Project:Credit Card Fraud Detection

Abstract:
It is vital that credit card companies are able to identify fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Such problems can be tackled with Data Science and its importance, along with Machine Learning, cannot be overstated. This project intends to illustrate the modelling of a data set using machine learning with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the data of the ones that turned out to be fraud. This model is then used to recognize whether a new transaction is fraudulent or not. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications.
Credit Card Fraud Detection is a typical sample of classification. In this process, we have focused on analysing and preprocessing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data.

Purpose:
High dependence on the internet has increased credit card transactions. As the credit card transactions become the most common method of payment for both online and offline transactions, credit card fraud rate also increases. In the past, credit card frauds have been increasing tremendously. According to the Nilson Report [3], the global credit card fraud losses reached $16.31 billion in 2014 and it is estimated to cross $35 billion in 2020. A lot of research has been developed to detect external card fraud which accounts for the majority of credit card frauds. Detecting fraudulent transactions using traditional methods of manual detection is time consuming and inefficient. Hence, it is necessary to develop a credit card fraud detection technique as a countermeasure to fight illegal activities. Methodology or Approach: Fraud detection is a complex issue that requires a substantial amount of planning before applying Machine Learning algorithms to it. Here, we are going to use a local outlier factor to calculate anomaly scores as well as an isolation forest algorithm. These two algorithms will help us to go through a set of almost 280,000 credit card transactions and will predict which ones are fraudulent. The work is implemented in Python. The class of 1 means that the transaction is fraudulent where as in our data set 0 would mean it’s a valid transaction. This network is going to predict -1 for outliers. It correctly predicts whether it is an outlier or a fraudulent transaction.

Main challenges involved in credit card fraud detection are:

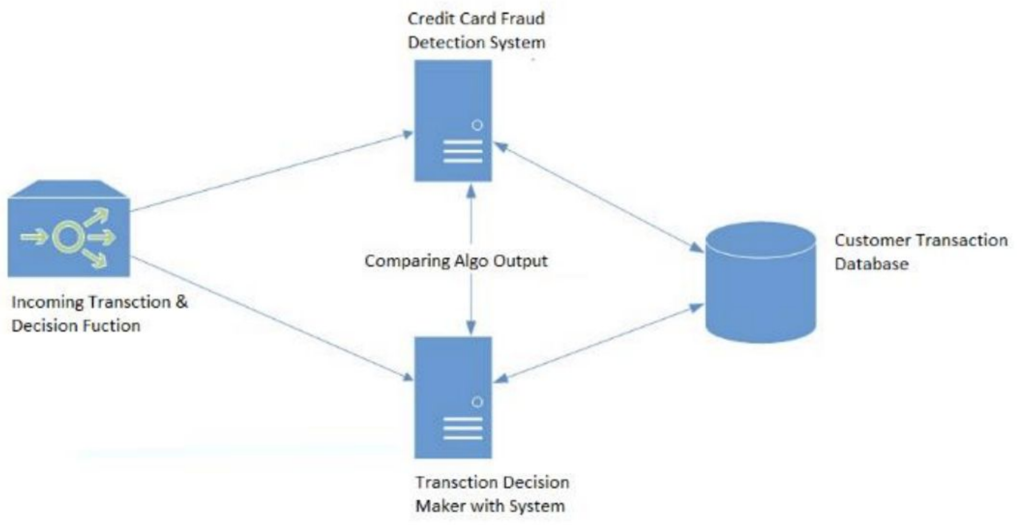
1. Enormous Data is processed every day and the model build must be fast enough to respond to the scam in time.
2. Imbalanced Data i.e most of the transactions *(99.8%)* are not fraudulent which makes it really hard for detecting the fraudulent ones
3. Data availability as the data is mostly private.
4. Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported.
5. Adaptive techniques used against the model by the scammers.

How to tackle these challenges?

