INFS_SP5_2023
Predictive Analytics
Assignment 2
Building a decision tree predictive model

Huining Huang: huahy057@mymail.unisa.edu.au
Lingjun Ji: Jiyly006@mymail.unisa.edu.au
Grahi Nileshkumar Brahmbhatt: bragy016@mymail.unisa.edu.au

Contents

Part 1: Introduction 2
Part 2: Related Work 2
Part 3: Data exploration 3
Part 4: Building decision tree models 10
References: 25
Appendix 26

Part 1: Introduction

Healthcare insurance is crucial for citizens' access to quality healthcare and national well-being. But it's threatened by healthcare insurance fraud, which involves deceptive activities among medical providers, patients, and insurance companies, posing a significant challenge in the healthcare sector. Insurance companies are frequently on the receiving end of these bad practices, which in turn has caused them to hike the prices of their insurance premiums, making healthcare costs surge periodically.[1] It is pretty evident that patients and their healthcare information can easily be exploited which later can hamper the overall cost.[1] Healthcare fraud has far-reaching consequences. It jeopardizes the healthcare system's stability, erodes patient trust in insurance, and can lead to unnecessary expenses and health risks from unnecessary treatments. It also poses financial risks to insurance companies and damages the reputation of all healthcare providers.

To detect and avoid the fraud, data mining techniques are applied.[2] Data mining aids in uncovering fraud patterns and unusual behavior in extensive medical data, helping the healthcare system spot unnecessary procedures and ensure appropriate care for patients. Predictive models using historical data can also identify potential fraudulent providers, enabling early detection and prevention of medical insurance fraud. This leads to cost reduction for insurance companies and, in turn, lower insurance premiums.

In this report, we will examine academic papers on healthcare technologies and algorithms to improve our approach. Also, we will leverage the dataset we explored in our previous work, containing medical insurance claims data with provider details, billing information, patient records, procedure data, and fraud indicators. Our primary goal is to extract meaningful features from this dataset and create a predictive decision tree model for identifying potential fraudulent providers.

Part 2: Related Work

In our earlier report, we reviewed several academic papers that explored healthcare fraud prediction from various angles. Incorporating the feedback, we received and reflecting on these papers has led to new insights and understandings. The feedback on our previous work encouraged us to enhance our feature engineering efforts. We realized that careful feature selection and construction are critical in healthcare fraud detection. This insight has reinforced our focus on designing and selecting informative features to enhance the accuracy of our predictive model.

First, the paper titled "Healthcare Provider Summary Data for Fraud Classification"[3] by J. M. Johnson emphasizes the critical role of feature engineering and dataset construction in healthcare fraud detection. The author leverages the latest publicly available data from CMS and introduces two new labeled Medicare Part B datasets for supervised learning. Their research demonstrates that, through careful selection and construction of the SbP feature set, significant improvements can be achieved in the performance of practical healthcare fraud detection models. This is of paramount importance to our project as we need to carefully consider how to design and select the most informative features to enhance the accuracy of our model.

Furthermore, the paper titled "A Comparative Analysis of Fraud Detection in Healthcare using Data Balancing & Machine Learning Technique"[4] by Nikita Agrawal et al. The paper's results indicate the superior performance of two data balancing techniques, namely Class Weighing Scheme (CWS) and Adaptive Synthetic Oversampling (ADASYN), in handling imbalanced datasets. This prompted us to consider the use of data balancing techniques to address the data imbalance issue. This approach is crucial for improving model performance metrics and mitigating the impact of skewed data distribution.

Additionally, the paper "Predicting health insurance claim frauds using supervised machine learning technique"[5] by Veena K et al. The paper emphasizes the high accuracy of the decision tree classifier, surpassing the performance of the other three algorithms it was compared to: logistic regression, random forest, and naive Bayes. We have decided to incorporate this algorithm into our model to leverage its exceptional accuracy in identifying potentially fraudulent activities.

Through the feedback and insights gained from these papers, we have gained insights into the pivotal roles played by feature engineering, data balancing and decision tree classifiers techniques in healthcare fraud prediction. These insights will guide us in formulating sound data preprocessing, feature extraction plans and methodological strategies for our project, ensuring that our project is better aligned with the challenges of healthcare fraud prediction.

Part 3: Data exploration

1. Introduction to the Data and Variables:

The data revolves around medical claims, shedding light on potential fraudulent activities within the healthcare sector. Diving deeper, the data is segmented into:

- Inpatient Data: Comprehensive insights about claims associated with patients admitted to hospitals. Key Features: Admission and discharge dates (which helps in determining the length of a patient's stay), Diagnosis codes (indicating the medical reasons behind hospitalizations).

- Outpatient Data: This dataset pertains to patients who avail medical care but do not stay admitted in the hospital for long durations. Key Features: Data indicating the types and frequencies of outpatient services availed.

- Beneficiary Details Data: This dataset encapsulates KYC (Know Your Customer) details pertaining to beneficiaries, offering a glimpse into their medical history and affiliations. Key Features: Health conditions (showcasing the medical history or the current health status of beneficiaries), Regional affiliations (offering a geographical overview which can be pivotal for region-specific analyses).

- Provider Potential Fraud Data: This is the cornerstone for our fraud detection analysis. It maps healthcare providers to potential fraudulent activities with a binary distinction - 'Yes' for potential fraud and 'No' for non-fraudulent.

The dataset comprises 138,556 beneficiaries, 558,211 claims, and 5,410 providers. On average, each beneficiary has around 4.029 claims. Filtering beyond the 99th percentile reveals beneficiaries and providers with notably high claim counts. The Cumulative Density Plot shows a key transition zone, hinting at a rise in unusual claims. Providers with claims exceeding 4,000 require deeper scrutiny due to their deviation from typical patterns.

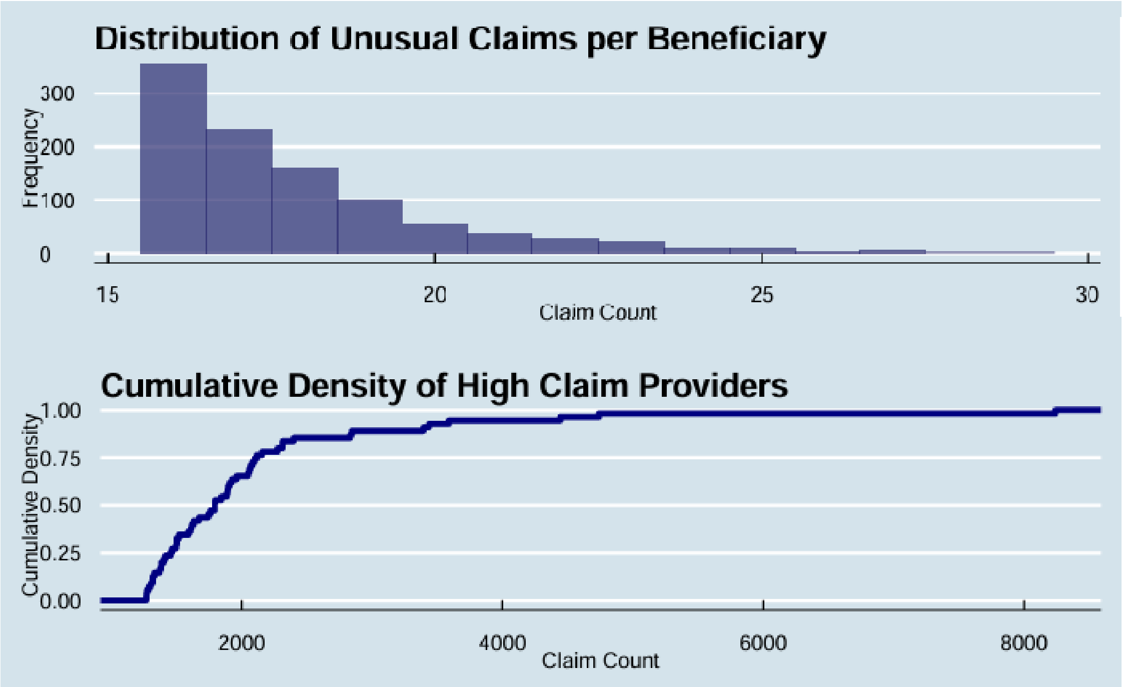


Figure 1: Distribution of Unusual Claims and Cumulative Density of High Claim Providers.

2. Merging Datasets and Data Cleaning:

To create a holistic dataset conducive for rigorous analysis, we integrated data from the aforementioned four distinct datasets. This amalgamation resulted in a dataset with 558,211 records spanned across 54 features. Key identifiers like ‘ClaimID’ and ‘BeneID’ were retained for granularity.

Certain columns underwent transformations for consistency:

- ‘Gender’ column was binarized.
- ‘RenalDiseaseIndicator’ was transformed such that ‘Y’ indicated the presence of renal disease.
- Null values in the ‘DeductibleAmtPaid’ column were imputed with ‘0’.
- In the quest for fraud detection, the ‘PotentialFraud’ column was binarized where ‘Yes’ indicated potential fraud.

3. Addressing Missing Data:

Understanding and navigating through missing data is paramount for any analytical endeavor.

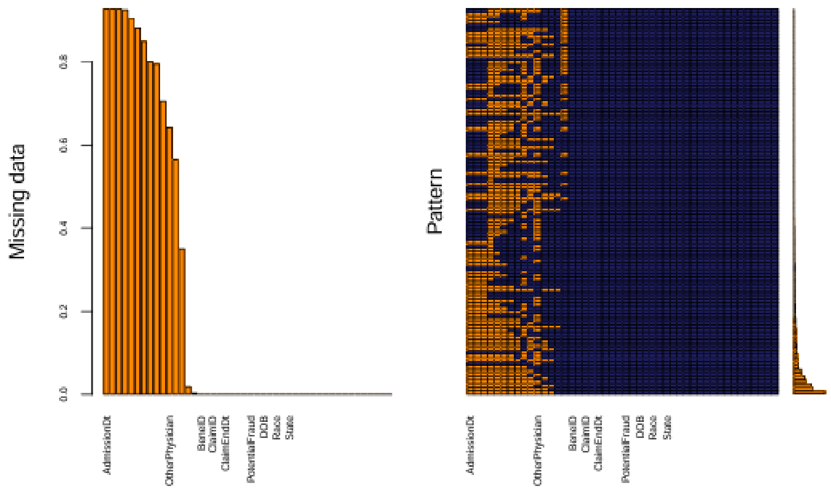


Figure 2: Missing Data Heatmap.

