A Model for Big Data Analytics to Improve Service Delivery in South African Public Hospitals

Mfanafuthi A. Makome

201601028


orcid.org/0000-0002-5797-5154

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Master of Computing

Supervisor: Dr. Olalekan Samuel Ogunleye

Co-Supervisor:

School of Computing and Mathematical Sciences

Faculty of Agriculture and Natural Sciences

May 2026

DECLARATION

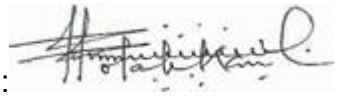
I **Mfanafuthi Armstrong Makome**, hereby declare that this dissertation entitled "***A Model for Big Data Analytics to Improve Service Delivery in South African Public Hospitals"*** submitted to the School of Computing and Mathematical Sciences, University of Mpumalanga for the award of the degree Master of Computing, is my original work and has never been submitted for assessment to any other university or for another qualification. I further declare that all sources used or consulted have been acknowledged by citation and inclusion in the reference list.

Also, note that part of this work has been accepted for publication as a conference paper titled ***"Leveraging Machine Learning and Big Data Analytics to Transform Service Delivery in Public Healthcare"*** at the IMITEC2025 Conference.

Signed:  Date: 30 Sept. 2025

Mfanafuthi A Makome

Student

Signed:  Date: 13th October 2025

Dr Olalekan Samuel Ogunleye

Supervisor

DEDICATION

This dissertation is dedicated to my mother, Lindiwe; my wife, Bonolo; my siblings; and my late family members, my father, my sister, and my uncle Louis (RIP), who have consistently encouraged me to pursue this qualification. Uncle Louis, who always reminded me that I still have the time and ability to achieve my goals.

ACKNOWLEDGEMENTS

I want to express my special dedication to my late supervisor, Prof. B. Kalema. Wherever you are, may your soul rest in peace. Your sudden passing was a shock to us, and it dealt a significant blow to our studies. However, I believe your prayers kept us going, and I know you left us in capable hands.

I would also like to express my gratitude to our current supervisor, Dr. O.S. Ogunleye, who welcomed us and guided us out of uncertainty, showing us that there is indeed light at the end of the tunnel. Most importantly, I thank Dr. W. Vambe, who has been a mentor throughout this journey and has always been available for calls and support.

I would like to express my gratitude to the Faculty of Agriculture and Natural Sciences, the School of Computing and Mathematical Sciences, and the staff at the University of Mpumalanga (UMP). Their contributions, resources and facilities created a supportive environment for research activities. The collaborative and engaging academic atmosphere at UMP contributed to shaping the results of this study.

The Vice Chancellor of the University of Mpumalanga, Professor Thoko Mayekiso, has created numerous opportunities, including a Vice Chancellor scholarship, an ATSP fellowship, and her role as an Associate Lecturer under her leadership, embodying UMP's slogan, *"Creating Opportunity."*

Finally, to Almighty God, for His gift of life and love, who has shown the anointing of favour. All glory to Him. AMEN.

TABLE OF CONTENTS

Table of Contents

LIST OF TABLES

LIST OF EQUATIONS

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligence
AIDS	Acquired immunodeficiency syndrome
AUC	Area under the curve
BDA	Big Data analytics
BERT	Bidirectional Encoder Representations from Transformers
COVID-19	Coronavirus disease 19
CRISP-DM	Cross-Industry Standard Process for Data Mining
DHMIS	District Health Management Information System
DFUs	Diabetic foot ulcers
EDA	exploratory data analysis
EHRs	Electronic health records
EMRs	Electronic Medical Records
HIV	Human immunodeficiency virus
HMS	Hospital management systems
HIS	Hospital information systems
IBM	International Business Machines Corporation
IT	Information Technology
ICT	Information and Communications Technology
KDD	Knowledge Discovery in Databases
ML	Machine learning
mHealth	Mobile healthcare
NLTK	Natural Language Toolkit
NHI	National Health Insurance
NoSQL	Not Only Structures Query Language
ROC	Receiver Operating Characteristic
SAMRC	South African Medical Research Council
SEMMA	Sample, Explore, Modify, Model, Assess
SHAP	SHapley Additive exPlanations

SVM	Support Vector Machine
SQL	Structured Query Language
UHC	Universal Health Coverage
VISP	Vaccine-induced seropositivity

ABSTRACT

Abstract: This study addresses the challenge of improving service delivery in South African public hospitals by utilising Big Data analytics and machine learning techniques. It employs a quantitative approach within the CRISP-DM framework. The research objectives included identifying factors that influence Big Data analytics, assessing various analysis methods, identifying patterns to improve services, developing a predictive model, and evaluating its efficiency and effectiveness. Utilising a synthetic dataset (Final_healthcare_big_data.csv), no real hospital/electronic health record data were used. The study implemented feature engineering and employed a Stacking Classifier consisting of XGBoost, Random Forest, CatBoost, and LightGBM to build the final predictive multiclass healthcare model categorizing outcomes as 'Improved,' 'Unchanged,' or 'Worsened.' Initially, during the trial stage, various algorithms were trained, including SVM and logistic regression; however, they were later dropped while building the final model as they were not suitable for this study, due to extended training durations unsuitable for multiclass. SHAP analysis highlighted critical features such as Satisfaction_Rating, Age, and Log_Wait_Time, and the stacked classifier model achieved an accuracy of 84% with AUC values of 0.93, 0.95, and 0.95, respectively. Patterns, like Wait_Length_Interaction, which had an impact on worsened outcomes, were identified, and a Streamlit application was developed to provide real-time insights. The findings demonstrate the model's potential to enhance healthcare decision-making by providing a scalable and interpretable solution. This study advances data science by laying a strong foundation for healthcare analytics in South Africa, with implications for resource allocation and improving patient care efficiency.

Keywords: *Healthcare, Big Data analytics (BDA), machine learning, healthcare outcomes, CRISP-DM, Service delivery, F1-Score, ROC Curves, Random Forest, XGBoost, CatBoost, LightGBM, Stacking Classifier, SVM, Logistic Regression, Multiclassification, Confusion Matrix, SMOTEENN, Python, Streamlit*

CHAPTER ONE: INTRODUCTION AND RESEARCH BACKGROUND

This chapter provides a general introduction to the field of study and presents the background to the research problem. This chapter is structured as follows: Section 1.4 outlines the aim, followed by the study's research objectives in Section 1.5 and research questions in Section 1.6, respectively. The research methodology to be adopted in this work is discussed in Section 1.7, while Section 1.8 highlights the significance of the study. The delimitations and limitations of the study are detailed in Section 1.9, while Section 1.10 outlines the ethical considerations that guided the research. Finally, Section 1.11 concludes the chapter.

1.1 Introduction to the Field of Study

The increasing use of information technology (IT) in various application areas has led to the generation of massive amounts of data. This data is generated at high speed, in significant volumes, and comes from various sources. This massive data, also known as big data, is characterized by its significant volume, high speed, and originates from various sources (Abdalla, 2022). In South African public hospitals, vast amounts of data are generated daily from diverse sources, including patient records, medical histories, and administrative records (Marutha & Ngoepe, 2018). The challenge lies not in collecting this data, but in effectively analyzing it to improve service delivery, as its characteristics challenge traditional transactional processing systems (Rahul et al., 2023).

According to IBM (2024), the primary distinction between big data analytics and traditional data analytics lies in how the data is handled and the analytical tools used. Traditional analytics focuses on structured data, which is usually stored in relational databases. These databases organise data in a way that is easily understandable by computers. Traditional data analytics utilizes statistical methods and tools such as Structured Query Language (SQL) to query these databases. In contrast, big data analytics deals with vast amounts of data in various formats, including structured, semi-structured, and

unstructured data that organisations gather, analyse and use to gain valuable insights (Favaretto et al., 2020). The complexity and diversity of this data necessitates more advanced analytical techniques. Big data analytics employs sophisticated methods, including text analytics, social media analytics, predictive analytics, in-memory analytics, graph analytics, statistical methods, machine learning and data mining to extract insights from complex datasets. Additionally, it often requires distributed processing systems such as Hadoop, Apache Spark, Google, and BigQuery, among others, to manage and process data (Sun et al., 2023).

Big Data analytics can be applied in various sectors to provide solutions through the analysis of available datasets. The analyzed data can be used to improve productivity and reduce costs, leading to financial gain and a competitive advantage (David & Torshin, 2023). For instance, in finance, Big Data Analytics helps detect fraudulent activities, manage risks, and ensure regulatory compliance. Big Data analytics (BDA) monitors real-time transactions and flags suspicious activities, thereby preventing fraud and enhancing security (Shoetan et al., 2024).

In the retail sector, BDA can be used to understand customer behaviour, optimise supply chains, and personalize marketing efforts. Retail giants like Amazon use BDA to analyse purchase patterns, manage inventory, and predict consumer trends, thereby increasing efficiency and customer satisfaction (Fisher & Raman, 2018). Furthermore, BDA can assist retailers in discovering new opportunities and innovations, refining their existing business models, and discontinuing unproductive areas.

On the other hand, governments use BDA to enhance public safety, improve transportation systems, and manage urban planning (Munné, 2016). Cities can optimise traffic flow, reduce congestion, and improve public transit services by analysing data from traffic sensors and public transportation systems (Cavanillas et al., 2018). Big Data plays a crucial role in disaster management and response. By analysing data from various sources, emergency services can predict natural disasters, coordinate response efforts, and allocate resources effectively, saving lives and reducing damage (Yu et al., 2018).

Public healthcare can benefit significantly from big data and predictive analytics, enhancing monitoring, assessment, epidemiological research, and population health needs analysis (Bandi et al., 2024). The vast amount of data is invaluable for evaluating population-based interventions and informing policymaking. Effective analysis of this big data could yield meaningful insights, aiding the health sector in enhancing service delivery. However, some public healthcare systems struggle with determining the optimal methods for analysing big data (Pastorino et al., 2019).

1.2 Background of the Study

South Africa's healthcare system operates on a two-tiered public and private healthcare model characterised by significant discrepancies. The public healthcare sector, funded by the state, serves approximately 71% of the population. In contrast, the private sector relies on individual contributions to medical aid schemes or health insurance and caters to around 27% of the population (Rensburg, 2021). The public healthcare sector faces chronic underfunding, while private healthcare remains prohibitively expensive for most South Africans (De Maaye & Coovadia, 2024). Over the years, South Africa’s public healthcare sector has experienced a continuous decline in service delivery. This is a result of persistent underfunding, which has impaired the ability of the health system to meet healthcare demands. Some healthcare facilities in South Africa are understaffed and overcrowded, resulting in longer patient waiting times, increased strain on the facility staff, and frustration for individuals seeking medical attention.

Subsequently, healthcare personnel often struggle to handle the volume of patients and patients’ data effectively and efficiently, resulting in delays in receiving care. A preliminary investigation revealed that the South African public health sector continues to employ traditional methods for storing, processing, and analyzing data (Katurura & Cilliers, 2018). Traditional methods lead to inefficient, high-cost, loss of files, limited analytical capability, and reduced quality of patient care (Nkoane, 2023). Furthermore, traditional data

processing methods are often insufficient for managing such complexity, leading to data silos and fragmented insights.

In response to these challenges and with the advancement of technology, other countries such as the United States of America (Mumtaz et al., 2023), China (Liang et al., 2021), Malaysia (Ghaleb et al., 2023) and Tanzania (Kala, 2023) among others, have adopted the use of technology in storing; BDA for processing and managing patients data. Big data analytics enables hospitals to make more informed decisions, optimize resource use, and improve overall management and patient care quality, as it offers valuable insights into resource allocation, patient care patterns, and administrative processes (Adekunle et al., 2024). Big Data in healthcare comes from various sources, including patient demographics, treatment history, diagnostic reports, and IoT devices (Malik et al., 2018). Proper management and analysis of this data are crucial for deriving reliable insights that can inform strategic decision-making (Goyal & Malviya, 2023). Big Data analytics in healthcare offers significant potential benefits, including improved treatment quality, better public health surveillance, and enhanced disease response (Mehta et al., 2019).

Despite the vast amounts of data being generated daily and the advantages of BDA, surprisingly, healthcare facilities in South Africa still rely on traditional methods to store, process, and analyse data. The underutilisation of BDA tools in healthcare significantly impedes the effective use of data for improving healthcare services (Wang et al., 2018). One of the significant issues is the inability to fully integrate and analyse the vast and varied data collected from numerous sources. Healthcare data, derived from electronic health records, medical imaging, wearable technology, genomics, and patient feedback, is inherently diverse and voluminous (Cremin et al., 2022). Moreover, healthcare systems lacking BDA capabilities often experience inefficiencies and missed opportunities to enhance patient outcomes and operational performance.

With the advantages of technology and BDA in the healthcare sector cited above, there is a need for a shift towards the adoption of technology and BDA in South Africa. New strategies and technologies must be developed to understand the nature, complexity, and

volume of healthcare data in order to derive meaningful information (Awrahman et al., 2022) in South Africa and advance towards a predictive system, enabling the prediction of future health outcomes based on current data (Nijjer et al., 2020). It will also facilitate population health management, early disease detection, and the discovery of therapeutic countermeasures (Fragala et al., 2019). This calls for a concerted effort to have a big data analytics model to improve service delivery in South African Public Hospitals.

1.3 Research Problem

Approximately 73% of South African households rely on public healthcare services as their first point of contact (Statistics South Africa, 2023), with an estimated 80 - 84% of the population dependent on the public system due to the high costs of private care (Maphumulo & Bhengu, 2019). Compounding this demand, foreign nationals also access these services, though no official statistics exist on their numbers or proportion, as the public health system does not record patients' citizenship or immigration status (Minister Aaron Motsoaledi, August 2025, as reported in Business Day/TimesLIVE).

Human resource shortages and systemic constraints result in prolonged waiting times, often exacerbating patients' conditions (Ngene et al., 2023). Service delays stem from multiple factors, including inadequate analysis of patients' historical medical data, limited real-time synchronization and pattern recognition for similar conditions, and insufficient examination of past disease outbreaks or related health datasets. Much of this data remains fragmented across incompatible formats, rendering it inaccessible to traditional transactional processing systems (Zali, 2023).

Big Data analytics (BDA) and advanced technologies could significantly enhance service delivery by enabling efficient processing, storage, and analysis of large-scale medical data. However, the South African public health sector continues to rely predominantly on conventional methods, with limited adoption of BDA (Muhunzi et al., 2023). Existing BDA models, largely developed in Western contexts, emphasize high-resource environments and lack adaptation to South African realities such as data silos in public facilities,

resource constraints in rural areas, multilingual patient records, and integration needs for the National Health Insurance (NHI) framework (Muhunzi et al., 2023; Yan et al., 2025).

Effective BDA implementation could facilitate real-time comparison of medical histories, faster diagnosis and prescription, and predictive insights into emerging health threats, benefiting both public and private sectors (Batko & Ślęzak, 2022; Hummel et al., 2024). Despite these potential gains, there remains a critical gap: the absence of a contextually appropriate BDA model tailored to South Africa's public healthcare challenges. This study addresses this gap by developing a South African-specific data analytics model to support improved service delivery and contribute to the realization of universal quality healthcare under the NHI, as enshrined in the Constitution

1.4 Research Aim and Objectives

The aim and objectives of this study are as follows;

1.4.1 Aim of the study

The aim of this study was to develop a big data analytics (BDA) model to service delivery in South African public hospitals.

1.4.2 Objectives

To achieve the above aim, this research was guided by the following objectives.

1. To identify factors influencing Big Data analytics in the health sector in the world and in the South African public healthcare sector context.
2. To assess the various technological techniques/tools and methods used to analyse health big data in the world and South African public healthcare sector context.
3. To use Big Data analytics techniques and tools to draw patterns and insights needed for improving service delivery in public healthcare.
4. To use machine learning (ML) classification techniques to develop a model of Big Data analytics that will be used to make better decisions and predictions needed for improved service delivery.
5. To test and evaluate the effectiveness and efficiency of the model.

1.5 Research Questions

The research was guided by the following research questions.

1.5.1 Primary Research Question

How can a Big Data analytics Model be developed to improve service delivery in South African public hospitals?

1.5.2 Secondary (sub) questions

1. What factors influence Big Data analytics in the health sector in the world and in the South African public healthcare sector context?
2. What methods are used to analyse Big Data in the health sector in the world and in the South African public healthcare sector context?
3. What Big Data patterns and insights are needed to improve public healthcare service delivery?
4. What machine learning (ML) techniques are suitable for developing an effective and efficient Big Data analytics model to improve decision-making and prediction for better service delivery?
5. How effective and efficient is the developed Big Data analytics model in supporting service delivery in public healthcare?

1.6 Justification of the Study

In the context of South African public hospitals, investing in BDA can address pressing issues such as resource allocation, patient management, and operational inefficiencies, which lead to poor public healthcare service delivery (Iyamu, 2020). Public hospitals often face challenges like limited resources, high patient volumes, and varying levels of care quality. Through the integration of BDA, hospital administrators can gain insights into patterns of patients' medical history and how specific diseases are treated and also learn which healthcare is good in specialisation relating to and trends that were previously undetectable, such predictive analytics can help anticipate patient admissions, allowing for better resource planning and reducing waiting times and improvements can lead to

enhanced patient satisfaction and better health outcomes (Bhati, Deogade & Kanyal, 2023).

According to Ferreira et al. (2023), enhanced service delivery results in improved patient outcomes, reduced wait times, and higher patient satisfaction levels in healthcare. Efficient resource management ensures that patients receive timely and adequate care, reducing the overall burden on the healthcare system. This can only be achieved by correctly analysing the data generated within healthcare settings. Furthermore, medical staff will benefit from optimised workflows and reduced administrative burdens, allowing them to focus more on patient care. The predictive capabilities of the BDA models can also assist in clinical decision-making, improving treatment outcomes (Paul et al., 2023).

The insights generated from this study and the developed model can guide policy formulation and implementation, ensuring that healthcare resources are allocated equitably and efficiently across the healthcare system, as supported by Ramezani et al. (2023). By analysing data to accurately identify common disease outbreaks in the regions and determine the necessary prescriptions, the model leverages data insights to predict outbreaks, ensure the availability of necessary medications, and support policymakers in making informed decisions for better healthcare planning and management. This is in line with Clack, (2023) findings. Policymakers can also identify areas that require targeted interventions to improve service delivery in under-resourced regions. Additionally, these insights can help policymakers make informed decisions about resource distribution and healthcare policies, ensuring that urban and rural hospitals are adequately equipped to meet patient needs (Chao et al., 2023).

Lastly, despite the proven benefits of BDA in healthcare, there is a notable lack of research focusing on its application in the South African public healthcare system (Aikman, 2019). Recent studies have shown that BDA can significantly improve healthcare outcomes by enhancing diagnostic accuracy, personalising treatment plans, and enabling predictive analytics (Nkwanyana et al., 2023). This calls for more research in this domain as most existing studies are centered on developed countries and lack contextualization from South African perspectives, which have wider cultural diversities

(Goundar et al., 2021). Furthermore, the deployment of Big Data analytics (BDA) in South Africa's healthcare sector must be customized to meet the distinct demands and challenges of its healthcare model. Moreover, integrating indigenous health datasets and managing data privacy issues are pivotal in crafting robust analytical solutions (Ahmed et al., 2023).

1.7 Contributions of the Study

This research aims to develop and deploy Big Data analytics models to enhance service delivery in South African public hospitals, thereby addressing a critical need in healthcare administration. Furthermore, it will assist the South African government-led NHI National Health Insurance(NHI) in providing universal access to quality health care for all South Africans (as enshrined in the Constitution). Therefore, the worth of this research extends to both theoretical and practical contributions.

1.7.1 Theoretical Contribution

This research contributes to the existing body of knowledge and the ongoing debate of Big Data analytics in the healthcare sector. The model developed will be used to extend research by testing and retesting it in various circumstances, thereby contributing to existing literature, particularly from Big Data analytics, public healthcare, and developing countries.

1.7.2 Practical Contribution

This research benefits healthcare practitioners, including hospital administrators, policymakers, and medical staff. The developed Big Data analytics model will be equipped to make predictions, enabling the management and control of future incidents and events. The model also provides actionable insights that can be used, ultimately improving patient care, optimize resource allocation, and enhance overall operational efficiency, to improve service delivery in the health sector. Furthermore, the model can be used to predict patient outcomes, disease treatment, admissions, and health records and adapt staffing levels, resulting in reduced wait times, enhanced patient satisfaction

and improved healthcare service delivery.

1.8 Research Delineation

This study utilised synthetic datasets to train and build the model. All experiments utilise synthetic data generated via Python code. The study was limited to synthetic datasets that were generated to mimic the health care domain used for decision-making. As much as big data models can be used for diagnosis and prescription, this was outside the scope of this study. No real hospital/electronic health record data were used. This work presents a proof-of-concept / prototype using fully synthetic data to assess the technical feasibility of the proposed method

1.9 Chapter Summary

This chapter discusses the introduction of the field of study, and the background of the study is highlighted. The chapter discussed the research problem, research questions and objectives. Furthermore, the chapter discusses the justification of the study, the importance, and the research contribution to the body of knowledge, and lastly, stipulates the constraints of the study. In Chapter Two, the literature review and related work underpinning this study will be discussed.

CHAPTER TWO: LITERATURE REVIEW

This chapter provides a detailed overview of Big Data and its applications in the South African public healthcare sector. Section 2.1 introduces the concept of Big Data, beginning with subsection 2.1.1, which discusses the application of big data analytics, and subsection 2.1.2, which examines the factors influencing big data. It then covers tools and techniques in big data, concluding with the benefits and challenges of big data. Following this, Section 2.2 discusses the South African Public Healthcare Sector, beginning with its challenges in subsection 2.2.1, technology advancements in 2.2.2, and current tools and techniques in the South African context in 2.2.3, before focusing on big data analytics in South African public health. The literature review then transitions to related work in Section 2.3, summarizing existing studies. Additionally, the chapter reviews current data analysis methods and technological systems in South African hospitals, compares related literature, and highlights observed gaps and differences. The discussion in this chapter addresses research questions and objectives one, two, and three. It further provides context for the forthcoming selection of machine learning approaches outlined in objective and question four.

2.1 The Concept of Big Data

The increasing growth in data has necessitated the development of new technologies and architectures to store, process, and analyse the data, leading to advancements in fields such as big data, machine learning, artificial intelligence, and big data analytics (Bose et al., 2023). Big data encompasses massive volumes of structured, semi-structured, and unstructured data generated and stored by organisations and individuals, characterised by the five V's: volume, velocity, variety, veracity, and value (Sunagar & Alam, 2020).

Big Data has emerged as a crucial frontier for innovation and development across sectors, including healthcare, where it enables better decision-making, greater efficiency, and competitive advantage (Rahul et al., 2023). In healthcare, sources like electronic health records (EHRs), medical imaging, genomic data, and wearables generate vast

datasets daily, necessitating BDA to uncover hidden patterns, improve patient outcomes, optimize resources, and inform public health strategies (Shaik et al., 2024; Zaripova et al., 2023)

.

2.1.1 Application of Big Data and Data Analytics in the health sector

Healthcare big data includes patients' histories, diagnoses, treatments, radiology images, clinical trials, insurance claims, and behavioral data (Dash et al., 2019; Oussous et al., 2018). BDA supports diagnoses, predictions, prescriptions, and forecasting disease outbreaks, though traditional systems struggle with unstructured, heterogeneous data (Hulsen et al., 2019; Pastorino et al., 2019). Benefits include enhanced decision-making, robust supply chains, and pandemic mitigation (e.g., COVID-19), but require technical, business, and strategic integration (Padarath & Moeti, 2023; Shabbir & Gardezi, 2020).

2.1.2 Factors influencing Big Data Analytics

Pugliese et al. (2021) note that the diversity of data in Big Data environments demands sophisticated tools and technologies to extract valuable insights. For example, NoSQL databases are often employed to handle unstructured data. In addition to these technologies, factors such as privacy protection when processing large datasets are important. Thus, managing and processing big data, particularly when utilizing machine learning, requires careful attention to trust, transparency, and ethical concerns.

Moreover, Gandomi and Haider (2015) emphasise that the ability to analyse various data types from multiple sources is crucial for gaining comprehensive insights and making informed decisions. Furthermore, factors such as the scale and complexity of Big Data are significant in shaping the application of ML technologies, which reinforces the importance of the tools and strategies identified by Pugliese et al. (2021).

According to Edu and Agozie (2022), real-time data processing capabilities are also vital in today's fast-paced digital world, as they support timely decision-making. Technologies that enable and facilitate real-time data streaming and analysis help organisations to respond promptly to emerging trends and issues. In this context, Edu and Agozie (2022) identified several key metrics and data sources, including patient historical records, decision-making processes, disease records, and data records, that play a significant role in adopting Big Data analytics.

According to Surbakti et al. (2019), the adoption and successful implementation of Big Data analytics in organisations are influenced by several key factors, including technological readiness, including the robustness of IT infrastructure and the perceived ease of use. These factors play a vital role, as organisations with advanced IT capabilities are better equipped to integrate Big Data analytics. Also, Lutfi et al. (2022) highlight the importance of ensuring data security and addressing privacy concerns, which builds trust and promotes acceptance across sectors.

In addition to technological and security considerations, organisational support plays a critical role, particularly from top management. A data-driven culture with skilled personnel significantly enhances the likelihood of successful big data analytics adoption and training programs to boost employees' data literacy (Prakash, 2024). Furthermore, Frisk and Bannister (2017) note that external factors such as market competition and regulatory frameworks further shape the adoption of big data analytics. With competitive pressure driving innovation and favourable government policies, technological integration can be facilitated, which is a driver for adopting big data technologies.

Table 2.1 presents factors from Aldossari et al. (2023), adapted here with critical evaluation for South African public healthcare.

Table 2.1 Factors influencing Big Data analytics (Adopted from Aldossari et al., 2023)

Category	Factor	Description	In the healthcare context	Critical Evaluation in the SA Public Healthcare Context
Technology	Data Storage Solutions	Scalable storage systems like Hadoop and NoSQL databases are essential for managing big data.	Needed for vast, secure healthcare data.	Essential but challenged by unreliable electricity and legacy systems in rural facilities; limits velocity processing.
	Computational Power	High processing power is needed to analyse large volumes of data in real-time.	Required for complex datasets	High needs conflict with resource constraints; private hospitals adopt faster than public ones.
	Data Integration	Integrating data from multiple sources seamlessly for holistic analysis.	Enables coordinated care	Fragmented silos (e.g., EMRs vs. paper) exacerbate delays; NHI demands better integration.
	Advanced Analytics Tools	Use sophisticated tools like machine learning and Artificial Intelligence (AI) for deep insights and predictions.	Enhances diagnostics/personalization	Global tools (e.g., TensorFlow) powerful but require adaptation for SA data bias/privacy (POPIA).
Human Resource	Skilled Personnel	Expertise in data science, analytics, and domain knowledge to interpret data effectively.	Translates data to medical insights	Severe shortages in public sector; training needed to bridge public-private divide.
Data Management	Data Quality/Volume/Variety/Velocity/Veracity/ Value	Accuracy, scale, diversity, speed, trust, insights.	Crucial for reliable care/outcomes	High variety/velocity from wearables/EHRs, but veracity low

				due to errors; value untapped without localized models.
Data Governance	Data Security/Privacy	Breach protection, regulatory compliance	Vital for sensitive patient data (e.g., HIPAA-like)	POPIA compliance critical, but breaches risk high from outdated infrastructure/third-party vulnerabilities.
Organizational Context	Organizational Readiness	Culture/infrastructure for data-driven decisions	Requires investment/cultural shift	Low in public hospitals due to bureaucracy/funding; NHI could drive but lags in practice.
	Cost Considerations	Balancing the costs of big data initiatives with expected returns and benefits.	While costly, Big Data analytics can lead to significant savings by improving efficiency and patient outcomes.	High upfront costs prohibitive in underfunded public system; ROI slow without govt support.
	Regulatory Compliance	Adherence to legal standards governing data usage and analysis.	Compliance with regulations is crucial to ensure that data analytics practices are legal and ethical.	NHI/POPIA amplify need, yet enforcement gaps hinder adoption.

2.1.3 Tools, Techniques, and Machine Learning Algorithms in Big Data Analytics

Processing big data requires specialized tools beyond traditional systems. Hadoop excels in distributed batch processing and storage, while Apache Spark offers faster in-memory computation, making it suitable for iterative analytics (Sewal & Singh, 2021; Vijay et al. (2024) note that Hadoop's effectiveness lies in its capabilities for batch processing and data storage, while Spark is praised for its in-memory processing capabilities, which significantly enhance the speed of data analytics tasks. Tools like MongoDB, Cassandra, and HBase are broadly used for data storage and managing unstructured or semi-

structured data (Muhammad et al., 2020). These NoSQL databases offer scalability and flexibility, making them suitable for handling the diverse data types often encountered in big data analytics.

Machine learning libraries such as Pandas, NumPy, Scikit-learn, TensorFlow, and PyTorch support complex analysis and modeling (Sundaram et al., 2023). Key algorithms applied in healthcare include:

- **Supervised learning**: Random forests and XGBoost for classification and regression (e.g., predicting readmissions or resource needs) - strengths include interpretability and handling imbalanced data common in disease datasets; weaknesses include sensitivity to noisy features.
- **Unsupervised learning:** K-means or DBSCAN clustering for patient segmentation or anomaly detection in outbreaks - effective for exploratory analysis but requires careful validation.
- **Deep learning:** Convolutional Neural Networks (CNNs) for medical imaging - high diagnostic accuracy but computationally intensive and less interpretable (black-box issue), raising concerns about clinical trust.

Natural Language Processing tools (NLTK, SpaCy, BERT) enable analysis of unstructured clinical notes (Tyagi & Bhushan, 2023). While these tools and algorithms deliver powerful insights globally, their deployment in resource-constrained environments demands adaptation for scalability, interpretability, and integration with legacy systems.

2.1.4 Benefits and challenges of Big Data Analytics in healthcare

Big data analytics in healthcare has immense benefits, especially in enhancing medical decision-making processes (Rehman et al., 2021). A study by Setyati et al. (2024) demonstrated its advantages, which include early disease detection and improved health management. However, Maxwell et al. (2023) argued that practical data analysis can only fully realise these benefits. Badr et al. (2024) also support the notion that the successful

analysis of Big Data can significantly enhance healthcare outcomes. However, Stoumpos et al. (2023) point out that while technology can revolutionise healthcare, inefficiencies and inaccuracies in data processing often hinder these advancements.

This situation highlights the pressing need for integrated, secure analytical solutions (Miandoab et al., 2023). Alowais et al. (2023) emphasise that manual data analysis can compromise its value, making automated data processing crucial. Udeh et al. (2024) simplify the concept by asserting that the true potential of Big Data lies in its analytics, because it drives more informed decision-making. Furthermore, Khosravi et al. (2024) emphasise that healthcare advancements hinge on uncovering valuable insights within Big Data, achievable only through sophisticated analytical tools such as AI and machine learning (Carrasco et al., 2024).

Healthcare systems worldwide are being increasingly encouraged to adopt Big Data analytics as a core strategy (Brossard et al., 2022). This adoption is expected to enhance treatment quality, improve public health monitoring, and strengthen responses to widespread health concerns (Paul et al., 2023).

Despite these benefits, significant challenges remain. One key issue is the lack of data integration due to the diverse sources and formats of healthcare data (Walker et al., 2023). Additionally, these challenges vary across countries, necessitating solutions tailored to specific national contexts (Hosseini et al., 2024).

In the healthcare sector, Big Data analytics can tackle the complexities within information systems. However, these analytics face several obstacles. Muhunzi et al. (2023) highlight the difficulty of developing robust methods critical for healthcare improvement. The potential of these methods is thought to lie in strategically layering various analytics applications to address issues such as clinical decision support, personalized care, public health management, and policy development (Hossin et al., 2023). Current systems, such as MapReduce, Hadoop, and STORM, are already employed to manage data storage, retrieval, analysis, visualisation, and security (Nuthalapati & Nuthalapati, 2024).

The challenges of Big Data in healthcare systems are multifaceted and complex. One of the primary challenges is the large volume of data generated, which is often unstructured and heterogeneous (Sivarajah et al., 2017). The rapid growth of healthcare data poses significant challenges in storing, searching, capturing, sharing, and analysing data in real-time. This is particularly critical in medical settings where timely decision-making is crucial for patient care (Rahul et al., 2023).

A study by Shah et al. (2019) highlights several challenges in applying Big Data analytics in the healthcare sector. These challenges include data cleaning difficulties, such as issues with normalization, noise reduction, and handling missing data. Additionally, the integration of diverse data from various sources, the evolving nature of healthcare data standards, and the need for scalability are significant obstacles (Sedláková et al., 2023). Technical challenges such as data storage, security, and the visualization of large, complex data sets also pose significant challenges. Moreover, ensuring the accuracy and consistency of data while overcoming errors in data entry and staging is crucial for effective healthcare analytics (Shah et al., 2019). Furthermore, it emphasises that these challenges can impede the progress and potential benefits of Big Data analytics in healthcare if not addressed appropriately.

The study by Batko and Ślęzak (2022) identifies several challenges associated with using Big Data analytics in healthcare, and the key challenges include managing healthcare data's sheer volume and complexity, integrating structured and unstructured data from various sources, and ensuring data quality and accuracy. Batko and Ślęzak (2022) also highlight the lack of adequate infrastructure and technical expertise to implement Big Data analytics effectively, and the diversity of data formats and the speed at which data must be processed further complicate the analysis process. Moreover, Batko and Ślęzak (2022) highlight the significant challenge of ensuring data privacy and security, given the sensitive nature of health information, and emphasize that while Big Data analytics has the potential to revolutionize healthcare, these challenges must be addressed to fully realize its benefits.

Ngesa (2023) emphasizes that data privacy and security pose significant challenges in Big Data analytics, particularly with the increasing adoption of cloud computing for data analysis. Furthermore, the risk of privacy disclosure is heightened, and transferring large volumes of data to and from the cloud can be both costly and time-consuming due to network limitations. Additionally, the lack of standardisation in data formats, along with the need for data cleaning and aggregation to reduce heterogeneity and improve analytical efficiency, further complicates the process (Guo et al., 2023). These Big Data challenges underscore the necessity for robust strategies to effectively manage and analyse Big Data in healthcare, ensuring that the benefits of technology are fully realised while maintaining patient privacy and security (Lekkala, 2023).

One of the primary challenges in Big Data analytics is managing the large volume of data sets, particularly when validating long-term predictions for diagnoses and medication. Healthcare practitioners require a comprehensive understanding of the data to deliver services effectively and accurately using big data. On the technical side, integrating and analysing the diverse types of healthcare big data to solve emerging issues remains a significant challenge (Goyal & Malviya, 2023).

Pool et al. (2024) identify multiple challenges in protecting personal health data within healthcare systems. They identify key issues, including the complexity of integrating and securing digital health records, the evolving nature of healthcare data standards, and the significant risks associated with data breaches. They further argue that the breaches underlined by the study are often facilitated by noncompliance with security policies, lack of awareness among healthcare workers, and vulnerabilities in third-party systems. Furthermore, the study highlights the difficulties in managing large volumes of data, the inadequacies in existing IT infrastructure, and the ongoing challenge of maintaining patient trust amidst increasing incidents of unauthorised data access. This, therefore, underscores the need for context-specific approaches and stronger data protection measures to address these persistent challenges effectively.

Managing large-scale datasets in healthcare poses problems related to storage, processing, and analysis (Williamson & Prybutok, 2024). Mentsiev et al. (2023) suggest that overcoming these issues requires new strategies for effective management of big data. This means that if healthcare data is properly integrated and analysed, it holds the potential to solve many persistent issues in the sector (Williamson & Prybutok, 2024).

Therefore, it is essential that we recognise the ongoing difficulties associated with big data. Additionally, one of the key challenges is that much of this data exists in outdated systems, making it difficult to integrate with newer data sources (Hariri et al., 2019). Also, the current problem has shifted from a lack of information to a lack of actionable insights for decision-making in healthcare (Tosi et al., 2024). This means that big data could remain an untapped resource without the right tools and methods (Hariri et al., 2019; Tosi et al., 2024). Sedláková et al. (2023) also emphasise that the difficulty often lies in the diverse nature of healthcare data, including structured and unstructured formats from various sources.

Implementing Information and Communications Technology (ICT) and big data in healthcare presents challenges, particularly regarding integration and the need for sufficient infrastructure, which vary depending on the country and healthcare system (Hassan et al., 2018). While some problems are context-specific, there are also common hurdles in using big data across the healthcare sector (Oleribe et al., 2019). These include the large volumes of data generated and the complications of linking older data with new systems due to differences in format. Despite these obstacles, the progress of data analytics in healthcare should not be halted. Au-Yong-Oliveira et al. (2021) argue that while technological advancements have driven much of the exploration of big data analytics, more academic research is urgently needed.

2.2 The South African Public Healthcare Sector

Maleka and Matli (2022) suggest that while telehealth has been beneficial in providing remote health services during the pandemic, the barriers outweigh the advantages.

Telehealth inhibits in-person human interaction, which is perceived as impersonal and not ideal for new patient consultations. Additionally, the lack of consistent access to internet connections and technological devices, particularly in rural areas with limited healthcare services, has been a major challenge. The socioeconomic advancement of rural livelihoods needs to be addressed to enable the full potential of telehealth. The researchers argued that while advanced technologies are available, healthcare providers may not have the necessary resources or the need to adapt to the latest technologies if their existing systems function effectively. Furthermore, the COVID-19 pandemic has further exacerbated the struggling healthcare services in rural and sparsely populated areas, with questions raised about the long-term effects on these regions.

Despite the government's introduction of the National Health Insurance (NHI) scheme to address these disparities and improve healthcare access and affordability, the success of this initiative remains uncertain, and the gap between the wealthy opting for the private health care system and the less privileged relying on the public healthcare system persists (Rammila, 2023). However, Gordon et al. (2020) argued that the NHI scheme has the potential to enhance South Africans' ability to pay for healthcare and address affordability concerns, which, in turn, has the potential to reduce socio-economic disparities in accessing healthcare in the country.

Universal Health Coverage is a critical goal for the South African government, particularly in the ongoing efforts to implement the NHI system and recover from the COVID-19 pandemic (Morar, 2024). Achieving Universal Health Coverage requires a comprehensive approach to health system reform, focusing on strengthening governance, leadership, human resources, financing, and service delivery (Schneider et al., 2023). The COVID-19 pandemic has exposed the strengths and weaknesses of South Africa's health system, emphasising the need for resilience and adaptability in the face of public health emergencies (Chabrol & David, 2023).

Several priority areas for health policy and systems research are being put in place ,and they include enhancing leadership capacity, ensuring accountability, improving the use of health information for decision-making, and fostering a culture of continuous quality

improvement (South African Medical Research Council (SAMRC), 2020). This suggests that effective health system governance and strategic investment in health infrastructure and workforce are essential to achieving equitable access to quality health services for all South Africans. As the country progresses towards implementing the NHI, it is crucial to continuously engage stakeholders, utilise evidence-based decision-making, and adapt strategies to address emerging health challenges and disparities (Michel et al., 2020).

2.2.1 Challenges faced by the South African public health sector

Maleka and Matli (2022) suggest that while telehealth has been beneficial in providing remote health services during the pandemic, the barriers outweigh the advantages. Telehealth inhibits in-person human interaction, which is perceived as impersonal and not ideal for new patient consultations. Additionally, the lack of consistent access to internet connections and technological devices, particularly in rural areas with limited healthcare services, has been a major challenge. The socioeconomic advancement of rural livelihoods needs to be addressed to enable the full potential of telehealth. The researchers argued that while advanced technologies are available, healthcare providers may not have the resources or need to adapt to the latest technologies if their systems function effectively. Furthermore, the COVID-19 pandemic has further exacerbated the struggling healthcare services in rural and sparsely populated areas, with questions raised about the long-term effects on these regions.

The South African public health sector faces significant challenges, particularly in rural areas. Key issues include socio-economic barriers such as high unemployment and poverty levels, which severely limit access to healthcare services. Many rural residents struggle to reach health facilities due to inadequate infrastructure and transportation options. Additionally, the healthcare system is often understaffed and lacks essential resources, leading to poor health outcomes. Willie and Maqbool (2023) emphasize that, despite providing free primary healthcare, many families still cannot afford necessary treatments, thereby exacerbating health disparities in these communities.

Moreover, the quality of care in public health facilities is a significant concern. Patients often encounter long distances to health centres, insufficient medication, and unhelpful attitudes from healthcare staff. The healthcare system's focus on biomedicine overlooks the social determinants of health, such as cultural expectations and community dynamics. As a result, rural populations remain vulnerable to health issues, with inadequate monitoring and evaluation of health services further complicating the situation. Willie and Maqbool (2023) suggested that addressing these systemic issues is crucial for improving access and quality of care in South Africa's public health sector.

Witter et al. (2024) highlighted resource constraints, such as inadequate funding, staffing shortages, and the limited availability of medical equipment and supplies, as other challenges in the health sector in South Africa. The system also grapples with inefficiencies compounded by bureaucratic delays, poor management practices, and corruption. These issues contribute to a strained healthcare delivery system that struggles to meet the needs of a growing and diverse population, especially in under-resourced areas.

Moreover, the public health sector faces significant challenges related to inequality and accessibility. Despite efforts to improve healthcare access, disparities persist, with rural and marginalised communities often receiving lower-quality care (Witter et al., 2024). The burden of diseases like HIV/AIDS, tuberculosis, and non-communicable diseases adds further pressure on the system. Additionally, the sector is struggling to keep up with the increasing demand for mental health services, which are critically under-resourced. These challenges underscore the need for comprehensive improvements to enhance the public health infrastructure and enhance service delivery nationwide.

