|  |
| --- |
| Credit Card Fraud Detection With Classification Algorithms In Python |
| Credit Card Fraud Detection using Machine Learning  Team members |
| |  |  |  | | --- | --- | --- | |  |  |  | |

Table of Contents

**[Executive Summary](#_Toc101288608)** [2](#_Toc101288608)

[**Project motivation/background** 2](#_Toc101288609)

[**Motivation** 4](#_Toc101288610)

[**Problem statement** 4](#_Toc101288611)

[**Research Objectives** 5](#_Toc101288612)

[**Proposed System** 6](#_Toc101288613)

[**Data preparation activities** 6](#_Toc101288614)

[**Dataset Description** 7](#_Toc101288615)

[**Data pre-processing** 8](#_Toc101288616)

[**Handling imbalanced data** 9](#_Toc101288617)

[**Data Analytics** 10](#_Toc101288618)

[**Techniques used** 12](#_Toc101288619)

[**1.** **Random Forest** 12](#_Toc101288620)

[**Working of Random Forest** 12](#_Toc101288621)

[**2.** **XGBoost** 13](#_Toc101288622)

[**Working of XGBoost** 13](#_Toc101288623)

[**3.** **LightGBM** 14](#_Toc101288624)

[**Working of LightGBM** 14](#_Toc101288625)

[**4.** **SMOTE** 15](#_Toc101288626)

[**Evaluation Parameters** 15](#_Toc101288627)

[**Steps of Methodology** 16](#_Toc101288628)

[**Findings** 26](#_Toc101288629)

[**Comparing Results** 27](#_Toc101288630)

[**Comparing Confusion Matrix** 29](#_Toc101288631)

[**Comparing Features Importance** 30](#_Toc101288632)

[**Features Correlation** 32](#_Toc101288633)

[**Business Implications/Intelligence** 34](#_Toc101288634)

[**Conclusion** 35](#_Toc101288635)

[**References** 35](#_Toc101288636)

# **Executive Summary**

Businesses are constantly drifting toward joining their administrations to the online environment as the extensive use of the Internet increases. The development of internet based. e-Commerce sites on the other hand has led to the situation where most individuals and companies rely upon online administrations for all their monetary transactions today. There has been an exponential increase in the use of internet banking together with an increase in the variety of online transactions, as a result of which there has been an exponential increase in credit card fraud as well.

There are several mechanisms for protecting credit card transactions, including methods of encrypting and tokenizing credit card data. Even though such approaches are effective in the majority of cases, they do not provide full protection against credit card fraud in all cases. Artificial Intelligence (AI) is a field that examines the ways in which computers can learn based on their previous experiences with data (in other words to become smarter and better at predicting events without being explicitly programmed to do so).

Researchers are currently pursuing the most optimal approach to detecting such frauds using data science. It is important to know that credit cards contain sensitive data, and they can be used to commit fraud affecting not only the holder of the card, but also banks, governments, and all types of financial sectors which leads to high losses on the financial front. Various Machine Learning algorithms are available that can be used for detecting fraud transactions so as to overcome such losses and to prevent such a scenario from occurring in the future. The methods such as Logistic Regression, Random Forests, Naive Bayes, K- Nearest Neighbour, and the Neural Networks are all techniques that have been used as part of the detection of fraudulent transactions. The objective of this study is to determine what algorithm model is most efficient among all the ones studied and to come up with an optimal solution among all of them.

# **Project motivation/background**

Since the world is moving towards a cashless society, there is going to be a consistent increase in transactions that are done online. There is no requirement for fraudsters to physically be present at the crime scene to commit fraud. There are many ways to hide their identities so that they can carry out these diabolical acts from the comfort of their own homes. There are a number of identity obscuring techniques which include the use of VPNs, to route traffic through the Tor network, etc. These techniques are difficult to trace back, and are considered as anonymous as possible. Internet financial losses can have a profound effect on individuals. Fraudsters, after stealing card details, are either able to use the cards for their own purchases or sell them on to other people, as is currently happening in India, where around 70 million credit and debit card details are being sold on the dark web (Tanouz et al., 2021). An incident in the UK involving credit card fraud that was one of the most serious in recent memory resulted in a total loss of GBP 17 million that was related to such an incident. An international fraudster group was involved in an incident that took place in the mid-2000s when they stole the details of more than 32,000 credit cards (Tiwari et al., 2021). Several researchers refer to this incident as the biggest credit card fraud ever perpetrated. As a result, millions of dollars have been lost through credit card fraud due to the ineffective security systems in place. During the cardholder's use of their card, as well as during card issuer's processing of the transaction, all transactions are guaranteed to be benign. However, fraudsters are actually trying to deceive and manipulate financial institution employees into believing that fraudulent transactions that they are conducting are legitimate.

Moreover, there is a certain amount of fraud which takes place constantly, obtaining financial gain without both the knowledge of the card issuer as well as the cardholder. The dark side of online transactions is that many times both the cardholder and the authorized institution are unaware that the transaction was fraudulent, and that is the most dangerous aspect of online transactions. As a consequence, it is very challenging to detect fraud in an environment which is flooded with thousands of legitimate transactions, particularly when the number of fraudulent transactions is significantly smaller than that of legitimate transactions. Fraud detection technologies have been deployed in several financial industries in order to fight fraud more effectively, including predictive analytics, data mining, and modelling algorithms that employ clustering (Sudeep Srivastava, 2022). There is an issue with all of these techniques where they are not effective without the support of ML algorithms, regardless whether they are supervised approaches or unsupervised approaches, since they will be able to distinguish between credit cards and other fraudulent transactions. It is worth noting, however, that even when using machine learning algorithms, as a robust method of detecting all fraudulent activity, they face an undeniable number of challenges (Alfaiz & Fati, 2022). It is essential that the widely used evaluation metrics for machine learning be at their highest level in an ideal model. There are many improvements that need to be made in this arena in order to get us closer to this ideal model. Credit card fraud detection is a difficult task that requires a variety of techniques, for example machine learning techniques, resampling procedures and cross-validation processes among others. If these factors are taken into consideration, then the performance of the model will be enhanced so that it can be validated by evaluation metrics.

## **Motivation**

In recent years, technology has experienced an unprecedented transformation in the finance industry, which has led to a major transformation of the industry. Probably the most visible transformation you can see has to do with the way we look at new payment transactions nowadays. In the last few years, digital payments have seen a phenomenal increase in the market.

Digital payment transactions are projected to total USD 4,934,741 million by 2020, according to Statista. Reports from the same company reveal that by the year 2024, Mobile Point of Sale payments will account for a total of 1800.4 million users (*• Value of Payment Card Fraud Worldwide 2027 | Statista*, n.d.).

The fact that digital payments have now become acceptable has led to a growing number of companies competing for this segment of the market and doing their best to make payments more user-friendly & customer-oriented.

With the rapid growth of electronic payments, there has also been an increase in electronic fraud. Dealing with fraud in the banking and commerce industries has been very challenging. Fraudsters have become adept at finding loopholes.

