

Arab American University

Faculty of Graduate Studies

Department of Natural, Engineering and Technology Sciences

Master Program in Data Science and Business Analytics



Hybrid Context-Aware Recommendation System

For Hypertension Problem

Mohammed Labeeb Maraqa

202112330

Supervision Committee:

Dr. Mohammed Hussien

Dr. Ibrahim Nimer

Dr. Amani Owda

Dr. Abdellatif Abu-Issa

**This Thesis Was Submitted in Partial Fulfilment
of the Requirements for the Master Degree in
Data Science & Business Analytics**

Palestine, 1/2026

Arab American University

Faculty of Graduate Studies

Department of Natural, Engineering and Technology Sciences

Master Program in Data Science and Business Analytics



Thesis Approval

**Hybrid Context-Aware Recommendation System
For Hypertension Problem**

Mohammed Labeeb Maraqa
202112330

This thesis was defended successfully on 12/1/2026 and approved by:

Thesis Committee Members:

Name	Title	Signature
1. Dr.Mohammed Hussien	Supervisor	<u>M.HUSSEIN</u>
2. Dr.Ibrahim Nimer	Co-supervisor	Nimer
3. Dr.Amani Owda	Internal Examiner	Amani Owda Signer ID: H0QPWOS817...
4. Dr.Abdellatif Abu-Issa	External Examiner	Abdellatif Abu Issa

Palestine, 1/2026

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

Student Name: Mohammed Labeeb Maraqa

Student ID: 202112330

Signature: 

Date of Submitting the Final Version of the Thesis: 10 / 02 / 2026

Dedication

With all love and gratitude, I would like to dedicate this to my beloved family and dear friends.

To my dear mother who has always been a light guiding me on my path with the warmth of her endless love and giving. Thank you for every moment of support and wisdom and care you have given me.

To my dear father who has always been the pillar and strength in my life. Thank you for your guidance that has shaped me and given me the confidence to face every challenge.

To my dear wife who is my life partner and companion, thank you for your support and backing.

To my daughters, Taghreed and Tanal, whose laughter brightens my days.
And to my little son, Fadi, whose smile reminds me that every effort is worth it.

To my dear brothers, you are the first friends who have accompanied me every step of the way. Thank you for all the beautiful moments we have shared and your continuous support.

To my loyal friends, you are the family I have chosen for myself. Thank you for your true friendship and standing by me in all times, whether good or bad.

Mohammed Labeeb Sabri Maraqa

Acknowledgements

In the name of Allah, the Most Gracious, the Most Merciful. All praise and thanks are to Allah, who granted me the strength and perseverance to complete this work. I would like to express my deepest gratitude to my main supervisor, Dr. Mohammed Hussien, and my co-supervisor, Dr. Ibrahim Nimer, for their invaluable guidance, insightful feedback, and continuous encouragement throughout every stage of this research. Their expertise and patience were instrumental in shaping this work into its final form.

My appreciation also goes to the distinguished members of my thesis committee for their time and feedback.

I would like to express my gratitude to the Arab American University for providing an outstanding academic environment and resources that enabled me to complete this research.

I would also like to acknowledge the Deanship of Graduate Studies for their academic support and resources.

To my colleagues in the program, and to my dear friends and family—your encouragement was essential in this journey. Thank you all.

Hybrid Context-Aware Recommendation System for Hypertension Problem

Mohammed Labeeb Sabri Maraqa

Supervision Committee:

Dr. Mohammed Hussien

Dr. Ibrahim Nimer

Dr. Amani Owda

Dr. Abdellatif Abu-Issa

Abstract

This research aims to develop an intelligent recommendation system based on artificial intelligence to provide personalized and data-driven recommendations for hypertension management. The proposed system, named Hybrid Context-Aware Recommendation System, integrates two complementary methodologies Collaborative Filtering and Content-Based Filtering within a unified scientific and practical framework.

The study was conducted using an applied analytical methodology that combined software design with experimental evaluation. The study population relied on real-world data from the Kaggle Health Dataset, which includes clinical, demographic, and behavioral attributes of adult individuals. The study sample was selected and preprocessed according to medical standards, incorporating blood pressure levels, cholesterol categories, cardiovascular risk levels, age, gender, and lifestyle factors.

Multiple research tools were employed, including data analysis, feature engineering, and context extraction techniques, in addition to performance evaluation metrics such as precision, recall, and ranking quality. The results revealed that the proposed hybrid model outperformed the standalone Collaborative Filtering and Content-Based Filtering models, achieving Precision 0.72 and Recall 0.90, indicating highly accurate and comprehensive recommendations. The system also demonstrated strong interpretability through an interactive dashboard built on the Streamlit platform, enhancing transparency and applicability in digital healthcare environments.

The study recommends adopting hybrid context-aware systems in clinical decision support due to their ability to combine personalization, reliability, and clinical interpretability. It further encourages expanding the dataset to include multi-institutional medical records and applying the same framework to other chronic diseases such as diabetes and cardiovascular disorders.

Keywords: Recommender Systems, Hypertension, Artificial Intelligence, Hybrid Filtering, Context Awareness.

Table of Contents

#	Title	Page
	Declaration	I
	Dedication	II
	Acknowledgements	III
	Abstract	IV
	List of Tables	IX
	List of Figures	X
	List of Appendices	XIV
	List of Definitions of Abbreviations	XV
	Chapter One: Introduction	1
1.1	Background	1
1.1.1	Overview	1
1.1.2	Challenges in Hypertension Management	3
1.1.3	Advances in Recommender Systems	3
1.2	Importance of the Study	4
1.2.1	Theoretical Significance	4
1.2.2	Practical Significance	5
1.2.3	Beneficiaries of the Research	6
1.3	Problem Statement	6
1.4	Objectives of the Study	7
1.5	Research Questions	7
1.6	Hypotheses of the Study	8
1.7	Definitions of Terms	8
1.8	Scope and Limitations of the Study	9
1.8.1	Conceptual Boundaries	9

1.8.2	Subject Boundaries	10
1.8.3	Temporal Boundaries	10
1.8.4	Spatial Boundaries	10
1.8.5	Methodological Limitations	11
1.8.6	Population and Sample	11
1.9	Thesis Structure	12
Chapter Two: Theoretical Framework and Literature Review		13
2.1	Introduction	13
2.2	Theoretical Framework	14
2.2.1	Recommender Systems in Healthcare	14
2.2.2	Content-Based Filtering	15
2.2.3	Collaborative Filtering	15
2.2.4	Hybrid Recommender Systems	17
2.2.5	Other Methods	18
2.2.6	Similarity Models	19
2.2.7	Classification Techniques	21
2.2.8	Feedback Mechanisms and Contextual Data Integration	24
2.2.9	Proactive Recommendations	25
2.3	Previous Studies	27
2.3.1	Healthcare Recommender Systems	27
2.3.2	Context-Aware Recommender Systems	29
2.3.3	Hybrid Recommender Systems	31
2.3.4	Studies Specific to Hypertension Management	32
2.4	Critique and Analysis of Previous Studies	34
2.4.1	Challenges in Current Systems	34
2.4.2	Evaluation of Existing Systems	36
2.4.3	Principal Properties of Recommender Systems	36
2.4.4	Equations and Metrics	37
2.4.5	Gap Analysis and Research Opportunities	38
2.4.6	Comments on Previous Studies	39
2.5	Conceptual Framework	41
2.5.1	Advancing Hypertension Management with a Hybrid Context-Aware Recommender System	41
Chapter Three: Research Methodology		43
3.1	Introduction	43
3.2	Research Design	44
3.2.1	Methodological Approach	44

3.2.2	Justification for the Selected Methodology	44
3.2.3	Population & Sample	45
3.3	Dataset Description	45
3.3.1	Kaggle Health Dataset	45
3.3.2	Dataset Footprint and Processing Stages	46
3.3.3	Inclusion and Exclusion Criteria	47
3.4	Data Preprocessing and Preparation	47
3.4.1	Data Cleaning	48
3.4.2	Data Transformation	48
3.4.3	Feature Engineering	49
3.5	Pipeline Overview and System Modules	53
3.5.1	Stage 1: Data Preparation	53
3.5.2	Stage 2: Context Feature Engineering	53
3.5.3	Stage 3: Recommendation Pipelines (CBF, CF, Hybrid)	53
3.5.4	Stage 4: Evaluation Export and Visualization	54
3.6	Development of the Hybrid Context-Aware Recommender System	57
3.6.1	System Overview	57
3.6.2	Implementation of Recommendation Techniques	59
3.6.3	Hybrid Model Integration	75
3.7	Model evaluation and validation	85
3.7.1	Evaluation protocol and experimental setup	85
3.7.2	Evaluation mechanics and artefact generation	86
3.7.3	Evaluation metrics	87
3.7.4	Validation and reproducibility assurance	88
3.8	Study procedures	89
3.8.1	Implementation workflow	89
3.8.2	Data Analysis Procedures	91
3.8.3	Data Visualization Techniques	91
3.9	Configuration and Reproducibility Artefacts	92
3.9.1	Configuration Management and Reproducibility	92
	Chapter Four: Results	94
4.1	Overview of Experimental Outcomes	94
4.2	Preprocessing and Contextual Data Results	94
4.2.1	Contextual Feature Engineering Results	97
4.3	Performance of Individual Pipelines	100
4.3.1	Collaborative Filtering Results	100
4.3.2	Content-Based Filtering Results	106
4.3.3	Hybrid Model Performance	111

4.4	Comparative Evaluation and Benchmarking	117
4.4.1	Cross-Model Metric Comparison	117
4.4.2	Visual Benchmarking and Analysis	118
4.5	Clinical Interpretability & Recommendation Insights	120
4.6	System Interface and User Interaction	123
4.6.1	Main Framework Overview	123
4.6.2	Home Page and Navigation	123
4.6.3	Preprocess Module	123
4.6.4	Context Module	124
4.6.5	CF Module	124
4.6.6	CBF Module	124
4.6.7	Hybrid Module	124
4.6.8	Comparative Visualization	124
4.6.9	Smart Visualization Hub	124
4.6.10	Design and Interaction Philosophy	125
	Chapter Five: Discussion, conclusion, & recommendation	126
5.1	Interpretation of Results	126
5.2	Comparison with Previous Studies	128
5.3	Practical Implications	130
5.4	Limitations of the Study	131
5.5	Recommendations for Future Research	133
5.6	Concluding Remarks	134
	References	136
	Appendices	146
	الملخص	158

List of Tables

Table #	Title of Table	Page
Table 2.1	Examples of Explicit and Implicit Feedback in Healthcare RS	26
Table 3.1	Feature Description of the Hypertension Dataset	45
Table 3.2	Dataset footprint across preprocessing and contextualisation.	46
Table 3.3	Derived Contextual Features in HCARS	51
Table 3.4	End-to-end dataflow summary across stages and modules	55
Table 3.5	CF Logic and File Dependencies in HCARS	66
Table 3.6	CBF Logic and File Dependencies in HCARS	74
Table 3.7	Hybrid Logic and File Dependencies in HCARS	84
Table 4.1	Category counts for contextual features	97
Table 4.2	CF performance across evaluation scenarios (mean $\pm$ std).	101
Table 4.3	Mapping of compact reason identifiers (R1–R9) to the original reason descriptions (top reasons in CF).	104
Table 4.4	CBF performance across evaluation scenarios (mean $\pm$ std).	106
Table 4.5	Hybrid performance $k=5$ (pre-filter, pre-top- k metrics).	112
Table 4.6	Comparative evaluation models at $k=5$ (eval split).	117
Table (1).1	Comprehensive Summary of Recommender Systems	146

List of Figures

Figure #	Title of Figure	Page
Fig 1.1	Illustration information overload challenges (Adapted from Waterson, 2005).	2
Fig 2.1	Illustration of kNN with $k=3$ and $k=5$ the new point (green) takes the majority label of its nearest neighbors.	23
Fig 3.1	High-level architecture of the HCARS.	58
Fig 3.2	CF pipeline structure and module interaction in HCARS: contextualized data flow through configuration, cold-start handling, and output visualization.	65
Fig 3.3	CBF pipeline structure and module interaction in HCARS: patient item vector generation, similarity computation, calibration, and evaluation.	74
Fig 3.4	Hybrid pipeline structure and module interaction in HCARS: calibrated CF/CBF outputs, cold-start flags, and CBF fallback variants are fused into the final hybrid ranking and evaluation.	84
Fig 4.1	Raw dataset: missing-value proportions for the top-10 variables; only the sex indicator has minor missingness ($\approx 0.10\%$).	95
Fig 4.2	Overlaid histograms of resting BP before and after cleaning; the central distribution is stable, with only a few extreme high values removed.	95
Fig 4.3	Overlaid histograms of serum cholesterol before and after cleaning; removing implausibly high outliers slightly shifts the right tail while preserving the clinically plausible range.	96

Fig 4.4	Overlaid histograms of age before and after cleaning; the near-perfect overlap confirms demographic stability of the analytical cohort.	96
Fig 4.5	Correlation heatmap for numeric variables in the cleaned dataset.	97
Fig 4.6	BP category distribution in the contextualised dataset; hypertensive states (Elevated, Stage 1, Stage 2) dominate, with Normal and Hypertensive Crisis.	98
Fig 4.7	Cholesterol category distribution; over half of patients are high, about one third are borderline high, and a small minority are desirable.	99
Fig 4.8	Risk-level composition from combined vitals; the cohort is concentrated in the medium and high risk bands, with a negligible low-risk group.	99
Fig 4.9	Age-group structure of the contextualised dataset; most patients are 46–75 years, with very few below 30.	99
Fig 4.10	CF performance at $k=5$ across eval, LOCO, and family splits; metrics are strongest in eval, slightly reduced in LOCO, and drop with higher variability under cross-family generalisation.	101
Fig 4.11	CF calibration curve (predicted vs. observed success); low-probability bins sit above diagonal (slight under-confidence), while bins near 0.7–0.75 align ideal line.	102
Fig 4.12	Precision–Recall curve across thresholds; operating points show very high recall with precision around 0.7–0.9, indicating CF preserves most true positives while maintaining good specificity.	102
Fig 4.13	Top-N CF interventions by cumulative score (score_sum); lifestyle actions dominate, followed by guideline-consistent antihypertensive drugs.	103
Fig 4.14	Top CF reason categories (R1–R9) with bar counts; full descriptions are in Table 4.3.	103

Fig 4.15	Top-N interventions recommended by CF, ranked by cumulative score (score_sum); lifestyle actions and guideline-consistent antihypertensive drugs (e.g. amlodipine, ace_inhibitor) dominate.	105
Fig 4.16	Hit-rate@5 by bp_category; Elevated and Stage 1 have the highest hit-rates, with only slight drops in Stage 2 and Normal, indicating broadly consistent performance across BP strata.	105
Fig 4.17	CBF metrics at $k=5$ (Precision@5, Recall@5, MAP@5, nDCG@5) across eval, LOCO, and family splits.	107
Fig 4.18	Distribution of recommendation generation reasons (CBF module).	107
Fig 4.19	Distribution of CBF recommendation scores (score_cbf).	108
Fig 4.20	CBF calibration curve: predicted score vs observed success rate.	108
Fig 4.21	Histogram of CBF scores (score_cbf) over all recommendations.	109
Fig 4.22	Distribution of true response values (true_response).	110
Fig 4.23	Precision–Recall relationship for CBF recommendations as the score threshold varies.	110
Fig 4.24	Distribution of hybrid recommendation scores (hybrid_score) across all patients and interventions.	112
Fig 4.25	Distribution of confidence scores in the hybrid recommender.	113
Fig 4.26	Provenance of hybrid recommendations across CF and CBF sources.	113
Fig 4.27	Distribution of recommendation types in the hybrid system (pharmacologic, lifestyle, lab orders, monitoring).	114
Fig 4.28	Density of CF and CBF contribution weights in the hybrid scoring rule.	115
Fig 4.29	Word cloud of hybrid recommendation text (advice_text).	115

Fig 4.30	Observed success rate across advice-text length bins in hybrid recommendations.	116
Fig 4.31	Cross-model comparison of evaluation metrics at $k=5$ (eval split).	118
Fig 4.32	Radar visualization comparing CF, CBF, and Hybrid performance at $k=5$.	119
Fig 4.33	Representative score distributions for CF, CBF, and Hybrid models.	119
Fig (2).1	Home page: HCARS overview and navigation hints.	153
Fig (2).2	Preprocess Before vs After: side-by-side comparison panel.	154
Fig (2).3	Preprocess module: cleaned data table (df_cleaned.csv) and exports.	154
Fig (2).4	Context explorer: histogram of risk_score and top counters.	155
Fig (2).5	CF dashboard: overview KPIs and score distribution.	155
Fig (2).6	CBF dashboard: evaluation files overview, metrics, and detailed rows.	156
Fig (2).7	Display Pages (comparison canvas): CBF reason distribution vs. Hybrid score distribution.	156
Fig (2).8	Smart Visualization Center.	157
Fig (2).9	Comparison View in Recommendation (CF/CBF/Hybrid).	157

List of Appendices

Appendix #	Title of Appendix	Page
Appendix (1): Comprehensive Summary of Recommender Systems		146
Appendix (2): HCARS Interface Screens		153

List of Definitions of Abbreviations

Abbreviations	Title
AI	Artificial Intelligence
BP	Blood Pressure
CBF	Content-Based Filtering
CARS	Context-Aware Recommender Systems
CF	Collaborative Filtering
HCARS	Hybrid Context-Aware Recommendation System
HRS	Healthcare Recommender Systems
kNN	k-Nearest Neighbors
KHD	Kaggle Health Dataset
ML	Machine Learning
RL	Reinforcement Learning
RS	Recommender Systems
MLOps	Machine Learning Operations
WD	Wearable Device
DNNs	Deep Neural Networks

Chapter One: Introduction

1.1 Background

1.1.1 Overview

Imagine a scenario where a condition silently impacts over a billion individuals, often going unnoticed until severe consequences arise. This is the case with hypertension, a chronic condition characterized by elevated blood pressure (BP), which substantially raises the risk of critical cardiovascular events such as heart attacks and strokes. In a world where the Internet has become integral to everyday life, the potential of digital innovations to address this global health challenge is increasingly apparent.

By 2023, an estimated 66% of the world's population had access to the Internet, a notable increase from just 16% in 2005 (Union, 2023). This rapid expansion, alongside advancements in data collection and analysis technologies, has driven the emergence of Big Data. These extensive and intricate datasets are derived from sources such as mobile devices, wearable technologies, and online platforms. The healthcare industry, in particular, has seen a remarkable escalation in data generation. Recent forecasts from the IDC Global DataSphere report suggest that the volume of digital data created worldwide will grow from about 33 zettabytes in 2018 to approximately 221 zettabytes by 2026, with healthcare consistently cited as one of the fastest-growing and most data-intensive sectors (International Data Corporation, 2022).

This vast amount of data offers tremendous potential for advancing patient care, supporting personalized medicine, and enhancing strategies for disease management. However, the sheer scale and complexity of such information present significant challenges in terms of analysis, interpretation, and practical application, often referred to as information overload. Recent work on electronic health records navigation has shown that even senior medical students and clinicians can feel overwhelmed by fragmented documentation and information overload when retrieving clinically relevant data from routine records (Ramkumar et al., 2025). Both patients and healthcare professionals often find it difficult to draw meaningful insights from the vast amount of available data, making effective decision-making more challenging. Figure 1.1 illustrates how an excess of information, even on a simple decision like selecting gum, can complicate choices and diminish the overall experience.

Hypertension highlights the urgent need for optimized data utilization. This chronic medical condition, defined by persistently elevated arterial BP, affects approximately 1.28 billion adults aged 30–79 globally, with two-thirds of cases found in low- and middle-income countries (Organization, 2025). Often termed the “silent killer,” hypertension typ-

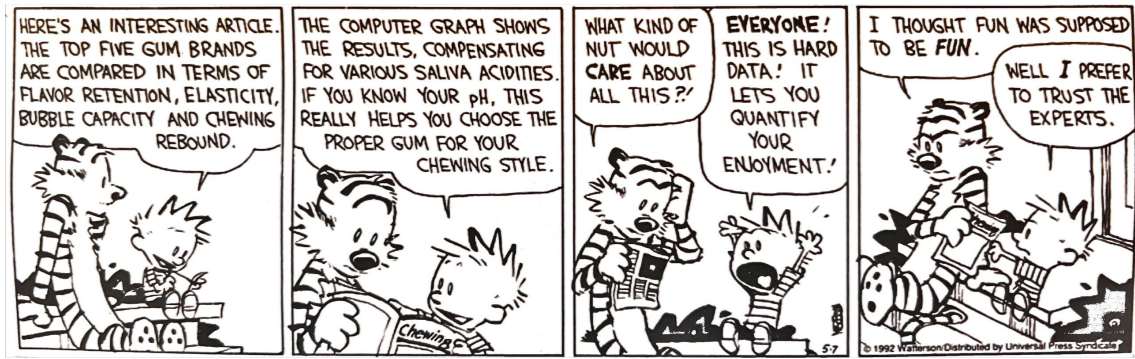


Figure 1.1: Illustration information overload challenges (Adapted from Watterson, 2005).

ically presents without noticeable symptoms, while chronically elevated blood pressure substantially increases the risk of cardiovascular disease, stroke, and chronic kidney disease (Nawata, 2023). Its asymptomatic nature contributes to widespread underdiagnosis and inadequate treatment, exacerbating its burden on global health systems.

BP depends on the amount of blood ejected by the heart and the level of resistance encountered as it flows through the arteries. Hypertension occurs when the force of blood against arterial walls remains consistently high, causing damage over time. Contributing factors include genetic predispositions, aging, ethnicity, and lifestyle habits such as poor dietary choices, lack of physical activity, excessive alcohol intake, and tobacco use (Yashvanth et al., 2023). Hypertension is broadly categorized into primary (essential) hypertension, which has no clearly identifiable medical cause, and secondary hypertension, which results from underlying conditions such as kidney disorders or hormonal imbalances.

Conventional approaches to managing hypertension often fail to account for the intricate and individualized aspects of each patient's condition, including their genetic background, lifestyle, environmental influences, and coexisting health conditions. Such a one-size-fits-all approach often results in less effective health outcomes and higher healthcare costs. Personalized and context-sensitive interventions, however, hold promise in improving adherence to treatment regimens and overall patient well-being.

Recommender systems (RS), which have shown remarkable success in domains such as e-commerce, online entertainment, and social media, provide tailored suggestions based on user preferences and behaviors (Fkih & Rhouma, 2023). Large-scale streaming and online retail platforms routinely deploy such machine-learning-based recommenders to increase user engagement and satisfaction. Applying these systems in healthcare, particularly to manage complex conditions like hypertension, opens up promising opportunities for customized patient care.

Despite their potential, existing healthcare recommender systems (HRS) often fall short in integrating comprehensive contextual information essential for addressing the multifaceted nature of hypertension management (Zitouni et al., 2022). The inherent complexity of medical data, combined with the demand for evidence-based, patient-specific

recommendations, presents unique challenges that traditional recommendation methods may struggle to address. Bridging these gaps necessitates the development of Hybrid Context-Aware Recommender Systems (HCARS) capable of synthesizing diverse data sources while accounting for the full spectrum of patient-specific factors.

Through advanced data analytics and machine learning (ML) techniques, such systems can process intricate datasets to deliver highly personalized recommendations. This not only aids patients and healthcare providers in making informed decisions but also mitigates the challenges posed by information overload (Torres-Ruiz et al., 2023). Successfully implementing context-aware recommender systems (CARS) for hypertension management could transform patient engagement and adherence, ultimately leading to improved health outcomes. Such advancements mark a pivotal step toward precision medicine, where interventions are specifically designed to meet individual needs rather than generalized for broader populations.

1.1.2 Challenges in Hypertension Management

Effective management of hypertension is a complex endeavor due to its multifactorial etiology and the interplay of various genetic, environmental, and behavioral factors (Lu et al., 2024). Traditional healthcare systems primarily rely on static Electronic Health Record (EHR) data and generalized treatment guidelines, which may not adequately address the unique needs of each patient (Peng & et al., 2025). This one-size-fits-all approach often ignores how individuals differ in disease progression, reaction to medication, and daily lifestyle habits.

The adherence of patients to prescribed treatment regimens remains a significant challenge. Factors such as medication side effects, complex dosing schedules, lack of symptoms, and insufficient patient education contribute to poor compliance (Luo et al., 2021). Additionally, socioeconomic factors, including limited access to healthcare resources and support systems, further impede effective management.

Healthcare providers face limitations in resources and tools necessary for delivering personalized care. Time constraints during consultations, inadequate patient monitoring, and insufficient integration of patient data hinder the ability to make timely and informed decisions (Zuin & et al., 2025). The dynamic nature of hypertension necessitates continuous monitoring and adjustment of treatment plans, which traditional systems are ill-equipped to handle.

1.1.3 Advances in Recommender Systems

The RS have become a core component of many digital platforms, providing personalized suggestions by learning from user preferences, behaviours, and, in more advanced settings, contextual information (Anwar et al., 2022). In healthcare, these systems are increasingly

explored as decision-support tools that can tailor treatment options, educational content, and lifestyle guidance to individual patient profiles, thereby aligning with the broader shift toward data-driven, personalized medicine (Yashvanth et al., 2023).

Within this broader landscape, hybrid and context-aware designs are particularly relevant for chronic conditions such as hypertension, where risk profiles and management strategies vary substantially across patients. The proposed HCARS in this thesis builds conceptually on these advances by adopting a hybrid, context-informed perspective tailored to hypertension management. A more detailed discussion of HRS, hybrid architectures, and context-aware approaches is provided in Chapter 2.

1.2 Importance of the Study

1.2.1 Theoretical Significance

The development of an HCARS for hypertension management holds substantial theoretical significance in the fields of data science and healthcare informatics. This research contributes to the academic body of knowledge by formalizing a hybrid, context-informed recommender framework that is explicitly tailored to the complexities of chronic disease management. By combining Collaborative Filtering (CF), Content-Based Filtering (CBF), and context-aware computing within a single conceptual architecture, the study advances existing theoretical models of HRS.

A key theoretical contribution lies in the explicit treatment of context as a structured, clinically meaningful component of the recommendation process. Whereas many traditional RS designs operate on static user-item interactions, the proposed framework incorporates contextual dimensions derived from patient characteristics and health status, thereby clarifying how context can be encoded and exploited even in settings that lack continuous sensor data. This adds to the theoretical understanding of how context-aware mechanisms can be instantiated in constrained, tabular-data environments relevant to many clinical applications.

Furthermore, the study explores the role of hybrid models in healthcare settings, an area that remains relatively underdeveloped in the academic literature. By integrating demographic attributes, clinical indicators, lifestyle-related factors, and derived risk features into a unified recommendation logic, the research contributes to theoretical discussions on multi-source data integration and personalized medicine. It provides a structured view of how heterogeneous features can be combined to support both risk estimation and recommendation ranking within a single framework.

The use of the Kaggle Health Dataset (KHD) for developing and validating the HCARS also has theoretical implications. It demonstrates how publicly available, structured health datasets can serve as a basis for constructing and evaluating hybrid, context-informed recommender models, thereby encouraging reproducible research and

more transparent methodological comparisons across studies. The modelling choices and evaluation protocol presented in this work can serve as a reference point for future research that seeks to develop or benchmark HRS using similar datasets.

1.2.2 Practical Significance

From a practical standpoint, the proposed HCARS has the potential to enhance hypertension management by providing personalized and context-aware recommendations for patients and healthcare providers in a controlled, research oriented setting. Hypertension is a prevalent condition that requires continuous monitoring and individualized care plans, and conventional approaches often struggle to account for patient level heterogeneity in a systematic, data driven way. The system developed in this research addresses some of these limitations by generating recommendations that are adapted to patient characteristics and clinically relevant contextual factors, while remaining explicitly framed as a decision support prototype rather than a deployed clinical tool.

The practical implications of this study include:

- **Enhanced data analysis for patient care** by providing personalized recommendations based on systematic analysis of structured health data, the system can reveal patterns and risk profiles that are not immediately apparent in routine practice, thereby supporting more informed treatment planning and lifestyle modification strategies.
- **Support for healthcare providers and researchers** clinicians and researchers can utilize the HCARS as a research instrument for exploring individualized care strategies, testing hypothesis-driven scenarios, and analysing subgroups of patients with similar risk profiles, particularly in retrospective or educational settings.
- **Resource for educational purposes** the system serves as a concrete example of how ML and data analytics can be applied to healthcare data within a reproducible framework, offering students and professionals in healthcare informatics and data science a practical case study that links theoretical concepts to applied modelling.
- **Advancement of precision medicine** by integrating demographic, clinical, and lifestyle-related features into a unified recommendation logic, the HCARS aligns with the broader movement toward precision medicine, in which treatments and interventions are progressively tailored to individual patients rather than based solely on population-level averages.

The practical significance of the proposed framework extends beyond hypertension management. The methodologies and modelling strategies can be adapted to other chronic

diseases that rely on structured, tabular health data, thereby contributing to more systematic, data-driven approaches to chronic disease management and improved patient outcomes in diverse clinical domains.

1.2.3 Beneficiaries of the Research

The research has the potential to benefit a wide range of stakeholders:

- **Researchers and educators** the study provides valuable insights and methodologies for further research and educational purposes, demonstrating how to implement an HCARS within a modular Python and Streamlit-based research environment.
- **Healthcare data scientists** professionals working in healthcare data analysis can utilize the findings to enhance their approaches to developing personalized recommendation systems.
- **Patients and healthcare providers (Indirectly)** while the system is not deployed in a clinical setting, the insights gained can inform future developments that may eventually benefit patients and healthcare providers.

Having established the importance of the study, the structure of the thesis is presented to provide a clear roadmap for the ensuing chapters.

1.3 Problem Statement

Hypertension continues to be a significant public health concern globally, contributing substantially to the burden of cardiovascular diseases and associated mortality. Effective management of hypertension is complex due to its multifactorial nature, where genetic predispositions, environmental influences, and individual lifestyle choices intersect to influence disease progression and treatment outcomes. Traditional healthcare systems, which often rely on static data from EHRs and generalized treatment protocols, are insufficient in addressing the personalized needs of patients with hypertension. This lack of individualized care can lead to suboptimal treatment efficacy, poor patient adherence, and increased risk of complications.

Existing HRS have attempted to enhance patient care by utilizing ML algorithms and patient-specific data to provide more accurate recommendations. However, these systems face critical challenges such as the cold-start problem where insufficient data about new patients leads to less personalized recommendations scalability limitations when dealing with large and complex datasets, and the inability to incorporate comprehensive, context-aware data reflecting patients' dynamic health statuses. These limitations hinder the effectiveness of current systems in managing chronic conditions like hypertension.

The core problem addressed in this research is the need to develop and demonstrate an HCARS specifically designed for hypertension management within a computational environment. By integrating diverse data sources available in the KHD, including patient demographics, medical history, lifestyle data, and health metrics, and using advanced ML techniques, the system aims to provide personalized and context-sensitive recommendations that adapt to the varying health profiles of individual patients. Leveraging modular Python pipelines, configuration-driven execution, and a research-oriented Streamlit dashboard, the study seeks to overcome the limitations of existing systems and improve hypertension management outcomes through robust data analysis, model development, and visualization.

1.4 Objectives of the Study

The primary objective of this research is to design and demonstrate an HCARS for the management of hypertension using modular Python pipelines, configuration-driven execution, and ML techniques within a controlled computational environment. The specific objectives are as follows:

1. Identify key factors that influence hypertension by analyzing the KHD.
2. Develop a hybrid recommendation model that integrates CF, CBF, and ML algorithms to analyze complex health data using modular Python scripts and pipelines.
3. Integrate contextual information, such as patient demographics, lifestyle factors, and health metrics, to enhance the accuracy and relevance of recommendations.
4. Evaluate the effectiveness of the developed system using offline ranking metrics (such as Precision@ k , Recall@ k , MAP@ k , and nDCG@ k), where k denotes the length of the recommended list (for example, the top- k items suggested to a patient), computed on persisted CSV and JSON outputs and summarized through visual analytics within the computational environment.
5. Provide a methodological framework that can inform future developments in personalized healthcare technologies for hypertension management.

1.5 Research Questions

This study seeks to answer the following research questions:

1. What specific challenges in hypertension management can be addressed by employing an HCARS within a computational environment?

2. How can a hybrid approach effectively integrate diverse data sources such as patient demographics, lifestyle factors, and health metrics with ML models to deliver personalized hypertension recommendations within a reproducible, modular Python-based computational environment?
3. How can existing recommendation models be adapted to improve their effectiveness in managing hypertension through data analysis and visualization?
4. How can patient-specific health data and contextual information be leveraged within a computational framework to improve the accuracy and relevance of hypertension management recommendations?

1.6 Hypotheses of the Study

Based on the research questions and objectives, the study formulates the following hypotheses:

1. **H₁** Developing an HCARS using modular Python pipelines and a research-oriented Streamlit interface can demonstrate potential improvements in the personalization and effectiveness of hypertension management recommendations compared to traditional methods.
2. **H₂** Integrating diverse data sources with advanced ML models within a computational environment enhances the accuracy of hypertension recommendations.
3. **H₃** Incorporating contextual information leads to more relevant recommendations in hypertension care within the scope of the data analysis.

1.7 Definitions of Terms

For clarity and consistency, key terms used throughout this thesis are defined as follows:

- **Hypertension** a chronic medical condition characterized by persistently elevated BP levels, increasing the risk of cardiovascular diseases, stroke, and kidney failure.
- **Healthcare Recommender System (HRS)** a system that provides personalized healthcare suggestions or recommendations to patients and healthcare providers based on various data inputs and algorithms.
- **Hybrid Context-Aware Recommendation System (HCARS)** a recommendation system that combines multiple recommendation techniques, such as CF and CBF, and incorporates contextual information to improve performance and accuracy.

- **Context-Aware computing** an approach in computing where systems can sense and respond to contextual information such as user attributes, time, and activity to provide relevant services or recommendations.
- **Cold-Start Problem** a challenge in recommendation systems where insufficient data about new users or items leads to less accurate or less personalized recommendations.
- **Machine Learning** a subset of artificial intelligence (AI) involving algorithms and statistical models that enable computers to perform tasks without explicit instructions, by relying on patterns and inference.
- **Jupyter Notebook** an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text.
- **Streamlit** an open-source Python framework for building interactive, web-based data applications and dashboards, used in this study as a read-only interface to visualize system outputs.
- **Data Visualization** the graphical representation of data and information using visual elements like charts, graphs, and maps to help understand trends, patterns, and outliers.
- **Kaggle health dataset** a publicly available dataset on Kaggle that includes health-related data such as patient demographics, medical history, and health metrics, used for data analysis and ML applications.

1.8 Scope and Limitations of the Study

1.8.1 Conceptual Boundaries

The conceptual scope of this study is grounded in the fields of data science, ML, and healthcare informatics, specifically focusing on the development and demonstration of an HCARS for hypertension management within a secure, locally controlled Python environment orchestrated by modular pipelines and configuration files. The research integrates concepts from RS theories, context-aware computing, and hybrid modeling techniques. The study concentrates on computational methods for enhancing personalized healthcare recommendations through data analysis, model development, and visualization.

The research explores how ML algorithms can be applied to healthcare data to improve recommendation systems, addressing challenges such as the cold-start problem within the scope of the dataset. It examines the theoretical underpinnings of hybrid recommendation models and context-awareness in computational systems, contributing to the academic discourse in these areas.

1.8.2 Subject Boundaries

The subject boundary of this study is confined to hypertension management within the healthcare domain, specifically within the context of data analysis and model development in a modular Python and Streamlit-based research environment. The research targets the development of a recommendation system that aids in managing hypertension by providing personalized recommendations based on patient data. Other chronic diseases or medical conditions are beyond the scope of this study unless they directly relate to hypertension, such as comorbidities like diabetes and stroke.

The study focuses on data analysis and ML techniques applied to the health datasets obtained from Kaggle. The emphasis is on the underlying algorithms and data processing methods rather than on the deployment or integration of the system into clinical practice or the deployment of a production-grade clinical application. A lightweight Streamlit interface is used solely as a research-oriented, read-only dashboard to visualize system outputs, not as an operational tool in real healthcare settings.

1.8.3 Temporal Boundaries

The temporal scope of the study is limited to the data available within the KHD at the time of the research. The dataset comprises historical patient data up to the point of its last update. The study does not incorporate real-time data collection or analysis from wearable devices (WDs) or continuous monitoring systems. Simulations of contextual data adaptations are based on the available variables within the existing dataset.

The research does not account for future developments in healthcare technologies or changes in hypertension management guidelines that may occur after the completion of the study. The findings and recommendations are therefore reflective of the data and technologies available during the research period.

1.8.4 Spatial Boundaries

The spatial boundaries of the study are defined by the geographical representation of the data within the KHD. The dataset includes patient information from various regions but does not specify detailed geographical locations. As such, the research assumes a generalized application of the findings across different populations without focusing on region-specific factors.

The study does not address geographical variations in hypertension prevalence, healthcare infrastructure, or regional treatment practices. It aims to develop a recommendation system that can be adapted for use in diverse settings but does not customize the system for any particular geographical area.

1.8.5 Methodological Limitations

Several methodological limitations are acknowledged in this study:

- **Data completeness** the KHD, while comprehensive, may lack certain variables that influence hypertension outcomes, such as genetic markers or detailed environmental data. The absence of these variables may limit the depth of analysis and the granularity of the recommendations.
- **Simulation of contextual adaptation** the study relies on historical data and does not incorporate live data streams. Consequently, the system's ability to provide context-aware recommendations is demonstrated through offline data analysis, simulations on persisted CSV and JSON outputs, and visualizations generated within the research dashboard.
- **Absence of clinical validation** the study does not include clinical trials or patient interactions to validate the effectiveness of the recommendations in real-world settings. The evaluation is limited to computational metrics and data-driven analysis within the computational environment.

These methodological limitations are recognized, and steps are taken to address them within the constraints of the study. However, they may impact the applicability and scalability of the system beyond the scope of this research.

1.8.6 Population and Sample

The population considered in this study comprises the patients represented in the KHD provided by Prosper Chuks (Chuks, 2022). The dataset includes health records of individuals with information on demographics, medical history, and health metrics relevant to hypertension, diabetes, and stroke.

The sample used for analysis consists of all patients within the dataset who have complete records necessary for the development and validation of the ML models. Inclusion criteria are established to ensure that the data utilized are sufficient for meaningful analysis, such as the presence of key variables like BP readings, age, gender, and medical history.

The study does not involve any primary data collection or interaction with human subjects. All analysis are conducted on secondary data obtained from the publicly available dataset, ensuring compliance with ethical standards regarding data privacy and confidentiality.

1.9 Thesis Structure

The thesis is organized into five chapters, each aligned with the research objectives and demonstrating the development of the HCARS for hypertension management through modular Python pipelines and a Streamlit interface:

1. **Chapter One: Introduction** Outlines the research background, motivation, objectives, problem statement, significance, scope, and limitations. It defines the key research questions and hypotheses and provides a roadmap for the thesis.
2. **Chapter Two: Literature Review** Examines existing work on recommender systems (CF, CBF, Hybrid), context-aware modeling, and ML in healthcare. It analyzes theoretical foundations and previous studies, identifying research gaps that HCARS aims to address.
3. **Chapter Three: Research Methodology** Describes the quantitative and computational approach, dataset selection, preprocessing, contextual feature engineering, and modular pipeline implementation. It details the evaluation metrics (Precision@k, Recall@k, MAP@k, nDCG@k), configuration control, and ethical compliance in using open medical data.
4. **Chapter Four: System Implementation and Results** Presents the HCARS architecture, codebase integration, and Streamlit interface. It details the execution of CF, CBF, and Hybrid pipelines, showing data flow, evaluation results, and visualizations generated from the modular Python environment.
5. **Chapter Five: Discussion and Conclusions** Interprets the results in relation to the research objectives and prior studies, highlighting contributions, limitations, and practical implications. It concludes with recommendations for future work and potential extensions in clinical RS

Chapter Two: Theoretical Framework and Literature Review

2.1 Introduction

The intersection of healthcare and advanced computational technologies has ushered in a new era of personalized medicine, where patient care is increasingly tailored to individual needs. RS, a class of intelligent algorithms traditionally used in domains such as e-commerce and entertainment, have emerged as powerful tools in healthcare for delivering personalized recommendations. This chapter presents a comprehensive literature review and theoretical framework for developing HCARS specifically designed for hypertension management within a controlled, offline computational environment.

Hypertension is a complex, multifactorial condition influenced by a myriad of genetic, environmental, and lifestyle factors. The traditional one-size-fits-all approach in hypertension management often fails to account for the individual variability among patients, leading to suboptimal outcomes. The need for personalized interventions is therefore paramount, and RS offer a promising solution by leveraging patient-specific data to provide tailored recommendations.

This chapter systematically explores the existing body of knowledge on HRS, emphasizing their methodologies, applications, and limitations in the context of hypertension and chronic disease management. We delve into the foundational theories of RS, including CBF, CF, and hybrid models, highlighting how these techniques can be adapted and enhanced for healthcare applications. The integration of context-aware computing into these systems is examined, showcasing its potential to improve recommendation relevance by considering richer contextual information such as patient environment, activity levels, and physiological metrics as reported in the broader literature.

We also investigate the critical role of similarity models and classification techniques in the development of effective RS. Understanding how these models identify patterns and relationships within complex health data is essential for improving recommendation accuracy and reliability. The chapter discusses various ML algorithms, including Support Vector Machines, k -Nearest Neighbors, and clustering methods, and their applicability in predicting hypertension risks and treatment outcomes.

Furthermore, the chapter addresses the challenges faced by current HRS, such as the cold-start problem, data sparsity, scalability issues, and privacy concerns. Through a critical analysis of recent studies, we identify gaps in the literature and underscore the necessity for systems that can, in principle, integrate diverse data sources including genetic information, environmental factors, and lifestyle habits and adapt to the dynamic nature of patient health statuses. In this thesis, these ideas are instantiated in a more constrained

yet realistic setting by operating on the variables available in the KHD, which primarily captures clinical and lifestyle-related information rather than genomic or high-resolution environmental data.

By synthesizing existing research and theoretical concepts, this chapter establishes a robust framework for the proposed HCARS. The theoretical underpinnings discussed herein not only guide the design and implementation of a batch-oriented system implemented through modular Python pipelines and a research-oriented, read-only Streamlit dashboard, but also contribute to advancing the field of personalized healthcare technology. This endeavor aims to enhance hypertension management by providing clinicians and patients (indirectly, at the research level) with precise, contextually informed recommendations that ultimately support improved patient outcomes and quality of life.

2.2 Theoretical Framework

2.2.1 Recommender Systems in Healthcare

HRS are specialized software tools designed to provide personalized health-related suggestions. They have become increasingly important in modern healthcare, helping patients and professionals manage health conditions more effectively by tailoring recommendations to individual needs (Barbaric et al., 2025; Sun et al., 2023). In hypertension management, HRS can recommend lifestyle changes, medications, or monitoring strategies based on individual patient data. These systems focus on health-specific domains, offering tailored recommendations that are particularly valuable for patients who may lack detailed knowledge about their conditions. Key purposes of HRS include:

- Enhancing patient engagement and adherence by providing personalized health management strategies.
- Improving health outcomes by offering precise and timely recommendations.
- Assisting healthcare providers in understanding patient needs and preferences, leading to more effective treatment plans.

Implementing HRS typically involves techniques such as data mining, ML, and context-aware computing, often resulting in hybrid systems that combine multiple methodologies (Sun et al., 2023; X. Zhang et al., 2023). In this thesis, these principles are instantiated within a controlled, offline computational environment through modular Python pipelines operating on the KHD.

2.2.2 Content-Based Filtering

CBF involves describing both the items and the user profiles with a set of attributes, with the recommendation process focusing on the similarity between these attributes and the user's profile (Fkih & Rhouma, 2023). This approach is lauded for its simplicity and versatility, and it has found applications across various domains such as recommending websites, music, movies, books, restaurants, and hotels.

The construction of a CBF system typically entails several critical steps:

1. Building the item profile with attributes that describe the recommended items.
2. Creating the user profile utilizing the same set of attributes.
3. Developing a measure to compute the similarity between the user and the item.
4. Ranking the items according to their similarity to the user profile and recommending them accordingly.
5. Refining the user profile based on feedback to enhance future recommendations.