Furthermore, Achieng and Ogundaini (2024) highlight that the implementation of BDA in South Africa faces several challenges, which include inadequate digital and analytical skills among healthcare professionals, unreliable electricity supply, and concerns about cybersecurity and data privacy. There is a pressing need for investments in ICT infrastructure and resources, as well as targeted training programs to upskill healthcare

workers in using BDA tools and techniques. Addressing these barriers requires a multi-stakeholder approach, involving governments, academia, and IT practitioners, to create an enabling environment for BDA (Achieng & Ogundaini, 2024). This approach can help mitigate the complexities and challenges currently hindering the effective use of BDA in the healthcare sector, thereby enhancing the overall public health surveillance and response capabilities in South Africa.

2.2.2 Technology advancement in the SA health sector

In recent years, technology has significantly transformed the public healthcare sector in South Africa, resulting in improved patient care and more efficient medical practices. One of the most impactful changes has been the move from paper-based records to electronic health records (Katurura & Cilliers, 2018). This change has made it easier for healthcare practitioners to access and share patient information, ensuring everyone involved in a patient's care is on the same page. With electronic health records, healthcare providers can quickly review a patient's medical history, make informed decisions, and coordinate care more effectively, ultimately leading to better health outcomes (Adeniyi et al., 2024).

Additionally, mobile health (mHealth) apps are transforming the way South Africans manage their health (National Department of Health, 2019). These apps offer various services, from booking doctors' appointments to reminding patients to take their medication (Sehlabo and Vambe, 2024). With a high percentage of smartphone users in South Africa, mHealth makes healthcare more accessible and convenient (Mahmood et al., 2023). These apps are also utilized in public health campaigns, helping to spread awareness about important issues such as HIV prevention and treatment, which are crucial in improving the nation's overall health.

In recent years, there has been a significant increase in the use of AI and ML tools. Many advanced hospitals, especially those in the private sector, have been adopting data-driven systems powered by Big Data (Moodley, 2023), due to the increasing use of various processes in healthcare. This has led to the adoption of more advanced technologies, such as Big Data analytics, which seek to analyze complex data and

produce faster and more efficient outcomes. Big data analytics tools aim to provide more insights by utilizing predictive, prescriptive, and diagnostic models to generate meaningful information. However, the increase in using these models and techniques has been primarily visible in private and more advanced hospitals, with very few instances in public hospitals, due to several reasons, including a lack of skilled personnel, resources, and funds (Nene & Hewitt, 2023).

Despite the numerous advantages of ICT in healthcare, several challenges persist, especially in rural areas where technological infrastructure is lacking (Nyasulu & Chawinga, 2018). Another issue, as highlighted by Moodley and James (2024), is the complexity of integrating technology into healthcare settings, which can often lead to unintended consequences. Although many hospitals in the public sector have adopted ICT to enhance healthcare processes, the expected benefits, such as workflow and record management improvements, have been slow to materialize (Motsi & Chimbo, 2023).

Information, communication, and technology hold significant potential to transform healthcare service delivery (Malungana & Motsi, 2024). Despite these benefits, Maphumulo and Bhengu (2019) argue that current technologies fail to meet patient needs due to poor quality, limited usability, and a lack of innovation. This shortfall is often attributed to inadequate awareness and expertise among medical and ICT practitioners. While ICT has the potential to improve the quality and accessibility of healthcare, progress has been slow, partly due to limited information access or unreliable data (Jinabhai et al., 2021).

Wang et al. (2018) argue that healthcare has not seen a return on investment in data analytics, primarily due to a lack of confidence in these systems. The lack of awareness regarding ICT's role in healthcare can stifle innovation and hinder overall adoption (Moodley & James, 2024). However, Mbunge et al. (2022) emphasise that the increasing complexity of healthcare operations, driven by the overwhelming volume of data, has necessitated ICT innovations to address challenges in medical diagnostics and treatment.

Healthcare is a data-intensive industry, and while data enables better health services, caution that the multitude of data sources can limit accessibility, quality, and utility (Makeleni & Cilliers, 2021). ICT solutions, such as electronic health records, generate vast amounts of health data, which, according to Makeleni & Cilliers (2021), are crucial for managing healthcare's complexities. Despite the potential of big data to improve service delivery, Bruintjies and Njenga (2024) noted that the implementation has often fallen short, creating a gap between potential and practice that continues to hinder the growth and quality of healthcare.

Finally, Obasa and Palk (2023) discuss that healthcare data is sourced from various origins, including web and social media, machine-to-machine interactions, and biometric data, which can come from internal and external sources. While big data has the potential to provide valuable insights for improving healthcare service delivery in the public sector, challenges in utilising this data effectively continue to impede progress (Chibuike et al., 2024).

2.2.3 Current tools and techniques in the SA public health sector

Big Data analytics is increasingly applied for integrated infectious disease surveillance. This involves analysing large volumes of health records to identify patterns and trends, which is crucial for early detection and monitoring of disease outbreaks (Achieng & Ogundaini, 2024). Despite its potential, the application of these analytics faces challenges such as inadequate regulatory frameworks and disparities in digital health infrastructure (Achieng & Ogundaini, 2024)

Moreover, frameworks have been proposed to guide the selection of Big Data analytics tools tailored for the South African healthcare sector (Iyamu, 2020). The framework aimed to assist healthcare practitioners in understanding and utilising big data effectively, addressing the traditional reliance on outdated data processing methods. Nakashololo & Iyamu (2023) developed a model that simplifies and supports the process of selecting and using Big Data Analytics (BDA) tools in organisations. Their study identified five key

factors that significantly influence this process: organisational needs, the approach taken (whether top-down or bottom-up), stakeholder involvement, the perceived value of BDA, and the organisation's structure. These factors collectively guide how organisations choose and implement BDA tools effectively.

Further studies have been conducted; for instance, Mbunge et al. (2022) found that South Africa adopted various digital technologies to deliver healthcare services during the COVID-19 pandemic. These included SMS-based solutions, mobile health apps, telemedicine, telehealth, WhatsApp systems, artificial intelligence, chatbots, and robotics. These technologies were used for disease screening, surveillance, treatment compliance, and public awareness. Additionally, virtual healthcare services such as teleconsultation, e-prescription, and specialised fields like teledermatology and teleoncology were implemented. Despite their benefits, these technologies faced challenges, including infrastructural, technological, organisational, financial, policy, regulatory, and cultural barriers.

Table 2.2 Technological Systems used by Hospitals in South Africa (Source: Cline & Luiz, 2013; Netcare, 2022)

Hospital	**Data analysis methods and techniques**	**Description**
Inkosi Albert Luthuli Central Hospital (IALCH) and Sebokeng Hospital	Health Information Systems (HIS):	They have implemented a comprehensive HIS, which includes modules for patient management, clinical documentation, laboratory information systems, and more. These systems help streamline operations

		and improve data accuracy.
Groote Schuur, Khayelitsha Hospital, Chris Hani Baragwanath, and Steve Biko	Electronic Medical Records (EMRs)	These are widely used to store patient data, making tracking patient history and treatments easier. They use EMRs as part of their hospital information systems (HIS).
Netcare Greenacres Hospital	CareOn system	Digital technology to improve health and care for patients. Records all clinically relevant data on each patient's unique electronic medical record
All district hospitals	District Health Management Information System (DHMIS),	A system for deriving a combination of health statistics from various sources, mainly from the routine information system used in the public sector to track health services delivery in sub-districts, districts, provinces and nationally

2.2.4 Big Data Analytics in the South African public healthcare sector

Welbotha and Van Den Berg (2024) noted that Big Data analytics in the South African public sector is essential for boosting operational efficiency and achieving a competitive edge. Van der Pol et al. (2021) identify shortcomings in current mobile health (mHealth) monitoring systems for patients with chronic disease in developing countries, highlighting their focus on information dissemination and healthcare worker monitoring over patient empowerment. Researchers identified key elements, including mobile technology, patient beliefs, social behavior, culture, environment, perceived behavioral control, and behavioral intention. Still, the study ultimately formulates a new model for a persuasive mHealth monitoring system to enhance patient participation and improve healthcare outcomes in these regions.

In addition, Kalema and Musoma (2019) and Kgasi et al. (2023) alluded to the fact that traditional technologies are increasingly inadequate for handling the velocity, variety, and volume of data generated in the public sector. The findings of this study provide a framework for public sector enterprises to effectively utilise Big Data analytics, helping them determine what actions to take, when to take them, and how to implement them effectively. This contributes to the sparse literature on Big Data analytics in the public sector. It offers practical guidance for management to make informed decisions, enhancing their capacity to serve the public and optimise their operations. Despite the benefits, however, the success of such systems depends heavily on the development of robust ICT regulatory and data governance frameworks that ensure data security, privacy, and proper management (Achieng & Ogundaini, 2024).

2.3 Synthesis of the Literature and Research Gap

The global literature on Big Data analytics in healthcare highlights the availability of powerful tools, such as Hadoop, Spark, and TensorFlow, as well as various techniques, including supervised and unsupervised machine learning algorithms. These resources offer significant potential benefits, including early disease detection, personalized care, and improved operational efficiency (Setyati et al., 2024; Badr et al., 2024; Khosravi et al., 2024).

However, several persistent challenges hinder the full realization of these benefits. Issues such as data integration difficulties, privacy and security concerns, high resource demands, and the limitations of traditional systems frequently arise (Walker et al., 2023; Muhunzi et al., 2023; Ngesa, 2023). To manage these complexities, structured methodologies like CRISP-DM are commonly used in high-resource Western contexts (Shearer, 2000). However, adaptations of these methodologies for healthcare settings in developing countries remain limited (Hosseini et al., 2024).

In the context of South African public healthcare, research consistently highlights several unique structural and contextual barriers. These include unreliable infrastructure (such as electricity and internet access), acute shortages of digital and analytical skills, fragmented data systems, underfunding, rural-urban disparities, and policy pressures stemming from the rollout of the National Health Insurance (NHI) (Achieng & Ogundaini, 2024; Witter et al., 2024; Nene & Hewitt, 2023).

While some studies focus on basic digital tools (like electronic health records, mHealth, health information systems, and district health management information systems) and emerging big data analytics (BDA) applications (such as infectious disease surveillance), the emphasis largely remains on identifying barriers or proposing high-level frameworks for tool selection. There is a lack of development of fully contextualized, implementable BDA models specifically tailored to improve service delivery.

The need for such a tailored Big Data analytics model is critical in addressing the persistent challenges faced by the healthcare system and in advancing healthcare outcomes (Nakashololo & Iyamu, 2023; Kgasi et al., 2023; Welbotha & Van Den Berg, 2024). Importantly, there is limited evidence of Big Data analytics models specifically designed for South African public hospitals that address the full lifecycle of analytics. This includes understanding hospital-specific business needs (such as reducing waiting times, optimizing scarce resources, and aligning with NHI objectives) and deploying solutions in constrained environments (Iyamu & Mgudlwa, 2021; Achieng & Ogundaini, 2024).

Although the CRISP-DM framework offered a robust and industry-standard process for Big Data analytics project development, its adaptation to incorporate the realities of South African public healthcare, such as fragmented and heterogeneous data sources, limited computational resources, compliance with the Protection of Personal Information Act (POPIA), and goals of equity driven by the NHI, remains underexplored in the literature.

This study sought to address this gap by developing a tailored Big Data analytics model to improve service delivery in South African public hospitals. This model was guided by an adapted CRISP-DM framework that explicitly considers these contextual constraints and priorities.

2.4 Related Work

Numerous studies have been conducted to explore the application of Big Data analytics and Machine Learning in healthcare, yielding promising results that motivate further research in this domain. Rajkomar et al. (2018) revealed the use of ML to predict patient results with the exploration of predictive analytics in hospital management. Furthermore, the utilisation of deep learning to predict disease onsets shows the potential for early intervention strategies. The studies lay the groundwork for implementing comparable techniques to enhance service delivery in South African public hospitals.

The study by Badawy et al. (2023) utilized machine learning algorithms to predict patient outcomes. Furthermore, their findings demonstrated the potential of ML in predicting

patient results, highlighting significant improvements in patient management and outcomes. The studies have been limited by the scope of the data and the diversity of patient populations, which may affect the generalizability of the results. Lastly, the study suggested further exploration of predictive analytics across different hospital settings and patient demographics to validate and enhance predictive models.

Bowe et al. (2022) illustrated how Machine Learning (ML) could identify patients at risk of severe complications, underscoring the importance of advanced analytics in healthcare. The study applied machine learning techniques to identify patients at risk of severe complications. Furthermore, the study emphasized the significance of advanced analytics in predicting and managing patient risk, thereby enhancing patient care and outcomes. The study's limitations include the variability in data quality and the need for extensive data preprocessing to ensure accuracy.

Precision public health is an emerging field driven by technological advancements, enabling more detailed descriptions and analyses of individuals and population groups. Its primary objective is to enhance the health outcomes of all population groups, ensuring that those in need of healthcare services receive quality care (Dolley, 2018). As a subset of big data, precision public health also focuses on disease prevention, reducing health disparities, and promoting quality healthcare by utilising evolving methods and cutting-edge technologies to achieve these goals (Dolley, 2018). The study explores how big data can be leveraged to enable more granular prediction and understanding of public health risks as well as to customize treatments for more specific and homogeneous subpopulations. The study indicated that Big Data is used to track disease spread using mobile phone data, predict future risks by using significant data sources like social media and the internet, and

Rostamzadeh et al. (2020) and Doutreligne et al. (2023) developed a data-driven framework that utilized electronic health records to identify potential bottlenecks in the care process, thereby enhancing workflow efficiency. The researchers developed a data-driven framework leveraging electronic health records to identify bottlenecks in the care

process. Furthermore, their findings improved workflow efficiency by pinpointing and addressing potential delays and inefficiencies in patient care. The study limitations include the reliance on electronic health record (EHR) data, which may vary in completeness and accuracy. The study recommended expanding the framework to include more diverse data sources and implementing continuous monitoring to further streamline healthcare processes.

A study by Turgay and Özçelik (2023) utilised data-driven techniques and Big Data analytics (BDA) to improve healthcare processes. Their study demonstrated that leveraging electronic health records (EHRs) and advanced analytics can identify inefficiencies in healthcare delivery and suggest improvements. The study faced challenges related to data privacy and security, as well as the integration of heterogeneous data sources. Moreover, the study aims to enhance the efficiency of healthcare delivery by identifying and addressing inefficiencies in the system.

Gupta (2023) examines the important role of Big Data and IoT in enhancing smart healthcare. He highlights the deficiencies of standard healthcare systems, particularly during the pandemic, and encourages the use of Big Data to improve healthcare access and reduce costs. Furthermore, the study discusses the sources of healthcare Big Data, its applications in diagnosis, prevention, personalised treatment, research, and its challenges, such as data privacy and security. He concludes that Big Data and IoT can significantly advance healthcare by providing actionable insights, decision-making,and personalised care.

Darrell and Estrin (2024) and Sarker (2024) demonstrated the use of predictive modelling techniques to forecast patient demand and optimise staff allocation, ultimately reducing wait times and enhancing resource utilisation. The study faced limitations in data accuracy and the real-time application of predictive models in a dynamic hospital environment. Lastly, the study suggested ongoing refinement of predictive models and incorporation of real-time data for more accurate and dynamic staff allocation.

2.5 Chapter Summary

This chapter reviews the literature on Big Data in the healthcare sector, with a focus on its growing importance in South African public healthcare. It covered the concept of Big Data, its benefits and challenges, including the tools and techniques for analytics and the factors influencing its implementation. The chapter highlighted issues faced by the South African healthcare system. It also assessed current data analysis methods and technologies in South African hospitals, compared relevant literature, and identified gaps and differences. Informed by this chapter, the next chapter will outline the methodology adopted for developing a data analytics model for the South African public health sector.

CHAPTER THREE: RESEARCH METHODOLOGY

This chapter outlines the research methodology employed in this study, focusing on the CRISP-DM process framework, which was adopted due to its suitability for data science projects. Therefore, the chapter is structured as follows: Section 3.1 presents the approach adopted for this study, followed by Section 3.2, which highlights the research paradigm. Section 3.2 outlines the research strategy followed. Then, Section 3.4 outlines the research methodology adopted in this study, followed by a discussion of the CRISP-DM process, which serves as the adopted flow framework for data science. This process is further explored in Section 3.5, where its six phases are discussed in subsections 3.5.1 to 3.5.6. The data collection and the analysis of the study are detailed in Section 3.6, while Section 3.7 explains the reliability and validity of the research. Finally, Section 3.8 concludes the chapter.

3.1 Research Approach

This study falls under a quantitative research approach, as the nature of our work involves working with numerical analysis and data interpretation in the healthcare domain, doing predictions using machine learning. Quantitative approaches are suitable for this research because they enable the interpretation of findings using percentages, graphs, and advanced data science tools, as explained by Kotronoulas (2023). This approach aligns with the study's goal of modeling and predicting outcomes based on quantifiable factors and statistical patterns. The availability of a big dataset with structured numerical and categorical variables enables thorough analysis using statistical techniques and machine learning algorithms.

Unlike qualitative approaches, which are better suited for exploratory studies with limited data or mixed methods, which may introduce complexity while offering little value in this context, the quantitative approach is well-aligned with the study's technical and analytical objectives. By focusing on quantifiable results and applying statistical validation, this approach ensures that the study objectives are met, enabling the development of a robust, interpretable, and high-performing model (Javed et al., 2024). This approach not

only promotes scientific research into healthcare data but also facilitates the application of findings in real-world health-care situations.

3.2 Research Paradigm

According to Creswell and Creswell (2018), the term "paradigm" refers to a research culture characterised by a shared set of beliefs, values, and assumptions held by a community of researchers regarding how research should be conducted. Johnson and Christensen (2020) define a paradigm in educational research as a framework that shapes the study and interpretation of knowledge, as well as the motivations and goals behind the research. Understanding the research paradigm gives researchers a clearer perspective on the research process (Creswell & Creswell, 2018).

This study adopted the positivist paradigm Cohen et al. (2018) because it is a scientific investigation that relies on statistical analysis of the data. The positivist research paradigm is particularly suitable for data science and computer science, where empirical and quantifiable data guide scientific investigation.

Since the research utilizes structured data from the Final_healthcare_big_data.csv file and applies statistical and machine learning tools to identify patterns typical of data-driven research in this field, a positivist approach is justified as it supports the creation of a predictive model that can be validated and replicated.
Additionally, this paradigm ensures that the study maintains the scientific rigour required in data science, and it blends well with structured phases of the CRISP-DM framework, which was adopted in this work.

3.3 Research Strategy

A research strategy outlines how a researcher would conduct their study, from the initial concept to the final results (Malhotra, 2017). This study adopts an experimental research technique, which entails systematically “manipulating independent variables under controlled conditions to determine their cause-and-effect relationship with dependent

variables” (Creswell, 2014; Ishtiaq, 2019). This is suitable for the fields of data science and computer science, focusing on developing and evaluating predictive models.

In our case, this aligns very well as the research uses independent variables such as patient demographics and hospital data (e.g., Satisfaction_Rating, Age, Log_Wait_Time, Resource_Bed_Interaction), a dependent variable (Outcome), controlled conditions (the dataset with standardised preprocessing and balancing via SMOTEENN e.g., scaling and imputation) ensured the a controlled experimental environment with the CRISP-DM, with the goal to have a model that predicts healthcare patient outcome in the health sector.

3.4 Research Methodology

Research methodology in computer science is “*an action plan, strategy, process, or design laying behind the choice of and methods and linking the choice of methods used*” (Alturki et al., 2025). Although there are other distinctions in the research methodology, the most common classification of research methodologies can be categorised broadly as qualitative, quantitative, and mixed methodology. Qualitative methodology concentrates on examining non-numerical data to extract profound contextual insights, whereas quantitative methodology prioritizes numerical data analysis and statistical interpretation to discern patterns and evaluate hypotheses (Oranga & Matere, 2023). Mixed methodology combines both approaches to capitalise on their complementary strengths, resulting in a comprehensive understanding of complex phenomena (Ahmed et al., 2024).

To achieve the above methodology, research methods are used. Research methods are defined as “*the techniques or procedures used to gather and analyse data related to a research question or hypothesis”* and can be either qualitative or quantitative methods (University of Newcastle Library, 2023). Based on this research aim, research questions, and objectives, this study adopted a quantitative research methodology and used statistical analysis, feature engineering, machine learning classification techniques, and

performance evaluation metrics. Quantitative methods were used to analyse and derive insights from the dataset, aligning with the study's quantitative methodology and the CRISP-DM process, as explained in detail in section 3.6.

3.5 Process Framework

In the context and field of data science in which our research falls into, there exist several research process frameworks that can be used in the field, such as Sample, Explore, Modify, Model, Assess(SEMMA) (Corrales et al., 2015), Knowledge Discovery in Databases(KDD) (Maimon & Rokach, 2005), Cross-Industry Standard Process for Data Mining (CRISP-DM) (Lind & Glas, 2022), among many others. Given the nature of our work, this study adopted the CRISP-DM process framework. CRISP-DM was preferred over other data mining approaches because of its comprehensive, iterative, and business-centered framework, which aligns with the objectives of this study (Lind & Glas,2022). CRISP-DM's compatibility with the quantitative nature of this study strengthens its applicability. The framework's emphasis on the evaluation and deployment phases encourages the use of statistical tests and machine learning algorithms, which are essential for quantitative analysis. Unlike SEMMA, which is less flexible for exploratory data tasks, and KDD, which focuses more on knowledge discovery than model deployment, CRISP-DM offers a balanced approach that enables the development, testing, and practical application of the healthcare outcomes classifier. This option ensures that the research follows a standardised procedure, enabling collaboration, documentation, and future replication, all of which are required for the study and potential real-world implementation in a healthcare environment. Talaviya (2023) defines CRISP-DM as "*a widely used methodology for guiding data mining projects".* It provides a phased approach to data mining, ensuring that projects are well-defined and managed, and deliver valuable results. Cross-Industry Standard Process for Data Mining has six-phase namely: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment, which offers a repeatable and well-understood workflow that meets the technical requirements of this study (Tripathi et al., 2021). This process framework provides for an iterative approach, allowing the researchers to revisit

previous phases (for example, data preparation) depending on insights acquired during modeling or experiments, ensuring that the model is continuously improved. Given the complexities of healthcare data, including numerical patient metrics and categorical variables, CRISP-DM's emphasis on thorough data understanding and preparation phases facilitating the identification and management of potential biases or anomalies, which is crucial for developing a reliable classifier. This structured but adaptable framework ensures that the research process is systematic and reproducible, increasing the credibility of the findings.

To achieve the research goals, this work adopted the six phases of CRISP-DM methodology, namely business (problem) understanding, health data understanding, data preparation, modeling, evaluation, and deployment, illustrated in Figure 3.1 below.

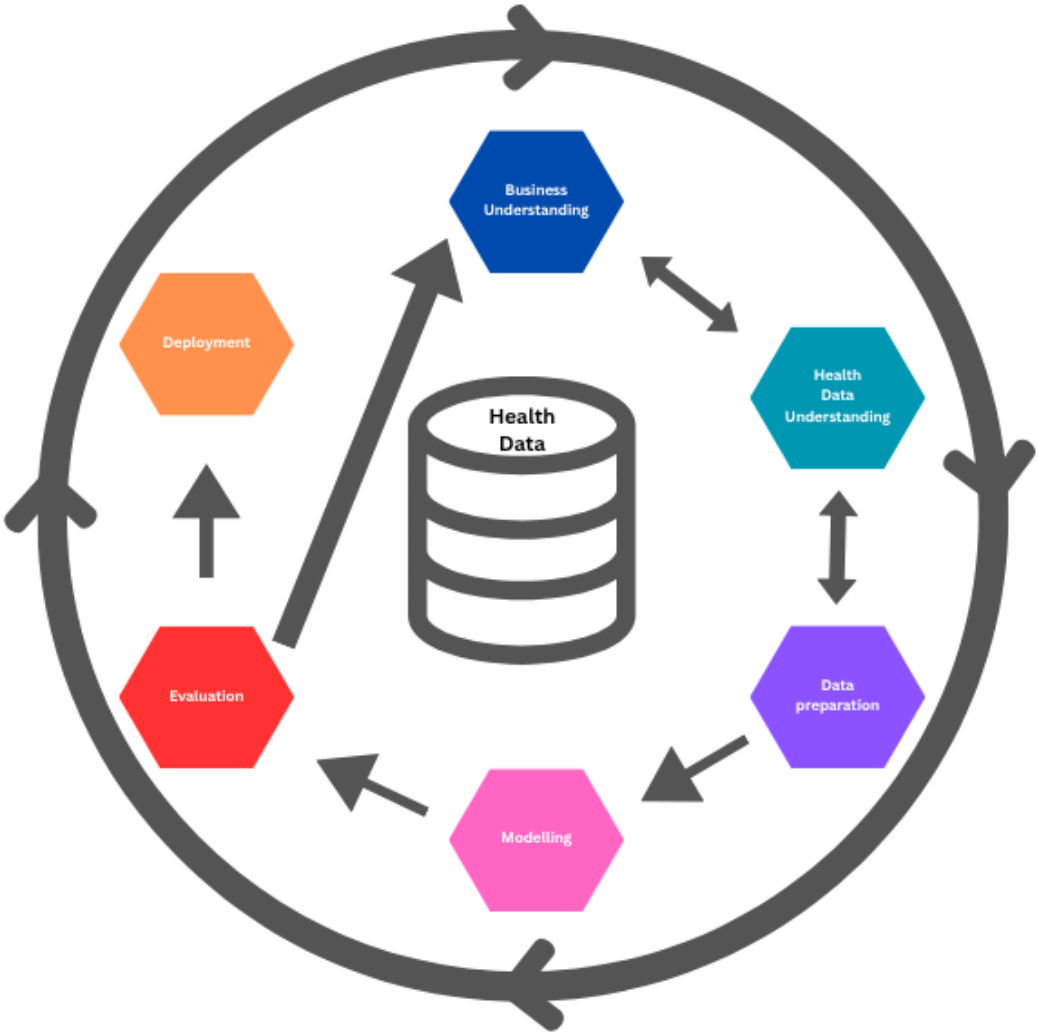


Figure 3.1 The CRISP-DM methodology design, adopted from "CRISP-DM for Data Science Teams: 5 Actions to Consider" by Saltz, J. (2020, May 29).

The following sub-sections detail how the phases were used, highlighting at each stage the methods and or software used to achieve each phase

3.5.1.1 Problem Understanding

This phase focuses on assessing the situation, defining the problem to be solved, and determining data mining goals (Sharma & Osei-Bryson, 2009).

To understand the research problem and identify a research gap in the health sector, the work used a literature review method (Auraria Library, 2023), a qualitative approach. The literature employed was within a 5-year range to identify a research gap. Using the latest

literature also helped us to align our work with the latest scholarly discourse and further enables us to base our studies on relevant and proven findings. The researcher focused on peer-reviewed papers from well-recognized journals on Scopus, Google Scholar, ScienceDirect, South African Medical Journal, and PubMed. Keywords such as healthcare, Big Data analytics, Machine learning in Healthcare, Service Delivery, South African Hospitals, and Predictive Analytics were used to narrow our search and obtain correct and relevant papers for our study.

Furthermore, the researcher employed the document inspection method (Office of Health Standards Compliance, 2024) to verify the problems identified in the literature by examining the South African government's health website and reports. This helped in contextualizing the research problem.

It was from these literature review papers, documents, and reports that the research gap was identified, and our research problem statement, as highlighted in Section 1.3, was built on. This stage helped us establish the research aim, objective, and questions for this study, as highlighted in Sections 1.4.1, 1.4.2, and 1.5.1, respectively.

3.5.1.2 Data Understanding

This phase involved collecting initial data, describing it, exploring it, and verifying its quality to understand the data's characteristics, identify potential problems, and assess its suitability for data mining, as explained by Jung (2024).

In our context, researchers examine, comprehend, and analyze synthetic data representing patient outcomes, resource utilization, and service delivery. The datasets were synthetically generated using Python code, which is included in the appendix, and comprised patient care records, resource utilisation statistics, and operational efficiency measurements. The datasets were processed in CSV (Comma-Separated Value) format, which is suitable for model development and evaluation and does not require conversion. The experiment began by generating synthetic datasets that resembled real-world data. Following this, the data underwent preprocessing, including handling missing values,

removing duplicates, transforming skewed distributions, and conducting a correlation analysis to identify the features most strongly associated with the target variable for feature selection. Additionally, features with low variance were removed. Idrees et al. (2019) emphasise the significance of historical data in developing prediction models, which informed our investigation. Furthermore, these insights, illustrated in updated Figures 4.5 - 4.7 (density plots, correlation heatmaps with interpretive notes), confirmed the synthetic data's plausibility, underscored the need for imbalance-aware techniques and metrics (F1-score, Precision-Recall AUC), and established a solid, clean foundation for subsequent data preparation and modelling. An exploratory data analysis (EDA) method, as described by Monsen (2024), was employed to understand the distributions, trends, and relationships within the data. As Komorowski et al. (2016) and IBM (2020) noted, EDA is significant as it helps detect data quality problems and comprehend distributions.

Ultimately, the data understanding phases provided crucial insights into the relevance, structure, and quality of the datasets used in this study. Through the EDA, key issues such as outliers, duplicates, and missing values were identified, and inconsistencies were noted for remediation in the next phase. The finding laid a strong foundation for the data preparation and model development phase, ensuring that subsequent analyses are built on clean and meaningful data.

3.5.1.3 Data Preparation

After realizing the missing values, outliers, and inconsistencies as stated above, there was a need to perform data preparation, a phase that helps prepare the data for analysis and modeling by cleaning, transforming, and formatting it, as highlighted by Subasi (2020a). This stage is crucial in model building, as it ensures that the data used for training is clean, accurate, and consistent, thereby enhancing model performance and yielding more reliable insights (Omoseebi et al., 2025).

As stated by Rizzoli, (2022), the performance of a model is intrinsically linked to the quality and pertinence of the data trained on; therefore, inadequate preparation can lead to

distorted models by inaccuracies, inconsistencies, or absent values, culminating in erroneous predictions and unreliable outcomes. Furthermore, Luengo et al. (2020) assert that adequate data preparation guarantees robust and trustworthy subsequent analysis.

Even though there exist several tools that can be used for cleaning, transforming, and formatting data, such as R (Kopra et al., 2024) and SQL(Ogbogu, 2022), among others, in this work we used Python (Subasi, 2020), because Python libraries make data cleaning more effective and systemically by offering robust capabilities to tackle everyday cleaning tasks. Furthermore, data cleaning with Python enhances data quality by eliminating mistakes and increasing data accuracy and completeness (Tiwari, 2021).

Data cleaning by handling missing values, removing duplicates, dropping redundant features, correcting data types, handling outliers, standardising and normalising by dropping redundant features was done using Python features, supported by the core libraries, namely Pandas and Numpy.

a) Data transformation is performed by mapping using Pandas series, feature engineering, encoding categorical variables, and separating the target feature.
b) Data formatting by standardising formats, string manipulation, data aggregation, and data export

Normalization and standardization were performed to ensure uniformity across variables, as recommended by Raju et al. (2020). This phase was critical to ensuring the quality of input data, aligning with best practices in machine learning and data analytics. The above stages will be detailed in the next chapter in Sections 4.2.1.3 to 4.2.1.11

After data preprocessing and using Python tools, the data was split into training, test ,and validation sets to facilitate model training, hyperparameter tuning, and final evaluation. During the trial experiment, the researchers started with a 60-20-20 split, allocating 60% of the data to the training set, 20% to the validation set, and 20% to the test set, as supported by Muraina (2022) as an ideal dataset split ratio for machine learning algorithms.

The training set was used to fit the models, the validation set was used for early stopping (e.g., in LightGBM, XGBoost, and CatBoost) and performance monitoring, and the test set was reserved for the models' final evaluation, as discussed in Section 4.2.1.11. For the final model, the study employed an 80:20 training-to-testing split, as using the aforementioned split set compromised accuracy. Therefore, the decision was made to adhere to the 80:20 split, which was achieved by utilizing cross-validation, as discussed in Section 4.2.1.11. This splitting technique ensured that the models were trained on a large percentage of the data, efficiently tuned with the validation set, and then assessed on an independent test set to produce an unbiased performance assessment.

3.5.1.4 Modelling

After the data preparation stage, the data were ready to be used in the model building. The modelling phase focuses on building prediction models using Python-based machine learning algorithms.

In the beginning, started with training models, such as random forest, Logistic regression, Support vector machine (SVM) (Cervantes et al., 2020), Random Forest (Liu et al., 2022), Extreme Gradient Boosting (XGBoost) (Wiens et al., 2025), Light Gradient Boosting Machine LightGBM (Soomro et al., 2024), and Categorical Boosting (CatBoost) (Shaikh,2023). SVM and logistic regression were dropped because they took over a day to complete or produce results, and they underperformed due to the nature of our classification problem. It was not ideal to use them.

In developing the final model, Random Forest, XGBoost, LightGBM, and CatBoost were used because they are robust ensemble machine learning techniques, predominantly used for classification and regression problems. The researcher developed and optimised machine learning models using ensemble algorithms (Random Forest, XGBoost, LightGBM, CatBoost).

The trial models were trained using a dataset split into a 60% train set, a 20% test set, and a 20% evaluation set. The training set was used to train the models, while the test set was used to assess the models' best-performing algorithm. However, the final model was trained using the 80:20 split, which is discussed in detail in the next chapter. Furthermore, the testing set was used for model tuning and hyperparameter optimization to find the most selective and effective model. Finally, the evaluation set was used to assess the generalization ability and determine how the model performs best in unseen data within the experimental context. This ensures that the final model was accurate on the training data and robust and reliable. This approach ensured a rigorous and unbiased assessment of their predictive capabilities in our work.

After the successful development and selection of the model, the next step was to evaluate it.

3.5.1.5 Evaluation

The evaluation phase assessed the efficiency, effectiveness, and overall performance of the model.

a) Effectiveness of the model

A confusion matric was used as a primary evaluation tool to measure the model's classification accuracy by analysing on class 0: true positives (Improved), false negatives(Improved as Unchanged), false negatives(Improved as Worsened), class 1: false positives(Unchanged as Improved), true positives (Unchanged), false negatives(Unchanged as Worsened); class 2: false positives(Worsened as Improved), false positives (Worsened as Unchanged), true positives (Worsened). Key performance metrics, including accuracy, precision, recall, and F1 score were computed from these parameters.

b) The efficiency of the model was assessed.

The model's efficiency was assessed by measuring the training and overall computation time, which also provided insights into the CPU performance. Experimental tests were conducted to analyse the effectiveness of the model. The results of the experiments ensured that fine-tuning of the model design and performance was done to meet the requirements for a secure, fast, and efficient predictive health care model. Furthermore, in a case where the model was to be deployed in the healthcare industry, the same outcomes are expected.

c) Expert evaluators

Expert evaluators were provided with a simplified model abstraction to assess the model. The abstraction outlined key inputs, output categories, and decision logic, enabling experts to evaluate the model's relevance, interpretability, and alignment to the healthcare domain. This approach allowed meaningful feedback on the model's structure, fairness, and practical applicability within the study context. Additionally, a user interface was developed to enhance the user experience.

3.5.1.6 Deployment

In the context of our work, the researchers did not deploy the model in the real world as it was outside the scope of the research. Deploying the model in the real world will be associated with risks, taking into account the sensitivity of the health sector and the limited time available due to ethical issues; hence, an experimental approach was opted for. While the study is experimental and does not involve a real-world deployment, testing the model's performance was based on experimental results obtained during the experimental tests when the model was run.

3.6 Data Collection and Analysis

This section focuses on data collection and analysis designed to ensure rigour, credibility, and alignment with the research questions, employing a quantitative approach to predict healthcare outcomes. The dataset, referred to as 'Final_healthcare_big_data.csv', contained 150,000 records from a simulated healthcare dataset, which includes variables

such as daily patient inflow, emergency response time, staff count, bed occupancy rate, visit frequency, age, resource utilisation, treatment outcome, and satisfaction rate.

The sampling method utilised a purposive sampling technique, resulting in a comprehensive dataset that offers a balanced representation of the target classes (Improved, Unchanged, Worsened) after balancing. This approach ensures sufficient variability for model training. The instrumentation involved Python-based data science tools; Pandas was used for data manipulation, Scikit-learn for preprocessing and modeling, and Imbalanced-learn for addressing class imbalances using SMOTEENN.

Adopting the CRISP-DM framework, the analytical process begins with data understanding, which includes assessing missing values, duplicates, and data types. This phase progresses to data preparation, feature engineering, and variable preprocessing. For the modelling phase, a StackingClassifier integrates RandomForest, XGBoost, LightGBM, and CatBoost, optimised with class weights to correct initial imbalances. The evaluation process incorporates classification reports, threshold adjustments, and SHAP analysis to enhance the interpretability of the results.

Software tools such as VS Code, Matplotlib, and Seaborn aid in visualisation and validation, while cross-validation ensures model robustness. Validity is maintained through feature selection and statistical evaluation, whereas reliability is enhanced by the iterative CRISP-DM framework and the use of established libraries, which helped to reduce human error and ensure consistency.

3.7 Reliability and Validity of Research

3.7.1 Reliability

According to Creswell and Creswell (2018), quantitative researchers aim for both validity and reliability in their studies. Reliability refers to the consistency of a measurement, indicating whether the results can be reproduced under the same conditions. LoBiondo-Wood & Haber (2014) further explain that reliability involves the consistency, dependability, and replicability of the results obtained from a research study.

Reliability in this study is ensured through the adoption of the CRISP-DM framework, which offers a standardised and repeatable approach to data mining and model creation in computer science. By utilising well-known Python libraries such as Scikit-learn, Imbalanced-learn, and SHAP, along with cross-validation during model assessment, we can reduce variability and minimise human error. This results in consistent and unbiased outcomes across multiple runs. Additionally, the dataset's preprocessing methods, such as handling duplicates and applying SMOTEENN for balancing, enhance reliability by ensuring data quality and representativeness. This is essential for the StackingClassifier's ability to accurately predict healthcare outcomes.

3.7.2 Validity

Creswell (2015) emphasised that validity involves assessing the accuracy of findings from the perspectives of the researcher, the participant, or the readers of the account.

In this study, validity is ensured through a rigorous feature selection process and statistical tests, such as precision-recall curves and ROC analysis. These methods confirm that the model accurately measures the target constructs, specifically healthcare outcome prediction. SHAP analysis further enhances the model's interpretability by highlighting key features, aligning with research objectives, and reducing the risk of overfitting. Within the CRISP-DM framework, the iterative evaluation phase enables ongoing validation against real-world healthcare needs, thereby ensuring both internal and external validity. This focus on reliability and validity enhances the study's credibility, making it a valuable contribution to research in data science and healthcare.

3.8 Algorithm Selection and Modeling Rationale

In the trial stage, we tested several machine learning algorithms: Logistic Regression, SVM, Random Forest, XGBoost, CatBoost, and LightGBM. The following models were dropped:

Logistic Regression: This model demonstrated mediocre performance, with an AUC of 0.65-0.75, and struggled to capture non-linear relationships.

SVM: Although it was competitive, it required excessive training time (5–10 times longer than other models) and did not scale well.

We retained and combined the following models using ensemble methods (stacking): Random Forest, XGBoost, CatBoost, and LightGBM. The rationale for this decision includes:

Performance with Tabular Healthcare Data: Tree-based ensembles perform exceptionally well on healthcare data, which often has non-linear relationships, mixed data types, and class imbalances.

Handling Missing Values and Categorical Data: These models handle missing values effectively, with CatBoost particularly strong at handling categorical variables. We also addressed class imbalance through weighted sampling.

Efficiency: LightGBM and XGBoost are particularly fast when working with large synthetic datasets.

Interpretability: Using feature importance and SHAP values helps build clinical trust in the algorithms used in sensitive settings.

The ensemble approach achieved superior performance, yielding higher AUC, F1, and recall metrics compared to the baseline models tested during the trials. For evaluation, we employed stratified 5-fold cross-validation (CV) and focused on the following metrics: AUC-ROC (primary), PR-AUC, F1 score, and recall (with a priority on sensitivity), as well as calibration and SHAP explanations.

3.9 Chapter Summary

This chapter outlined the research methodology used in the study. The study began by presenting the research approach, paradigm, and strategy, followed by a detailed description of the methodological framework that guided the research. The framework is structured around the CRISP-DM process flow, comprising six phases: problem understanding, data understanding, data preparation, modeling, evaluation, and deployment. The chapter also described the procedures for data collection and analysis, ensuring transparency in how the dataset was obtained and processed. Finally, it addressed issues of reliability and validity to establish the credibility and trustworthiness of the research.

CHAPTER FOUR: DESIGN AND IMPLEMENTATION

This chapter outlines the design and implementation of the study. It provides a comprehensive overview of the experimental design, detailing the procedures for training, testing, and evaluating the healthcare prediction model. The chapter is structured as follows: Section 4.1 presents the design. In subsection 4.1.1, a design overview diagram is provided, followed by the algorithm flow in subsection 4.1.2. Finally, subsection 4.1.3 outlines the software and hardware used during the study. Section 4.2 focuses on the implementation of the study, discussing data preparation and understanding in subsection 4.2.1, followed by final model development in subsection 4.2.2. Section 4.3 outlines the process of developing the user interface using Streamlit. Finally, Section 4.4 concludes the chapter.

4.1 Design

4.1.1 Model design overview

Figure 4.1 illustrates the end-to-end big data analytics model workflow, which was followed when implementing the final model. As shown in Fig. 4.1, a person logs into the computer and then runs a Python code. The training data is used to build and train machine learning models using various algorithms such as Random Forest, XGBoost, LightGBM, and CatBoost. Once trained, the model was tested using the test data. The model evaluation stage involves assessing its performance using precision, accuracy, recall, and F1-score metrics. The final results are then used to generate insights for decision-making. This will prompt them to import libraries. The data is then extracted from the generated dataset in CSV format, which is suitable for our study. The use of synthetic datasets is becoming increasingly important in healthcare. Real medical data is highly sensitive and protected by stringent privacy rules and laws; therefore, researchers and developers often do not have direct access to it (Appenzeller et al., 2022). Synthetic data helps address privacy concerns and increases the availability and diversity of real data. In addition, real-world data is scarce, which may take time to complete the work; the

cost of purchasing data, as organisations charge exorbitant fees to sell it, and the time required for collection. This is vital for training AI models that enhance patient outcomes and advance healthcare research, enabling researchers to conduct tests without compromising patient identities.

