INSURANCE FRAUD DETECTION USING ML

ARUL ARASU N (Roll No: 20Z306)

DHANASEELAN V (Roll No: 20Z313)

DINESH BAABU R (Roll No: 20Z315)

LOKAJIT G (Roll No: 20Z328)

SUDARSHAN S (Roll No: 20Z350)

19Z720 – PROJECT WORK

BACHELOR OF ENGINEERING

**Branch:** COMPUTER SCIENCE AND ENGINEERING



NOVEMBER 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE - 641 004

INSURANCE FRAUD DETECTION USING ML

Bonafide record of work done by

Arul Arasu N (20Z306)

Dhanaseelan V (20Z313)

Dinesh Baabu R (20Z315)

Lokajit G (20Z328)

Sudarshan S (20Z350)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

Of Anna University

NOVEMBER 2023

…………………….. ……………………….

Dr.Santhi V Dr. Sudha Sadhasivam G

Faculty Guide Head of the Department

CERTIFICATE

Certified that this report titled “**Insurance Fraud Detection using ML**”, for the Project work I (19Z720) is a bonafide work of

Arul Arasu N (20Z306)

Dhanaseelan V (20Z313)

Dinesh Baabu R (20Z315)

Lokajit G (20Z328)

Sudarshan S (20Z350)

who have carried out the work under my supervision for the partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Coimbatore Dr.Santhi V

Date: Designation: Professor

Department of Computer Science and Engineering

PSG College of Technology

Coimbatore - 641004

COUNTERSIGNED

HEAD

Department of Computer Science and Engineering

PSG College Of Technology - 641004

# ACKNOWLEDGEMENT

We would like to thank the management of **PSG College of Technology** for providing us the infrastructure and helping us envision new ideas and make them work successfully. We would also like to express our heartfelt gratitude to our **Principal Dr.K.Prakasan** for bestowing us with this valuable opportunity to work in our area of interest.

We also extend our sincere thanks to our **Dr.G.Sudha Sadasivam, Head and Professor, Department of Computer Science and Engineering** for constantly supporting us to develop our ideas and present our project to the faculty committee as a part of our partial fulfilment of the requirements leading to the awarding BE degree.

We take immense pleasure in thanking our project guide **Dr. V. Shanthi, Professor and Programmer Coordinator (G2), Department of Computer Science and Engineering** for being a pillar of support, without whose guidance, unparalleled cooperation and constructive criticisms, this project wouldn’t have been fruitful.

We would also like to thank our tutor **Dr. Vijayalakshmi S, Assistant Professor (Sl. Gr.), Department of Computer Science and Engineering**, for their encouragement and constant vigilance during the course of this project.

A sincere thanks to all the **Panel Members** for reviewing our project and all the **Faculty Members and Staffs** of the department for offering us the required support during the course of the project.

CONTENTS

CHAPTER Page No.

# SYNOPSIS

The project's primary goal is to create a healthcare insurance policy that enforces specific patient precautions aligned with their policy terms. To achieve this, the project utilizes a dataset of insurance claims from admitted patients in healthcare or hospital settings. Machine learning (ML) models, powered by ML algorithms, are employed to analyze patterns within the data and identify potential instances of fraudulent behavior. By doing so, the project aims to calculate the percentage of fraudulent claims and distinguish them from legitimate ones. To ensure the security and transparency of this process, the legitimate dataset is stored on a blockchain network, leveraging blockchain technology.

The prevalence of fraudulent activities in the insurance industry necessitates innovative solutions. This proposal combines cutting-edge blockchain technology with advanced machine learning techniques to address the issue effectively. According to a 2015 report, healthcare fraud resulted in a significant loss of £303.8 million, with potentially hidden losses estimated to range from £3.73 to £5.74 billion. The approach involves analyzing complex historical feedback data using sophisticated machine learning algorithms. These algorithms possess adaptive capabilities, continuously improving their ability to detect fraudulent patterns by learning from past cases. This approach serves as a powerful tool for early fraud detection and represents a substantial investment in terms of time and resources. The integration of blockchain technology further bolsters the system by establishing a secure and transparent record of virtual insurance transactions. This not only enhances trust but also ensures clear accountability in the process. The ultimate aim is to reshape the insurance landscape, combat fraud, enhance operational efficiency, inspire trust, and provide security. These efforts can expedite claims processing, reduce financial losses, promote a culture of integrity, and benefit both insurers and policyholders as they collectively prepare for an uncertain future.

In summary, the project seeks to revolutionize healthcare insurance by utilizing machine learning. Through the analysis of historical insurance claims, ML algorithms can detect fraudulent behavior. This innovative approach aims to combat widespread insurance fraud, improve operational efficiency, build trust, and ensure security, ultimately benefiting both insurers and policyholders in an ever-evolving landscape.

# LIST OF FIGURES

| Fig.No | Description |  |
| --- | --- | --- |
| Fig 4 | System Architecture |  |
| Fig 5.1 | Distribute of Class labels (Potential Fraud) |  |
| Fig 5.2 | Distribute of Data in the ‘Gender’ Feature |  |
| Fig 5.3 | Distribute of Data in the ‘NoOfMonth\_PartACov’ Feature |  |
| Fig 5.4 | Histogram /PDF of the date in ‘IpAnnualReimbursementAmt’ Feature |  |
| Fig 5.5 | Box Plot of the data in the ‘OPAnnualReimbursementAmt’ Feature |  |
| Fig 5.6 | Box Plot of the data in the ‘ClaimSettlement Delay’ Feature |  |
| Fig 5.7 | Box Plot of the data in the ‘Treatment Duration’ Feature |  |
| Fig 5.8 | Distribution of data(Top 30) in the ‘ClmProcedure Code\_1’ Feature |  |
| Fig 5.9 | Distribution of data(Top 30) in the ‘ClmProcedure Code\_5’ Feature |  |
| Fig 5.10 | Distribution of data(Top 30) in the ‘Attending Physical ’ Feature |  |
| Fig 6.1 | Correlation plot of the feature |  |
| Fig 6.2 | Correlation plot of the feature with Class Label |  |
| Fig 7.1 | Directory Structure |  |
| Fig 7.2 | Decision Tree Classifier Model |  |
| Fig 7.3 | A)Confusion Matrix B)Precision Matrix C)Recall Matrix |  |
| Fig 7.4 | Performance Matrix |  |
| Fig 7.5 | ROX C And AUC |  |
| Fig 7.5.1 | Pictorial Representation of Random Forest |  |

**CHAPTER 1**

# INTRODUCTION

In the era of digital world most of the people play some handsome tricks that plays the role the society, organization and the other some sustainable services that could illegally may generate fraud data on foster research of health care services, insurance policy services, bank services and some sustain factors of services that may hazardously incur the fraudulent activities that make economic growth anonymously. These factors affecting the other intellect services of a patient’s information may implant the technology of Machine Learning.

Machine learning may detect fraudulent datasets of patient’s insurance claims admitted in a Health care or Hospital that analyzes the behavior patterns and identifies data that need to predict its statistics occurring in the dataset. This could ensure a secure phase from unwanted data that purifies the data to validate on the health care or hospital and insurance policy as well.

This project work collects insurance claim records and performs necessary data processing techniques to make it fitting to the machine learning models. Different machine learning models are trained with the data and evaluated with various performing metrics to ensure to come up with the right model. The features contribution towards the learning is identified through model interpretability. The details of the project work would be discussed in the forthcoming chapters.

**CHAPTER 2**

# LITERATURE SURVEY

1. **“** A novel fraud detection and prevention method for healthcare claim processing using machine learning and blockchain technology Anokye Acheampong Amponsah ∗ , Adebayo Felix Adekoya, Benjamin Asubam Weyori “ <https://pdf.sciencedirectassets.com/780449>

This research paper provides a detailed explanation of a new approach to healthcare fraud detection and prevention using machine learning and blockchain technology. The authors explain that healthcare fraud is a serious problem in many countries, causing significant economic losses to insurance companies and government agencies The proposed method uses decision tree classification algorithms to analyze healthcare information and they are used to identify potentially misleading information and to include fraudulent information.

The authors also discuss the potential benefits of using blockchain technology in healthcare claims. They explain that blockchain provides a secure and transparent way to store and share health information, which can help prevent fraud and ensure that the rightful beneficiaries receive fair compensation. The use of blockchain technology can help reduce administrative costs and improve health care coverage.

However, the authors also acknowledge that there are some challenges in applying this approach to real-world health care settings. For example, there may be concerns about data privacy and security, as well as resistance from healthcare providers and insurance companies accustomed to traditional ways of handling data The authors suggest that these challenges can be addressed through policy with care and collaboration between stakeholders.

1. Sathya, M., & Balakumar , B. . (2022). Insurance Fraud Detection Using Novel Machine Learning Technique. International Journal of Intelligent Systems and Applications in Engineering, 10(3), 374–381. <https://ijisae.org/index.php/IJISAE/article/view/2178>

This paper discusses the problem of fraudulent claims in the insurance industry and the limitations of traditional fraud detection techniques. The paper proposes a novel approach for detecting insurance fraud using a hybrid machine learning classifier called eRFSVM, which combines Random Forest and Support Vector Machine algorithms. The proposed approach also utilizes block chain technology for secure information sharing among insurance agencies.

The paper evaluates the proposed approach using a confusion matrix and various classification metrics. The results show that the eRFSVM classifier outperforms traditional fraud detection techniques in terms of accuracy and efficiency. The proposed approach shows remarkable performance in evaluating the authenticity of customer claims, with an exceptional accuracy of 97.176%. Additionally, the value of specificity and sensitivity is 96.158% and 96.634% respectively.

The paper highlights the potential of machine learning techniques for fraud detection in the insurance industry and the importance of secure information sharing among insurance agencies. The proposed approach can help insurance companies to detect fraudulent activities effectively and reduce unwarranted expenses. The paper also discusses of the proposed approach for detecting insurance fraud include the requirement for a large amount of data to train the machine learning model effectively, reliance on the accuracy of the data provided by insurance agencies, potential difficulty in detecting new types of fraud, and the need for significant computational resources to process large amounts of data.

1. N. R. Bhamidipati et al., "ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing," 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 2021, pp. 55-64, doi: 10.1109/Blockchain53845.2021.00019. <https://par.nsf.gov/servlets/purl/10315023>

This paper discusses the benefits of using blockchain technology for insurance claims processing and introduces ClaimChain, a secure blockchain platform designed to replace traditional NICB/ISO database architecture used in the auto-insurance industry. The paper is organized into several sections, including a discussion of related works, an overview of the ClaimChain approach, and a description of the platform's security features.

The authors explain how blockchain technology offers transparency and auditability, enabling distributed trust among participating peers, and how smart contracts can reduce operation and maintenance costs while improving processing time. They also describe how ClaimChain uses machine learning and NICB-identified red flags to detect and prevent fraud, and how the platform's threat modelling approach helps to improve security at both the infrastructure and application levels.

The paper includes a detailed analysis of a dataset of insurance claims hosted on the ClaimChain testbed, which reveals that the majority of fraudulent claims identified have no police report or witnesses. The authors conclude by highlighting the potential benefits of ClaimChain for the insurance industry and outlining future research directions. Overall, the PDF file provides a comprehensive overview of the benefits and challenges of using blockchain technology for insurance claims processing and introduces a promising new platform for improving the efficiency and security of this process.

1. R. Roy and K. T. George, "Detecting insurance claims fraud using machine learning techniques," 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-6, doi: 10.1109/ICCPCT.2017.8074258. <https://ieeexplore.ieee.org/document/8074258>

This paper focuses on detecting auto/vehicle insurance fraud using machine learning techniques. The authors focus on creating a set of rules and anomalies for creating raw data, which is dependent on a set of attributes. They then compare the performance of decision trees, random forests, and Naïve Bayes in detecting insurance fraud using a confusion matrix. The methodology adopted involves dividing the data into training and testing sets, and comparing the accuracy, precision, and recall of each method. The authors found that decision trees and random forests outperformed Naïve Bayes in detecting insurance fraud. The evaluation of the methodology involved comparing the performance of each method using a confusion matrix, and calculating accuracy, precision, and recall.

1. Urunkar, Abhijeet & Khot, Amruta & Bhat, Rashmi & Mudegol, Nandini. (2022). Fraud Detection and Analysis for Insurance Claims using Machine Learning. 406-411. 10.1109/SPICES52834.2022.9774071.

<https://www.researchgate.net/publication/360646224_Fraud_Detection_and_Analysis_for_Insurance_Claim_using_Machine_Learning>

The traditional method for detecting fraud relies on manual intervention and heuristics, which can be limited in their ability to detect complex fraud schemes. The authors propose a machine learning approach that can analyze large amounts of data and identify patterns indicative of fraud. They use a combination of supervised and unsupervised learning algorithms, including decision trees, random forests, and clustering. The authors evaluate their approach using a dataset of insurance claims and compare their results to those of traditional fraud detection methods. They find that their machine learning approach outperforms traditional methods in terms of accuracy and efficiency. However, they also note that there are challenges to implementing these techniques in real-world insurance settings, such as data privacy concerns and the need for ongoing model maintenance.The authors suggest that it may not be practical to define optimum algorithmic techniques or use feature engineering processes for higher performance due to the inherent characteristics of various datasets. Instead, they propose that the models be used for specific business contexts and user priorities, allowing loss management units to focus on new fraud situations and ensure that the models are adapting to detect.

1. Inayatulloh, Siti Elda Hiererra, Prasetya Cahaya S, Rozil Toyob, Nico Djundharto Djajasinga , Sawqi Saad El Hasan, Rofiq Noorman Haryadi, Rivaldhy N. Muhammad,“ Blockchain technology of fraud Detection and Risk Prevention in Insurance Industry “. Proceedings of the 3rd South American International Industrial Engineering and Operations Management Conference, Asuncion, Paraguay, July 19-21, 2022. [47.docx (googleusercontent.com)](https://doc-10-8g-apps-viewer.googleusercontent.com/viewer/secure/pdf/77gfp996nr6b4qqh2qgl1g8e4t0lv9ej/2hvusaf9arr9domf1324i9sh39rgtnn0/1692286725000/gmail/06715741469926710199/ACFrOgBrcrsOqvV5LN7lBDorohY40T4RHb6Pmfi68FHikCPu2-EMrPf6mT8SWdxhBckrhWlX9brmAXk1f64M_NUAoHF405fIy0X4JjtAQK7WYAurqidDhI5O2iu2SBw=?print=true&nonce=2d7vgp0r0fbdq&user=06715741469926710199&hash=kupj6g9lt95ouaek0nsggtu7sgf38bmt)

Implementing Insurance Fraud Claim that customers register the insurance as a smart contract between customer and insurance participant that is stored as a new block/ledger broadcasting into the blockchain network . Validating process will be taken by each participant/node in the blockchain network which takes the advantage for difficulty in manipulation . After the validation from all blockchains network a new block will be part of the blockchain network and submission of the customer as an Insurance Company Customer is completed , the customer can access the approval via Smartphone or PC . The Submission claim from customer creates a new block as form of a ledger and broadcasted to the blockchain network . Each participant/node in the blockchain network will validate the claim submission from the customer and the process of potential fraud when submitting a claim will be lost. After the successful validation process the customer gets the result via Smartphone or PC finding out the submitted claims aren’t fake that revels in their application.