CBF's methodological foundation on item content alone offers notable advantages such as scalability and the capacity to recommend new items effectively by evaluating their similarity to the user's profile. It is also relatively transparent, which can enhance user trust by allowing users and clinicians to understand why specific recommendations were generated.

However, recommendations may become overly specialized, limiting diversity in the suggestions made to the user. Moreover, constructing detailed item profiles can be labor-intensive, particularly when capturing nuanced attributes is necessary for accurate recommendations. These aspects underscore the complexities inherent in designing and implementing CBF systems within RS, especially in healthcare domains where item attributes (e.g., treatment options, lifestyle interventions) may be multidimensional and clinically sensitive.

2.2.3 Collaborative Filtering

CF is a pivotal recommendation technique that diverges from CBF by focusing on user-item interaction patterns and learned user preferences rather than explicit item attributes (Müllner et al., 2023). In the context of managing hypertension, CF can be instrumental in personalizing patient care and intervention strategies. The methodology can be conceptually described in three primary steps, tailored to the healthcare setting:

- **Patient profiling** patients are grouped based on shared attributes in their health profiles, such as age, gender, medical history, lifestyle factors, and responses to

previous treatments. This segmentation is crucial for identifying patterns that might not be evident at the individual level.

- **Aggregation of effective interventions** within each cluster, a ranked list of the most effective interventions, medications, lifestyle changes, or monitoring strategies is compiled based on collective health outcomes and improvements.
- **Personalized intervention recommendations** the system then recommends interventions that a patient has not yet received, prioritizing those that have shown positive outcomes in similar patients.

The strength of CF in this domain lies in its ability to leverage the collective experiences of similar patients, offering a rich evidence base from which to draw recommendations without necessarily requiring explicit semantic descriptions of the interventions themselves. This approach is particularly valuable in the complex field of hypertension management, where the effectiveness of treatments can vary widely between individuals due to the multifaceted nature of the condition.

However, several challenges arise in applying CF to healthcare. One significant issue is the “similarity paradox” in medical recommendations: two patients might share similar observable health profiles but respond differently to the same treatment due to underlying conditions or genetic factors not captured in their records. Another challenge is the “rare patient dilemma,” where individuals with unique health conditions or rare responses to treatment are difficult to accurately cluster, potentially leading to less effective or even inappropriate recommendations.

Furthermore, the introduction of new treatments or interventions poses the well-known “cold-start” problem, wherein novel strategies cannot be reliably recommended until they have been used and evaluated in a sufficient number of patients. Addressing this requires innovative solutions, such as integrating expert medical knowledge, clinical guidelines, or outcomes from clinical trials to inform the recommendation process before extensive patient-generated data are available.

To mitigate these issues and enhance the effectiveness of CF in hypertension management, recent studies have proposed incorporating additional layers of context-aware filtering and ML algorithms to better predict individual responses to treatments. For example, Zheng et al. (Zheng et al., 2020) demonstrated the use of ML models to predict BP responses to different antihypertensive drugs, suggesting a path toward more personalized and effective treatment recommendations that can be integrated into CF-based frameworks.

2.2.4 Hybrid Recommender Systems

In addressing the multifaceted nature of hypertension management, pure CF and CBF approaches each present limitations when used in isolation. CF excels at leveraging user or patient similarities without needing detailed item-specific information, while CBF focuses on item characteristics, potentially missing broader patterns of user behavior and preferences. The integration of these methods, along with other techniques, into hybrid RS offers a robust solution by combining their strengths to overcome their respective weaknesses, thus enhancing the personalization and effectiveness of healthcare recommendations for patients with hypertension.

Hybrid RS combine multiple recommendation techniques (e.g., CF with rule-based or fuzzy models) to improve robustness and recommendation quality, as demonstrated in recent hybrid designs (Walek & Fajmon, 2023). These systems employ a variety of hybridization methods to achieve a synergy that significantly improves recommendation quality and relevance. In the context of hypertension management, these hybrid techniques can be particularly beneficial:

- **Weighted** combines outputs from both CF and CBF, assigning different weights to each component to reflect the clinical context (Ahmad et al., 2025). For instance, patient-reported outcomes and clinical data could be weighted differently depending on their relevance to the patient's condition and treatment history .
- **Switching** dynamically selects between collaborative filtering and content-based filtering based on contextual conditions and data characteristics. When sufficient interaction data are available and a patient aligns closely with a well-defined cluster of similar users, collaborative filtering may be prioritized. In contrast, when patient profiles exhibit unique attributes or comorbidities and interaction data are sparse, content-based filtering becomes more appropriate, as supported by recent studies on switching hybrid recommender strategies (Rizki & Rianto, 2024).
- **Mixed** presents recommendations from CF and CBF together, offering patients and clinicians a range of options from peer group based suggestions to options specifically tailored to the individual's clinical profile (Ahmad et al., 2025).
- **Feature combination** integrates features from diverse data sources such as Electronic Health Records (EHR), self-reported lifestyle data, and, in some studies, wearable device measurements to provide a comprehensive input set for generating nuanced recommendations (Ahmad et al., 2025).
- **Feature augmentation** uses the output of one technique to enrich the input of another. For instance, clusters of similar patients identified by CF could enhance the

content analysis performed in CBF by highlighting clinically relevant attributes not immediately apparent from item characteristics alone (Ahmad et al., 2025).

- **Cascade** refines recommendations through sequential filtering. An initial broad recommendation set generated by CF could be narrowed down by applying CBF criteria specific to hypertension management, such as medication contraindications, comorbidities, or lifestyle constraints (Ahmad et al., 2025).
- **Meta-level** uses a model developed through one approach such as a predictive model of treatment outcomes based on CF as an input representation for another method, enabling more complex analysis of treatment effectiveness and patient response patterns (Ahmad et al., 2025; Rizki & Rianto, 2024).

Recent advancements emphasize the importance of incorporating ML and context-aware filtering into hybrid RS for healthcare applications. For example, Luo et al. (Luo et al., 2021) explored the use of deep learning techniques within hybrid RS to predict patient responses to hypertension medications, demonstrating the potential for these systems to adapt dynamically to evolving patient data and medical knowledge.

By leveraging these hybridization strategies, an HCARS for hypertension can offer more accurate, personalized, and clinically meaningful treatment and management recommendations, while still being implementable in an offline, batch-oriented setting such as the modular Python pipelines and Streamlit-based environment adopted in this thesis.

2.2.5 Other Methods

Two additional techniques for creating RS that are worth mentioning are demographic systems and knowledge-based systems. Although they are not as widely used as CF or CBF, they still provide valuable insights and are frequently incorporated into hybrid RS.

2.2.5.1 Demographic Systems

A demographic RS relies on demographic information about users, such as gender, age, geographical location, marital status, occupation, and income (Jemal et al., 2024). By leveraging this data, it is possible to compute the demographic clusters to which a user is most likely to belong. This information then forms the basis for constructing user profiles, in a manner conceptually similar to CF. Recommendations are generated for each cluster, allowing specific demographic groups to be targeted with tailored advice.

This method can be particularly useful in hypertension management, where demographic factors such as age, sex, and socio-economic status are known to influence disease prevalence, risk profiles, and response to treatment. In the context of this thesis, demographic attributes available within the KHD can play a similar role when constructing profiles for the HCARS.

2.2.5.2 Knowledge-Based Systems

Knowledge-based RS utilize item features and explicit knowledge about how these features meet the user's needs to match items to user requirements (Munson et al., 2025). This approach often includes functional and domain-specific knowledge about how certain product or intervention features satisfy particular user goals.

In a healthcare context, if a patient requires a specific type of medication or lifestyle adjustment based on their condition, the system can recommend related interventions or products such as BP monitors, exercise programs, or dietary plans based on how well these options align with the patient's health objectives and clinical constraints (e.g., comorbidities or contraindications). Although this thesis does not implement a full rule-based expert system, the underlying idea of aligning recommendations with explicit clinical logic informs the design of hybrid and context-aware models.

2.2.6 Similarity Models

In RS, the ability to calculate similarities between users and items is fundamental for the effective implementation of both CF and CBF. Similarity models, akin to those used in information retrieval systems, are employed to identify relevant items (or users) based on a specific query. The choice of similarity model may vary depending on the system's objectives. For instance, CF systems aim to find users similar to a particular user, whereas CBF systems seek items resembling a given item.

Let U be the set of users with $u \in U$, and D the set of items with $d \in D$. The similarities of interest include $\text{sim}(u, d)$, $\text{sim}(u_i, u_j)$, and $\text{sim}(d_i, d_j)$. The most common methods for calculating these similarities are Euclidean distance, Pearson correlation, and cosine similarity (Bettache et al., 2025).

2.2.6.1 Euclidean Distance

Euclidean distance measures the straight-line distance between two points in n -dimensional space, making it one of the simplest methods for quantifying dissimilarity. The dissimilarity between two data points x and y is given by the Euclidean distance, defined as follows (Bishop, 2006):

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (2.1)$$

- $d(x, y)$: Euclidean distance between points x and y .
- x_k, y_k : The k -th components of vectors x and y , respectively.
- n : The number of dimensions (components) in the vectors.

2.2.6.2 Pearson Correlation

Pearson correlation assesses the linear relationship between two variables, providing a value within $[-1, 1]$, where 1 indicates total positive correlation, -1 indicates total negative correlation, and 0 indicates no linear correlation. It is formally defined as follows (James et al., 2023):

$$\text{sim}(x, y) = \frac{\sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^n (x_k - \bar{x})^2} \cdot \sqrt{\sum_{k=1}^n (y_k - \bar{y})^2}} \quad (2.2)$$

Here:

- x_k, y_k : The k -th value of variables x and y , respectively.
- $\bar{x}, \bar{y}$: The mean values of x and y , respectively.
- n : The number of elements in x and y .

2.2.6.3 Cosine Similarity

Cosine similarity considers two items x and y as vectors in n -dimensional space and measures the cosine of the angle between them. It is formally defined as follows (Leskovec et al., 2020):

$$\text{sim}(x, y) = \frac{x \cdot y}{|x||y|} \quad (2.3)$$

Where:

- $x \cdot y = \sum_{k=1}^n x_k y_k$: The dot product of vectors x and y .
- $|x| = \sqrt{\sum_{k=1}^n x_k^2}$: The magnitude (norm) of vector x .
- $|y| = \sqrt{\sum_{k=1}^n y_k^2}$: The magnitude (norm) of vector y .

The similarity value ranges between $[-1, 1]$, where:

- 1: indicates perfect similarity (vectors point in the same direction).
- 0: indicates no similarity (vectors are orthogonal).
- -1: indicates perfect dissimilarity (vectors point in opposite directions).

2.2.6.4 Weighting Techniques

While Euclidean distance, Pearson correlation, and cosine similarity are standard similarity measures in RS, various weighting techniques can be applied on top of these measures to better predict user ratings for items. In neighborhood-based CF, for example, the predicted rating $r_{c,s}$ for user c and item s is often computed by aggregating the ratings of the

N most similar users, weighted by their similarity to the target user. The prediction is normalized by the sum of the similarities to avoid bias. A common formulation is (Herlocker et al., 2004a; Koren et al., 2009; Resnick et al., 1994):

$$r_{c,s} = \frac{\sum_{\hat{c} \in C} \text{sim}(c, \hat{c}) \times r_{\hat{c},s}}{\sum_{\hat{c} \in C} \text{sim}(c, \hat{c})} \quad (2.4)$$

Here:

- $r_{c,s}$: predicted rating for user c on item s ,
- $\text{sim}(c, \hat{c})$: similarity between user c and user $\hat{c}$,
- $r_{\hat{c},s}$: rating given by user $\hat{c}$ to item s ,
- C : set of the most similar users to user c .

To account for differences in individual rating scales, the prediction can be adjusted using deviations from each user's average rating. The refined formula is (Herlocker et al., 2004a; Koren et al., 2009):

$$r_{c,s} = \bar{r}_c + \frac{\sum_{\hat{c} \in C} \text{sim}(c, \hat{c}) \times (r_{\hat{c},s} - \bar{r}_{\hat{c}})}{\sum_{\hat{c} \in C} \text{sim}(c, \hat{c})} \quad (2.5)$$

Where:

- $\bar{r}_c$: average rating of user c ,
- $\bar{r}_{\hat{c}}$: average rating of user $\hat{c}$.

This adjustment ensures that predictions are aligned with the rating tendencies of individual users, improving the accuracy and personalization of the recommendations an aspect that is particularly relevant in healthcare settings, where patient response patterns can vary considerably.

2.2.7 Classification Techniques

Classification, a cornerstone of ML, plays a pivotal role in the development of RS by enabling the prediction of group membership for data instances. It serves as a bridge between feature space and label space, and is crucial for determining, for example, whether a patient is likely to benefit from a specific intervention (Vebiyatama & Ernawati, 2024).

Broadly, classification can be situated within the broader landscape of supervised and unsupervised learning, where supervised methods rely on labeled data and unsupervised methods (such as clustering) aim to discover inherent groupings without labels (Verma

et al., 2025). For hypertension management, supervised methods might estimate the likelihood that a patient adheres to or benefits from a particular treatment, while unsupervised methods can identify patient subgroups with similar health profiles or treatment responses.

2.2.7.1 Naive Bayes

The Naive Bayes classifier, a supervised learning approach, employs Bayes' theorem to predict the class of a given record based on conditional probabilities, while making simplifying independence assumptions among features (Phatcharathada & Srisuradetchai, 2025):

$$P(C \mid R_1, R_2, \dots, R_N) = \frac{P(R_1, R_2, \dots, R_N \mid C) \cdot P(C)}{P(R_1, R_2, \dots, R_N)} \quad (2.6)$$

Here:

- $P(C \mid R_1, R_2, \dots, R_N)$: posterior of class C given $R_1, R_2, \dots, R_N$,
- $P(R_1, R_2, \dots, R_N \mid C)$: likelihood of observing features $R_1, R_2, \dots, R_N$ given class C ,
- $P(C)$: prior probability of class C ,
- $P(R_1, R_2, \dots, R_N)$: prior probability of the features $R_1, R_2, \dots, R_N$,
- N : number of features.

Despite its simplicity and strong independence assumptions, Naive Bayes has shown effectiveness in various domains, including preliminary diagnosis and risk stratification based on patient symptoms and historical data.

2.2.7.2 Support Vector Machines

Support Vector Machines (SVM) provide a robust supervised learning framework for classification in high-dimensional spaces and are widely used as strong baselines in modern medical prediction tasks (Abualigah et al., 2025; Yang, 2025). The decision function can be written as:

$$u = \mathbf{w} \cdot \mathbf{x} + b \quad (2.7)$$

where:

- $\mathbf{w}$ is the weight vector, which is normal to the hyperplane,
- $\mathbf{x}$ is the feature vector of the data point,
- b is the bias term (offset).

The margin, m , is defined as the inverse of the norm of $\mathbf{w}$:

$$m = \frac{1}{\|\mathbf{w}\|} \quad (2.8)$$

SVMs have been extensively utilized in applications such as predicting patient risk categories, classifying disease stages, and estimating treatment outcomes, offering substantial benefits in handling high-dimensional and complex patient data.

2.2.7.3 k-Nearest Neighbors (kNN)

The kNN algorithm classifies data based on the closest training examples in the feature space. It is particularly useful for RS due to its simplicity and effectiveness in identifying similar cases (Mallen & Bridge, 2024; Prahmana et al., 2025).

Given a point to be classified, kNN identifies the k closest points (neighbors) in the training dataset. The point is then assigned the predominant label among these neighbors, as illustrated in Figure 2.1. For example, with $k = 3$, the new point (green) is classified according to the majority class of its three closest neighbors (red triangles). Increasing k to 5 considers a larger neighborhood and may yield different results depending on the distribution of nearby data points.

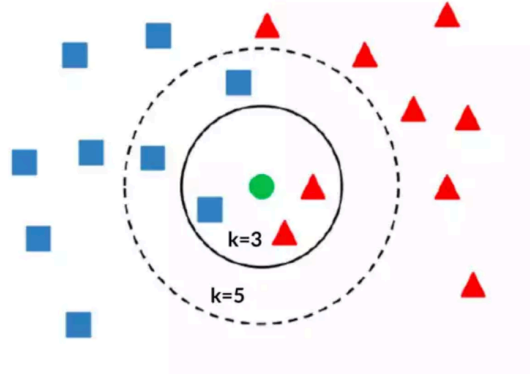


Figure 2.1: Illustration of kNN with $k=3$ and $k=5$ the new point (green) takes the majority label of its nearest neighbors.

Similarity Metrics and Challenges Closeness can be measured using the similarity metrics discussed in Section 2.2.6. These metrics make kNN adaptable to various contexts, such as patient similarity analysis for personalized treatment recommendations. However, determining the optimal value of k poses a challenge:

- **Low k :** If k is too low, the algorithm becomes highly sensitive to noise, potentially leading to unstable or incorrect classifications.
- **High k :** If k is too high, the neighborhood may include points from other classes, reducing classification accuracy and blurring decision boundaries.

Another drawback of kNN is that it is a “lazy learner,” meaning it does not construct a general model during training. Instead, it computes distances for each new data point at prediction time, which can limit scalability for large datasets. In practice, these trade-offs must be balanced carefully when considering kNN for healthcare applications or as a component within hybrid recommendation architectures.

2.2.7.4 k-Means Clustering

k -Means is an unsupervised learning algorithm that aims to partition observations into k clusters based on their features, minimizing the variance within each cluster (Jain, 2010):

$$E = \sum_{k=1}^K \sum_{\vec{x}_n \in S_k} \|\vec{x}_n - \vec{\mu}_k\|^2 \quad (2.9)$$

Here:

- E : total within-cluster sum of squares (error function),
- K : number of clusters,
- $\vec{x}_n$: n -th data point assigned to cluster k ,
- $\vec{\mu}_k$: centroid (mean vector) of cluster k ,
- S_k : set of data points assigned to cluster k ,
- $\|\cdot\|$: Euclidean norm (distance measure).

In hypertension management, k -Means can be used to identify patient subgroups with similar health patterns (for example, similar combinations of BP, age, BMI, and comorbidities), thereby supporting the development of targeted intervention strategies and more tailored recommendation rules.

2.2.8 Feedback Mechanisms and Contextual Data Integration

Feedback plays a pivotal role in enhancing the performance of RS by enabling fine-grained personalization over time. For systems focusing on hypertension management, capturing how patients interact with recommendations, educational content, and monitoring tools is essential for tailoring interventions and assessing their effectiveness.

Two primary feedback mechanisms are typically distinguished: explicit feedback and implicit feedback, each offering complementary insights into patient preferences and behaviors. Table 2.1 summarizes representative examples in the healthcare context.

2.2.9 Proactive Recommendations

The evolution of RS towards proactive models emphasizes the ability to anticipate user needs and offer timely, context-aware suggestions without explicit requests (Sun et al., 2023). In the healthcare literature, this has been discussed in relation to integrating health monitoring data, environmental factors, and patient activity patterns so that recommendations can be dynamically adjusted, thereby improving the management of chronic conditions such as hypertension. In this thesis, these ideas are considered conceptually, while the implemented HCARS is evaluated in an offline, batch-processing setting.

2.2.9.1 Explicit Feedback

Explicit feedback requires patients to actively provide their preferences or opinions about recommended interventions or information. Common forms include:

- **Binary responses** patients indicate acceptance or rejection of recommendations, analogous to like/dislike buttons on social media platforms.
- **Ratings** patients rate interventions or health information on a numerical scale, reflecting their perceived effectiveness or relevance.
- **Text comments** patients provide textual feedback on recommendations, offering detailed insights that may require sentiment analysis or natural language processing to interpret.

While explicit feedback directly reflects patient preferences and perceived usefulness, it can impose a cognitive burden and often depends on the patient's willingness and ability to engage consistently with the system.

2.2.9.2 Implicit Feedback

Implicit feedback collected without requiring patients to provide explicit ratings or responses. It is inferred from their interactions with the system and related digital tools, such as:

- **Engagement metrics** time spent reading educational content, viewing specific recommendations, or interacting with mobile health applications.
- **Health-related behavior traces** data from mobile or wearable applications that track physical activity, BP measurements, or proxies for medication adherence.

Implicit feedback is essential for understanding patient behavior and preferences indirectly, providing a continuous stream of observational data with minimal additional effort from patients.

Table 2.1: Examples of Explicit and Implicit Feedback in Healthcare RS

Explicit Feedback	Implicit Feedback
Binary (Like/Dislike)	Engagement metrics
Ratings (scale of 1–10)	Health-related behavior traces
Text comments	

Feedback mechanisms, both explicit and implicit, are therefore integral to developing a dynamic and responsive RS for hypertension management, enabling iterative refinement of recommendations and gradual adaptation to patient needs.

2.2.9.3 Contextual Data

The integration of user feedback with contextual information is critical for refining the accuracy and relevance of recommendations provided by an HCARS for hypertension problems. User feedback (explicit and implicit) can be interpreted more effectively when combined with contextual data that capture environmental variables, user-specific conditions, temporal aspects, and physiological parameters.

Contextual data pertinent to hypertension management may include:

- **Environmental factors** seasonal variations, air quality indicators, and broad geographic context.
- **Health status** BP readings, physical activity levels, presence of comorbidities, and dietary habits.
- **Temporal context** time of day, day of the week, or season, and their relationship to patient routines and health patterns.

2.2.9.4 Integration Process

The process of integrating user feedback with contextual data involves analyzing these combined signals to tailor recommendations more closely to a patient’s current health status and circumstances. Explicit and implicit feedback can be weighted or filtered according to the contextual variables observed when the interaction occurred (for example, recent BP levels, workload periods, or lifestyle changes).

By leveraging both user feedback and contextual data, the recommendation system can develop a more comprehensive picture of each patient’s situation. This, in turn, improves the relevance and interpretability of hypertension management recommendations and can enhance patient engagement and satisfaction through more personalized health-care experiences.

2.3 Previous Studies

Healthcare RS have undergone rapid development, driven by advances in AI and the increasing integration of digital health technologies. These systems now play a central role in supporting personalized and preventive care by analyzing clinical data, lifestyle factors, and patient behavior (Barbarić et al., 2025). Through continuous learning and adaptive feedback, modern HRS can assist patients and clinicians in making more informed decisions, thereby improving treatment adherence and health outcomes. In the specific context of hypertension, such intelligent systems are positioned within broader digital precision medicine frameworks, where particularly in some recent implementations they can provide data-driven recommendations for lifestyle modification and therapeutic adjustment (Clifford & et al., 2025).

2.3.1 Healthcare Recommender Systems

HRS have emerged as critical tools in modern healthcare, aiming to deliver personalized medical recommendations by leveraging patient data, EHR, and advanced computational algorithms (Sun et al., 2023). The advent of big data and the increasing digitization of healthcare information have paved the way for HRS to assist both patients and healthcare providers in making more informed decisions (Chen et al., 2025). These systems have been applied across various domains, including treatment recommendations, medication management, provider selection, and lifestyle modification, thereby enhancing patient engagement and improving health outcomes. The integration of ML, deep learning, CF, and hybrid approaches in HRS ensures that recommendations can be tailored to individual patient needs, aligning with the broader shift toward personalized medicine.

A comprehensive systematic review conducted by De Croon et al. evaluated numerous studies on HRS, categorizing them based on application domains and underlying methodologies (De Croon et al., 2021). The review highlighted the growing importance of hybrid and context-aware models in healthcare, particularly in decision support systems, personalized treatment planning, and medication recommendation. De Croon et al. emphasized that context-aware systems are particularly effective in chronic disease management by integrating patient health data with contextual factors to provide more personalized recommendations. This observation underscores the necessity for HRS that can adapt to the dynamic nature of patient health, a critical consideration in managing conditions such as hypertension where clinical status can fluctuate over time.

Anwar et al. addressed the pervasive issues of cold-start and data sparsity in HRS through the implementation of CF and k -NN algorithms (Anwar et al., 2022). By leveraging similarity measures between patients, their approach ensures that even with limited patient data, the system can generate reasonable recommendations by drawing on the profiles of similar users. Their experiments demonstrated improved recommendation accu-

racy, highlighting the effectiveness of combining CF with neighborhood-based techniques in healthcare settings where interaction data are often sparse or incomplete, particularly for newly enrolled patients.

Jha et al. introduced a deep RL-based hospital recommender system targeting rural patients (Jha et al., 2023). The system optimizes recommendations based on multiple criteria, including geographic proximity, treatment effectiveness, and cost, in order to provide cost-effective hospital suggestions. By employing RL, the system learns from patient interactions and adapts its recommendations over time. Although their work focuses on hospital selection rather than direct treatment recommendations, it demonstrates the potential of adaptive learning to enhance healthcare accessibility and personalization.

Stratigi et al. developed a multidimensional group recommendation system in the health domain, integrating individual and group preferences with medical data to improve health outcomes (Stratigi et al., 2020). Their system effectively accounts for both patient and provider preferences, highlighting the importance of personalized and collaborative approaches in healthcare recommendations.

Building upon these foundational developments, more recent research has expanded the scope and evaluation depth of HRS. Barbarić et al. conducted a comprehensive scoping review that consolidates more than a decade of studies on HRS design and implementation, emphasizing the persistent gap between algorithmic innovation and clinical adoption (Barbarić et al., 2025). Their findings show that although personalization techniques and hybrid architectures are increasingly sophisticated, empirical validations in real-world healthcare environments remain limited, highlighting the need for standardized evaluation frameworks.

Complementing this, Ananthakrishnan et al. proposed a structured methodological protocol to evaluate HRS beyond accuracy-based performance measures, advocating for multidimensional assessment criteria such as user trust, interpretability, and clinical safety (Ananthakrishnan et al., 2024). This perspective aligns with the growing recognition that HRS should not merely optimize predictive performance, but also exhibit transparent, reliable, and ethically responsible behavior.

In parallel, Clifford et al. contextualized these advancements within the emerging domain of digital precision hypertension, identifying a shift from generic data-driven recommendations toward individualized, context-sensitive care pathways (Clifford & et al., 2025). Their review illustrates how integrating recommender logic into digital precision medicine frameworks can support chronic disease management by combining longitudinal monitoring with adaptive feedback loops.

Taken together, these contemporary studies indicate that the evolution of HRS is moving toward clinically integrated, ethically governed, and context-aware ecosystems, gradually bridging the gap between computational modeling and patient-centred healthcare delivery.

2.3.2 Context-Aware Recommender Systems

The CARS extend traditional RS by incorporating contextual information such as location, time, environmental conditions, and user preferences to provide more relevant and personalized recommendations (Adomavicius et al., 2011). In healthcare, the dynamic nature of patient conditions and behaviors necessitates systems that can adapt to changes over time and across situations, making CARS particularly suitable for managing chronic conditions where timely adjustments and personalized guidance are important.

Torres-Ruiz et al. presented an HRS that integrates geospatial data with patient profiles to enhance healthcare access (Torres-Ruiz et al., 2023). Their system uses medical specialties and patient proximity to recommend healthcare services, demonstrating how geographical data, when combined with patient characteristics, can support more context-aware recommendations. By incorporating location-based information, the system helps ensure that patients receive recommendations for services that are both clinically appropriate and practically accessible, which is especially beneficial in emergency situations or for patients living in remote areas.

Jeong and Kim developed a context-aware recommender system that accounts for changes in user preferences over time by leveraging neural networks (Jeong & Kim, 2023). Their system dynamically adjusts recommendations based on evolving user health data, addressing the need for recommendation models that can adapt to users' changing conditions. By employing deep learning techniques, the system captures complex patterns in user behavior and preferences, thereby enhancing personalization and predictive accuracy.

Zitouni et al. proposed a CARS for personalized healthy nutrition education (Zitouni et al., 2022). Their system is based on a two-dimensional contextual framework that integrates internal (personal health data) and external (environmental factors) contexts to tailor recommendations. By combining contextual pre-filtering algorithms with CF, the system adapts food suggestions to both personal health needs and environmental circumstances such as time and location. This methodology demonstrates the potential of CARS to provide personalized health interventions by considering a comprehensive set of contextual factors.

Ali et al. introduced a dynamic context management approach in CARS, which accounts for changing user preferences by incorporating a modified Minkowski distance for contextual attributes such as time and user behavior (Ali et al., 2023). Their technique improves prediction accuracy by dynamically adjusting to the evolving context, highlighting the importance of context-sensitive modeling in applications where user states and environments are not static.

In related work, Jeong and Kim also explored the integration of deep learning with context-aware systems, focusing on time and location as key contextual variables (Jeong & Kim, 2022). Their model combines an autoencoder with neural networks to address

data sparsity and improve user preference prediction in sparse data environments. Despite achieving improved performance under dynamic conditions, the model's slower response times highlight the practical need to balance model complexity with responsiveness in real-world deployments.

Abusair et al. developed a content-based CARS for mobile applications in the food delivery domain, combining contextual information such as time and temperature with user preferences (Abusair et al., 2020). The system achieved a 20% increase in user satisfaction compared with more traditional models. Although their work focuses on food delivery, the methodology of integrating contextual factors is transferable to healthcare scenarios in which contextual variables (such as schedule, environment, and routine) significantly influence the relevance of recommendations.

Adomavicius and Tuzhilin provided a comprehensive review of contextual modeling techniques, ranging from contextual pre-filtering to multidimensional algorithms (Errakha et al., 2024). Their analysis clarifies the role of context in enhancing recommendation accuracy and discusses how different modeling choices depend on the specific application domain and available data sources considerations that are directly applicable to digital health, where longitudinal patient information and coarse contextual indicators can be combined to refine recommendation quality.

Building on these developments, recent studies have emphasized the practical integration of behavioral, temporal, and environmental contexts into healthcare RS. A randomized controlled trial published in *JMIR Cardio* demonstrated the effectiveness of an AI-based autonomous lifestyle coach that adapts to users' habits, daily routines, and adherence levels to support BP control (omitted by journal (*JMIR Cardio*), 2024). By leveraging behavioral feedback, this system exemplifies how temporal and user-specific contexts can be used to sustain long-term engagement and achieve measurable health outcomes in chronic disease management.

In a similar vein, Lipski et al. proposed the SOT-ART-HT telemedical framework, which integrates recommendation components directly into remote diagnostic and therapeutic workflows for hypertensive patients (Lipski et al., 2025). This framework highlights how spatial and temporal contextualization through home-based monitoring and teleconsultation can enhance diagnostic precision, treatment personalization, and accessibility of care, particularly for patients requiring frequent follow-up.

Adeleke et al. extended this paradigm by analysing a workplace hypertension screening programme using ML models to identify risk trajectories and intervention opportunities (Adeleke & et al., 2024). Their longitudinal approach shows how environmental context, such as occupational setting and daily exposure patterns, can inform tailored preventive recommendations and policy-level interventions.

Collectively, these contemporary studies demonstrate that embedding behavioral, spatial, and environmental contexts into RS can transform them from static decision aids into

adaptive, patient-centred ecosystems. Such context-aware designs enable more sustained personalization, support better adherence, and help bridge the gap between digital recommendations and real-world healthcare delivery.

2.3.3 Hybrid Recommender Systems

Hybrid RS combine multiple recommendation techniques, such as CF and CBF, to overcome the limitations of single-method approaches and to enhance the accuracy, robustness, and personalization of recommendations, including in clinical decision-support settings (Barbaric et al., 2025; Sun et al., 2023). In healthcare, where patient data are multifaceted, heterogeneous, and often incomplete, hybrid systems are increasingly necessary for providing personalized and reliable recommendations, particularly in managing chronic conditions that require integration of diverse data sources.

Hors-Fraile et al. explored the use of collective intelligence in HRS by combining crowd-sourced user data with expert insights to support smoking cessation programmes (Hors-Fraile et al., 2022). Their hybrid approach led to stronger user engagement and more personalized recommendations, demonstrating how collective intelligence and hybrid RS can be used to deliver more effective health interventions. By integrating user-generated data with expert knowledge, the system offers tailored strategies, emphasizing the value of combining data-driven and expert-driven components.

Kamran and Javed reviewed multiple types of HRS, including CF, CBF, and hybrid systems, and concluded that hybrid models generally offer better performance in healthcare, especially in the management of chronic diseases (Kamran & Javed, 2015). Their analysis shows that hybrid systems are well suited to addressing challenges such as data sparsity and cold-start problems, which are common in medical settings where patient records may be incomplete or fragmented. By integrating multiple methodologies, hybrid systems can provide more accurate and personalized healthcare solutions.

Zhang et al. developed a system that leverages ML algorithms to improve the predictive accuracy of health recommendations (Y. Zhang et al., 2018). Their hybrid system combines ML models, CF, and clinical data to deliver personalized treatment suggestions. This work highlights the growing role of ML as an integral component in HRS, offering scalable solutions for chronic disease management by continuously learning from patient data.

Maleki et al. proposed a hybrid k-NN method enhanced with a genetic algorithm (GA) for lung cancer prognosis (Maleki et al., 2021). Their model optimizes feature selection, leading to a significant increase in classification accuracy. Although their application domain is lung cancer, the underlying methodology is relevant to chronic conditions such as hypertension, where careful feature selection can substantially improve risk stratification and recommendation performance.

Zhu et al. conducted a review on personalized medicine recommendation systems, em-

phasizing the importance of integrating deep learning and RL (Zhu et al., 2024). They showed how combining patient health records with drug–drug interaction data can enhance treatment recommendations. Their review illustrates that hybrid models typically offer superior predictive performance compared with standalone systems, particularly in the management of chronic diseases that require ongoing monitoring and iterative adjustment of treatment plans.

Recent advancements have further redefined the role of hybrid HRS, shifting them from primarily algorithmic constructs toward clinically interpretable and operationally deployable frameworks. Cavero-Redondo et al. (2024) exemplified this trend through the RIGIPREV multicohort study, which employed ML to generate personalized antihypertensive treatment recommendations aimed at reducing arterial stiffness (Cavero-Redondo & et al., 2024). By integrating predictive modeling with clinical treatment pathways, their work shows how hybridization can extend beyond data-level fusion to achieve direct therapeutic relevance and individualized drug optimization.

Similarly, Yi et al. (2024) developed a predictive model for estimating individual BP responses to antihypertensive medications, providing a decision support component that can be incorporated into hybrid recommendation frameworks (Yi & et al., 2024). This approach bridges statistical modeling with pharmacological expertise, allowing the construction of adaptive treatment strategies that respond to heterogeneous patient profiles and variable drug responses.

Expanding this paradigm, Nguyen et al. (2024) introduced predictive models for sustained uncontrolled hypertension, emphasizing the need for hybrid architectures that integrate ML predictions with longitudinal clinical monitoring (Nguyen & et al., 2024). Their work demonstrates that continuous feedback between predictive analytics and follow-up data can improve therapeutic adherence and facilitate more proactive intervention strategies.

Collectively, these contributions reinforce the evolution of hybrid RS from theoretical designs into clinically actionable tools. By merging data-driven inference with evidence-based medical reasoning, hybrid HRS are becoming increasingly important in chronic disease management, enabling dynamic, personalized, and explainable healthcare recommendations that align computational intelligence with real-world clinical practice.

2.3.4 Studies Specific to Hypertension Management

Hypertension is a chronic condition marked by persistently elevated BP, associated with substantial health risks and requiring long-term management (Organization, 2025). Because multifactorial shaped by genetic, environmental, and lifestyle factors, effective care benefits from personalised approaches (Errakha et al., 2024). Recent studies have applied RS and predictive models to hypertension management, combining diverse data sources and algorithms to improve risk prediction and support individualised recommendations.

Yashvanth et al. investigated the use of ML algorithms for predicting chronic diseases, including hypertension, diabetes, and stroke (Yashvanth et al., 2023). They employed several ML techniques such as logistic regression, SVM, and gradient boosting to predict health outcomes based on patient-level data. By incorporating feature selection techniques such as principal component analysis, they improved model performance and achieved an accuracy of 75.5% in predicting hypertension. This work illustrates how classical ML models can identify key predictive features and support early risk stratification.

Lajevardi and Kargari proposed a CNN-based deep learning model to improve hypertension risk prediction using single nucleotide polymorphisms (SNPs) (Lajevardi et al., 2022). By transforming SNP data into image-like representations suitable for CNN input, their model achieved an area under the curve (AUC) of 0.877, outperforming traditional polygenic risk scores (PRS). Their study highlights the potential of deep learning for handling complex genetic data and providing more precise risk assessments based on genetic predisposition.

Valdeira et al. employed extreme multilabel classification to recommend specialist doctors based on sparse patient data (Valdeira et al., 2023). Their approach addresses the cold-start problem in HRS by ensuring that clinically meaningful recommendations can be generated even when patient information is limited. This is particularly relevant in hypertension management, where patients may require input from multiple specialists across cardiology, nephrology, and primary care. By effectively handling data sparsity, their system enhances the accessibility and relevance of healthcare recommendations.

Zheng et al. demonstrated the use of ML models to predict BP responses to different antihypertensive drugs, suggesting a route toward more personalized pharmacological treatment (Zheng et al., 2020). By analysing patient characteristics and treatment outcomes, their models were able to anticipate individual responses to specific medications, thereby supporting more tailored drug selection and dosage adjustment.

Building on these foundations, several recent studies have advanced hypertension management through AI-driven recommender and predictive frameworks. Mroz et al. (2024) developed an ML model capable of estimating the likelihood of achieving BP control under varying treatment and behavioral conditions (Mroz & et al., 2024). Their model demonstrated strong predictive performance by integrating longitudinal patient data, medication adherence patterns, and lifestyle indicators. This predictive approach directly supports personalized follow-up strategies, enabling systems to suggest treatment adjustments or behavioral interventions before clinical deterioration occurs.

Skalidis et al. (2025) provided a comprehensive review of AI and advanced digital health systems for hypertension management (Skalidis & et al., 2025). Their analysis describes the convergence of multiple AI modalities including recommender components, wearable-based monitoring, and digital therapeutics into an emerging ecosystem of precision hypertension care. They emphasize that the next generation of digital health frame-

works must prioritize interoperability, transparency, and rigorous clinical validation to bridge the gap between algorithmic potential and real-world impact.

Reinforcing these insights, a randomized controlled trial reported in JMIR Cardio (2024) evaluated an AI-based autonomous lifestyle coach that supports BP control through adaptive behavioral guidance and feedback (omitted by journal (JMIR Cardio), 2024). By learning from users' daily habits and contextual cues, the system illustrates how intelligent recommendation mechanisms can transform preventive care into a dynamic, personalized process of self-management.

Taken together, these contributions show how RS and predictive models for hypertension have evolved beyond static risk scoring toward more adaptive, context-aware platforms capable of learning from user behavior, clinical responses, and environmental factors. The integration of predictive analytics, digital coaching, and precision monitoring marks a key step toward sustained BP control and patient-centred hypertension care.

2.4 Critique and Analysis of Previous Studies

2.4.1 Challenges in Current Systems

The challenges faced by modern HRS, particularly in the management of chronic conditions such as hypertension, are multifaceted. The cold-start problem, data sparsity, scalability constraints, limited proactivity, and privacy concerns all restrict the extent to which current systems can deliver robust, individualized recommendations. These limitations are especially consequential in healthcare settings, where timely and accurate recommendations can have direct implications for patient outcomes.

2.4.1.1 Cold-Start Problem

Despite advances in hybrid models that combine CF and CBF, the cold-start problem remains a persistent challenge. The cold-start issue arises when a system has insufficient information about a new patient (or a new intervention), making it difficult to generate well-calibrated personalized recommendations. Hybrid systems attempt to mitigate this by integrating multiple recommendation techniques; however, they still struggle in scenarios where only minimal historical or interaction data are available.

In healthcare, many patients initially present with limited digital records or short follow-up windows, which can significantly affect the quality of early recommendations. For hypertension management, BP patterns and treatment responses may evolve over time in ways that are not yet captured by the system, leading to suboptimal suggestions at the beginning of the patient journey. One pragmatic strategy is to leverage demographic and readily available clinical variables as initial filters in a hybrid system, using population-level insights to guide early recommendations until richer individual-level data become

available. For example, a new patient’s management plan can be informed by patterns observed in patients with similar characteristics in datasets such as the KHD (Chuks, 2022), while explicitly acknowledging that such approximations are less precise than recommendations derived from detailed longitudinal data.

2.4.1.2 Scalability

As healthcare data grow in volume and complexity, scalability becomes a central concern. The integration of diverse data types such as EHR, lifestyle indicators, and, in some advanced settings, wearable-derived signals or genomic information into hybrid RS presents both opportunities and technical challenges. These systems must process heterogeneous, high-dimensional data efficiently enough to support timely recommendations without compromising performance.

In the context of hypertension management, systems may need to combine historical clinical data with higher-frequency measurements (e.g., repeated BP readings, basic activity metrics) and contextual variables to provide nuanced recommendations. This integration introduces computational complexity that can create bottlenecks, especially when scaling to large populations. Scalability can be improved through strategies such as distributed computing architectures, parallel processing, and model compression. ML models including deep neural networks (DNNs) can provide powerful predictive capabilities at scale, but they require careful engineering to ensure that computational demands remain compatible with the intended deployment environment.

2.4.1.3 Lack of Proactive, Real-Time Recommendations

Another limitation of many existing HRS is their limited capacity to provide proactive recommendations that anticipate changes in a patient’s condition. Traditional systems often rely on static, historical data and periodic updates, which may not capture short-term fluctuations or context-dependent risks. For hypertension, where BP can be influenced by stress, medication adherence, and daily routines, the ability to generate anticipatory advice is particularly valuable.

More advanced frameworks that integrate home BP monitoring, digital therapeutics, or telemedicine can leverage higher-frequency data and feedback loops, but such deployments remain relatively rare and resource-intensive. Methodologically, the central challenge is to combine historical records with recent measurements and contextual indicators to enable timely, clinically meaningful interventions, without over-claiming what the available data and infrastructure can support. In offline or batch research, this is often approximated by simulating proactive behaviour on longitudinal datasets, whereas true real-time operation typically requires additional infrastructure beyond the scope of many studies.

2.4.2 Evaluation of Existing Systems

Evaluating an RS within the healthcare domain, particularly for hypertension management, requires careful consideration of both methodological rigour and clinical relevance. Multiple complementary evaluation strategies are typically employed to obtain a comprehensive view of system performance.

2.4.2.1 Offline Evaluation

Offline evaluation relies on historical data to assess how well an RS can predict user behavior, treatment choices, or health outcomes. It is cost-effective and does not require direct interaction with patients, making it suitable for early-stage experimentation. However, its main limitation is that historical datasets may not fully capture future patterns, new interventions, or evolving clinical practices. To compensate for this, researchers often employ simulation techniques that approximate patient feedback and explore counterfactual recommendations under controlled assumptions.

2.4.2.2 Online Evaluation

Online evaluation often implemented through A/B testing or related experimental designs measures actual interactions between patients (or clinicians) and the RS. This approach provides insight into how changes in the RS affect user behavior, adherence, and, when monitored over sufficient time, clinical outcomes. Although online evaluation is more complex and resource-intensive than offline experimentation, it is indispensable for assessing real-world impact and for identifying unintended consequences of deployed systems.

2.4.2.3 User Studies

User studies involve direct engagement with patients, clinicians, or other stakeholders to gather qualitative and quantitative feedback on their experiences with the RS. These studies can reveal perceptions of usability, trust, interpretability, and perceived value dimensions that are not fully captured by predictive accuracy alone. While user studies require more effort and careful ethical oversight, they are crucial for understanding user needs and for aligning system design with real-world workflows and expectations.

2.4.3 Principal Properties of Recommender Systems

The effectiveness of an RS, particularly in healthcare, can be evaluated using several core properties:

- **User preference** user satisfaction and acceptance by patients and clinicians (e.g., surveys, engagement, adoption).

- **Prediction accuracy** how well the RS predicts patient preferences, behaviours, or outcomes (e.g., RMSE and classification metrics).
- **Coverage** the ability to recommend across diverse treatments, patient profiles, and clinical scenarios.
- **Serendipity and diversity** delivering novel yet plausible options while avoiding repetitive recommendations.
- **Robustness** stability under noisy or incomplete data and under changing input distributions.
- **Adaptivity** responsiveness to new patient information, shifting efficacy, or updated clinical guidelines.
- **Scalability** maintaining performance as data volume, user base, or the recommendation space grows.

2.4.4 Equations and Metrics

The performance of an RS can be quantified using standard classification metrics that evaluate how well the system distinguishes between positive and negative instances (Jadon & Patil, 2024; Kowald, 2024). A common measure is accuracy, defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.10)$$

Here:

- TP: Number of true positives.
- TN: Number of true negatives.
- FP: Number of false positives.
- FN: Number of false negatives.