(Pezoulas et al., 2024). The data will undergo data preprocessing, and the resulting data will be processed. The data is then split into 80% (training) and 20% testing. The training data is used to build and train machine learning models using various algorithms such as Random Forest, XGBoost, LightGBM, and CatBoost. Once trained, the model was tested using the test data. The model evaluation stage involves assessing its performance using precision, accuracy, recall, and F1-score metrics. The final results are then used to generate insights for decision-making.

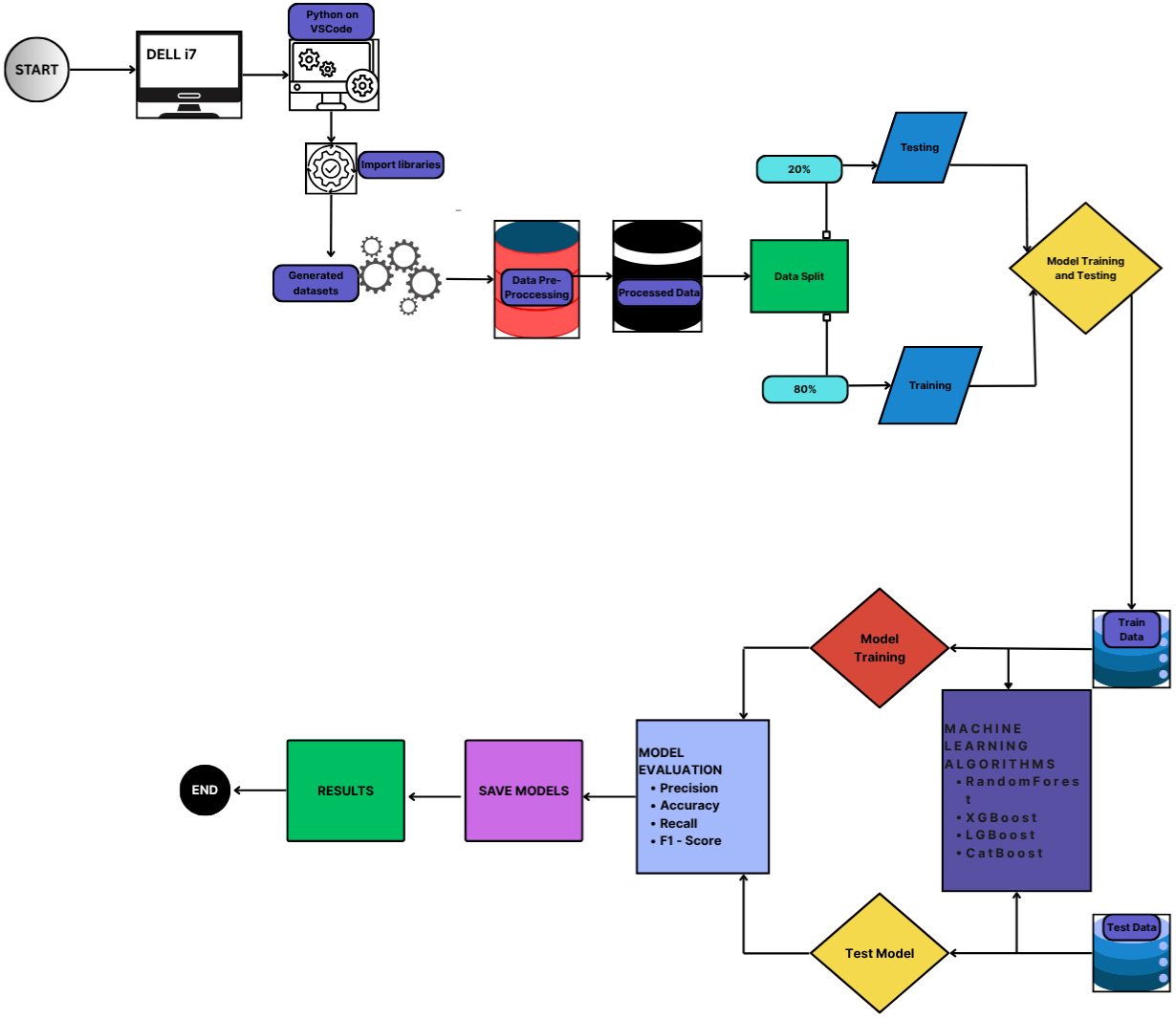


Figure 4.1 Model Design Overview

4.1.2 Model flow algorithm

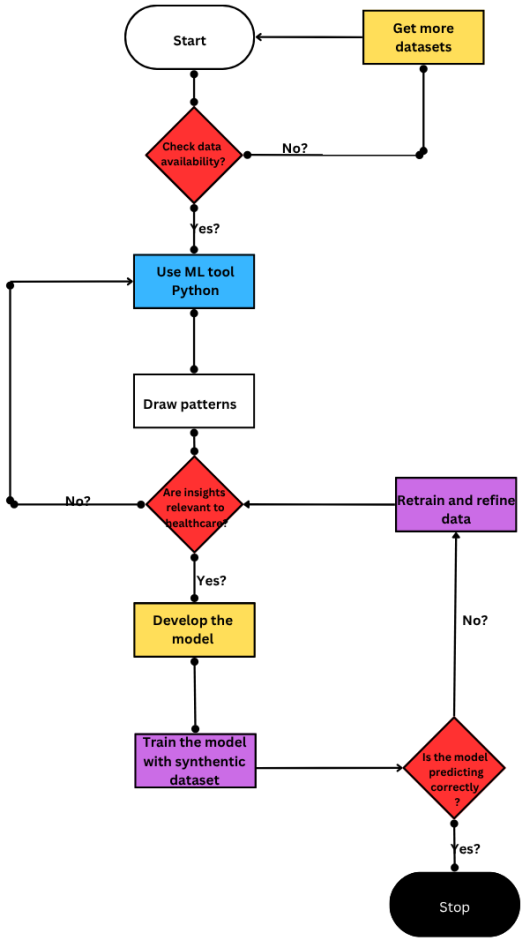


Figure 4.2 BDA Model Flow algorithm

4.1.3 Software and Hardware Requirements

To run the model as described above, it requires certain minimum hardware and software requirements.

4.1.3.1 Hardware requirements

The minimum requirement to efficiently run machine learning depends on model complexity, framework, and dataset size. A computer with a multi-core CPU (Core i3, 8GB RAM, and 256 GB of storage) can be suitable for a basic ML model. However, in our case, the experimentations were conducted on a personal laptop with the following specifications: a Dell Latitude 3540, equipped with 16GB of RAM, system BIOS Dell Inc. 1.18.0 version, and a 13th Gen Intel® Core™ i5-1335U, 1300 MHz, 10 Core(s) Logical processor. This hardware setup provided sufficient computational capacity to handle the large dataset size, train the stacking classifier, and support the development of the Streamlit interface. Figure 4.3 shows the System information of the hardware used.

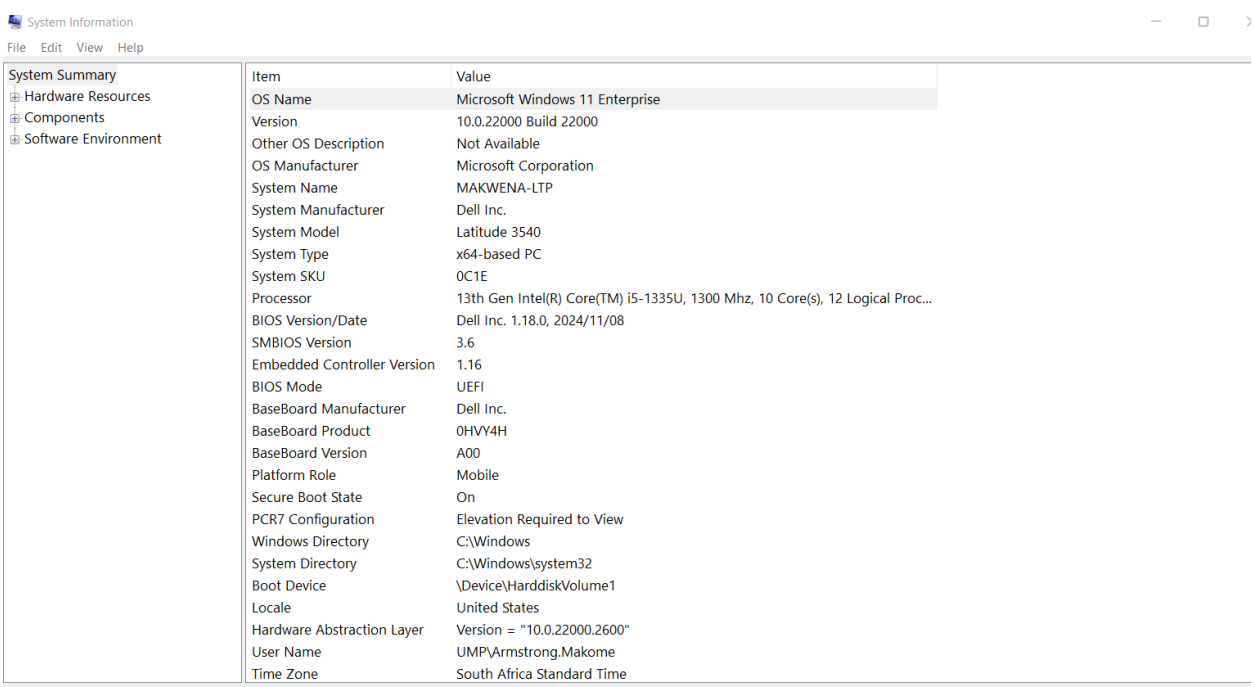


Figure 4.3 System information of the hardware used

4.1.3.2 Software used

The minimum software requirements to run the model and its supported libraries are Windows 10, macOS, or Linux operating systems. In this study, the computer was running on the Windows 11 Enterprise operating system. Visual Studio Code version 1.100.3 is used as the primary IDE, as it supports Python. Furthermore, Python version 3.13.0 was used, which was compatible with some of the libraries and the environment setup. The necessary Python libraries, including Pandas, NumPy, and Scikit-learn, were imported to implement the proposed models and interface. These libraries make designing and evaluating machine learning models easier, allowing for more efficient experimentation without requiring a complete start from scratch. The purpose of these libraries and how to import them is explained in detail in Section 4.2.1.1.

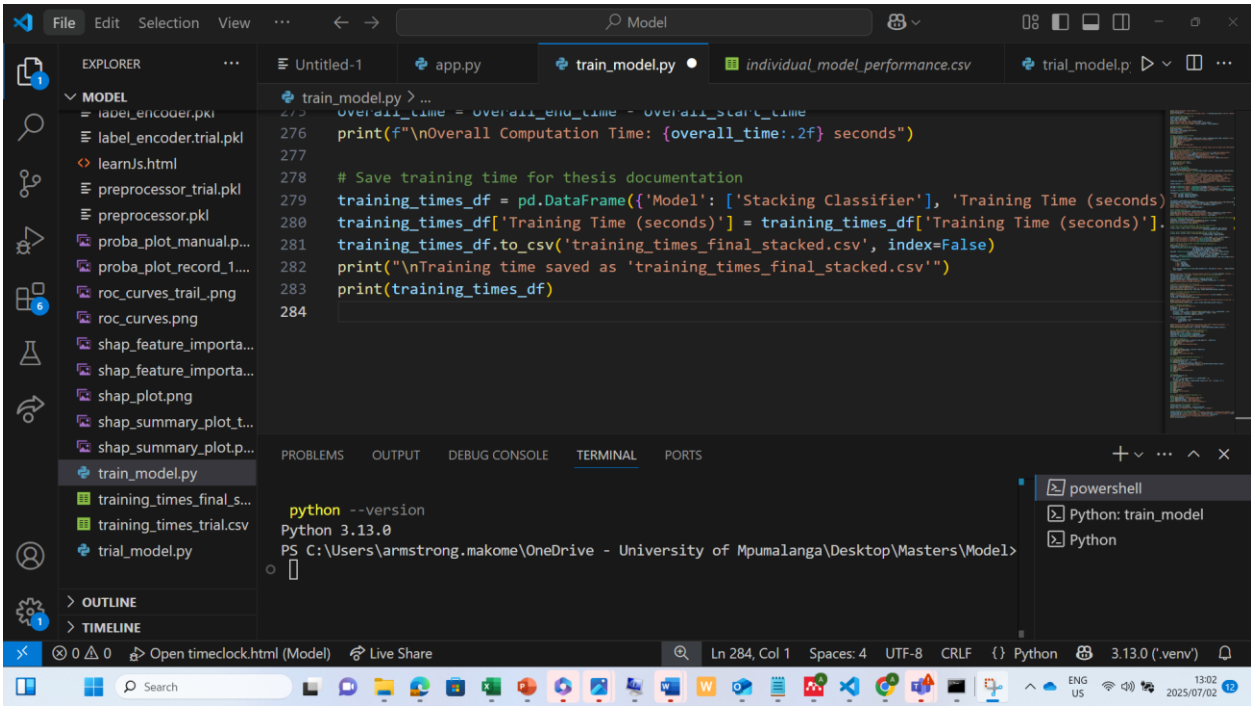


Figure 4.4 Python version 3.13.0 installed on this PC

4.2 Implementation

4.2.1 Data understanding / Preparation

4.2.1.1 Importing libraries After installing the minimum hardware, operating system, VS Code, and Python, as highlighted above, Python libraries were imported using the following commands: *pip install pandas, pip install shap, pip install xgboost, and pip install matplotlib*. Then, the dataset files, in our case, were imported, specifically Final_healthcare_bigdata.csv. Appendix A shows the full code, including all the libraries used.

Below are the machine learning and development libraries that were imported for this study:
The purpose of the imported ML and development libraries is explained below.

- **Pandas:** Used for data manipulation and analysis, facilitating the loading and preprocessing of the synthetic dataset.
- **NumPy:** Used for numerical computing, supporting processes on engineered features like log transformations.
- **Scikit-learn:** Provided tools for preprocessing (StandardScaler, OneHotEncoder, OrdinalEncoder), model selection (train_test_split, StackingClassifier), and evaluation (confusion_matrix, classification_report).
- **Random Forest, XGBoost, LightGBM, CatBoost:** XGBoost, LightGBM, and CatBoost are used to implement base models in the stacking classifier, offering advanced gradient boosting capabilities.
- **SHAP:** Utilised for model interpretability, generating SHAP values to explain feature importance in the Streamlit interface.
- **Matplotlib and Seaborn:** used for data visualization, including the construction of confusion matrices and prediction probability charts.
- **Streamlit:** Used for developing the interactive web user interface, allowing users to upload/ input patient data and display predictions.

- **Joblib:** Facilitated model persistence to save the trained pipeline and preprocessing model files.

Appendix A below shows an example of the pip command executed to install the Pandas library.

4.2.1.2 Loading, preparation and visualisation of datasets

To load the dataset and display a summary of the data overview, the code snippet shown in Figure 4.5 was run. This code loads the dataset and provides a summary of the data structure. Upon successful loading of the dataset saved as 'Final_healthcare_big_data.csv, the output confirms that the data were successfully loaded. In our case, the dataset contained 150,000 records and 17 columns, which define its shape.

```
# LOAD & PREPARE DATA (CRISP-DM: Business Understanding, Data Understanding)
print("=== Data Understanding ===")

# Load the dataset (150,000 instances)
data = pd.read_csv('Final_healthcare_big_data.csv')
print("Dataset loaded successfully.")
```

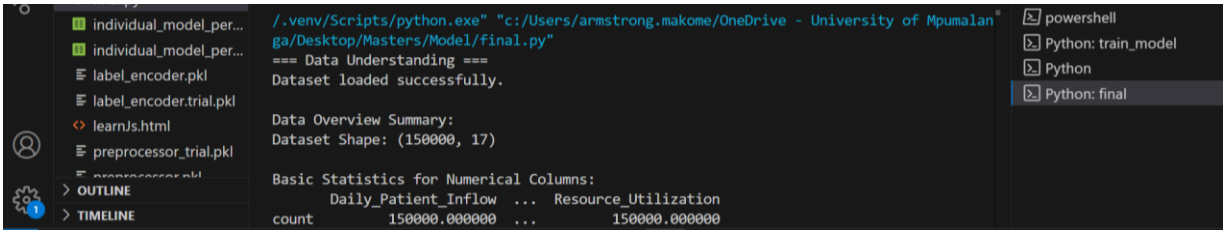


Figure 4.5 Load data and show loaded successfully

After the code was successfully executed, the column information for our datasets will be seen as shown in Figure 4.6 below shows that

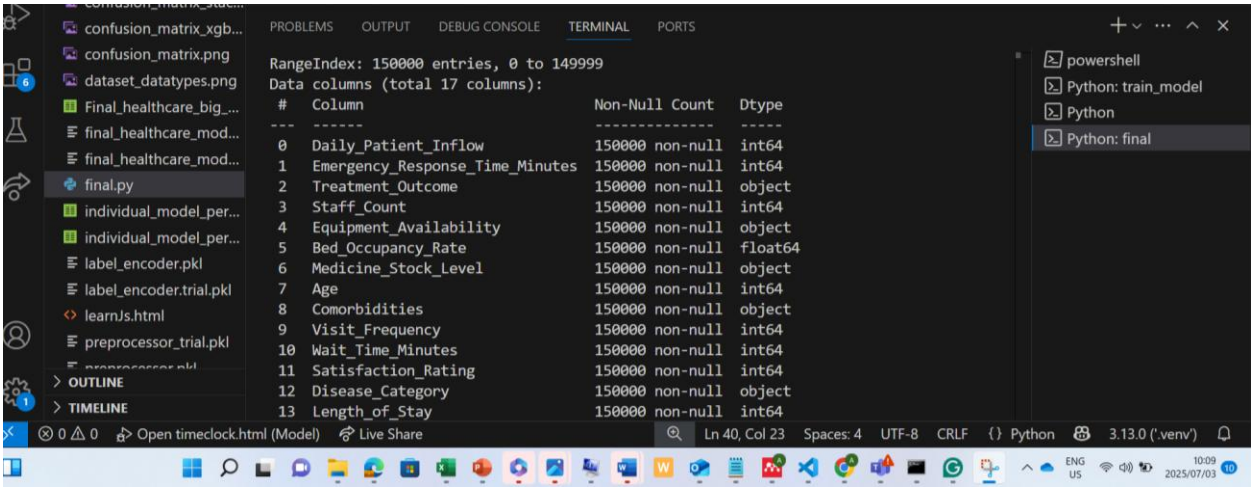


Figure 4.6 Column information

To see the initial class distribution of the original dataset, the following code was used.

```
print("\nClass Distribution of Target Variable ('Outcome'):")
print(data['Outcome'].value_counts())
```

Figure 4.7 below displays the class distribution across all three classes (Improved, Unchanged and Worsened).

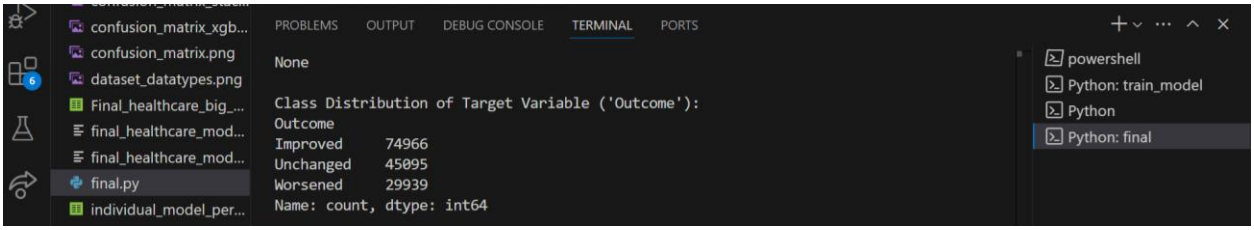


Figure 4.7 Class distribution

4.2.1.3 Data Preparation and Preprocessing

After the above process, the datasets were cleaned by checking or handling missing values, further checking duplicates, and dropping them if found. The following code was used to check for missing values.

```
missing_values = data.isnull().sum()
```

The reason for doing data cleaning and checking missing values, as supported by Ridzuan and Zainon (2019), is to identify the errors, detect the errors ,and correct them by checking missing values and removing duplicates, which guarantees data accuracy, consistency, and reliability, eventually leading to further reliable analysis and model performance (Durgapal, 2023). The snippet of code below was used to check for missing values.

Figure 4.8 below shows that there were no missing values found.

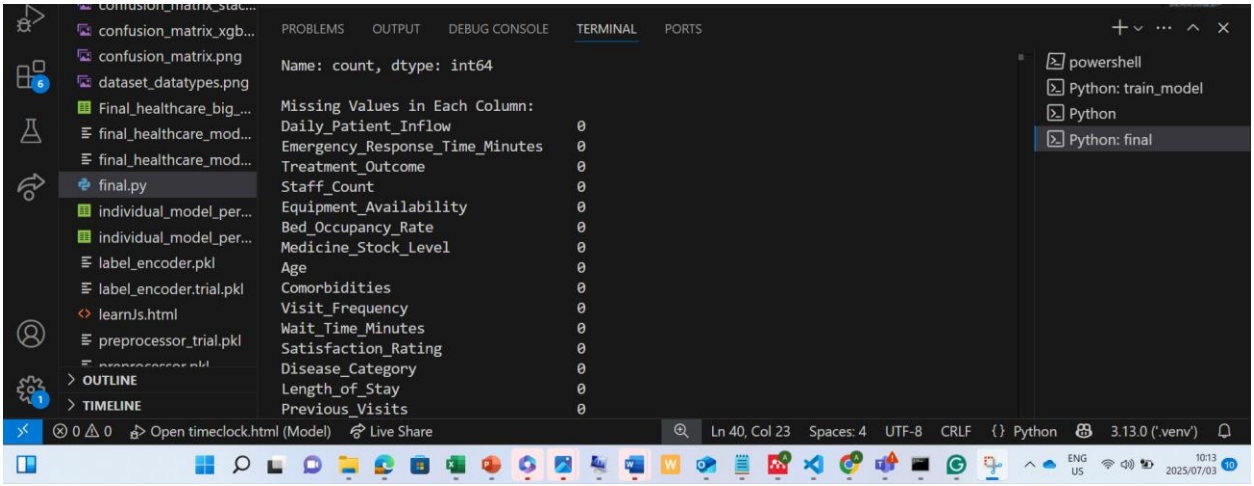


Figure 4.8 Missing values

To check for duplicates and drop them, the code below was used.

```
initial_shape = data.shape
data = data.drop_duplicates()
final_shape = data.shape
```

Figure 4.9 below shows that there were no duplicates found in the datasets.

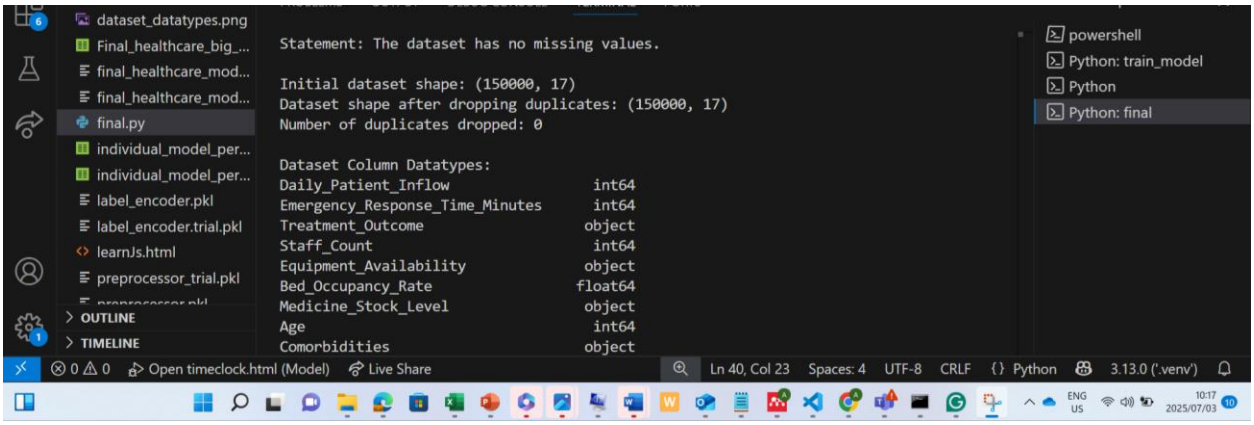


Figure 4.9 Checking and dropping duplicates

To drop redundant features, the following code was used.

```
X_df = pd.DataFrame(X_processed, columns=feature_names)
X_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'], axis=1, inplace=True,
errors='ignore')
```

Figure 4.10 displays the output of the redundant features which were dropped.

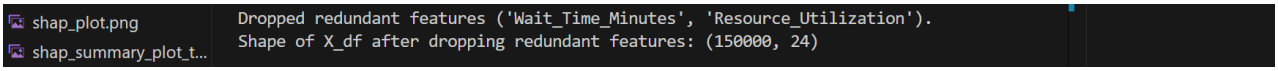


Figure 4.10 Dropping redundant features

4.2.1.5 Feature engineering

The snippet of code below was used to perform feature engineering. According to Murel and Kavlakoglu (2024), feature engineering converts raw data into pertinent information for application in machine learning models. Further, it refers to the process of developing features for predictive models.

```
data['Resource_Bed_Interaction'] = data['Resource_Utilization'] *
data['Bed_Occupancy_Rate']
data['Wait_Length_Interaction'] = data['Wait_Time_Minutes'] *
data['Length_of_Stay']
data['Log_Wait_Time'] = np.log1p(data['Wait_Time_Minutes'])
data['Log_Resource_Utilization'] = np.log1p(data['Resource_Utilization'])
data['Satisfaction_Staff_Interaction'] = data['Satisfaction_Rating'] *
data['Staff_Count']
```

```
print("Feature engineering completed.")
```

Figure 4.11 below displays the output indicating that feature engineering was performed successfully.

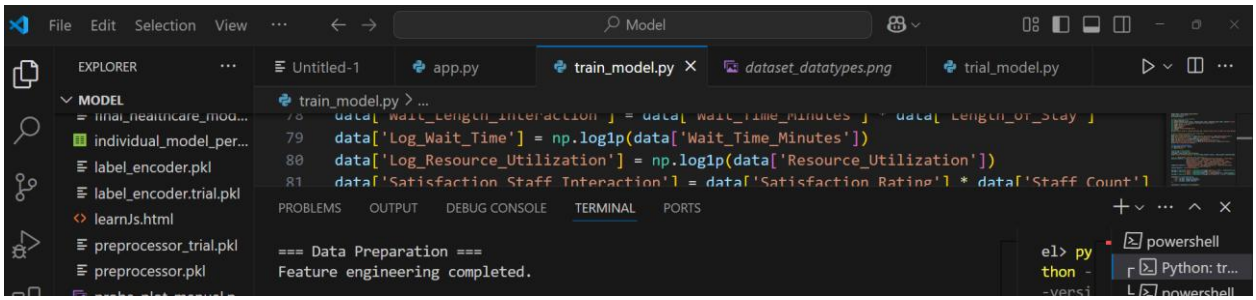


Figure 4.11 Feature engineering

4.2.1.6 Separate feature and target

The snippet of code below was used to separate features and the target variable.

```
X = data.drop('Outcome', axis=1)
y = data['Outcome']
```

Feature separation guarantees that the model learns from the correct data and prevents mistakes that include the target in the input features (Rajandran, 2023).

4.2.1.7 Encoding the target variable

The snippet of the code below shows how the target variable was encoded. The primary purpose of the target encoder is to substitute categories with a metric reflecting their potential impact on the target variable (Trevisan, 2022). The target variable was encoded using a method known as target encoding. This approach replaces categorical values with the mean of the target variable in each category. This effectively turns the target variable into numerical values.

```
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
print("\nLabel Encoding Mapping:", dict(zip(label_encoder.classes_,
label_encoder.transform(label_encoder.classes_))))
```

Figure 4.12 below shows that the target variable was performed successfully, and the labelling of the encoding mapping output.

Figure 4.12 Encoding the target variable

4.2.1.8 Preprocessing for data preparation

The code below shows how the data was prepared through preprocessing. Preprocessing and preparation are fundamental steps in transforming raw data into meaningful information because raw data may be incomplete, redundant, or noisy (Adari et al., 2023). The snippet code below was used for rounding features by data type: numerical, categorical, and ordinal.

```
numerical_features = ['Daily_Patient_Inflow', 'Emergency_Response_Time_Minutes',
'Staff_Count',
                      'Bed_Occupancy_Rate', 'Visit_Frequency',
'Wait_Time_Minutes',
                      'Length_of_Stay', 'Previous_Visits',
'Resource_Utilization', 'Age',
                      'Resource_Bed_Interaction', 'Wait_Length_Interaction',
'Log_Wait_Time',
                      'Log_Resource_Utilization',
'Satisfaction_Staff_Interaction']
categorical_features = ['Treatment_Outcome', 'Equipment_Availability',
'Medicine_Stock_Level',
                        'Comorbidities', 'Disease_Category']
ordinal_features = ['Satisfaction_Rating']
```

The snippet of the code below was used to impute missing values using median, scale numerical data using StandardScaler, mode, and encode using OneHotEncoder.

```
num_pipe = Pipeline([('imputer', SimpleImputer(strategy='median')), ('scaler',
StandardScaler())])
cat_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
                     ('onehot', OneHotEncoder(drop='first', sparse_output=False,
handle_unknown='ignore'))])
```

The snippet code below was used to impute missing values and standardise ordinal features. Further, the researchers used ColumnTransformer to apply preprocessing to each group of features, accordingly, combining all pipelines. Lastly, fit and transform the input features using the preprocessing pipeline to generate new feature names after transformation was done.

```
ord_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
('scaler', StandardScaler())])

preprocessor = ColumnTransformer([
    ('num', num_pipe, numerical_features),
    ('cat', cat_pipe, categorical_features),
    ('ord', ord_pipe, ordinal_features)
])

X_processed = preprocessor.fit_transform(X)
cat_feature_names =
preprocessor.named_transformers_['cat'].named_steps['onehot'].get_feature_names_o
ut(categorical_features)
feature_names = numerical_features + list(cat_feature_names) + ordinal_features
```

Figure 4.13 below shows output that the preprocessing was done successfully and depicts the shape of x successfully processed. For more detailed code, go to Appendix A.

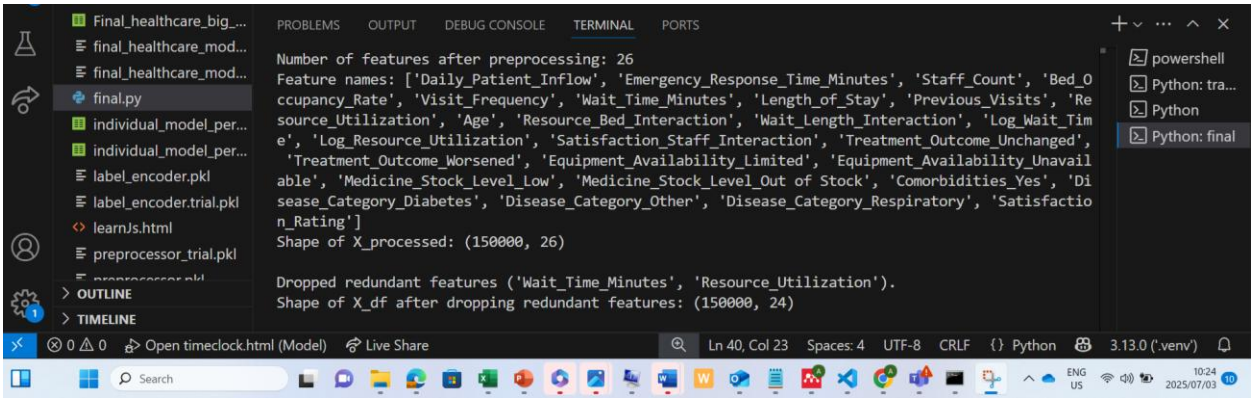


Figure 4.13 Preprocessing of data

4.2.1.9 Balancing the datasets

The snippet of code below was used to balance the datasets during the data preparation stage, using SMOTEENN.

```
smote_enn = SMOTEENN(random_state=42)
X_balanced, y_balanced = smote_enn.fit_resample(X_df.values, y_encoded)
```

The purpose of balanced datasets, particularly in machine learning, is to ensure that each class in the data contains an equal or representative number of instances (Mujahid et al., 2024). This is critical for keeping models from being biased towards the majority class and enhancing overall model performance. Siddique et al. (2021) state that SMOTEENN deals perfectly with oversampling and undersampling, two often-used balancing strategies.

Figure 4.14 below shows that the code was successfully executed, and the class distribution can be seen.

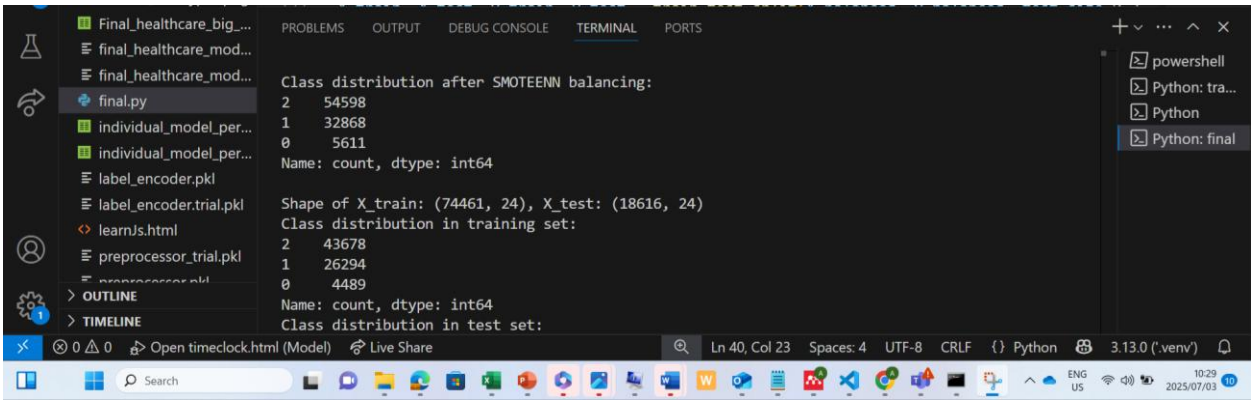


Figure 4.14 Class distribution after SMOTEENN

4.2.1.10 Dataset and Splitting

The dataset included 150,000 patient records, with a class distribution of improved 49.98%, unchanged 30.06%, and worsened 19.96%. The dataset was split into training and test sets using scikit-learn's train_test_split function, with a test size of 20% and a train size of 80%. The snippet code below was used for data splitting.

```
X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced,
test_size=0.2,
                                                    stratify=y_balanced,
random_state=42)
```

Figure 4.15 shows the output of splitting the dataset performed successfully. For more detailed code, refer to Appendix A.

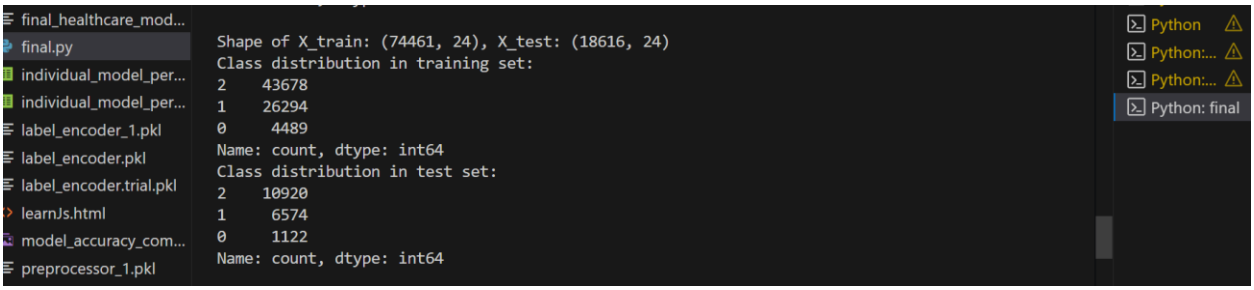


Figure 4.15 Data Splitting

4.2.1.11 Handling validation

Validation was handled internally via 5-fold cross-validation in a classifier, which acknowledges that the cv=5 parameter in the classifier provides internal validation through 5-fold cross-validation on the training set, splitting the 80% training and 20% validation fold per iteration. Below is the snippet of code that was used to handle validation.

```
final_estimator=XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42),
    cv=5, n_jobs=1
)
```

4.2.1.12 Algorithms Evaluation

During the development phase, the researchers evaluated various classification algorithms (Random Forest, XGBoost, LightGBM, and CatBoost) before developing the final model. This assisted in selecting the best-performing algorithm for our final model development. To improve prediction performance and accuracy, the study used a stacking classifier that included four machine learning algorithms: Random Forest,

XGBoost, LightGBM, and CatBoost, along with a final XGBoost estimator. Stacking is an ensemble learning approach that leverages the capabilities of multiple base models to increase accuracy and resilience (Dey & Mathur, 2023), making it ideal for the multi-class classification task in this work. Random Forest uses decision trees to handle non-linear relationships (Liu et al., 2022), XGBoost and LightGBM use gradient boosting to achieve high performance on structured data (Wiens et al., 2025), and CatBoost excels at handling categorical features and imbalanced datasets, which is critical given the synthetic dataset's class imbalance (fewer "Worsened" cases) (Nguyen & Ngo, 2025).

4.2.1.13 Methodological rationale for data preparation and model selection

This study aimed to develop a robust multiclass prediction capability that can support evidence based decision making for service delivery improvement in South African public hospitals. The dataset used for the experiments is large and structured, containing both numerical and categorical variables, and the prediction target represents three clinically meaningful operational outcomes, namely improved, unchanged, and worsened. In this setting, the modelling pipeline must achieve three things at the same time. First, it must protect the validity of evaluation by preventing information leakage from the target into predictors and by using an evaluation strategy that reflects generalisation to unseen cases. Second, it must transform the raw columns into a form that machine learning algorithms can use reliably, while preserving the meaning of the variables. Third, it must adopt model families that are well suited to tabular data with mixed feature types, potential non-linear relationships, and interaction effects that reflect real service delivery dynamics. The methodological reasoning for the key design choices is explained below.

Rationale for cleaning and preparing the data

Even when a dataset appears well formed, cleaning checks remain necessary because model performance can be artificially inflated by hidden duplicates, inconsistent records, or unintended redundancy. Duplicate rows can bias training by repeating the same patterns, reducing the effective diversity of the training set and producing overly optimistic results (Kuhn & Johnson, 2019). The duplicate check therefore served as a safeguard for

reliability and reproducibility, ensuring that the learned decision rules are driven by genuine variation rather than repeated records.

With respect to missing values, two methodological principles guided the design. The first is that real world healthcare operational data often contain missingness due to capture errors, workflow interruptions, or system integration gaps (Little & Rubin, 2019). The second is that an implementation that is intended as a transferable prototype should remain robust even when deployed on new data whose missingness pattern may differ from the training data. For that reason, the preprocessing pipeline used imputation strategies that are appropriate for variable type, using median imputation for continuous variables to reduce sensitivity to skewness and extreme values, and most frequent imputation for categorical and ordinal variables to preserve plausible category membership (Schafer & Graham, 2002; Kuhn & Johnson, 2019). This is methodologically defensible because it reduces the risk of discarding records, which is particularly important in healthcare contexts where rare events and minority classes may already be underrepresented (Buda et al., 2018).

Rationale for encoding categorical and ordinal variables

Most machine learning algorithms used in this study require numeric inputs, yet several predictors are categorical by nature, such as disease category or equipment availability. One-hot encoding was used for nominal variables because it represents categories without imposing an artificial ordering, which would be incorrect for purely nominal fields (Hastie et al., 2009; Kuhn & Johnson, 2019). In addition, the encoding configuration was designed to handle previously unseen categories at inference time, which is important in practical deployment where new codes or values can appear (James et al., 2021).

For ordinal variables such as satisfaction rating, the methodological goal is to preserve the ranking information, since the difference between low and high satisfaction carries meaning that can relate to service outcomes. Treating such variables as purely nominal can discard useful structure, while treating them as continuous without care can overstate distance between levels. A pragmatic approach is to represent ordinal levels consistently

and apply scaling within the pipeline so that the variable contributes comparably during model training (Kuhn & Johnson, 2019; James et al., 2021).

The target variable was encoded to numeric labels to support multiclass learning, since the algorithms ultimately optimise loss functions defined over numeric class indices even when the classes are semantically named (Hastie et al., 2009). This encoding does not change the meaning of the target classes. It is simply a representational step that enables model fitting and evaluation.

Rationale for transformations and feature engineering

Healthcare service delivery outcomes are rarely driven by single variables acting independently. Operational pressure often emerges through combinations such as high waiting time together with longer length of stay, or high resource utilisation in a context of high bed occupancy. Feature engineering was therefore used to capture interaction effects that reflect these compound conditions. Interaction terms are widely used to allow models to represent relationships that are otherwise difficult to learn from raw variables alone, especially when the true mechanism involves multiplicative or conditional effects (Hastie et al., 2009; Kuhn & Johnson, 2019). In practical terms, variables such as wait time and length of stay may each correlate with outcomes, but their joint escalation can signal congestion and delayed throughput, which aligns with a worsened service outcome scenario.

Log transformations were applied to variables such as waiting time and resource utilisation to reduce skewness and compress extreme values. In operational datasets, these measures often exhibit long tailed distributions where a small number of observations can dominate scale and variance. Log transformation is a standard technique for stabilising variance and improving the learnability of relationships for many modelling approaches (Box & Cox, 1964; Osborne, 2002). The use of a log plus one transform is particularly practical because it is defined at zero and supports non negative measures common in operational metrics.

