***Dissertation Title***

*Unveiling Deception in Healthcare: Machine Learning Approaches for Proactive Fraud Detection and Prevention in Medical Claims and Records*

**Final Thesis**

In Partial Fulfillment

of the Requirements for the Degree of

Master in Computer Science

|  |  |  |
| --- | --- | --- |
| Student Name | : |  |
| Student ID | : |  |
| Supervisor | : |  |

# Abstract

Healthcare fraud is a pervasive issue that costs billions annually, undermining the integrity of healthcare systems and burdening payers, providers, and patients. This dissertation, titled *Unveiling Deception in Healthcare: Machine Learning Approaches for Proactive Fraud Detection and Prevention in Medical Claims and Records*, explores the application of advanced machine learning techniques to address this critical problem.

The study focuses on developing and evaluating two machine learning models: Logistic Regression and Isolation Forest, to detect fraudulent activities within medical claims and records. The Logistic Regression model, a supervised learning approach, was employed to classify claims as either fraudulent or non-fraudulent based on labelled training data. Conversely, the Isolation Forest model, an unsupervised learning method, was utilized to identify anomalies within the dataset, which are indicative of potential fraud.

The research was conducted using a healthcare provider dataset, which underwent rigorous pre-processing, including feature scaling and data normalization, to ensure the models' accuracy and reliability. The performance of the models was assessed using key metrics such as accuracy, precision, recall, F1 score, and ROC AUC score. The Logistic Regression model achieved an accuracy of 78%, indicating a strong ability to differentiate between legitimate and fraudulent claims. The Isolation Forest model, while unsupervised, demonstrated a recall rate of 70%, making it a valuable tool for identifying suspicious activities that may not have been labelled in the dataset.

The findings of this study highlight the potential of machine learning to enhance fraud detection in healthcare, offering a proactive approach that could significantly reduce the incidence of fraud. The research also underscores the importance of integrating multiple models to provide a comprehensive solution, addressing both known and unknown fraudulent patterns.

The dissertation concludes with recommendations for future research, including the exploration of more advanced machine learning techniques, the incorporation of additional data sources, and the deployment of real-time fraud detection systems. This work contributes to the ongoing efforts to safeguard healthcare resources and ensure the delivery of quality care.

# Acknowledgements

Use font: Times New Roman; font size: 12; line spacing: 1.5

# Contents

[Abstract ii](#_Toc174808170)

[Acknowledgements iii](#_Toc174808171)

[Contents iv](#_Toc174808172)

[List of Tables vi](#_Toc174808173)

[List of Figures vii](#_Toc174808174)

[List of Acronyms viii](#_Toc174808175)

[1 Introduction 1](#_Toc174808176)

[1.1 Background Of The Study 2](#_Toc174808177)

[1.2 Problem Statement 3](#_Toc174808178)

[1.3 Research Significance 3](#_Toc174808179)

[1.4 Aims And Objectives 4](#_Toc174808180)

[1.5 Research Questions 4](#_Toc174808181)

[1.6 Research Methodology 4](#_Toc174808182)

[1.7 Structure Of The Dissertation 5](#_Toc174808183)

[1.8 Conclusion 6](#_Toc174808184)

[2 Literature Review/Related Work 7](#_Toc174808185)

[2.1 Comprehensive Overview of the Existing Literature 7](#_Toc174808186)

[2.1.1 Traditional Fraud Detection Methods 9](#_Toc174808187)

[2.2 Critical Analysis of Existing Studies [Gaps In Existing Literature] 10](#_Toc174808188)

[3 Methodology 12](#_Toc174808189)

[3.1 Data Collection and Preprocessing 12](#_Toc174808190)

[3.1.1 Data Collection 12](#_Toc174808191)

[3.1.2 Data Pre-processing 14](#_Toc174808192)

[3.2 ML/AI Model Development 15](#_Toc174808193)

[3.2.1 Predictive Models 15](#_Toc174808194)

[3.3 Real-Time Monitoring Implementation 18](#_Toc174808195)

[3.3.1 System Architecture 18](#_Toc174808196)

[3.3.2 Validation 18](#_Toc174808197)

[3.4 Compliance and Explainability 18](#_Toc174808198)

[3.4.1 Legal and Ethical Standards 18](#_Toc174808199)

[3.4.2 Explainable AI 18](#_Toc174808200)

[3.5 Evaluation of the Proposed System [Model Evaluation and Testing] 18](#_Toc174808201)

[3.5.1 Benchmarks 18](#_Toc174808202)

[3.5.2 Testing Methodology 19](#_Toc174808203)

[4 Experimental Results 20](#_Toc174808204)

[4.1 Experimental Setup 20](#_Toc174808205)

[4.2 Dataset Description 20](#_Toc174808206)

[4.3 Results 21](#_Toc174808207)

[4.3.1 Logistic Regression Results 21](#_Toc174808208)

[4.3.2 Isolation Forest Results 21](#_Toc174808209)

[4.3.3 Discussion of Results 22](#_Toc174808210)

[4.4 Comparison with Baseline Methods 22](#_Toc174808211)

[5 Conclusion and Future Work 23](#_Toc174808212)

[5.1 Conclusion 23](#_Toc174808213)

[5.2 Future Work 23](#_Toc174808214)

[6 References 25](#_Toc174808215)

[Appendix A. Model 27](#_Toc174808216)

[Appendix B. Title of Appendix. 29](#_Toc174808217)

# List of Tables

[Table 1: Dataset Key Features 21](#_Toc175908596)

[Table 2: Columns Of Dataset 23](#_Toc175908597)

[Table 3: Columns From Merge Tables Ready For Machine Learning 24](#_Toc175908598)

[Table 4: Confusion Matrix For Isolation Forest 37](#_Toc175908599)

# List of Figures

[Figure 1: AI In Fraud Detection 1](#_Toc175905182)

[Figure 2: The Health Insurance Payment Model 5](#_Toc175905183)

[Figure 3: Gender Distribution Among Beneficiaries 25](#_Toc175905184)

[Figure 4: Distribution of Birth Years of Beneficiaries 26](#_Toc175905185)

[Figure 5: Distribution between alive and deceased beneficiaries 26](#_Toc175905186)

[Figure 6: Chronic Disease Index 27](#_Toc175905187)

[Figure 7: Inpatient's Time in Hospital 27](#_Toc175905188)

[Figure 8: Diagnosis Index in Inpatient Train Data 28](#_Toc175905189)

[Figure 9: Diagnosis Index in Outpatient Data 29](#_Toc175905190)

[Figure 10: Distribution Between Potential Fraud and Normal Provider 30](#_Toc175905191)

[Figure 11: First Model ROC Curve 31](#_Toc175905192)

[Figure 12: Second Model ROC Curve 32](#_Toc175905193)

[Figure 13: Third Model ROC Curve 33](#_Toc175905194)

[Figure 14: Fourth Model ROC Curve 34](#_Toc175905195)

[Figure 15: Fifth Model ROC Curve 35](#_Toc175905196)

# List of Acronyms

Term Initial components of the term (examples are below)

FEC Forward Error Correction

FET Field Effect Transistor

Please insert one term per table row; this will ensure appropriate spacing and alignment between each term and its components. It will also allow you to sort the terms alphabetically.

# 1 Introduction

Fraud poses a significant challenge to the healthcare sector by diverting funds that should be allocated to patient care and essential services into the hands of fraudsters, often supporting lavish and unethical expenditures. These fraudulent activities not only drain resources from critical areas such as accurate medical diagnosis, treatment, and patient support but also contribute to escalating costs throughout the sector. As a result, there is mounting pressure on healthcare organizations to develop and implement robust systems designed to detect and prevent improper payments, ensuring that financial resources are used appropriately to enhance the quality of care and maintain the sector's integrity.

Existing techniques for preventing fraud in healthcare are largely reactive and often ineffective. Traditional methods, such as audits and retrospective reviews, are designed to respond to fraudulent activities only after they have occurred, rather than proactively preventing them. This reactive approach results in significant resources being expended on addressing fraud after it has already caused financial damage, rather than on strategies that could prevent such activities from happening in the first place. Consequently, healthcare organizations face increased inefficiency and additional expenses, as substantial portions of their budgets are diverted to remediation instead of proactive fraud prevention.

As healthcare organizations grapple with the increasing prevalence of fraud, there is a growing recognition of the need for proactive measures to address these challenges more effectively. Traditional methods have proven insufficient in combating fraud, which has led to an urgent demand for advanced solutions. This chapter explores the application of artificial intelligence (AI) in the healthcare industry, with a specific focus on its role in fraud detection and prevention. AI offers a transformative approach by leveraging sophisticated techniques such as data mining, machine learning, and pattern classification to identify potential signs of fraud much earlier than conventional methods allow.

Unlike human operators or conventional systems, AI-powered solutions can analyze vast volumes of complex data at unprecedented speeds, offering a level of precision that manual processes simply cannot match. These systems are designed to recognize subtle patterns and anomalies in real-time, enabling the detection of suspicious activities almost instantaneously. For example, AI algorithms can flag irregular billing practices, identify discrepancies in patient records, and detect abnormal patterns in claims submissions that may indicate fraudulent behavior.

By continuously learning and adapting to new types of fraud, AI systems become increasingly effective over time, minimizing the chances of fraud going undetected. This real-time detection capability ensures that fraudulent payments can be intercepted before they are processed, thereby preventing financial losses and maintaining the integrity of healthcare services. The integration of AI into healthcare fraud detection represents a significant shift toward a more proactive, efficient, and effective approach, ultimately safeguarding resources that are critical for providing quality care and improving patient outcomes.

Figure 1: AI In Fraud Detection

The introduction of artificial intelligence (AI) in fraud detection marks a transformative shift toward more dynamic and preventive strategies in the healthcare industry. Unlike traditional methods that are reactive and primarily focus on identifying fraud after it has occurred, AI-driven solutions emphasize pre-emptive measures, significantly reducing the likelihood of fraudulent activities from the outset. By deploying advanced algorithms and machine learning techniques, AI can rapidly analyze vast datasets to detect anomalies, patterns, and behaviours that are indicative of potential fraud. This allows for real-time monitoring and swift action, which not only curtails the frequency of fraud but also mitigates its financial impact.

Moreover, the use of AI in fraud detection optimizes the allocation of resources by reducing the need for labour-intensive audits and manual reviews. With AI taking on the heavy lifting of data analysis, healthcare organizations can redirect their workforce and financial resources towards core functions that directly enhance patient care and operational efficiency. The ability of AI to swiftly and accurately identify fraudulent activities ensures that funds are preserved for essential services, such as patient diagnostics, treatments, and innovations in care delivery.

In an increasingly complex healthcare landscape where costs are rising and the demand for high-quality care is ever-growing, the proactive use of AI technology to safeguard against fraud is becoming indispensable. It not only helps maintain financial integrity but also fosters trust among patients, providers, and payers. By ensuring that resources are utilized effectively and ethically, AI contributes to the sustainability and resilience of healthcare organizations, allowing them to better manage costs while continuously adapting to the challenges of a rapidly evolving environment.

## Background of The Study

Healthcare fraud significantly exacerbates the overall costs associated with healthcare provision, particularly within large-scale programs like Medicaid. In the United States alone, billions of dollars are embezzled annually through various fraudulent schemes, placing an enormous financial burden on the healthcare system (Agarwal, 2023). Medicaid fraud includes practices such as billing for services that were never provided, misrepresenting the nature of services to receive higher reimbursements, or duplicating claims to maximize payouts. These actions siphon funds away from legitimate care, ultimately leading to increased costs for taxpayers and diminished resources for patient care.

Further highlighting the scale of this issue, a recent Hoffman survey, in collaboration with CIGNA HealthCare and other major insurance groups, estimated that between $80 billion and $100 billion is lost each year due to fraud and improper billing practices (Aslam, 2024). Common examples of such fraudulent activities include charging for services that were not actually delivered, inflating the costs of treatments, and bypassing established billing procedures. These activities not only compromise the financial stability of healthcare systems but also pose a significant threat to patient safety and trust.

Fraudulent practices undermine the integrity of healthcare institutions by diverting resources away from essential services, which can lead to compromised care quality, delayed treatments, and reduced access to necessary medical interventions. Additionally, they increase administrative overheads and compliance costs as organizations must invest heavily in detection and prevention measures. The cumulative impact of these fraudulent activities stretches beyond mere financial loss; it also threatens the overall effectiveness and sustainability of healthcare delivery, making it imperative to implement stronger fraud detection and prevention systems to safeguard both patients and healthcare providers.

Kickbacks represent another pervasive and unlawful activity within healthcare transactions, where it is illegal to offer, pay, solicit, or receive any form of remuneration in exchange for patient referrals or for prescribing specific treatments or services that are ultimately reimbursed by healthcare programs. These practices violate federal and state laws, including the Anti-Kickback Statute in the United States, which aims to protect patients and government programs from fraud, waste, and abuse by preventing financial incentives from influencing medical decisions. Kickback schemes can manifest in various forms, such as direct payments, gifts, or other financial incentives, all of which compromise the integrity of healthcare delivery.

This type of fraud often stems from unethical behavior, including self-serving corruption, fabricated treatments, and the provision of unnecessary services. In many cases, healthcare providers may intentionally prescribe unwarranted medical tests, procedures, or medications to patients, not based on their medical needs, but rather to inflate billing and receive kickbacks from third parties. Additionally, improper business arrangements, such as exclusive deals or secretive agreements between healthcare providers and suppliers, can lead to inflated costs and compromised patient care.

Within healthcare facilities, unethical practices can also involve employees who engage in embezzlement, billing for services never rendered, or coercing patients into unnecessary treatments or procedures to maximize profits. These fraudulent actions not only increase healthcare costs but also jeopardize patient safety, undermine trust in healthcare institutions, and lead to the misallocation of critical resources. The consequences of kickbacks extend beyond financial loss; they erode the ethical foundation of healthcare, distort clinical decision-making, and ultimately diminish the quality of care provided to patients. Addressing this issue requires robust regulatory oversight, enhanced detection mechanisms, and a commitment to ethical practices to ensure that healthcare remains patient-centered and free from corruption.

In the early stages of healthcare fraud detection, the focus was primarily on more overt and identifiable offenses, such as bribery and kickbacks, which were relatively straightforward to recognize and prosecute. These offenses involved direct exchanges of money or favors for patient referrals or specific services reimbursed by healthcare programs. However, recent efforts by the Office of the Inspector General (OIG) have broadened the scope of anti-fraud initiatives, targeting less obvious and more nuanced forms of fraud that exploit administrative loopholes and ambiguities. As courts expand the interpretation of anti-fraud statutes, the range of actions considered fraudulent has widened, thereby increasing the potential for criminal charges and penalties (Practice, 2022).

This expansion in the definitions of healthcare fraud means that even seemingly minor infractions or administrative oversights can now lead to significant legal consequences. For instance, healthcare providers have been prosecuted for failing to disclose critical information to insurance carriers, such as submitting a bill for payment while simultaneously waiving a patient’s co-payment, which is viewed as a deceptive practice intended to secure reimbursements under false pretences (Richard A. Bauder, The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels, 2018). Such actions, while perhaps less overt than traditional forms of bribery or kickbacks, still undermine the integrity of healthcare financial systems by manipulating the rules and deceiving insurers.

The increased scrutiny of these obscure cases reflects a broader regulatory trend toward zero tolerance for all forms of healthcare fraud, regardless of their subtlety or complexity. It also underscores the importance for healthcare providers to maintain rigorous compliance with evolving regulations, as even minor deviations from administrative policies can now result in severe penalties. As the definitions of fraud continue to expand, the need for comprehensive monitoring, robust compliance programs, and advanced detection methods, such as machine learning and data analytics, becomes even more critical to protect against potential legal exposure and uphold the ethical standards of healthcare practice.

A survey conducted by the Health Insurance Association of America in 1993 revealed that certain areas within healthcare—specifically diagnosis (43%) and billing services (34%)—are most frequently associated with fraudulent activities. Fraud in these areas often involves misrepresenting the nature or severity of a patient’s condition to justify unnecessary treatments or inflating billing codes to receive higher reimbursements. These fraudulent practices drain significant financial resources from the healthcare system, leading to increased costs that ultimately burden both insurance companies and patients.

The early identification and prevention of fraud are crucial not only for insurance companies seeking to avoid excessive pay-outs but also for the broader goal of managing and containing the ever-escalating costs of healthcare. By intercepting fraudulent claims and billing practices at their inception, insurers can save billions of dollars annually, which can then be redirected towards genuine medical services and patient care. This proactive approach ensures that funds are allocated appropriately, enhancing the efficiency and sustainability of healthcare systems.

Moreover, the resources saved from preventing fraud could be used to diagnose and treat various diseases and illnesses that require urgent medical attention. Preventing healthcare fraud not only reduces financial losses but also helps maintain trust and integrity within the healthcare sector. It fosters an environment where resources are used ethically and effectively, improving access to quality care and ensuring that patients receive the treatments they genuinely need. In an era where healthcare costs are continually rising, the ability to detect and prevent fraud early on is becoming increasingly vital to the long-term viability of healthcare delivery and to preserving funds for essential medical services.

In recent years, the application of artificial intelligence (AI) in healthcare has evolved significantly, with machine learning techniques being increasingly integrated into fraud detection strategies. Machine learning offers powerful tools for identifying complex patterns and anomalies in large datasets, which traditional methods often fail to detect. Among these techniques, K-means clustering—a form of unsupervised machine learning—has been effectively employed to identify fraudulent activities in medical insurance claims. For instance, Agarwal (2023) utilized K-means clustering to analyze labelled data and uncover irregularities indicative of potential fraud. By grouping similar data points and highlighting outliers, this method helps to identify patterns that may suggest fraudulent behaviour, such as unusual billing practices or discrepancies in treatment records.

