**Table of Contents**

**Chapter 1. INTRODUCTION**

* 1. What is Credit Card Fraud
  2. What is Credit Card Fraud Detection
  3. How to prevent Credit Card Fraud

**Chapter 2. LITERATURE REVIEW**

2.1 Table 2.1

2.2 Fig 2.1

**Chapter 3. METHODOLOGY**

3.1 Block Diagram

3.2 Explanation

3.3 Dataset

3.4 Algorithms

**Chapter 4. RESULT & DISCUSSION**

4.1 Visualisation

4.2 Accuracy Rate

4.3 AUC Score

4.4 Recall

4.5 F1 Score

4.6 Comparison between the accuracy of past results and present result

**Chapter 5. PERFORMANCE MEASURE**

5.1 Classification Matrices

5.2 Precision

5.3 Recall

5.4 F-1 Score

5.5Performance Measure of different Algorithms

5.6 Confusion Matrix Heatmap

**Chapter 6. HARDWARE**

6.1 Table 6.1

**Chapter 7. CONCLUSION & FUTURE SCOPE**

**REFERENCES**

**CHAPTER 1: INTRODUCTION**

* 1. **What is Credit Card Fraud?**

Credit card fraud is a type of financial fraud that occurs when someone uses another person's credit card or credit card information to make unauthorized purchases or transactions. This can happen in a variety of ways, including through stolen or lost credit cards, skimming devices, phishing scams, or hacking into databases that contain credit card information. The fraudulent activity may involve small or large transactions, and can have a significant impact on the victim's finances and credit score.

* 1. **What is Credit Card Fraud Detection?**

Credit card fraud detection refers to the process of identifying fraudulent credit card transactions, which are unauthorized and illegal. This involves the use of various techniques and algorithms to analyze patterns and identify anomalies in the transaction data. Credit card fraud detection is important to prevent financial losses for both the cardholders and the card issuers. With the increasing use of credit cards for online transactions, the need for effective fraud detection systems has become even more critical.

* 1. **How to Prevent Credit Card Fraud**

Keep your credit cards safe: Always keep your credit cards in a safe place and never lend them to anyone. Also, make sure to sign the back of your credit card as soon as you receive it.

Monitor your account regularly: Keep a close eye on your credit card transactions and immediately report any unauthorized charges to your credit card issuer.

Use secure websites: When making online purchases, use secure websites that have a valid SSL certificate. Look for the padlock icon in the address bar of your browser, which indicates that the website is secure.

Don't share your card information: Never give out your credit card information to anyone, especially over the phone or via email.

Set up alerts: Many credit card issuers offer alerts that notify you of suspicious activity on your account. Set up these alerts to stay informed about your account activity.

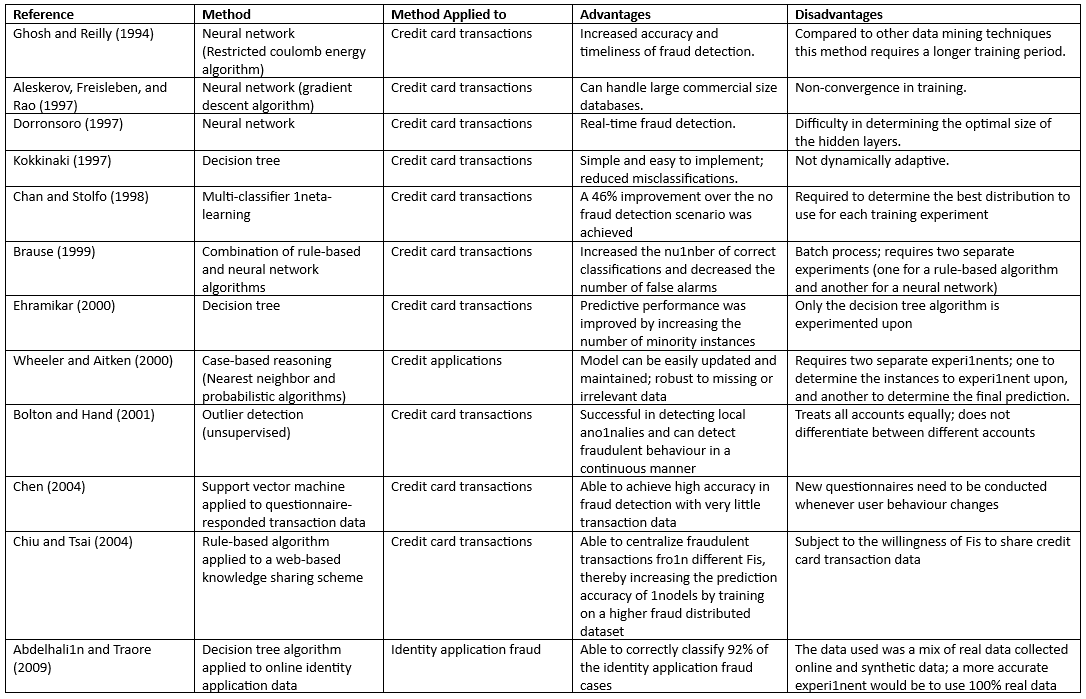
Check your credit report: Regularly check your credit report to ensure that all of the information is accurate and there are no unauthorized accounts or inquiries.

Use EMV chip-enabled cards: EMV chip-enabled cards are more secure than traditional magnetic stripe cards and can help prevent credit card fraud.

**CHAPTER 2: LITERATURE REVIEW**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no. | Study | Techniques | Result |
| 1 | Suresh K Shirgave  Chetan J. Awati Rashmi More , Sonam S. Pat il (2019) | * Random Forest * Logistic Regression * KNN * SVM * Naive Bayes | * 96.2% * 94.7% * 94.2% * 93.8% * 93.7% |
| 2 | Sonal Mehndiratta, Mr. Kamal Gupta (2019) | * ANN * KNN | * 90.3% * 93.8% |
| 3 | K.Ratna Sree Valli , P.Jyothi , G.Varun Sai , R.Rohith Sai Subash (2020) | * Naïve Bayes * Logistic Regression * Random Forest | * 90.8% * 99.5% * 100% |
| 4 | Nayan Uchhana, Ravi Ranjan, Shashank Sharma, Deepak Agrawal, Anurag Punde (2021) | * SVM * Naive Bayes * Logistic Regression * Random Forest | * 92% * 91.1% * 83.6% * 89% |
| 5 | Aisha Mohammad Fayyomi, Derar Eleyan, Amina Eleyan (2021) | * Random Forest * Logistic Regression * Decision Tree * KNN * K-means | * 92.2% * 94.6% * 88% * 75.6% * 53.2% |
| 6 | Meera AlEmad (2022) | * KNN * Naive Bayes * Logistic Regression * SVM | * 99.8% * 97.7% * 99.9% * 99.9% |
| 7 | Rejwan Bin Sulaiman, Vitaly Schetinin, Paul Sant (2022) | * Random Forest * ANN * SVM * KNN | * 96.7% * 92% * 91% * 72% |
| 8 | Fawaz Khaled Alarfaj; Iqra Malik; Hikmat Ullah Khan; Naif Almusallam; Muhammad Ramzan; Muzamil Ahmed (2022) | * Decision Tree * KNN * Logistic Regression * SVM * Random Forest * XG Boost | * 99.93% * 99.95% * 99.91% * 99.93% * 99.92% * 99.94% |

