# ABSTRACT

**PROJECT TITLE: Credit Card Fraud Detection**

It is essential for credit card firms to be able to detect fraudulent credit card purchases so that consumers are not charged for things they have not purchased.This project aims to demonstrate the modelling of a data collection using Credit Card Fraud Detection using Machine Learning. Identification of credit card fraud includes modelling past credit card transactions with the data of those who have turned out to be fraud.This model is then used to consider whether or not a new transaction is fraudulent. Our goal here is to identify 100 percent of fraudulent transactions while reducing the classification of incorrect frauds.Credit Card Fraud Detection is an example of classification problem. In this process, we have focusedon analysing and pre-processing data sets as well applied different classification techniques to select best model with and without sampling on the PCA transformed Credit Card Transaction data.

## **Understanding the Problem Statement**

## Online Payments does not require physical card.

* Anyone who knows the details of card can make fraud transactions.
* Currently, card holders come to know only after the fraud transaction is carried out

## 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.

## **Introduction**

Unauthorized and unauthorized use of an account by someone other than the owner of that account is 'fraud' in credit card transactions.Required preventive steps can be taken to avoid this abuse and the actions of such fraudulent activities can be observed in the future to mitigate it and protect it from similar incidents.This is a very important problem which requires automation of the solution to this problem. This dilemma is especially challenging from the learning perspective, as it is defined by various factors such as class imbalance.The number of transactions which are legitimate greatly outnumber fraudulent ones. The transaction trends very frequently alter their statistic properties over time.

Frauds are classified as: -

* Credit Card Frauds: Online and Offline
* Card Theft
* Account Bankruptcy
* Application Fraud
* Device Intrusion
* Telecommunication Fraud

## **How Real-WorldSystem Works: -**

* In real world examples, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize.
* Machine learning algorithms are employed to analyse all the authorized transactions and report the suspicious ones.
* These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent.
* The investigators providea feedback to the automated system which is used to train and update the algorithm to eventually

improve the fraud-detection performance over time.

•Credit Card Frauds: Online and Offline

•Card Theft

•Account Bankruptcy

•Device Intrusi**Introduction to Market Sentiment**

## **DATA SET**

* Dataset used is from Kaggle.
* The dataset contains 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 284,315 transactions.
* It has 30 input features and 1 target variable. The dataset is highly unbalanced, the positive class (frauds) account for 0.173% of all transactions.
* It contains only numerical input variables which are the result of a PCA transformation. Due to confidentiality issues, Kaggle doesn’t provide the background information about the 28 features out of 30. The only Features defined are ‘Time’ and ‘Amount’.
* ‘Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ‘Amount’ is the Transaction Amount. ‘Class’ is the target variable and it is 1 in case of fraud and 0 otherwise.

## **Project Flow**

1. GATHERING DATA

2. DATA PRE-PROCESSING or EDA

3. RESAMPLING TO GET RID OF IMBALANCED DATA

4. APPLIED DIFFERENT CLASSIFICATION TECHNIQUES

5. EVALUATION

6. CONCLUSION

## **RESAMPLING TECHNIQUES USED**

**Under Sampling:**We take random sample of non-fraud class to match number of fraud samples. This makes sure that the training data has equal amount of fraud and non-fraud samples. But we lose information

**OverSampling:** Random oversampling duplicates examples from the minority class in the training dataset and can result in overfitting for some models

**(SMOTE) Synthetic Minority Oversampling Technique:** It overcomes the data imbalance by oversampling the minority class (fraud cases) using nearest neighbours of fraud cases to create new synthetic fraud cases instead of duplicating the minority samples

## **CLASSIFICATION MODELS USED**

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Support Vector Classifier

## **MODEL EVALUATION IMPORTANT TERMINOLOGIES**

**Precision** as the name suggests, tells us how precise (how sure) is our model in detecting fraud transactions

**Recall** is the amount of fraud cases our model is able to detect.

**Precision/Recall Tradeoff:** The more precise (selective) our model is, the less cases it will detect. Example: Assuming that our model has a precision of 95%, Let's say there are only 5 fraud cases in which the model is 95% precise or more that these are fraud cases. Then let's say there are 5 more cases that our model considers 90% to be a fraud case, if we lower the precision there are more cases that our model will be able to detect.

