Credit Card Fraud Detection

PRML Course Project

Introduction

What is a Credit card...?

A credit card is a credit facility provided by banks that allow customers to borrow money within a pre-approved limit.

So hackers/scammers use some methods to copy our card details using different techniques to use our credit cards.

In an era of digitalization, the usage of credit card is rapidly increasing and as a result, the cases associated with credit card fraud are on raise. So, fraud detection has become an important instrument and, in many cases, the best strategy to prevent fraud. Among the existing techniques, Machine Learning (ML) is crucial in detecting fraud.

Conventional Fraud Detection

- The rules of making a decision on determining schemes should be set manually.
- Takes an enormous amount of time
- Multiple verification methods are needed; thus, inconvenient for the user
- Finds only obvious fraud activities

Machine Learning-based Fraud Detection

- Detecting fraud automatically
- Real-time streaming
- Less time needed for verification methods
- Identifying hidden correlations in data

Credit Card Fraud Detection with Machine Learning is a process of data investigation by a Data Science team and the development of a model that will provide the best results in revealing and preventing fraudulent transactions.

This is achieved through bringing together all meaningful features of card users' transactions, such as Date, User Zone, Product Category, Amount, Provider, Client's Behavioral Patterns, etc.

The information is then run through a subtly trained model that finds patterns and rules so that it can classify whether a transaction is fraudulent or is legitimate.

Minutes of 1st meet

#Ajenda of the project

# Dataset used

#Features (28) - It contains only numeric input variables which are the result of a PCA transformation.

# Models to be implemented

Minutes of Last meet

# Overview of Project (classifying fraud or not)

# Explained about 3 different models

# Confusion matrix for models Logistic Regression, Decision tree, Random forest

# Considering Recall score from confusion matrix

#Compared Recall score of all three

# Aiming for better score if possible, so thought of implementing two other models

Comments

# Real life example of fraud detection

# What is the agenda for next project

DATASET

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 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

How can we tackle these challenges …?

The model must be simple and fast enough to detect the anomaly and classify it as a fraudulent transaction as quickly as possible.

Parameters of the data are reduced to protect the privacy of the user .

MODELS

Ø **Logistic Regression**

Ø **Decision Trees**

Ø **Random Forest Classifier**

Based on the accuracy score we will use that model to detect the fraud transactions.

Precision

Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly).

Precision helps us to visualize the reliability of the machine learning model in classifying the model as positive

Recall

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples.The higher the recall, the more positive samples detected.

Unlike Precision, Recall is independent of the number of negative sample classifications. Further, if the model classifies all positive samples as positive, then Recall will be 1

F1-Score

The F1 score is a measure of a test's accuracy—it is the harmonic mean of precision and recall. It can have a maximum score of 1 (perfect precision and recall) and a minimum of 0. Overall, it is a measure of the preciseness and robustness of your model.

Confusion Matrix

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix.

Confusion matrix accuracy is not meaningful for unbalanced classification

Logistic Regression

This type of statistical model is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

Logistic Regression is used when the dependent variable(target) is categorical.

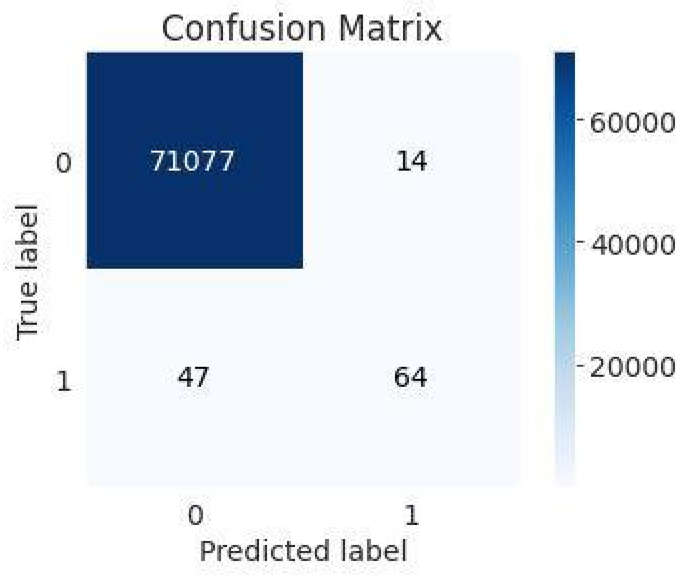
For example:

⇒ To predict whether an email is spam (1) or (0)

⇒ Whether the tumor is malignant (1) or not (0)

⇒ Whether the the person is male (1) or female (0)

Problems with this model



With the Logistic Regression Model, we have:

71077 transactions classified as normal and were actually normal;

14 transactions classified as fraud but that were really normal (type 1 error);

47 transactions classified as normal but which were fraud (type 2 error);

64 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 42 out of 100 fraudulent transactions.

# Accuracy score of the Logistic Regression model is 0.9991432824920649

# F1 score of the Logistic Regression model is 0.6772486772486772

# Recall of theLogistic Regression model is 0.57

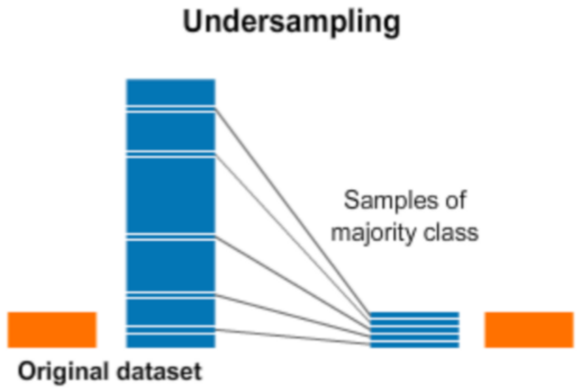
Accuracy in a highly unbalanced data set does not represent a correct value for the efficiency of a model.

Initially, a method should be applied to balance the data before taking into account any performance evaluation metrics.

** Accuracy in a highly unbalanced data set does not represent a correct value for the efficiency of a model. Initially, a method should be applied to balance the data before taking into account any performance evaluation metrics.

Under sampling-Working with unbalanced data

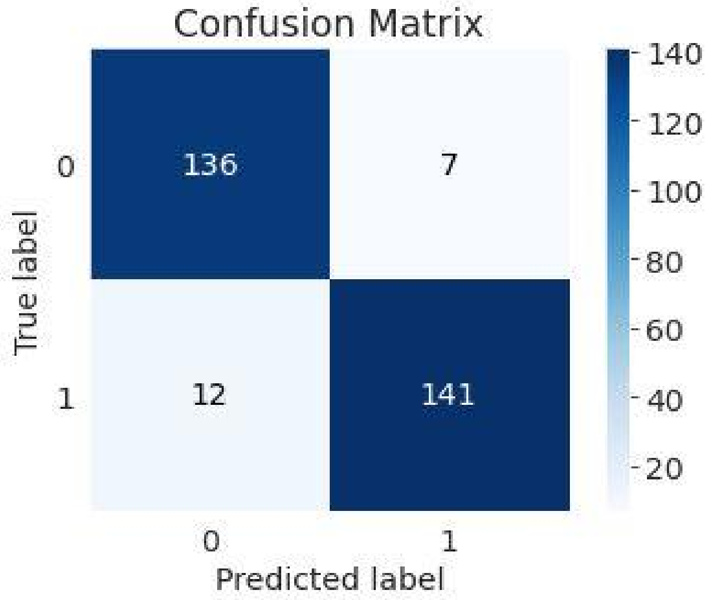
Undersampling is the technique of removing major class records from the sample. In this case, it is necessary to remove random records from the legitimate class (No fraud), in order to obtain a number of records close to the amount of the minority class (fraud) in order to train the model.



Applying the undersampling technique

In this case, we will use the undersampling technique to obtain a uniform division between fraud and valid transactions. This will make the training set small, but with enough data to generate a good classifier.

