**CREDIT CARD FRAUD DETECTION SYSTEM USING MACHINE LEARNING**

*A project report submitted to ICT Academy of Kerala*

*in partial fulfillment of the requirements*

*for the certification of*

**CERTIFIED SPECIALIST**

**IN**

**DATA SCIENCE & ANALYTICS**

Submitted by

**Maneesha K**

**Amrutha P Nair**

**Shamiya C K**

**Erza Nizar**

A close up of a sign

Description automatically generated

**ICT ACADEMY OF KERALA**

**THIRUVANANTHAPURAM, KERALA, INDIA**

**January 2021**

**LIST OF FIGURES**

1 Proposed system 11

2 Countplot of Class 15

3 Scatter Plot of amount and time 16

4 Distribution plot of Amount 17

5 Pairplot 18

6 Heatmap 19

7 Histogram 20

**LIST OF TABLE AND ABBREVIATIONS**

**List of Tables**

1 Dataset description 9

2 Results 38

**List of Abbreviations**

PCA *Principal Component Analysis*

SMOTE Synthetic Minority Over-sampling technique

**TABLE OF CONTENTS**

Abstract 5

1. Problem Definition 6

1.1 Project overview 6

1.2 Problem statement 6

1.3 Objectives 6

1.4 Domain knowledge 6

2. Introduction 7

3. Literature review 8

4. Dataset Analysis 9

4.1 Importing python libraries 10

4.2 Importing the data 10

4.3 Data source 10

5. Proposed Method 11

5.1 Data collection 11

5.2 Data Preprocessing 14

5.3 Exploratory data Analysis 15

5.4 Modelling 21

5.5 Result 38

6. Conclusion 40

References

**ABSTRACT**

A credit card is a convenient financial product that can be used for everyday purchases. Frauds in credit card transactions are common today as most of us are using the credit card payment methods more frequently. This is due to the advancement of technology and increase in online transactions and credit card fraud comes out as a serious problem.The main challenges in credit card fraud detection are to recognize fraudulent credit card transactions so that the customers of credit card companies are not charged for items that they did not purchase. Huge amount of data is processed everyday and most of the transactions are not fraudulent, which makes it really hard for detecting the fraudulent ones. Therefore, there is need for effective methods to reduce the loss.

So, we can use Machine learning techniques due to their beneficial characteristics to build a good fitting model to identify fraudulent transactions.Our aim is to build a machine learning model that must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible. For this study we have used the publically available credit card fraud detection dataset from Kaggle. The dataset is highly imbalanced with 492 frauds out of 284,807 transactions. We have applied different machine learning techniques and their performances are compared based on accuracy, precision, recall and execution time. The model that gives the best performance in less time is used for fraud detection.

**1. PROBLEM DEFINITION**

* 1. **Project overview**

We are given a dataset containing theinformation about the credit card transactions of people, the information that they are fraud or not, and the objective is to differentiate between them. This is the case we are going to deal with. Our ultimate intent is to tackle this situation by building classification models to classify and distinguish fraud transactions.

**1.2 Problem statement**

Modelling of 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 analyzing and pre-processing data sets including balancing the dataset as well as the deployment of multiple machine learning algorithm on the PCA transformed Credit Card Transaction data.

**1.3 Objectives**

The objectives of the project are to implement machine learning algorithms to detect credit card frauds with respect to time and amount of transaction.

**1.4 Domain Knowledge**

This project come under the domain of Banking and finance.Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage

**2. INTRODUCTION**

A credit card is a physical medium for selling goods or services without having cash in hand. Credit card fraud detection is the procedure to identify whether a transaction is normal or abnormal. In recent years, as there is advancement of technology, most of them are using credit card for buying their needs so the frauds associated with it is also rising gradually. In the present world, almost all the enterprises from small to big industries are using the credit card as mode of payment.

Credit card fraud can be due to use of card by unauthorized cardholder using false identity taking bank’s official in confidence, or it may also be due to use of stolen credit cards. The fraudulent transactions are divided into two kinds: online fraud and offline fraud; the first is performed by using a stolen physical card for example at the mall etc, while the second is committed by stealing victim identities such as credit card numbers, name of the credit card holder, expiry date, and passwords .The detection of Credit card fraud becomes a relevant research area in the last few years and several studies use Machine learning and data mining approaches to handle this problem.

The major challenges associated with this problem is that fraudulent transactions often look like legitimate ones and the credit card databases are not easily available. The classification of fraudulent transactions with highly imbalanced classes of data is also a challenging task. Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behaviour, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated. This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumbers fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

In our study we have used all the supervised machine learning techniques to detect the fraudulent transaction, we have also taken care of the imbalanced dataset we have. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications

**3.LITERATURE REVIEW**

Fraud act as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already.

Samidha Khatri, Aishwarya Arora, and Arun Prakash Agrawal from Amity University, Uttar Pradesh conducted a comparative study for Credit card fraud detection using different supervised machine learning techniques. They have used Decision tree, Random Forest, Naïve Bayes, and Logistic Regression classifiers for fraud detection. The analysis shows that the sensitivity of the kNN model is greater than that of Decision tree, but as time taken by kNN for testing the data is very large, they choose Decision Tree over kNN. In case of fraud detection. According to them ensure that minimum time is taken for prediction, therefore, Decision Tree is the preferred mode.

Fatima Zohrafrom University Sidi Mohammed Ben Abdellah Atlas, Fez, Moroccoproposed a method in which they try to detect fraudulent transactions using two artificial neural network classifiers, Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM), applied on the credit card fraud dataset. The performance of these classifiers is evaluated based on accuracy, recall, precision, and classification time. The results show that the accuracy of MLP and ELM classifiers achieves respectively 97.84% and 95.46%. Otherwise, ELM is very fast for predicting new fraudulent transactions.