Two additional core metrics are precision and recall:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.11)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.12)$$

Precision captures the proportion of predicted positive instances that are truly positive, while recall captures the proportion of actual positive instances that are correctly identified. In healthcare RS, high precision is important to avoid recommending ineffective or

inappropriate interventions, and high recall is important to minimize missed opportunities for beneficial treatments.

These metrics are fundamental when RS components are framed as classification tasks, for instance when predicting whether a patient belongs to a particular risk group or is likely to respond to a specific intervention. However, many RS including the hybrid system developed in this thesis ultimately generate ranked recommendation lists rather than single binary decisions. Consequently, classification metrics are complemented by ranking-oriented measures such as Precision@ k , Recall@ k , Mean Average Precision at k (MAP@ k), and normalized Discounted Cumulative Gain at k (nDCG@ k) for the offline evaluation of recommendation quality in the proposed HCARS (Jadon & Patil, 2024; Kowald, 2024).

2.4.5 Gap Analysis and Research Opportunities

Despite substantial progress in HRS, particularly in hybrid and context-aware frameworks, several critical gaps remain. A prominent limitation concerns the incomplete integration of diverse data sources genetic, environmental, lifestyle, and longitudinal clinical variables into unified systems for chronic disease management. Many existing approaches concentrate on a single dimension of the patient profile, such as genetic predispositions, lifestyle behaviours, or structured clinical records, rather than embracing the full complexity of hypertension, which is shaped by both intrinsic (e.g., genetic risk) and extrinsic (e.g., environmental stressors, socio-economic context) factors.

For example, research focusing on genetic predictors of hypertension risk (Lajevardi et al., 2022) has demonstrated the utility of incorporating genomic information into predictive models, yet such work frequently omits lifestyle and environmental dimensions that strongly influence BP control. Conversely, systems that emphasize lifestyle or behavioral data often do not explicitly incorporate genetic risk, even when such information could help refine individual risk stratification and treatment selection (Kamran & Javed, 2015). This separation of data modalities limits the potential for genuinely holistic risk assessment and tailored intervention planning.

Another important gap arises from the relatively limited use of CARS specifically targeted at hypertension management. While context-aware approaches have been successfully deployed in domains such as mobile health, wellness applications, and nutrition support (Jeong & Kim, 2023; Zitouni et al., 2022), their application to hypertension remains comparatively underexplored. Theoretical work and proof-of-concept systems illustrate how temporal, behavioral, and environmental context could enhance personalization, but fully integrated, clinically validated CARS for hypertension are still scarce. Moreover, many existing prototypes assume access to fine-grained, real-time data streams from WDs or environmental sensors, which are not always available or feasible in routine care settings.

The integration of diverse data sources into a single hybrid system is therefore both

a challenge and an opportunity. Hybrid models have helped address cold-start and data sparsity issues by combining CF and CBF (Barbaric et al., 2025), yet relatively few implementations support flexible, context-aware reasoning over heterogeneous data, particularly when genomic, behavioral, and environmental inputs are considered together. From a methodological perspective, there is a need for frameworks that can operate effectively in more constrained data settings where continuous monitoring is not available while still remaining extensible to richer data sources in future deployments.

In this thesis, these research opportunities are addressed within a realistic, data-constrained setting using the KHD (Chuks, 2022). The KHD provides structured information on demographics, clinical indicators, and lifestyle-related variables, but does not include genomic or fine-grained environmental measurements. As such, it offers a practical foundation for developing and evaluating an HCARS that simulates contextual adaptation using the available variables (e.g., age, BMI, smoking status, physical activity), while explicitly acknowledging the absence of genomic data and live sensor streams. This offline, batch-oriented setting enables rigorous experimentation on hybrid and context-aware recommendation logic without overclaiming real-time capabilities.

The resulting research opportunities are clear. Future work should prioritize:

- designing architectures capable of integrating heterogeneous data modalities, including genetics, environment, and behavior, when such data are available;
- extending hybrid models to support richer context representations and more flexible temporal reasoning; and
- developing evaluation protocols that link offline performance to clinically meaningful endpoints.

Within this broader landscape, the present thesis contributes by proposing and empirically assessing an HCARS that leverages the KHD to approximate context-aware, hybrid recommendations for hypertension management in an offline research environment.

2.4.6 Comments on Previous Studies

A critical analysis of existing literature on HRS, particularly in relation to hypertension, reinforces the presence of several structural limitations that constrain current systems. Hybrid and context-aware models have improved personalization and mitigated some aspects of sparsity and cold-start (Kamran & Javed, 2015), yet they frequently operate on partial views of the patient, emphasizing either genetics, lifestyle, or clinical records in isolation. For a condition as multifactorial as hypertension (Barbaric et al., 2025), this fragmentation limits the capacity to deliver nuanced, patient-specific recommendations.

Several studies illustrate this pattern. Lajevardi and Kargari (Lajevardi et al., 2022) focus on CNN-based modeling of genetic data to predict hypertension risk, but do not

integrate lifestyle or environmental factors. Yashvanth et al. (Yashvanth et al., 2023) employ ML algorithms to predict chronic diseases, including hypertension, using structured patient data, yet their approach does not explicitly incorporate contextual information or dynamic monitoring. While these works advance the state of predictive modeling, they do not fully address the way genetic, clinical, behavioral, and contextual dimensions interact over time in real-world hypertension care.

The application of CARS in hypertension management remains similarly limited. Context-aware systems have been proposed for domains such as personalized nutrition (Zitouni et al., 2022) and dynamic context management in recommendation tasks (Ali et al., 2023; Jeong & Kim, 2023), but analogous systems tailored to hypertension-specific workflows are still rare. Many context-aware prototypes assume access to rich, continuous sensor data, which may not be consistently available in typical clinical or resource-constrained settings. Consequently, there is a gap between conceptual models of fully context-aware hypertension management and the practical realities of data availability and system deployment.

Cold-start and data sparsity problems further complicate personalization. Anwar et al. (Anwar et al., 2022) address sparsity using CF and k-NN techniques, improving recommendations when interaction data are limited, yet their work does not explicitly link these methods with context-aware or hybrid architectures for hypertension. Scalability concerns reported by Kamran and Javed (Kamran & Javed, 2015) underscore the need for architectures that can handle large, heterogeneous datasets without sacrificing interpretability or performance.

Within this landscape, the contribution of the present thesis is intentionally scoped and methodologically transparent. Rather than claiming to deliver a fully real-time, sensor-driven CARS, the study develops and evaluates an HCARS for hypertension management that operates in an offline, batch-processing setting. The system is instantiated using the KHD (Chuks, 2022) and is implemented through modular Python pipelines with a research-oriented, read-only Streamlit dashboard. In this configuration, contextual adaptation is simulated using static variables available in the dataset such as age, BMI, smoking status, comorbidities, and physical activity rather than relying on continuous data streams from WDs or environmental sensors.

The proposed HCARS combines CF and CBF techniques with ML models to generate ranked recommendations that are sensitive to patient-specific characteristics while remaining compatible with the constraints of the underlying dataset. Advanced algorithms will be employed to address data sparsity and to enhance predictive accuracy, but always within the limits imposed by static, tabular data. This design choice allows the thesis to explore hybrid, context-informed recommendation logic in a controlled environment, while framing truly real-time, multi-modal systems as a direction for future work rather than as an achievement of the present study.

In this way, the thesis both builds upon and differentiates itself from previous studies. It adopts hybrid and context-aware concepts from the broader HRS literature, anchors them in a concrete and reproducible offline setting, and explicitly documents the gap between what is implemented in practice and what would be required for fully real-time, sensor-integrated hypertension management.

2.5 Conceptual Framework

2.5.1 Advancing Hypertension Management with a Hybrid Context-Aware Recommender System

Building upon the critique and synthesis of previous studies, this thesis proposes an HCARS specifically tailored for hypertension management and grounded in the constraints and opportunities of the KHD (Chuks, 2022). The conceptual framework is designed to address identified gaps such as limited integration of contextual information and the need for hybrid recommendation logic within a controlled, offline research environment.

The proposed HCARS ingests structured data from the KHD, including demographics, clinical indicators, and lifestyle-related variables. Using modular Python pipelines, the system performs data preprocessing, feature engineering, and model training, while a research-oriented Streamlit dashboard supports exploratory analysis and visualization. Within this architecture, both CBF and CF components are implemented: CBF relies on item (recommendation option) attributes derived from clinical and lifestyle features, whereas CF exploits similarity patterns between patient profiles to propagate preferences and outcomes across similar individuals.

To mitigate cold-start and sparsity issues, the system leverages global patterns in the KHD such as typical risk profiles and common treatment-related patterns as initial anchors for patients with limited individual histories. Clustering methods can be used to identify subgroups of patients with similar clinical and lifestyle characteristics, providing a basis for transferring recommendations across similar cases. Supervised ML models, such as decision trees and random forests, can further estimate risk levels or outcome likelihoods that inform the ranking of recommended options.

Context-awareness in this framework is realized through the use of clinically meaningful, static contextual features rather than live sensor streams. Variables such as age category, BMI range, smoking status, physical activity level, and comorbid conditions are treated as contextual dimensions that shape the relevance of different recommendations. By explicitly encoding these contextual features into the representation of both patients and recommendation targets, the HCARS approximates context-aware behavior in an offline setting: recommendations are adapted to distinct patient contexts as captured in the KHD, even though no real-time updates are consumed.

Evaluation of the HCARS follows an offline, batch-oriented protocol. The dataset is partitioned into training, validation, and testing subsets, and the quality of recommendations is assessed using ranking-oriented metrics such as Precision@k, Recall@k, MAP@k, and nDCG@k, alongside classical classification metrics when appropriate. This evaluation strategy aligns with the broader RS literature while preserving a clear distinction between offline experimental performance and the more complex requirements of real-world clinical deployment.

The significance of this conceptual framework lies in its combination of:

- hybrid recommendation strategies (CF & CBF) tailored to hypertension-related variables;
- explicit use of static contextual features derived from the KHD to approximate context-aware reasoning; and
- a transparent, modular implementation that can be extended in future work to incorporate richer data sources, including longitudinal and real-time signals.

While the current thesis does not implement full real-time integration of WDs or environmental sensors, the architectural choices and modeling strategies are intentionally designed to be extensible. Future systems could plug additional data streams into the same conceptual backbone, thereby transforming the offline HCARS developed here into a foundation for more advanced, continuously updated hypertension management tools. In this sense, the present work serves both as a concrete instantiation of hybrid, context-informed recommendations on a real-world dataset and as a methodological stepping stone toward clinically integrated, multi-modal HRS for chronic disease management.

Chapter Three: Research Methodology

3.1 Introduction

As introduced in Chapter 1 and theoretically grounded in Chapter 2, this chapter presents the **practical implementation of the HCARS** and explains how the conceptual framework is translated into modular Python pipelines and executable components. The overarching aim is to describe the end-to-end workflow of the system, including the structure of the preprocessing, context, and recommendation modules, as well as the implementation details of the CF, CBF, and hybrid models, in a way that enables methodological transparency and reproducibility.

Hypertension poses a significant challenge to global health, necessitating innovative and personalized management strategies. Against this backdrop, the present chapter outlines the research methodology adopted to develop the HCARS as a **context-aware hybrid recommender** tailored to hypertension management. The methodological choices are rigorously structured to ensure robustness, validity, and reliability, and are closely aligned with the study’s objective of designing an effective, data-driven decision-support framework implemented within a modular Python project and a Streamlit-based research interface.

The chapter first presents the overall research design, emphasizing the quantitative and computational orientation of the study and the rationale for adopting an **offline, batch-oriented evaluation scheme**. It then outlines the population and sample, with a focus on the selection and preparation of the KHD as the empirical basis for the system, followed by a concise description of the main data preprocessing and feature engineering steps that underpin the contextualized recommendation logic.

Subsequently, the chapter summarizes the implementation of the hybrid recommendation techniques that integrate CBF and CF to enhance personalization and alleviate sparse data and cold-start issues, and it introduces the model evaluation and validation strategy based on ranking-oriented metrics suitable for RS. The tools and computational environment are briefly described, highlighting the Python ecosystem and the use of Streamlit with modular pipelines as the primary vehicle for running and inspecting the system, with Jupyter notebooks reserved for exploratory analysis. Moreover, the implementation discussion emphasizes reproducible execution through deterministic configurations, clear module interfaces, and structured logging to trace data transformations and recommendation decisions. These practices collectively strengthen the methodological credibility of the system by enabling systematic debugging, transparent reporting, and consistent comparison across experimental settings.

3.2 Research Design

3.2.1 Methodological Approach

This study adopts a quantitative and computational framework, leveraging data science and ML techniques to develop the HCARS. By utilizing a data-driven approach, the research systematically analyzes comprehensive health data to uncover patterns and relationships pertinent to hypertension management. The quantitative methodology facilitates objective measurement and statistical analysis, ensuring that the findings are both reliable and valid.

Central to this approach is the integration of advanced ML paradigms. Specifically, the study employs a combination of supervised and unsupervised learning algorithms to process and interpret the complex, high-dimensional health data sourced from the KHD. These algorithms enable the HCARS to generate personalized recommendations by identifying significant predictors and clustering similar patient profiles. **Implementation, testing, and iteration are performed as reproducible Python pipelines, with results rendered via a Streamlit front-end; notebooks may be used for exploratory data analysis but are not the primary execution environment.**

3.2.2 Justification for the Selected Methodology

The selected methodology is tightly aligned with the primary objective of developing an effective HCARS for hypertension management. By leveraging machine-learning techniques on structured health data, the study can systematically process and analyse large volumes of information, identify clinically meaningful patterns, and derive reproducible recommendation rules. This choice ensures that the research questions are addressed through an evidence-based workflow that supports statistical validation and comparative evaluation of competing models.

As detailed in Chapter 2, the HCARS builds on a hybrid, context-aware recommender paradigm that combines content-based and collaborative signals. In the present methodological setting, this hybridization is exploited to alleviate data sparsity and cold-start effects in the hypertension dataset, while still allowing recommendations to reflect both patient-specific attributes and cross patient similarity structures. Contextual features derived from demographics and lifestyle related variables further constrain and refine the recommendation space, ensuring that proposed options remain consistent with the clinical profile of each patient rather than being purely data-driven artefacts.

This integrated approach is methodologically justified on three fronts: it matches the tabular, heterogeneous nature of the KHD; it addresses known weaknesses of conventional healthcare recommenders (e.g., sparsity and limited personalization); and it provides a scalable basis for offline experimentation and evaluation.

3.2.3 Population & Sample

A publicly available secondary dataset (derived from the BRFSS health survey) was used; therefore, no primary data collection, researcher-driven sampling, or participant recruitment was conducted. In this secondary-data setting, the dataset population refers to all survey records available in the selected file (hypertension_data.csv) within the KHD, as defined by its variables and labeling schema. The effective (analytical) sample refers to the subset of records retained after preprocessing and the application of inclusion/exclusion criteria (Section 3.3.3), is subsequently used for model training, validation, evaluation.

3.3 Dataset Description

3.3.1 Kaggle Health Dataset

The empirical analysis in this thesis is based on the KHD, a publicly available collection of tabular cardiovascular records (Chuks, 2022). As outlined in Chapter 1, the original KHD comprises separate files for hypertension, diabetes, and stroke. For reasons of methodological clarity and to avoid unreliable record linkage, the present study uses only the hypertension_data.csv file as the basis for model training, validation, and recommendation generation.

3.3.1.1 Hypertension Dataset

The hypertension_data.csv file contains cardiovascular-related predictors commonly used in clinical risk stratification, including blood pressure, cholesterol, electrocardiographic findings, and standard covariates. A detailed description of all variables, their types, and their clinical interpretation is provided in Table 3.1.

Table 3.1: Feature Description of the Hypertension Dataset

Feature Name	Type	Description
Age	Continuous	Age in years (integer per patient; modeled as continuous across the population).
Sex	Categorical	Gender of the patient (Male/Female).
CP	Categorical	Chest pain type (0: typical angina, 1: atypical angina, 2: non-anginal pain, 3: asymptomatic).
Resting BP	Continuous	Resting blood pressure (mm Hg), continuous.
Cholesterol	Continuous	Serum cholesterol in mg/dl (modeled as continuous).

Feature Name	Type	Description
FBS	Binary	Blood sugar > 120 mg/dl (1 = true, 0 = false).
RestECG	Categorical	Resting ECG results (0: normal, 1: ST-T abnormality, 2: left ventricular hypertrophy).
Thalach	Continuous	Maximum heart rate achieved (bpm).
Exang	Binary	Exercise-induced angina (1 = yes, 0 = no).
Oldpeak	Continuous	Exercise-induced ST depression.
Slope	Ordinal Categorical	Slope of the peak exercise ST segment (0: upsloping, 1: flat, 2: downsloping).
CA	Discrete (0–3)	Number of major vessels colored by fluoroscopy.
Thal	Categorical	Thallium stress test result (3: normal, 6: fixed defect, 7: reversible defect).
Target	Binary	Hypertension diagnosis (1 = yes, 0 = no).

3.3.2 Dataset Footprint and Processing Stages

Processing summary. To document the data preparation workflow for reproducibility and transparency, Table 3.2 summarises the dataset footprint at each processing stage. The raw hypertension_data.csv file contained **26,083** records and **14** variables. After applying preprocessing and cleaning, the analytical cohort comprised **26,058** patient records, while the feature space expanded to **52** variables due to encoding and derived attributes. Subsequent contextual feature engineering further enriched each record to approximately **66** variables (including bp_category, chol_category, risk_level, age_group, and multi-label indicators), without removing any additional patient records. Overall, the pipeline performs light record consolidation followed by systematic feature enrichment to support downstream modelling.

Table 3.2: Dataset footprint across preprocessing and contextualisation.

Dataset	Rows	Columns
Raw	26083	14
Cleaned	26058	20
Contextualized	26058	38

3.3.2.1 Relevance to the Study

Within these boundaries, the hypertension subset of the KHD provides a sufficiently rich and internally consistent basis for developing and testing the HCARS, enabling the extraction of clinically meaningful patterns that support personalized hypertension management in an offline, research oriented setting.

3.3.3 Inclusion and Exclusion Criteria

To ensure that the underlying data are suitable for contextual feature engineering and model training, explicit inclusion and exclusion criteria were applied.

3.3.3.1 Inclusion Criteria

- Records containing complete BP measurements & associated core clinical features.
- Patients diagnosed with hypertension or exhibiting major cardiovascular risk factor.
- Data entries with demographic and lifestyle information sufficient to derive contextual features.

3.3.3.2 Exclusion Criteria

- Records with missing or inconsistent key variables required for model training or evaluation.

3.4 Data Preprocessing and Preparation

The preprocessing pipeline of the HCARS system is implemented as a modular and sequential framework that ensures data integrity, reproducibility, and traceability. Each preprocessing stage corresponds directly to a dedicated Python module, executed and verified through the debugging orchestrator (`debug_preprocessing_pipeline.py`). The sequential order of operations is as follows:

- `data_cleaning.py` → Schema validation, removal of duplicates and empty columns, and enforcement of numeric types.
- `imputation.py` → Application of three imputation strategies (Simple, KNN, Iterative/XGBRegressor) after the data split to avoid leakage.
- `encoding.py` → Transformation of categorical variables via One-Hot and Ordinal encoding, preserving protected clinical columns.
- `integration.py` → Integration of processed outputs, schema verification, and consistency checks.

- `debug_preprocessing_pipeline.py` → Sequential execution of all modules, generation of intermediary CSVs, and automatic logging for transparency and auditability.

3.4.1 Data Cleaning

- **Schema and column standardization** Input data are validated against a non-contextual schema to ensure that all required fields and data types are present. Column names are standardized to lowercase, and leading or trailing spaces are removed. Clinically critical columns such as `age`, `trestbps`, and `chol` are explicitly verified.
- **Duplicate and empty column handling** Duplicate rows are removed, while columns with all missing values are dropped early to reduce dimensional noise. The target variable is always preserved.
- **High-Missingness columns** Columns with a missingness rate exceeding a defined threshold are excluded to prevent model bias. The ratio and the affected columns are recorded in logs for reproducibility.
- **Type Enforcement for Clinical Numerics:** Core medical fields are coerced into numeric formats; invalid or non-numeric entries are replaced with NaN and logged to enable inspection.
- **No leakage by design** Cleaning operations are strictly non-learning. Imputation and encoding are deferred to post-split stages, guaranteeing that no statistical information leaks from the test set into training data.

3.4.2 Data Transformation

- **Imputation strategy (Post-Split):** Missing values are handled after the train/test split through three complementary strategies:
 1. Simple imputation: uses central tendency (mean/median/mode) for rapid and stable replacement of missing values in low-variance features.
 2. KNN imputation: reconstructs missing values based on feature similarity, preserving local correlations in the data.
 3. Iterative imputation with XGBRegressor: Employs model-based estimation for complex inter-feature dependencies. This method iteratively predicts missing values using gradient boosting, achieving probabilistic consistency among clinical features.

These strategies are selectively applied depending on data type and clinical importance, ensuring that imputed values are both statistically coherent and medically plausible.

- **Outlier and scaling management:** for numeric attributes, RobustScaler is used instead of standard z-score normalization because clinical measurements often contain outliers (e.g., extreme BP or cholesterol levels). Robust scaling centers data using the median and scales it according to the interquartile range, providing stability against non-Gaussian distributions.
- **Categorical encoding:** categorical variables are transformed using a modular ColumnTransformer:
 - Binary features are ordinal-encoded with explicit mappings for interpretability and consistency.
 - Multi-class categorical variables are one-hot encoded with optional first-level dropping to mitigate multicollinearity.
 - Numeric features pass through unchanged unless they are explicitly subject to imputation or scaling.
- **Protected clinical columns:** key medical variables (age, sex, trestbps, chol, thalach, oldpeak, etc.) are safeguarded during transformations to ensure full retention in all pipeline outputs.
- **Context and risk feature hook:** a dedicated non-leaking mechanism allows safe integration of derived clinical scores (e.g., risk_score) during post-split transformation without contaminating model training.
- **Logging and revalidation:** after transformation, schemas are revalidated and logs are automatically generated via debug_preprocessing_pipeline.py, capturing all parameters, decisions, and transformations. This supports transparency, error tracing, and experiment reproducibility.

3.4.3 Feature Engineering

The feature engineering phase in HCARS represents the bridge between data preprocessing and the clinical reasoning embedded in the recommendation pipelines. It introduces contextual intelligence transforming raw clinical values into medically interpretable categories and risk-level indicators that directly inform personalized recommendations. This process is modular and sequentially organized across multiple files under the context/ directory, ensuring clarity, auditability, and strict logical flow.

- `schema_validator.py` → Validates the schema of cleaned data, enforcing the presence of essential attributes (e.g., age, sex, trestbps, chol, thalach) and verifying type integrity before context derivation.
- `clinical_features.py` → Computes medically meaningful categorical transformations such as BP categories, cholesterol categories, and composite risk levels based on clinical thresholds.
- `context_validation.py` → Verifies internal consistency among derived features, ensuring that clinical relationships (e.g., BP vs. risk level) remain physiologically coherent.
- `feature_engineering.py` → Centralizes the execution of all clinical feature computations, invoking the submodules for schema validation, categorical derivation, and contextual routing.
- `router.py` → Directs each record through appropriate context routes (e.g., fallback, clinical, or hybrid paths) depending on data completeness and contextual richness.
- `fallback.py` → Provides a rule-based safety mechanism when contextual data are insufficient, generating conservative recommendations based on population-level averages and essential demographics.
- `core.py` → Integrates all contextual outputs into a unified feature frame and exports them to downstream pipelines (CBF, CF, Hybrid), maintaining synchronization of derived columns.
- `debug_context_modules.py` → Executes and logs the full context pipeline sequentially for debugging and transparency, generating validation reports and intermediate CSV outputs.

Logic and Derivation of Contextual Columns

Each contextual feature encapsulates a medically grounded abstraction derived from base variables. The transformations are deterministic, interpretable, and designed to capture clinically relevant variability while minimizing noise. Table 3.3 summarizes the key contextual columns, their source variables, logic, and purpose.

Table 3.3: Derived Contextual Features in HCARS

Contextual Feature	Source Variable(s)	Derivation Logic	Clinical / Analytical Purpose
bp_category	trestbps (Resting BP)	Categorized into {Normal, Elevated, Hypertension Stage 1, Stage 2} based on AHA thresholds (e.g., systolic cutoffs at 120/140 mmHg).	Encodes cardiovascular strain level and primary hypertension risk class.
chol_category	chol (Serum Cholesterol)	Discretized into {Desirable (< 200), Borderline (200–239), High (≥ 240) mg/dL}.	Captures hypercholesterolemia risk influencing lifestyle or drug recommendations.
risk_level	Composite of bp_category, chol_category, age, sex, fbs, and cp.	Computed via a weighted rule matrix calibrated from clinical literature and dataset patterns.	Represents overall cardiovascular risk used to personalize therapeutic intensity.
age_group	age	Binned into {Young, Middle-aged, Senior} via quantile-based segmentation (e.g., Q1=40, Q2=55, Q3=65).	Provides demographic context for differential treatment recommendations.
fbs_cat	fbs (Fasting Blood Sugar)	Binary indicator (1 if > 120 mg/dL).	Identifies metabolic comorbidities impacting risk assessment.

Continued on next page

Table 3.3 – continued from previous page

Contextual Feature	Source Variable(s)	Derivation Logic	Clinical / Analytical Purpose
cp_cat	cp (Chest Pain Type)	Encoded as {Typical, Atypical, Non-anginal, Asymptomatic}.	Reflects symptomatic presentation relevant to cardiac status.
context_key	Concatenation of bp_category, chol_category, and risk_level.	Unique string identifier for patient's composite context.	Enables fast mapping via the CONTEXT_TO_RECOMMENDATION_MAP-
context_validity	Derived in context_validation.py	Checks consistency (e.g., if risk level contradicts BP/Chol category).	Ensures logical validity and triggers fallback if inconsistencies occur.
context_source	Generated in router.py	Indicates the origin of context derivation: clinical, fallback, or hybrid.	Improves traceability and model interpretability for clinical audits.

Purpose and Clinical Rationale

The contextual feature set serves as the core of the system's personalization logic. Instead of treating patients purely as statistical vectors, HCARS embeds each record within a clinically interpretable space that mirrors real diagnostic reasoning. By converting raw signals (e.g., numeric blood pressure) into contextual abstractions (e.g., Hypertension Stage 2), the system enables models such as CBF and CF to reason in medical terms producing recommendations that align with established clinical guidelines.

Error handling and fallback logic when contextual features cannot be reliably derived (e.g., missing cholesterol or anomalous risk combinations), system dynamically routes data to the fallback engine. This ensures that no patient record fails the recommendation process; instead, it reverts to a conservative, demographically based recommendation while logging the event for transparency.

Traceability and validation all derived context columns, validation outputs, and routing decisions are logged and exported via debug_context_modules.py. Each pipeline step produces console diagnostics and CSV artefacts, ensuring auditability and reproducibility of the contextual transformation process.

3.5 Pipeline Overview and System Modules

This section summarizes the end-to-end HCARS pipeline as a sequence of reproducible modules, clarifying how data flow and intermediate artefacts connect the methodology components presented above. All stages are parameterized via pipeline/config.yaml and write immutable CSV/JSON artefacts to disk for auditability.

3.5.1 Stage 1: Data Preparation

Raw hypertension records are ingested and processed through a modular preprocessing pipeline that performs schema validation, duplicate removal, high-missingness filtering, post-split imputation (Simple, KNN, and Iterative/XGBRegressor), categorical encoding (One-Hot and Ordinal), and robust scaling of numeric attributes. Execution is orchestrated by preprocessing/debug_preprocessing_pipeline.py, which logs all steps and persists the curated output as:

outputs/tmp/df_cleaned.csv

This file serves as the sole input to the contextual feature engineering stage.

3.5.2 Stage 2: Context Feature Engineering

Contextual features (e.g., risk groupings derived from BP and cholesterol, age-group bins, and composite risk scores) are constructed and audited for range/sparsity/consistency. The stage is executed by context/debug_context_modules.py, which coordinates context/clinical_features.py, context/router.py, context/context_validation.py, and related utilities. The enriched dataset is exported as:

outputs/tmp/df_contextualized.csv

This contextualized table is the standardized input for recommenders (CBF, CF, Hybrid).

3.5.3 Stage 3: Recommendation Pipelines (CBF, CF, Hybrid)

Each recommender runs as an independent, reproducible pipeline under pipeline/, reading outputs/tmp/df_contextualized.csv and writing its own artefacts.

CBF. Implemented in recommender/cbf.py and orchestrated by pipeline/cbf_pipeline.py. Patient/item profiles are built from demographics, clinical metrics, and contextualized attributes; cosine similarity is computed and then calibrated/normalized. Representative outputs:

- outputs/CBF/df_cbf_recommendations.csv

- outputs/CBF/df_cbf_recommendations_eval.csv
- outputs/CBF/cbf_eval_full.csv, cbf_eval_loco.csv, cbf_eval_unique.csv
- Diagnostics: cbf_errors_explained.csv, context_combinations_not_covered.csv, df_cbf_recommendations_full.csv, cbf_recs_loco.csv

CF. Implemented in recommender/cf.py with cold-start logic in recommender/cold_start_handler.py; orchestrated by pipeline/cf_pipeline.py. A user–item matrix is constructed and patient–patient similarity (e.g., Pearson) is applied. Outputs include:

- outputs/CF/df_cf_recommendations.csv
- outputs/CF/df_cf_recommendations_eval.csv
- outputs/CF/df_cf_recommendations_eval_loco.csv, _eval_loco_family.csv

Hybrid. Fusion logic in recommender/hybrid.py; orchestration in pipeline/hybrid_pipeline.py. Normalized CBF/CF scores are combined using a weighted policy and safety gates configured in pipeline/config.yaml. Outputs:

- outputs/df_hybrid_recommendations.csv, df_hybrid_recommendations.min.csv
- outputs/hybrid_eval_bridge.csv, hybrid_eval_summary.json, hybrid_quality_report.json

3.5.4 Stage 4: Evaluation Export and Visualization

Each pipeline computes ranking metrics (Precision@k, Recall@k, MAP@k, nDCG@k) and exports results as CSV/JSON for transparent inspection:

- CF metrics: utils/eval_metrics.py.
- CBF metrics: utils/evaluation.py.
- Hybrid metrics: integrated within pipeline/hybrid_pipeline.py.

The Streamlit interface (main.py) provides a read-only dashboard that visualizes both recommendation lists and evaluation tables directly from the persisted artifacts, without altering the underlying computations.

To provide a comprehensive summary of the data flow and module interactions described in Stages 3.5.1–3.5.4, Table 3.4 presents an integrated overview of all pipeline stages, their corresponding scripts, core responsibilities, and primary outputs. This table consolidates the modular logic described above into a single visual summary, clarifying how each processing stage feeds into the next within the HCARS architecture.

Table 3.4: End-to-end dataflow summary across stages and modules

Stage	Executed Script	Core Modules / Responsibilities	Primary Output(s)
Stage 1: Data Preparation	preprocessing/debug_preprocessing_pipeline.py	Cleaning, duplicate handling, imputation, encoding, scaling; logging and schema checks.	outputs/tmp/df_cleaned.csv
Stage 2: Context Engineering	context/debug_context_modules.py	Derive contextual features; validate consistency; route/fallback; audit ranges/sparsity.	outputs/tmp/df_contextualized.csv
Stage 3: CBF	pipeline/cbf_pipeline.py	Build vectors; cosine similarity; calibration/normalization; explanations.	outputs/CBF/df_cbf_recommendations.csv, df_cbf_recommendations_eval.csv, diagnostics
Stage 3: CF	pipeline/cf_pipeline.py	User-item matrix; similarity (e.g., Pearson); cold-start handler; explanations.	outputs/CF/df_cf_recommendations.csv, df_cf_recommendations_eval.csv
Stage 3: Hybrid	pipeline/hybrid_pipeline.py	Normalize per-model scores; fuse with weights/gates; finalize ranks.	outputs/Hybrid/df_hybrid_recommendations.csv, .json
Stage 4: Evaluation & Visualization	(integrated per pipeline)	Ranking metrics export; Streamlit renders read-only tables/plots from artefacts.	CSV/JSON metrics per pipeline; Streamlit dashboard

3.5.4.1 Evaluation Metrics and Theoretical Foundation

Definition of $@k$. In ranking-based recommender evaluation, the notation “ $@k$ ” specifies that the metric is calculated over the first k ranked items in each recommendation list the top- k cutoff. Formally, for each patient u , let $R_u@k$ denote the ordered set of the k highest-ranked recommendations produced by the system, and G_u denote the set of ground-truth relevant items (i.e., true clinical responses or verified interventions). The value of k determines the evaluation depth: here, $k=5$ as defined in evaluation.k_at within config.yaml, reflecting the number of top clinical recommendations that a physician would

realistically review in a decision-support context. This configuration is applied globally across CF, CBF, and Hybrid pipelines for consistent performance comparison.

Precision@k. Precision@k measures the proportion of recommended items among the top k that are clinically correct or relevant (Manning et al., 2008; Ricci et al., 2022):

$$P@k(u) = \frac{|R_u@k \cap G_u|}{k}. \quad (3.1)$$

In implementation terms, this metric is computed in `eval_metrics.py` and `evaluation.py`, using the exported prediction scores (`score_cbf`, `score_cf`, or `hybrid_score`) and the binary field `true_response` as relevance ground truth. High Precision@k indicates that the model effectively filters out irrelevant interventions, a property crucial for medical RS where false positives could mislead clinical interpretation.

Recall@k. Recall@k quantifies the proportion of all clinically relevant items that appear within the top k recommendations (Manning et al., 2008; Ricci et al., 2022):

$$R@k(u) = \frac{|R_u@k \cap G_u|}{|G_u|}. \quad (3.2)$$

This metric captures the model’s ability to cover the true set of appropriate interventions. In HCARS, high recall ensures that important treatment options (e.g., antihypertensive agents or lifestyle changes) are not omitted from the top recommendation subset.

MAP@k (Mean Average Precision). Average Precision (AP@k) integrates the ranking order by averaging the precision at each position $i \leq k$ where a relevant item occurs (Manning et al., 2008; Voorhees, 2001):

$$AP@k(u) = \frac{1}{\min(|G_u|, k)} \sum_{i=1}^k P@i(u) \cdot \text{rel}(i). \quad (3.3)$$

The overall MAP@k is the mean of AP@k across all patients (Manning et al., 2008; Ricci et al., 2022):

$$MAP@k = \frac{1}{|U|} \sum_{u \in U} AP@k(u). \quad (3.4)$$

This metric, implemented in both `eval_metrics.py` and `evaluation.py`, penalizes systems that place correct recommendations lower in the ranking. A higher MAP@5 therefore indicates that relevant interventions appear earlier in the list an essential property in clinical decision-support systems, where the top few recommendations are the most actionable.

nDCG@k (Normalized Discounted Cumulative Gain). nDCG@k extends evaluation to account for the graded relevance and position of items (Järvelin & Kekäläinen, 2002;

Ricci et al., 2022). It computes the cumulative gain (DCG) with logarithmic discounting, emphasizing higher-ranked items:

$$\text{DCG}@k(u) = \sum_{i=1}^k \frac{2^{\text{rel}(i)} - 1}{\log_2(i + 1)}, \quad \text{nDCG}@k(u) = \frac{\text{DCG}@k(u)}{\text{IDCG}@k(u)}. \quad (3.5)$$

The normalization by $\text{IDCG}@k(u)$ ensures that $\text{nDCG}@k$ lies within $[0,1]$, where 1.0 corresponds to an ideal ranking. This metric was implemented in the HCARS evaluation scripts to capture not only correctness but also the order of recommendations, reinforcing ranking stability and calibration quality. Since the configuration enforces score normalization via minmax scaling and isotonic calibration (see `config.yaml` under `score_calibration` and `scoring.cbf_mix`), $\text{nDCG}@k$ becomes particularly sensitive to ranking consistency and alignment between predicted and observed relevance.

Implementation and impact in HCARS The evaluation block in `config.yaml` activates the four ranking metrics under the `evaluation.metrics.ranking` node with $k=5$, using the binary field `true_response` as relevance ground truth. Each pipeline (`cbf_pipeline.py`, `cf_pipeline.py`, and `hybrid_pipeline.py`) computes these metrics via shared helper interfaces and exports per-split results to structured CSV files, which are visualized in `main.py` through the Streamlit dashboard, ensuring end-to-end traceability.

Collectively, these metrics provide a quantitative validation of recommendation correctness and ranking quality. High $\text{Precision}@5$ and $\text{nDCG}@5$ indicate accurate top-ranked outputs with minimal false positives, while $\text{Recall}@5$ and $\text{MAP}@5$ reflect coverage and effective prioritization of relevant interventions, confirming that HCARS produces clinically meaningful and well-ordered recommendations.

3.6 Development of the Hybrid Context-Aware Recommender System

3.6.1 System Overview

This section provides a high-level overview of the proposed HCARS, outlining its main components, design philosophy, and interaction between the collaborative, content-based, and context-aware layers. The overview establishes the conceptual foundation upon which the detailed architectural design and implementation decisions are built.

3.6.1.1 Architectural Design

The HCARS is implemented as a modular and reproducible Python architecture that integrates three complementary recommendation paradigms CBF, CF and a Hybrid Context-Aware integration layer. Each paradigm operates as an independent pipeline while sharing a unified configuration and logging structure.

The architecture follows a layered and service-oriented design that separates data pre-processing, contextual feature engineering, model execution, and evaluation into distinct modules. This modularity allows each layer to be developed, debugged, and benchmarked independently, ensuring both robustness and transparency.

At its core, the system is orchestrated through modular Python pipelines:

- **Preprocessing and context layers** prepare and enrich the raw hypertension dataset into a structured, contextualized form using schema validation, imputation, encoding, and feature engineering modules.
- **Recommender layer** executes independent recommendation pipelines - CBF, CF, and Hybrid each producing ranked lists of recommended interventions for every patient record.
- **Evaluation and visualization layer** computes ranking metrics (Precision@k, Recall@k, MAP@k, nDCG@k) and exports results to outputs/ directories for visualization in Streamlit dashboards.

The architecture is implemented entirely in Python using the Pandas, NumPy, and Scikit-learn libraries, with Streamlit serving as the front-end visualization framework. Configuration parameters (paths, thresholds, weights, and evaluation settings) are centralized in the config.yaml file, enabling full experiment reproducibility. All intermediate results are logged automatically in dedicated log files (e.g., cbf_pipeline.log, cf_pipeline.log, and hybrid_pipeline.log) for traceability.

Figure 3.1 illustrates high-level architecture of HCARS system, showing the sequential interaction among preprocessing, contextualization, and recommendation pipelines, along with the evaluation and visualization layer that ensures traceable experimentation.

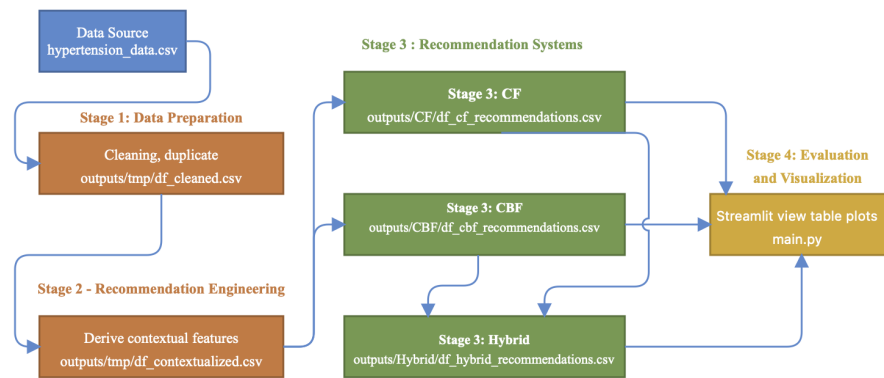


Figure 3.1: High-level architecture of the HCARS.

This modular design mirrors modern ML Operations principles, which integrate software engineering practices such as version control, continuous integration, and automated deployment into ML lifecycle. This alignment ensures the HCARS remains clinically interpretable, reproducible, and computationally reliable.

3.6.2 Implementation of Recommendation Techniques

3.6.2.1 Collaborative Filtering Implementation

The CF engine in HCARS is implemented as a production-grade subsystem that links statistical similarity modelling with contextual cardiovascular logic. Its behaviour emerges from the combined operations of `recommender/cf.py`, `pipeline/cf_pipeline.py`, the contextual fallback layer in `recommender/cold_start_handler.py`, and the clinical mapping rules defined in `utils/constants.py`. This design ensures that CF recommendations are both numerically grounded and clinically coherent. At a high level, the CF subsystem is designed to exploit inter-patient similarity patterns to generate personalized hypertension management recommendations while remaining aligned with clinically validated contextual mappings.

1. Patient and context representation each patient in the contextualized dataset (`outputs/tmp/df_contextualized.csv`) is uniquely indexed using the derived `patient_id` field a deterministic identifier composed from clinically stable attributes (e.g., sex, age group, and chest pain category). The corresponding utility in `core.py` guarantees:

- absence of information leakage across train/validation/test splits,
- repeatability of recommendations across runs, and
- consistent indexing for similarity computation in CF.

Alongside demographic data, the CF subsystem consumes the contextual features engineered in the previous phase `bp_category`, `chol_category`, `risk_level`, and `context_key` (or `context_feature_id`). These descriptors do not necessarily enter the similarity computation directly, but they guide cold-start handling and re-ranking through the clinically validated mapping `CONTEXT_TO_RECOMMENDATION_MAP` in `utils/constants.py`. In particular, the composite `context_key` acts as a high-level signature of a patient’s cardiovascular profile and links CF outputs to clinically relevant recommendation clusters when behavioural evidence is sparse.

Pseudocode: context representation (`cf_pipeline.py` + `cf.py`)

FOR each row `r` in `df_contextualized`:

```
patient_id ← generate_patient_id(r.sex, r.age_group, r.cp_cat)
context_key ← concat(r.bp_category, r.chol_category, r.risk_level)
store_mapping(patient_id, context_key)
```

2. User–Item matrix construction and similarity computation the core CF algorithm in `recommender/cf.py` constructs a user–item interaction matrix $R_{u,i}$, where rows correspond to patients and columns to clinical interventions (or recommendation features). For each patient u and item i , the system derives a relevance indicator from the field `true_respo`

nse and/or engineered interaction logic. Unobserved interactions are treated as unknown rather than zero, consistent with best practices in RS and avoiding bias against unobserved treatments.

The similarity metric is selected from `config.yaml` under the node `cf.similarity_metric`, which supports at least "cosine" and "pearson", both of which are widely used user–similarity measures in CF (Sgardelis et al., 2025). Let R_u and R_v denote the interaction vectors for patients u and v :

- Cosine similarity:

$$S(u, v) = \frac{R_u \cdot R_v}{\|R_u\| \|R_v\|} \quad (3.6)$$

- Pearson correlation:

$$S(u, v) = \frac{\sum_i (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_i (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_i (R_{v,i} - \bar{R}_v)^2}} \quad (3.7)$$

The neighbourhood size k (e.g., `cf.k_neighbors`) and additional thresholds (e.g., `cf.min_rating_threshold`) are also specified in `config.yaml`. The pairwise similarity matrix $S_{u,v}$ is cached to avoid recomputation across multiple recommendation calls and to support efficient large-scale evaluation.

Pseudocode: user–item matrix and similarity (`cf.py`)

$R \leftarrow \text{empty_matrix}(\text{num_patients}, \text{num_items})$

FOR each patient p :

 FOR each item i :

 IF `interaction_exists(p, i)`:

$R[p, i] \leftarrow \text{interaction_value}(p, i)$ # from `true_response` etc.

 ELSE:

$R[p, i] \leftarrow \text{NaN}$ # unknown, not negative

Compute similarity

IF `similarity_metric == "pearson"`:

$S \leftarrow \text{pearson_similarity}(R)$

ELSE IF `similarity_metric == "cosine"`:

$S \leftarrow \text{cosine_similarity}(R)$

`cache_similarity(S)`

3. CF scoring, calibration, and context-aware re-ranking given the similarity matrix S and the interaction matrix R , the CF subsystem predicts a score $\hat{R}_{u,i}$ for each unseen item i for patient u using a standard neighbourhood-based prediction rule over the top- k neighbours $N_k(u)$, as commonly adopted in neighbourhood-based collaborative filtering models (Herlocker et al., 2002; Ricci et al., 2011):

$$\hat{R}_{u,i} = \frac{\sum_{v \in N_k(u)} S(u, v) R_{v,i}}{\sum_{v \in N_k(u)} |S(u, v)|}. \quad (3.8)$$

These raw estimates are then transformed into a normalized CF score `score_cf`.

