INFS 5100 – Predictive Analytics

Assignment 3 – Classification

Anisha Mariam Abraham
110406080

Table of Contents

Introduction and Recap

Detecting healthcare fraud is essential to protect patients, maintain provider integrity, and ensure affordable insurance. Although fraud constitutes the most extensive category of federal offences, healthcare fraud does not receive the same level of attention or visibility (Flynn, 2016). Several studies, such as those by Akbar et al. (2020), Nabrawi and Alanzi (2023), Johnson and Khoshgoftaar (2023), and Kumaraswamy et al. (2022), have focused on healthcare fraud detection, emphasizing machine learning models, feature engineering, and data preprocessing. These studies utilized various datasets and machine learning techniques to enhance fraud detection accuracy. For instance, Akbar et al. (2020) used XGBoost to improve recall and accuracy, while Nabrawi and Alanzi (2023) achieved high accuracy with Random Forest (RF). Johnson and Khoshgoftaar (2023) combined RF and XGBoost, highlighting data aggregation and feature engineering, and Kumaraswamy et al. (2022) focused on logistic regression and RF models. Common techniques across these studies included data balancing, feature selection, and preprocessing to handle large healthcare datasets effectively.

In our data exploration, we found that most beneficiaries are aged 80-100 with complex health profiles, leading to potential fraudulent claims, especially posthumous ones. Analyzing top Diagnosis Group Codes for this age group helps identify fraud patterns. Race 1 comprises 84.4% of beneficiaries with a 38% fraud rate, while Races 3 and 5 have higher fraud rates of 45% and 44%, respectively, indicating the need for tailored fraud prevention. Outpatient claims show a higher fraud rate (58%) compared to inpatient claims (37%). Claim durations are right-skewed, with a mean of 9 days and a median of 6 days. Reimbursement amounts also exhibit right-skewness with significant outliers. Moderate positive correlations were found between IPAnnualDeductibleAmt and IPAnnualReimbursementAmt (0.64), and between Admission_Duration and InscClaimAmtReimbursed (0.63), justifying their inclusion in the dataset due to no multicollinearity concerns.

Three decision tree models with varying parameters were compared to determine the best approach. Model 1, with MaxDepth 6, Minsplit 100, and CP 0.0003, showed an accuracy of 81.9%, a high F1 score of 0.72, and an AUC of 0.90. Model 2, with MaxDepth 5, Minsplit 50, and CP 0.000002, achieved an accuracy of 80.5% and a recall of 0.9994 but had a lower F1 score of 0.657. Model 3, with MaxDepth 7, Minsplit 30, and CP 0.0000004, had the highest accuracy at 82.38% and an AUC of 0.905 but a slightly lower F1 score of 0.718. Despite Model 3's higher accuracy, Model 1 was preferred for its balanced performance and reliability. Model 1 was selected as the best model due to its strong overall performance, simplicity, and practical applicability. Key attributes like OtherPhysician, OperatingPhysician, Address_freq, and DiagnosisGroupCode drive its decision-making process.

Data Exploration and Feature Selection

Initially, we examined the basic statistics of each feature, including mean, median, standard deviation, and range, to get a sense of the central tendency and variability. In our data exploration, we found that most beneficiaries are aged 80-100, often with complex health profiles that can lead to fraudulent claims, especially when posthumous claims are involved. Analyzing top Diagnosis Group Codes for this age group, both alive and deceased, helps identify potential fraud patterns. Demographically, Race 1 comprises 84.4% of beneficiaries with a 38% fraud rate, while smaller racial groups, Races 3 and 5, have higher fraud rates at 45% and 44%, respectively, indicating the need for tailored fraud prevention. Outpatient claims show a higher fraud rate (58%) compared to inpatient claims (37%). Claim durations are right-skewed, with a mean of 9 days and median of 6 days, and reimbursement amounts also exhibit right-skewness with significant outliers. Correlation analysis revealed moderate positive relationships between IPAnnualDeductibleAmt and IPAnnualReimbursementAmt (0.64) and between Admission_Duration and InscClaimAmtReimbursed (0.63), justifying their inclusion in the dataset due to the absence of multicollinearity concerns.

Feature engineering played a crucial role in enhancing the dataset's quality. To enhance the dataset for decision tree models, rows with missing values in the Potential_Fraud column were removed to avoid imputation. Feature engineering included creating a new "Address" feature by combining "State" and "County" columns, and removing the redundant "RenalDiseaseIndicator" column since "ChronicCond_KidneyDisease" already captures kidney disease information. Frequency encoding was applied to the "Address" column, replacing each unique address with its frequency in the dataset. Target encoding was used for several categorical variables (e.g., AttendingPhysician, ClmDiagnosisCode_1) by replacing each category with the mean of the PotentialFraud variable within that category. Numeric data types were ensured for financial and age-related columns, and diagnosis and procedure codes. Categorical columns, such as "Race," chronic conditions, physician identifiers, and "PatientType," were converted to factors. The PotentialFraud column was converted from 'Yes' and 'No' to numerical values 1 and 0, respectively, to facilitate model processing.

a.

