Fraudulent Activity Detection in Credit Card Data Using Data Mining Techniques

Milestone: Project Report

Group 6

Student 1	**Yaswanth Reddy Nalamalapu**
Student 2	**Hrudhvik Nangineni**
Student 3	**Udhay Chityala**

617-777-5405 (Tel of Student 1)
703-740-6758 (Tel of Student 2)
857-381-5521 (Tel of Student 3)

nalamalapu.y@northeastern.edu
nangineni.h@northeastern.edu
chityala.u@northeastern.edu

Percentage of Effort Contributed by Student 1: 33.33%

Percentage of Effort Contributed by Student 2: 33.33%

Percentage of Effort Contributed by Student 3: 33.33%

Signature of Student 1: **Yaswanth Reddy Nalamalapu**

Signature of Student 2: **Hrudhvik Nangineni**

Signature of Student 3: **Udhay Chityala**

Submission Date: **04-23-2024**

Abstract:

For the past few years, data mining algorithms have been employed by financial institutions such as banks and insurance companies. These organizations utilize data mining techniques in various applications, including predicting business failures, analyzing marketing trends, and detecting fraudulent activities. In this study, our aim is to conduct an analysis to identify some effective data mining techniques like Random Forests, for detecting fraud in Credit card usage and payments. We will evaluate and compare these techniques based on specific criteria to determine their suitability.

Problem Setting:

Credit card fraud is a growing concern in the credit card industry, with data accessed through automated teller machines (ATMs), barcode readers in stores, banks, and online banking systems. The main goal is to identify several types of credit card fraud and evaluate different detection methods. Financial institutions like banks and credit card companies face several types of fraud, leading to the exploration of strategies to reduce these incidents. This type of fraud affects both businesses and individuals, causing mistrust and increased living costs. Despite efforts to detect fraud by analyzing past transactions, the evolving tactics of fraudsters make detection challenging. Ongoing research aims to find effective ways to prevent and combat financial fraud.

Also, as one of the avenues used, data mining is a method that uses techniques such as statistical analysis, machine learning, and mathematical methods to extract valuable insights from large datasets. Specifically for fraud detection, statistical methods are typically grouped into supervised and unsupervised techniques. In supervised methods, models are trained using samples of both fraudulent and legitimate transactions to classify new transactions as fraudulent or legitimate. This approach harnesses the power of data mining to effectively detect financial fraud.

Problem definition:

The Specific problem analyzed in this project is on Fraud detection in Credit Card Transactions using data mining methods are:

- What features or combinations are most influential in determining Frauds?
- What Data mining tasks to be used on the dataset for effective results to be generated?
- How do different machine learning algorithms compare in predicting whether it is a fraud or not?
- What aspects of the statistical metrics generated are considered and which Values have more importance?
- How well does the developed model generalize to new test data or real-world data?

By addressing these questions through data analytics and machine learning techniques, we aim to identify which of the ones used is a robust predictive model that can reliably classify the Frauds in Credit Card Transactions.

Data Sources:

Credit Card Transactions Fraud Detection Dataset (kaggle.com)

(https://www.kaggle.com/datasets/kartik2112/fraud-detection)

The dataset is a simulated credit card transaction dataset generated using the Sparkov Data Generation tool created by Brandon Harris, taken from Kaggle where the simulator comes with a set list of merchants, customers, and types of transactions and the using a Python library called "faker", it creates a new list based on the number of customers and merchants specified for the simulation.

Data Description:

The dataset contains both legitimate and fraudulent transactions for 1000 customers across 800 merchants, covering the period from January 1, 2019, to December 31, 2020.

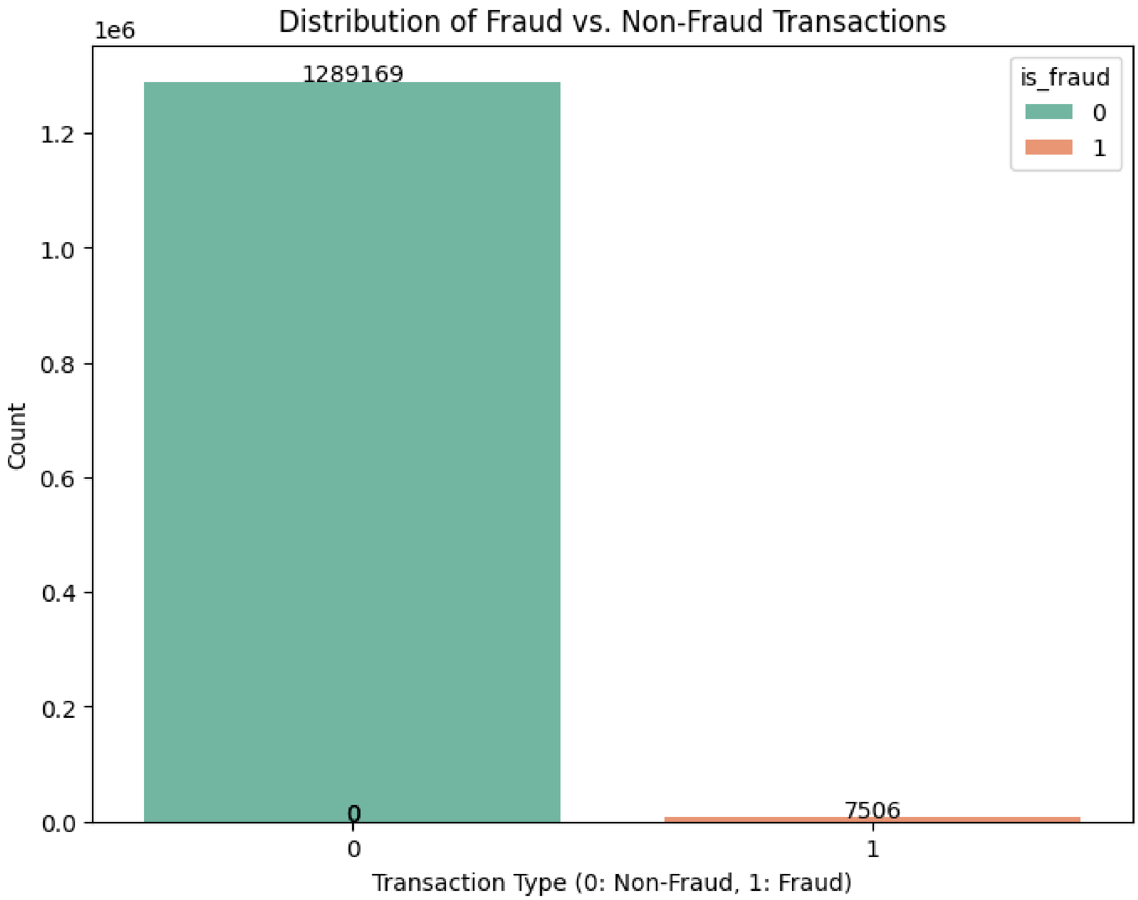
Information regarding the dataset:

- Number of Rows: 1.856 million rows (556k in test set, 1.30 million in train set)
- Number of Columns: 23 columns (features or variables)

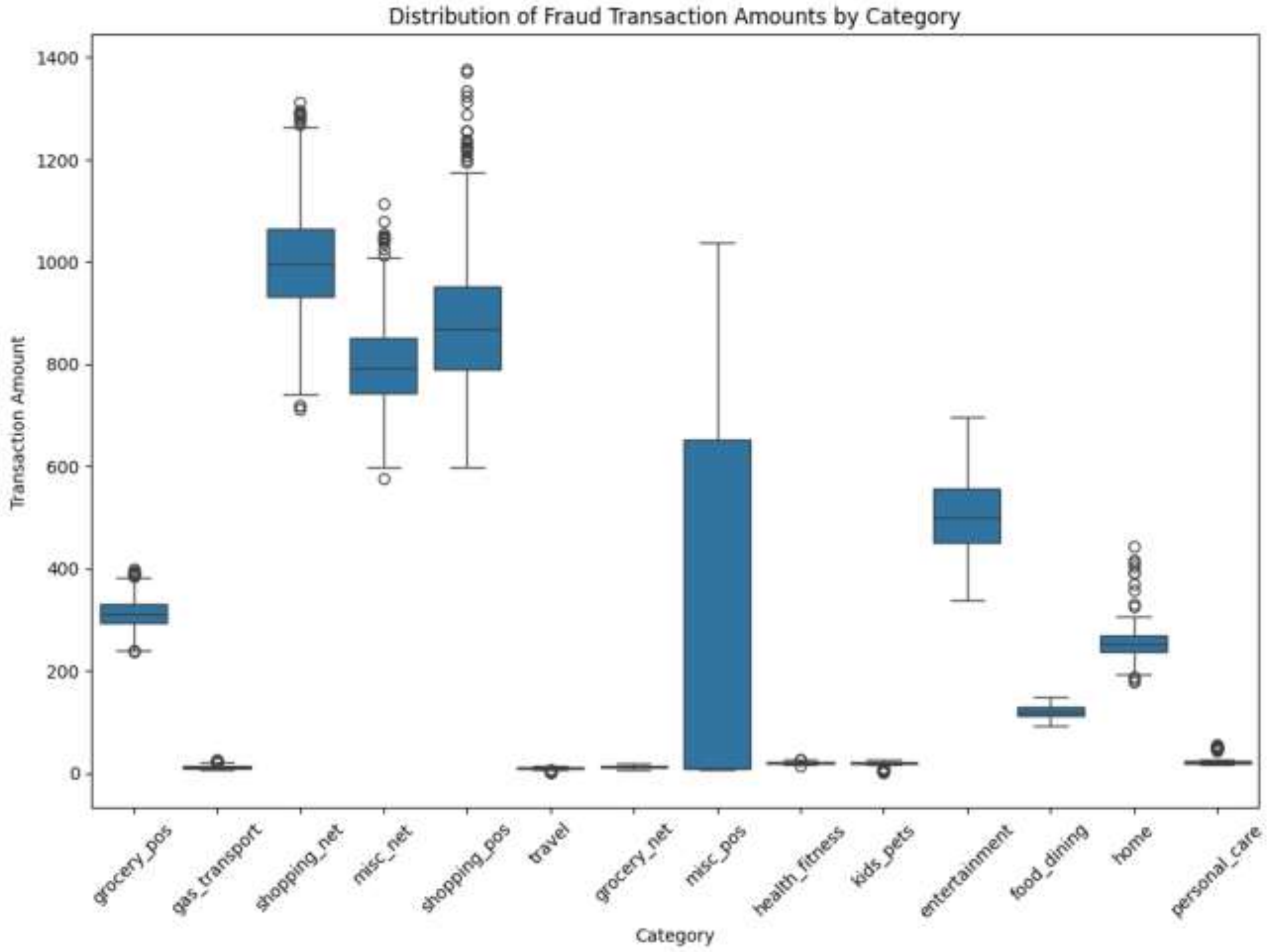
Sample Variable Names:

- index - Unique Identifier for each row
- trans_date_trans_time - Transaction DateTime
- cc_num - Credit Card Number of Customer
- merchant - Merchant Name
- category - Category of Merchant
- amt - Amount of Transaction
- first - First Name of Credit Card Holder
- last - Last Name of Credit Card Holder
- gender - Gender of Credit Card Holder
- street - Street Address of Credit Card Holder
- city - City of Credit Card Holder
- state - State of Credit Card Holder
- zip - Zip of Credit Card Holder
- lat - Latitude Location of Credit Card Holder
- long - Longitude Location of Credit Card Holder
- city_pop - Credit Card Holder's City Population
- job - Job of Credit Card Holder
- dob - Date of Birth of Credit Card Holder
- trans_num - Transaction Number
- unix_time - UNIX Time of transaction
- merch_lat - Latitude Location of Merchant
- merch_long - Longitude Location of Merchant
- is_fraud - Fraud Flag <--- Target Class

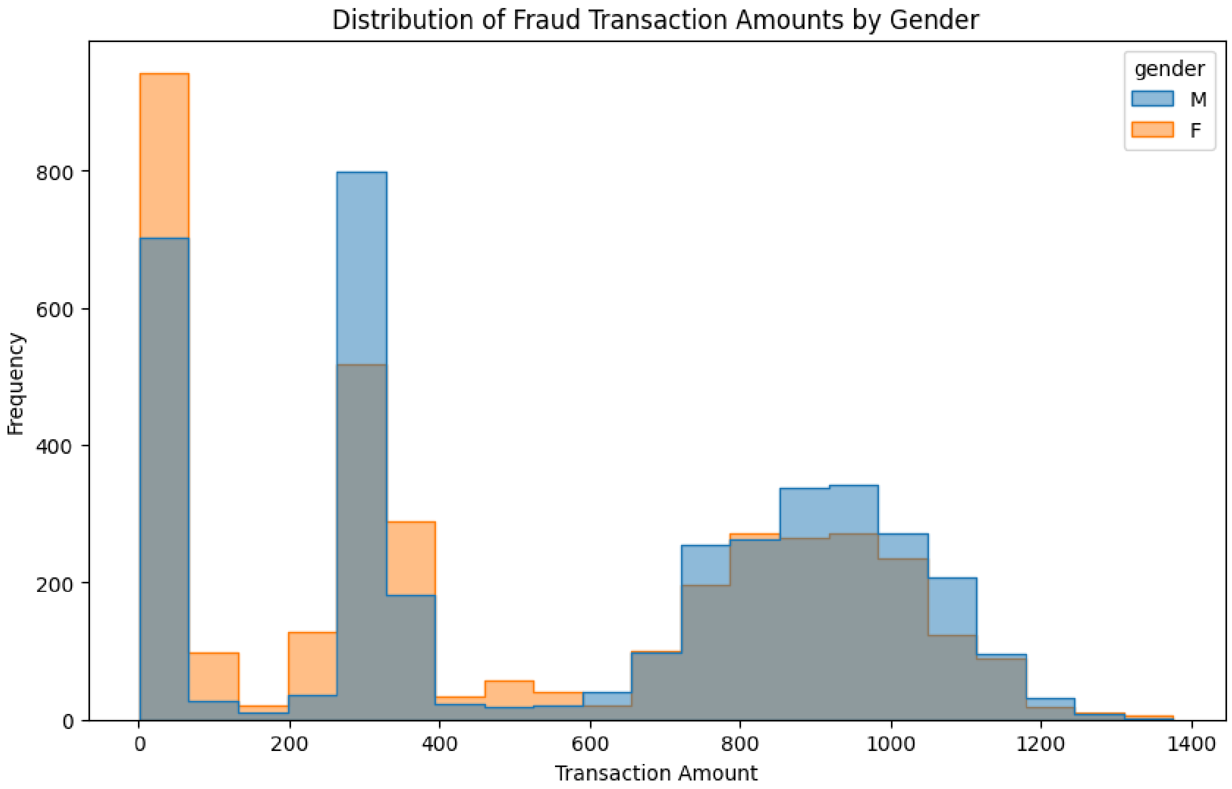
Data Exploration:



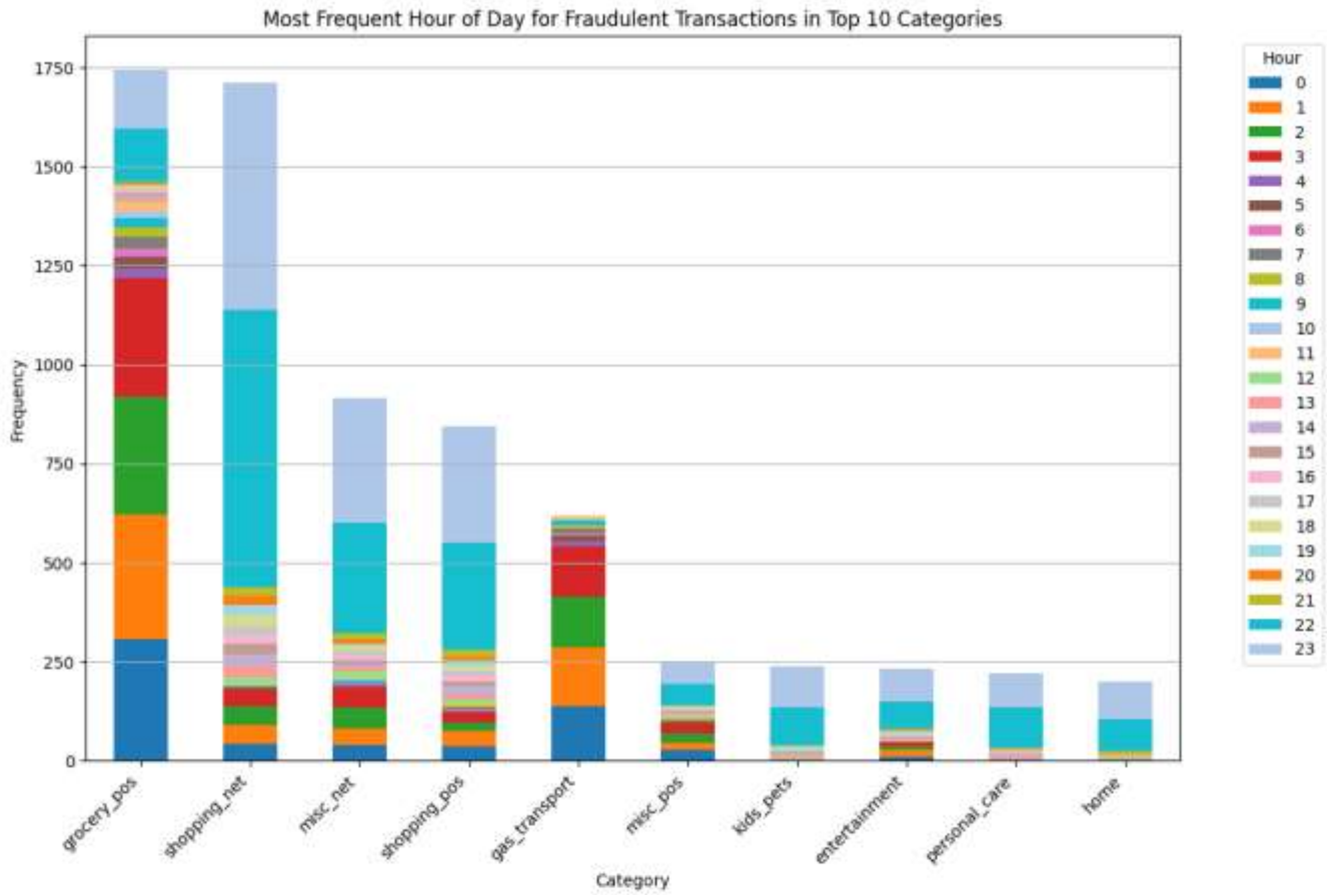
- The dataset is highly imbalanced, with most transactions being non-fraudulent (0). There are significantly fewer fraudulent transactions (1) compared to non-fraudulent ones.
- This imbalance in the data may need to be addressed through techniques like oversampling or undersampling for effective fraud detection modeling.



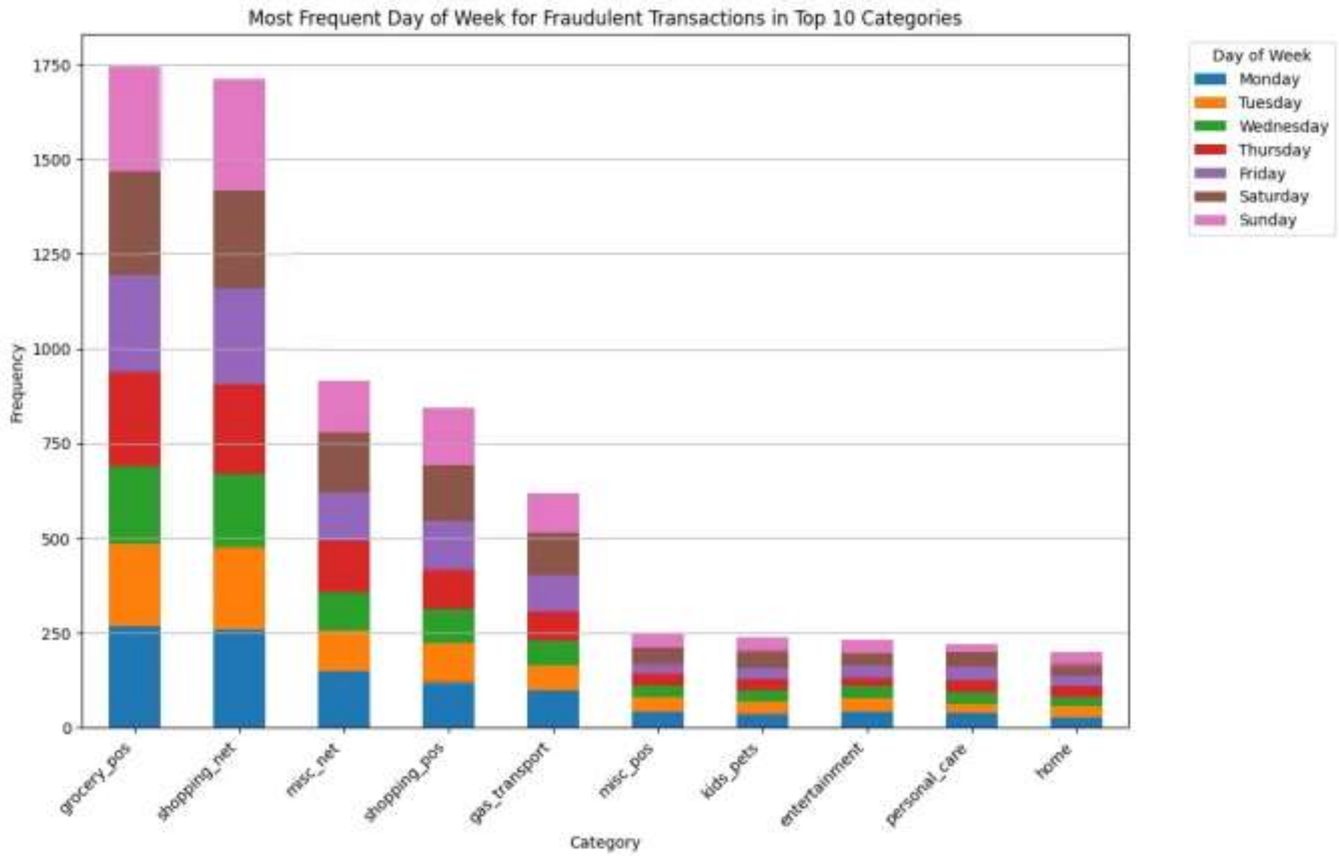
- The "personal_loan" category seems to have the highest fraud transaction amount, significantly higher than other categories.
- Categories like "medical_aid", "fitness_app", and "fuel_car" appear to have relatively lower fraud amounts.
- There is a wide range of fraudulent transaction amounts across categories, with some categories having higher variability than others.
- Certain categories, such as "kid_app" and "pet_app," have outliers with relatively high fraud transaction amounts compared to the rest of the data points in those categories.