**Table 2.1** This is the table of Literature Review

****

**Fig 2.1** This is the fig. of literature review

**CHAPTER 3: METHODOLOGY**

**3.1** **Block Diagram**

Training Set

Model Evaluation

Apply Algorithm

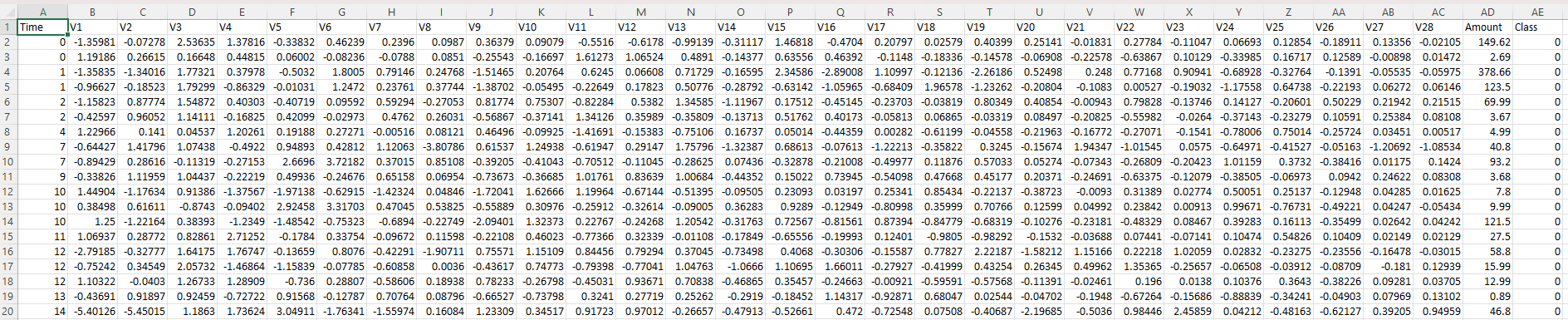
Testing Set

Credit card Transactions Data

**3.2 Explanation**

At first, we had collected our dataset. Then we processed our dataset. After that we visualized our dataset. Then we split our data into training and testing. Then our model evaluation was done.

**3.3 Data set**

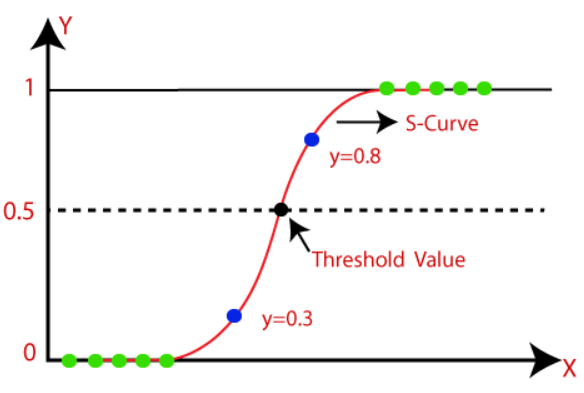


We had taken CSV data. It has 284808 rows and 30 columns and its size is 143MB.

**3.4 Algorithms**

***Logistic Regression: -*** Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).



**Fig 3.4.1**

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The above image (Fig. 3.4.1) is showing the logistic function.

***Logistic Function (Sigmoid Function):***

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1.

The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

***Steps in Logistic Regression*:** To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

Data Pre-processing step

Fitting Logistic Regression to the Training set

Predicting the test result

Test accuracy of the result (Creation of Confusion matrix)

Visualizing the test set result.

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

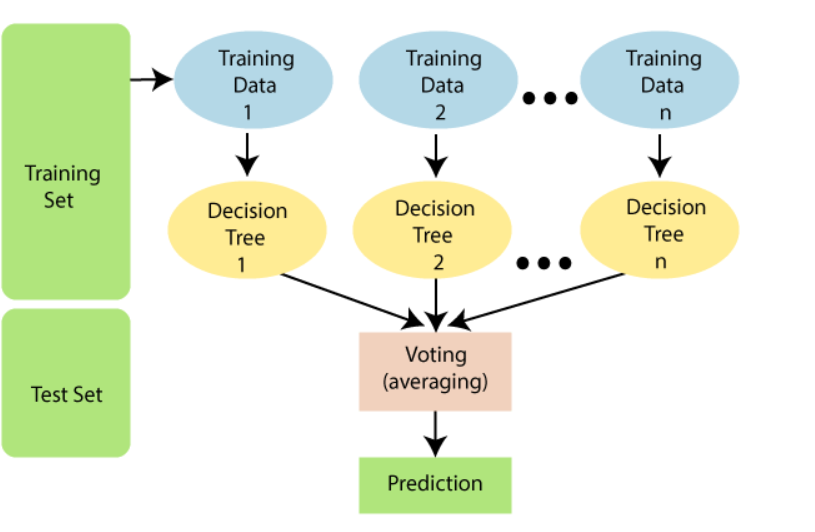
We know the equation of the straight line can be written as:

In Logistic Regression y can be between 0 and 1 only, so for this let’s divide the above equation by (1-y):

But we need range between –[infinity] to +[infinity], then take logarithm of the equation it will become:

The above equation is the final equation for Logistic Regression.

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



**Fig 3.4.2**

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.

***Procedure: -***

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

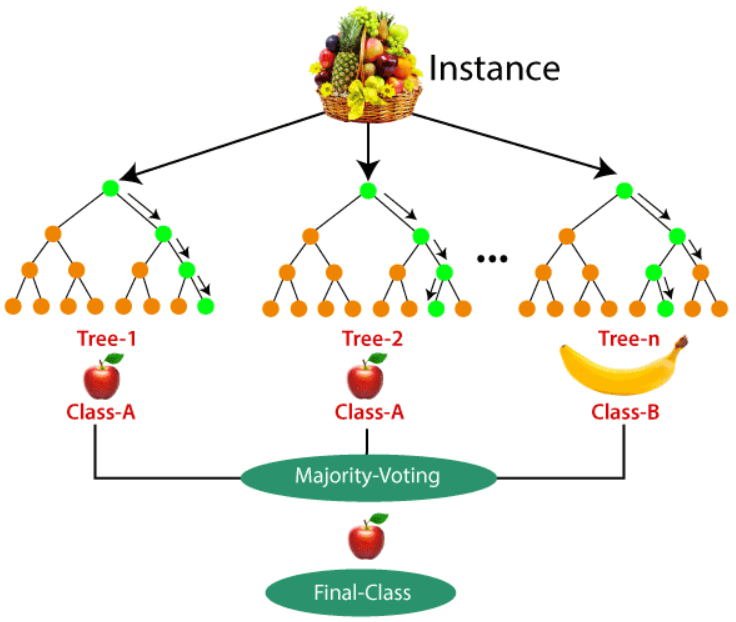
Step-2: Build the decision trees associated with the selected data points (Subsets).

