**Introduction**

The project is about accurately detecting fraudulent credit/debit card transaction using machine learning techniques. Nilson reports that U.S. card fraud (credit, debt, etc) was reportedly $9 billion in 2016 and expected to increase to $12 billion by 2020. For perspective, in 2017 both PayPal's and Mastercard's revenue was only $10.8 billion each. In addition, The scam usually occurs when someone accesses consumers credit or debit card numbers from unsecured websites or via an identity theft scheme to fraudulently obtain money or property. Due to its recurrence and the harm it may cause to both individuals and financial institutions, it is crucial to take preventive measures as well as identifying when a transaction is fraudulent. The project is undertaken because I am inspired by the experiences that I had at working best buy and I would like to contribute detecting fraudulent credit/debit card transaction so that the consumers do not suffer. I have seen many fraudulent credit/debits card transaction and the purpose of this research would be correctly classified and predict credit/debit card fraudulent transaction using machine learning techniques.

**Selection of Data**

The dataset utilized in this project was provided by [Kaggle](https://www.kaggle.com/mlg-ulb/creditcardfraud) and contains transactions made by credit cards in September 2013 by European cardholders. The transactions reported occurred within two days.

Data preview:

Table

Description automatically generated

The dataset is highly unbalanced since it has 492 (0.17%) frauds out of 284,807 transactions. All the features in the data set are numeric. Thanks to the confidentiality of the client, the columns have been renamed to V1, V2, ..., V28, and their features have undergone a PCA transformation, which consists of zeroing one or more of the smallest key components, resulting in a lower-dimensional data projection that retains the full data variance. The only two exceptions were the features Time and Number, which included seconds elapsed between each transaction and the first transaction in the dataset, and the transaction amount, respectively.

**Methods**

Tools:

* NumPy, Pandas, Matplotlib, seaborn and Scikit-learn for data analysis and inference
* GitHub for version control
* Jupyter as notebook

Inference methods used with Scikit:

* Linear regression model
* Decision Tree
* Test and train
* Confusion Matrix

**Result**

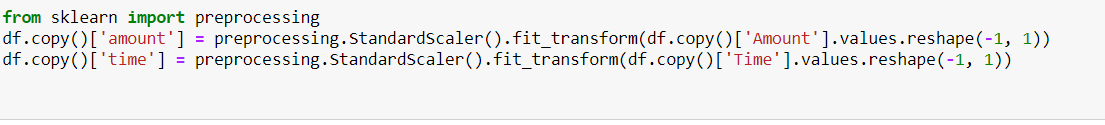
**A picture containing chart

Description automatically generated**

We start with analyzing the correlation between the variables. As we can see we did not observe any strong correlation between the variables. That is because our data is highly unbalanced. After preprocessing the data, we should see a better-looking correlation matrix.

**Preprocessing:**

In order to make sure the data is easily interpreted by the machine language I used StandarScaler library to have a mean value 0 and standard deviation of 1



**Training and test set:**

The reason for using training and testing data set is because I can minimize the effects of data discrepancies and better understand the characteristics of the model. Once I process the model by using training set, I can test the model by making predictions against the test set.

Graphical user interface, text, application

Description automatically generated

Here I did split my dataset into train and test data. The size of the test data is 0.25, the default value.

**Balancing the dataset:**

As the dataset indicates and also I mentioned it earlier that this dataset is highly unbalanced and therefore our training dataset could be biased and machine learning algorithm could display unsatisfactory results. To make sure I do not get biased results I used RandomUnderSampler() to randomly balance the data

A picture containing diagram

Description automatically generated

Chart, bar chart

Description automatically generated

As we see on the balanced graph that both class 0 and 1 has 351 entries where unbalanced dataset has 284315 entries for class 0 and 492 for class 1. Since dataset is balanced now, I would not have biased and unsatisfactory results when I use machine learning algorithm.

**Logistic Regression:**

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. In our case, it will establish the probability of a transaction belonging to Class 0 or 1, which is regular or fraudulent.



In terms of quality of prediction, the model has an accuracy of 98% and ROC AUC score of 94%. This model shows that my logistic regression model performed really well. From the confusion matrix I concluded that 98% of regular transaction were correctly predicted as regular and 90% of fraudulent transaction were correctly predicted as frauds.

**Decision Tree:**

**A picture containing graphical user interface

Description automatically generated**

In terms of quality of prediction, the Decision tree model has an accuracy of 89% and ROC AUC score of 93%. This model shows that my Decision tree model performed good but not as good as logistic regression. From the confusion matrix I concluded that 89% of regular transaction were correctly predicted as regular and 98% of fraudulent transaction were correctly predicted as frauds.

**Discussion**

Our goal was to correctly classified fraudulent and regular transaction using machine learning techniques. Both models Logistic Regression and Decision Tree performed well in classifying regular and fraudulent transaction. In term of accuracy and precision Logistic regression performed better compare to decision tree where logistic regression got 98% and 94% respectively for regular and fraudulent activities. Even though Decision tree did not perform as well as Logistic regression still this model’s performance was satisfactory.

One of the reason for satisfactory result was preprocessing and balancing the data. Without performing this task the data would not be balanced and I would not have accurate prediction.

The biggest drawback from this analysis is the occurrence of false positive. The accuracy was between 90-97% and in some cases the algorithm incorrectly detects a fraud. So my future research would be shrink the wrong detection of fraud and my aim is 3-4% mark of false positive.

**Important Finding**

The most important finding is data after the preprocessing and balancing because with these two tasks I would have entirely different model which would be way less accurate. In addition the accuracy and precise score for detecting fraudulent and regular transaction using logistic regression and decision tree was most important finding because it was my research question and after my analysis I got satisfactory results.