Some redundant features were removed after transformation and feature creation to reduce unnecessary duplication of information. When both a raw variable and a deterministic transformation of that variable are retained, the model can overemphasise

a single underlying concept, increasing sensitivity to noise and making interpretation less stable. Feature reduction in such cases is not only a computational choice but also a methodological one that promotes parsimony and reduces the risk of overfitting (Guyon & Elisseeff, 2003; Kuhn & Johnson, 2019).

Rationale for the training and testing split strategy

The study adopted an eighty percent training and twenty percent testing split, with stratification to preserve the class distribution. The methodological purpose of stratification is to ensure that each split contains representative proportions of the outcome classes so that evaluation is not distorted by random overrepresentation of the majority class in either the training or testing subset (Kohavi, 1995; James et al., 2021). Given the large sample size, an eighty twenty split provides a strong balance between learning capacity and evaluation credibility. The training set remains large enough for complex models to learn stable patterns, while the test set remains sufficiently large to provide reliable estimates of performance across the three classes.

A clear separation between training and testing is essential because model selection and tuning performed on the same data used for final evaluation can lead to overly optimistic performance estimates, especially for flexible models such as ensembles (Hastie et al., 2009). The test set therefore functions as an independent benchmark that approximates deployment on unseen cases.

Rationale for cross validation as internal validation

Cross validation was used to provide internal validation during model building, supporting more reliable hyperparameter selection and reducing sensitivity to a single split. The central methodological benefit of cross validation is that it produces a performance estimate averaged across multiple folds, which reduces variance and offers a more stable view of generalisation performance than a single validation split (Kohavi, 1995; Hastie et al., 2009). A fivefold configuration was appropriate because it provides a good compromise between computational cost and estimation stability, particularly for large datasets and ensemble models where training can be expensive (James et al., 2021).

In addition, cross validation is particularly important for stacking because it helps ensure that the meta learner is trained on out of fold predictions, which reduces leakage and prevents the meta learner from simply memorising the training labels through overly optimistic base model outputs (Wolpert, 1992; van der Laan et al., 2007).

Rationale for handling class imbalance

Although the dataset contains all three classes in meaningful volumes, the worsened class remains the minority, and minority class performance is often the most operationally important because it can represent deterioration in service delivery. Class imbalance can bias learning algorithms toward the majority class, resulting in high overall accuracy but poorer sensitivity for the minority outcome (He & Garcia, 2009; Krawczyk, 2016). For that reason, the study applied SMOTEENN, a hybrid resampling approach that combines synthetic minority oversampling with a cleaning step that removes ambiguous samples using edited nearest neighbours. The methodological advantage is that it can increase minority representation while reducing class overlap that might otherwise degrade decision boundaries (Batista et al., 2004; He & Garcia, 2009).

Resampling alone is not always sufficient, especially when misclassification costs are asymmetric in practice. The study therefore also incorporated class weighting during training so that errors on underrepresented outcomes are penalised more heavily. Cost sensitive learning through class weights is a recognised strategy for improving minority class recall without discarding majority class information (Elkan, 2001; Buda et al., 2018). Using both resampling and weighting is methodologically reasonable because they address different aspects of imbalance. Resampling changes the effective training distribution, while weighting changes the optimisation pressure during learning.

Principles of the selected ensemble models and their suitability for this task

The study implemented Random Forest, XGBoost, LightGBM, CatBoost, and a stacking ensemble. These choices are defensible for three key reasons. First, the dataset is structured and tabular, a setting in which tree based ensembles have repeatedly shown strong performance relative to linear models, especially when relationships are non linear and involve interactions (Hastie et al., 2009; James et al., 2021). Second, the predictors

include mixed types and engineered interactions, and tree based methods naturally handle non linear splits and feature interactions without requiring explicit specification of functional forms. Third, the research objectives emphasise both predictive performance and actionable insight, and tree based ensembles integrate well with post hoc explainability approaches such as SHAP for feature attribution, which supports interpretation of drivers of outcomes (Lundberg & Lee, 2017b).

Random Forest is a bagging based ensemble that builds many decision trees using bootstrap samples and random feature selection at splits. Its key principle is variance reduction through averaging, which produces stable predictions and reduces sensitivity to noise, a valuable property for large operational datasets (Breiman, 2001). Because Random Forest can model complex patterns without strong parametric assumptions, it is appropriate for service delivery outcomes that can depend on multiple interacting operational factors.

Gradient boosting methods such as XGBoost and LightGBM build trees sequentially, where each new tree corrects errors made by the previous ensemble. The underlying idea is additive modelling under a differentiable loss, producing strong predictive performance when tuned appropriately (Friedman, 2001; Hastie et al., 2009). XGBoost is particularly well known for its regularisation and practical optimisation strategies that improve accuracy and limit overfitting, making it suitable for high dimensional engineered feature spaces (Chen & Guestrin, 2016). LightGBM is designed for efficiency on large datasets through histogram based splitting and optimised tree growth strategies, which supports scalability and faster experimentation without sacrificing predictive quality (Ke et al., 2017).

CatBoost was included because it is designed to handle categorical features effectively through ordered target statistics that reduce target leakage during encoding and improve stability, especially where categorical predictors carry meaningful signal (Prokhorenkova et al., 2018). Even when one hot encoding is applied, CatBoost can still provide competitive performance and robustness in mixed feature settings, and it often performs well when the dataset contains categorical structure and complex interactions.

Rationale for stacking as the final modelling approach

No single algorithm is consistently best across all datasets. Stacking addresses this by combining diverse base learners and training a meta learner to integrate their strengths. The methodological principle is that different learners capture different aspects of the data generating process. When their errors are not perfectly correlated, a meta learner can improve overall generalisation (Wolpert, 1992; van der Laan et al., 2007). In this study, Random Forest contributes stability and variance reduction, while boosting methods contribute strong bias reduction and fine grained decision boundaries. The stacking design is therefore aligned with the goal of producing a more accurate and robust multiclass predictor for service delivery outcomes.

Importantly, stacking was paired with cross validation to ensure that the meta learner is trained on predictions that reflect out of sample behaviour rather than on predictions generated from the same data used to fit the base models. This reduces leakage and strengthens the credibility of the evaluation (Wolpert, 1992; van der Laan et al., 2007).

4.2.2 Modelling Development

4.2.2.1 Model Experimental Development Process

The Stacked model (random forest, XGBoost, LGBM, and Catboost) models were trained using the training data, while hyperparameter tuning was conducted using the validation set. This code was used to train the algorithms (RF model, XGB model, LGBM model, and CatBoost model) with the class weight proposed by Razali et al. (2025), which offers more balanced predictions by penalizing the model for erroneously classifying a minority class. Furthermore, stacking of the model was introduced according to Lu et al. (2023), which involves using multiple learning algorithms to generate an ideal model, thereby increasing both the model's generalizability and performance.

The snippet of the code below shows the code that was used to build the final model, this is the engine part of the code that was used to build the final model from the stacked algorithms, and the output on the console displays all the algorithms trained and further shows the final stacked classifier trained successfully, which is an indication that our code

was executed successfully. For more detailed information on the code, refer to Appendix A.

```
print("\n=== Modeling ===")
class_weights = {0: 5.0, 1: 1.0, 2: 1.0}
rf_model = RandomForestClassifier(n_estimators=500, max_depth=None,
min_samples_split=2,
                                  min_samples_leaf=2, max_features='log2',
                                  class_weight=class_weights, random_state=42)
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42,
                          n_estimators=1000, max_depth=10, learning_rate=0.1,
                          subsample=0.9, colsample_bytree=1.0, reg_lambda=1,
reg_alpha=1)
lgbm_model = LGBMClassifier(class_weight=class_weights, random_state=42)
catboost_model = CatBoostClassifier(verbose=0, early_stopping_rounds=10,
auto_class_weights='Balanced')

stacking = StackingClassifier(
    estimators=[
        ('rf', rf_model),
        ('xgb', xgb_model),
        ('lgbm', lgbm_model),
        ('cat', catboost_model)
    ],
    final_estimator=XGBClassifier(use_label_encoder=False,
eval_metric='mlogloss', random_state=42),
    cv=5, n_jobs=1
)

print(f"Processing Training Set: Training Stacking Classifier on
{X_train.shape[0]} instances...")
stacking_start_time = time.time()
stacking.fit(X_train, y_train)
stacking_end_time = time.time()
stacking_training_time = stacking_end_time - stacking_start_time
print(f"Training time for Stacking Classifier: {stacking_training_time:.2f}
seconds")
print("Stacking classifier trained successfully.")
```

The snippet of code shows the actual chosen final model.

```

joblib.dump(stacking, 'final_healthcare_model_1.pkl')
joblib.dump(preprocessor, 'preprocessor_1.pkl')
```

```
joblib.dump(label_encoder, 'label_encoder_1.pkl')
print("\nModels and preprocessors saved successfully.")
```

The performance of the stacked model was evaluated using metrics such as accuracy, precision, recall, and F1 score.

The evaluation phase examined the efficacy of the predictive models to confirm their alignment with the study objectives. Metrics including accuracy, precision, recall/ training time, and F1-score were employed, as advised by Syed et al. (2021), to assess the models' reliability. Validation techniques, such as cross-validation, were applied to prevent overfitting, guided by Lenth (1995) and Wani et al. (2018), who emphasized the importance of rigorous evaluation in predictive modeling. Despite the absence of deployment, the models underwent conceptual testing to demonstrate their ability to yield actionable insights for improving healthcare services. Furthermore, utilising the confusion matrix during the evaluation step ensured a thorough assessment of the model's performance. It offered actionable insights to inform the continued enhancement of predictive algorithms and the study's objective of optimising service delivery in South African public hospitals. The study employed experimental evaluation instead of a real-world or simulation. Furthermore, use classification metrics and SHAP explainability to assess model performance.

a) Confusion Matrix

According to Susmaga (2004) and Beauxis-Aussalet and Hardman (2014), the confusion matrix is a prevalent tool in machine learning for assessing and visualising model performance in supervised learning.

A confusion matrix for a multi-class classification displays the frequency of correct and incorrect predictions for each class. Varoquaux and Colliot (2023) indicate that it provides a detailed view of model errors, indicating which classes are often misclassified. In

healthcare, it helps identify the systematic diagnostic mistakes between similar conditions and guides model refinement (Varoquaux & Colliot, 2023).

b) Accuracy

According to Rezvani and Wang (2023), accuracy is a critical statistic and the ultimate purpose of performance analysis since it shows how accurate the model's predictions are. Equation 4.1 illustrates how accuracy is computed as the ratio of properly categorised examples (True Positive + True Negative) to the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 4.1 Accuracy

Accuracy measures the proportion of correctly predicted instances over the total instances. In a healthcare setting, it remains applicable in multi-class settings; however, it may be misleading in imbalanced datasets, which are common in healthcare, where certain conditions are rare (Varoquaux & Colliot, 2023). Furthermore, the researchers indicate that high accuracy might still indicate poor performance in minority classes, resulting in insufficient performance as a standalone metric.

c) Precision and Recall

Precision measures the model's ability to identify positive instances for a specific class, focusing on reducing false positives. According to Kulkarni et al. (2020), precision is calculated as the ratio of true positives to the total predicted positives for a given class, as shown in Equation 4.2

$$Precision = \frac{TP}{TP + FP}$$

Equation 4.2 Precision

As per Kulkarni et al. (2020), recall is the ratio of true positive (TP) instances to the total actual positive instances for a given class, calculated as shown in Equation 4.3:

$$Recall = \frac{TP}{TP \ + \ FN}$$

Equation 4.3 Recall

Precision quantifies how many of the predicted positives are truly positive. Recall indicates how many actual positive instances were correctly predicted. Varoquaux and Colliot (2023) allude to a multiclassification, where precision and recall can be computed per class and averaged using macro or micro methods. Furthermore, the researchers added that these metrics are important in medical diagnosis, where false positives and negatives come with different risks.

d) F1 Score

Rezvani and Wang (2023) describe the F1-score as the harmonic mean of precision and recall, providing a balanced measure of a model's performance, particularly in scenarios with imbalanced classes. The F1-score is calculated using Equation 4.4:

$$F1 - Score = 2 \, x \frac{Precision \, x \, Recall}{Precision \ + \ Recall}$$

Equation 4.4 F1-score

F1 Score is the harmonic mean of precision and recall, further balancing the trade-off of the two. Varoquaux and Colliot (2023), in the context of a multiclassification, the F1 score can be calculated for each class and averaged. Furthermore, the researchers indicate that this metric is particularly valuable in healthcare, where false negatives and false positives can lead to critical misdiagnosis.

e) ROC Curve

The ROC curve plots the true positive rate against the false positive rate. A one vs all approach can generate a multiclass classification ROC curve for each class. The area under the curve, namely AUC, provides a single measure of model performance, though macro averaging may overrepresent rare classes if not weighted correctly. In the healthcare context, it can be used to evaluate diagnostic models that distinguish between multiple diseases or patient outcomes and further inform clinical decision-making by identifying thresholds that minimise false positives/negatives for critical conditions (Varoquaux and Colliot, 2023).

The process involved an iterative refinement, exploring various categorical features and techniques, such as SMOTEEN, which is explained in Section 4.2.1.9 in detail, stacking, ensembling, and threshold adjustment, to improve model accuracy and achieve balance. Lastly, the results were analysed, interpreted, and visualised. Figure 4.16 shows the entire experimental development process.

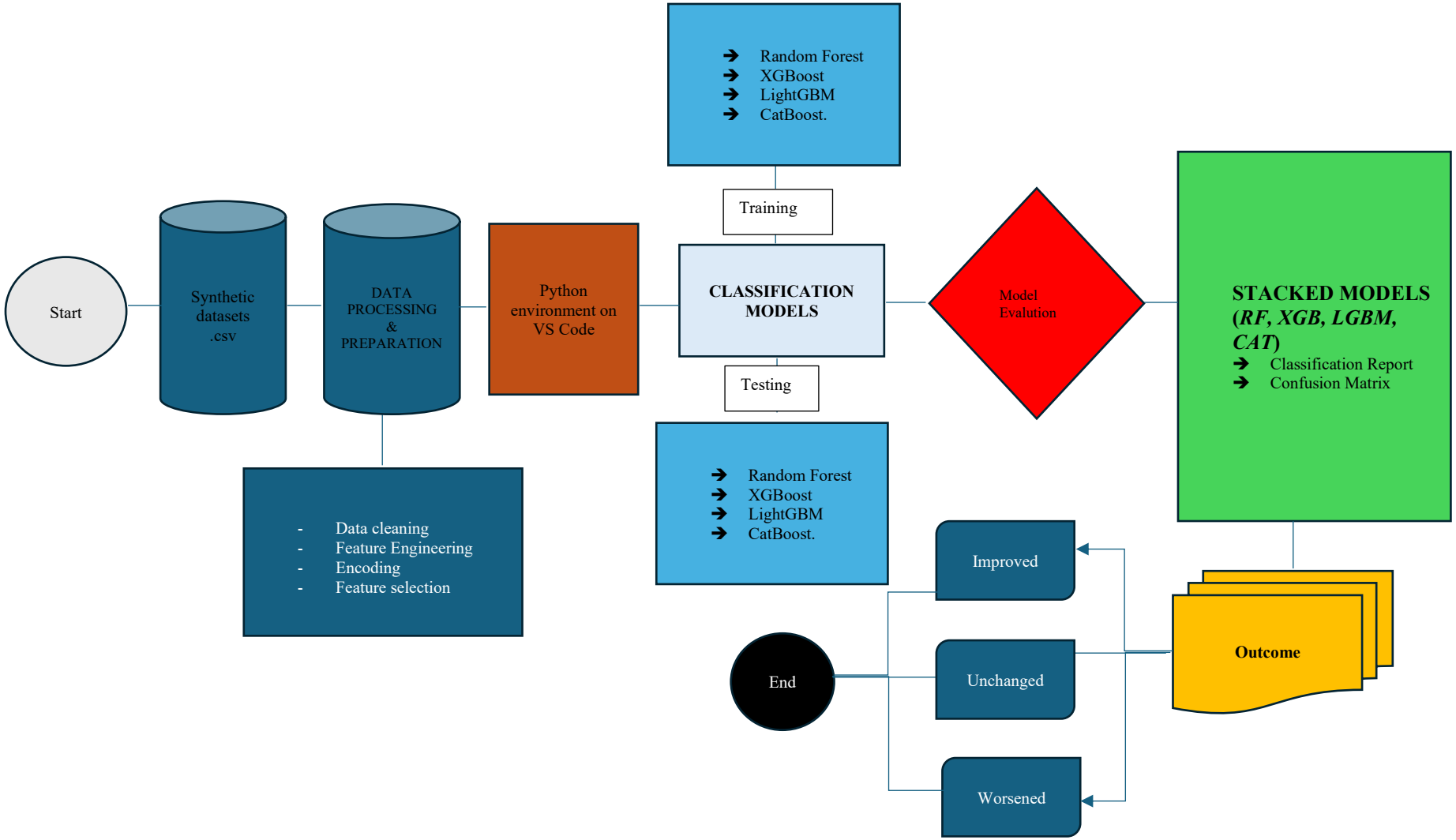


Figure 4.16 Model Experimental Development Process

4.3 Model User Interface Development: Implementation and Code

4.3.1 Streamlit

"Streamlit is a powerful and user-friendly open-source Python library that makes it easier to build interactive web applications for machine learning" (GeeksforGeeks, 2020). In this work, Streamlit was used to develop an interactive online application that visualizes and interprets the predictive model for healthcare outcomes, allowing users to investigate the classifier's performance and feature relevance (via SHAP values) in real-time. This was significant because it enhanced the study's practical utility during the CRISP-DM Deployment phase, allowing healthcare stakeholders to interact with the model's insights without requiring technical expertise. This improvement facilitated better decision-making and aligned with the research goal of delivering a user-friendly, interpretable solution.

Furthermore, Streamlit's quick prototyping capacity facilitated iterative development, guaranteeing that the application could be updated based on feedback, thereby increasing the study's legitimacy and usefulness in a healthcare environment. The full code for Streamlit can be accessed in Appendix B. We ran a code to develop the user interface landing page. Figure 4.20 illustrate the command used to run the user interface, which is executed by running the streamlit run app.py command in the terminal. The figure below shows the running Streamlit app on the browser; in this case, our default browser is Chrome, and it is running on localhost.

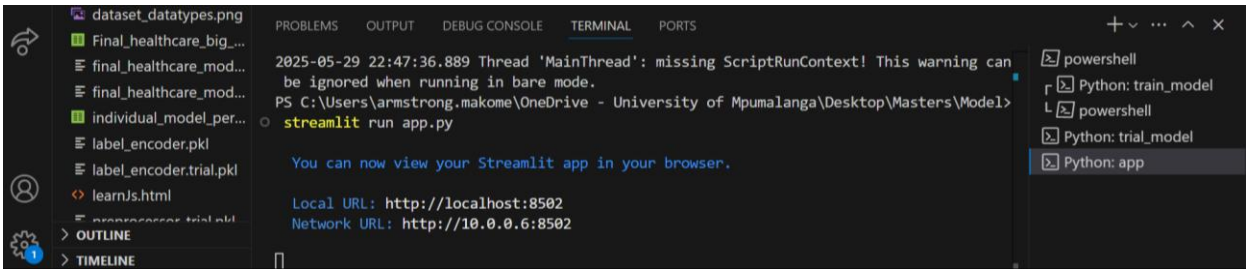


Figure 4.17 Running streamlit on localhost

Figure 4.21 depicts the running app on the browser.

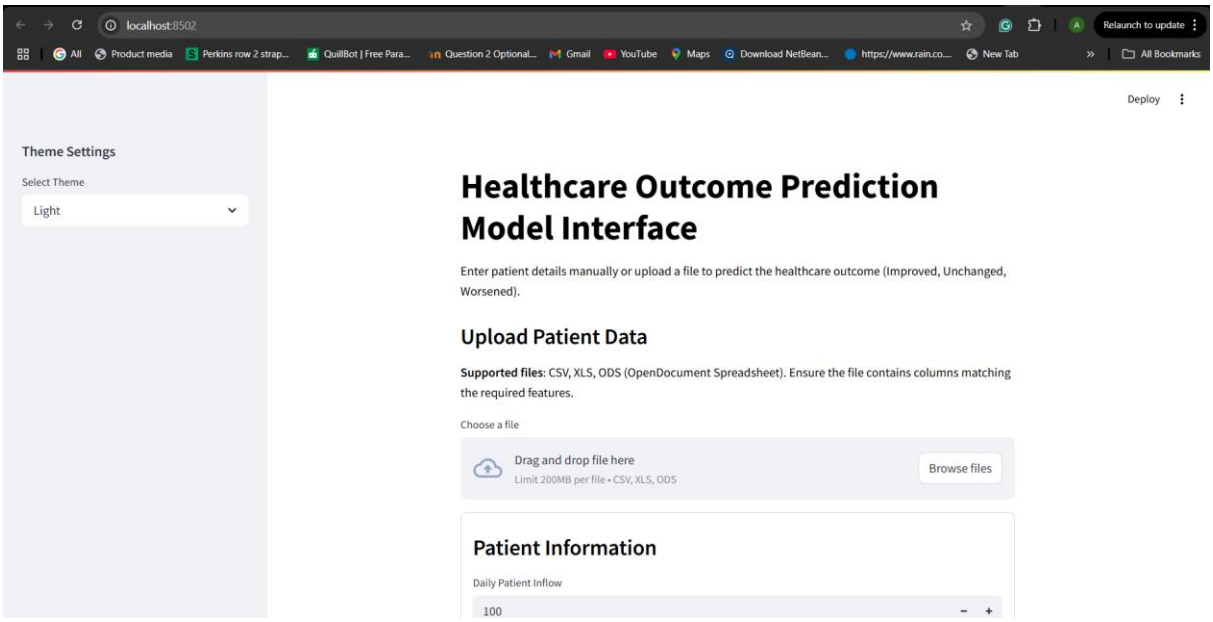


Figure 4.18: Running user interface

4.4 Chapter Summary

This chapter detailed the design and implementation of a healthcare model. It began with the experimental setup, which outlined the model's design, process flow, and the necessary hardware and software requirements. The implementation section focused on data preparation and model development, utilising evaluation metrics such as accuracy, precision, recall, F1 score, and the ROC curve to assess the classifier's performance and inform the final model selection. Finally, the chapter discussed the development of an interactive user interface using Streamlit, showcasing the practical deployment of the model.

CHAPTER FIVE: RESULTS AND DISCUSSION

This chapter presents the results of the tested model. The evaluation aimed to test and evaluate the enhanced algorithm's prediction effectiveness and efficiency in South Africa. Thus, present the results of the stacking classifier model, analyse its performance, and discuss its implications for healthcare outcome prediction. In so doing, answering the fifth (5^{th}) objective of this research, which is to test and evaluate the effectiveness and efficiency of the model. The results of the model are presented in Section 5.1. The explanation of the classification report for the models is covered in Subsection 5.1.1, and the explanation of the confusion matrices along with their accuracy is covered in Subsection 5.1.2. The explanations of the model's effectiveness and efficiency are covered in Subsections 5.1.3 and 5.1.4, respectively. The analysis of the ROC curve is covered in Subsection 5.1.5. The comparisons in similar work and the advantages of the model are discussed in Section 5.2, and lastly, the chapter summary is in Section 5.3

5.1 Results

After training the model, test runs were conducted to evaluate its effectiveness in classifying patient outcomes. The algorithms were all trained on the training set and tested using a test set of data, which had the same dataset size for all.

A classification report was used to analyse model performance, providing numerous important evaluation metrics. These comprised the accuracy, precision, recall, F1-score, and the area under the Receiver Operating Characteristic (ROC) curve (AUC). These criteria were used to assess how well each model performed in accurately predicting the result categories and to facilitate a comparison of the various algorithms. The results from this evaluation are reported in the following sections.

5.1.1 Classification Report of the Algorithms

The classification report is a tool used in machine learning to evaluate the performance of a classification model, as supported by Kharwal (2021). Vujovic (2021) emphasises the use of classification reports to quantify the quality of predictions made by a classification algorithm.

The classification report includes precision, recall, F1 Score, Support, and overall accuracy of the model. In this section, each model classification report is explained individually, which includes random forest, CatBoost, XGBoost, LightGBM, and a stacked ensemble classifier.

5.1.1.1 Random Forest classification report

Table 5.1 Random Forest evaluation

	Precision	**Recall**	**F1-Score**	**Support**
accuracy			0.70	18616
macro avg	0.81	0.45	0.44	18616
weighted avg	0.77	0.70	0.64	18616

Table 5.1 presents the evaluation metrics of the Random Forest model, achieving an accuracy of 70%. In 18616 support (instances), 70% of them were correctly classified. The F1-score and recall suggest that the model was able to identify the correct instances and balance the metrics, indicating a good ability to classify the majority of instances correctly. However, the F1 Score and recall reveal that the model struggles to perform well across all classes, particularly for the minority classes. These results indicate that while the model performs reasonably well, it does not equally balance performance across all classes.

5.1.1.2 CatBoost classification report

Table 5.2 CatBoost evaluation

	Precision	Recall	F1-Score	Support
accuracy			0.55	18616
macro avg	0.50	0.61	0.49	18616
weighted avg	0.64	0.55	0.58	18616

Table 5.2 presents the evaluation metrics of the CatBoost model, achieving an accuracy of 55%. In 18616 support (instances), 55% of them were correctly classified. While the model exhibits moderate performance, the metrics reveal imbalances in its performance across all classes. The F1-score and the recall suggest that the model has a relatively better capacity to identify instances from all classes, including the minority ones. However, struggles to maintain consistent precision across them. The precision indicates that many of the predicted positive cases are incorrect, which may result in higher false positives. This may indicate that, while the model performs better on the majority class, it still faces challenges in accurately classifying minority categories.

5.1.1.3 XGBoost classification report

Table 5.3 XGBoost evaluation

	Precision	Recall	F1-Score	Support
accuracy			0.82	18616
macro avg	0.73	0.58	0.60	18616
weighted avg	0.81	0.82	0.80	18616

Table 5.3 presents the evaluation metrics of the XGBoost model, achieving an accuracy of 82%. In 18616 support (instances), 82% of them were correctly classified, indicating a strong overall classification performance. The F1-Score and Recall suggest that the model has improved its ability to identify correct instances across all classes, including

the minority ones. Furthermore, the F1-score and recall confirm that the model performs consistently well across the dataset, particularly favouring the majority classes while maintaining a reasonable balance across all classes. These improvements over the other models indicate that the enhancements made, potentially through dataset balancing or tuning algorithms, have increased the model's ability to generalise and correctly classify more instances without compromising the performance of the minority classes.

5.1.1.4 LightGBM classification report

Table 5.4 LightGBM evaluation

	Precision	**Recall**	**F1-Score**	**Support**
accuracy			0.53	18616
macro avg	0.50	0.54	0.36	18616
weighted avg	0.63	0.53	0.46	18616

Table 5.4 shows the evaluation metrics of the LightGBM model with an accuracy o Table 5.4 presents the evaluation metrics of the LightGBM model, achieving an accuracy of 53%. In 18616 supports (instances), 53% of them were correctly classified. However, the model shows a lower performance, particularly in the F1-score, indicating its difficulty in treating all classes equally. With the help of precision and recall, the model performs slightly better on the majority of classes but still struggles to accurately balance the metrics across all classes.

5.1.1.5 Stacked classifier classification report

Table 5.5 Stacked classifier evaluation

	Precision	Recall	F1-Score	Support
accuracy			0.82	18616
macro avg	0.73	0.58	0.60	18616
weighted avg	0.81	0.82	0.80	18616

Table 5.5 presents the evaluation metrics of the stacked ensemble classifier model, achieving an accuracy of 82%. In 18616 support (instances), 82% of them were correctly classified. This indicates that the model correctly classified a significant proportion of the test data. F1-score and Recall results suggest a strong and consistent performance, especially on the majority classes. Results reveal that the model performs relatively well across all classes, including the minority ones. These results indicate better balance performance than the previous models, such as Random Forest, CatBoost, LightGBM, and XGBoost, which exhibited more disparity between precision and recall across classes.

Notably so, these metrics reveal that the model maintains high performance even when accounting for class imbalance. The stacked classifier combines the strengths of multiple base model learners, allowing it to generalise better and reduce overfitting on individual class distributions.

5.1.2 Confusion Matrix of the Algorithms

In this section of our work, each confusion matrix of the algorithms was explained briefly.

5.1.2.1 Random Confusion Matrix

Figure 5.1 below displays the confusion matrix for the Random Forest classifier generated for a three-class patient outcome prediction problem - Improved, Unchanged, and

Worsened. The matrix provides insight into the model's ability to correctly differentiate between the classes, highlighting areas of strength and limitation.

- In the case of the Improved class, only 9 instances were correctly predicted, while 83 were misclassified as Unchanged, and a significant 1030 were incorrectly predicted as Worsened. This reveals that the model struggles significantly to identify instances from the Improved class, resulting in very low recall and precision for this category.
- For the Unchanged class, 2220 instances were correctly classified. Still, 4352 were misclassified as Worsened, demonstrating a high false negative rate. Even though the model performs better here than with the Improved class, its predictive impact remains limited due to a high overlap with the worsened class.
- The Worsened class demonstrates the best performance, with 10838 instances correctly classified, and only 82 misclassified as Unchanged. This indicates a strong bias in the model's prediction toward the "Worsened" outcome, likely due to class imbalance or insufficient learning of patterns in the minority class.

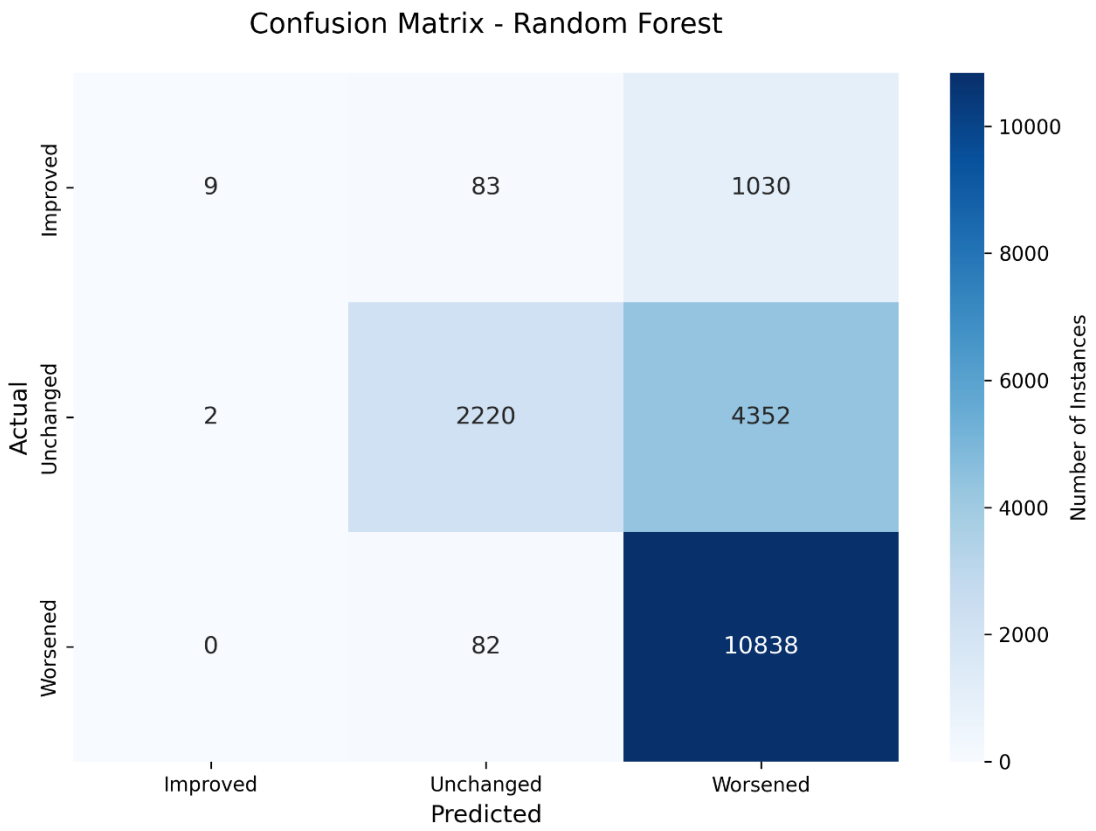


Figure 5.1 Random Forest Confusion Matrix

The matrix demonstrates that the Random Forest model is significantly biased towards accurately detecting the predominant class, Worsened, while exhibiting subpar performance on the minority classes, Improved and Unchanged. This imbalance indicates the previously noted poor macro average F1-score and recall in the assessment measures, highlighting the necessity for further data balancing or model calibration to increase classification fairness and reliability across all class categories.

5.1.2.2 CatBoost Confusion Matrix

Figure 5.2 depicts the CatBoost classifier's confusion matrix, which highlights its performance across three patient outcome classes: improved, unchanged, and worsened. This matrix gives further information on how effectively the model differentiates between real and expected categories.

- In the Improved class, the model predicted 903 occurrences correctly, whereas 162 were misclassified as Unchanged and 57 as Worsened. This demonstrates a significant classification performance for this minority class as compared to the Random Forest model, which previously demonstrated weak recall in this category. The comparatively high true positive count and low false positives indicate superior accuracy and recall.
- 2889 instances of the Unchanged class were successfully categorised. However, 1582 were mistakenly forecasted as improved, and 2103 as worsened. While the model exhibits adequate sensitivity for this
- In the Worsened class, 6440 cases were correctly identified; however, 20244 were incorrectly classified as Improved and 2236 as Unchanged. Despite being the majority class, the model does not dominate it, as the Random Forest model does. The distribution of mistakes points to more balanced learning.

Therefore, the CatBoost model outperforms Random Forest in terms of classification balance across all classes, with substantially higher recall for the Improved class and more fair performance overall. However, the model remains confused between Unchanged and Worsened, which might be owing to overlapping feature distributions or inadequate class separation in the dataset. These results are consistent with the macro-average F1-score of 0.49 and recall of 0.61, indicating acceptable but imperfect class-wise performance.

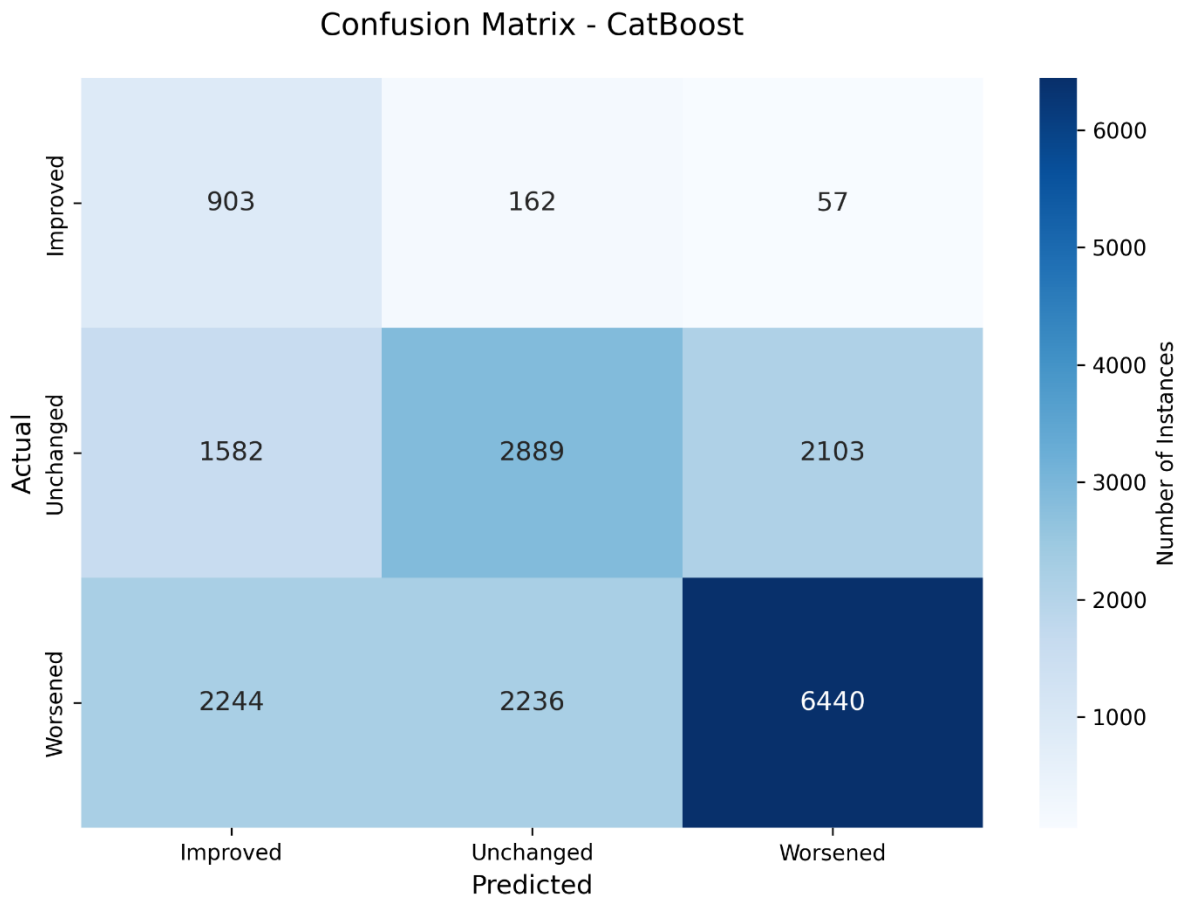


Figure 5.2 CatBoost Confusion Matrix

5.1.2.3 XGBoost Confusion Matrix

Figure 5.3 shows the XGBoost classifier's confusion matrix, which evaluates its classification performance on three patient outcome categories: Improved, Unchanged, and Worsened. The matrix gives information on how well the model distinguishes between different classes.

- The model properly categorized 87 occurrences in the Improved class, while 363 were incorrectly classified as Unchanged, and a substantial 672 as Worsened. While outperforming models such as the Random Forest model in this class, the

result still reveals a challenge in accurately detecting minority class occurrences, notably owing to overlap with the dominating class.

- This demonstrates a rather good recall for the Unchanged class, while there is still some confusion with the Worsened class, which may indicate semantic or feature similarities between these two labels.
- The Worsened class performs the best, with 10480 accurate predictions and just 402 misclassifications, compared to 38 as Unchanged and 0 as Improved.
- This demonstrates XGBoost's high prediction power and sensitivity for the majority class.

The confusion matrix demonstrates that XGBoost outperforms in the dominating Worsened class and exhibits decent generalization in the Unchanged category; however, it underperforms in the Improved category, a frequent pattern in unbalanced healthcare datasets. The error distribution is consistent with typical issues in multi-class classification, where models often favor the majority class until explicitly corrected by balancing or threshold adjustment. These findings are consistent with predicted evaluation metrics, since weighted averages are likely to show strong accuracy and recall, whereas macro averages, particularly for precision and F1, may reveal poorer performance on underrepresented classes.

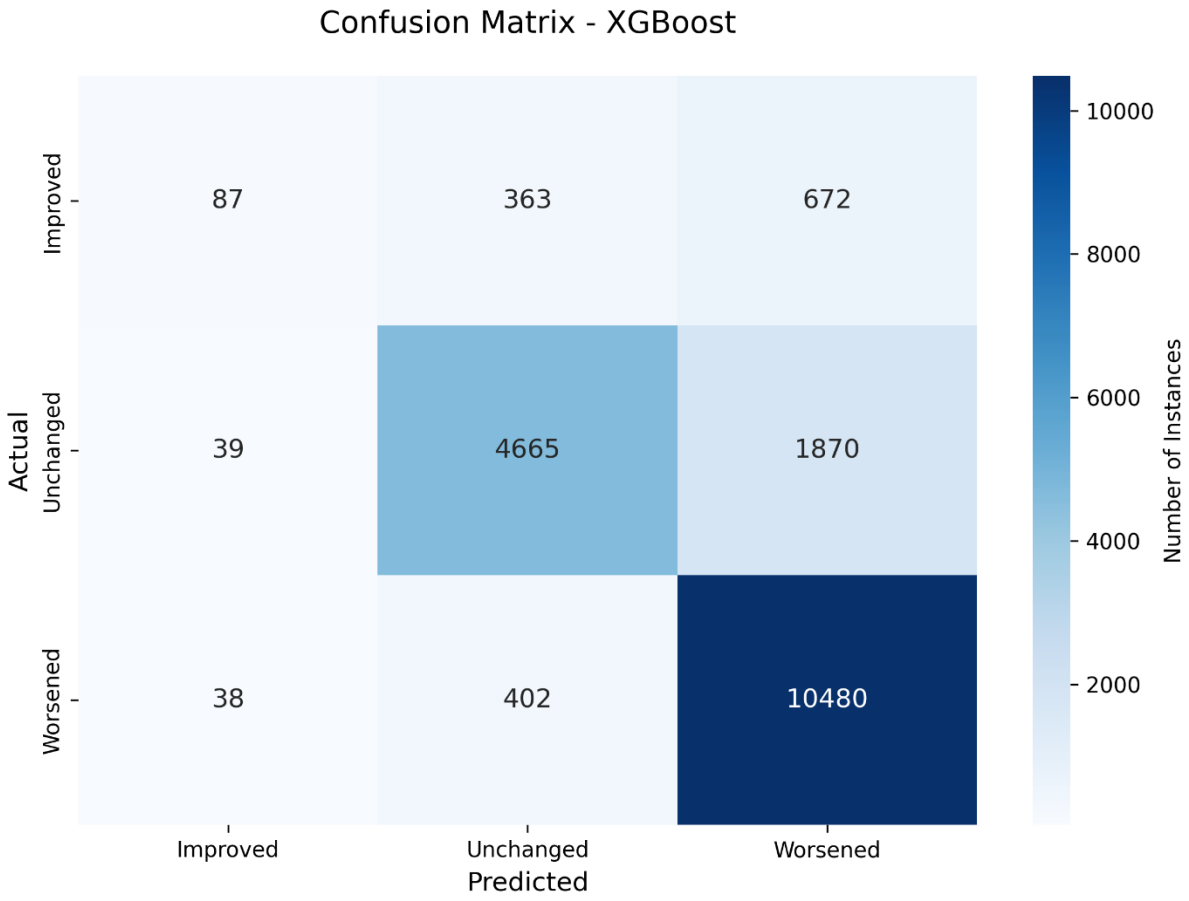


Figure 5.3 XGBoost Confusion Matrix

5.1.2.4 LightGBM Confusion Matrix

Figure 5.4 illustrate the LightGBM classifier's confusion matrix, which demonstrates its classification performance across three patient outcome categories: Improved, Unchanged, and Worsened. The matrix illustrates the model's ability to distinguish between classes and highlights areas of strength and potential misclassification.