• Columns like ‘AdmissionDt’ and ‘DischargeDt’ had substantial missing data (~92.75%). However, these columns are pivotal for computing the ‘length of stay’ feature, a significant metric for fraud detection.

• For the columns ‘ClmDiagnosisCode_2-10’ that encapsulate a myriad of medical conditions, missing values varied. It’s paramount here to judiciously apply domain knowledge to ascertain their inclusion or exclusion from the analysis.

• Columns with missing values above 30% were extricated from the dataset. Those with missing values below this threshold were retained for imputation.

Procedure for Missing Data Imputation: Considering the non-random nature of the missing data, the Multiple Imputation by Chained Equations (MICE) technique was chosen. This method, leveraging other variables within the dataset, is adept at predicting missing values. Noteworthy parameters for this imputation included: • m=5 for generating five imputed datasets.

- maxit = 50 indicating the number of iterative rounds.

- Adoption of the mean matching method for imputation.

1. Exploratory data analysis

The bar plot showcases the distribution of potential fraudulent claims within our dataset. A clear class imbalance is evident, with a vast majority of claims labeled as non-fraudulent (represented by ‘0’). Such imbalances are pivotal to recognize as they can influence modeling outcomes, potentially skewing predictive accuracy on unseen data. Further investigation could delve into understanding the underlying features that contribute to the minority fraudulent class.

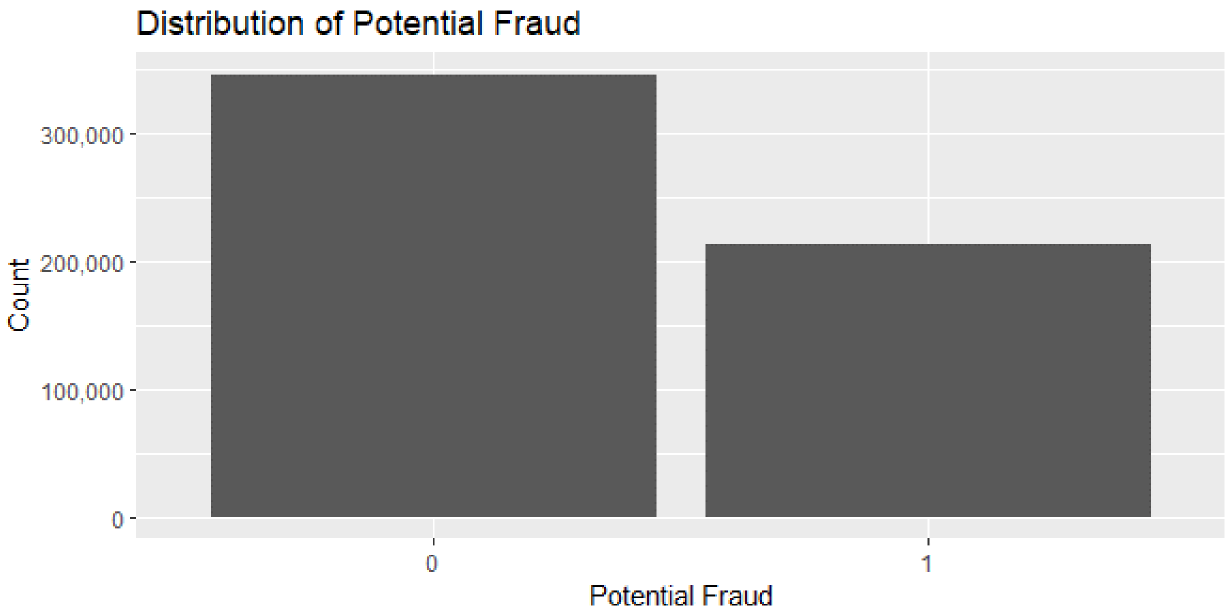


Figure3: Distribution of Potential Fraudulent Claims.

This graph highlights the gender and racial distributions within our dataset. While there's a slight preponderance of one gender over the other, the racial distribution is dominated by one particular category. Delving deeper, one could explore if fraudulent claims are more prevalent within a specific gender or racial group. Such insights can guide targeted fraud detection strategies, ensuring that potential biases are minimized.

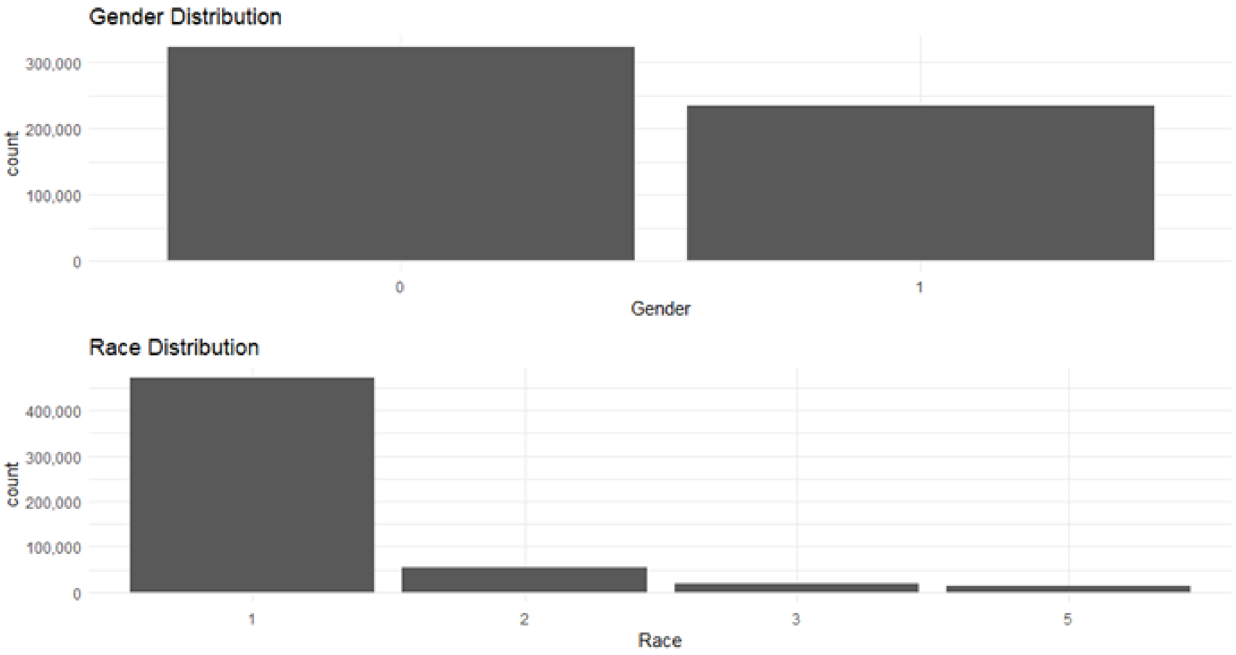


Figure4: Distribution of the gender and racial groups.

The bar chart provides a glimpse into the frequency of various chronic conditions among beneficiaries. 'IschemicHeart' and 'Diabetes' emerge as predominant conditions. However, it would be insightful to investigate if any specific chronic condition is more associated with fraudulent claims. Such findings can be instrumental in healthcare planning, as they can hint at potentially suspicious patterns of claim submissions.

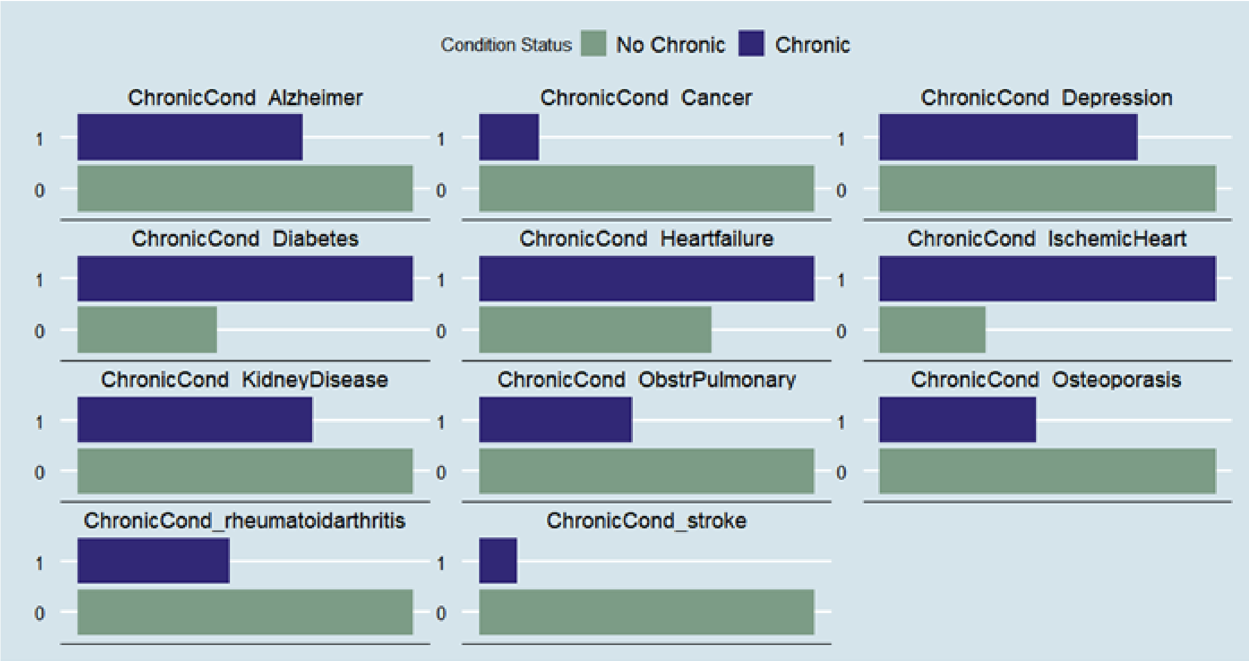


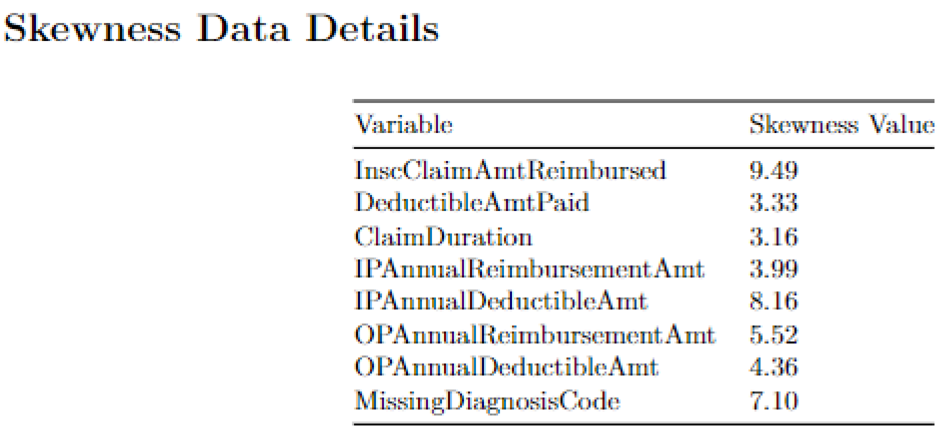
Figure 5: Distribution of Chronic Conditions.