1. Sun C., Li Q., Li H., Shi Y., Zhang S., Guo W., “Patient Cluster Divergence Based Healthcare Insurance Fraudster Detection”, IEEE Access, Vol. 7, pp. 14162–14170, 2019. <https://ieeexplore.ieee.org/document/8576507>

Sun et al presented a novel approach for detecting frauds, called Patient Cluster Divergence-based Healthcare Insurance Fraudster Detection (PCDHIFD) in presence of camouflage responses. For the experimental purpose, the health care dataset was chosen and the dataset consisted of around 40M admission records of 10000 patients of the previous five years. The proposed technique worked in 3 steps for three basic records: Life history of patients, diagnosis record, and medical practitioners attended. Steps were in this sequence: first of all, a patient graph was constructed based on most similar info for the patient level hospital admission. Then a clustering-based graph algorithm was used for finding the peak and real meaning for individual clusters. Lastly, the difference in the patient cluster was found and the probability of fraud for each patient was calculated.

**CHAPTER 3**

# SYSTEM REQUIREMENTS

This chapter focuses on the hardware and software requirements essential to develop, train, test, and implement the system and its modules. It also discusses the dataset used, along with the feasibility of the system.

## 3.1 HARDWARE REQUIREMENTS

The hardware requirements mentioned are the minimum hardware settings required to create and use the proposed project in a computer system.

* **Processor (CPU)**: Intel Core i5 or higher processor
* **RAM**: 4 GB OR 8GB RAM , even more beneficial performance
* **Storage**: 500 MB or more space disk
* Graphics Card (GPU): NVIDIA GPU
* Operating System: Windows OS

## 3.2 SOFTWARE REQUIREMENTS

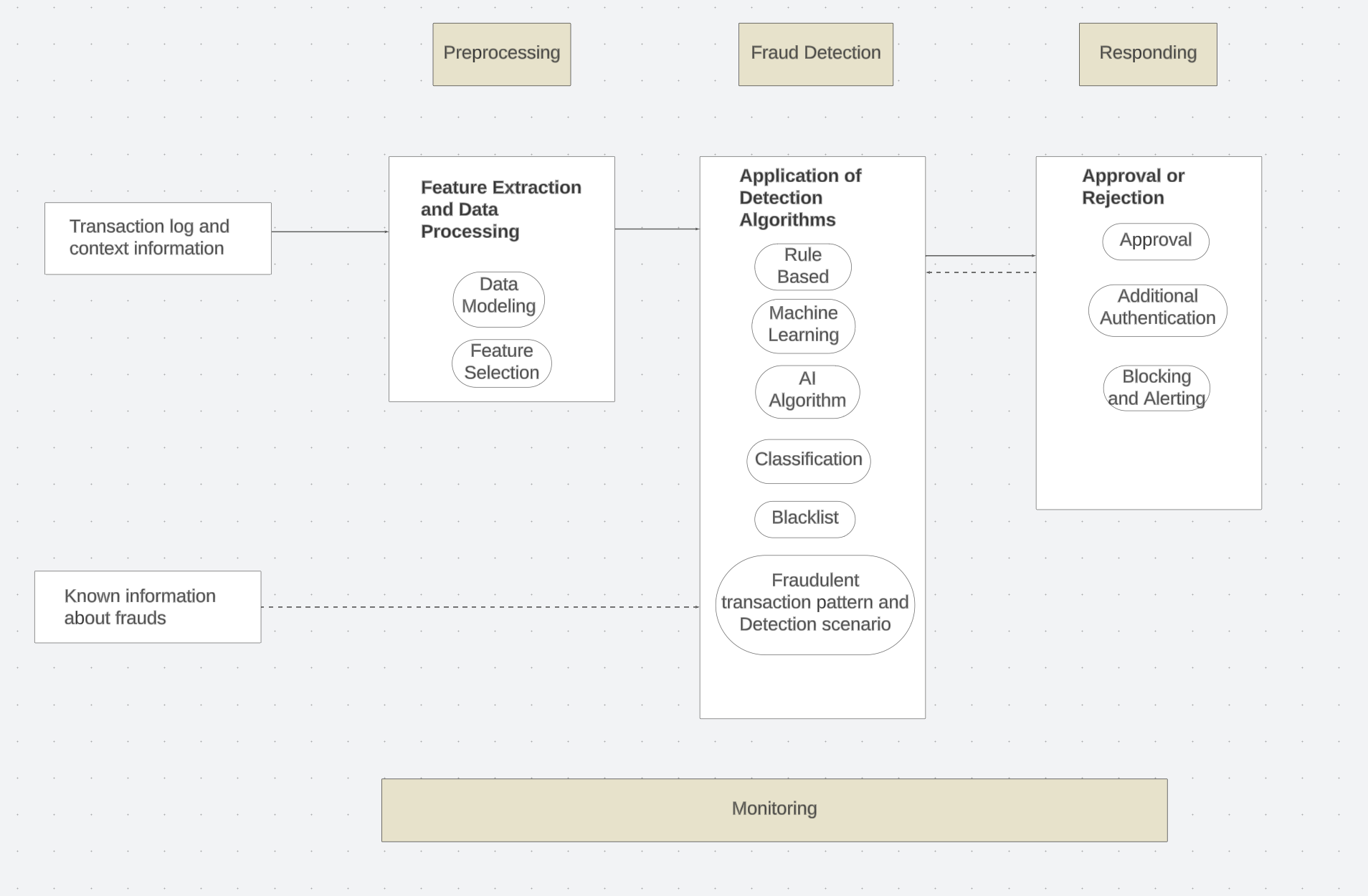
* **GOOGLE COLAB:** Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.
* **VS CODE:** A streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

**CHAPTER 4**

# SYSTEM DESIGN AND IMPLEMENTATION

## 4.1 SYSTEM ARCHITECTURE

The system design plots the sequence of phases in building a model to classify the insurance claims with optimal performance. Initially the data is fetched from the data store as individual records in an uninterpretable way for the model. The data must be pre-processed before being injected to the model. This phase includes data cleaning, feature engineering by deriving necessary attributes from existing ones and feature selection. The next phase is the building of models using different machine learning algorithms with the training data. The trained model is able to classify the unsupervised data as fraudulent or legit. The model shows the result of the legitimacy of the input to the user which is considered as the responding phase. The model’s decision is supported by its confidence score which should also be considered.

**Fig 4.1 - System architecture**

**CHAPTER - 5**

# DATA PREPARATION

## 5.1 DATASET INTRODUCTION

This project uses eight data files out of which 4 are for training purposes and 4 are used in testing the model. There are 4 types of data where each type has one file for training and testing. The types of data file are discussed below:

A) Inpatient Data

This data provides insights about the claims filed for those patients who are admitted in the hospitals. It also provides additional details like their admission and discharge dates and admit diagnosis code.

B) Outpatient Data

This data provides details about the claims filed for those patients who visit hospitals and not admitted in it.

C) Beneficiary Details Data

This data contains beneficiary KYC details like health conditions, region they belong to etc.

D) Provider Data

This file contains Provider ID. In the training file, the Provider ID is mapped with target value - Potential Fraud, whereas in the testing file only the Provider ID is given.

## 5.2 EXPLORATORY DATA ANALYSIS

### 5.2.1 Provider Dataset

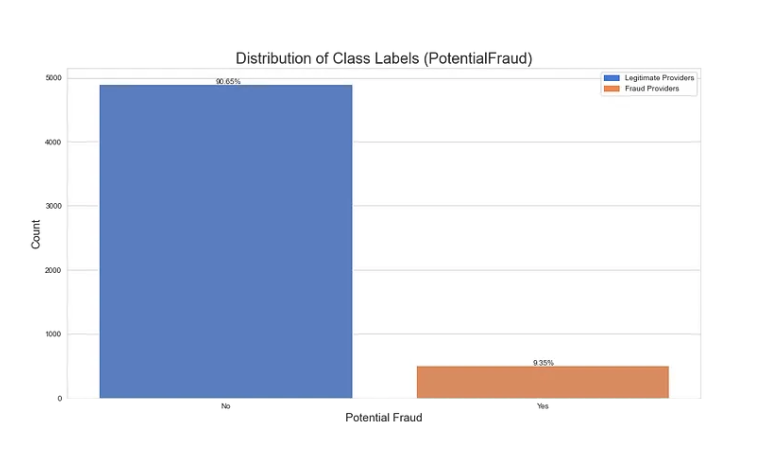


Fig 5.1 – Distribution of Class Labels

Fig 5.1 shows the distribution of class labels in the given dataset and it indicates that dataset is hugely imbalanced when classified based on the training data.

### 5.2.2 Beneficiary Dataset

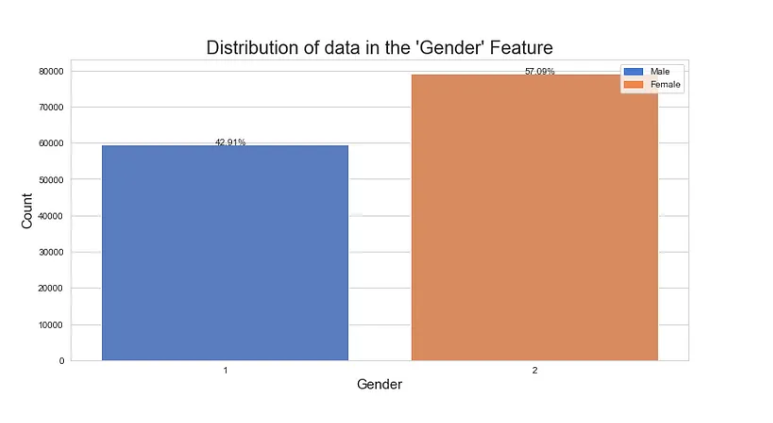


Fig 5.2 – Distribution of data in the ‘Gender’ Feature

Fig 5.2 shows the distribution of data in the ‘Gender’ Feature. It shows 57.09% of the beneficiaries are females and 42.92% are males. As opposed to the previous graph, this one is close to being perfectly balanced.

### 5.2.3 In-patient and Out-patient Dataset

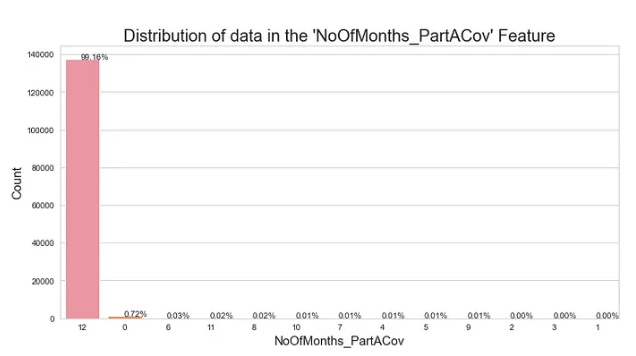


Fig 5.3 – Distribution of data in the ‘Number of Months in Part A Coverage’

Fig 5.3. shows the distribution of data in Number of Months in Part A Coverage. Both Part A and Part B coverages mostly have values of 12 corresponding to the duration of the plan. Since the other values are negligible, this feature can be neglected.

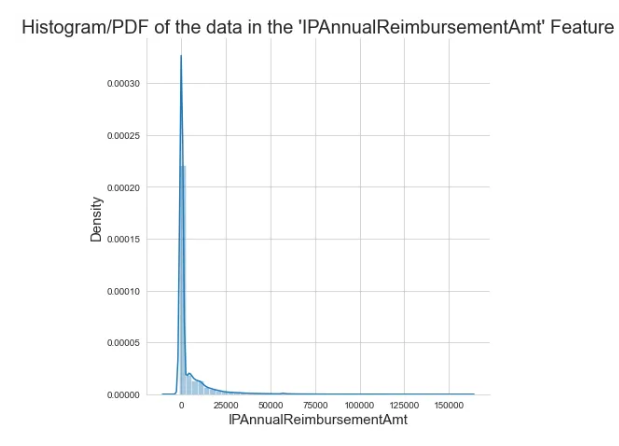


Fig 5.4 – Histogram of the data in the IP Annual reimbursement Amount

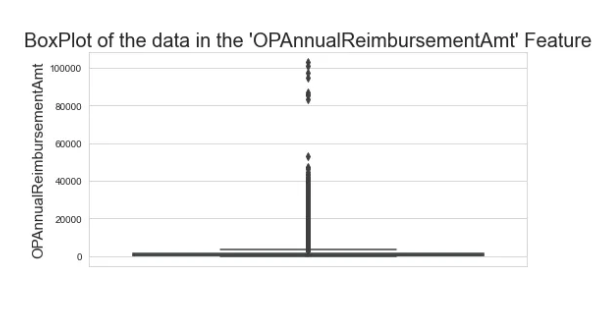


Fig 5.5 – Histogram of the data in the OP Annual reimbursement Amount

For Inpatient (Fig 5.4) and Outpatient (Fig 5.5) Annual Reimbursement Amount feature ranges mostly between 0 to 5000 and the Inpatient and Outpatient Annual Deductible Amount feature ranges between 0 to 2000. This means that the rest of the plotted points are outliers which shouldn’t be removed as they may be potential frauds.

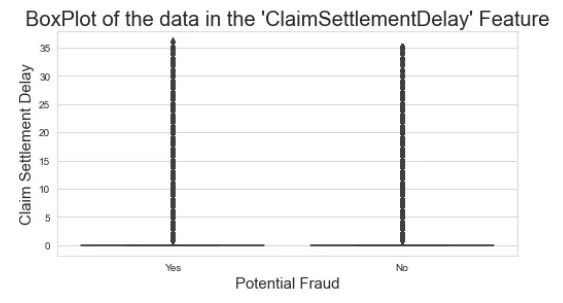


Fig 5.6 – Box Plot of Claim Settlement Delay

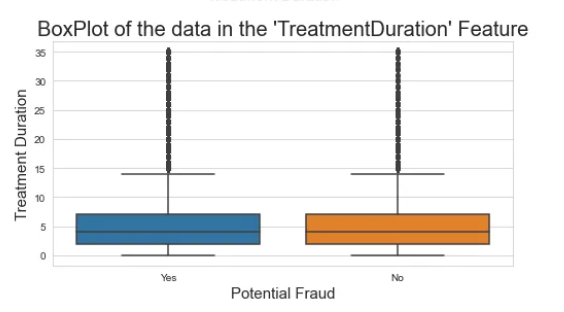


Fig 5.7 – Box Plot of Treatment Duration

Claim Settlement Delay (Fig 5.6) and Treatment Duration (Fig 5.7) feature are similar but the Treatment Duration feature is present only in the Inpatient dataset which can be removed in case of merges between IP and OP datasets.



Fig 5.8 (a) Distribution of top 50 categories of the ‘ClmAdmitDiagnosisCode’

(b) Distribution of bottom 50 categories of the ‘ClmAdmitDiagnosisCode’

From the Fig 5.8 it is evident that,

* The top 50 categories of the ‘ClmAdmitDiagnosisCode’ feature are present for both positive (fraud) and negative (non-fraud) cases.
* The ‘ClmAdmitDiagnosisCode’ feature are not present for both the classes but are present for either of them. For example, category ‘V452’ is present only once for fraud cases and category ‘37883’ is present only once for the non-fraud case.
* The top 5 Claim Admit Diagnosis Codes are ‘V7612’, ‘42731’, ‘78605’, ‘4019’ and ‘25000’.