Using the "new" classifier for balanced data



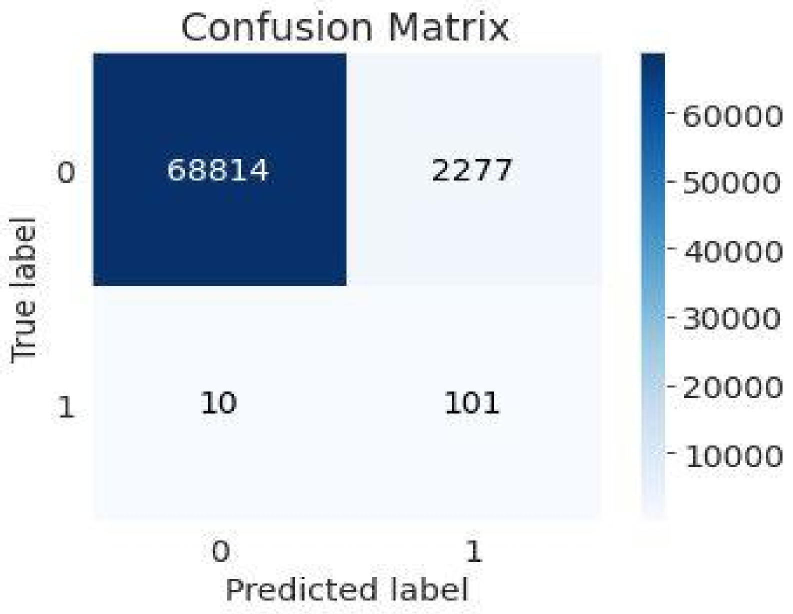
# Accuracy score of the Logistic Regression model after using undersampling for balanced data is 0.9358108108108109

# F1 score of the Logistic Regression model after using undersampling for balanced data is 0.9368770764119602

Accuracy has decreased, but sensitivity has greatly increased. Looking at the confusion matrix, we can see a much higher percentage of correct classifications of fraudulent data.

Unfortunately, a greater number of fraud classifications almost always means a correspondingly greater number of valid transactions also classified as fraudulent.

Using the "new" classifier for the original data set



68814 transactions classified as normal and were actually normal;

2277 transactions classified as fraud but that were really normal (type 1 error);

10 transactions classified as normal but which were fraud (type 2 error);

101 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 9 out of 100 fraudulent transactions.

# Accuracy score of the Logistic Regression model after using undersampling for original dataset is 0.9678801157270863

# F1 score of the Logistic Regression model after using undersampling for original dataset is 0.08115709120128566

Recall of the Decision Tree model is 0.91

Conclusion

The algorithm was much better at capturing fraudulent transactions (47 classification errors at the beginning of the project to 10 current), but much worse at incorrectly labeling valid transactions (14 to 2277).

Before using under sampling technique,

Miss Classifications :

Fraud transactions - 42%

Valid transactions - 0.02%

After using under sampling technique,

Miss Classifications :

Fraud transactions - 9%

Valid transactions - 3%

Decision trees

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

We make some assumptions while implementing the Decision-Tree algorithm. These are listed below:-

1. At the beginning, the whole training set is considered as the root.
2. Feature values need to be categorical. If the values are continuous then they are discretized prior to building the model.

3. Records are distributed recursively on the basis of attribute values.
4. Order to place attributes as root or internal node of the tree is done by using some statistical approach.

In a Decision Tree algorithm, there is a tree-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from the root node to leaf node represent classification rules.

Root Node

- It represents the entire population or sample. This further gets divided into two or more homogeneous sets.

Splitting

- It is a process of dividing a node into two or more sub-nodes.

Decision Node

- When a sub-node splits into further sub-nodes, then it is called a decision node.

Leaf/Terminal Node

- Nodes that do not split are called Leaf or Terminal nodes.