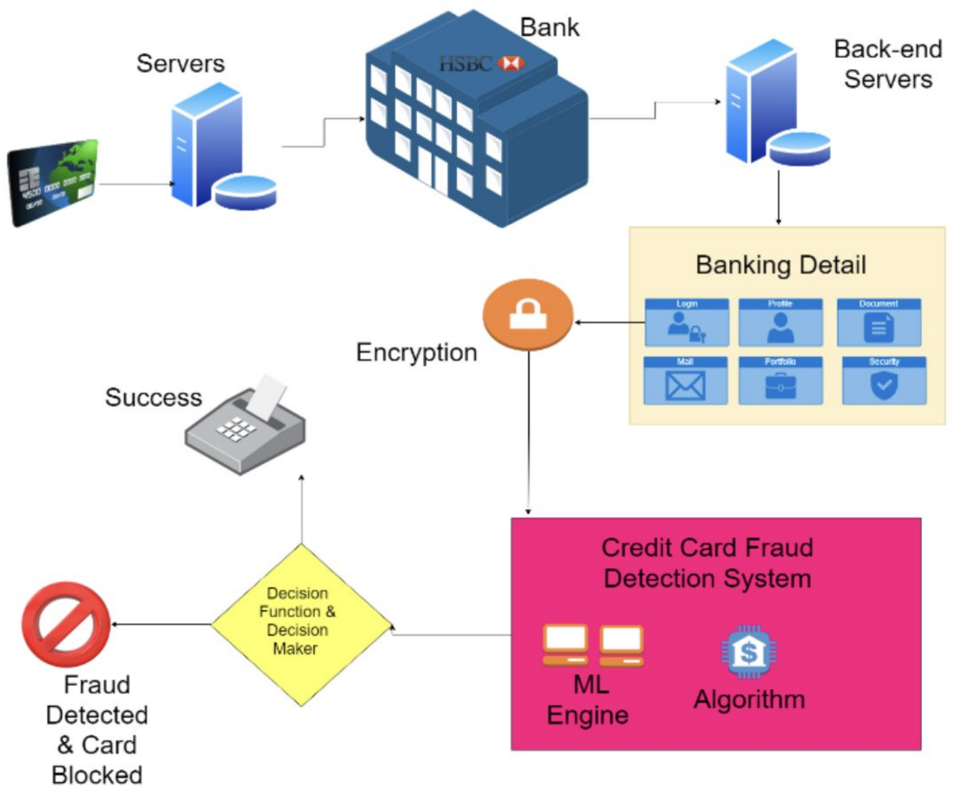
1. The model used must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible.
2. Imbalance can be dealt with by properly using some methods which we will talk about in the next paragraph
3. For protecting the privacy of the user the dimensionality of the data can be reduced.
4. A more trustworthy source must be taken which double-check the data, at least for training the model.
5. We can make the model simple and interpretable so that when the scammer adapts to it with just some tweaks we can have a new model up and running to deploy.

METHODOLOGY

The approach that this paper proposes, uses the latest machine learning algorithms to detect anomalous activities, called outliers. The basic rough architecture diagram can be represented with the following figure:


Credit Card Fraud Detection System
Incoming Transction & Decision Fuction
Comparing Algo Output
Customer Transaction Database
Transction Decision Maker with System

When looked at in detail on a larger scale along with real life elements, the full architecture diagram can be represented as follows:



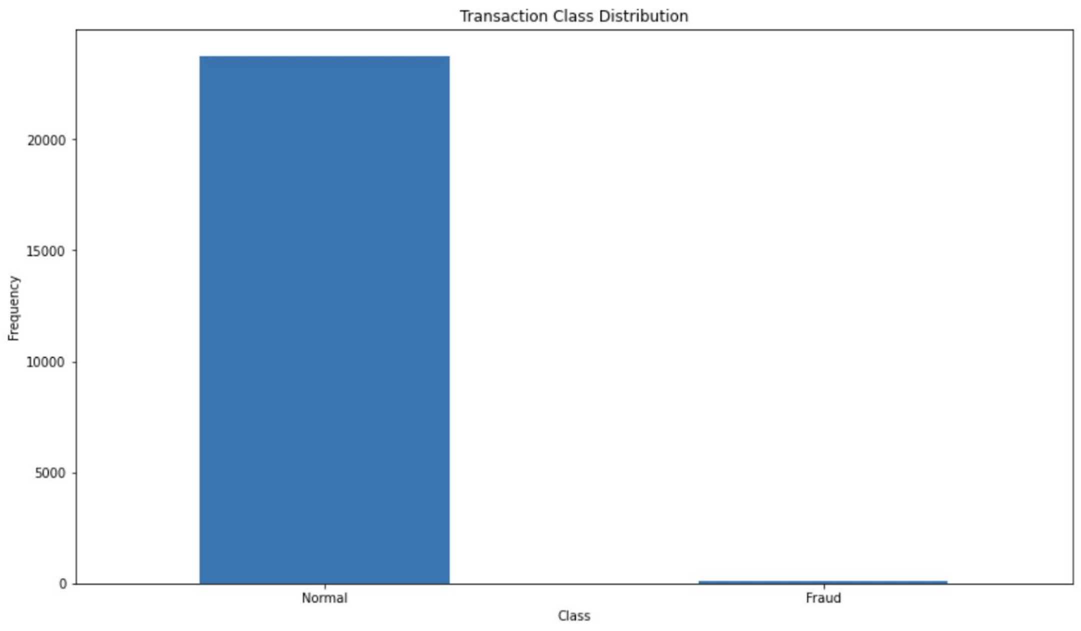
About the Dataset:

For this project we are using a dataset that is hosted on Kaggle.com.
https://www.kaggle.com/mlg-ulb/creditcardfraud

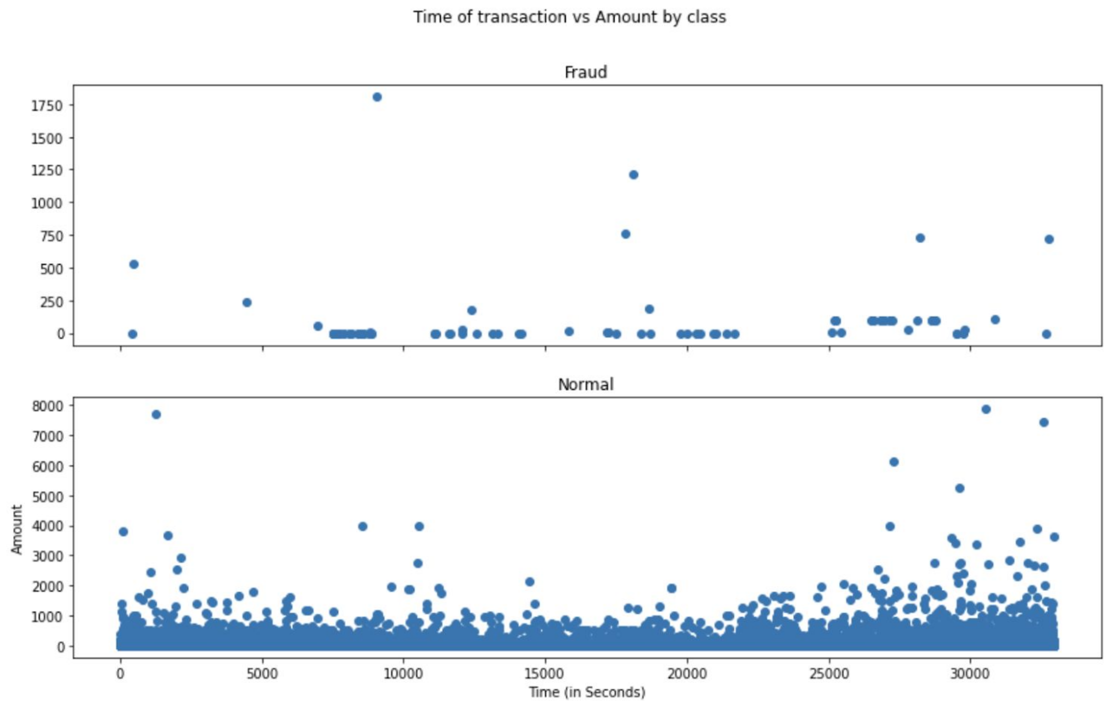
The datasets contain transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numeric input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise

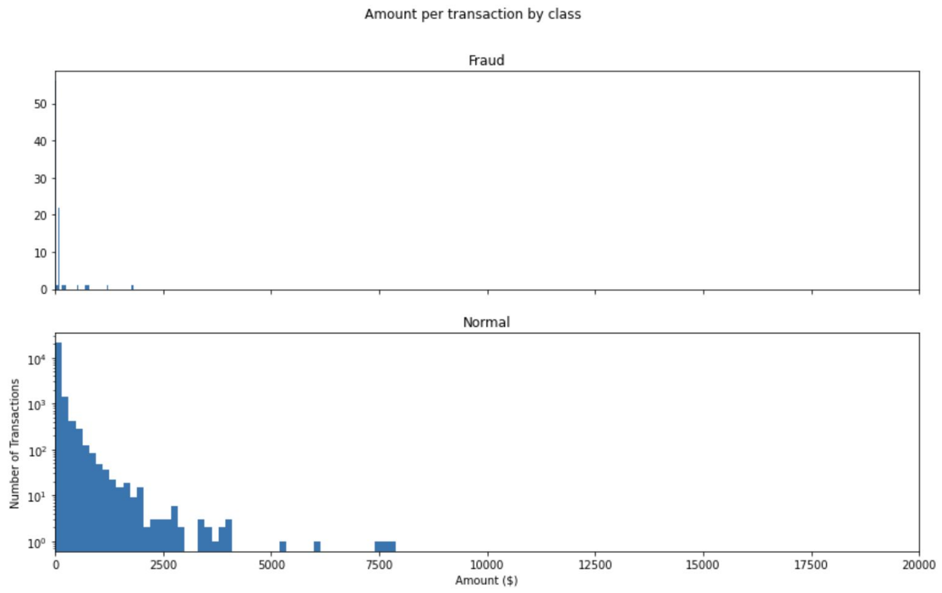
We plot different graphs to check for inconsistencies in the dataset and to visually comprehend it:



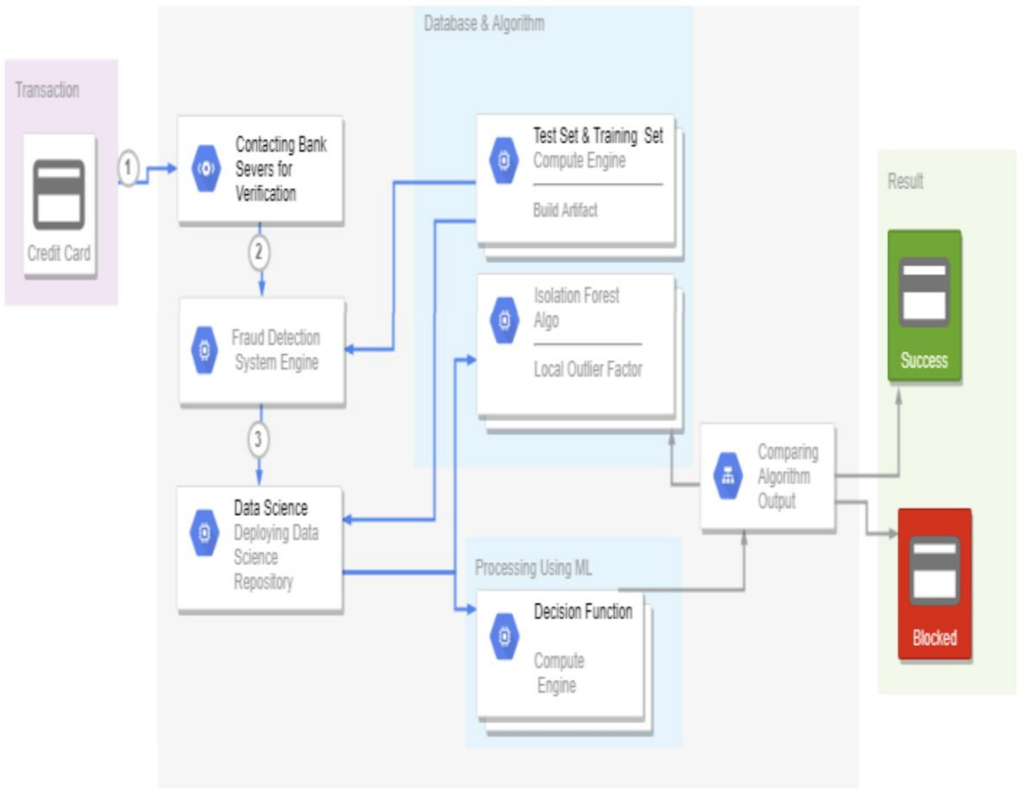
This graph shows that the number of fraudulent transactions is much lower than the legitimate ones.