1. Feature Engineering:

Feature engineering is the backbone of any machine learning project, transforming raw data into insightful features that can significantly improve model performance.

• Temporal Features & Medical Conditions: Features like 'Age at the Time of Claim' and 'Claim Processing Time' were derived. A consolidated feature was also created to encapsulate the severity of chronic conditions, offering an in-depth view of patient health over time. • Features Transformation: Given the skewness and kurtosis in the dataset, transformations, predominantly logarithmic, were used. This not only addressed skewness but also mitigated outlier impact.

Table 1: skewness of the features.



Skewness Data Details

Variable	Skewness Value
InscClaimAmtReimbursed	9.49
DeductibleAmtPaid	3.33
ClaimDuration	3.16
IPAnnualReimbursementAmt	3.99
IPAnnualDeductibleAmt	8.16
OPAnnualReimbursementAmt	5.52
OPAnnualDeductibleAmt	4.36
MissingDiagnosisCode	7.10

6. Feature Selection:

The feature selection was data-driven, leveraging the mlr3 package. Based on information gain criteria, features like ‘Provider’, and ‘Attending Physician’ emerged as paramount predictors. Conversely, features like ‘ChronicCond__Cancer’, ‘ChronicCond__Depression’, and ‘Gender’ demonstrated minimal influence on model prediction. Moving forward, our strategy is meticulously crafted, focusing on features with high information gain, ensuring the model’s robustness and accuracy.

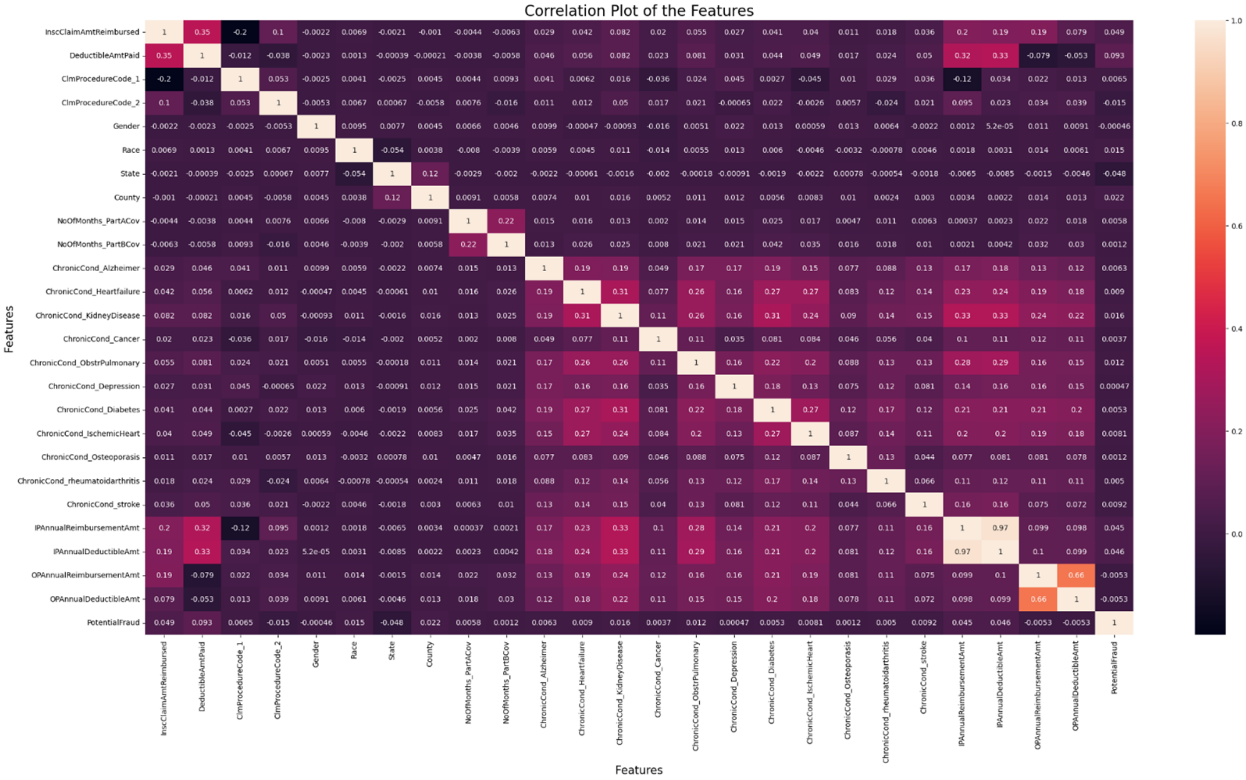


Figure6 : Heatmap of Correlation Matrix used for Feature Selection.

Table 2: Features Selected for Model Building.

Feature Name	Description & Reason for Selection
AdmitCode	Binary variable indicating if a claim admission diagnosis code is present. Helpful in identifying the primary reason for medical services.
DiagnosisCode_*	Binary variables representing different claim diagnosis codes. Provide insights into types of medical conditions and may associate with fraudulent claims.
Gender	Binary representation of gender. Might affect medical service utilization patterns.
Race	Categorical variable for racial categories. Different racial groups might have different medical needs.
DiseaseIndicator	Indicates presence or absence of renal disease. Patients with this might have specific medical service utilization patterns.
State, County	Represent geographical locations. Geographical patterns might emerge in fraudulent activities.
ChronicCond_*	Series of binary variables for various chronic conditions. Influence type and frequency of required medical services.
PotentialFraud	Binary variable for potential fraud. Target variable for a fraud detection task.
Age	Represents age of beneficiaries. Different age groups might have different utilization patterns.
Admission	Indicates if admission was during a weekend. Might see different types of claims or emergencies.
IsDead	Binary variable indicating if beneficiary is deceased. Might influence kind of services rendered.
ClaimDelay	Categorical variable for claim settlement delay. Delays might indicate complications or disputes.
Duration	Duration of treatment. Longer durations might associate with more severe conditions.
ClaimAmount, TotalAmount	Amounts related to claims. High or unusual amounts might be red flags for fraudulent activities.
Log_*	Log-transformed values of features to normalize skewed distributions.
UniquePhys, PhysRole	Numeric values for count of unique physicians and their roles. Multiple roles or frequent changes might be suspicious.
SamePhys	Indicates if the same physician plays multiple roles. Might indicate a lack of specialization or potential fraud.
PHY*	Binary variables for specific physician identifications. Specific physicians might associate with specific types of claims.

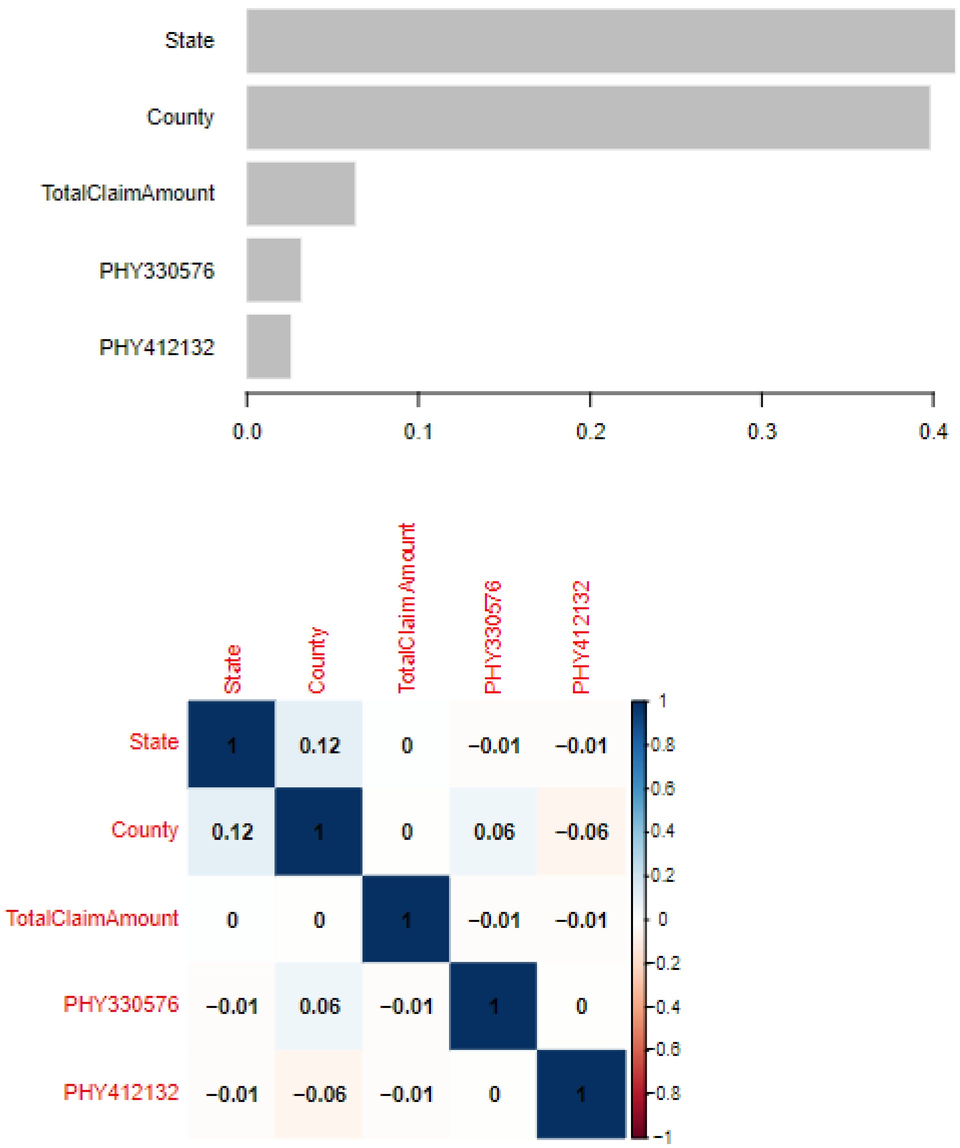


Figure 5: The top 5 features with the highest information gain and a Heatmap showcasing the correlation matrix to make sure the features are not highly correlated.