It has therefore become increasingly important for companies to manage vulnerability effectively and close security loopholes by detecting fraudulent activity through machine learning and predictive analytics. VynZ Research identified fraud detection and prevention as a significantly large market, anticipated to reach a value of USD 85.3 billion over the next five years, growing at a compound annual growth rate of 17.8%.

## **Problem statement**

The performance of this fraud detection system is complicated by a few issues that are difficult to resolve for researchers. Among one of the most challenging issues is the absence of high-quality literature in the field that provides test results and authentic data. Frequently, the reason for this is that the sensitive financial information associated with the fraud needs to be used in a confidential manner in order to protect the customer's privacy and security. We present a list of some of the properties a fraud detection system needs to have in order to deliver results that are meaningful:

* The fraud detection system ought to be able to cope with slanted distributions since only a small percentage of all credit card transactions are fraudulent.
* The system must be prepared to deal with the noise in a proper manner. A main cause of noise is the fact that the information has errors in it, such as wrong dates, incorrect figures, etc. There is a significant amount of noise present in the actual information that reduces the amount of generalization that may be achieved, regardless of how extensive the training set may be.
* Also, there is an issue with overlapping information, which makes this a challenging area of study. Occasionally, a transaction might be classified as a fallacious transaction as opposed to an actual transaction or vice versa.
* The system should be prepared to adapt to new forms of fraud as they arise. Expert fraudsters aim to carry out their work as fast as possible and to use as many methods as are available to them since they are aware that new fraud methods will become ineffective once they become standard practices in the future.
* The classifier system needs to be evaluated properly by the means of proper measurements. In order to evaluate the overall precision, the dispersion of the measurements needs to have a slanted distribution, since even with high levels of precision, most false transactions can be incorrectly classified as legitimate.

## **Research Objectives**

The goal of the proposed method is to detect fraudulent credit card purchases using supervised machine learning algorithms. With the purpose of detecting fraud transactions, we came up with a technique whereby we extract the features of a dataset, using the extracted features to train the models for the sole purpose of detecting fraud transactions. The following objectives will be met by the proposed solution:

* To determine the level of fraud activity associated with credit cards with high efficiency and accuracy.
* To train a high-performance model on a given dataset using both label and feature information.
* To determine whether a transaction is fraud or not using the trained model.

## **Proposed System**

The data flow into a system model is defined as the system architecture. During the pre-processing phase, steps are taken to convert the raw data into useful information. On the basis of the characteristics extracted from the data, an algorithm is then selected that is used to classify a specific transaction as legal or illegal based on the features it is derived from.

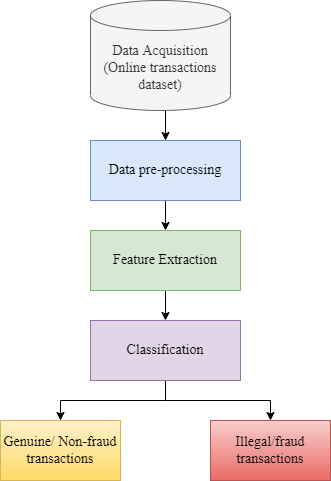


Figure 1: The data flow diagram of proposed system.

# **Data preparation activities**

It is always impossible to make a comparison between a card transaction and a previous transaction that the customer has made. In real-world situations, this sort of unfamiliarity - concept drift problems - are a very difficult problem to solve. The concept drift can be described as a variable that changes over time and in unexpected ways. As a result, significant imbalances in the data occur.

## **Dataset Description**

A dataset which was used in this study was the Credit Card Fraud Detection dataset which is available on Kaggle (*Credit Card Fraud Detection | Kaggle*, n.d.-a), and that could be downloaded from there. It is intended to serve as a reference dataset for the analysis and interpretation of big data mining and fraud detection research conducted by the Worldline Group and the Machine Learning Group (<http://mlg.ulb.ac.be>) of ULB (Université Libre de Bruxelles).

This dataset consists of data from credit card transactions made by European cardholders for the month of September 2013. Within two days, they have 492 fraudulent transactions out of 284,807 transactions within this dataset. A substantial portion of the dataset's transactions fall into the positive class (frauds), accounting for 0.172% of all transactions.

There are only numerical variables that are the results of a PCA transformation that are included in this analysis. Among the features V1, V2, …., V28, we can see they are the principal components resulting from PCA, however, 'Time' and 'Amount' have not been transformed by PCA. A feature called "Time" represents the number of seconds that passed between each transaction and the first transaction found in the dataset. With 'Amount', one can show the amount a transaction has, this feature can even be used for instance-dependant and cost-sensitive learning. In the 'Class' feature, in the case of fraud, it takes the value of 1 while in the case of non-fraud, it takes the value of 0.

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Feature** | **Description** |
| 1. | Time | The time in seconds between the current and the first transaction. |
| 2. | Amount | Amount of the transaction |
| 3. | Class | 1. Class label represents not fraud 2. Class label represents fraud |

Table 1: Description of attributes of dataset.

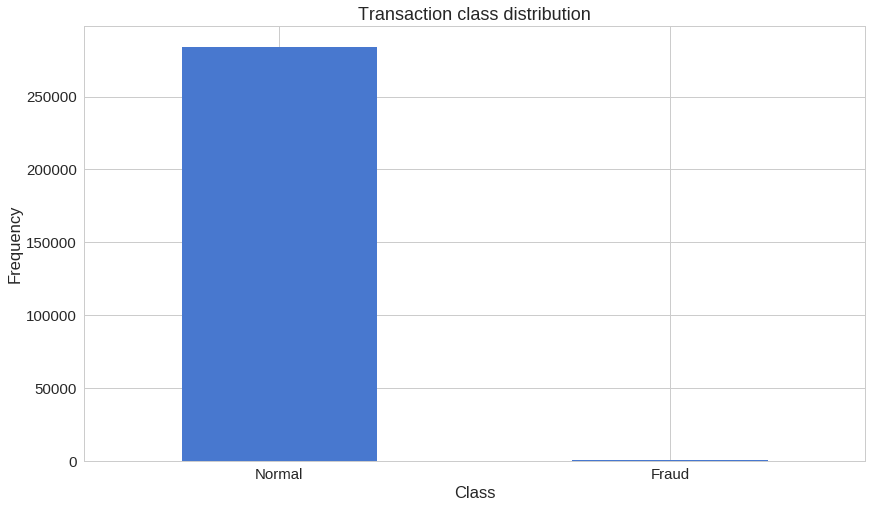


Figure 2: Distribution of samples between two classes.

## **Data pre-processing**

Pre-processing is not necessary for every dataset if certain criteria are met. As part of these criteria there is the requirement to determine if there are any missing values that might cause the prediction values to be altered. Based on our analysis of the data, we can see that the dataset contains 284,807 values for each feature, which means that no value is missing from any of these features. Pre-processing of the data is therefore not required. The correlation matrix is shown in Figure 3 along with a heat map. It has been shown that correlation matrix is an effective technique for assisting us in determining whether any particular part of the report could be removed in order to ensure the accuracy of the proposed system. Due to the fact that the correlation matrix has shown that all features are associated with the 'Class' feature irrespective of whether there is a strong correlation or not, on the basis of this, one may conclude that it is not necessary to pull out any particular attribute when analyzing the correlation matrix; as a result, there is no requirement to perform any pre-processing. There is another reason for the reduction in dimension of the features of “V1” to “V28” as a result of Principal Component Analysis (PCA) transformation. As a result of this process, sensitive , the original data for these features was preserved. The dataset is, however, already processed, so we don't have to process it again, this decision is made deliberately made to avoid the pre-processing stage in order to obtain an approach that is more realistic.