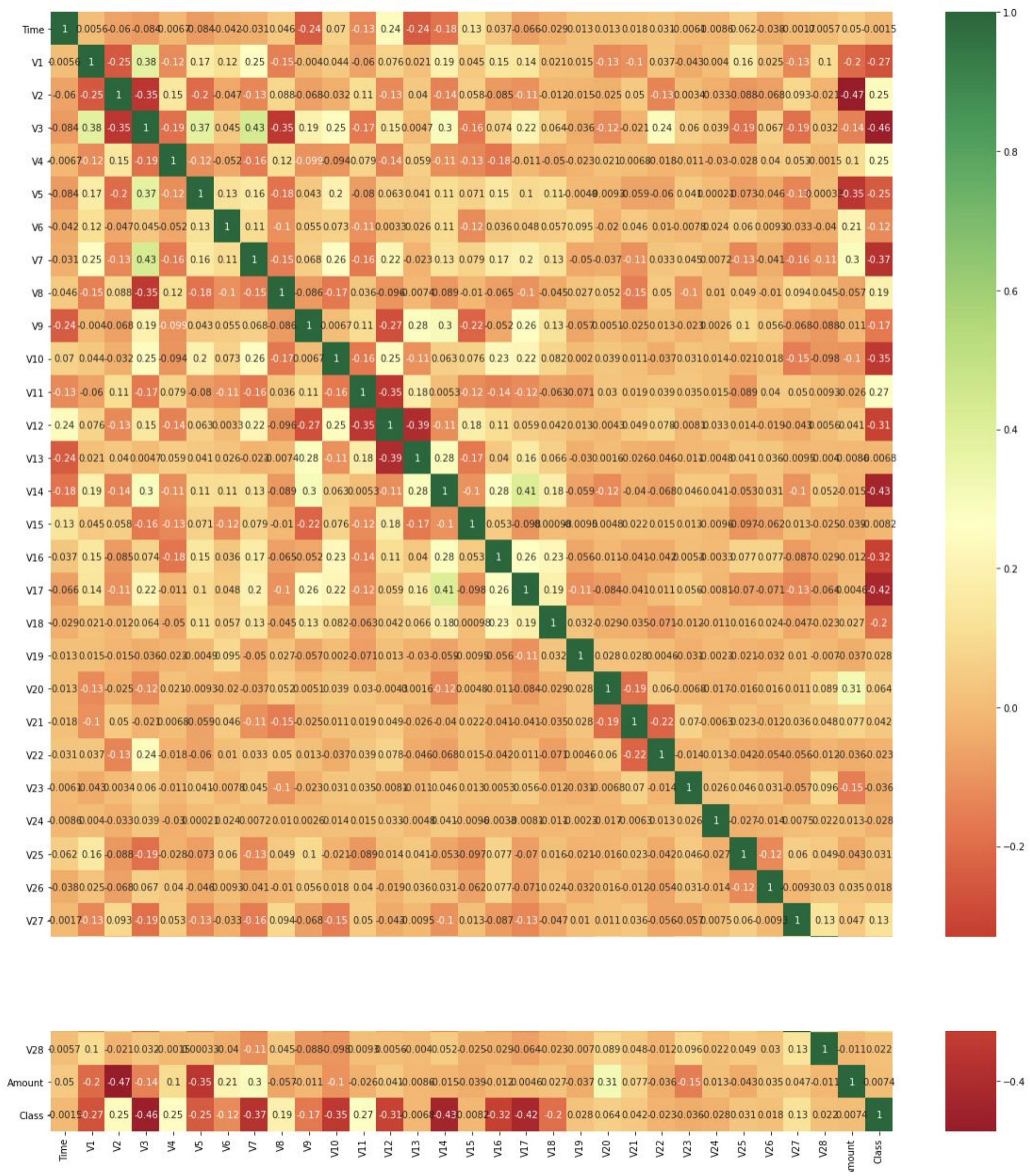
This graph shows the times at which transactions were done within two days. It can be seen that the least number of transactions were made during night time and highest during the days.



