Detecting Fraudulent Activities in Credit Card Transactions

Kritthika Shanmugam, Priyanka Jaina and Sathiyashivani Sathish Kumar

Team Number: 4, DS520: Data Mining, School of Technology and Computing

City University of Seattle)

[shanmugamkritthika@cityuniversity.edu](mailto:shanmugamkritthika@cityuniversity.edu)

[jainapriyanka@cityuniversity.edu](mailto:jainapriyanka@cityuniversity.edu)

[sathishkumarsathiya@cityuniversity.edu](mailto:sathishkumarsathiya@cityuniversity.edu)

**Abstract**

We live in the golden age of technology; internet shopping has become a simple and time-saving choice for customers all over the world. Credit card payments have a vital role in facilitating these transactions. In this, the fraudulent activity in credit card transactions risks banks and consumers. To reduce financial losses and to protect our transactions, a reliable credit card fraud detection system is required. This paper analyzes the issues connected with credit card fraud and emphasizes the importance of a good detection system. Credit card fraud is detected by machine learning techniques such as Naive Bayes, Logistic Regression, SVM and Decision Trees. This paper focuses on analyzing data to acquire information on credit card fraud. It involves implementing and evaluating the Decision Tree algorithm as part of the fraud detection system. The goal is to advance our understanding of credit card fraud detection methods and improve the overall security of online transactions.

**Keywords**

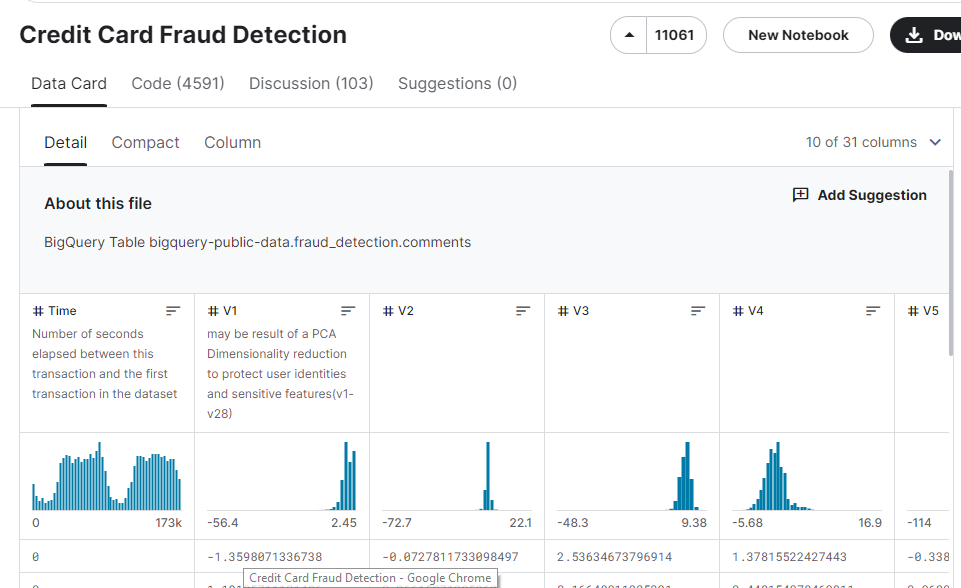
Data Mining, Credit Card Fraud, Detection Classification, Decision Trees, Techniques and Challenges, electronic payment, fraud detection

1. **PROBLEM STATEMENT**

In the current world, credit card frauds have been a widespread problem. This paper has a primary goal to create a strong model to prevent and detect fraud and to make financial transactions safer and more secure. Using R programming environment, we are developing a Decision Tree model for detecting fraud transactions. The problem is to be the complex task of dealing with the increasing prevalence of online transactions, which requires us to distinguish between legitimate and fraudulent transaction activities quickly and accurately. This study's goals are to increase transaction security, reduce financial losses, and develop an effective fraud detection system. We ensure this by achieving prominent levels of accuracy, precision, and recall rates while minimizing false positives, that our fraud detection process is reliable and efficient. Successfully achieving this goal means creating a system that is skilled at navigating the complexities of credit card transactions. It is not just about detecting fraud in the short term; it is also about making meaningful contributions to the broader field of fraud prevention by harnessing the power of machine learning. This paper operates on the success of building trust not only among financial institutions but also among individual users, ensuring a secure financial environment in today's changing landscape of digital transactions. We are using a Kaggle dataset in this project. This chosen dataset had an imbalance, presenting added challenges.