- In the Improved class, the model properly identified 888 instances, whereas 18 were incorrectly classified as Unchanged and 216 as Worsened. This performance demonstrates a notable recognition of this minority class, with high recall and minimal misclassification, outperforming numerous other models in accurately recognizing this outcome.

- In the case of the Unchanged class, there were 4635 correct predictions, with 1735 incorrectly labelled as Improved and 204 as Worsened. While the model retains a high true positive rate, the number of instances misclassified as the Improved class is significant, which may imply feature similarity or decision boundary overlap between these two classes.
- In the case of the Worsened class, 8723 cases were correctly identified, whereas 2116 were incorrectly classified as Improved and 81 as Unchanged. Despite being the majority class, the relatively large proportion of misclassifications into the Improved class implies that the model is either over-predicting this label or failing to differentiate severe outcome indicators.

LightGBM exhibits balanced and robust classification performance, with relatively good recall and precision across all classes. It demonstrates enhanced generalisation to the minority Improved class while maintaining good accuracy in the majority worsened class. However, the misclassification of Unchanged and the other two classes, Improved, suggests that more refinement, such as advanced feature engineering or threshold adjustment, may improve class separability.

These results are consistent with a high macro-average recall and F1-score, demonstrating the model's capacity to perform well not just worldwide, but also inside each class separately.

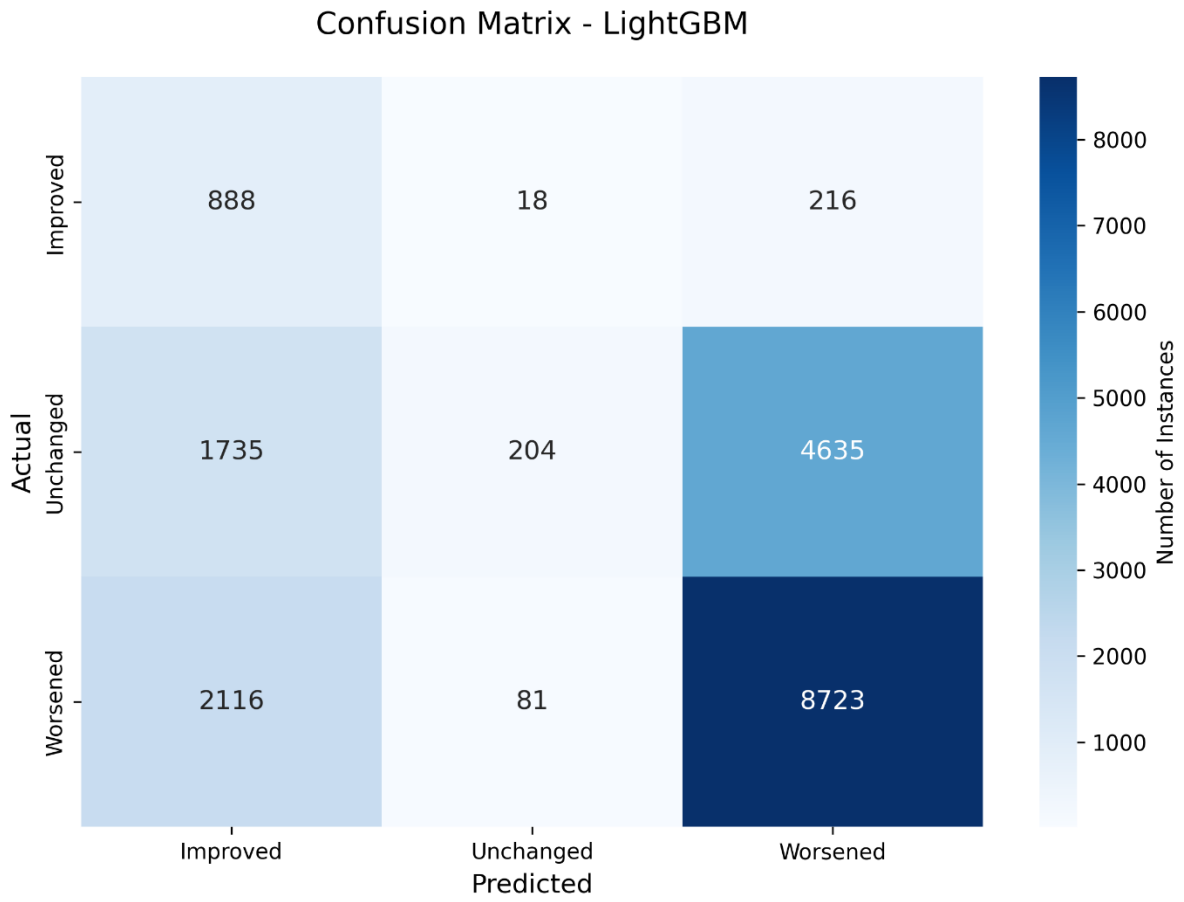


Figure 5.4 LightGBM Confusion Matrix

5.1.2.5 Stacked classifier Confusion Matrix

Figure 5.5 illustrates the confusion matrix of the stacking classifier after threshold adjustment, demonstrating its predictive accuracy across three outcome classes: Improved, Unchanged, and Worsened. The matrix shows a well-calibrated model with better sensitivity balance across all classes, highlighting the impact of threshold adjustment on class discrimination.

- In the case of the Improved class, the model properly categorised 686 instances, with 169 misclassified as Unchanged and 267 as Worsened. This represents a significant improvement over previous models, including Random Forest, CatBoost, LightGBM, and XGBoost. The assembled classifier has enhanced sensitivity to this minority class, implying that threshold adjustment was helpful in improving the model's capacity to recognise fewer common outcomes.
- The model correctly predicted 5159 occurrences in the Unchanged class, whereas 681 were incorrectly categorised as Improved and 734 as Worsened. These findings demonstrate a high recall and balanced misclassification, demonstrating that the model is well identifying this intermediate category.
- In the Worsened class, 9537 cases were accurately identified, with 600 misclassified as Improved, and 783 as unchanged. While performance is significantly lower than in prior models that greatly favoured this majority class, the trade-off has resulted in a more balanced overall classifier, enhancing fairness across all classes

The threshold-adjusted stacking classifier performs the most balanced and reasonable of all the models tested. While it marginally reduces precision in the majority class, it also significantly increases recall and precision in the Improved and Unchanged classes. This makes it ideal for healthcare applications in which precise identification of minority outcomes, such as patient improvement, is just as critical as the majority result.

The confusion matrix supports previous metric evaluation that showed greater macro average recall and F1-scores, demonstrating that threshold adjustment may be an effective strategy for minimising class imbalance effects and improving real-world model applicability.

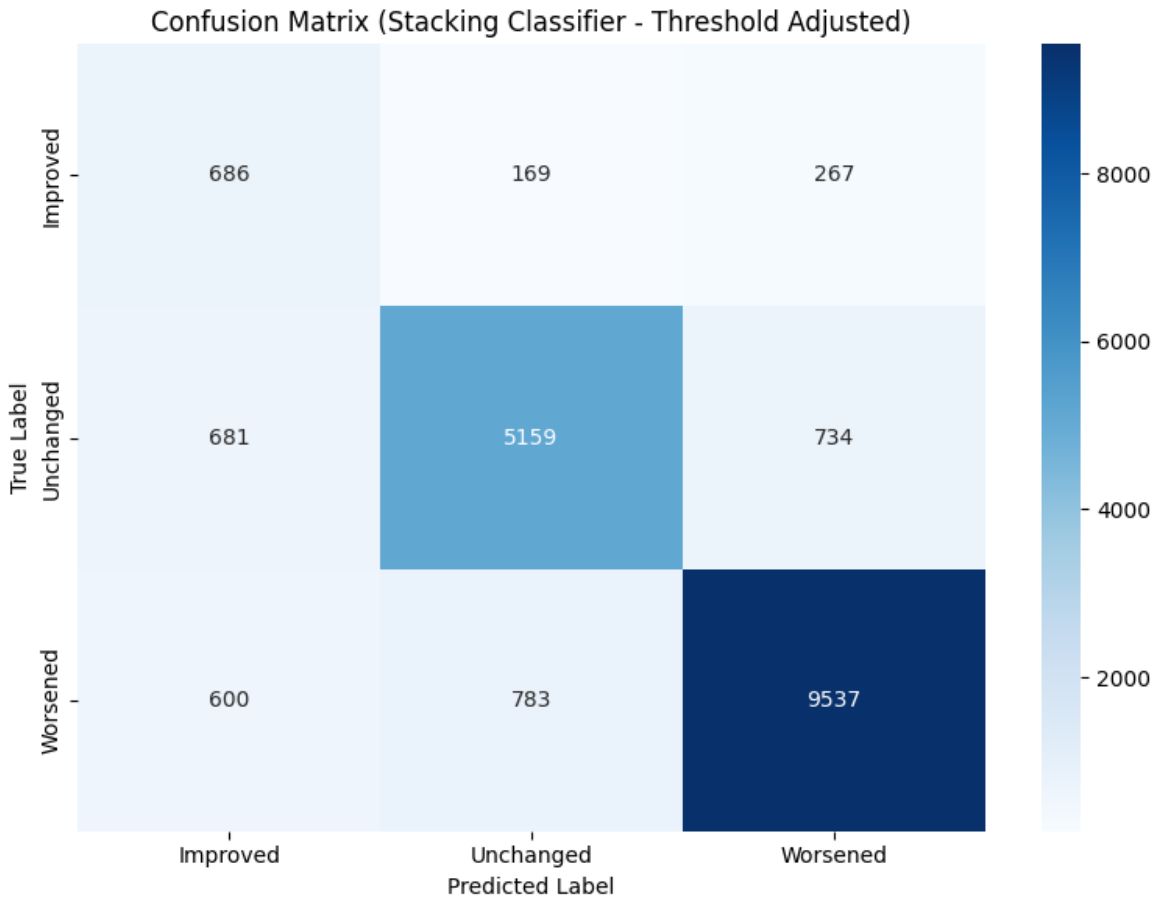


Figure 5.5 Stacked classifier confusion matrix

5.1.3 Effectiveness of the model

This section presents a comparative analysis of the classification algorithms used in this study. The study compared the effectiveness of the models and tabulated the results in Table 5.6 to better understand their metrics. The findings were further represented in a graphical representation for improved understanding. To enhance interpretability and enable clear comparison of each model's strengths and limitations across different outcome classes.

5.1.3.1 Algorithm comparisons table

Table 5.6 Model classification comparisons

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.702	0.77	0.702	0.644
XGBoost	0.818	0.807	0.818	0.796
LightGBM	0.527	0.626	0.527	0.457
CatBoost	0.55	0.644	0.55	0.578
Stacking Classifier	0.84	0.826	0.84	0.83

Table 5.6 shows the summary of all models' accuracies and evaluation metrics.

5.1.3.2 Algorithm comparisons bar graph

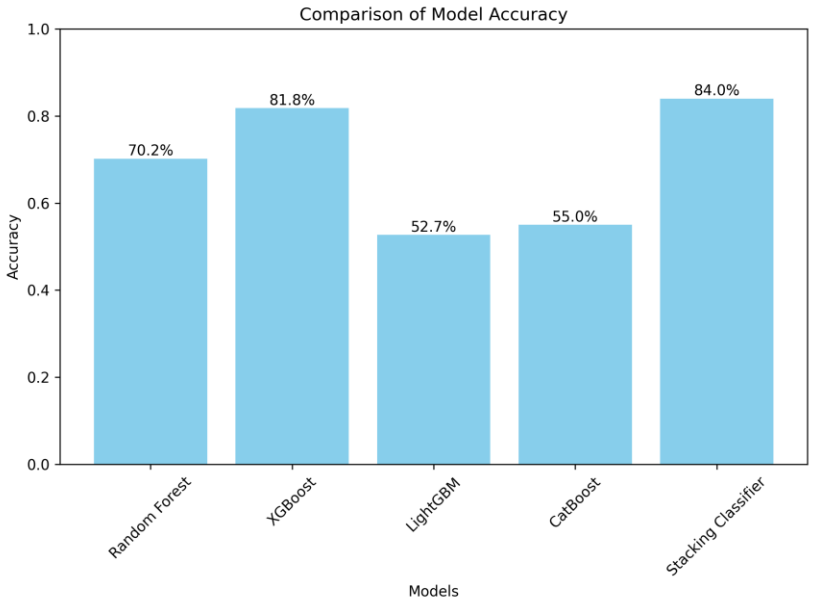


Figure 5.6 Bar graph showing models' accuracy comparisons

Figure 5.6 illustrates the accuracy of five different algorithms in the model: Random Forest, XGBoost, LightGBM, CatBoost, and the stacked classifier. The Stacking classifier achieved the highest accuracy at 84%, followed by XGBoost at 81.8%, CatBoost at 55%, and LightGBM at 52.7%, which had the lowest accuracy.

5.1.4 Efficiency of the model

In this section, the study presented and compared the computational times of the algorithms, which were a crucial factor in selecting the final model.

5.1.4.1 Model computational times comparisons

Table 5.7 Model computational times comparisons

Model	Computational times (seconds)
Random Forest	147.205
XGBoost	65.517
LightGBM	1.831
CatBoost	29.583
Stacking Classifier	3485.762

Table 5.7 shows the computational times in seconds of five different algorithms in the model: Random Forest, XGBoost, LightGBM, CatBoost, and the Stacking assembled classifier. The first one to have the least computational time is LightGBM, with 1.831 seconds, followed by CatBoost with 29.583 seconds, then XGBoost with 65.517 seconds, then Random Forest with 147.205 seconds, and the highest computational time is the Stacking classifier with 3,485.762 seconds.

5.1.5 ROC Curves

The ROC Curves were used to assess the classifier's capacity to discriminate between 'Improved', 'Unchanged', and 'Worsened' healthcare outcomes by displaying the True Positive Rate versus False Positive Rate. They optimise thresholds and offer AUC values to enable strong, reliable predictions inside the CRISP-DM framework. This contributes to the quantitative method by assessing model performance and aligns with the study objective of credible healthcare outcome analysis. ROC curve visually shows model performance at various thresholds and further evaluates how well the classifiers differentiate between classes (Bhandari, 2020). ROC graphs are widely used in medical decision-making (Fawcett, 2006)

Figure 5.7 presents the classification model's multi-class ROC (Receiver Operating Characteristic) curves, illustrating its discriminatory ability across the three outcome classes: Improved, Unchanged, and Worsened. The ROC curve plots the True Positive Rate (sensitivity) against the False Positive Rate at various threshold levels, with the Area Under the Curve (AUC) serving as a summary measure of the model's performance.

- The AUC for the Improved class is 0.93, which indicates a high capacity of the model to distinguish between the Improved class and the others. Despite being a minority class in the dataset, this strong AUC score suggests that the model was effectively trained to identify instances of patient improvement.
- The Unchanged and Worsened classes got an AUC of 0.95, indicating that the model performed well in distinguishing these classes. The ROC curves for these classes are closely related to the top-left corner of the graph, indicating a high true positive rate with few false positives.

The ROC curves indicate that the classification model, most likely the threshold-optimized stacked ensemble, exhibits robust and balanced prediction capabilities across all three classes. The curves exhibit low variation from the ideal performance line (top-left), indicating that the model performs consistently in terms of overall accuracy and class-wise discrimination.

This performance level is crucial in healthcare outcome prediction, where the precise classification of minority classes, such as Improved, is vital for equitable and effective clinical decision-making.

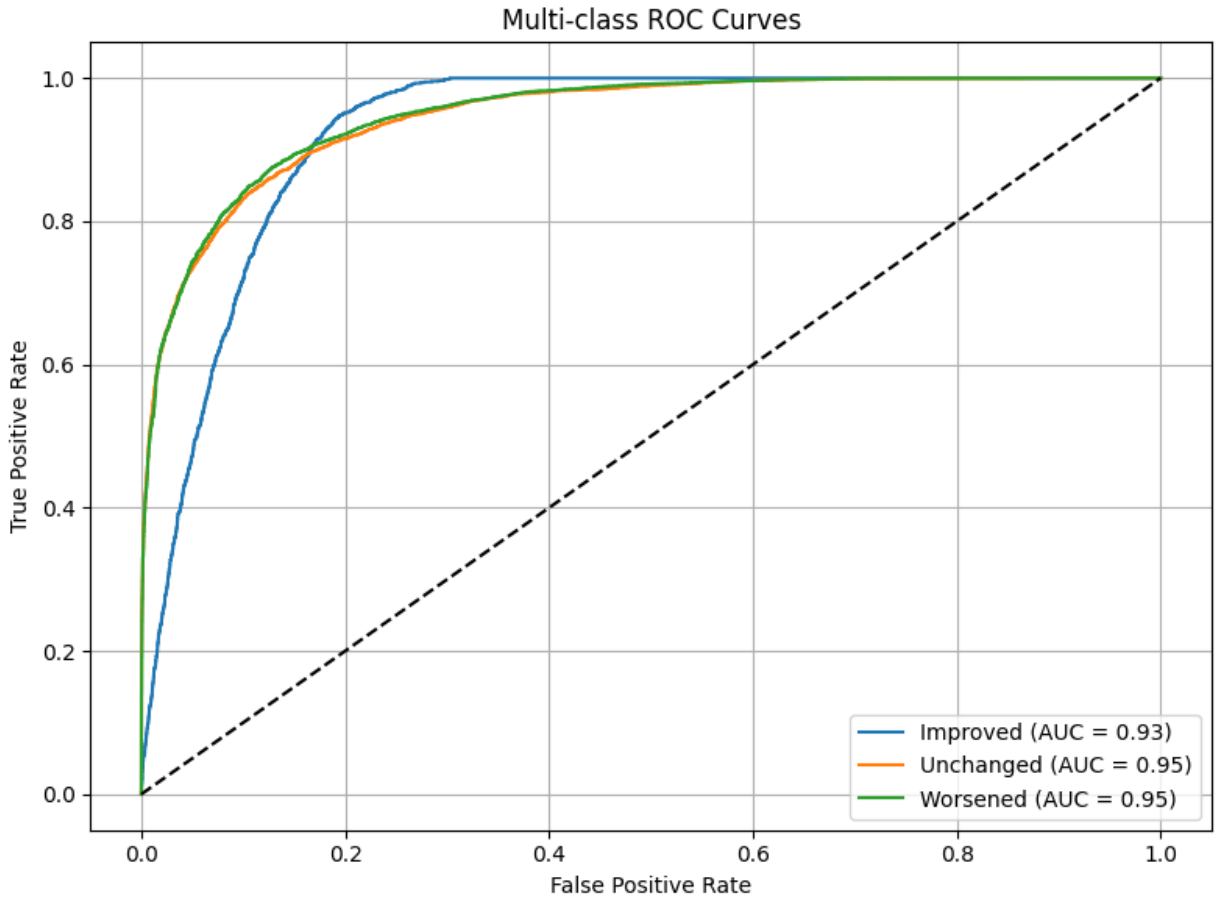


Figure 5.7 Multiclass ROC Curves for the stacked classifier

5.1.6 Analysis of SHAP Features

*"*SHapley Additive exPlanations (*SHAP) explains the prediction of a data sample by quantifying the contribution of each feature to the algorithm's prediction"* (Rasheed et al., 2022). Lundberg & Lee (2017a) provided SHAP as a cohesive framework for interpreting intricate model predictions, attributing feature significance values using a novel class of additive measures and a distinctive solution with advantageous qualities. Therefore, in the context of our work, the SHAP feature was used to explain how an individual feature contributes to the model's prediction.

Figure 5.8 depicts the SHAP feature significance plot, which shows the average contribution of each feature to the model's prediction for each of the three outcome classes: Improved (Class 0), Unchanged (Class 1), and Worsened (Class 2). SHAP values give a consistent and interpretable measure of feature influence, reflecting both size and direction of impact on model outputs.

- Top Influential Features:
    - Previous_Visits and Visit_Frequency are the strongest indicators, especially for Class 0 (Improved). Their dominating blue bars indicate that patients with more frequent past hospital encounters are likelier to have better outcomes, maybe due to continuing treatment or tighter monitoring.
    - Satisfaction_Rating is likewise highly valued across classes, influencing forecasts for Class 1 (Unchanged) and Class 2 (Worsened). This is consistent with real-world expectations, since patient satisfaction might indicate treatment quality and participation, influencing recovery trajectories.
- Clinical and Operational Drivers:
    - Comorbidities_Yes, Age, and Treatment_Outcome_Unchanged all significantly contribute to Classes 1 and 2, demonstrating the model's sensitivity to patient health state and history.
    - Bed_Occupancy_Rate, Daily_Patient_Inflow, Staff_Count, and Equipment_Availability_Limited are all fairly relevant, especially in predicting "Worsened" outcomes (Class 2). These characteristics represent resource constraints and hospital load, both of which have been shown to influence healthcare delivery.

- Time and efficiency metrics:
    - Variables such as Emergency_Response_Time_Minutes, Length_of_Stay, Wait_Length_Interaction, and Log_Wait_Time all have a significant impact, particularly in Class 2, showing that greater delays and inefficiencies in care delivery are related to worsening patient conditions.

- Disease and Interaction Variables:
    - Features such as Disease_Category_Diabetes, Disease_Category_Other, and Resource_Bed_Interaction provide a small but significant contribution, demonstrating that individual diagnoses and resource utilisation patterns influence patient outcomes.

The SHAP analysis demonstrates that a combination of historical patient data, satisfaction-related factors, and hospital resource and efficiency indicators drives the model's findings. Notably, the model retains interpretability and fairness using a broad and clinically relevant feature set to differentiate between the outcome classes. This interpretability is critical for establishing confidence in the model's predictions in the healthcare environment.

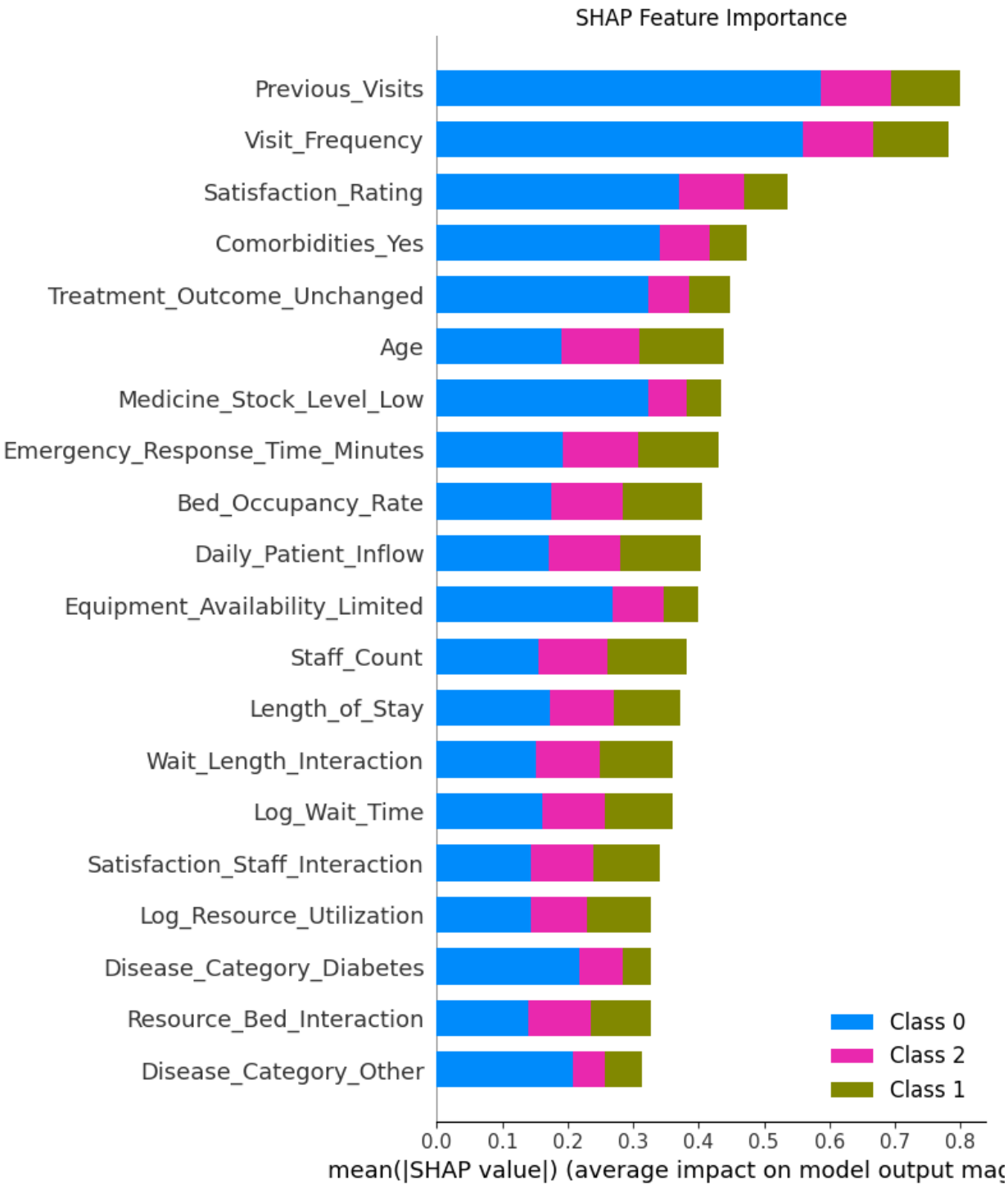


Figure 5.8 SHAP Feature analysis plot

Figure 5.9 illustrates the SHAP summary plot, which displays the distribution of SHAP values for the top features that contribute to the model's predictions. This graph illustrates the magnitude of each feature's influence on the model's output and direction, indicating whether it tends to push the prediction towards or away from a specific class outcome. The form and spread of the distributions indicate that the top characteristics exert constant and considerable effect, as seen by the larger violin-like shapes. The narrower bands in the figure represent characteristics with a minor or localised impact on the model's output.

Furthermore, this SHAP summary plot indicates that a few highly significant variables drive the model's decisions, and the influence of these factors changes between instances in a complex and frequently non-linear manner. The use of SHAP provides a transparent, interpretable tool crucial in the healthcare environment, as it demonstrates how much and in what way each feature influences predictions.

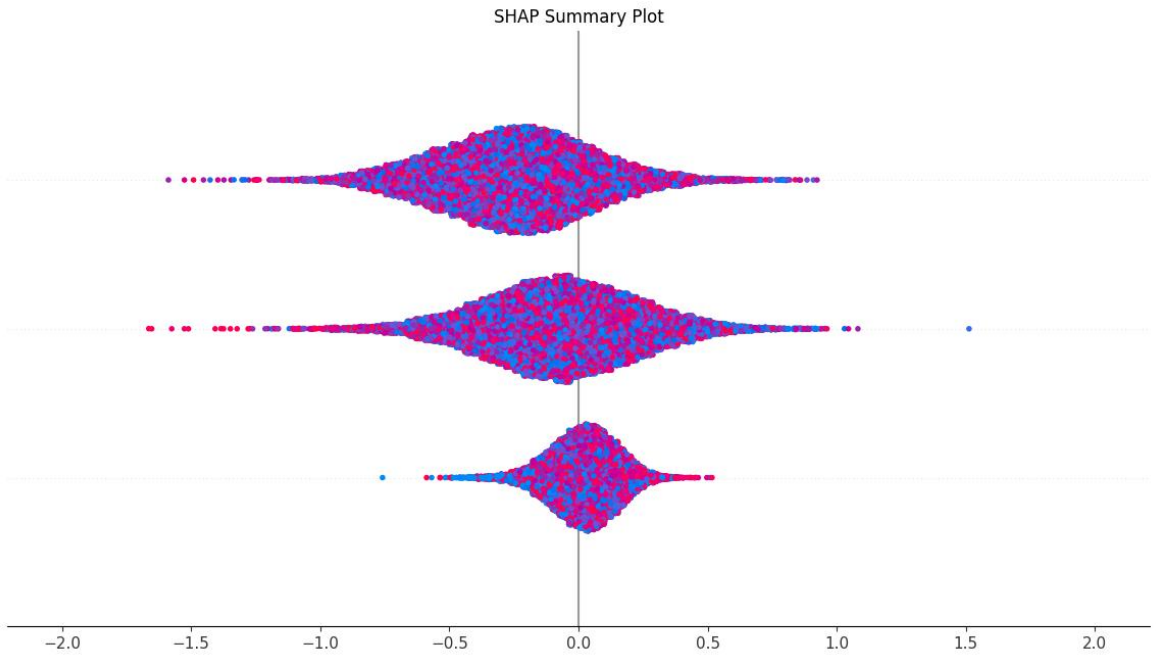


Figure 5.9 SHAP summary plot

5.2 Discussion of the results

5.2.1 Model accuracy

The XGBoost model reached an accuracy of 81%, which is notably higher than several studies. Mishra et al. (2018) utilized an assembled XGBoost and Gradient Boost Model to design a Decision Support System for healthcare prediction, focusing on detection, diagnosis, and treatment, with accuracies of 78% for XGB ensembled, 77.15% for GBM, and 76.85% for XGB-GBM ensembled. Due to the massive amount of data that is generated from the healthcare datasets and also the low quality of data, which results in poor model performance or low accuracy, as supported by (Budach et al., 2022), the nature of healthcare data (Salmi et al., 2024), model complexity, and improper algorithm selection (Wang et al., 2024). The high accuracy of the model is due to the proper cleaning, preparation, and processing of data before use (Shankar & Shewale, 2024).

In recent years, research has led to the development of various machine learning-based predictive models for healthcare outcomes. However, prior studies have primarily focused on comparing the prediction accuracy of different models, often selecting the one with the highest performance as the main outcome. Ensemble learning, a technique that combines predictions from multiple models to create a more robust and accurate model, offers several advantages. These include enhanced precision, reduced overfitting, greater resilience, adaptability across different algorithms, and improved stability.

Our current study focused on developing a predictive model for healthcare that achieves a balanced and robust outcome. To reach this goal, we employed a stacked classifier that includes Random Forest, XGBoost, LightGBM, and CatBoost, ultimately achieving an accuracy of 84%. This performance surpasses that of several previously mentioned algorithms.

For comparison, Chen et al. (2024) utilised an ensemble of Logistic Regression, Random Forest, Gradient Boosting, and a meta-learner (stacked) to predict mortality rates for cardiovascular diseases and Atrial Fibrillation, achieving an accuracy of approximately

72%. Similarly, Yan and Feng (2022) combined K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), XGBoost, and a meta-learner (MLP) to predict cancer survival, reaching an accuracy of 82.1% for gastric cancer and 83.3% overall, which are still slightly lower than our model's results.

Additionally, Ameen et al. (2025) implemented a stacked ensemble of SVM, KNN, Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Random Forest for multi-omics cancer classification, achieving an accuracy of approximately 81%, which is also lower than our findings.

5.2.2 Computational Efficiency of Stacking Model

The study on model accuracy and training time indicated a clear trade-off between predictive performance and computational cost. The Stacking Classifier proved to be the most accurate model with 84.0%, with computational times of 3485.762 - 5339.8 seconds of computation to give results that vary due to resources used, such as hardware and running software, illustrating the resource needs of an ensemble-based model. XGBoost was the best compromise, achieving 81.8% accuracy in just 65.517 seconds, in support of Nalluri et al. (2020) 's results, which show that XGBoost effectively balances speed and performance.

In many machine learning and medical informatics studies, the time required for training or inference is often overlooked. Studies tend to focus on predictive performance metrics such as accuracy, sensitivity, specificity, and AUC, which are more closely related to clinical decision-making and publication standards (de Hond et al., 2022). Reporting runtime can be challenging, as computational time varies significantly depending on hardware configurations, software applications, and dataset size (Zheng et al., 2025). This variability makes such measures less reproducible or comparable across different domains.

Additionally, in early feasibility studies, researchers often prioritise showcasing methodological innovations or improvements in predictions over operational considerations. These operational aspects are typically addressed in later translational or implementation-focused research. As a result, the lack of timing data reflects a disciplinary norm that values the validity and generalizability of the model more highly than computational efficiency unless real-time application is a primary research focus (Topol, 2019).

5.2.3 Advantages of the model

Our stacked ensemble classifier model is versatile and robust, supporting various algorithms that allow for easy evaluation and selection based on available metrics. The examination is comprehensive, encompassing both standard accuracy metrics and confusion matrices. In a healthcare environment where precision is paramount, this capability ensures better control over false positives and negatives, potentially improving patient outcomes by providing more reliable diagnostic decision-making.

Furthermore, our model-enhanced performance is a critical advantage in the healthcare environment. The combination of the strengths of multiple algorithms reduces bias and variance, delivering more accurate and reliable predictions.

5.3 Chapter Summary

This chapter gave the outcomes of the tested model. The model classification report was evaluated for each model, considering the accuracy, precision, recall, and F1 score. The study included a confusion matrix for all the algorithms. The model's efficacy and efficiency were explained, which addressed the research's fifth objective. Also performed additional analysis using the ROC Curve and SHAP analysis. Additionally, I performed further analysis using the ROC Curve and SHAP analysis. Lastly, the study reviewed the comparison in terms of accuracy and training time, followed by comparisons with comparable work and the merits of my model in the chapter summary. The following chapter discusses the conclusion, recommendations, and future work.

CHAPTER SIX: CONCLUSION AND FUTURE WORKS

This chapter reflects on the research conducted from problem identification and motivation in Chapter One to the results presented in Chapter Five. It emphasises how each objective and research question were addressed, ultimately providing a general conclusion for each finding. The study, grounded in computer science and data science, undertook the challenge of enhancing healthcare service delivery in South African public hospitals by utilizing Big Data analytics and machine learning. The chapter is organised as follows: Section 6.1 provides a recap of the research aim, Section 6.2 provides a summary of each chapter, highlighting the objectives and research questions answered, Section 6.3 offers overall conclusions regarding the model and the study; and Section 6.4 presents recommendations for future work related to the model.

6.1 Recap of the Aim

The primary aim of this study was to design and implement a big data analytics model that can support decision making relevant to service delivery in South African public hospitals, and to demonstrate the technical feasibility of the approach using a synthetic dataset that mimics the healthcare domain. In this sense, the study's contribution is best understood as a working prototype that illustrates how large scale healthcare variables can be structured, analysed, and used for multiclass outcome prediction, while recognising that practical impact claims require further validation on real hospital data.

6.2 Achievement of Research Objectives

This section highlights the chapters and sections where each objective was answered partially or fully.

First Objective: To identify factors influencing Big Data analytics in the South African public healthcare sector.

This objective was answered fully in the literature review in Chapter Two. In the study's experiment, these factors were tested, in which feature significance was established using SHAP analysis on a dataset. The analysis, conducted during the CRISP-DM Data Preparation and Modeling phases, identified key influencing factors, including Satisfaction_Rating, Age, Log_Wait_Time, and Satisfaction_Staff_Interaction, as the

most important features. In contrast, Daily_Patient_Inflow and Length_of_Stay were deemed the least important. These features, obtained using the Stacking Classifier with Random Forest and XGBoost, highlight characteristics crucial for predicting healthcare outcomes, allocating resources, and managing patients in South African public hospitals.

Second Objective: To assess the different methods used to analyse Big Data in South African public hospitals.

This objective was answered fully in Chapter Two by conducting a comprehensive literature review, which revealed a range of established and advanced methodologies for Big Data analytics in healthcare. Traditional methods included statistical analysis and cohort studies, but newer methods also utilized machine learning models, such as XGBoost and Random Forest. The review concluded that Random Forest is preferred for its ability to handle large datasets and provide robust classification, a finding supported by its use in this study. In contrast, Logistic Regression was noted for its effectiveness with binary outcomes, which aligns with the need to categorize patient improvement levels in the South African domain.

Third Objective: To use Big Data analytics techniques and tools to draw patterns and insights needed for improving service delivery in public healthcare.

The objective was answered in Chapter Four, during the experimental phase by utilising the CRISP-DM framework to apply Big Data analytics to the synthetic dataset. The analysis on the synthetic dataset illustrated how the proposed analytics workflow can surface associations and candidate patterns that may be useful for forming operational hypotheses. However, these patterns should be validated on real hospital datasets before any conclusions are drawn about practical bottlenecks or service improvements. Techniques such as feature engineering (e.g., Resource_Bed_Interaction) and SMOTEENN for balancing the dataset revealed trends, including the relationship between Wait_Length_Interaction and poor outcomes. This provided valuable insights into the bottlenecks in service delivery. Additionally, tools like Python (Pandas, Scikit-learn) and visualisation libraries (Matplotlib, Seaborn) helped identify these trends. These patterns should be interpreted as analytically derived signals within the synthetic dataset, which

can be used to formulate plausible hypotheses about operational bottlenecks that merit investigation in real hospital environments once suitable data access and approvals are in place (Pastorino et al., 2019; Favaretto et al., 2020).

Fourth Objective: To use machine learning (ML) classification techniques to develop a model of Big Data analytics that will be used to make better decisions and predictions needed for improved service delivery.

The objective was answered by training and testing several machine learning algorithms, including Logistic Regression, Random Forest, Support Vector Machine (SVM), XGBoost, CatBoost, and LightGBM, on a balanced dataset during the CRISP-DM modelling phase in Chapters Three and Four. However, due to factors such as training and computational time, the researchers decided to exclude Logistic Regression, which was more effective for linear problems than for classifying complex datasets like ours. SVM was also dropped due to its lengthy training and computational time.

The study focused on the remaining algorithms, which performed well and were suitable for our work. To build a robust model, the researchers opted for a Stacking Classifier that combines XGBoost, Random Forest, CatBoost, and LightGBM. This ensemble model emerged as the best performer, effectively managing multiclass classification problems and outperforming both Logistic Regression (which is not well-suited for multiclass problems) and SVM (which was impractical due to long training and computational times).

Overall, the algorithm selection and ensemble design demonstrate a feasible modelling strategy for multiclass prediction under the study's experimental conditions, with practical deployment remaining a future step that depends on real data evaluation and clinical workflow alignment (de Hond et al., 2022; Topol, 2019).

Fifth Objective: To test and evaluate the effectiveness and efficiency of the model.

This objective was answered during the evaluation phase of the CRISP-DM framework in Chapter Five, when the Stacking Classifier was thoroughly tested and validated using performance metrics. These results indicate promising predictive performance for the prototype on the synthetic dataset under the chosen evaluation design. The findings

should therefore be interpreted as evidence of technical feasibility, with further validation on real hospital data required before considering operational use.

These results indicate that the prototype can achieve strong predictive performance within the synthetic dataset and can serve as a feasibility baseline for subsequent external validation studies using real hospital data, where performance, reliability, and operational usefulness can be assessed more directly (de Hond et al., 2022; Zheng et al., 2025).

6.3 Overall Conclusions

The study demonstrated that a stacking ensemble combining XGBoost, Random Forest, CatBoost, and LightGBM can achieve strong multiclass predictive performance on the synthetic dataset used in this dissertation, with an accuracy of 84 percent. In practical terms, this supports the methodological conclusion that ensemble learning is a suitable candidate approach for modelling complex, non-linear interactions among service delivery related variables in the designed feature space. At the same time, the conclusions of this dissertation remain bounded to prototype feasibility. Because the model has not been trained or evaluated on real hospital information system data, the results do not provide evidence of realised improvements in hospitals, nor do they establish causal effects on patient care or operational performance. Instead, the contribution is a demonstrated pipeline and artefact that can be taken forward for empirical validation and context specific evaluation in future work (Hevner et al., 2004; Collins et al., 2015; Steyerberg, 2019).

Beyond predictive performance, the use of explainability analysis helped identify features that were most influential in the model's predictions, which strengthens interpretability and supports transparent discussion of what the model appears to learn. Importantly, such feature importance signals are not equivalent to confirmed causal drivers of service delivery outcomes. Rather, they provide an evidence-informed basis for further investigation, including confirmation on real hospital data and the assessment of whether the same relationships hold under clinical documentation practices, operational constraints, and data quality variation (Pastorino et al., 2019; Favaretto et al., 2020).

Accordingly, the main conclusion of this dissertation is that the proposed model constitutes a functional prototype that validates the end-to-end pipeline from data

preparation and feature engineering to model training, evaluation, and interpretability reporting. While the work is motivated by service delivery challenges in South African public hospitals, the evidence produced here should be read as demonstrating modelling capability and methodological readiness, not as demonstrating realised improvements in hospital operations. Any statements about practical impact are therefore framed as potential benefits that remain contingent on further validation using real-world clinical datasets and prospective evaluation in operational settings (de Hond et al., 2022; Zheng et al., 2025).

6.4 Recommendations and Future Work

A logical next step is a staged validation programme using real hospital datasets obtained with appropriate ethics approval, governance clearance, and privacy safeguards. Future work should evaluate the model on real-world data sources used in public hospitals, assess generalisation across facilities and time periods, and report both discrimination and calibration, following established guidance for prediction model development and validation (Collins et al., 2015; Steyerberg, 2019). A pilot implementation may then be considered to study usability and workflow fit, including monitoring for dataset shift, missing data behaviour, and performance drift after deployment.

Furthermore, future research can extend the prototype by testing alternative ensemble configurations, incorporating cost sensitive learning where needed, and conducting stakeholder informed evaluation to ensure that model outputs are interpretable and appropriate for the intended decision context.

ETHICAL ISSUES

Creswell and Creswell (2018) emphasise that ethics involve beliefs about what is right and wrong from a moral standpoint when engaging with participants or analysing archival data.