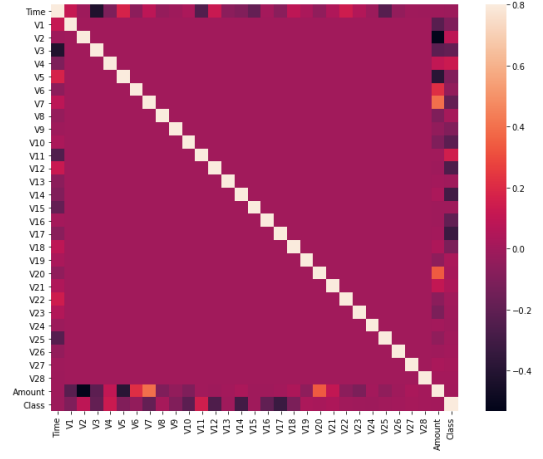


Figure 3: Attribute correlation matrix (the X and Y axis show the different attributes within the dataset).

The correlation matrix of the dataset is shown in Figure 3. In this matrix, it is shown that attribute class is independent of both the transaction amount and the time of the transaction. In fact, it is evident even from the matrices above that the class of the transaction is determined by the PCA applied attributes.

## **Handling imbalanced data**

Developing a predictive model for a class imbalanced dataset entails developing a model based on the data that contains historical classification datasets.

When working with balanced datasets, the challenge arises from the fact that most machine learning algorithms don't give much attention to, or in turn give poor results for, the minority class, even though in general it is the performance of the minority class that counts the most.

The oversampling of minority groups is an approach that can be used to overcome imbalanced datasets. It is possible to duplicate examples in the minority class by using the simple approach. However, these examples don't contribute any additional information to the model. In fact, new examples can be created by synthesizing what is already known. In this context, there is the Synthetic Minority Oversampling Technique, which is a method of augmentation for a minority population and is sometimes referred to as SMOTE for short (SWASTIK SATPATHY, 2020).

The SMOTE algorithm works by picking examples near one another in the feature space, setting up a line to connect them in the feature space, and creating a new sample in the feature space at the point where that line meets the previous sample.

In particular, a randomly selected example from a class of minority members is first chosen. After that, k of the closest neighbors of that example are found (normally, k is 5). There is a choice of a neighbor randomly selected and a synthetic example is produced by selecting a random point in the feature space between the two examples.

There was a great deal of imbalance in the credit card data. The original data set contained 99.83% of the real-world transactions as non-fraudulent, compared to 0.17% that were fraudulent. Using the original data set in case of a fraud detection algorithm would not be a wise decision for a very simple reason: in light of the fact that more than 99% of all transactions are not fraudulent, an algorithm that consistently indicates non-fraudulent transactions would be more accurate than 99%. However, that is exactly the opposite of what we are trying to achieve.  It is not our goal to achieve 99% accuracy by never representing a transaction as fraudulent but, rather, we want to detect a transaction as fraudulent and label it in that manner. For the creation of our balanced training data set, we will use SMOTE algorithm. We will compare the results of our chosen machine learning algorithm before and after using SMOTE algorithm.

# **Data Analytics**

The proposed workflow of system is shown below:

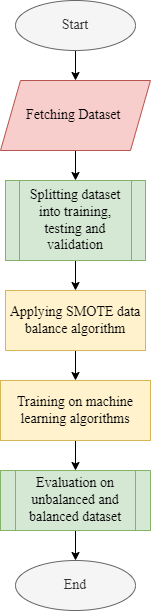


Figure 4: Steps of proposed system.

There are five steps that make up the proposed system. The start of the process involves retrieving data, in this case data was downloaded from Kaggle’s credit card fraud detection dataset. Data samples are then divided up between training and testing. In other words, 80 percent of the data will be used for training purposes, and 20 percent for testing purposes. After taking the data balance algorithm, SMOTE, which applies a balance approach to the training dataset, a variety of machine learning algorithms are then trained on the dataset. Finally, evaluation measures are used to evaluate the machine learning algorithms.

## **Techniques used**

The techniques we selected for the proposed approach are Random Forest Xboost and LightGBM.

After experiencing a multitude of Classification Algorithms, we had obtained the knowledge that the best algorithms in terms of efficiency and accuracy in classification algorithms these days would be Random Forest and XG Boost, so we decided to utilize them both together for and also LightGBM for designing our model and compare their results.

## **Random Forest**

A random forest algorithm is one of the most common types of supervised machine learning algorithms based on ensemble learning. It is a type of learning in which there is a combination of different algorithms that helps form a model that gives more accurate predictions by repeating the algorithm multiple times. In a random forest algorithm, multiple algorithms of the same type are combined, forming a forest of decision trees, thus the name "Random Forest". A random forest algorithm is useful both for regression as well as classification problems (Sruthi E R, 2021).

**Working of Random Forest**

Performing the random forest algorithm involves a number of basic steps:

1. Select N records at random from the dataset.
2. From this selection, create a decision tree depicting the probability of this outcome.
3. We can choose how many trees you want to include in your algorithm by repeating steps.
4. Each tree in the forest predicts what category the new record belongs to.
5. It is then decided which category will be assigned the new record, based upon the votes of the majority.

