Insurance Fraud Detection Using Machine Learning

**Abstract – The increasing fraudulent activities inside the insurance coverage industry requires innovative and novel answers. This paper ambitions to plan a version for the detection of fraudulent activities inside the subject of health insurance. Many present solutions lag because of constrained feature selection. The proposed version classifies insurance claims with a high degree of precision, efficacy, and dependability. An extensive dataset comprising coverage claims from patients is tuned using pre-processing techniques, consisting of standardization and encoding techniques and most effective features that facilitate learning are retained. Sampling strategies are used to moderate the imbalance nature in facts which was discovered in Exploratory Data Analysis. The data is trained on four different machine learning algorithms like logistic regression, decision tree, random forests, and XGBoost. On comparison the eXtreme Gradient Boosting version with SMOTE oversampling is recognized as remarkable in performance metrics like minimum log loss, misclassification price, false fantastic fee (FPR), and maximum place below the curve (AUC), F1 score, Balanced Accuracy Score (BACC), Matthew Correlation Coefficient (MCC). This model allows early detection of coverage fraud which benefits each insurer and policyholders.**

***Keywords - Insurance frauds, machine learning, logistic regression, decision tree, random forest, xgboost, fraud detection.***

I Introduction

Insurance serves as a safety assurance in an uncertain world providing financial protection and peace of mind. But, the unfortunate reality is it is being exploited by some individuals by taking advantage of the trust placed in insurance systems through deceitful practices known as insurance frauds. These fraudulent activities not only threaten the financial stability of insurance companies but also lead to increased premiums for honest policyholders The Coalition Against Insurance Fraud estimates a minimum of $80 billion every year in the US alone [].

To overcome this issue, an exhaustive yet innovative solution must be developed. Manual reviewing is not considered a good choice as it is time consuming and tedious. Machine learning has proven to be an outstanding concept in data analyzing. Machine learning may detect fraudulent insurance claims admitted in a health care or hospital by analyzing 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.

Several existing approaches are studied and reviewed in Section II with their pros and cons. Their limitations and scope for improvement are identified and assimilated as a guidance. An extensive dataset is essential in identifying unusual behavioral patterns in fraudulent claims. Also the dataset must possess the characteristic of diversity in the source of data. This paper has collected insurance claim records and has performed necessary data processing techniques to make it fitting to the machine learning models. Different feature selection methods are experimented and compared to recognize the best technique suitable to the given dataset. Exploratory Data Analysis(EDA) is performed on the dataset to inspect the distribution of values and variables.

The findings of the EDA are studied to find the imbalance in the distribution of classes. It leads to adopting sampling techniques to maintain balance in the dataset. Different machine learning models are trained with the data under different sampling 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 which is an important evidence that substantiates the findings of the model with reliability. The details of the process would be discussed in the forthcoming chapters.

II. Literature Survey

Anokye et al, in their paper provide a detailed explanation of a new approach to healthcare fraud detection and prevention using machine learning and blockchain technology [1]. 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. 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.

Sathya et al, This paper discusses the problem of fraudulent claims in the insurance industry and the limitations of traditional fraud detection techniques [2]. 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 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 demonstrates exceptional performance in determining the veracity of customer claims, with an exceptional accuracy of 97.176%. Also the values of specificity and sensitivity are exceptionally high. 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.

R. Roy and K. T. George introduced a paper on "Detecting insurance claims fraud using machine learning techniques," [4] that 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.

Urunkar et al proposed a machine learning approach that can analyze large amounts of data and identify patterns indicative of fraud [5]. 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 imply that defining the perfect algorithmic methods or implementing feature engineering techniques for improved performance might not be feasible due to the unique traits of different datasets. Their suggestion involves utilizing models tailored to specific business contexts and user preferences. This approach allows loss management units to concentrate on identifying new instances of fraud and ensuring that the models continuously adapt to detect them.

Sun et al introduced a novel approach, named Patient Cluster Divergence-based Healthcare Insurance Fraudster Detection (PCDHIFD) [7], designed to address the issue of camouflage responses in healthcare insurance fraud detection. To conduct their experiments, they utilized a substantial healthcare dataset consisting of around 40 million admission records from 10,000 patients over a five-year period. The methodology consisted of three key steps: first, the construction of a patient graph based on the most similar information at the patient level; second, the application of a clustering-based graph algorithm to identify significant clusters; and finally, the calculation of patient cluster divergence to determine the probability of fraud for each patient. This approach provides a comprehensive and effective means of detecting potential fraudsters within the healthcare insurance system, taking into account patient histories, diagnoses, and medical practitioner data.