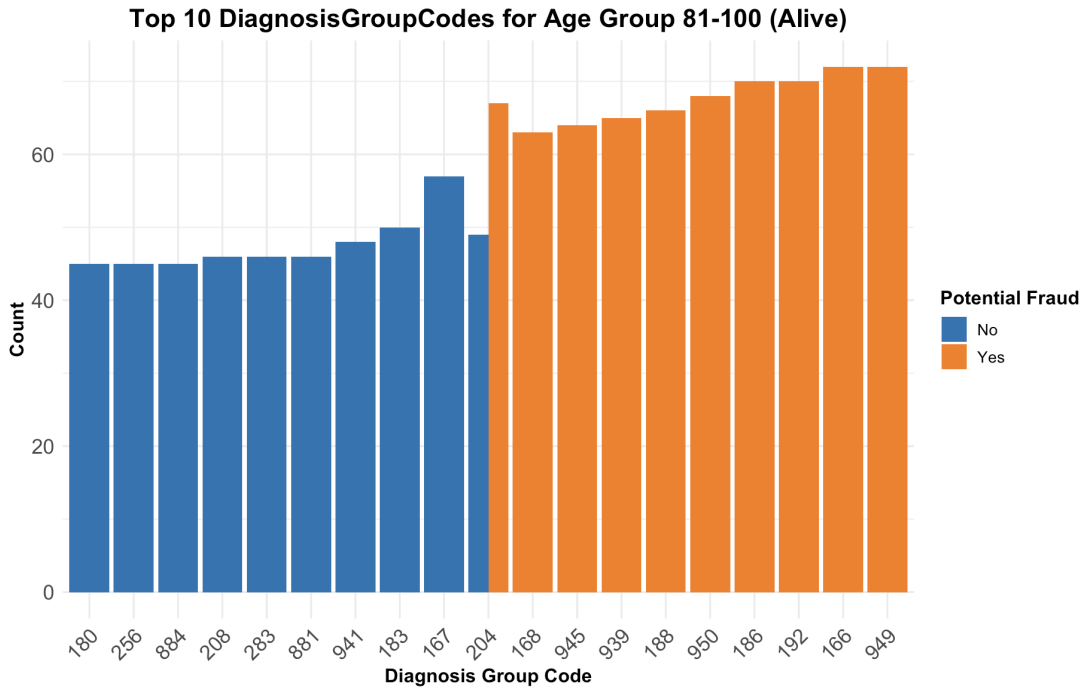


Figure 1: Decision tree for predicting fraud

Figure 1 represents the frequency of the top 10 Diagnosis Group Codes for the age group 81-100, categorized by Potential Fraud. Each Diagnosis Group Code is plotted against its corresponding frequency count, with two separate bars representing non-fraudulent (Potential Fraud = 0) and fraudulent (Potential Fraud = 1) claims. By visualizing the top 10 Diagnosis Group Codes, it becomes apparent which medical conditions or diagnoses are most prevalent within this age group, and whether there are any notable variations in frequency between claims categorized as fraudulent and non-fraudulent.

The plot allows for a quick comparison of the frequency of Diagnosis Group Codes between the two categories of claims, aiding in the identification of any potential trends or anomalies. For example, if certain Diagnosis Group Codes are disproportionately more frequent in fraudulent claims compared to non-fraudulent ones, it could indicate a pattern of fraudulent activity associated with specific medical conditions.

As for alternative representations, other visualizations such as stacked bar charts or heatmaps could also be considered. Stacked bar charts would allow for a direct comparison of the total frequency of Diagnosis Group Codes across both categories, while heatmaps could provide a more comprehensive view by displaying the frequency distribution of Diagnosis Group Codes across different age groups and potential fraud categories simultaneously. These alternative representations could offer additional insights into the relationship between Diagnosis Group Codes, age groups, and potential fraud.

b. The feature selection process involved several steps to identify the most relevant features for the final dataset. Initially, we considered domain knowledge and statistical analysis to identify potentially important features. Features that showed significant differences between the fraudulent and non-fraudulent classes were prioritized.

The dataset provides a comprehensive overview of various factors influencing healthcare utilization and potential fraud detection, including demographic information, medical conditions, financial aspects, healthcare provider details, and specific claim information. Demographic features like Race, State, County, and Age help analyze healthcare patterns across different groups, while medical condition indicators such as ChronicCond_KidneyDisease and ChronicCond_Cancer provide insights into the health status of beneficiaries. Financial information captured through features like IPAnnualReimbursementAmt and DeductibleAmtPaid is crucial for understanding the economic aspects of healthcare utilization. Details about healthcare providers, including AttendingPhysician and OperatingPhysician, allow for the analysis of provider behaviour and patient care patterns.

Additionally, the dataset includes critical variables for fraud detection, such as PotentialFraud, which is essential for identifying suspicious activity. Detailed claim information, including diagnosis and procedure codes, as well as the duration of claims and admissions, helps in understanding the nature of healthcare services used. The newly added features, PatientType, Is_deceased, and Admission_duration, further enhance the dataset by providing information on the type of patient record, the deceased status of beneficiaries, and the duration of hospital admissions, respectively. These variables collectively offer a robust framework for analyzing healthcare data and detecting fraudulent activities.