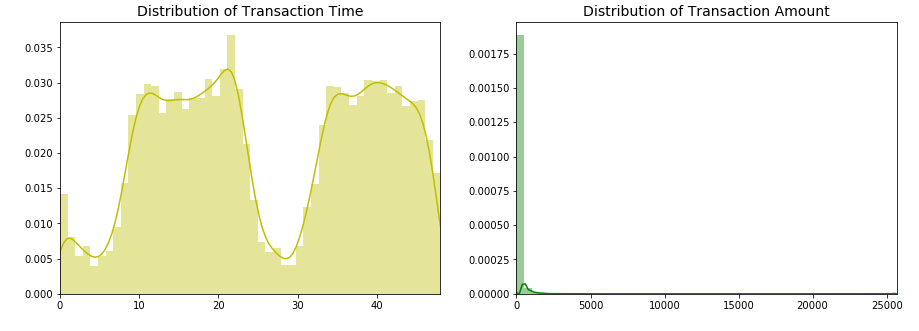
**OTHER IMPORTANT TERMS**

1. True Positives: Correctly Classified Fraud Transactions
2. False Positives: Transactions which are not fraud but classified as fraud
3. True Negative: Correctly Classified Non-Fraud Transactions
4. False Negative: Transactions which are fraud but classified as not fraud
5. Precision (1-Specificty): TP/ (TP + FP) i.e. ratio of correct predicted positive to the total predicted positive
6. Recall or Sensitivity: TP/ (TP + FN) i.e. what percentage of fraud is correctly identified
7. F1 Score = 2(Recall. Precision) / (Recall + Precision) is the weighted average of Precision and Recall.

**Summary of dataset**

* The data set we have used is already in the form of principal components to maintain the privacy of data.
* The dataset has a total of 284807 rows with 30 features and 1 target feature.
* There are no missing values
* Most of the transactions were Non-Fraud (99.83%) and Fraud transactions are 0.17% in the data frame. The dataset is highly imbalanced.
* The features given in the data frame from V1 to V28 are principal vectors
* Amount and Time are not principal factors and are in the original form.

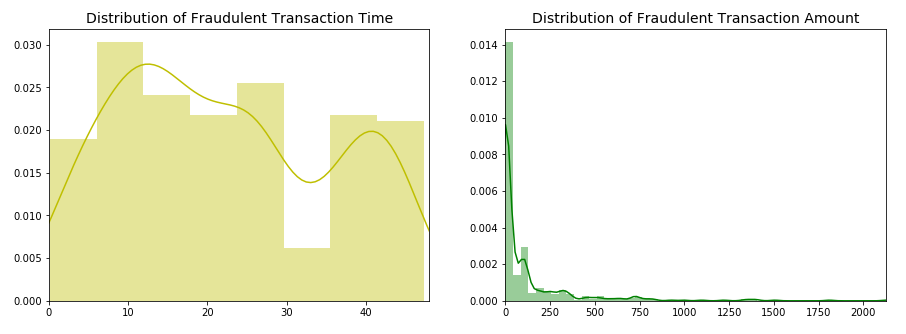
**Distribution of Transaction time & Amount**

****

### Summary:

* Amount: There are mostly small amount transactions in the database
* Amount: Mean comes out to be 88 dollars and 75th percentile is at 77 dollars
* Time: as it is evident from the distribution that most of the transactions occured during the daytime

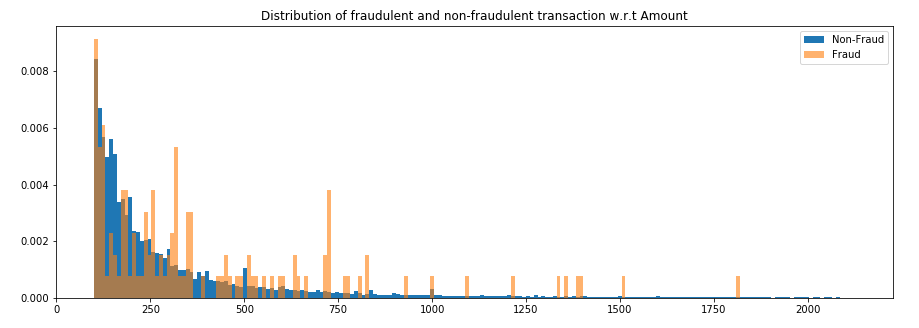
**Distribution of Fraudulent Transaction time & Amount**



