Credit Card Fraud Detection

1. Introduction: Provide an overview of the problem of credit card fraud and the importance of developing accurate fraud detection models.

   According to globenewswire, in 2021, credit card fraud losses exceeded $32 billion and losses are expected to reach $397.4 billion over the next 10 years. Therefore, it is important for credit card companies to recognize fraudulent credit card transactions and build trust with their customers.

   The main objective of this article is to develop a machine learning model that will identify fraudulent transactions. This task should be considered to be in the production stage and not complete to be directly be used in real situation as some aspects will be skipped due to time, computer power, etc. But, we'll use different machine algorithms to train a classification model in order to identify fraudulent transactions.

2. Data collection and preprocessing: Describe the process of collecting and preparing the data for analysis. This may include information about the sources of data, the size of the dataset, and any cleaning or normalization that was done.

   For this post, we will use a transaction dataset to build a model that predicts fraud based on certain characteristics. The data used in this paper is provided by kaggle and can be downloaded here after creating an account.

   Here is a quick description of the dataset contents:

   - Transactions made by credit cards European cardholders.
   - Data collected over 2 days in September 2013
   - A total of 284,807 transactions
   - 492 frauds recorded (0.172%)
   - It contains 31 columns and 28 columns resulting from a PCA transformation
   - Free of missing data

   More details can be found on the kaggle

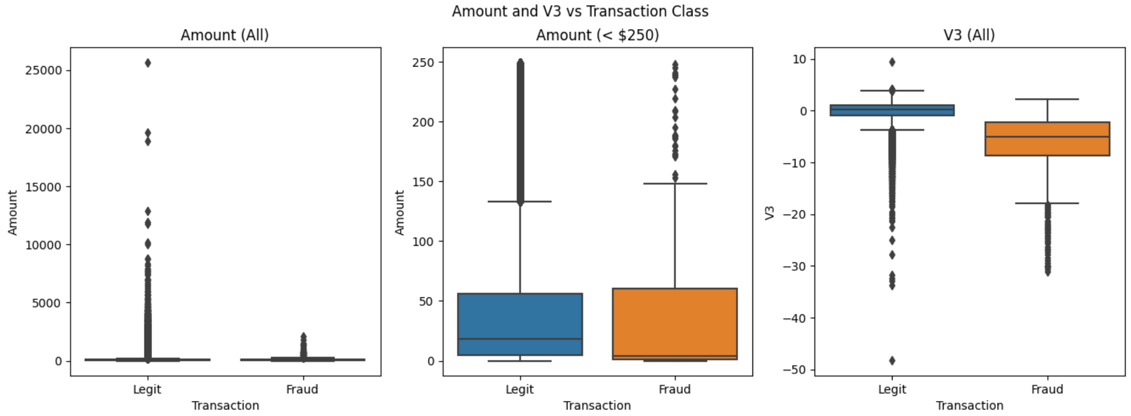
3. Exploratory Data Analysis (EDA): Discuss the steps taken to understand the data, such as visualizations or statistical analyses. Explain any patterns or insights that were discovered during EDA.

   For a dataset with a large number of features, it is necessary to select and retain those with high potential in terms of predictive power or use dimension reduction techniques to reduce their number. But for credit card fraud detection, of the 30 features, 28 are already PCA results. The absence of the original features - for privacy reasons - used in their creation makes it difficult to identify what they correspond to.

4. Feature Engineering: Explain the process of selecting and transforming the variables used in the model. This may include dimensionality reduction, feature scaling, or creating new variables.

   We can still perform feature engineering to identify the features that distinguish fraudulent transactions from the legit ones, and boxplots can be useful for this purpose.