Part 4: Building decision tree models

To prevent medical fraud, we employ a decision tree model to detect anomalies in the data. we set the random seed to 1000 ensures that running the code multiple times will produce the same results. Given the relatively large dataset of 558,211 data points, careful consideration is given to the data partitioning ratios to balance the requirements of model training and performance evaluation. After careful consideration, we have adopted the following partitioning scheme: 80% of the data is allocated to the training set to ensure thorough model training, 10% is allocated to the validation set for parameter tuning, and the remaining 10% is reserved for the test set for final performance evaluation. This partitioning strategy maximizes the utilization of the extensive dataset, ensuring comprehensive model training and reasonable performance assessment.

To achieve a better decision tree model. we fine-tune the model by adjusting three key parameters. The cp parameter is a crucial hyperparameter in decision trees. It controls the size of the tree by penalizing its complexity. A higher cp value leads to a smaller tree, while a lower cp value leads to a larger tree. The minsplit and maxdepth parameters also affect the size of the tree. `minsplit` sets the minimum number of observations that must exist in a node for a split to be attempted,`maxdepth` restricts the maximum depth of any node of the final tree, essentially limiting the number of splits that can happen in any decision path from the root node to a terminal node. A higher minsplit value leads to a smaller tree, while a higher maxdepth value leads to a larger tree. We employ these parameter adjustments to generate four distinct decision tree models for thorough comparison and to enhance the model's overall performance.

Table 3: Decision Tree Model Parameters Setting.

Model	CP	Minsplit	Maxdepth
Baseline(Full grow)	0.00	/	/
Model1	0.01	20	30
Model2	0.001	20	8
Model3	0.001	5	8

Pivot Study and Comparative Analysis

In the pivot study, the model's complexity parameter (cp) was optimized using 10-fold cross-validation with the rpart algorithm. The tuning grid ranged from 0 to 0.01 for cp values, in increments of 0.002. The optimal cp value was determined to be 0, suggesting that a full tree model was the most accurate according to RMSE. But it does not mean it is the best model. The cross-validated RMSE was approximately 0.468, with an Rsquared value of 0.168 and a MAE of 0.333. As a foundational step, this optimized model will serve as the Baseline Model for comparison.

Table 4: K-Fold Cross-Validation for Best CP in the Pivot Study.

cp	RMSE	Rsquared	MAE
0.000	0.468	0.167	0.334
0.002	0.464	0.088	0.429
0.004	0.473	0.052	0.447
0.006	0.475	0.044	0.451
0.008	0.479	0.026	0.459
0.010	0.482	0.014	0.465

The Decision Tree Models and Their Performance

First, we set up a model with no parameter constraints, allowing it to grow to its maximum extent to achieve optimal performance. Next, we utilized K-Fold Cross-Validation to compare results on the validation set and determined that the best value for the ‘cp’ parameter is 0.001, while we kept the default values for ‘minsplit’ (20) and ‘maxdepth’ (30) to assess the decision tree’s performance under the optimal ‘cp’ value. Subsequently, we employed ‘rpart’ with a lower ‘maxdepth’ setting of 8, resulting in the creation of the third decision tree model, allowing us to observe the impact of ‘maxdepth.’ Finally, based on the third model, we reduced ‘minsplit’ to 5 to obtain the fourth model and assess the effects of ’minsplit.”

The Baseline Model

In machine learning, a baseline model serves as a point of reference for comparing the performance of other models. It is usually a simple, non-tuned model that sets the initial benchmark for predictive accuracy or error.

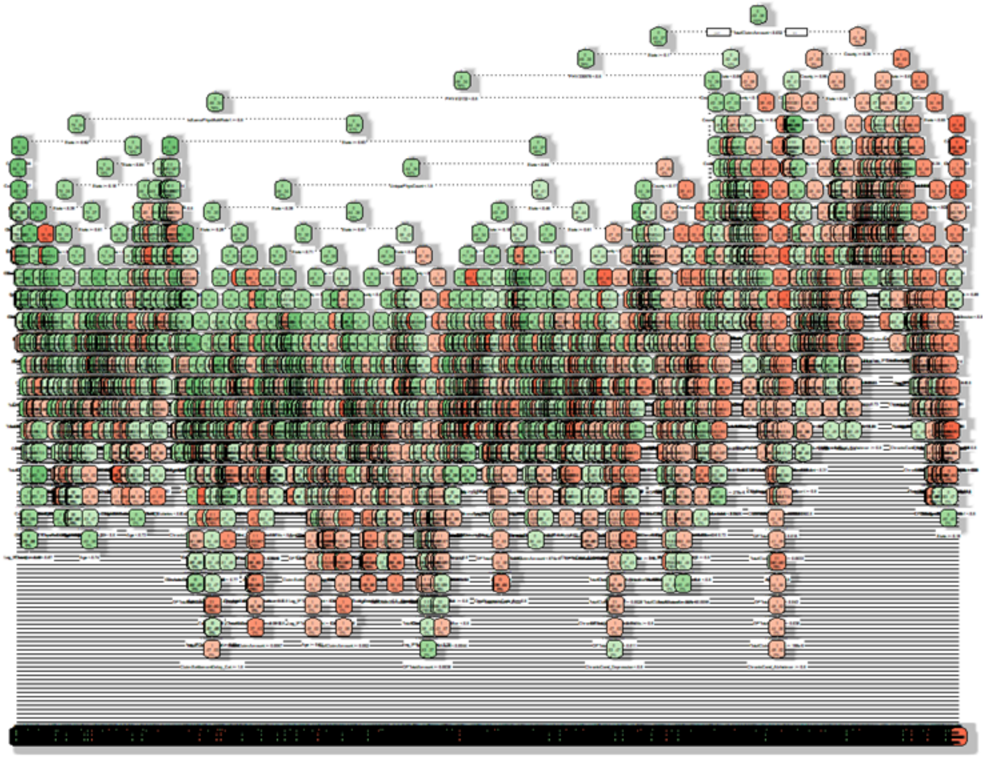
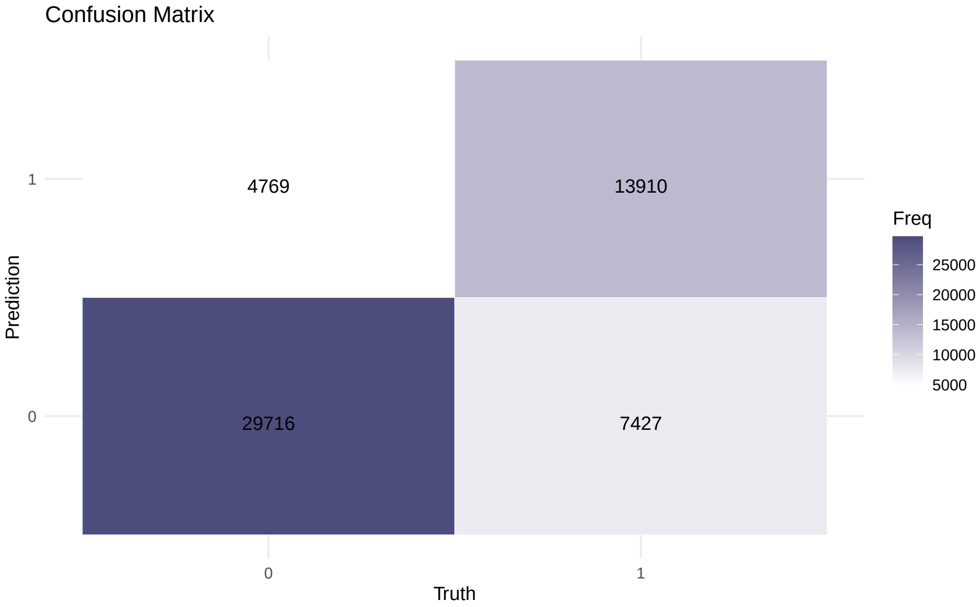


Figure 6: The Baseline Model, the full grown decision tree.

Setting the complexity parameter to zero (`cp=0`) means that the decision tree will grow until each terminal node contains observations from only one class or until other constraints like `minsplit` or `maxdepth` are met. In other words, the tree will be fully grown and will likely be very complex. A `cp` value of zero often risks overfitting the model to the training data. Overfitting occurs when a model learns the training data too well, including its noise and outliers, which leads to poor generalization to new or unseen data. A fully grown tree (cp=0) is often harder to interpret compared to a pruned tree, which might hinder understanding of the data’s underlying structure.

Table 5: The Baseline Model's Performance and statistics.

Confusion Matrix



Statistics

Metric	Value
Accuracy	0.7815
95% CI	(0.7781, 0.7849)
No Information Rate	0.6178
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.5261
McNemar's Test P-Value	< 2.2e-16
Sensitivity	0.8617
Specificity	0.6519
Pos Pred Value	0.8000
Neg Pred Value	0.7447
Prevalence	0.6178
Detection Rate	0.5323
Detection Prevalence	0.6654
Balanced Accuracy	0.7568

The baseline model exhibits an accuracy of approximately 78.15%, indicating that it correctly classifies nearly 78.15% of instances. The Kappa statistic, with a value of about 0.5261, suggests moderate agreement between the model's predictions and actual classifications. Sensitivity, or recall, stands at around 86.17%, highlighting the model's strong ability to correctly identify instances of the positive class. Precision, which is approximately 80.00%, reveals the proportion of true positive predictions among all positive predictions. Furthermore, the F1-Score, a combination of precision and recall.

$$\mathit{F\text{-}Score} = (1+\beta^2)\cdot\frac{\text{Precision}\times\text{Recall}}{\beta^2\times\text{Precision}+\text{Recall}}$$

In this report, we assume all $\beta = 1$. Given this assumption, the F1-Score can be calculated as:

$$F1\text{-}Score = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

Substituting in the provided values:

$$F1\text{-}Score = 2 \times \left(\frac{0.80 \times 0.8617}{0.80 + 0.8617} \right)$$

So, the F1-Score is approximately 0.8278.