In this study, ethical considerations focused on several key issues:

- **Data privacy, confidentiality and anonymisation**: All personal identifiers were removed from the dataset, and aggregated results were used in reporting to ensure anonymity and protect participant confidentiality throughout the study.
- **Data Usage Transparency**: The researcher documented the origin and synthetic nature of the Final_healthcare_big_data.csv dataset, ensuring transparency regarding its use in predicting healthcare outcomes without involving real patients.
- **Ethics Clearance and Research Permission**: The researcher applied to the University of Mpumalanga Research Ethics Committee for ethical clearance to conduct this research.

REFERENCES

Abdalla, H. B. (2022). A brief survey on big data: technologies, terminologies and data-intensive applications. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00659-3

Achieng, M. S., & Ogundaini, O. O. (2024). Big Data Analytics for Integrated Infectious Disease. Surveillance in sub-Saharan Africa. *SA Journal of Information Management, 26*(1).https://doi.org/10.4102/sajim.v26i1.1668

Adari, G. K., Raja, M., & Vijaya, P. (2023). Machine learning in genomics: identification and modelling of anticancer peptides. *Elsevier EBooks*, 25–68. https://doi.org/10.1016/b978-0-323-98352-5.00007-0

Adekunle, N., Okolo, A., None Tolulope Olorunsogo, & None Oloruntoba Babawarun. (2024). Leveraging big data and analytics for enhanced public health decision-making: A global review. *GSC Advanced Research and Reviews, 18*(2), 450–456. https://doi.org/10.30574/gscarr.2024.18.2.0078

Adeniyi, A. O., Arowoogun, J. O., Chidi, R., Okolo, C. A., & Babawarun, O. (2024). The impact of electronic health records on patient care and outcomes: A comprehensive review. *World Journal of Advanced Research and Reviews*, *21*(2), 1446–1455. https://doi.org/10.30574/wjarr.2024.21.2.0592

Ahmed, A., Pereira, L., & Jane, K. (2024, September). Mixed Methods Research: Combining Both Qualitative and Quantitative Approaches. *ResearchGate*. https://www.researchgate.net/publication/384402328_Mixed_Methods_Research_Combining_both_qualitative_and_quantitative_approaches

Ahmed, A., Xi, R., Hou, M., Shah, S. A., & Hameed, S. (2023). Harnessing Big Data Analytics for Healthcare: A Comprehensive Review of Frameworks, Implications, Applications, and Impacts. *IEEE Access, 11*, 112891–112928. https://doi.org/10.1109/access.2023.3323574

Aikman, N. (2019). The crisis within the South African healthcare system: A multifactorial disorder. *South African Journal of Bioethics and Law, 12*(2), 52. https://doi.org/10.7196/sajbl.2019.v12i2.00694

Aldossari, S., Mokhtar, U. A., & Tarmizi, A. (2023). Factor Influencing the Adoption of Big Data Analytics: a Systematic Literature and Experts Review. *SAGE Open, 13*(4). https://doi.org/10.1177/21582440231217902

Alowais, S. A., Alghamdi, S. S., Alsuhebany, N., Alqahtani, T., Alshaya, A., Almohareb, S. N., Aldairem, A., Alrashed, M., Saleh, K. B., Badreldin, H. A., Yami, A., Harbi, S. A., & Albekairy, A. M. (2023). Revolutionizing healthcare: the role of artificial intelligence in clinical practice. *BMC Medical Education, 23*(1). https://doi.org/10.1186/s12909-023-04698-z

Alturki, A., Gable, G. G., & Wasana Bandara. (2025). *The Design Science Research Roadmap: In Progress Evaluation*. AIS Electronic Library (AISeL). https://aisel.aisnet.org/pacis2013/160

Ameen, A., Alganmi, N., & Bajnaid, N. (2025). Stacked Deep Learning Ensemble for Multiomics Cancer Type Classification: Development and Validation Study. *JMIR Bioinformatics and Biotechnology*, *6*, e70709–e70709. https://doi.org/10.2196/70709

Appenzeller, A., Leitner, M., Philipp, P., Krempel, E., & Beyerer, J. (2022). Privacy and Utility of Private Synthetic Data for Medical Data Analyses. *Applied Sciences, 12*(23), 12320. https://doi.org/10.3390/app122312320

Au-Yong-Oliveira, M., Pesqueira, A., Sousa, M. J., Dal Mas, F., & Soliman, M. (2021). The Potential of Big Data Research in HealthCare for Medical Doctors' Learning. *Journal of Medical Systems, 45*(1). https://doi.org/10.1007/s10916-020-01691-7

Auraria Library. (2023). *Research Guides: Research Methods: Literature Reviews*. Guides.auraria.edu. https://guides.auraria.edu/researchmethods/literaturereviews

Awrahman, B. J., Aziz Fatah, C., & Hamaamin, M. Y. (2022). A review of the role and challenges of big data in healthcare informatics and analytics. *Computational Intelligence and Neuroscience, 2022*(5317760), 1–10. https://doi.org/10.1155/2022/5317760

Badawy, M., Ramadan , N., & Hefny, H. A. (2023). Healthcare predictive analytics using machine learning and deep learning techniques: a survey. *Journal of Electrical Systems and Inf Technol, 10*(40), 40. https://doi.org/10.1186/s43067-023-00108-y

Badr, Y., Kader, L. A., & Shamayleh, A. (2024). The Use of Big Data in Personalized Healthcare to Reduce Inventory Waste and Optimize Patient Treatment. *Journal of Personalized Medicine, 14*(4), 383–383. https://doi.org/10.3390/jpm14040383

Bandi, M., Kumar, A., Vemula, R., & Vallu, S. (2024). Predictive Analytics in Healthcare: Enhancing Patient Outcomes through Data-Driven Forecasting and... *International Journal of Machine Learning*, *8*(8), 20. https://www.researchgate.net/publication/387306392

Batko, K., & Ślęzak, A. (2022). The Use of Big Data Analytics in Healthcare. *Journal of Big Data, 9*(1). ncbi. https://doi.org/10.1186/s40537-021-00553-4

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations, 6*(1), 20–29

Bhandari, A. (2020, June 15). *AUC-ROC Curve in Machine Learning Clearly Explained*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Bhati, D., Deogade, M. S., & Kanyal, D. (2023). Improving Patient Outcomes Through Effective Hospital Administration: A Comprehensive Review. *Cureus, 15*(10). https://doi.org/10.7759/cureus.47731

Beauxis-Aussalet, E., & Hardman, L. (2014). Simplifying the Visualization of Confusion Matrix. *BNAIC - Benelux Conference on Artificial Intelligence*. https://www.researchgate.net/publication/302412429_Simplifying_the_Visualization_of_Confusion_Matrix

Bose, S., Dey, S. K., & Bhattacharjee, S. (2023). *Big data, data analytics and artificial intelligence in accounting: an overview*. 32–51. https://doi.org/10.4337/9781800888555.00007

Bowe, A. K., Lightbody, G., Staines, A., & Murray, D. M. (2022). Big data, machine learning, and population health: predicting cognitive outcomes in childhood. *Pediatric Research*. https://doi.org/10.1038/s41390-022-02137-1

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological), 26*(2), 211–252.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Brossard, P.-Y., Minvielle, E., & Sicotte, C. (2022). The path from big data analytics capabilities to value in hospitals: A scoping review. *BMC Health Services Research, 22*(1). https://doi.org/10.1186/s12913-021-07332-0

Bruintjies, A. N., & Njenga, J. (2024). Factors affecting Big Data adoption in a government organisation in the Western Cape. *South African Journal of Information Management, 26*(1). https://doi.org/10.4102/sajim.v26i1.1690

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, 106*, 249–259.

Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N. S., Patzlaff, H., Harmouch, H., & Naumann, F. (2022). *The Effects of Data Quality on Machine Learning Performance*. https://doi.org/10.48550/arxiv.2207.14529

Carrasco, G., Islam, Md. Mafiqul., & Hasan, I. (2024). Machine Learning Applications in Healthcare: Current Trends and Future Prospects. *Deleted Journal, 1*(1). https://doi.org/10.60087/jaigs.v1i1.33

Cavanillas, J. M., Curry, E., & Wahlster, W. (2018). New Horizons for a Data-Driven Economy A Roadmap for Usage and Exploitation of Big Data in Europe. *Springer Nature*. https://doi.org/10.1007/978-3-319-21569-3

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing, 408*(1), 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Chabrol, F., & David, P.-M. (2023). How resilience affected public health research during COVID-19 and why we should abandon it. *Global Public Health, 18*(1). https://doi.org/10.1080/17441692.2023.2212750

Chao, K., Sarker, N. I., Ali, I., Radin, B., Azman, A., & Shaed, M. M. (2023). Big data-driven public health policy making: Potential for the healthcare industry. *Heliyon*, *9*(9), e19681–e19681. https://doi.org/10.1016/j.heliyon.2023.e19681

Chen, P., Sun, J., Chu, Y., & Zhao, Y. (2024). Predicting in-hospital mortality in patients with heart failure combined with atrial fibrillation using stacking ensemble model: an analysis of the medical information mart for intensive care IV (MIMIC-IV). *BMC Medical Informatics and Decision Making*, *24*(1). https://doi.org/10.1186/s12911-024-02829-0

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Chibuike, M. C., Sara, G. S., & Adele, B. (2024). Overcoming Challenges for Improved Patient-Centric Care: A Scoping Review of Platform Ecosystems in Healthcare. *IEEE Access, 12*, 14298–14313. https://doi.org/10.1109/access.2024.3356860

Clack, L. (2023, June 20). *Using Data Analytics to Predict Outcomes in Healthcare*. Journal of AHIMA. https://journal.ahima.org/page/using-data-analytics-to-predict-outcomes-in-healthcare

Cline, G. B., & Luiz, J. M. (2013). Information technology systems in public sector health facilities in developing countries: the case of South Africa. *BMC Medical Informatics and Decision Making, 13*(1). https://doi.org/10.1186/1472-6947-13-13

Cohen, L., Manion, L., & Morrison, K. (2018). *Research Methods in Education* (8th ed.). Routledge. https://doi.org/10.4324/9781315456539

Collins, G. S., Reitsma, J. B., Altman, D. G., & Moons, K. G. M. (2015). Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): Explanation and elaboration. *Annals of Internal Medicine, 162*(1), W1–W73.

Corrales, D. C., Ledezma, A., & Corrales, J. C. (2015). A Conceptual Framework for Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal. *Journal of Computers, 10*(6), 396–405. https://doi.org/10.17706/jcp.10.6.396-405

Cremin, C. J., Dash, S., & Huang, X. (2022). Big data: Historic advances and emerging trends in biomedical research. *Current Research in Biotechnology*, *4*, 138–151. https://doi.org/10.1016/j.crbiot.2022.02.004

Creswell, J. W., & Creswell, J. D. (2018). *Research design: qualitative, quantitative, and Mixed Methods Approaches* (5th ed.). SAGE Publications.

Creswell, J.W. (2015). *A Concise Introduction to Mixed Methods Research*. Thousand Oaks: Sage

Creswell, J. W. (2014). Research Design: Qualitative, Quantitative and Mixed Methods Approaches (4th ed.). Thousand Oaks, CA: Sage.

Darrell, T., & Estrin, D. (2024). *Revolutionising Healthcare with Predictive Analytics: The Role of Machine Learning in Patient Care*. https://doi.org/10.13140/RG.2.2.36486.59203

Dash, S., Shakyawar, S. K., Sharma, M., & Kaushik, S. (2019). Big data in healthcare: Management, analysis and prospects. *Journal of Big Data, 6*(1), 1–25. springer. https://doi.org/10.1186/s40537-019-0217-0

David, P., & Torshin, I. (2023). Global Trends in Data Analytics: Navigating the Big Data Landscape. *Computer Science Articles, 1*(1), 1–5. https://doi.org/10.13140/RG.2.2.28801.84320

de Hond, A. A. H., Leeuwenberg, A. M., Hooft, L., Kant, I. M. J., Nijman, S. W. J., van Os, H. J. A., Aardoom, J. J., Debray, T. P. A., Schuit, E., van Smeden, M., Reitsma, J. B., Steyerberg, E. W., Chavannes, N. H., & Moons, K. G. M. (2022). Guidelines and quality criteria for artificial intelligence-based

prediction models in healthcare: a scoping review. *Npj Digital Medicine*, *5*(1), 1–13. https://doi.org/10.1038/s41746-021-00549-7

De Maaye, T., & Coovadia, A. (2024). Advocacy in a Collapsing Public Healthcare Sector. *Wits Journal of Clinical Medicine, 6*(1). https://doi.org/10.18772/26180197.2024.v6n1a6

Dey, R., & Mathur, R. (2023). Ensemble Learning Method Using Stacking with Base Learner, A Comparison. *Lecture Notes in Networks and Systems*, 159–169. https://doi.org/10.1007/978-981-99-3878-0_14

Dolley, S. (2018). Big Data's Role in Precision Public Health. *Frontiers in Public Health, 6*. https://doi.org/10.3389/fpubh.2018.00068

Doutreligne, M., Struja, T., Abecassis, J., Morgand, C., Varoquaux, G., & Celi, L. A. (2023). Step-by-step causal analysis of Electronic Health Records to ground decision making. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-3222036/v1

Dunbray, N., Rane, R., Nimje, S., Katade, J., & Mavale, S. (2021). A Novel Prediction Model for Diabetes Detection Using Gridsearch and A Voting Classifier between Lightgbm and KNN. *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 1–7. https://doi.org/10.1109/gcat52182.2021.9587551

Durgapal, A. (2023, July 30). Data Preprocessing — Handling Duplicate Values and Outliers in a dataset. Medium. https://medium.com/@ayushmandurgapal/handling-duplicate-values-and-outliers-in-a-dataset-b00ce130818e

Edu, S. A., & Agozie, D. Q. (2022). Exploring Factors Influencing Big Data and Analytics Adoption in Healthcare Management. *IGI Global EBooks*, 1433–1449. https://doi.org/10.4018/978-1-6684-3662-2.ch069

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (pp. 973–978)

Favaretto, M., De Clercq, E., Schneble, C. O., & Elger, B. S. (2020). What is your definition of Big Data? Researchers' understanding of the phenomenon of the decade. *PLOS ONE*, *15*(2), e0228987. https://doi.org/10.1371/journal.pone.0228987

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fdil, E. M., & Souidi, E. M. (2024). Sybil Attack Detection in VANETs Using CatBoost Classifier. *Lecture Notes in Networks and Systems*, 428–437. https://doi.org/10.1007/978-3-031-62273-1_27

Fereidooni, D., Karimi, Z., & Ghasemi, F. (2024). Non-destructive test-based assessment of uniaxial compressive strength and elasticity modulus of intact carbonate rocks using stacking ensemble models. *PLoS ONE, 19*(6), e0302944–e0302944. https://doi.org/10.1371/journal.pone.0302944

Ferreira, D. C., Vieira, I., Pedro, M. I., Caldas, P., & Varela, M. (2023). Patient Satisfaction with Healthcare Services and the Techniques Used for Its Assessment: a Systematic Literature Review and a Bibliometric Analysis. *Healthcare, 11*(5), 639. https://doi.org/10.3390/healthcare11050639

Fisher, M., & Raman, A. (2018). Using Data and Big Data in Retailing. *Production and Operations Management, 27*(9), 1665–1669. https://doi.org/10.1111/poms.12846

Fragala, M. S., Shiffman, D., & Birse, C. E. (2019). Population Health Screenings for the Prevention of Chronic Disease Progression. *The American Journal of Managed Care, 25*(11). https://www.ajmc.com/view/population-health-screenings-for-the-prevention-of-chronic-disease-progression

Frisk, J. E., & Bannister, F. (2017). Improving the use of analytics and big data by changing the decision-making culture. *Management Decision, 55*(10), 2074–2088. https://doi.org/10.1108/md-07-2016-0460

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Gandomi, A., & Haider, M. (2015). Beyond the Hype: Big Data Concepts, Methods, and Analytics. *International Journal of Information Management, 35*(2), 137–144.

GeeksforGeeks. (2020, November 24). *A Beginners Guide To Streamlit*. GeeksforGeeks. https://www.geeksforgeeks.org/python/a-beginners-guide-to-streamlit/

Ghaleb , E. A. A., Dominic, E. A. A., Singh, N., & Ahmed, M. (2023). Assessing the Big Data Adoption Readiness Role in Healthcare between Technology Impact Factors and Intention to Adopt Big Data. *Sustainability*, *15*(15), 11521–11521. https://doi.org/10.3390/su151511521

Gordon, T., Booysen, F., & Mbonigaba, J. (2020). Socio-economic Inequalities in the Multiple Dimensions of Access to healthcare: the Case of South Africa. *BMC Public Health, 20*(1). https://doi.org/10.1186/s12889-020-8368-7

Goundar, S., Karpagam Masilamani, Bhardwaj, A., & Chandramohan Dhasarathan. (2021). Big Data Analytics in Healthcare. *Advances in Data Mining and Database Management Book Series*, 85–98. https://doi.org/10.4018/978-1-7998-6673-2.ch006

Gül, V. (2023, April 23). *Machine Learning Part-4 (Random Forest-GBM-XGBoost-LightGBM-CatBoost)*. Medium. https://medium.com/@veribilimi35/machine-learning-part-4-random-forest-gbm-xgboost-lightgbm-catboost-10e4e69eef33

Guo, H., Nguyen, H., Vu, D.-A., & Bui, X.-N. (2019). Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach. *Resources Policy*, 101474. https://doi.org/10.1016/j.resourpol.2019.101474

Guo, M., Wang, Y., Yang, Q., Li, R., Zhao, Y., Li, C., Zhu, M., Yao, C., Xin, J., Song, S., Li, Q., & Gao, R. (2023). Normal Workflow and Key Strategies for Data Cleaning Toward Real-World Data: Viewpoint. *Interactive Journal of Medical Research, 12*, e44310–e44310. https://doi.org/10.2196/44310

Gupta, P. M. (2023). The Role of Big Data in Smart Healthc. International Journal of Internet of Things, 11(1), 11–18. https://doi.org/10.5923/j.ijit.20231101.02

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Goyal, P., & Malviya, R. (2023). Challenges and opportunities of big data analytics in healthcare. *Health Care Science, 2*(5). https://doi.org/10.1002/hcs2.66

Grover, V. (2015). *RESEARCH APPROACH: AN OVERVIEW*. ResearchGate; www.researchgate.net. https://www.researchgate.net/publication/273352276_RESEARCH_APPROACH_AN_OVERVIEW

Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of Big Data, 7*(1). https://doi.org/10.1186/s40537-020-00369-8

Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in Big Data analytics: survey, opportunities, and Challenges. *Journal of Big Data, 6*(1), 1–16. springeropen. https://doi.org/10.1186/s40537-019-0206-3

Hassan, M. K., El Desouky, A. I., Elghamrawy, S. M., & Sarhan, A. M. (2018). Big Data Challenges and Opportunities in Healthcare Informatics and Smart Hospitals. *Security in Smart Cities: Models, Applications, and Challenges*, 3–26. https://doi.org/10.1007/978-3-030-01560-2_1

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.

Hosseini, S. M., Boushehri, S. A., & Alimohammadzadeh, K. (2024). Challenges and solutions for implementing telemedicine in Iran from health policymakers' perspective. *BMC Health Services Research, 24*(1). https://doi.org/10.1186/s12913-023-10488-6

Hossin, M. A., Du, J., Mu, L., & Asante, I. O. (2023). Big Data-Driven Public Policy Decisions: Transformation Toward Smart Governance. *SAGE Publications, 13*(4). https://doi.org/doi.org/10.1177/215824402312151

Hulsen, T., Jamuar, S. S., Moody, A. R., Karnes, J. H., Varga, O., Hedensted, S., Spreafico, R., Hafler, D. A., & McKinney, E. F. (2019). From Big Data to Precision Medicine. *Frontiers in Medicine, 6*(34). https://doi.org/10.3389/fmed.2019.00034

Hummel, P., Braun, M., Bischoff, S., Samhammer, D., Seitz, K., Fasching, P. A., & Dabrock, P. (2024). Perspectives of patients and clinicians on big data and AI in health: a comparative empirical investigation. *AI & SOCIETY*. https://doi.org/10.1007/s00146-023-01825-8

I

BM. (2020). *What Is Exploratory Data Analysis?* | *IBM*. Www.ibm.com; IBM. https://www.ibm.com/topics/exploratory-data-analysis

IBM. (2024, April 15). *What is Big Data Analytics?* | *IBM*. Www.ibm.com. https://www.ibm.com/topics/big-data-analytics

Idrees, S. M., Alam, M. A., Agarwal, P., & Ansari, L. (2019). Effective Predictive Analytics and Modeling Based on Historical Data. *Communications in Computer and Information Science*, 552–564. https://doi.org/10.1007/978-981-13-9942-8_52

Ishtiaq, M. (2019). creswell, J. W. (2014). research design: Qualitative, quantitative and mixed methods approaches (4th ed.). thousand oaks, CA: Sage. *English Language Teaching*, *12*(5), 40. https://doi.org/10.5539/elt.v12n5p40

Iyamu, T. (2020). A framework for selecting analytics tools to improve healthcare big data usefulness in developing countries. *SA Journal of Information Management, 22*(1). https://doi.org/10.4102/sajim.v22i1.1117

Iyamu, T., & Mgudlwa, S. (2021). ANT Perspective of Healthcare Big Data for Service Delivery in South Africa. *Journal of Cases on Information Technology*, *23*(1), 65–81. https://doi.org/10.4018/jcit.2021010104

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

Javed, H., El-Sappagh, S., & Abuhmed, T. (2024). Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust AI applications. *Artificial Intelligence Review, 58*(1). https://doi.org/10.1007/s10462-024-11005-9

Jinabhai, C. C., Onwubu, S. C., Sibiya, M. N., & Thakur, S. (2021). Accelerating implementation of District Health Information Systems: Perspectives from healthcare workers from KwaZulu-Natal, South Africa. *SA Journal of Information Management, 23*(1). https://doi.org/10.4102/sajim.v23i1.1435

Johnson, R. B., & Christensen, L. (2020). *Educational Research: Quantitative, Qualitative, and Mixed Approaches.* Sage Publications.

Johnson, O., Brown, W., & Wilson, G. (2024). The Role of Big Data Analytics in Retail Marketing and Supply Chain Optimization. *Preprints.org*. https://doi.org/10.20944/preprints202407.2058.v1

Jung, D. (2024). Business Data Understanding. *Springer, Cham, 1*(3), 49–110. https://doi.org/10.1007/978-3-031-59907-1_3

Kahn, T. (2025, August 6). *State does not know how many foreigners use SA's public health system*. BusinessLIVE; Business Day. https://www.businesslive.co.za/bd/national/health/2025-08-06-state-does-not-know-how-many-foreigners-use-sas-public-health/

Kala, E. S. M. (2023). Challenges of Technology in African Countries: a Case Study of Zambia. *Open Journal of Safety Science and Technology*, *13*(4), 202–230. https://doi.org/10.4236/ojsst.2023.134011

Kalema, B. M., & Musoma, R. M. (2019). *Mobile Health monitoring systems model for chronic diseases patients in developing countries*. https://doi.org/10.1109/iemcon.2019.8936181

Katurura, M. C., & Cilliers, L. (2018). Electronic health record system in the public health care sector of South Africa: A systematic literature review. *African Journal of Primary Health Care & Family Medicine, 10*(1). https://doi.org/10.4102/phcfm.v10i1.1746

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*

Kgasi, M., Chimbo, B. & Motsi, L. (2023). mHealth self-monitoring model for medicine adherence of diabetic patients in developing countries: A structural equation modelling approach. *JMIR Form Res, 2023*;7:e49407
https://doi.org/10.2196/49407

Kharwal, A. (2021, July 7). *Classification Report in Machine Learning | Aman Kharwal*. AmanXai. https://amanxai.com/2021/07/07/classification-report-in-machine-learning/

Khosravi, M., Zare, Z., Mojtabaeian, S. M., & Izadi, R. (2024). Artificial Intelligence and Decision-Making in Healthcare: A Thematic Analysis of a Systematic Review of Reviews. *Health Services Research and Managerial Epidemiology, 11*, 1–15. https://doi.org/10.1177/23333928241234863

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (pp. 1137–1143).

Kolyshkina, I., & Simoff, S. (2021). Interpretability of Machine Learning Solutions in Public Healthcare: The CRISP-ML Approach. *Frontiers in Big Data*, *4*. https://doi.org/10.3389/fdata.2021.660206

Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). Exploratory Data Analysis. *Secondary Analysis of Electronic Health Records, 2*(15), 185–203. springer. https://doi.org/10.1007/978-3-319-43742-2_15

Kopra, J., Tikka, S., Heinäniemi, M., López-Pernas, S., & Saqr, M. (2024). An R Approach to Data Cleaning and Wrangling for Education Research. *ResearchGate*, 95–119. https://doi.org/10.1007/978-3-031-54464-4_4

Kotronoulas, G. (2023). An Overview of the Fundamentals of Data management, analysis, and Interpretation in Quantitative Research. *Seminars in Oncology Nursing, 39*(2), 151398. Sciencedirect. https://doi.org/10.1016/j.soncn.2023.151398

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221–232.

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. Data Democracy, 83–106. https://doi.org/10.1016/b978-0-12-818366-3.00005-8

Kumar, N., Hema, K., Hordiichuk, V., Menon, R., Catherene, Aarthy, C. J., & Gonesh, C. (2023). *Harnessing the Power of Big Data: Challenges and Opportunities in Analytics*. 44(2). https://doi.org/10.52783/tjjpt.v44.i2.193

Lekkala, L. R. (2023). Applications and Challenges in Healthcare Big Data. *International Journal of Information Technology (IJIT), 4*(1), 72–79. https://doi.org/10.17605/OSF.IO/AT8JR

Lenth, R. V. (1995). Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap. *Technometrics, 37*(4), 458–459. https://doi.org/10.1080/00401706.1995.10484382

Li, H., Nyirandayisabye, R., Dong, Q., Niyirora, R., Hakuzweyezu, T., Zardari, I. A., & Nkinahamira, F. (2024). Crack damage prediction of asphalt pavement based on tire noise: A comparison of machine learning algorithms. *Construction and Building Materials, 414*, 134867. https://doi.org/10.1016/j.conbuildmat.2024.134867

Liang, J., Li, Y., Zhang, Z., Shen, D., Xu, J., Zheng, X., Wang, T., Tang, B., Lei, J., & Zhang, J. (2021). Adoption of Electronic Health Records (EHRs) in China During the Past 10 Years: Consecutive Survey Data Analysis and Comparison of Sino-American Challenges and Experiences. *Journal of Medical Internet Research*, *23*(2), e24813. https://doi.org/10.2196/24813

Lind, E., & Glas, S. (2022). *DATA MINING IN PRACTICE : An application of the CRISP-DM framework in healthcare (Dissertation*). Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-197610

Little, R. J. A., & Rubin, D. B. (2019). Statistical analysis with missing data (3rd ed.). Wiley.

Liu, L., Wu, X., Li, S., Li, Y., Tan, S., & Bai, Y. (2022). Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection. *BMC Medical Informatics and Decision Making, 22*(1). https://doi.org/10.1186/s12911-022-01821-w

LoBiondo-Wood, G. & Haber, J. (2014). *Nursing Research - E-Book*. 8th ed. Philadelphia: Mosby, p.290.

Lu, M., Hou, Q., Qin, S., Zhou, L., Hua, D., Wang, X., & Cheng, L. (2023). A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting. *Water, 15*(7), 1265. https://doi.org/10.3390/w15071265

Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2020). *Big Data Preprocessing* (1st ed.). Springer International Publishing. https://doi.org/10.1007/978-3-030-39105-8

Lundberg, S., & Lee, S.-I. (2017a). *A Unified Approach to Interpreting Model Predictions*. ArXiv.org. https://doi.org/10.48550/arXiv.1705.07874

Lundberg, S. M., & Lee, S.-I. (2017b). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*.

Lutfi, A., Alsyouf, A., Almaiah, M. A., Alrawad, M., Abdo, A. A. K., Al-Khasawneh, A. L., Ibrahim, N., & Saad, M. (2022). Factors Influencing the Adoption of Big Data Analytics in the Digital Transformation Era: Case Study of Jordanian SMEs. *Sustainability, 14*(3), 1802. https://doi.org/10.3390/su14031802

Mahmood, S., Coovadia, A., Laher, A. E., & Adam, A. (2023). mHealth app usage amongst paediatric department doctors in South Africa. *African Health Sciences, 23*(3). https://doi.org/10.4314/ahs.v23i3.24

Maimon, O., & Rokach, L. (2005). Introduction to Knowledge Discovery in Databases. *Data Mining and Knowledge Discovery Handbook*, 1–17. https://doi.org/10.1007/0-387-25465-x_1

Makeleni, N., & Cilliers, L. (2021). Critical success factors to improve data quality of electronic medical records in public healthcare institutions. *SA Journal of Information Management, 23*(1). https://doi.org/10.4102/sajim.v23i1.1230

Maleka, N. H., & Matli, W. (2022). A review of telehealth during the COVID-19 emergency situation in the public health sector: challenges and opportunities. *Journal of Science and Technology Policy Management, 15*(4), 707–724. Journal of Science and Technology Policy Management. https://doi.org/10.1108/jstpm-08-2021-0126

Malhotra, G. (2017). Strategies in Research. *International Journal of Advance Research, Ideas and Innovations in Technology*, *2*(5). www.IJARnD.com.

Malik, M. M., Abdallah, S., & Ala'raj, M. (2018). Data mining and predictive analytics applications for the delivery of healthcare services: a systematic literature review. *Annals of Operations Research, 270*(1-2), 287–312. https://doi.org/10.1007/s10479-016-2393-z

Malungana, L., & Motsi, L. (2024). An investigation of healthcare professionals' intention to use Smart Card Technology. *South African Journal of Information Management, 26*(1). https://doi.org/10.4102/sajim.v26i1.1663

Mantzaris, E., & Pillay, P. (2021). Legal Profession and Corruption in Health Care: Some Reflective Realities in South Africa. *Frontiers in Public Health, 9*. https://doi.org/10.3389/fpubh.2021.688049

Maphumulo, W. T., & Bhengu, B. R. (2019). Challenges of Quality Improvement in the Healthcare of South Africa post-apartheid: a Critical Review. *Curationis, 42*(1), 1–9. https://doi.org/10.4102/curationis.v42i1.1901

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251–266.

Marutha, N. S., & Ngoepe, M. (2018). Medical records management framework to support public healthcare services in Limpopo province of South Africa. *Records Management Journal, 28*(2), 187–203. https://doi.org/10.1108/rmj-10-2017-0030

Mash, R., Williams, B., Stapar, D., Hendricks, G., Steyn, H., Schoevers, J., Wagner, L., Abbas, M., Kapp, P., Perold, S., Swartz, S., Viljoen, W., & Bello, M. (2022). Retention of medical officers in district health services, South Africa: a descriptive survey. *BJGP Open, 6*(4), BJGPO.2022.0047. https://doi.org/10.3399/BJGPO.2022.0047

Matahela, R. S., Adekola, A. P., & Mavhandu-Mudzusi, A. H. (2023). Exploring quality standards implementation at a South African municipality's health facilities. *Curationis, 46*(1). https://doi.org/10.4102/curationis.v46i1.2416

Maxwell, L., Shreedhar , P., & Krishnan, A. (2023). How do we measure the costs, benefits, and harms of sharing data from biomedical studies? A scoping review. *Open Res Europe , 3*(151). https://doi.org/doi.org/10.12688/openreseurope.16063.1

Mbunge, E., Batani, J., Gaobotse, G., & Muchemwa, B. (2022). Virtual healthcare services and digital health technologies deployed during coronavirus disease 2019 (COVID-19) pandemic in South Africa: a systematic review. *Global Health Journal, 6*(2). https://doi.org/10.1016/j.glohj.2022.03.001

Mehta, N., Pandit, A., & Shukla, S. (2019). Transforming Healthcare with Big Data Analytics and Artificial Intelligence: A Systematic Mapping Study. *Journal of Biomedical Informatics*, *100*, 103311. https://doi.org/10.1016/j.jbi.2019.103311

Mentsiev, A., Aygumov, T., & Amirova, E. (2023). DATA-DRIVEN DIGITAL TRANSFORMATION: CHALLENGES AND STRATEGIES FOR EFFECTIVE BIG DATA MANAGEMENT. *RT&A, 18*(5). https://doi.org/10.24412/1932-2321-2023-575-526-531

Miandoab, A. T., Samad-Soltani, T., Jodati, A., & Rezaei-Hachesu, P. (2023). Interoperability of heterogeneous health information systems: a systematic literature review. *BMC Medical Informatics and Decision Making, 23*(1). https://doi.org/10.1186/s12911-023-02115-5

Michel, J., Obrist, B., Bärnighausen, T., Tediosi, F., McIntyre, D., Evans, D., & Tanner, M. (2020). What we need is health system transformation and not health system strengthening for universal health coverage to work: Perspectives from a National Health Insurance pilot site in South Africa. *South African Family Practice, 62*(1). https://doi.org/10.4102/safp.v62i1.5079

Mishra, A. K., Keserwani, P. K., Samaddar, S. G., & Lamichaney, H. B. (2018). A Decision Support System in Healthcare Prediction. *Lecture Notes in Electrical Engineering*, 156–167. https://doi.org/10.1007/978-981-10-8240-5_18

Mohammed, S. H., Ahamad, S., & Hameed, M. A. (2024). Enhancements in Random Forest Algorithms for Improving Healthcare Applications. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. https://doi.org/10.46254/in04.20240192

Monsen, K. A. (2024). Exploratory Data Analysis. *Springer EBooks, 7*(8), 105–115. https://doi.org/10.1007/978-3-031-54111-7_8

Moodley, K. (2023). Artificial intelligence (AI) or augmented intelligence? How big data and AI are transforming healthcare: Challenges and opportunities. *South African Medical Journal, 114*(1), 22–26. https://doi.org/10.7196/samj.2024.v114i1.1631

Moodley, K., & James, R. (2024). The adoption of ICT and robotic automation systems in the pharmaceutical industry. *South African Journal of Information Management, 26*(1). https://doi.org/10.4102/sajim.v26i1.1744

Morar, R. (2024). *Navigating the road to Universal Health Coverage in South Africa - News*. Mandela.ac.za. https://news.mandela.ac.za/Opinion-pieces/Opinion-pieces-archive/Navigating-the-road-to-Universal-Health-Coverage-i

Motsi, L., & Chimbo, B. (2023). Success factors for evidence-based healthcare practice adoption. *South African Journal of Information Management, 25*(1), 1–9. https://doi.org/10.4102/sajim.v25i1.1622

Muhammad, J., Ren, H., Gilliam, C., Deng, W., & Ogwal, E. (2020). Comparison of Hbase, Cassandra, and MongoDB. *ResearchGate*. https://doi.org/10.13140/RG.2.2.19775.43686

Muhunzi, D., Kitambala, L., & Mashauri, H. L. (2023). Big Data Analytics in the Healthcare Sector: Opportunities and Challenges in Developing Countries. A Literature Review. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-2869049/v1

Mujahid, M., Kına, E., Rustam, F., Villar, M. G., Alvarado, E. S., Diez, I. D. L. T., & Ashraf, I. (2024). Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. *Journal of Big Data, 11*(1), 1–32. https://doi.org/10.1186/s40537-024-00943-4

Mukwena, N. V., & Manyisa, Z. M. (2022). Factors influencing the preparedness for the implementation of the national health insurance scheme at a selected hospital in Gauteng Province, South Africa. *BMC Health Services Research, 22*(1). https://doi.org/10.1186/s12913-022-08367-7

Mumtaz, H., Riaz, M. H., Wajid, H., Saqib, M., Zeeshan, M. H., Khan, S. E., Chauhan, Y. R., Sohail, H., & Vohra, L. I. (2023). Current challenges and potential solutions to the use of digital health technologies in evidence generation: a narrative review. *Frontiers in Digital Health*, *5*(5). https://doi.org/10.3389/fdgth.2023.1203945

Munné, R. (2016). Big Data in the Public Sector. *New Horizons for a Data-Driven Economy*, 195–208. https://doi.org/10.1007/978-3-319-21569-3_11

Muraina, I. O. (2022, February 3). *IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND*... ResearchGate; unknown. https://www.researchgate.net/publication/358284895_IDEAL_DATASET_SPLITTING_RATIOS_IN_MACHINE_LEARNING_ALGORITHMS_GENERAL_CONCERNS_FOR_DATA_SCIENTISTS_AND_DATA_ANALYSTS

Murel , J., & Kavlakoglu, E. (2024, January 20). *Feature Engineering*. Ibm.com. https://www.ibm.com/think/topics/feature-engineering

Nahm, F. S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, *75*(1), 25–36. https://doi.org/10.4097/kja.21209

Nakashololo, T. M., & Iyamu, T. (2023). Decision support model for big data analytics tools. *South African Journal of Information Management, 25*(1). https://doi.org/10.4102/sajim.v25i1.1678

Nalluri, M., Eluri, N. R., & Pentela, M. (2020). A Scalable Tree Boosting System: XG Boost. *International Journal of Research Studies in Science, Engineering and Technology,* *7*(12), 36–51. https://doi.org/10.22259/2349-476X.0712005

National Department of Health. (2019). *National Digital Health Strategy for South Africa*. National Department of Health Republic of South Africa. https://www.health.gov.za/wp-content/uploads/2020/11/national-digital-strategy-for-south-africa-2019-2024-b.pdf

Ndebele, N. C., & Ndlovu, J. (2023). The Impact of Staffing Moratoria on the Delivery of Quality Health Care Services in the Department of Health. *African Journal of Governance and Development, 12*(2), 120–140. https://doi.org/10.36369/2616-9045/2023/v12i2a8

Nene, S. E., & Hewitt, L. M. (2023). Implementing artificial intelligence in South African public hospitals: A conceptual framework. *Acta Commercii, 23*(1), 6. https://doi.org/10.4102/ac.v23i1.1173

Netcare. (2022). *Pioneering electronic medical record system “a game changer” for hospital*. Www.netcare.co.za. https://www.netcare.co.za/News-Hub/Articles/pioneering-electronic-medical-record-system-a-game-changer-for-hospital

Ngene, N. C., Khaliq, O. P., & Jagidesa Moodley. (2023). Inequality in health care services in urban and rural settings in South Africa. *PubMed, 27*(5s), 87–95. https://doi.org/10.29063/ajrh2023/v27i5s.11

Ngesa, J. (2023). Tackling security and privacy challenges in the realm of big data analytics. *World Journal of Advanced Research and Reviews, 21*(2), 552–576. https://doi.org/10.30574/wjarr.2024.21.2.0429

Nguyen, N., & Ngo, D. (2025). Comparative analysis of boosting algorithms for predicting personal default. *Cogent Economics & Finance, 13*(1). https://doi.org/10.1080/23322039.2025.2465971

Nijjer, S., Saurabh, K., & Raj, S. (2020). Predictive Big Data Analytics in Healthcare. *Big Data Analytics and Intelligence: A Perspective for Health Care*, 75–91. https://doi.org/10.1108/978-1-83909-099-820201009

Nkoane, N. L. (2023). Experiences of operational managers regarding record keeping by new professional nurses in public hospitals in the North West province, South Africa. *Health SA Gesondheid*, *28*, 2257. https://doi.org/10.4102/hsag.v28i0.2257

Nkwanyana, A., Mathews, V., Zachary, I., & Bhayani, V. (2023). Skills and competencies in health data analytics for health professionals: a scoping review protocol. *BMJ Open, 13*(11), e070596–e070596. https://doi.org/10.1136/bmjopen-2022-070596

Nuthalapati, S. B., & Nuthalapati, A. (2024). TRANSFORMING HEALTHCARE DELIVERY VIA IOT-DRIVEN BIG DATA ANALYTICS IN A CLOUD-BASED PLATFORM. *Journal of Population Therapeutics & Clinical Pharmacology*, 2559–2569. https://doi.org/10.53555/jptcp.v31i6.6975

Nyasulu, C., & Chawinga, W. D. (2018). The role of information and communication technologies in the delivery of health services in rural communities: Experiences from Malawi. *SA Journal of Information Management, 20*(1). https://doi.org/10.4102/sajim.v20i1.888

Obasa, A. E., & Palk, A. C. (2023). Responsible application of artificial intelligence in health care. *South African Journal of Science, 119*(5/6). https://doi.org/10.17159/sajs.2023/14889

Ogbogu, K. (2022, August 2). *Data Cleaning and Preparation With SQL*. Medium. https://medium.com/@kelechiogbogu/data-cleaning-and-preparation-with-sql-f5f7e539808

Office of Health Standards Compliance. (2024). Annual report 2023/24. Retrieved from https://nationalgovernment.co.za/entity_annual/3961/2024-office-of-health-standards-compliance-(ohsc)-annual-report.pdf

Oleribe, O. E., Momoh, J., Uzochukwu, B. S., Mbofana, F., Adebiyi, A., Barbera, T., Williams, R., & Taylor Robinson, S. D. (2019). Identifying key challenges facing healthcare systems in Africa and potential solutions. *International Journal of General Medicine, 12*(1), 395–403. https://doi.org/10.2147/IJGM.S223882

Omoseebi, A., Ola, G., & Tyler, J. (2025, February 15). *Data Preparation and Feature Engineering*. ResearchGate.https://www.researchgate.net/publication/389860294_Data_Preparation_and_Feat

ure_Engineering?enrichId=rgreq-447351434d763118f5ff141223cb4c9b-XXX&enrichSource=Y292ZXJQYWdlOzM4OTg2MDI5NDtBUzoxMTQzMTI4MTMxNjA1NDQ0OEAxNzQyMDMwMTgyNzkz&el=1_x_2&_esc=publicationCoverPdf

Oranga, J., & Matere, A. (2023). Qualitative Research: Essence, Types and Advantages. *Open Access Library Journal, 10*(12), 1–9. https://doi.org/10.4236/oalib.1111001

Osborne, J. W. (2002). Notes on the use of data transformations. *Practical Assessment, Research, and Evaluation, 8*(6).