Pruning

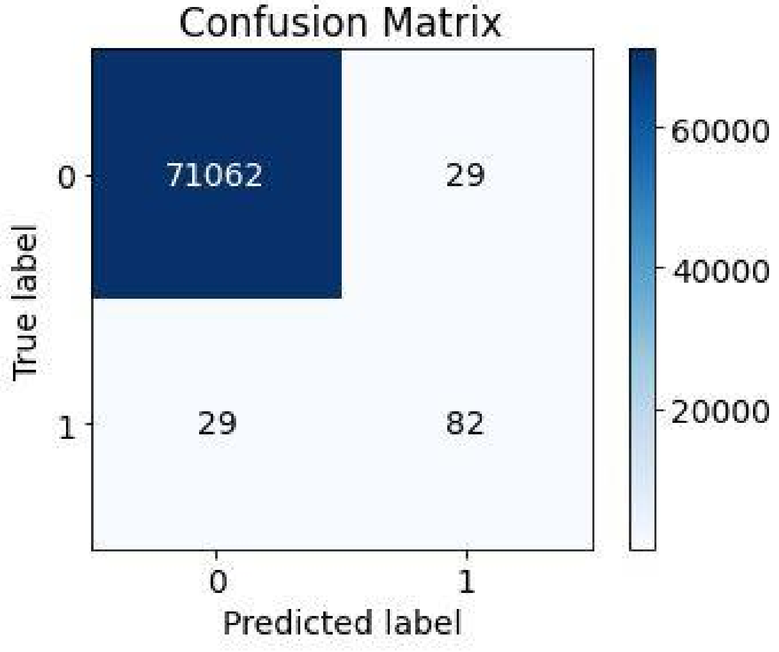
- When we remove sub-nodes of a decision node, this process is called pruning. It is the opposite process of splitting.

Branch/Subtree

- A subsection of an entire tree is called a branch or sub-tree.

Parent and Child Node

- A node, which is divided into sub-nodes is called the parent node of sub-nodes where sub-nodes are the children of a parent node.



71062 transactions classified as normal and were actually normal;

29 transactions classified as fraud but that were really normal (type 1 error);

29 transactions classified as normal but which were fraud (type 2 error);

82 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 26 out of 100 fraudulent transactions.

```
# Accuracy score of the Decision Tree model is 0.9991854161399961

# F1 score of the Decision Tree model is 0.7387387387387387

# Recall of the Decision Tree model is 0.74
```

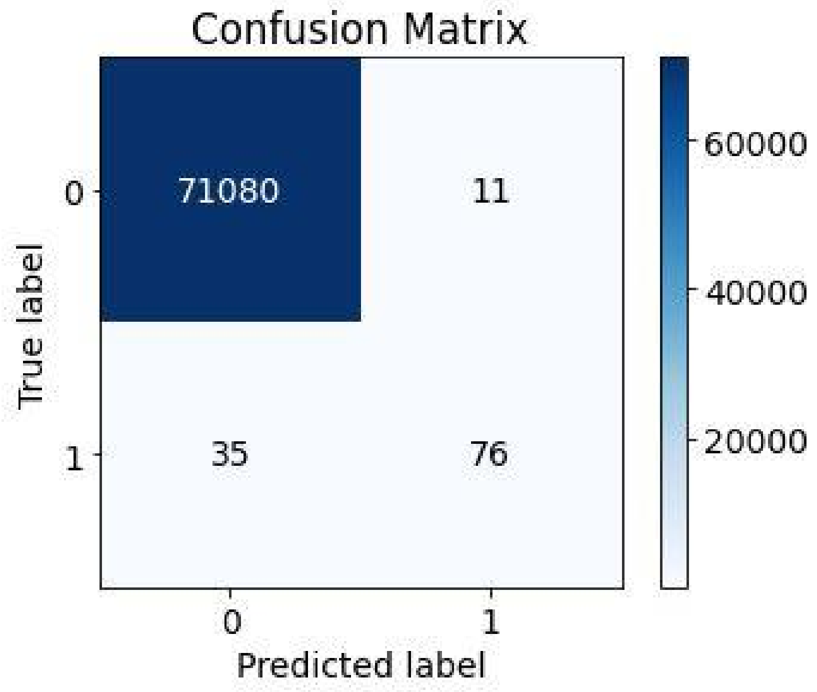
Random-Forest

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The random forest uses many trees, and it makes a prediction by averaging the predictions of each component tree. It generally has much better predictive accuracy than a single decision tree and it works well with default parameters. If you keep modeling, you can learn more models with even better performance, but many of those are sensitive to getting the right parameters.