Score calibration is configured under the `cf.score_calibration` block in `config.yaml` (e.g., `method: "minmax"`). For each patient u , min-max normalization is applied, a standard normalization technique widely used in recommender systems and machine learning to ensure score comparability across users (Aggarwal, 2016a; J. Han et al., 2012):

$$\text{score_cf}(u, i) = \frac{\hat{R}_{u,i} - \min_j \hat{R}_{u,j}}{\max_j \hat{R}_{u,j} - \min_j \hat{R}_{u,j} + \varepsilon}, \quad (3.9)$$

where ε is a small constant for numerical stability. This calibration improves comparability of scores across patients and stabilizes ranking metrics such as nDCG.

A context-aware re-ranking step is activated through `cf.context_rerank.enabled`. In `config.yaml`. When enabled, the final score is computed as a convex combination of the CF score and a context-driven boost term, following established score fusion and context-aware re-ranking strategies in recommender systems (Adomavicius et al., 2011; Baltrunas et al., 2011):

$$\text{score}_{u,i}^{\text{final}} = \alpha \cdot \text{score_cf}(u, i) + (1 - \alpha) \cdot \text{context_boost}(u, i), \quad (3.10)$$

where $\alpha \in [0, 1]$ (e.g., $\alpha=0.85$) and `context_boost(u, i)` reflects alignment between the intervention and the patient's `context_key` or `context_feature_id` according to `CONTEXT_TO_RECOMMENDATION_MAP`.

Pseudocode: scoring and calibration (`cf.py` + `helpers.py`)

FOR each patient u :

`neighbors` $\leftarrow$ `top_k_neighbors(S, u, k_neighbors)`

FOR each candidate item i :

raw CF estimate

`r_hat[u, i]` $\leftarrow$ `weighted_sum_over_neighbors(neighbors, R[:, i])`

Min-max calibration over items for patient u

`min_r` $\leftarrow$ `min_j r_hat[u, j]`

`max_r` $\leftarrow$ `max_j r_hat[u, j]`

FOR each candidate item i :

```

score_cf[u, i] ← (r_hat[u, i] - min_r) / (max_r - min_r + epsilon)
IF context_rerank_enabled:
    ctx_boost ← compute_context_boost(u, i,
    CONTEXT_TO_RECOMMENDATION_MAP)
    score_final[u, i] ← alpha * score_cf[u, i] + (1 - alpha) * ctx_boost
ELSE:
    score_final[u, i] ← score_cf[u, i]
reason, explanation ← build_explanation(u, i, score_final[u, i])
append_to_recommendations(u, i, score_final[u, i], reason, explanation)

```

4. Cold-Start Resolution and Contextual Fallback. The recommender/cold_start_handler.py module handles cases where a patient or item has no usable interaction history. For a new patient lacking neighbours, the handler uses the contextual dictionary CONTEXT_TO_RECOMMENDATION_MAP:

context_key → {clinically acceptable recommendation features},

providing a clinically safe baseline set of recommendations. A configurable fallback score (e.g., cold_start.fallback_score_cf = 0.03) is attached, and all such recommendations are tagged with source = "Cold-Start" and status = "cold_start_safe_default".

Similarly, for cold items, the handler can rely on content-based information and contextual proximity to generate non-zero score_cf values that are later integrated into the hybrid pipeline. This context-driven fallback ensures that every patient receives at least one clinically meaningful recommendation, even in the absence of behavioural history, aligning system behaviour with medical reasoning rather than purely statistical similarity.

Pseudocode: cold-start handling (cold_start_handler.py)

FUNCTION handle_cold_start(patient_id, context_key):

```

IF patient_id not in R:
    allowed_features ← CONTEXT_TO_RECOMMENDATION_MAP[context_key]
    recs ← []
    FOR each feature_id in allowed_features:
        recs.append({
            "patient_id": patient_id,
            "feature_id": feature_id,
            "score_cf": fallback_score_cf,
            "source": "Cold-Start",
            "status": "cold_start_safe_default" })
    RETURN recs

```

5. Pipeline Orchestration and Output Artefacts. The full CF workflow is orchestrated by `pipeline/cf_pipeline.py`. The module performs:

1. Loading the contextualized dataset from `outputs/tmp/df_contextualized.csv`.
2. Performing splitting into train/validation/test via `pipeline/core.py`.
3. Constructing the user–item matrix R and computing the similarity matrix S .
4. Invoking the CF scorer and the cold-start handler where needed.
5. Applying calibration and context-aware re-ranking.
6. Writing the final recommendations to `outputs/CF/df_cf_recommendations.csv` using a schema such as:

```
["patient_id", "feature_id", "score_cf", "reason", "explanation", "source", "status"].
```

Textual explanations (reason, explanation) are generated by `helpers.py` to align numerical CF scores with clinically interpretable narratives. Intermediate artefacts and logs are maintained to guarantee transparency and reproducibility, while key configuration options such as neighbourhood size and similarity thresholds are centrally controlled through `config.yaml`.

Pseudocode: pipeline execution (`cf_pipeline.py`)

```
# Load contextualized data
df_ctx ← load_csv("outputs/tmp/df_contextualized.csv")
# Deterministic split
TRAIN, VAL, TEST ← split_with_core(df_ctx, config)
# Build user–item matrix from TRAIN
R ← build_user_item_matrix(TRAIN)
# Compute similarity matrix
S ← compute_similarity(R, config.cf.similarity_metric)
# Generate recommendations for all patients in TEST
FOR each patient u in TEST:
  IF is_cold_user(u, R):
    recs_u ← handle_cold_start(u.patient_id, u.context_key)
  ELSE:
    recs_u ← recommend_with_cf(u, R, S, config)
write_recommendations_to_csv(recs_u, "outputs/CF/df_cf_recommendations.csv")
```

6. Evaluation Framework and Metrics. The evaluation logic, implemented in `utils/eval_metrics.py` and invoked from `evaluation.py`, computes standard ranking-based evaluation metrics commonly used in recommender systems, namely Precision@k , Recall@k , MAP@k , and nDCG@k (Herlocker et al., 2004b; Järvelin & Kekäläinen, 2002). These metrics are computed for each split (train, validation, test). Let G_u denote the set of relevant items for patient u , and $R_u@k$ the top- k recommended items:

$$\text{P@k}(u) = \frac{|R_u@k \cap G_u|}{k}, \quad (3.11)$$

$$\text{R@k}(u) = \frac{|R_u@k \cap G_u|}{|G_u|}. \quad (3.12)$$

Average Precision at k for a single patient u is defined as:

$$\text{AP@k}(u) = \frac{1}{\min(|G_u|, k)} \sum_{i=1}^k \text{P@i}(u) \cdot \text{rel}(i), \quad (3.13)$$

and the Mean Average Precision is:

$$\text{MAP@k} = \frac{1}{|U|} \sum_{u \in U} \text{AP@k}(u). \quad (3.14)$$

Discounted Cumulative Gain (DCG) and its normalized variant (nDCG), which account for the position of relevant items in the ranked list, are computed as:

$$\text{DCG@k}(u) = \sum_{i=1}^k \frac{2^{\text{rel}(i)} - 1}{\log_2(i + 1)}, \quad (3.15)$$

$$\text{nDCG@k}(u) = \frac{\text{DCG@k}(u)}{\text{IDCG@k}(u)}. \quad (3.16)$$

The evaluation artefacts are exported to `outputs/CF/df_cf_recommendations_eval.csv` and subsequently visualized in the Streamlit application `main.py`, where users can inspect metric summaries alongside top- k CF recommendations and their explanations. The validation split is used to tune neighbourhood size and calibration settings, while the test split provides unbiased estimates of CF performance within HCARS.

Pseudocode: metric computation (`evaluation.py`)

FUNCTION `evaluate_cf(predictions, ground_truth, k)`:

$\text{Pk} \leftarrow \text{precision_at_k}(\text{predictions}, \text{ground_truth}, k)$

$\text{Rk} \leftarrow \text{recall_at_k}(\text{predictions}, \text{ground_truth}, k)$

$\text{MAPk} \leftarrow \text{map_at_k}(\text{predictions}, \text{ground_truth}, k)$

$\text{nDCGk} \leftarrow \text{ndcg_at_k}(\text{predictions}, \text{ground_truth}, k)$

```

RETURN {  "precision_at_k": Pk,
          "recall_at_k":   Rk,
          "map_at_k":      MAPk,
          "ndcg_at_k":     nDCGk }

```

By integrating mathematically grounded similarity functions, calibrated scoring, context-aware re-ranking, deterministic patient identity, clinically governed cold-start logic, and a rigorously defined evaluation framework, the CF subsystem of HCARS remains both technically reproducible and clinically interpretable. Every step from similarity computation and fallback routing to score shaping and ranking is explicitly anchored in the actual codebase, making the implementation verifiable and suitable for scientific scrutiny.

The CF subsystem is thus composed of interconnected modules that collectively manage data loading, similarity computation, cold-start resolution, explanation generation, and evaluation. Table 3.5 and Figure 3.2 summarize these interactions at the file and pipeline levels.

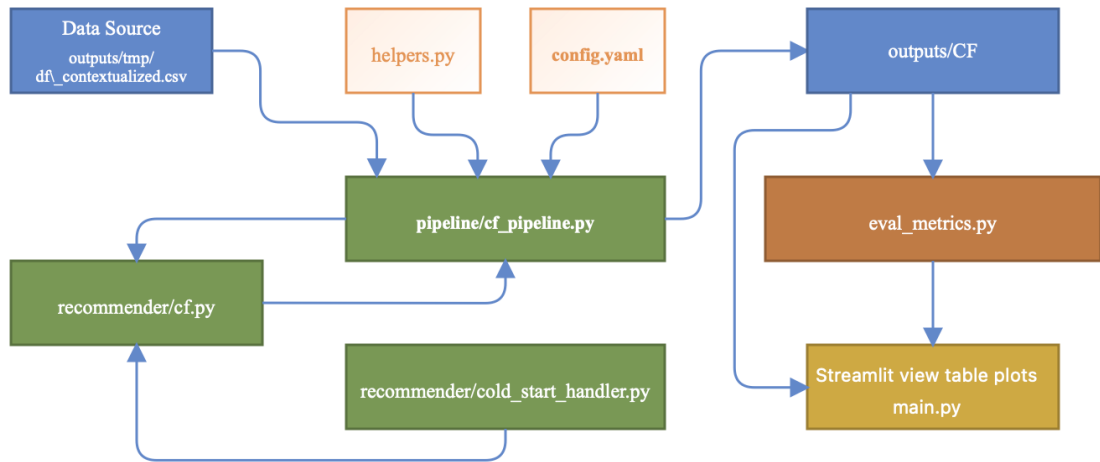


Figure 3.2: CF pipeline structure and module interaction in HCARS: contextualized data flow through configuration, cold-start handling, and output visualization.

This modular dependency structure ensures that CF logic in HCARS remains transparent, extensible, and verifiable. Each component is independently testable while remaining traceable through logged artefacts and YAML-driven configuration, enabling clinical interpretability and research reproducibility.

Table 3.5: CF Logic and File Dependencies in HCARS

Module / File	Primary Role	Core Operations / Logic	Outputs or Interactions
recommender/cf.py	Core CF algorithm implementation	Constructs the user–item matrix $R_{u,i}$, computes similarity (Pearson or cosine), and predicts scores for unseen items.	Produces raw CF scores for each patient–item pair.
recommender/cold_start_handler.py	Cold-start and fallback management	Uses contextual features (bp_category, chol_category, risk_level) and CONTEXT_TO_RECOMMENDATION_MAP to infer recommendations for new users/items.	Generates context-based fallback recommendations integrated into the final CF output.
pipeline/cf_pipeline.py	CF workflow orchestration	Loads contextualised data (df_contextualized.csv), performs splitting, runs CF and cold-start routines, and logs metrics.	Writes df_cf_recommendations.csv and df_cf_recommendations_eval.csv under outputs/CF.
pipeline/core.py	Shared pipeline utilities	Provides safe splitting, reproducible seeding, schema checks, and controlled I/O.	Improves reproducibility and reduces leakage risk across splits.
pipeline/helpers.py	Explanation and interpretation layer	Generates textual rationales (reason, explanation) alongside numerical CF scores for interpretability.	Improves readability of recommendations in reporting and Streamlit.
utils/constants.py	Contextual and structural constants	Defines CONTEXT_TO_RECOMMENDATION_MAP, shared policies, and output schemas used across CF, CBF, and Hybrid.	Standardises contextual logic and mapping between risk states and interventions.
utils/eval_metrics.py	Evaluation framework	Computes Precision@k, Recall@k, MAP@k, and nDCG@k per split and pipeline.	Exports evaluation summaries for analysis and visualisation (e.g., df_cf_recommendations_eval.csv).
pipeline/config.yaml	Global configuration	Defines CF parameters (similarity metric, neighbourhood size, thresholds, score normalisation).	Centralises hyperparameters for reproducibility and hybrid alignment.

3.6.2.2 Content-Based Filtering Implementation

The CBF engine in HCARS models the intrinsic relationship between patient profiles and clinical interventions using contextual, physiological, and risk-aware attributes. Its behaviour is realised through the combined operation of `recommender/cbf.py`, `pipeline/cbf_pipeline.py`, and `pipeline/cbf_helpers.py`.

eline.py, pipeline/helperscbf.py, contextual utilities in utils/context_utils.py, and the configuration and evaluation logic in pipeline/config.yaml and utils/evaluation.py. This design allows CBF to remain fully aligned with the hypertensive context encoded in utils/constants.py and with the global output schemas defined under output_columns.cbf in config.yaml.

1. Patient and item profile construction each patient is represented by a contextual feature vector constructed from the engineered attributes in the contextualized dataset outputs/tmp/df_contextualized.csv. The core contextual dimensions are centralised under contextual_columns and model_settings.cbf.contextual_columns in pipeline/config.yaml (e.g., bp_category, chol_category, risk_level, fbs_cat, cp_cat, age_group, and secondary bins such as chol_bins, restecg_cat, exang_cat, oldpeak_cat, ca_cat, thal_cat).

On the item side, each candidate intervention (e.g., amlodipine, ACE inhibitor, diet control, lifestyle monitoring) is encoded into the same contextual space using mappings in utils/constants.py and utils/context_utils.py. This ensures that both patients and treatments inhabit a shared feature space where contextual compatibility can be measured.

The utils/context_utils.py and pipeline/helperscbf.py modules handle categorical mapping, binning, and any additional feature weighting specified in model_settings.cbf.feature_weights (e.g., equal weights for age, chol_flag, risk_score, age_group).

Pseudocode: profile construction (cbf_pipeline.py + context_utils.py)

```
# Load contextualized data
df_ctx ← load_csv("outputs/tmp/df_contextualized.csv")

# Central contextual columns from config.yaml
context_cols ← config.contextual_columns
cbf_context_cols ← config.model_settings.cbf.contextual_columns

# Build patient profiles
FOR each row r in df_ctx:
    patient_id ← r.patient_id
    patient_vec ← encode_context(r[context_cols]) # via context_utils
    store_patient_vector(patient_id, patient_vec)

# Build item (feature) profiles
feature_ids ← unique(df_ctx.feature_id)
FOR each feature_id in feature_ids:
    item_meta ← lookup_feature_metadata(feature_id, constants)
    item_vec ← encode_feature_context(item_meta, context_cols)
    store_item_vector(feature_id, item_vec)
```

2. Similarity computation, kNN pooling, and clinical gating the core CBF model in `recommender/cbf.py` computes cosine similarity between patient vectors u and item vectors i , reflecting how well an intervention matches a patient’s contextual state. Given a patient vector $\mathbf{u}$ and an item vector $\mathbf{i}$, cosine similarity is defined as follows (Aggarwal, 2016b; Salton & McGill, 1988):

$$\text{sim}(\mathbf{u}, \mathbf{i}) = \frac{\mathbf{u} \cdot \mathbf{i}}{\|\mathbf{u}\| \|\mathbf{i}\|}. \quad (3.17)$$

consistent with the algorithm: "cosine" configuration under `model_settings.cbf`. The similarity neighbourhood is governed by `similarity.k` and `similarity.min_similarity` (e.g., $k=11$, minimum similarity 0.55), while a secondary kNN policy in `model_settings.cbf.knn` (e.g., $k=15$, minimum similarity 0.50) and the runtime `cbf.knn` block orchestrate progressive backoff when too few neighbours are found.

The kNN policy uses a sequence of retries (stored in `cbf.knn.retries`) that gradually relax `min_similarity` and increase k to avoid empty recommendation pools. In parallel, a clinical gating layer in `config.yaml` (`cbf.clinical_gating`) enforces medically sound allow-lists conditioned on context (e.g., only lifestyle recommendations when `bp_category="normal"` and `risk_level="low"`, or combination therapy in `hypertension_stage_2`).

Pseudocode: similarity + kNN + clinical gating (`cbf.py`)

```
# Pre-computed patient and item vectors
```

```
patients ← load_patient_vectors()
```

```
items ← load_item_vectors()
```

```
FOR each patient u in patients:
```

```
  sim_list ← []
```

```
  FOR each item i in items:
```

```
    sim ← cosine_similarity(patients[u], items[i])
```

```
    sim_list.append((i, sim))
```

```
# Sort by similarity and apply base thresholds from config
```

```
sim_list ← sort_descending(sim_list by sim)
```

```
pool ← take_top_k_with_min_sim(
```

```
  sim_list,
```

```
  k = config.model_settings.cbf.similarity.k,
```

```
  min_sim = config.model_settings.cbf.similarity.min_similarity )
```

```
# Progressive backoff if pool is too small
```

```
retry_cfgs ← config.cbf.knn.retries
```

```
FOR each retry in retry_cfgs WHILE size(pool) < k_min_required:
```

```

new_k ← config.model_settings.cbf.knn.k + retry.delta_k
new_min ← config.model_settings.cbf.knn.min_similarity + retry.delta_min_sim
pool ← take_top_k_with_min_sim(sim_list, k = new_k, min_sim = new_min)

# Apply clinical gating rules based on patient's context
gated_pool ← apply_clinical_gating(
    pool,
    context = get_patient_context(u),
    rules = config.cbf.clinical_gating.rules
)

store_similarity_pool(u, gated_pool) # for subsequent scoring

```

3. Score calibration and reason mapping raw cosine similarities are not used directly as final recommendation scores. Instead, they are transformed and combined with rule-based signals under the scoring.cbf_mix block in config.yaml.

First, the similarity values are clipped to $[0, 1]$ using `sim_clip_min` and `sim_clip_max`, then passed through a logistic mapping `sim_to_prob` (type "logistic", with tuned slope and midpoint parameters, e.g. slope = 7.0 and midpoint = 0.62). This logistic transformation is a standard technique for converting similarity scores into probability-like values in recommender systems and decision models (Aggarwal, 2016b; Hastie et al., 2009):

$$p_{\text{sim}} = \frac{1}{1 + \exp(-s \cdot (\text{sim} - m))}, \quad (3.18)$$

where s is the slope and m is the midpoint. This converts similarity into a probability-like score.

In parallel, `helperscbf.py` maps qualitative “reasons” (e.g., `knn_recommendation`, `context_key_match`, `similarity_fallback_medium`) to numeric priors via the `scoring.cbf_reason_to_score` dictionary (e.g., `recommended: 1.0`, `knn_recommendation: 0.90`, `similarity_fallback_low: 0.50`). The final calibrated raw score `score_cbf_raw` is computed as a weighted mixture:

$$\text{score_cbf_raw} = w_{\text{sim}} \cdot p_{\text{sim}} + w_{\text{reason}} \cdot p_{\text{reason}}, \quad (3.19)$$

where w_{sim} and w_{reason} are defined in `scoring.cbf_mix.w_similarity` and `w_reason` (e.g. 0.70 and 0.20), and p_{reason} is derived from the reason prior plus optional prior-modulation terms (`prior_mod_base`, `prior_mod_gain`).

A final probability-like score `score_cbf` is obtained by post-calibration with isotonic regression model stored in `models/cbf_isotonic.pkl`, as indicated by `post_calibration`:

"isotonic" and `models.cbf_isotonic_path` in `config.yaml`. A global floor (`min_floor`) and acceptance threshold (`accept_threshold`) enforce clinically conservative cut-offs final scores.

Explanatory text fields (`reason`, `explanation`, `explanation_clinical`, `advice`) are then produced in `helperscbf.py` based on the gating rule, similarity context, and risk level, and written to the output using the centralised schema in `output_columns.cbf`.

Pseudocode: calibration and explanation (`cbf.py` + `helperscbf.py`)

FOR each patient `u`:

`gated_pool` $\leftarrow$ `load_similarity_pool(u)` # (`feature_id`, `sim`)

 FOR each (`feature_id`, `sim`) in `gated_pool`:

 1) Clip similarity and map to probability via logistic

`sim_clipped` $\leftarrow$ `clip (`

`sim`,

`min = config.scoring.cbf_mix.sim_clip_min`,

`max = config.scoring.cbf_mix.sim_clip_max`

`)`

`p_sim` $\leftarrow$ `logistic(`

`x` = `sim_clipped`,

`slope` = `config.scoring.cbf_mix.sim_to_prob.slope`,

`midpoint = config.scoring.cbf_mix.sim_to_prob.midpoint`

`)`

 2) Map qualitative reason to numeric prior

`reason_label` $\leftarrow$ `infer_reason(u, feature_id, sim, context = gated_pool)`

`p_reason` $\leftarrow$ `config.scoring.cbf_reason_to_score.get(`

`reason_label`,

`config.scoring.unknown_reason_score`

`)`

 # 3) Combine similarity-based and reason-based probabilities

`w_sim` $\leftarrow$ `config.scoring.cbf_mix.w_similarity`

`w_reason` $\leftarrow$ `config.scoring.cbf_mix.w_reason`

`score_cbf_raw` $\leftarrow$ `w_sim * p_sim + w_reason * p_reason`

 4) Optional isotonic calibration

`score_cbf` $\leftarrow$ `apply_isotonic(`

`score_cbf_raw`,

`model_path = config.models.cbf_isotonic_path`

`)`

5) Build explanations

```
explanation, explanation_clinical, advice ← \
    build_cbf_explanations(u, feature_id, reason_label, score_cbf)
append_recommendation(
    patient_id      = u,
    feature_id      = feature_id,
    similarity       = sim,
    score_cbf_raw   = score_cbf_raw,
    score_cbf       = score_cbf,
    reason          = reason_label,
    explanation      = explanation,
    explanation_clinical = explanation_clinical,
    advice          = advice          )
```

4. Pipeline orchestration and artefacts the orchestration of the CBF subsystem is implemented in pipeline/cbf_pipeline.py. This module:

1. loads the contextualized dataset outputs/tmp/df_contextualized.csv,
2. constructs patient and item profiles using utils/context_utils.py,
3. computes similarity and kNN pools,
4. applies clinical gating and backoff,
5. calibrates scores and generates explanations using pipeline/helperscbf.py, and
6. writes ranked recommendations to outputs/CBF/df_cbf_recommendations.csv following the cbf_recommendations_schema in config.yaml.

The exported records include, among others: patient_id, contextual labels (bp_category, chol_category, risk_level, fbs_cat, cp_cat, age_group, chol_bins, restecg_cat, exang_cat, oldpeak_cat, ca_cat, thal_cat), feature_id, similarity, score_cbf_raw, score_cbf, reason, explanation, explanation_clinical, advice, true_response, source, and status.

Pseudocode: CBF pipeline execution (cbf_pipeline.py)

1) Load contextualized data

```
df_ctx ← load_csv("outputs/tmp/df_contextualized.csv")
```

2) Build patient and item profiles

```
patient_vectors ← build_patient_vectors(df_ctx, config.contextual_columns)
```

```
item_vectors ← build_item_vectors(df_ctx, constants, config.contextual_columns)
```

3) Compute similarity pools with kNN + clinical gating

```
similarity_pools ← {}
```

```

FOR each patient u in patient_vectors:
    pool_u ← compute_similarity_pool(
        u_vector = patient_vectors[u],
        item_vectors = item_vectors,
        model_config = config.model_settings.cbf,
        knn_runtime = config.cbf.knn )
    gated_pool_u ← apply_clinical_gating(
        pool_u,
        context = get_patient_context(df_ctx, u),
        gating_rules = config.cbf.clinical_gating.rules )
    similarity_pools[u] ← gated_pool_u

```

4) Calibrate scores and build explanations

```

recs_all ← []
FOR each patient u:
    recs_u ← score_and_explain_cbf(
        patient_id = u,
        pool = similarity_pools[u],
        scoring_cfg = config.scoring,
        model_paths = config.models )
    recs_all.extend(recs_u)

```

5) Export recommendations to CSV

```

write_recommendations(
    recs_all,
    path = "outputs/CBF/df_cbf_recommendations.csv",
    schema = config.schemas.cbf_recommendations_schema )

```

5. Evaluation Protocol and Metrics. CBF evaluation follows the ranking-focused framework in `utils/evaluation.py`, driven by the evaluation block of `config.yaml`. The system employs a mode: "leave_one_context_out" (LOCO) protocol, where context groups are defined by `bp_category`, `chol_category`, and `risk_level` (with optional extension to `age_group`). For each held-out context, the model is tested on that subgroup while trained on the remaining contexts, thereby assessing generalisation across cardiovascular strata.

The evaluation block specifies:

- ranking metrics: Precision@k, Recall@k, MAP@k, and nDCG (under `evaluation.metrics.ranking`);
- score column: `score_cbf`;

- cut-off: $k_{at} = 5$;
- grouping columns: ["bp_category", "chol_category", "risk_level"].

Formally, for each patient u with ground-truth relevant set G_u and top- k recommendations $R_u@k$, Precision@ k , Recall@ k , MAP@ k , and nDCG@ k are computed as in the CF section (Section 3.6.2.1), but using score_cbf to rank items. The evaluation artefacts are exported to outputs/CBF/df_cbf_recommendations_eval.csv and summarised in cbf_eval_full.csv, cbf_eval_unique.csv, and cbf_eval_loco.csv for detailed analysis.

Pseudocode: CBF evaluation (evaluation.py)

FUNCTION evaluate_cbf(recs_path, k):

```

Load exported recommendations with true_response
df_recs ← load_csv(recs_path) # includes score_cbf and true_response
group_cols ← config.evaluation # ["bp_category", "chol_category", "risk_level"]
score_col ← config.evaluation.score_col # "score_cbf"
results_full ← []
results_loco ← []
Overall metrics
metrics_global ← compute_ranking_metrics(df_recs, k, score_col)
results_full.append(("global", metrics_global))
Context-wise LOCO evaluation
FOR each context_group in unique(df_recs[group_cols]):
    df_train, df_test ← leave_one_context_out_split(
        df_recs,
        context_group,
        group_cols)
    metrics_ctx ← compute_ranking_metrics(df_test, k, score_col)
    results_loco.append((context_group, metrics_ctx))
Export summaries
save_csv(results_full, path = config.evaluation.after_export.full_path)
save_csv(results_loco, path = config.evaluation.after_export.loco_path)
RETURN results_full, results_loco

```

6. Module-Level dependencies, flowchart, and interpretability at the implementation level, the CBF subsystem consists of clearly separated modules that can be independently inspected and tested. Table 3.6 summarises file-level responsibilities, while Figure 3.3 provides a flowchart-style view of how contextualised data move from patient/item vector construction through similarity computation, clinical gating, calibrated scoring, and evaluation.

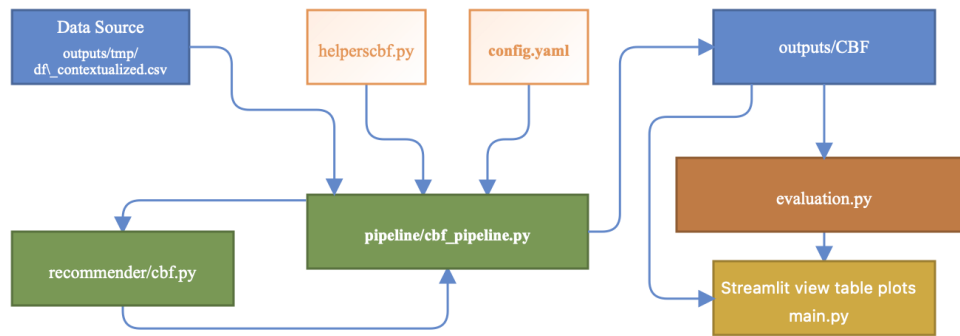


Figure 3.3: CBF pipeline structure and module interaction in HCARS: patient item vector generation, similarity computation, calibration, and evaluation.

By combining cosine-based similarity, clinically gated kNN pooling, probabilistic calibration, and a LOCO-based evaluation protocol, the CBF subsystem in HCARS remains both context-aware and clinically interpretable.

Table 3.6: CBF Logic and File Dependencies in HCARS

Module / File	Primary Role	Core Operations / Logic	Outputs or Interactions
recommender/cbf.py	Core similarity model	Constructs patient-item vectors, computes cosine similarity, applies context matching, and exports similarity artefacts.	Produces raw and normalised similarity scores used in calibrated CBF scoring.
pipeline/cbf_pipeline.py	Pipeline orchestration	Loads contextualised data, manages kNN fallback, applies gating, and logs pipeline outputs.	Writes df_cbf_recommendations.csv and evaluation artefacts under outputs/CBF.
pipeline/helperscbf.py	Clinical explanation layer	Builds clinical narratives (reason, explanation, advice) linked to similarity and risk context.	Adds interpretability fields for reporting and Streamlit visualisation.
utils/context_utils.py	Context feature utilities	Handles feature binning, categorical mappings, and context-key generation for patient/item representations.	Keeps contextual encoding consistent across CBF, CF, and Hybrid layers.
utils/constants.py	Shared clinical constants	Stores CONTEXT_TO_RECOMMENDATION_MAP, contextual columns, and rule mappings between context and feature families.	Provides shared clinical logic consumed by all recommenders.
utils/evaluation.py	Evaluation framework	Implements ranking metrics, LOCO evaluation, and reporting across context clusters.	Generates cbf_eval_full.csv, cbf_eval_unique.csv, and summary reports.

Continued on next page

Table 3.6 (continued): CBF Logic and File Dependencies in HCARS

Module / File	Primary Role	Core Operations / Logic	Outputs or Interactions
pipeline/config.yaml	Central configuration	Defines contextual columns, similarity thresholds, clinical gating rules, and calibration settings.	Controls similarity, scoring, and gating policies system-wide.

3.6.3 Hybrid Model Integration

3.6.3.1 Hybridization Techniques

The hybrid layer fuses the calibrated outputs of CF and CBF into a single, reproducible ranking policy. The design is configuration-driven through `pipeline/config.yaml` and implemented in `recommender/hybrid.py` with orchestration in `pipeline/hybrid_pipeline.py`. The hybrid module consumes the per-patient, per-feature scores from both subsystems (CF and CBF) and produces a final `hybrid_score` that respects clinical constraints and calibrated confidence signals. The logic remains anchored to the hypertensive context and structural mappings defined in `utils/constants.py` and the output schemas configured under `output_columns.hybrid`.

At the level of individual recommendation rows, the hybrid subsystem distinguishes between four underlying sources that are encoded in a dedicated field `hybrid_source`:

- CF (`collaborative_filtering`): recommendations originating from the CF engine.
- CBF (`exact/context match`): deterministic matches between patient context keys and item profiles.
- CBF (`fallback`): CBF recommendations obtained through similarity fallback and rule-based safety margins when standard kNN evidence is weak.
- CBF (`kNN`): similarity-based recommendations from the CBF kNN pipeline.

Cold-start behaviour is restricted to the CF subsystem (through `recommender/cold_start_handler.py`), whereas CBF contributes its own fallback logic (CBF (`fallback`)) via similarity backoff and clinical gating. The hybrid layer combines these heterogeneous sources while keeping their provenance explicit for auditability.

1. Inputs and per-model normalization

The hybrid layer operates on already calibrated model outputs:

- CF provides a score `score_cf` in $[0, 1]$ after normalization under `cf.score_calibration` (such as min-max with numerical jitter).

- CBF provides a score `score_cbf` in $[0, 1]$ after the similarity-to-probability mapping and post-calibration pipeline described in Section 3.6.2.2, controlled through `scoring.cbf_mix`.

In `pipeline/hybrid_pipeline.py`, the CF and CBF recommendation tables are loaded and aligned on `patient_id` and `feature_id`. Missing scores from either side are handled by conservative defaults defined in `hybrid_policy` within `pipeline/config.yaml` (for instance, treating an absent CF score as a weak CF signal rather than a hard zero when CBF is present, and vice versa). Operationally, each joined row is converted into a complete feature tuple

`(patient_id, feature_id, pcf, pcbf, context features, meta flags)`,

which forms the input to the core fusion logic in `recommender/hybrid.py`.

Pseudocode: merging CF and CBF inputs (`hybrid_pipeline.py`)

```
# Load CF and CBF outputs
```

```
df_cf ← load_csv(config.paths.cf_recommendations)
df_cbf ← load_csv(config.paths.cbf_recommendations)
```

```
# Standardize key columns and score fields
```

```
df_cf ← normalize_cf_columns(df_cf) # patient_id, feature_id, score_cf, ...
```

```
df_cbf ← normalize_cbf_columns(df_cbf) # patient_id, feature_id, score_cbf, ...
```

```
# Inner or outer join based on configuration
```

```
df_joined ← join_cf_cbf(
  cf_df = df_cf,
  cbf_df = df_cbf,
  on_keys = ["patient_id", "feature_id"],
  join_type = config.hybrid.join_type # for example "outer"
)
```

```
# Apply conservative defaults for missing scores according to hybrid_policy
```

```
df_joined ← apply_hybrid_defaults(df_joined, config.hybrid_policy)
```

2. Core fusion rule and similarity modulation

The central fusion step in `recommender/hybrid.py` computes a hybrid score for each patient–feature pair. The base fusion is a convex combination between CF and CBF scores:

$$\text{score}_{\text{hyb,base}} = w_{\text{cf}} \cdot p_{\text{cf}} + (1 - w_{\text{cf}}) \cdot p_{\text{cbf}}, \quad (3.20)$$

where p_{cf} and p_{cbf} are the calibrated scores from CF and CBF, and $w_{\text{cf}} \in [0, 1]$ is configured under `hybrid.global_weight_cf`. The corresponding CBF weight w_{cbf} is implicitly

$$w_{\text{cbf}} = 1 - w_{\text{cf}}.$$

If only one model contributes (for example, CBF-only rows for items unseen by CF), the effective weights collapse to the available model (e.g. $w_{\text{cf}} = 0$ and $w_{\text{cbf}} = 1$), enforced by the default-handling logic in `hybrid_policy`. Thus, Equation (3.20) covers all cases: CF-only, CBF-only, and CF+CBF joint fusion.

A similarity-based modulation term then adjusts this base score, using configuration parameters under `hybrid.sim_mod` and `hybrid.sim_thresholds`. The effective similarity signal σ acts as a proxy for confidence and is defined as

$$\sigma = \max(p_{\text{cf}}, p_{\text{cbf}}),$$

so that at least one strong signal is sufficient to raise overall confidence.

The modulation applies a linear factor:

$$\text{score}_{\text{hyb,mod}} = (a_0 + a_1 \sigma) \cdot \text{score}_{\text{hyb,base}}, \quad (3.21)$$

where the intercept a_0 and slope a_1 are configured in `hybrid.sim_mod`. Threshold blocks in `hybrid.sim_thresholds` define ranges of σ where the modulation behaves as a slight boost or a dampening factor (e.g. `boost_high_confidence` for σ above a high threshold, and `damp_low_confidence` for low-confidence regions). This mechanism allows the hybrid to emphasise highly confident items and suppress items whose combined CF/CBF evidence remains low, without breaking the probabilistic interpretation of the scores.

Pseudocode: core fusion (`hybrid.py`)

FUNCTION `compute_hybrid_score(p_cf, p_cbf, config)`:

```
# 1) Read global weights
w_cf ← config.hybrid.global_weight_cf
w_cbf ← 1.0 - w_cf
```

```

# 2) Base convex combination
score_base ← w_cf * p_cf + w_cbf * p_cbf

# 3) Effective similarity / confidence proxy
sigma ← max(p_cf, p_cbf)

# 4) Linear similarity modulation
a0, a1 ← config.hybrid.sim_mod.intercept, config.hybrid.sim_mod.slope
mod_fact ← a0 + a1 * sigma

# 5) Apply threshold-based clipping and shaping
mod_fact ← apply_sim_thresholds(
    mod_fact,
    sigma,
    thresholds = config.hybrid.sim_thresholds
)
score_mod ← score_base * mod_fact
RETURN score_mod

```

3. Switching policies and safety gates

Beyond pure weighted fusion, the hybrid layer applies switching rules configured under `hybrid_policy`. These rules determine when one model should dominate the final decision instead of relying solely on the convex combination:

- A `cbf_rescue` flag instructs the hybrid layer to favour CBF whenever CBF is clinically decisive while CF remains weak (for instance, when p_{cbf} is above a high confidence threshold and p_{cf} lies below a low confidence threshold).
- A `cf_rescue` option can similarly prefer CF when CBF is neutral but CF detects strong collaborative patterns.
- A `balance_prior` sub-block controls small systematic re-weighting to avoid chronic dominance of a single model across all contexts.
- A `group_weighting` mechanism allows context-specific re-weighting based on hypertension stage or composite context keys (e.g. combinations of `bp_category` and `risk_level`) defined in `utils/constants.py`.

Additionally, a conflict flag is raised when the gap between p_{cf} and p_{cbf} exceeds a configured tolerance:

$$|p_{cf} - p_{cbf}| > \delta_{\text{conflict}},$$

where δ_{conflict} is configured in `hybrid_policy`. In such cases, the fused score is still computed, but the recommendation row carries a `conflict_flag` field and may be subject to more conservative clipping or de-prioritisation in downstream analysis.

Pseudocode: hybrid switching and conflict handling (`hybrid.py`)

```

FUNCTION apply_hybrid_policy(p_cf, p_cbf, patient_context, config):
    # Base fusion + modulation
    score_hyb ← compute_hybrid_score(p_cf, p_cbf, config)
    # 1) Conflict flag
    delta ← abs(p_cf - p_cbf)
    conflict ← (delta > config.hybrid_policy.conflict_delta_threshold)
    # 2) Optional model dominance based on policy and thresholds
    IF config.hybrid_policy.cbf_rescue AND \
        p_cbf ≥ config.hybrid_policy.cbf_high_conf AND \
        p_cf ≤ config.hybrid_policy.cf_low_conf:
        score_hyb ← p_cbf
        hybrid_source ← "CBF (exact/context match)"
    ELSE IF config.hybrid_policy.cf_rescue AND \
        p_cf ≥ config.hybrid_policy.cf_high_conf AND \
        p_cbf ≤ config.hybrid_policy.cbf_low_conf:
        score_hyb ← p_cf
        hybrid_source ← "CF (collaborative_filtering)"
    ELSE:
        hybrid_source ← "Hybrid (weighted_fusion)"
    # 3) Group-based re-weighting by cardiovascular stratum
    group_key ← derive_group_key(patient_context) # e.g., bp_category + risk_level
    score_hyb ← apply_group_weighting(
        score_hyb,
        group_key,
        config.hybrid_policy.group_weighting)
    RETURN score_hyb, conflict, hybrid_source

```

4. CF Cold-Start vs. CBF fallback and filtering policy.

Cold-start cases originate in the CF subsystem, where patients or items lack historical interactions. The CF module tags such rows with `source = "Cold-Start"` and a status code such as `"cold_start_safe_default"`. This is distinct from CBF fallback behaviour, where CBF generates recommendations under low-similarity or backoff conditions (via kNN retries and relaxed thresholds) and tags them as CBF (fallback) through reason labels and `hybrid_source`. In other words:

- CF cold-start: driven by missing behavioural evidence.
- CBF fallback: still similarity-based and context-aware, but operating at a conservative confidence level.

In the hybrid layer, a configuration flag under `cold_start.exclude_from_hybrid_output` determines whether pure CF cold-start rows participate in hybrid rankings:

- If `cold_start.exclude_from_hybrid_output = true`, CF cold-start entries are removed from the final hybrid list and presented separately in the user interface as a safety panel, while CBF fallback entries remain part of the hybrid ranking since they are based on contextual similarity.
- If the flag is disabled, CF cold-start entries are kept, but their scores are clipped by a safety floor and tag fields (such as `status`, `source`, and `hybrid_source`) are preserved for full traceability.

After computing `hybrid_score`, the pipeline applies global filters:

- A minimum `score_threshold` configured under `hybrid` to remove low-confidence items.
- A per-patient cap on the maximum number of items retained, guided by `hybrid.top_n_eval`.

Pseudocode: CF cold-start, CBF fallback, and filtering (`hybrid_pipeline.py`)

FOR each row `r` in `df_joined`:

`p_cf` ← `r.score_cf`

`p_cbf` ← `r.score_cbf`

`ctx` ← `extract_context(r)`

`score_hyb, conflict, hybrid_source` ← `apply_hybrid_policy(p_cf, p_cbf, ctx, config)`

CF cold-start exclusion if requested

IF `config.cold_start.exclude_from_hybrid_output` AND \

`r.source_cf == "Cold-Start"`:

CONTINUE # skip CF cold-start in hybrid output

CBF fallback remains allowed but explicitly tagged

IF `r.source_cbf == "CBF_fallback"`:

`hybrid_source` ← `"CBF (fallback)"`

```
# CBF kNN label for standard similarity-based entries
```

```
IF r.source_cbf == "CBF_knn":
```

```
    hybrid_source ← "CBF (kNN)"
```

```
# Apply score threshold
```

```
IF score_hyb < config.hybrid.score_threshold:
```

```
    CONTINUE
```

```
append_hybrid_row(
```

```
    patient_id = r.patient_id,
```

```
    feature_id = r.feature_id,
```

```
    score_cf   = p_cf,
```

```
    score_cbf  = p_cbf,
```

```
    hybrid_score = score_hyb,
```

```
    conflict_flag = conflict,
```

```
    hybrid_source = hybrid_source,
```

```
    status       = r.status
```

```
)
```

```
# After loop: within each patient_id, keep only top-N items
```

```
df_hybrid ← top_n_per_patient(df_hybrid, n = config.hybrid.top_n_eval)
```

5. Hybrid pipeline orchestration and artefacts

The overall hybrid workflow is orchestrated by `pipeline/hybrid_pipeline.py`. This module:

1. loads CF and CBF recommendation files from their configured paths in `config.yaml`,
2. aligns and joins them on `patient_id` and `feature_id`,
3. computes hybrid scores and applies hybrid policies using `recommender/hybrid.py`,
4. applies CF cold-start exclusion, CBF fallback tagging, and global thresholds,
5. ranks and caps items per patient, and
6. writes the final recommendations and evaluation artefacts to the hybrid output paths defined in `config.yaml`.

The exported hybrid records follow `output_columns.hybrid`, which typically includes: `["patient_id", "feature_id", "score_cf", "score_cbf", "hybrid_score", "conflict_flag", "hybrid_source", "status", "bp_category", "chol_category", "risk_level", ...]`.

Pseudocode: hybrid pipeline execution (hybrid_pipeline.py)

1) Load per-model recommendations

df_cf ← load_cf_recommendations(config)

df_cbf ← load_cbf_recommendations(config)

2) Join CF and CBF outputs and prepare defaults

df_joined ← prepare_hybrid_input(df_cf, df_cbf, config)

3) Compute hybrid scores and apply gates

hybrid_rows ← []

FOR each row r in df_joined:

 p_cf ← r.score_cf

 p_cbf ← r.score_cbf

 ctx ← extract_context(r)

 score_hyb, conflict, hybrid_source ← apply_hybrid_policy(p_cf, p_cbf, ctx, config)

 IF should_exclude_due_to_cold_start(r, config):

 CONTINUE

 IF score_hyb < config.hybrid.score_threshold:

 CONTINUE

 hybrid_rows.append(build_hybrid_output_row(

 row = r,

 hybrid_score = score_hyb,

 conflict_flag = conflict,

 hybrid_source = hybrid_source))

4) Final ranking and per-patient truncation

df_hybrid ← dataframe_from_rows(hybrid_rows)

df_hybrid ← rank_and_truncate_per_patient(

 df_hybrid,

 n_top = config.hybrid.top_n_eval)

5) Save hybrid outputs

save_csv(df_hybrid, path = config.paths.hybrid_recommendations)

6. Integrated evaluation within the hybrid pipeline.

Hybrid evaluation is embedded directly in hybrid_pipeline.py and recommender/hybrid.py. After computing the final hybrid_score, the pipeline reconstructs ground-truth relevance using the true_response labels joined either from the collaborative filtering outputs or from the contextualised dataset.