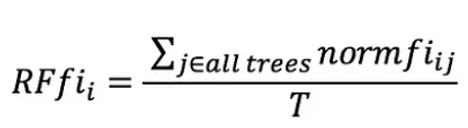
Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

***Example: -*** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:





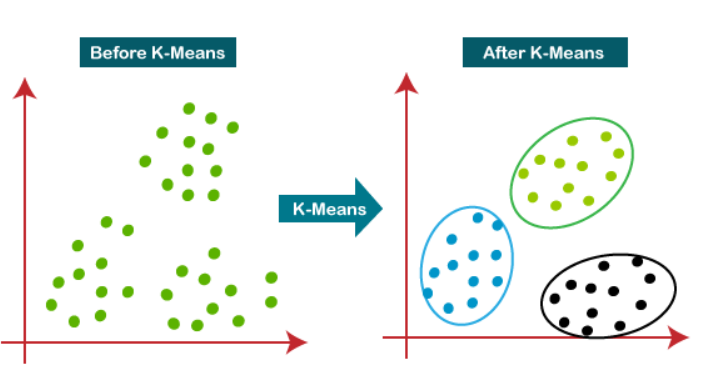
The final feature importance, at the Random Forest level, is it’s average over all the trees. The sum of the feature’s importance value on each trees is calculated and divided by the total number of trees.

* RFfi sub(i)= the importance of feature i calculated from all trees in the Random Forest model
* normfi sub(ij)= the normalized feature importance for i in tree j
* T = total number of trees

***K-means Classifier: -*** 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.

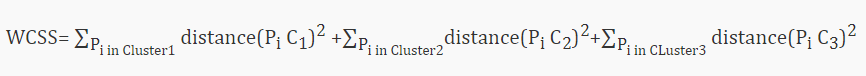
The k-means clustering algorithm mainly performs two tasks:

* 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.



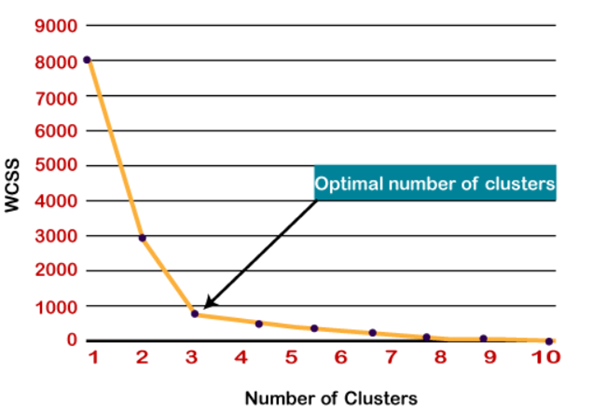
**Fig 3.4.3**

***Elbow Method: -*** This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:



In the above formula of WCSS,

∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.



**Fig 3.4.4** WCSS graph

***Procedure: -***

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

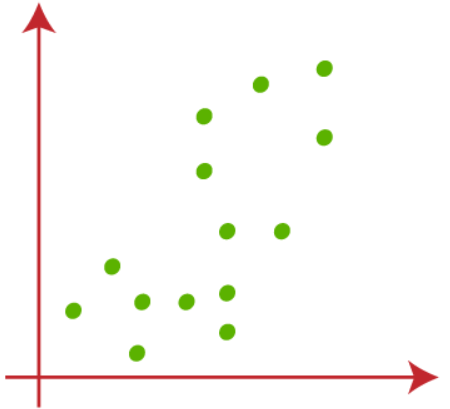
Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

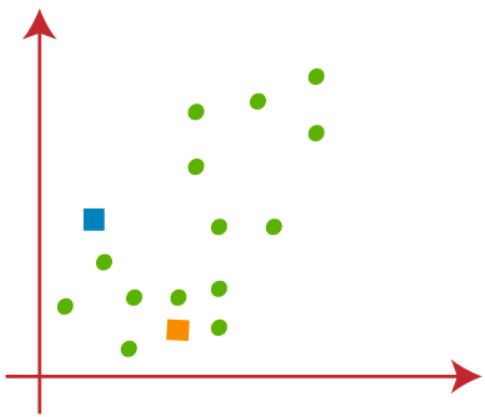
Step-7: The model is ready.

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:

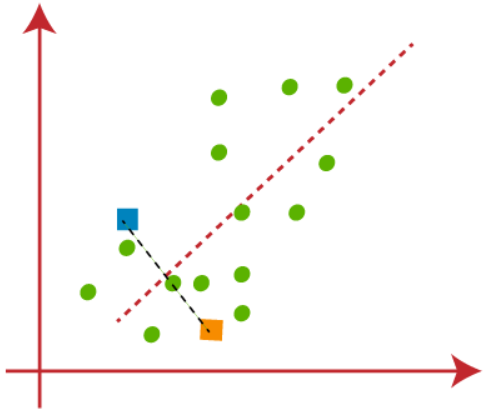


Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.

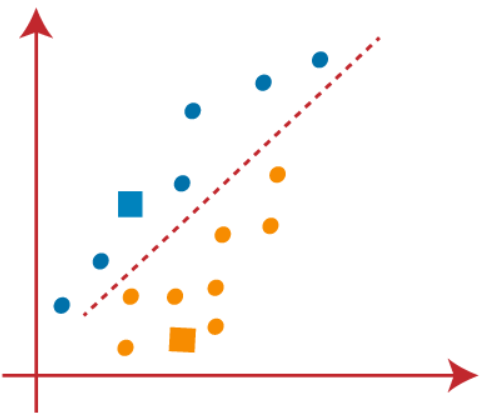
We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



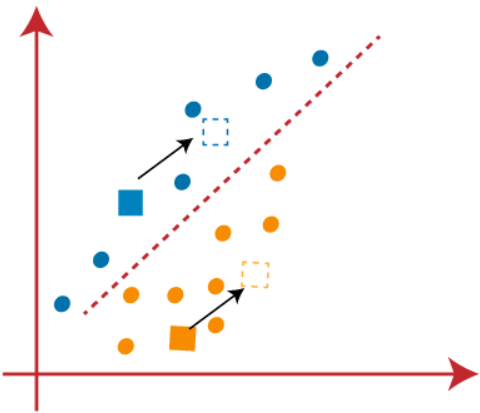
Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



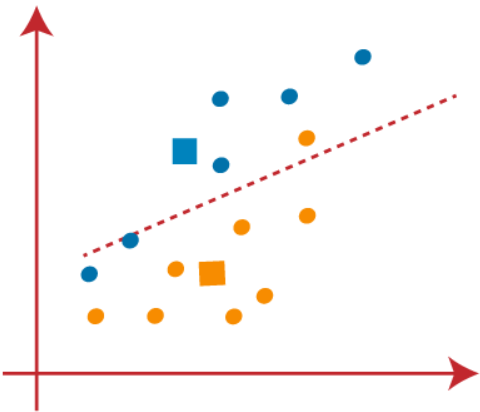
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:



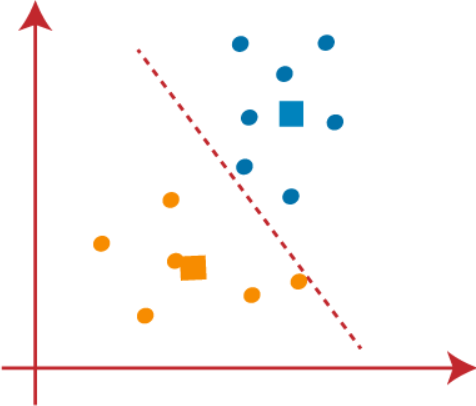
From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