71080   transactions classified as normal and were actually normal;

11 transactions classified as fraud but that were really normal (type 1 error);

35 transactions classified as normal but which were fraud (type 2 error);

76 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 31 out of 100 fraudulent transactions.

# Accuracy score of the Random Forest model is 0.9993539507317211

# F1 score of the Random Forest model is 0.7676767676767677

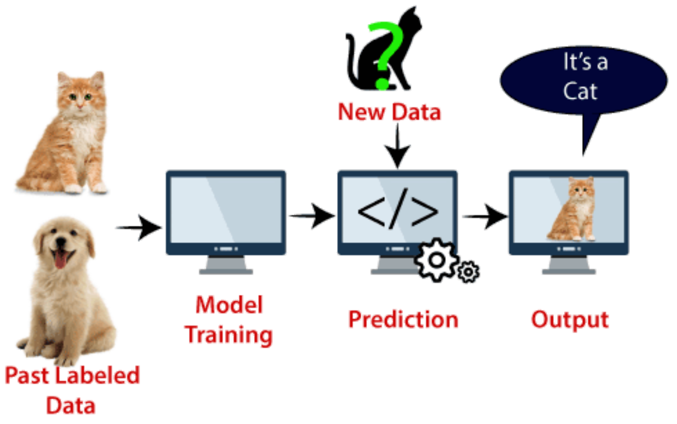
# Recall of the Random Forest model is 0.68

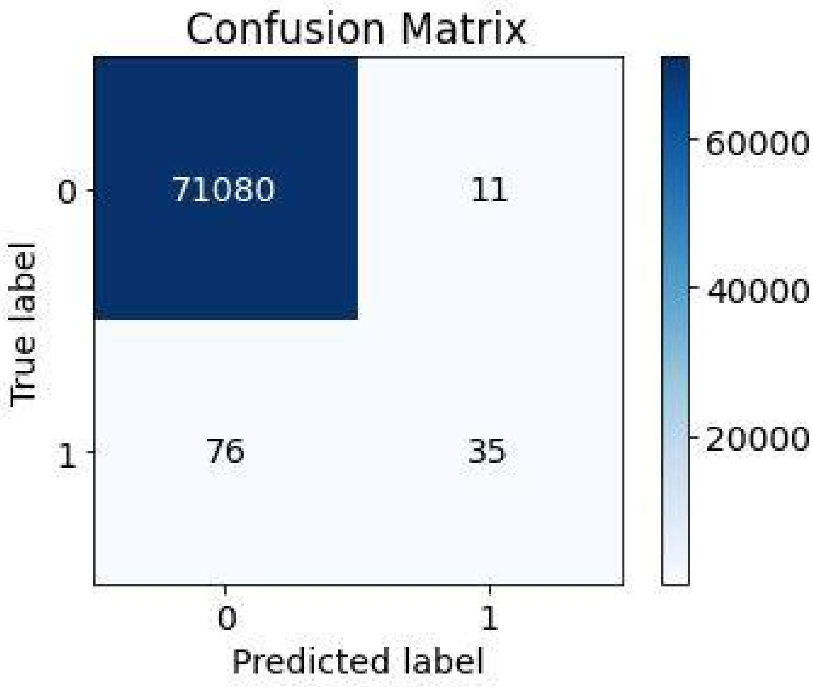
Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence the algorithm is termed as Support Vector Machine.

SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as the support vector creates a decision boundary between these two data (cat and dog) and chooses extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat.





71080   transactions classified as normal and were actually normal;

11 transactions classified as fraud but that were really normal (type 1 error);

76 transactions classified as normal but which were fraud (type 2 error);

35 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 68 out of 100 fraudulent transactions.