III System Design

Fig 1 shows 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 [10]. 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.

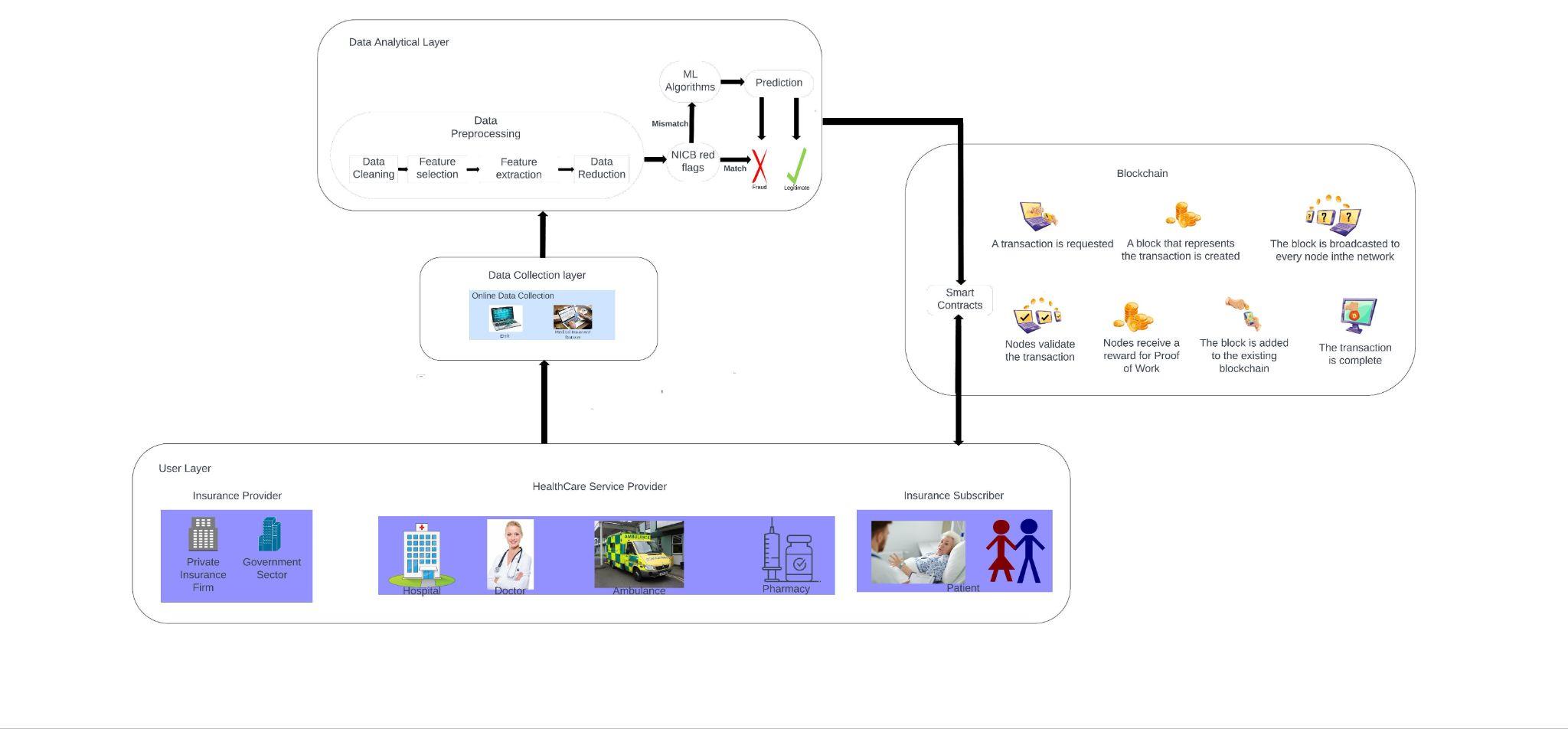


Fig 1 – System Architecture

1. *A. Dataset Introduction*

This project uses eight data files [8] 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:

* Inpatient Data offers valuable information on the insurance claims submitted for individuals who have been admitted to hospitals. It includes additional details such as admission and discharge dates, as well as the admit diagnosis code.
* Outpatient Data provides information on insurance claims for patients who visit hospitals but are not admitted. It captures relevant details associated with their hospital visits.
* Beneficiary Details Data encompasses KYC details of beneficiaries, including health conditions and the region they are affiliated with.
* Provider Data 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.