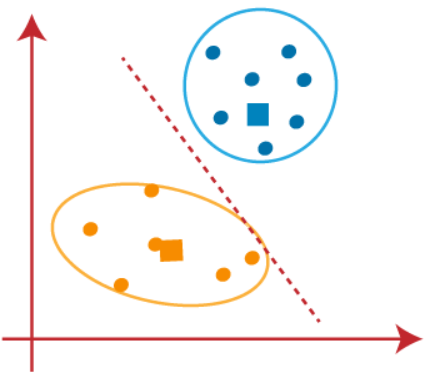
We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



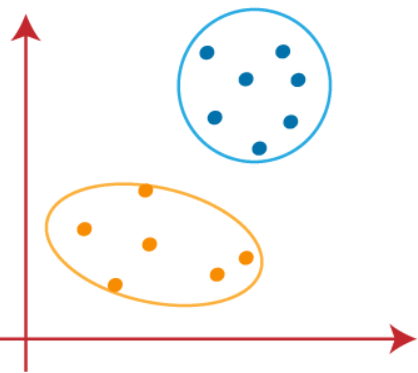
As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:

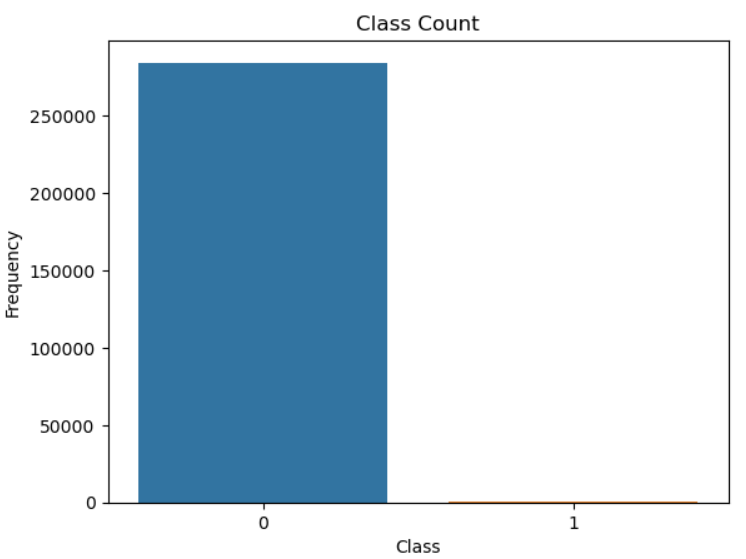
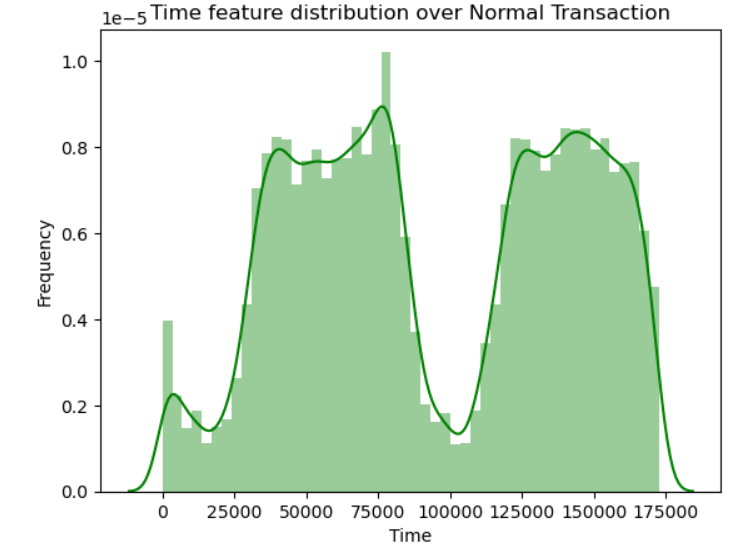


As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



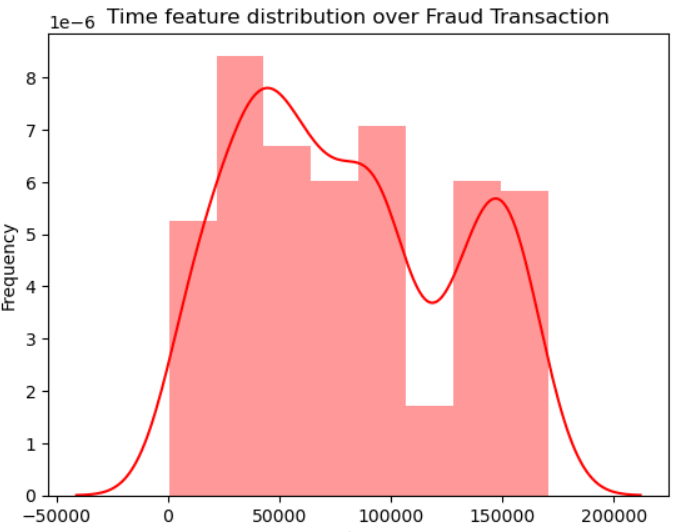
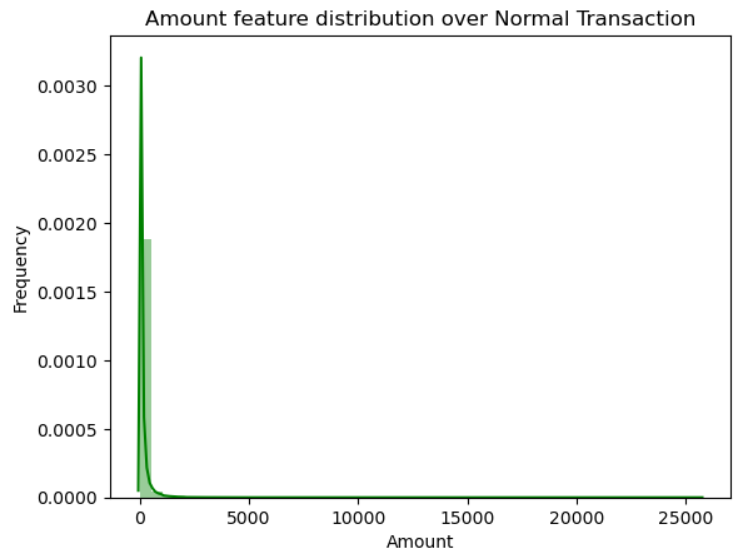
**CHAPTER 4: RESULT & DISCUSSION**