No changes were made to the outliers identified during the outlier detection process. Outliers were assessed using visualizations such as box plots and scatter plots, with extreme values in financial columns like IPAnnualReimbursementAmt examined closely. While outliers were flagged for potential impact analysis, it was determined that they were legitimate but extreme values and were retained in the dataset without modification.

Scaling is a preprocessing technique used to standardize the range of features in a dataset, ensuring that each variable contributes equally to the analysis. Scaling has been applied to various features within the "sampled_data" dataset using the scale() function in R.

Each feature transforms to centre its values around the mean and scale them to have a standard deviation of one. This process effectively rescales the data to a common range, making it easier to compare and interpret the impact of different variables in subsequent analyses.

For instance, features like "Address_freq," "IPAnnualReimbursementAmt," and "IPAnnualDeductibleAmt" have been scaled, ensuring that their values are centered at zero and have a standard deviation of one. Similarly, other features such as "OperatingPhysician," "OtherPhysician," and "DiagnosisGroupCode" undergo the same scaling process, making them comparable in terms of their influence on the analysis.

Methods such as feature importance from tree-based algorithms were employed to refine the feature set. These methods helped to identify and retain features that contributed most to the model's predictive power while discarding those that were less informative or redundant.

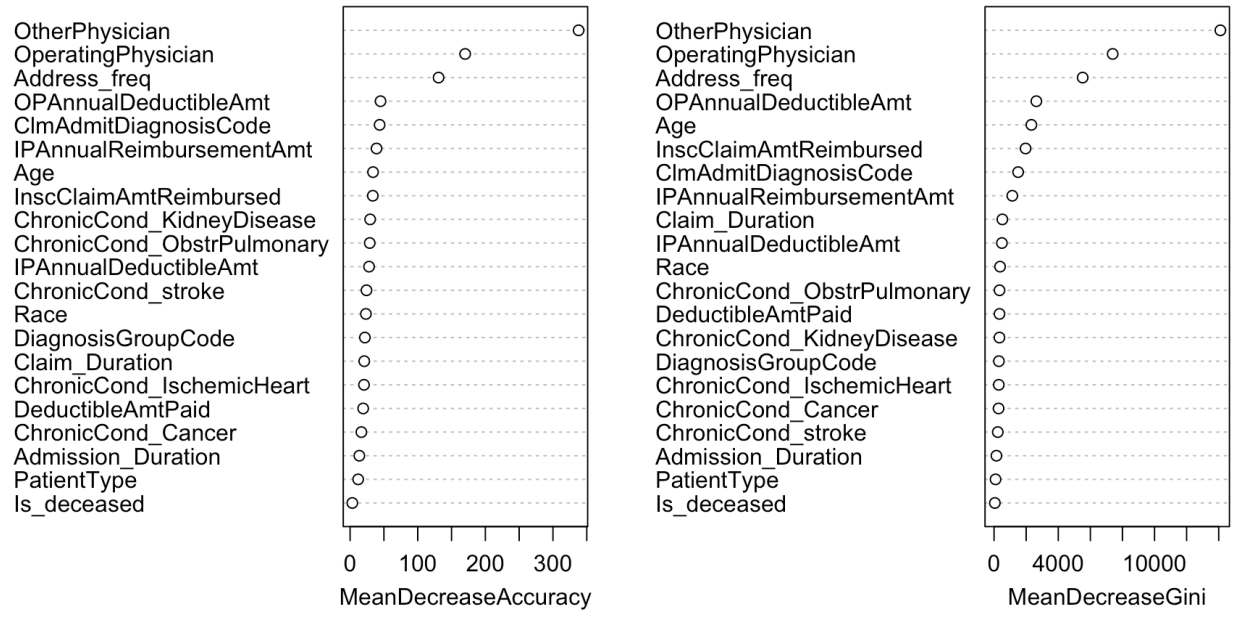


Figure 2: Feature Importance from decision tree

In Figure 2, MeanDecreaseAccuracy provides an aggregate measure of how important the feature is for the overall accuracy of the model. Higher values indicate more important features. MeanDecreaseGini reflects how much a feature contributes to the homogeneity of the nodes and leaves in the resulting Random Forest. Higher values indicate that the feature plays a more significant role in improving the purity of the nodes, and hence, it is more important. In both cases, OtherPhysician, OperatingPhysician and Address_freq were considered as most important features.

Building Classification Models

Naive Bayes Classifier

It is a simple classifier that has its foundation on the well known Bayes's theorem for conditional probability. The algorithm provides probabilistic outputs, meaning it can estimate the likelihood of a transaction being fraudulent. The Naive Bayes classifier used here is the Gaussian Naive Bayes classifier. It assumes that the features follow a Gaussian (normal) distribution. This is appropriate for continuous data, which often fits this assumption reasonably well after scaling. The code includes continuous features such as `Address_freq`, `IPAnnualReimbursementAmt`, `Age`, etc. Gaussian Naive Bayes is suitable for such continuous variables.

A random sample of 100,000 rows is extracted from the dataset to manage computational resources effectively. Features are scaled to have a mean of zero and a standard deviation of one. This scaling is a common preprocessing step for Gaussian Naive Bayes, ensuring that the features are normally distributed and have similar scales. Features like `Address_freq`, `IPAnnualReimbursementAmt`, `Age`, and others are scaled to have a mean of zero and a standard deviation of one. This step is essential for models sensitive to the scale of input features, such as Naive Bayes.