**2. DATASET SELECTION**

The dataset we used on this task originates from Kaggle supply and it is miles a good platform for diverse datasets. As we analyzed the dataset, this is the records from the credit card transactions performed by using European cardholders at some stage in September 2013, encompassing 284,807 transactions over a two-day length. We decided on this dataset as it aligns with the task's core goals, centered on figuring out fraudulent activities within credit score card transactions. Its relevance lies in supplying an actual-world situation, proposing a 0.172% fraud price, and mirroring the magnificence imbalance frequently met in actual credit score card transactions. This dataset gives a combination of numerical functions because of Principal Component Analysis (PCA) to enhance safety, alongside original features 'Time' and 'Amount' imparting temporal and transactional dimensions. The limitations we confronted in this dataset consist of the class imbalance throughout version education, decreased statistics due to PCA, and potential omissions of sure capabilities or contextual details because of confidentiality worries. The dataset's temporal and geographical specificity to September 2013 and European cardholders may also affect the generalizability of findings. As we understand and navigate those limitations, the knowledgeable development and assessment of a robust credit card fraud detection version.



**Limitation**

The dataset employed in this paper has a limitation that should be corrected for better model performance. First, we need to solve the class imbalance within the dataset, with a mere 0.172% fraud rate, poses a challenge for robust model training. Additionally, the dataset's temporal and geographical specificity, focused on European transactions in September 2013, may impact the model's generalizability to broader contexts. The use of Principal Component Analysis (PCA) for feature engineering reduces interpretability, and potential confidentiality concerns might result in omitted features or contextual details. Furthermore, the absence of external factors and limited original features may hinder the model's ability to capture the full complexity of credit card transactions.

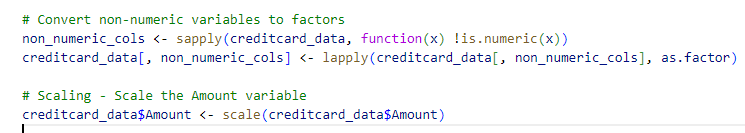
Another obstacle really worth citing is the constraint imposed with the aid of the huge dataset, which avoided us from using cloud-based platforms like PostCloud. Due to memory troubles, trying to work with the dataset in a cloud surroundings brought about crashes. As a result, the task became carried out with the use of Visual Studio Code, which presented a greater strong environment for dealing with the dataset and undertaking analyses.

**3. USEFULNESS**

The model we developed for fraud detection algorithm detects fraudulent activities in real-time transactions and prevents money loss for both customer and bank covering the overall security infrastructure. As this focuses on prioritizing security, this project provides customers and financial institutions with a trusted platform and provides confidence in the financial ecosystem. Strong encryption methods and secure ways of transmitting data protect important financial details, guaranteeing confidentiality and reliability.

**4. DATA CLEANING**

Data cleaning in R involves making dataset quality and integrity. The dataset cleaning is getting ready for our usage in the model. Firstly, it addresses missing values, keeps information integrity and accuracy and it additionally gets rid of duplicates, standardizes variable formats, and filters outliers to enhance evaluation reliability. By validating information entries and correcting discrepancies, facts consistency is ensured. The data cleaning complements dataset satisfactory for enhancing accuracy rate of detecting fraud transaction and enhancing the validity of challenge findings.





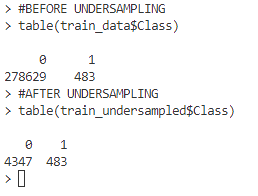


This screenshot shows the converting of the non-numeric variables of column ‘creditcard\_data’ to factors, ensuring categorical data compatibility. It also scales the "Amount" variable, standardizing numeric values for consistent analysis in the credit card dataset.