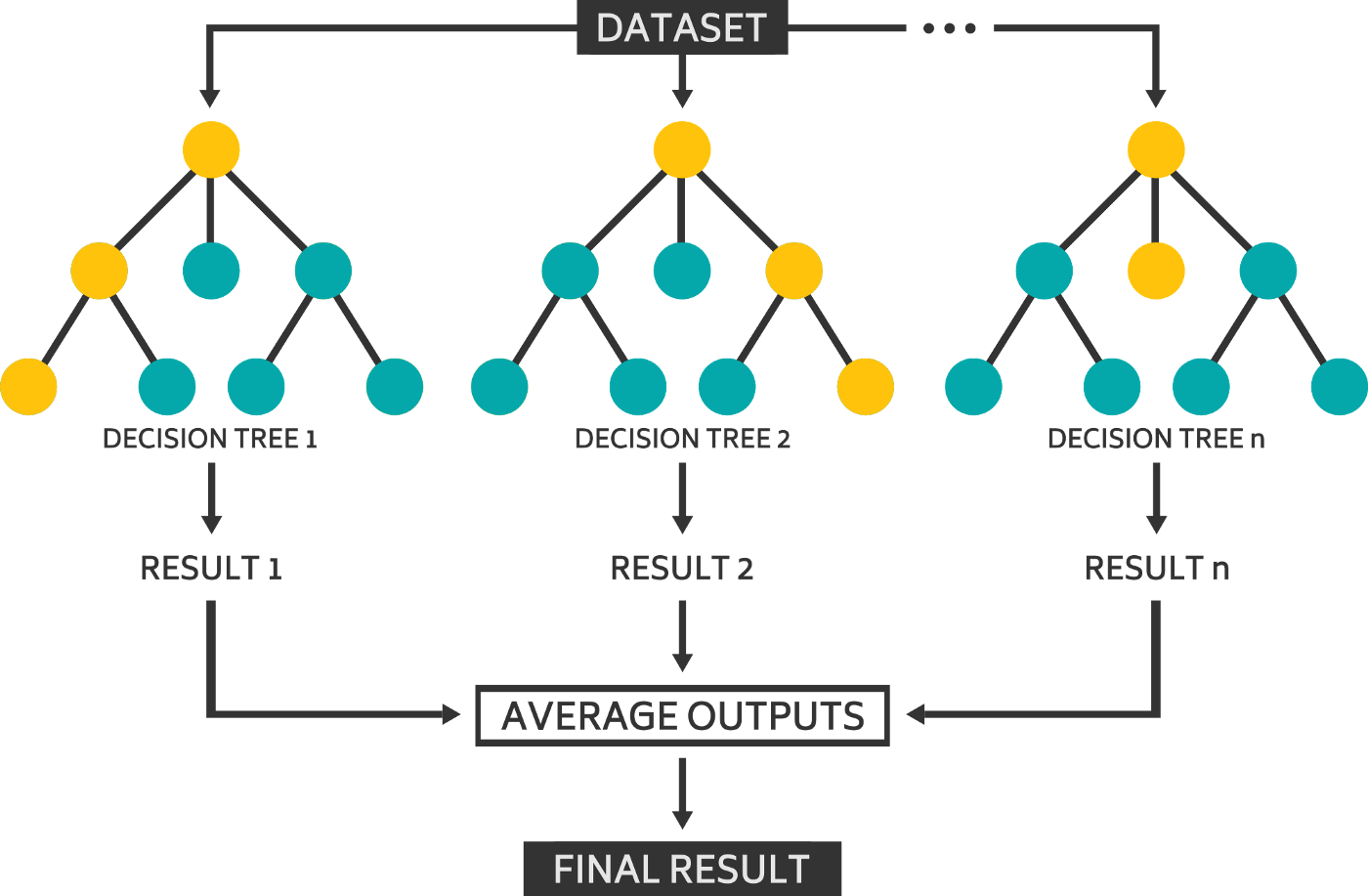


Figure 5: Workflow diagram of random forest algorithm.

## **XGBoost**

There is also a classifier called XGBoost, which is a machine learning ensemble classifier based on decision tree. The XGBoost framework is based on a classification framework constructed by grouping classification trees with regression tree (CART).   There has been extensive use of XGBoost to develop state-of-the-art algorithms when it comes to solving data problems (e.g., Kaggle competitions), as it is an extremely efficient and scalable machine learning algorithm meant to improve tree growth (Ramraj et al., 2016).

**Working of XGBoost**

1. The weak learners in gradient boosting are regression trees, where each of the leaves of each regression tree is a continuous score, mapped to a point in the input data.
2. A regularized (L1 and L2) objective function minimized by XGBoost is a combination of a convex loss function (based on the difference between the predicted and actual output, and vice versa) as well as a penalty term for the complexity of the model (the regression tree functions, etc.).
3. Iteratively, the tree prediction is made by adding new trees that predict the errors or residuals of the previous trees, and these predictions are then merged together to generate the final prediction.
4. In this way, gradient boosting minimizes any loss associated with adding a new model by using a technique known as gradient descent.

## **LightGBM**

LightGBM is an efficient and high-performance technique based on a decision tree algorithm for gradient boosting which can be utilized for a variety of purposes, including ranking, classification, and many other machine learning tasks(Pranjalk, 2017).

**Working of LightGBM**

1. With LightGBM, trees are split level-wise rather than the other boosting algorithms which grow trees level-wise.
2. LightGBM grows only the leaves that produce the biggest delta loss. The leaf-wise algorithm has the advantage of having a lower loss than level-wise algorithm since the leaf is fixed.

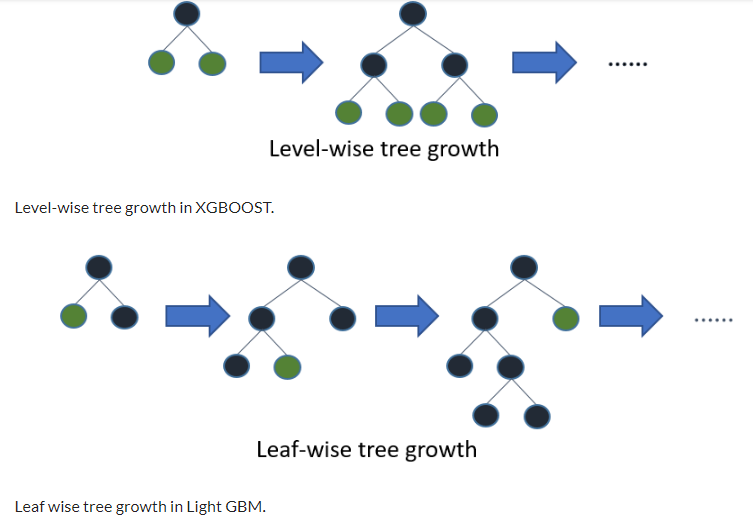


Figure 6: Difference between XGBoost and LightGBM.

## **SMOTE**

SMOTE is referred to as synthetic minority oversampling technique and is a method of oversampling to address the problem of class unbalance by increasing the number of minority samples analyzed. The new minorities data points are created by placing the new data points between the real minorities data points. In this regard, these data points help to shift the classifier's bias towards the generated data points.

## **Evaluation Parameters**

Before using evaluation parameters (Nicholson, 2019), we must understand following concepts.

**True Positives (TP)** - These are the values that are correctly predicted positives. That is, if the value of the actual value of the class is found to be true, then the predicted value must also be true.

**True Negatives (TN)** - These are negative values which have been correctly predicted, which in turn means that the value of the actual class, and the value of the predicted class, are also negative.

**False Positives (FP)** – These are the cases when a class is actually no, but a class is predicted to be yes.

**False Negatives (FN)** – This occurs when the class of the actual is yes, but the predicted class is no.

The following parameters will be used for evaluating the results of machine learning algorithms.