After scaling, the data is split into new training and testing sets following the same 70-30 ratio. The Naive Bayes classifier is then retrained on the scaled data, this time with Laplace smoothing applied. Laplace smoothing, with a parameter of 1, helps to handle cases of zero probabilities which can occur in Naive Bayes when a category in a feature does not appear in the training set. Laplace smoothing is a technique used to handle the zero-frequency problem, where a category in the test data was not seen in the training data. It allows you to specify the amount of smoothing to apply. Tuning this parameter can sometimes improve the model's performance, especially when dealing with sparse data or when there are many categories.

‘Positive’ class: 0	Actual 0	Actual 1
Predicted 0	16965	4964
Predicted 1	1691	6380

Metric	Value
Accuracy	0.7782
95% Confidence Interval	(0.7734, 0.7829)
Kappa	0.5
Sensitivity	0.9094
Specificity	0.5624
Pos Pred Value	0.7736
Neg Pred Value	0.7905
Prevalence	0.6219
Detection Rate	0.5655
Balanced Accuracy	0.7359

Figure 3: Evaluation metrics for Naive Bayes classifier

The Naive Bayes classifier achieved an accuracy of 77.82% which indicates that approximately 77.82% of the total predictions made by the model were correct, with a 95% confidence interval of (77.35%, 78.29%). This range gives us an idea of the reliability and variability of the accuracy estimate. Sensitivity was high at 90.94%, indicating the model correctly identified a large proportion of actual positives. Specificity was moderate at 56.24%, suggesting room for improvement in identifying actual negatives. The model's Kappa value was 0.5, indicating moderate agreement beyond chance. Other metrics, including Precision (PPV) and NPV, further highlight the classifier's performance. The Naive Bayes classifier shows a significant improvement over the No Information Rate, as indicated by the P-Value.

Support Vector Machine

In potential fraud detection, SVMs can accurately identify subtle anomalies and irregularities in transaction behavior by constructing optimal hyperplanes that separate legitimate transactions from fraudulent ones. (Dheepa & Dhanapal, 2012). Let's plot the data with the two most important features to see whether the classes are linearly separable:

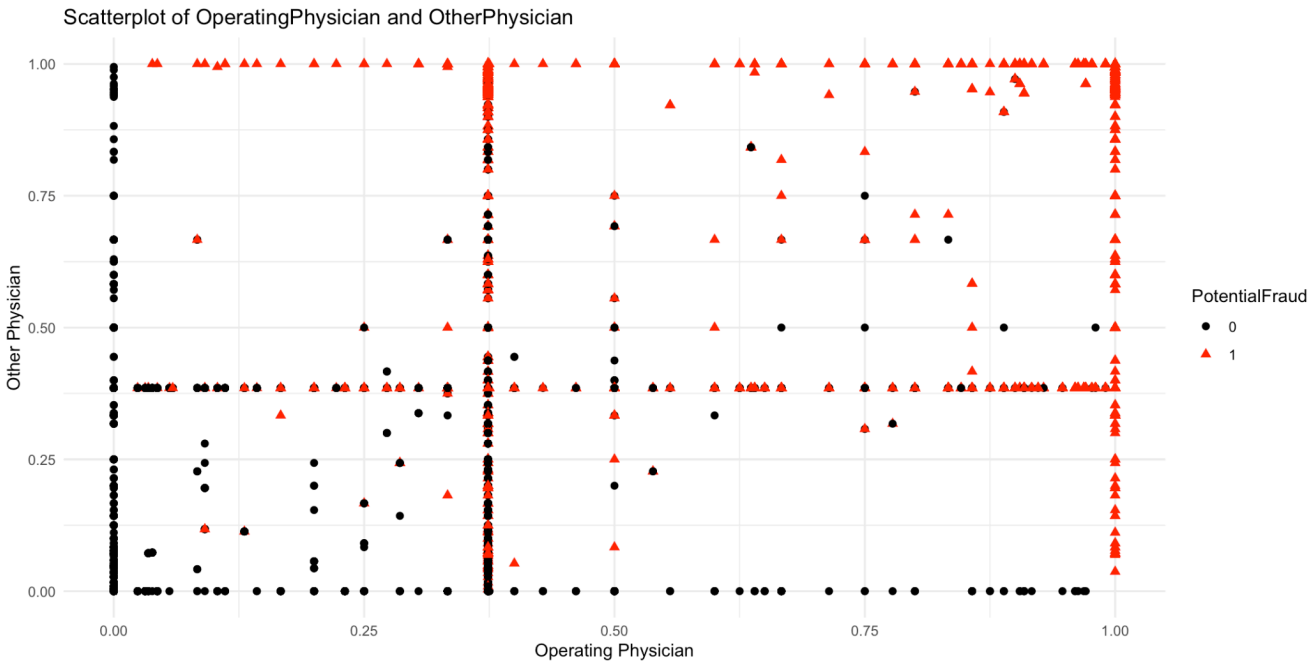


Figure 4: Scatterplot to study linear separability

From Figure 4, we can see that the data is not linearly separable. When dealing with a dataset where a linear hyperplane fails to provide adequate separation between classes, it is crucial to explore alternative kernel functions to achieve better classification results. This scenario necessitates the use of more sophisticated techniques, such as the polynomial kernel, to map the input features into a higher-dimensional space where a separating hyperplane can be effectively constructed.