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences, 30*(4), 431–448. https://doi.org/10.1016/j.jksuci.2017.06.001

Padarath, A., & Moeti, T. (2023). South African Health Review 2022: health systems recovery after COVID-19. *South African Health Review, 25*. https://doi.org/10.61473/001c.87567

Pastorino, R., De Vito, C., Migliara, G., Glocker, K., Binenbaum, I., Ricciardi, W., & Boccia, S. (2019). Benefits and challenges of big data in healthcare: An overview of the european initiatives. *European Journal of Public Health, 29*(3), 23–27. https://doi.org/10.1093/eurpub/ckz168

Paul, M., Maglaras, L., Ferrag, M. A., & Almomani, I. (2023). Digitization of Healthcare sector: A Study on Privacy and Security Concerns. *ICT Express, 9*(4), 571–588. https://doi.org/10.1016/j.icte.2023.02.007

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77.

Pervin, N., & Mokhtar, M. (2022). The Interpretivist Research paradigm: a Subjective Notion of a Social Context. *International Journal of Academic Research in Progressive Education and Development*, *11*(2), 419–428. https://doi.org/10.6007/IJARPED/v11-i2/12938

Pezoulas, V. C., Zaridis, D. I., Mylona, E., Androutsos, C., Apostolidis, K., Tachos, N. S., & Fotiadis, D. I. (2024). Synthetic data generation methods in healthcare: A review on open-source tools and methods. *Computational and Structural Biotechnology Journal, 23,* 2892–2910. https://doi.org/10.1016/j.csbj.2024.07.005

Pool, J. K., Akhlaghpour, S., Fatehi, F., & Burton-Jones, A. (2024). A systematic analysis of failures in protecting personal health data: A scoping review. *International Journal of Information Management, 74*(102719), 102719–102719. https://doi.org/10.1016/j.ijinfomgt.2023.102719

Prakash, D. (2024). Data-Driven Management: The Impact of Big Data Analytics on Organizational Performance*. International Journal for Global Academic & Scientific Research, 3*(2), 12–23. https://doi.org/10.55938/ijgasr.v3i2.74

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*.

Pugliese, R., Regondi, S., & Marini, R. (2021). Machine learning-based approach: Global trends, research directions, and regulatory standpoints. *Data Science and Management, 4*, 19–29. https://doi.org/10.1016/j.dsm.2021.12.002

Rahul, K., Banyal, R. K., & Arora, N. (2023). A systematic review on big data applications and scope for industrial processing and healthcare sectors. *Journal of Big Data, 10*(1). https://doi.org/10.1186/s40537-023-00808-2

Rajandran, R. (2023, April 2). *Decoding Features and Targets in Machine Learning: The Keys to Model Success*. Marketcalls. https://www.marketcalls.in/machine-learning/decoding-features-and-targets-in-machine-learning-the-keys-to-model-success.html

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., Sundberg, P., Yee, H., Zhang, K., Zhang, Y., Flores, G., Duggan, G. E., Irvine, J., Le, Q., Litsch, K., & Mossin, A. (2018). Scalable and accurate deep learning with electronic health records. *Npj Digital Medicine, 1*(1). https://doi.org/10.1038/s41746-018-0029-1

Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., & Padma, V. (2020, August 1). *Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification*. IEEE Xplore. https://doi.org/10.1109/ICSSIT48917.2020.9214160

Ramezani, M., Amirhossein Takian, Bakhtiari, A., Rabiee, H. R., Sadegh Ghazanfari, & Mostafavi, H. (2023). The application of artificial intelligence in health policy: a scoping review. *BMC Health Services Research, 23*(1). https://doi.org/10.1186/s12913-023-10462-2

Rammila, D. (2023). Evaluating the potential impact of National Health Insurance on medical scheme members' rights to have access to health-care services in South Africa. *Law, Democracy and Development, 27*, 360–391. https://doi.org/10.17159/2077-4907/2023/ldd.v27.14

Rasheed, K., Qayyum, A., Ghaly, M., Al-Fuqaha, A., Razi, A., & Qadir, J. (2022). Explainable, trustworthy, and ethical machine learning for healthcare: A survey. *Computers in Biology and Medicine*, *149*(106043), 106043. https://doi.org/10.1016/j.compbiomed.2022.106043

Razali, M. N., Arbaiy, N., Lin, P.-C., & Ismail, S. (2025). Optimizing Multiclass Classification Using Convolutional Neural Networks with Class Weights and Early Stopping for Imbalanced Datasets. *Electronics, 14*(4), 705–705. https://doi.org/10.3390/electronics14040705

Rehman, A., Naz, S., & Razzak, I. (2021). Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities. *Multimedia Systems, 28*(4), 1339–1371. https://doi.org/10.1007/s00530-020-00736-8

Rensburg, R. (2021, July 7). *Healthcare in South Africa: How Inequity Is Contributing to Inefficiency - Wits University*. Www.wits.ac.za. https://www.wits.ac.za/news/latest-news/opinion/2021/2021-07/healthcare-in-south-africa-how-inequity-is-contributing-to-inefficiency.html

Rezvani, S., & Wang, X. (2023). A broad review on class imbalance learning techniques. *Applied Soft Computing, 143*, 110415–110415. https://doi.org/10.1016/j.asoc.2023.110415

Ridzuan, F., & Zainon, W. M. N. W. (2019). A Review on Data Cleansing Methods for Big Data. *Procedia Computer Science, 161*(1), 731–738. https://doi.org/10.1016/j.procs.2019.11.177

Rizzoli, A. (2022). *Training Data Quality: Why It Matters in Machine Learning*. Www.v7labs.com. https://www.v7labs.com/blog/quality-training-data-for-machine-learning-guide

RMB. (2021). Healthcare innovation – the SA impact - Rand Merchant Bank. Www.rmb.co.za. https://www.rmb.co.za/news/healthcare-innovation-the-sa-impact

Rostamzadeh, N., Abdullah, S. S., & Sedig, K. (2020). Data-Driven Activities Involving Electronic Health Records: An Activity and Task Analysis Framework for Interactive Visualization Tools. *Multimodal Technologies and Interaction, 4*(1), 7. https://doi.org/10.3390/mti4010007

Salmi, M., Atif, D., Oliva, D., Abraham, A., & Ventura, S. (2024). Handling imbalanced medical datasets: review of a decade of research. *Artificial Intelligence Review, 57*(10). https://doi.org/10.1007/s10462-024-10884-2

Sarker, M. (2024). Revolutionizing Healthcare: The Role of Machine Learning in the Health Sector. *Journal of Artificial Intelligence General Science (JAIGS) ISSN:3006-4023, 2*(1), 36–61. https://doi.org/10.60087/jaigs.v2i1.96

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods, 7*(2), 147–177.

Schneider, H., Kredo, T., Odendaal, W. A., & Abdullah, F. (2023). Healthcare Delivery. *South African Medical Journal, 113*(2), 61–64. https://doi.org/10.7196/samj.2023.v113i2.16798

Sedláková, J., Daniore, P., Andrea Horn Wintsch, Wolf, M., Stanikić, M., Haag, C., Sieber, C., Schneider, G., Staub, K., Ettlin, D. A., Grübner, O., Rinaldi, F., & Viktor von Wyl. (2023). Challenges and best practices for digital unstructured data enrichment in health research: A systematic narrative review. *PLOS Digital Health, 2*(10), e0000347–e0000347. https://doi.org/10.1371/journal.pdig.0000347

Setyati, R., Astuti, A., Utami, T. P., Adiwjaya, S., & Hasyim, D. M. (2024). The Importance of Early Detection in Disease Management. *Journal of World Future Medicine Health and Nursing, 2*(1), 51–63. https://doi.org/10.55849/health.v2i1.692

Sehlabo, R., & Vambe, W. T. (2024). Bilingual Medication (Med-Alert) Reminder Application for Patients. *IEEE*. https://doi.org/10.23919/ist-africa63983.2024.10569165

Sewal, P., & Singh, H. (2021, October 1). *A Critical Analysis of Apache Hadoop and Spark for Big Data Processing*. IEEE Xplore. https://doi.org/10.1109/ISPCC53510.2021.9609518

Shabbir, M. Q., & Gardezi, S. B. W. (2020). Application of big data analytics and organizational performance: the mediating role of knowledge management practices. *Journal of Big Data, 7*(1), 1–17. Springeropen. https://doi.org/10.1186/s40537-020-00317-6

Shah, G., Shah, A., & Shah, M. (2019). Panacea of challenges in real-world application of big data analytics in healthcare sector. *Journal of Data, Information and Management*, 107–116. https://doi.org/10.1007/s42488-019-00010-1

Shaik, T., Tao, X., Li, L., Xie, H., & Velásquez, J. D. (2024). A survey of multimodal information fusion for smart healthcare: Mapping the journey from data to wisdom. *Information Fusion, 102*, 102040. https://doi.org/10.1016/j.inffus.2023.102040

Shaikh, B. (2023, July 26). *CatBoost: A Solution for Building Model with Categorical Data*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2023/07/catboost-building-model-with-categorical-data/

Shankar, B. S., & Shewale, P. L. (2024). DATA CLEANING TECHNIQUES AND THEIR IMPACT ON MODEL ACCURACY. *International Research Journal of Modernization in Engineering Technology and Science, 6*(12). IRJMETS. e-issn: 2582-5208

Sharma, S., & Osei-Bryson, K.-M. (2009). Framework for formal implementation of the business understanding phase of data mining projects. *Expert Systems with Applications*, *36*(2, Part 2), 4114–4124. https://doi.org/10.1016/j.eswa.2008.03.021

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5, 13–22.

Shi, X., Cheng, Y., & Xue, D. (2019). Classification Algorithm of Urban Point Cloud Data based on LightGBM. *IOP Conference Series: Materials Science and Engineering, 631*, 052041. https://doi.org/10.1088/1757-899x/631/5/052041

Shoetan, P. O., Oyewole, A. T., Okoye, C. C., & Ofodile, O. C. (2024). REVIEWING THE ROLE OF BIG DATA ANALYTICS IN FINANCIAL FRAUD DETECTION. *Finance & Accounting Research Journal, 6*(3), 384–394. https://doi.org/10.51594/farj.v6i3.899

Siddique, A., Jan, A., Majeed, F., Qahmash, A. I., Quadri, N. N., & Wahab, M. O. A. (2021). Predicting Academic Performance Using an Efficient Model Based on Fusion of Classifiers. *Applied Sciences, 11*(24), 11845. https://doi.org/10.3390/app112411845

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research, 70*(1), 263–286.

Sivarajah, U., Kumar, S., Kumar, V., Chatterjee, S., & Li, J. (2024). A study on big data analytics and innovation: From technological and business cycle perspectives. *Technological Forecasting and Social Change, 202*. https://doi.org/10.1016/j.techfore.2024.123328

Soomro, A. A., Mokhtar, A. A., Hussin, H. B., Lashari, N., Oladosu, T. L., Jameel, S. M., & Inayat, M. (2024). Analysis of machine learning models and data sources to forecast burst pressure of petroleum

corroded pipelines: A comprehensive review. *Engineering Failure Analysis, 155*, 107747. https://doi.org/10.1016/j.engfailanal.2023.107747

South African Medical Research Council (SAMRC). (2020). STRATEGIC PLAN. In *Parliament of South Africa*. Parliament of South Africa. https://www.parliament.gov.za/storage/app/media/Docs/tpap/5f50e3e1-6740-4666-a403-565b99c77ee4.pdf. https://static.pmg.org.za/SAMRC_Strategic_Plan_2025-2030.pdf.

Steyerberg, E. W. (2019). *Clinical prediction models* (2nd ed.). Springer.

Stoumpos, A. I., Kitsios, F., & Talias, M. A. (2023). Digital Transformation in Healthcare: Technology Acceptance and Its Applications. *International Journal of Environmental Research and Public Health, 20*(4). https://doi.org/10.3390/ijerph20043407

Subasi, A. (2020a). Data preprocessing. In *Practical Machine Learning for Data Analysis Using Python* (pp. 27–89). https://doi.org/10.1016/b978-0-12-821379-7.00002-3

Subasi, A. (2020b*). Practical machine learning for data analysis using python*. (pp. 1–26). Academic Press.

Sun, X., He, Y., Wu, D., & Huang, J. Z. (2023). Survey of Distributed Computing Frameworks for Supporting Big Data Analysis. *Big Data Mining and Analytics, 6*(2), 154–169. https://doi.org/10.26599/bdma.2022.9020014

Sun, Z., Strang, K., & Li, R. (2018). Big Data with Ten Big Characteristics. Proceedings of the 2nd *International Conference on Big Data Research - ICBDR 2018*. https://doi.org/10.1145/3291801.3291822

Sunagar, P., & Alam, M. (2020). *Big Data Application - an overview | ScienceDirect Topics*. Www.sciencedirect.com. https://www.sciencedirect.com/topics/computer-science/big-data-application

Sundaram, J., Gowri, K., Devaraju, S., Gokuldev, S., Jayaprakash, S., Anandaram, H., Manivasagan, C., & Thenmozhi, M. (2023). An Exploration of Python Libraries in Machine Learning Models for Data Science. *Advances in Computational Intelligence and Robotics Book Series*, 1–31. https://doi.org/10.4018/978-1-6684-8696-2.ch001

Surbakti, F. P. S., Wang, W., Indulska, M., & Sadiq, S. (2019). Factors influencing effective use of big data: A research framework. *Information & Management, 57*(1), 103146. https://doi.org/10.1016/j.im.2019.02.001

Susmaga, R. (2004). Confusion Matrix Visualization. *Intelligent Information Processing and Web Mining*, 107–116. https://doi.org/10.1007/978-3-540-39985-8_12

Syed, S., Morseth, B., Hopstock, L. A., & Horsch, A. (2021). A novel algorithm to detect non-wear time from raw accelerometer data using deep convolutional neural networks. *Scientific Reports, 11*(1). https://doi.org/10.1038/s41598-021-87757-z

Talaviya, A. (2023, October 30). *CRISP-DM framework: A foundational data mining process model*. Medium. https://medium.com/@avikumart_/crisp-dm-framework-a-foundational-data-mining-process-model-86fe642da18c

Tiwari, N. (2021, June 14). *Data Cleaning Using Pandas | Data Cleaning for Beginners*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/data-cleaning-using-pandas/

Tiwari, V. (2024). ANALYSIS OF PHYSICO-CHEMICAL PARAMETERS OF RIVER GARRA AT SHAHJAHANPUR (U.P.). Knowledgeable Research: A Multidisciplinary Peer-Reviewd Refereed Journal, 2(10), 55–67. http://dx.doi.org/10.57067/0zr57x43

Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, *25*(1), 44–56. https://www.nature.com/articles/s41591-018-0300-7

Tosi, D., Kokaj, R., & Roccetti, M. (2024). 15 years of Big Data: a systematic literature review. *Journal of Big Data, 11*(1). https://doi.org/10.1186/s40537-024-00914-9

Trevisan, V. (2022, March 17). *Target-encoding Categorical Variables | Towards Data Science*. Towards Data Science. https://towardsdatascience.com/dealing-with-categorical-variables-by-using-target-encoder-a0f1733a4c69/

Tripathi, S., Muhr, D., Brunner, M., Jodlbauer, H., Dehmer, M., & Emmert-Streib, F. (2021). Ensuring the Robustness and Reliability of Data-Driven Knowledge Discovery Models in Production and Manufacturing. *Frontiers in Artificial Intelligence*, *4*. https://doi.org/10.3389/frai.2021.576892

Turgay, S., & Özçelik, Ö. F. (2023). Data-Driven Approaches to Hospital Capacity Planning and Management. *Information and Knowledge Management, 4*(2), 6–14. https://doi.org/10.23977/infkm.2023.040202

Tyagi, N., & Bhushan, B. (2023). Demystifying the Role of Natural Language Processing (NLP) in Smart City Applications: Background, Motivation, Recent Advances, and Future Research Directions. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-023-10312-8

University of Newcastle Library. (2023). *Research Methods: What are research methods?* Newcastle.edu.au. https://libguides.newcastle.edu.au/researchmethods

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology, 6*(1).

Van der Pol, N., Ntinga, X., Mkhize, M., & van Heerden, A. (2021). A scoping review of mHealth use in South Africa. *Development Southern Africa, 39*(4), 1–13. https://doi.org/10.1080/0376835x.2021.1904837

Varoquaux, G., & Colliot, O. (2023). Evaluating Machine Learning Models and Their Diagnostic Value. *Neuromethods*, 601–630. https://doi.org/10.1007/978-1-0716-3195-9_20

Vijay, V., Sharma, V., Srivastava, V., & Kumar, V. (2024). A Comparative Study on Hadoop MapReduce and Apache Spark Framework for Big Data Analytics. International Journal of Research Publication and Reviews, 5(2), 3228–3232. https://doi.org/10.55248/gengpi.5.0224.0601

Vujovic, Ž. Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, *12*(6). https://doi.org/10.14569/ijacsa.2021.0120670

Walker, D. M., Tarver, W. L., Jonnalagadda, P., Ranbom, L., Ford, E. W., & Rahurkar, S. (2023). Perspectives on challenges and opportunities for interoperability: Findings from key informant interviews with stakeholders in ohio. *JMIR Medical Informatics, 11*(11), e43848. https://doi.org/10.2196/43848

Wang, Y., Kung, L., & Byrd, T. A. (2018). Big Data analytics: Understanding Its Capabilities and Potential Benefits for Healthcare Organizations. *Technological Forecasting and Social Change, 126*(1), 3–13. https://www.sciencedirect.com/science/article/pii/S0040162516000500

Wang, W., Jin, Y.-H., Liu, M., He, Q., Xu, J.-Y., Wang, M.-Q., Li, G.-W., Fu, B., Yan, S.-Y., Zou, K., & Sun, X. (2024). Guidance of development, validation, and evaluation of algorithms for populating health status in observational studies of routinely collected data (DEVELOP-RCD). *Military Medical Research, 11*(1). https://doi.org/10.1186/s40779-024-00559-y

Wani, F. J., Rizvi, S. E. H., Sharma, M. K., & Bhat, M. I. J. (2018). A study on cross validation for model selection and estimation. *INTERNATIONAL JOURNAL of AGRICULTURAL SCIENCES, 14*(1), 165–172. https://doi.org/10.15740/has/ijas/14.1/165-172

Welbotha, D., & Van Den Berg, C. (2024). Big data analytics capabilities and the organisational performance of South African retailers. *South African Computer Journal, 36*(2), 103–123. https://doi.org/10.18489/sacj.v36i2.17937

Wiens, M., Verone-Boyle, A., Henscheid, N., Podichetty, J. T., & Burton, J. (2025). A Tutorial and Use Case Example of the eXtreme Gradient Boosting (XGBoost) Artificial Intelligence Algorithm for Drug Development Applications. *Clinical and Translational Science, 18*(3). https://doi.org/10.1111/cts.70172

Williamson, S. M., & Prybutok, V. (2024). Balancing Privacy and Progress: A Review of Privacy Challenges, Systemic Oversight, and Patient Perceptions in AI-Driven Healthcare. *Applied Sciences, 14*(2), 675. https://doi.org/10.3390/app14020675

Willie, M. M., & Maqbool, M. (2023, March 31). *Access to Public Health Services in South Africa's Rural Eastern Cape Province*. Social Science Research Network. https://doi.org/10.2139/ssrn.4405870

Witter, S., van der Merwe, M., Twine, R., Mabetha, D., Hove, J., Tollman, S. M., & D'Ambruoso, L. (2024). Opening decision spaces: A case study on the opportunities and constraints in the public health sector of Mpumalanga Province, South Africa. *PloS One, 19*(7), e0304775–e0304775. https://doi.org/10.1371/journal.pone.0304775

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259.

Yan, F., & Feng, Y. (2022). A two-stage stacked-based heterogeneous ensemble learning for cancer survival prediction. *Complex & Intelligent Systems*. https://doi.org/10.1007/s40747-022-00791-w

Yan, A. S., Maguire, T. K., & Chen, J. (2025). Revisiting the rural and urban divide in hospital health information technology adoption: Evidence from 2023. *The Journal of Rural Health*, *41*(4). https://doi.org/10.1111/jrh.70100

Yehoshua, R. (2023, August 9). *XGBoost: The Definitive Guide (Part 1) - TDS Archive - Medium*. Medium; TDS Archive. https://medium.com/data-science/xgboost-the-definitive-guide-part-1-cc24d2dcd87a

Yu, M., Yang, C., & Li, Y. (2018). Big Data in Natural Disaster Management: A Review. *Geosciences, 8*(5), 165. https://doi.org/10.3390/geosciences8050165

Zali, M. (2023). *South Africa seeks to improve healthcare sector as burden of disease continues to rise*. Alliance for Science. https://allianceforscience.org/blog/2023/07/south-africa-seeks-to-improve-healthcare-sector-as-burden-of-disease-continues-to-rise/

Zaripova, R., Kosulin, V., Shkinderov, M., & Rakhmatullin, I. (2023). Unlocking the potential of artificial intelligence for big data analytics. *E3S Web of Conferences, 460*(460), 04011. International Scientific Conference on Biotechnology and Food Technology (BFT-2023). https://doi.org/10.1051/e3sconf/202346004011

Zheng, X., Jia, J., Dong, S., Wang, Y., Lu, R., Chen, Y., & Wang, Y. (2025). Training and inference Time Efficiency Assessment Framework for machine learning algorithms: A case study for hyperspectral image classification. *International Journal of Applied Earth Observation and Geoinformation*, *141*, 104591–104591. https://doi.org/10.1016/j.jag.2025.104591

Zhou, K., Wang, W., Wu, C., & Hu, T. (2020). Practical evaluation of encrypted traffic classification based on a combined method of entropy estimation and neural networks. *Etri Journal, 42*(3), 311–323. https://doi.org/10.4218/etrij.2019-0190

APPENDICES

Appendix A: Final Model Code

```
#CODE MODEL
import pandas as pd
import numpy as np
import shap
import matplotlib
matplotlib.use('Agg')  # Use non-interactive backend to avoid Tkinter dependency
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
```

```
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_curve,
auc, precision_recall_curve
from imblearn.combine import SMOTEENN
import joblib
import warnings
import time  # Added for measuring computation time

warnings.filterwarnings('ignore')
np.random.seed(42)

# Record the start time for overall computation
overall_start_time = time.time()

# === 1. LOAD & PREPARE DATA (CRISP-DM: Business Understanding, Data
Understanding) ===
print("=== Data Understanding ===")
# Load the dataset (150,000 instances)
data = pd.read_csv('Final_healthcare_big_data.csv')
print("Dataset loaded successfully.")

# Data Overview Summary
print("\nData Overview Summary:")
print(f"Dataset Shape: {data.shape}")
print("\nBasic Statistics for Numerical Columns:")
print(data.describe())
print("\nColumn Information:")
print(data.info())
print("\nClass Distribution of Target Variable ('Outcome'):")
print(data['Outcome'].value_counts())


# Snippet to Check for missing values
missing_values = data.isnull().sum()
print("\nMissing Values in Each Column:")
print(missing_values)
```

```
print("\nStatement: The dataset has no missing values." if missing_values.sum()
== 0 else "\nStatement: The dataset contains missing values.")

# Our code to check for duplicates and drop them
initial_shape = data.shape
data = data.drop_duplicates()
final_shape = data.shape
print(f"\nInitial dataset shape: {initial_shape}")
print(f"Dataset shape after dropping duplicates: {final_shape}")
print(f"Number of duplicates dropped: {initial_shape[0] - final_shape[0]}")

# Visualize dataset datatypes
print("\nDataset Column Datatypes:")
print(data.dtypes)
dtype_counts = data.dtypes.value_counts()
print("\nDatatype Counts:")
print(dtype_counts)

# Figure: Visualize column datatypes
plt.figure(figsize=(8, 6))
sns.barplot(x=dtype_counts.values, y=dtype_counts.index, hue=dtype_counts.index,
palette='viridis', legend=False)
plt.title('Dataset Column Datatypes (Two Object and Int64 Types)')
plt.xlabel('Number of Columns')
plt.ylabel('Datatype')
plt.tight_layout()
plt.savefig('dataset_datatypes.png')
plt.close()
print("\nFigure saved as 'dataset_datatypes.png' showing columns with two object
and int64 datatypes.")

# Feature engineering (CRISP-DM: Data Preparation)
print("\n=== Data Preparation ===")
data['Resource_Bed_Interaction'] = data['Resource_Utilization'] *
data['Bed_Occupancy_Rate']
data['Wait_Length_Interaction'] = data['Wait_Time_Minutes'] *
data['Length_of_Stay']
data['Log_Wait_Time'] = np.log1p(data['Wait_Time_Minutes'])
data['Log_Resource_Utilization'] = np.log1p(data['Resource_Utilization'])
data['Satisfaction_Staff_Interaction'] = data['Satisfaction_Rating'] *
data['Staff_Count']
print("Feature engineering completed.")

# Separate features and target
X = data.drop('Outcome', axis=1)
```

```
y = data['Outcome']

# Encode the target variable
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
print("\nLabel Encoding Mapping:", dict(zip(label_encoder.classes_,
label_encoder.transform(label_encoder.classes_))))

# === 2. PREPROCESSING (CRISP-DM: Data Preparation) ===
numerical_features = ['Daily_Patient_Inflow', 'Emergency_Response_Time_Minutes',
'Staff_Count',
                      'Bed_Occupancy_Rate', 'Visit_Frequency',
'Wait_Time_Minutes',
                      'Length_of_Stay', 'Previous_Visits',
'Resource_Utilization', 'Age',
                      'Resource_Bed_Interaction', 'Wait_Length_Interaction',
'Log_Wait_Time',
                      'Log_Resource_Utilization',
'Satisfaction_Staff_Interaction']
categorical_features = ['Treatment_Outcome', 'Equipment_Availability',
'Medicine_Stock_Level',
                        'Comorbidities', 'Disease_Category']
ordinal_features = ['Satisfaction_Rating']

num_pipe = Pipeline([('imputer', SimpleImputer(strategy='median')), ('scaler',
StandardScaler())])
cat_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
                     ('onehot', OneHotEncoder(drop='first', sparse_output=False,
handle_unknown='ignore'))])
ord_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
('scaler', StandardScaler())])

preprocessor = ColumnTransformer([
    ('num', num_pipe, numerical_features),
    ('cat', cat_pipe, categorical_features),
    ('ord', ord_pipe, ordinal_features)
])

X_processed = preprocessor.fit_transform(X)
cat_feature_names =
preprocessor.named_transformers_['cat'].named_steps['onehot'].get_feature_names_o
ut(categorical_features)
feature_names = numerical_features + list(cat_feature_names) + ordinal_features
print(f"\nNumber of features after preprocessing: {len(feature_names)}")
print("Feature names:", feature_names)
```

```
print(f"Shape of X_processed: {X_processed.shape}")

# Drop redundant features
X_df = pd.DataFrame(X_processed, columns=feature_names)
X_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'], axis=1, inplace=True,
errors='ignore')
print("\nDropped redundant features ('Wait_Time_Minutes',
'Resource_Utilization').")
print(f"Shape of X_df after dropping redundant features: {X_df.shape}")

# === 3. BALANCE DATA AND SPLIT INTO TRAINING AND TEST SETS (CRISP-DM: Data
Preparation) ===
smote_enn = SMOTEENN(random_state=42)
X_balanced, y_balanced = smote_enn.fit_resample(X_df.values, y_encoded)
print("\nClass distribution after SMOTEENN balancing:")
print(pd.Series(y_balanced).value_counts())

# Split data into training and test sets (80-20 split)
X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced,
test_size=0.2,
                                                    stratify=y_balanced,
random_state=42)
print(f"\nShape of X_train: {X_train.shape}, X_test: {X_test.shape}")
print(f"Class distribution in training
set:\n{pd.Series(y_train).value_counts()}")
print(f"Class distribution in test set:\n{pd.Series(y_test).value_counts()}")

# === 4. DEFINE AND TRAIN STACKING CLASSIFIER (CRISP-DM: Modeling) ===
print("\n=== Modeling ===")
class_weights = {0: 5.0, 1: 1.0, 2: 1.0}
rf_model = RandomForestClassifier(n_estimators=500, max_depth=None,
min_samples_split=2,
                                  min_samples_leaf=2, max_features='log2',
                                  class_weight=class_weights, random_state=42)
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42,
                          n_estimators=1000, max_depth=10, learning_rate=0.1,
                          subsample=0.9, colsample_bytree=1.0, reg_lambda=1,
reg_alpha=1)
lgbm_model = LGBMClassifier(class_weight=class_weights, random_state=42)
catboost_model = CatBoostClassifier(verbose=0, early_stopping_rounds=10,
auto_class_weights='Balanced')

stacking = StackingClassifier(
    estimators=[
```

```
        ('rf', rf_model),
        ('xgb', xgb_model),
        ('lgbm', lgbm_model),
        ('cat', catboost_model)
    ],
    final_estimator=XGBClassifier(use_label_encoder=False,
eval_metric='mlogloss', random_state=42),
    cv=5, n_jobs=1
)

print(f"Processing Training Set: Training Stacking Classifier on
{X_train.shape[0]} instances...")
stacking_start_time = time.time()  # Record start time for Stacking Classifier
stacking.fit(X_train, y_train)
stacking_end_time = time.time()  # Record end time for Stacking Classifier
stacking_training_time = stacking_end_time - stacking_start_time
print(f"Training time for Stacking Classifier: {stacking_training_time:.2f}
seconds")
print("Stacking classifier trained successfully.")

# Evaluate on validation set (for monitoring)
# Note: No explicit validation set with 80-20 split; validation handled by cv=5
internally
print("Note: Validation is handled internally via 5-fold cross-validation in
StackingClassifier.")

# === 5. EVALUATION (CRISP-DM: Evaluation) ===
print("\n=== Evaluation ===")
print(f"Processing Test Set: Evaluating Stacking Classifier on {X_test.shape[0]}
instances...")
y_pred = stacking.predict(X_test)
y_pred_proba = stacking.predict_proba(X_test)

print("Evaluation Phase: Generating classification report for Stacking Classifier
on test set...")
print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))

# === 6. THRESHOLD ADJUSTMENT ===
print("\n=== Threshold Adjustment ===")
thresholds = {}
y_pred_adj = np.copy(y_pred)
for cls in range(3):
    precision, recall, thresh = precision_recall_curve(y_test == cls,
y_pred_proba[:, cls])
```

```
    f1_scores = 2 * (precision * recall) / (precision + recall + 1e-8)
    thresholds[cls] = thresh[np.argmax(f1_scores)]

for i in range(len(y_pred_proba)):
    for cls in range(3):
        if y_pred_proba[i, cls] >= thresholds[cls]:
            y_pred_adj[i] = cls
            break

print("Evaluation Phase: Generating classification report after threshold
adjustment...")
print("\nClassification Report after Threshold Adjustment:")
print(classification_report(y_test, y_pred_adj,
target_names=label_encoder.classes_))

# === 7. SHAP INTERPRETABILITY (CRISP-DM: Evaluation) ===
print("\nComputing SHAP values for interpretability...")
xgb_model.fit(X_train, y_train)  # Retrain for SHAP
shap_explainer = shap.TreeExplainer(xgb_model)
X_test_df = pd.DataFrame(X_test, columns=X_df.columns)
shap_values = shap_explainer.shap_values(X_test_df)

# SHAP Feature Importance
shap.summary_plot(shap_values, X_test_df, plot_type="bar", show=False)
plt.title("SHAP Feature Importance")
plt.tight_layout()
plt.savefig('shap_feature_importance_1.png')
plt.close()

# SHAP Summary Plot
shap.summary_plot(shap_values, X_test_df, show=False)
plt.title("SHAP Summary Plot")
plt.tight_layout()
plt.savefig('shap_summary_plot_1.png')
plt.close()

# === 8. VISUALIZATIONS (CRISP-DM: Evaluation) ===
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred_adj)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
yticklabels=label_encoder.classes_)
plt.title('Confusion Matrix (Threshold Adjusted)')
plt.xlabel('Predicted Label')
```

```
plt.ylabel('True Label')
plt.tight_layout()
plt.savefig('confusion_matrix_1.png')
plt.close()

# ROC Curves
plt.figure(figsize=(8, 6))
for i in range(3):
    fpr, tpr, _ = roc_curve(y_test == i, y_pred_proba[:, i])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{label_encoder.classes_[i]} (AUC =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Multi-class ROC Curves')
plt.legend(loc='best')
plt.grid()
plt.tight_layout()
plt.savefig('roc_curves_1.png')
plt.close()

# === 9. SAVE MODELS (CRISP-DM: Deployment) ===
print("\n=== Deployment ===")
joblib.dump(stacking, 'final_healthcare_model_1.pkl')
joblib.dump(preprocessor, 'preprocessor_1.pkl')
joblib.dump(label_encoder, 'label_encoder_1.pkl')
print("\nModels and preprocessors saved successfully.")

# Record the end time for overall computation
overall_end_time = time.time()
overall_time = overall_end_time - overall_start_time
print(f"\nOverall Computation Time: {overall_time:.2f} seconds")

# Save training time for thesis documentation
training_times_df = pd.DataFrame({'Model': ['Stacking Classifier'], 'Training
Time (seconds)': [stacking_training_time]})
training_times_df['Training Time (seconds)'] = training_times_df['Training Time
(seconds)'].round(2)
training_times_df.to_csv('training_times_final_stacked_1.csv', index=False)
print("\nTraining time saved as 'training_times_final_stacked_1.csv'")
print(training_times_df)
```

Appendix B: Trial Model Code

```
import pandas as pd
import numpy as np
import shap
import matplotlib
matplotlib.use('Agg')  # Use non-interactive backend to avoid Tkinter dependency
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, StratifiedKFold,
cross_val_score
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_curve,
auc, precision_recall_curve
from imblearn.combine import SMOTEENN
import joblib
import warnings
import time  # Added for measuring computation time

warnings.filterwarnings('ignore')
np.random.seed(42)

# Record the start time for overall computation
overall_start_time = time.time()

# === 1. LOAD & PREPARE DATA (CRISP-DM: Business Understanding, Data
Understanding) ===
print("=== Data Understanding ===")
# Load the dataset (150,000 instances)
data = pd.read_csv('Final_healthcare_big_data.csv')
print("Dataset loaded successfully.")

# Data Overview Summary
print("\nData Overview Summary:")
print(f"Dataset Shape: {data.shape}")  # Should show (150000, number_of_columns)
print("\nBasic Statistics for Numerical Columns:")
print(data.describe())
```

```
print("\nColumn Information:")
print(data.info())
print("\nClass Distribution of Target Variable ('Outcome'):")
print(data['Outcome'].value_counts())  # Should show Improved: 9045, Unchanged:
52965, Worsened: 87990

# Check for missing values
missing_values = data.isnull().sum()
print("\nMissing Values in Each Column:")
print(missing_values)
print("\nStatement: The dataset has no missing values." if missing_values.sum()
== 0 else "\nStatement: The dataset contains missing values.")

# Check for duplicates and drop them
initial_shape = data.shape
data = data.drop_duplicates()
final_shape = data.shape
print(f"\nInitial dataset shape: {initial_shape}")
print(f"Dataset shape after dropping duplicates: {final_shape}")
print(f"Number of duplicates dropped: {initial_shape[0] - final_shape[0]}")

# Visualize dataset datatypes
print("\nDataset Column Datatypes:")
print(data.dtypes)
dtype_counts = data.dtypes.value_counts()
print("\nDatatype Counts:")
print(dtype_counts)

# Figure: Visualize column datatypes
plt.figure(figsize=(8, 6))
sns.barplot(x=dtype_counts.values, y=dtype_counts.index, hue=dtype_counts.index,
palette='viridis', legend=False)
plt.title('Dataset Column Datatypes (Two Object and Int64 Types)')
plt.xlabel('Number of Columns')
plt.ylabel('Datatype')
plt.tight_layout()
plt.savefig('dataset_datatypes.png')
plt.close()
print("\nFigure saved as 'dataset_datatypes.png' showing columns with two object
and int64 datatypes.")

# Feature engineering (CRISP-DM: Data Preparation)
print("\n=== Data Preparation ===")
data['Resource_Bed_Interaction'] = data['Resource_Utilization'] *
data['Bed_Occupancy_Rate']
```

```
data['Wait_Length_Interaction'] = data['Wait_Time_Minutes'] *
data['Length_of_Stay']
data['Log_Wait_Time'] = np.log1p(data['Wait_Time_Minutes'])
data['Log_Resource_Utilization'] = np.log1p(data['Resource_Utilization'])
data['Satisfaction_Staff_Interaction'] = data['Satisfaction_Rating'] *
data['Staff_Count']
print("Feature engineering completed.")

# Separate features and target
X = data.drop('Outcome', axis=1)
y = data['Outcome']

# Encode the target variable
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
print("\nLabel Encoding Mapping:", dict(zip(label_encoder.classes_,
label_encoder.transform(label_encoder.classes_))))

# === 2. PREPROCESSING (CRISP-DM: Data Preparation) ===
numerical_features = ['Daily_Patient_Inflow', 'Emergency_Response_Time_Minutes',
'Staff_Count',
                      'Bed_Occupancy_Rate', 'Visit_Frequency',
'Wait_Time_Minutes',
                      'Length_of_Stay', 'Previous_Visits',
'Resource_Utilization', 'Age',
                      'Resource_Bed_Interaction', 'Wait_Length_Interaction',
'Log_Wait_Time',
                      'Log_Resource_Utilization',
'Satisfaction_Staff_Interaction']
categorical_features = ['Treatment_Outcome', 'Equipment_Availability',
'Medicine_Stock_Level',
                        'Comorbidities', 'Disease_Category']
ordinal_features = ['Satisfaction_Rating']

num_pipe = Pipeline([('imputer', SimpleImputer(strategy='median')), ('scaler',
StandardScaler())])
cat_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
                     ('onehot', OneHotEncoder(drop='first', sparse_output=False,
handle_unknown='ignore'))])
ord_pipe = Pipeline([('imputer', SimpleImputer(strategy='most_frequent')),
('scaler', StandardScaler())])

preprocessor = ColumnTransformer([
    ('num', num_pipe, numerical_features),
    ('cat', cat_pipe, categorical_features),
```

```
    ('ord', ord_pipe, ordinal_features)
])

X_processed = preprocessor.fit_transform(X)
cat_feature_names =
preprocessor.named_transformers_['cat'].named_steps['onehot'].get_feature_names_o
ut(categorical_features)
feature_names = numerical_features + list(cat_feature_names) + ordinal_features
print(f"\nNumber of features after preprocessing: {len(feature_names)}")
print("Feature names:", feature_names)
print(f"Shape of X_processed: {X_processed.shape}")

# Drop redundant features
X_df = pd.DataFrame(X_processed, columns=feature_names)
X_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'], axis=1, inplace=True,
errors='ignore')
print("\nDropped redundant features ('Wait_Time_Minutes',
'Resource_Utilization').")
print(f"Shape of X_df after dropping redundant features: {X_df.shape}")

# === 3. BALANCE DATA (CRISP-DM: Data Preparation) ===
smote_enn = SMOTEENN(random_state=42)
X_balanced, y_balanced = smote_enn.fit_resample(X_df.values, y_encoded)
print("\nClass distribution after SMOTEENN balancing:")
print(pd.Series(y_balanced).value_counts())

X_train, X_test, y_train, y_test = train_test_split(X_balanced, y_balanced,
test_size=0.2,
                                                    stratify=y_balanced,
random_state=42)
print(f"\nShape of X_train: {X_train.shape}, X_test: {X_test.shape}")

# === 4. DEFINE AND EVALUATE INDIVIDUAL MODELS (CRISP-DM: Modeling & Evaluation)
===
print("\n=== Individual Model Evaluation ===")

# Define the individual models (same as in the stacking classifier)
class_weights = {0: 5.0, 1: 1.0, 2: 1.0}
rf_model = RandomForestClassifier(n_estimators=500, max_depth=None,
min_samples_split=2,
                                  min_samples_leaf=2, max_features='log2',
                                  class_weight=class_weights, random_state=42)
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
random_state=42,
                          n_estimators=1000, max_depth=10, learning_rate=0.1,
```

```
                          subsample=0.9, colsample_bytree=1.0, reg_lambda=1,
reg_alpha=1)
lgbm_model = LGBMClassifier(class_weight=class_weights, random_state=42)
catboost_model = CatBoostClassifier(verbose=0, early_stopping_rounds=10,
auto_class_weights='Balanced')

# List of models to evaluate
models = {
    'Random Forest': rf_model,
    'XGBoost': xgb_model,
    'LightGBM': lgbm_model,
    'CatBoost': catboost_model
}

# Dictionary to store performance metrics, confusion matrices, and training times
performance_results = []
confusion_matrices = {}
training_times = {}

# Evaluate each model independently
for model_name, model in models.items():
    print(f"\nTraining {model_name}...")
    start_time = time.time()  # Record start time for this model
    model.fit(X_train, y_train)
    end_time = time.time()  # Record end time
    training_time = end_time - start_time
    training_times[model_name] = training_time
    print(f"Training time for {model_name}: {training_time:.2f} seconds")

    y_pred = model.predict(X_test)

    # Compute confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    confusion_matrices[model_name] = cm

    # Generate and save heatmap for confusion matrix
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=label_encoder.classes_,
yticklabels=label_encoder.classes_,
                annot_kws={"size": 12}, cbar_kws={'label': 'Number of
Instances'})
    plt.xlabel('Predicted', fontsize=12)
    plt.ylabel('Actual', fontsize=12)
    plt.title(f'Confusion Matrix - {model_name}', fontsize=14, pad=20)
```

```
    plt.tight_layout()
    plt.savefig(f'confusion_matrix_{model_name.lower().replace(" ", "_")}.png',
dpi=300)
    plt.close()
    print(f"Saved confusion matrix heatmap for {model_name} as
'confusion_matrix_{model_name.lower().replace(' ', '_')}.png'")

    # Compute and print detailed classification report
    print(f"\nDetailed Classification Report for {model_name}:")
    print(classification_report(y_test, y_pred,
target_names=label_encoder.classes_))

    # Extract only overall weighted average metrics for combined table
    report = classification_report(y_test, y_pred,
target_names=label_encoder.classes_, output_dict=True)
    metrics = {
        'Model': model_name,
        'Accuracy': report['accuracy'],
        'Precision': report['weighted avg']['precision'],
        'Recall': report['weighted avg']['recall'],
        'F1-Score': report['weighted avg']['f1-score'],
        'Training Time (seconds)': training_time
    }
    performance_results.append(metrics)

# === 5. TRAIN AND EVALUATE STACKING CLASSIFIER (CRISP-DM: Modeling & Evaluation)
===
print("\n=== Stacking Classifier Modeling & Evaluation ===")
stacking_start_time = time.time()  # Record start time for stacking classifier
stacking = StackingClassifier(
    estimators=[
        ('rf', rf_model),
        ('xgb', xgb_model),
        ('lgbm', lgbm_model),
        ('cat', catboost_model)
    ],
    final_estimator=XGBClassifier(use_label_encoder=False,
eval_metric='mlogloss', random_state=42),
    cv=5, n_jobs=1
)

stacking.fit(X_train, y_train)
stacking_end_time = time.time()  # Record end time for stacking classifier
stacking_training_time = stacking_end_time - stacking_start_time
training_times['Stacking Classifier'] = stacking_training_time
```

```
print(f"Training time for Stacking Classifier: {stacking_training_time:.2f}
seconds")
print("Stacking classifier trained successfully.")

# Evaluate Stacking Classifier
y_pred_stacking = stacking.predict(X_test)
y_pred_proba_stacking = stacking.predict_proba(X_test)

# Compute and print detailed classification report
print("\nDetailed Classification Report for Stacking Classifier:")
print(classification_report(y_test, y_pred_stacking,
target_names=label_encoder.classes_))

# Extract only overall weighted average metrics for combined table
report_stacking = classification_report(y_test, y_pred_stacking,
target_names=label_encoder.classes_, output_dict=True)
metrics_stacking = {
    'Model': 'Stacking Classifier',
    'Accuracy': report_stacking['accuracy'],
    'Precision': report_stacking['weighted avg']['precision'],
    'Recall': report_stacking['weighted avg']['recall'],
    'F1-Score': report_stacking['weighted avg']['f1-score'],
    'Training Time (seconds)': stacking_training_time
}
performance_results.append(metrics_stacking)

# Convert performance results to a DataFrame and save as a combined table
performance_df = pd.DataFrame(performance_results)
performance_df = performance_df.round(3)  # Round to 3 decimal places for
readability
print("\nCombined Performance Evaluation Table for All Models (Including Training
Times):")
print(performance_df)
performance_df.to_csv('combined_model_performance_with_times.csv', index=False)
print("Combined performance table saved as
'combined_model_performance_with_times.csv'")

# === 6. CREATE BAR GRAPH COMPARISON BASED ON ACCURACY ===
print("\n=== Generating Bar Graph Comparison Based on Accuracy ===")
models = performance_df['Model']
accuracies = performance_df['Accuracy']

plt.figure(figsize=(8, 6))
bars = plt.bar(models, accuracies, color='skyblue')
plt.xlabel('Models')
```

```
plt.ylabel('Accuracy')
plt.title('Comparison of Model Accuracy')
plt.xticks(rotation=45)
plt.ylim(0, 1)  # Ensure y-axis ranges from 0 to 1

# Add percentage labels on top of each bar
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height,
             f'{height:.1%}',
             ha='center', va='bottom')

plt.tight_layout()
plt.savefig('model_accuracy_comparison.png', dpi=300)
plt.close()
print("Bar graph saved as 'model_accuracy_comparison.png'")

# === 7. THRESHOLD ADJUSTMENT ===
print("\n=== Threshold Adjustment ===")
thresholds = {}
y_pred_adj = np.copy(y_pred_stacking)
for cls in range(3):
    precision, recall, thresh = precision_recall_curve(y_test == cls,
y_pred_proba_stacking[:, cls])
    f1_scores = 2 * (precision * recall) / (precision + recall + 1e-8)
    thresholds[cls] = thresh[np.argmax(f1_scores)]

for i in range(len(y_pred_proba_stacking)):
    for cls in range(3):
        if y_pred_proba_stacking[i, cls] >= thresholds[cls]:
            y_pred_adj[i] = cls
            break

print("\nClassification Report for Stacking Classifier after Threshold
Adjustment:")
print(classification_report(y_test, y_pred_adj,
target_names=label_encoder.classes_))

# Confusion Matrix for Stacking Classifier (Threshold Adjusted)
cm_stacking = confusion_matrix(y_test, y_pred_adj)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_stacking, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
yticklabels=label_encoder.classes_)
plt.title('Confusion Matrix (Stacking Classifier - Threshold Adjusted)')
```

```
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.tight_layout()
plt.savefig('confusion_matrix_stacking_trail.png')
plt.close()
print("Saved confusion matrix heatmap for Stacking Classifier as
'confusion_matrix_stacking_trail.png'")

# === 8. SHAP INTERPRETABILITY (CRISP-DM: Evaluation) ===
print("\nComputing SHAP values for interpretability...")
xgb_model.fit(X_train, y_train)  # Retrain for SHAP
shap_explainer = shap.TreeExplainer(xgb_model)
X_test_df = pd.DataFrame(X_test, columns=X_df.columns)
shap_values = shap_explainer.shap_values(X_test_df)

# SHAP Feature Importance
shap.summary_plot(shap_values, X_test_df, plot_type="bar", show=False)
plt.title("SHAP Feature Importance")
plt.tight_layout()
plt.savefig('shap_feature_importance_trail.png')
plt.close()

# SHAP Summary Plot
shap.summary_plot(shap_values, X_test_df, show=False)
plt.title("SHAP Summary Plot")
plt.tight_layout()
plt.savefig('shap_summary_plot_trail.png')
plt.close()

# === 9. VISUALIZATIONS (CRISP-DM: Evaluation) ===
# ROC Curves
plt.figure(figsize=(8, 6))
for i in range(3):
    fpr, tpr, _ = roc_curve(y_test == i, y_pred_proba_stacking[:, i])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{label_encoder.classes_[i]} (AUC =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Multi-class ROC Curves')
plt.legend(loc='best')
plt.grid()
plt.tight_layout()
plt.savefig('roc_curves_trail_.png')
```

```
plt.close()

# === 10. SAVE MODELS (CRISP-DM: Deployment) ===
print("\n=== Deployment ===")
joblib.dump(stacking, 'final_healthcare_model_trial.pkl')
joblib.dump(preprocessor, 'preprocessor_trial.pkl')
joblib.dump(label_encoder, 'label_encoder.trial.pkl')
print("\nModels and preprocessors saved successfully.")

# Record the end time for overall computation
overall_end_time = time.time()
overall_time = overall_end_time - overall_start_time
print(f"\nOverall Computation Time: {overall_time:.2f} seconds")

# Save training times to a file for thesis documentation (optional, can be
removed if combined table suffices)
training_times_df = pd.DataFrame(list(training_times.items()), columns=['Model',
'Training Time (seconds)'])
training_times_df['Training Time (seconds)'] = training_times_df['Training Time
(seconds)'].round(2)
training_times_df.to_csv('training_times_trial.csv', index=False)
print("\nTraining times saved as 'training_times_trial.csv'")
print(training_times_df)
```