   For that let's look at two the boxplots of 2 different features bellow: Amount and V3

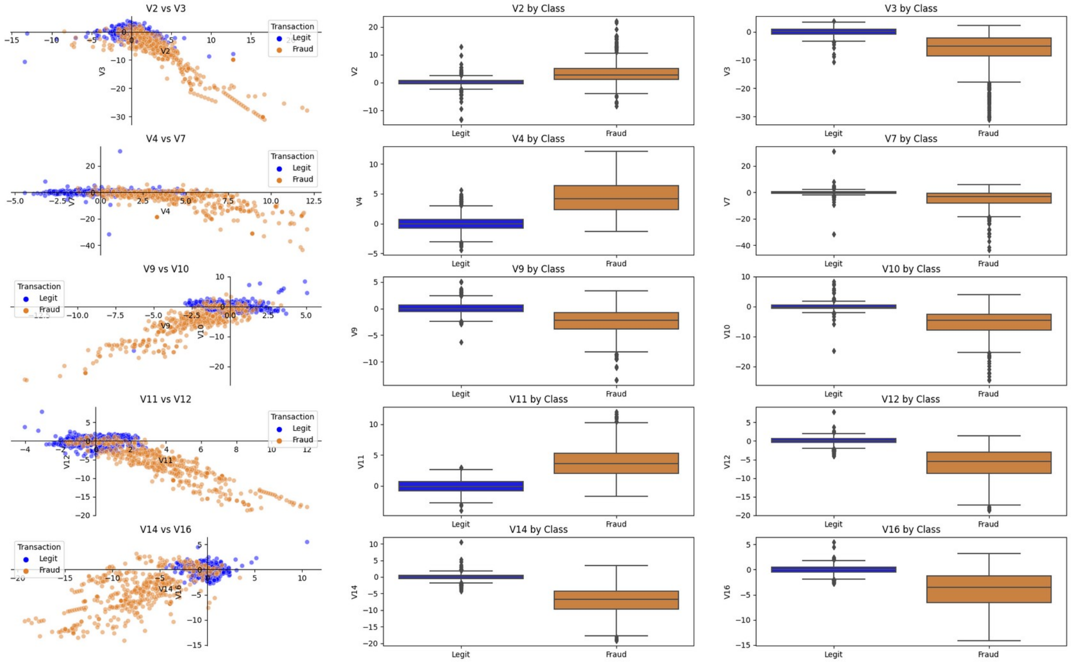


Even if the left boxplots – Amount (All) – gives quite few details as the boxes seems to be just lines, it's still possible possible to notice that the credit card fraudsters tend to keep transactions bellow a certain amount (less than $2500), but, compared to all the other transactions these Amount are considered outliers.

Let's "zoom in" for more details. The middle boxplots – Amount (< 250) – is the version of the previous boxplots that focuses on the amount range with most of the transactions (amount less than $250). In this range, identifying potential legit and fraudulent transaction is less realistic as their respective boxplots are almost identical.

On the other hand, the feature V3 – from a PCA – seems to have higher potential ability in differentiating legit transactions from the fraudulent ones. Indeed, the component V3 tends to make that difference by assigning high score to legit transactions and lower scores to the fraudulent ones.

In total, there are 30 features and they're all numerical. We can write a script to identify those with a high potential differentiation power based on a custom criteria. For instance, we can decide that a feature has that high potential if for its two boxplots B1 (legit) and B2 (fraud), the 1st quartile of B1 is greater than the 3rd quartile of B2, or the opposite (e.i the 1st quartile of B2 is greater than the 3rd quartile of B1). We can achieve this by write a simple script in Python (see code here). Next, plotting these features gives the figure bellow:



We can notice that the variables selected by our script have a certain potential in discriminating fraudulent transactions from the legit ones. For the other variables we can try different transformations such as logarithm, square, trigonometric etc and rerun the previous script to check for new variables with high discrimination potential.

But we will just will try only the square transformation. This shows that by squaring all the remaining variables the index V_21 and V_27 (their squared values) was spotted by the script (see the below graph)

5. Model Selection: Describe the process of selecting a suitable algorithm to solve the problem, such as logistic regression, decision tree, random forest, or neural network. Discuss the advantages and disadvantages of each algorithm and why you chose a specific one.