In another paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection.

**4.DATASET ANALYSIS**

The dataset contains the transactions made by credit cards in September 2013 by European Card holders. 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 numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, the original features and more background information about the data are not available. 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.

Confusion matrix accuracy is not meaningful for unbalanced classification. Dataset link:<https://www.kaggle.com/mlg-ulb/creditcardfraud>

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data type |
| Time | Time gap between first transaction and following one | float |
| V1, V2...V28 | PCA transformed values | float |
| Amount | Amount of money transacted | float |
| Class | Class label (0-normal 1-fraud) | int |

**4.1 Importing python libraries**

For this project, our primary packages are going to be Pandas to work with data, NumPy to work with arrays, scikit-learn for data split, building and evaluating the classification models, and finally the xgboost package for the xgboost classifier model algorithm. We have imported all of our primary packages into our python environment.

**4.2 Importing the data**

In this step we have to import the dataset, for that we can use the read function. The libraries needed for there is pandas. We can use excel or CSV format according to our dataset. Here we have our data set in a CSV format and we used the read\_csv () function to import dataset

**4.3 Data Source**

This dataset is from Kaggle

Dataset link:<https://www.kaggle.com/mlg-ulb/creditcardfraud>

**5. PROPOSED METHOD**

The proposed method consists of the following methods

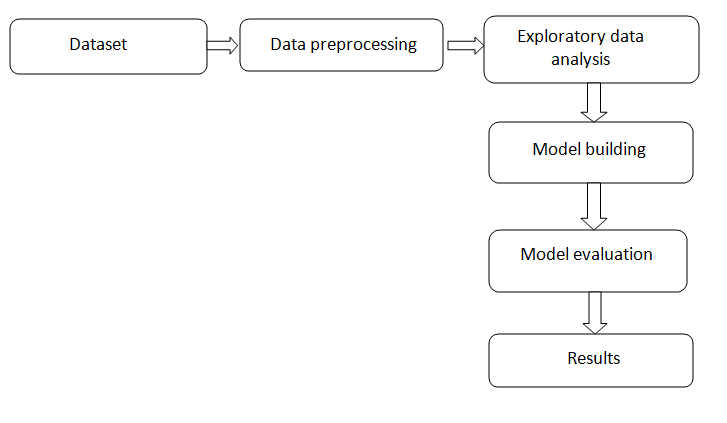


Fig 1 Proposed method

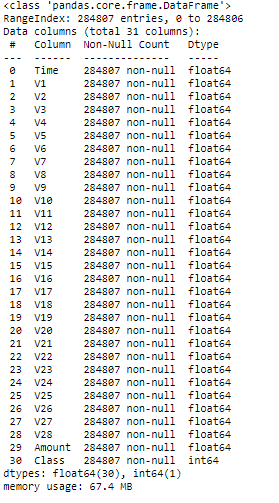
**5.1 Data collection**

We have collected the credit card dataset from Kaggle. The detailed description of the dataset is given in section 2.

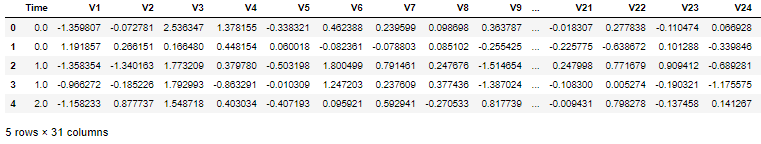
The dataset is loaded into python Jupiter notebook environment using pandas read\_csv () function (credit card dataset is in csv format). Then the basic details like shape, columns etc are checked.

The credit card dataset consists of 284807 rows and 31 columns including the class attribute.

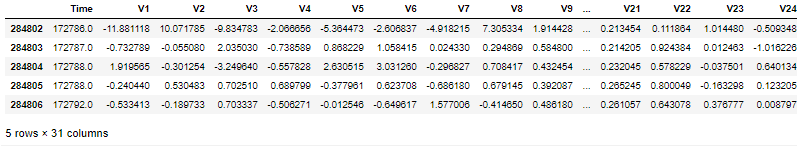
The detailed information of the dataset is checked using info () function. Pandas **info ()** functionis used to view the basic details like column name, null values, dtypes, memory usage.



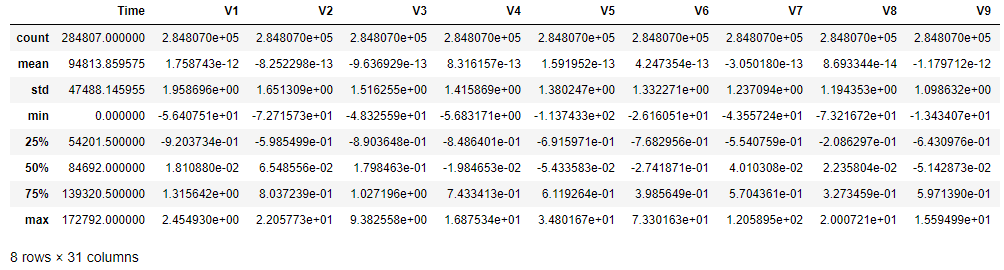
Pandas **head ()** method is used to return top n (5 by default) rows of a data frame or series.



Pandas **tail ()** method is used to return bottom n (5 by default) rows of a data frame or series.



Pandas **describe ()** is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values.

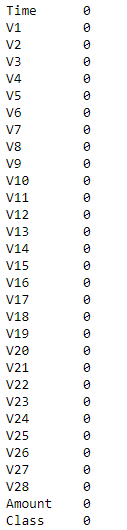


The data set contains 284,807 transactions. The mean value of all transactions is 88.35USD while the largest transaction recorded in this data set amounts to 25,691USD. The vast majority of transactions are relatively small and only a tiny fraction of transactions comes even close to the maximum.