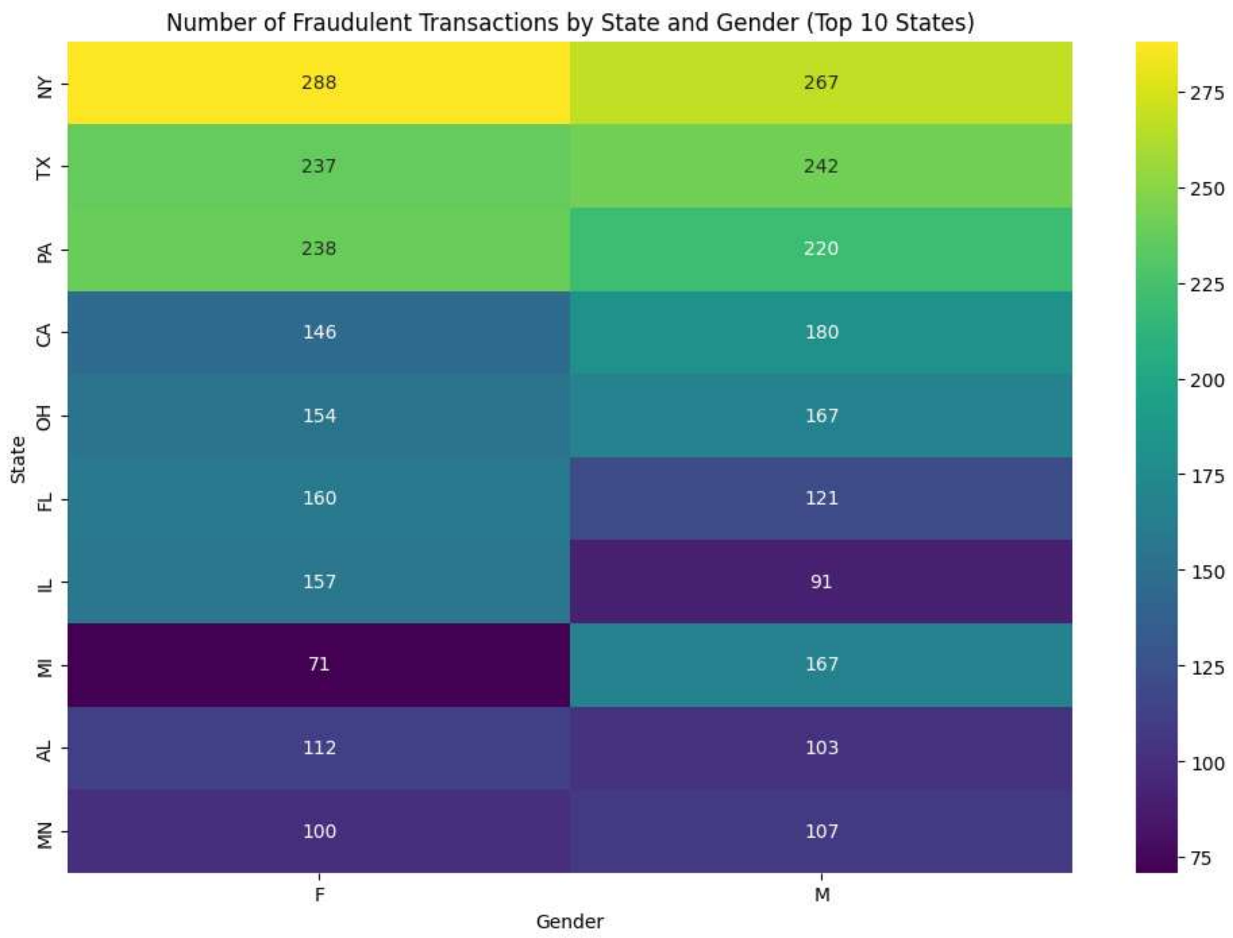
- There appears to be more fraud transactions, with higher amounts committed by males compared to females, as indicated by the taller bars towards the right side for males.
- Both genders exhibit a right-skewed distribution, with a higher frequency of lower transaction amounts and a few outliers with remarkably high transaction amounts.
- The highest fraud transaction amount is committed by a male, as evident from the rightmost bar for the male category.
- The distribution for females has a slightly lower variance compared to males, suggesting that fraud transaction amounts committed by females tend to be more clustered around lower values.