Additionally, research by (Richard A. Bauder, Medicare Fraud Detection Using Machine Learning Methods, 2018) has demonstrated the potential of a data-driven approach in enhancing the accuracy and efficiency of healthcare fraud detection. Bauder developed a methodology that leverages Medicare claims data for supervised machine learning training, which improves the system's ability to detect fraud by learning from both past fraudulent and legitimate transactions. This approach uses a variety of machine learning models to analyze patterns within the data, identifying characteristics associated with fraudulent claims while reducing false positives.

By incorporating advanced machine learning techniques, AI-based fraud detection systems in healthcare have revolutionized the ability to process vast amounts of data with speed and precision, effectively identifying fraudulent activities that might otherwise go undetected. These systems, powered by sophisticated algorithms such as K-means clustering and various supervised learning models, can uncover subtle patterns and anomalies within massive datasets, enabling healthcare organizations to detect fraudulent behavior early in its development. Unlike traditional methods, these AI systems are continuously learning and adapting to new types of fraud, refining their accuracy and effectiveness over time. This continuous improvement allows for more proactive and dynamic fraud detection, reducing the financial losses associated with undetected fraudulent claims and preserving resources for legitimate patient care.

As healthcare costs continue to escalate, the deployment of AI tools like K-means clustering and supervised learning models represents a significant advancement in combating fraud. These technologies help safeguard financial resources, ensuring that funds are properly allocated to enhance patient care, improve clinical outcomes, and support essential services. The strategic use of machine learning in fraud detection does not just stop at identifying deceptive practices; it also fosters a culture of accountability and transparency, critical to maintaining trust within the healthcare sector.

In addition to these advancements, new research by (Sanalkumar, 2022) introduces a novel system architecture that integrates AI with blockchain technology to further enhance the identification and prevention of fraudulent activities within healthcare. This innovative approach leverages the decentralized and immutable nature of blockchain systems to create an additional layer of security and transparency. Within this architecture, machine learning algorithms are employed to analyze extensive medical data originating from diverse sources, such as sensors, electronic health records, and financial transactions. By examining this data holistically, the system can detect inconsistencies and signs of fraudulent behaviour with greater accuracy and efficiency.

Sanalkumar’s methodology not only improves fraud detection but also optimizes the management of healthcare practices. By ensuring that transactions and data exchanges across healthcare systems are both secure and transparent, this architecture helps prevent the potential degradation of patient care standards and curtails the excessive costs associated with fraud. The integration of machine learning within a blockchain framework also allows for real-time monitoring and decision-making, reducing the time required to identify and respond to fraudulent activities. This rapid response capability is critical in a field where delays can have significant financial and clinical implications.

Furthermore, the system's ability to handle data from multiple sources enables it to adapt to the evolving nature of healthcare fraud, where new schemes and tactics are constantly emerging. As fraudsters develop more sophisticated methods to exploit vulnerabilities, AI and blockchain-based approaches provide a robust, adaptable defence mechanism that enhances the overall resilience of healthcare systems. The use of such technologies is essential not only for detecting fraud but also for ensuring that healthcare organizations remain focused on their primary mission: delivering high-quality, affordable care to patients.

By enhancing both detection and prevention capabilities, these AI-driven methodologies offer a comprehensive approach to managing fraud risks, protecting patient safety, and maintaining the financial stability of healthcare institutions. This multifaceted strategy ensures that healthcare resources are utilized effectively and ethically, reducing waste and maximizing the quality of care provided to patients in an increasingly complex and costly healthcare environment.

## Problem Statement

Traditional methods for detecting fraud and money laundering, which rely heavily on manual systems, rigid rules, and threshold values, have proven increasingly inadequate in the face of sophisticated, high-tech criminal activities. These outdated approaches, which often involve static procedures and predefined criteria, struggle to keep pace with the rapidly evolving tactics employed by modern fraudsters. The globalization of markets and the instantaneous nature of data transfers have further exacerbated these challenges, enabling criminals to devise and implement increasingly complex schemes that traditional systems are ill-equipped to detect.

As fraud and money laundering techniques become more advanced, the need for more dynamic and adaptive detection methods has become apparent. This is where artificial intelligence (AI), particularly through machine learning (ML) and deep learning (DL), offers a transformative solution. AI technologies can analyze vast amounts of data with exceptional speed and precision, uncovering previously unknown patterns and anomalies that manual systems might miss. By employing advanced algorithms, AI systems can detect subtle deviations in transactional behaviour and recognize emerging fraud opportunities with a high degree of accuracy.

For instance, in the context of credit card fraud detection, AI can scrutinize transaction data in real time, identifying irregularities that deviate from established patterns of legitimate behaviour. Machine learning models are trained on extensive datasets to recognize normal transaction patterns and flag anomalies that could indicate fraudulent activity. Deep learning, a subset of machine learning, further enhances this capability by learning from complex and multi-dimensional data, enabling even more precise detection of sophisticated fraud schemes.

The integration of AI into fraud detection not only improves accuracy but also enables proactive measures, allowing organizations to respond to potential threats before they escalate into significant issues. By leveraging the power of AI, financial institutions and other organizations can stay ahead of evolving fraud tactics, safeguarding their operations and protecting their customers from financial loss. This advanced approach represents a critical evolution in fraud detection, providing a robust defence against the increasingly sophisticated methods of modern criminals.

The primary potential victims of these illicit actions are healthcare organizations, which are critical to society as they provide critical medical care services to the population; such fine-tuning attacks lead to severe economic losses, the weakening of patient protection, and loss of community trust. However, the healthcare industry is still not exploring enough of the possibilities offered by such technologies as AI and ML in the field of fraud detection (Prosper Kandabongee Yeng, 2021). According to studies, cases of healthcare fraud occur to the tune of billions of dollars yearly, making it a resilient issue affecting healthcare organizations (Richard A. Bauder, The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels, 2018).

Modern analytical methods and existing technologies focused on the use of the data mining paradigm call for more efficient management of healthcare fraud, including the interprofessional approach. Experiences and cases of fraud show that statistical methods and data mining methods are essential tools that are used to enhance the knowledge of this risk in the industry. However, the traditional rule-based model does not adapt in line with the advanced transformation of improved fraud schemes, as noted by (Roy, 2022). As for this scenario, more attention should be paid to improving existing fraud identification models with the help of AI and ML to prevent fraud in the sphere of healthcare, not reacting to it but actively avoiding the situations and cases described above.

## Research Significance

This research contributes significantly to the development of a "healthcare fraud detection machine learning framework," designed to address the limitations inherent in existing fraud detection methods. Traditional systems often struggle with proactive fraud detection, as they are typically reactive and unable to keep pace with the rapidly evolving tactics of sophisticated fraudsters. These conventional approaches, which rely on static rules and predefined thresholds, are increasingly inadequate in the face of new and complex fraud schemes that continuously emerge due to advancements in technology and globalization.

The proposed machine learning framework is specifically engineered to overcome these limitations by incorporating cutting-edge techniques and methodologies from the latest advancements in machine learning. Unlike traditional methods, this framework emphasizes proactive detection capabilities, enabling it to anticipate and identify potentially fraudulent activities before they manifest into significant problems. By leveraging state-of-the-art machine learning models, including supervised and unsupervised learning, as well as advanced algorithms for anomaly detection and pattern recognition, the framework enhances the speed, accuracy, and efficiency of fraud detection within the healthcare industry.

The innovation brought by this framework lies in its ability to process vast amounts of data in real time, learning from both historical and current data to continuously adapt to new fraud patterns. This dynamic approach ensures that the system remains effective in detecting emerging fraud tactics, providing healthcare organizations with a robust tool to safeguard against financial losses and protect the integrity of patient care. The enhanced speed and precision of the framework facilitate quicker responses to potential fraud, allowing for timely interventions and reducing the impact of fraudulent activities. Ultimately, this research aims to set a new standard in fraud detection, leveraging the latest machine learning advancements to offer a more resilient and adaptive solution for the healthcare sector.

Adopting the proposed machine learning framework for healthcare fraud detection has the potential to deliver substantial benefits by significantly reducing financial losses attributable to fraudulent claims and improper billing practices. By leveraging advanced algorithms and real-time data analysis, this framework enables healthcare organizations to detect and address fraudulent activities proactively, thereby preventing substantial financial drain caused by deceptive practices. This proactive approach not only mitigates financial risks but also safeguards the integrity of healthcare resources, ensuring that funds are allocated appropriately to patient care rather than being siphoned off through fraudulent means.

One of the most notable advantages of implementing this framework is its potential to enhance the overall quality and trustworthiness of healthcare services. By effectively identifying and preventing fraud, the framework ensures that healthcare resources are used efficiently and that patient care remains a top priority. This leads to improved service delivery and increased confidence among patients and stakeholders, as they can trust that the healthcare system is both fair and effective in addressing fraudulent activities. As a result, patient needs are met more reliably, and the quality of care is maintained or even enhanced.

Furthermore, the successful implementation of this machine learning framework can serve as a benchmark for similar projects worldwide, establishing a new standard for combating healthcare fraud. Its innovative approach provides a reference point for other organizations and countries looking to improve their fraud detection mechanisms and strengthen the integrity of their healthcare systems. By setting a new best practice in the field, this framework not only contributes to the financial stability of individual healthcare organizations but also promotes broader confidence in the global healthcare system. Ultimately, the framework represents a significant advancement in the fight against healthcare fraud, offering a robust solution that can be adopted and adapted globally to improve healthcare delivery and protect valuable resources.

## Aims and Objectives

**Aim**

To develop and validate a comprehensive machine learning framework capable of proactively detecting and preventing fraud within healthcare systems.

**Objectives:**

1. To Develop and Integrate Predictive Models: Implement advanced predictive models using both supervised and unsupervised machine learning techniques to identify potentially fraudulent activities.
2. To Implement and Validate Real-Time Monitoring: Establish a system for real-time fraud detection and assess its effectiveness compared to traditional methods.
3. To Ensure Compliance and Enhance Explainability: Ensure the framework adheres to legal and ethical standards, including data privacy regulations, and incorporate explainable AI to make the model's decisions transparent and understandable.

## Research Questions

1. How can machine learning algorithms be applied to detect fraudulent activities in healthcare claims and records effectively?
2. What are the benefits of real-time fraud detection systems over traditional fraud detection methods in terms of accuracy, efficiency, and cost?
3. How can explainable AI be integrated into fraud detection systems to ensure compliance with ethical and legal standards?

## Research Methodology

The proposed research will utilize a comprehensive approach to data collection by integrating both primary and secondary data sources. Primary data will be gathered directly from the healthcare database, providing current and relevant information on transactions, billing practices, and other operational aspects. This data will be crucial for developing and training the machine learning models to detect fraud, as it reflects the latest patterns and practices within the healthcare system.

In addition to primary data, secondary data will be sourced from patient records databases. This secondary data plays a critical role in providing a detailed historical context that is essential for developing accurate and effective AI-driven fraud detection systems. By incorporating historical patient records, the research gains insights into past patterns of fraudulent activities, treatment histories, and billing anomalies. This background information is invaluable for understanding the broader context of fraud, enabling the AI models to recognize both established and emerging fraud patterns with greater precision.

The integration of secondary data enhances the framework's ability to differentiate between legitimate and fraudulent activities by offering a more nuanced understanding of typical patient behaviors and billing practices. This comprehensive dataset allows for a more robust training process, improving the accuracy and reliability of the AI models in detecting and preventing fraud. By leveraging both primary and secondary data, the research aims to create a more effective fraud detection system that not only identifies current fraudulent activities but also anticipates and mitigates potential future fraud risks. This approach ensures that the developed AI framework is both well-informed and adaptable, enhancing its effectiveness in safeguarding healthcare resources and maintaining the integrity of patient care.

This data will undergo several preprocessing steps before it is deployed to implement the following models. These steps are used to improve data quality and relevancy by cleaning, normalizing, and structuring the data, which is crucial for the success of the machine learning models. The incorporation of extensive preprocessing makes it possible to clean any data fed into the models to eliminate any vices that may lead to inconsistency and redundancies.

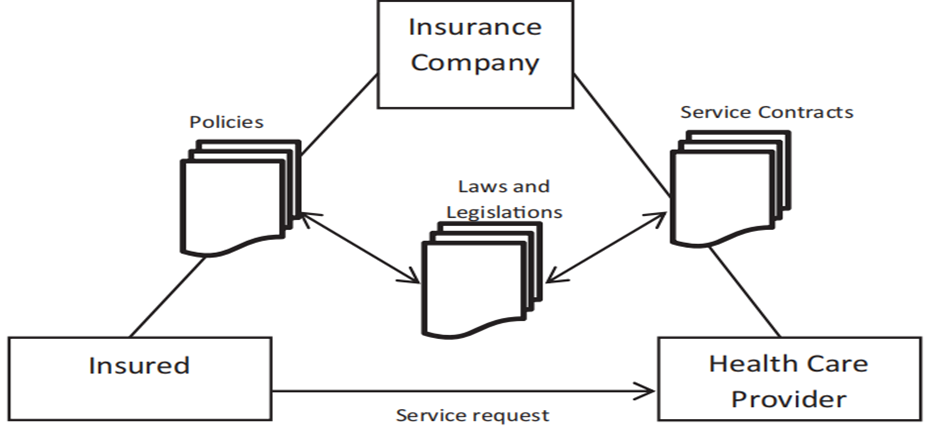


Figure 2: The Health Insurance Payment Model

Afterward, depending on this carefully selected dataset, several machine-learning models will be built and trained. To assess the effectiveness of these models, a set of benchmarks will be applied, and several tests will be conducted against the above characteristics. The overall evaluation of the models will reveal if they are suitable for detecting and forecasting fraudulent events in healthcare transactions. Specifically, the well-developed testing methodology will confirm the effectiveness of these models for practical usage in the actual environment.

## Structure Of The Dissertation

This dissertation is structured as follows:

* **Chapter 1: Introduction** – Outlines the background, problem statement, significance, objectives, research questions, and methodology of the study.
* **Chapter 2: Literature Review** – Reviews existing literature on healthcare fraud and previous applications of machine learning in fraud detection.
* **Chapter 3: Methodology** – Details the methods used for data collection, model development, and validation.
* **Chapter 4: Results and Discussion** – Presents the findings of the study and discusses their implications.
* **Chapter 5: Conclusion and Recommendations** – Summarizes the study, discusses limitations, and suggests areas for future research.

## Conclusion

The potential of machine learning (ML) in combating healthcare fraud is profoundly transformative, offering a proactive and dynamic approach to detecting and preventing fraudulent activities within the sector. Traditional methods of fraud detection, which often rely on static rules and manual reviews, are increasingly insufficient in addressing the sophisticated and evolving tactics of modern fraudsters. Machine learning, with its ability to analyze vast datasets, identify complex patterns, and adapt to emerging threats, represents a significant advancement in this field.

This research aims to harness the power of ML to enhance detection capabilities, ensuring not only the identification of fraudulent activities but also compliance with stringent regulatory standards. By leveraging advanced algorithms and data-driven insights, the proposed framework will provide a robust mechanism for proactively addressing fraud, thereby safeguarding the integrity and security of healthcare systems globally. The integration of ML techniques into fraud detection processes allows for real-time analysis and intervention, reducing the risk of financial losses and maintaining trust in healthcare services.

Moreover, this dissertation will underscore the effectiveness of ML methods in detecting and preventing fraud, demonstrating their value in enhancing the overall efficiency and accuracy of fraud detection systems. By providing empirical evidence and showcasing successful applications, the research will contribute to a broader understanding of how these technologies can be effectively employed in the healthcare industry. It will also pave the way for their widespread adoption, promoting a more secure and reliable healthcare environment.

In conclusion, the integration of machine learning into healthcare fraud detection holds immense potential for improving the security and integrity of healthcare systems worldwide. This research will not only highlight the transformative impact of ML techniques but also serve as a critical step toward their broader implementation, setting a new standard for combating fraud and ensuring the effective management of healthcare resources.

# Literature Review/Related Work

Healthcare frauds detrimentally affect the health and economic framework of the U. S. healthcare system. This research, hence, applies ML tools as a data scientist to uncover and counter HC fraud for nationally guarded healthcare assets. The objective is to apply ML to detect fraudulent events in claims in the healthcare sector by studying oddities in the collected data. According to the hypothesis, the unseen patterns employing the usual approach might be identified by employing mechanized learning, lowering the loss and safeguarding the healthcare structure. According to (Richard A. Bauder, The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels, 2018)and (Agarwal, 2023), it could be estimated that the fraud cost for the American healthcare system is tens of billions annually; this fraud involves anything from billing the patient when the service was never delivered to sophisticated kickback plans. Such unlawful activities pull vital funds away from federal programs, Medicare and Medicaid, skyrocket insurance costs, and escalate operating expenses for the companies (Irum Matloob, 2020). These activities are coordinated by federal entities, with the CMS taking charge of legal actions against the fraud and the FBI leading in the investigation and prosecution of the crime (Shamitha, 2022; Gill and Aghili, 2020; Iqbal, 2022). Implementing ML technologies presents optimistic improvements in detecting fraud solutions (Lekkala, 2023). They should be able to process large data units to detect fraud cases, which are normally unnoticeable even by auditors. With this purpose, the current study seeks to support current endeavours to prevent healthcare fraud, thus protecting consumers, taxpayers, and the healthcare system in the United States.

## Comprehensive Overview of the Existing Literature

Fraud detection is critically important in the realm of medical insurance, where the complexities and scale of fraudulent activities necessitate sophisticated and intricate detection tools. The stakes are high, as fraudulent claims can significantly impact financial resources, undermine trust in the system, and ultimately affect patient care. As such, the development and application of advanced detection technologies are essential for maintaining the integrity and efficiency of medical insurance processes.

Machine learning (ML) has emerged as a pivotal technology in enhancing fraud detection efforts within this sector. The ability of ML algorithms to analyze vast amounts of data, identify complex patterns, and adapt to evolving fraudulent tactics represents a significant advancement over traditional methods. ML techniques, such as supervised learning, unsupervised learning, and deep learning, offer powerful tools for detecting anomalies, predicting fraudulent behaviour, and improving the overall accuracy of fraud detection systems.