**5.2. Data preprocessing**

Data Preprocessing is an important task in data mining that is used to transform the raw data into an efficient format. The quality of data has a greater influence on the quality of the model being built. Data preprocessing helps in improving the accuracy of the model and also increases the speed for execution. So, it is necessary to preprocess the data before feeding it into the model. Preprocessing includes data cleaning, data integration, and data transformation, and data reduction

Pandas **isna ()** function is used to detect missing values. Here our credit card dataset contains no missing values.



**5.3. Exploratory Data Analysis**

**Count plot of Class attribute**

To know how many samples are available for both the fraudulent and normal transactions we have ploted a countplot using seaborns countplot()function. We have also printed the counts percentage of each class. The figure is shown here.

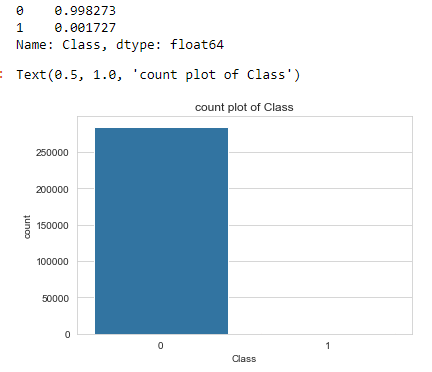
****

Fig 2. Countplot of Class

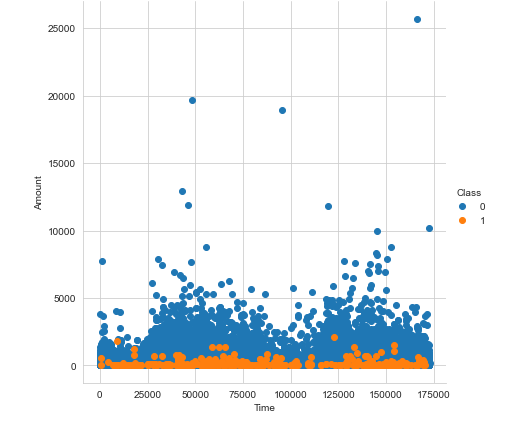
Observations:

* From the above graph it is clear that 99.8273 % of our total observations belongs to normal class (0) and only 0.17% belongs to fraudulent class.This graph shows that the number of fraudulent transactions is much lower than the legitimate ones.

**Scatter plot of Amount and Time attributes**

Scatter plots are useful to show individual values plot on a two dimensional cartesian X & Y plane from two Series in a Pandas DataFrame.

Here we have plotted the scatter plot between Amount and Time to know when most of the fraudulent transactions has taken place and the amounts.

****

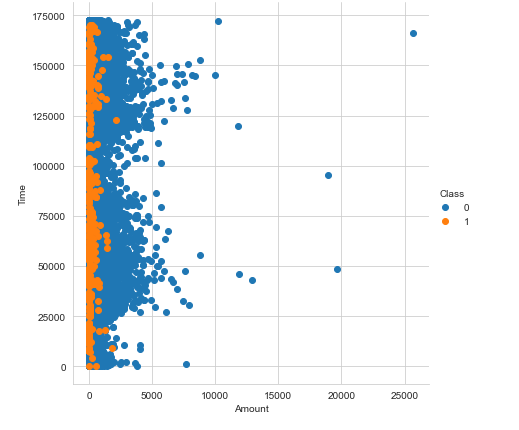
****

Fig 3. Scatter plot of amount and Time

Observations:

* From the two scatter plots it is visible that there are frauds only on the transactions which have transaction amount less than 2500. Transactions above approximately 2500 have no fraud.
* As per with the time, frauds in the transactions are evenly distributed with time.

**Distribution plot of Amount**

Distribution plots are used to visualize univariate distributions of observations

They can be used to identify outliers, identify how normal a dataset is, and whether there are potential gaps in your dataset, along with other applications. We have plotted the distribution plot of Amount

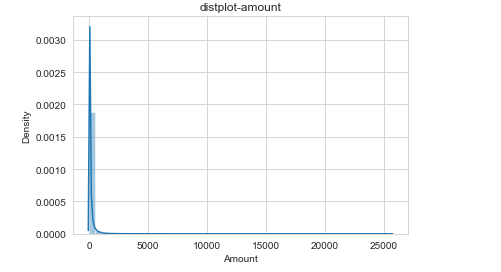
****

Fig 4 distplot of amount

Observation:

* 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. Amount is not normally distributed. So before any analysis, we have to scale the data and transform.

**Pairplot of Amount ,time and class attribute**

Pair plots can play a similar role to individual scatter plots as they provide a variety of visualizations. We have plotted all the features except the PCA transformed values.

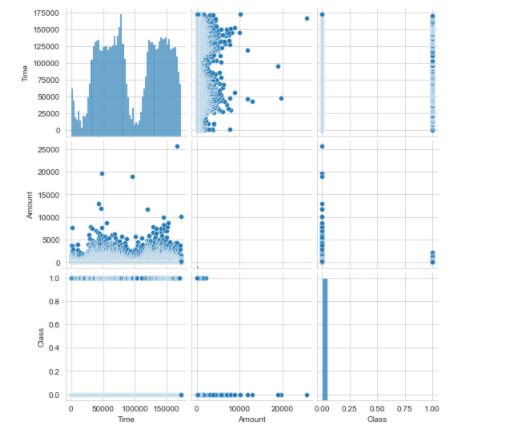


Fig 5. Pairplot

Observatons

* The figure shows pairwise relationships between amount,time and class attributes in the dataset.