   There are several algorithms than can be used in our situation to train a classification model. Among there there.

   - Logistic Regression:

   - Support Vector Machines:

- Tree-based algorithms such as Decision Tree and Random Forest: These algorithms are known for their ability to handle imbalanced data better than others such as logistic regression or support vector machines.

- Deep Learning:

Logistic regression, support vector machines, and decision trees are sensitive to outliers in the training data. Outliers can have a significant impact on the decision boundary, and hence on the classification results, especially for algorithms that rely on the distance between data points, such as SVM. In logistic regression and decision trees, outliers can affect the model coefficients and the splitting criteria, respectively, leading to biased or unstable models. Therefore, it is important to handle outliers appropriately before or during training to improve the performance and stability of these algorithms.

Robust methods: Some machine learning algorithms, such as random forests and support vector machines, are naturally robust to outliers. Using these algorithms can be an effective way to handle outliers in the data.

Ensemble methods: Ensemble methods like bagging and boosting can also help handle outliers by averaging or combining the predictions of multiple models. This can help reduce the impact of outliers on the overall performance of the model.

We'll restrict our choice to the first four

6. Model training and evaluation: Discuss how the model was trained on the data and how its performance was evaluated. Explain the metrics used to assess the model's performance, such as accuracy, precision, recall, and F1 score. Provide the results of the evaluation and discuss any issues or challenges encountered during this stage.

   The quality of datasets and their distribution play an important role when its comes to training a model. This even more important when data dataset is highly imbalanced as the one we're dealing with. Even if there are training algorithms that can handle this such datasets there are still some common methods to address imbalanced data.

- Resampling the data: This involves either undersampling the majority class or oversampling the minority class.
- Adjusting the classification threshold: By default, most classifiers have a threshold of 0.5 for classifying samples as positive or negative. Adjusting this threshold can help improve performance on the minority class
- Using different performance metrics: Accuracy is not always the best metric to evaluate the performance of a classification model on imbalanced data. Indeed the proportion of fraudulent transaction in our dataset is 0.172% which is less than 1%. A simple way to have a model with at least 99% of accuracy (which is pretty good for a balanced dataset), we can just decide flag all transaction as legit in our predictions. This will take us away from our main objective which to help credits card companies

to offer a more secure service to their clients by spotting fraudulent transactions. lMetrics such as precision, recall, F1 score, and AUC-ROC can be more informative.

The fact that the dataset is highly unbalanced makes the use of accuracy measure irrelevant. Therefore we will focus on measures such as Area under cover (AUC), precision, recall, and f1-score.

The table presents the performance results of five different machine learning models trained to identify fraudulent transactions of credit cards. The evaluation metrics used to assess their performance include Accuracy, AUC, Precision, Recall, and f1 score for both legitimate and fraudulent transactions.

All the models achieved high accuracy levels ranging from 0.9345 to 0.9503. The Random Forest model achieved the highest accuracy of 0.9503, while the Decision Tree model achieved the lowest accuracy of 0.9323.