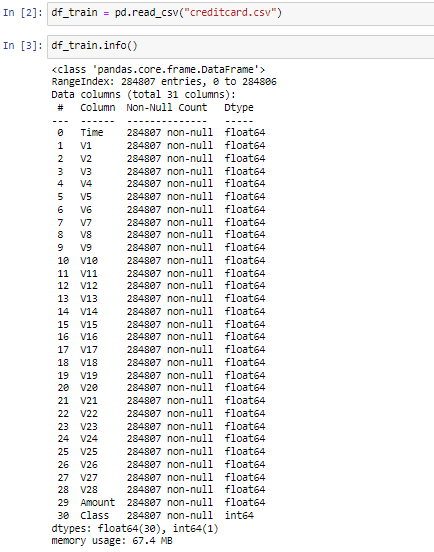
1. **Accuracy** - Accuracy can be viewed as a performance measure due to its intuitive nature. It is calculated as a ratio of the correctly predicted observations compared to the total observations. It can be calculated as:
2. **Precision** - When we look at the precision we measure the ratio between the number of correctly predicted positive results and the total number of predicted positive results.  The higher the precision, the lower the false positive rate. It can be calculated as:
3. **Recall**(Sensitivity) - The recall of a prediction is the ratio of positive responses correctly predicted to all responses in the class. It can be calculated as:
4. **F1 score** - F1 score is the sum of Precision and Recall as a weighted average. Because of this, this score both considers false positives as well as false negatives. It can be calculated as:
5. **ROC -** The Receiver Operating Characteristics Curve, or ROC curve for short, is a metric that measures the effectiveness of a model for classifying data. By plotting the ROC curve, we can visualize the ratio of the true positive rate to the false positive rate, which allows us to evaluate the reliability of the classifier model.
6. **Matthews correlation coefficient** **(MCC)** is used in assessment of the model's performance to determine whether it performs well. It can be calculated as:

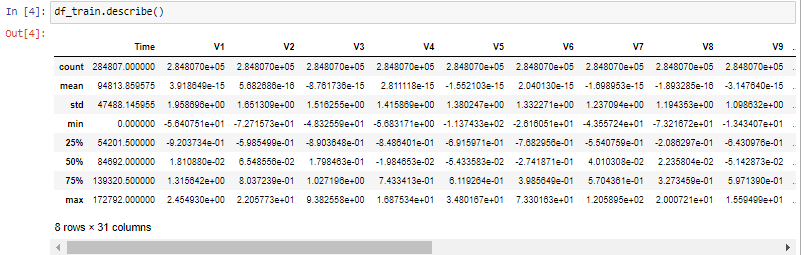
## **Steps of Methodology**

In this section, we will describe the steps involved in running the machine learning algorithms on chosen dataset. First, we will see results of machine learning algorithms on unbalanced dataset. Then we will see results of machine learning algorithm after applying SMOTE algorithm to balance the dataset.

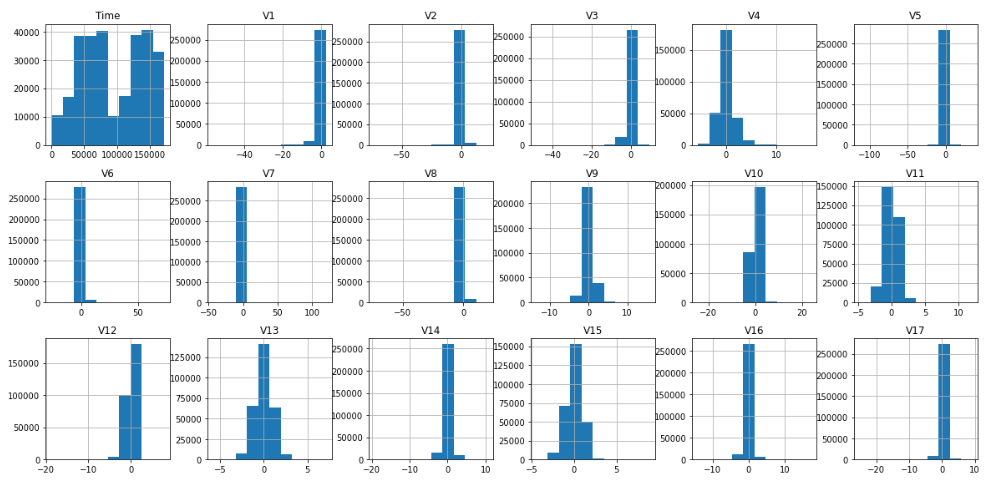
1. ***Read Dataset***

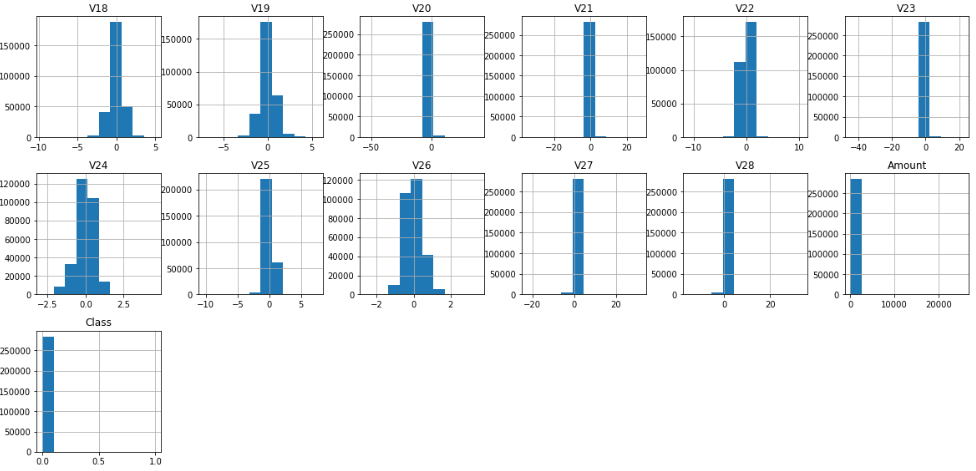
The dataset we are going to be importing will be the 'CSV file' in the form of a Pandas DataFrame. Further, we will be examining our dataset in depth to gain a greater understanding of the type, quantity, and distribution of data within our dataset. This can be achieved by using the built-in describe feature of Pandas, as shown below:





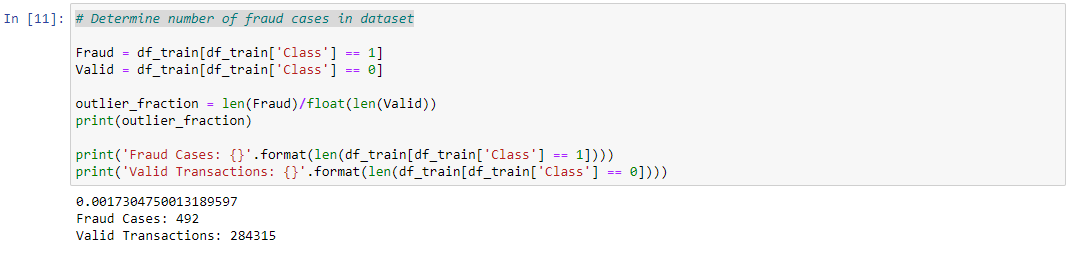
1. ***Plotting the histograms of each parameters***



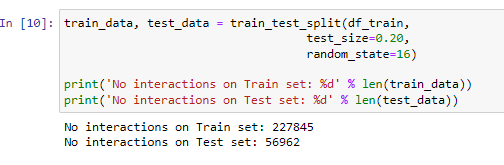


The PCA Dimensionality reduction has been applied to the V1 through V28 versions as a way of protecting sensitive information and user identities.