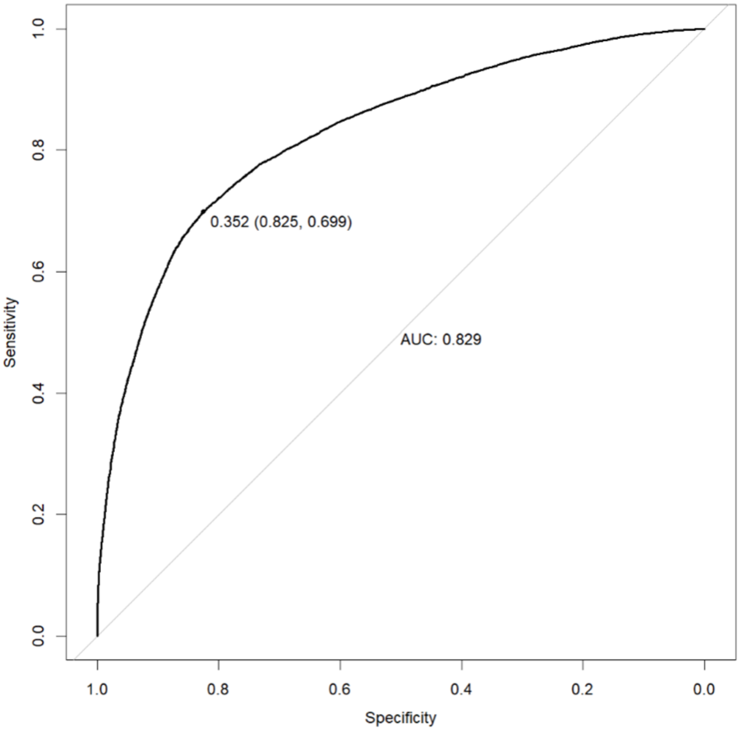


Figure 7: The AUC and ROC of the Baseline Model.

As shown in Figure6, a full-grow model with an AUC of 0.829 suggests that the model performs well in distinguishing between classes. At the threshold of 0.352, the model achieves a good balance with 82.5% specificity (correctly identifying negatives) and 69.9% sensitivity (correctly detecting positives).

Model 1: cp=0.01, minsplit = 20, maxdepth = 30

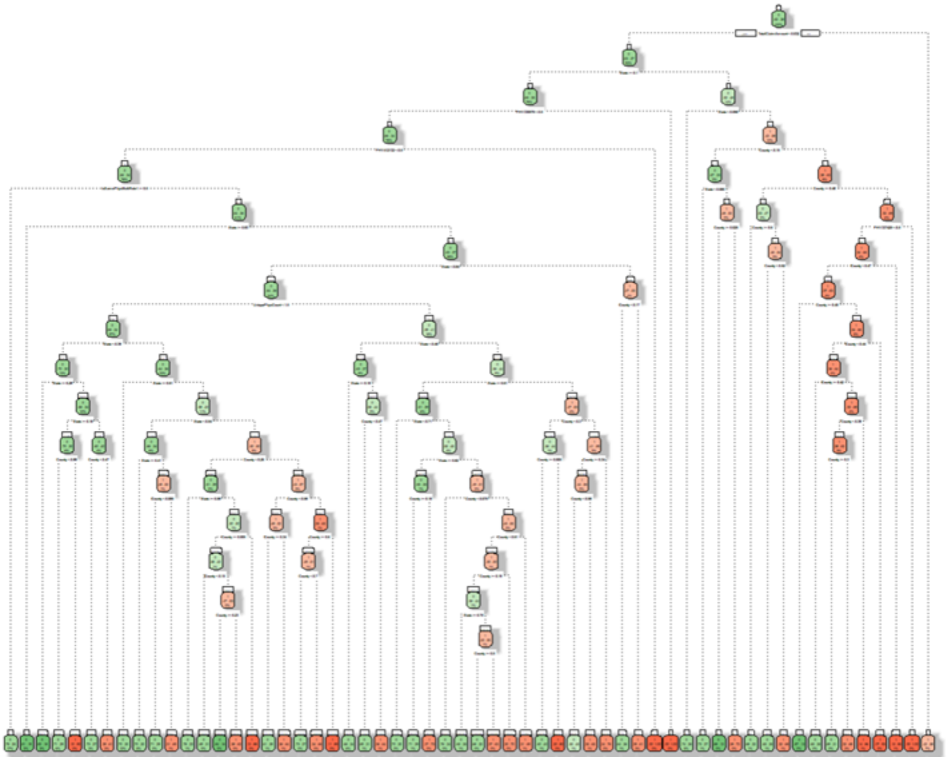
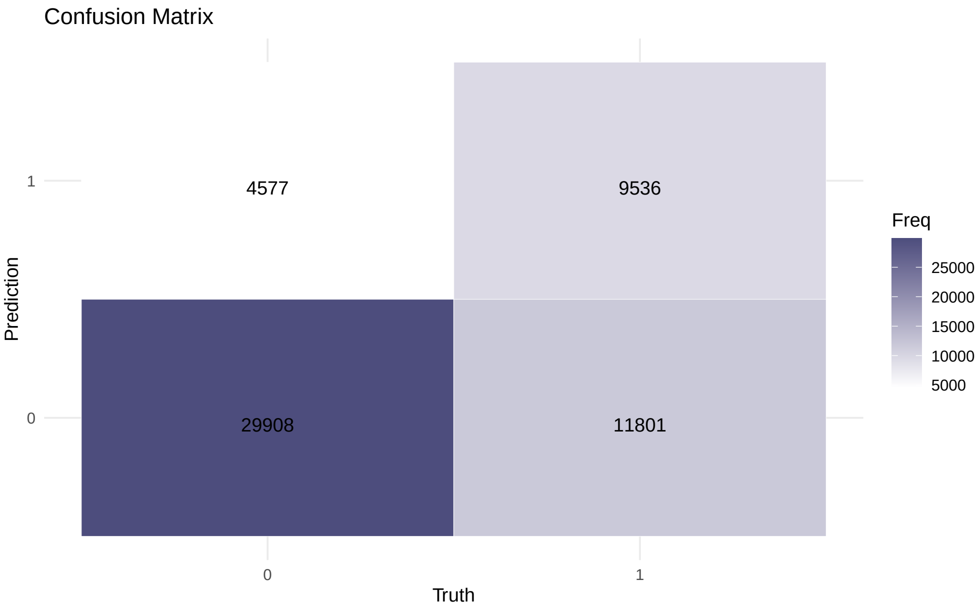


Figure 8: The Model 1, the decision tree with cp=0.01, minsplit = 20, maxdepth = 30.

As shown in Figure 8, even though the Model 1 still exhibits considerable depth in the tree when viewed on the graph, the model remains relatively complex, making it evident that it has performed multiple intricate data partitions to arrive at the final classification decisions. However, in comparison to the fit.full model, the tree's overall structure and individual nodes can now be roughly discerned.

Table 6: The Model 1's Performance and statistics.

Confusion Matrix



Statistics

Metric	Value
Accuracy	0.7066
95% CI	(0.7028, 0.7104)
No Information Rate	0.6178
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.3359
McNemar's Test P-Value	< 2.2e-16
Sensitivity	0.8673
Specificity	0.4469
Pos Pred Value	0.7171
Neg Pred Value	0.6757
Prevalence	0.6178
Detection Rate	0.5358
Detection Prevalence	0.7472
Balanced Accuracy	0.6571

Model 1 have an accuracy of 70.66%, which indicates the proportion of correctly classified instances. The precision is approximately 71.71%, demonstrating the proportion of true positive predictions among all positive predictions. The recall is about 86.73%, representing the model's ability to correctly identify the positive class. The Kappa statistic is 0.3359, suggesting a fair level of agreement between predicted and actual classifications. The F1-Score is approximately 0.7854 suggests a reasonably good balance between precision and recall in the classification model.

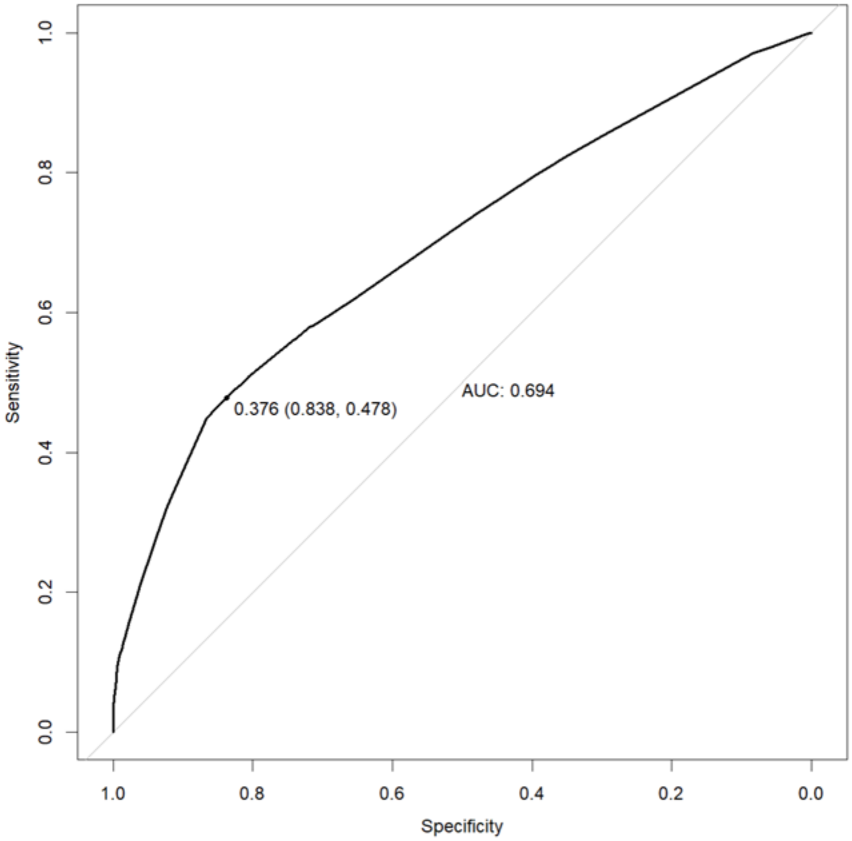


Figure 9: The AUC and ROC of the Model 1.

Figure 9 displays the AUC of the Model 1's performance. The AUC of 0.694 indicates moderate model performance in distinguishing between classes. The optimal threshold of 0.376 balances sensitivity (47.8%) and specificity (83.8%), suggesting that the model can detect positives reasonably well while minimizing false positives.