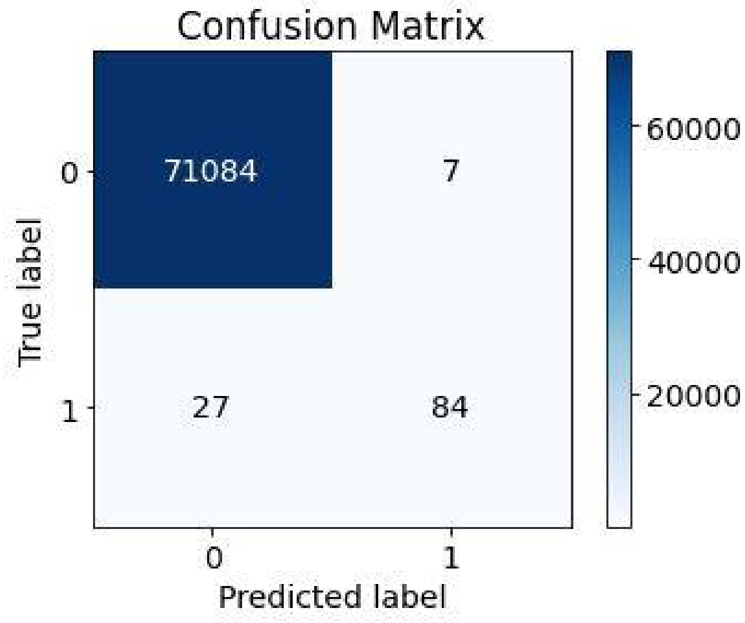
# Accuracy score of the SVM model is 0.9987781242099941

# F1 score of the SVM model is 0.4458598726114649

# Recall of the SVM model is 0.37

XG-Boost

XGBoost is **a popular and efficient open-source implementation of the gradient boosted trees algorithm**. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.



71084 transactions classified as normal and were actually normal;

7 transactions classified as fraud but that were really normal (type 1 error);

27 transactions classified as normal but which were fraud (type 2 error);

84 transactions were classified as fraud and were actually fraud.

Thus, although the accuracy was excellent, the algorithm wrongly classified about 24 out of 100 fraudulent transactions.

```
# Accuracy score of the XG-Boost model is 0.999522485323446

# F1 score of the Random XG-Boost is 0.8316831683168316

# Recall of the XG-Boost model is 0.78
```

K-Means

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm is, how the algorithm works, along with the Python implementation fof k-means clustering.

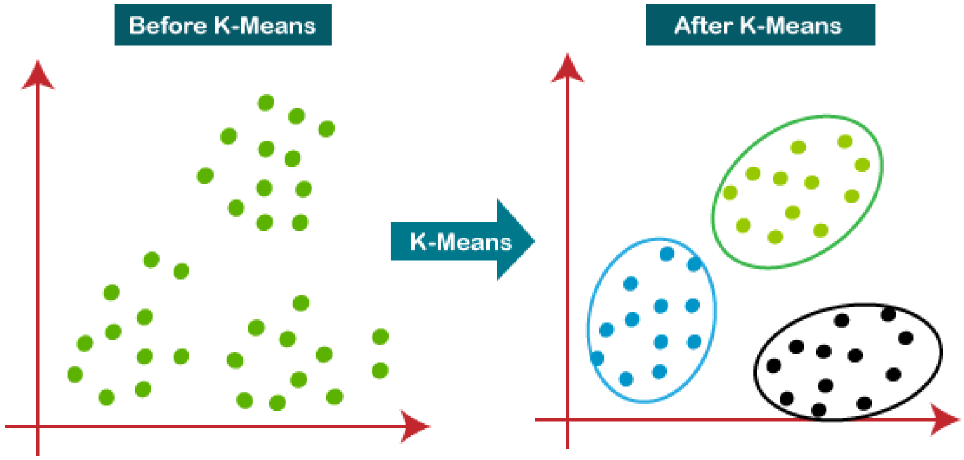
It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