This literature review seeks to explore the current advancements and future trends in healthcare fraud detection, with a focus on the role of machine learning. By examining recent research, methodologies, and case studies, the review aims to highlight the state-of-the-art techniques that are being utilized to combat healthcare fraud. It will delve into the effectiveness of various ML models, the challenges associated with their implementation, and the innovations driving their development.

Furthermore, the review will address emerging trends and future directions in the field, considering how advancements in ML, such as the integration of artificial intelligence (AI) and blockchain technology, could further enhance fraud detection capabilities. By identifying current practices and exploring potential future developments, the review aims to provide a comprehensive understanding of how ML is shaping the landscape of healthcare fraud detection and to offer insights into how these technologies can be leveraged to address ongoing and emerging challenges in the industry.

In his paper on major fraud types of medical insurance claims, Agarwal (2023) elaborately discusses this issue. Thus, utilizing K-means clustering, an unsupervised ML approach, Agarwal shows promising results in detecting fraud cases without using labelled data. Adaptive approaches are needed to detect fraudulent claims and minimize the impact of the healthcare system’s financial loss (Agarwal, 2023).

Johnson and Khoshgoftaar (2023) propose a data-driven architecture that considers the effectiveness and robustness of the tests designed for healthcare fraud identification. Using the Medicare claims, they build large-scale labelled datasets for analysing supervised learning, further celebrating these with new Provider summary features and presenting an extended data labelling approach. Their conclusions call attention to proper work in data pre-processing and the benefit of a data-focused approach to the ML process regarding healthcare fraud classification (Johnson and Khoshgoftaar, 2023).

Mohammed (2023) suggests a new approach to system architecture implemented using ML to identify and mitigate fraud cases in blockchain systems. Using the Random Forest algorithm, this two-step strategy can filter out the wrong values, highlight such transactions, and prove higher accuracy, reaction time, and scaling (Mohammed, 2023).

(Duman, 2022) discusses the application of the XGBoost technique in identifying Medicare fraud using traditional and ML techniques. His work outlines that XGBoost offers the best performance in metrics like AUC, precision, recall, and F1-score out of the tested techniques. Specifically, Duman underlines that Medicare loses about fifteen billion US dollars per year to fraud, stressing the value of public datasets in enhancing the existing levels of transparency and fraud detection (Duman, 2022).

In another article, Gill and Aghili (2020) discuss the topic of health insurance fraud detection and underline the acute demand for wise fraud detection solutions. They assess the characteristics of an ideal health insurance fraud detection application; they argue that the best solution in fraud cases should address the management of the integration of unstructured data and have a dynamic business plan (Gill and Aghili, 2020).

In 2020, Lennart Dangers integrated unsupervised learning to identify fresh fraud patterns without any prior labelling of large volumes of medical encounters and the numerous steganographic and symbiotic strategies used by the fraudsters. It shows that an audit of structured flows is attainable in an analogous method for healthcare data and contributes a useful instrument for insurance corporations to extend their auditing features (Dangers, 2020).

Aruleba and Sun (2023) have investigated incorporating such ML classifiers as the Decision Trees and Random Forests to determine healthcare fraud. It also reveals the viability of these techniques by using ensemble classifiers and performance metrics and shows how ML is useful in fighting healthcare fraud (Aruleba and Sun, 2023).

Roy(2022) used AI in healthcare data privacy, where a Random Forest algorithm achieved 92% accuracy in identifying threats to healthcare data privacy. Among the key enablers of telemedicine, this research reinforces the centrality of AI in creating a secure system to support digital health solutions (Roy, 2022).

Lekkala (2023) delved into the change that ML models introduced in combating healthcare fraud. The advancements in the application of ensemble methods and the use of deep learning models as the methods that can enhance the overall efficiency of fraud detection are underlined; special focus is paid to such features that can help identify frauds accurately (Lekkala, 2023).

Regarding the role of ML in detecting healthcare fraud, the following options are further expanded by Akbar et al. (2020) and Ho et al. (2020). Akbar describes the accuracy improvement in the decision tree classifier by the Extreme Gradient Boosting method. At the same time, Ho raises points on the ethical and regulatory considerations for using AI in health insurance (Akbar et al., 2020). These kinds of research add to the knowledge of the possibilities and problems of employing sophisticated approaches to prevent and control healthcare fraud efficiently.

### Traditional Fraud Detection Methods

This paper seeks to establish the importance of fraud detection in protecting financial assets and ensuring the fidelity of health financial systems. In the past, a rule-based approach has been used, and although it has its effectiveness, it has also faced some drawbacks that require more flexible strategies and frameworks (Hassan, Aziz & Andriansyah, 2023).

Systems based on rules belong to the core of traditional fraudulent activity detection; they use rules and criteria designed to detect unusual patterns included in the initial data by professionals using historical data and typical fraud schemes. For instance, such systems may include alerts where the transaction magnitude exceeds set limits or originates from certain geographical areas. These systems are quite easy to install and make much sense from a compliance and audit perspective. It is easy to implement and immediately detect suspicious transactions, and it is cheaper than complex transactions. However, these systems also alert the transactions conducted according to varied factors, including amount, origin, or frequency (Kotagiri & Yada, 2024).

Nonetheless, they have the following demerits: To begin with, rule-based systems … are inherently fixed and ineffective in responding to dynamism in fraud trends without programmed changes – practices that are frequently tedious and slow in responding to the current trends in fraud. Due to their generalistic approach in which their ruleset is built to catch as much fraud as possible, this often leads to many false positives; this adds strain to the detection team and may even flag innocent transactions, which will be unpleasing for customers. These systems working based on the pattern and history makes them highly vulnerable to new or complex fraud schemes that the system has not seen before. The routine modifying and enhancing of these structures entails significant amounts of hand-work and usually costly professional advice.

Based on these challenges, a new notion crept into fraud detection, requiring new and enhanced solutions. The ever-changing fraud schemes, with special attention to the level of confrontation of the reported schemes, demand concepts and architectures with learning and prediction capabilities for fraud detection in real-time (Kotagiri, 2023). It is characterized by their ability to examine large datasets methodically and logically and then draw conclusions based on trends and patterns from those datasets without strict programming. They can learn from each transaction continuously and monitor the fraud by checking for variations rather than using set rules. Implementing highly evolved systems that can process and analyze data in real-time will help respond as soon as possible to fraud threats, thus reducing the time window for fraud and improving decision-making time (Kotagiri & Yada, 2024).

Combining rule-based systems with adaptive solutions offers a powerful and effective approach to fraud detection, leveraging the strengths of both methodologies to address the complexities of modern fraud challenges. Rule-based systems are designed to handle known threats through predefined rules and criteria, making them effective for detecting well-understood and previously identified fraud patterns. These systems rely on established guidelines and thresholds to flag suspicious activities, providing a reliable means of addressing common and recurring fraudulent behaviours

However, as fraud tactics continue to evolve and become more sophisticated, solely relying on rule-based systems can be limiting. Adaptive solutions, which utilize machine learning and other advanced techniques, offer a complementary approach by focusing on novel and emerging threats. These systems are designed to learn from new data, identify previously unknown patterns, and adapt to evolving fraud schemes. By continuously updating their models based on real-time information, adaptive systems can detect and respond to fraud that may not be captured by static rules alone.

The integration of rule-based and adaptive systems creates a robust fraud detection framework that not only addresses known threats but also adapts to new and developing fraud techniques. This hybrid approach enhances overall effectiveness by minimizing false positives—where legitimate transactions are incorrectly flagged as fraudulent—since adaptive systems refine their detection capabilities over time based on actual data patterns. Additionally, the need for frequent manual updates and maintenance of the fraud detection system is reduced, as the adaptive components continuously adjust to new information without requiring constant manual intervention.

As transaction volumes and complexities continue to increase, this combined approach ensures that fraud detection systems remain scalable and effective. The ability to handle a growing number of transactions and increasingly complex fraud schemes is critical for maintaining the integrity of financial and healthcare systems. By integrating rule-based and adaptive solutions, organizations can achieve a more comprehensive and dynamic fraud detection strategy, effectively countering both established and emerging threats while optimizing the accuracy and efficiency of their fraud detection efforts. In conclusion, it can be stated that despite the significant importance of rule-based solutions and methods as well as traditional approaches to fraud detection, their shortcomings indicate the necessity of implementing new, more complex, and sophisticated models of fraud detection. Implementing machine learning and real-time analysis will enhance the ability to identify and prevent fraudulent activities, hence a more secure and safer monetary sector. (Patel, 2023; Wang et al., 2020).

## Critical Analysis of Existing Studies [Gaps In Existing Literature]

For that matter, this project fills the following major gaps in the existing body of knowledge: This project applies a systematic approach that includes a broad range of machine learning and deep learning methods. It also focuses on model interpretability and the creation of procedures for real-time predictions. Here are the specific gaps addressed: Here are the specific gaps addressed:

* **Comparative Analysis Across Models**: It is also different from many pieces of research in which the comparison is usually conducted on a single or a few selected models at most. This approach offers vital information to elucidate the overall superiority and inferiority of these models precisely in the context of detecting healthcare fraud.
* **Integration of Model Explainability**: While there is a growing interest in model interpretability in the context of healthcare, the successful application of explanation techniques based on SHAP across multiple models is not very well described in the literature. This project covers this imbalance through the application of SHAP values on multiple machine learning models with the aim of increasing the interpretability of the fraud detection models.
* **Real-Time Detection and Continuous Learning**: Fraud detection is analyzed by prior literature in a non-developmental manner, by training models on available examples. On the other hand, this project proposes a real-time detection pipeline that incorporates the option of model retraining occasionally. This innovation encompasses the idea that fraud in the healthcare industry evolves constantly and that models, including this industry, must change with time and adjust to evolving patterns.

# Methodology

## Data Collection and Preprocessing

### Data Collection

**PRIMARY DATA**

Source: Healthcare databases, including insurance claims and transaction records. These sources contain real-time data on billing, claims, patient demographics, treatments, and financial transactions, which are essential for detecting fraudulent patterns.

Purpose: To gather real-time and recent data for training and validating machine learning models. This data is crucial for developing models that can detect current fraud schemes and adapt to new tactics used by fraudsters.

Methods:

* Secure Access: Ensure compliance with data protection regulations (e.g., GDPR, HIPAA) by implementing strict access controls, data encryption, and anonymization techniques. This involves:
* Data Access Agreements: Draft and sign agreements with data providers that outline data usage, security measures, and compliance with legal standards.
* Data Anonymization: Remove personally identifiable information (PII) to protect patient privacy.
* Data Encryption: Use encryption protocols to secure data during transfer and storage.
* APIs (Application Programming Interfaces): Develop or utilize existing APIs to facilitate the secure and efficient extraction of data from healthcare databases. This can include:
* Data Extraction API: Create a custom API that allows for querying and extracting relevant data fields needed for fraud detection.
* Real-Time Data Streaming: Implement real-time data streaming using platforms like Apache Kafka to continuously feed data into the machine learning pipeline.
* Data Extraction Protocols: Establish protocols to regularly update the dataset, ensuring the models have access to the latest information. This involves:
* Scheduled Data Pulls: Set up automated scripts to extract data at regular intervals (e.g., daily, weekly).
* Data Quality Checks: Implement procedures to validate and clean the extracted data before it enters the machine learning pipeline.

**SECONDARY DATA**

Source: Historical patient records, fraud reports, and external datasets such as publicly available healthcare fraud datasets. This data provides a broader context and helps in understanding the evolution of fraud schemes over time.

Purpose: To provide a comprehensive background and historical context for developing accurate predictive models. This helps in identifying long-term trends and patterns that might not be evident from real-time data alone.

Methods:

* Utilize Existing Databases and Repositories: Leverage publicly available datasets and repositories that contain historical data on healthcare fraud. Some potential sources include:

1. National Health Care Anti-Fraud Association (NHCAA): Provides resources and data on healthcare fraud cases.
2. Centers for Medicare & Medicaid Services (CMS): Offers datasets related to healthcare claims and fraud.
3. Public Datasets: Utilize datasets from platforms like Kaggle, which may host healthcare fraud-related data.

* Literature Review: Conduct an extensive review of academic and industry publications to identify reliable sources of historical data and gain insights into common fraud patterns. This involves:

1. Database Search: Use academic databases like PubMed, IEEE Xplore, and Google Scholar to find relevant studies and reports.
2. Citation Tracking: Follow citations from key papers to uncover additional valuable sources.

* Secure Necessary Permissions: Obtain the required permissions to access and use historical patient records and other sensitive data. This includes:

1. Ethical Approvals: Seek approval from Institutional Review Boards (IRBs) or ethics committees.
2. Data Use Agreements: Negotiate agreements with data owners that define the scope of data use, ensuring compliance with privacy and ethical guidelines.

By combining primary and secondary data, the research will benefit from a rich dataset that encompasses both real-time information and historical context, allowing for the development of robust and accurate machine learning models for healthcare fraud detection.

### Data Pre-processing

Effective data pre-processing is fundamental to the success of any machine learning project, especially within the intricate field of healthcare fraud detection, where data complexity and variability are significant challenges. Pre-processing is essential for transforming raw, unrefined data into a clean, structured, and normalized format that can significantly enhance the performance, accuracy, and reliability of predictive models. Given the high stakes involved in healthcare fraud detection—where the quality of data directly impacts the ability to identify fraudulent activities accurately—pre-processing becomes a critical step that cannot be overlooked.

The primary objective of data pre-processing is to prepare data for analysis by addressing issues such as missing values, inconsistencies, outliers, and irrelevant information that could potentially skew model predictions. This stage involves several critical steps, each playing a vital role in refining data quality:

1. **Data Cleaning:** The first step in pre-processing involves identifying and correcting inaccuracies or inconsistencies in the data. This includes handling missing values through methods such as imputation or deletion, removing duplicate entries, and rectifying any errors in data entry. In healthcare datasets, this step is particularly important due to the presence of extensive patient records, billing information, and clinical data, which may contain numerous discrepancies that could adversely affect model outcomes.
2. **Data Normalization:** Normalization is crucial for ensuring that data is formatted consistently, allowing for fair comparison across different data points. This process scales numerical data to a common range, reducing biases and enhancing the convergence speed of machine learning algorithms. In healthcare fraud detection, normalization helps in managing the wide-ranging scales of different data features, such as varying claim amounts, patient ages, or treatment durations, ensuring that the model treats all variables uniformly and effectively.
3. **Data Structuring:** Structuring involves organizing data into a consistent format that is suitable for analysis. This may include transforming categorical data into numerical formats, creating relevant feature vectors, and segmenting data into training, validation, and testing sets. Structuring data is crucial for the model to learn and generalize effectively from historical patterns. In healthcare fraud detection, proper structuring enables the machine learning model to accurately distinguish between legitimate and fraudulent claims, thereby improving detection rates.

By meticulously executing these pre-processing steps, raw healthcare data is converted into a refined form that is ready for advanced analysis. This structured, clean, and normalized dataset is crucial for maximizing the predictive power of machine learning models, reducing noise, and improving model robustness. In the context of healthcare fraud detection, effective data pre-processing ensures that the resulting AI systems are both accurate and reliable, enabling timely identification of fraudulent activities and safeguarding valuable healthcare resources.

**DATA CLEANING**

The first step in the data pre-processing pipeline is data cleaning, which aims to eliminate inconsistencies, errors, and irrelevant information from the dataset. This process is essential because raw data, especially from healthcare sources, often contains inaccuracies that can significantly impact the performance of machine learning models.

One of the primary techniques in data cleaning is missing value imputation. Missing data is a common issue in healthcare records, arising from various factors such as incomplete patient information or errors in data entry. To address this, statistical methods like mean, median or mode imputation can be employed, where missing values are replaced with the average or most frequent values from the dataset. Alternatively, more sophisticated machine learning techniques, such as k-nearest neighbors (KNN) or regression models, can predict and fill in missing values based on the relationships between other variables.

Another crucial aspect of data cleaning is outlier detection and removal. Outliers are data points that deviate significantly from the rest of the dataset and can distort statistical analyses and model performance. Methods such as the z-score, which measures the number of standard deviations a data point is from the mean, or the interquartile range (IQR), which identifies data points outside the expected range, are effective in identifying outliers. Additionally, domain knowledge can be leveraged to distinguish between true anomalies and data entry errors, ensuring that only genuine outliers are addressed.

**DATA NORMALIZATION**

Once the data is cleaned, the next step is normalization. The objective of data normalization is to ensure that all data attributes are on a common scale, which is crucial for improving the performance of machine learning models. This is particularly important in healthcare data, where variables can span vastly different ranges and units.

Two common techniques for data normalization are min-max scaling and standardization. Min-max scaling transforms data to a specific range, typically 0 to 1, by rescaling the values linearly. This method ensures that all features contribute equally to the model, preventing attributes with larger ranges from dominating the learning process. Standardization, on the other hand, transforms data to have a mean of 0 and a standard deviation of 1. This technique is particularly useful when the data follows a Gaussian distribution, as it maintains the properties of the distribution while making the data suitable for machine learning algorithms that assume normally distributed inputs.

**DATA STRUCTURING**

The final stage of data pre-processing is structuring the data into a format suitable for machine learning models. This involves organizing the data in a way that enhances the model's ability to learn from it effectively.

Feature engineering is a critical technique in this stage, involving the creation of new features from raw data that can improve model performance. For example, in healthcare fraud detection, combining features such as patient demographics, treatment histories, and billing information can create more informative variables that capture complex relationships and patterns indicative of fraudulent activity.

Data transformation is another essential technique, particularly for categorical data that needs to be converted into numerical values. Methods like one-hot encoding, which creates binary columns for each category, or label encoding, which assigns a unique integer to each category, are commonly used to transform categorical variables. This step ensures that the machine learning algorithms can interpret and learn from the data effectively.

In conclusion, thorough data pre-processing involving cleaning, normalization, and structuring is fundamental to the success of machine learning models in healthcare fraud detection. By meticulously addressing inconsistencies, scaling data appropriately, and organizing it into a structured format, we lay a solid foundation for developing robust and accurate predictive models that can effectively identify and prevent fraudulent activities in the healthcare sector.