Model 2: cp=0.001, minsplit=20, maxdepth=8

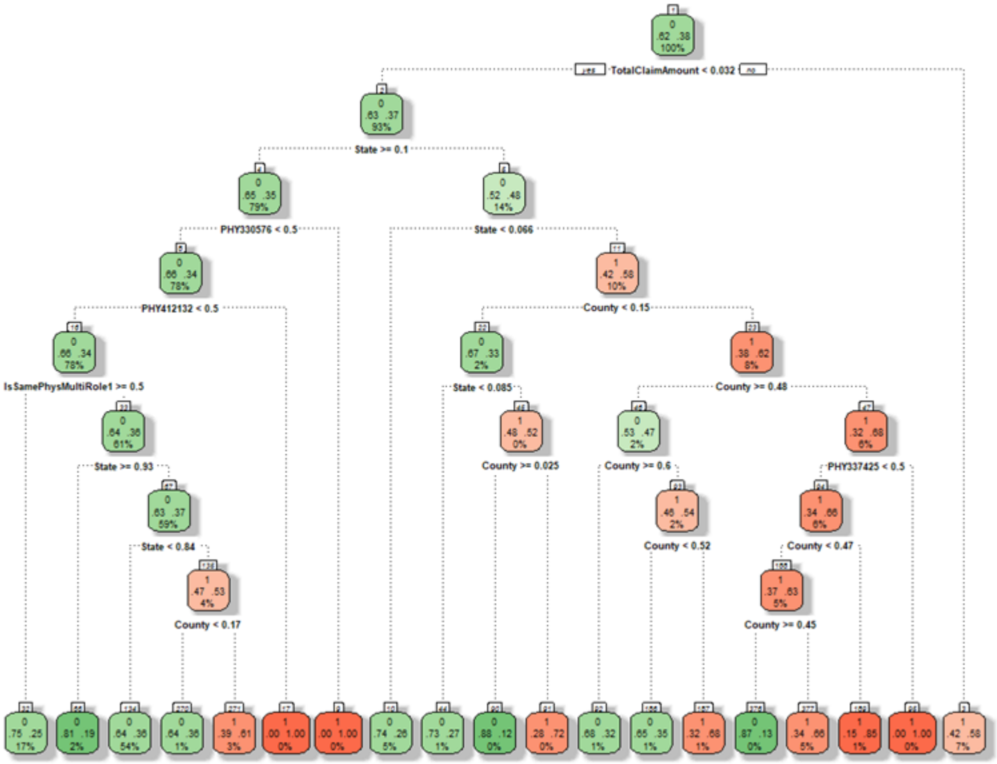
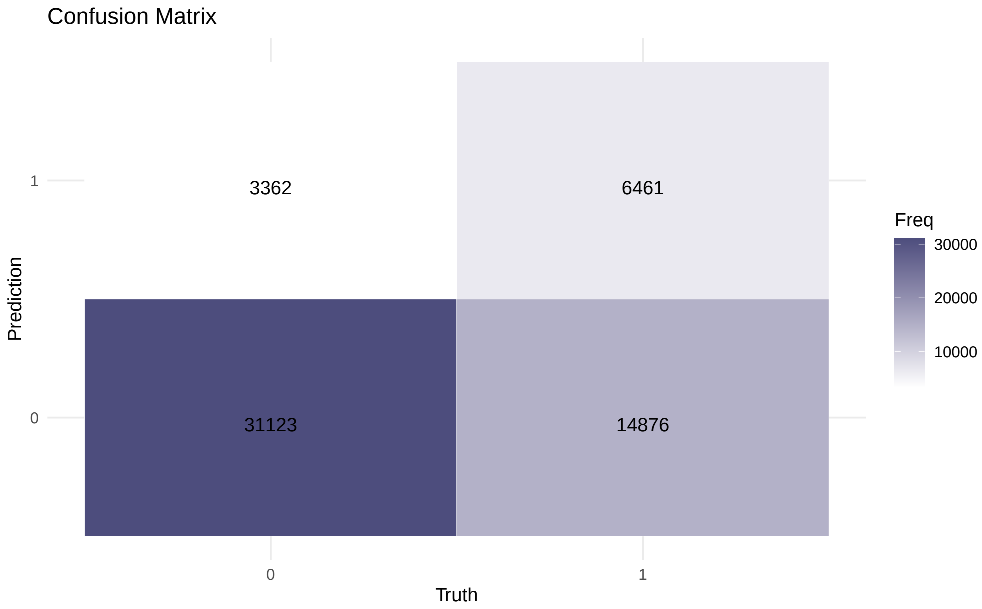


Figure 10: The Model 2, the decision tree with cp=0.001, minsplit=20, maxdepth=8.

In Figure 10, you can clearly see the overall structure of the Model 2. By reducing the model's parameter 'maxdepth,' significant simplification of the model was achieved, making all the structures and nodes clearly visible. The model depth has been well constrained to 8. This demonstrates the substantial impact of 'maxdepth' when dealing with complex decision tree models.

Table 5: The Model2's Performance and statistics.

Confusion Matrix



Statistics

Metric	Value
Accuracy	0.6733
95% CI	(0.6694, 0.6772)
No Information Rate	0.6178
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.2289
McNemar's Test P-Value	< 2.2e-16
Sensitivity	0.9025
Specificity	0.3028
Pos Pred Value	0.6766
Neg Pred Value	0.6577
Prevalence	0.6178
Detection Rate	0.5575
Detection Prevalence	0.8240
Balanced Accuracy	0.6027
'Positive' class	0

As indicated by the accuracy of 67.33%, demonstrates its ability to correctly classify instances. However, it's important to note that the model's kappa value of 0.2289 suggests only a fair level of agreement beyond what would be expected by chance. The recall is relatively high at 90.25%, highlighting the model's proficiency in correctly identifying the positive class. On the other hand, precision value of 0.6766, indicates a relatively good rate of accuracy for positive predictions. The F-Score is approximately 0.7746 indicates a good balance between precision and recall in Model 2's performance.

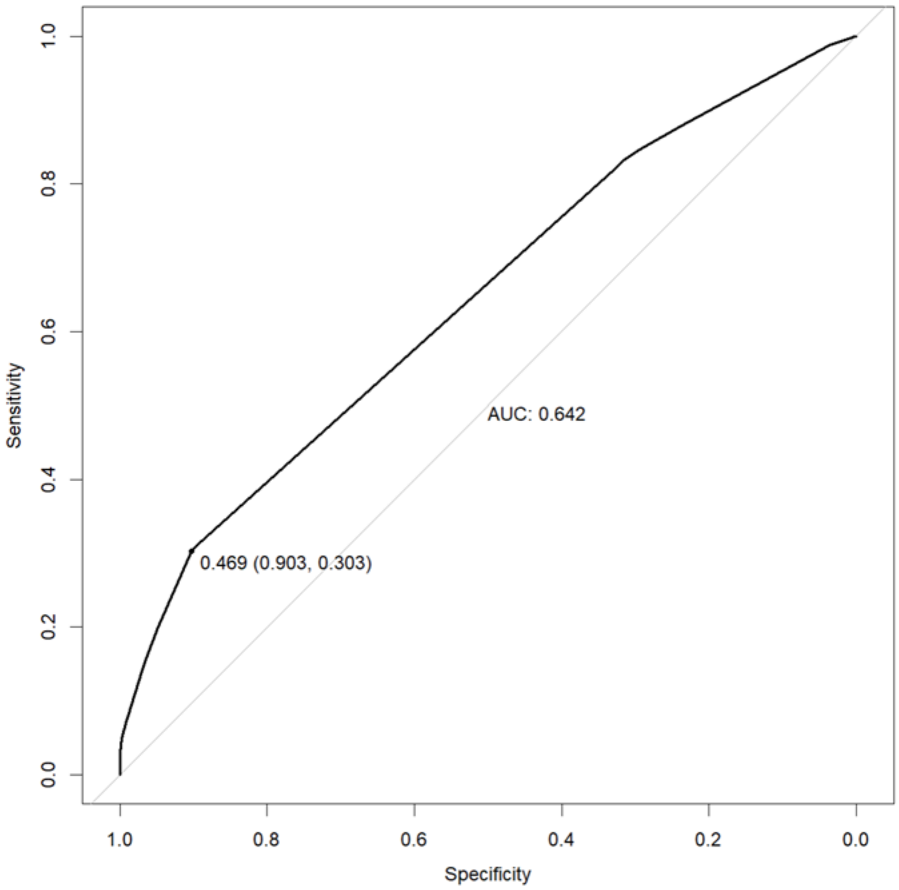


Figure 11: The AUC and ROC of the Model 2.

In Model 2's AUC plot (Figure 10), The AUC value of 0.642 suggests that the model has moderate discriminatory power, but it may not be well-balanced in terms of sensitivity and specificity. The optimal threshold of 0.469 prioritizes specificity (minimizing false negatives (90%)) over sensitivity (correctly identifying positives (30%)).