**4.1 Visualization**

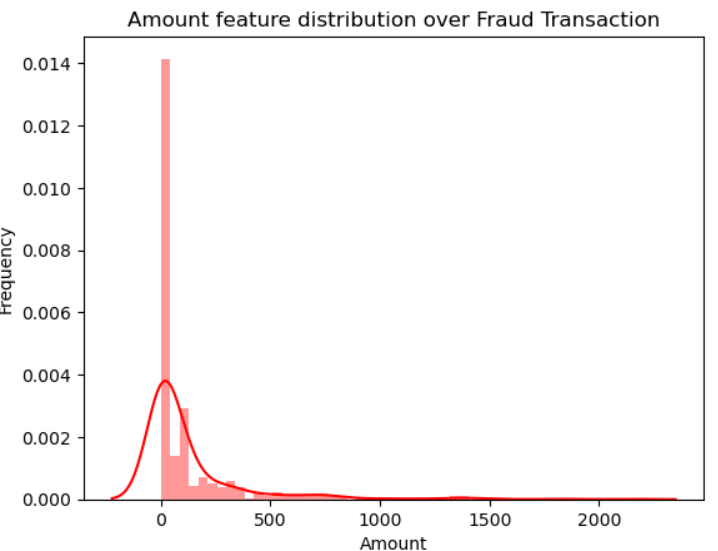
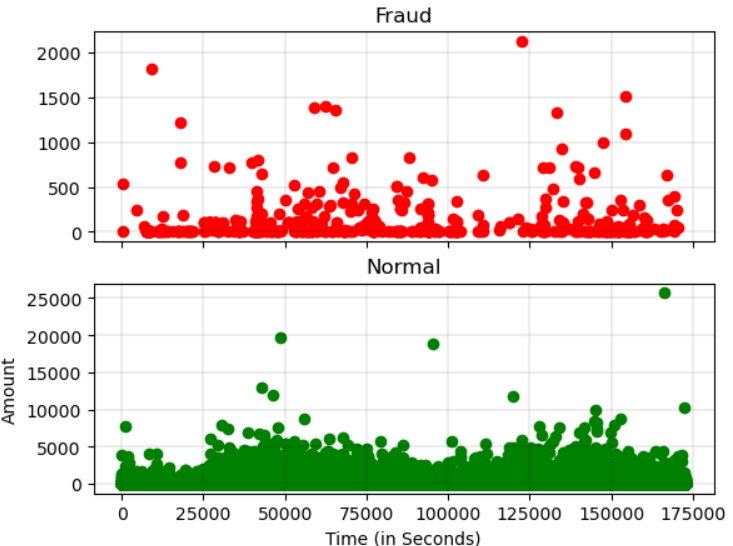
**** ****

Here 0 represents Normal Transaction & 1 represents Time Feature distribution over normal trans.

Fraud Transaction

Time Feature distribution over Fraud transaction Amount Feature distribution over normal trans.

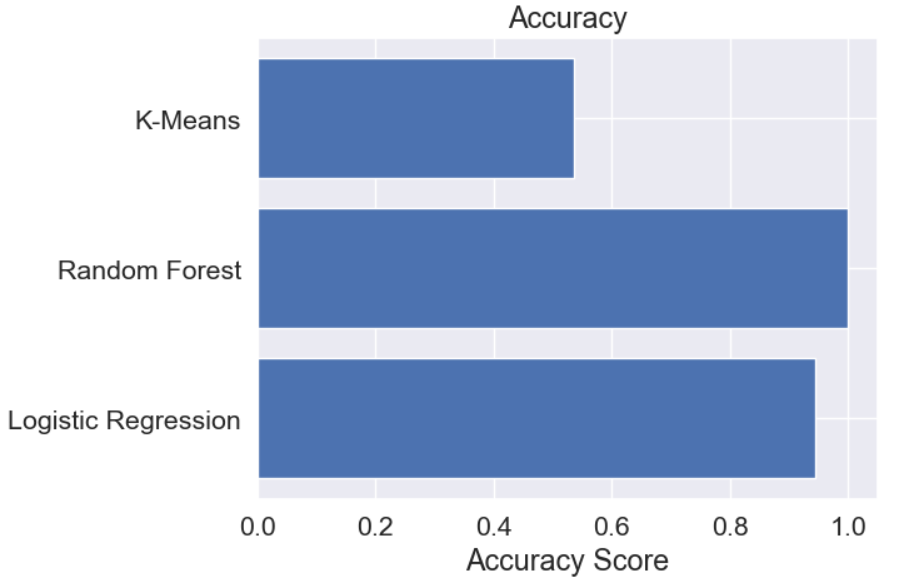
Amount Feature distribution over Fraud Transaction Time of transaction vs amount by Class

**4.2 Accuraacy Rate**

***Split Ratio:- 80-20***

****

****

****

**Fig 4.2.1**

We split our dataset into 80:20 ratio. We use three different algorithm such as Logistic Regression, Random Forest and K-means . And we got the highest accuracy about 99.98% in Random Forest and lowest accuracy about 53.47% in K-means.

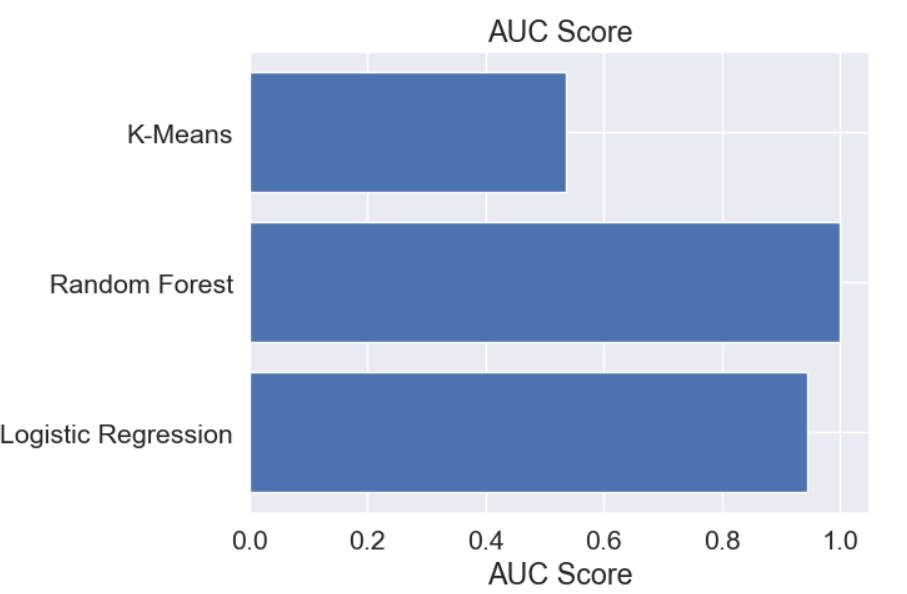
**4.3 AUC Score**

***Split Ratio:- 80-20***

 - Logistic Regression

 - Random Forest

- K-means



**Fig 4.3.1**

We split our dataset into 80:20 ratio. We use three different algorithm such as Logistic Regression, Random Forest and K-means . And we got the highest AUC score about 0.99 in Random Forest and lowest AUC score about 0.53 in K-means.

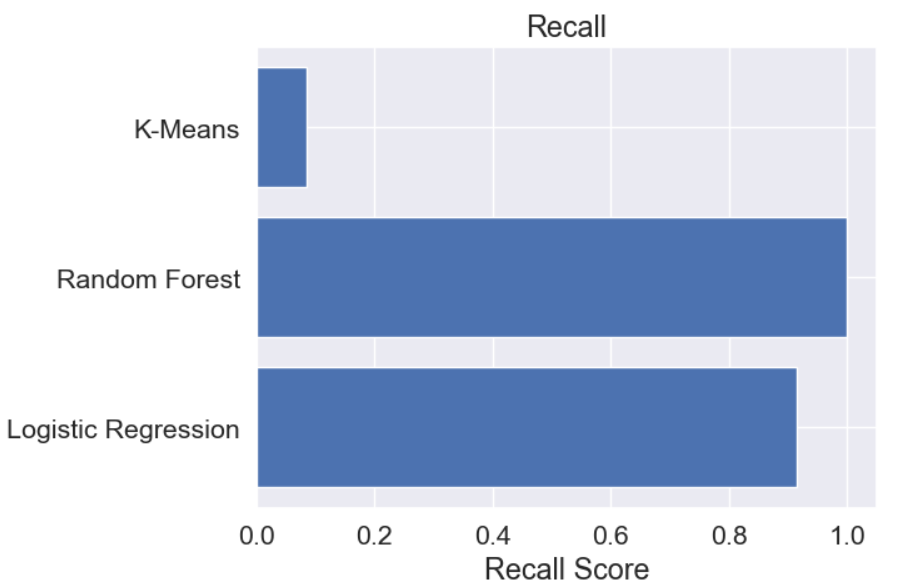
**4.4 Recall**

***Split Ratio:- 80-20***

**** - Logistic Regression

**-** Random Forest

**-** K-means

****

**Fig 4.4.1**

We split our dataset into 80:20 ratio. We use three different algorithm such as Logistic Regression, Random Forest and K-means . And we got the highest Recall about 0.99 in Random Forest and lowest Recall about 0.53 in K-means.