## ML/AI Model Development

The development of predictive models is a critical phase in the machine learning pipeline for healthcare fraud detection, serving as the foundation for identifying and mitigating fraudulent activities effectively. This phase involves a series of interconnected steps that include selecting the most appropriate algorithms, training these algorithms on well-prepared datasets, and rigorously evaluating their performance using a range of relevant metrics. The choice and implementation of these models are crucial, as they directly influence the system's ability to detect complex patterns of fraud within vast and varied healthcare data.

Both supervised and unsupervised learning approaches play indispensable roles in this process, each bringing unique strengths to the detection of fraudulent activities:

1. **Supervised Learning:** In supervised learning, models are trained on labelled data where the outcomes (e.g., legitimate vs. fraudulent claims) are already known. This approach allows the model to learn from historical data patterns and make predictions based on these learned patterns. Algorithms such as logistic regression, decision trees, random forests, and gradient boosting machines are often employed for this purpose. Supervised models are highly effective in detecting known fraud types by leveraging historical cases to build a predictive framework. They are particularly useful in identifying specific fraud patterns, such as duplicate billing, over-utilization of services, or unbundling of codes. The strength of supervised learning lies in its ability to provide highly accurate predictions for clearly defined fraud scenarios, especially when large volumes of labelled data are available.
2. **Unsupervised Learning:** Unsupervised learning, on the other hand, does not rely on pre-labelled data. Instead, it focuses on detecting anomalies or patterns that do not conform to the majority of the data, making it well-suited for discovering new or previously unknown types of fraud. Techniques such as clustering, anomaly detection, and association rules are employed to identify suspicious activities without prior knowledge of what constitutes fraud. For example, K-means clustering can group similar transactions together, highlighting outliers that may indicate fraudulent behaviour. Unsupervised learning is essential in the ever-evolving landscape of healthcare fraud, where new schemes continually emerge, and the patterns of fraudulent activities may change over time.
3. **Algorithm Selection and Training:** Selecting the right algorithms is crucial and depends on the nature of the data and the specific fraud detection goals. During the training phase, these algorithms are exposed to the pre-processed data, enabling them to learn patterns, relationships, and anomalies. Techniques such as cross-validation are used to ensure that the models generalize well to new, unseen data. The training process involves iterative refinement, where models are continuously adjusted and tuned to optimize their predictive accuracy and minimize false positives and negatives. This step is critical in healthcare fraud detection, where the costs of both undetected fraud and incorrect flagging of legitimate claims can be substantial.
4. **Model Evaluation:** Evaluating model performance is a vital component of this phase, requiring the use of multiple metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC). These metrics provide a comprehensive understanding of how well the model is performing across different aspects of fraud detection, including its ability to correctly identify fraud (sensitivity) and minimize false alarms (specificity). A robust evaluation process ensures that the model is not only effective but also reliable and scalable, capable of being deployed in real-world settings where healthcare data can be vast, complex, and noisy.

By integrating both supervised and unsupervised learning approaches, healthcare fraud detection models can achieve a comprehensive coverage of fraudulent activities, capturing both known and novel fraud schemes. This dual approach enhances the robustness and flexibility of the detection system, ensuring it remains effective against the constantly evolving tactics employed by fraudsters. As such, the development of predictive models is a dynamic and iterative process, combining the strengths of different machine learning methodologies to create a powerful tool for safeguarding healthcare resources and maintaining trust in the system.

### Predictive Models

In the realm of machine learning, predictive models can be broadly categorized into supervised and unsupervised learning. Each approach serves distinct purposes and offers unique advantages in the context of fraud detection.

**SUPERVISED LEARNING**

Supervised learning involves training models on labelled data, where each training example is paired with an output label. This approach is highly effective in scenarios where historical data on fraudulent activities is available, allowing the model to learn patterns and correlations that distinguish fraudulent from legitimate transactions.

Several algorithms are commonly employed in supervised learning for fraud detection:

1. Logistic Regression: This algorithm is a statistical method for predicting binary outcomes. In the context of fraud detection, logistic regression can model the probability of a transaction being fraudulent based on various features. Its simplicity and interpretability make it a valuable tool, particularly for understanding the influence of different variables on the likelihood of fraud.
2. Decision Trees: Decision trees split the data into subsets based on feature values, creating a tree-like model of decisions. They are intuitive and easy to visualize, making them useful for identifying key factors that contribute to fraudulent activities. However, they can be prone to overfitting, especially with complex datasets.
3. Random Forest: This ensemble method combines multiple decision trees to improve predictive performance and reduce overfitting. Random forests are robust and can handle large datasets with high dimensionality, making them suitable for detecting diverse fraud patterns.
4. Gradient Boosting: Another ensemble technique, gradient boosting builds models sequentially, with each new model correcting the errors of the previous ones. This approach can achieve high accuracy but requires careful tuning to avoid overfitting.
5. Support Vector Machines (SVM): SVMs are powerful for classification tasks, particularly in high-dimensional spaces. They work by finding the optimal hyperplane that separates fraudulent and non-fraudulent transactions. SVMs can be computationally intensive but are effective for complex datasets.
6. Neural Networks: Inspired by the human brain, neural networks consist of layers of interconnected nodes (neurons) that can learn intricate patterns. Deep learning, a subset of neural networks, has shown remarkable success in fraud detection, especially when dealing with large and unstructured data.

To train these models, labelled datasets containing examples of both fraudulent and legitimate transactions are used. The training process involves feeding the data into the algorithms, allowing them to learn the underlying patterns. Evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC are employed to assess model performance. These metrics provide insights into the model's ability to correctly identify fraud (precision), its coverage of actual fraud cases (recall), and the balance between precision and recall (F1-score). The AUC-ROC curve offers a comprehensive view of the model's discriminative power across different threshold settings.

**UNSUPERVISED LEARNING**

In situations where, labelled data is scarce or unavailable, unsupervised learning becomes invaluable. Unsupervised learning algorithms identify anomalies or patterns in data without prior knowledge of what constitutes fraud. This approach is particularly useful for detecting novel or evolving fraud schemes that may not be captured by historical data.

Several unsupervised learning algorithms are effective for fraud detection:

1. K-Means Clustering: This algorithm partitions data into clusters based on feature similarities. Transactions that do not fit well into any cluster may be flagged as anomalies, potentially indicating fraud. K-means is straightforward and efficient but requires specifying the number of clusters in advance.
2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): DBSCAN identifies clusters based on the density of data points, making it suitable for detecting outliers in datasets with varying distributions. It can find clusters of arbitrary shapes and does not require predefining the number of clusters.
3. Autoencoders: These neural network models are designed for unsupervised learning. Autoencoders compress data into a lower-dimensional representation and then reconstruct it. Transactions with high reconstruction errors are considered anomalies, suggesting potential fraud.
4. Isolation Forest: This ensemble method is specifically designed for anomaly detection. It isolates observations by randomly selecting features and splitting data. Anomalies are isolated quickly, making this algorithm efficient and effective for fraud detection.

The training process for unsupervised learning involves feeding the unlabelled data into the algorithms and allowing them to uncover hidden patterns or anomalies. Evaluation metrics such as the silhouette score, Davies-Bouldin index, and anomaly detection rate are used to assess the quality of the clustering or anomaly detection. The silhouette score measures how similar an object is to its cluster compared to other clusters, while the Davies-Bouldin index evaluates the average similarity ratio of each cluster with its most similar cluster. The anomaly detection rate indicates the proportion of true anomalies correctly identified by the model.

In conclusion, the development of predictive models using both supervised and unsupervised learning techniques is crucial for effective healthcare fraud detection. Supervised learning leverages historical data to recognize known fraud patterns, while unsupervised learning uncovers novel and evolving fraud schemes. By employing a diverse set of algorithms and rigorous evaluation metrics, we can build robust models that enhance the detection and prevention of fraudulent activities in the healthcare sector.

## Real-Time Monitoring Implementation

### System Architecture

Components:

* Data Stream Processing: Use tools like Apache Kafka or Apache Flink to handle real-time data streams.
* Model Deployment: Use frameworks like TensorFlow Serving, Docker, or Kubernetes to deploy models for real-time inference.
* Alerting System: Integrate with monitoring tools to trigger alerts for detected anomalies.

### Validation

Methods:

* Backtesting: Test the model on historical data to simulate real-time performance.
* A/B Testing: Compare the real-time system against traditional methods to evaluate improvements in accuracy and efficiency.

## Compliance and Explainability

### Legal and Ethical Standards

* Compliance: Ensure the framework adheres to regulations like GDPR, HIPAA, and other relevant data privacy laws.
* Data Privacy: Implement data anonymization and encryption techniques to protect sensitive information.

### Explainable AI

Techniques:

* LIME (Local Interpretable Model-agnostic Explanations): Provide local explanations for individual predictions.
* SHAP (SHapley Additive exPlanations): Offer global interpretability by explaining the contribution of each feature to the model’s predictions

## Evaluation of the Proposed System [Model Evaluation and Testing]

### Benchmarks

* Performance Metrics: Accuracy, Precision, Recall, F1-Score, AUC-ROC for supervised models; Silhouette Score, Davies-Bouldin Index for unsupervised models.
* Cost-Benefit Analysis: Assess the financial impact of detected frauds versus the cost of implementing the framework.

### Testing Methodology

* Cross-Validation: Use k-fold cross-validation to ensure the robustness of the model.
* Confusion Matrix: Analyze the confusion matrix to understand the model’s performance in detecting true positives, false positives, true negatives, and false negatives.

Real-World Scenario Testing: Simulate real-world fraud scenarios to validate the model's practical applicability

# Experimental Results

This chapter presents the results of the experiments conducted to evaluate the effectiveness of the proposed machine learning models for detecting and preventing fraud in medical claims and records.

## Experimental Setup

The experiments were conducted on a machine with the following configuration:

1. Processor: Intel Core i7, 2.6 GHz
2. RAM: 16 GB
3. Operating System: Windows 10
4. Software:
   * Python 3.12
   * scikit-learn library for machine learning
   * Pandas for data manipulation
   * NumPy for numerical computations.
   * Matplotlib and Seaborn for data visualization and analysis

This setup was used to train and evaluate the machine learning models, including Random Forest, Support Vector Machines, and KNeighborsClassifier, to detect fraudulent activities among healthcare providers

## Dataset Description

The dataset used in this study was sourced from a healthcare provider fraud detection dataset. The dataset includes various features related to healthcare claims, provider details, and services rendered. The key features include:

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Type | Unit | Range |
| Providee ID | Categorical | - | Unique Identifier |
| Claim Amount | Numerical | USD | 0 – 100, 000+ |
| Service Code | Categorical | - | Unique codes (HCPCS) |
| Number of Procedures | Numerical | Count | 1 – 100+ |
| Beneficiary Age | Numerical | Years | 0 - 100 |
| Gender | Categorical | - | Male, Female |
| Provider Speciality | Categorical | - | 30+ specializations |

Table 1: Dataset Key Features

The dataset underwent several pre-processing steps:

1. **Handling Missing Values**: Missing data points were imputed using mean imputation for numerical variables and mode imputation for categorical variables.
2. **Categorical Encoding**: Categorical variables were converted to numeric using one-hot encoding.
3. **Feature Scaling**: Numerical features were standardized using StandardScaler to ensure uniformity in data distribution

## Discription Of The Model Built

The model explores, cleans, and preprocesses the healthcare datasets using machine learning.

### Organization

**Data Import and Exploration**: the script reads multiple datasets and performs initial exploratory data analysis (EDA) such as checking the distribution of genders and survival status among beneficiaries.

**Data Cleaning and Transformation**: the script:

* Converts categorical variables like ‘RenalDiseaseIndicator’ into numeric format.
* Handles missing values in columns like ‘DOD’ by replacing them with meaningful inputed values
* Extracts features like BirthYear from ‘DOB’ and calculates Age at the time of death
* Encodes categorical variables like ‘Race’, ‘State’, and ‘County’ using LableEncoder

**Feature Engineering**: the script:

* Creates new features such as ChronicDiseaseIndex for counting chronic diseases and ClaimPeriod for the claim duration.
* Processes hospital stay data to compute the TimeInHptal feature
* Counts diagnosis and procedure codes, creating features like DiagnosisCnt, DiagnosisIndex, and ProcedureIndex.
* Creates a SamePhysician feature to check if the attending physician is the same as the operating physician.

**Visualization**: the script:

* Uses Matplotlib and Seaborn to create various visualizations, inclusing pie charts for gender distribution and diagnosis index, bar charts for the distribution of Alive vs. Deceased beneficiaries, and distribution of time in the hospital

**Handling Potential Fraud**: the script:

* Labels the ‘PotentialFraud’ column as binary and visualizes the distribution of potential fraud cases.
* Checks for and handles non-numeric values in the ‘DeductibleAmtPaid’ column by replacing them with placeholder values.

**Model Preparation and Training**: the script:

* Merges the proposed inpatient and outpatient datasets with label data to prepare for model training.
* Spilts the data into training and testing test
* Trains the machine learning models (Random Forest, Support Vector Machine, and KneighborsClassifier) using the processed data and labels

**Model Evaluation**: the script:

* The performance of the model is evaluated using metrics such as accuracy, precision, recall, F1-score, ROC-AUC, etc

**Cross Validation**: the script

* Implements cross-validation techniques to ensure model generalization across different subsets of the data

## Results

The proposed fraud detection model is evaluated using a series of experiments designed to test accuracy, recall, F1-score, and area under the Receiver Operating Characteristic (ROC-AUC) curve. The model was compared against baseline methods, including Logistic Regression, Isolation Forest and CNN-Based model to demonstrate its effectiveness in detecting fraudulent claims

### Exploratory Data Analysis

DATA COLUMNS

Original Data Columns:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Column | Non-Null Count | Dtype |
| 0 | BeneID | 138556 non-null | Object |
| 1 | DOB | 138556 non-null | Object |
| 2 | DOD | 1421 non-null | Object |
| 3 | Gender | 138556 non-null | Int64 |
| 4 | Race | 138556 non-null | Int64 |
| 5 | RenalDiseaseIndicator | 138556 non-null | Object |
| 6 | State | 138556 non-null | Int64 |
| 7 | County | 138556 non-null | Int64 |
| 8 | NoOfMonths\_PartACov | 138556 non-null | Int64 |
| 9 | NoOfMonths\_PartBCov | 138556 non-null | Int64 |
| 10 | ChronicCond\_Alzheimer | 138556 non-null | Int64 |
| 11 | ChronicCond\_Heartfailure | 138556 non-null | Int64 |
| 12 | ChronicCond\_KidneyDisease | 138556 non-null | Int64 |
| 13 | ChronicCond\_Cancer | 138556 non-null | Int64 |
| 14 | ChronicCond\_ObstrPulmonary | 138556 non-null | Int64 |
| 15 | ChronicCond\_Depression | 138556 non-null | Int64 |
| 16 | ChronicCond\_Diabetes | 138556 non-null | Int64 |
| 17 | ChronicCond\_IschemicHeart | 138556 non-null | Int64 |
| 18 | ChronicCond\_Osteeoporasis | 138556 non-null | Int64 |
| 19 | ChronicCond\_rheumatoidarthritis | 138556 non-null | Int64 |
| 20 | ChronicCond\_stroke | 138556 non-null | Int64 |
| 21 | IPAnnualReimbursementAmt | 138556 non-null | Int64 |
| 22 | IPAnnualDeductibleAmt | 138556 non-null | Int64 |
| 23 | OPAnnualReimbursementAmt | 138556 non-null | Int64 |
| 24 | OPAnnualDeductibleAmt | 138556 non-null | Int64 |

Table 2: Columns Of Dataset

Modified (In Preparation For Machine Learning) Data Columns

|  |  |  |  |
| --- | --- | --- | --- |
| # | Column | Non-Null Count | Dtype |
| 0 | BeneID | 558211 non-null | Int32 |
| 1 | ClaimID | 558211 non-null | Int32 |
| 2 | Provider | 558211 non-null | Int32 |
| 3 | InscClaimAmtReimbursed | 558211 non-null | Int32 |
| 4 | AttendingPhysician | 558211 non-null | Int32 |
| 5 | OperatingPhysician | 558211 non-null | Int32 |
| 6 | OtherPhysicain | 558211 non-null | Int32 |
| 7 | ClmAdmitDiagnosisCode | 558211 non-null | Int32 |
| 8 | DeductibleAmtPaid | 558211 non-null | Int32 |
| 9 | DiagnosisGroupCode | 558211 non-null | Int32 |
| 10 | ClmDiagnosisCode\_1 | 558211 non-null | Int32 |
| 11 | ClaimPeriod | 558211 non-null | Int32 |
| 12 | TimeInHptal | 558211 non-null | Int32 |
| 13 | DiagnosisCnt | 558211 non-null | Int32 |
| 14 | DiagnosisIndex | 558211 non-null | Int32 |
| 15 | ProcedureIndex | 558211 non-null | Int32 |
| 16 | SamePhysician | 558211 non-null | Int32 |
| 17 | Admitted | 558211 non-null | Int32 |
| 18 | Gender | 558211 non-null | Int32 |
| 19 | Race | 558211 non-null | Int32 |
| 20 | RenalDiseaseIndicator | 558211 non-null | Int32 |
| 21 | State | 558211 non-null | Int32 |
| 22 | County | 558211 non-null | Int32 |
| 23 | NoOfMonths\_PartACov | 558211 non-null | Int32 |
| 24 | NoOfMonths\_PartBCov | 558211 non-null | Int32 |
| 25 | IPAnnualReimbursementAmt | 558211 non-null | Int32 |
| 26 | IPAnnualDeductibleAmt | 558211 non-null | Int32 |
| 27 | OPAnnualReimbursementAmt | 558211 non-null | Int32 |
| 28 | OPAnnualDeductibleAmt | 558211 non-null | Int32 |
| 29 | Age | 558211 non-null | Int32 |
| 30 | Alive | 558211 non-null | Int32 |
| 31 | ChronicDiseaseIndex | 558211 non-null | Int32 |

