# IBM Coursera Capstone Project

# Credit Card Fraud

By:Tarushi Pathak

## Introduction

Many of us fall prey to the fraud messages and calls sent to us and end up losing a huge amount of our savings. Sometimes the fraud messages are too obvious and other times they are extremely well planned and hard to figure out. Either way, the person ends up losing a huge amount of money.
It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Hence, in this project I'll be focusing on using the means of clustering the credit card purchase data in order to understand the fraud character traits which will further help the credit companies to recognize fraudulent credit card transactions faster.

### Business Problem

The objective of this capstone project is to analyse the data and classify them on the basis whether a fraud would be committed or not. Using Data Science Methodology, we will aim to find an answer to the question , Is the given data and trait of the corresponding transaction pointing to fraud or not? , using data analysis and machine learning techniques.

## Target Audience

The credit card companies using this machine learning model will be able to assure protection from fraud to their customers to a certain extent. Customers ,also, will shift to those companies who have better results at detecting fraud. Hence,our target audience will be the credit card companies.

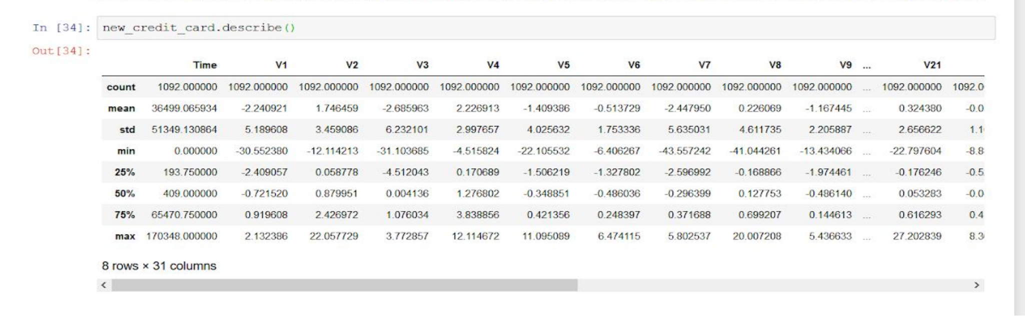
## Data

We will be using kaggle's credit card fraud detection set.It can be find at this link: https://www.kaggle.com/mlg-ulb/creditcardfraud

As per the data provider,the datasets contains transactions made by credit cards in September 2013 by european cardholders.
The 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 numerical 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-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

## Methodology

The given data was normalized beforehand. While studying the data I realized how imbalanced it was. 98% of data had cases which weren't fraud. The model would have overfit and compromised its performance. Hence, resampling was done such that the data contained 600 no fraud cases and 492 fraud cases.
Further EDA was performed on the resampled data.
Below are the basic statistical values I obtained.

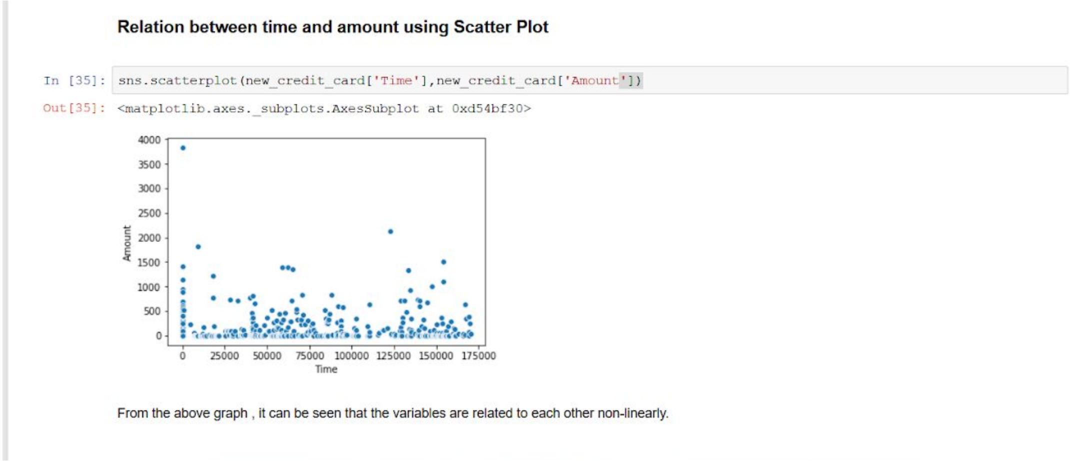


```
In [34]: new_credit_card.describe()
Out[34]:
```