The polynomial kernel, which is a non-linear kernel function, allows the Support Vector Machine (SVM) to capture the complex relationships within the data by transforming the input features into polynomial combinations. A cost argument allows us to specify the cost of a violation to the margin. When the cost argument is small, then the margins will be wide and many support vectors will be on the margin or will violate the margin. Parameter tuning is a critical step in machine learning aimed at optimizing the performance of a model. The primary goal of parameter tuning is to find the best set of parameters that maximize the model's performance. Proper tuning can significantly improve metrics such as accuracy, precision, recall, and F1-score.

Parameter tuning for Support Vector Machines (SVM) using 10-fold cross-validation involves optimizing the 'cost' parameter, which controls the trade-off between maximizing the margin and minimizing the classification error. The dataset is divided into 10 equal parts (folds). The SVM model is trained on 9 parts and tested on the remaining part. This process is repeated 10 times, with each fold used as a test set once. The average performance across all folds is considered to estimate the model's effectiveness. The SVM model with a cost of 1 achieved the lowest average error rate of 0.1914286 across the 10-fold cross-validation. This is the best performance among the different 'cost' values tested. The error rate is the highest at 0.2028571 for Cost = 1000, indicating overfitting as the model penalizes errors too heavily.

	Cost	Error	Dispersion
1	1e-01	0.1958571	0.01607028
2	1e+00	0.1914286	0.01502832
3	1e+01	0.1982857	0.01613155
4	1e+02	0.1974286	0.01488886
5	1e+03	0.2028571	0.01514855

Figure 5: Parameter tuning results

With an accuracy of 81.17%, the model demonstrates a commendable ability to correctly classify claims as fraudulent or not. Notably, the model exhibits high sensitivity (97.00%), effectively identifying non-fraudulent claims, but its specificity (55.12%) in detecting fraudulent claims is comparatively lower. This imbalance suggests that while the model excels in recognizing genuine claims, it could benefit from enhancements to accurately flag suspicious activities. Additionally, the Kappa value of 0.5655 reflects a moderate level of agreement beyond chance, further affirming the model's reliability.

'Positive' class: 0	Actual 0	Actual 1
Predicted 0	18097	5091
Predicted 1	559	6253

Metric	Value
Accuracy	0.8117
95% Confidence Interval	(0.8072, 0.8161)
Kappa	0.5655
Sensitivity	0.9700
Specificity	0.5512
Pos Pred Value	0.7804
Neg Pred Value	0.9179
Prevalence	0.6219
Detection Rate	0.6032
Balanced Accuracy	0.7606

Figure 6: Evaluation metrics for SVM

KNN

KNN checks how similar a data point is to its neighbour and classifies the data point into the class it is most similar to. Unlike most algorithms, KNN is a non-parametric model which means that it does not make any assumptions about the data set. KNN is a lazy algorithm, this means that it memorizes the training data set instead of learning a discriminative function from the training data.

A range of `k` values is defined using Elbow method. For each iteration, a kNN classifier is instantiated with the current `k` value and trained on the training data. The classifier's performance is evaluated on both the training and testing datasets by predicting the target variable and calculating the accuracy. These accuracy values, along with the corresponding `k` values, are stored in the data frame.