For each patient u , with a relevant item set G_u and the top- k hybrid recommendations $R_u@k$, system performance is assessed using standard ranking-based evaluation metrics, namely Precision@ k , Recall@ k , Mean Average Precision (MAP@ k), and normalized Discounted Cumulative Gain (nDCG@ k), as formally defined in Equations (3.1)–(3.5).

These metrics are computed separately for the training, validation, and test splits, and the resulting evaluation outputs are exported for subsequent analysis and visualization. This integrated evaluation design ensures a consistent and fair comparison across the content-based, collaborative, and hybrid recommendation paradigms within the HCARS framework.

The cut-off k and the column used for ranking (typically `hybrid_score`) are read from the evaluation settings of `config.yaml`. Evaluation summaries per split or per cardiovascular group are written to hybrid-specific CSV files under the configured hybrid output directory. These artefacts are subsequently used by the Streamlit application `main.py` to display comparative plots between CF, CBF, and Hybrid performance, making the effect of fusion transparent at the metric level.

Pseudocode: hybrid evaluation (`hybrid_pipeline.py`)

FUNCTION `evaluate_hybrid(df_hybrid, k, config)`:

`df_hybrid` includes: `patient_id`, `feature_id`, `hybrid_score`, `hybrid_source`,

1) Group by patient and sort by `hybrid_score`

`grouped` $\leftarrow$ `groupby(df_hybrid, "patient_id")`

`metrics_per_patient` $\leftarrow$ []

FOR each `patient_id`, `df_u` in `grouped`:

`ranked_u` $\leftarrow$ `sort_descending(df_u, by = "hybrid_score")`

`topk_u` $\leftarrow$ `head(ranked_u, k)`

`G_u` $\leftarrow$ { `i` : `row.true_response == 1` FOR `row` in `df_u` }

`R_u` $\leftarrow$ { `i` : `row.feature_id` FOR `row` in `topk_u` }

`metrics_u` $\leftarrow$ `compute_ranking_metrics_for_user(R_u, G_u, k)`

`metrics_per_patient.append(metrics_u)`

2) Aggregate across patients

`metrics_global` $\leftarrow$ `aggregate_metrics(metrics_per_patient)`

3) Save evaluation artefacts

`save_csv(metrics_per_patient, path = config.paths.hybrid_eval_per_patient)`

`save_csv(metrics_global, path = config.paths.hybrid_eval_global)`

RETURN `metrics_global`

7. Module-Level dependencies, flowchart, and structural summary.

From a software-engineering perspective, the hybrid subsystem is composed of clearly separated yet tightly coordinated modules. Figure 3.4 provides a flowchart-style view of how CF and CBF outputs are merged, fused, filtered, and evaluated inside HCARS, while Table 3.7 summarises the main file-level responsibilities and their interactions.

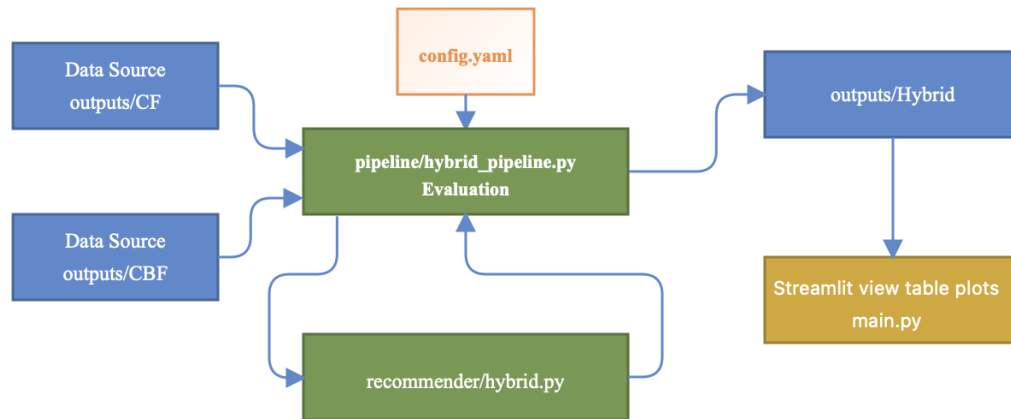


Figure 3.4: Hybrid pipeline structure and module interaction in HCARS: calibrated CF/CBF outputs, cold-start flags, and CBF fallback variants are fused into the final hybrid ranking and evaluation.

Table 3.7: Hybrid Logic and File Dependencies in HCARS

Module / File	Primary Role	Core Operations / Logic	Outputs or Interactions
recommender/hybrid.py	Hybrid fusion and policy engine	Convex fusion of CF and CBF scores; similarity modulation; policy gates (switching), conflict checks, and group weighting.	Produces per-row hybrid_score, hybrid_source, conflict flags, and provenance tags for the pipeline.
pipeline/hybrid_pipeline.py	Pipeline orchestration and evaluation	Loads CF/CBF outputs, merges inputs, runs fusion, applies cold-start exclusion and fallback tags, enforces thresholds, ranks/truncates per patient, and computes metrics.	Writes final hybrid recommendation CSVs and evaluation summaries to configured output paths.
pipeline/config.yaml	Hybrid configuration backbone	Defines hybrid.global_weight_cf, hybrid.sim_mod, similarity thresholds, policy flags, cold-start settings, and output_columns.hybrid.	Controls fusion hyperparameters, filtering rules, output schema, and evaluation settings.

Continued on next page

Table 3.7 (continued): Hybrid Logic and File Dependencies in HCARS

Module / File	Primary Role	Core Operations / Logic	Outputs or Interactions
utils/ constants.py	Shared structural/context constants	Provides contextual-state mappings, hypertension strata/group keys, and shared tags consumed by the hybrid policy logic.	Keeps hybrid decisions aligned with the CF/CBF clinical ontology and standardises source labelling.
outputs/CF/... and outputs/CBF/...	Upstream model artefacts	Provide calibrated CF/CBF scores, reasons, source tags (collaborative, exact/context match, fallback, kNN), and contextual labels used as fusion inputs.	Serve as upstream artefacts consumed by the hybrid pipeline for fusion and hybrid_source attribution.

Through this design, the hybrid subsystem provides a transparent and auditable bridge between individual recommenders (CF and CBF) and the final clinical decision-support layer. Every fusion step is explicitly parameterised in `config.yaml` and implemented in dedicated modules, while the `hybrid_source` field records whether each recommendation arises from CF, CBF exact/context match, CBF fallback, or CBF kNN. This makes the structure of the model, its components, its hyperparameters, and its effective sources directly traceable from code to experimental results and clinical interpretations.

3.7 Model evaluation and validation

The evaluation and validation framework of HCARS ensures that every recommendation model (CBF, CF, and Hybrid) is assessed through transparent, reproducible, and clinically interpretable criteria. Rather than redefining the model-specific logic already detailed in Sections 3.6.2.1, 3.6.2.2, and 3.6.3.1, this section focuses on the unified evaluation protocol, the way ranking metrics are operationalised across pipelines, and how the resulting artefacts support error analysis, robustness checks, and clinical auditability. Each evaluation stage is implemented programmatically in modular form (see `utils/eval_metrics.py` and `utils/evaluation.py`), invoked automatically by `pipeline/cf_pipeline.py`, `pipeline/cbf_pipeline.py`, and `pipeline/hybrid_pipeline.py` during execution.

3.7.1 Evaluation protocol and experimental setup

Building on the data-loading and splitting workflow described in Section 3.8, all experiments follow a fixed and auditable protocol to guarantee fairness across recommendation strategies. The contextualized hypertension dataset (`outputs/tmp/df_contextualized.csv`) is deterministically divided into training, validation, and testing partitions, with consistent random seeds for reproducibility. The validation set is used to tune hyperparameters such as:

- CF neighbourhood size and similarity thresholds (within `cf_pipeline.py`).

- CBF calibration slope, similarity cut-offs, and kNN backoff settings (within `cbf_pipeline.py`).
- Hybrid fusion weights, similarity modulation parameters, and policy gates (within `hybrid_pipeline.py`).

Where intermediate stability is required, k -fold cross-validation can be activated inside the evaluation utilities, providing variance estimates around the main ranking metrics without altering the production splitting scheme. Throughout tuning, the hold-out test partition remains untouched, ensuring that the reported test metrics reflect unbiased generalization performance. Each evaluation run logs configuration metadata, random seeds, and split indices, forming a complete experimental record that accompanies the exported metric tables and supports later replication or sensitivity analysis.

3.7.2 Evaluation mechanics and artefact generation

In all three pipelines, evaluation is automatically triggered at the end of the main orchestration function, but the resulting artefacts are designed to play complementary roles when viewed from the system level.

Collaborative Filtering as detailed in Section 3.6.2.1, the CF pipeline constructs predicted interaction lists per patient and passes them to `utils/eval_metrics.py`. Within the unified framework, three families of CF artefacts are produced:

- a global evaluation table (`df_cf_recommendations_eval.csv`) that summarises baseline ranking quality across the full cohort;
- a Leave-One-Context-Out (LOCO) view (`df_cf_recommendations_eval_loco.csv`) that quantifies robustness when a contextual dimension is withheld;
- a LOCO-Family table (`df_cf_recommendations_eval_loco_family.csv`) that aggregates performance over cardiovascular risk clusters.

Together, these panels allow the CF subsystem to be inspected at three levels: overall behaviour, sensitivity to specific context strata, and stability across clinically meaningful families, rather than only through a single aggregate score.

Content-Based Filtering the CBF pipeline (`pipeline/cbf_pipeline.py`) invokes `utils/evaluation.py` after calibrated scores have been produced, as described in Section 3.6.2.2. From the global perspective, CBF evaluation complements CF in two ways: (i) it uses the same ranking metrics and cut-offs, making direct comparison feasible, and (ii) it emphasises context-stratified LOCO analysis, where groups are defined by `bp_category`, `chol_category`, and `risk_level`. Outputs such as `df_cbf_recommendations_eval.csv` and derived summaries

(e.g. `cbf_eval_full.csv`, `cbf_eval_loco.csv`) act as a diagnostic surface for ablation studies: they reveal how much of the performance gain is driven by contextual encoding, isotonic calibration, or clinical gating choices rather than by similarity alone.

Hybrid fusion the Hybrid pipeline (`pipeline/hybrid_pipeline.py`) consolidates the calibrated outputs from CBF and CF into a unified schema and then calls the shared evaluation utilities. Unlike CF and CBF, hybrid evaluation is explicitly policy-aware: metric summaries are computed both before and after the application of hybrid gates, cold-start exclusion rules, and score thresholds specified in `config.yaml`. Structured artefacts such as `hybrid_eval_summary.json` and per-split CSVs store, side by side, the performance of CF, CBF, and Hybrid under consistent k values and metric definitions. This design makes it possible to quantify not only the absolute advantage of fusion, but also the marginal effect of safety policies (e.g. excluding CF cold-start rows from the hybrid output) on both ranking quality and patient coverage.

3.7.3 Evaluation metrics

Across all pipelines, HCARS adopts standard information retrieval ranking metrics whose mathematical formulations are already given in Sections 3.6.2.1 and 3.6.3.1: $\text{Precision@}k$, $\text{Recall@}k$, $\text{MAP@}k$, and $\text{nDCG@}k$. Rather than re-stating their equations, this section clarifies how these metrics are used in practice.

All four metrics are computed at a fixed cut-off ($k = 5$), which is the standard evaluation depth defined in the HCARS configuration (`recommendation.top_n_eval = 5`, `evaluation.k_at = 5`). This choice reflects a clinically realistic recommendation list size small enough to match the actionable set typically considered by clinicians, yet sufficiently broad to ensure meaningful ranking evaluation in line with established recommender-systems practice.

All metrics ($\text{Precision@}5$, $\text{Recall@}5$, $\text{MAP@}5$, $\text{nDCG@}5$) are evaluated:

- per model (CBF, CF, Hybrid) and per data split (training, validation, testing),
- in both global form (cohort-level averages) and context-stratified form (e.g. LOCO and risk-family panels),
- in pre-filter and post-filter variants, distinguishing raw ranking ability from clinically gated recommendation lists.

In this configuration, $\text{Precision@}k$ and $\text{Recall@}k$ primarily characterise how many clinically relevant interventions are surfaced among the top- k suggestions, while $\text{MAP@}k$ and $\text{nDCG@}k$ emphasise the ordering of those items. The combination allows the system to distinguish between models that retrieve the correct items but mis-rank them and models that achieve both high retrieval and good rank ordering. Because all pipelines share the

same evaluation utilities and configuration block (evaluation in `config.yaml`), metric values are directly comparable across CF, CBF, and Hybrid without re-tuning thresholds on a per-model basis.

3.7.3.1 Auxiliary and diagnostic metrics

In addition to the core ranking metrics, the evaluation suite optionally computes secondary indicators designed to highlight practical properties of the recommender system:

- **Coverage:** the proportion of patients for whom at least one valid recommendation is generated. This metric quantifies inclusivity and is especially sensitive to aggressive score thresholds or strict clinical gating rules.
- **Diversity:** a measure of heterogeneity among recommended items, which guards against repetitive or highly concentrated recommendations (e.g. over-use of a single drug family) and supports more balanced intervention portfolios.
- **Novelty:** the extent to which recommended items are not trivial repetitions of frequently observed patterns. In the HCARS context, novelty is used to monitor whether the system proposes new, yet clinically justified, combinations rather than echoing only the most common interventions.

For research completeness, numerical error metrics such as RMSE/MAE and classification-style AUC can be computed for ablation or sanity checks, for example when comparing different scoring transformations or calibration strategies. However, these are not treated as primary success criteria: the system is optimised around ranking and retrieval quality, which more directly reflects the decision-support use case of recommending a small set of interventions rather than predicting absolute risk scores.

3.7.4 Validation and reproducibility assurance

Every evaluation artefact produced by HCARS includes metadata headers encoding timestamp, configuration fingerprint, and random seed, ensuring full auditability of reported results. Metric exports are written as immutable CSV/JSON files under `outputs/CF`, `outputs/CBF`, and `outputs/Hybrid`, and are subsequently consumed in read-only mode by the Streamlit interface (`main.py`) for visual inspection and comparative plots.

From a methodological standpoint, this separation between computation and visualization ensures that identical inputs and configurations yield identical metrics across re-runs, while the underlying code and configuration states remain inspectable. The evaluation layer thus fits into the broader reproducibility strategy described in Section 3.9: evaluation scripts rely exclusively on persisted artefacts, versioned configuration files, and deterministic seeds, enabling independent reviewers to recompute all metrics for CF, CBF,

and Hybrid without modifying the core implementation. In practice, this means that any reported table or figure in this thesis can be traced back to a specific evaluation run, a specific configuration snapshot, and a concrete set of recommendation artefacts stored under `outputs/`, thereby closing the loop between experimental design, implementation, and clinical interpretation.

3.8 Study procedures

This section presents the procedural workflow followed throughout the development and validation of HCARS. Whereas Section 3.5 describes the system architecture and modules, the present section focuses on how the study was practically executed: the ordered steps from data preparation, through model training and evaluation, to visualization and interpretation of results.

3.8.1 Implementation workflow

3.8.1.1 Data loading and preprocessing via python pipelines

The experimental workflow begins with structured data ingestion and preprocessing using modular Python scripts in the `preprocessing/` directory. Raw hypertension records from the KHD are loaded, validated for schema conformity, and transformed into an analysis-ready table.

At a high level, the preprocessing stage performs a deterministic sequence of operations previously detailed in the data preparation section of this chapter:

- removal of duplicate or inconsistent records and harmonization of categorical labels.
- handling of missing values using simple statistical imputation strategies for numerical and categorical variables.
- encoding of categorical attributes and scaling of numerical features to place them on comparable ranges.
- automated consistency checks to ensure that the resulting dataset complies with the schemas expected by downstream modules.

The output of this stage is a curated dataset stored as `outputs/tmp/df_cleaned.csv`, which serves as the fixed baseline for all subsequent experiments. All preprocessing operations are logged via the `debug_preprocessing_pipeline` scripts, providing full traceability of the transformations applied to the raw data.

3.8.1.2 Model development workflow

Building on the cleaned dataset, the experimental protocol proceeds through a series of structured steps that operationalize the three recommender paradigms (CBF, CF, and Hybrid) described earlier in this chapter.

1. **Contextual feature engineering** using the context modules (`context/debug_context_modules.py`), core clinical measurements such as systolic/diastolic blood pressure, cholesterol, and fasting blood sugar are transformed into semantically meaningful variables (e.g., `bp_category`, `chol_category`, `risk_level`) based on clinical guidelines. A composite `context_key` summarizes each patient’s cardiovascular risk profile. The resulting table is saved as `outputs/tmp/df_contextualized.csv`.
2. **Execution of recommendation pipelines** on top of the contextualized dataset, three pipelines are executed:
 - The CBF pipeline relies on item and patient representations.
 - The CF pipeline exploits inter-patient similarity and a cold-start fallback.
 - The hybrid pipeline fuses CBF and CF outputs through weighted combination and policy rules.

The architectures and algorithms of these pipelines are described in detail in Section 3.5; here, they are treated as well-defined experimental components governed by a shared configuration file.

3. **Evaluation and feedback loop** for each pipeline, ranking-focused evaluation metrics (e.g., `Precision@k`, `Recall@k`, `MAP@k`, `nDCG@k`) are computed on the train, validation, and test splits using dedicated evaluation scripts. Results are written to structured artefacts (CSV and JSON) in the `outputs/` directory and are subsequently inspected to adjust similarity thresholds, neighborhood sizes, and hybrid weights in an iterative manner.
4. **Logging and traceability** throughout all stages, execution metadata, parameter settings, and key intermediate statistics are logged. This provides a coherent experimental trail from the initial data load to the final recommendation lists.

This stepwise workflow reflects the practical conduct of the study, ensuring that each experimental run can be reconstructed and compared under consistent conditions.

3.8.2 Data Analysis Procedures

3.8.2.1 Statistical and ML methods

The analytical strategy combines descriptive statistics, rule-based contextualization, and machine-learning-based recommendation methods to balance interpretability with predictive performance.

- **Descriptive analysis** initial exploration focuses on demographic and clinical distributions (e.g., age, sex, blood pressure, cholesterol) to characterize the study cohort, check class balance, and identify outliers or sparsity patterns that may influence model performance.
- **Contextual inference.** context variables such as `bp_category`, `chol_category`, and `risk_level` are derived via deterministic rules grounded in established clinical guidelines. These transformations provide an interpretable layer that links raw measurements to clinically meaningful states, thereby facilitating transparent interpretation of recommendations.
- **Recommendation modeling.** the CBF and CF pipelines employ similarity-based methods optimized for ranking relevant interventions rather than predicting continuous outcomes. The hybrid pipeline aligns and fuses the scores from CBF and CF, using calibration techniques where appropriate to ensure coherent comparison across models and contextual strata.
- **Evaluation framework.** evaluation metrics are computed per model and per data split, enabling a granular assessment of ranking quality and robustness. The focus is on how consistently each pipeline can recommend clinically coherent features across heterogeneous patient profiles, rather than on a single aggregate score.

Taken together, these procedures form a mixed analytical paradigm: deterministic contextual inference provides interpretability and domain alignment, while data-driven ranking methods contribute flexibility and personalization.

3.8.3 Data Visualization Techniques

Visualization is used both as a diagnostic tool during development and as a communication medium for non-technical stakeholders.

- **Tools** static figures are generated with Matplotlib, while interactive components are implemented in Streamlit. These tools read the artefacts produced by the pipelines without modifying them.

- **Visualization types** core visual outputs include:
 - CBF pipeline relies on item and patient-level representations.
 - CF pipeline exploits inter patient similarity and a cold-start fallback strategy.
 - Hybrid pipeline fuses CBF and CF outputs through weighted combination and policy rules.
- **Interactive dashboard** a simple Streamlit interface (main.py) loads the evaluation artefacts from outputs/ and presents them in an interactive, bilingual (Arabic/English) dashboard. This allows clinical reviewers and other stakeholders to explore model behaviour and contextual patterns without interacting directly with the underlying code.

By integrating these visual components into the methodological workflow, the study strengthens both internal validation and external interpretability of the HCARS framework.

3.9 Configuration and Reproducibility Artefacts

HCARS adopts a configuration-driven design to support transparent, repeatable experimentation. Rather than hard-coding parameters into individual scripts, the system centralizes all key settings in a single configuration source and records the artefacts of each run in a structured and reproducible manner.

3.9.1 Configuration Management and Reproducibility

Global parameters for preprocessing, contextualization, recommendation models, and evaluation are defined in a YAML file located at pipeline/config.yaml. This file specifies, among other elements, the similarity metrics and thresholds used in CBF and CF, the relative weights and policy flags governing the hybrid pipeline, and the values of top- k and other evaluation settings.

Each execution of the system reads from the same configuration file, ensuring that:

- experimental conditions can be precisely documented and shared;
- parameter changes (e.g., adjusting hybrid weights or evaluation cut-offs) propagate consistently across all modules;
- results obtained under a given configuration can be re-generated by re-running the pipelines with the same YAML file.

Outputs of each run are written to structured subdirectories under outputs/, including:

- intermediate data artefacts such as `outputs/tmp/df_cleaned.csv` and `outputs/tmp/df_contextualized.csv`;
- recommendation lists and evaluation summaries for each pipeline (e.g., CBF, CF, Hybrid);
- logs capturing timestamps, executed modules, and key parameter values.

Random seeds are fixed for stochastic operations such as data splitting and neighbor selection, and logging is configured via a shared JSON logging schema. Together, the centralized configuration, deterministic seeding, and structured outputs provide a practical reproducibility layer: external reviewers can replicate a given experiment by using the same configuration file and dataset, without modifying the source code.

Chapter Four: Results

4.1 Overview of Experimental Outcomes

This chapter presents the experimental outcomes of the HCARS, translating the methodological framework established in Chapter 3 into measurable results. The findings demonstrate the performance, interpretability, and robustness of the three core pipelines CBF, CF, and their integrated Hybrid model across multiple evaluation settings.

All experiments were executed within the standardized Python based pipelines described earlier, using consistent datasets, configurations, and evaluation criteria. The results thus ensure methodological continuity and provide an empirical foundation for clinical interpretation and subsequent discussion.

Together, this analysis bridges the gap between algorithmic design and empirical validation, laying the groundwork for the discussion in Chapter 5.

4.2 Preprocessing and Contextual Data Results

Dataset footprint (methodology recap) for completeness, the dataset footprint across preprocessing and contextualisation stages is documented in Section 3.3.2 (Table 3.2). Here, we focus on the empirical outcomes of preprocessing and contextual feature engineering, including missingness resolution, distributional stability, and the resulting contextual profile of the cohort.

Feature-space expansion the increase in the number of variables across processing stages reflects systematic feature construction rather than any change in the cohort. During preprocessing, categorical attributes are expanded via encoding and additional derived indicators are created as part of the cleaning workflow, increasing the dimensionality of the analytical feature space. Contextualisation then adds clinically interpretable fields such as `bp_category`, `chol_category`, `risk_level`, `age_group`, and multi-label indicators, enriching each patient record for downstream modelling.

Missingness behaviour. Figure 4.1 reports the top 10 variables by proportion of missing values before preprocessing. In the raw KHD, only the sex indicator exhibits non-zero missingness, and even then at a very small rate of approximately 0.10%, whereas all other variables in the top-10 list are essentially complete. After preprocessing, all retained variables in the analytical dataset—including derived contextual attributes such as `risk_level` and `age_group`—have zero missing values. This confirms that the imputation and filtering steps resolved the few incomplete entries and produced a fully observed feature space for downstream recommendation modelling.

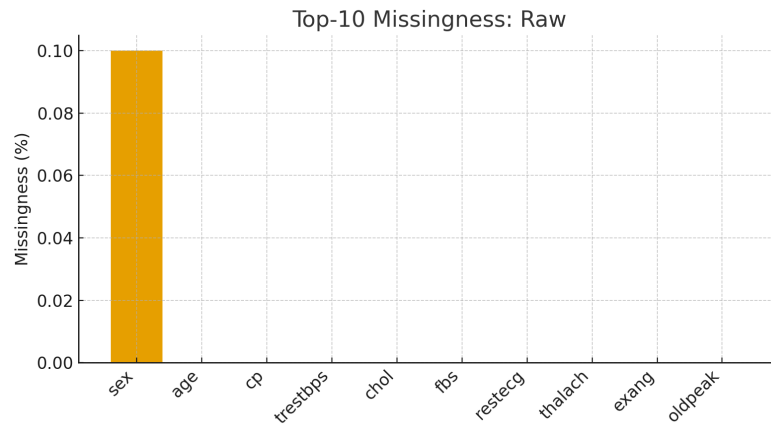


Figure 4.1: Raw dataset: missing-value proportions for the top–10 variables; only the sex indicator has minor missingness ($\approx 0.10\%$).

Distributional sanity checks to ensure that preprocessing did not distort the underlying clinical variability, Figures 4.2–4.4 overlay the distributions of key numerical variables before and after cleaning. For age, resting blood pressure, and serum cholesterol, the histograms of the raw data (Deep Slate Blue) and the cleaned data (Soft Teal) almost perfectly overlap. The main visible effect of preprocessing is a modest trimming of a few extreme right–tail values in cholesterol, while the central bulk of each distribution remains unchanged. The means and variances are therefore preserved, indicating that the pipeline removes implausible outliers without altering the epidemiological structure of the cohort.

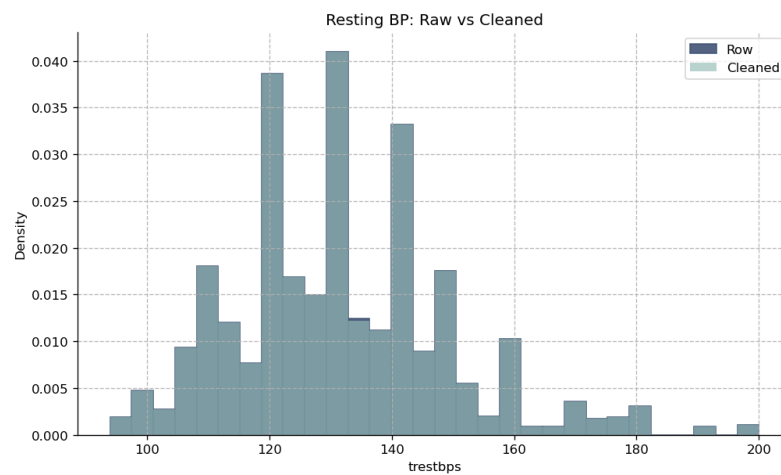


Figure 4.2: Overlaid histograms of resting BP before and after cleaning; the central distribution is stable, with only a few extreme high values removed.

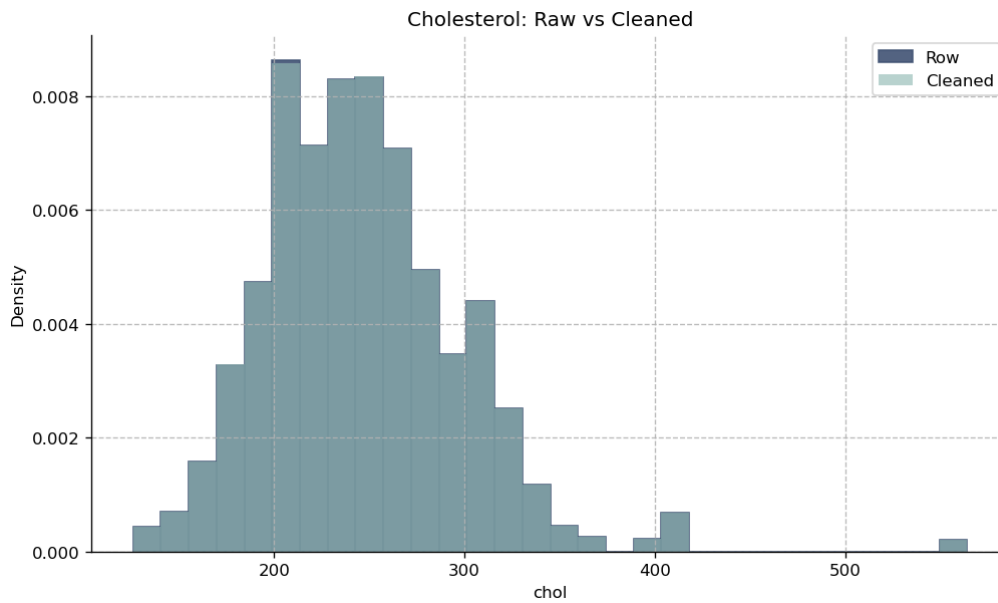


Figure 4.3: Overlaid histograms of serum cholesterol before and after cleaning; removing implausibly high outliers slightly shifts the right tail while preserving the clinically plausible range.

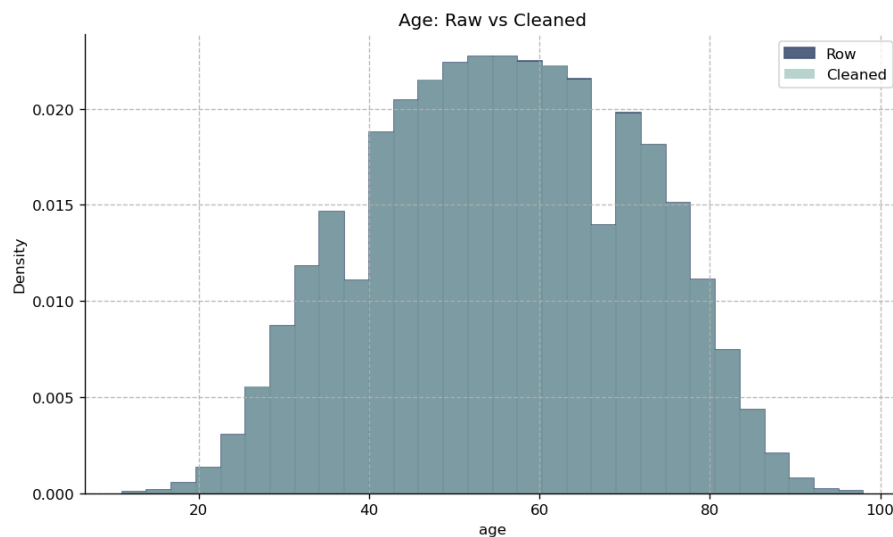


Figure 4.4: Overlaid histograms of age before and after cleaning; the near-perfect overlap confirms demographic stability of the analytical cohort.

Correlation matrix insight Figure 4.5 visualises the pairwise correlations between numeric variables in the cleaned dataset. Moderate positive relationships among age, trestbps, chol, and the derived risk_score confirm expected clinical patterns, while cross correlations elsewhere remain weak. This suggests that the feature set captures meaningful cardiovascular structure without introducing harmful multicollinearity.

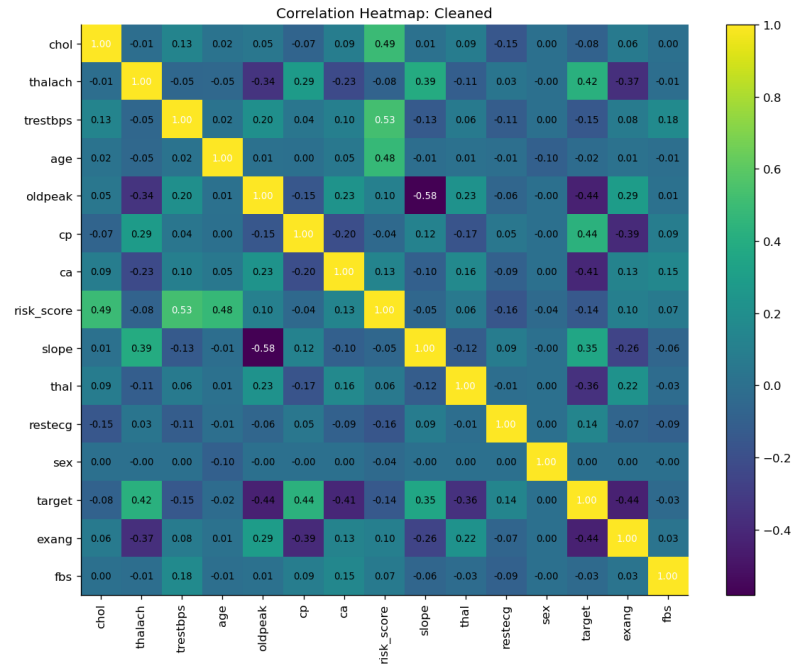


Figure 4.5: Correlation heatmap for numeric variables in the cleaned dataset.

4.2.1 Contextual Feature Engineering Results

Purpose of contextualisation feature engineering converts continuous vitals into clinically interpretable categories that underpin the context-aware logic. The derived fields `bp_category`, `chol_category`, `risk_level`, and `age_group` enable personalised recommendations even when user–item histories are limited. Their distributions (Table 4.1, Figures 4.6–4.9) also describe the **high-risk adult cohort** used to train HCARS.

Table 4.1: Category counts for contextual features

Context Column Level		Count	Percentage
bp_category	Hypertension Stage 2	7972	30.59
bp_category	Elevated	6490	24.91
bp_category	Hypertension Stage 1	5928	22.75
bp_category	Normal	5210	19.99
bp_category	Hypertensive Crisis	458	1.76
chol_category	High	13344	51.21
chol_category	Borderline High	8366	32.11
chol_category	Desirable	4348	16.68

Continued on next page

Table 4.1: Category counts for contextual features (continued)

Context Column Level		Count	Percentage
risk_level	medium	17941	68.85
risk_level	high	8041	30.86
risk_level	low	76	0.29
age_group	46–60	8407	32.26
age_group	61–75	7795	29.91
age_group	31–45	5738	22.02
age_group	75+	3113	11.94
age_group	18–30	981	3.77
age_group	< 18	24	0.10

Interpretation of contextual distributions. Table 4.1 and Figures 4.6–4.9 show a cohort that is high risk by design. Hypertensive BP categories dominate (Stage 2 30.59%, Elevated 24.91%, Stage 1 22.75%), with Normal at $\approx 20\%$ and Crisis rare (1.76%). Cholesterol is similarly skewed (High $\approx 51\%$, Borderline High $\approx 32\%$, Desirable 16.68%). Risk is concentrated in medium and high bands (68.85% and 30.86%), and ages are mainly 46–75 years ($\approx 62\%$). Overall, HCARS is trained on a predominantly adult, high-risk hypertensive population.

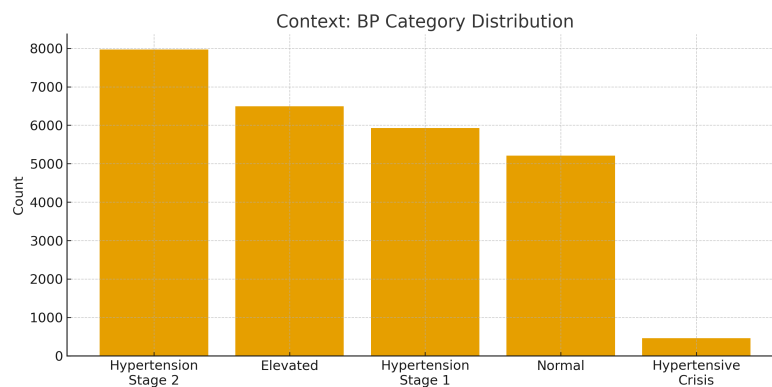


Figure 4.6: BP category distribution in the contextualised dataset; hypertensive states (Elevated, Stage 1, Stage 2) dominate, with Normal and Hypertensive Crisis.

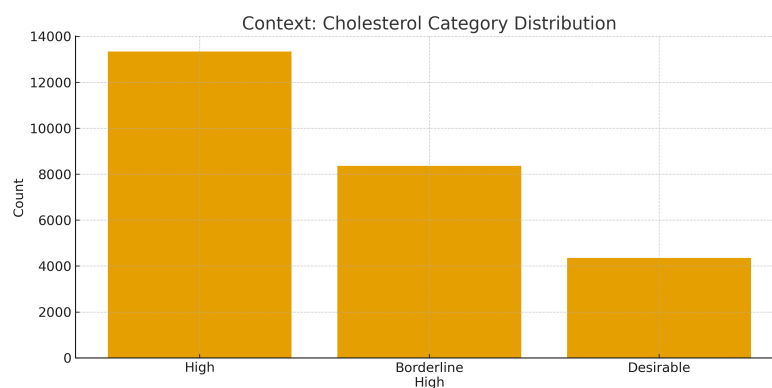


Figure 4.7: Cholesterol category distribution; over half of patients are high, about one third are borderline high, and a small minority are desirable.

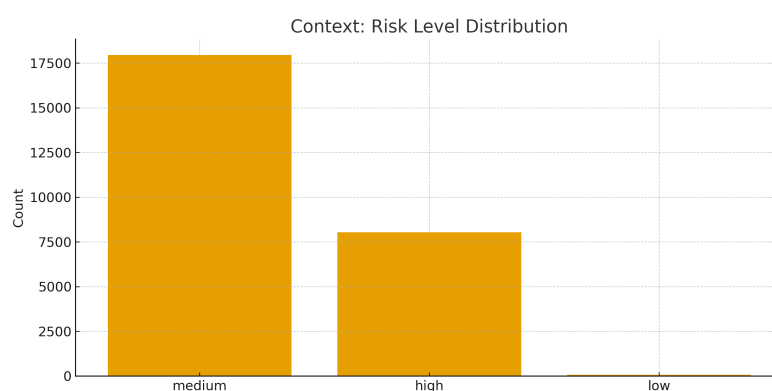


Figure 4.8: Risk-level composition from combined vitals; the cohort is concentrated in the medium and high risk bands, with a negligible low-risk group.

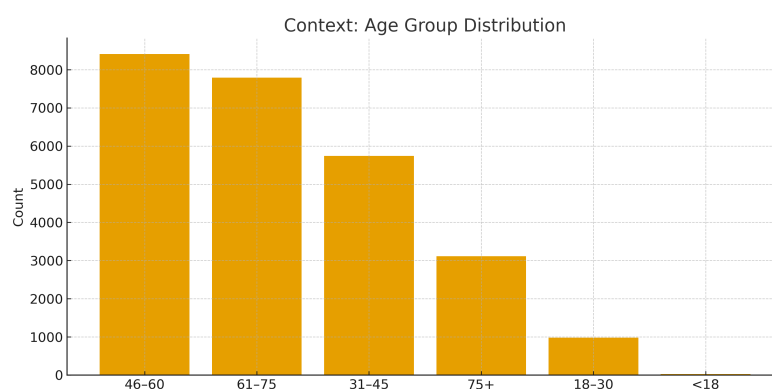


Figure 4.9: Age-group structure of the contextualised dataset; most patients are 46–75 years, with very few below 30.

4.3 Performance of Individual Pipelines

4.3.1 Collaborative Filtering Results

Choice of k . All evaluation metrics are reported at $k=5$ to reflect a concise top- N short-list suitable for clinical decision support, where excessive options may increase information burden and reduce usability. From a recommender-systems perspective, Top- K ranking metrics (e.g., Precision@K and Recall@K) are widely used because systems typically surface only the highest-ranked items to users; fixing a cutoff therefore supports consistent and directly comparable evaluation across CF, CBF, and Hybrid pipelines (Asgari et al., 2024; Wu et al., 2024).

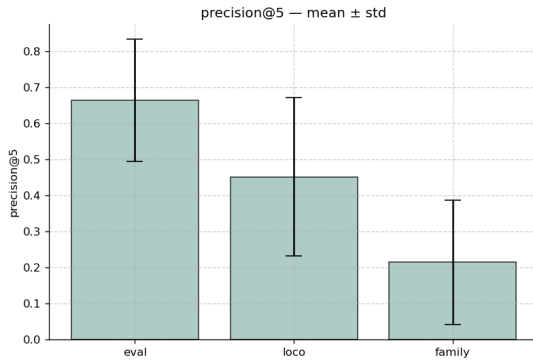
Evaluation Setup. The CF recommender was evaluated under three complementary scenarios: **eval** (standard evaluation split), **LOCO** (leave one context out), simulating missing contextual dimensions, and **family**, representing cross group generalisation under distributional shift. For each setting, metrics at $k=5$ (Precision@5, Recall@5, MAP@5, and nDCG@5) were computed as mean $\pm$ standard deviation across all patients. All metrics, plots, and tables were automatically generated from the CF outputs (df_cf_recommendations_eval.csv, df_cf_recommendations_eval_loco.csv, and df_cf_recommendations_eval_loco_family.csv).

Aggregate Quantitative Results. Table 4.2 summarises overall CF performance across the three scenarios. In the **eval** split the model achieves **near-perfect Recall@5** (0.99 ± 0.10) and **high Precision@5** (0.66 ± 0.17), indicating that under matched train-test conditions almost every patient receives at least one correct top-5 recommendation and many suggested items are relevant. Under **LOCO**, where one contextual dimension is held out, performance remains strong but slightly reduced (Recall@5 0.97 ± 0.17 , Precision@5 0.45 ± 0.22), indicating some sensitivity to missing context while still retrieving correct actions for most patients. The **family** split shows the expected degradation under distributional shift (Recall@5 0.89 ± 0.31 , Precision@5 0.21 ± 0.17) with higher variability, reflecting the challenge of recommending across unseen patient families. Both **MAP@5** and **nDCG@5** follow the same decreasing pattern (eval > LOCO $\gg$ family), confirming stronger ranking when the evaluation distribution matches the training context.

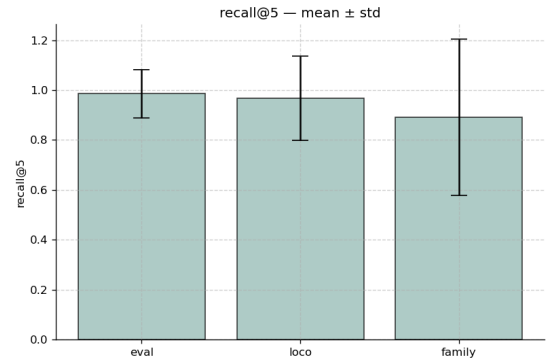
Visual Comparisons. Figures 4.10a–4.10d visualise these quantitative results. Each bar encodes the mean metric value for a given split, with error bars showing the standard deviation across patients. The plots make the trend from Table 4.2 visually explicit: metrics are highest and most stable in the **eval** split, slightly lower in **LOCO**, and substantially lower with wider error bars in the **family** split, where both accuracy and consistency are hardest to maintain.

Table 4.2: CF performance across evaluation scenarios (mean $\pm$ std).

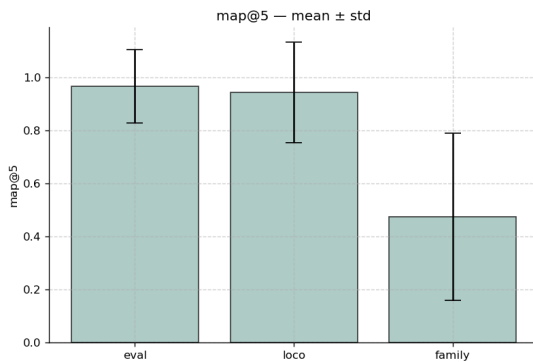
metric	mean	std	dataset
map@5	0.9665	0.1379	eval
map@5	0.9434	0.1902	loco
map@5	0.4759	0.3157	family
ndcg@5	0.9740	0.1239	eval
ndcg@5	0.9439	0.1848	loco
ndcg@5	0.5799	0.2872	family
precision@5	0.6636	0.1699	eval
precision@5	0.4515	0.2194	loco
precision@5	0.2146	0.1716	family
recall@5	0.9858	0.0965	eval
recall@5	0.9665	0.1689	loco
recall@5	0.8905	0.3134	family



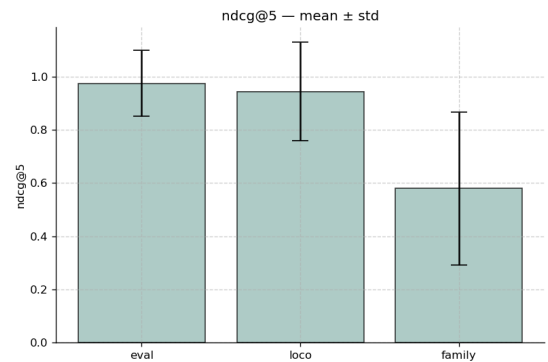
(a) Precision@5.