**Correlation heatmap**

We plot a heatmap to get a coloured representation of the data and to study the correlation between out predicting variables and the class variable. This heatmap is shown below:

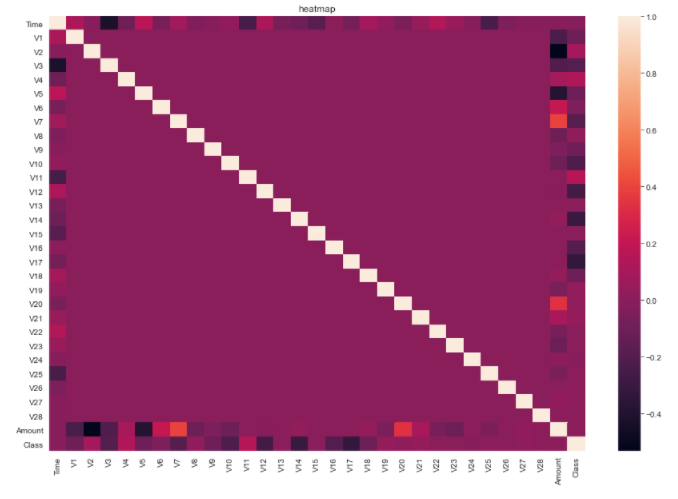
****

Fig 6 correlation heatmap

Observation:

* ‘Class’ is less correlated with ‘Amount’ and ‘Time’ which suggests it is hard to predict whether transaction is fraudulent or not from ‘Amount’ and ‘Time’ details of transaction.‘Class’ is negatively correlated with ‘V3’, ‘V7’, ‘V10’, ‘V12’, ‘V14’, ‘V17’ and positively correlated with ‘V2’, ‘V4’, ‘V11’. The other correlations are relatively small.
* There are no significant correlations between the reduced features (‘V1’ to V’28'). Therefore **we don’t drop any of the columns as they are fairly unrelated to each other**.

**Histogram of columns V1,V2,..V28**

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

We have plotted histogram of all PCA transformed attributes(v1,v2,..v28)

****

Fig7 . Histogram

**5.4. Modelling**

**5.4.1. Data Preparation**

While the anonymized features have been scaled and seem to be centered around zero, our time and amount features have not. Not scaling them as well would result in certain machine learning algorithms that give weights to features (logistic regression) or rely on a distance measure (KNN) performing much worse. To avoid this issue, first we have standardized both the time and amount column.

**Standard Scaler on Time and Amount:**

* StandardScaler is a common tool when working with classification problems like this. It transforms the data to where there is a mean of 0 and a standard deviation of 1, thus standardizing the data into a normal distribution. Especially with working on such a wide range of amounts and time

We should not be happy for high accuracy of test set. If the precision/recall score is low then there is grave problem in the model.

* Low precision and high recall means the model detects most of the features as fraud including the genuine transactions.
* High precision and low recall mean the model precisely detects the non-frauds but labels the frauds as genuine too.

By considering these issues, we have to build a model that can have highly balanced precision-recall/ f1 score, for that we have to check beyond accuracy score.

**5.4.2 Train test splitting for imbalnced dataset**

 Creating a training data set that will allow our algorithms to pick up the specific characteristics that make a transaction more or less likely to be fraudulent. Using the original data set would not prove to be a good idea for a very simple reason: Since over 99% of our transactions are non-fraudulent, an algorithm that always predicts that the transaction is non-fraudulent would achieve an accuracy higher than 99%. Nevertheless, that is the opposite of what we want. We do not want a 99% accuracy that is achieved by never labeling a transaction as fraudulent, we want to detect fraudulent transactions and label them as such. Before giving data into the machine learning models, we have to split the data into test and train sets. Train test splitting is done using train\_test\_split () function in sklearn. We have used 70% of the data for training and 30 % for testing

**Different Sampling methods to deal with Data Imbalance:**

1. Random Under-Sampling : Random draws are taken from the non-fraud observations i.e the majority class to match it with the Fraud observations i.e the minority class. This means, we are throwing away some information from the dataset which might not be ideal always. The Fig below illustrates this methodology.
2. Random Over-Sampling: In this case, we do exact opposite of under-sampling i.e duplicate the minority class or the fraudulent observations at random to increase the number of the minority class till we get a balanced dataset. Here the possible limitation is, we are creating a lot of duplicates with this method
3. SMOTE:( Synthetic Minority Over-sampling technique): Smote is another method that uses synthetic data with KNN instead of using duplicate data. Each minority class example along with their k-nearest neighbours is considered. Then synthetic examples are created along the line segments that join any/all the minority class examples and their k-nearest neighbours

With only over-sampling, the decision boundary becomes smaller while with SMOTE we can create larger decision regions thereby improving the chance of capturing the minority class better.

**5.4.3. Training different Machine Learning models**

Our main objective in this process is to find the best model for our given case. The evaluation metrics we are going to use are the reacall score,precision score ,f1 score ,accuracy score and finally the confusion matrix.

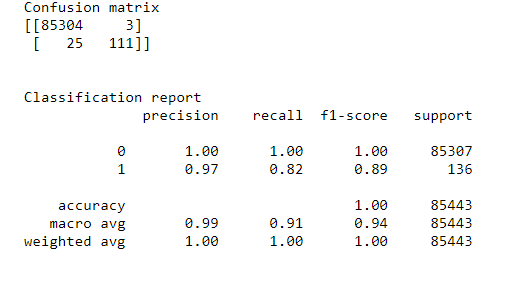
* **TP (True positive)/TN (True negative)** are the cases of correct predictions i.e predicting Fraud cases as Fraud (TP) and predicting non-fraud cases as non-fraud (TN)
* **FP (False positive)** are those cases that are actually non-fraud but the model predicted as Fraud
* **FN (False negative)** are those cases that are actually fraud but the model predicted as non-Fraud
* **Accuracy:** Measures how many majority as well as minority classes could be correctly classified., Accuracy = (TP +TN)/(TP+FP+FN+TN)
* **Recall** = TP/ (TP+FN): Recall measures out of all the actual fraud cases, how many the model could predict correctly as fraud. This is an important metric here.
* **Precision** = TP / (TP + FP): Precision measures how accurately the model is able to capture fraud i.e out of the total predicted fraud cases, how many actually turned out to be fraud.
* **F-score** = 2\*TP/ (2\*TP + FP +FN) = 2\* [(Precision \*Recall)/ (Precision +Recall)]; this is a balance between precision and recall.Precision and recall are inversely related, hence F-score is a good measure to achieve a balance between the two.