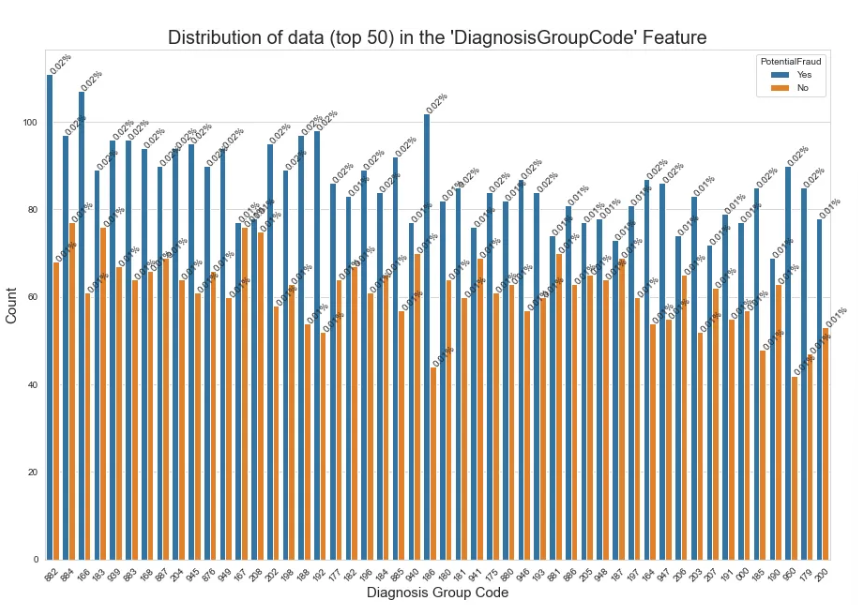


Fig 5.9(a) – Distribution of data (top 50) in the Diagnosis Group Code

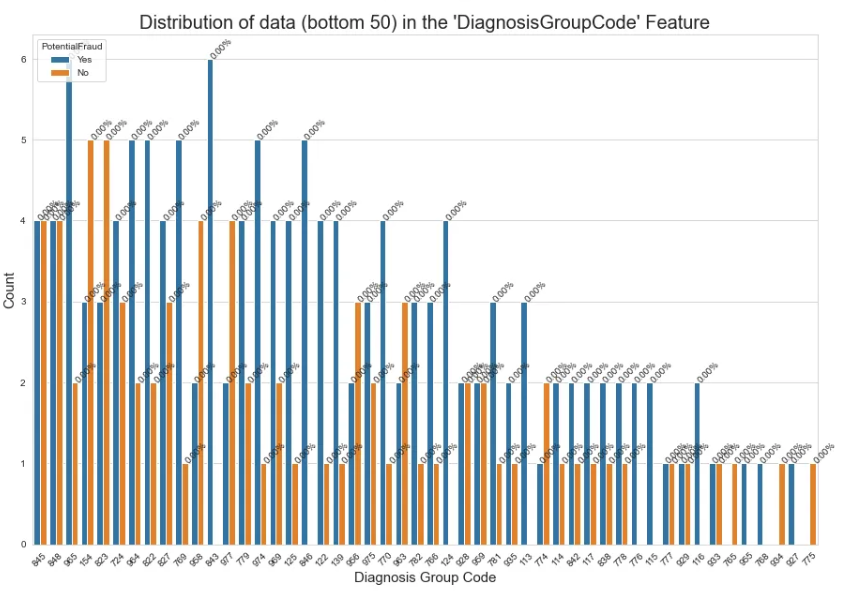


Fig 5.9(b) – Distribution of data (bottom 50) in the Diagnosis Group Code

The points to be inferred from Fig 5.9 are,

* From the top 50 categories of the ‘DiagnosisGroupCode’ feature, we cannot deduce much information.
* From the bottom 50 categories of the ‘DiagnosisGroupCode’ feature, we can see that there are few codes that occur only once (Count = 1). In the ‘ClmAdmitDiagnosisCode’ feature, there are comparatively lot of codes that had only one occurrence.

## 5.3 DATA PREPROCESSING

### 5.3.1 Data Cleanup

* Renamed the 'County' column to 'Country' for consistency.
* Standardized binary columns ('Renal Disease Indicator' and 'Chronic Condition') to have values of 0 or 1.
* Replaced missing values in 'DeductibleAmtPaid' with 0.
* Dropped columns with all null values.
* Encoded 'PotentialFraud' labels as 1 for 'Yes' and 0 for 'No'.

### 5.3.2 Data Processing

* Converted date-related columns to DateTime format.
* Calculated 'ClaimSettlementDelay' as the difference between 'ClaimEndDt' and 'ClaimStartDt'.
* Calculated 'TreatmentDuration' as the difference between 'DischargeDt' and 'AdmissionDt' for inpatient records.

### 5.3.3 Feature Engineering

* Created a new feature 'Age' based on beneficiaries' date of birth and date of death.
* Added 'IsDead' feature to indicate whether a beneficiary is deceased.
* Created new features 'TotalClaimAmount,' 'IPTotalAmount,' and 'OPTotalAmount' by aggregating relevant columns.
* Generated an 'IsInpatient' feature based on the presence of 'DiagnosisGroupCode.'

**CHAPTER 6**

# FEATURE SELECTION

## 6.1 CHI-SQUARED FEATURE SELECTION

Chi-squared (χ²) feature selection is a statistical technique used in machine learning and data analysis to identify and select the most relevant features from a dataset, particularly when dealing with categorical or discrete data. It operates on the principle of measuring the statistical independence between each feature and a target variable. The method constructs contingency tables to tabulate the relationships between feature categories and the target variable. By calculating a chi-squared statistic based on observed and expected counts within these tables, the technique quantifies the strength of association between each feature and the target. Features with higher chi-squared scores, indicative of a significant relationship, are considered more informative and are prioritized for inclusion in predictive models. Chi-squared feature selection is valuable for dimensionality reduction, enhancing model interpretability, and improving predictive accuracy in classification tasks by focusing on the most discriminating attributes. It is a versatile tool in the data scientist's toolbox, especially when handling datasets with a multitude of categorical variables.

### 6.1.1 Functions and Libraries:

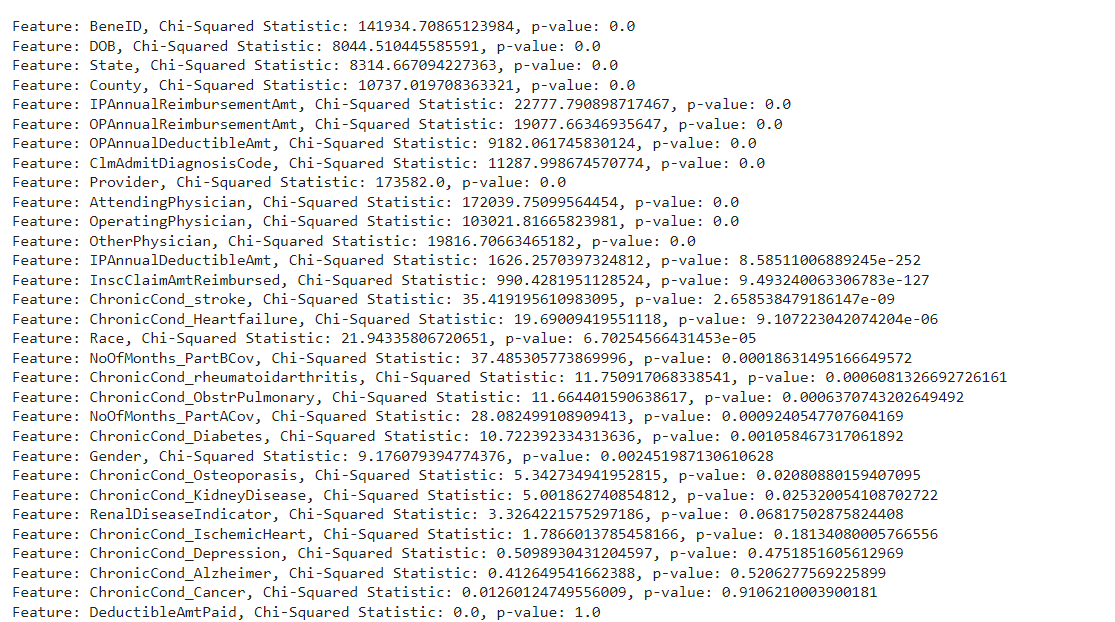
pd.crosstab() (Pandas):

* pd.crosstab() is a Pandas function that is used to create a contingency table (also known as a cross-tabulation or a contingency matrix).
* It takes two or more categorical variables as input and counts the occurrences of each combination of categories between those variables.

scipy.stats.chi2\_contingency() (SciPy):

* scipy.stats.chi2\_contingency() is a function from the SciPy library that is used to perform a chi-squared test of independence on a contingency table.
* It takes a contingency table as input and calculates the chi-squared statistic, the p-value, and other statistics to assess whether there is a statistically significant association between the categorical variables represented in the table.

### 6.1.2 Execution Output:



The above output gives out the potential p-value for all the features to our target value . The closer the value to 0 is highly associated with the target value and closer to 1 , the value is less associated with the target value.



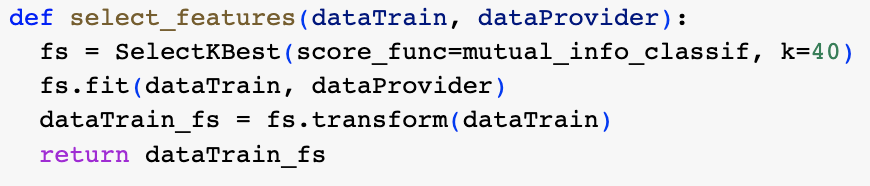
The above output shows the p-values that are highly associated with the target variable.

## 6.2 MUTUAL INFORMATION

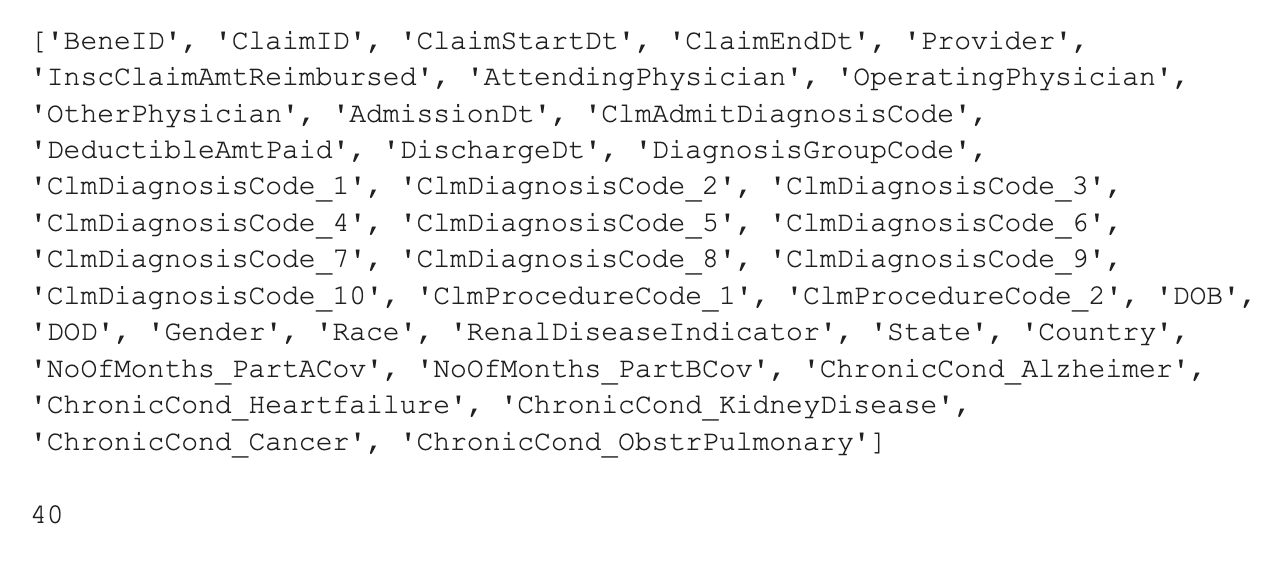
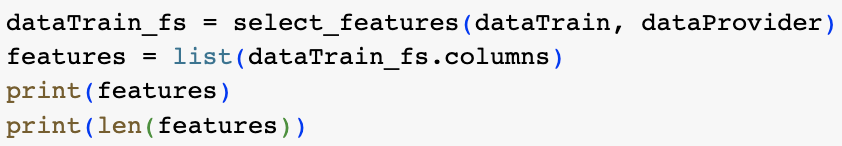
Mutual information (MI) is a measure of the statistical dependence or information shared between two random variables. In the context of feature selection, mutual information is used to assess the relationship between each feature (independent variable) and the target variable (dependent variable or class labels).

It quantifies the reduction in uncertainty (entropy) of the target variable when the feature is known. In other words, it measures how much knowing the feature's value reduces the uncertainty about the target variable. Features that have high mutual information with the target variable provide a lot of information about the target and are considered more relevant.

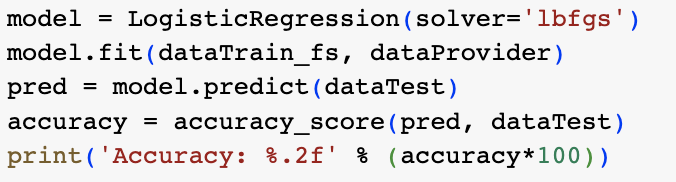
### 6.2.1 SelectKBest Function



### 6.2.2 Selected Features



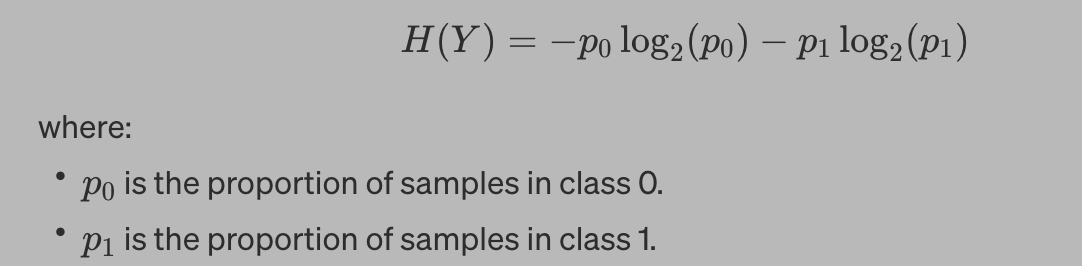
### 6.2.3 Training with Logistic Regression



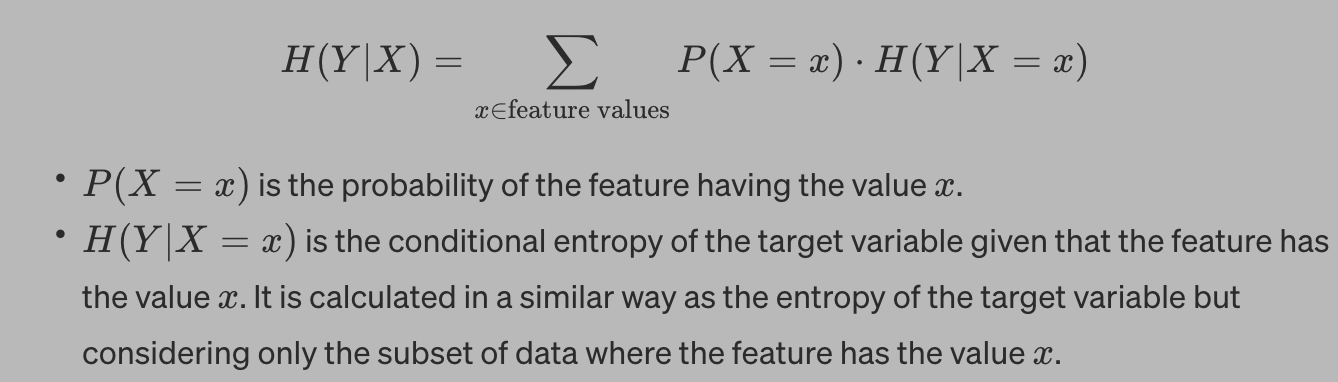


### 6.2.4 Calculation of Reduction in Entropy

Total Entropy *H(Y)* is given by,



Entropy of target variable is calculated as follows,



## 6.3 SPEARMAN’S CORRELATION COEFFICIENT

Spearman's correlation coefficient, often denoted as ρ (rho), is a statistical measure used to quantify the strength and direction of the monotonic relationship between two variables.