| | Time | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | ... | V21 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | 1092.000000 | ... | 1092.000000 | 1092.0 |
| mean | 36499.065934 | -2.240921 | 1.746459 | -2.685963 | 2.226913 | -1.409386 | -0.513729 | -2.447950 | 0.226069 | -1.167445 | ... | 0.324380 | -0.0 |
| std | 51349.130864 | 5.189608 | 3.459086 | 6.232101 | 2.997657 | 4.025632 | 1.753336 | 5.635031 | 4.611735 | 2.205887 | ... | 2.656622 | 1.1 |
| min | 0.000000 | -30.552380 | -12.114213 | -31.103685 | -4.515824 | -22.105532 | -6.406267 | -43.557242 | -41.044261 | -13.434066 | ... | -22.797604 | -8.8 |
| 25% | 193.750000 | -2.409057 | 0.058778 | -4.512043 | 0.170689 | -1.506219 | -1.327802 | -2.596992 | -0.168866 | -1.974461 | ... | -0.176246 | -0.5 |
| 50% | 409.000000 | -0.721520 | 0.879951 | 0.004136 | 1.276802 | -0.348851 | -0.486036 | -0.296399 | 0.127753 | -0.486140 | ... | 0.053283 | -0.0 |
| 75% | 65470.750000 | 0.919608 | 2.426972 | 1.076034 | 3.838856 | 0.421356 | 0.248397 | 0.371688 | 0.699207 | 0.144613 | ... | 0.616293 | 0.4 |
| max | 170348.000000 | 2.132386 | 22.057729 | 3.772857 | 12.114672 | 11.095089 | 6.474115 | 5.802537 | 20.007208 | 5.436633 | ... | 27.202839 | 8.3 |

8 rows × 31 columns

As can be seen below, the relation between time and amount is non linear.

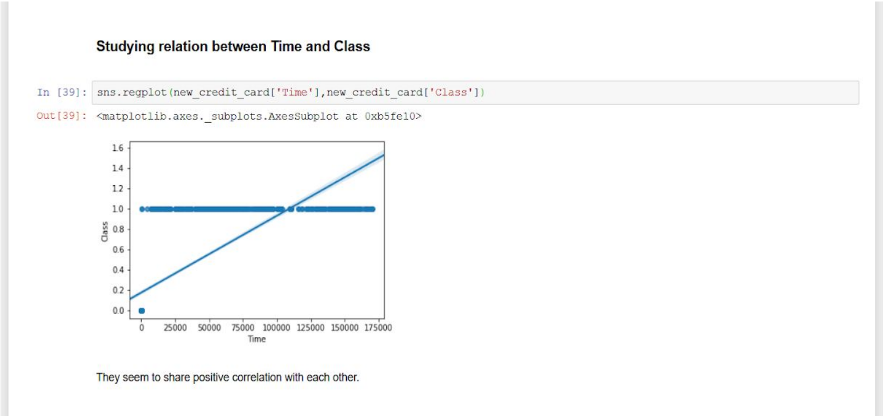


### Relation between time and amount using Scatter Plot

```
In [35]: sns.scatterplot(new_credit_card['Time'],new_credit_card['Amount'])
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0xd54bf30>
```

From the above graph , it can be seen that the variables are related to each other non-linearly.

Below can be seen that there is a positive correlation between time and class.

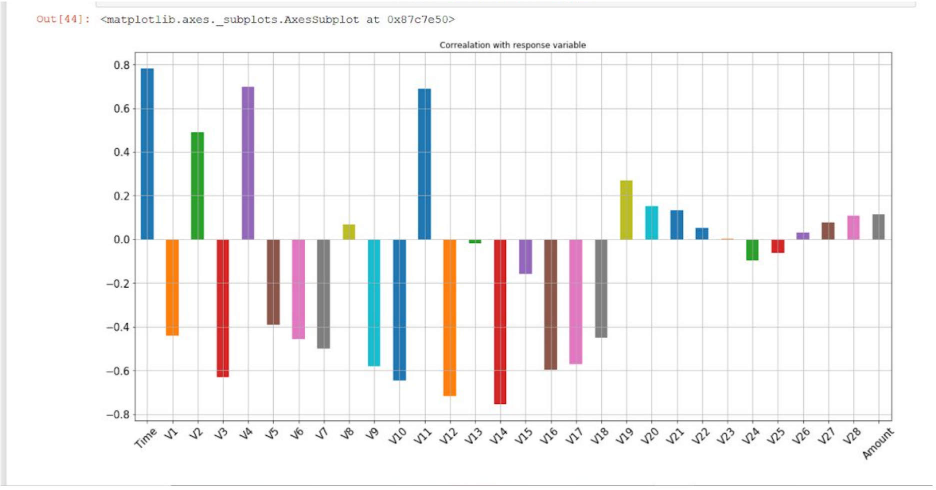


### Studying relation between Time and Class

```
In [39]: sns.regplot(new_credit_card['Time'],new_credit_card['Class'])
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0xb5fe10>
```

They seem to share positive correlation with each other.

A correlation heatmap is used to study correlation between the given parameters of a dataset. I used the correlation values obtained to get the below graph.



```
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x87c7e50>
```

From the above analysis we can conclude that the columns V8,V13,V22,V23,V25,V26 are the least correlated variables with class.We also noticed that Time seems to have the highest positive correlation and V14 the highest negative correlation. Variables V8,V13,V22,V23,V25 and V26 were dropped from the dataset.
After that Logistic Regression Model was implemented as it's performance with binary categorical variables is said to be unparalleled and it did give a good performance**.**

## Results

In this project,I successfully implemented our machine learning model.
I used confusion matrix to determine the final accuracy of the model
I used Logistic Regression and got an accuracy of 88.10%.

## Conclusions

From the above analysis we can conclude that the features V8,V13,V22,V23,V25,V26 are the least correlated variables with class.We also noticed that Time seems to have the highest positive correlation and V14 the highest negative correlation. Surprisingly amount had a very small positive correlation with fraud.