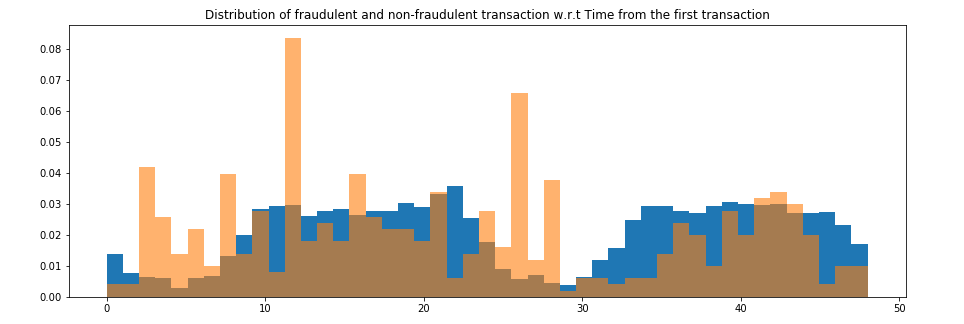
### Summary:

* Amount: There are higher fraud cases for smaller amounts
* The chances of occurence of fraud during night time is greater as compared to day time.

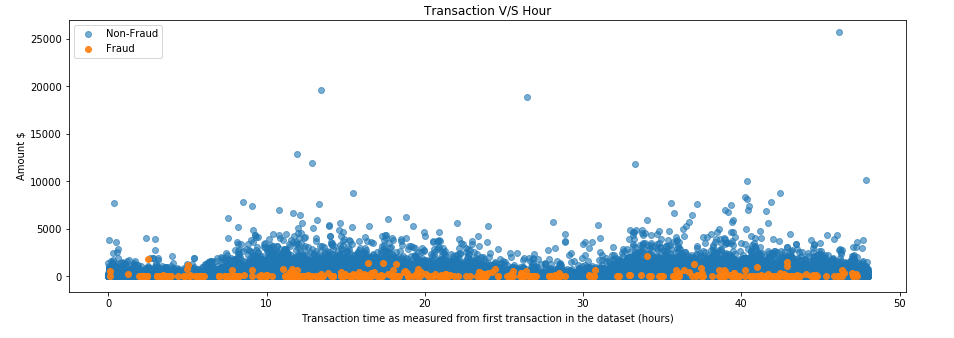
**Distribution of fraudulent and non-fraudulent transaction wrt Amount**



**Distribution of fraudulent and non-fraudulent transaction wrt Time for first transaction**

****

**Transaction vs Hour**

****

### Hence, it is difficult to clearly draw a line which can separate fraud and non-fraud transactions

#### As we already know that, V1 to V28 are principal components and before converting features into principal components, scaling is done. Therefore, we are required to do scale "Amount" and "Time", before proceeding further for model building.

# Handling imbalanced data

We can either do **undersampling** of data or **oversampling** of data. What happens in both of them is described below

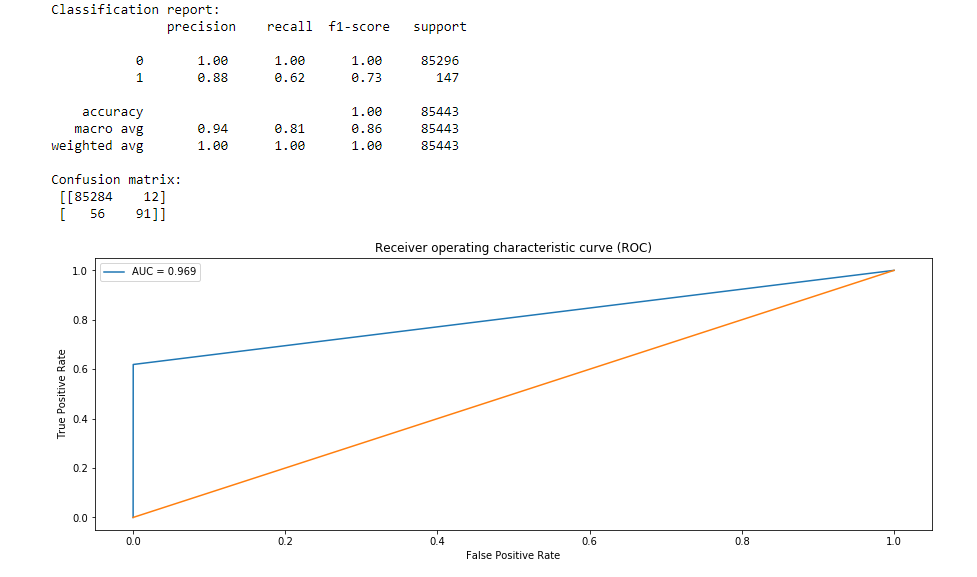
* UnderSampling: we take ramdom draws from non-fraud observations to match the number of fraud observations. But we randomly loose alot of information in this
* OverSampling: Random oversampling duplicates examples from the minority class in the training dataset and can result in overfitting for some models