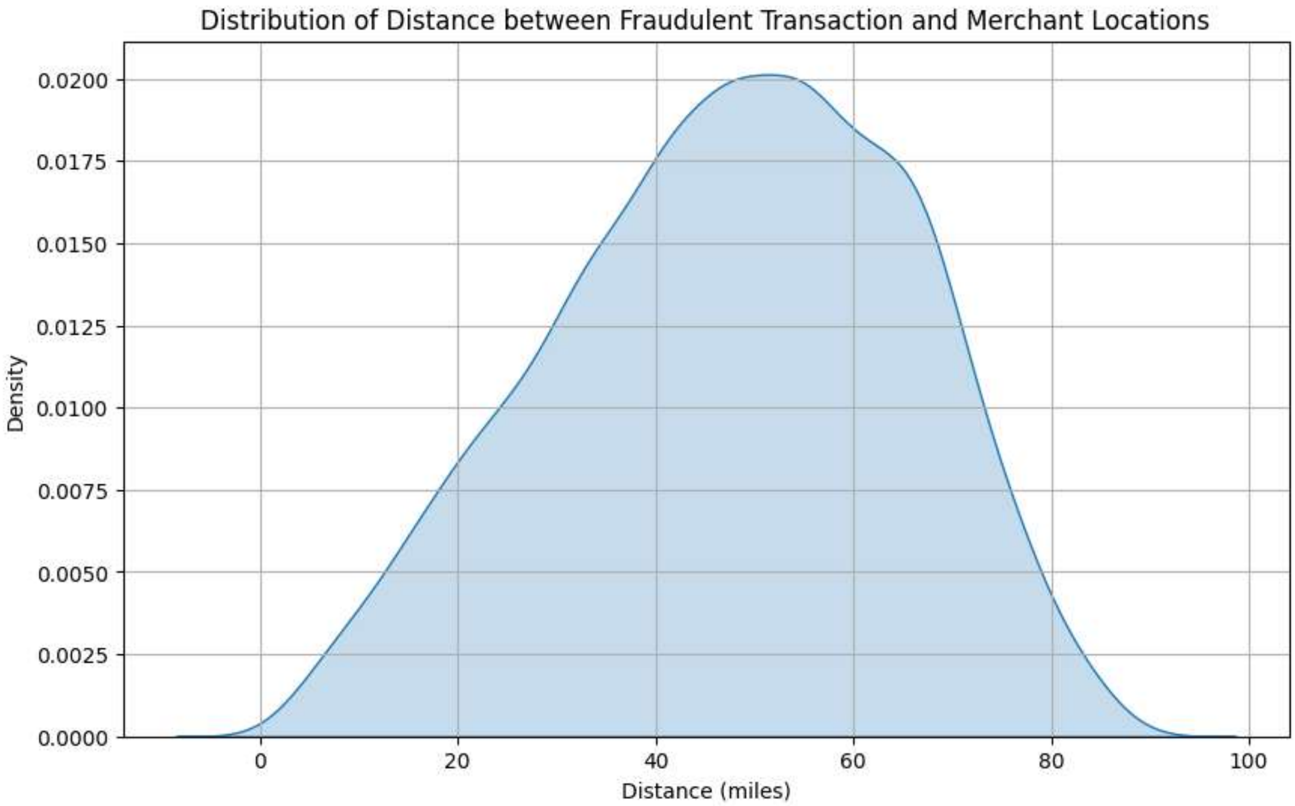
- The "grocery_pos" category stands out with the highest frequency of fraudulent transactions occurring at midnight (0th hour).
- Categories like "shopping_net" and "misc_net" show high frequencies during the late evening hours (9-11 PM).
- The "personal_care" category seems to have an even distribution of frequent fraud hours across various times of the day.
- Some categories, such as "entertainment" and "kids_pets," exhibit a broader distribution of frequent fraud hours, spanning from late night to early morning.
- The "home" category seems to have a low frequency of fraudulent transactions across all hours.



- Most fraudulent transactions occur on Mondays across the top 10 categories.
- Categories like "Grocery Stores" and "Shopping Clubs" show a higher frequency of fraudulent transactions on Mondays.
- Some categories like "Kids' Apparel" and "Personal Care" exhibit a more uniform distribution across weekdays.



- New York (NY) has the highest number of fraudulent transactions for both genders (male and female).
- For most of the top 10 states, the number of fraudulent transactions is higher for males compared to females.
- The difference in fraudulent transactions between genders is relatively small for states like Pennsylvania (PA) and Ohio (OH).



- The distribution of distance between fraudulent transaction and merchant locations follows a bell curve (normal distribution).
- The highest density is observed around the middle range (30-70 miles), suggesting fraudulent transactions often occur within a moderate distance from merchant locations.
- The distribution tapers off at both extremes, indicating fewer fraudulent transactions occur at very short or very long distances from merchant locations.

Data Mining Tasks:

1. Data Handling:

Since the dataset does not contain any missing values, preprocessing was initiated without the need for imputation techniques.

2. Data Preprocessing:

The dataset underwent several preprocessing steps. Firstly, columns containing private customer data such as 'trans_num', 'first', 'last', 'unix_time', 'dob', and 'cc_num' were dropped due to redundancy or irrelevance for the analysis.

Next, label encoding was applied to categorical columns including 'merchant', 'category', 'gender', 'street', 'city', 'state', and 'job'.