**Random Forest Classifier**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees

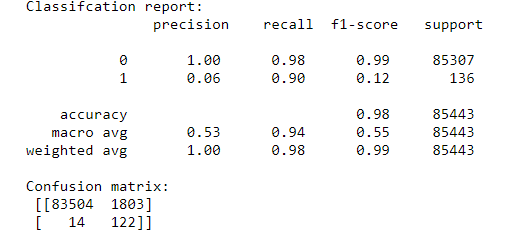
We have trained Random Forest classifier with and without sampling technique. First we have trained Random Forest without any sampling technique and then applied random under sampler, random over sampler and SMOTE respectively. The confusion matrix and accuracy,precision,recall are shown in the results below

* **Without any sampling**:



* + No sampling result interpretation: Without any sampling we are able to capture 111 fraudulent transactions. The precision is 99%,and the recall is 91%. This means that there are quite a few fraudulent transactions that our model is not able to capture. In this case, more than precision we are more concerned about recall because we want to catch as many fraudulent cases as possible
* **Random Under-Sampling:**

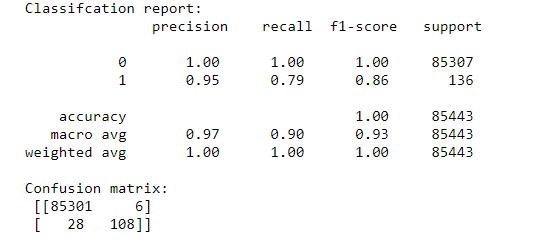
We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the random forest model. Resampling is done using RandomUnderSampler from imblearn.under\_sampling class



### Under-sampling result interpretation: With under-sampling , though the model is able to capture 122 fraud cases with significant improvement in recall, the accuracy and precision falls drastically. This is because the false positives have increased phenomenally and the model is penalizing a lot of genuine transactions. With undersampling the recall improves to 94% and we are able to capture more Frauds (122 vs 111).But precision drops a lot and so does the F-score

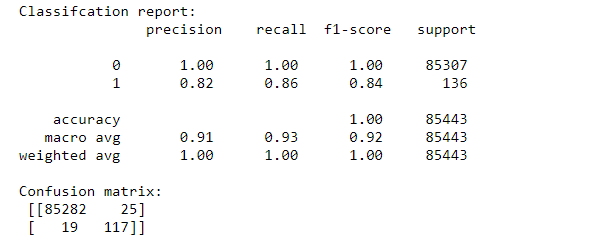
* **Random Over-Sampling**:

We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the random forest model. Resampling is done using RandomOvererSampler from imblearn.over\_sampling class



* + Over-sampling result interpretation: Over-sampling method has the high precision and accuracy and the recall is also good at 90%. False positives is pretty low as well.
* **SMOTE** :

Finally we have implemented the SMOTE method with the random forest model. SMOTE is imported from imblearn.over\_sampling class.



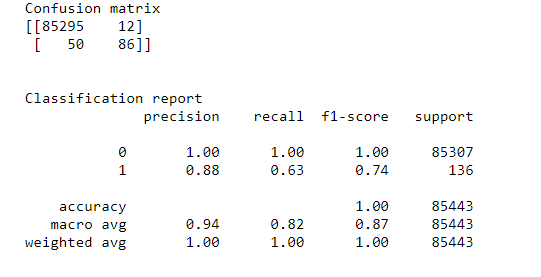
* SMOTE further improves the over-sampling method with 9 more frauds caught in the net and though false positives increased a bit the recall is pretty healthy at 93%.

**Logistic Regression**

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. Logistic regression has become an important tool in the discipline of machine learning

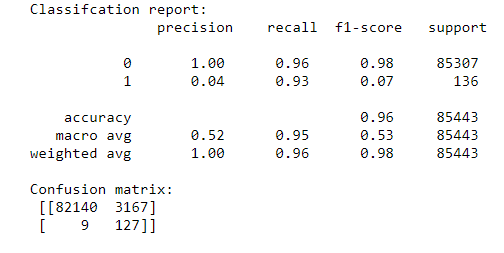
We have trained Logistic Regression with and without sampling technique. First we have trained Logistic Regression without any sampling technique and then applied random under sampler, random over sampler and SMOTE respectively. The confusion matrix and accuracy,precision,recall are shown in the results below

* **Without any sampling:**



* + No sampling result interpretation: Without any sampling we are able to capture 86 fraudulent transactions. The precision is 94%,and the recall is 82%. This means that there are quite a few fraudulent transactions that our model is not able to capture. In this case,more than precision we are more concerned about recall because we want to catch as many fraudulent cases as possible
* **Random Under-Sampling:**

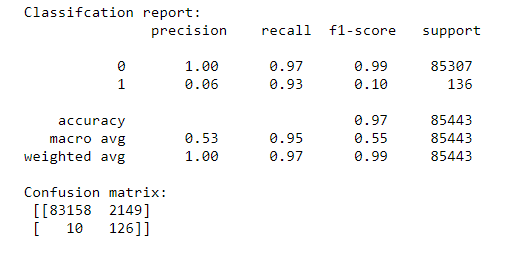
We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Logistic Regression model. Resampling is done using RandomUnderSampler from imblearn.under\_sampling class