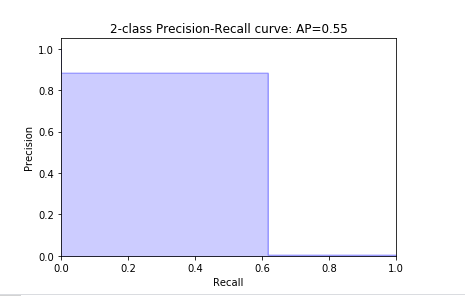
Apart from these two, we have another method, which is proved more effective as compared to other two mentioned.

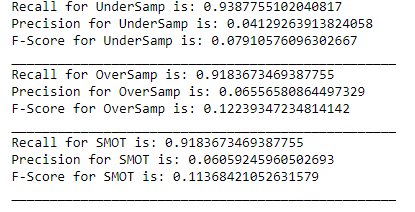
(SMOTE) Synthetic Minority Oversampling Technique: It overcomes the data imbalance by oversampling the minority class (fraud cases) using nearest neighbors of fraud cases to create new synthetic fraud cases instead of duplicating the monority samples.

**Evaluation of various ML models**

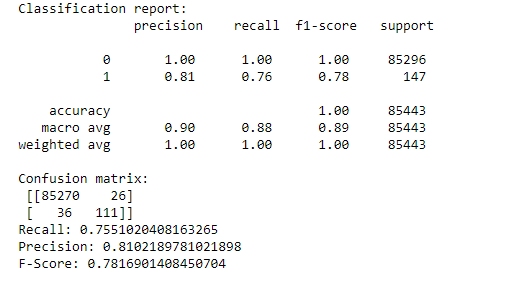
**Logistic Regression**

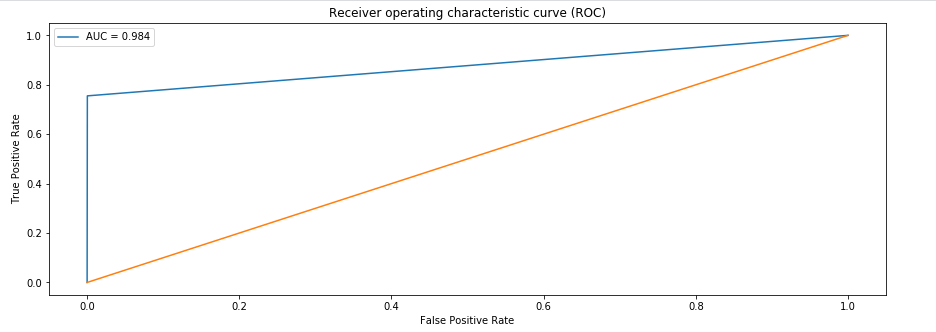
****

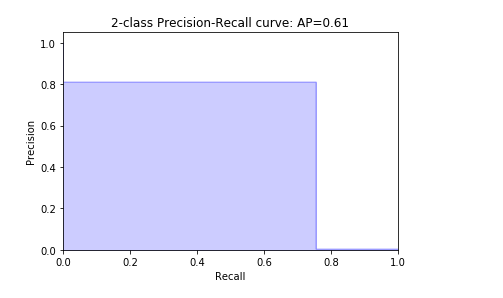


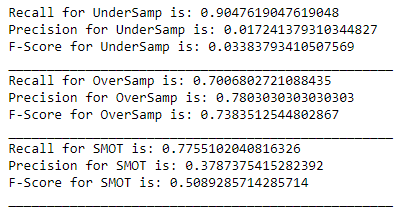