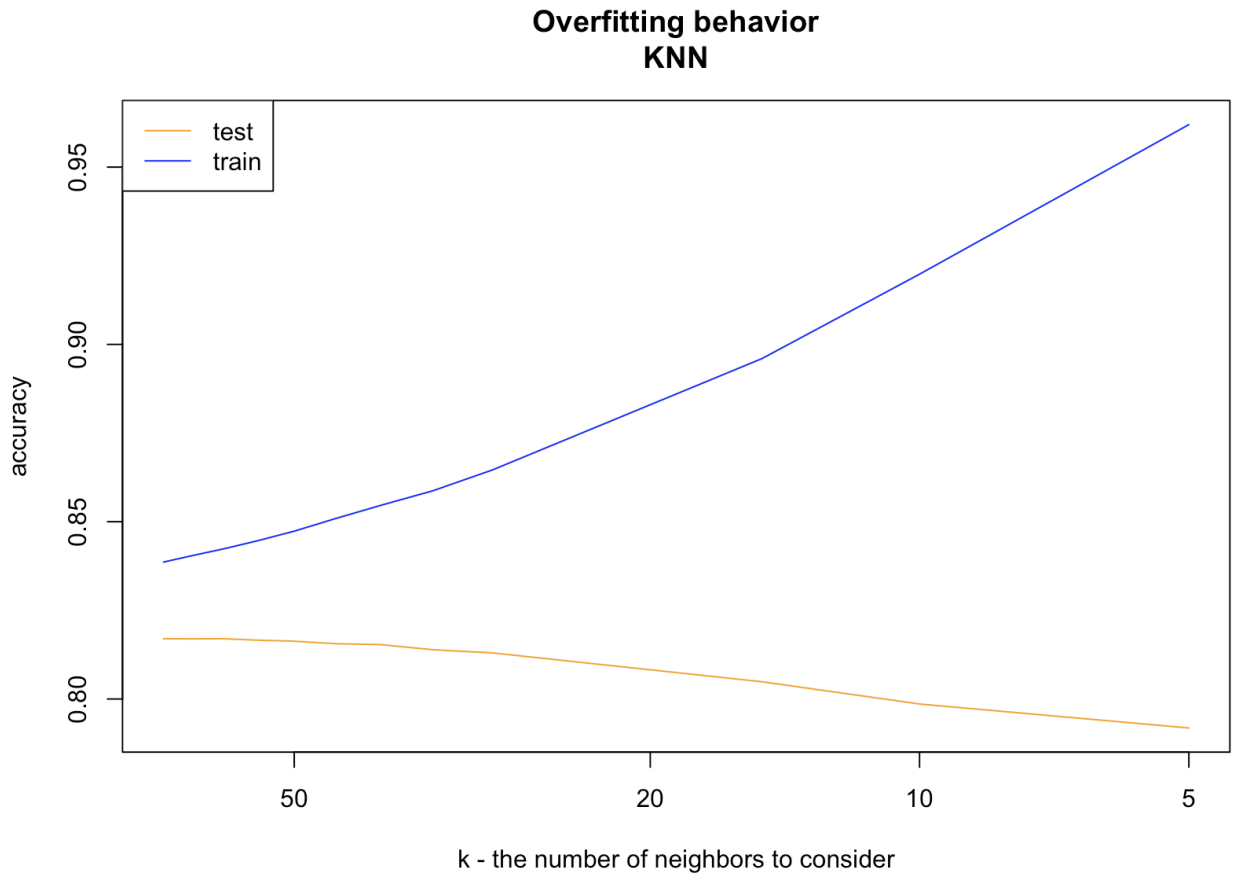


Figure 7: Plot to study overfitting behaviour via Elbow method

	acc_train	acc_test	k
1	0.8386143	0.8170000	70
2	0.8404429	0.8169667	65
3	0.8423286	0.8170000	60
4	0.8446000	0.8166000	55
5	0.8473286	0.8163000	50
6	0.8508571	0.8156000	45
7	0.8546286	0.8153333	40
8	0.8587143	0.8139000	35
9	0.8646286	0.8130000	30
10	0.8960000	0.8048667	15
11	0.9198000	0.7986000	10
12	0.9619714	0.7918333	5

Figure 8: Values of k and their accuracy on trainData and testData

After completing the loop, the results are visualized using a plot that shows the training and test accuracy for different `k` values in Figure 7. This plot helps to identify the overfitting behaviour of the kNN model. Overfitting is indicated when the model performs significantly better on the training data compared to the test data, especially for lower values of `k`. Conversely, higher values of `k` tend to generalize better but may underfit the data. The plot helps in selecting an optimal `k` value that balances these effects.

Finally, the code trains a kNN model using the optimal `k` value from Figure 8 (in this case, `k = 60`), based on the results from the earlier experiments. The model is trained on the full training dataset and its performance is evaluated on the test dataset. A confusion matrix is generated to provide a detailed assessment of the model's performance, including metrics such as accuracy, precision, recall, and F1-score. The matrix shows that out of the actual non-fraudulent cases (class 0), the model correctly identified 17,729 instances while misclassifying 4,563 as fraudulent. For the actual fraudulent cases (class 1), the model correctly identified 6,781 instances but misclassified 927 as non-fraudulent.

‘Positive’ class: 0	Actual 0	Actual 1
Predicted 0	17729	4563
Predicted 1	927	6781

Metric	Value
Accuracy	0.817
95% Confidence Interval	(0.8126, 0.8214)
Kappa	0.5848
Sensitivity	0.9503
Specificity	0.5978
Pos Pred Value	0.7953
Neg Pred Value	0.8797
Prevalence	0.6219
Detection Rate	0.5910
Balanced Accuracy	0.7740

Figure 9: Evaluation metrics for kNN

The overall accuracy of the model is 81.7%, with a 95% confidence interval of (81.26%, 82.14%). This accuracy indicates that the model is correct in its predictions approximately 82% of the time. The Kappa statistic, which accounts for agreement by chance, is 0.5848, suggesting moderate agreement. In terms of specific performance metrics, the model demonstrates high sensitivity (recall) of 95.03%, meaning it correctly identifies a high proportion of actual non-fraudulent cases. However, the specificity is 59.78%, indicating a moderate ability to correctly identify fraudulent cases. The Positive Predictive Value (precision) for non-fraudulent predictions is 79.53%, meaning that around 80% of the instances predicted as non-fraudulent are indeed non-fraudulent. The Negative Predictive Value is 87.97%, indicating that about 88% of the instances predicted as fraudulent are fraudulent.

Random forest

In a random forest classification, multiple decision trees are generated using different random subsets of the data and features. Each decision tree acts like an expert, offering its classification opinion. Predictions are made by aggregating the classifications from all the decision trees and selecting the most frequent result. Random forest algorithms have been shown to significantly improve the detection of fraudulent credit card transactions. By analyzing large datasets of past transactions, the algorithm can distinguish between normal and fraudulent behaviour with high accuracy (Naik et al., 2023), (Xuan et al., 2018).