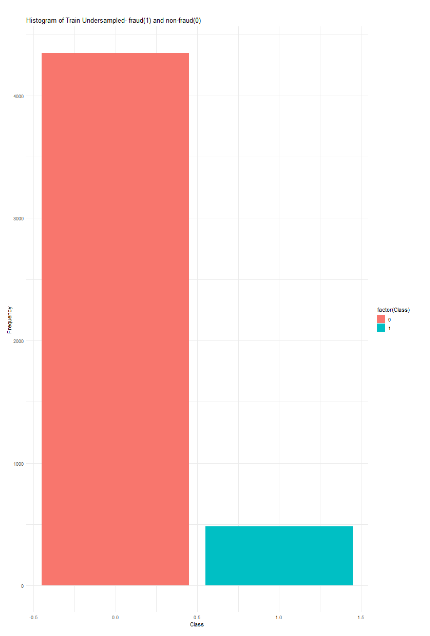
**Undersampling**

To solve this limitation class imbalance, we did an under-sampling strategy and implemented to rectify the disproportionate representation of fraudulent and non-fraudulent transactions within the training dataset.

This method randomly selects a subset of the majority class instances (non-fraudulent transactions) to equalize their numbers with those of the minority class (fraudulent transactions).



The "ovun.sample" function from the ROSE package is the method we employed to perform the undersampling and we also generated a balanced training dataset labeled as "train\_undersampled."



The histogram screenshot illustrates fraud and non-fraud undersampling data, displaying a bar plot of distribution between fraudulent and non-fraudulent transactions. Before performing undersampling, the training data shows a drawback of class imbalance with 278,629 observations in the majority class (non-fraud) and only 483 observations in the minority class (fraud). To solve this, we used undersampling technique in the training data and it comprises 4,830 samples, while the test set contains 5,695 samples. This balanced distribution of fraud and non-fraud data improves the stage for the subsequent stages of model training and evaluation. The class distribution was rebalanced, resulting in 4,347 observations for the majority class and 483 observations for the minority class after the undersampling technique. We started this process with these preprocessing steps as it aims to address the imbalance, for equal representation of both classes for training a machine learning model. Then, Adjusting the training/test split ratio may further enhance the balance in class distribution for improved model evaluation on the test set and by diminishing the prevalence of non-fraudulent data in the training data, the under-sampling technique aimed to reduce the dominance of the majority class and prevent model bias toward predicting non-fraudulent transactions.

**5. DATA EXPLORATION AND ANALYSIS**

The dataset on fraudulent activities in credit card transactions is further analyzed with the aid of us using R programming. We need to realize the dataset summaries and information, to further expand our model and so generated the usage of the capabilities like 'str ()' and 'table ()' shows the dataset's structure and class distribution, presenting a preliminary information of its composition and the prevalence of fraud.

In the numeric summary provided of our credit card transaction dataset, key variables including 'Time', 'V1' to 'V28', 'Amount', and 'Class' are provided, shedding mild on their distribution and characteristics. This dataset contains 1725 entries, offering a snapshot of diverse transaction attributes. This numerical exploration serves as a foundational step in the dataset's composition.

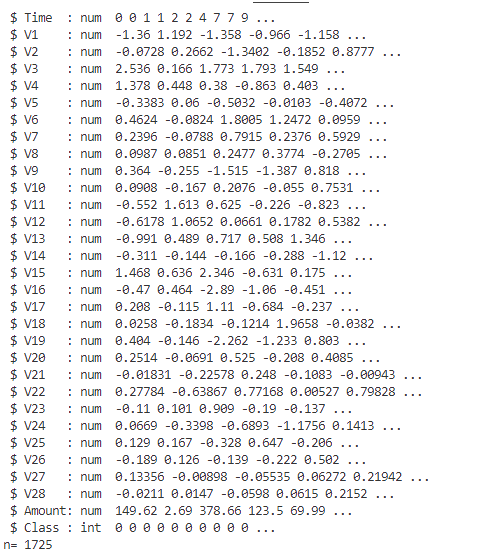
**Time:** It Represents the time elapsed since the first transaction in the dataset.

**V1 to V28:** This is the numeric variables resulting from a dimensionality reduction process. Due to confidential reasons these data were not provided as it is mentioned in Kaggle source.