*B. Exploratory Data Analysis*

Checking out data, also called Exploratory Data Analysis (EDA), is like the detective work of data analysis. It's where you dig into your datasets at the start to figure out what's going on, find interesting stuff, and guess some ideas. Think of it as getting to know your data before you jump into more complicated math or computer stuff. EDA uses simple summaries, pictures, and graphs to uncover connections, trends, and things that seem a bit strange in the data. By doing this detective work, analysts can make smart choices about what to do next with their analysis, fix any data issues, and decide how to build their models.

1. *Provider Dataset*

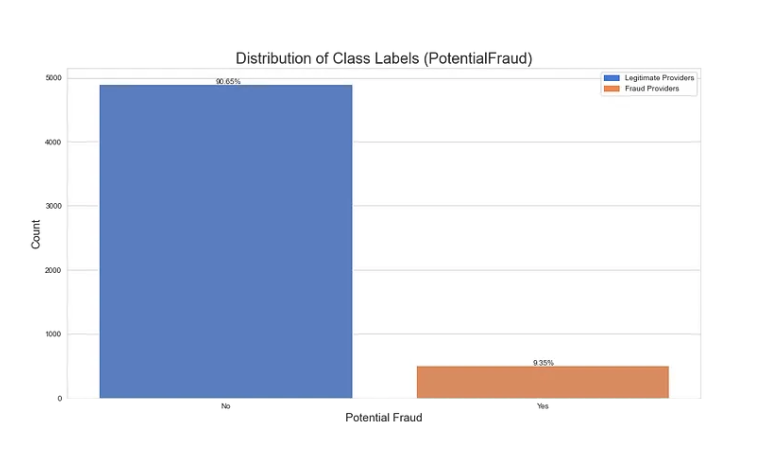


Fig 2 – Distribution of Class Labels

Fig 2 shows the distribution of class labels in the given dataset and it indicates that the dataset is hugely imbalanced when classified based on the training data [9]

1. *Beneficiary Dataset*

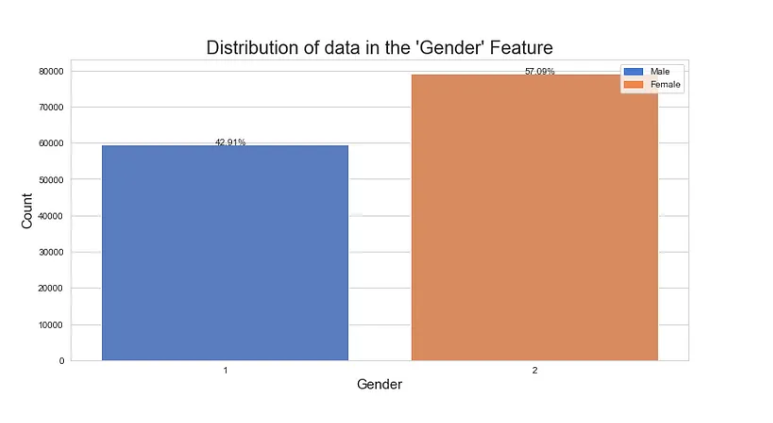


Fig 3 – Distribution of data in the ‘Gender’ Feature

Fig 3 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.

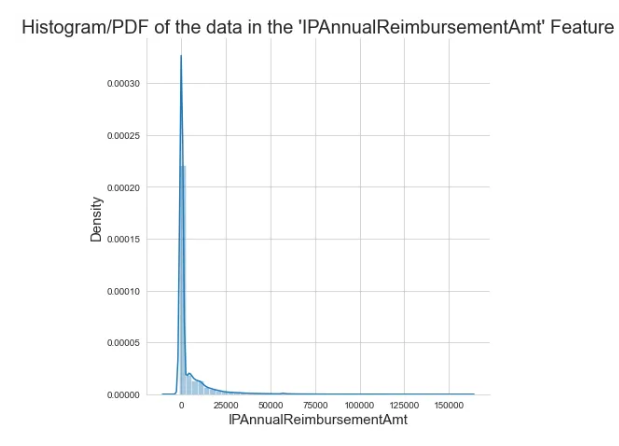


Fig 4 – Histogram of the data in the IP Annual reimbursement Amount

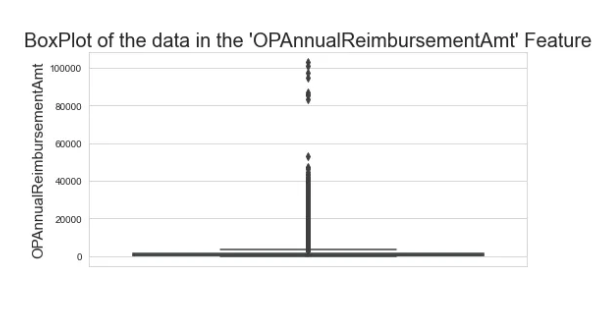


Fig 5 – Histogram of the data in the OP Annual reimbursement Amount

For Inpatient (Fig 4) and Outpatient (Fig 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.