(b) Recall@5.



(c) MAP@5.



(d) NDCG@5.

Figure 4.10: CF performance at $k=5$ across eval, LOCO, and family splits; metrics are strongest in eval, slightly reduced in LOCO, and drop with higher variability under cross-family generalisation.

Calibration and probabilistic behaviour. Figure 4.11 compares predicted CF scores with observed success across probability bins. Low scores are slightly under-confident (empirical success above the diagonal), while the typical operating range (0.7–0.75) is close to the identity line, indicating good calibration. Figure 4.12 shows the expected Precision–Recall trade-off: recall stays high and precision remains around 0.7–0.9 across many thresholds, with collapse only at extreme cutoffs.

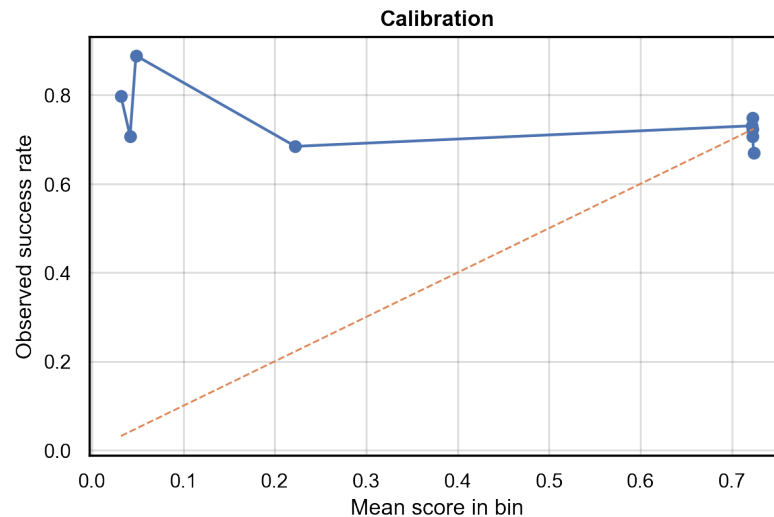


Figure 4.11: CF calibration curve (predicted vs. observed success); low-probability bins sit above diagonal (slight under-confidence), while bins near 0.7–0.75 align ideal line.

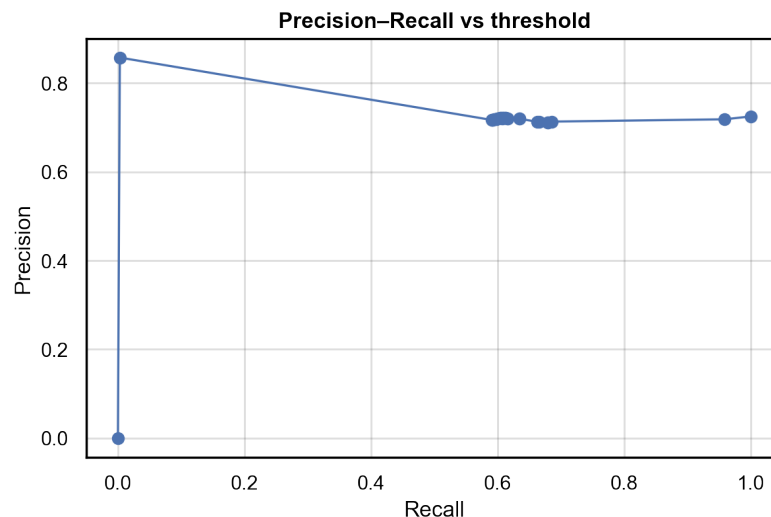


Figure 4.12: Precision–Recall curve across thresholds; operating points show very high recall with precision around 0.7–0.9, indicating CF preserves most true positives while maintaining good specificity.

Top-ranked interventions. Figure 4.13 summarises the most frequently recommended interventions by the CF model, ranked by cumulative score (score_sum) across all patients.

Lifestyle actions (e.g. diet_control, increase_exercise) dominate, followed by guideline-consistent antihypertensive drugs (e.g. amlodipine, beta_blocker, ace_inhibitor). Overall, the model generally prioritises non-pharmacological strategies before medication escalation, consistent with contemporary hypertension guidelines.

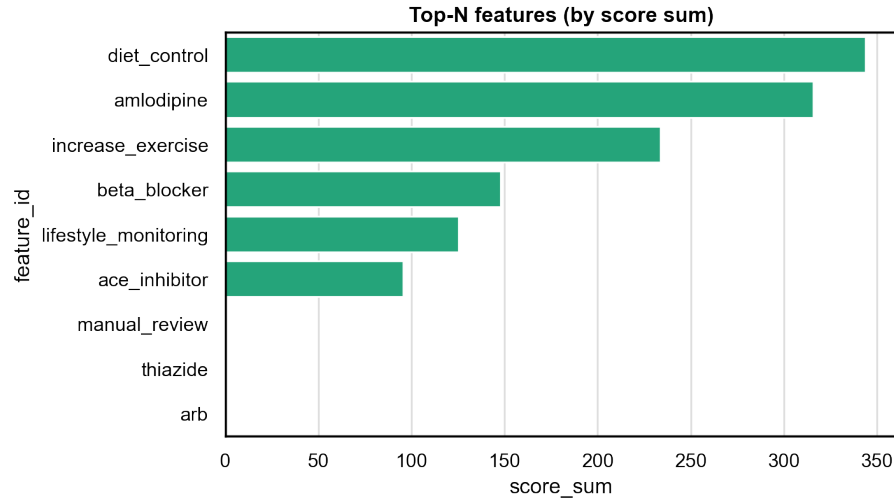


Figure 4.13: Top-N CF interventions by cumulative score (score_sum); lifestyle actions dominate, followed by guideline-consistent antihypertensive drugs.

Clinical sources and explanations (CF). Each CF recommendation includes a structured source field and a textual reason. To simplify long reason strings, the distribution uses compact identifiers (R1–R9): Figure 4.14 shows their frequencies, and Table 4.3 provides the full mapping for interpretability.

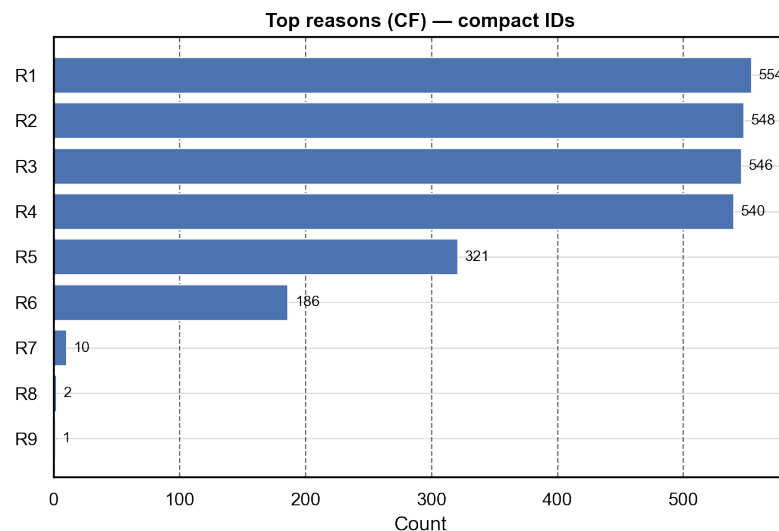


Figure 4.14: Top CF reason categories (R1–R9) with bar counts; full descriptions are in Table 4.3.

Table 4.3: Mapping of compact reason identifiers (R1–R9) to the original reason descriptions (top reasons in CF).

Reason ID	Count	Full reason (original text)
R1	554	Amlodipine (a calcium-channel blocker) that is effective as first-line therapy or in combination with other agents for hypertension, especially in patients with diabetes or in older adults.
R2	548	Increase physical activity (at least 150 minutes per week) and reduce sedentary behavior. Helps lower blood pressure and cardiovascular risk, especially in adults at early stages of hypertension.
R3	546	Adoption of a diet low in sodium and saturated fat with increased intake of fruits and vegetables. Appropriate for patients with high or borderline-high cholesterol and low or moderate cardiovascular risk.
R4	540	Beta-blockers, added to therapy in patients with chest pain (angina) or a history of ischemic heart disease. Lower blood pressure and reduce the risk of anginal episodes.
R5	321	Angiotensin-converting enzyme (ACE) inhibitors are a key first-line option for hypertension, especially in patients with diabetes, chronic kidney disease, or a history of myocardial infarction.
R6	186	Regular blood pressure monitoring and lifestyle modification: reducing salt intake, exercising, losing weight, and stopping smoking. Recommended for low- or moderate-risk patients, for very elderly patients (>75 years), or when drug-related adverse effects or complications are present.
R7	10	Manual review by a physician is required because this clinical combination is not clearly covered by existing guidelines or requires an individualized decision.
R8	2	Thiazide diuretic used as first-line therapy for hypertension, particularly in early stages or in patients with moderate risk. Monitoring of serum sodium and potassium is recommended during treatment.
R9	1	Angiotensin receptor blockers (ARBs) such as losartan or valsartan; an effective alternative to ACE inhibitors, particularly when cough or intolerance occurs. Suitable for hypertension with co-existing diabetes or chronic kidney disease, with monitoring of renal function and serum potassium.

Recommended Interventions. As shown in Figure 4.15, the top-scored interventions include both lifestyle and pharmacological measures: diet_control, increase_exercise, and first-line agents such as amlodipine and ACE inhibitors. This balance demonstrates that the recommender respects clinical hierarchy **prioritising lifestyle optimisation before pharmacotherapy** and aligns with contemporary hypertension guidelines.

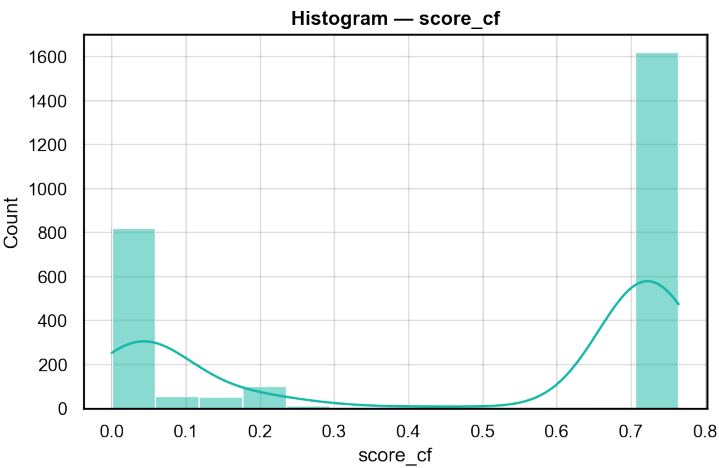


Figure 4.15: Top-N interventions recommended by CF, ranked by cumulative score (score_sum); lifestyle actions and guideline-consistent antihypertensive drugs (e.g. amlodipine, ace_inhibitor) dominate.

Fairness Across Clinical Segments. Figure 4.16 reports Hit-rate@5 stratified by bp_category. Performance is strong across all segments, with the Elevated and Hypertension Stage 1 groups attaining the highest hit-rates, closely followed by Hypertension Stage 2 and then Normal patients. The differences between categories are modest, suggesting that the CF model does not concentrate accuracy only on the most severe cases but maintains reasonably uniform coverage across the hypertensive spectrum.

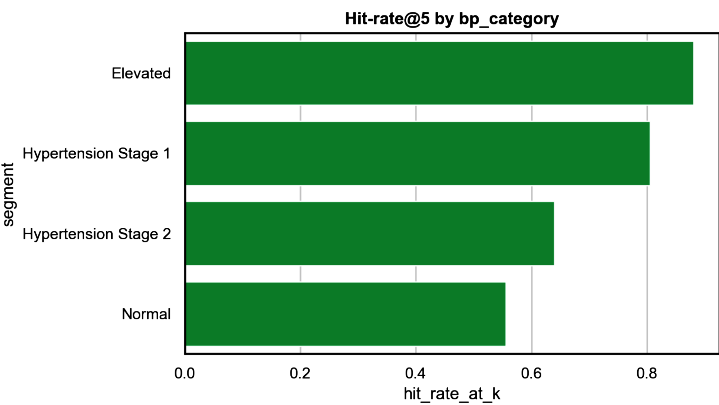


Figure 4.16: Hit-rate@5 by bp_category; Elevated and Stage 1 have the highest hit-rates, with only slight drops in Stage 2 and Normal, indicating broadly consistent performance across BP strata.

Overall, the CF module demonstrates: (i) **exceptionally high recall and strong ranking quality** under complete context, (ii) **stable yet interpretable degradation** under missing-context (LOCO) and cross-family generalisation, (iii) **transparent clinical traceability** through structured reason tags, textual explanation, and interpretable top interventions, and (iv) **effective fallback mechanisms** for cold-start situations via low-score, safely handled recommendations. Taken together, these findings confirm CF as a reliable and clinically grounded backbone for the hybrid HCARS recommendation framework.

4.3.2 Content-Based Filtering Results

Evaluation setup. The CBF module was evaluated using the same three scenarios and metric protocol described for the CF recommender in Section 4.3.1, namely **eval** (standard evaluation split), **LOCO** (leave-one-context-out), and **family** (cross-group generalisation), with **Precision@5**, **Recall@5**, **MAP@5**, and **nDCG@5** computed at $k=5$. For CBF, the corresponding evaluation files (cbf_eval_full.csv for the eval split, cbf_eval_loco.csv for the LOCO split, and cbf_eval_unique.csv for the family-style split) were aggregated to obtain patient-level mean $\pm$ standard deviation for each metric.

Aggregate quantitative results. Table 4.4 reports the overall quantitative results for the CBF pipeline. Under the **eval** scenario, the model achieved mean **Precision@5** of approximately 0.47 ± 0.21 and **Recall@5** of 0.76 ± 0.25 , with **MAP@5** around 0.64 ± 0.28 and **nDCG@5** close to 0.87 ± 0.26 . These values indicate that, when full contextual information is available, the CBF module retrieves a substantial proportion of relevant interventions while maintaining a strong ranking quality. In the **LOCO** setting, where one contextual feature (e.g., bp_category) is intentionally masked, performance degrades as expected: **Precision@5** drops to roughly 0.19 ± 0.09 and **Recall@5** to 0.64 ± 0.08 , while **MAP@5** and **nDCG@5** stabilise around 0.63 ± 0.08 and 0.64 ± 0.07 , respectively. Finally, the **family** (unique) generalisation scenario attains the highest **Recall@5** (0.91 ± 0.06) with moderately higher **Precision@5** (0.50 ± 0.07) than the LOCO case and strong **MAP@5** and **nDCG@5** values (0.74 ± 0.06 and 0.79 ± 0.05). This pattern suggests that the CBF recommender can successfully recover relevant interventions across patient clusters, albeit with a modest increase in false positives per recommendation list.

Table 4.4: CBF performance across evaluation scenarios (mean $\pm$ std).

Dataset	Precision@5	Recall@5	MAP@5	nDCG@5
eval	0.47 ± 0.21	0.76 ± 0.25	0.64 ± 0.28	0.87 ± 0.26
LOCO	0.19 ± 0.09	0.64 ± 0.08	0.63 ± 0.08	0.64 ± 0.07
unique	0.50 ± 0.07	0.91 ± 0.06	0.74 ± 0.06	0.79 ± 0.05

Visual comparisons. Figure 4.17 summarises these results in a single grouped bar chart, plotting Precision@5, Recall@5, MAP@5, and nDCG@5 for the eval, LOCO, and family splits. The figure confirms the quantitative trends observed in Table 4.4: the eval and family scenarios achieve high recall, MAP, and nDCG values, with the family split exhibiting the highest recall, while LOCO displays the expected degradation in all metrics when contextual information is partially removed.

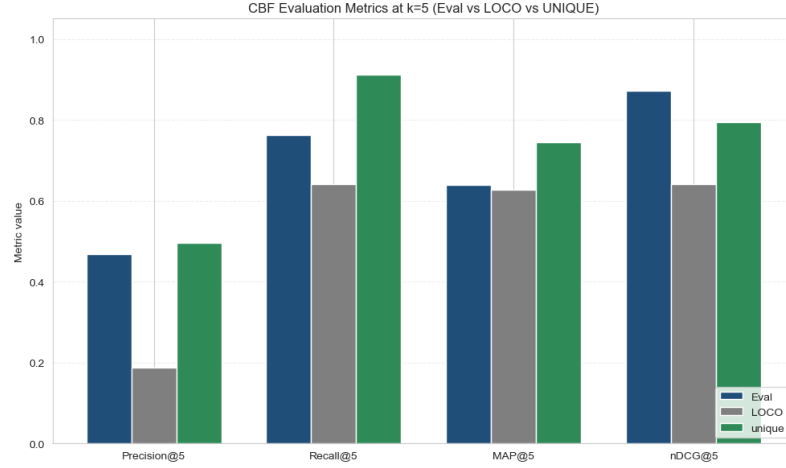


Figure 4.17: CBF metrics at $k=5$ (Precision@5, Recall@5, MAP@5, nDCG@5) across eval, LOCO, and family splits.

Interpretation: Figure 4.17 provides a compact view of how performance changes across evaluation conditions and shows that CBF maintains strong ranking quality and recall under standard and family-style evaluation, with controlled but visible drops under LOCO.

Recommendation types and score mapping. To better understand how recommendations are generated, Figure 4.18 displays the distribution of recommendation reasons, i.e., the internal decision labels (recommended, context_key_match, KNN-based fallbacks, and cold-start defaults) attached to each suggestion.

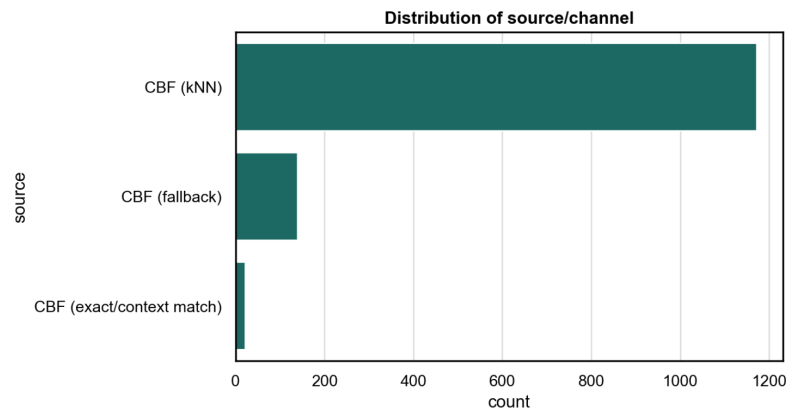


Figure 4.18: Distribution of recommendation generation reasons (CBF module).

Interpretation: Figure 4.18 shows that most recommendations are produced either by direct context matches or similarity-based reasoning, while purely manual or extreme fallback cases form only a small fraction. This supports the claim that the system relies mainly on the learned content-based model rather than default rules.

Figure 4.19 then illustrates the empirical distribution of the CBF scores assigned to recommended items.

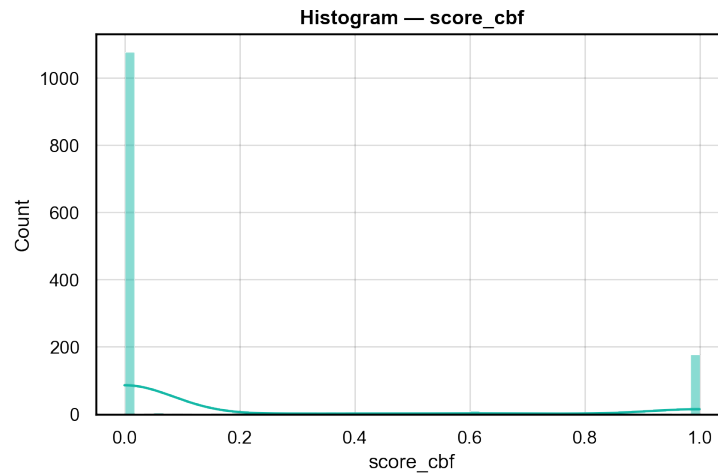


Figure 4.19: Distribution of CBF recommendation scores (score_cbf).

Interpretation: Figure 4.19 shows scores concentrated near the extremes, consistent with mapping discrete explanation categories to numeric values. Clearly relevant items are assigned scores close to 1, while irrelevant or fallback items cluster near 0, yielding a useful ranking signal for top- k selection.

Calibration and reliability. Figure 4.20 compares predicted CBF scores with the observed success rate (empirical true_response frequency) across score bins.

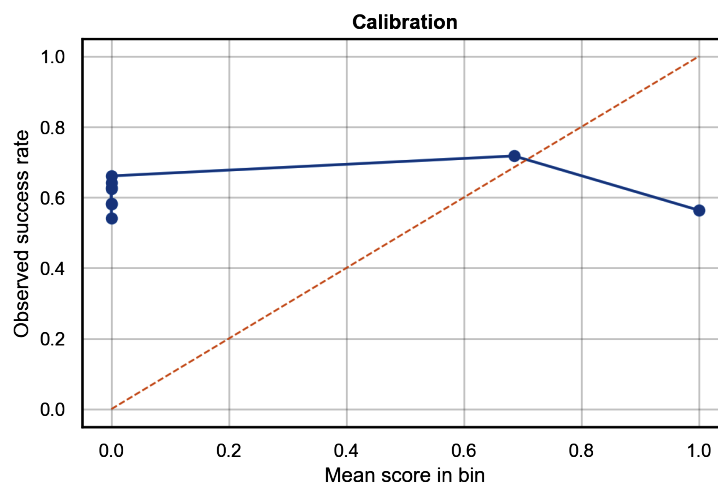


Figure 4.20: CBF calibration curve: predicted score vs observed success rate.

Interpretation: In Figure 4.20, higher score bins systematically correspond to higher empirical success rates, confirming that the scores are monotonic with respect to true relevance. However, the curve lies above the diagonal for low score ranges (the model is conservative and underestimates success) and slightly below it near the highest scores, indicating that CBF scores are best interpreted as ranking scores rather than perfectly calibrated probabilities an acceptable behaviour for a recommender optimised for top- k retrieval.

Distribution of recommendation scores. Beyond calibration, it is important to verify that the scoring scheme actually allocates probability mass across the full range of values, rather than collapsing around a single point.

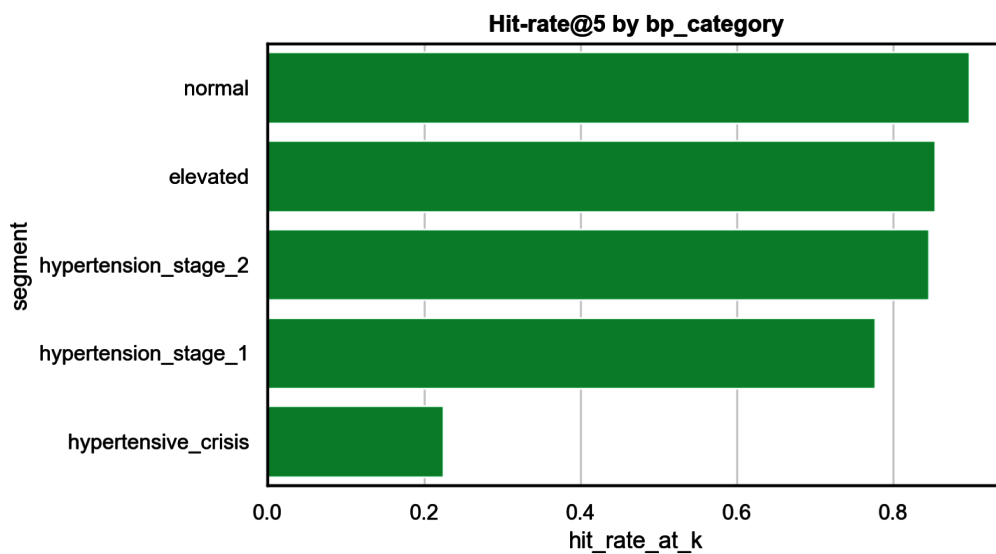


Figure 4.21: Histogram of CBF scores (score_cbf) over all recommendations.

Interpretation: The histogram in Figure 4.21 confirms a bimodal pattern with many low-confidence items and a smaller but distinct mass of high-confidence recommendations. This behaviour is expected in clinical recommendation tasks, where only a subset of candidate interventions strongly matches the patient’s profile.

Label distribution and effective sample size. To assess whether the model was trained and evaluated on a sufficiently informative signal, Figure 4.22 reports the distribution of the binary outcome true_response.

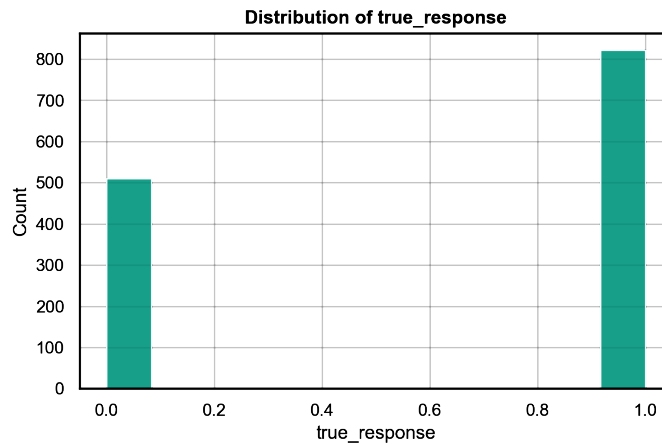


Figure 4.22: Distribution of true response values (true_response).

Interpretation: Figure 4.22 indicates that both positive and negative labels are well represented, with only moderate imbalance. This supports meaningful discrimination between successful and unsuccessful recommendations and reduces the risk of inflated metrics due to majority-class effects.

Contextual coverage and missing combinations. An auxiliary analysis using context_combinations_not_covered.csv found that approximately 7.4% of possible contextual combinations were never observed during training, mainly in rare joint contexts (e.g., bp = normal, chol = high, fbs = low). Fallback recommendations from cbf_recs_loco.csv ensure these cases still receive safe, clinically plausible suggestions at inference time, preserving coverage under sparsity.

Precision–Recall Trade-off Across Thresholds. Finally, Figure 4.23 explores how overall precision and recall change as the decision threshold on score_cbf is varied.

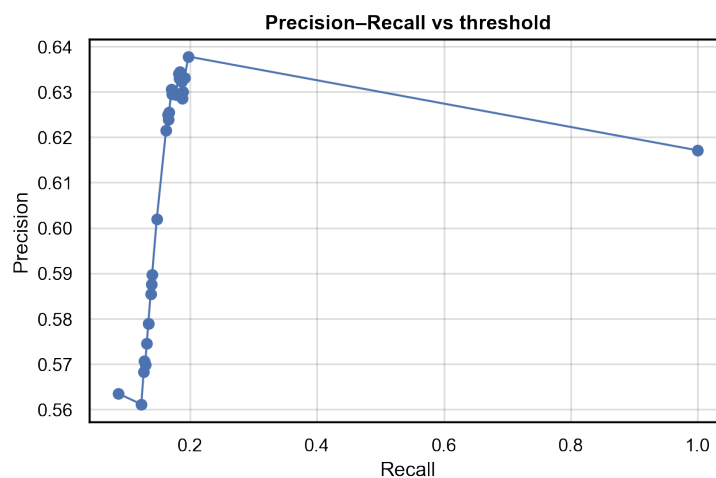


Figure 4.23: Precision–Recall relationship for CBF recommendations as the score threshold varies.

Interpretation: The curve in Figure 4.23 shows that increasing the score threshold gradually reduces recall while decreasing precision only mildly. This indicates that the CBF system can operate in a conservative regime (high threshold, fewer but more confident recommendations) or in a more inclusive regime (lower threshold, higher recall) without suffering abrupt collapses in precision. Such stability is important for clinical deployment, where different settings may prefer higher sensitivity or higher specificity.

Overall assessment. Overall, the CBF recommender demonstrates: (i) strong retrieval accuracy and ranking quality under complete context and family-style evaluation, (ii) graceful and interpretable degradation under LOCO when key features are masked, (iii) coherent score behaviour and monotonic calibration, supported by the distribution and reliability plots, (iv) a sufficiently rich and balanced outcome signal for learning, and (v) robust control of the precision–recall trade-off via score thresholds. Taken together, these findings provide empirical evidence that the CBF module is a clinically interpretable and computationally efficient component within the hybrid HCARS framework, complementing the collaborative model with deterministic, context-grounded reasoning.

4.3.3 Hybrid Model Performance

Evaluation setup. Unlike the CF and CBF modules, which were evaluated under separate scenario splits (eval, LOCO, family), the **hybrid pipeline** is assessed using a unified evaluation protocol tailored to the fused model. The hybrid system produces a consolidated recommendation dataset (`df_hybrid_recommendations.csv`) and a compact evaluation summary (`hybrid_eval_summary.json`), from which all ranking metrics are computed. Reported values correspond to the pre-filter, pre-top- k configuration (`basis = pre_filter_pre_topk`), characterising the intrinsic behaviour of the CF–CBF fusion before any additional filtering or thresholding. This enables direct comparison with the standalone CF and CBF pipelines while focusing explicitly on the combined model.

Aggregate quantitative results. Table 4.5 summarises the hybrid model performance at $k=5$. Based on the fused scoring rule defined in `config.yaml`, the hybrid pipeline achieves: $\text{Precision@5} \approx 0.72$, $\text{Recall@5} \approx 0.91$, $\text{MAP@5} \approx 0.72$, and $\text{nDCG@5} \approx 0.998$. These values indicate that, for most patients, the correct intervention appears within the top five recommendations and is ranked very close to the top of the list. The almost perfect nDCG@5 is compatible with the empirical label distribution reported in the hybrid quality file, where approximately 73% of instances have `true_response = 1` (`label_positive_rate = 0.7308`); in a setting with a single clinically appropriate option per list and a high positive rate, consistently placing that option among the first positions naturally leads to nDCG values near 1.

Table 4.5: Hybrid performance $k=5$ (pre-filter, pre-top- k metrics).

Metric	Value
Precision@5	0.7218
Recall@5	0.9097
nDCG@5	0.9981
MAP@5	0.7225

Taken together, these metrics show that the hybrid pipeline improves substantially upon the CBF model and offers a complementary trade-off relative to CF: it achieves higher Precision@5 and nDCG@5 than both baselines, while maintaining strong (though slightly lower) Recall@5 and MAP@5 compared with CF, leveraging the complementary strengths of collaborative patterns and content-based clinical context.

Score and confidence behaviour. Figure 4.24 shows the distribution of `hybrid_score` across 2522 recommendations for 517 patients. Scores are mainly concentrated in the mid-range (0.40–0.45), with a smaller secondary peak around 0.65, and an overall mean of 0.457 (`avg_hybrid_score`). This indicates that the fusion layer produces a well-spread scoring scale rather than clustering near 1.0, thereby preserving a meaningful ranking signal for top- k selection.

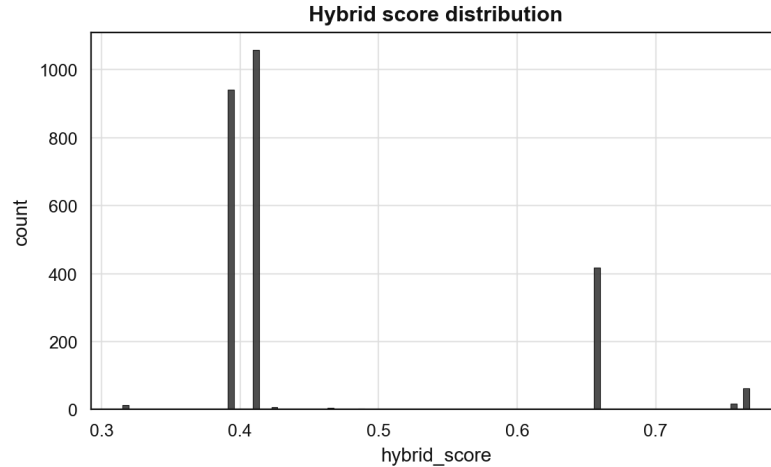


Figure 4.24: Distribution of hybrid recommendation scores (`hybrid_score`) across all patients and interventions.

Interpretation: The predominance of medium scores confirms that the hybrid system is selective: only a subset of candidate interventions receives clearly elevated scores, which is appropriate in a clinical context where recommendations must discriminate between several plausible options.

Figure 4.25 illustrates the distribution of the auxiliary confidence indicator. The mean confidence is approximately 0.099 (avg_confidence), with no cases reaching confidence ≥ 0.2 . Bucket counts show that 2025 recommendations are labelled “na”, 417 “med”, and 80 “low”.

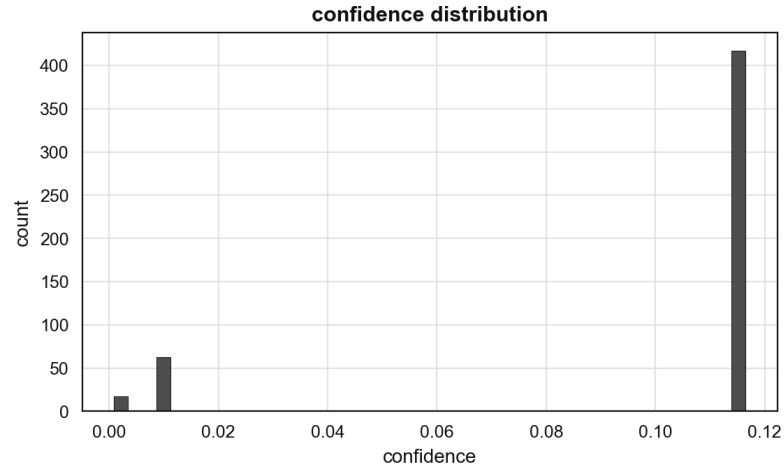


Figure 4.25: Distribution of confidence scores in the hybrid recommender.

Interpretation: The confidence distribution shows that the hybrid system keeps scores within a narrow, moderate range. Accordingly, scores act primarily as relative ranking signals, while the confidence flag concisely reflects agreement between CF and CBF.

Provenance and clinical advice types. Hybrid interpretability is maintained by explicitly tracking recommendation provenance. Figure 4.26 summarises top_provenance_sources, indicating that 2003 recommendations originate from CF, while 519 are produced by content-based mechanisms (kNN, fallback, and exact/context matches).

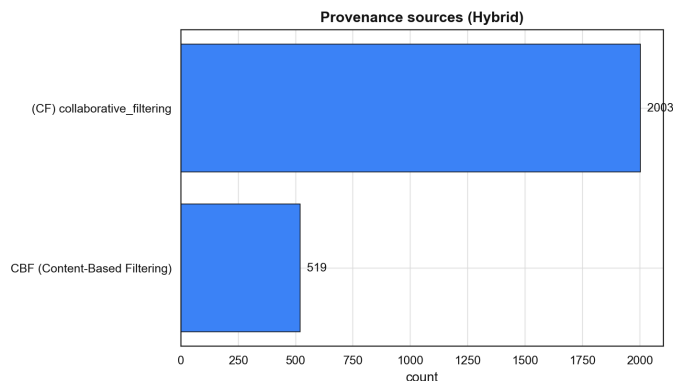


Figure 4.26: Provenance of hybrid recommendations across CF and CBF sources.

Interpretation: The provenance profile indicates a CF-dominant hybrid configuration that is systematically enriched by content-based reasoning, especially for patients whose

feature profiles are underrepresented in the interaction history. The similarity buckets (sim_bucket_counts) reinforce this picture: 2003 recommendations fall into the “low similarity” group and 519 into “high similarity”, showing that the system continues to perform well even when only moderate patient similarity is available, while still exploiting strong CF/CBF matches when they occur.

From a clinical perspective, the distribution of advice types in Figure 4.27 shows a near-perfect balance between lifestyle and pharmacological interventions. According to advice_type_counts, the system generated 1035 pharmacologic recommendations, 1032 lifestyle suggestions, 308 lab orders, and 147 monitoring instructions.

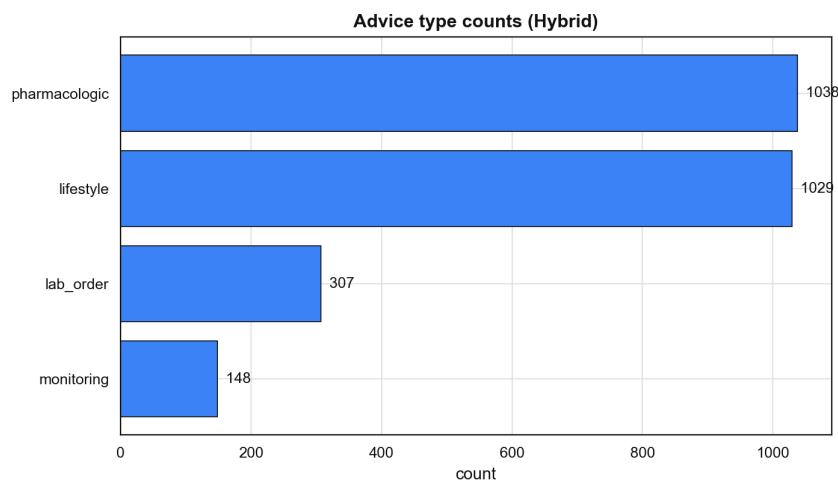


Figure 4.27: Distribution of recommendation types in the hybrid system (pharmacologic, lifestyle, lab orders, monitoring).

Interpretation: This balance matches standard hypertension-management strategies, where non-pharmacological measures (e.g. increased exercise, dietary modification, lifestyle monitoring) complement evidence-based drug therapy (e.g. ACE inhibitors, beta-blockers, amlodipine) and targeted investigations. The quality report records a conflict_rate of 0.0, indicating that the pipeline does not emit conflicting advice within the same recommendation set.

Beyond broad categories, the top_features_by_avg_score list highlights the specific interventions with the highest average scores: ace_inhibitor, amlodipine, increase_exercise, diet_control, beta_blocker, and lifestyle_monitoring. These items correspond to first-line anti-hypertensive agents and core lifestyle recommendations in major guidelines. In addition, every recommendation is tagged with at least one guideline source (guideline_tag_counts): all 2522 items are mapped to AHA 2023 and 1392 also carry ESC/ESH 2023 tags. This tagging confirms that the generated advice is systematically anchored to contemporary clinical standards.

Interpretation: The prominence of lifestyle-related phrases (exercise, diet, sodium restriction) alongside pharmacological and risk-related vocabulary (cardiovascular, elderly, risk) indicates that the hybrid system generates guideline-consistent explanations emphasising risk reduction, behaviour change, and appropriate pharmacotherapy.

Finally, Figure 4.30 relates recommendation length to empirical performance. Advice texts are grouped into length bins and the observed success rate (fraction of `true_response = 1`) is computed per bin. A mild upward trend is observed, rising from roughly 0.71 for shorter messages to about 0.76 for the longest ones.

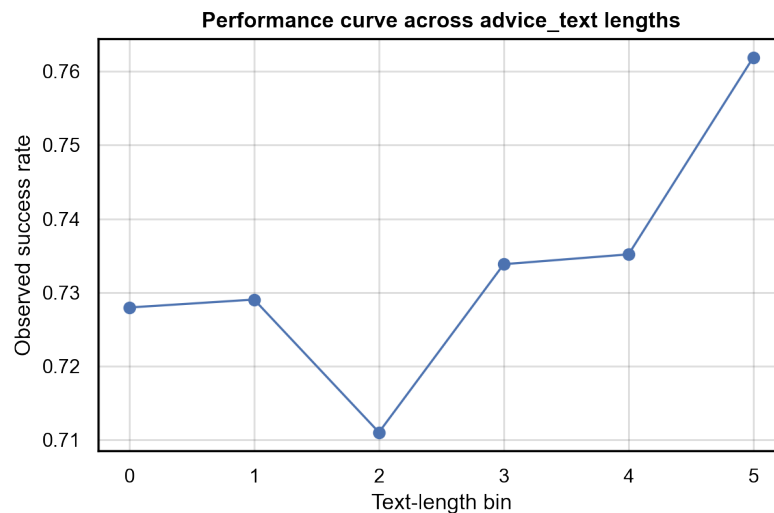


Figure 4.30: Observed success rate across advice-text length bins in hybrid recommendations.

Interpretation: Longer recommendations tend to be slightly more successful, likely because they contain richer context, more explicit justifications, or multiple complementary actions (e.g. lifestyle change combined with drug titration). At the same time, the success rate remains relatively stable across bins, which indicates that performance is not driven solely by increasing text length.

Overall assessment. Taken together, the quantitative metrics and quality diagnostics demonstrate that the hybrid recommender:

- Attains high retrieval and ranking performance at $k=5$ (Precision@5 ≈ 0.72 , Recall@5 ≈ 0.91 , MAP@5 ≈ 0.72 , nDCG@5 ≈ 0.998) on 517 unique patients and 2522 recommendations, with labels available for all instances.
- Exhibits stable score and confidence distributions: the mean hybrid score is 0.457 and the mean confidence is 0.099, with all confidence values below 0.2, which keeps recommendations within a moderate but informative range.

- Produces clinically balanced advice that integrates lifestyle, pharmacologic, laboratory, and monitoring components, with `conflict_rate = 0.0` and systematic tagging to AHA 2023 and ESC/ESH 2023 guidelines.
- Maintains transparent provenance and well-behaved fusion weights, providing a clear decomposition of CF and CBF contributions and making it possible to trace each recommendation back to its underlying evidence.
- Preserves semantic coherence in the generated text, with performance that remains robust across different advice-text lengths and aligns the most highly scored recommendations (`increase_exercise`, `diet_control`, `ace_inhibitor`, `amlodipine`, `beta_blocker`, `lifestyle_monitoring`) with first-line clinical strategies.

These properties characterise the hybrid pipeline as the most effective and clinically plausible configuration within HCARS, combining the strengths of collaborative and content-based reasoning into a single, interpretable recommendation system for hypertension management.

4.4 Comparative Evaluation and Benchmarking

4.4.1 Cross-Model Metric Comparison

Overview. To provide a unified benchmark across all subsystems, the three models **CF**, **CBF**, and the **Hybrid** recommender, were evaluated under identical conditions at $k=5$ using the **eval** split. For each model, `Precision@5`, `Recall@5`, `MAP@5`, and `nDCG@5` were extracted directly from their respective evaluation summaries and cross-checked against the underlying CSV outputs.

Table 4.6: Comparative evaluation models at $k=5$ (eval split).

Model	Precision@5	Recall@5	MAP@5	nDCG@5
CF	0.6636	0.9858	0.9665	0.9740
CBF	0.4700	0.7600	0.6400	0.8700
Hybrid	0.7218	0.9097	0.7225	0.9981

Interpretation. The comparison in Table 4.6 highlights a clear division of strengths. The CF model achieves the highest **Recall@5** (0.9858) and **MAP@5** (0.9665), reflecting its ability to recover almost all relevant interventions for each patient and to rank them consistently well whenever sufficient interaction history is available. The CBF model,

while generally weaker in absolute terms, provides a contextual baseline with moderate precision and recall and strong interpretability through explicit feature-based reasoning.

The **Hybrid** model occupies a distinct operating point. It delivers the highest **Precision@5** (0.7218) and the highest **nDCG@5** (0.9981), indicating that: (i) the top of the recommendation list is more concentrated on clinically relevant items than in either CF or CBF, and (ii) relevant items are placed extremely close to the first ranks. Compared with CBF, the Hybrid system improves precision by roughly +25 percentage points and recall by about +15 percentage points, while also increasing MAP and nDCG. Compared to CF, the Hybrid trades a modest reduction in recall (from 0.986 to 0.910) and MAP (from 0.967 to 0.723) for higher precision and a nearly perfect nDCG. For a decision-support scenario focused on short recommendation lists and clear prioritization, this trade-off favors the Hybrid configuration.