**Amount:** This denotes the transaction amount for each entry.

**Class:** This column indicates the class label, with '0' typically representing non-fraudulent transactions and '1' representing fraudulent transactions.

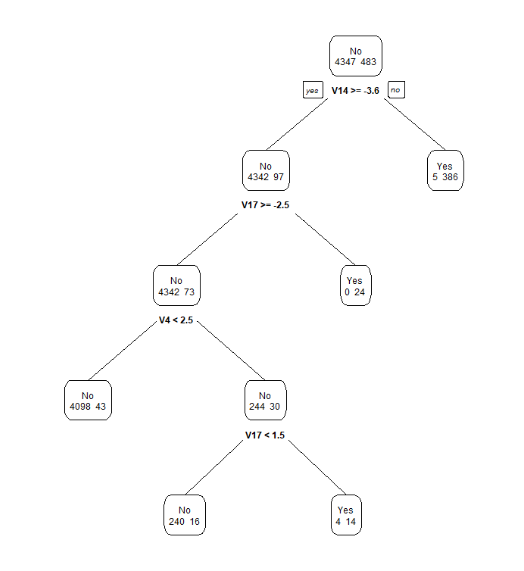
These numerical summaries, information visualization techniques together with 'prp()' function are hired to visualize the decision tree model, presenting a visual representation of its structure and decision-making process. In addition to the decision tree visualization, we use other data visualization strategies inclusive of charts and plots to discover relationships among variables and discover potential trends. We also want to understand the scatter plots, histograms, and box plots, which are particularly useful for visualizing the distribution of numeric variables and detecting outliers or unusual patterns.



The exciting component we found out is heatmaps and correlation matrices which can screen correlations among variables. This topic visually exploring the dataset helps to identify patterns or anomalies that may improve further investigation, enhancing the effectiveness of fraud detection algorithms. These integrating summaries, statistics, and data visualization makes the program provide a complete exploration of the dataset, to the extraction of thrilling and revealing perspectives on credit card fraud. This method permits us to benefit from a deeper understanding of the data, discover potential predictors of fraudulent activity, and facilitates to create greater accurate and effective fraud detection models. As we do thorough data exploration and analysis, we can find valuable information and details that inform decision-making and pressure improvements in fraud detection strategies.

**6. MACHINE LEARNING METHOD**

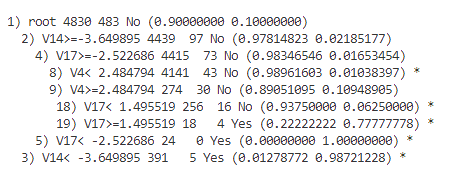
In this project we choose a machine learning method for detecting fraudulent activities in credit card transactions, as we looked for higher method used for financial information, we observed few studies papers for better analyses detailing the model's effectiveness and drawbacks. After this attention, a decision tree classifier emerged as the best preference. We like the usage of decision trees due to the fact they are terrific at showing us how choices are made. They are clean to apprehend and help us see precisely what elements are critical for detecting fraud. This aligns with the task's goal of correctly pinpointing fraudulent transactions while furnishing intelligible rationales, essential for effective risk management in financial institutions.



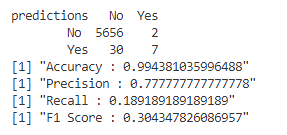
**Finally, we deliberate to work with the decision tree model because they are good at handling complex relationships and interactions betwe****en distinct features and for offering better outcomes within the project. To work with decision trees also stems from their adaptability in handling non-linear relationships and interactions among features, making them ideally suited for the project. In the next tiers of our task, we caught to an established methodology, related to preprocessing, training, and testing the model. We now begin with the credit card transaction data from Kaggle source, a comprehensive dataset reflecting various transaction types and patterns. This data preprocessing addressed some problems consisting of missing values, duplicates, and inconsistencies, guaranteeing the quality of the dataset. The feature engineering suggests extracting pertinent information such as transaction amount, merchant category, and timing, aiming to unearth valuable information for fraud detection.**

**7. MODEL EVALUATION**

In the model's evaluation phase, we perform training and assessment of a decision tree model designed for credit card fraud detection. Prior to model development using the rpart package, essential data preprocessing steps, together with scaling and the introduction of a target variable, are accomplished. The model's overall performance is then evaluated on the test set, employing metrics including accuracy and precision. An exact breakdown of predictions is supplied through a confusion matrix, providing information into the model's efficacy in figuring out fraudulent activities.