1. *Data Preprocessing*
2. *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'.

1. *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.

1. *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' [11].

IV. Implementation

1. *Feature Selection*

The given dataset is then gone through feature selection to find if any features could be eliminated. To eliminate features there should be a valid substantiating parameter to justify. Hence three different feature selection methods which are widely used for categorical features are experimented and the results are compared to find the best method.

1. *a. Chi-Squared Feature Selection*

Chi-squared feature selection is a statistical technique used in machine learning and data analysis to identify and select the most appropriate features from a dataset, especially when dealing with categorical or discrete data. It measures the statistical independence between each feature and a target variable which is potential fraud, and constructs contingency tables to tabulate the relationships between feature categories and the target variable. This method is performed for the given dataset and the results are tabulated. The features are considered closely associated with the target variable ‘potential fraud’ when the chi-squared score is high and vice versa. The advantages of this method are dimensionality reduction, model interpretability, and improving predictive accuracy in classification tasks.

*b. Mutual Information*

Mutual Information (MI) is yet another metric that correlates the interconnection between data attributes. In this case, the independent variables, and the target variable are plotted to determine the associativity. High MI scores indicate the value of attributes in providing insights into the target variable. Further, the Reduction in Entropy can be calculated as:

Reduction in Entropy = Total Entropy - Entropy of Target Variable

Total Entropy H(Y) is given by,

.…(1)

Where,

p0 -> proportion of samples in class 0

p1 -> proportion of samples in class 1

And the Entropy of target variable is calculated as follows,

.…(2)

Where,

P(X = x) -> probability of feature having value x

H( Y | X = x) -> conditional entropy of target variable given that the feature has value x.

*c. Spearman’s Correlation Coefficient*

Spearman's correlation coefficient (ρ) is a statistical tool used to measure the connection between variables. It involves assigning ranks to each variable's values individually and determining the disparities in these ranks for each pair of data points. This helps assess the strength and direction of the monotonic relationship between the two variables.

To calculate ρ, rank the values of each variable separately, calculate the differences between the ranks for each pair of data points, square each difference, and then calculate the Spearman's correlation coefficient. The range of ρ values ranges from -1 to 1, with -1 indicating a perfect negative relationship, 1 indicating a perfect positive relationship, and 0 indicating no particular relationship.

In real-world data, most ρ values fall between -1 and 1, with closer values indicating stronger correlation between variables. This tool is often used when data is ordinal or when a linear relationship is not appropriate.

We notice that some features exhibit strong Spearman Correlation Coefficients.

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.

1. *Encoding*

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.

1. *Data Sampling*

An imbalanced dataset is a situation where a minority class has fewer examples than the majority class. To address this, techniques like balanced class weights, under-sampling, and over-sampling are used to ensure fairness and effective learning from all classes. Each of these techniques is applied sequentially after standardization and before fitting through the model using pipelines [18].

### *Balanced Class Weights*

In classification tasks with imbalanced class distributions, balanced class weights come into play. The purpose of this method is to avoid the model ending up favoring the majority class and performing poorly on the minority class. Balanced class weights solve this by assigning varying importance to the classes, giving extra weight to the minority group. Hence the model promotes a fairer and more accurate outcome by learning from both classes equally during training.

### *Under-Sampling*

Under-sampling is performed when the data belonging to the majority class exceeds data associated with minority exceedingly. This approach helps prevent the model from getting swamped by the majority class. In this method, random samples from the majority class are taken until it equalizes the minority class samples. Now the dataset with balanced distribution of samples to both classes are fed into the model.

### *Over-Sampling*

Over-sampling is yet another method of balancing the dataset with respect to samples in either of the classes. It is similar to under-sampling but the difference is here the minority class samples are populated to even up the majority class samples. It involves creating synthetic samples for the minority class using techniques like SMOTE (Synthetic Minority Over-sampling Technique) [17]. This increases the number of instances in the minority class.

1. *Model Training*

The data under various sampling methods is then trained on four machine learning algorithms namely Logistic Regression, Decision Tree, Random Forest and eXtreme Gradient Boost. The performance of four algorithms are discussed below and the results are summarized.