Additionally, temporal features such as 'hour', 'minute', 'day_of_week', 'day_of_month', 'month', and 'year' were extracted from the 'trans_date_trans_time' column to enhance the dataset's temporal analysis.

Given that most of the remaining columns are categorical, normalization was deemed unnecessary in this scenario, except for the 'amt' column.

3. Handling Imbalance Data:

The dataset presents a significant class imbalance issue, with non-fraudulent cases outnumbering instances of fraud. This skew towards the majority class poses a significant challenge in effectively training models to detect the minority class.

To address this class imbalance, a hybrid approach combining oversampling and undersampling techniques was employed. Specifically, Synthetic Minority Over-sampling Technique (SMOTE) was utilized for oversampling, while Random Undersampling was employed for undersampling. This strategy aims to mitigate the class imbalance and improve the model's ability to accurately detect instances of fraud.

4. Train Test Split:

The dataset was split into three subsets for training and evaluation purposes. Seventy percent of the data was allocated for training the machine learning models, while 15% was set aside for validation and another 15% for testing.

This partitioning strategy ensures that the models are trained on a sufficiently large portion of the data while also allowing for independent evaluation on unseen data. The validation set aids in fine-tuning model hyperparameters and monitoring performance during training, while the test set provides a final assessment of the model's generalization capability on new, unseen data.

5. Model Training and Evaluation:

A variety of machine learning algorithms, including Random Forest, Naive Bayes, XGBoost, and Multi-Layer Perceptron (MLP), were implemented and assessed on the training, validation, and test datasets.

Each algorithm underwent training using the training subset and was subsequently evaluated on the validation subset to fine-tune parameters and assess performance. Finally, the models were tested on the independent test subset to gauge their ability to generalize unseen data.

This comprehensive evaluation process allows for a thorough comparison of the performance of different algorithms and facilitates the selection of the most suitable model for the specific task at hand.

Data Mining Models/Methods:

For this project, the entire training dataset was utilized to train various models, while the test dataset was split evenly to form validation and test sets for evaluating model performance.

The models employed in this project are:

1. Random Forests:

This versatile algorithm utilizes ensemble learning by constructing multiple decision trees using bootstrapped samples of the training data and random subsets of features. By aggregating the predictions of these trees, Random Forest achieves robustness and high accuracy, making it effective in handling high-dimensional data and mitigating overfitting.

2. Naive Bayes:

Based on Bayes theorem, this classifier selects the decision with the highest probability. It calculates posterior likelihood by using known values and probabilities, providing a method for determining the probability of an outcome given certain conditions.

3. XGBoost:

A powerful gradient boosting library, XGBoost builds an ensemble of decision trees sequentially, with each new tree correcting the errors of the previous ones. Known for its excellent performance, scalability, and flexibility, XGBoost achieves state-of-the-art results in various machine learning competitions and real-world applications.

4. MLP (Multilayer Perceptron):

A type of feedforward artificial neural network, MLP consists of input, hidden, and output layers with interconnected nodes. Nonlinear activation functions in the hidden and output layers enable the network to learn complex patterns. MLPs are trained using backpropagation and gradient descent to minimize the difference between predicted and actual outputs.

In conclusion, the chosen data mining models and methods prove effective for addressing the problem of fraud detection, offering a diverse range of approaches to tackle the issue.

Performance Evaluation:

In an unbalanced dataset for fraud detection, where the number of non-fraudulent transactions (majority class) is much higher than fraudulent transactions (minority class), the following metrics and AUCs are often considered more important:

Precision, Recall, F1-Score, and the derivation of True Negative and False Positive Rates from the confusion matrix, along with AU-ROC & PR-AUC.

In fraud detection, precision indicates how many of the predicted fraudulent transactions are fraudulent and high precision means fewer false alarms. Also, recall indicates how many of the actual fraudulent transactions were correctly predicted and high recall means fewer missed fraudulent transactions.

Here, the Data set is used directly without handling processes on the model, and the performance metrics generated by this led us to process the data before training the model for better optimization.

Project Results:

I. Performance of Machine Learning models on Imbalance data

- Validation Evaluation:

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	1.0	0.96	0.8	0.86	0.8	0.76
Naive Bayes	0.99	0.61	0.73	0.65	0.73	0.35
XG Boost	1.0	0.92	0.8	0.85	0.8	0.72
MLP	1.0	0.78	0.5	0.5	0.5	0.29

- Test Evaluation:

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	1.0	0.96	0.8	0.86	0.8	0.76
Naive Bayes	0.99	0.62	0.73	0.65	0.73	0.35
XG Boost	1.0	0.92	0.82	0.85	0.82	0.74
MLP	1.0	0.83	0.5	0.5	0.5	0.34

- Performance on Imbalanced Data:

Random Forest consistently showed excellent performance across all metrics with Accuracy at 1.0 which may be due to overfitting and recall at 0.8 on the Validation set. Naive Bayes performed well in terms of Accuracy, which may be due to classifying everything with the majority class due to its high probability but lagged in Precision and Recall. XG Boost also demonstrated robust performance, especially in Precision and F1-Score. MLP has struggled with Recall and F1-Score, indicating difficulty in identifying fraud cases.

Here Random Forests and XG Boost performed adequately in comparison to others, which may be due to their nature of generating various rules for model performance to check different possibilities.