1. ***Determine number of fraud cases in dataset***



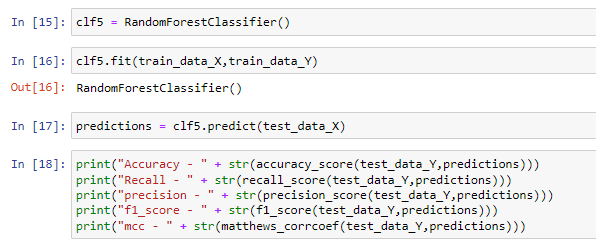
1. ***Splitting data into training and testing***



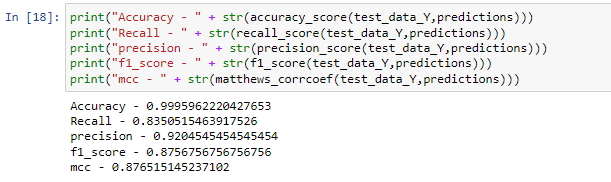
**Random Forest Algorithm**

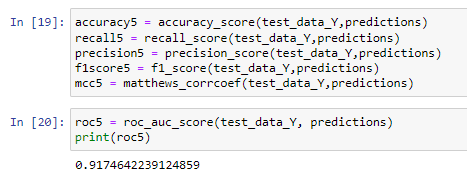
1. ***Applying random forest classifier on unbalanced dataset***

Here we apply random forest classifier without applying any balanced technique on dataset.



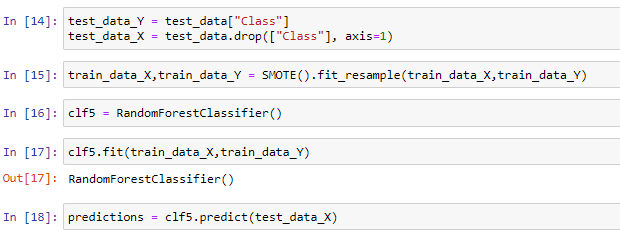
***Output***



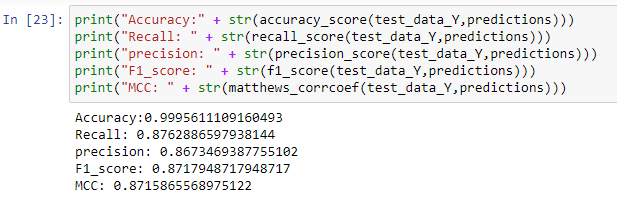


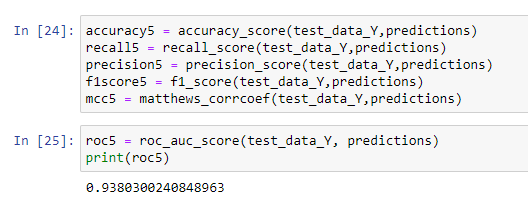
1. ***Applying random forest classifier on balanced dataset***

Here we apply random forest classifier by applying smote balancing technique on dataset.



***Output***





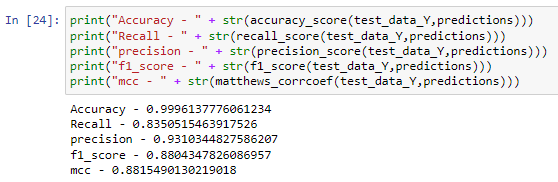
**XGBoost Algorithm**

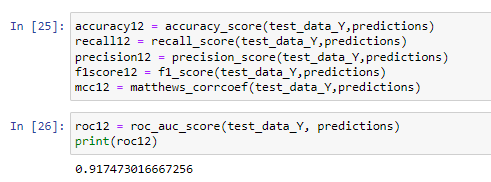
1. ***Applying XGBoost classifier on unbalanced dataset***

Here we apply XGBoost classifier without applying any balanced technique on dataset.



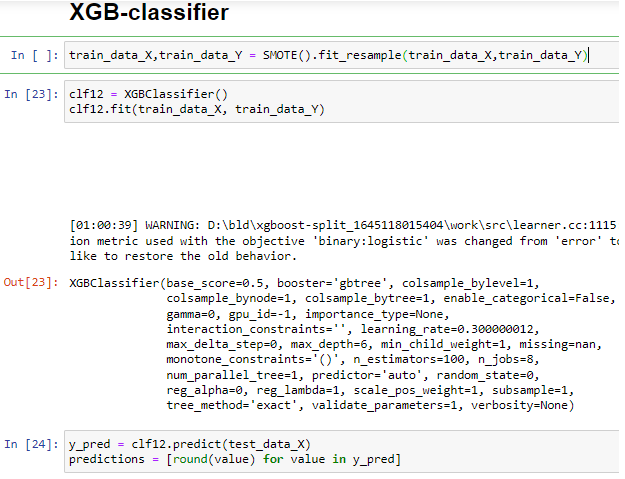
***Output***



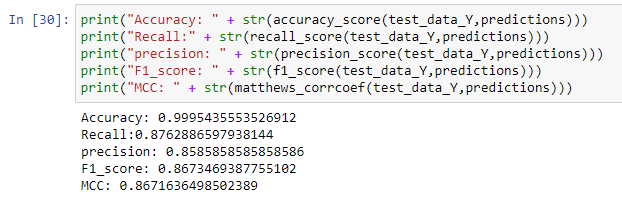


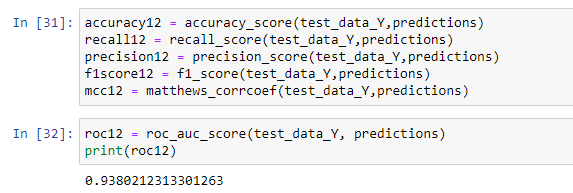
1. ***Applying XGBoost classifier on balanced dataset***

Here we apply XGBoost classifier by applying smote balancing technique on dataset.



***Output***

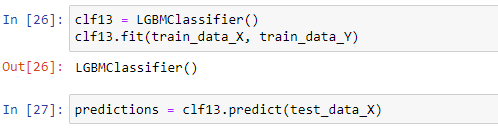




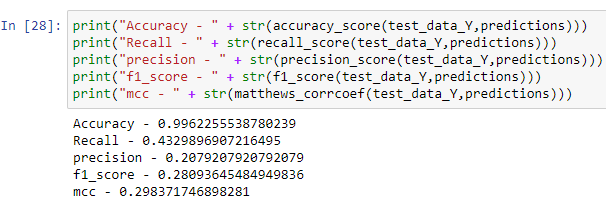
**LightGBM Algorithm**

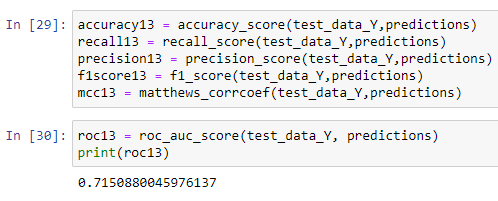
1. ***Applying LightGBM classifier on unbalanced dataset***

Here we apply LightGBM classifier without applying any balanced technique on dataset.