Appendix C: Streamlit App – User Interface Code

```
import streamlit as st
import pandas as pd
import numpy as np
import shap
import matplotlib
matplotlib.use('Agg')  # Use non-interactive backend for Matplotlib
import matplotlib.pyplot as plt
import joblib
import time
import io

# Load the saved model, preprocessor, and label encoder
```

```
model = joblib.load('final_healthcare_model_1.pkl')
preprocessor = joblib.load('preprocessor_1.pkl')
label_encoder = joblib.load('label_encoder_1.pkl')

# Define feature lists (same as in the model)
numerical_features = ['Daily_Patient_Inflow', 'Emergency_Response_Time_Minutes',
'Staff_Count',
                      'Bed_Occupancy_Rate', 'Visit_Frequency',
'Wait_Time_Minutes',
                      'Length_of_Stay', 'Previous_Visits',
'Resource_Utilization', 'Age',
                      'Resource_Bed_Interaction', 'Wait_Length_Interaction',
'Log_Wait_Time',
                      'Log_Resource_Utilization',
'Satisfaction_Staff_Interaction']
categorical_features = ['Treatment_Outcome', 'Equipment_Availability',
'Medicine_Stock_Level',
                        'Comorbidities', 'Disease_Category']
ordinal_features = ['Satisfaction_Rating']

# Define feature names (replicating the training script's logic)
cat_feature_names =
preprocessor.named_transformers_['cat'].named_steps['onehot'].get_feature_names_o
ut(categorical_features)
feature_names = numerical_features + list(cat_feature_names) + ordinal_features

# Dark Mode Toggle
st.sidebar.subheader("Theme Settings")
theme = st.sidebar.selectbox("Select Theme", ["Light", "Dark"])

# Apply custom CSS based on theme selection
if theme == "Dark":
    st.markdown("""
        <style>
        /* Main app background and default text color */
        .stApp {
            background-color: #1E1E1E;
            color: #FFFFFF !important;
        }
        /* Headers */
        h1, h2, h3, h4, h5, h6 {
            color: #FFFFFF !important;
        }
        /* General text elements */
        p, div, span, label {
```

```
        color: #FFFFFF !important;
    }
    /* Form container */
    .stForm {
        background-color: #1E1E1E;
        color: #FFFFFF !important;
    }
    /* Text input fields */
    .stTextInput > div > div > input {
        background-color: #2E2E2E;
        color: #FFFFFF !important;
        border: 1px solid #555555;
    }
    /* Placeholder text for inputs */
    .stTextInput > div > div > input::placeholder {
        color: #AAAAAA !important;
    }
    /* Labels for text inputs */
    .stTextInput > div > div > label {
        color: #FFFFFF !important;
    }
    /* Selectbox (dropdowns) */
    .stSelectbox > div > div > select {
        background-color: #2E2E2E;
        color: #FFFFFF !important;
        border: 1px solid #555555;
    }
    /* Labels for selectbox */
    .stSelectbox > div > div > label {
        color: #FFFFFF !important;
    }
    /* Selectbox options (dropdown menu items) */
    .stSelectbox > div > div > select > option {
        background-color: #2E2E2E;
        color: #FFFFFF !important;
    }
    /* Slider */
    .stSlider > div > div > div > div {
        background-color: #2E2E2E;
        color: #FFFFFF !important;
    }
    /* Slider labels (the numbers/values shown) */
    .stSlider > div > div > div > div > div {
        color: #FFFFFF !important;
    }
```

```
        /* Labels for sliders */
        .stSlider > div > div > label {
            color: #FFFFFF !important;
        }
        /* Buttons */
        .stButton > button {
            background-color: #4CAF50;
            color: white;
            border: none;
        }
        /* DataFrame */
        .stDataFrame {
            background-color: #2E2E2E;
            color: #FFFFFF !important;
        }
        /* DataFrame text */
        .stDataFrame table {
            color: #FFFFFF !important;
        }
        /* Fallback for any other form-related text */
        .stForm * {
            color: #FFFFFF !important;
        }
        </style>
    """, unsafe_allow_html=True)
else:
    st.markdown("""
        <style>
        .stApp {
            background-color: white;
            color: black;
        }
        h1, h2, h3, h4, h5, h6 {
            color: black;
        }
        .stTextInput > div > div > input {
            background-color: white;
            color: black;
        }
        .stSelectbox > div > div > select {
            background-color: white;
            color: black;
        }
        .stSlider > div > div > div > div {
            background-color: white;
```

```
        }
        .stButton > button {
            background-color: #4CAF50;
            color: white;
        }
        .stDataFrame {
            background-color: white;
            color: black;
        }
        </style>
    """, unsafe_allow_html=True)

# Streamlit UI
st.title("Healthcare Outcome Prediction Model Interface")
st.write("Enter patient details manually or upload a file to predict the
healthcare outcome (Improved, Unchanged, Worsened).")

# Function to convert Matplotlib figure to Streamlit image (used for Prediction
Probabilities graph)
def fig_to_streamlit(fig):
    buf = io.BytesIO()
    fig.savefig(buf, format="png", bbox_inches="tight")
    buf.seek(0)
    return buf

# Initialize session state for storing prediction history
if 'prediction_history' not in st.session_state:
    st.session_state.prediction_history = []

# File upload section
st.subheader("Upload Patient Data")
st.markdown("**Supported files**: CSV, XLS, ODS (OpenDocument Spreadsheet).
Ensure the file contains columns matching the required features.")
uploaded_file = st.file_uploader("Choose a file", type=['csv', 'xls', 'ods'])

# Process uploaded file
if uploaded_file is not None:
    start_time = time.time()
    try:
        # Read file based on extension
        file_extension = uploaded_file.name.split('.')[-1].lower()
        if file_extension == 'csv':
            input_data = pd.read_csv(uploaded_file)
        elif file_extension in ['xls', 'xlsx']:
            input_data = pd.read_excel(uploaded_file)
```

```
        elif file_extension == 'ods':
            input_data = pd.read_excel(uploaded_file, engine='odf')

        # Validate required columns
        required_columns = numerical_features + categorical_features +
ordinal_features
        missing_columns = [col for col in required_columns if col not in
input_data.columns]
        if missing_columns:
            st.error(f"Missing required columns: {', '.join(missing_columns)}")
        else:
            st.success("File uploaded successfully! Processing data...")

            # Interactive Data Exploration for Uploaded Files
            st.subheader("Explore Uploaded Data")
            st.dataframe(input_data)

            column_to_plot = st.selectbox("Select a column to visualize",
input_data.columns)
            fig, ax = plt.subplots(figsize=(8, 4))
            input_data[column_to_plot].hist(ax=ax, bins=20, color='purple')
            ax.set_title(f"Histogram of {column_to_plot}")
            ax.set_xlabel(column_to_plot)
            ax.set_ylabel("Frequency")
            plt.tight_layout()
            st.image(fig_to_streamlit(fig), caption=f"Distribution of
{column_to_plot}")
            plt.close(fig)

            # Compute derived features
            input_data['Resource_Bed_Interaction'] =
input_data['Resource_Utilization'] * (input_data['Bed_Occupancy_Rate'] / 100)
            input_data['Wait_Length_Interaction'] =
input_data['Wait_Time_Minutes'] * input_data['Length_of_Stay']
            input_data['Log_Wait_Time'] =
np.log1p(input_data['Wait_Time_Minutes'])
            input_data['Log_Resource_Utilization'] =
np.log1p(input_data['Resource_Utilization'])
            input_data['Satisfaction_Staff_Interaction'] =
input_data['Satisfaction_Rating'] * input_data['Staff_Count']

            # Preprocess and predict
            prediction_start_time = time.time()
            input_processed = preprocessor.transform(input_data)
            input_df = pd.DataFrame(input_processed, columns=feature_names)
```

```
            input_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'], axis=1,
inplace=True, errors='ignore')

            predictions = model.predict(input_df)
            predictions_proba = model.predict_proba(input_df)
            predicted_labels = label_encoder.inverse_transform(predictions)
            prediction_time = time.time() - prediction_start_time

            # Display results
            st.subheader("Prediction Results")
            for i, (label, proba) in enumerate(zip(predicted_labels,
predictions_proba)):
                st.write(f"Record {i+1}: Predicted Outcome: **{label}**")
                st.write("Prediction Probabilities:")
                for class_label, prob in zip(label_encoder.classes_, proba):
                    st.write(f"{class_label}: {prob:.2f}")

                # Probability bar chart for first record
                if i == 0:
                    fig, ax = plt.subplots(figsize=(8, 4))
                    ax.bar(label_encoder.classes_, proba, color='skyblue')
                    ax.set_title(f"Prediction Probabilities for Record {i+1}")
                    ax.set_xlabel("Outcome")
                    ax.set_ylabel("Probability")
                    plt.tight_layout()
                    st.image(fig_to_streamlit(fig), caption=f"Prediction
Probabilities for Record {i+1}")
                    plt.close(fig)

            # Downloadable predictions
            results = pd.DataFrame({
                'Predicted_Label': predicted_labels,
                **{f'Prob_{label}': predictions_proba[:, i] for i, label in
enumerate(label_encoder.classes_)}
            })
            st.download_button("Download Predictions",
results.to_csv(index=False), "predictions.csv")

            # SHAP explanation for first record
            st.subheader("Model Explanation (SHAP) - First Record")
            shap_start_time = time.time()
            explainer = shap.TreeExplainer(model.estimators_[1])  # Use XGBoost
for SHAP
            shap_values = explainer.shap_values(input_df.iloc[[0]])
```

```
            plt.figure(figsize=(10, 6))
            shap.summary_plot(shap_values, input_df.iloc[[0]], plot_type="bar",
show=False)
            plt.title("SHAP Feature Importance for First Record")
            plt.tight_layout()
            plt.savefig('shap_plot.png')
            plt.close()
            st.image('shap_plot.png', caption="SHAP Feature Importance")

            shap_time = time.time() - shap_start_time

            # Display timing information
            st.write(f"Prediction Time: {prediction_time:.2f} seconds")
            st.write(f"SHAP Computation Time: {shap_time:.2f} seconds")
            st.write(f"Total Processing Time: {time.time() - start_time:.2f}
seconds")

    except Exception as e:
        st.error(f"Error processing file: {str(e)}")

# Input form (only shown if no file is uploaded)
if uploaded_file is None:
    with st.form("patient_form"):
        st.subheader("Patient Information")

        # Numerical inputs
        daily_patient_inflow = st.number_input("Daily Patient Inflow",
min_value=0, value=100)
        emergency_response_time = st.number_input("Emergency Response Time
(Minutes)", min_value=0.0, value=15.0)
        staff_count = st.number_input("Staff Count", min_value=0, value=50)
        bed_occupancy_rate = st.number_input("Bed Occupancy Rate (%)",
min_value=0.0, max_value=100.0, value=75.0)
        visit_frequency = st.number_input("Visit Frequency", min_value=0,
value=3)
        wait_time_minutes = st.number_input("Wait Time (Minutes)", min_value=0.0,
value=30.0)
        length_of_stay = st.number_input("Length of Stay (Days)", min_value=0.0,
value=5.0)
        previous_visits = st.number_input("Previous Visits", min_value=0,
value=2)
        resource_utilization = st.number_input("Resource Utilization",
min_value=0.0, value=80.0)
        age = st.number_input("Age", min_value=0, value=45)
```

```
        # Derived features
        resource_bed_interaction = resource_utilization * (bed_occupancy_rate /
100)
        wait_length_interaction = wait_time_minutes * length_of_stay
        log_wait_time = np.log1p(wait_time_minutes)
        log_resource_utilization = np.log1p(resource_utilization)

        # Categorical inputs
        treatment_outcome = st.selectbox("Treatment Outcome", ["Improved",
"Unchanged", "Worsened"])
        equipment_availability = st.selectbox("Equipment Availability",
["Available", "Limited", "Unavailable"])
        medicine_stock_level = st.selectbox("Medicine Stock Level", ["Adequate",
"Low"])
        comorbidities = st.selectbox("Comorbidities", ["Yes", "No"])
        disease_category = st.selectbox("Disease Category", ["Other",
"Cardiovascular", "Diabetes", "Respiratory"])

        # Ordinal input
        satisfaction_rating = st.slider("Satisfaction Rating", min_value=1,
max_value=5, value=3)
        satisfaction_staff_interaction = satisfaction_rating * staff_count

        # Submit button
        submitted = st.form_submit_button("Predict Outcome")

    # Process manual input and predict
    if submitted:
        start_time = time.time()
        # Create input dataframe
        input_data = pd.DataFrame({
            'Daily_Patient_Inflow': [daily_patient_inflow],
            'Emergency_Response_Time_Minutes': [emergency_response_time],
            'Staff_Count': [staff_count],
            'Bed_Occupancy_Rate': [bed_occupancy_rate / 100],
            'Visit_Frequency': [visit_frequency],
            'Wait_Time_Minutes': [wait_time_minutes],
            'Length_of_Stay': [length_of_stay],
            'Previous_Visits': [previous_visits],
            'Resource_Utilization': [resource_utilization],
            'Age': [age],
            'Resource_Bed_Interaction': [resource_bed_interaction],
            'Wait_Length_Interaction': [wait_length_interaction],
            'Log_Wait_Time': [log_wait_time],
            'Log_Resource_Utilization': [log_resource_utilization],
```

```
            'Treatment_Outcome': [treatment_outcome],
            'Equipment_Availability': [equipment_availability],
            'Medicine_Stock_Level': [medicine_stock_level],
            'Comorbidities': [comorbidities],
            'Disease_Category': [disease_category],
            'Satisfaction_Rating': [satisfaction_rating],
            'Satisfaction_Staff_Interaction': [satisfaction_staff_interaction]
        })

        # Preprocess and predict
        prediction_start_time = time.time()
        input_processed = preprocessor.transform(input_data)
        input_df = pd.DataFrame(input_processed, columns=feature_names)
        input_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'], axis=1,
inplace=True, errors='ignore')

        prediction = model.predict(input_df)
        prediction_proba = model.predict_proba(input_df)[0]
        predicted_label = label_encoder.inverse_transform(prediction)[0]
        prediction_time = time.time() - prediction_start_time

        # Save to prediction history
        history_entry = {
            'Predicted_Label': predicted_label,
            'Prob_Improved': prediction_proba[0],
            'Prob_Unchanged': prediction_proba[1],
            'Prob_Worsened': prediction_proba[2],
            'Wait_Time_Minutes': wait_time_minutes,
            'Staff_Count': staff_count,
            'Timestamp': time.strftime("%Y-%m-%d %H:%M:%S")
        }
        st.session_state.prediction_history.append(history_entry)

        # Display prediction
        st.subheader("Prediction Result")
        st.write(f"Predicted Outcome: **{predicted_label}**")
        st.write("Prediction Probabilities:")
        for label, prob in zip(label_encoder.classes_, prediction_proba):
            st.write(f"{label}: {prob:.2f}")

        # Model Confidence Indicator
        confidence = max(prediction_proba) * 100
        st.write(f"Model Confidence: {confidence:.2f}%")

        # Probability bar chart
```

```
        fig, ax = plt.subplots(figsize=(8, 4))
        ax.bar(label_encoder.classes_, prediction_proba, color='skyblue')
        ax.set_title("Prediction Probabilities")
        ax.set_xlabel("Outcome")
        ax.set_ylabel("Probability")
        plt.tight_layout()
        st.image(fig_to_streamlit(fig), caption="Prediction Probabilities")
        plt.close(fig)

        # Downloadable prediction
        results = pd.DataFrame({
            'Predicted_Label': [predicted_label],
            **{f'Prob_{label}': [prob] for label, prob in
zip(label_encoder.classes_, prediction_proba)}
        })
        st.download_button("Download Prediction", results.to_csv(index=False),
"prediction.csv")

        # Visualize the input values that led to the prediction
        st.subheader("Input Values Visualization")

        # Numerical Features (Bar Chart)
        numerical_values = {
            'Daily Patient Inflow': daily_patient_inflow,
            'Emergency Response Time (Min)': emergency_response_time,
            'Staff Count': staff_count,
            'Bed Occupancy Rate (%)': bed_occupancy_rate,
            'Visit Frequency': visit_frequency,
            'Wait Time (Min)': wait_time_minutes,
            'Length of Stay (Days)': length_of_stay,
            'Previous Visits': previous_visits,
            'Resource Utilization': resource_utilization,
            'Age': age,
            'Satisfaction Rating': satisfaction_rating
        }

        fig, ax = plt.subplots(figsize=(10, 6))
        ax.barh(list(numerical_values.keys()), list(numerical_values.values()),
color='lightgreen')
        ax.set_title("Numerical Input Features")
        ax.set_xlabel("Value")
        ax.set_ylabel("Feature")
        plt.tight_layout()
        st.image(fig_to_streamlit(fig), caption="Numerical Features Used for
Prediction")
```

```
        plt.close(fig)

        # Categorical Features (Bar Chart for simplicity)
        categorical_values = {
            'Treatment Outcome': treatment_outcome,
            'Equipment Availability': equipment_availability,
            'Medicine Stock Level': medicine_stock_level,
            'Comorbidities': comorbidities,
            'Disease Category': disease_category
        }

        fig, ax = plt.subplots(figsize=(10, 6))
        ax.barh(list(categorical_values.keys()), [1] * len(categorical_values),
color='lightcoral')
        for i, (key, value) in enumerate(categorical_values.items()):
            ax.text(0.5, i, value, ha='center', va='center', color='black',
fontweight='bold')
        ax.set_title("Categorical Input Features")
        ax.set_xlabel("Category (Value Displayed)")
        ax.set_ylabel("Feature")
        ax.set_xlim(0, 1.5)
        plt.tight_layout()
        st.image(fig_to_streamlit(fig), caption="Categorical Features Used for
Prediction")
        plt.close(fig)

        # Benchmarking Against Averages
        averages = {
            'Daily Patient Inflow': 80,
            'Emergency Response Time (Min)': 12.0,
            'Staff Count': 60,
            'Bed Occupancy Rate (%)': 70.0,
            'Wait Time (Min)': 20.0
        }

        st.subheader("Benchmarking Against Averages")
        comparison = {
            'Feature': [],
            'Your Value': [],
            'Average': [],
            'Difference': []
        }

        for feature, avg in averages.items():
            user_value = numerical_values[feature]
```

```
            comparison['Feature'].append(feature)
            comparison['Your Value'].append(user_value)
            comparison['Average'].append(avg)
            comparison['Difference'].append(user_value - avg)

        comparison_df = pd.DataFrame(comparison)
        st.table(comparison_df)

        # SHAP explanation
        st.subheader("Model Explanation (SHAP)")
        shap_start_time = time.time()
        explainer = shap.TreeExplainer(model.estimators_[1])
        shap_values = explainer.shap_values(input_df)

        plt.figure(figsize=(10, 6))
        shap.summary_plot(shap_values, input_df, plot_type="bar", show=False)
        plt.title("SHAP Feature Importance for Prediction")
        plt.tight_layout()
        plt.savefig('shap_plot.png')
        plt.close()
        st.image('shap_plot.png', caption="SHAP Feature Importance")

        shap_time = time.time() - shap_start_time

        # Actionable Recommendations
        st.subheader("Actionable Recommendations")
        # Aggregate SHAP values across classes by taking the mean of absolute
values
        shap_values_aggregated = np.abs(shap_values).mean(axis=2)  # Shape: (1,
24)
        shap_values_df = pd.DataFrame(shap_values_aggregated,
columns=input_df.columns)
        top_negative_features =
shap_values_df.iloc[0].sort_values(ascending=False).head(3).index.tolist()

        st.write("Top features impacting the outcome (based on SHAP
importance):")
        for feature in top_negative_features:
            st.write(f"- {feature}: This feature has a high impact on the
prediction. Consider reviewing its value.")

        # Specific suggestion for Wait_Time_Minutes if it's a top feature
        if 'Wait_Time_Minutes' in top_negative_features:
            suggested_wait_time = wait_time_minutes * 0.8  # Reduce by 20%
```

```
            st.write(f"Suggestion: Reducing Wait Time to
{suggested_wait_time:.1f} minutes may improve the outcome.")

        # Interactive Feature Impact Simulator
        st.subheader("Feature Impact Simulator")
        with st.expander("Adjust Key Features"):
            new_wait_time = st.slider("Adjust Wait Time (Minutes)",
min_value=0.0, max_value=60.0, value=float(wait_time_minutes))
            new_staff_count = st.slider("Adjust Staff Count", min_value=0,
max_value=100, value=int(staff_count))

            if st.button("Simulate New Outcome"):
                input_data['Wait_Time_Minutes'] = [new_wait_time]
                input_data['Staff_Count'] = [new_staff_count]
                input_data['Wait_Length_Interaction'] = [new_wait_time *
length_of_stay]
                input_data['Log_Wait_Time'] = [np.log1p(new_wait_time)]
                input_data['Satisfaction_Staff_Interaction'] =
[satisfaction_rating * new_staff_count]

                input_processed = preprocessor.transform(input_data)
                input_df = pd.DataFrame(input_processed, columns=feature_names)
                input_df.drop(['Wait_Time_Minutes', 'Resource_Utilization'],
axis=1, inplace=True, errors='ignore')

                new_prediction = model.predict(input_df)
                new_prediction_proba = model.predict_proba(input_df)[0]
                new_predicted_label =
label_encoder.inverse_transform(new_prediction)[0]

                st.write(f"New Predicted Outcome: **{new_predicted_label}**")
                st.write("New Prediction Probabilities:")
                for label, prob in zip(label_encoder.classes_,
new_prediction_proba):
                    st.write(f"{label}: {prob:.2f}")

        # Historical Prediction Dashboard
        st.subheader("Prediction History")
        if st.session_state.prediction_history:
            history_df = pd.DataFrame(st.session_state.prediction_history)
            st.dataframe(history_df)

            fig, ax = plt.subplots(figsize=(10, 6))
            ax.plot(history_df['Timestamp'], history_df['Prob_Improved'],
label='Improved', marker='o')
```

```
            ax.plot(history_df['Timestamp'], history_df['Prob_Unchanged'],
label='Unchanged', marker='o')
            ax.plot(history_df['Timestamp'], history_df['Prob_Worsened'],
label='Worsened', marker='o')
            ax.set_title("Prediction Probabilities Over Time")
            ax.set_xlabel("Timestamp")
            ax.set_ylabel("Probability")
            ax.legend()
            plt.xticks(rotation=45)
            plt.tight_layout()
            st.image(fig_to_streamlit(fig), caption="Prediction Trends")
            plt.close(fig)

        # Real-Time Feedback
        st.subheader("Feedback")
        with st.form("feedback_form"):
            rating = st.slider("Rate the prediction accuracy (1-5)", 1, 5, 3)
            comments = st.text_area("Comments (optional)")
            feedback_submitted = st.form_submit_button("Submit Feedback")
            if feedback_submitted:
                st.success("Thank you for your feedback!")

        # Display timing information
        st.write(f"Prediction Time: {prediction_time:.2f} seconds")
        st.write(f"SHAP Computation Time: {shap_time:.2f} seconds")
        st.write(f"Total Processing Time: {time.time() - start_time:.2f}
seconds")
```

Appendix D: Re-run model(prevent data leakage)

```
import pandas as pd
import numpy as np
import shap
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
```

```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_curve,
auc, precision_recall_curve
from imblearn.combine import SMOTEENN
import joblib
import warnings
import time

warnings.filterwarnings('ignore')
np.random.seed(42)

# ──────────────────────────────────────────────────────────────
# Timing & logging
# ──────────────────────────────────────────────────────────────
overall_start_time = time.time()
print("=== Healthcare Outcome Prediction Pipeline (Leakage Corrected) ===")

# ──────────────────────────────────────────────────────────────
# 1. LOAD DATA
# ──────────────────────────────────────────────────────────────
print("\n[1] Loading dataset...")
data = pd.read_csv('Final_healthcare_big_data.csv')
print(f"Dataset shape: {data.shape}")

# Basic checks
print("\nTarget distribution:")
print(data['Outcome'].value_counts(normalize=True).round(3))

print("\nMissing values:\n", data.isnull().sum().sum(), "total")
data = data.drop_duplicates()
print(f"Shape after dropping duplicates: {data.shape}")

# ──────────────────────────────────────────────────────────────
# 2. DEFINE FEATURE GROUPS
# ──────────────────────────────────────────────────────────────
numerical_features = [
    'Daily_Patient_Inflow', 'Emergency_Response_Time_Minutes', 'Staff_Count',
    'Bed_Occupancy_Rate', 'Visit_Frequency', 'Wait_Time_Minutes',
    'Length_of_Stay', 'Previous_Visits', 'Resource_Utilization', 'Age'
]
```

```
categorical_features = [
    'Treatment_Outcome', 'Equipment_Availability', 'Medicine_Stock_Level',
    'Comorbidities', 'Disease_Category'
]

ordinal_features = ['Satisfaction_Rating']

target_col = 'Outcome'

# ──────────────────────────────────────────────────────────────
# 3. TRAIN-TEST SPLIT — MUST BE VERY EARLY
# ──────────────────────────────────────────────────────────────
print("\n[3] Performing stratified train-test split...")
X_raw = data.drop(columns=[target_col])
y_raw = data[target_col]

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y_raw)
print("Label encoding mapping:", dict(zip(label_encoder.classes_,
range(len(label_encoder.classes_)))))

X_train_raw, X_test_raw, y_train, y_test = train_test_split(
    X_raw, y_encoded,
    test_size=0.20,
    stratify=y_encoded,
    random_state=42
)

print(f"Train: {X_train_raw.shape[0]:,} rows   Test: {X_test_raw.shape[0]:,}
rows")
print("Train class distribution:\n",
pd.Series(y_train).value_counts(normalize=True).round(3))

# ──────────────────────────────────────────────────────────────
# 4. FEATURE ENGINEERING (applied separately to train & test)
# ──────────────────────────────────────────────────────────────
def create_features(df):
    df = df.copy()
    df['Resource_Bed_Interaction'] = df['Resource_Utilization'] *
df['Bed_Occupancy_Rate']
    df['Wait_Length_Interaction'] = df['Wait_Time_Minutes'] *
df['Length_of_Stay']
    df['Log_Wait_Time'] = np.log1p(df['Wait_Time_Minutes'])
    df['Log_Resource_Utilization'] = np.log1p(df['Resource_Utilization'])
```

```
    df['Satisfaction_Staff_Interaction'] = df['Satisfaction_Rating'] *
df['Staff_Count']
    return df

print("\n[4] Creating engineered features...")
X_train = create_features(X_train_raw)
X_test  = create_features(X_test_raw)

# Update numerical features list
numerical_features += [
    'Resource_Bed_Interaction', 'Wait_Length_Interaction',
    'Log_Wait_Time', 'Log_Resource_Utilization',
    'Satisfaction_Staff_Interaction'
]

# ──────────────────────────────────────────────────────────────
# 5. PREPROCESSING PIPELINE (fit only on train)
# ──────────────────────────────────────────────────────────────
print("\n[5] Building and fitting preprocessor (train only)...")

num_pipe = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

cat_pipe = Pipeline([
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(drop='first', sparse_output=False,
handle_unknown='ignore'))
])

ord_pipe = Pipeline([
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('scaler', StandardScaler())
])

preprocessor = ColumnTransformer([
    ('num', num_pipe, numerical_features),
    ('cat', cat_pipe, categorical_features),
    ('ord', ord_pipe, ordinal_features)
], remainder='drop')

# Fit & transform
X_train_pre = preprocessor.fit_transform(X_train)
X_test_pre  = preprocessor.transform(X_test)
```

```
# Get feature names
cat_ohe_names =
preprocessor.named_transformers_['cat'].named_steps['onehot'].get_feature_names_o
ut(categorical_features)
feature_names = numerical_features + list(cat_ohe_names) + ordinal_features

print(f"Features after preprocessing: {len(feature_names)}")
print("First 10 feature names:", feature_names[:10])

# For SHAP / visualization convenience
X_train_pre_df = pd.DataFrame(X_train_pre, columns=feature_names)
X_test_pre_df  = pd.DataFrame(X_test_pre,  columns=feature_names)

# ──────────────────────────────────────────────────────────────
# 6. RESAMPLING — ONLY TRAINING DATA
# ──────────────────────────────────────────────────────────────
print("\n[6] Applying SMOTEENN to training set only...")
smote_enn = SMOTEENN(random_state=42)
X_train_bal, y_train_bal = smote_enn.fit_resample(X_train_pre, y_train)

print("Post-resampling train class distribution:")
print(pd.Series(y_train_bal).value_counts(normalize=True).round(3))
print(f"Training set size after resampling: {X_train_bal.shape[0]:,} rows")

# ──────────────────────────────────────────────────────────────
# 7. MODEL DEFINITION & TRAINING
# ──────────────────────────────────────────────────────────────
print("\n[7] Defining and training Stacking Classifier...")

class_weights = {0: 5.0, 1: 1.0, 2: 1.0}

rf_model = RandomForestClassifier(
    n_estimators=500, max_depth=None, min_samples_split=2,
    min_samples_leaf=2, max_features='log2',
    class_weight=class_weights, random_state=42, n_jobs=2
)

xgb_model = XGBClassifier(
    eval_metric='mlogloss', random_state=42,
    n_estimators=1000, max_depth=10, learning_rate=0.1,
    subsample=0.9, colsample_bytree=1.0,
    reg_lambda=1, reg_alpha=1, n_jobs=2
)
```

```
lgbm_model = LGBMClassifier(
    class_weight=class_weights, random_state=42,
    n_estimators=1000, num_leaves=31, learning_rate=0.1,
    n_jobs=2, verbose=-1
)

cat_model = CatBoostClassifier(
    verbose=0, early_stopping_rounds=10,
    auto_class_weights='Balanced', random_seed=42,
    thread_count=2
)

stacking = StackingClassifier(
    estimators=[
        ('rf', rf_model),
        ('xgb', xgb_model),
        ('lgbm', lgbm_model),
        ('cat', cat_model)
    ],
    final_estimator=XGBClassifier(
        eval_metric='mlogloss', random_state=42,
        n_estimators=200, max_depth=5, learning_rate=0.1
    ),
    cv=5,
    n_jobs=1,
    verbose=2
)

train_start = time.time()
stacking.fit(X_train_bal, y_train_bal)
train_time = time.time() - train_start

print(f"\nStacking classifier training completed in {train_time:.1f} seconds")

# ──────────────────────────────────────────────────────────────
# 8. EVALUATION
# ──────────────────────────────────────────────────────────────
print("\n[8] Evaluating on clean test set...")
y_pred = stacking.predict(X_test_pre)
y_pred_proba = stacking.predict_proba(X_test_pre)

print("\nClassification Report (default threshold):")
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))

# ──────────────────────────────────────────────────────────────
```

```
# 9. THRESHOLD OPTIMIZATION (per class – F1 maximization)
# ──────────────────────────────────────────────────────────────
print("\n[9] Optimizing per-class decision thresholds...")
thresholds = {}
y_pred_adj = np.zeros_like(y_pred)

for cls in range(len(label_encoder.classes_)):
    precision, recall, thresh = precision_recall_curve(y_test == cls,
y_pred_proba[:, cls])
    f1_scores = 2 * (precision * recall) / (precision + recall + 1e-10)
    best_idx = np.argmax(f1_scores)
    thresholds[cls] = thresh[best_idx] if best_idx < len(thresh) else 0.5
    print(f"Class {label_encoder.classes_[cls]} → best threshold =
{thresholds[cls]:.4f}")

# Apply adjusted thresholds
for i in range(len(y_pred_proba)):
    scores = y_pred_proba[i]
    best_class = None
    best_score_above_thresh = -1
    for cls in range(len(scores)):
        if scores[cls] >= thresholds[cls] and scores[cls] >
best_score_above_thresh:
            best_score_above_thresh = scores[cls]
            best_class = cls
    y_pred_adj[i] = best_class if best_class is not None else np.argmax(scores)

print("\nClassification Report (adjusted thresholds):")
print(classification_report(y_test, y_pred_adj,
target_names=label_encoder.classes_))

# ──────────────────────────────────────────────────────────────
# 10. VISUALIZATIONS
# ──────────────────────────────────────────────────────────────
print("\n[10] Generating visualizations...")

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred_adj)
plt.figure(figsize=(7, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
            yticklabels=label_encoder.classes_)
plt.title('Confusion Matrix – Threshold Adjusted')
plt.xlabel('Predicted')
plt.ylabel('True')
```

```
plt.tight_layout()
plt.savefig('confusion_matrix_corrected.png', dpi=150)
plt.close()

# ROC Curves
plt.figure(figsize=(8, 6))
for i, cls_name in enumerate(label_encoder.classes_):
    fpr, tpr, _ = roc_curve(y_test == i, y_pred_proba[:, i])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f'{cls_name} (AUC = {roc_auc:.3f})')
plt.plot([0,1], [0,1], 'k--', lw=1)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Multi-class ROC Curves')
plt.legend(loc='lower right')
plt.grid(alpha=0.3)
plt.tight_layout()
plt.savefig('roc_curves_corrected.png', dpi=150)
plt.close()

# ──────────────────────────────────────────────────────────────
# 11. SHAP (using XGBoost as example)
# ──────────────────────────────────────────────────────────────
print("\n[11] Computing SHAP values (XGBoost)...")
xgb_model.fit(X_train_bal, y_train_bal)  # retrain single model for clean SHAP
explainer = shap.TreeExplainer(xgb_model)
shap_values = explainer.shap_values(X_test_pre_df)

shap.summary_plot(shap_values, X_test_pre_df, plot_type="bar", show=False,
                  max_display=15)
plt.title("SHAP Feature Importance (XGBoost)")
plt.tight_layout()
plt.savefig('shap_importance_corrected.png', dpi=150)
plt.close()

shap.summary_plot(shap_values, X_test_pre_df, show=False)
plt.title("SHAP Summary Plot (XGBoost)")
plt.tight_layout()
plt.savefig('shap_summary_corrected.png', dpi=150)
plt.close()

# ──────────────────────────────────────────────────────────────
# 12. SAVE ARTIFACTS
# ──────────────────────────────────────────────────────────────
print("\n[12] Saving models and artifacts...")
```

```
joblib.dump(stacking, 'healthcare_stacking_model_corrected.pkl')
joblib.dump(preprocessor, 'preprocessor_corrected.pkl')
joblib.dump(label_encoder, 'label_encoder_corrected.pkl')

training_times = pd.DataFrame({
    'Model': ['Stacking Classifier'],
    'Training Time (s)': [round(train_time, 1)]
})
training_times.to_csv('training_times_corrected.csv', index=False)

# ──────────────────────────────────────────────────────────────
# Final timing
# ──────────────────────────────────────────────────────────────
overall_time = time.time() - overall_start_time
print(f"\nPipeline completed in {overall_time/60:.1f} minutes ({overall_time:.0f}
seconds)")
print("All outputs saved. You can now use the corrected model.")
```

Appendix E: Dataset

Final_healthcare_big_data.csv