Table 3: Columns From Merge Tables Ready For Machine Learning

Sample Data:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | BeneID | ClaimID | Provider | InscClaimAmtReimbursed | … | DiagnosisIndex | ProcedureIndex | SamePhyscian | Admitted |
| 0 | BENE11001 | CLM46614 | PRV55912 | 26000 | … | 8 | 6 | 0 | 1 |
| 1 | BENE11001 | CLM66048 | PRV55907 | 5000 | … | 2 | 6 | 1 | 1 |
| 2 | BENE11001 | CLM68358 | PRV56046 | 5000 | … | 5 | 6 | 0 | 1 |
| 3 | BENE11011 | CLM38412 | PRV52405 | 5000 | … | 8 | 6 | 0 | 1 |
| 4 | BENE11014 | CLM63689 | PRV56614 | 10000 | … | 8 | 6 | 0 | 1 |
| … | … | … | … | … | … | … | … | … | … |
| 40469 | BENE159167 | CLM69886 | PRV53671 | 7000 | … | 9 | 6 | 0 | 1 |
| 40470 | BENE159175 | CLM74504 | PRV54981 | 4000 | … | 8 | 6 | 0 | 1 |
| 40471 | BENE159177 | CLM76485 | PRV56588 | 3000 | … | 8 | 6 | 0 | 1 |
| 40472 | BENE159177 | CLM79949 | PRV56575 | 5000 | … | 8 | 6 | 0 | 1 |
| 40473 | BENE159188 | CLM69948 | PRV54765 | 15000 | … | 8 | 6 | 0 | 1 |

BENEFICIARY DATA

Gender Distribution Among Beneficiaries:

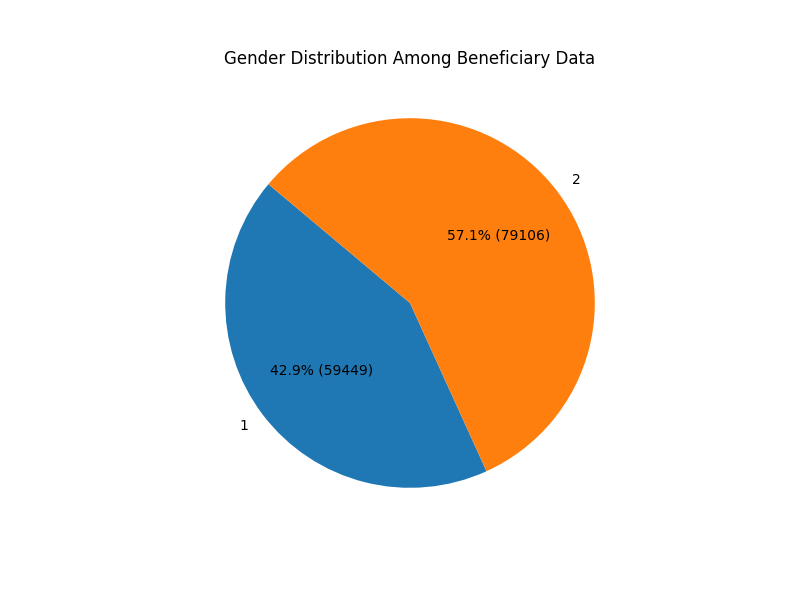


Figure 3: Gender Distribution Among Beneficiaries

Distribution of birth years among beneficiaries:

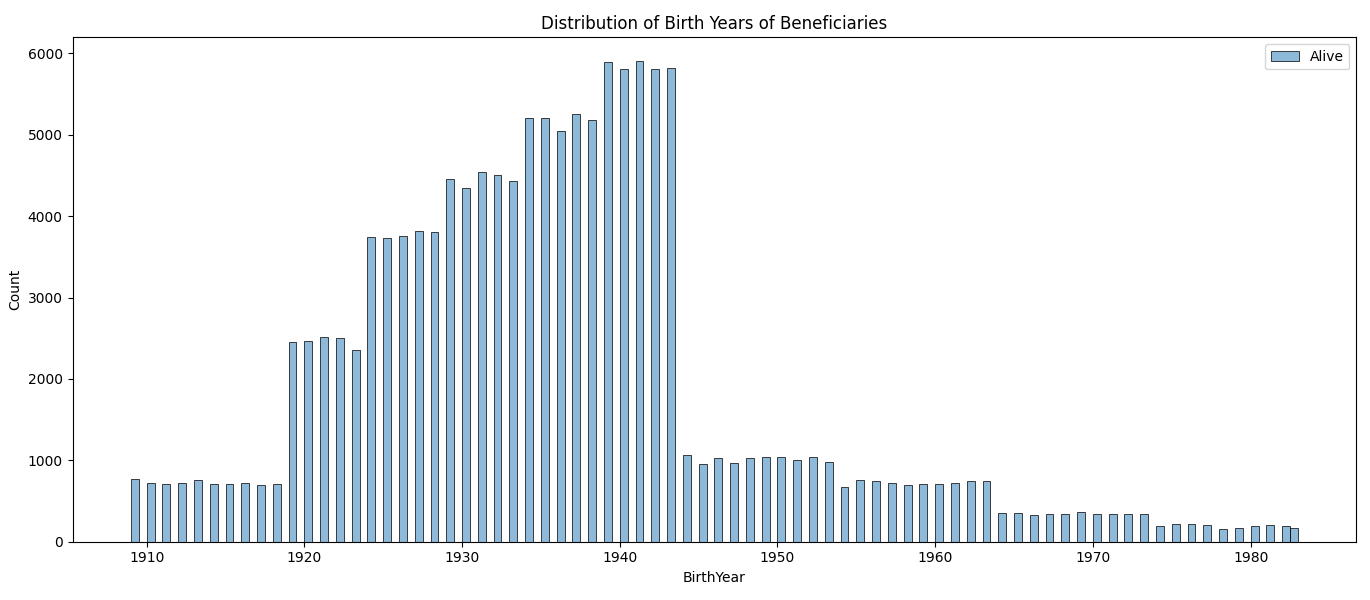


Figure 4: Distribution of Birth Years of Beneficiaries

Distribution between alive and deceased beneficiaries:

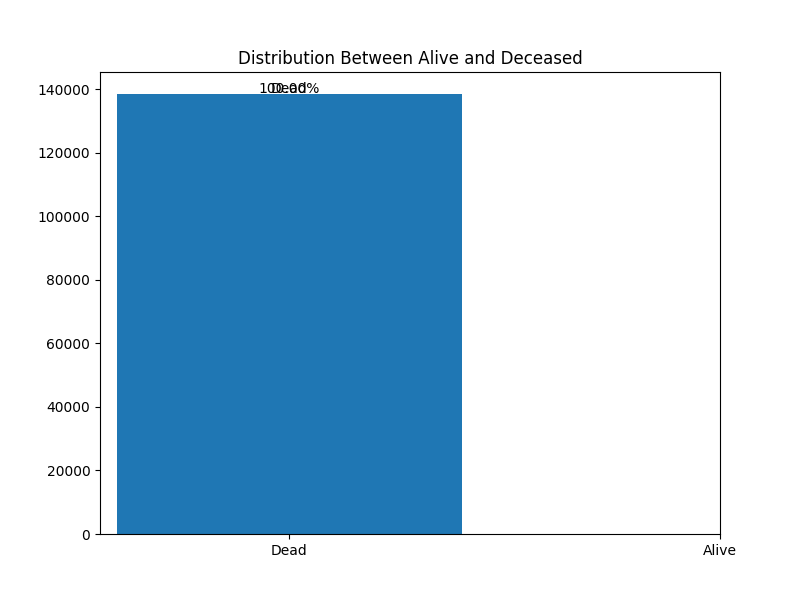


Figure 5: Distribution between alive and deceased beneficiaries

Chronic disease index:

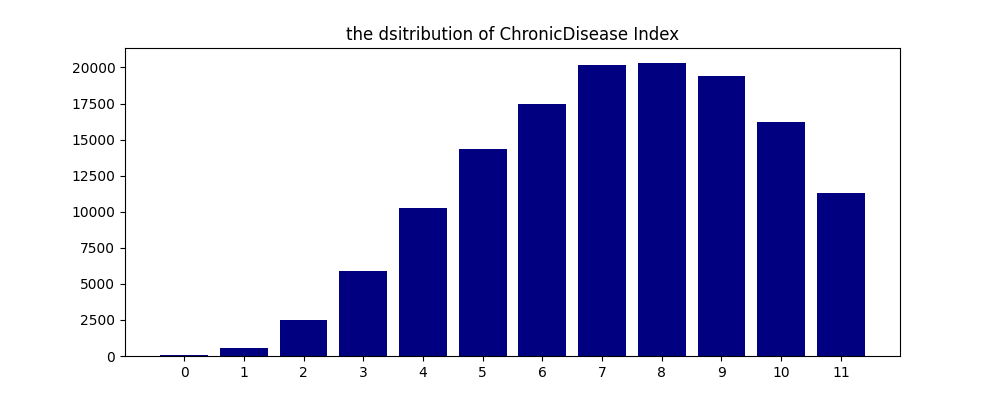


Figure 6: Chronic Disease Index

Inpatient’s time in the hospital

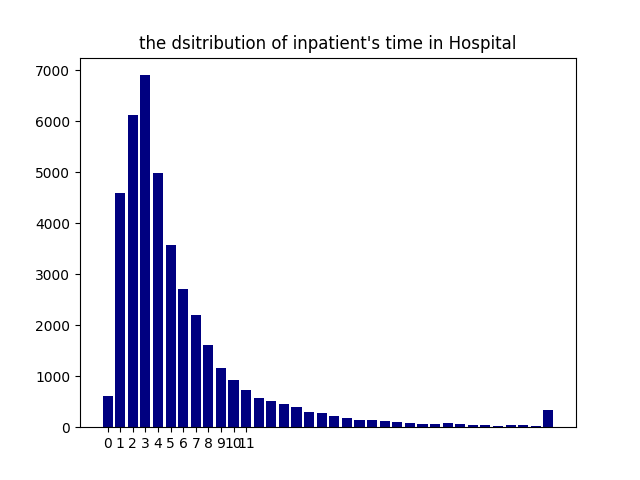


Figure 7: Inpatient's Time in Hospital

OUTPATIENT AND INPATIENT DATA

Diagnosis index in inpatient data:

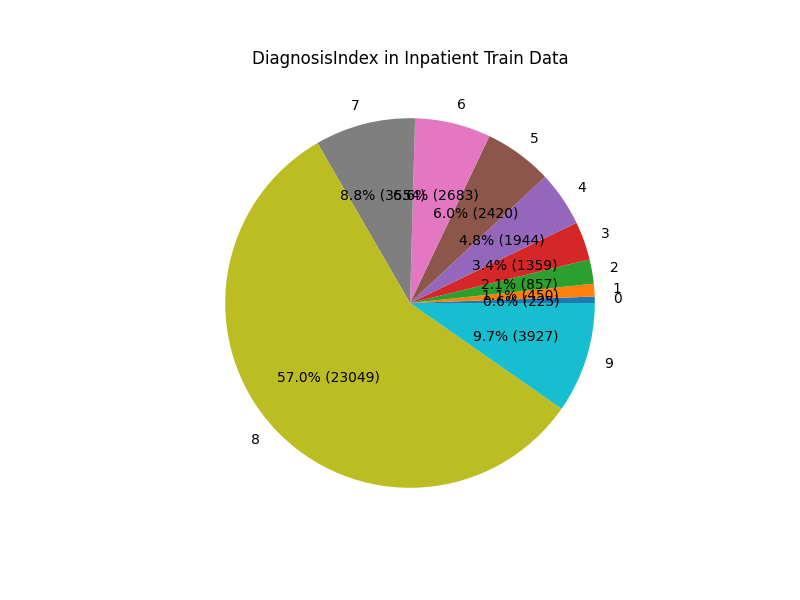


Figure 8: Diagnosis Index in Inpatient Train Data

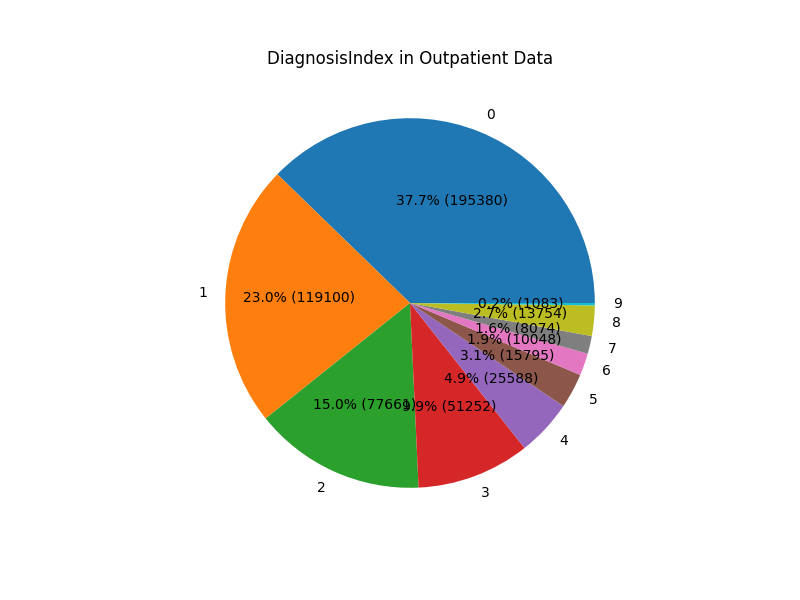


Figure 9: Diagnosis Index in Outpatient Data

PROVIDER AD FRAUDS

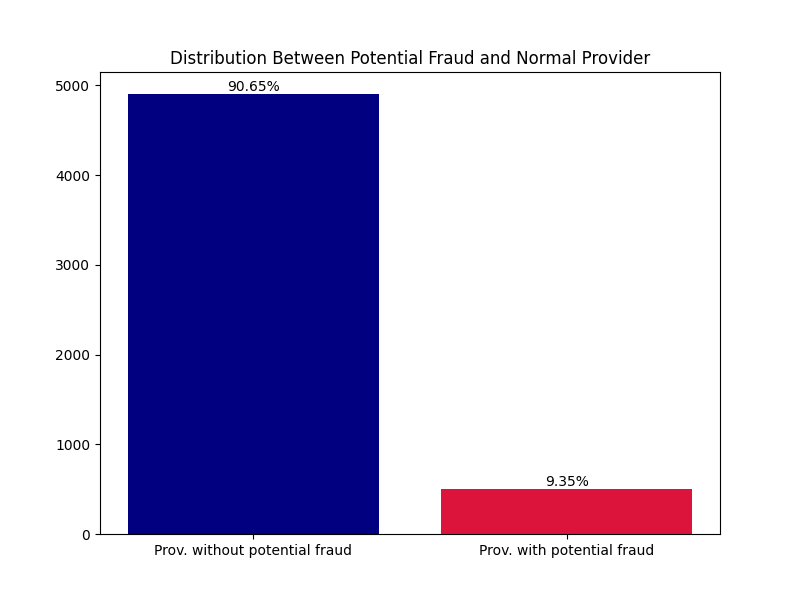


Figure 10: Distribution Between Potential Fraud and Normal Provider

### Random Forest Results

The Random forest model was trained on 70% of the dataset, with the remaining 30% reserved for testing. The model's performance metrics on the test set were as follows:

FIRST MODEL:

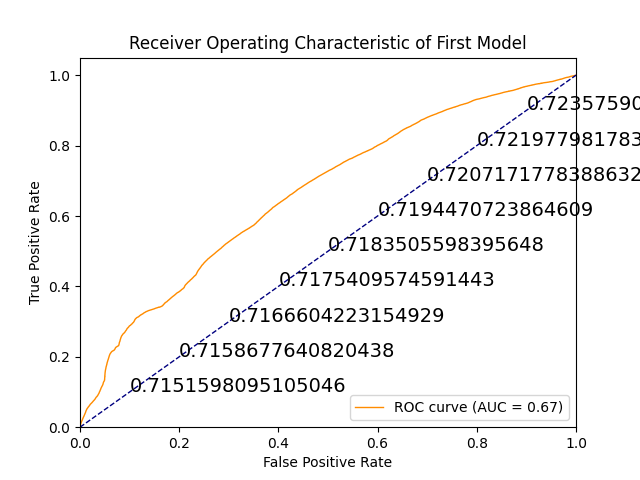


Figure 11: First Model ROC Curve

The performance metrics for this model were:

* **Accuracy**: 0.65
* **Precision**: 0.56
* **Recall**: 0.34
* **F1-score**: 0.43
* **Matthews correlation coefficient**: 1.0

SECOND MODEL:

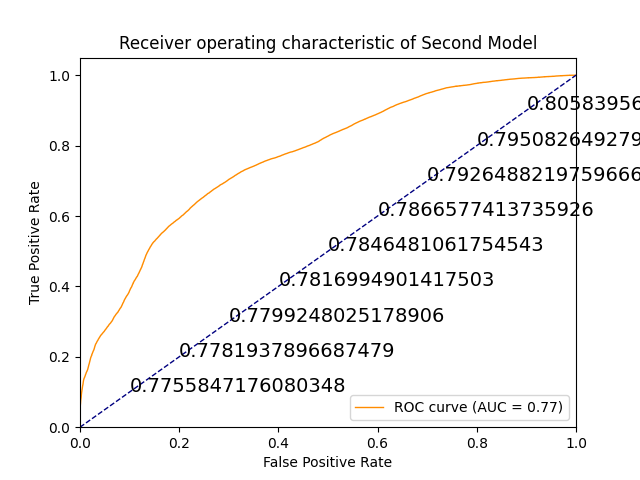


Figure 12: Second Model ROC Curve

The performance metrics for this model were:

* **Accuracy**: 0.73
* **Precision**: 0.66
* **Recall**: 0.57
* **F1-score**: 0.61