***Output***

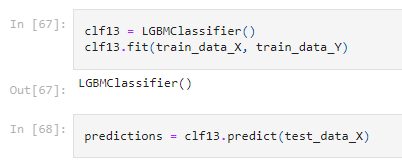




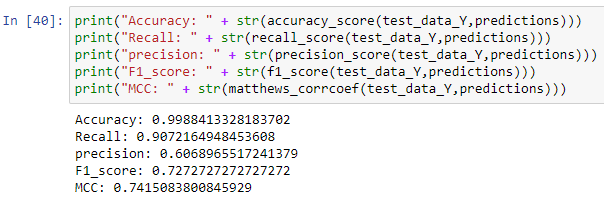
1. ***Applying LightGBM classifier on balanced dataset***

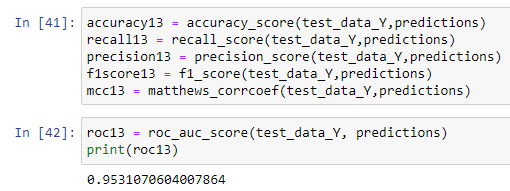
Here we apply LightGBM classifier by applying smote balancing technique on dataset.





***Output***





# **Findings**

With regard to the application designed to detect the fraudulent use of a credit card, an example of a numerical feature might include: the amount of the transaction being analyzed. In this case, the target variable will be y: which is a feature for which we would like to know its value based on the set of characteristics x. y can be categorical or continuous. With the problem of fraud that we are currently dealing with, a valuable variable that we prefer to use as a metric is the ground truth about the nature of the transaction: is transaction a fraudulent or not? The purpose of a machine learning algorithm is to model the relationship between a feature set X and a target variable Y.

The results of our observations can be found below. The results of this study are based on a dataset with 20% of the data used for testing, with a random state being 16.

## **Comparing Results**

***Results of ML algorithms for test cases on unbalanced dataset.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Recall** | **Precision** | **F1 Score** | **ROC** | **MCC** |
| **Random Forest** | 0.99959 | 0.83505 | 0.92045 | 0.87567 | 0.91746 | 0.87651 |
| **XGBoost** | 0.99961 | 0.83505 | 0.93103 | 0.88043 | 0.91747 | 0.88154 |
| **LightGBM** | 0.99590 | 0.43298 | 0.20792 | 0.28093 | 0.71508 | 0.29837 |

As shown in the above table, we have summarized both machine learning algorithms without considering any data balancing algorithms. Based on the results of this study, it is clear that Random forest and XGBoost are found to be more accurate in comparison with LightGBM. The XGBoost and Random Forest algorithms are found to be the best performing in terms of MCC.

***Results of ML algorithms for test cases on balanced dataset.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Recall** | **Precision** | **F1 Score** | **ROC** | **MCC** |
| **Random Forest** | 0.99959 | 0.87628 | 0.86734 | 0.87179 | 0.93803 | 0.87158 |
| **XGBoost** | 0.99954 | 0.87628 | 0.85858 | 0.86734 | 0.93802 | 0.86716 |
| **LightGBM** | 0.99884 | 0.90721 | 0.60869 | 0.72727 | 0.95310 | 0.74150 |

The above table gives a summary of the results for both machine learning algorithms which utilizes SMOTE as a data balancing algorithm. We found that recall of Random Forest increases along with LightGBM and XGBoost. The recall of XGBoost also increases but Random Forest performs best when it comes to F1, ROC and MCC. On other hand, the performance of LightGBM increases but is low as compared to other two algorithms.

Figure 7: Performance of ML algorithms before applying balancing algorithm on dataset.

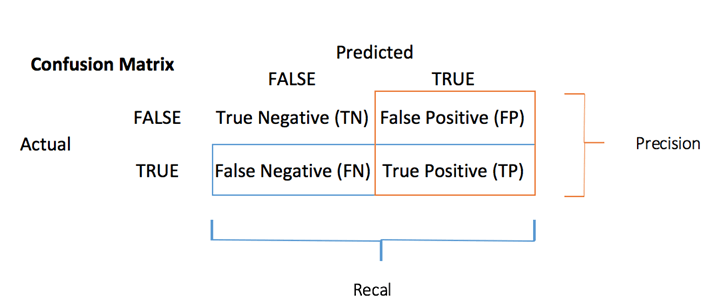
Figure 8: Performance of ML algorithms after applying balancing algorithm on dataset.

If we look at above graphs, it becomes clearer that both Random Forest and XGBoost algorithms have slight effect on evaluation parameters even after applying smote technique. But on other hand, the precision and MCC in case of LightGBM decreases. From this analysis, we can say that Random Forest and XGBoost are good choice for the chosen dataset of credit card fraud.

## **Comparing Confusion Matrix**

It is a technique used to measure the accuracy of machine learning classifications. The goal of this kind of table is to get a clearer picture of how the classification model performs when it is applied to a set of data where the true value is known.

Confusion matrix is represented as follows:



|  |  |
| --- | --- |
| ***Confusion matrix of*** ***Random Forest without balancing algorithm*** | ***Confusion matrix of*** ***Random Forest with balancing algorithm*** |
|  |  |

Figure 9: Evaluation of RF algorithm before and after balancing dataset.

|  |  |
| --- | --- |
| ***Confusion matrix of*** ***XGBoost without balancing algorithm*** | ***Confusion matrix of*** ***XGBoost with balancing algorithm*** |
|  |  |

Figure 10: Evaluation of XGBoost algorithm before and after balancing dataset.

|  |  |
| --- | --- |
| ***Confusion matrix of*** ***LightGBM without balancing algorithm*** | ***Confusion matrix of LightGBM with balancing algorithm*** |
|  |  |

Figure 11: Evaluation of LightGBM algorithm before and after balancing dataset.

## **Comparing Features Importance**

In this section, we will see the importance and dominance of features for unbalanced dataset and after using smote algorithm.

|  |  |
| --- | --- |
| ***Random Forest without balancing algorithm*** | ***Random Forest with balancing algorithm*** |
|  |  |

Figure 12: Features importance in case of RF algorithm before and after balancing dataset.

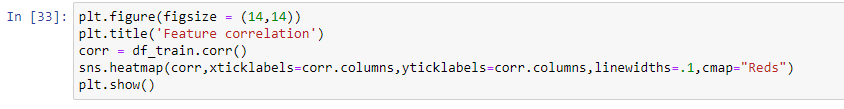
|  |  |  |
| --- | --- | --- |
| ***XGBoost without balancing algorithm*** | ***XGBoost with balancing algorithm*** | |
|  |  | |
|  | |  | |

Figure 13:Features importance in case of LightGBM algorithm before and after balancing dataset.

From graphs, we can see that ‘V17’ is dominant in cases of both Random Forest and XGBoost when no balancing algorithm is used. After applying balancing technique that is smote, ‘V14’ becomes more dominant in all three cases.