*a. Decision Tree*

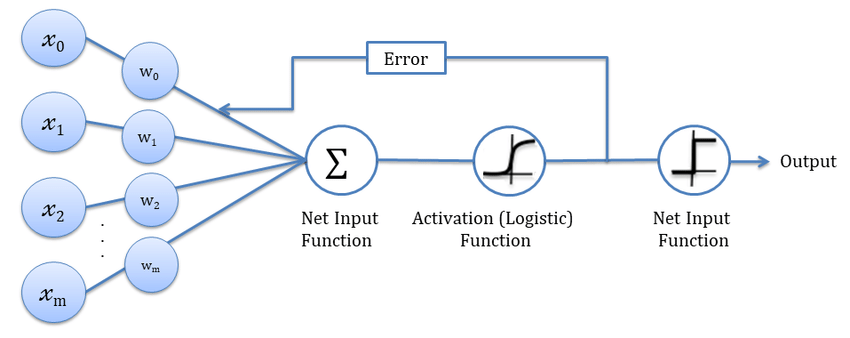
Decision tree modeling is a simple but effective machine learning algorithm for both classifying and predicting data. Its main objective is to partition the dataset into subsets that are as similar in a certain feature relative to the target variable [14]. This division process keeps repeating, forming a tree structure that guides predictions, moving from the starting point (the root node) down to a specific endpoint (a leaf node). The nitty-gritty of how this works will be discussed in the upcoming sections. What's interesting is that decision tree algorithms excel at figuring out which features are crucial for classification, making feature selection an inherent part of their operation. This feature-selection process can be fine-tuned using different approaches, such as the SSV criterion. [9].

Evaluation of Decision Tree Classifier is performed using,

1. Best Hyperparameters on Response Encoded Data without Sampling
2. Best hyperparameters on Response Encoded Data with balanced Class Weights
3. Best hyperparameters on Response Encoded Data with Random Undersampling
4. Hyperparameter tuning on Response Encoded Data with SMOTE Oversampling

## *b. Logistic Regression*

Logistic regression is a statistical model used for binary classification, which is widely used in datasets whose relationship is non-linear. This model uses logistic (sigmoid) function to transform the linear combination of features by calculating the probability of the target variable belonging to a particular class [13]. The feature coefficients are learned from the labeled data and the best fit is observed. It enables understanding the influence of individual features on the prediction, improving the interpretability of the model.



## *c. Random Forest*

Random Forest machine learning algorithm ensemble method that handles classification and regression tasks. A strong model is built using multiple decision trees [15]. The overfitting issues in individual decision trees is addressed in this model and the model's ability to make generalizable predictions is enhanced. "Bootstrapping" is used for data sampling . It reduces the risk of overfitting. A random subset of features is selected to split in a decision tree. Each decision tree is constructed using a variant of the decision tree algorithm, such as CART. There are two types of tasks namely, classification tasks and regression tasks. In classification tasks, the results are combined through a majority vote, while in regression tasks, predictions from different trees are averaged.

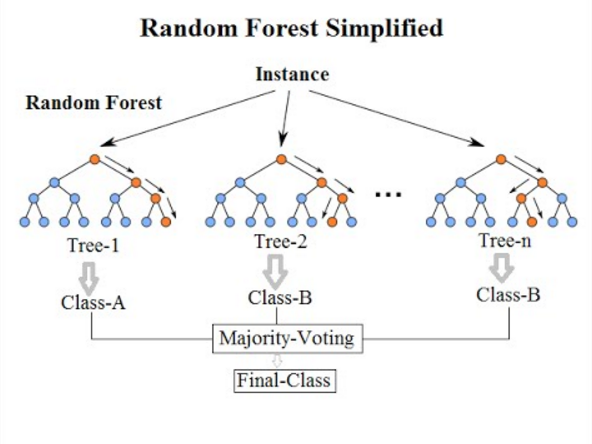


Fig 6 – Pictorial Representation of Random Forest

## *d. Xgboost*

XGBoost, an abbreviation for Extreme Gradient Boosting, represents a robust and widely favored machine learning technique recognized for its remarkable capabilities across diverse data-centric assignments [12]. Categorized within gradient boosting algorithms, it stands out for its proficiency and success in managing structured data sets. 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 modeling and competitive data science competitions.