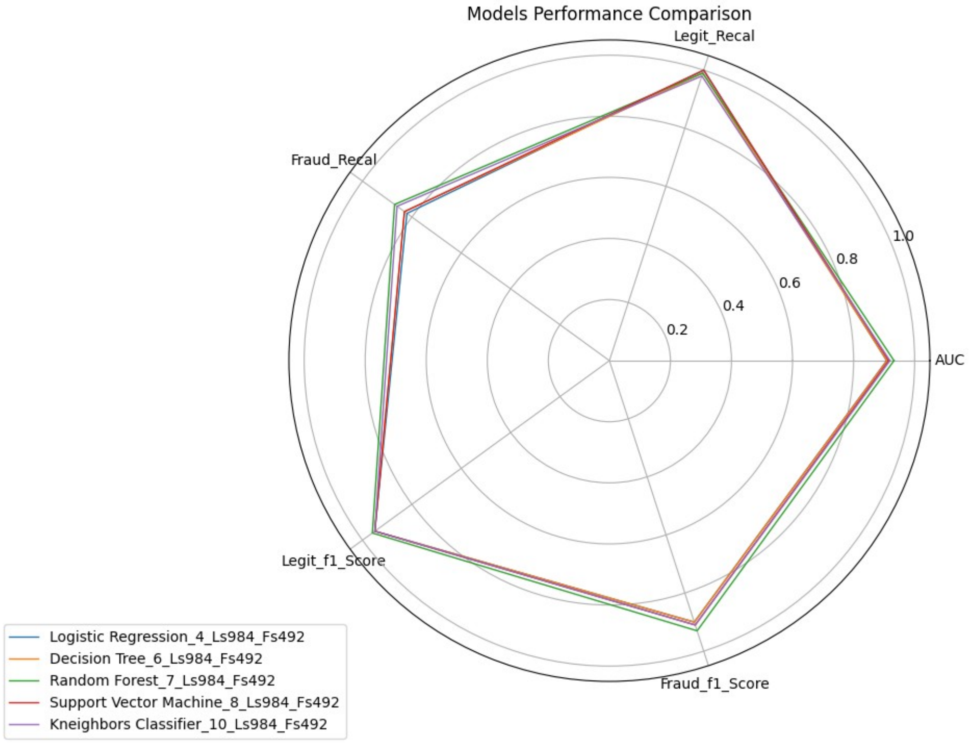
The AUC (Area Under Curve) values range from 0.9085 to 0.9324, and the Random Forest model has the highest AUC of 0.9324, indicating that this model can effectively distinguish between fraudulent and legitimate transactions.

In terms of precision, all models achieved a high precision for fraudulent transactions, with values ranging from 0.96 to 0.99, indicating that these models have a low false-positive rate. The Random Forest model achieved the highest precision score of 0.99.

The Recall metric measures the ability of the model to identify fraudulent transactions, and all models achieved high recall scores for both fraudulent and legitimate transactions, indicating that the models can detect a high percentage of fraudulent transactions. The Random Forest model achieved the highest recall score of 0.87 for fraudulent transactions.

The f1 score is the harmonic mean of precision and recall, and it provides a measure of the model's overall performance. All models achieved high f1 scores ranging from 0.90 to 0.93, with the Random Forest model achieving the highest f1 score of 0.93.

Overall, the Random Forest model appears to be the best-performing model based on the high values of AUC, precision, recall, and f1 score. However, all models performed well and achieved high accuracy levels, indicating that they can effectively identify fraudulent transactions.



Performance table:

Model Name	**Acc**	**AUC**	**Legit Precision**	**Fraud Precision**	**Legit Recal**	**Fraud Recal**	**Legit f1 Score**	**Fraud f1 Score**
Logistic Regression	0.9345	0.9085	0.91	0.99	1	0.82	0.95	0.9
Decision Tree	0.9323	0.9097	0.92	0.97	0.99	0.83	0.95	0.9
Random Forest	0.9503	0.9324	0.93	0.99	0.99	0.87	0.96	0.93
Support Vector Machine	0.9391	0.9149	0.92	0.99	1	0.83	0.95	0.91
Kneighbors Classifier	0.9368	0.9190	0.93	0.96	0.98	0.86	0.95	0.91

7. Conclusion and Future Work: Summarize the key findings of the analysis and discuss potential future work to improve the model's accuracy and performance. This may include ideas for incorporating additional data sources, exploring different machine learning techniques, or refining the model architecture.
   We discovered that

8. References: Include a list of sources used in the analysis, such as research papers, articles, or tutorials.
9. Code: Include a link to the code used to build the model, so that readers can replicate your work and explore the results on their own.