# **Credit Card Fraud Transaction Detection**

Pratik Karunakar Poojari (NUID- 002645173)

## Introduction

### Background:

Credit card fraud is a significant concern for financial institutions and their customers. In the United States alone, credit card fraud losses were estimated to be $11 billion in 2020. Fraudulent transactions can be difficult to detect since they often involve small amounts and can be mixed with legitimate transactions. However, detecting fraud is crucial as it can help prevent financial losses and protect customers' sensitive information.

### Motivation:

The rise of online shopping and contactless payments has made credit card fraud detection more challenging. Fraudsters have become more sophisticated and can use stolen card information to make unauthorized purchases, leaving banks and their customers vulnerable. It is important that credit card companies can recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Therefore, there is a need for advanced fraud detection systems that can quickly and accurately detect fraudulent transactions.

### Goal:

The goal of credit card transaction fraud detection is to develop a model that can accurately distinguish fraudulent transactions from legitimate ones. This can be achieved using various machine learning techniques such as Random Forest Algorithm, Multi-Layer Perceptron, Linear Regression, K-Means Clustering, etc.

## Methodology

Credit card fraud detection necessitates the use of an ML-based algorithm to accomplish feature extraction and model evaluation. In this project, three machine learning techniques: Random Forest, Linear Regression, and K-Means Clustering are compared and analysed by utilizing the dataset from Kaggle that contains transactions made by credit cards September 2013 by European cardholders to create a fraud transaction detection system.

Following is the methodology employed by this project:

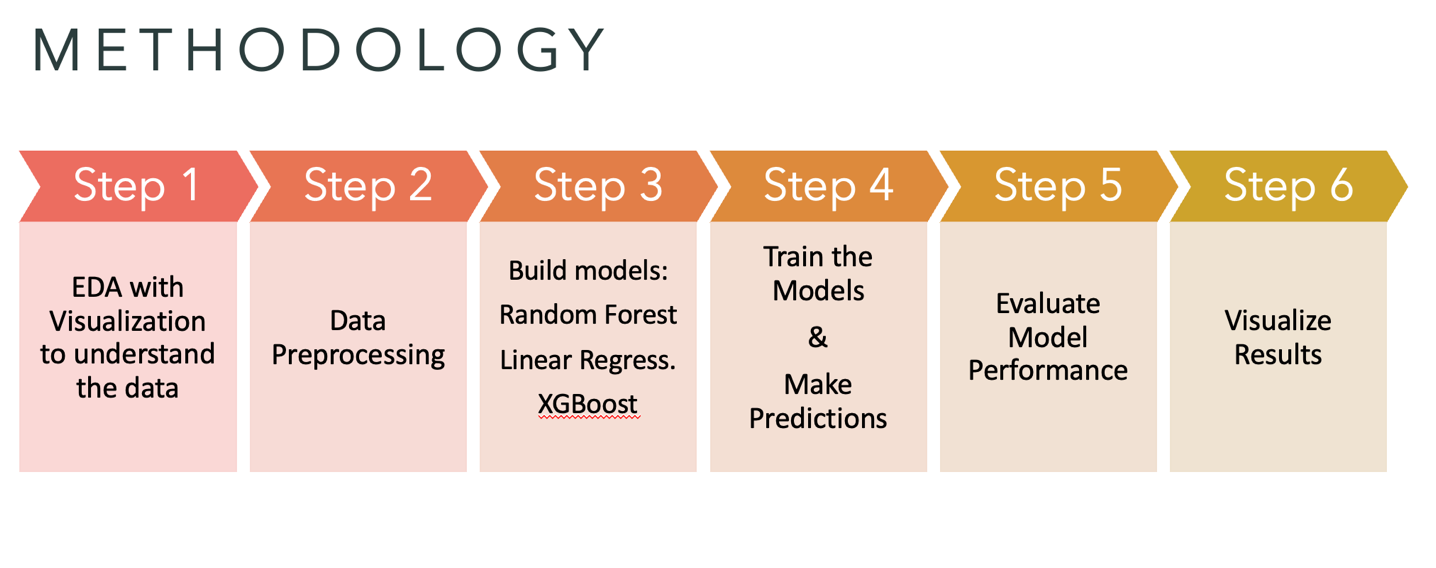


Figure : Methodology

Credit card transactions by European cardholders in September 2013 are contained in this dataset. A total of 492 frauds were detected out of 284,807 transactions that occurred over a two-day period. There is a high degree of unbalance in the dataset. There are 0.172% of all transactions that are classified as positive (frauds).