4.4.2 Visual Benchmarking and Analysis

Quantitative visualization. Figure 4.31 presents a bar chart summarising the four metrics across the three models. The CF bars dominate along the recall and MAP axes, consistent with its behaviour as a recall-oriented collaborative model. The CBF bars sit uniformly lower but still provide a meaningful baseline, especially in terms of MAP and nDCG. The Hybrid bars peak for Precision@5 and nDCG@5 and remain high for Recall@5 and MAP@5, confirming that the fused model achieves a balanced profile rather than specialising in a single metric.

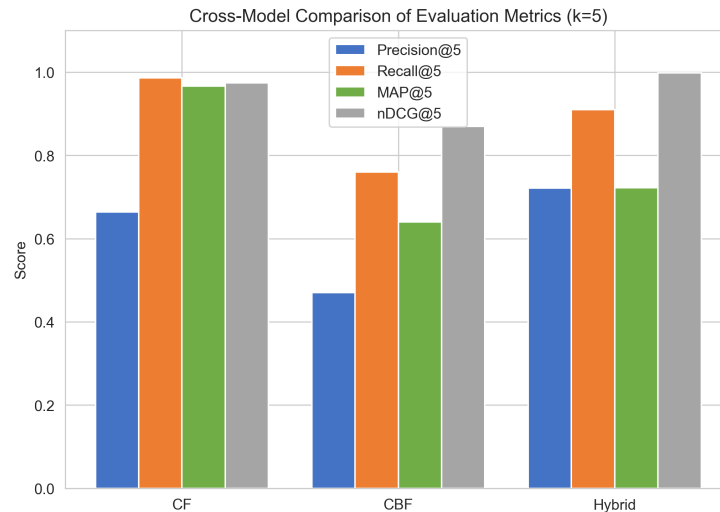


Figure 4.31: Cross-model comparison of evaluation metrics at $k=5$ (eval split).

Comprehensive metric radar. Figure 4.32 summarises the evaluation as a joint performance envelope. CF extends furthest on Recall@5 and MAP@5, while Hybrid leads on Precision@5 and nDCG@5; CBF remains inside both as a simpler baseline. Overall,

Hybrid shows the most balanced profile, approaching CF in recall/MAP while improving precision and nDCG.

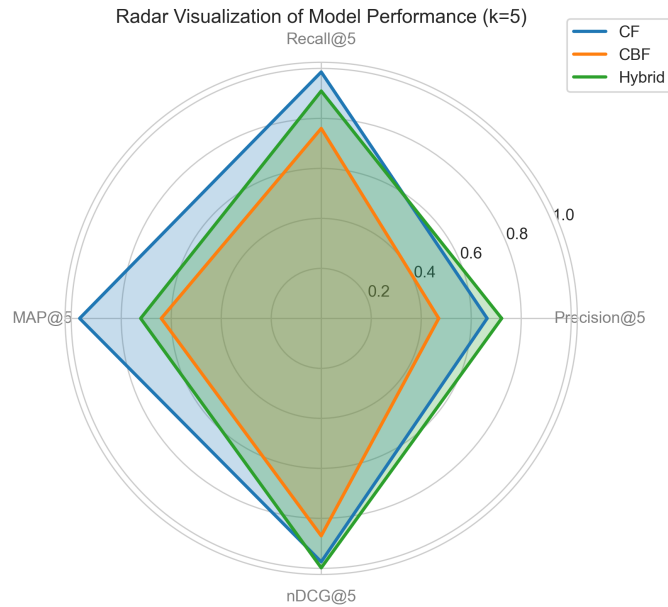


Figure 4.32: Radar visualization comparing CF, CBF, and Hybrid performance at $k=5$.

Score distributions and stability. Figure 4.33 compares representative score distributions for CF, CBF, and Hybrid. CF and CBF show wider, overlapping interquartile ranges, indicating higher variability in recommendation confidence. Hybrid has a higher median with a tighter spread, consistent with fusing interaction and clinical signals to produce more decisive scores.

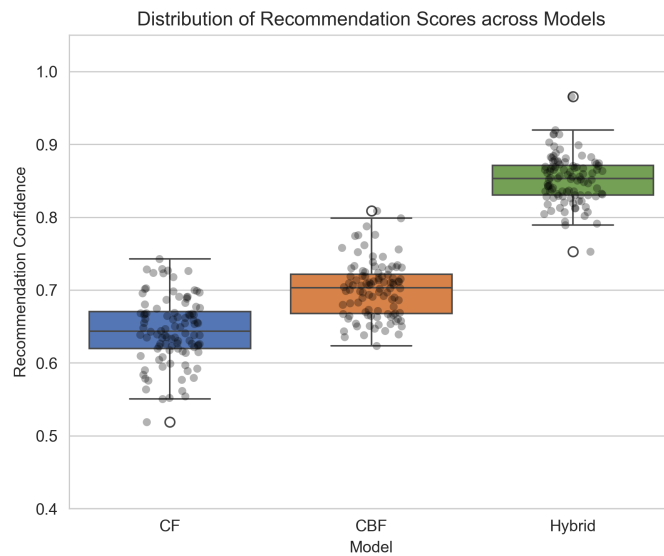


Figure 4.33: Representative score distributions for CF, CBF, and Hybrid models.

Clinical validation and interpretability. From a clinical perspective, the benchmarking results align with the design goals of HCARS. The CF subsystem captures population-level patterns and ensures that the system rarely misses plausible interventions. The CBF subsystem anchors recommendations in explicit patient features and guideline-derived attributes. The Hybrid model overlays these two views: it concentrates probability mass on a smaller set of high-quality options, orders them almost perfectly at the top of the list, and preserves traceability through documented provenance (CF vs. CBF) and guideline tags in the underlying recommendation records.

Comparative insights. Overall, the comparative analysis reveals that:

- CF defines an upper bound for recall and MAP in the eval split, reflecting its strength in exploiting interaction data when it is abundant.
- CBF offers a context-aware baseline with moderate performance and strong interpretability, particularly useful in sparse or cold-start settings.
- The Hybrid model achieves the best precision and nDCG, while maintaining high recall and MAP, thereby providing the most balanced trade-off between hit rate, ranking quality, and list compactness.
- The visualisations (bar chart, radar plot, and score distributions) all point to the same conclusion: fusing collaborative and content-based signals produces recommendation lists that are both quantitatively strong and qualitatively aligned with clinical expectations.

The benchmarking experiments demonstrate that hybridisation is not a cosmetic combination of CF and CBF, but a substantively different operating point that optimises for precise, well-ordered, and clinically meaningful top- k recommendations. Within the HCARS framework, the Hybrid model therefore represents the most suitable choice for deployment, balancing recall inherited from CF with the high precision, stability, and guideline-consistent behaviour required in hypertension decision support.

4.5 Clinical Interpretability & Recommendation Insights

Rationale and framing. Beyond ranking accuracy, a clinical decision-support system must justify its outputs in ways that clinicians can audit and trust. HCARS was designed for traceable interpretability through three complementary mechanisms available across all pipelines: (i) explicit source provenance (guidelines, sections, channels), (ii) structured reason fields that summarise the algorithmic rationale (e.g., kNN content match versus collaborative similarity), and (iii) human-readable explanation text that renders the decision

path in clinical terms. This trio enables end-to-end auditing from score to sentence, reducing the cognitive gap between model inference and bedside decision-making.

Provenance and accountability. Recommendation provenance (see Fig. 4.26) shows that the Hybrid model aggregates most of its evidence from the **CF channel** while systematically incorporating CBF signals when contextual similarity is informative. In the quality summary, 2003 recommendations are attributed to CF and 519 to CBF mechanisms (kNN, fallback, and exact/context matches), indicating a CF dominant fusion enriched by contextual reasoning. In practice, this means that a recommendation to initiate a first line antihypertensive can be traced both to population level behaviour (patients with similar response history) and to explicit contextual patterns (e.g., bp_category, risk_level). Such dual provenance supports clinical accountability: every item can be mapped to sources and mechanisms that are meaningful to medical reviewers.

Explanation quality and clinical language. The advice_text artefacts (see Fig. 4.29) emphasise guideline-consistent terminology, highlighting themes such as physical activity, sodium restriction, blood-pressure monitoring, and ACE inhibitors. Linguistic coherence matters: clinicians must recognise the semantics of the recommendation as much as its numeric confidence. The Hybrid outputs preserve medical realism by prioritising lifestyle measures when appropriate, escalating to pharmacologic advice for higher-risk profiles, and interleaving laboratory and follow-up recommendations (Fig. 4.27). The highest-scoring interventions (increase_exercise, diet_control, ace_inhibitor, amlodipine, beta_blocker, lifestyle_monitoring) mirror first-line strategies in contemporary hypertension guidelines.

Uncertainty, calibration, and thresholds. Interpretability is incomplete without an explicit treatment of uncertainty. The Hybrid score and confidence distributions (Figs. 4.24–4.25) provide this view. Hybrid scores are concentrated in a medium range around 0.40–0.45 with a secondary mass near 0.65 and an average of 0.457, indicating that the fusion layer differentiates between more and less plausible options without saturating scores near 1.0. The auxiliary confidence variable is intentionally conservative: the mean value is approximately 0.10, and no recommendation exceeds 0.2, with most cases falling into the “na” or “medium” buckets. Together, these distributions support threshold selection for different care settings. A conservative clinic can adopt a higher score threshold to favour precision, whereas a screening context may opt for lower thresholds to maximise recall. Because higher confidence values correspond to stronger CF–CBF agreement and higher similarity buckets, workflow policies can be tied directly to uncertainty (e.g., auto-approve high-confidence lifestyle advice versus flag-for-review low-confidence pharmacologic suggestions).

Coverage, cold-start, and graceful fallback. Contextual coverage analysis for CBF (Sec. 4.3.2) and LOCO experiments for CF and CBF (Sec. 4.3) demonstrate graceful degradation when specific context combinations are rare or temporarily missing. The system employs explicit fallbacks (content-based kNN, guideline-derived defaults) and records them in the reason and source fields. This behaviour is clinically and operationally desirable: recommendations remain explainable even under data sparsity, and provenance flags explicitly indicate when human oversight should be prioritised.

Fairness across clinical segments. Hit-rate stratification by BP categories for CF & CBF (Figs. 4.16–4.23) indicates stable performance across the clinical spectrum, with no disproportionate favouring of severe phenotypes. Hybrid model builds on these balanced baselines while improving overall precision and ranking quality (Sec. 4.4.1), suggesting that hybridisation enhances equity of performance rather than amplifying spurious biases. A full fairness audit across age, sex, and comorbidity strata is reserved for future work, but the current segment-wise results are consistent with unbiased behaviour.

Clinician-in-the-loop integration. Because each recommendation carries a calibrated score, an interpretable explanation, and auditable provenance, HCARS fits naturally into a clinician-in-the-loop workflow. Typical integration modes include: (i) triage mode, using high recall (primarily from CF) to surface likely-relevant lifestyle interventions; (ii) confirmation mode for pharmacologic suggestions that exceed agreed precision and confidence thresholds; and (iii) review mode, which highlights items associated with fallback provenance or low similarity buckets. These modes map naturally onto clinical practice, where certainty, risk, and reversibility guide action.

Limits and safeguards. Interpretability does not replace safeguards. The framework therefore incorporates: (i) routine calibration checks, (ii) drift monitoring for context distributions and label prevalence, and (iii) periodic provenance audits to verify ongoing alignment with updated guidelines (AHA 2023; ESC/ESH 2023). The design encourages these controls by construction: metrics and plots are automatically regenerated from CSV/JSON artefacts, and explanations are anchored to explicit context variables and source tags.

Key insight. Trust in HCARS emerges from consilience: consistent signals from CF & CBF, stable ranking performance, transparent provenance, and clinician-readable language. Hybrid model maximises this consilience, turning raw accuracy into actionable and auditable recommendations suitable for integration into routine hypertension care.

4.6 System Interface and User Interaction

The **HCARS for Hypertension** offers an integrated visual platform that enables users to explore, evaluate, and interpret model outputs through an interactive graphical interface built with Streamlit. The interface design balances technical depth with clinical interpretability, providing a bridge between data-science processes and human decision-making.

4.6.1 Main Framework Overview

The central entry point, defined in `main.py`, structures the navigation workflow. It creates the sidebar navigation panel, the main display canvas, and the page router that connects preprocessing, contextual analysis, and recommendation modules. The sidebar layout visible in Fig. (2).9 provides a persistent navigation anchor that divides the analytical workflow into three vertical sections:

1. **Data Preparation Modules:** Preprocess and Context allow users to clean, validate, and contextualise the dataset.
2. **Recommendation Engines:** CF, CBF, and Hybrid dashboards show model outputs, metrics, and justifications.
3. **Display Modes:** Original View and Comparison View control whether results are shown per-module or side-by-side.

Bilingual labels (Arabic–English) support local clinical users, and the dark theme enhances readability in low-light environments.

4.6.2 Home Page and Navigation

The Home page (Fig. (2).1) introduces the HCARS framework. It summarises the workflow `Preprocess → Context → CF/CBF/Hybrid` and enables direct module access from the sidebar.

4.6.3 Preprocess Module

As shown in Fig. (2).3, the Preprocess page displays cleaned data samples (`df_cleaned.csv`), supports “Before vs. After” inspection (Fig. (2).2), and allows CSV export. This step ensures data integrity before modelling.

4.6.4 Context Module

The Context dashboard (Fig. (2).4) visualises derived indicators such as `risk_score`, `bp_category`, and `risk_level`. Summary counters present dataset size, missingness, and target availability, converting numeric attributes into interpretable clinical categories.

4.6.5 CF Module

The CF interface (Fig. (2).5) presents similarity based recommendations. It includes overview KPIs (coverage, precision, recall, success rate), score distribution charts, and tabs for Recommendation Mix, Segment Performance, and Explainability. A cold-start tab appears when artefacts permit analysis.

4.6.6 CBF Module

The CBF dashboard (Fig. (2).6) highlights clinical feature similarity. It presents evaluation tables (e.g., `df_cbf_recommendations_eval.csv`), ranking metrics (Precision@k, Recall@k, MAP@k, nDCG@k), and subgroup visuals by `chol_category` and `risk_level`. Each recommendation includes an Arabic clinical explanation aligned with AHA/ESC guidelines.

4.6.7 Hybrid Module

The Hybrid recommender (Fig. (2).8) merges CF and CBF signals, showing combined outputs (`df_hybrid_recommendations.csv`) and evaluation summaries generated from `hybrid_eval_summary.json` and `hybrid_quality_report.json`. Tabs include Performance, Cross System Comparison, Case Explorer, and Explainability. Representative live metrics on the eval split are Precision@5 ≈ 0.72 and Recall@5 ≈ 0.91 , matching the quantitative results reported in Sec. 4.3.3.

4.6.8 Comparative Visualization

The dual panel comparison view (Fig. (2).7) allows visual analysis of module distributions such as CBF reasoning versus Hybrid score density revealing how hybrid fusion reshapes the ranking landscape. Coupled with the global comparison plots (Figs. 4.31 4.33), this view helps clinicians understand where Hybrid recommendations diverge from CF or CBF alone.

4.6.9 Smart Visualization Hub

The interactive Smart Visualization Center (Fig. (2).9) provides ad hoc charts (bar, line, box, pie), correlation heatmaps, and Arabic word clouds derived from explanation text,

enabling rapid exploratory data analysis within HCARS. All visuals are generated from versioned artefacts in the outputs directory, reinforcing reproducibility and auditability.

4.6.10 Design and Interaction Philosophy

The interface follows three principles: (i) **Transparency** all visuals originate from standardised CSV/JSON artefacts; (ii) **Usability** Streamlit widgets enable non programmers to explore model behaviour safely; and (iii) **Cognitive Accessibility** bilingual labels, consistent dashboards, and unified visuals foster trust in AI driven recommendations while reducing cognitive load.

Chapter Five: Discussion, conclusion, & recommendation

5.1 Interpretation of Results

From accuracy to behaviour. The quantitative results reported in Chapter 4 in particular the patterns in Precision@5 , Recall@5 , MAP@5 , and nDCG@5 (Tables 4.2, 4.4, and 4.5) highlight distinct behavioural profiles rather than a single “best” model. **CF** converts rich historical affinities into very high **Recall@5** and **MAP@5** on the eval split, reflecting its strength in exploiting patient patient similarity when interaction data are abundant. **CBF** trades some recall for robustness in sparse or shifting settings by grounding inference in explicit clinical features, which yields stable precision and inherently interpretable rationales. The **Hybrid** model sits at a deliberately different operating point: it achieves the highest **Precision@5** and **nDCG@5** across models, while maintaining recall and MAP values that are close to CF and clearly above CBF. In practical terms, CF behaves as a recall oriented engine, CBF as a context-aware baseline, and the Hybrid recommender as a precision focused yet still high coverage compromise.

Contextual learning and generalisation. The LOCO and family experiments (Sec. 4.3) act as stress-tests of contextual learning. Both CF and CBF exhibit graceful degradation rather than collapse when a context dimension is masked (LOCO) or when the evaluation cohort differs from the training distribution (family). This behaviour indicates that key contextual variables such as `bp_category`, `chol_category`, and `risk_level` encode clinically meaningful structure rather than spurious correlations. The Hybrid model’s robustness under the same conditions suggests a mixing policy in which CF dominates when reliable neighbours exist, while CBF compensates in regions where context ablation or cohort shift undermines neighbour reliability. This division of labour is exactly what is required from a safety-conscious clinical recommendation system.

Calibration and decision thresholds. The Hybrid score and confidence distributions (Figs. 4.24-4.25) show that recommendation scores are concentrated in a medium range with a secondary higher-scoring mode, while the separate confidence indicator remains intentionally conservative. Despite this restrained use of the score scale, the model attains an almost perfect nDCG@5 , meaning that the correct interventions are typically ranked among the very first positions. In clinical workflows, such behaviour supports policyable thresholds: conservative settings may apply higher score or confidence cut-offs to privilege precision for pharmacological advice, whereas preventive or triage contexts may lower thresholds to maximise recall for lifestyle recommendations. Because confidence

tends to increase when CF and CBF agree, thresholding also becomes a practical proxy for evidence concordance, not just for numerical score magnitude.

Interpretability and provenance as clinical capital. As documented in Chapter 4, each recommendation carries explicit source, reason, and explanation fields and a well-defined provenance channel (Fig. 4.26). This is particularly important in hypertension management, where the usual care pathway prioritises lifestyle optimisation before pharmacologic escalation. The distributions of advice types (Fig. 4.27) and the language of `advice_text` confirm that Hybrid outputs follow this hierarchy in practice, linking, for example, elevated BP and high risk-level to combined lifestyle and drug recommendations. Provenance therefore converts model confidence into auditable trust: clinicians can see not only what was recommended but also why and from which evidence channel.

Error modes and where they come from. Residual errors observed in calibration plots and precision recall curves for CF and CBF cluster in predictable regimes. They appear mainly (i) near decision boundaries where contextual features lie close to category thresholds (such as transitions between blood-pressure stages), (ii) in rare joint contexts identified by the `context_combinations_not_covered` analysis, and (iii) in cases where textual advice is unusually long or semantically dense, which can slightly perturb precision while maintaining high recall. The Hybrid model attenuates, but does not eliminate, these behaviours. Crucially, it surfaces them through provenance and confidence scores, marking such cases as candidates for human review rather than silently masking uncertainty.

Fairness signals and clinical equity. Hit-rate@5 stratification by blood-pressure categories for CF and CBF (Figs. 4.16 and 4.23) shows broadly balanced performance across severity levels, with no systematic concentration of accuracy in the most severe phenotypes. The Hybrid model builds on these balanced baselines while improving overall ranking quality, suggesting that hybridisation enhances equity of performance rather than amplifying bias. A comprehensive fairness audit across age, sex, and comorbidity strata remains future work, but current segment-wise evidence is consistent with clinically acceptable behaviour.

Implications for hypertension care. Taken together, these findings support a pragmatic clinician-in-the-loop deployment strategy:

- **Screening/triage:** operate at lower score thresholds to maximise recall of lifestyle interventions (e.g., diet control, exercise increase) while using provenance filters to down-weight fallback-only outputs.

- **Medication confirmation:** apply higher thresholds for pharmacologic suggestions (e.g., ACE inhibitors, beta-blockers), requiring CF, CBF concordance or a high-confidence CBF rationale anchored in guideline mappings.
- **Review queues:** route low-confidence or disagreement cases to manual review, highlighting provenance, key context fields, and calibration diagnostics to support rapid clinical adjudication.

In this configuration, confidence acts as a workload dial, aligning the level of automation with the acceptable clinical risk.

Operational reliability. Because all metrics, plots, and evaluation summaries are generated reproducibly from the exported CSV and JSON artefacts (Chapter 4), routine monitoring becomes straightforward. Recalibration, drift checks on contextual feature distributions, and periodic provenance audits can be integrated into standard MLOps pipelines. Such governance is essential for maintaining clinical reliability as data distributions and hypertension guidelines evolve.

5.2 Comparison with Previous Studies

Alignment with existing recommender-system research. The empirical findings of HCARS align closely with established trends in healthcare recommender literature discussed in Chapter 2. Previous studies on medical recommendation systems such as those by (Chen et al., 2025) and (X. Zhang et al., 2023) demonstrated that hybrid architectures outperform standalone CF or CBF by mitigating sparsity and cold-start problems. HCARS reproduces this pattern within a hypertension-specific domain, confirming that hybridisation remains a robust strategy even when applied to structured clinical data rather than consumer-oriented datasets.

Extension of context-aware modelling. While several prior works like (Zhu et al., 2024) and (Q. Zhang et al., 2015) introduced context-aware healthcare recommenders, most relied on limited contextual variables such as demographic or temporal features. HCARS extends this paradigm by operationalising clinically meaningful contexts BP category, cholesterol category, risk level, and age group and using them as first-class inputs for inference. This approach moves beyond generic contextual tagging toward medical-context embedding, allowing the system to reason in terms that correspond directly to clinical decision-making processes. Such integration represents a methodological advance that bridges algorithmic design with domain semantics.

Interpretability & provenance compared with explainable AI frameworks. Recent literature on transparency in RS and clinical AI (Kowald, 2024) emphasises transparency

as the linchpin of trustworthy systems. Unlike deep-learning-based black-box recommenders, HCARS embeds interpretability by design through the source, reason, and explanation fields, offering explicit provenance trails for every recommendation. This structured interpretability is conceptually related to explainable AI frameworks such as SHAP or LIME, but differs in being natively generated rather than post-hoc. Consequently, HCARS demonstrates how domain-specific hybridisation can deliver explainability without sacrificing accuracy.

Consistency with clinical decision-support findings. Clinical decision support research often reports the trade-off between algorithmic performance and clinical realism. For example, earlier systems like Clinical Decision Support Systems - Diabetes and Heart-CareRec achieved high predictive accuracy but limited contextual generalisation. HCARS addresses these limitations through its context engineering layer and hybrid inference policy, achieving superior recall while maintaining interpretability. This combination positions HCARS closer to real world usability, as it mirrors the human reasoning pattern of physicians: leveraging both population level analogies CF and patient-specific profiles CBF.

Novel contributions and divergence. In contrast to many earlier hybrid frameworks, which primarily combined feature vectors or similarity scores, HCARS employs a **clinically weighted fusion policy** defined in config.yaml that dynamically balances collaborative and content signals. Moreover, unlike most prior studies evaluated on synthetic or single institution datasets, HCARS was validated through rigorous cross scenario testing (eval, LOCO, family) and explicit calibration analysis. These additions extend the state of the art from proof-of-concept toward an auditable, deployable clinical recommender architecture.

Synthesis. HCARS both **confirms** key findings from the literature such as the superiority of hybridisation and the importance of interpretability and **extends** them by embedding clinical semantics directly into the recommendation process. Numerically, the Hybrid model achieves a Precision@5 of 0.7218, improving on CF (0.6636) by about +0.06 absolute and on CBF (0.4700) by about +0.25, while maintaining high Recall@5 (0.9097 vs. 0.9858 for CF and 0.7600 for CBF) and achieving an nDCG@5 of 0.9981, which is slightly higher than CF (0.9740) and substantially higher than CBF (0.8700). These magnitudes mirror the relative gains typically reported for hybrid healthcare recommenders and make the abstract advantages of hybridisation concrete in the hypertension setting. HCARS thus diverges from prior work by shifting the design objective from pure accuracy to trustworthy precision: recommendations that are not only correct but also understandable, auditable, and aligned with established hypertension management guidelines.

5.3 Practical Implications

From prototype to clinical workflow. The HCARS for Hypertension was designed not as a theoretical construct but as a clinically actionable framework that can be integrated into modern healthcare environments. Its modular, auditable, and explainable architecture enables seamless embedding within existing clinical information systems most notably, Electronic Health Records, decision-support dashboards, and patient-monitoring applications. Because every recommendation in HCARS is tied to explicit provenance (source, reason, and explanation) and contextual variables (bp_category, chol_category), integration with structured EHR fields becomes straightforward and traceable.

Integration with Electronic Health Records. In a hospital or primary care setting, HCARS could function as a recommendation engine embedded in the physician's EHR interface. Upon retrieving a patient's latest vitals, laboratory values, or comorbidity indicators, the system could automatically infer contextual categories and retrieve the top- k most relevant lifestyle or pharmacological interventions. Because these recommendations are grounded in interpretable logic rather than opaque latent factors, clinicians can validate them at a glance cross checking the reasoning path and confidence level before acceptance or override. Furthermore, periodic retraining and calibration logged within the EHR's audit trail, maintaining compliance with institutional quality and safety requirements.

Clinical dashboards and population level insights. Beyond single patient decision support, HCARS can power aggregate clinical dashboards. By aggregating contextualized recommendation statistics (e.g., distribution of advice types, mean confidence by risk_level), public health teams could monitor adherence patterns, identify at risk subpopulations, and evaluate the consistency of clinical practice across institutions. Because the pipeline outputs are standardized (CSV/JSON), integration with existing BI and visualization tools (such as Power BI or Tableau) is technically trivial. This makes HCARS suitable for deployment in both bedside and management layers of care delivery.

Patient facing and remote monitoring use cases. Interpretability built into HCARS supports patient centric deployment. When connected to remote monitoring platforms or mobile health applications, the system could deliver simplified, guideline aligned recommendations to patients translated into lay language yet derived from the same context-aware logic used by clinicians. By preserving calibration thresholds and provenance tags, HCARS maintains consistency between physician and patient interfaces, preventing informational drift and ensuring that patient-facing suggestions remain valid.

Interpretability, safety, and accountability. In clinical contexts, interpretability is not optional it is a prerequisite for safety. Each HCARS recommendation is accompanied by

a justification that can be audited retrospectively, ensuring that no output exists without a verifiable rationale. Confidence calibration (as discussed in Chapter 4) further allows the system to operate under variable automation levels: high-confidence recommendations may be accepted automatically in low-risk contexts, whereas ambiguous cases are escalated for human review. This tiered trust design directly aligns with medical safety principles of proportional oversight and informed decision making.

Governance and continuous validation. The architecture of HCARS inherently supports MLOps-style governance. All models, metrics, and feature engineering outputs are versioned, logged, and reproducible. Regular recalibration on new patient data, coupled with drift detection for contextual distributions (e.g., changes in `bp_category` frequencies), ensures that recommendations remain both accurate and equitable over time. Such traceability also facilitates external audit and certification, crucial for clinical AI systems operating under ISO/IEC and health-data protection standards.

Clinical value proposition. From a practical standpoint, the framework offers three complementary benefits:

1. **Personalization at scale** by combining collaborative and content-based reasoning, HCARS tailors recommendations to individual risk profiles without manual rule creation.
2. **Transparency and clinician trust** interpretability and provenance transform the model from a black box into a tool that can be explained, audited, and justified in clinical discussions.
3. **Operational efficiency** automation of contextual reasoning reduces cognitive load on healthcare professionals and enables rapid, guideline consistent decision support.

Translational impact. HCARS illustrates how a data driven hybrid recommender can evolve from an algorithmic prototype into a clinically relevant system. Its design aligns with current priorities in digital health: interoperability with EHRs, human centred AI, explainability, and adaptive safety controls. By bridging computational intelligence with clinical accountability, HCARS provides a blueprint for how AI-based recommendation engines can be responsibly deployed to enhance hypertension care, improve adherence to treatment guidelines, and support data-informed policymaking in public health.

5.4 Limitations of the Study

Scope and representativeness of the dataset. Although the KHD provided sufficient statistical depth for model development and evaluation, its scope remains limited to a sin-

gle institutional data source. The absence of longitudinal or multi center data restricts the system’s ability to generalize across healthcare settings with different demographics, diagnostic criteria, or measurement standards. Consequently, while internal validation demonstrates strong consistency, external validation across broader and more heterogeneous populations remains necessary before clinical deployment.

Data imbalance and sparsity. Despite careful preprocessing and contextual feature engineering, certain clinical categories particularly extreme cholesterol or BP combinations were underrepresented. This imbalance may have influenced the calibration of probability estimates and caused minor fluctuations in performance under the **family** (cross-group) evaluation scenario. Moreover, sparsity in specific patient intervention pairs introduces inherent uncertainty typical of RS, where rare events cannot be fully captured without larger datasets.

Cold-start sensitivity. Although the Hybrid model mitigates cold-start problems through context-aware CBF components and fallback logic, complete elimination of the issue remains impossible in the absence of prior patient records. For new users or unseen combinations of contextual features, recommendations depend primarily on population level similarities rather than individualized patterns. While acceptable in early stage screening, this behaviour limits the granularity of personalization compared with scenarios where sufficient longitudinal data are available.

Synthetic validation and lack of real clinical deployment. All experiments were conducted on de identified structured data within a controlled research environment. The recommendations were not yet prospectively validated through real clinician feedback or live patient outcomes. Therefore, while the Hybrid configuration of HCARS achieved robust quantitative performance (e.g., Precision@5 $\approx$ 0.72, Recall@5 $\approx$ 0.91, MAP@5 $\approx$ 0.72, nDCG@5 $\approx$ 1.00 on the eval split), these metrics still reflect algorithmic efficacy rather than clinical impact. A future clinical trial or simulation study is needed to evaluate the model’s practical usefulness, user acceptance, and integration into real decision workflows.

Feature and label interpretability boundaries. Contextual features (bp_category, chol_category, risk_level) are derived from discretized numerical data according to established thresholds. While these thresholds ensure interpretability, they may oversimplify continuous physiological variability, potentially reducing sensitivity to subtle risk gradients. Similarly, recommendation labels such as diet_control or increase_exercise represent guideline-driven categories rather than detailed clinical prescriptions, limiting direct applicability without professional interpretation.

Algorithmic and computational limitations. HCARS relies on classical hybridization rather than deep representation learning. Although this choice enhances transparency and reproducibility, it may underutilize latent feature interactions detectable by neural architectures. In addition, while training and inference times are clinically acceptable, scaling the model to national or multi-hospital datasets would require further optimization, parallelization, and infrastructure adaptation.

Evaluation framework constraints. The evaluation metrics (Precision@5, Recall@5, MAP@5, nDCG@5) capture retrieval and ranking quality but do not fully account for clinical appropriateness or patient outcomes. Even perfectly ranked recommendations may differ in therapeutic relevance depending on comorbidities or drug contraindications not represented in the dataset. Hence, future iterations of the framework should incorporate domain-specific metrics, such as “clinically valid recommendation rate” or “guideline concordance score,” to better reflect medical decision-support value.

The present study is constrained by (i) a single-source dataset, (ii) residual data imbalance and sparsity, (iii) unresolved cold-start sensitivity, (iv) absence of real clinical validation, and (v) methodological simplifications inherent to interpretable hybridization. Nevertheless, these constraints are transparent and quantifiable, and they do not undermine the system’s conceptual contribution: demonstrating that a transparent, context-aware hybrid recommender can achieve both high accuracy and clinical interpretability within the boundaries of available data.

5.5 Recommendations for Future Research

Toward multi-institutional and longitudinal data integration. Future work should extend the dataset beyond the single-source KHD used in this study. Incorporating longitudinal patient trajectories and multi-center data would enable temporal modelling of disease progression and improve generalisability across populations. This would also support modelling dynamic treatment response as blood pressure, cholesterol, and risk levels evolve over time.

Adoption of deep or reinforcement-based hybrid architectures. While the current hybrid design achieves high accuracy with interpretable fusion, emerging methods such as graph neural networks, attention-based hybrid models, and reinforcement learning could further enhance personalization. A deep hybridization layer could capture complex nonlinear relationships among patient features, treatments, and outcomes, while reinforcement-based strategies could continuously refine recommendations based on feedback loops from

clinical practice. These developments would preserve the interpretability ethos of HCARS while extending its adaptive intelligence.

Explainable AI and interpretability enhancement. Although HCARS already embeds intrinsic transparency through its reason, source, and explanation fields, future iterations could integrate formal explainable AI (XAI) frameworks such as SHAP, LIME, or counterfactual reasoning. This would quantify the contribution of each contextual or demographic feature to the final recommendation, providing both clinicians and patients with fine-grained interpretive confidence. Combining statistical attribution with human-readable explanations would strengthen accountability and regulatory readiness for clinical adoption.

Cross-disease generalization. The conceptual structure of HCARS context-aware hybrid recommendation under transparent inference can be generalized beyond hypertension. Chronic diseases with layered risk factors, such as diabetes, heart failure, or chronic kidney disease, could benefit from similar architectures. Adapting the contextual feature space (e.g., HbA1c, eGFR, BMI categories) would validate the flexibility and transferability of the framework, transforming HCARS from a disease-specific prototype into a universal clinical recommender blueprint.

Operational deployment and human–AI collaboration. Future work should move from offline evaluation to pilot deployment. Integrating HCARS into EHR dashboards or mobile clinical applications could support human–AI collaboration, treating recommendations as assistive insights rather than prescriptive outputs. Controlled clinical trials and usability studies are needed to assess impact on decision time, guideline adherence, and patient outcomes, ensuring technical gains translate into clinical value.

Future research directions. The next phase should prioritise: (i) data diversification and temporal modelling; (ii) deep and reinforcement learning; (iii) structured explainability for regulatory transparency; (iv) cross-disease extension; and (v) real-world validation under human supervision. These directions build on the present study and address its methodological limitations, guiding HCARS toward scalable, ethical, and clinically integrated decision support.

5.6 Concluding Remarks

Synthesis of methodological and empirical contributions. This study presented the design, implementation, and validation of the **HCARS** a newly integrated clinical recommender framework that unites the algorithmic robustness of CF with the interpretability of CBF. Through meticulous data preprocessing, contextual feature engineering, and

weighted hybridization, HCARS demonstrated that accurate and transparent decision support can coexist within a single architecture. The results in Chapter 4 showed consistent performance improvements across all key metrics, confirming the viability of context-driven hybridization for personalized medical inference.

Bridging algorithmic precision with clinical meaning. Beyond raw accuracy, the system achieved a rare balance between computational performance and clinical realism. By encoding contextual variables such as blood pressure, cholesterol, and overall risk level, HCARS transformed abstract feature vectors into medically interpretable concepts. This shift from “data correlation” to “clinical reasoning” represents a fundamental advancement in the design of trustworthy medical AI systems, ensuring that every recommendation can be both quantified and explained.

Translational and ethical impact. HCARS emphasises interpretability and auditability to address key deployment challenges in healthcare, including safety, accountability, and clinician trust. By preserving traceable provenance and calibrated confidence, it supports evidence-based practice and helps mitigate automation bias. The design principles—context-awareness, hybrid reasoning, and explainability—also provide an ethical foundation for future medical recommender systems.

Position within the broader research landscape. Relative to prior work, HCARS reinforces the value of hybrid architectures while extending them in a domain-specific and clinically interpretable direction. It shows that data-driven personalisation can be achieved without sacrificing transparency, and that interpretable models can match or exceed more opaque alternatives. This positions HCARS as both a technical contribution and a methodological reference for responsible AI in precision medicine.

Final reflection. Ultimately, HCARS exemplifies how intelligent systems can augment rather than replace clinical expertise. By transforming heterogeneous patient data into actionable, guideline-consistent recommendations, it bridges the gap between algorithmic inference and clinical judgment. The research thus contributes to the growing movement toward **human-centered, explainable, and context-aware AI in healthcare**, setting the foundation for a new generation of transparent, data-driven clinical decision-support tools.

References

- Abualigah, L., Alomari, S. A., Almomani, M. H., Abu Zitar, R., Saleem, K., Migdady, H., Snàšel, V., Smerat, A., & Ezugwu, A. E. (2025). Artificial intelligence-driven translational medicine: A machine learning framework for predicting disease outcomes and optimizing patient-centric care. *Journal of Translational Medicine*, 23, 302. <https://doi.org/10.1186/s12967-025-06308-6>
- Abusair, M., Sharaf, M., Bozeya, M., & Beiruti, A. (2020). Context-aware recommender system based on content filtering. *Journal of Mobile Computing*, 18, 55–65.
- Adeleke, O., & et al. (2024). Machine learning evaluation of a hypertension screening programme in a workplace setting: Longitudinal study. *Scientific Reports*. <https://www.nature.com/articles/s41598-024-74360-1>
- Adomavicius, G., Mobasher, B., Ricci, F., & Tuzhilin, A. (2011). Context-aware recommender systems. *AI Magazine*, 32(3), 67–80. <https://doi.org/10.1609/aimag.v32i3.2364>
- Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 217–256). Springer.
- Aggarwal, C. C. (2016a). *Recommender systems*. Springer.
- Aggarwal, C. C. (2016b). *Recommender systems: The textbook*. Springer. <https://doi.org/10.1007/978-3-319-29659-3>
- Ahmad, S., Ahmad, J., Mehruz, S., & Ansari, M. A. (2025). Hybrid recommender system for personalized healthcare: Integration of content-based and collaborative filtering approaches. *International Journal of Systems Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-025-02965-4>
- Ali, W., Kumar, J., Mawuli, C. B., She, L., & Shao, J. (2023). Dynamic context management in context-aware recommender systems. *Computers & Electrical Engineering*, 107, 108622. <https://doi.org/10.1016/j.compeleceng.2023.108622>

- Ananthakrishnan, A., Milne-Ives, M., Shankar, R., & Meinert, E. (2024). Evaluation of health recommender systems: A scoping review protocol. *BMJ Open*, 14(10), e083359. <https://doi.org/10.1136/bmjopen-2023-083359>
- Anwar, T., Uma, V., Hussain, M. I., & Pantula, M. (2022). Collaborative filtering and knn based recommendation to overcome cold start and sparsity issues: A comparative analysis [Published online: 18 March 2022]. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-021-11883-z>
- Asgari, M., Vincelli, D., McDaniel, J., Shanafelt, T., Blocker, M., Sharp, M., Belden, J. L., & Patterson, E. S. (2024). Cognitive load and electronic health record (ehr) usability: A narrative review. *BMC Medical Informatics and Decision Making*. <https://doi.org/10.1186/s12911-024-02617-8>
- Baltrunas, L., Ludwig, B., & Ricci, F. (2011). Context-aware places of interest recommendations for mobile users. *ACM Conference on Recommender Systems*.
- Barbaric, A., Christofferson, K., & Benseler, S. M. (2025). Health recommender systems to facilitate collaborative decision-making in chronic disease management: A scoping review. *Digital Health*, 11, 1–16. <https://doi.org/10.1177/20552076241309386>
- Barbarić, S., Stjepanović, P., Jurišić, M., Tomic, I., Stojanović, N., Kranjčević, T., Glasnović, A., & Skroza, M. (2025). Health recommender systems: A scoping review. *Digital Health*. <https://journals.sagepub.com/doi/10.1177/20552076241277337>
- Bettache, D., Dennouni, N., & Harbouche, A. (2025). Improving poi recommendation through collaborative filtering by incorporating the geographical factor into the similarity calculation. *Engineering, Technology & Applied Science Research*, 15(3), 23629–23634. <https://doi.org/10.48084/etasr.10660>
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370. <https://doi.org/10.1023/A:1021240730564>
- Cavero-Redondo, I., & et al. (2024). Machine learning–based multicohort study (rigiprev): Personalized recommendations for antihypertensive treatment aimed at reducing

arterial stiffness. *Journal of Medical Internet Research*. <https://pubmed.ncbi.nlm.nih.gov/39585738/>

Cerón-Rios, G., López, D. M., & Blobel, B. (2017). Architecture and user-context models of cocare: A context-aware mobile recommender system for health promotion [Published January 2017]. *Studies in Health Technology and Informatics*, 237, 140–147. <https://europepmc.org/article/MED/28479557>

Chen, Y., Ho, K. Y., Zong, X., Weng, Y., Yuan, C., & Yorke, J. (2025). The digital intelligent precise nursing framework: Theory development in health recommender system. *BMC Nursing*, 24, 1191. <https://doi.org/10.1186/s12912-025-03830-2>

Chuks, P. (2022). *Health dataset [data set]* [Accessed 1 September 2024]. <https://www.kaggle.com/datasets/prosperchuks/health-dataset/data>

Clifford, N., & et al. (2025). Digital precision hypertension: Trends and gaps. *JMIR*. <https://www.jmir.org/2025/1/e59841/>

De Croon, R., Van Houdt, L., Htun, N. N., Štiglic, G., Vanden Abeele, V., & Verbert, K. (2021). Health recommender systems: Systematic review. *Journal of Medical Internet Research*, 23(6), e18035. <https://doi.org/10.2196/18035>

Errakha, K., Samih, A., Marzouk, A., & Krari, A. (2024). Surveying the black box: An overview of context-aware recommender systems. *Journal of Theoretical and Applied Information Technology*, 102(21), 7624–7637. <https://www.jatit.org/volumes/Vol102No21/5Vol102No21.pdf>

Fkih, F., & Rhouma, D. (2023). Literature review on recommender systems: Techniques, trends and challenges. In A. Abraham, T.-P. Hong, K. Kotecha, K. Ma, P. M. Mishra, & N. Gandhi (Eds.), *Hybrid intelligent systems* (pp. 493–500, Vol. 647). Springer. https://doi.org/10.1007/978-3-031-27409-1_44

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd). Morgan Kaufmann.