Accuracy measures the overall correctness of predictions, while precision assesses the accuracy of positive predictions in a binary classification model. A high accuracy suggests overall correctness, while high precision signifies accurate identification of positive instances among predicted positives.



The assessment results of our pruned decision tree model demonstrate commendable standard performance. The confusion matrix shows that the model correctly predicted most transactions, with 5,656 identified as non-fraudulent and 7 correctly identified as fraudulent. However, it additionally misclassified 30 non-fraudulent transactions as fraudulent and missed 2 actual fraudulent transactions. We observed the key performance of metrics highlights the model's effectiveness: an accuracy of 99.44%, indicating the model's correctness across predictions. The precision of 77.78% underscores the reliability of the model when it predicts a transaction as fraudulent, the recall (sensitivity) of 18.92% indicates an ability trouble in capturing all actual fraudulent cases, as the model missed a notable portion. The F1 Score, balancing precision, and recall, stands at 30.43%, indicating a trade-off between correctly identifying actual fraud cases and minimizing false positives. We realize that the high precision underscores the model's accuracy in detecting actual fraud, the lower recall suggests there is room for improvement in capturing a higher proportion of true fraud cases.

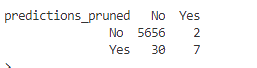
**Confusion matrix**

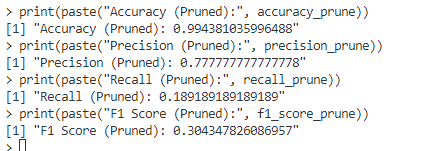
The confusion matrix plays an important position in assessing the effectiveness of a binary type of version specifically in the realm of fraud detection. The confusion matrix has four vital components- True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). True Positives constitute information in which the version correctly identifies fraudulent transactions, True Negatives denotes correct predictions of non-fraudulent transactions. False Positives occur even as the model incorrectly shows non-fraudulent as fraudulent, and False Negatives arise whilst it incorrectly categorizes fraudulent as non-fraudulent as those define the confusion matrix in specific way. To calculate the critical metrics like accuracy, precision, recall (sensitivity), and specificity, accomplished from confusion matrix values.

**8. MODEL IMPROVEMENT**

The output of the original and pruned decision tree models on the test set shows us the overall performance metrics. We find both models results are a commendable accuracy of 99.44%, suggesting a high degree of overall correctness in predictions. To know more about this, we did a deeper analysis, and it exposes positive demanding situations in handling fraudulent transactions, as indicated by way of a precision of 77.78%, recall of 18.92%, and an F1 Score of 30.43%.

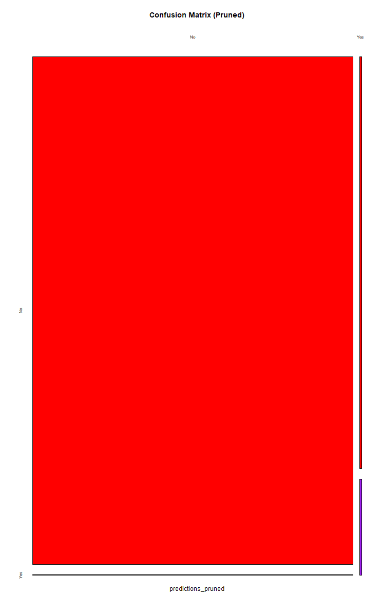
We see the metrics highlight a potential trade-off between precision and recall, emphasizing the desire for cautious attention based on precise use-case requirements. The precision metric reflects the accuracy of positive predictions, while recall measures the ability to identify all actual positive instances. The model's current configuration shows a risk of missing an extensive portion of fraudulent activities, emphasizing the importance of similarly optimization. Both the original and pruned models show consistent performance, indicating that pruning did no longer appreciably affects the assessment metrics.