Several numerical variables are contained in it, which are the results of a PCA transformation. Data features, background information, and original features have unfortunately been withheld due to confidentiality issues. With PCA, the principal components are V1, V2, ... V28. The table below outlines the features that were not calculated through the PCA algorithm.

Table : Dataset Description

|  |  |  |
| --- | --- | --- |
| Sr no | Feature | Description |
| 1. | Time | Records the duration between the first and last transaction in the dataset |
| 2. | Amount | Amount of the transaction |
| 3. | Class | Class 0: Non-Fraudulent Transactions  Class 1: Fraudulent Transactions |

### Exploratory Data Analysis:

Several Python functions, namely head, describe, and info, were used to analyze the data's features and learn more about the structure and contents.

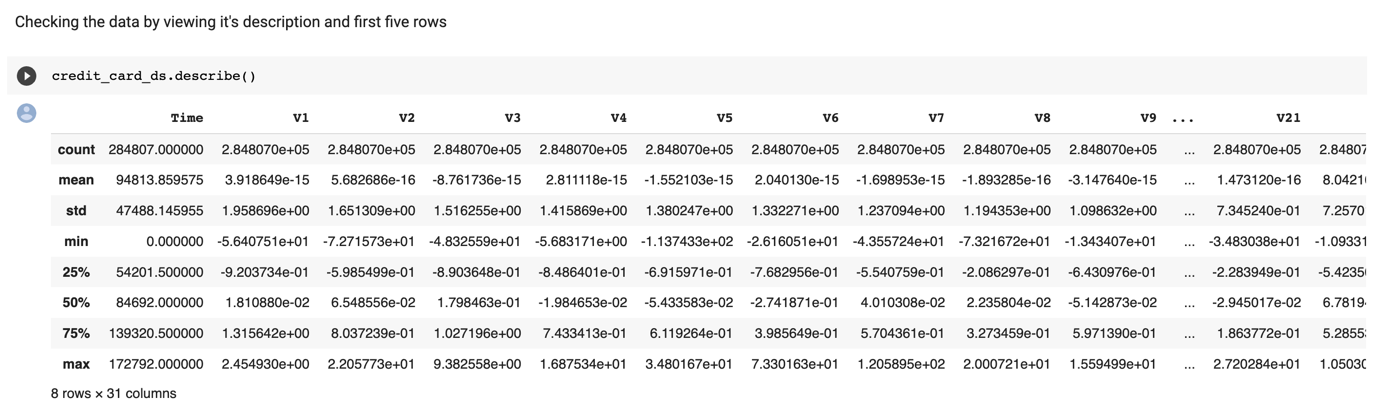


Figure : Description of Data

Graphical user interface

Description automatically generated with medium confidence

Figure : Initial Data

We plotted a bar plot of the target variable column to check the distribution of the number of fraudulent and authentic credit card transactions

Chart, bar chart

Description automatically generated

Figure : Class Distribution - Legit v/s Fraud Transactions

Next, we plotted a histogram to check the distribution of the feature ‘Amount’

Chart, histogram

Description automatically generated

Figure : Distribution of column 'Amount'

In the following step we plotted a histogram across one of the PCA Vectors, in this case V1 for checking the distribution

Chart, histogram

Description automatically generated

Figure : Distribution across V1

Next, we created a bar chart of correlations.

Chart, bar chart

Description automatically generated

Figure : Correlation with 'Class' column

Finally, we plotted the distribution of the column ‘Time’.

Chart, histogram

Description automatically generated

Figure : Distribution of column 'Time'

To understand the skewness of the data, we plotted a histogram.

Chart, histogram

Description automatically generated

Figure : Skewness Disribution

### Data Observations

* The data includes 8 Rows and 31 Columns.
* The Columns Names from V1 to V28 are hidden as it contains people's confidential transactional information. Hence, they are already converted to PCA Vectors.
* We have Time and Amount column, and the Class column is the target column and represents if the transaction is a fraud or not. '0' means it's not a fraud and '1' means it's a fraud. On further analysis, we can see that the number of frauds (492) in the data are much less than the legit transactions (284315). This indicates the data is highly unbalanced and might require under sampling or oversampling.
* The data does not have any null values in any of the columns.
* From the histograms, we can observe that the data has maximum transactional amounts below 100 with a mean of 88. The values of PCA vector V1 are maximum between -5 to 5.
* Further, we have also checked the skewness across each column by plotting its distribution across histogram and have fixed it using a Power Transformer

### Data Pre-processing

#### Overcoming the problem of an imbalanced dataset by using sampling techniques

The data we are using is rather uneven, and we have found that there is a large difference between the number of real and fraudulent transactions. To solve this problem, we choose to produce a more balanced dataset by oversampling the fraudulent values and under sampling the legitimate values.