Even though all four models boast high accuracies, hovering around 100%, we notice differences in other measures. This happens because our data is lopsided, with very few instances of fraud compared to non-fraud. Random Forest and XGBoost stand out in spotting fraudulent cases better, with higher precision and recall. This could be because they use decision trees, which are good at making rules for classification. On the flip side, Naive Bayes doesn't perform as well. It relies on probabilities, and since our data has a lot more non-fraud cases, it tends to favor those when predicting. Multilayer Perceptron (MLP) also struggles. Trained mostly on non-fraud data, it tends to predict non-fraudulent outcomes for almost every case. These are just our initial thoughts based on the data, but more analysis is needed for a complete understanding.

II. Performance of Machine Learning models After Handling Imbalance data

1. Oversampling 20% and Undersampling 20%

- Validation Evaluation:

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.95	0.97	0.85	0.90	0.85	0.87
Naive Bayes	0.91	0.88	0.75	0.8	0.75	0.73
XG Boost	0.96	0.96	0.89	0.92	0.89	0.90
MLP	0.94	0.91	0.85	0.88	0.85	0.82

- Test Evaluation`

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.95	0.97	0.85	0.90	0.85	0.87
Naive Bayes	0.91	0.88	0.75	0.8	0.75	0.73
XG Boost	0.96	0.96	0.89	0.92	0.89	0.90
MLP	0.94	0.92	0.85	0.88	0.83	0.84

2. Oversampling 20% and Undersampling 30%

- Validation Evaluation:

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.93	0.96	0.86	0.9	0.86	0.89
Naive Bayes	0.88	0.88	0.76	0.8	0.76	0.77
XG Boost	0.95	0.96	0.90	0.92	0.90	0.91
MLP	0.91	0.93	0.82	0.86	0.82	0.84

- Test Evaluation

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.93	0.96	0.86	0.90	0.86	0.89
Naive Bayes	0.88	0.88	0.76	0.80	0.76	0.77
XG Boost	0.95	0.96	0.90	0.93	0.90	0.91
MLP	0.91	0.92	0.82	0.86	0.82	0.84

3. Oversampling 20% and Undersampling 40%

- Validation Evaluation

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.92	0.95	0.87	0.90	0.87	0.90
Naive Bayes	0.85	0.88	0.76	0.79	0.76	0.79
XG Boost	0.94	0.95	0.91	0.93	0.91	0.92
MLP	0.90	0.91	0.85	0.87	0.85	0.86

- Test Evaluation

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AU-ROC**	**PR-AUC**
Random Forest	0.92	0.95	0.87	0.90	0.87	0.90
Naive Bayes	0.86	0.88	0.76	0.79	0.76	0.80
XG Boost	0.94	0.95	0.91	0.93	0.91	0.92
MLP	0.90	0.91	0.85	0.87	0.85	0.86

- Performance After Handling Imbalance with 20% Oversampling and 20% Undersampling:

After balancing the dataset, Random Forest maintained high performance with minor drops in Accuracy and Precision which says that there are no overfitting issues after data handling, but improved Recall. Naive Bayes showed improvement in most metrics but had a slight decrease in Precision. XG Boost and MLP also showed improved performance across most metrics, particularly in Recall and F1-Score. In this case XG Boost has the highest performance overall, showing its effectiveness.

- Performance After Handling Imbalance with 20% Oversampling and 30% Undersampling:

With increased undersampling to 30%, Random Forest experienced a slight decrease in accuracy and precision but an increase in recall causing a slight increase in the values of AU-ROC, PR-AUC. Naive Bayes & XG Boost also showed a similar trend with all metrics, especially Recall. MLP maintained strong performance but had minor reductions in Recall and F1-Score. Also, for this model the AU- ROC was decreased while PRAUC is increased which can mean that it is improving its performance in correctly identifying the positive class, which, in the context of fraud detection, is identifying actual fraud cases with higher precision.

- Performance After Handling Imbalance with 20% Oversampling and 40% Undersampling:

In this case where the undersampling is once again increased to 40%, every model has experienced a slight decrease in accuracy and precision but an increase in recall causing a slight increase in the values of AU-ROC, PR-AUC making this the case which has generated the highest important metrics which are used for determining the model value.

Comparing the results across various sampling depths with the original non-sampled data, we observe generally good model performance. As we delve deeper into the fraud cases by increasing their representation in the sample, accuracy improves. However, this improvement comes at the expense of higher errors, as evidenced by decreasing PRAUC and AU-ROC scores. This trade-off isn't ideal for fraud detection, where misidentification can lead to substantial losses.

Interestingly, at certain sampling depths, the performance gap between the Test and Validation sets is minimal. This suggests that the sampling techniques effectively balanced the dataset. Consequently, the models trained on these balanced datasets consistently perform well across both validation and test sets.

Impact of the Project Outcomes:

There are many Impacts that the project has highlighted, those are:

- Due to the nature of the Imbalanced data set many Machine learning models do not perform well on the dataset without any data handling methods, but there are some which may not give us best results but perform averagely like in the case of Random Forests and XG Boost.
- These two methods are considered enabled decision tree algorithms, so this type of problem performs well for the algorithms based on decision trees.
- The data handling techniques used are Random undersampling and SMOTE in combination for getting good performance across various models.
- As we increase the depth of Fraud cases by increasing the percentage of Undersampling we get better results which verifies our method of using combination of SMOTE and Random Undersampling to form a balanced dataset for performance evaluation.
- By observing the values of AUROC & PRAUC, which are our important performance metrics increasing with the percentage of Undersampling we can conclude that a higher percent of this in combination of SMOTE gives us better model performance.
- Even the lowest metrics after data handling are way better than some of the model performance metrics generated without it.

- Test and Validation sets generating almost similar metrics suggests that the sampling techniques effectively balanced the dataset.