Model 3: cp=0.001, minsplit=5, maxdepth=8

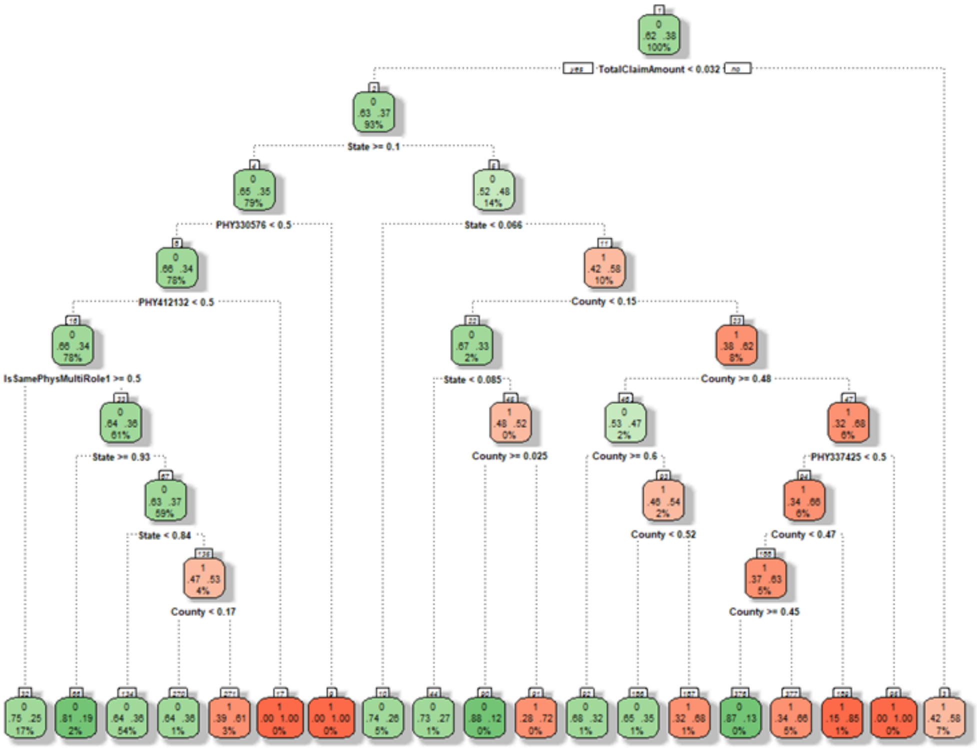


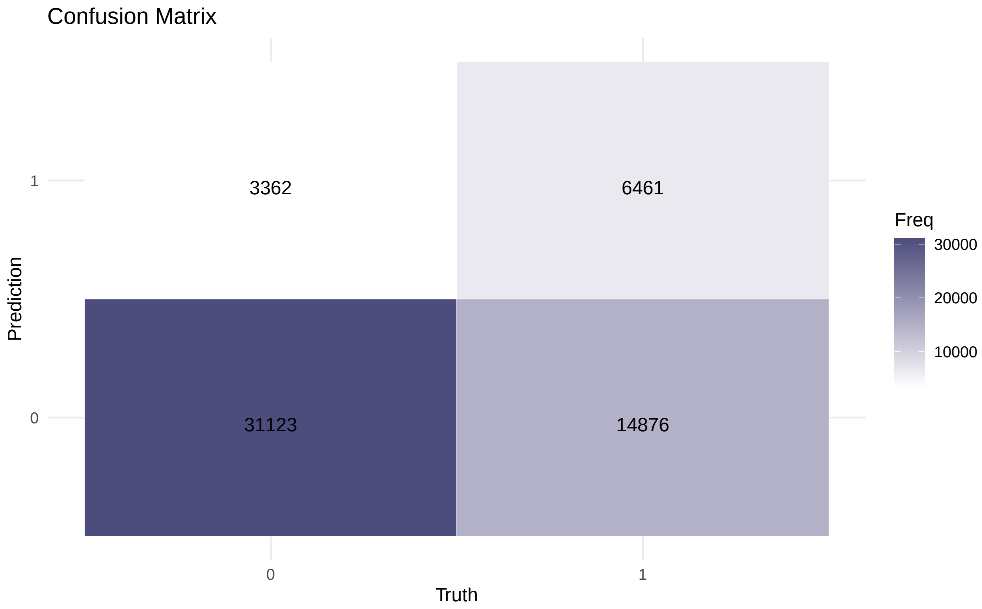
Figure 1: Image Caption

Figure 12: The Model 3, the decision tree with cp=0.001, minsplit=5, maxdepth=8.

Model 3's complexity was further reduced by lowering the minsplit parameter to 5, in addition to the previously reduced maxdepth. This was done to simplify the model further and observe the impact of minsplit. However, as shown in Figure 12, the decision tree model remained unchanged compared to the Model 2. This could indicate that the minsplit parameter has a limited influence on this dataset.

Table 6: The Model 3's Performance and statistics.

Confusion Matrix



Statistics

Metric	Value
Accuracy	0.6733
95% CI	(0.6694, 0.6772)
No Information Rate	0.6178
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.2289
McNemar's Test P-Value	< 2.2e-16
Sensitivity	0.9025
Specificity	0.3028
Pos Pred Value	0.6766
Neg Pred Value	0.6577
Prevalence	0.6178
Detection Rate	0.5575
Detection Prevalence	0.8240
Balanced Accuracy	0.6027
'Positive' class	0

From Table 6, it can be observed that the confusion matrix and statistical results of the Model 3 are quite similar to the performance of the Model 2. Both models exhibit an accuracy 67.33%, a kappa value 0.2289, a recall of 90.25%, a precision value of 0.6766, and an F-score approximately 0.7746.

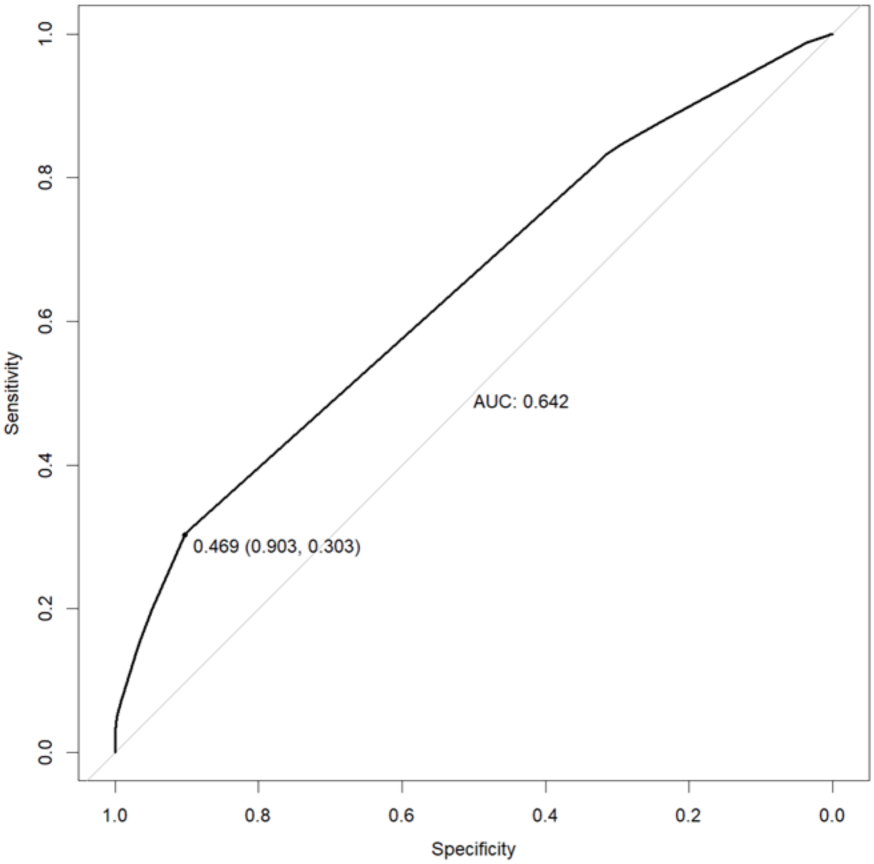


Figure 13: The AUC and ROC of the Model 3.

The AUC chart (Figure 13) for the Model 3 clearly shows the same coordinates and performance as the Model 2.

Model Comparison

Model	Accuracy	Kappa	Recall	Precision	F-score	AUC	Xerror
Baseline	0.7815	0.5261	0.8617	0.8	0.8278	0.829	0.5741198
Model1	0.7066	0.3359	0.8673	0.7171	0.7854	0.694	0.7686583
Model2	0.6733	0.2289	0.9025	0.6766	0.7746	0.642	0.8574061
Model3	0.6733	0.2289	0.9025	0.6766	0.7746	0.642	0.8580986

In the comparison of these models, Baseline and Model 1 perform relatively well, while Model 2 and Model 3 show slightly weaker performance.

The Baseline model excels in various aspects. It achieves relatively high scores in terms of accuracy, Kappa statistics, F-score, and AUC. This indicates that the Baseline model successfully classifies most of the samples, outperforming random classification. Moreover, it boasts a high recall rate, signifying its ability to effectively identify positive-class samples, while maintaining a high level of precision, striking a crucial balance. In cross-validation, its Xerror is relatively low, indicating that the model's performance remains relatively consistent across different data subsets.

The Model 1 demonstrates exceptional recall performance, nearly on par with Baseline model, albeit with slightly lower accuracy and precision. Despite the lower accuracy, the F-score remains relatively high, signifying Model1's strong overall performance. However, Model 1 exhibits a relatively higher Xerror in cross-validation, suggesting that its performance may vary significantly across different data subsets.

The only distinction between the Model 2 and Model 3 lies in the Xerror metric. Xerror measures cross-validation error, assessing the performance variation of the model across different data subsets. In this case, Model 2 has a relatively high Xerror (85.74%), and Model 3's Xerror is slightly higher at 85.81%. This suggests that the performance of Model 2 and Model 3 exhibits significant instability across different data subsets, possibly due to inconsistent model behavior on various data subsets. Consequently, these two models demonstrate relatively unstable performance and are slightly inferior in overall performance compared to Baseline and Model 1.

In conclusion, Baseline and Model 1 demonstrate relatively stable and balanced performance. However, it's essential to consider some key factors since Baseline Model is a fully grown decision tree model. Firstly, fully grown models are prone to overfitting, excelling in fitting the training data but lacking in generalization to new data. Secondly, such models are often very complex, challenging to interpret, and demand significant computational resources. Lastly, to address overfitting and complexity issues, pruning is typically required for fully grown trees. Therefore, after weighing performance, model complexity, and computational resources, we conclude that the Model1 model is the preferred choice.

The Best Model: Model 1

By examining Figure 8 and the Model 1 summary in the **Appendix**, it becomes clear that the Model 1 relies on several crucial attributes for making predictions. 'TotalClaimAmount' emerges as a central factor, acting as the initial decision point and underscoring its importance in detecting potential fraud. 'State' is consistently influential, indicating regional variations in potential fraud cases.

Additionally, 'PHY330576' and 'PHY412132' play pivotal roles, especially when combined with 'SamePhys,' emphasizing their predictive significance. 'UniquePhys' and 'County' are also notable, leading to distinct branches in the tree when specific thresholds are met. 'County' is used in multiple splits, highlighting its critical role in the model's decision-making process.

In summary, 'ClaimAmount' and 'State' are primary drivers of predictions in the Model 1. However, 'PHY330576,' 'PHY412132,' 'SamePhys,' 'UniquePhys,' and 'County' are also essential variables for making accurate predictions regarding potential fraud, taking into account geographic and physician-related factors.

References:

[1] A. Bhardwaj, S. Kumar and A. Naidu, “Predictive analysis and supervised detection for fraudulent cases in healthcare,” 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 416-421, doi: 10.1109/Confluence52989.2022.9734195.

[2] V. Rawte and G. Anuradha, “Fraud detection in health insurance using data mining techniques,” 2015 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, 2015, pp. 1-5, doi: 10.1109/ICCICT.2015.7045689.

[3] J. M. Johnson and T. M. Khoshgoftaar, “Healthcare Provider Summary Data for Fraud Classification” 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI), San Diego, CA, USA, 2022, pp. 236-242, doi: 10.1109/IRI54793.2022.00060.

[4] N. Agrawal and S. Panigrahi, “A Comparative Analysis of Fraud Detection in Healthcare using Data Balancing & Machine Learning Techniques” 2023 International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-4, doi: 10.1109/IC3S57698.2023.10169634.