The first step in calculating Spearman's correlation is to rank the values of each variable separately. Next, you calculate the differences between the ranks for each pair of data points

For example, if you have two datasets X and Y , then

After calculating the differences, you square each of them (d\_i^2)

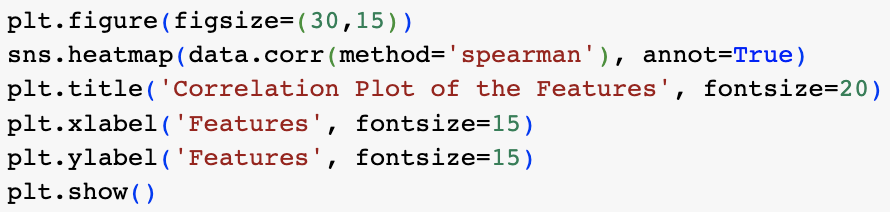
Finally, Spearman's correlation coefficient is calculated

(Where: Σ represents the sum of the values and n is the number of data points.)

Spearman's correlation is often used when the data is ordinal (ranked categories) or when you suspect that a linear relationship is not appropriate for the data.

The resulting ρ value ranges from -1 to 1:

* A ρ value of 1 indicates a perfect monotonic positive relationship, meaning that as one variable increases, the other always increases.
* A ρ value of -1 indicates a perfect monotonic negative relationship, meaning that as one variable increases, the other always decreases.
* A ρ value of 0 indicates no monotonic relationship between the variables; they are independent of each other.



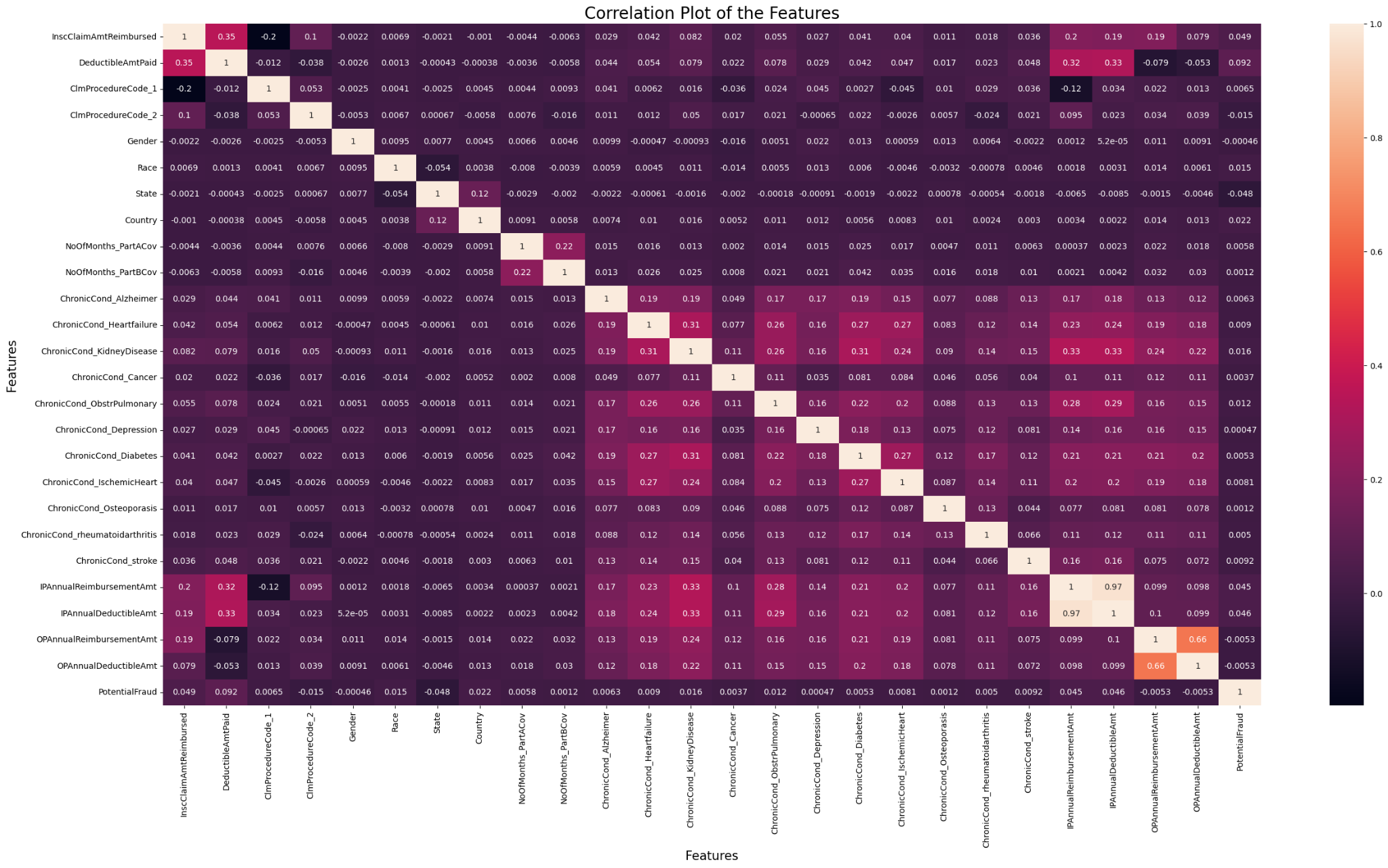


Fig 6.1 Correlation Plot of the Features

Here, we see a few features having high Spearman Correlation Coefficient.

Spearman Correlation Coefficient between some of the features having coefficient greater than 0.70 are:

1. TreatmentDuration and DiagnosisGroupCode: 0.99

2. TreatmentDuration and ClmProcedureCode\_1: 0.74

3. DiagnosisGroupCode and ClmProcedureCode\_1: 0.74

4. ClmDiagnosisCode\_3 and ClmDiagnosisCode\_4: 0.74

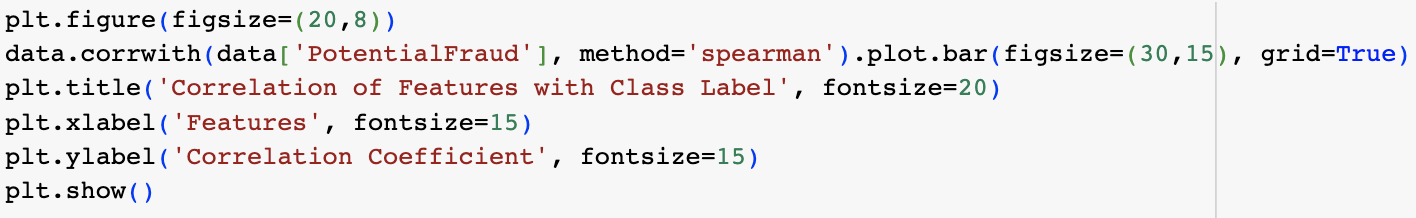
5. ClmDiagnosisCode\_4 and ClmDiagnosisCode\_5: 0.77

6. ClmDiagnosisCode\_5 and ClmDiagnosisCode\_6: 0.84

7. ClmDiagnosisCode\_5 and ClmDiagnosisCode\_7: 0.73

8. ClmDiagnosisCode\_6 and ClmDiagnosisCode\_7: 0.87

9. ClmDiagnosisCode\_6 and ClmDiagnosisCode\_8: 0.77



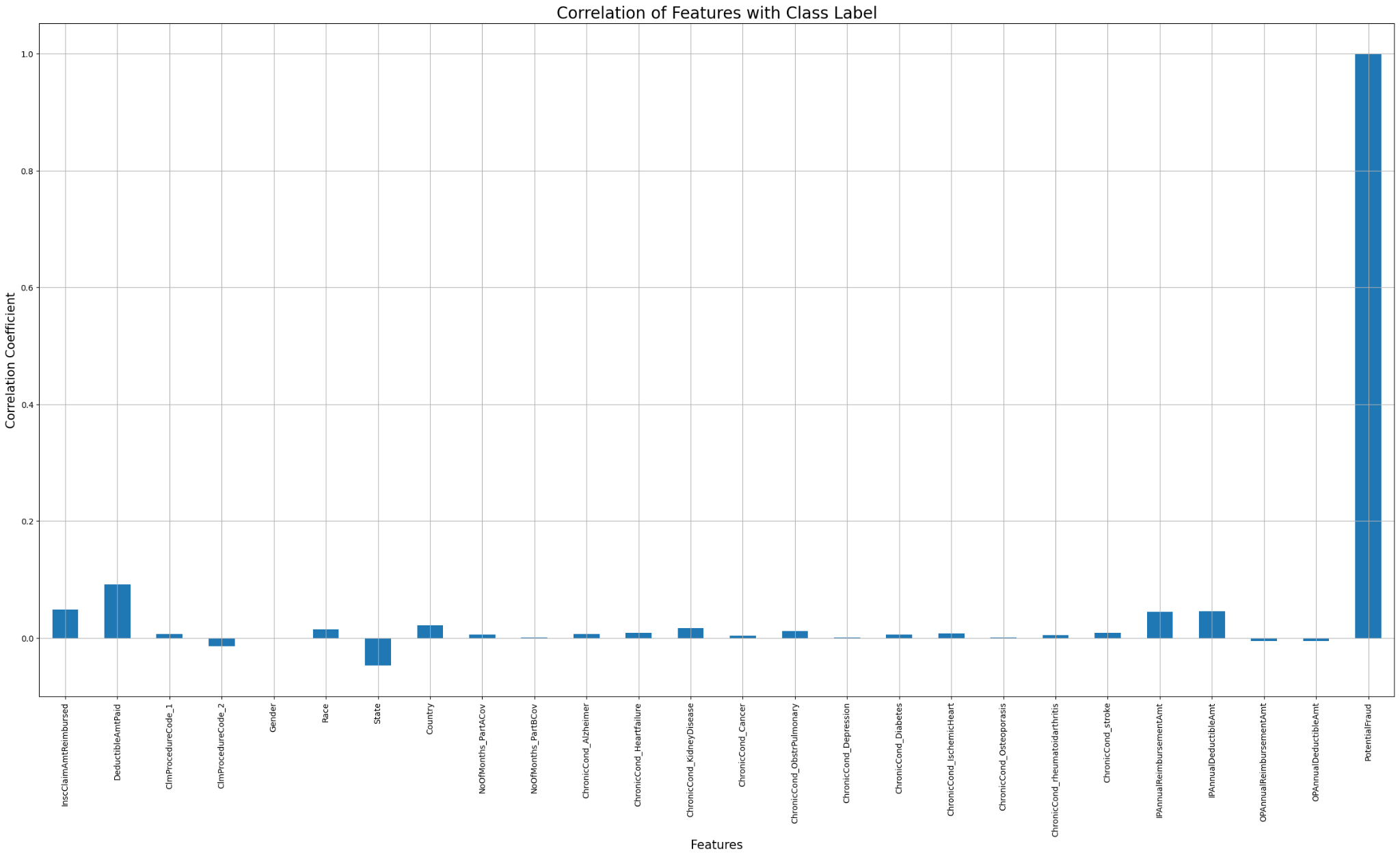


Fig 6.2 – Correlation of Features with Class Label

IPAnnualReimbursementAmt and IPAnnualDeductibleAmt (0.97) and OPAnnualReimbursementAmt and OPAnnualDeductibleAmt (0.66) are the highly correlated features and can be used to find the total claim amount.

When plotted against the class label, 'Gender', 'NoOfMonths\_PartACov', 'ChronicCond\_Depression', 'ChronicCond\_Osteoporasis' and 'OPAnnualReimbursementAmt' are found to be very less correlated. But removal shouldn’t be done because a combination of these features might prove useful.

# CHAPTER 7

# MACHINE LEARNING MODELLING

This chapter details the model training using machine learning algorithms. The data after preprocessing and feature selection is fed to the model for learning. Since the machine learning models could interpret only numerical data, it is necessary to convert any categorical data to numerical. Encoding techniques are used to accomplish this task. In particular response encoding and one-hot encoding are adopted to transform the categorical features ‘State’, ‘Country’ to numerical. This finalized data is fed into four machine learning algorithms for performance comparison. They are i) Decision Tree, ii) Logistic Regression, iii) Random Forest, iv) XGBoost.

## 7.1 DATA SAMPLING

An imbalanced dataset occurs when one class (the minority class) has significantly fewer instances than another class (the majority class). The necessity for using suitable techniques arises from the challenges posed by imbalanced data, and their use depends on the specific problem and machine learning algorithm being employed. Data sampling techniques like balanced class weights, under-sampling, and over-sampling are used to address the issue of imbalanced datasets.

### 7.1.1 Balanced Class Weights

Balanced class weights are typically used in classification tasks, where the class distribution is imbalanced. Without balanced class weights, the model might be biased towards the majority class, leading to poor performance on the minority class. It assigns different weights to the classes, giving more importance to the minority class. This encourages the model to pay equal attention to both classes during training.

### 7.1.2 Under-Sampling

Under-sampling is necessary when you have a large amount of data for the majority class, but you want to balance the class distribution to prevent the model from being overwhelmed by the majority class. It involves randomly removing a subset of instances from the majority class to match the number of instances in the minority class.

### 7.1.3 Over-Sampling

Over-sampling is used when you have a limited amount of data for the minority class, and you want to balance the class distribution. It helps the model better learn the patterns in the minority class. It involves creating synthetic samples for the minority class using techniques like SMOTE (Synthetic Minority Over-sampling Technique). This increases the number of instances in the minority class.

## 7.2 DECISION TREE

Decision tree modelling is a machine learning technique used for both classification and regression tasks. The primary goal is to split the data into subsets that are as homogeneous as possible concerning the target variable. This process continues recursively, resulting in a tree that can make predictions by following a path from the root node to a leaf node. The implementation is discussed in the following sections. Decision tree algorithms are known for their ability to detect important features for classification, making feature selection inherent in their process. The trees' capabilities can be used for feature selection, and feature selection based on the SSV criterion can be designed in different ways[9].

### 7.2.2 Dependencies:

The libraries used in the Decision Tree Modelling are discussed below:

1. **pandas**: pandas is a fundamental library for data manipulation and analysis. You use it extensively for loading, cleaning, and transforming datasets. It provides data structures like DataFrames, making it easy to work with tabular data.
2. **numpy**: numpy is used for numerical operations in Python. It provides support for arrays and matrices, which are essential for performing calculations on the data efficiently.
3. **matplotlib.pyplot**: matplotlib is a popular data visualization library, and pyplot is its submodule. You use it to create various types of plots and charts to visualize data and model performance.
4. **seaborn**: seaborn is built on top of matplotlib and provides a high-level interface for creating attractive statistical graphics. It simplifies the process of creating complex visualizations.
5. **sklearn (Scikit-Learn):** Scikit-Learn is a comprehensive machine learning library in Python. You import specific modules and functions for tasks such as model selection, data splitting, performance metrics, and hyperparameter tuning.
6. **RandomizedSearchCV:** This is a class from Scikit-Learn used for hyperparameter tuning. You use it to search for the best hyperparameters of your Decision Tree Classifier efficiently.
7. **StandardScaler:** StandardScaler is a preprocessing technique from Scikit-Learn for standardizing features. It scales features to have a mean of 0 and a standard deviation of 1, making them suitable for many machine learning algorithms.
8. **mpatches:** This module is part of matplotlib and is used for creating patches or shapes in plots. In your project, it's used for creating legends in heatmap plots.
9. **os:** The os module provides functions for interacting with the operating system. You may use it for tasks related to file paths and directory manipulation.
10. **glob:** The glob module is used for finding all the pathnames matching a specified pattern. It can be handy for loading multiple files or datasets.
11. **tqdm:** tqdm is a library for creating progress bars in Python scripts. It can be useful for tracking the progress of time-consuming tasks, especially in data processing and model training.
12. **datetime:** The datetime module provides classes for working with dates and times. You use it to convert and manipulate date-related columns in your dataset.