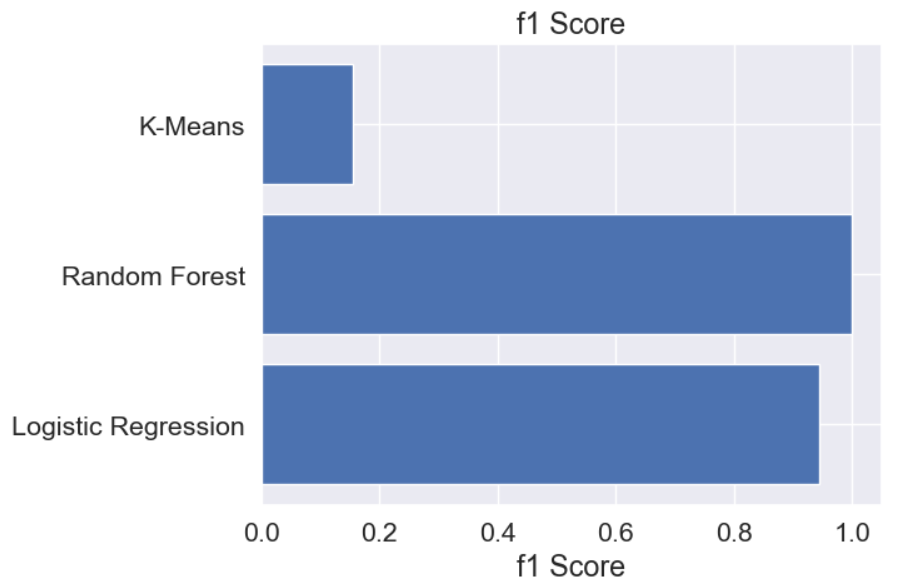
**4.5 F1 Score**

***Split Ratio:- 80-20***

**** - Logistic Regression

** -** Random Forest

**** - K-means

****

**Fig 4.5.1**

We split our dataset into 80:20 ratio. We use three different algorithm such as Logistic Regression, Random Forest and K-means . And we got the highest f1 score about 0.99 in Random Forest and lowest Recall score about 0.15 in K-means.

**4.6** **Comparison between the accuracy of past results and present result**

PAST

|  |  |  |
| --- | --- | --- |
| Researcher Name | Algorithm Name | Accuracy |
| Suresh K Shirgave  Chetan J. Awati Rashmi More , Sonam S. Pat il (2019) | * Random Forest * Logistic Regression | * 96.2% * 94.7% |
| Aisha Mohammad Fayyomi, Derar Eleyan, Amina Eleyan (2021) | * Random Forest * Logistic Regression * K-means | * 92.2% * 94.6% * 53.2% |
| Rejwan Bin Sulaiman, Vitaly Schetinin, Paul Sant (2022) | * Random Forest | * 96.7% |

PRESENT

|  |  |
| --- | --- |
| Algorithm Name | Accuracy |
| Logistic Regression | 94.48% |
| Random Forest | 99.98% |
| K-means | 53.47% |

**CHAPTER 5:** **PERFORMANCE MEASURE**

**5.1 Classification Matrices**

Classification problems are one of the world’s most widely researched areas. Use cases are present in almost all production and industrial environments. Speech recognition, face recognition, text classification- the list is endless.

Classification models have discrete output, so we need a metric that compares discrete classes in some form. Classification Metrics evaluate a model’s performance and tell you how good or bad the classification is, but each of them evaluates it in a different way.

**5.2 Precision**

Precision is the ratio of **true positives** to the total of the true positives and false positives. Precision looks to see how much junk positives got thrown in the mix. If there are no bad positives (those FPs), then the model had 100% precision. The more FPs that get into the mix, the uglier that precision is going to look.

To calculate a model’s precision, we need the positive and negative numbers from the confusion matrix.

***Precision = TP/(TP + FP)***

**5.3 Recall**

Recall goes another route. Instead of looking at the number of false positives the model predicted, recall looks at the number of **false negatives** that were thrown into the prediction mix.

***Recall = TP/(TP + FN)***

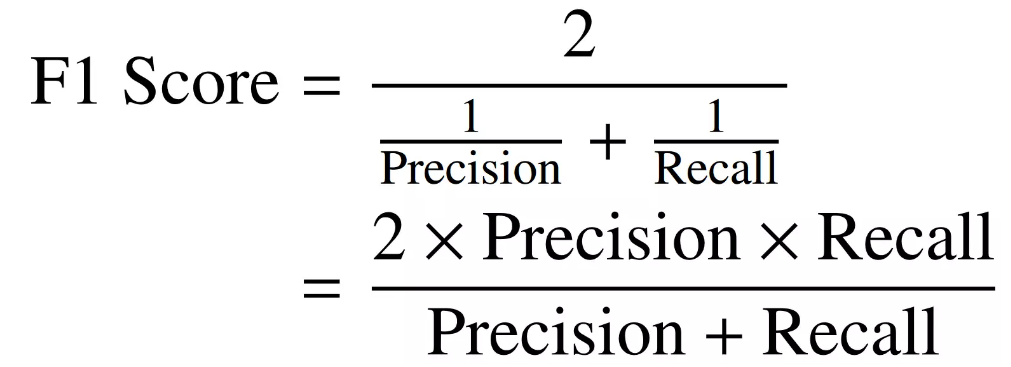
The recall rate is penalized whenever a false negative is predicted. Because the penalties in precision and recall are opposites, so too are the equations themselves. Precision and recall are the yin and yang of assessing the confusion matrix.

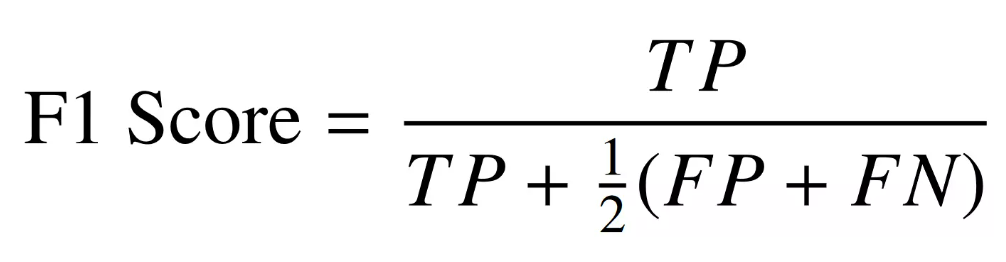
**5.4 F-1 Score**

It is a machine learning evaluation metric that measures a model’s accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset. This can be a reliable metric only if the dataset is class-balanced; that is, each class of the dataset has the same number of samples.