This graph represents the amount that was transacted. A majority of transactions are relatively small and only a handful of them come close to the maximum transacted amount. After checking this dataset, we plot a histogram for every column. This is done to get a graphical representation of the dataset which can be used to verify that there are no missing values in the dataset. This is done to ensure that we don't require any missing value imputation and the machine learning algorithms can process the dataset smoothly.



After this analysis, we plot a heatmap to get a coloured representation of the data and to study the correlation between our predicting variables and the class variable. This heatmap is shown below



Model Prediction

Now it is time to start building the model .The types of algorithms we are going to use to try to do anomaly detection on this dataset are as follows

Isolation Forest Algorithm :

One of the newest techniques to detect anomalies is called Isolation Forests. The algorithm is based on the fact that anomalies are data points that are few and different. As a result of these properties, anomalies are susceptible to a mechanism called isolation.

This method is highly useful and is fundamentally different from all existing methods. It introduces the use of isolation as a more effective and efficient means to detect anomalies than the commonly used basic distance and density measures. Moreover, this method is an algorithm with a low linear time complexity and a small memory requirement. It builds a good performing model with a small number of trees using small sub-samples of fixed size, regardless of the size of a data set.

Typical machine learning methods tend to work better when the patterns they try to learn are balanced, meaning the same amount of good and bad behaviors are present in the dataset.

How Isolation Forests Work The Isolation Forest algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. The logic argument goes: isolating anomaly observations is easier because only a few conditions are needed to separate those cases from the normal observations. On the other hand, isolating normal observations require more conditions. Therefore, an anomaly score can be calculated as the number of conditions required to separate a given observation.

The way that the algorithm constructs the separation is by first creating isolation trees, or random decision trees. Then, the score is calculated as the path length to isolate the observation.

Local Outlier Factor(LOF) Algorithm

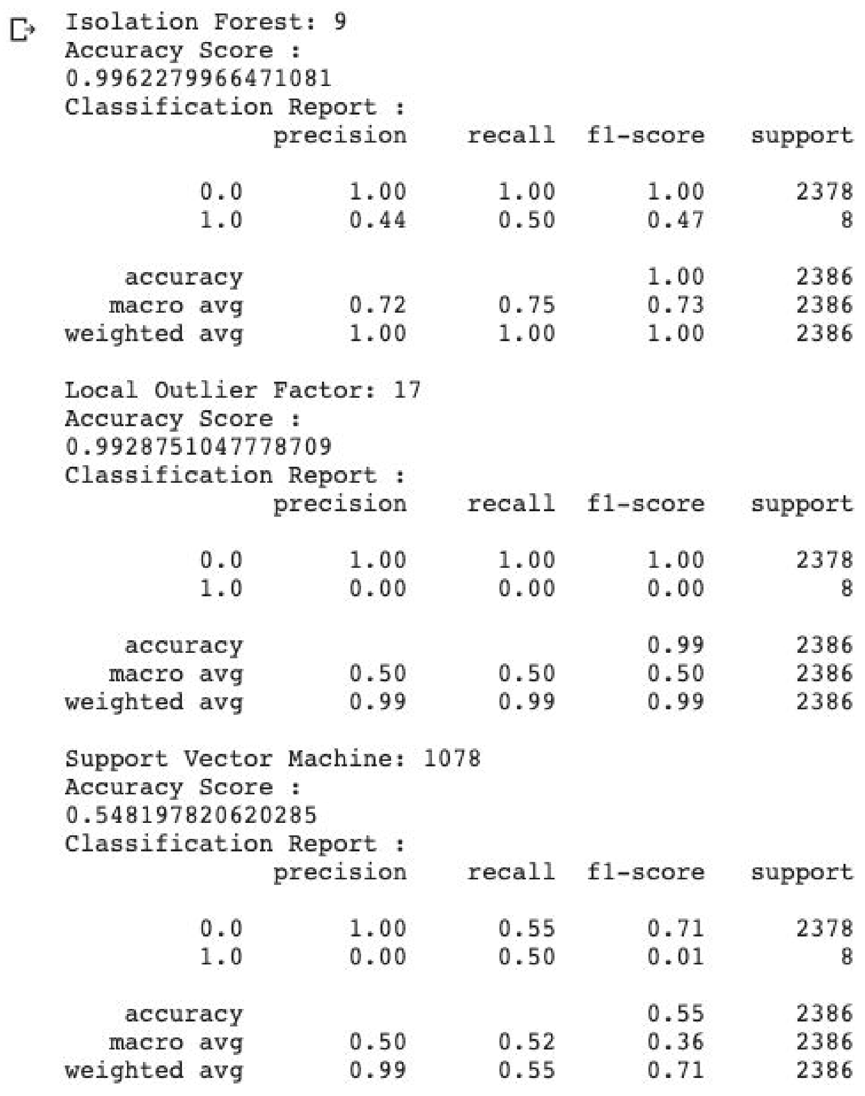
The LOF algorithm is an unsupervised outlier detection method which computes the local density deviation of a given data point with respect to its neighbors. It is considered as outlier samples that have a substantially lower density than their neighbors.