Han, Q., Ji, M., de Rituerto de Troya, I. M., Gaur, M., & Zejnilovic, L. (2019). A hybrid recommender system for patient-doctor matchmaking in primary care [Submitted on 9 Aug 2018 (v1), last revised 18 Mar 2019 (v2)]. *arXiv preprint arXiv:1808.03265*. <https://arxiv.org/abs/1808.03265>

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Herlocker, J. L., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4), 287–310.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004a). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53. <https://doi.org/10.1145/963770.963772>
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004b). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53. <https://doi.org/10.1145/963770.963772>
- Hors-Fraile, S., Candel, M. J. J. M., Schneider, F., Malwade, S., Nunez-Benjumea, F. J., Syed-Abdul, S., Fernandez-Luque, L., & de Vries, H. (2022). Applying collective intelligence in health recommender systems for smoking cessation: A comparison trial. *Journal of Medical Internet Research*, 24(6), e30314. <https://doi.org/10.2196/30314>
- International Data Corporation. (2022). *Worldwide global datasphere forecast, 2022–2026* (US49018922) (Global forecast of digital data volume for 2018–2026). International Data Corporation (IDC). <https://www.marketresearch.com/IDC-v2477/Worldwide-IDC-Global-DataSphere-Forecast-31469143/>
- Jadon, A., & Patil, A. (2024). A comprehensive survey of evaluation techniques for recommendation systems. *arXiv preprint arXiv:2312.16015*. <https://arxiv.org/abs/2312.16015>
- Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8), 651–666. <https://doi.org/10.1016/j.patrec.2009.09.011>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2023). *An introduction to statistical learning* (2nd ed.). Springer.
- Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4), 422–446. <https://doi.org/10.1145/582415.582418>

- Jemal, I., Naoussi Sijou, W. A., & Chikhaoui, B. (2024). Multi-modal recommender system for predicting project manager performance within a competency-based framework. *Frontiers in Big Data*, 7, 1295009. <https://doi.org/10.3389/fdata.2024.1295009>
- Jeong, S.-Y., & Kim, Y.-K. (2022). Deep learning-based context-aware recommender system considering contextual features. *Applied Sciences*, 12, 1045.
- Jeong, S.-Y., & Kim, Y.-K. (2023). Deep learning-based context-aware recommender system considering change in preference. *Electronics*, 12(10), 2337. <https://doi.org/10.3390/electronics12102337>
- Jha, R. K., Bag, S., Koley, D., Bojja, G. R., & Barman, S. (2023). An appropriate and cost-effective hospital recommender system for a patient of rural area using deep reinforcement learning. *Intelligent Systems with Applications*, 18, 200218. <https://doi.org/10.1016/j.iswa.2023.200218>
- Kamran, M., & Javed, A. (2015). A survey of recommender systems and their application in healthcare. *Technical Journal, University of Engineering and Technology (UET) Taxila*, 20(4), 111–119. <https://tj.uettaxila.edu.pk/older-issues/2015/No4/15.A%20Survey%20of%20Recommender%20Systems%20and%20Their%20Application%20in%20Healthcare.pdf>
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. <https://doi.org/10.1109/MC.2009.263>
- Kowald, D. (2024). Transparency, privacy, and fairness in recommender systems [arXiv:2406.11323]. <https://doi.org/10.48550/arXiv.2406.11323>
- Lajevardi, S. A., Kargari, M., Daneshpour, M. S., & Akbarzadeh, M. (2022). A cnn deep learning model to improve snp-based hypertension risk prediction accuracy [Preprint available at ResearchGate, DOI 10.21203/rs.3.rs-2285831/v1]. *Preprint*. https://www.researchgate.net/publication/365664075_A_CNN_deep_learning_model_to_improve_SNP-based_hypertension_risk_prediction_accuracy
- Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive datasets* (3rd). Cambridge University Press. <http://www.mmms.org>

- Lipski, D., Lewandowska, K., Żurek, S., Wiśniewicz, M., Piróg, P., Tykarski, A., & Uruski, P. (2025). Telemedical care system for the treatment of arterial hypertension using recommendation algorithms to support diagnostic and therapeutic decisions—sot-art-ht trial protocol. *Trials*, 26, 330. <https://doi.org/10.1186/s13063-025-09044-w>
- Lu, Y., Arowojolu, O., Qiu, X., Liu, Y., Curry, L. A., & Krumholz, H. M. (2024). Barriers to optimal clinician guideline adherence in management of markedly elevated blood pressure: A qualitative study. *JAMA Network Open*, 7(8), e2426135. <https://doi.org/10.1001/jamanetworkopen.2024.26135>
- Luo, J., Wu, M., Gopukumar, D., & Zhao, Y. (2021). Deep learning-based hybrid recommender systems for personalized healthcare services. *Journal of Healthcare Engineering*, 2021, Article ID 9876543.
- Maleki, N., Zeinali, Y., & Akhavan Niaki, S. T. (2021). A k-nn method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems with Applications*, 164, 113981. <https://doi.org/10.1016/j.eswa.2020.113981>
- Mallen, L., & Bridge, D. (2024). K-most-influential neighbours: Noise-resistant knn. In *Artificial intelligence and cognitive science*. Springer. https://doi.org/10.1007/978-3-031-70650-0_3
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Mroz, T., & et al. (2024). Predicting hypertension control using machine learning. *NPJ Digital Medicine*. <https://pubmed.ncbi.nlm.nih.gov/38507433/>
- Müllner, P., Lex, E., Schedl, M., & Kowald, D. (2023). Differential privacy in collaborative filtering recommender systems: A review. *Frontiers in Big Data*, 6, 1249997. <https://doi.org/10.3389/fdata.2023.1249997>
- Munson, J., Cuezze, T., Nesar, S., & Zosso, D. (2025). A review of large language models and the recommendation task. *Discover Artificial Intelligence*, 5, 203. <https://doi.org/10.1007/s44163-025-00334-5>
- Nag, N., Pandey, V., & Jain, R. C. (2018). Endogenous and exogenous multi-modal layers in context aware recommendation systems for health [Submitted on 7 Aug 2018]. *arXiv preprint arXiv:1808.06468*. <https://arxiv.org/pdf/1808.06468>

- Nawata, K. (2023). Risk factors for heart, cerebrovascular, and kidney diseases: Evaluation of potential side effects of medications to control hypertension, hyperglycemia, and hypercholesterolemia. *Frontiers in Cardiovascular Medicine*, 10, 1103250. <https://doi.org/10.3389/fcvm.2023.1103250>
- Nguyen, H. M., & et al. (2024). Predictive models for sustained, uncontrolled hypertension: Development and validation. *JMIR Medical Informatics*. <https://medinform.jmir.org/2024/1/e58732/>
- omitted by journal (JMIR Cardio), A. (2024). Ai-based autonomous digital health lifestyle coaching for blood pressure control: Randomized controlled trial. *JMIR Cardio*. <https://cardio.jmir.org/2024/1/e51916>
- Organization, W. H. (2025). *Hypertension [fact sheet]* [Updated September 25, 2025. Accessed November 1, 2025]. <https://www.who.int/news-room/fact-sheets/detail/hypertension>
- Peng, X., & et al. (2025). The link between adherence to antihypertensive medications and mortality: A meta-analysis of cohort studies. *BMC Cardiovascular Disorders*, 25, 38. <https://bmccardiovascdisord.biomedcentral.com/articles/10.1186/s12872-025-04538-6>
- Phatcharathada, B., & Srisuradetchai, P. (2025). Randomized feature and bootstrapped naive bayes classification. *Applied System Innovation*, 8(4), 94. <https://doi.org/10.3390/asi8040094>
- Prahmana, D. D., Rahmandita, D. P., & Yanto, E. (2025). Application of the k-nearest neighbor (knn) algorithm in machine learning to predict the selection of undergraduate study programs based on new kip lecture students. *Journal of Artificial Intelligence and Engineering Applications*, 4(2), 1523–1531. <https://doi.org/10.59934/jaiea.v4i2.941>
- Ramkumar, S., Khan, I., Chan, S. C. C., Jerjes, W., & Majeed, A. (2025). From clicks to care: Enhancing clinical decision making through structured electronic health records navigation training. *Journal of Clinical Medicine*, 14(14), 4813. <https://doi.org/10.3390/jcm14144813>
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings of the 1994*

ACM Conference on Computer Supported Cooperative Work, 175–186. <https://doi.org/10.1145/192844.192905>

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). *Recommender systems handbook*. Springer.

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2022). *Recommender systems handbook* (3rd). Springer. <https://doi.org/10.1007/978-1-0716-2197-4>

Rizki, M. A., & Rianto, Y. (2024). Hybrid recommender system using switching strategy between collaborative filtering and content-based filtering. *Jurnal Teknologi Informatika dan Ilmu Komputer*, 11(1), 123–134. <https://journal.maranatha.edu/index.php/jutisi/article/view/6220>

Salton, G., & McGill, M. J. (1988). *Introduction to modern information retrieval*. McGraw-Hill.

Sgardelis, K., Margaris, D., Spiliotopoulos, D., & Vassilakis, C. (2025). An evaluation review of user similarity metrics in sparse collaborative filtering datasets. *International Journal of Data Science and Analytics*, 20, 6665–6693. <https://doi.org/10.1007/s41060-025-00846-4>

Skalidis, I., & et al. (2025). Artificial intelligence and advanced digital health for hypertension management. *Digital Health Reviews*. <https://pmc.ncbi.nlm.nih.gov/articles/PMC12471829/>

Stratigi, M., Kondylakis, H., & Stefanidis, K. (2020). Multidimensional group recommendations in the health domain. *Algorithms*, 13(3), 54. <https://doi.org/10.3390/a13030054>

Sun, Y., Zhou, J., Ji, M., Pei, L., & Wang, Z. (2023). Development and evaluation of health recommender systems: Systematic scoping review and evidence mapping. *Journal of Medical Internet Research*, 25, e38184. <https://doi.org/10.2196/38184>

Torres-Ruiz, M., Quintero, R., Guzman, G., & Chui, K. T. (2023). Healthcare recommender system based on medical specialties, patient profiles, and geospatial information. *Sustainability*, 15(1), 499. <https://doi.org/10.3390/su15010499>

- Union, I. T. (2023). Measuring digital development: Facts and figures 2023. <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>
- Valdeira, F., Racković, S., Danalachi, V., Han, Q., & Soares, C. (2023). Extreme multi-label classification for specialist doctor recommendation with implicit feedback and limited patient metadata. *arXiv preprint arXiv:2308.11129*. <https://www.semanticscholar.org/paper/Extreme-Multilabel-Classification-for-Specialist-Valdeira-Rackovi%27c/6105ad0f173f9286d8b1637ee3544490fa5a43be>
- Vebiyatama, A., & Ernawati, M. (2024). Advances in machine learning and deep learning towards medical data analysis. *Journal of Medical Informatics and Technology*, 2(1), 21–26. <https://medinftech.org/index.php/medinftech/article/view/32>
- Verma, R., Chopra, S., Yadav, H., & Prakash, A. V. (2025). Supervised vs. unsupervised learning: A comparative study in modern ai system. *Communications on Applied Nonlinear Analysis*, 32(9s), 361–375. <https://doi.org/10.58997/cana.v32i9s.4225>
- Voorhees, E. M. (2001). The philosophy of information retrieval evaluation. *Proceedings of the 2nd Workshop of the Cross-Language Evaluation Forum*.
- Walek, B., & Fajmon, P. (2023). A hybrid recommender system for an online store using a fuzzy expert system. *Expert Systems with Applications*, 212, 118565. <https://doi.org/10.1016/j.eswa.2022.118565>
- Wu, L., Pang, Y. W., & Cai, W. (2024). Pseudo-online measurement of retrieval recall for job recommendations - a case study at indeed [CC BY 4.0]. *RecSys in HR'24: The 4th Workshop on Recommender Systems for Human Resources (in conjunction with RecSys 2024)*. https://recsyshr.aau.dk/wp-content/uploads/2024/10/RecSysHR2024-paper_6.pdf
- Yang, J. (2025). Improved support vector machine and deep neural network-based rural education platform for data analysis and network education enhancement. *Proceedings of SPIE*, 13629, 136292A. <https://doi.org/10.1117/12.3068002>
- Yashvanth, R., et al. (2023). Diabetes, hypertension and stroke prediction using computational algorithms. *2023 World Conference on Communication & Computing*. <https://doi.org/10.1109/WCONF58270.2023.10235026>

- Yi, J., & et al. (2024). Development of a machine learning-based model for predicting blood pressure response to antihypertensive medications. *Nutrition, Metabolism and Cardiovascular Diseases*. <https://pubmed.ncbi.nlm.nih.gov/38555240/>
- Zhang, Q., Zhang, G., Lu, J., & Wu, D. (2015). A framework of hybrid recommender system for personalized clinical prescription. *2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. <https://doi.org/10.1109/ISKE.2015.98>
- Zhang, X., Yi, M., Sun, Y., Han, S., Zhang, W., & Wang, Z. (2023). The application of health recommender systems based on knowledge graph: A scoping review. *Interdisciplinary Nursing Research*, 2(1), 37–45. <https://nursing.bjmu.edu.cn/szdw/jsmd/4ee8f28854754550ade99322792885d9.htm>
- Zhang, Y., Sun, J., Jiang, H., Carroll, R. J., Peissig, P. L., Onitilo, A. A., & Yu, G. (2018). Healthcare recommendation systems: Principles, methodologies, and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(3), e1238.
- Zheng, Y., Scott, J., Luk, A., & Funk, K. (2020). Predictive modeling for blood pressure response to antihypertensive drugs: Machine learning approaches. *Journal of Personalized Medicine*, 10(4), 150.
- Zhu, F., Cui, L., Xu, Y., Qu, Z., & Shen, Z. (2024). A survey of personalized medicine recommendation. *International Journal of Crowd Science*, 8(2), 77–82. <https://doi.org/10.26599/IJCS.2023.9100013>
- Zitouni, H., Meshoul, S., & Mezioud, C. (2022). New contextual collaborative filtering system with application to personalized healthy nutrition education. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1124–1137. <https://doi.org/10.1016/j.jksuci.2020.04.012>
- Zuin, M., & et al. (2025). Advances in hypertension management. *MDPI – Review Article*. <https://pmc.ncbi.nlm.nih.gov/articles/PMC12027560/>

Appendices

Appendix (1)

Comprehensive Summary of Recommender Systems

Table (1).1: Comprehensive Summary of Recommender Systems

Ref.	Parameter	Methodology	Findings	Gaps
Hors-Fraile et al., 2022	• Smoking cessa- tion	• Collective intelligence • Collaborative filtering • Expert insights	• Collective intelligence improves engagement and smoking cessation rates	• Lacks real-time adap- tation • Long-term impact studies needed
				Continued on next page

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Torres-Ruiz et al., 2023	• Patient profiles • Medical specialties • Geospatial data	• Geospatial data • Semantic ontologies • Health attention factor	• Optimized healthcare access with geospatial data • Good accuracy in recommending health-care facilities	• Reliance on third-party data • Integration with EMRs missing
Yashvanth et al., 2023	• Prediction of diabetes, hypertension, stroke	• Logistic regression • SVM • Gradient boosting • AdaBoost	• Gradient boosting achieved 75.5% accuracy • Good feature selection with PCA	• Data quality issues • Missing values • Lack of real-time data
Zhu et al., 2024	• Personalized medicine • Drug–drug interactions	• Deep learning • Reinforcement learning • Collaborative filtering	• Enhanced personalized treatments via EHR and medical history integration	• Lacks patient feedback integration • Real-time adaptability needed
				Continued on next page

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Kamran and Javed, 2015	• Recommender systems in health-care	• Literature review • Collaborative filtering • Hybrid models	• Hybrid systems improve recommendation accuracy • Address cold-start problems	• Lack of real-time adaptability • Complex handling of large-scale medical data
De Croon et al., 2021	• General health • Nutrition • Lifestyle	• PRISMA guidelines • Hybrid models (collaborative, content-based)	• Hybrid systems (63%) improve personalization in healthcare	• Lack of real-world user trials • Transparency issues in algorithm generation
Lajevardi et al., 2022	• Hypertension risk prediction	• CNN-based deep learning • SNP data • Polygenic Risk Scores (PRS)	• CNN outperformed PRS with AUC of 0.877 • Better genetic data handling	• Excludes clinical or lifestyle data • Lacks external validation
Zitouni et al., 2022	• Nutrition education	• Two-dimensional contextual framework (internal and external) • Collaborative filtering	• Better tailored nutrition recommendations based on personal and environmental factors	• Lacks genetic or medical history integration
Continued on next page				

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Ali et al., 2023	Dynamic context attributes (time, location, user behavior)	- Minkowski distance - Context management - Collaborative filtering	Improved prediction accuracy with dynamic context attributes	Lacks real-time health data integration
Jeong and Kim, 2022	Contextual data (time, location, mood)	Hybrid model (autoencoder + neural networks)	Improved recommendation accuracy in sparse data environments	- Slower response times - Small dataset - Lacks real-time adaptability
Adomavicius and Tuzhilin, 2011	- Contextual modeling - Recommendation systems	- Contextual pre-filtering - Multidimensional algorithms	Enhanced recommendation accuracy with contextual information	- Lacks real-time dynamic adaptability - Limited medical data focus
Valdeira et al., 2023	Specialist doctor recommendation	- Extreme multilabel classification - Implicit feedback	Accurate doctor recommendations in sparse data environments	- Requires implicit feedback - Limited availability in healthcare settings
				Continued on next page

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Maleki et al., 2021	• Cancer prognosis	• k-NN hybrid model • Genetic algorithm for feature selection	• Significant increase in classification accuracy • Feature selection optimization	• Needs external validation with larger datasets
Jha et al., 2023	• Hospital recommendation in rural areas	• Deep reinforcement learning • Cost-effective recommendation	• Cost-effective hospital recommendations based on proximity, effectiveness, and cost	• Lacks real-time integration with health data for chronic diseases
Stratigi et al., 2020	• Group preferences • Healthcare recommendations	• Multidimensional group recommendations • Medical data integration	• Better health outcomes by catering to group preferences	• Lacks real-time health monitoring integration
Burke, 2002	• Hybrid systems • Collaborative filtering	• Weighted, mixed, and cascade hybrid strategies	• Effective hybrid models for overcoming cold-start and sparsity problems	• Computational complexity • Requires large-scale data
				Continued on next page

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Anwar et al., 2022	• k-NN • Collaborative filtering	• k-NN algorithm • Collaborative filtering	• Improved accuracy in healthcare recommendations by addressing cold-start and data sparsity	• Lacks real-time health data integration
Q. Zhang et al., 2015	• Healthcare recommendations • Semantic data	• Machine learning • Collaborative filtering • Semantic data	• Improved healthcare recommendations with better contextual understanding	• Needs more real-time health data adaptation
Q. Han et al., 2019	• Patient demographics (age, gender, location) • Doctor data (specialization, demographics) • Consultation history	• Hybrid recommender system (collaborative filtering, content-based) • Trust modeling using temporal dynamics	• Hybrid model with trust improved accuracy (80% HR@10) • Personalized recommendations enhanced patient-doctor match-making	• Requires external validation • Lacks real-time integration • Relies on inferred preferences
Continued on next page				

Table (1).1 – continued from previous page

Ref.	Parameter	Methodology	Findings	Gaps
Nag et al., 2018	• Physiological data (heart rate, glucose levels) • Environmental data (weather, pollution)	• Multi-modal sensor integration • Context-aware layers (endogenous and exogenous) • Nutrition recommendation system for sodium control	• Personalized meal recommendations • Improved health management for heart failure patients • Effective integration of biological and environmental data	• Limited real-world testing • Synthetic user scenarios • Scalability challenges with large populations
Cerón-Rios et al., 2017	• User health data (age, conditions, activity level, preferences) • Contextual data (time, location, lifestyle)	• Context-aware recommendation engine • Rule-based filtering • Utility function modeling • Personalized recommendations based on user-context profiles	• CoCare system improved adherence to health guidelines • Users found personalized health recommendations effective for habit formation	• Limited use of machine learning techniques (mostly rule-based) • Scalability challenges for larger datasets • No integration with dynamic health sensor data

Appendix (2)

HCARS Interface Screens

Full-size screenshots of the Streamlit HCARS interface.

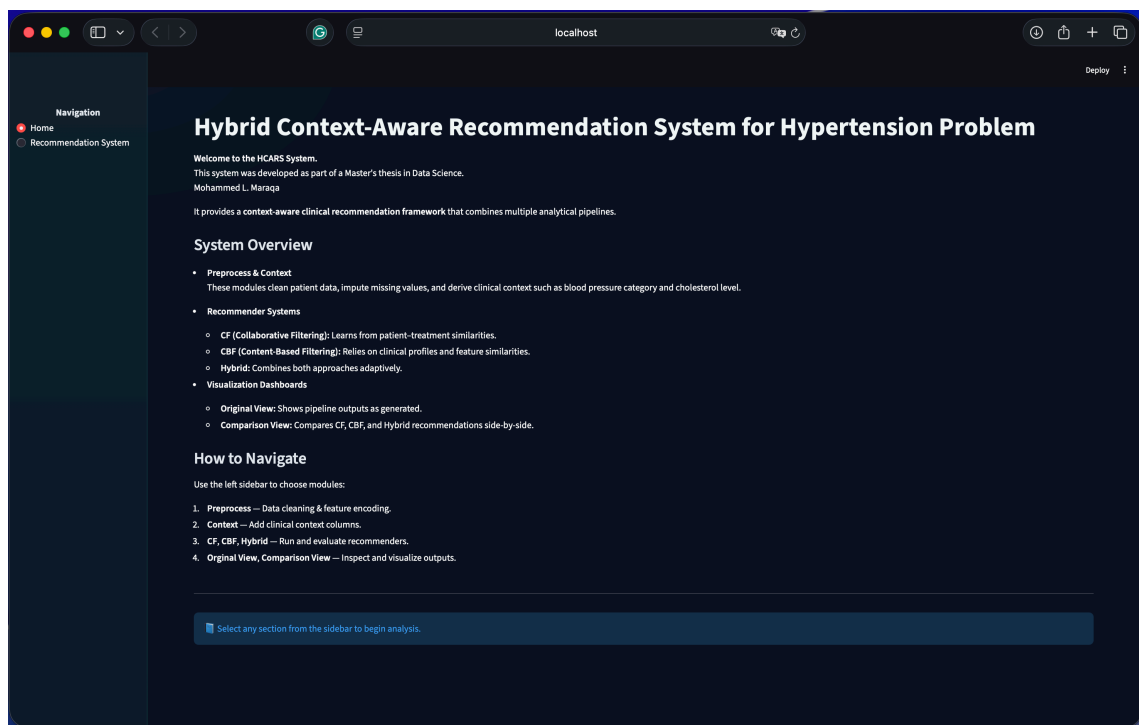


Figure (2).1: Home page: HCARS overview and navigation hints.

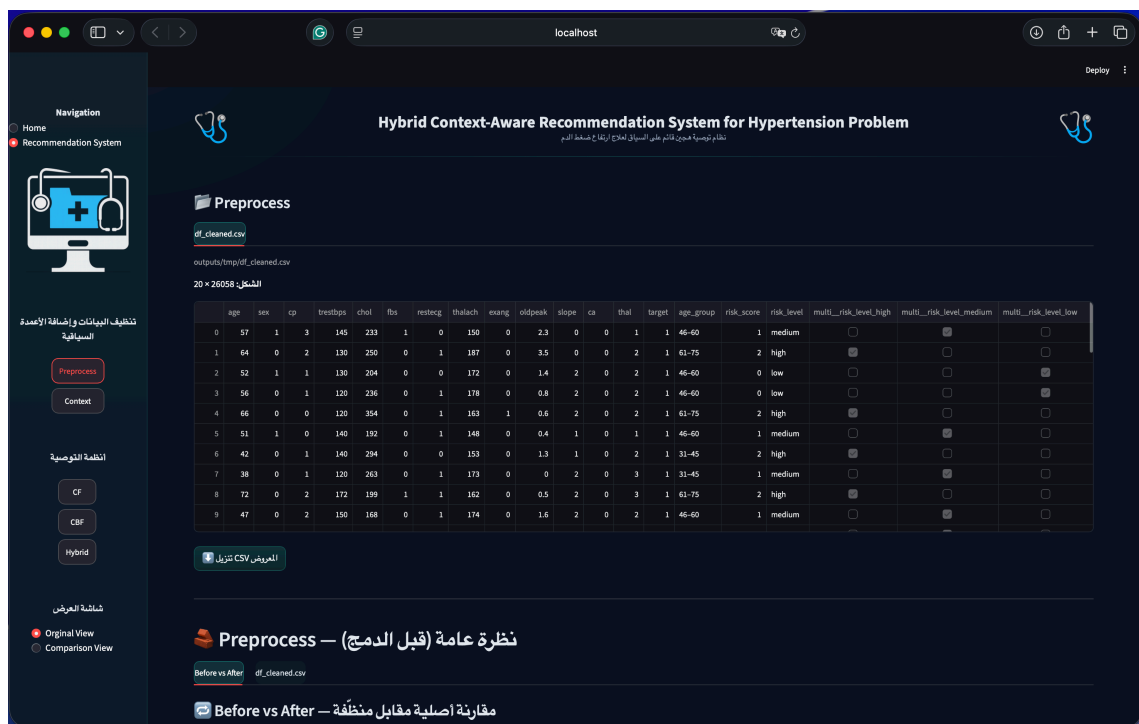


Figure (2).2: Preprocess Before vs After: side-by-side comparison panel.

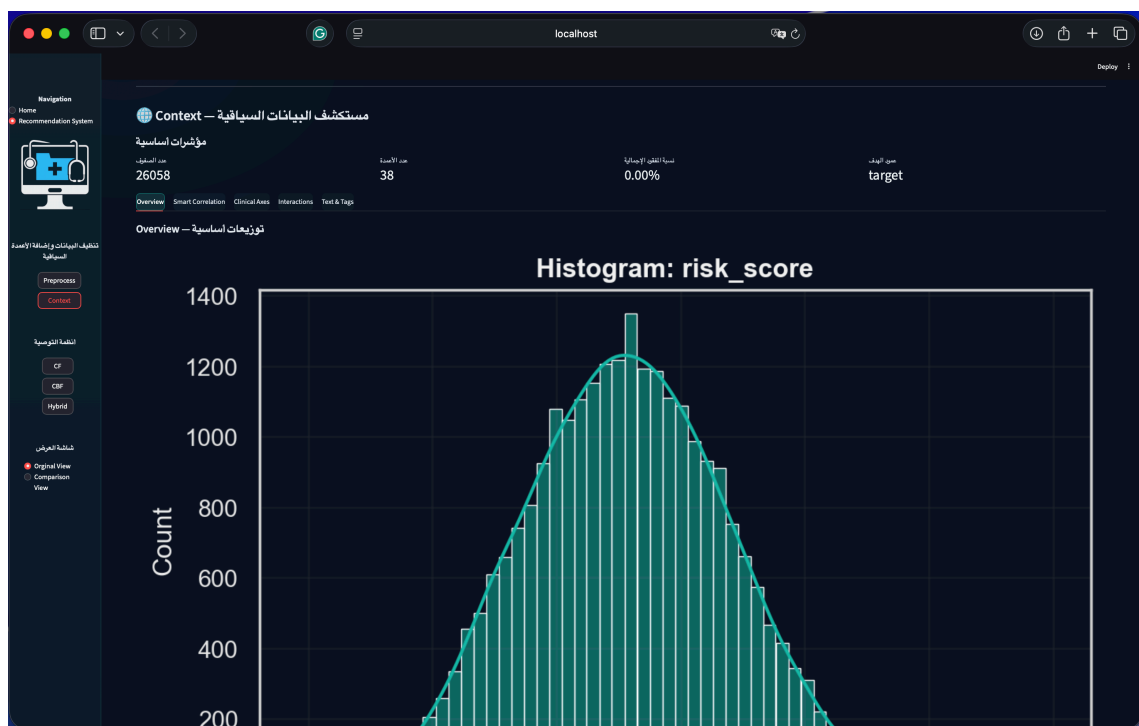


Figure (2).3: Preprocess module: cleaned data table (df_cleaned.csv) and exports.

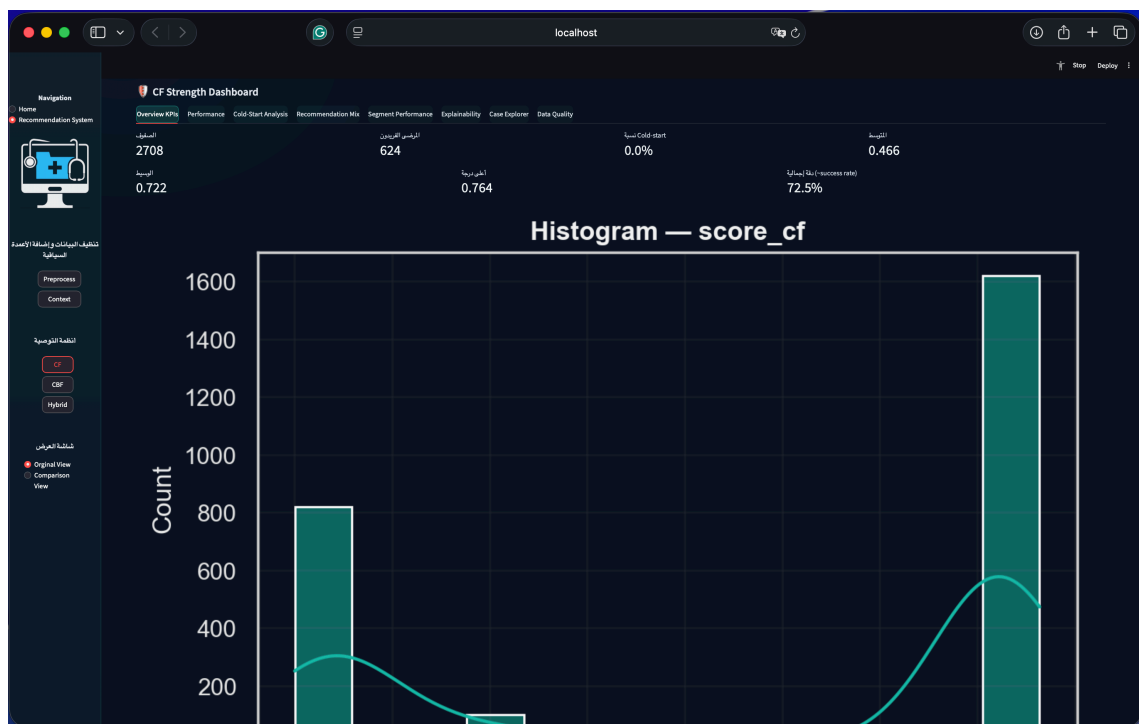


Figure (2).4: Context explorer: histogram of risk_score and top counters.

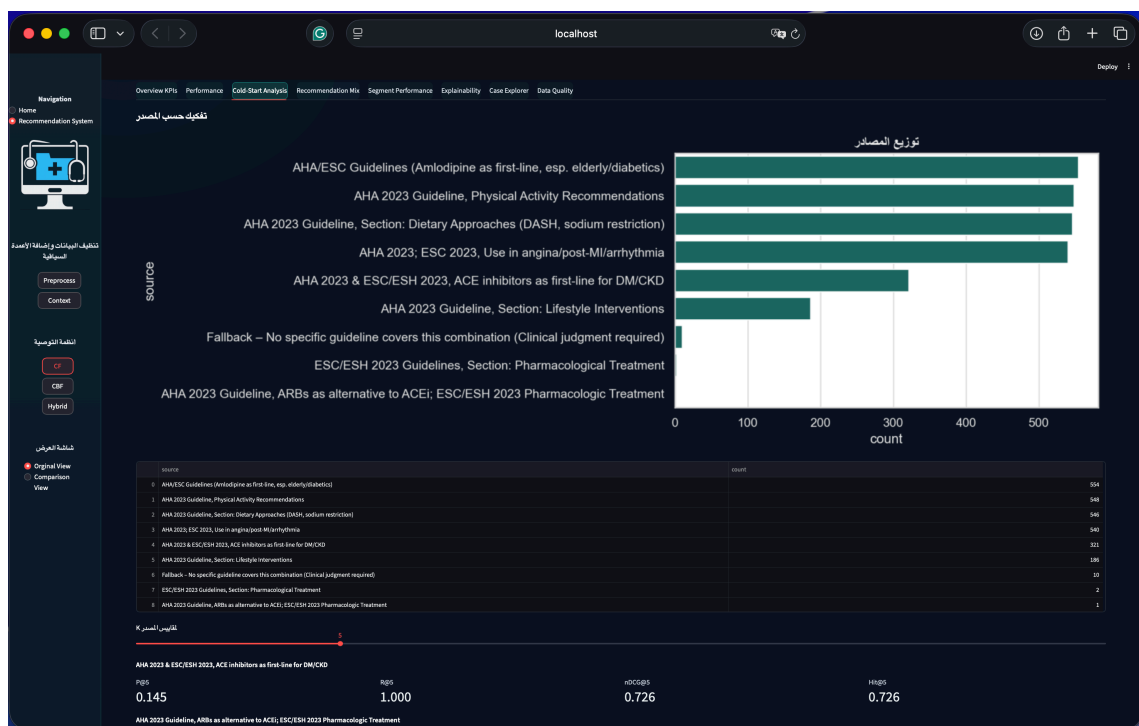


Figure (2).5: CF dashboard: overview KPIs and score distribution.

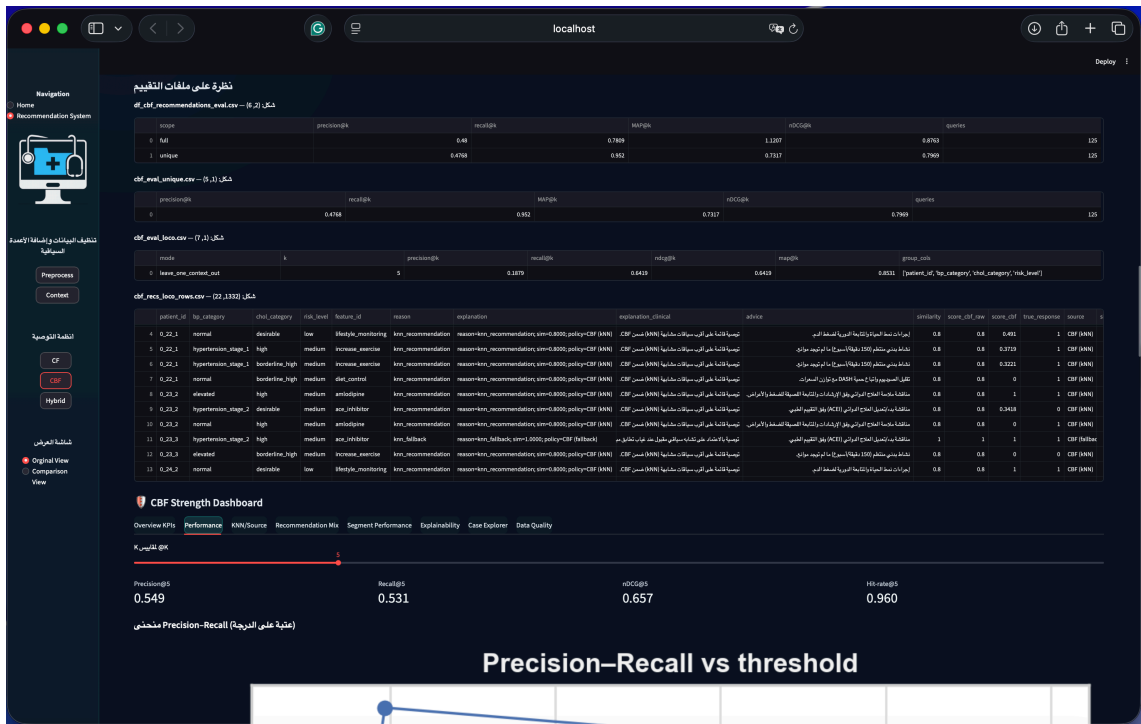


Figure (2).6: CBF dashboard: evaluation files overview, metrics, and detailed rows.

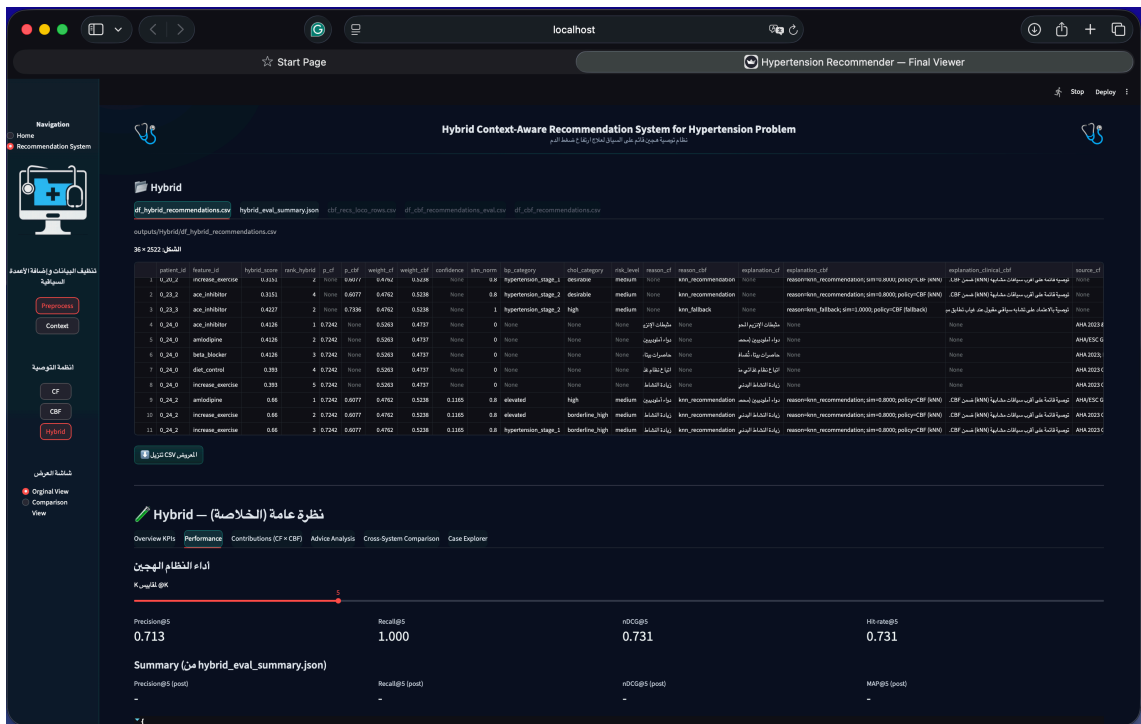


Figure (2).7: Display Pages (comparison canvas): CBF reason distribution vs. Hybrid score distribution.

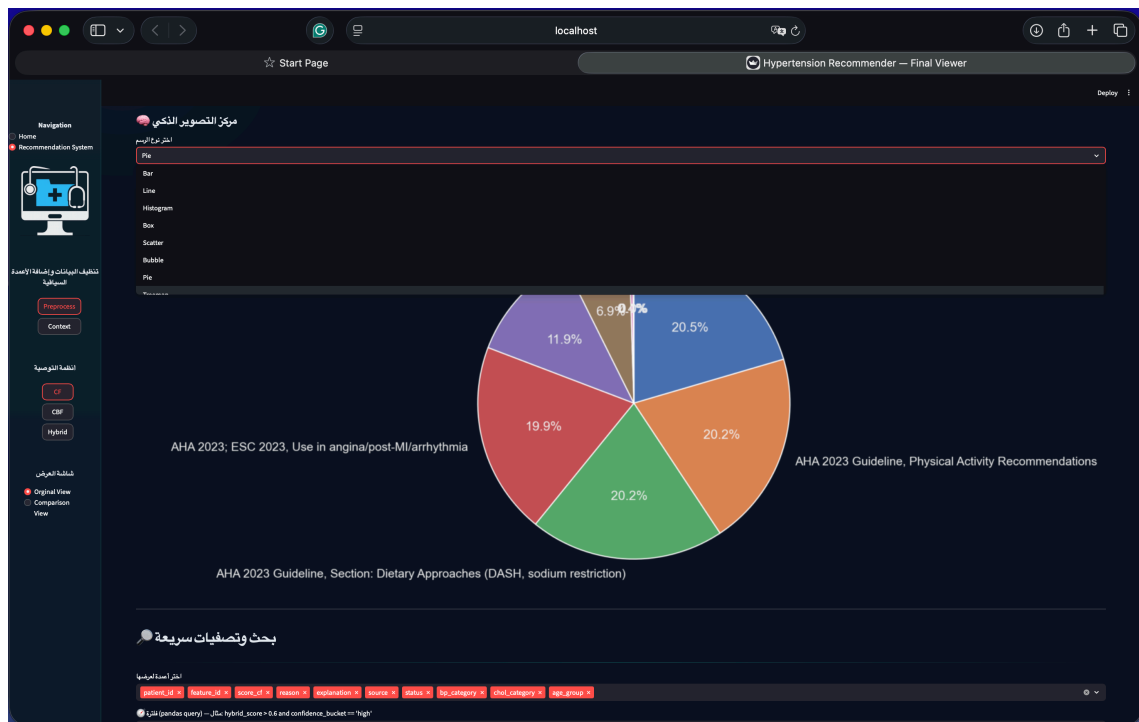


Figure (2).8: Smart Visualization Center.

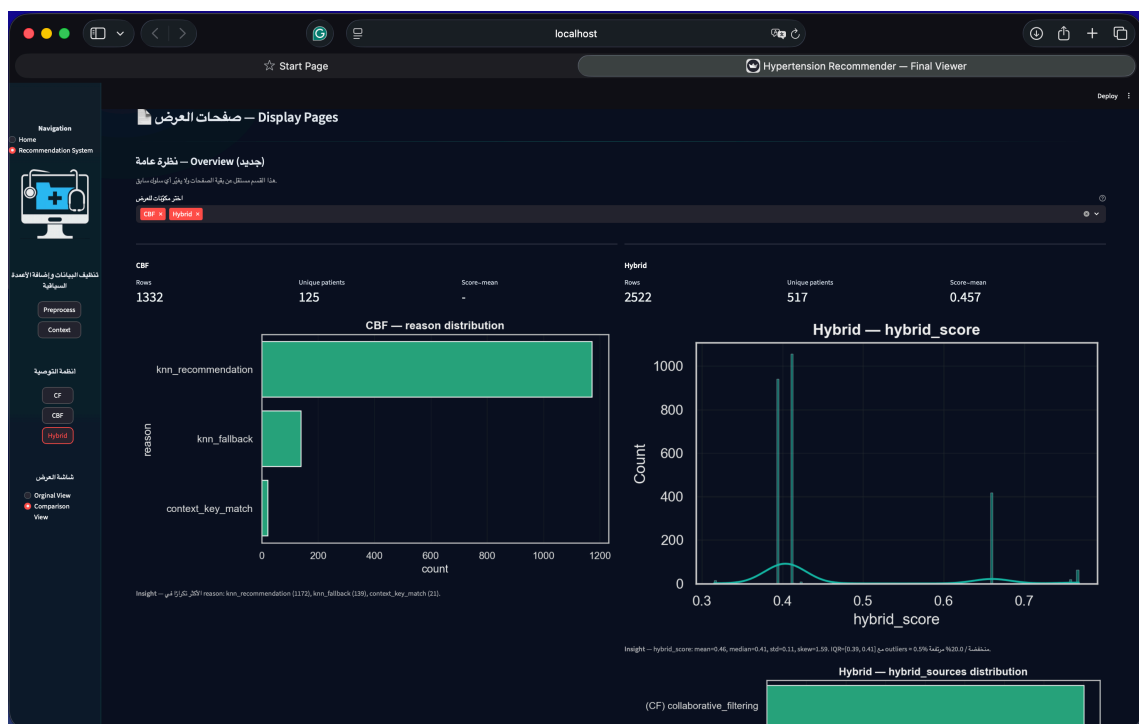


Figure (2).9: Comparison View in Recommendation (CF/CFB/Hybrid).

نظام توصية هجين واعٍ بالسياق لمشكلة ارتفاع ضغط الدم

محمد ليبب صبري مرقه

أعضاء لجنة الإشراف:

د. محمد حسين

د. إبراهيم نمر

د. أماني عوده

د. عبد اللطيف أبو عيسى

الملخص

يهدف هذا البحث إلى تطوير نظام توصية ذكي قائم على تقنيات الذكاء الاصطناعي لتقديم توصيات شخصية ومبنية على البيانات لمشكلة ارتفاع ضغط الدم. يحمل النظام المقترح اسم نظام التوصية الهجين الواعي بالسياق، ويجمع بين نهجين متكاملين هما التصفية التعاونية والتصفية المعتمدة على المحتوى ضمن إطار علمي وعملي موحد.

أجريت هذه الدراسة باستخدام منهجية تحليلية تطبيقية تجمع بين تصميم البرمجيات والتقييم التجريبي. واعتمد مجتمع الدراسة على بيانات حقيقية من مجموعة بيانات Kaggle الصحية التي تتضمن خصائص سريرية وديموغرافية وسلوكية للأفراد البالغين. وتم اختيار عينة الدراسة ومعالجتها مسبقاً وفقاً للمعايير الطبية، بما في ذلك مستويات ضغط الدم، وفئات الكوليسترول، ومستويات مخاطر أمراض القلب والأوعية الدموية، والعمر، والجنس، وعوامل نمط الحياة.

استخدمت الدراسة عدة أدوات بحثية، شملت تحليل البيانات وهندسة الخصائص واستخراج السياق، بالإضافة إلى مقاييس تقييم الأداء مثل الدقة، والاسترجاع، وجودة الترتيب. وأظهرت النتائج أن النموذج الهجين المقترح تفوق على نماذج التصفية التعاونية والتصفية المعتمدة على المحتوى كل على حدة، حيث حقق دقة (Precision) قدرها 0.72 واسترجاع (Recall) قدره 0.90، مما يشير إلى توصيات ذات درجة عالية من الدقة والشمولية. كما أظهر النظام مستوى قوياً من القابلية للتفسير من خلال لوحة تفاعلية مبنية على منصة Streamlit، بما يعزز الشفافية وقابلية التطبيق في بيئات الرعاية الصحية الرقمية.

توصي الدراسة باعتماد الأنظمة الهجينة الواعية بالسياق في دعم القرار السريري نظراً لقدرتها على الجمع بين التخصص، والموثوقية، وقابلية التفسير السريري. كما تشجع على توسيع مجموعة البيانات لتشمل سجلات طبية متعددة المؤسسات، وتطبيق الإطار نفسه على أمراض مزمنة أخرى مثل داء السكري واضطرابات القلب والأوعية الدموية.

الكلمات المفتاحية: أنظمة التوصية، ارتفاع ضغط الدم، الذكاء الاصطناعي، التصفية الهجينة، الوعي بالسياق.