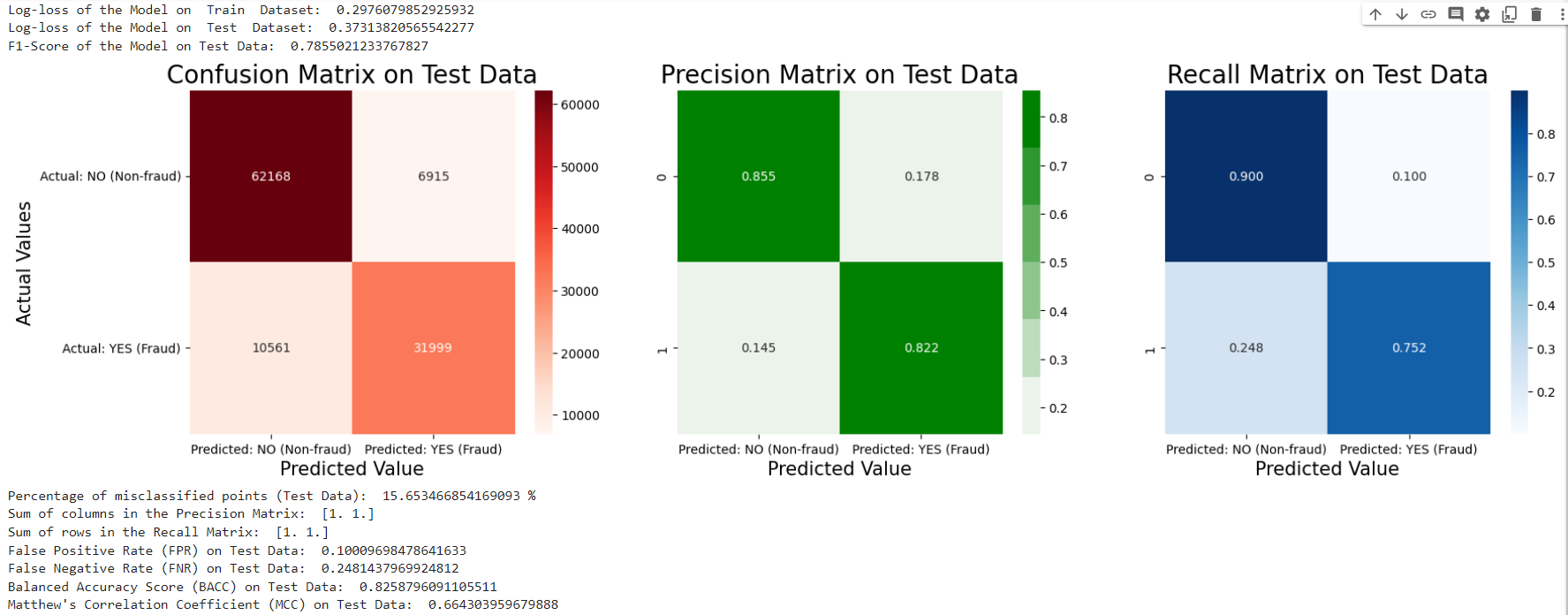
1. *Training the Model with best hyperparameters on Response Encoded Data with SMOTE Oversampling*