The randomForest function from the randomForest package is used to construct this model with 100 trees (ntree = 100). This ensemble method combines the predictions of multiple decision trees to improve overall accuracy and reduce overfitting compared to a single decision tree. The predictions made on the test data are compared with the actual values to compute the model's accuracy.

'Positive' class: 0	Actual 0	Actual 1
Predicted 0	17729	4563
Predicted 1	927	6781

Metric	Value
Accuracy	0.817
95% Confidence Interval	(0.8126, 0.8214)
Kappa	0.5848
Sensitivity	0.9503
Specificity	0.5978
Pos Pred Value	0.7953
Neg Pred Value	0.8797
Prevalence	0.6219
Detection Rate	0.5910
Balanced Accuracy	0.7740

The matrix indicates the number of true positives (17630), true negatives (7475), false positives (3869), and false negatives (1026). The accuracy of the model is 0.8368, meaning that 83.68% of the predictions made by the model are correct. This is a strong indicator of the model's overall effectiveness. The 95% confidence interval (0.8326, 0.841) provides a range within which the true accuracy of the model is likely to fall, giving additional context to the accuracy measurement. The Kappa statistic is 0.6351 measures the agreement between the predicted and actual classifications while accounting for the possibility of agreement occurring by chance. Sensitivity (0.9450) and specificity (0.6589) are measures of the model's ability to correctly identify true positives and true negatives, respectively. The positive predictive value (0.8200) and negative predictive value (0.8793) indicate the proportion of positive and negative predictions that are correct, respectively.

Figure 10: Evaluation metrics of Random forest

Model Comparison and Conclusion

Metric	Naive Bayes	Support Vector Machine	k-Nearest Neighbors	Random Forest
Accuracy	77.82%	81.17%	81.7%	83.68%
95% CI	(77.35%, 78.29%)	(80.72%, 81.61%)	(81.26%, 82.14%)	(83.26%, 84.10%)
Kappa	0.5	0.5655	0.5848	0.6351
Sensitivity	90.94%	97.00%	95.03%	94.50%
Specificity	56.24%	55.12%	59.78%	65.89%
Balanced Accuracy	73.59%	76.06%	77.40%	80.20%

Figure 11: Comparison between different models

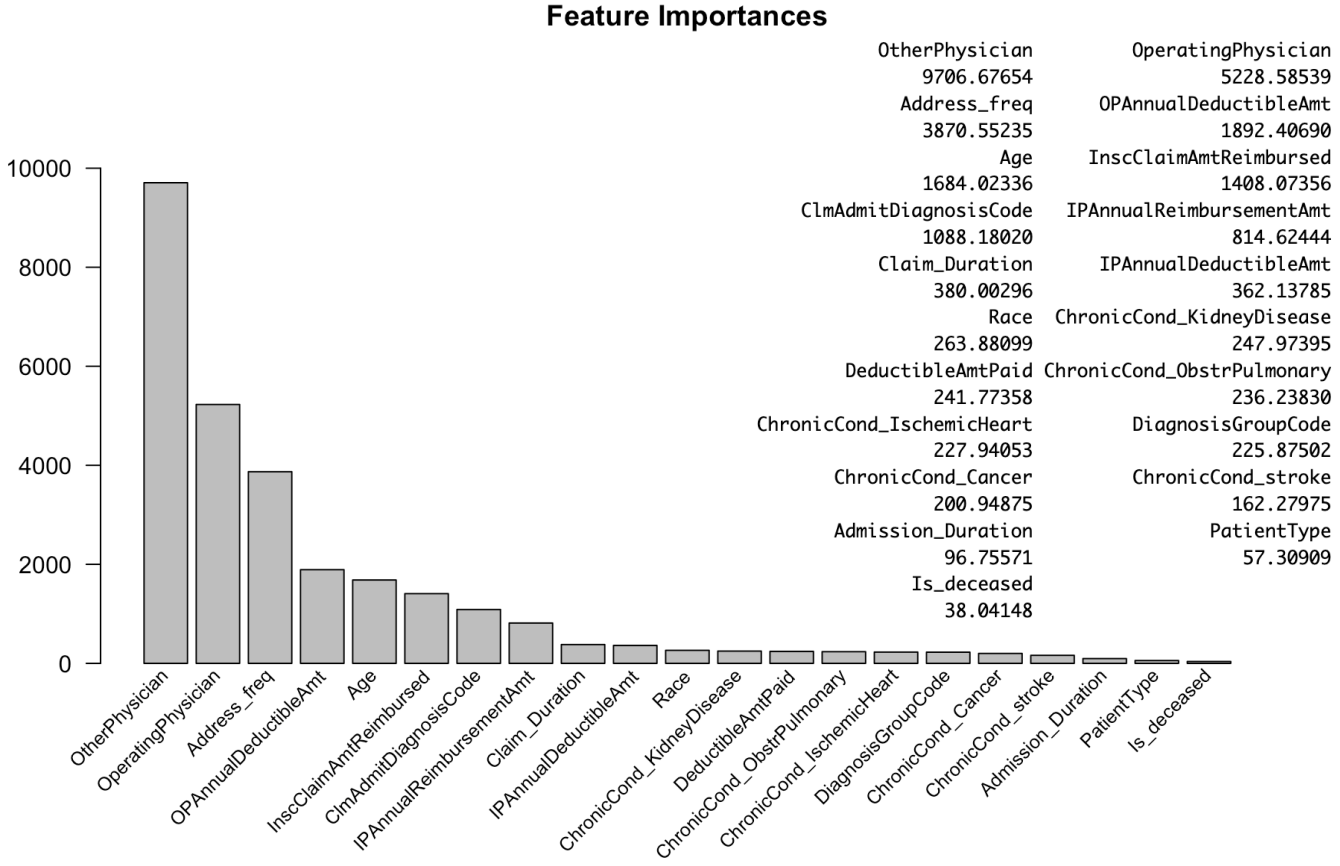
The Random Forest model demonstrates superior performance across multiple metrics, making it the best choice for detecting fraudulent transactions. Its highest accuracy of 83.68% indicates that the model correctly classifies a large proportion of the transactions, both fraudulent and non-fraudulent. This high accuracy suggests that the model is effective at learning the patterns and nuances in the data that distinguish legitimate transactions from fraudulent ones.