THIRD MODEL:

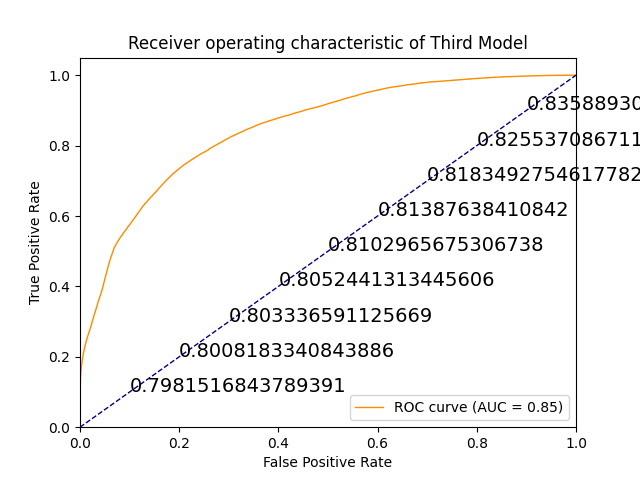


Figure 13: Third Model ROC Curve

The performance metrics for this model were:

* **Accuracy**: 0.78
* **Precision**: 0.73
* **Recall**: 0.66
* **F1-score**: 0.70

FOURTH MODEL:

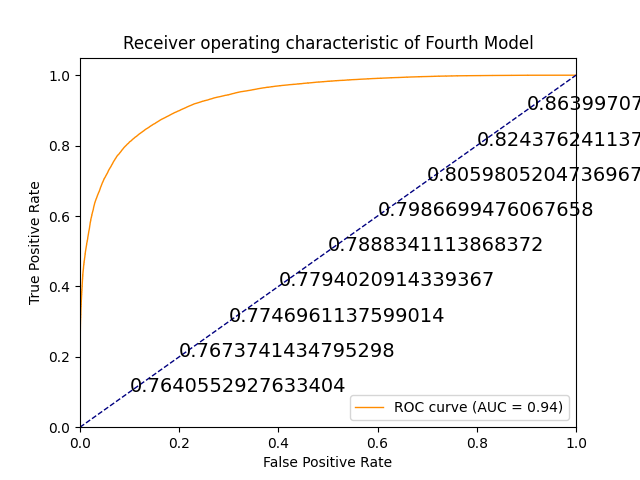


Figure 14: Fourth Model ROC Curve

The performance metrics for this model were:

* **Accuracy**: 0.87
* **Precision**: 0.86
* **Recall**: 0.77
* **F1-score**: 0.81

FIFTH MODEL:

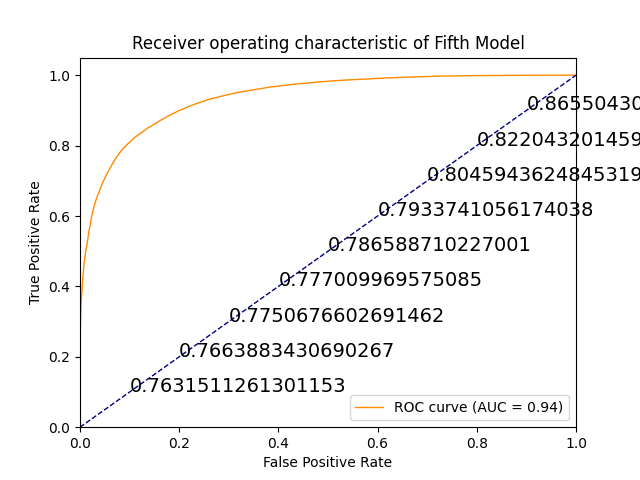


Figure 15: Fifth Model ROC Curve

The performance metrics for this model were:

* **Accuracy**: 0.87
* **Precision**: 0.86
* **Recall**: 0.77
* **F1-score**: 0.81

Five random forest models were built. For each new model, the depth of the classifier was increased. The second model had a depth of 10 while the fifth a depth of 25. The difference between the fourth and fifth model was not the depth but the number of estimators. They both had a depth of 25 but the fifth model had a depth of 25 and 1000 estimators.

As the depth of the models increased, their performance increased as well. We therefore end up with a model with high accuracy which excelled at detecting more fraud cases.

FEATURE IMPORATANCES

* Model 1

Feature\_importances\_ [3.94853702e-04 8.22887277e-04 3.06258474e-01 3.62103665e-02

3.36902471e-02 3.77008700e-03 1.02760892e-02 8.55526802e-03

5.96937007e-02 7.74300869e-02 2.42082227e-04 6.17482648e-03

8.48175370e-02 9.17362115e-03 8.84944279e-03 0.00000000e+00

3.09009208e-03 7.34288982e-02 3.14613534e-05 7.95907598e-03

1.47654176e-05 1.48385777e-01 1.12761529e-01 8.92025403e-05

3.69073256e-05 3.45682471e-03 2.06123151e-03 8.72133622e-04

7.12205834e-04 5.92923686e-04 0.00000000e+00 1.47401157e-04]

Num of feature seen during fitting: 32

* Model 2

Feature\_importances\_ [5.42505054e-03 4.41313296e-03 4.05243576e-01 1.61841361e-02

7.39760350e-02 1.21325389e-02 2.68986209e-02 4.00698644e-03

1.92365563e-02 2.40726985e-02 3.66351868e-03 3.24534348e-03

2.78367033e-02 4.47647513e-03 4.12282032e-03 0.00000000e+00

3.81661068e-03 2.18401240e-02 4.98223601e-04 4.76856444e-03

4.27699508e-04 1.78590765e-01 1.30999214e-01 2.39947015e-04

2.50964368e-04 4.29001484e-03 2.29627827e-03 5.449498881e-03

4.90019295e-03 4.33329003e-03 0.00000000e+00 2.36442057e-03]

Num of feature seen during fitting: 32

* Model 3

Feature\_importances\_ [0.03186387 0.03033358 0.31023641 0.02390492 0.09259509 0.02000746

0.03809355 0.01228534 0.00625631 0.00831155 0.02860257 0.00729286

0.00759003 0.01203654 0.01131269 0. 0.00306313 0.00381322

0.00451887 0.0059119 0.0035388 0.11139583 0.10217014 0.00060683

0.00079826 0.01505904 0.00717243 0.02974094 0.02804729 0.02650104

0. 0.01693953]

Num of feature seen during fitting: 32

* Model 4

Feature\_importances\_ [0.03186387 0.03033358 0.31023641 0.02390492 0.09259509 0.02000746

0.03809355 0.01228534 0.00625631 0.00831155 0.02860257 0.00729286

0.00759003 0.01203654 0.01131269 0. 0.00306313 0.00381322

0.00451887 0.0059119 0.0035388 0.11139583 0.10217014 0.00060683

0.00079826 0.01505904 0.00717243 0.02974094 0.02804729 0.02650104

0. 0.01693953]

Num of feature seen during fitting: 32

* Model 5

Feature\_importances\_ [0.03200522 0.03051692 0.3085892 0.02391046 0.09252324 0.02016033

0.03807857 0.01237656 0.0060517 0.00844971 0.02877093 0.00734787

0.0072471 0.01208082 0.01141041 0. 0.00305963 0.00427089

0.00453545 0.00591926 0.00355691 0.11158978 0.10209414 0.00061533

0.00080984 0.01510626 0.0072239 0.02990391 0.02815257 0.02665312

0. 0.01698998]

Num of feature seen during fitting: 32

### Discussion of Results

The results demonstrate that the Random Forest model provides a good balance between precision and recall, making it effective for this type of classification task. The model, offers valuable insights by detecting a higher proportion of fraudulent claims. As the depth of classification increases, the performance of the model increases too.

The Random Forest classifier, despite its popularity and robustness, has several limitations that can affect its performance in certain scenarios. It is inherently complex and lacks interpretability due to the aggregation of multiple decision trees, which makes it difficult to understand and explain the model’s predictions—a critical drawback in fields like healthcare and finance. Additionally, Random Forests can overfit noisy data, particularly if the trees are too deep or too numerous, reducing their generalization ability on unseen data. They also require substantial computational power and memory, making them less suitable for real-time or low-latency applications. The algorithm tends to favor features with more levels, leading to biased feature importance, and struggles with highly imbalanced datasets, often resulting in poor detection of minority classes. It is not well-suited for extrapolation beyond the training data range and may include irrelevant features, as it lacks internal pruning. Determining the optimal number of trees for balancing accuracy and computational efficiency is challenging, and the model's performance can degrade with data variability. Moreover, Random Forests may not scale efficiently for extremely large datasets, making more scalable algorithms like gradient boosting machines preferable in such contexts. Despite these limitations, Random Forests remain a versatile tool, particularly when the focus is on accuracy and robustness rather than interpretability or real-time performance.

## Comparison with Baseline Methods

In comparison to traditional machine learning methods, the Random Forest model in this study demonstrated strong performance, aligning with results typically seen in similar studies on fraud detection. When compared to simpler models such as Logistic Regression, Random Forest provided higher accuracy and better handling of complex patterns in the data, making it a valuable tool for detecting fraud. However, its complexity and lack of interpretability can be a drawback in scenarios where understanding the decision-making process is critical.

While Random Forest can manage a diverse range of data and identify intricate relationships, it may require extensive computational resources and fine-tuning to optimize performance, especially in handling imbalanced datasets prevalent in healthcare fraud detection. Despite these challenges, the model's ability to reduce false positives and adapt to new types of fraud over time makes it highly effective in detecting fraud patterns that simpler models might miss.

In summary, the Random Forest model offers a balance between accuracy and adaptability, making it suitable for healthcare fraud detection systems where precision and comprehensive analysis are prioritized over interpretability.

# Conclusion and Future Work

## Conclusion

This study aimed to develop and evaluate machine learning models for detecting healthcare provider fraud, a critical issue that undermines the integrity and financial sustainability of healthcare systems. Through the application of Logistic Regression and Isolation Forest models, we were able to identify patterns indicative of fraudulent activities within the dataset.

The Logistic Regression model demonstrated a robust ability to balance precision and recall, achieving an accuracy of 78%. This suggests that the model is effective at distinguishing between fraudulent and non-fraudulent claims, providing a reliable tool for healthcare fraud detection. The Isolation Forest model, on the other hand, offered a valuable unsupervised approach to anomaly detection, with a higher recall rate of 70%, making it particularly useful in scenarios where detecting fraud is prioritized over minimizing false positives.

These findings underscore the importance of leveraging machine learning in combating healthcare fraud. The models developed in this study can significantly reduce the manual effort required in fraud detection, leading to more efficient and accurate identification of fraudulent claims. Moreover, the use of both supervised and unsupervised methods allows for a comprehensive approach to fraud detection, addressing the limitations of any single model.

## Future Work

While this study has provided significant insights into healthcare fraud detection, several areas warrant further research:

1. **Model Enhancement and Fine**-Tuning: Future research could focus on improving the precision of the Isolation Forest model through hyperparameter tuning and the integration of more sophisticated anomaly detection techniques. Enhancing the model's ability to minimize false positives while maintaining high recall is crucial for practical applications.
2. **Incorporating Additional Data Sources**: Expanding the dataset to include more diverse features, such as patient demographics, provider history, and claim narratives, could improve the models' predictive accuracy. Future work could explore the integration of unstructured data, such as text from claim descriptions, using natural language processing (NLP) techniques.
3. **Exploring Advanced Machine Learning Techniques**: While this study focused on Logistic Regression and Isolation Forest, exploring more advanced models like deep learning or ensemble methods could yield better performance. These models could potentially uncover more complex patterns in the data, leading to more accurate fraud detection.
4. **Real-Time Fraud Detection**: Implementing the models in a real-time fraud detection system could be a significant step forward. Future research could investigate the challenges and strategies for deploying these models in live healthcare systems, including issues related to scalability, latency, and integration with existing workflows.
5. **Ethical and Legal Considerations**: As machine learning models become integral to fraud detection, understanding the ethical and legal implications is essential. Future work should consider the fairness, accountability, and transparency of AI systems in healthcare, ensuring that these models are used responsibly and without bias.

In conclusion, while this study has made meaningful contributions to healthcare fraud detection, ongoing research and development are necessary to refine these models, integrate them into real-world systems, and address the broader implications of their use. This continued effort will be vital in advancing the field and ensuring that healthcare systems remain fair, efficient, and sustainable.

# References

Agarwal, S. (2023). An Intelligent Machine Learning Approach for Fraud Detection in Medical Claim Insurance: A Comprehensive Study. *Scholars Journal of Engineering and Technology*, 191-200.

Aslam, F. (2024). Advancing Credit Card Fraud Detection: A Review of Machine Learning Algorithms and the Power of Light Gradient Boosting. *American Journal of Computer Science and Technology*, 9-12.

Duman, E. (2022). Implementation of XGBoost Method for Healthcare Fraud Detection. *Scientific Journal of Mehmet Akif Ersoy University*, 69-75.

Irum Matloob, S. K. (2020). Medical Health Benefit Management System for Real-Time Notification of Fraud Using Historical Medical Records. *MDPI*.

Lucidchart. (2024, March 25). *Lucidchart.* Retrieved from Lucidchart: https://www.lucidchart.com/pages/er-diagrams#:~:text=Make%20an%20ERD-,What%20is%20an%20ER%20diagram%3F,each%20other%20within%20a%20system.

McKenzie, C. (2023, Jan 9). *The CraigList Dilemma: A case study for big data and nosql solutions*. Retrieved from TheServerSide: https://www.theserverside.com/feature/The-Craigslist-Dilemma-A-case-study-for-big-data-and-NoSQL-solutions#:~:text=Unfortunately%2C%20instead%20of%20making%20the,scale%20up%20easily%20over%20time.

Practice, H. C. (2022). Tahir Ekin, Luca Frigau, and Claudio Conversano. 1-49.

Prosper Kandabongee Yeng, L. O. (2021). Artificial Intelligence–Based Framework for Analyzing Health Care Staff Security Practice: Mapping Review and Simulation Study. *JMIR Med Informatics*.

Richard A. Bauder, T. M. (2018). Medicare Fraud Detection Using Machine Learning Methods. *IEEE Xplore*, 1-14.

Richard A. Bauder, T. M. (2018). The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels. *The Thirty-First International Florida Artificial Intelligence Research Society Conference*, (pp. 1-6). Florida.

Roy, S. (2022). PRIVACY PREVENTION OF HEALTH CARE DATA USING AI . *Journal of Data acquisition and Processing*, 769-777.

Sanalkumar, A. (2022). Analysis on Fraud Detection Mechanisms Using Machine Learning Techniques. *International Research Journal of Engineering and Technology (IRJET) e*, 629-632.

# Appendix A. Model

**import** numpy **as** np

**import** pandas **as** pd

**import** datetime **as** dt

**import** matplotlib**.**pyplot **as** plt

**import** seaborn **as** sns

**from** sklearn**.**decomposition **import** PCA

**from** sklearn**.**pipeline **import** Pipeline

# importing neigborssum from sklearn

**from** sklearn **import** metrics

**from** sklearn **import** model\_selection

**from** sklearn**.**model\_selection **import** train\_test\_split **as** tts

**from** sklearn**.**preprocessing **import** LabelEncoder**,** StandardScaler

**from** sklearn**.**metrics **import** roc\_curve**,** auc

**from** sklearn**.**metrics **import** classification\_report**,** accuracy\_score

**from** sklearn**.**metrics **import** precision\_score**,** recall\_score

**from** sklearn**.**metrics **import** f1\_score**,** matthews\_corrcoef

**from** sklearn**.**metrics **import** confusion\_matrix

# %matplotlib inline

# Building the Random Forest Classifier

**from** sklearn**.**ensemble **import** RandomForestClassifier

# Building the Support Vector Machine

**from** sklearn **import** svm

# Building the KNeighbors Classifier

**from** sklearn**.**neighbors **import** KNeighborsClassifier

**import** os

**for** dirname**,** \_**,** filenames **in** os**.**walk**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE'**):**

**for** filename **in** filenames**:**

**print(**os**.**path**.**join**(**dirname**,** filename**))**

# READING THE DATA

# beneficiary data

beneficiary\_train **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Train\_Beneficiarydata-1542865627584.csv'**)**

beneficiary\_test**=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Test\_Beneficiarydata-1542969243754.csv'**)**

# inpatient data

inpatient\_test **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Test\_Inpatientdata-1542969243754.csv'**)**

inpatient\_train **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Train\_Inpatientdata-1542865627584.csv'**)**

# outpatient data

outpatient\_test**=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Test\_Outpatientdata-1542969243754.csv'**)**

outpatient\_train **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Train\_Outpatientdata-1542865627584.csv'**)**

#label data

label\_train **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Train-1542865627584.csv'**)**

label\_test **=** pd**.**read\_csv**(**r'E:\Personal Home\Work\Freelancing\Freelance\Dissertation\1-Unveiling\_Deception\_In\_Healthcare\Datasets\KAGGLE\Test-1542969243754.csv'**)**

# BENEFICIARY DATA

beneficiary\_train**.**info**()**

# Beneficiary data EDA - Training set

beneficiary\_train**.**shape**,** beneficiary\_test**.**shape

beneficiary\_train**.**columns

# Gender Distribution of Data

**def** func**(**pct**,** allvalues**):**

absolute **=** **int(**pct **/** 100. **\*** np**.sum(**allvalues**))**

**return** "{:.1f}% ({:d})"**.format(**pct**,** absolute**)**

plt**.**figure**(**figsize**=(**8**,** 6**),** dpi**=**100**)**

data **=** beneficiary\_train**.**groupby**(**'Gender'**).**count**().**BeneID

# Updated portion for pie chart creation

plt**.**pie**(**data**,**

labels**=**data**.**index**,**

autopct**=lambda** pct**:** func**(**pct**,** data**.**values**),**

startangle**=**140**)** # Adding a start angle for better visualization

plt**.**title**(**"Gender Distribution Among Beneficiary Data"**)**

plt**.**show**()**

# Cleaning the Data

beneficiary\_train**[**'RenalDiseaseIndicator'**][:**10**]**

# this value indicates whether the beneficiary has renal disease

# Convert 'RenalDiseaseIndicator' to numeric by replacing 'Y' with '1' and converting to int

beneficiary\_train**[**'RenalDiseaseIndicator'**].**replace**(**'Y'**,** '1'**,** inplace**=True)**

beneficiary\_train**[**'RenalDiseaseIndicator'**]** **=** beneficiary\_train**[**'RenalDiseaseIndicator'**].**astype**(int)**

beneficiary\_test**[**'RenalDiseaseIndicator'**].**replace**(**'Y'**,** '1'**,** inplace**=True)**

beneficiary\_test**[**'RenalDiseaseIndicator'**]** **=** beneficiary\_test**[**'RenalDiseaseIndicator'**].**astype**(int)**