# Decision Tree Classifier

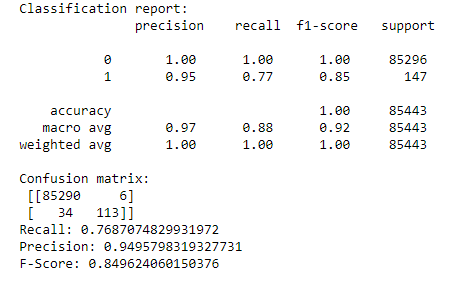


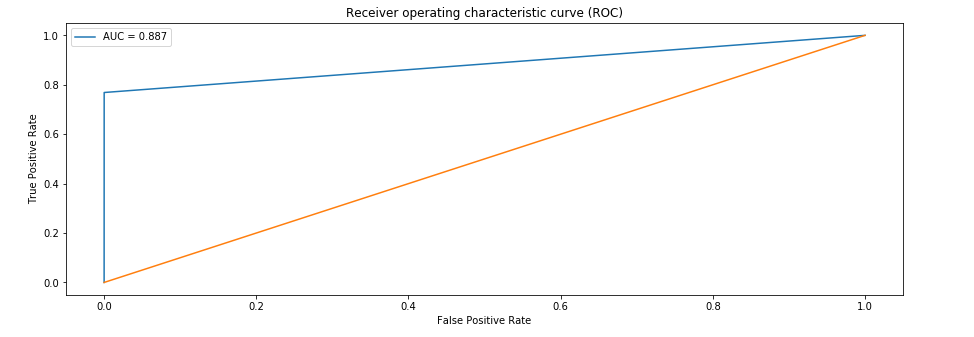


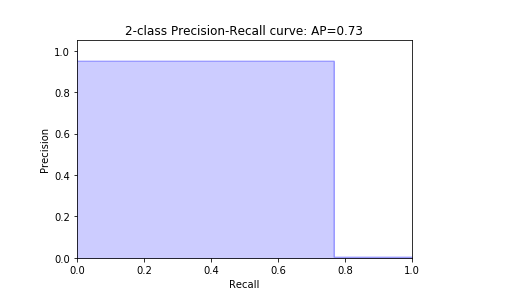




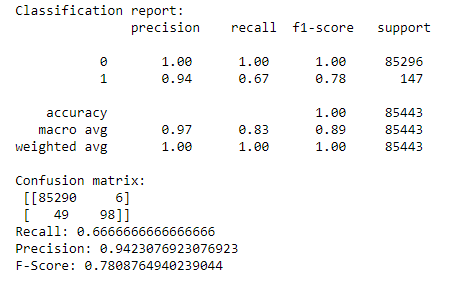
# Random Forest Classifier

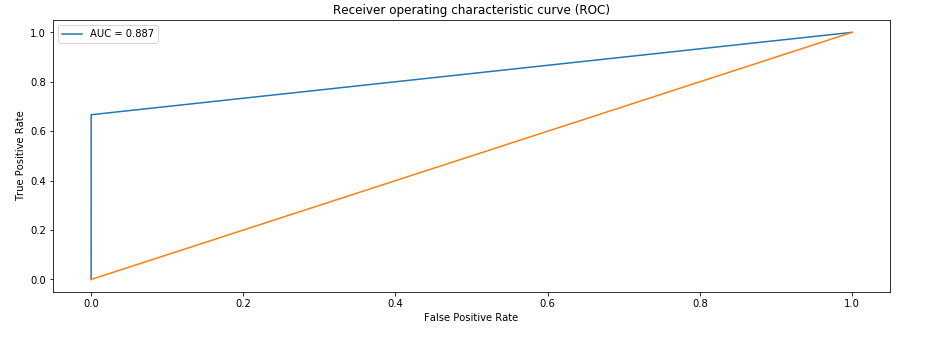


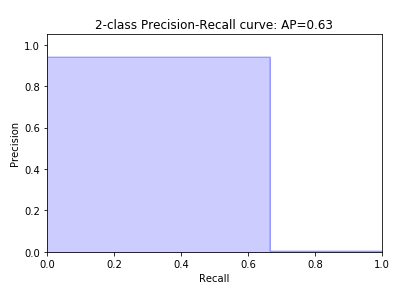




# Support Vector Classifier







**Results and Conclusion**

* **The best model suited for this scenario is Random Forest Classifier**
* **Re-sampling does not result in high precision and recall in all the cases**
* **The occurrence of fraud during in night time is greater**

**References used**

* **www.kaggle.com**
* **www.towardsdatascience.com**
* **www.geeksforgeeks.org**
* **www.stackoverflow.com**