Nevertheless, real-world datasets are heavily class-imbalanced, often making this metric unviable. For example, if a binary class dataset has 90 and 10 samples in class-1 and class-2, respectively, a model that only predicts “class-1,” regardless of the sample, will still be 90% accurate. Accuracy computes how many times a model made a correct prediction across the entire dataset. The F1 score combines precision and recall using their harmonic mean, and maximizing the F1 score implies simultaneously maximizing

both precision and recall. Thus, the F1 score has become the choice of researchers for evaluating their models in conjunction with accuracy.

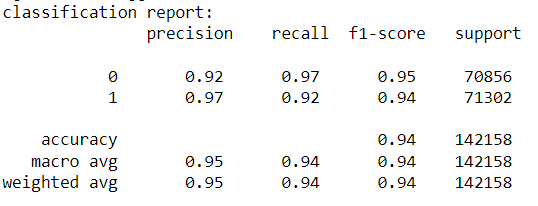




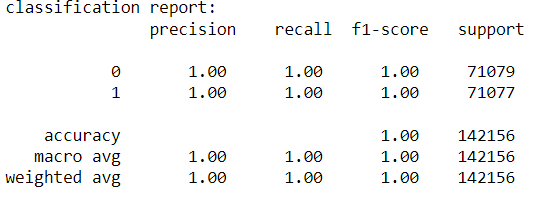
**5.5Performance Measure of different Algorithms**

***Split Ratio: - 80-20***

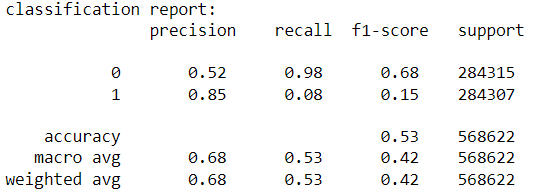
**Logistic Regression:-**

****

**Random Forest:-**

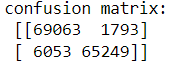
****

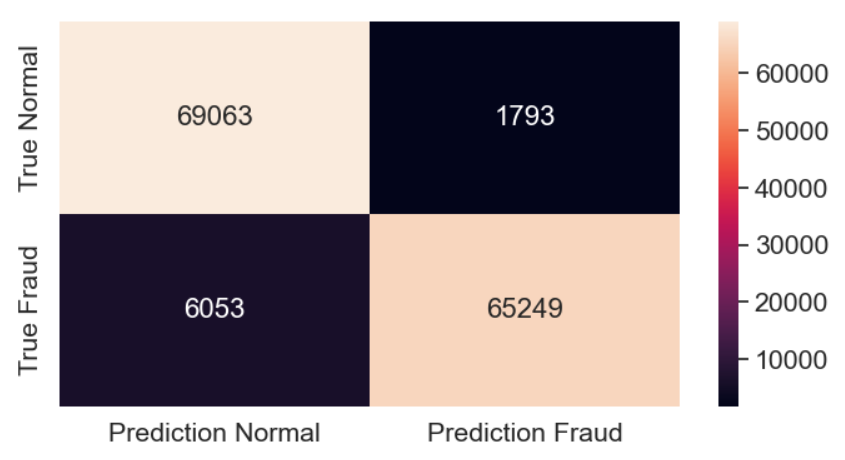
**K-means:-**

****

**5.6 Confusion Matrix Heatmap**

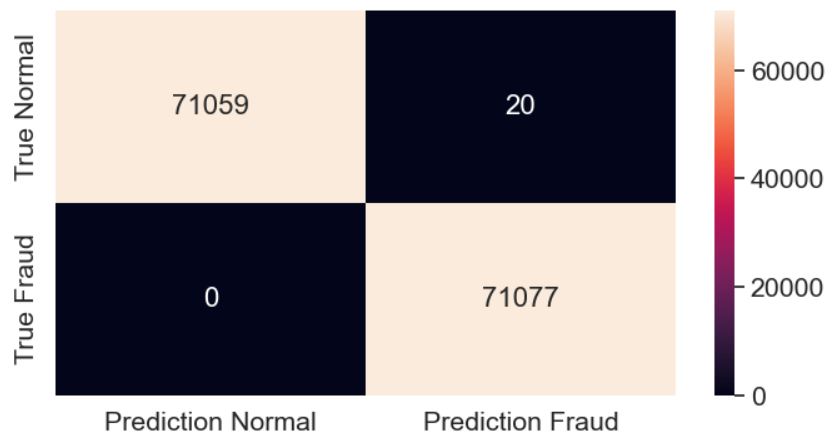
**Logistic Regression:-**

****

****

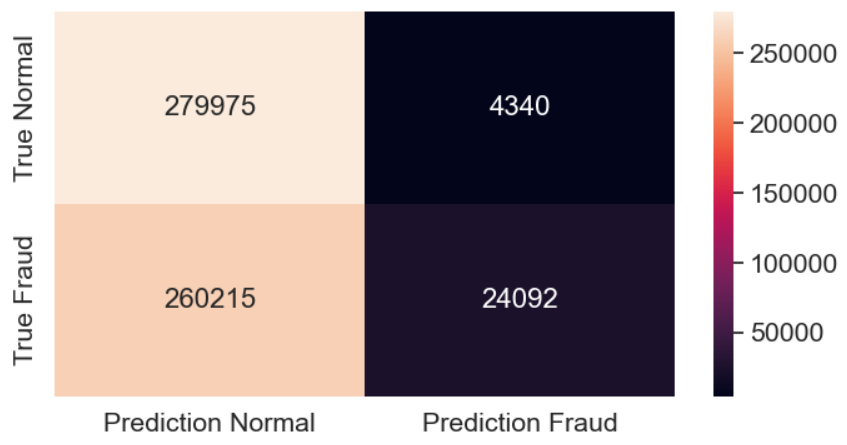
**Random Forest:-**

****

****

**K-means:-**

****

****

**CHAPTER 6: HARDWARE**

|  |  |
| --- | --- |
| Components | Specifications |
| Processor Brand | **‎Intel** |
| Processor Type | **‎Core i5** |
| Processor Speed | **‎2.50 GHz** |
| Processor Count | **‎1** |
| RAM Size | **‎8 GB** |
| Memory Technology | **‎DDR4** |
| Computer Memory Type | **‎DDR4 SDRAM** |
| Maximum Memory Supported | **‎24 GB** |
| Memory Clock Speed | **‎2496 MHz** |
| Graphics Coprocessor | **‎NVidia** |
| Graphics Chipset Brand | **‎NVIDIA** |
| Graphics Card Description | **‎Dedicated** |
| Graphics RAM Type | **‎GDDR4** |
| Graphics Card Ram Size | **‎4 GB** |
| Graphics Card Interface | **‎Integrated** |

**Table 6.1**

**CHAPTER 7:** **CONCLUSION & FUTURE SCOPE**

In Conclusion Credit card fraud becomes a serious concern to the world. Fraud brings huge financial losses to the world. This urged Credit card companies have been invested money to create and develop techniques to reveal and reduce fraud. The prime goal of this study is to define algorithms that confer the appropriate, and can be adapted by credit card companies for identifying fraudulent transactions more accurately, in less time and cost. Different machine learning algorithms are compared, including Logistic Regression, Random Forest and K-means clustering. Because not all scenarios are the same, a scenario-based algorithm can be used to determine which scenario is the best fit for that scenario.