### Under-sampling result interpretation: With under-sampling , though the model is able to capture 127 fraud cases with significant improvement in recall (95%), the accuracy and precision falls drastically. This is because the false positives have increased phenomenally and the model is penalizing a lot of genuine transactions. With undersampling the recall improves to 95% and we are able to capture more Frauds (127 vs 86).But precision drops a lot and so does the F-score

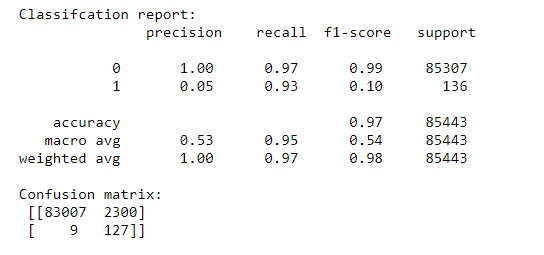
* **Random Over-Sampling:**

We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Logistic Regression model. Resampling is done using RandomOverSampler from imblearn.over\_sampling class



* + Over-sampling result interpretation: Over-sampling method has the precision 53% and accuracy and the recall is also good at 95%.
* **SMOTE:**

We have implemented the SMOTE method with the Logistic Regression model. SMOTE is imported from imblearn.over\_sampling class.

****

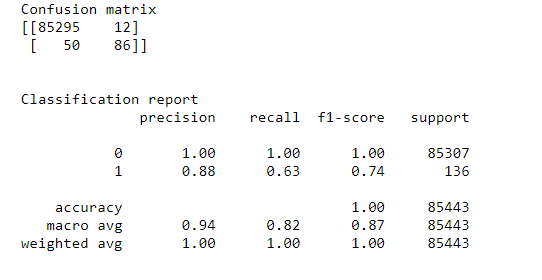
* SMOTE further improves the over-sampling method with 1 more frauds caught in the net and though false positives increased a bit the recall is pretty healthy at 95%.

**Decision Tree**

Decisions Trees is a powerful group of supervised Machine Learning models that can be used for both classification and regression.Decision Trees have been some of the Machine Learning models to deliver the best results at competitions.Decision trees are one of the most fundamental Machine Learning tools which are used for both classification and regression tasks.

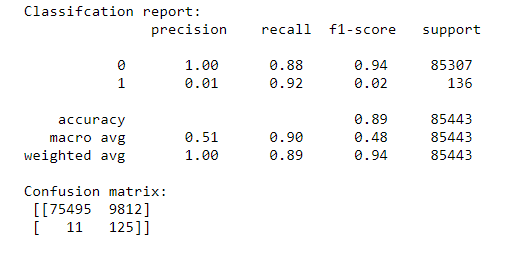
We have trained Decision Tree with and without sampling technique. First we have trained Decision Tree without any sampling technique and then applied random under sampler, random over sampler and SMOTE respectively. The confusion matrix and accuracy,precision,recall are shown in the results below

* **Without any sampling**



* + No sampling result interpretation: Without any sampling we are able to capture 86 fraudulent transactions. The precision is 94%,and the recall is 82%. This means that there are quite a few fraudulent transactions that our model is not able to capture. In this case, more than precision we are more concerned about recall because we want to catch as many fraudulent cases as possible
* **Random Under-Sampling:**

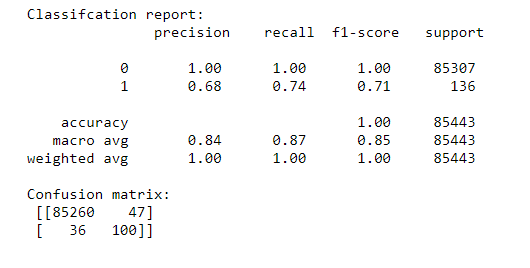
We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Decision Tree model. Resampling is done using RandomUnderSampler from imblearn.under\_sampling class



### Under-sampling result interpretation: With under-sampling , though the model is able to capture 128 fraud cases with significant improvement in recall (90%), the accuracy and precision falls drastically. This is because the false positives have increased phenomenally and the model is penalizing a lot of genuine transactions. With undersampling the recall improves to 90% and we are able to capture more Frauds (125 vs 86).But precision drops a lot and so does the F-score

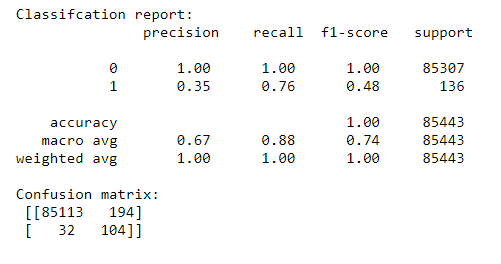
* **Random Over-Sampling:**

We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Decsion Tree model. Resampling is done using RandomOverSampler from imblearn.over\_sampling class



* + Over-sampling result interpretation: Over-sampling method has the high precision and accuracy and the recall is at 87%.we are able to capture more Frauds (100 vs 86).
* **SMOTE:**

We have implemented the SMOTE method with the Logistic Regression model. SMOTE is imported from imblearn.over\_sampling class.



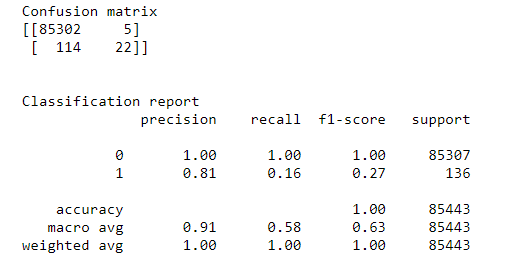
* SMOTE further improves the over-sampling method with 4 more frauds caught in the net and though false positives increased a bit the recall is pretty healthy at 88%.