## **Features Correlation**

We can find the dependency of features as follows:



***Output***

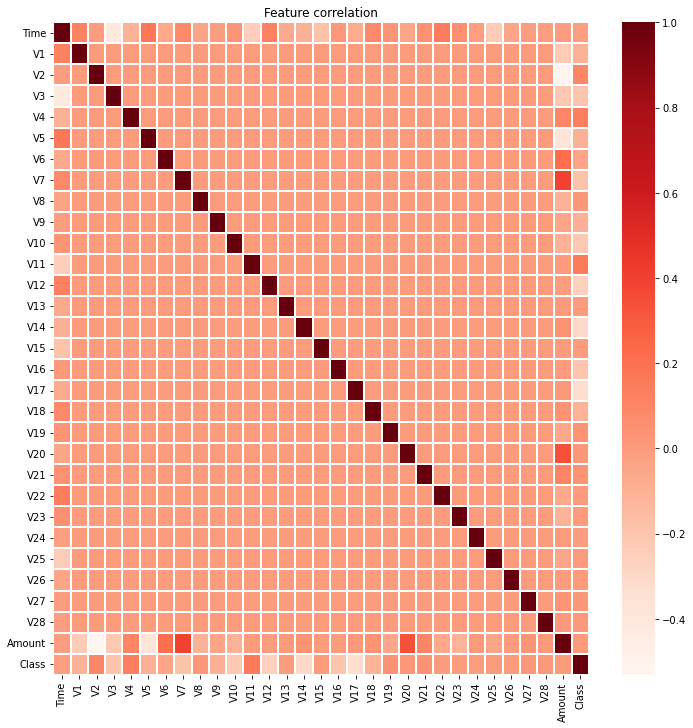


Figure 14: Correlation between features.

In correlation, the growth of both values is positive when they increase together, and the growth of one value decreases as the growth of the other increases

The correlation can have the following values:

* In this particular example, 1 reflects a perfect positive correlation (between amount and V7, amount and V20.)
* 0 reflects no correlation (between features V1-V28)
* -1 reflects perfect negative correlation (like between 'Time' and V3, then 'Amount' and V2, then between 'Amount' and V5).

# **Business Implications/Intelligence**

One of the top business objectives for most banks is retaining highly profitable customers. In spite of this, however, banking fraud is becoming an increasing threat to the achievement of this goal. It is a concern on the part of both banks and their customers, both because substantial financial losses could be incurred, as well as a breach of trust and credibility.

The Nilson Report estimates that by 2020, there will be $30 billion in frauds committed against banks around the world. The number of fraudulent transactions is also increasing with the emergence of various digital channels for payments, in addition to the rise in the number of digital payment gateways (*Credit Card Fraud Detection | Kaggle*, n.d.-b).

It is estimated that fraudulent credit card transactions cause billions of dollars in losses every year. For fraud losses to be reduced, it is vital to develop effective fraud detection algorithms, which increasingly employ automated machine learning techniques as they assist fraud investigators in their investigations. However, it is quite challenging to come up with fraud detection algorithms that are designed considering the non-stationary distribution of the data, the highly imbalanced distribution of the classes and the continuous stream of transactions. In the meantime, public data are scarce due to government policy regarding confidentiality, thereby leaving a multitude of questions unanswered about the best way to approach this problem.

It is no longer a trend but a necessity to monitor and prevent credit card fraud using machine learning systems in the banking industry. It is no longer a trend but a necessity according to experts. The use of machine learning is helping these institutions decrease the amount of time and effort spent on manual reviews, the number of chargebacks and fees assessed, as well as the number of transactions denied.

In order to make their fraud protection systems more effective, financial institutions have to make use of ML-powered systems that reduce substantially the risk of missing suspect transactions, the likelihood of human errors, and the incidence of security breaches. Algorithms based on machine learning have the potential to handle enormous volumes of data in order to prevent fraud.

By using machine learning, companies not only save time and energy that would otherwise be used on traditional predictive analytics, but they also allow them to stay safe from fraud.

# **Conclusion**

As the use of credit card fraud as a way to commit fraud is increasing at a frightening rate, credit card fraud detection is becoming an important topic of research. The purpose of this paper is to present a robust framework that can be used to process large volumes of data, whereby the functionality of the framework may be extended in order to extract real time data from different distant source systems. It is then used to create a strong analytical model using the extracted data. We have implemented three different analytical techniques in order to increase the analytical accuracy of fraud prediction. An evaluation of the effectiveness of these analytical models is done by running them on a credit card dataset by using various evaluation patterns and confusion matrix.

Throughout the project, we discuss how machine learning can be used to aid in detecting credit card fraud and identifying the source of it. This paper discusses the various machine learning algorithms that are being evaluated on the basis of accuracy, precision, recall, precision, ROC, MCCC and many other factors.  In the analysis conducted, 'V14' is found to be the most dominant feature in the case of smote balancing algorithms, it is found that 'V17' is the most dominant feature when no data balancing algorithms are employed in the case.

In future, the performance of algorithms can be improved by using the techniques for tuning the hyperparameters like randomized searchCV and grid searchCV etc. We can study more machine learning techniques including neural networks for building more efficient model. The balancing techniques including borderline-smote and hybrid approaches can be used to study the effects on evaluation metrics.

# **References**

*• Value of payment card fraud worldwide 2027 | Statista*. (n.d.). Retrieved April 18, 2022, from https://www.statista.com/statistics/1264329/value-fraudulent-card-transactions-worldwide/

Alfaiz, N. S., & Fati, S. M. (2022). Enhanced Credit Card Fraud Detection Model Using Machine Learning. *Electronics*, *11*(4), 662.

*Credit Card Fraud Detection | Kaggle*. (n.d.-a). Retrieved April 18, 2022, from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

*Credit Card Fraud Detection | Kaggle*. (n.d.-b). Retrieved April 19, 2022, from https://www.kaggle.com/code/gauravduttakiit/credit-card-fraud-detection/notebook#Decision-Tree-on-balanced-data-with-ADASYN

Nicholson, C. (2019). *Evaluation Metrics for Machine Learning - Accuracy, Precision, Recall, and F1 Defined | Pathmind*. https://wiki.pathmind.com/accuracy-precision-recall-f1

Pranjalk. (2017). *Light GBM vs XGBOOST: Which algorithm takes the crown*. https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

Ramraj, S., Uzir, N., Sunil, R., & Banerjee, S. (2016). Experimenting XGBoost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, *9*, 651–662.

Sruthi E R. (2021). *Random Forest | Introduction to Random Forest Algorithm*. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Sudeep Srivastava. (2022). *How Machine Learning Helps in Financial Fraud Detection?* https://appinventiv.com/blog/role-of-machine-learning-in-financial-fraud-detection/

SWASTIK SATPATHY. (2020). *SMOTE | Overcoming Class Imbalance Problem Using SMOTE*. https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

Tanouz, D., Subramanian, R. R., Eswar, D., Reddy, G. V. P., Kumar, A. R., & Praneeth, C. H. V. N. M. (2021). Credit card fraud detection using machine learning. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 967–972.

Tiwari, P., Mehta, S., Sakhuja, N., Kumar, J., & Singh, A. K. (2021). Credit Card Fraud Detection using Machine Learning: A Study. *ArXiv Preprint ArXiv:2108.10005*.