### Decision Tree Classifier

The dataset is split into training and testing sets using an 80:20 ratio. This module utilized the DecisionTreeClassifier for fraud detection. It performs hyperparameter tuning using RandomizedSearchCV with various combinations of 'max\_depth' and 'min\_samples\_split.' A pipeline consisting of Standardization of features, response encoding and hyperparameter tuning through RandomizedSearchCV. The decision tree is trained on four ways,

i) best hyperparameters on Response Encoded Data without sampling

ii) best hyperparameters on Response Encoded Data with balanced Class Weights

iii) best hyperparameters on Response Encoded Data with Random Undersampling

iv) hyperparameter tuning on Response Encoded Data with SMOTE Oversampling

### 7.2.4 Evaluation of Decision Tree Classifier

The decision tree trained under four types of sampling is evaluated based on the certain performance metrics and the model’s performance is summarized based on their scores in Table 7.1. All the scores are calculated on the Test data unless it is specified as ‘Train’.

i) Best Hyperparameters on Response Encoded Data Without Sampling

Fig 7.1 shows the confusion matrix, precision matrix and recall matrix of the model’s performance on test data and Fig 7.2 shows ROC curve for the decision tree model. The various performance metrics are also listed below,

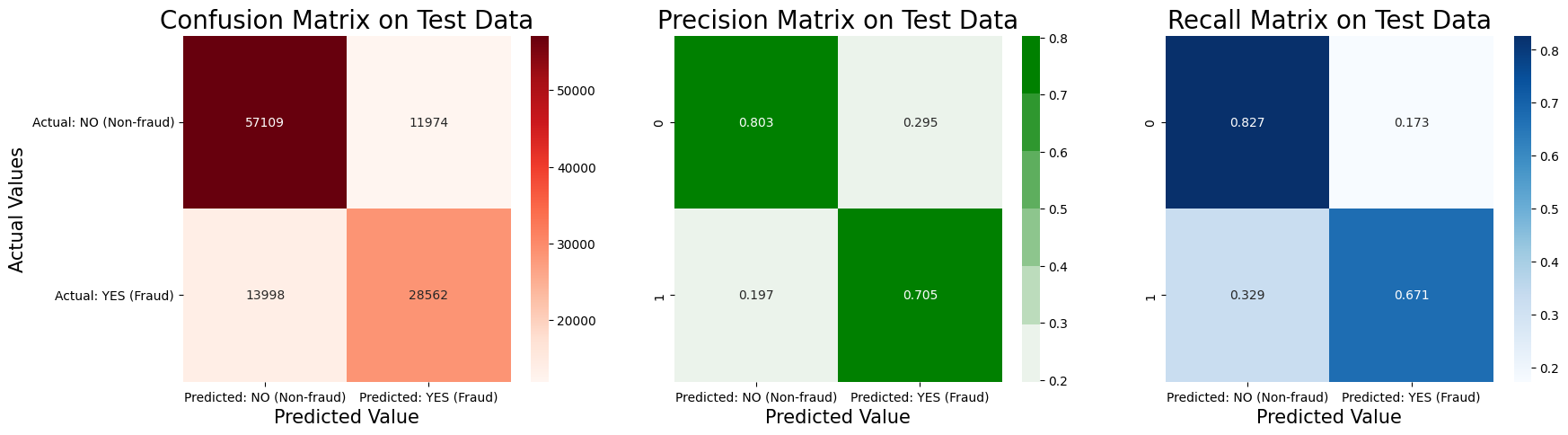


Fig 7.1 – Performance of Decision Tree on Data without Sampling

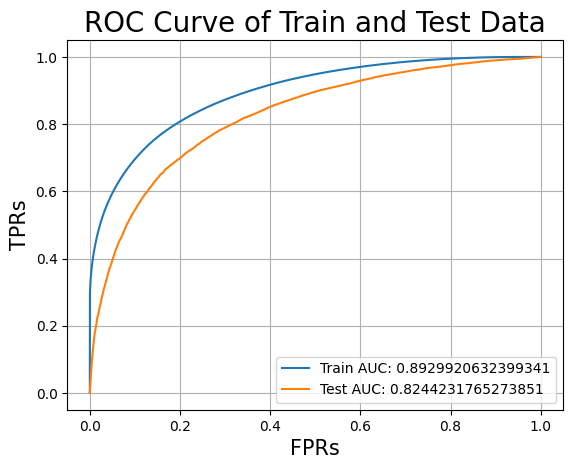


Fig 7.2 AUC Curve for Decision Tree on Data without Sampling

Log-loss of the Model on Train Dataset: 0.43683861992046585

Log-loss of the Model on Test Dataset: 0.6220033012152046

F1-Score of the Model on Test Data: 0.6844669898545654

Percentage of misclassified points (Test Data): 22.62031654470052 %

False Positive Rate (FPR) on Test Data: 0.1459838165684756

False Negative Rate (FNR) on Test Data: 0.35641447368421053

Balanced Accuracy Score (BACC) on Test Data: 0.7488008548736569

Matthew's Correlation Coefficient (MCC) on Test Data: 0.5117835195993226

ii) best hyperparameters on Response Encoded Data with balanced Class Weights

Fig 7.3 shows the confusion matrix, precision matrix and recall matrix of the model’s performance on test data and Fig 7.4 shows ROC curve for the decision tree model. The various performance metrics are also listed below,

Log-loss of the Model on Train Dataset: 0.4501686022698587

Log-loss of the Model on Test Dataset: 0.6562270319478715

F1-Score of the Model on Test Data: 0.7079818228055238

Percentage of misclassified points (Test Data): 23.656655589692143 %

False Positive Rate (FPR) on Test Data: 0.2296802397116512

False Negative Rate (FNR) on Test Data: 0.24774436090225563

Balanced Accuracy Score (BACC) on Test Data: 0.7612876996930467

Matthew's Correlation Coefficient (MCC) on Test Data: 0.512826589667137



Fig 7.3 - Performance of Decision Tree on Data with Balanced Class Weights

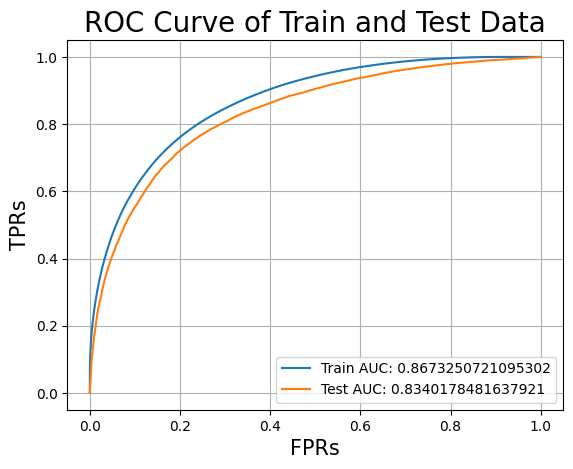


Fig 7.4 - AUC Curve for Decision Tree on Data with Balanced Class Weights

iii) best hyperparameters on Response Encoded Data with Random Undersampling

Fig 7.5 shows the confusion matrix, precision matrix and recall matrix of the model’s performance on test data and Fig 7.6 shows ROC curve for the decision tree model. The various performance metrics are also listed below,

Log-loss of the Model on Train Dataset: 0.45807164499130837

Log-loss of the Model on Test Dataset: 0.6428926487796232

F1-Score of the Model on Test Data: 0.7015953952861803

Percentage of misclassified points (Test Data): 24.007774782118002 %

False Positive Rate (FPR) on Test Data: 0.22801557546719164

False Negative Rate (FNR) on Test Data: 0.25965695488721807

Balanced Accuracy Score (BACC) on Test Data: 0.7561637348227952

Matthew's Correlation Coefficient (MCC) on Test Data: 0.5036169762407696

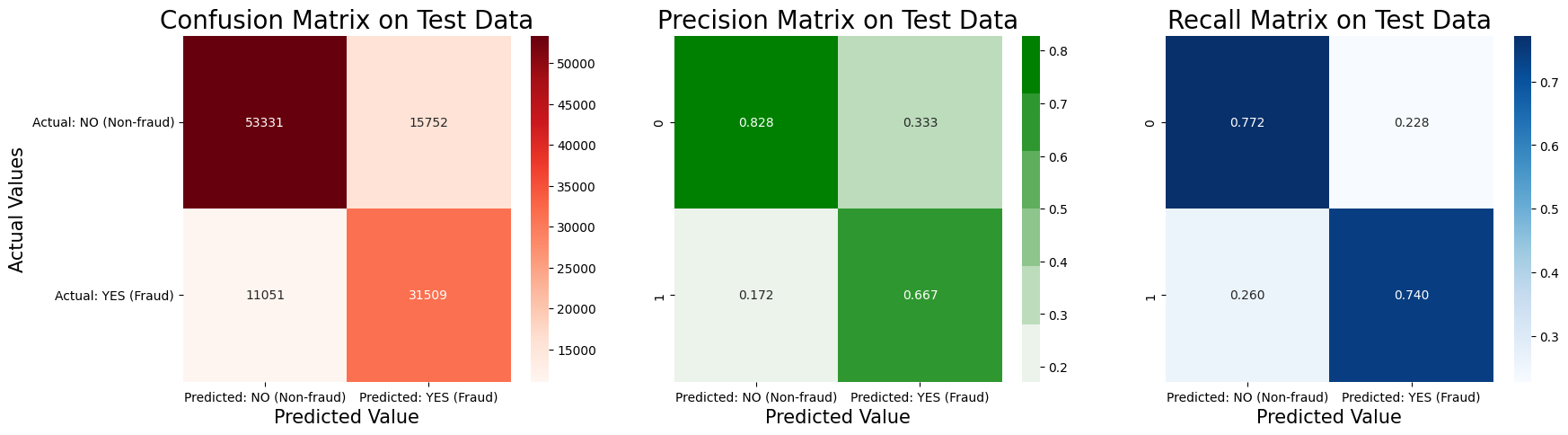


Fig 7.5 - Performance of Decision Tree on Data with Random Undersampling

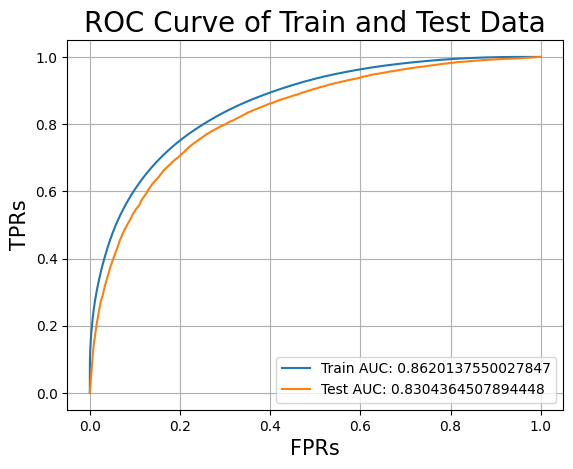


Fig 7.6 - AUC Curve for Decision Tree on Data with Random Undersampling

iv) hyperparameter tuning on Response Encoded Data with SMOTE Oversampling

Fig 7.7 shows the confusion matrix, precision matrix and recall matrix of the model’s performance on test data and Fig 7.8 shows ROC curve for the decision tree model. The various performance metrics are also listed below,

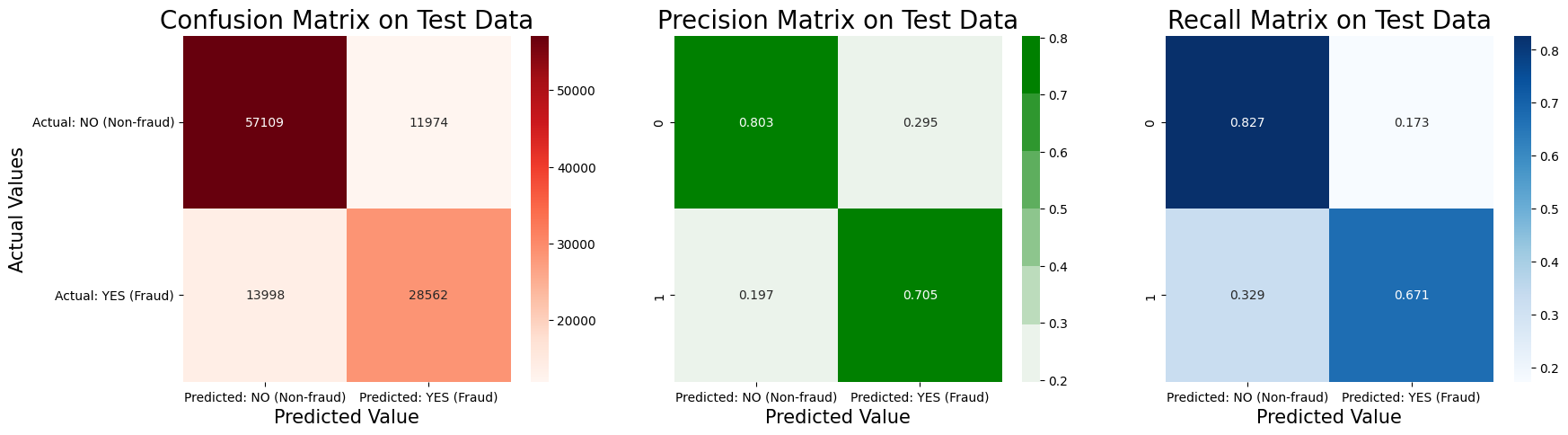


Fig 7.7 - Performance of Decision Tree on Data with SMOTE Oversampling

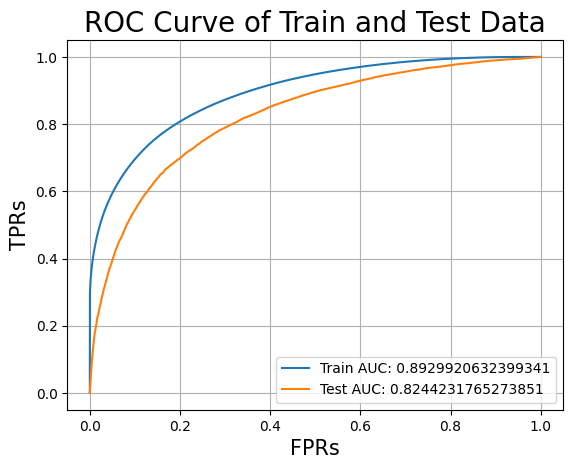


Fig 7.8 - AUC Curve for Decision Tree on Data with SMOTE Oversampling

Log-loss of the Model on Train Dataset: 0.40211044559944387

Log-loss of the Model on Test Dataset: 0.63622406049923

F1-Score of the Model on Test Data: 0.6874458457687493

Percentage of misclassified points (Test Data): 23.263437922664206 %

False Positive Rate (FPR) on Test Data: 0.17332773620138095

False Negative Rate (FNR) on Test Data: 0.3289003759398496

Balanced Accuracy Score (BACC) on Test Data: 0.7488859439293847

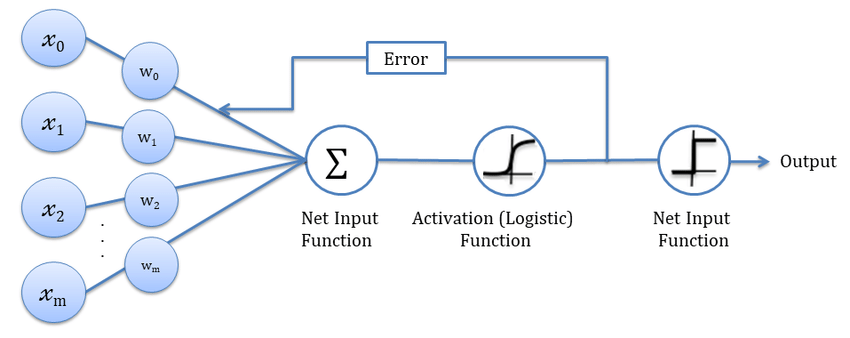
Matthew's Correlation Coefficient (MCC) on Test Data: 0.5027361861545789

| Sampling | Log-loss | | F1-Score | Misclassified Points % | FPR | FNR | BACC | MCC | AUC | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Test | Train | Test |
| (i) | 0.436 | 0.622 | 0.684 | 22.620 | 0.145 | 0.356 | 0.748 | 0.511 | 0.864 | 0.832 |
| (ii) | 0.450 | 0.656 | 0.707 | 23.656 | 0.229 | 0.247 | 0.761 | 0.512 | 0.867 | 0.834 |
| (iii) | 0.458 | 0.642 | 0.701 | 24.007 | 0.228 | 0.259 | 0.756 | 0.503 | 0.862 | 0.830 |
| (iv) | 0.402 | 0.636 | 0.687 | 23.263 | 0.173 | 0.328 | 0.748 | 0.502 | 0.892 | 0.824 |