**Gradient Boosting Classifier**

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees

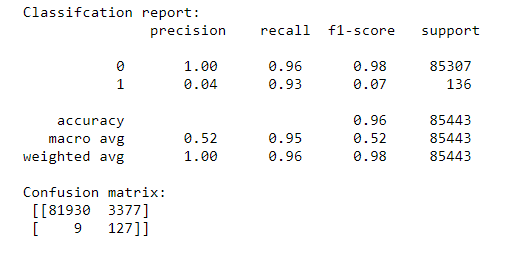
We have trained Gradient Boosting classifier with and without sampling technique. First we have trained Gradient Boosting without any sampling technique and then applied random under sampler, random over sampler and SMOTE respectively. The confusion matrix and accuracy,precision,recall are shown in the results below

* **Without any sampling:**



* + No sampling result interpretation: Without any sampling we are able to capture only 22 fraudulent transactions. The precision is 91%,and the recall is only 58%. This means that there are quite a few fraudulent transactions that our model is not able to capture. In this case, more than precision we are more concerned about recall because we want to catch as many fraudulent cases as possible
* **Random Under-Sampling:**

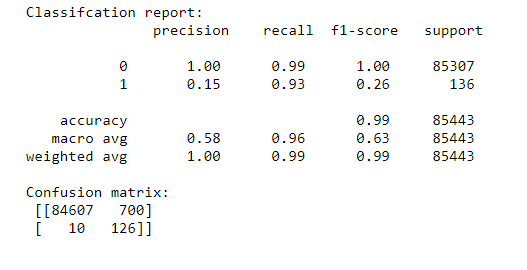
We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Gradient Boosting model. Resampling is done using RandomUnderSampler from imblearn.under\_sampling class



### Under-sampling result interpretation: With under-sampling , though the model is able to capture 127 fraud cases with significant improvement in recall (95%), the accuracy and precision falls drastically. This is because the false positives have increased phenomenally and the model is penalizing a lot of genuine transactions. With under sampling the recall improves to 95% and we are able to capture more Frauds (127 vs 22).But precision drops a lot and so does the F-score

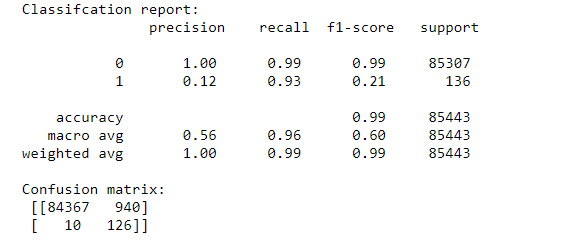
* **Random Over-Sampling**:

We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the Gradient Boosting model. Resampling is done using RandomOverSampler from imblearn.over\_sampling class



* + Over-sampling result interpretation: Over-sampling method has the precision 58% and the recall is at 96%.We are able to capture more Frauds (126 vs 22).
* **SMOTE:**

We have implemented the SMOTE method with the Gradient Boosting model. SMOTE is imported from imblearn.over\_sampling class.

****

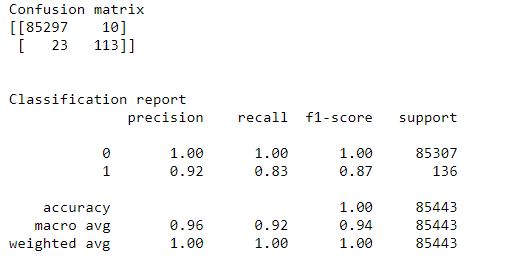
* SMOTE in this case does not improves the over-sampling method and though false positives increased a bit the recall is pretty healthy at 96%.

**XGB Classifier**

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

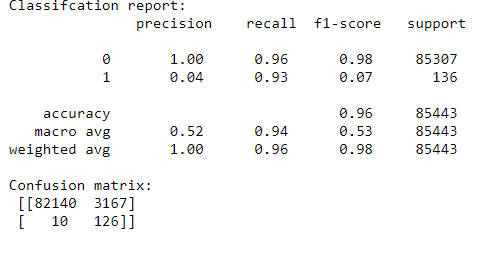
We have trained XGBclassifier with and without sampling technique. First we have trained XGB without any sampling technique and then applied random under sampler, random over sampler and SMOTE respectively. The confusion matrix and accuracy,precision,recall are shown in the results below

* **Without any sampling**



* + No sampling result interpretation: Without any sampling we are able to capture 113 fraudulent transactions. The precision is 96%,and the recall is 92%. This means that there are quite a few fraudulent transactions that our model is not able to capture. In this case, more than precision we are more concerned about recall because we want to catch as many fraudulent cases as possible
* **Random Under-Sampling:**

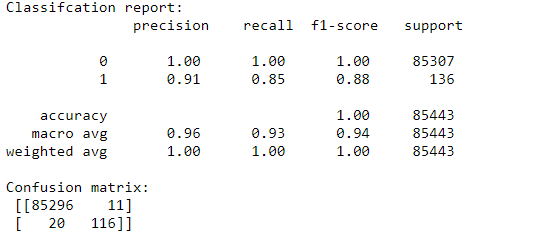
We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the XGB model. Resampling is done using RandomUnderSampler() from imblearn.under\_sampling class



### Under-sampling result interpretation: With under-sampling , though the model is able to capture 126 fraud cases with significant improvement in recall (94%), the accuracy and precision falls drastically. This is because the false positives have increased phenomenally and the model is penalizing a lot of genuine transactions. With undersampling the recall improves to 95% and we are able to capture more Frauds (126 vs 113).But precision drops a lot and so does the F-score