In this project, we explored the use of different machine learning algorithms such as Logistic Regression, Random Forest, and K-Means Clustering for credit card fraud detection. We also used techniques such as SMOTE Tomek for sampling unbalanced data.

We found that the Random Forest algorithm performed the best with an accuracy of 99.98% and an F1 score of 0.99. However, it is important to note that the choice of algorithm and parameters may vary depending on the specific dataset and problem at hand.

**Future Scope:**

The future scope of credit card fraud detection using machine learning algorithms is vast and promising. Some of the potential areas of development and improvement are:

* Integration of more advanced machine learning techniques such as deep learning and neural networks to improve the accuracy of fraud detection.
* Exploration of real-time monitoring and detection systems that can analyse transactions in real-time and raise alerts for suspicious activities.
* Development of a unified fraud detection system that can be used across different industries and financial institutions.
* Integration of big data analytics to detect fraud patterns across a large volume of transactions.
* Use of blockchain technology to create a decentralized and secure platform for credit card transactions, making it difficult for fraudsters to tamper with the data.
* Adoption of a hybrid approach that combines rule-based systems and machine learning algorithms to improve fraud detection accuracy and reduce false positives.

Overall, the future of credit card fraud detection lies in the integration of technology and advancements in machine learning algorithms to develop more sophisticated and secure fraud detection systems.

**REFERENCES**

[1] Adepoju, O., Wosowei, J., lawte, S., & Jaiman, H. (2019). Comparative evaluation of credit card fraud detection using machine learning techniques. 2019 Global Conference for Advancement in Technology (GCAT).

[2] Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications, 11(12).

[3] Shiyang Xuan,Guanjun Liu,Zhenchuan Li,Lutao Zheng,Shuo Wang, Jiang,”Random Forest for credit card fraud detection”,Int.conf.on Networking,Sensing and control,2018.

[4] Y. Sahin , and Duman,E.,(2011) “Detecting credit card fraud by ANN and logistic regression.” In Innovations in Intelligent Systems and Applications(INISTA),2011 international Symposium on (pp.315-319).IEEE

[5] Y. Sahin, S. Bulkan, and E. Duman, “A cost-sensitive decision tree approach for fraud detection,” Expert Syst. Appl., vol. 40, no. 15,pp. 5916–5923, 2013

[6] Sahin Y. and Duman E. (2011),”Detecting Credit Crad Fraud by Decision Trees and Support Vector Machines”, International Multi-Conference Of Engineers and Computer Scientists(IMECS 2011),Mar 16-18,Hong Kong,Vol.1,pp.1-6

[8] A. Agarwal, ―Survey of Various Techniques used for Credit Card Fraud Detection,‖ Int. J. Res. Appl. Sci. Eng. Technol., vol. 8, no. 7, pp. 1642–1646, 2020, doi: 10.22214/ijraset.2020.30614.

[9] C. Reviews, ―a Comparative Study : Credit Card Fraud,‖ vol. 7, no. 19, pp. 998–1011, 2020.

[10] A. DalPozzolo, O. Caelen, and G. Bontempi, “When is undersampling effective in unbalanced classification tasks?” in Machine Learning and Knowledge Discovery in Databases. Cambridge, U.K.: Springer, 2015

[11] N. Mahmoudi and E. Duman, “Detecting credit card fraud bymodified fisher discriminant analysis,” Expert Syst. Appl., vol. 42, no. 5,pp. 2510–2516, 2015

[12] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Detecting credit card fraud using periodic features,” in Proc. 14th Int. Conf. Mach. Learn. Appl., Dec. 2015, pp. 208–213.

[13] Anuruddha Thennakoon, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga,”Realtime Credit Card Fraud Detection Using Machine Learning ,”Int. Conf. on Cloud Computing,Data Science & Engineering,2019.

[14] S. Wang, L.L. Minku, and X. Yao, “Resampling-based ensemble methods for online class imbalance learning.” Trans, Knowl., Data Eng., vol 27, no. 5, pp. 1356-1368, May 2015.

[15] R. R. Popat and J. Chaudhary, ―A Survey on Credit Card Fraud Detection Using Machine Learning,‖ Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018, vol. 25, no. 01, pp. 1120–1125, 2018, doi: 10.1109/ICOEI.2018.8553963.

[16] O. Adepoju, J. Wosowei, S. Lawte, and H. Jaiman, ―Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques,‖ 2019 Glob. Conf. Adv. Technol. GCAT 2019, pp. 1–6, 2019, doi: 10.1109/GCAT47503.2019.8978372.

[17] M. Deepa and D. Akila, ―Survey Paper for Credit Card Fraud Detection Using Data Mining Techniques,‖ Int. J. Innov. Res. Appl. Sci. Eng., vol. 3, no. 6, p. 483, 2019, doi: 10.29027/ijirase.v3.i6.2019.483-489.

[18] P. Save, P. Tiwarekar, K. N., and N. Mahyavanshi, ―A Novel Idea for Credit Card Fraud Detection using Decision Tree,‖ Int. J. Comput. Appl., vol. 161, no. 13, pp. 6–9, 2017, doi: 10.5120/ijca2017913413.

[19] J. Vimala Devi and K. S. Kavitha, ―Fraud Detection in Credit Card Transactions by using Classification Algorithms,‖ Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017, pp. 125–131, 2018, doi: 10.1109/CTCEEC.2017.8455091.

[20] R. R. Popat and J. Chaudhary, ―A Survey on Credit Card Fraud Detection Using Machine Learning,‖ Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018, no. Icoei, pp. 1120–1125, 2018, doi: 10.1109/ICOEI.2018.8553963.

[21] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, ―Random forest for credit card fraud detection,‖ ICNSC 2018 - 15th IEEE Int. Conf. Networking, Sens. Control, pp. 1–6, 2018, doi: 10.1109/ICNSC.2018.8361343.

[22] S. Mittal and S. Tyagi, ―Performance evaluation of machine learning algorithms for credit card fraud detection,‖ Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019, pp. 320–324, 2019, doi: 10.1109/CONFLUENCE.2019.8776925.

[23] X. Yu, X. Li, Y. Dong, and R. Zheng, ―A Deep Neural Network Algorithm for Detecting Credit Card Fraud,‖ Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020, pp. 181–183, 2020, doi: 10.1109/ICBAIE49996.2020.00045.

[24] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, ―Credit Card Fraud Detection using Pipeling and Ensemble Learning,‖ Procedia Comput. Sci., vol. 173, pp. 104–112, 2020, doi: 10.1016/j.procs.2020.06.014.

[25] R. San Miguel Carrasco and M.-A. Sicilia-Urban, ―Evaluation of Deep Neural Networks for Reduction of Credit Card Fraud Alerts,‖ IEEE Access, vol. 8, pp. 186421–186432, 2020, doi: 10.1109/access.2020.3026222.

[26] G. Kibria and M. Sevkli, ―Application of Deep Learning for Credit Card Approval : A Comparison with Application of Deep Learning for Credit Card Approval : A Comparison with Two Machine Learning Techniques,‖ no. January, pp. 0–5, 2021, doi: 10.18178/ijmlc.2021.11.4.1049.