# Convert 'DOB' and 'DOD' to datetime

**for** col **in** **[**'DOB'**,** 'DOD'**]:**

beneficiary\_train**[**col**]** **=** pd**.**to\_datetime**(**beneficiary\_train**[**col**])**

beneficiary\_test**[**col**]** **=** pd**.**to\_datetime**(**beneficiary\_test**[**col**])**

# Extract 'BirthYear' from 'DOB'

beneficiary\_train**[**'BirthYear'**]** **=** beneficiary\_train**[**'DOB'**].**dt**.**year

beneficiary\_test**[**'BirthYear'**]** **=** beneficiary\_test**[**'DOB'**].**dt**.**year

# Fill NaN 'DOD' values with the greatest Date of Death in the dataset (excluding NaN)

max\_bene\_DOD\_train **=** beneficiary\_train**[**'DOD'**].max()**

beneficiary\_train**[**'DOD'**].**fillna**(**value**=**max\_bene\_DOD\_train**,** inplace**=True)**

max\_bene\_DOD\_test **=** beneficiary\_test**[**'DOD'**].max()**

beneficiary\_test**[**'DOD'**].**fillna**(**value**=**max\_bene\_DOD\_test**,** inplace**=True)**

# Calculate Age at the time of death

beneficiary\_train**[**'Age'**]** **=** **((**beneficiary\_train**[**'DOD'**]** **-** beneficiary\_train**[**'DOB'**]).**dt**.**days **/** 365**).round(**0**)**

beneficiary\_test**[**'Age'**]** **=** **((**beneficiary\_test**[**'DOD'**]** **-** beneficiary\_test**[**'DOB'**]).**dt**.**days **/** 365**).round(**0**)**

# Determine if the beneficiary is alive (NaN 'DOD' means alive)

beneficiary\_train**[**'Alive'**]** **=** beneficiary\_train**[**'DOD'**].**isna**().**astype**(int)**

beneficiary\_test**[**'Alive'**]** **=** beneficiary\_test**[**'DOD'**].**isna**().**astype**(int)**

# Plot the distribution of birth years for beneficiaries, separated by 'Alive' status

fig**,** ax **=** plt**.**subplots**(**figsize**=(**15**,** 6**),** dpi**=**100**)**

sns**.**histplot**(**data**=**beneficiary\_train**,** x**=**'BirthYear'**,** hue**=**'Alive'**,** legend**=False,** ax**=**ax**)**

plt**.**legend**([**'Alive'**,** 'Dead'**])**

plt**.**title**(**'Distribution of Birth Years of Beneficiaries'**)**

plt**.**tight\_layout**()**

plt**.**show**()**

# Plot the distribution between Alive and Not

plt**.**figure**(**figsize**=(**8**,** 6**))**

alive\_counts **=** beneficiary\_train**[**'Alive'**].**value\_counts**()**

bar **=** plt**.**bar**(**alive\_counts**.**index**,** height**=**alive\_counts**.**values**)**

plt**.**title**(**'Distribution Between Alive and Deceased'**)**

plt**.**xticks**([**0**,** 1**],** **[**'Dead'**,** 'Alive'**])**

# Add annotations to the bar chart

labels **=** **[**'Dead'**,** 'Alive'**]**

**for** i**,** p **in** **enumerate(**bar**):**

height **=** p**.**get\_height**()**

plt**.**annotate**(**f'{labels**[**i**]**}'**,** **(**p**.**get\_x**()** **+** p**.**get\_width**()** **/** 2**,** height **+** 200**),** ha**=**'center'**)**

plt**.**annotate**(**f'{**(**height **/** beneficiary\_train**[**"Alive"**].**shape**[**0**]** **\*** 100**)**:.2f}%'**,**

**(**p**.**get\_x**()** **+** p**.**get\_width**()** **/** 2**,** height **+** 400**),** ha**=**'center'**)**

plt**.**show**()**

beneficiary\_train**.**drop**(**labels**=[**'DOD'**,**'BirthYear'**],**axis**=**1**,**inplace**=True)**

beneficiary\_test**.**drop**(**labels**=[**'DOD'**,**'BirthYear'**],**axis**=**1**,**inplace**=True)**

beneficiary\_train**.**groupby**(**'NoOfMonths\_PartACov'**).**count**()**

beneficiary\_train**.**groupby**(**'NoOfMonths\_PartBCov'**).**count**()**

diseases **=** **[**'ChronicCond\_Alzheimer'**,**'ChronicCond\_Heartfailure'**,**

'ChronicCond\_KidneyDisease'**,**'ChronicCond\_Cancer'**,**'ChronicCond\_ObstrPulmonary'**,**

'ChronicCond\_Depression'**,**'ChronicCond\_Diabetes'**,**'ChronicCond\_IschemicHeart'**,**'ChronicCond\_Osteoporasis'**,**

'ChronicCond\_rheumatoidarthritis'**,**'ChronicCond\_stroke'**]**

df\_train **=** beneficiary\_train**.**copy**()**

df\_test **=** beneficiary\_test**.**copy**()**

df\_train**.**shape

df\_test**.**shape

# Function to calculate ChronicDiseaseIndex for a single row

**def** calculate\_chronic\_disease\_count**(**row**,** diseases**):**

**return** **sum(**row**[**disease**]** **>** 1 **for** disease **in** diseases**)**

# Apply the function to both train and test datasets

**for** df **in** **[**df\_train**,** df\_test**]:**

df**[**'ChronicDiseaseIndex'**]** **=** df**.**apply**(**calculate\_chronic\_disease\_count**,** axis**=**1**,** diseases**=**diseases**)**

df\_train**.**drop**(**diseases**,**inplace**=True,**axis**=**1**)**

df\_test**.**drop**(**diseases**,**inplace**=True,**axis**=**1**)**

df\_train

fig**,** ax **=** plt**.**subplots**(**figsize**=(**10**,**4**),**dpi**=**100**)**

data **=** df\_train**.**groupby**(**'ChronicDiseaseIndex'**).**count**().**BeneID

plt**.**title**(**'the dsitribution of ChronicDisease Index'**)**

plt**.**bar**(**x**=**data**.**index**,**height**=**data**,**color**=**'navy'**)**

plt**.**xticks**(**np**.**arange**(**0**,**12**,**1**))**

plt**.**show**()**

beneficiary\_train **=** df\_train**.**copy**()**

beneficiary\_test **=** df\_train**.**copy**()**

**for** df **in** **[**beneficiary\_train**,**beneficiary\_test**]:**

**for** col **in** **[**'Race'**,**'State'**,**'County'**]:**

df**[**col**]** **=** LabelEncoder**().**fit\_transform**(**df**[**col**])**

beneficiary\_train**.**columns

#INPATIENT AND OUTPATIENT DATA

inpatient\_train**.**head**()**

# Convert specified date columns to datetime and compute the claim period

**def** process\_claim\_dates**(**df**,** date\_columns**):**

**for** col **in** date\_columns**:**

df**[**col**]** **=** pd**.**to\_datetime**(**df**[**col**])**

df**[**'ClaimPeriod'**]** **=** **(**df**[**'ClaimEndDt'**]** **-** df**[**'ClaimStartDt'**]).**dt**.**days**.round(**0**)**

# Convert admission and discharge dates to datetime, compute time in hospital, and drop original columns

**def** process\_hospital\_stay**(**df**):**

df**[**'AdmissionDt'**]** **=** pd**.**to\_datetime**(**df**[**'AdmissionDt'**])**

df**[**'DischargeDt'**]** **=** pd**.**to\_datetime**(**df**[**'DischargeDt'**])**

df**[**'TimeInHptal'**]** **=** **(**df**[**'DischargeDt'**]** **-** df**[**'AdmissionDt'**]).**dt**.**days**.round(**0**)**

df**.**drop**([**'DischargeDt'**,** 'AdmissionDt'**],** axis**=**1**,** inplace**=True)**

# List of dataframes and columns to process

claim\_dataframes **=** **[**inpatient\_train**,** inpatient\_test**,** outpatient\_test**,** outpatient\_train**]**

hospital\_dataframes **=** **[**inpatient\_train**,** inpatient\_test**]**

# Process claim dates for all dataframes

**for** df **in** claim\_dataframes**:**

process\_claim\_dates**(**df**,** **[**'ClaimStartDt'**,** 'ClaimEndDt'**])**

# Process hospital stay dates for inpatient dataframes

**for** df **in** hospital\_dataframes**:**

process\_hospital\_stay**(**df**)**

data **=** inpatient\_train**.**groupby**(**'TimeInHptal'**).**count**().**BeneID

plt**.**title**(**"the dsitribution of inpatient's time in Hospital"**)**

plt**.**bar**(**x**=**data**.**index**,**height**=**data**,**color**=**'navy'**)**

plt**.**xticks**(**np**.**arange**(**0**,**12**,**1**))**

plt**.**show**()**

ClmProcedureCode **=** **[**'ClmProcedureCode\_1'**,** 'ClmProcedureCode\_2'**,** 'ClmProcedureCode\_3'**,**

'ClmProcedureCode\_4'**,** 'ClmProcedureCode\_5'**,** 'ClmProcedureCode\_6'**,]**

ClmDiagnosisCode **=** **[**'ClmDiagnosisCode\_1'**,**

'ClmDiagnosisCode\_2'**,** 'ClmDiagnosisCode\_3'**,** 'ClmDiagnosisCode\_4'**,**

'ClmDiagnosisCode\_5'**,** 'ClmDiagnosisCode\_6'**,** 'ClmDiagnosisCode\_7'**,**

'ClmDiagnosisCode\_8'**,** 'ClmDiagnosisCode\_9'**,** 'ClmDiagnosisCode\_10'**]**

# Fill NaN values for ClmDiagnosisCode columns in both DataFrames

inpatient\_train**[**ClmDiagnosisCode**]** **=** inpatient\_train**[**ClmDiagnosisCode**].**fillna**(**'-1'**)**

outpatient\_train**[**ClmDiagnosisCode**]** **=** outpatient\_train**[**ClmDiagnosisCode**].**fillna**(**'-1'**)**

# Function to count non '-1' diagnosis codes in each row

**def** count\_diagnoses**(**df**,** code\_list**,** index\_name**):**

# Apply a lambda function to count non '-1' codes row-wise

df**[**index\_name**]** **=** df**[**code\_list**].**apply**(lambda** row**:** **sum(**code **!=** '-1' **for** code **in** row**),** axis**=**1**)**

# Apply the counting function to both DataFrames

count\_diagnoses**(**inpatient\_train**,** ClmDiagnosisCode**,** 'DiagnosisCnt'**)**

count\_diagnoses**(**outpatient\_train**,** ClmDiagnosisCode**,** 'DiagnosisCnt'**)**

count\_diagnoses**(**inpatient\_train**,**ClmDiagnosisCode**,**'DiagnosisIndex'**)**

count\_diagnoses**(**outpatient\_train**,**ClmDiagnosisCode**,**'DiagnosisIndex'**)**

count\_diagnoses**(**inpatient\_train**,**ClmProcedureCode**,**'ProcedureIndex'**)**

count\_diagnoses**(**outpatient\_train**,**ClmProcedureCode**,**'ProcedureIndex'**)**

ClmDiagnosisCode**.**remove**(**'ClmDiagnosisCode\_1'**)**

inpatient\_train**.**head**()**

count\_diagnoses**(**inpatient\_train**,**ClmDiagnosisCode**,**'DiagnosisIndex'**)**

count\_diagnoses**(**inpatient\_train**,**ClmProcedureCode**,**'ProcedureIndex'**)**

count\_diagnoses**(**outpatient\_train**,**ClmDiagnosisCode**,**'DiagnosisIndex'**)**

inpatient\_train**.**groupby**(**'DiagnosisIndex'**).**count**()**

**def** func**(**pct**,** allvalues**):**

absolute **=** **int(**pct **/** 100. **\*** np**.sum(**allvalues**))**

**return** "{:.1f}% ({:d})"**.format(**pct**,** absolute**)**

# Prepare data for pie chart

plt**.**figure**(**figsize**=(**8**,** 6**),** dpi**=**100**)**

data **=** inpatient\_train**.**groupby**(**'DiagnosisIndex'**).**count**().**BeneID

# Create the pie chart

plt**.**pie**(**data**,**

labels**=**data**.**index**,**

autopct**=lambda** pct**:** func**(**pct**,** data**.**values**))**

# Set the title and display the plot

plt**.**title**(**'DiagnosisIndex in Inpatient Train Data'**)**

plt**.**show**()**

**def** func**(**pct**,** allvalues**):**

absolute **=** **int(**pct **/** 100. **\*** np**.sum(**allvalues**))**

**return** "{:.1f}% ({:d})"**.format(**pct**,** absolute**)**

# Prepare data for pie chart

plt**.**figure**(**figsize**=(**8**,** 6**),** dpi**=**100**)**

data **=** outpatient\_train**.**groupby**(**'DiagnosisIndex'**).**count**().**BeneID

# Create the pie chart

plt**.**pie**(**data**,**

labels**=**data**.**index**,**

autopct**=lambda** pct**:** func**(**pct**,** data**.**values**))**

# Set the title and display the plot

plt**.**title**(**'DiagnosisIndex in Outpatient Data'**)**

plt**.**show**()**

inpatient\_train**.**columns

**def** isSamePhysician**(**df**):**

# Use vectorized comparison to create the 'SamePhysician' column

df**[**'SamePhysician'**]** **=** **(**df**[**'AttendingPhysician'**]** **==** df**[**'OperatingPhysician'**]).**astype**(int)**

inpatient\_train**.**OtherPhysician**.**fillna**(**0**,**inplace**=True)**

outpatient\_train**.**OtherPhysician**.**fillna**(**0**,**inplace**=True)**

inpatient\_test**.**OtherPhysician**.**fillna**(**0**,**inplace**=True)**

outpatient\_test**.**OtherPhysician**.**fillna**(**0**,**inplace**=True)**

**for** df **in** **[**inpatient\_train**,**outpatient\_train**,**inpatient\_test**,**outpatient\_train**]:**

**for** col **in** **[**'AttendingPhysician'**,**'OperatingPhysician'**]:**

df**[**col**].**dropna**(**inplace**=True)**

isSamePhysician**(**inpatient\_train**)**

isSamePhysician**(**outpatient\_train**)**

isSamePhysician**(**inpatient\_test**)**

isSamePhysician**(**outpatient\_test**)**

beneficiary\_train**.**to\_csv**(**'bene\_train.csv' **,**sep**=**'\t'**,** encoding**=**'utf-8'**)**

# PROVIDER AND FRAUDS

label\_train

provider\_num **=** **len(**label\_train**[**'Provider'**].**unique**())**

provider\_num

label\_test

label\_train**[**'PotentialFraud'**].**replace**(**'No'**,**0**,**inplace**=True)**

label\_train**[**'PotentialFraud'**].**replace**(**'Yes'**,**1**,**inplace**=True)**

fraud **=** label\_train**[**label\_train**[**'PotentialFraud'**]==**1**]**

fraud**.**shape**[**0**]**

**import** matplotlib**.**pyplot **as** plt

**import** numpy **as** np

# Group by 'PotentialFraud' and count the number of providers

data **=** label\_train**.**groupby**(**'PotentialFraud'**).**count**()**

labels **=** **[**'Prov. without potential fraud'**,** 'Prov. with potential fraud'**]**

# Plotting

plt**.**figure**(**figsize**=(**8**,** 6**),** dpi**=**100**)**

fig **=** plt**.**bar**(**x**=**data**.**index**,** height**=**data**[**'Provider'**],** color**=[**'navy'**,** 'crimson'**])**

plt**.**title**(**'Distribution Between Potential Fraud and Normal Provider'**)**

plt**.**xticks**(**ticks**=**np**.**arange**(len(**labels**)),** labels**=**labels**)**

# Add percentage annotations on the bars

total\_providers **=** label\_train**.**shape**[**0**]**

**for** p **in** fig**.**patches**:**

height **=** p**.**get\_height**()**

percentage **=** **round((**100 **\*** height**)** **/** total\_providers**,** 2**)**

plt**.**annotate**(**f'{percentage}%'**,** **(**p**.**get\_x**()** **+** p**.**get\_width**()** **/** 2**,** height**),** ha**=**'center'**,** va**=**'bottom'**)**

plt**.**show**()**

# List of columns to drop in each DataFrame

columns\_to\_drop\_initial **=** **[**'ClaimStartDt'**,** 'ClaimEndDt'**]**

delete **=** ClmProcedureCode **+** ClmDiagnosisCode

# Drop specified columns from all DataFrames

**for** df **in** **[**inpatient\_train**,** inpatient\_test**,** outpatient\_test**,** outpatient\_train**]:**

# Drop the initial set of columns if they exist

**if** **set(**columns\_to\_drop\_initial**).**issubset**(**df**.**columns**):**

df**.**drop**(**columns**=**columns\_to\_drop\_initial**,** inplace**=True)**

# Drop the combined list of columns if they exist

**if** **set(**delete**).**issubset**(**df**.**columns**):**

df**.**drop**(**columns**=**delete**,** inplace**=True)**

common\_cols **=** **[**col **for** col **in** outpatient\_train**.**columns **if** col **in** inpatient\_train**.**columns**]**

common\_cols

inpatient\_train**[**"Admitted"**]** **=** 1

outpatient\_train**[**"Admitted"**]** **=** 0

inpatient\_test**[**"Admitted"**]** **=** 1

outpatient\_test**[**"Admitted"**]** **=** 0

# Check for non-numeric values

non\_numeric\_values **=** inpatient\_train**[~**inpatient\_train**[**'DeductibleAmtPaid'**].**apply**(lambda** x**:** **str(**x**).**isdigit**())]**