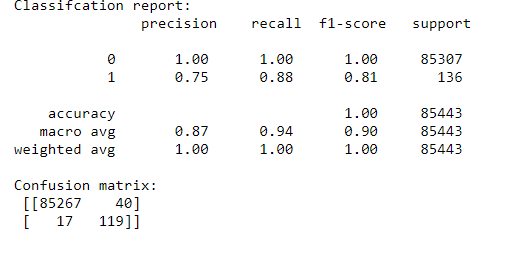
* **Random Over-Sampling:**

We have used the pipeline module from the imblearn library. The pipeline module will help to combine the resampling method with the XGB model. Resampling is done using RandomOverSampler from imblearn.over\_sampling class



* + Over-sampling result interpretation: Over-sampling method has the high precision and accuracy and the recall is at 93%.we are able to capture more Frauds (116 vs 113).
* **SMOTE:**

We have implemented the SMOTE method with the XGB model. SMOTE is imported from imblearn.over\_sampling class.

****

* SMOTE further improves the over-sampling method with 3 more frauds caught in the net and though false positives increased a bit the recall is pretty healthy at 94%.

**5.5. Results**

The results obtained from different Machine Learning models with and without sampling techniques are tabulated below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Sampling method | Accuracy | Precision | Recall | F1-score |
| Random Forest | No sampling | 1.00 | 0.99 | 0.91 | 0.94 |
| Under sampling | 0.98 | 0.53 | 0.94 | 0.55 |
| Over Sampling | 1.00 | 0.97 | 0.90 | 0.93 |
| SMOTE | 1.00 | 0.91 | 0.93 | 0.92 |
| Logistic Regression | No sampling | 1.00 | 0.94 | 0.82 | 0.87 |
| Under sampling | 0.96 | 0.52 | 0.95 | 0.53 |
| Over Sampling | 0.97 | 0.53 | 0.95 | 0.55 |
| SMOTE | 0.97 | 0.53 | 0.95 | 0.54 |
| Decsion Tree | No sampling | 1.00 | 0.94 | 0.82 | 0.87 |
| Under sampling | 0.84 | 0.51 | 0.90 | 0.48 |
| Over Sampling | 1.00 | 0.84 | 0.87 | 0.85 |
| SMOTE | 1.00 | 0.67 | 0.88 | 0.74 |
| Gradient Boosting | No sampling | 1.00 | 0.91 | 0.58 | 0.63 |
| Under sampling | 0.96 | 0.52 | 0.95 | 0.52 |
| Over Sampling | 0.99 | 0.58 | 0.96 | 0.63 |
| SMOTE | 0.99 | 0.56 | 0.96 | 0.60 |
| XGB | No sampling | 1.00 | 0.96 | 0.92 | 0.94 |
| Under sampling | 0.96 | 0.52 | 0.94 | 0.53 |
| Over Sampling | 1.00 | 0.96 | 0.93 | 0.94 |
| SMOTE | 1.00 | 0.87 | 0.94 | 0.90 |

In this our case , more than accuracy we are more concerned about recall because we want to catch as many fraudulent cases as possible. Eventhough Gradient boosting has highest recall it takes lot of time for execution. So we can’t select it as the best model for this problem, because the fraud detection must be taken place in less amount of time. In that sense, Random Forest classifier with SMOTE method obtains better results (better precision, recall, f1-score and accuracy) with less amount of time. So we will choose Random Forest Classifier with SMOTE method for Credit card fraud detection problem.

**6.CONCLUSION**

* 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.
* In our use case of fraud detection, the one metric that is most important is recall. This is because banks/financial institutions are more concerned about catching most of the fraud cases because fraud is expensive and they might lose a lot of money over this.
* Hence, even if there are few false positives i.e flagging of genuine customers as fraud it might not be too unmanageable because this only means blocking some transactions.
* In our study a balanced precision,recall,accuracy and F1-score is obtained with Rnadom Forest Classifier with SMOTE sampling method. The execution time was also less.
* However, blocking too many genuine transactions is also not a feasible solution, hence depending on the risk appetite of the financial institution we can go with either a simple over-sampling method or SMOTE .
* As a future work, we can also tune the parameters of the model, to further enhance the model results using grid search. We can also try Local Outlier Fraction and Isolation Forest algorithms for anomaly detection.

**REFERENCES**

[1] “Credit Card Fraud Detection Based on Transaction Behaviour -by John Richard D. Kho, Larry A. Vea” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017

[2] CLIFTON PHUA1, VINCENT LEE1, KATE SMITH1 & ROSS GAYLER2 “A Comprehensive Survey of Data Mining-based Fraud Detection Research” published by School of Business Systems, Faculty of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800, Australia

[3] “Survey Paper on Credit Card Fraud Detection by Suman”, Research Scholar, GJUS&T Hisar HCE, Sonipat published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014

[4] “Research on Credit Card Fraud Detection Model Based on Distance Sum – by Wen-Fang YU and Na Wang” published by 2009 International Joint Conference on Artificial Intelligence

[5] “Credit Card Fraud Detection through Par enclitic Network Analysis By Massimiliano Zanin, Miguel Romance, Regino Criado, and Santiago Moral” published by Hindawi Complexity Volume 2018, Article ID 5764370, 9 pages

[6] “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy” published by IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 29, NO. 8, AUGUST 2018

[7] “Credit Card Fraud Detection-by Ishu Trivedi, Monika, Mrigya, Mridushi” published by International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016 [8] David J. Wetson, David J.Hand,M Adams, Whitrow and Piotr Jusczak “Plastic Card Fraud Detection using Peer Group Analysis” Springer, Issue 2008.