[5] V. K et al., “Predicting health insurance claim frauds using supervised machine learning technique” 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICONSTEM56934.2023.10142604.

Appendix

n= 446568

node), split, n, loss, yval, (yprob) * denotes terminal node

```
 1) root 446568 170393 0 (0.61843885 0.38156115)
   1) TotalClaimAmount< 0.03223657 416216 152855 0 (0.63275078 0.36724922)
     1) State>=0.1037736 351577 122129 0 (0.65262517 0.34737483)
       1) PHY330576< 0.5 349526 120078 0 (0.65645474 0.34354526)
        2)  PHY412132< 0.5 347805 118357 0 (0.65970299 0.34029701)
          1)  IsSamePhysMultiRole1>=0.5 74705  18879 0 (0.74728599 0.25271401) *
          2)  IsSamePhysMultiRole1< 0.5 273100  99478 0 (0.63574515 0.36425485)
            1)  State>=0.9339623 11134   2083 0 (0.81291539 0.18708461) *
            2)  State< 0.9339623 261966  97395 0 (0.62821511 0.37178489)
             3)   State< 0.8396226 242534  87068 0 (0.64100703 0.35899297)
               1)   UniquePhysCount< 1.5 163415  54822 0 (0.66452284 0.33547716)
                 1)   State< 0.3867925 60126  17965 0 (0.70121079 0.29878921)
                  2)    State>=0.2924528 14820   2730 0 (0.81578947 0.18421053) *
                  3)    State< 0.2924528 45306  15235 0 (0.66373107 0.33626893)
                    1)    State>=0.1792453 27310   8069 0 (0.70454046 0.29545954)
                      1)    County< 0.9914915 26986   7776 0 (0.71185059 0.28814941) *
                      2)    County>=0.9914915 324     31 1 (0.09567901 0.90432099) *
                    2)    State< 0.1792453 17996   7166 0 (0.60180040 0.39819960)
                      1)    County< 0.4654655 11239   3054 0 (0.72826764 0.27173236) *
                      2)    County>=0.4654655 6757   2645 1 (0.39144591 0.60855409) *
                 2)   State>=0.3867925 103289  36857 0 (0.64316626 0.35683374)
                  3)    State>=0.6132075 54689  16498 0 (0.69833056 0.30166944) *
                  4)    State< 0.6132075 48600  20359 0 (0.58109053 0.41890947)
                    1)    State< 0.5377358 25923   8931 0 (0.65547969 0.34452031)
                      1)    State>=0.4056604 21058   6400 0 (0.69607750 0.30392250) *
                      2)    State< 0.4056604 4865   2334 1 (0.47975334 0.52024666)
                        1)    County< 0.08508509 2033    590 0 (0.70978849 0.29021151) *
                        2)    County>=0.08508509 2832    891 1 (0.31461864 0.68538136) *
                    2)    State>=0.5377358 22677  11249 1 (0.49605327 0.50394673)
                      1)    County< 0.2552553 8079   3172 0 (0.60737715 0.39262285)
                        1)    State>=0.5943396 3041    672 0 (0.77902006 0.22097994) *
                        2)    State< 0.5943396 5038   2500 0 (0.50377134 0.49622866)
                         3)     County>=0.005005005 4223   1783 0 (0.57778830 0.42221170)
                           1)     County< 0.1451451 2078    638 0 (0.69297401 0.30702599) *
                           2)     County>=0.1451451 2145   1000 1 (0.46620047 0.53379953)
                             1)     County>=0.2252252 391     62 0 (0.84143223 0.15856777) *
                             2)     County< 0.2252252 1754    671 1 (0.38255416 0.61744584) *
                         4)     County< 0.005005005 815     98 1 (0.12024540 0.87975460) *
                      2)    County>=0.2552553 14598   6342 1 (0.43444307 0.56555693)
                        1)    County< 0.5855856 9550   4562 1 (0.47769634 0.52230366)
                         2)     County>=0.3358358 4477   1756 0 (0.60777306 0.39222694) *
                         3)     County< 0.3358358 5073   1841 1 (0.36290164 0.63709836) *
                        2)    County>=0.5855856 5048   1780 1 (0.35261490 0.64738510)
                         3)     County>=0.5955956 3244   1590 1 (0.49013564 0.50986436)
                           1)     County< 0.6956957 1268    347 0 (0.72634069 0.27365931) *
                           2)     County>=0.6956957 1976    669 1 (0.33856275 0.66143725) *
                         4)     County< 0.5955956 1804    190 1 (0.10532151 0.89467849) *
               2)   UniquePhysCount>=1.5 79119  32246 0 (0.59243671 0.40756329)
                 1)   State< 0.4622642 36921  13509 0 (0.63411067 0.36588933)
                  2)    State>=0.1792453 26853   9058 0 (0.66268201 0.33731799) *
                  3)    State< 0.1792453 10068   4451 0 (0.55790624 0.44209376)
                    1)    County< 0.4654655 6058   1874 0 (0.69065698 0.30934302) *
                    2)    County>=0.4654655 4010   1433 1 (0.35735661 0.64264339) *
                 2)   State>=0.4622642 42198  18737 0 (0.55597422 0.44402578)
                  3)    State>=0.6132075 26474  10380 0 (0.60791720 0.39208280)
                    1)    State< 0.7075472 9998   2955 0 (0.70444089 0.29555911) *
                    2)    State>=0.7075472 16476   7425 0 (0.54934450 0.45065550)
                      1)    State>=0.8207547 6090   2120 0 (0.65188834 0.34811166)
                        1)    County>=0.1751752 5303   1548 0 (0.70808976 0.29191024) *
                        2)    County< 0.1751752 787    215 1 (0.27318933 0.72681067) *
                      2)    State< 0.8207547 10386   5081 1 (0.48921625 0.51078375)
                        1)    County< 0.07507508 1396    336 0 (0.75931232 0.24068768) *
                        2)    County>=0.07507508 8990   4021 1 (0.44727475 0.55272525)
                         3)     County< 0.6056056 6591   3286 1 (0.49855864 0.50144136)
                           1)     County>=0.1851852 5058   2232 0 (0.55871886 0.44128114)
                             1)     State>=0.745283 2742    955 0 (0.65171408 0.34828592) *
                             2)     State< 0.745283 2316   1039 1 (0.44861831 0.55138169)
                              3)      County>=0.4954955 575    182 0 (0.68347826 0.31652174) *
                              4)      County< 0.4954955 1741    646 1 (0.37105112 0.62894888) *
                           2)     County< 0.1851852 1533    460 1 (0.30006523 0.69993477) *
                         4)     County>=0.6056056 2399    735 1 (0.30637766 0.69362234) *
                  4)    State< 0.6132075 15724   7367 1 (0.46851946 0.53148054)
                    1)    County< 0.2952953 6061   2654 0 (0.56211846 0.43788154)
                      1)    County>=0.005005005 5456   2169 0 (0.60245601 0.39754399) *
                      2)    County< 0.005005005 605    120 1 (0.19834711 0.80165289) *
                    2)    County>=0.2952953 9663   3960 1 (0.40981062 0.59018938)
                      1)    County>=0.3358358 8010   3556 1 (0.44394507 0.55605493)
                        1)    County< 0.5755756 3644   1624 0 (0.55433589 0.44566411) *
                        2)    County>=0.5755756 4366   1536 1 (0.35180944 0.64819056) *
                      2)    County< 0.3358358 1653    404 1 (0.24440411 0.75559589) *
             4)   State>=0.8396226 19432   9105 1 (0.46855702 0.53144298)
               1)   County< 0.1651652 6140   2208 0 (0.64039088 0.35960912) *
               2)   County>=0.1651652 13292   5173 1 (0.38918146 0.61081854) *
        3)  PHY412132>=0.5 1721      0 1 (0.00000000 1.00000000) *
       2) PHY330576>=0.5 2051      0 1 (0.00000000 1.00000000) *
     2) State< 0.1037736 64639  30726 0 (0.52465230 0.47534770)
      3)  State< 0.06603774 20672   5293 0 (0.74395317 0.25604683) *
      4)  State>=0.06603774 43967  18534 1 (0.42154343 0.57845657)
        1)  County< 0.1451451 6753   2222 0 (0.67096105 0.32903895)
          1)  State< 0.08490566 5187   1403 0 (0.72951610 0.27048390) *
          2)  State>=0.08490566 1566    747 1 (0.47701149 0.52298851)
            1)  County>=0.02502503 522     65 0 (0.87547893 0.12452107) *
            2)  County< 0.02502503 1044    290 1 (0.27777778 0.72222222) *
        2)  County>=0.1451451 37214  14003 1 (0.37628312 0.62371688)
          1)  County>=0.4754755 10058   4765 0 (0.52624776 0.47375224)
            1)  County>=0.5955956 2834    894 0 (0.68454481 0.31545519) *
            2)  County< 0.5955956 7224   3353 1 (0.46414729 0.53585271)
             3)   County< 0.5155155 3128   1084 0 (0.65345269 0.34654731) *
             4)   County>=0.5155155 4096   1309 1 (0.31958008 0.68041992) *
          2)  County< 0.4754755 27156   8710 1 (0.32073943 0.67926057)
```

```
         1)  PHY337425< 0.5 25745   8710 1 (0.33831812 0.66168188)
          2)   County< 0.4654655 21918   8133 1 (0.37106488 0.62893512)
            1)   County>=0.4454454 1371    172 0 (0.87454413 0.12545587) *
            2)   County< 0.4454454 20547   6934 1 (0.33747019 0.66252981)
              1)   County< 0.4354354 18895   6822 1 (0.36104790 0.63895210)
               2)    County>=0.4154154 2127    745 0 (0.64974142 0.35025858) *
               3)    County< 0.4154154 16768   5440 1 (0.32442748 0.67557252)
                 1)    County< 0.3853854 12855   4906 1 (0.38164138 0.61835862)
                   1)    County>=0.2952953 2014    622 0 (0.69116187 0.30883813) *
                   2)    County< 0.2952953 10841   3514 1 (0.32413984 0.67586016) *
                 2)    County>=0.3853854 3913    534 1 (0.13646818 0.86353182) *
              2)   County>=0.4354354 1652    112 1 (0.06779661 0.93220339) *
          3)   County>=0.4654655 3827    577 1 (0.15077084 0.84922916) *
         2)  PHY337425>=0.5 1411      0 1 (0.00000000 1.00000000) *
2) TotalClaimAmount>=0.03223657 30352  12814 1 (0.42217976 0.57782024) *
```