**Cross-validation**

Cross-validation is achieved primarily based on the decision tree models result and uses cost-complexity. The cost-complexity pruning is executed for preventing overfitting in decision tree models, this is a common situation when dealing with complex datasets. While we have been working with cross-validation, we had a few challenges associated with fine-tuning the "complexity parameter" (cp). The parameter is managed from low to high values between 0 to 1. This parameter says how plenty pruning is executed for the dataset, with the higher values gives more severe pruning. After few changes, finding the right balance in adjusting the "cp" parameter was problematic because setting it too high should oversimplify the model, lacking important patterns within the data and placing it too low may want to cause inflicting overfitting and lowering performance on the test set. This process is accomplished with careful modifications of parameters and clean assessment to make certain the pruned decision tree model performed optimally higher. To avoid overfitting, we now set the "cp" value for pruning at 0.01, indicating low complexity. We also evaluated the pruned decision tree model on the test set, computing various metrics like accuracy, precision, recall, and F1-score to understand its effectiveness in identifying fraudulent transactions. This iterative manner of cross-validation was crucial for ensuring the model's robustness and generalizability. Obtaining results from cross-validation informed the final decision on the "cp" parameter, made the model working successfully. This code carried out in pruning adjusts the "cp" parameter, representing the cost complexity of the tree. Tuning this parameter unearths the proper balance between model complexity and performance on unseen data.



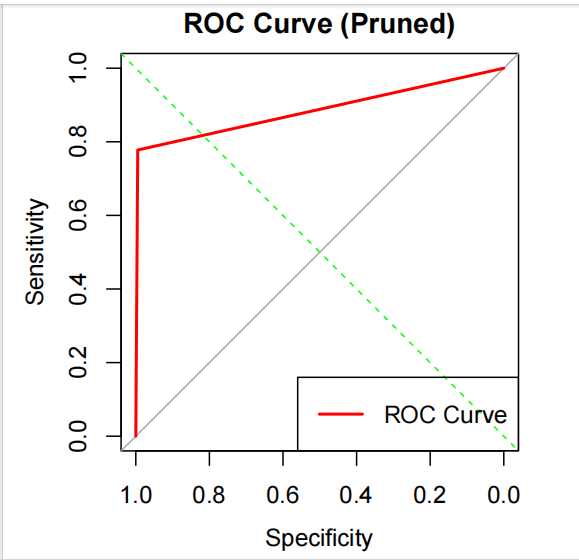
**Evaluating Fraud Detection with ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve)**

For representing credit card fraud detection, we use the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC). These are the tools used for comparing binary classification models. The ROC curve indicates the delicate balance between sensitivity (true positive rate) and specificity (true negative rate) across various classification thresholds. AUC is a quantitative degree by using calculating the area under the ROC curve.



In the realm of fraud detection, credit card datasets often face class imbalances, with fraudulent transactions being much less common. AUC and ROC, being robust metrics, provide a comprehensive evaluation of model performance under such conditions. With values starting from 0 to 1, a higher AUC signals superior model performance, 0.5 suggests random classification, and values under 0.5 imply suboptimal overall performance. ROC and AUC are instrumental in providing a holistic understanding of a model's capabilities across different thresholds. They are specifically precious in scenarios with imbalanced classes or nuanced model uncertainty.

For our pruned decision tree model, achieving an AUC of 0.886 indicates robust discrimination between fraud and non-fraud instances. The visually putting ROC curve similarly emphasizes the model's effectiveness in distinguishing across various classification thresholds. These metrics together verify the pruned decision tree's prowess in identifying fraudulent transactions, establishing it as an asset in the realm of fraud detection.



**9.** **REFERENCES**

MLG-ULB. (2018). Credit Card Fraud Detection. Kaggle. <https://www.kaggle.com/mlg-ulb/creditcardfraud>

Dal Pozzolo, A. (2015). “Adaptive Machine Learning for Credit Card Detection.” PhD thesis.

Safa, M. U., & Ganga, R. M. (2019). Credit Card Fraud Detection Using Machine Learning. International Journal of Research in Engineering, November-2019.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed Data Mining in Credit Card Fraud Detection. IEEE Intelligent Systems, pp 67-