Fig 7 – Performance Metrics - XGBoost with SMOTE Oversampling

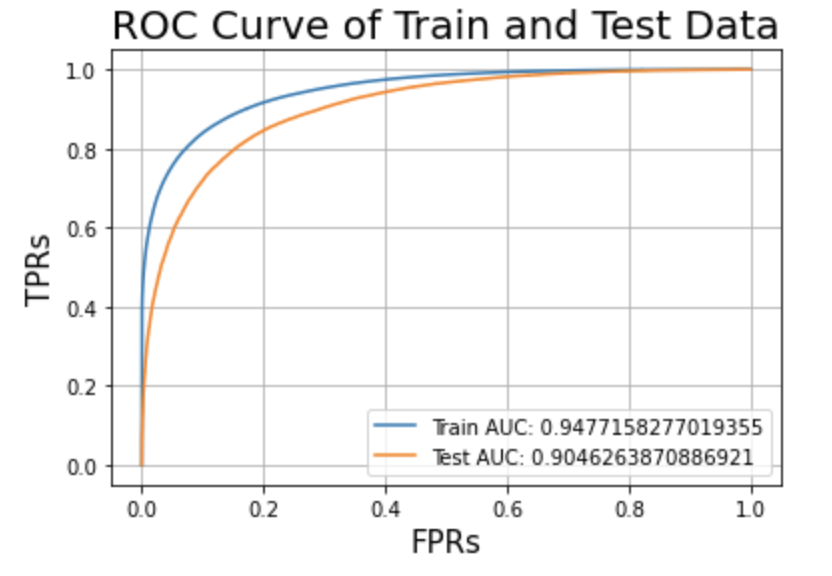


Fig 8 – ROC Curve - XGBoost with SMOTE Oversampling

## 

## 

## 

## *C. Best Model Selection*

| Model | Train Data | |
| --- | --- | --- |
| AUC | Log-loss |
| LR | 0.74094 | 0.57481 |
| LR - BCW | 0.74139 | 0.59742 |
| LR - RU | 0.74152 | 0.5988 |
| LR - SMOTE | 0.79221 | 0.54082 |
| DT | 0.86694 | 0.43362 |
| DT - BCW | 0.8656 | 0.45284 |
| DT - RU | 0.86093 | 0.45972 |
| DT - SMOTE | 0.89373 | 0.40121 |
| RF | 0.89265 | 0.40513 |
| RF - BCW | 0.97359 | 0.27579 |
| RF - RU | 0.97355 | 0.28446 |
| RF - SMOTE | 0.94208 | 0.32997 |
| XGB | 0.92598 | 0.36781 |
| RF - RU | 0.92511 | 0.36703 |
| RF - SMOTE | 0.94222 | 0.35101 |

| Model | TEST DATA | | | |
| --- | --- | --- | --- | --- |
| Log-loss | AUC | F1 score | % misclassification |
| LR | 0.57316 | 0.74464 | 0.54777 | 29.59 |
| LR - BCW | 0.5965 | 0.74497 | 0.62397 | 31.29 |
| LR - RU | 0.59831 | 0.74483 | 0.62432 | 31.49 |
| LR - SMOTE | 0.60103 | 0.71757 | 0.57263 | 31.49 |
| DT | 0.62828 | 0.83325 | 0.68552 | 22.56 |
| DT - BCW | 0.63311 | 0.83151 | 0.70404 | 24.1 |
| DT - RU | 0.64241 | 0.85882 | 0.69991 | 24.61 |
| DT - SMOTE | 0.49893 | 0.85931 | 0.70691 | 21.83 |
| RF | 0.47671 | 0.85847 | 0.69996 | 21.74 |
| RF - BCW | 0.48681 | 0.83704 | 0.70570 | 22.93 |
| RF - RU | 0.49762 | 0.82412 | 0.72196 | 23.72 |
| RF - SMOTE | 0.38447 | 0.90323 | 0.76912 | 17.42 |
| XGB | 0.46366 | 0.84323 | 0.69563 | 22.74 |
| XGB- RU | 0.46962 | 0.84257 | 0.71539 | 22.9 |
| RF - SMOTE | 0.39356 | 0.90463 | 0.76733 | 16.81 |

| Model | TEST DATA | | | |
| --- | --- | --- | --- | --- |
| FPR | FNR | BACC | MCC |
| LR | 0.15316 | 0.52982 | 0.65951 | 0.32963 |
| LR - BCW | 0.30928 | 0.31988 | 0.65981 | 0.36923 |
| LR - RU | 0.31590 | 0.31532 | 0.68533 | 0.36381 |
| LR - SMOTE | 0.23993 | 0.44527 | 0.65785 | 0.32107 |
| DT | 0.14606 | 0.35484 | 0.74595 | 0.51308 |
| DT - BCW | 0.23682 | 0.24741 | 0.75604 | 0.50407 |
| DT - RU | 0.22995 | 0.26201 | 0.75401 | 0.49935 |
| DT - SMOTE | 0.16501 | 0.30479 | 0.76734 | 0.53118 |
| RF | 0.13582 | 0.34274 | 0.76072 | 0.53673 |
| RF - BCW | 0.18353 | 0.28954 | 0.76346 | 0.52609 |
| RF - RU | 0.22632 | 0.25297 | 0.76535 | 0.52084 |
| RF - SMOTE | 0.12422 | 0.26005 | 0.80785 | 0.62339 |
| XGB | 0.13828 | 0.34523 | 0.75798 | 0.53072 |
| XGB- RU | 0.21905 | 0.24508 | 0.76793 | 0.52764 |
| XGB- SMOTE | 0.10379 | 0.27263 | 0.81179 | 0.63867 |

The table uses the following abbreviations - Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), XGBoost (XGB) - the machine learning models paired with - Balanced Class Weights (BCW), Random Undersampling (RU), Synthetic Minority Oversampling Technique (SMOTE) - which is assessed through performance metrics like - Area Under the Curve (AUC), False Positive Rate (FPR), False Negative Rate (FNR), Balanced Accuracy (BACC) and Mathews correlation coefficient (MCC).