We created a new dataset named "test\_over" by combining the original dataset of valid transactions with 3000 instances randomly chosen from the original dataset of fraud transactions using replacement. This strategy aids in improving the dataset's representation of the minority class (fraud).

Then, to choose examples from the majority class (legal transactions), which is closest to the minority class (fraud transactions), in terms of distance, we applied the NearMiss method with 10 nearest neighbours.

Then, using a bar chart with the x-axis representing the class labels (legit and fraud) and the y-axis representing the amount of instances, we printed the number of legitimate and fraudulent values in the sampled dataset. After the oversampling and under sampling processes, this bar chart gives a clear visual representation of the achieved balance between the two classes.

Chart, bar chart

Description automatically generated

Figure : Bar plot after sampling

#### Training and Testing Data

Now that we have completed the sampling, the data is now balanced, we will continue with splitting the data into Training and Testing. 70% of the total data will be used to train the ML model, with the remaining 30% used to test the model's accuracy.

#### Training and Testing Machine Learning Models:

We used three different machine learning algorithms. Trained and tested them all with the same dataset. These three algorithms are as follows:

##### **Random Forest Modeling:**

As part of an ensemble machine learning technique called random forest modeling, different decision trees are combined to produce a more reliable and accurate prediction model. The model is given unpredictability and variety through the use of feature subsampling and bootstrapping, which helps to lessen overfitting and boost performance. To generate the final result, the predictions from the several trees are combined using a majority vote or average. Random Forest models are widely utilized in many applications, including classification, regression, and feature selection, in industries including banking, healthcare, marketing, and image recognition, among others. They are well known for their propensity to handle big datasets, noisy data, and missing values.

##### **Linear Regression Modeling:**

Logistic regression is a widely used Machine Learning method that belongs to the Supervised Learning approach. It is used to predict the categorical dependent variable from a group of independent factors. As a result, the conclusion must be categorical or discrete. It can be No or Yes, 0 or 1, True or False, etc., but instead of giving the precise values like 0 and 1, it delivers the probability values that fall between 0 and 1, allowing it to classify whether the transaction is fraudulent or non-fraudulent.

##### **eXtreme Gradient Boosting (XGBoost):**

For supervised machine learning tasks including classification, regression, and ranking, the XGBoost (eXtreme Gradient Boosting) algorithm is an optimized implementation of the gradient boosting algorithm. It sequentially constructs an ensemble of decision trees, using regularization methods and gradient descent to update parameters. Due to its high prediction accuracy and speed, XGBoost is frequently utilized in industries including finance, healthcare, advertising, and recommendation systems. It is renowned for its effectiveness, scalability, and advanced functionality.

##### **Multi-Layer Perceptron**

An artificial neural network architecture called a multi-layer perceptron (MLP) is utilized for supervised learning tasks including classification and regression. It is made up of numerous layers of connected nodes, each of which applies an activation function to generate an output. MLP employs backpropagation to learn weights from training data and can have several hidden layers. Due to its capacity to recognize intricate patterns and flexible architectural design, it is adaptable and widely employed in a variety of applications.

## Result and Analysis

### Performance Evaluation

On the credit card dataset, we used standard evaluation approaches to assess how well the four algorithms (Random Forest Classifier, Linear Regression, XGBoost, and Multi-Layer Perceptron) performed. A confusion matrix was utilized to compare the actual target values to the values that each model predicted. Additionally, in order to evaluate the models' performance graphically, we produced Receiver Operating Characteristic (ROC) curves and calculated the Area Under the Curve (AUC) to depict each model's overall performance.

#### Random Forest Modeling:

We used a confusion matrix, a 2 x 2 matrix that contrasts the predicted values (tst\_prdc) of the model with the actual target values (s\_tst), to evaluate the effectiveness of the Random Forest Classifier model trained on our dataset. The confusion matrix gave us information about the model's performance in terms of true positives, false positives, true negatives, and false negatives. This information allowed us to assess how well the model identified fraudulent transactions while minimizing false alarms.