One of the key advantages of the Random Forest model is its high Kappa statistic of 0.6351. The Kappa statistic measures the agreement between the predicted and actual classifications, accounting for the possibility of agreement occurring by chance. A Kappa value above 0.6 indicates substantial agreement, which underscores the reliability of the Random Forest model in providing consistent and dependable predictions.

The Random Forest model also excels in terms of specificity, achieving a rate of 65.89%. Specificity measures the model's ability to correctly identify true negative cases, or in this context, non-fraudulent transactions. A higher specificity means the model is better at avoiding false positives, where legitimate transactions are incorrectly flagged as fraudulent. This reduces unnecessary alerts and investigations, making the fraud detection process more efficient and less burdensome on resources.

The model's balanced accuracy of 80.20% further highlights its robustness. Balanced accuracy is the average of sensitivity and specificity, providing a comprehensive measure of the model's

performance across both classes. The Random Forest's ability to maintain a high balanced accuracy indicates that it effectively handles the trade-off between detecting fraudulent transactions (sensitivity) and correctly identifying non-fraudulent ones (specificity).



The Random Forest model identified several key features that are highly predictive of fraudulent activities. The involvement of multiple physicians (OtherPhysician and OperatingPhysician), high frequency of certain addresses, and large deductible amounts are among the top indicators. These findings align with common fraud detection practices, which focus on unusual or suspicious patterns in billing and treatment.

Financial variables like deductible amounts and reimbursement amounts are critical, as they directly relate to the monetary aspects of claims. Chronic conditions and demographic details like age and race also provide valuable context, helping to identify patterns that deviate from typical medical practices.

The analysis concludes that the Random Forest classifier is the most effective model for detecting fraudulent transactions, outperforming Naive Bayes, Support Vector Machine, and k-Nearest Neighbors across various performance metrics. With the highest accuracy of 83.68%, substantial Kappa value of 0.6351, and the best Positive Predictive Value of 82.00%, the Random

Forest model demonstrates a superior capability to correctly classify both fraudulent and non-fraudulent transactions. Its high specificity and balanced accuracy indicate a robust performance in identifying genuine transactions while accurately flagging suspicious ones. The key features identified, such as the involvement of multiple physicians and significant financial variables, provide valuable insights into fraudulent patterns. These results underscore the Random Forest model's potential to significantly enhance fraud detection processes, offering reliable and actionable predictions for organizations to reduce financial losses and improve operational efficiency.

References

1. Flynn, K 2016, 'Financial fraud in the private health insurance sector in Australia perspectives from the industry', Journal of Financial Crime, vol. 23, no. 1, pp. 143–158.

2. Akbar, NA, Sunyoto, A, Rudyanto Arief, M & Caesarendra, W 2020, 'Improvement of decision tree classifier accuracy for healthcare insurance fraud prediction by using Extreme Gradient Boosting algorithm', in *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, IEEE, pp. 110–114.

3. Johnson, JM & Khoshgoftaar, TM 2023, 'Data-Centric AI for Healthcare Fraud Detection', *SN Computer Science*, vol. 4, no. 4, pp. 389–389.

4. Nabrawi, E., & Alanazi, A. (2023). Fraud Detection in Healthcare Insurance Claims Using Machine Learning. *Risks, 11*(9), 160.

5. Kumaraswamy, N, Markey, MK, Barner, JC & Rascati, K 2022, 'Feature engineering to detect fraud using healthcare claims data', Expert Systems with Applications, vol. 210, pp. 118433-.

6. Dheepa, V., & Dhanapal, R. (2012). Behavior based credit card fraud detection using support vector machines. *ICTACT Journal on Soft Computing, 2*(4), 391.

7. Naik, L., Shekar, K., Madhu, R., & Vinod, T. (2023). Credit Card Fraud Detection using Random Forest. International Journal of Advanced Research in Science, Communication and Technology.

8. Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018). Random forest for credit card fraud detection. 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 1-6.