On observing the scores of various models (Fig 7.6), it is inferred that the Decision Tree model, Random Forest model, and XGBoost model all showed low false negative rates and high scores on evaluation metrics. The XGBoost model with Synthetic Minority Oversampling Technique achieved the best scores for most criteria, making it the final model.

*D.* *Model Interpretability*

Explainable AI (XAI) is a concept which includes a set of processes used to perceive insights from the decisions generated by AI and machine learning models in a human-understandable form.

LIME (Local Interpretable Model-agnostic Explanations) is one such concept which is used to understand how confident the final XGBoost model classifies an input [16]. It focuses on a local group of data points and it is applicable to any machine learning model as it treats the model as black box. LIME takes a datapoint and produces samples around the datapoint. It then uses RBF to generate weights for the points generated. The points closer to the original data are assigned a greater weight compared to the points distant.

1. *Datapoint with Positive Prediction*

LIME takes the first data point and predicts it as positive based on its Linear Ridge Regression with a confidence of 89% (Fig 8.1). The values of ‘IsSamePhysMultiRole1’ and ‘PhysRoleCount’ increases the claims chances to be classified as fraudulent. Other values such as ‘PHY412132’, ‘TreatmentDuration’, ‘ClmProcCode\_9904’, ‘PHY337425’, ‘ClmProcCode\_8154’, ‘TotalClaimAmount’, ‘ClmProcedureCode\_2’, ‘PHY330576’ decrease the chance a bit for the claim to be classified as fraudulent.

1. *Datapoint with Negative Prediction*

LIME takes the zeroth data point and predicts it as negative based on its Linear Ridge Regression with a confidence of 79% (Fig 8.2). The values of ‘IsSamePhysMultiRole1’ and ‘PhysRoleCount’ increases the claims chances to be classified as non-fraudulent. Other values such as ‘PHY412132’, ‘TreatmentDuration’, ‘ClmProcCode\_9904’, ‘PHY337425’, ‘ClmProcCode\_8154’, ‘TotalClaimAmount’, ‘ClmProcedureCode\_2’, ‘PHY330576’ decrease the chance a bit for the claim to be classified as non-fraudulent.

V. Conclusion

In conclusion, this project successfully adopts a machine learning model for the detection of fake insurance claims, a critical issue in the insurance industry. By analyzing the previous solutions for this problem it is inferred that wide feature engineering and optimal feature selection can improve the model’s performance. The system model is developed and detailed with a sequence of sub-phases in each phase. Exploratory Data Analysis helped to identify the non-contributing features and exposed the imbalanced nature in the dataset.

Data preprocessing techniques like data cleaning and standardization tuned the dataset towards learning. Three different feature selection techniques namely Spearman correlation, Mutual information and Chi-squared feature selection were attempted and spearman correlation was found to elucidate the relationship between the features and target variable. The imbalance in the dataset is moderated using balanced class weights, undersampling and oversampling. The data was trained with machine learning models like decision trees, random forests, logistic regression, and XGBoost. Different encoding techniques like response encoding and one-hot encoding were helped to ease the model’s training process.The models were assessed based on real-world performance metrics like accuracy, precision, recall, and F1-score. The results provided valuable insights, helping in the understanding of when each method shines and where it might fall short.

However, it became evident that the XGBoost algorithm with response encoding, when combined with the Spearman correlation and oversampling using SMOTE, outperformed the competition in terms of overall accuracy with the following values for the chosen performance metrics : Log loss - 0.39356 , Misclassification percentage - 16.81%, FPR - 0.10379, AUC - 0.90463, F1 score - 0.76733, BACC - 0.81179, MCC - 0.63867.

This model will greatly benefit the insurance industry, enhancing its ability to combat fraudulent activities and protect the interests of both insurers and policyholders. Furthermore, this research underscores the importance of selecting the right combination of machine learning algorithms and feature selection techniques to achieve optimal results in complex, real-world problem-solving scenarios.

References

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. 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.
6. 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) [6]
7. 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>
8. <https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis>
9. <https://www.kaggle.com/datasets/arashnic/imbalanced-data-practice>
10. [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%20)
11. 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.
12. <https://xgboost.readthedocs.io/en/stable/python/python_intro.html>
13. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>
14. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
15. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
16. <https://towardsdatascience.com/lime-explain-machine-learning-predictions-af8f18189bfe>
17. <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>
18. <https://scikit-learn.org/0.16/modules/generated/sklearn.pipeline.Pipeline.html#:~:text=The%20purpose%20of%20the%20pipeline,as%20in%20the%20example%20below.>