Table 7.1 Summary of Performance of Decision Tree under various sampling

## 7.3 LOGISTIC REGRESSION

Logistic regression is a statistical model used for binary classification, where it predicts one of two possible outcomes based on input features. It calculates the probability of the target variable belonging to a particular class by using a logistic (sigmoid) function to transform the linear combination of features. It's trained with labeled data and learns feature coefficients to best fit the observed outcomes. One of its strengths is interpretability, as it allows us to understand the influence of individual features on the prediction. Logistic regression finds applications in various domains, including medicine, finance, and marketing, making it a valuable tool for decision-making in situations involving binary classification problems.



### 7.3.1 Hyperparameter Tuning on One Hot Encoded Data without sampling

Output:

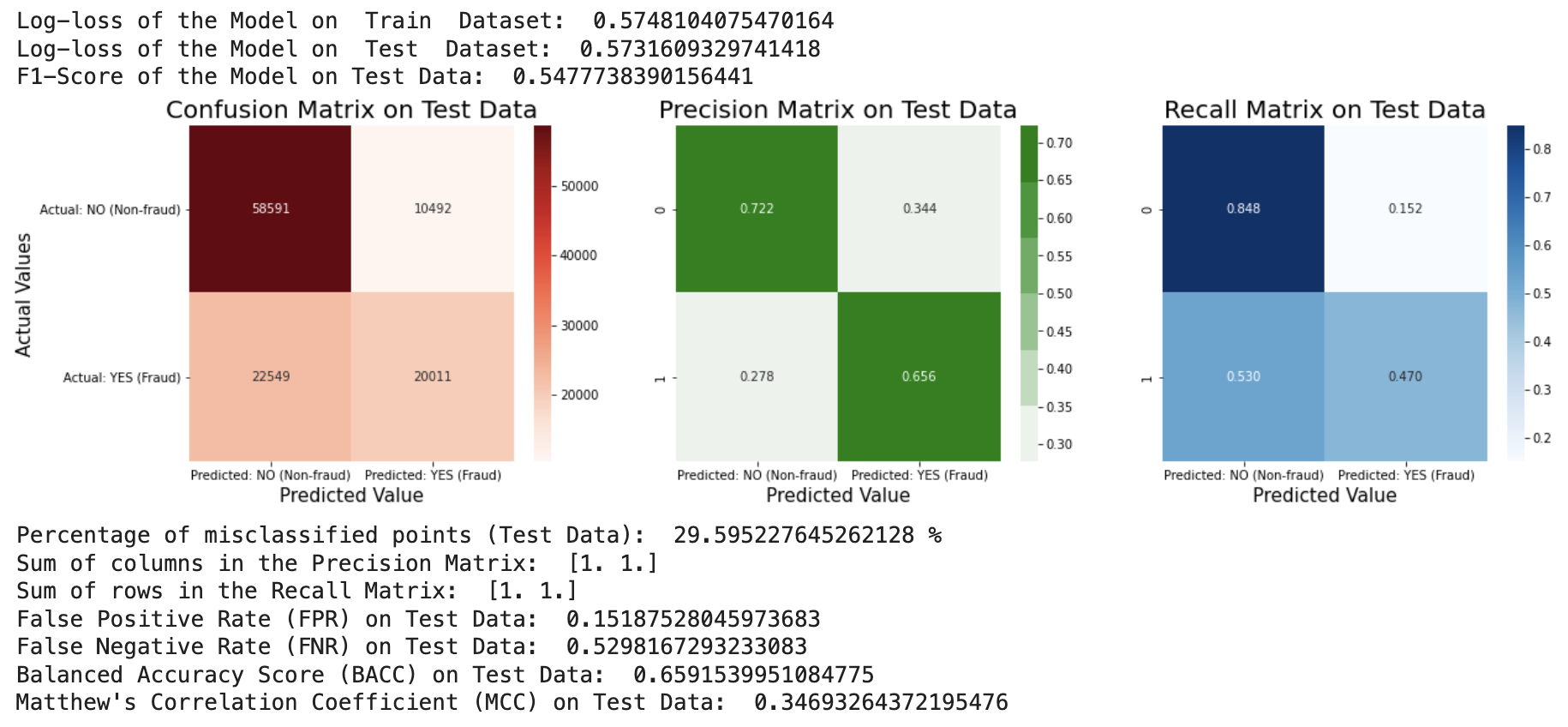


fig 7.3.1(a) - Performance metrics

Fig 7.3.1(b) - ROC curve of Hyperparameter Tuning on One Hot Encoded Data without sampling

### 7.3.2. Hyperparameter Tuning on One Hot Encoded Data with balanced Class Weights

Output:

### 

### 

Fig 7.3.2 (a) - Performance Metrics of Hyperparameter Tuning on One Hot Encoded Data with balanced Class Weights

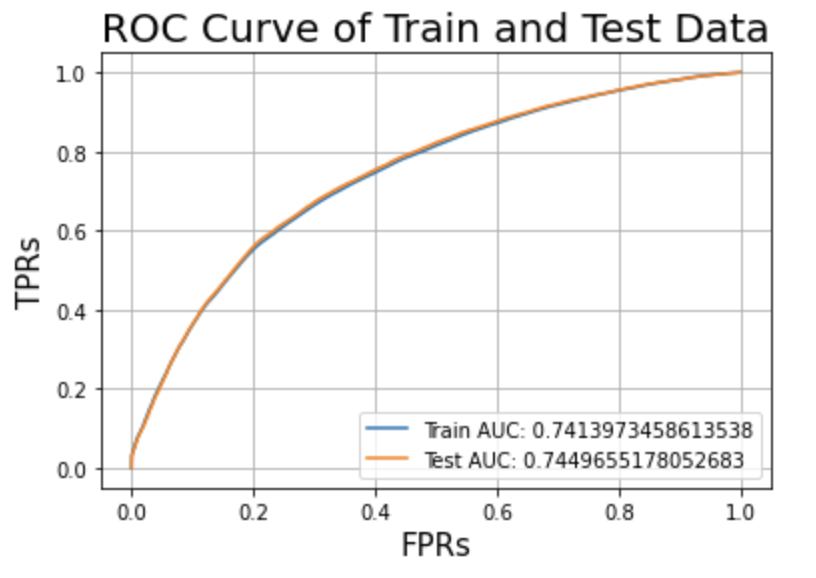


Fig 7.3.2 (b) - ROC of Hyperparameter Tuning on One Hot Encoded Data with balanced Class Weights

## 

### 7.3.3. Train the Model now with best hyperparameters on One Hot Encoded Data with Random Undersampling

Output:



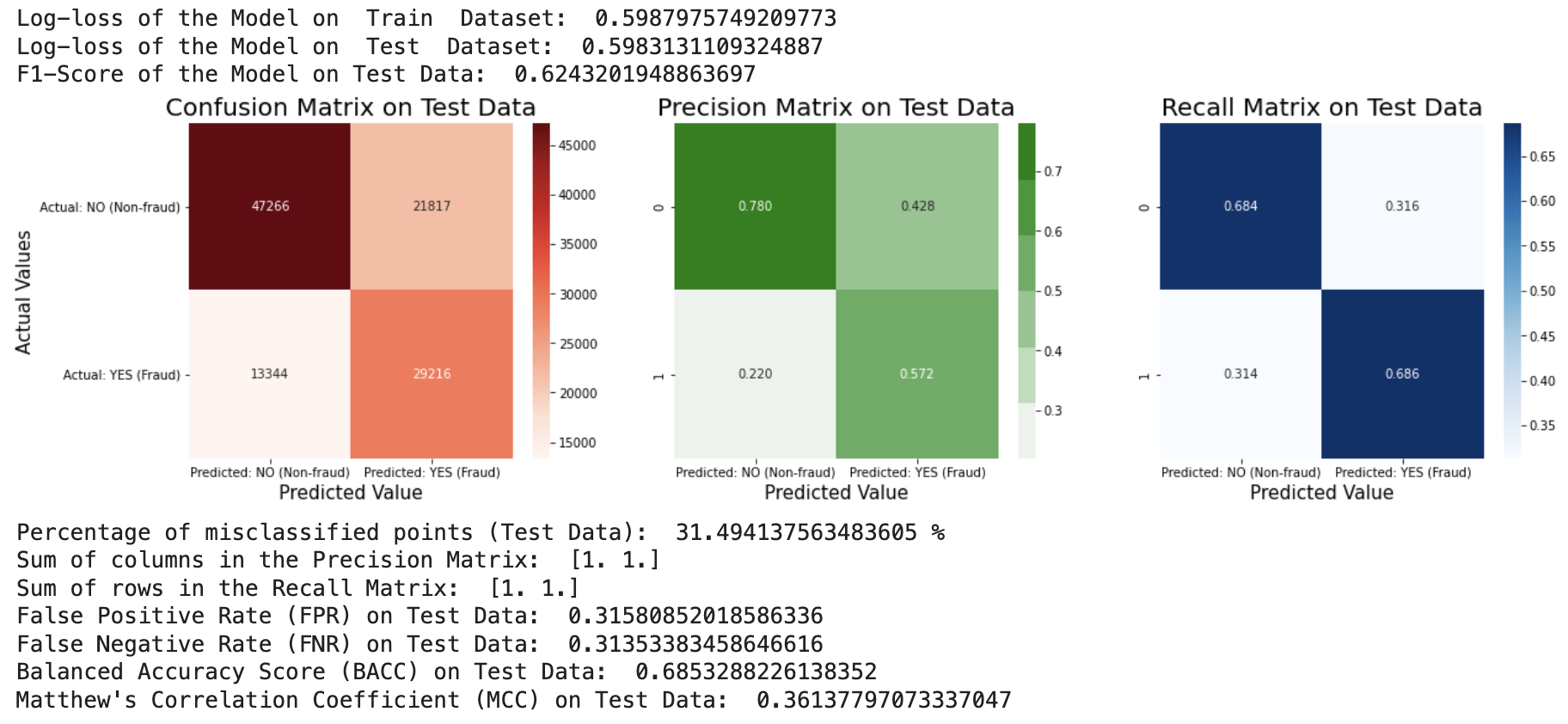


Fig 7.3.3(a) - Performance metrics of Logistic Regression with Random Undersampling

### 

Fig 7.3.3(b) - ROC curve of Logistic Regression with Random Undersampling

### 

### 7.3.4. Train the Model now with best hyperparameters on One Hot Encoded Data with SMOTE Oversampling

Output:





Fig 7.3.4(a) - Performance metrics of Logistic Regression with SMOTE Oversampling

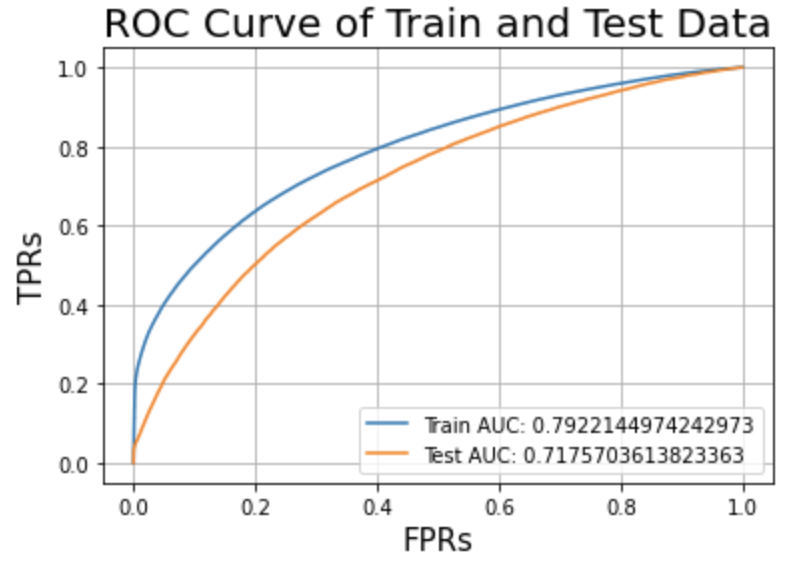


Fig 7.3.4(b) - ROC curve of Logistic Regression with SMOTE Oversampling

## 7.4 RANDOM FOREST

A Random Forest is a popular machine learning ensemble method used for both classification and regression tasks. It is composed of multiple decision trees, and the name "Random Forest" suggests the idea of combining many individual trees to form a robust and accurate model. The key idea behind a Random Forest is to reduce overfitting (a common problem in decision trees) and improve the model's generalization performance.

**Data Sampling**: A random subset of the training data is selected through a process called "bootstrapping" (random sampling with replacement). This means that each decision tree in the forest is trained on a slightly different dataset. This diversity helps reduce overfitting.

**Feature Selection:** At each node of each decision tree, instead of considering all the features, only a random subset of features is considered for making the split. This further diversifies the trees.

**Tree Building**: Each decision tree in the forest is constructed using a variant of the decision tree algorithm, such as the CART (Classification and Regression Trees) algorithm. The tree is built by repeatedly splitting the data into subsets based on the selected features and their values.

**Voting/Averaging:** For classification tasks, the results from all individual trees are combined through a majority vote, and for regression tasks, the predictions are averaged to produce the final prediction.

Random Forests offer several advantages:

* They are highly resistant to overfitting because of the ensemble nature and the use of bootstrapping.
* They can handle a large number of features, even when some of them are irrelevant or noisy.
* They provide an estimate of feature importance, which can be useful for feature selection.
* They generally have good predictive accuracy and work well "out of the box."