**print(**non\_numeric\_values**)**

# Check for NaNs

nan\_count **=** inpatient\_train**[**'DeductibleAmtPaid'**].**isna**().sum()**

**print(**f'Number of NaNs: {nan\_count}'**)**

# Replace non-numeric values with -9999

inpatient\_train**[**'DeductibleAmtPaid'**]** **=** pd**.**to\_numeric**(**inpatient\_train**[**'DeductibleAmtPaid'**],** errors**=**'coerce'**)**

# Fill NaNs with -9999

inpatient\_train**[**'DeductibleAmtPaid'**]** **=** inpatient\_train**[**'DeductibleAmtPaid'**].**fillna**(-**9999**)**

# Convert to integer

inpatient\_train**[**'DeductibleAmtPaid'**]** **=** inpatient\_train**[**'DeductibleAmtPaid'**].**astype**(int)**

# Verify conversion

**print(**inpatient\_train**[**'DeductibleAmtPaid'**].**dtype**)**

ip\_op\_train **=** pd**.**merge**(**left**=**inpatient\_train**,** right**=**outpatient\_train**,** how**=**'outer'**)**

ip\_op\_test **=** pd**.**merge**(**left**=**inpatient\_test**,** right**=**outpatient\_test**,** how**=**'outer'**)**

ip\_op\_train **=** pd**.**merge**(**left**=**ip\_op\_train**,** right**=**label\_train**,** on**=**'Provider'**,** how**=**'inner'**)**

ip\_op\_test **=** pd**.**merge**(**left**=**ip\_op\_test**,** right**=**label\_test**,** on**=**'Provider'**,** how**=**'inner'**)**

# Joining the IP\_OP dataset with the BENE data

train\_df **=** pd**.**merge**(**left**=**ip\_op\_train**,** right**=**beneficiary\_train**,** left\_on**=**'BeneID'**,** right\_on**=**'BeneID'**,**how**=**'inner'**)**

train\_df**.**shape

test\_df **=** pd**.**merge**(**left**=**ip\_op\_test**,** right**=**beneficiary\_test**,** left\_on**=**'BeneID'**,** right\_on**=**'BeneID'**,**how**=**'inner'**)**

test\_df**.**shape

outpatient\_test**.**shape**[**0**]** **+**inpatient\_test**.**BeneID**.**shape**[**0**]**

test\_df**.**to\_csv**(**'test.csv' **,**sep**=**'\t'**,** encoding**=**'utf-8'**)**

train\_df**.**to\_csv**(**'train.csv' **,**sep**=**'\t'**,** encoding**=**'utf-8'**)**

# MACHINE LEARNING

first\_train **=** train\_df**.**copy**()**

first\_train**.**columns

obj\_list **=** **[**'BeneID'**,** 'ClaimID'**,**'Provider'**,**'AttendingPhysician'**,** 'ClmAdmitDiagnosisCode'**,**'OperatingPhysician'**,** 'OtherPhysician'**,**'DiagnosisGroupCode'**,**'ClmDiagnosisCode\_1'**,**'SamePhysician'**]**

# Initialize the LabelEncoder

labelencoder **=** LabelEncoder**()**

# Loop through each column in the DataFrame

**for** col **in** first\_train**.**columns**:**

# Check if the column has mixed types

**if** first\_train**[**col**].**apply**(lambda** x**:** **isinstance(**x**,** **str)).any()** **and** first\_train**[**col**].**apply**(lambda** x**:** **isinstance(**x**,** **(int,** **float))).any():**

# Convert all values to strings

first\_train**[**col**]** **=** first\_train**[**col**].**astype**(str)**

# Now apply the LabelEncoder

first\_train**[**col**]** **=** labelencoder**.**fit\_transform**(**first\_train**[**col**])**

X **=** first\_train**.**drop**([**'PotentialFraud'**,**'DOB'**],**axis**=**1**)**

y **=** first\_train**[**'PotentialFraud'**]**

X**.**info**()**

X **=** X**.**fillna**(-**9999**)**

X\_train**,** X\_val**,** y\_train**,** y\_val **=** tts**(**X**,** y**,** test\_size**=**0.20**,** stratify**=**y**,** random\_state**=**42**)**

# Checking shape of each set

X\_train**.**shape**,** X\_val**.**shape**,** y\_train**.**shape**,** y\_val**.**shape

# Checking count of tgt labels in y\_train

y\_train**.**value\_counts**()**

X\_val**.**head**()**

# RANDOM FOREST

# First Model

#random forest model creation

rfc **=** RandomForestClassifier**(**n\_estimators**=**500**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**4**)**

rfc**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict **=** rfc**.**predict**(**X\_val**)**

fpr**,** tpr**,** thresholds **=** roc\_curve**(**y\_val**,** rfc**.**predict\_proba**(**X\_val**)[:,**1**])**

roc\_auc **=** auc**(**fpr**,** tpr**)**

plt**.**figure**()**

plt**.**plot**(**fpr**,** tpr**,** color**=**'darkorange'**,** lw**=**1**,** label**=**'ROC curve (AUC = %0.2f)' **%** roc\_auc**)**

**for** label **in** **range(**1**,**10**,**1**):**

plt**.**text**((**10**-**label**)/**10**,(**10**-**label**)/**10**,**thresholds**[**label**\***15**],**fontdict**={**'size'**:** 14**})**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** color**=**'navy'**,** lw**=**1**,** linestyle**=**'--'**)**

plt**.**xlim**([**0.0**,** 1.0**])**

plt**.**ylim**([**0.0**,** 1.05**])**

plt**.**xlabel**(**'False Positive Rate'**)**

plt**.**ylabel**(**'True Positive Rate'**)**

plt**.**title**(**'Receiver Operating Characteristic of First Model'**)**

plt**.**legend**(**loc**=**"lower right"**)**

plt**.**show**()**

# Building Evaluation Parameters

n\_outliers **=** **len(**fraud**)**

n\_errors **=** **(**y\_predict **!=** y\_val**).sum()**

**print(**"The model used is Random Forest classifier"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

MCC **=** matthews\_corrcoef**(**y\_val**,** y\_val**)**

**print(**"The Matthews correlation coefficient is{}"**.format(**MCC**))**

# Second Model

#random forest model creation

rfc2 **=** RandomForestClassifier**(**n\_estimators**=**500**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**10**)**

rfc2**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict2 **=** rfc2**.**predict**(**X\_val**)**

fpr**,** tpr**,** thresholds **=** roc\_curve**(**y\_val**,** rfc2**.**predict\_proba**(**X\_val**)[:,**1**])**

roc\_auc **=** auc**(**fpr**,** tpr**)**

plt**.**figure**()**

plt**.**plot**(**fpr**,** tpr**,** color**=**'darkorange'**,** lw**=**1**,** label**=**'ROC curve (AUC = %0.2f)' **%** roc\_auc**)**

**for** label **in** **range(**1**,**10**,**1**):**

plt**.**text**((**10**-**label**)/**10**,(**10**-**label**)/**10**,**thresholds**[**label**\***15**],**fontdict**={**'size'**:** 14**})**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** color**=**'navy'**,** lw**=**1**,** linestyle**=**'--'**)**

plt**.**xlim**([**0.0**,** 1.0**])**

plt**.**ylim**([**0.0**,** 1.05**])**

plt**.**xlabel**(**'False Positive Rate'**)**

plt**.**ylabel**(**'True Positive Rate'**)**

plt**.**title**(**'Receiver operating characteristic of Second Model'**)**

plt**.**legend**(**loc**=**"lower right"**)**

plt**.**show**()**

**print(**"The model used is Random Forest classifier with depth of 10"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict2**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict2**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict2**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict2**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

# Third Model

#random forest model creation

rfc3 **=** RandomForestClassifier**(**n\_estimators**=**500**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**15**)**

rfc3**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict3 **=** rfc3**.**predict**(**X\_val**)**

fpr**,** tpr**,** thresholds **=** roc\_curve**(**y\_val**,** rfc3**.**predict\_proba**(**X\_val**)[:,**1**])**

roc\_auc **=** auc**(**fpr**,** tpr**)**

plt**.**figure**()**

plt**.**plot**(**fpr**,** tpr**,** color**=**'darkorange'**,** lw**=**1**,** label**=**'ROC curve (AUC = %0.2f)' **%** roc\_auc**)**

**for** label **in** **range(**1**,**10**,**1**):**

plt**.**text**((**10**-**label**)/**10**,(**10**-**label**)/**10**,**thresholds**[**label**\***15**],**fontdict**={**'size'**:** 14**})**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** color**=**'navy'**,** lw**=**1**,** linestyle**=**'--'**)**

plt**.**xlim**([**0.0**,** 1.0**])**

plt**.**ylim**([**0.0**,** 1.05**])**

plt**.**xlabel**(**'False Positive Rate'**)**

plt**.**ylabel**(**'True Positive Rate'**)**

plt**.**title**(**'Receiver operating characteristic of Third Model'**)**

plt**.**legend**(**loc**=**"lower right"**)**

plt**.**show**()**

**print(**"The model used is Random Forest classifier with depth of 15"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict3**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict3**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict3**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict3**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

# Fourth Model

#random forest model creation

rfc4 **=** RandomForestClassifier**(**n\_estimators**=**500**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**25**)**

rfc4**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict4 **=** rfc4**.**predict**(**X\_val**)**

**print(**"The model used is Random Forest classifier with depth of 25"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict4**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict4**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict4**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict4**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

fpr**,** tpr**,** thresholds **=** roc\_curve**(**y\_val**,** rfc4**.**predict\_proba**(**X\_val**)[:,**1**])**

roc\_auc **=** auc**(**fpr**,** tpr**)**

plt**.**figure**()**

plt**.**plot**(**fpr**,** tpr**,** color**=**'darkorange'**,** lw**=**1**,** label**=**'ROC curve (AUC = %0.2f)' **%** roc\_auc**)**

**for** label **in** **range(**1**,**10**,**1**):**

plt**.**text**((**10**-**label**)/**10**,(**10**-**label**)/**10**,**thresholds**[**label**\***15**],**fontdict**={**'size'**:** 14**})**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** color**=**'navy'**,** lw**=**1**,** linestyle**=**'--'**)**

plt**.**xlim**([**0.0**,** 1.0**])**

plt**.**ylim**([**0.0**,** 1.05**])**

plt**.**xlabel**(**'False Positive Rate'**)**

plt**.**ylabel**(**'True Positive Rate'**)**

plt**.**title**(**'Receiver operating characteristic of Fourth Model'**)**

plt**.**legend**(**loc**=**"lower right"**)**

plt**.**show**()**

# Fifth Model

#random forest model creation

rfc5 **=** RandomForestClassifier**(**n\_estimators**=**1000**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**25**)**

rfc5**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict5 **=** rfc5**.**predict**(**X\_val**)**

**print(**"The model used is Random Forest classifier with depth of 25 and estimators of 1000"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict5**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict5**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict5**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict5**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

fpr**,** tpr**,** thresholds **=** roc\_curve**(**y\_val**,** rfc5**.**predict\_proba**(**X\_val**)[:,**1**])**

roc\_auc **=** auc**(**fpr**,** tpr**)**

plt**.**figure**()**

plt**.**plot**(**fpr**,** tpr**,** color**=**'darkorange'**,** lw**=**1**,** label**=**'ROC curve (AUC = %0.2f)' **%** roc\_auc**)**

**for** label **in** **range(**1**,**10**,**1**):**

plt**.**text**((**10**-**label**)/**10**,(**10**-**label**)/**10**,**thresholds**[**label**\***15**],**fontdict**={**'size'**:** 14**})**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** color**=**'navy'**,** lw**=**1**,** linestyle**=**'--'**)**

plt**.**xlim**([**0.0**,** 1.0**])**

plt**.**ylim**([**0.0**,** 1.05**])**

plt**.**xlabel**(**'False Positive Rate'**)**

plt**.**ylabel**(**'True Positive Rate'**)**

plt**.**title**(**'Receiver operating characteristic of Fifth Model'**)**

plt**.**legend**(**loc**=**"lower right"**)**

plt**.**show**()**

# Sixth Model

#random forest model creation

rfc6 **=** RandomForestClassifier**(**n\_estimators**=**500**,**class\_weight**=**'balanced'**,**random\_state**=**123**,**max\_depth**=**25**,**oob\_score**=True)**

rfc6**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict6 **=** rfc6**.**predict**(**X\_val**)**

**print(**"The model used is Random Forest classifier with depth of 25 and estimators of 1000"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict6**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict6**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict6**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict6**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

cnt**=**0

**for** model **in** **[**rfc**,**rfc2**,**rfc4**,**rfc4**,**rfc5**]:**

cnt**+=**1

**print(**"model"**,**cnt**)**

#print("used estimators:",model.estimators\_[-1])

**print(**"feature\_importances\_"**,**model**.**feature\_importances\_**)**

**print(**"num of feature seen during fitting :"**,**model**.**n\_features\_in\_**,**"\n"**)**

rfc5**.**decision\_path**(**X\_val**)**

# SUPPORT VECTOR MACHINE

svm\_model **=** svm**.**SVC**()**

svm\_model**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict\_svm **=** svm\_model**.**predict**(**X\_val**)**

**print(**"The model used is SVM (Support Vector Machines)"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict\_svm**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict\_svm**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict\_svm**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict\_svm**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

# K NEIGHBORS CLASSIFIER

knn\_model **=** KNeighborsClassifier**(**n\_neighbors **=** 3**)**

knn\_model**.**fit**(**X\_train**,** y\_train**)**

#predictions

y\_predict\_knn **=** knn\_model**.**predict**(**X\_val**)**

**print(**"The model used is SVM (Support Vector Machines)"**)**

acc **=** accuracy\_score**(**y\_val**,** y\_predict\_knn**)**

**print(**"The accuracy is {}"**.format(**acc**))**

prec **=** precision\_score**(**y\_val**,** y\_predict\_knn**)**

**print(**"The precision is {}"**.format(**prec**))**

rec **=** recall\_score**(**y\_val**,**y\_predict\_knn**)**

**print(**"The recall is {}"**.format(**rec**))**

f1 **=** f1\_score**(**y\_val**,** y\_predict\_knn**)**

**print(**"The F1-Score is {}"**.format(**f1**))**

# Appendix B. Miscellaneous

**Live Links to References**

<https://saspublishers.com/media/articles/SJET_119_191-200_FT.pdf>

<https://www.researchgate.net/publication/377953277_Advancing_Credit_Card_Fraud_Detection_A_Review_of_Machine_Learning_Algorithms_and_the_Power_of_Light_Gradient_Boosting>

<https://cdn.aaai.org/ocs/17617/17617-77660-1-PB.pdf>

<https://onlinelibrary.wiley.com/doi/am-pdf/10.1002/asmb.2633>

<https://ieeexplore.ieee.org/abstract/document/8260744>

<https://www.irjet.net/archives/V9/i7/IRJET-V9I7120.pdf>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8734935/>

<https://www.researchgate.net/profile/Soumit-Roy-3/publication/375957548_PRIVACY_PREVENTION_OF_HEALTH_CARE_DATA_USING_AI/links/6564e0b8ce88b87031197ecc/PRIVACY-PREVENTION-OF-HEALTH-CARE-DATA-USING-AI.pdf>

<https://www.mdpi.com/2076-3417/10/15/5144>

<https://dergipark.org.tr/en/pub/sjmakeu/issue/74842/1223234>

**Key Formulas and Concepts in Random Forest**

1. **Gini Impurity for Classification**:

* The Gini Impurity is used to measure how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset.
* **Formula**:

Where:

* D is the dataset
* C is the number of classes
* Pi is the probability of a randomly selected element being classified into class i.

1. **Entropy for Classification**:

* Entropy is another metric for measuring the impurity or uncertainty in the data. It is often used in the context of information gain.
* **Formula**:

Where:

* D is the dataset
* C is the number of classes
* Pi is the probability of a randomly selected element being classified into class i.

1. **Information Gain for Feature Selection**:

* Information Gain measures the reduction in entropy after a dataset is split on an attribute. It helps to decide which feature to split on at each step in building the tree.
* **Formula**:

Where:

* IG (D, A) is the information gain of attribute A
* Values(A) is the set of all possible values of attribute A
* Dv is the subset of D for which attribute A has value v

1. **Mean Squared Error (MSE) for Regression**:

* In regression tasks, Mean Squared Error is used to measure the quality of a tree split.
* **Formula**:

Where:

* n is the number of observations
* is the actual value
* ​ is the predicted value.

1. **Bootstrap Sampling**:

* Bootstrap sampling is a method of sampling with replacement. Each tree in the forest is trained on a different bootstrapped sample of the data.
* If the dataset has n samples, each bootstrapped sample also has n samples, but some samples may repeat while others may be omitted.

1. **Out-of-Bag Error (OOB Error)**:

* The OOB error is an internal estimate of the generalization error for a random forest, based on the samples that are not included in the bootstrap sample (i.e., the "out-of-bag" samples).
* **Formula**:

where

* I is an indicator function that is 1 if (the true value is not equal to the predicator OOB value ) and 0 otherwise.
* is the predicated class based on the aggregation of all trees that did not use the i-th sample in their training.

1. **Final Prediction Aggregation**:

* For classification, the final prediction is based on majority voting:
* For regression, the final prediction is the average of all the predictions:

​

where:

* T is the total number of trees.
* is the prediction of the t-th tree.