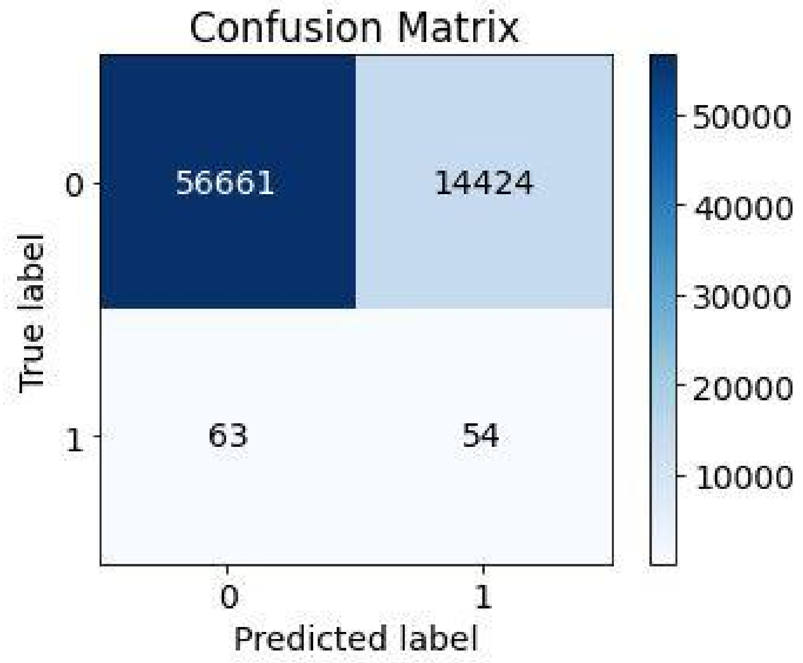
The below diagram explains the working of the K-means Clustering Algorithm:



The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has data points with some commonalities, and it is away from other clusters.



56661 transactions classified as normal and were actually normal;

14424 transactions classified as fraud but that were really normal (type 1 error);

63 transactions classified as normal but which were fraud (type 2 error);

54 transactions were classified as fraud and were actually fraud.

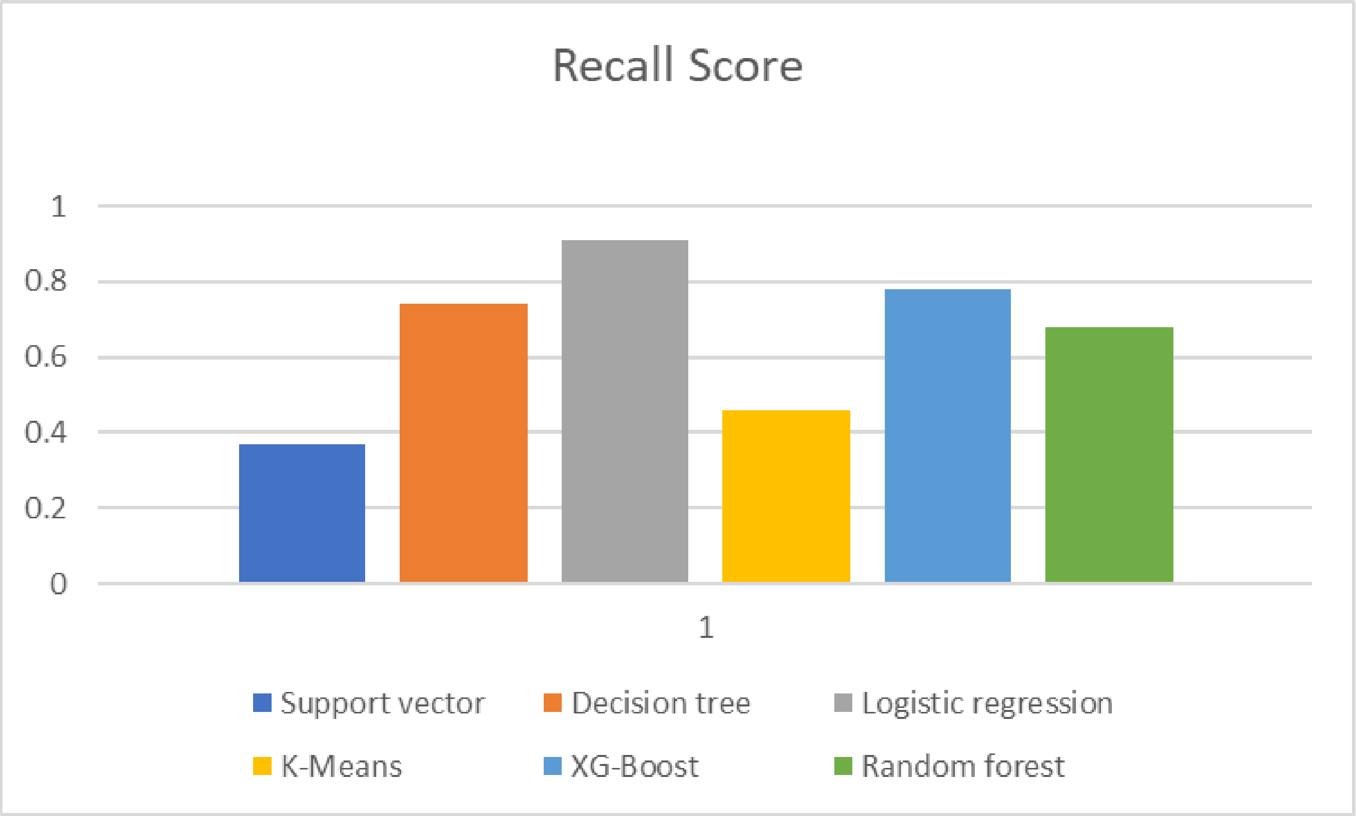
Thus, although the accuracy was excellent, the algorithm wrongly classified about 53 out of 100 fraudulent transactions.