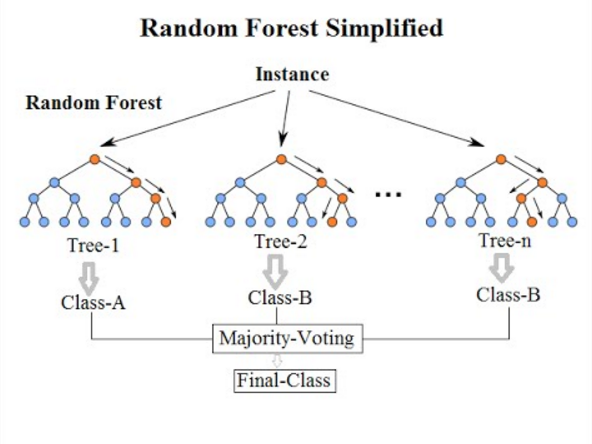
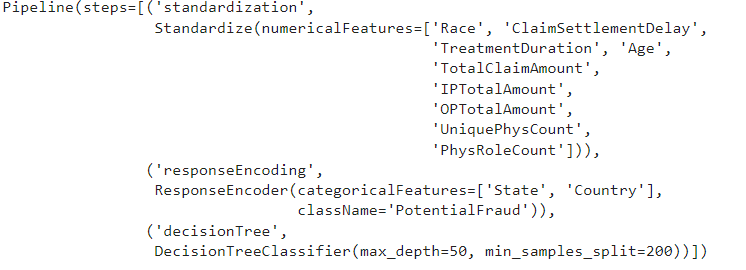


Fig 7.4(a) - Pictorial Representation of Random Forest

### 7.4.1. Hyperparameter Tuning on Response Encoded Data without sampling

Output :





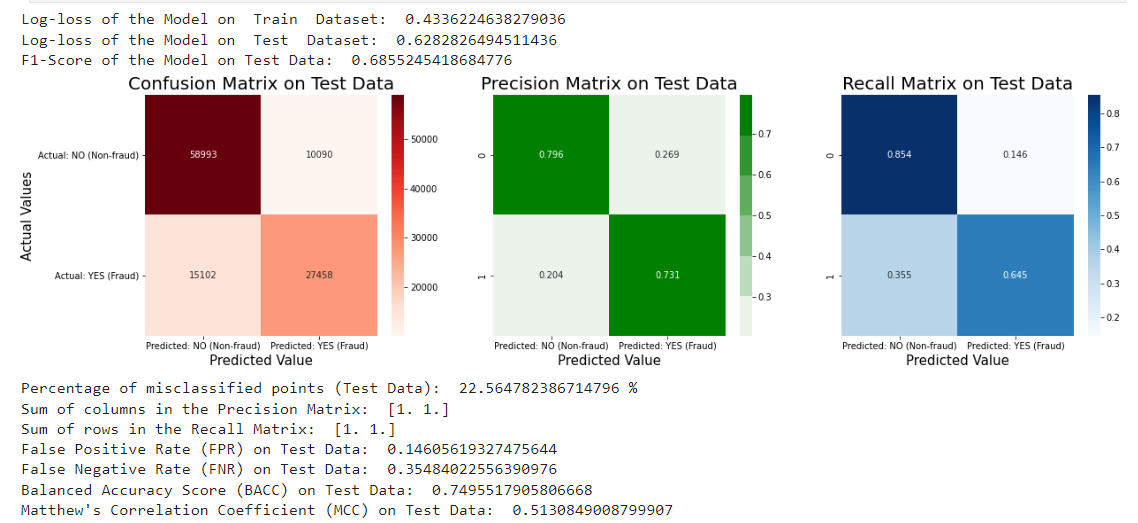


Fig 7.4.1 (a) - Performance Metrics of Random Forest without sampling

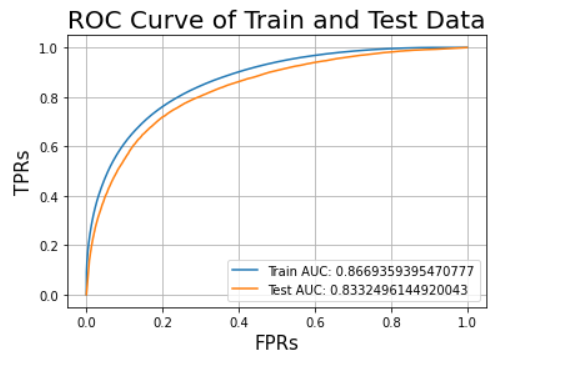


Fig 7.4.1 (b) - ROC Curve Of Random Forest without sampling

### 

### 7.4.2 Hyperparameter Tuning on Response Encoded Data with balanced Class Weights

Output:







Fig 7.4.2(a) - Performance Metrics Of Balanced Class Weights

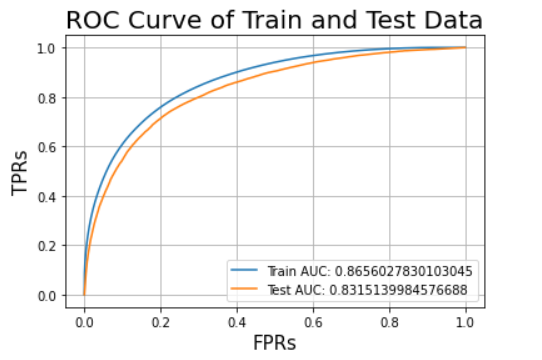


Fig 7.4.2(b) - ROC Curve Of Balanced Class Weights

### 7.4.3 Hyperparameter Tuning on Response Encoded Data with Random Undersampling

Output:



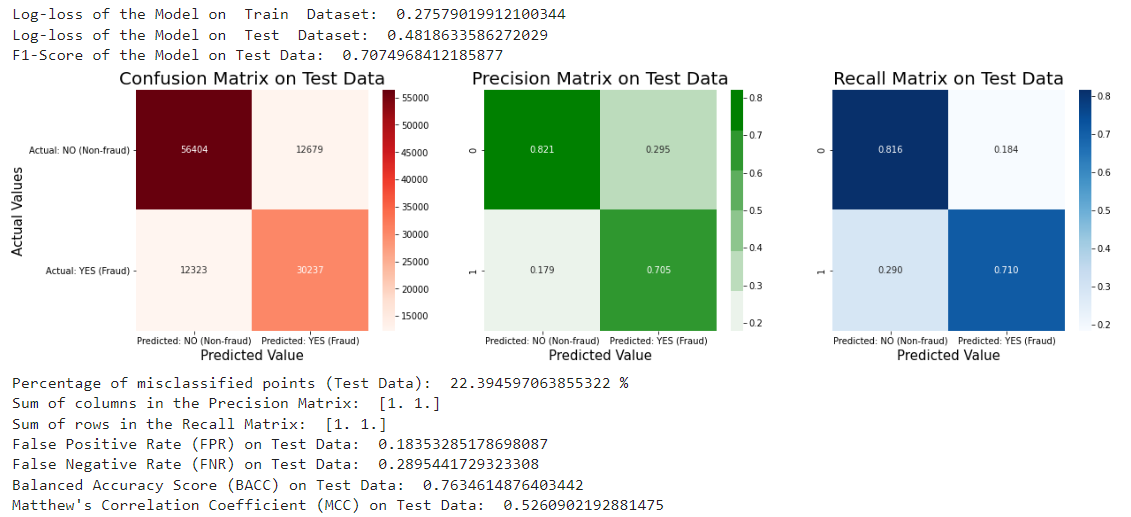


Fig 7.4.3(a) - Performance Metrics Of Random Forest with Random Undersampling

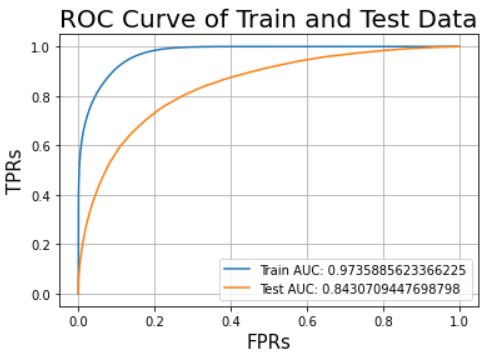


Fig 7.4.3 (b) - ROC curve of Random Forest with Random Undersampling

### 7.4.4 Training the Model with best hyperparameters on Response Encoded Data with SMOTE Oversampling

Output:



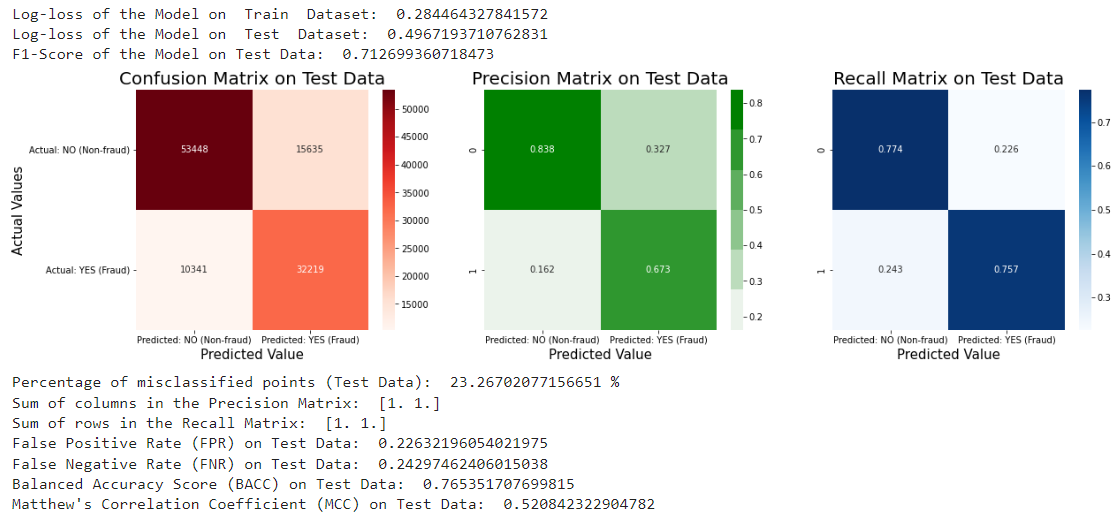


Fig 7.4.4 (a) - Performance Metrics - Random Forest with SMOTE Oversampling

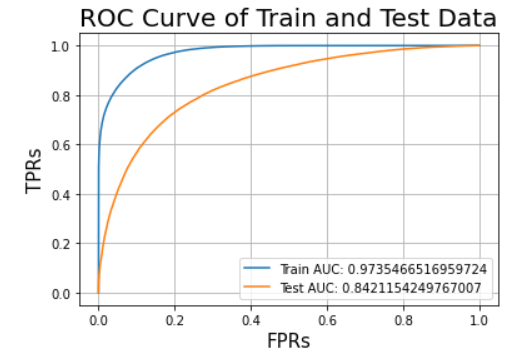


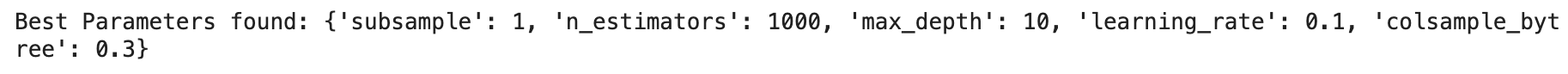
Fig 7.4.4(b) - ROC Curve Of Random Forest with SMOTE Oversampling

## 7.5 XGBOOST

XGBoost, short for Extreme Gradient Boosting, is a powerful and popular machine learning algorithm known for its exceptional performance in various data-driven tasks. It falls under the category of gradient boosting algorithms and is particularly renowned for its efficiency and effectiveness in handling structured datasets. XGBoost combines multiple decision trees to create an ensemble model, which enhances predictive accuracy and reduces overfitting. It employs a gradient descent optimization technique to iteratively improve the model's predictions. XGBoost is highly versatile, capable of handling regression, classification, and ranking problems. Its robustness, speed, and feature selection capabilities make it a favored choice among data scientists and machine learning practitioners for achieving state-of-the-art results in predictive modelling and competitive data science competitions.

### 7.5.1 Hyperparameter Tuning on Response Encoded Data without sampling

Output :



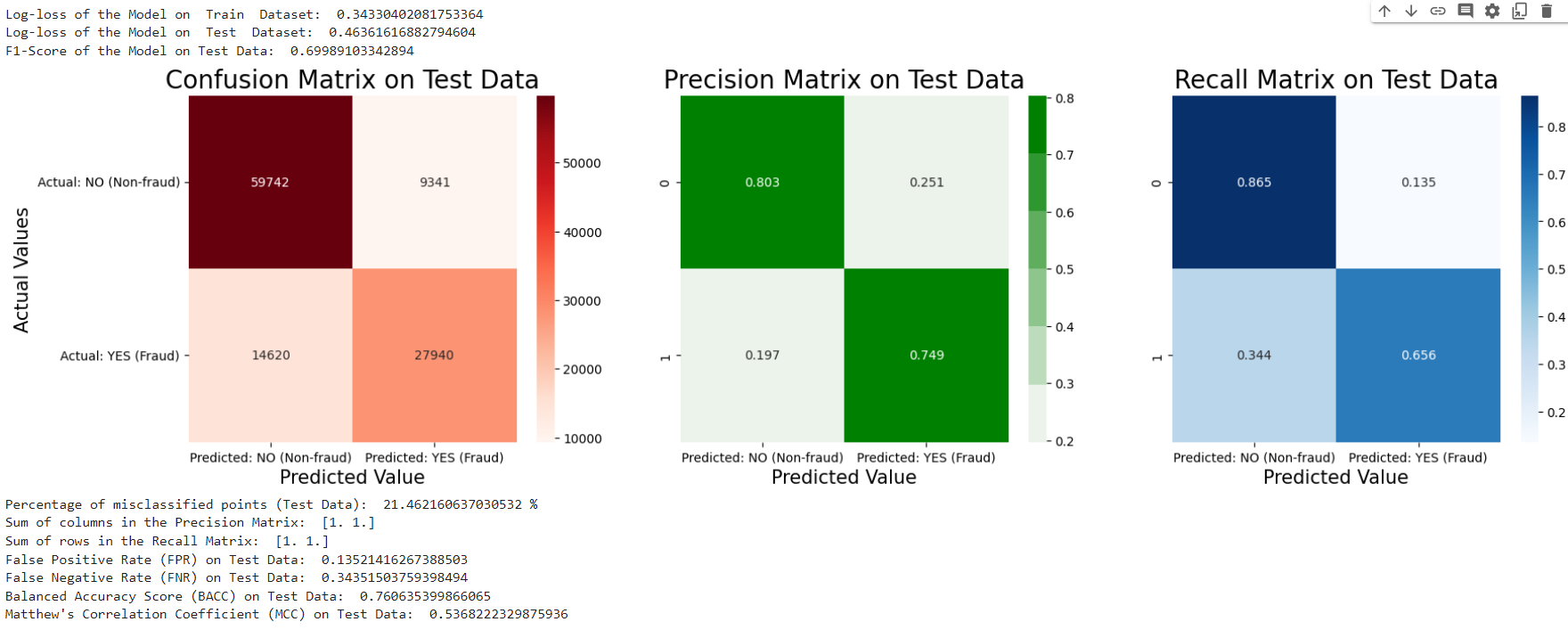


Fig 7.5.1 (a) - Performance Metrics - XGBoost without Sampling

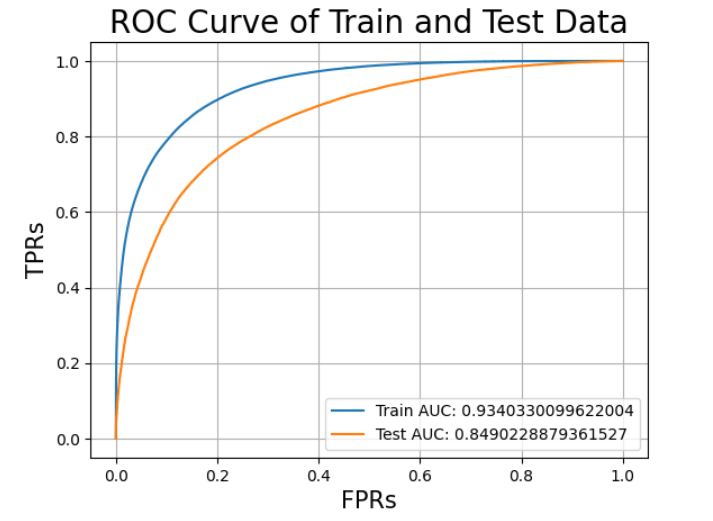
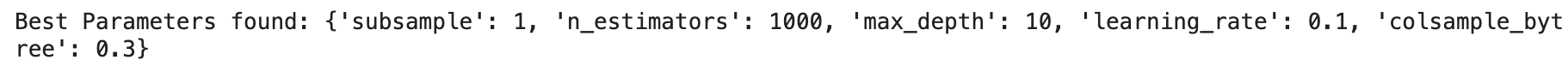