We used a Receiver Operating Characteristic (ROC) curve in addition to the confusion matrix to visually assess the performance of the model. The trade-off between the false positive rate (FPR) and true positive rate (TPR) at various categorization thresholds is depicted by the ROC curve. Additionally, we measured the Area Under the Curve (AUC), which offers a broad indicator of the model's effectiveness. We were able to assess the performance of the Random Forest Classifier model comprehensively and determine which model was most suited to the needs of the credit card firm by taking into account the data from the confusion matrix, classification report, and ROC curve.

|  |  |
| --- | --- |
| Chart  Description automatically generated  Figure 11: Test Confusion Matrix for Random Forest | Chart, line chart  Description automatically generated  Figure 12: ROC Curve for Random Forest |

#### Linear Regression Modeling:

We followed the same procedure as with the Random Forest Classifier in order to evaluate the model's performance. We used a confusion matrix to compare the actual target values to the values that the model had predicted. We were able to examine the true positives, false positives, true negatives, and false negatives using the confusion matrix, which gave us a better understanding of the model's capacity to identify fraudulent transactions and prevent erroneous warnings.

To further assess the model's effectiveness graphically, we created a Receiver Operating Characteristic (ROC) curve. At various categorization thresholds, the ROC curve demonstrated the trade-off between the false positive rate (FPR) and true positive rate (TPR). We also determined the Area Under the Curve (AUC), which demonstrated the model's general performance.

We were able to thoroughly assess the performance of the linear regression model and come to wise judgments for the demands of the credit card company by taking into account the confusion matrix and ROC curve.

|  |  |
| --- | --- |
| Chart, treemap chart  Description automatically generated  Figure 13: Test Confusion Matrix for Linear Regression | Chart, line chart  Description automatically generated  Figure 14: ROC Curve for Linear Regression |

#### XGBoost:

Using a confusion matrix and ROC curve, the effectiveness of the XGBoost model on the credit card fraud detection dataset was assessed. The model's ability to correctly forecast both fraudulent and non-fraudulent transactions was revealed by the confusion matrix. The area under the curve (AUC), which represents the overall effectiveness of the model, represented the trade-off between the false positive rate (FPR) and true positive rate (TPR) at various classification levels. The efficiency of the XGBoost model in identifying fraudulent transactions and preventing false positives can be measured by examining the ROC curve and AUC score.

|  |  |
| --- | --- |
| Chart  Description automatically generated  Figure 15: Test Confusion Matrix for XGBoost | Chart, line chart  Description automatically generated  Figure 16: ROC Curve for XGBoost |

#### Multi-Layer Perceptron:

A Multilayer Perceptron (MLP) Classifier model is fitted and assessed for credit card fraud detection in this snippet of code. On the training set, the model is trained using 50 hidden units and a random state of 2. The testing set's results are then predicted using the trained model, and the model's effectiveness is measured using a confusion matrix. With annotations for true positive, true negative, false positive, and false negative values, the confusion matrix is presented using a heatmap.

Next, predictions are produced on the testing set using a new training of the MLP Classifier model without specifying the number of hidden units. Plotting the ROC (Receiver Operating Characteristic) curve requires the calculation of the false positive rate (fpr), true positive rate (tpr), and threshold values. To evaluate the effectiveness of the model, the area under the ROC curve (AUC) is determined. The trade-off between true positive rate and false positive rate is depicted by the ROC curve. The plot legend includes an AUC score display.

|  |  |
| --- | --- |
| Chart, treemap chart  Description automatically generated  Figure 17: Test Confusion Matrix for MLP | Chart, line chart  Description automatically generated  Figure 18: ROC Curve for MLP |

## Conclusion

In order to address the problem of data imbalance in the credit card fraud detection dataset, this research used three different models: Random Forest Classification, Linear Regression, and XGBoost. The real values were under sampled to 3000 while the fraudulent values were oversampled to 3000 in order to balance the imbalance. Then, a 70:30 split between the training and test sets was applied to the dataset.

The models were trained on the training set, and the accuracy of their predictions on the testing set was used to gauge their performance. When compared to the Linear Regression model, which performed well in terms of accuracy, precision, and how it appeared on the Confusion Matrix and ROC Curves, the Random Forest Classifier and XGBoost models showed overfitting.

Based on these results, it can be said that the Linear Regression model performed better than the other two models and is the best choice for this project's evaluation of credit card fraud transactions.

## Future Scope:

Since two of the models overfitted the data, we can explore more models to detect fraudulent credit card transactions.

## Dataset Link:

<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

## References:

[1] Muaz, A., Jayabalan, M., & Thiruchelvam, V. (2020). A Comparison of Data Sampling Techniques for Credit Card Fraud Detection. International Journal of Advanced Computer Science and Applications, 11(6). <https://doi.org/10.14569/ijacsa.2020.0110660>​

[2] Jake VanderPlas, Python Data Science Handbook.​

[3] Shen, A., Tong, R., & Deng, Y. (2007). Application of Classification Models on Credit Card Fraud Detection. 2007 International Conference on Service Systems and Service Management.