# Accuracy score of the K-Means model is 0.7965366141400523

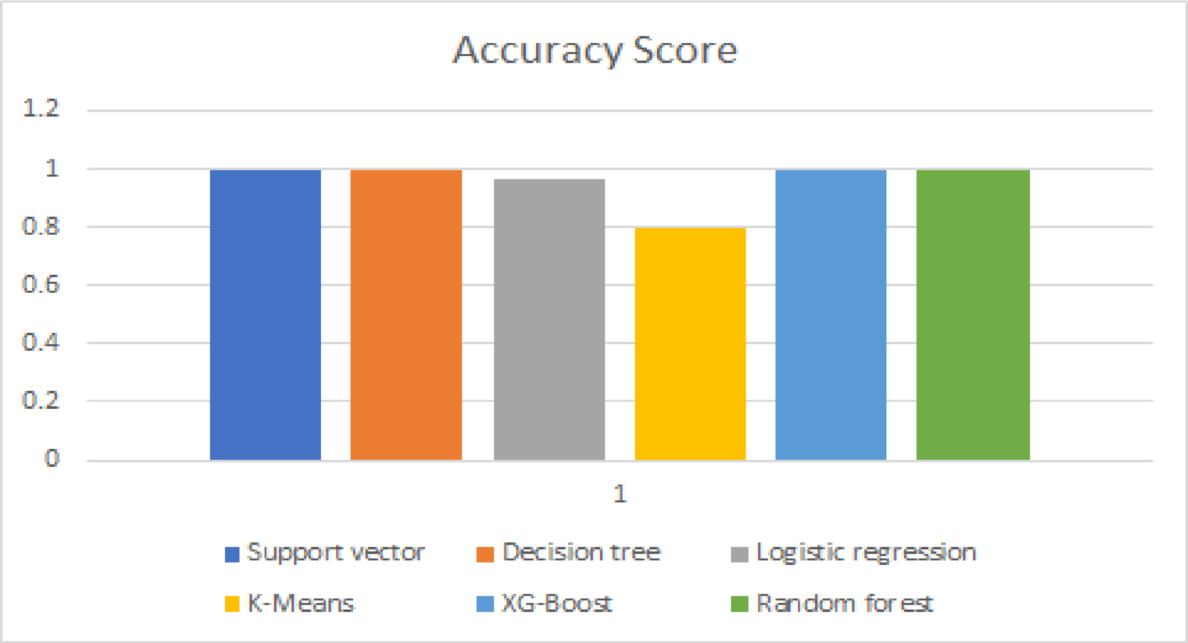
# F1 score of the K-means model is 0.0073997944501541625

# Recall of the K-means model is 0.47

Recall Score of all implemented models



Accuracy


Accuracy Score
Support vector
Decision tree
Logistic regression
K-Means
XG-Boost
Random forest