Fig 7.5.1(b) - ROC curve Of XGBoost without Sampling

### 7.5.2. Hyperparameter Tuning on Response Encoded Data with Random Undersampling

Output:



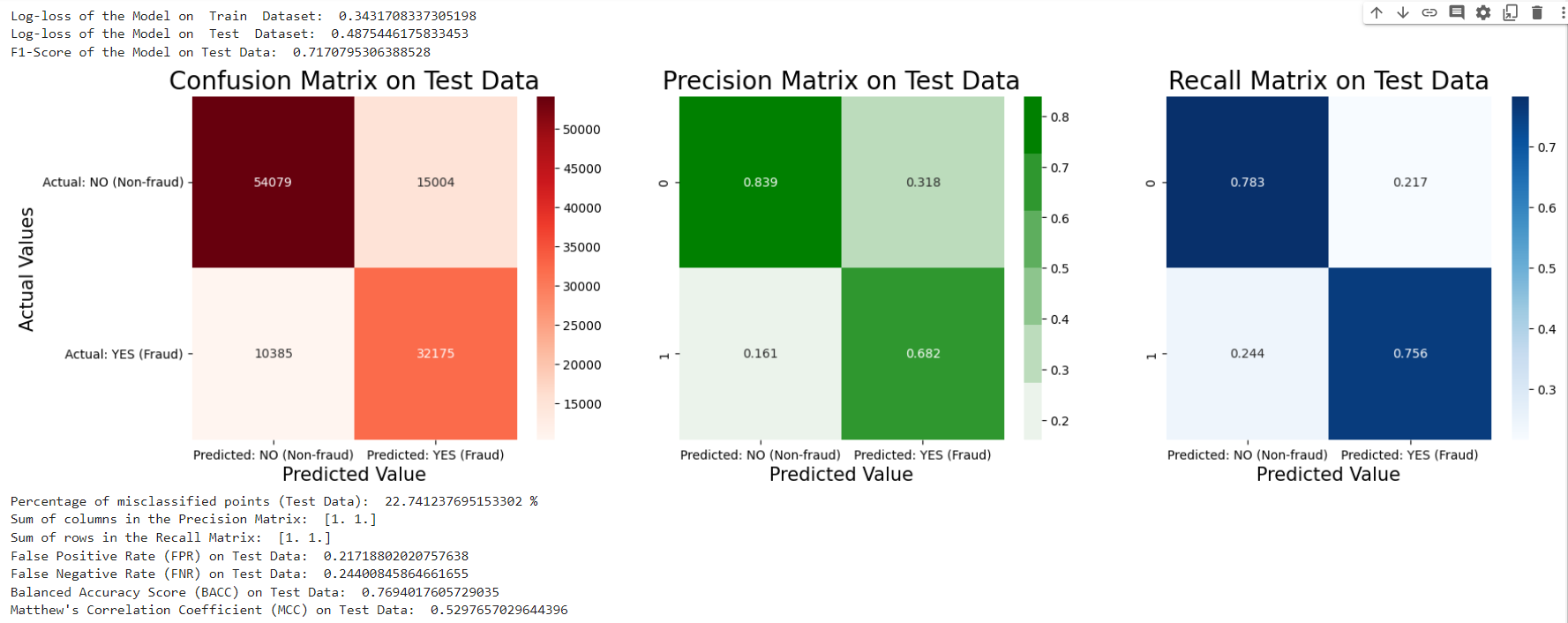


Fig 7.5.2(a) - Performance Metrics - XGBoost with Random Undersampling

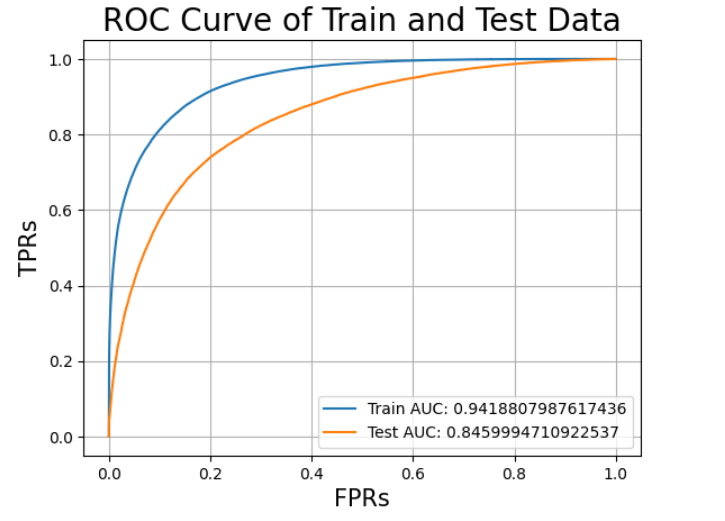


Fig 7.5.2(b) - ROC curve - XGBoost with Random Undersampling

### 7.5.3. Training the Model with best hyperparameters on Response Encoded Data with SMOTE Oversampling

## 

Fig 7.5.3 (a) - Performance Metrics - XGBoost with SMOTE Oversampling

## 

Fig 7.5.3(b) - ROC Curve - XGBoost with SMOTE Oversampling

## 7.6 BEST MODEL SELECTION

On observing the scores of various models it is inferred that,

1. The Decision Tree with balanced class weights received the least 'FNR' score of 0.24791.

2. Random Forest with balanced class weights received the minimum 'Train AUC' and minimum 'Train Log-loss'.

3. XGBoost with SMOTE Oversampling received the least 'Test Log-loss', 'percentage misclassification', 'FPR' and the maximum 'Test AUC', 'F1-Score', 'BACC' and 'MCC' score.

As a conclusion **XGBoost Model with SMOTE oversampling** gives the best scores for most of the performance metrics and it is chosen as the final model.

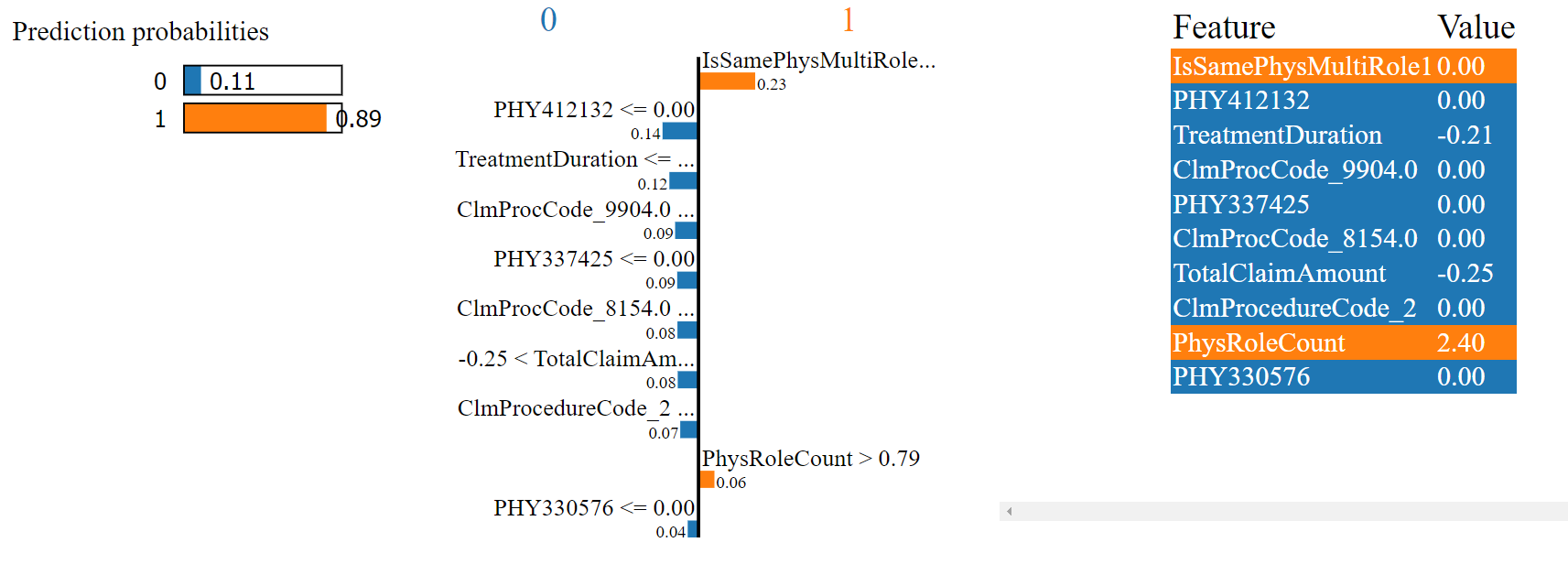
# CHAPTER - 8

# MODEL INTERPRETABILITY

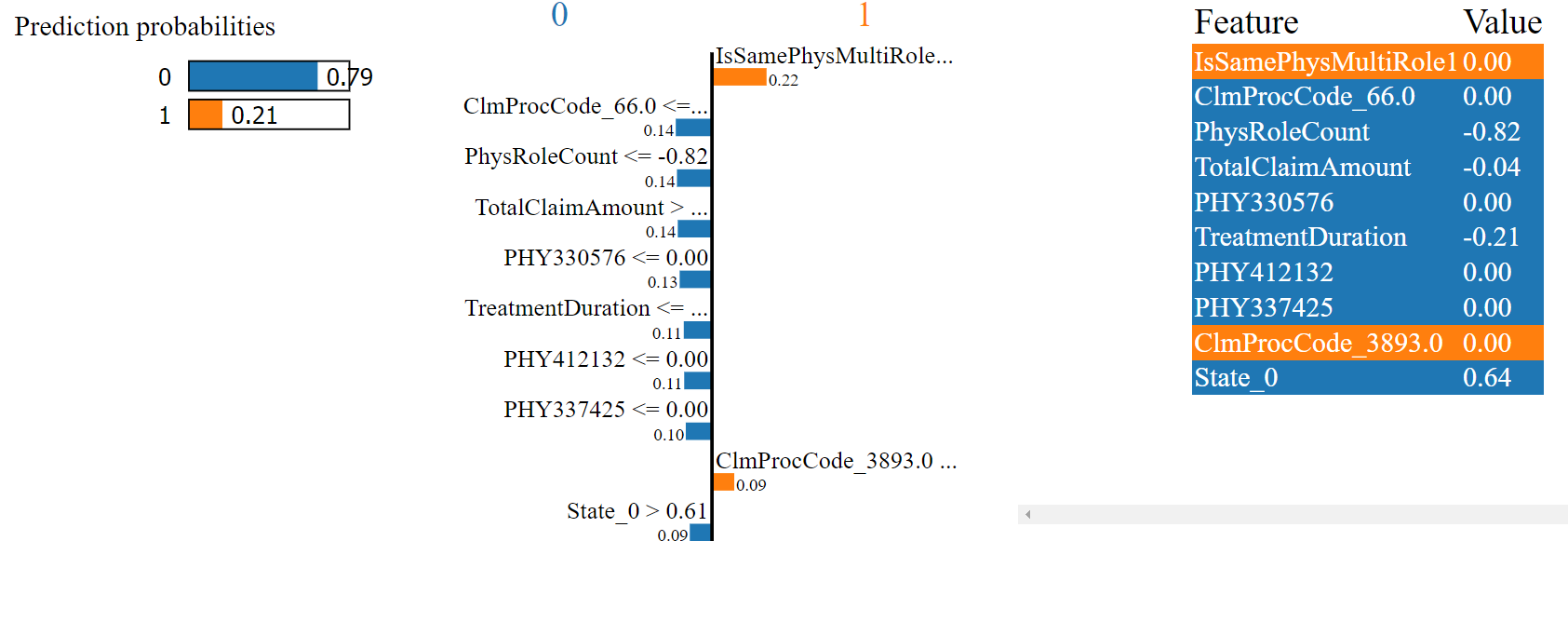
## 

## LIME

Positive



Negative



# CHAPTER 9

# CONCLUSION

The literature survey underscores the pressing concern of privacy preservation in social network clustering. Papers across various journals and conferences collectively highlight the significance of combining encryption and decentralized computation to achieve privacy-preserving insights from network data. By considering diverse methodologies and techniques, this survey pinpoints the necessity of striking a balance between data utility and individual privacy, a balance that the proposed solution aims to achieve by drawing insights from these influential works.

# CHAPTER- 10

# BIBLIOGRAPHY

1. **“** A novel fraud detection and prevention method for healthcare claim processing using machine learning and blockchain technology Anokye Acheampong Amponsah ∗ , Adebayo Felix Adekoya, Benjamin Asubam Weyori **“**
2. Sathya, M. ., & Balakumar , B. . (2022). Insurance Fraud Detection Using Novel Machine Learning Technique. *International Journal of Intelligent Syms and Applications in Engineering*, *10*(3), 374–381. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2178
3. “ Naga Ramya Bhamidipati\*, Varsha Vakkavanthula\*, George Stafford\*, Masrik Dahir\*, Roshan Neupane\*, Ernest Bonnah, Songjie Wang, J. V. R. Murthy, Khaza Anuarul Hoque, Prasad Calyam University of Missouri-Columbia, USA; Jawaharlal Nehru Technological University, India. “
4. R. Roy and K. T. George, "Detecting insurance claims fraud using machine learning techniques," 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-6, doi: 10.1109/ICCPCT.2017.8074258.
5. Sun C., Li Q., Li H., Shi Y., Zhang S., Guo W., “Patient Cluster Divergence Based Healthcare Insurance Fraudster Detection”, IEEE Access, Vol. 7, pp. 14162–14170, 2019. <https://ieeexplore.ieee.org/document/8576507>
6. <https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis>
7. <https://www.kaggle.com/datasets/arashnic/imbalanced-data-practice>
8. [https://labelyourdata.com/articles/data-collection-methods-AI#:~:text=Simply%20 put%2C%20data%20collection%20 is,fed%20into%20an%20 ML%20mod](https://labelyourdata.com/articles/data-collection-methods-AI#:~:text=Simply%20put%2C%20data%20collection%20is,fed%20into%20an%20ML%20model)
9. K. Grabczewski and N. Jankowski, "Feature selection with decision tree criterion," Fifth International Conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro, Brazil, 2005, pp. 6 pp.-, doi: 10.1109/ICHIS.2005.43.