The number of neighbors considered, (parameter n_neighbors) is typically chosen 1) greater than the minimum number of objects a cluster has to contain, so that other objects can be local outliers relative to this cluster, and 2) smaller than the maximum number of close by objects that can potentially be local outliers. In practice, such information is generally not available, and taking n_neighbors=20 appears to work well in general.

Observation and Result

The code prints out the number of false positives it detected and compares it with the actual values. This is used to calculate the accuracy score and precision of the algorithms.The fraction of data we used for faster testing is 10% of the entire dataset. The complete dataset is also used at the end andboth the results are printed.These results along with the classification report for each algorithm is given in the output as follows, where class 0means the transaction was determined to be valid and 1 means it was determined as a fraud transaction.This result matched against the class values to check for false positives.

Results When 10% Data is feeded.



```
Isolation Forest: 9
Accuracy Score :
0.9962279966471081
Classification Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      2378
         1.0       0.44      0.50      0.47         8

    accuracy                           1.00      2386
   macro avg       0.72      0.75      0.73      2386
weighted avg       1.00      1.00      1.00      2386

Local Outlier Factor: 17
Accuracy Score :
0.9928751047778709
Classification Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      2378
         1.0       0.00      0.00      0.00         8

    accuracy                           0.99      2386
   macro avg       0.50      0.50      0.50      2386
weighted avg       0.99      0.99      0.99      2386

Support Vector Machine: 1078
Accuracy Score :
0.548197820620285
Classification Report :
              precision    recall  f1-score   support

         0.0       1.00      0.55      0.71      2378
         1.0       0.00      0.50      0.01         8

    accuracy                           0.55      2386
   macro avg       0.50      0.52      0.36      2386
weighted avg       0.99      0.55      0.71      2386
```

- Isolation Forest detected 9 errors versus Local Outlier Factor detecting 17 errors vs. SVM detecting 1078 errors
- Isolation Forest has a 99.62% more accurate than LOF of 99.28% and SVM of 54.81
- When comparing error precision & recall for 3 models , the Isolation Forest performed much better than the LOF as we can see that the detection of fraud cases is around 27 % versus LOF detection rate of just 2 % and SVM of 0%.
- So overall the Isolation Forest Method performed much better in determining the fraud cases which is around 30%.
- We can also improve on this accuracy by increasing the sample size or use deep learning algorithms however at the cost of computational expense.We can also use complex anomaly detection models to get better accuracy in determining more fraudulent cases

<u>Conclusion</u>

Credit card fraud is without a doubt an act of criminal dishonesty. This project has listed out the most common methods of fraud along with their detection methods and reviewed recent findings in this field.Since the entire dataset consists of only two days' transaction records, its only a fraction of data that can be made available if this project were to be used on a commercial scale. Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.

<u>Future Enhancements</u>

The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.

Also We can create a used based apps for people to track their payments and to help them to identify whether the given site is safe or not as we can recognise these sites by compiling the data of whether any other customer reported it fraud or not .

References:

1)geeksforgeeks

2)tutorialspoint

3)youtube

4)Google

<u>Submitted By</u>

<u>Team Hunters</u>

-)Umang Singhal

-)Aditi Patel

-)Utkarsh Tiwari

-)Shubham Mittal