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

**AALAP MISTRY**

**DE MONTFORT UNIVERSITY**

**ABSTRACT**

The Purpose of this article is to review various machine learning algorithms used for credit card fraud detection and to identify the best performing algorithm based on various parameters like AUC (Area Under Curve), Sensitivity, Specificity, Recall and Accuracy of Model. It uses credit card dataset available on Kaggle.

Furthermore, this article will help other researchers to predict credit card fraudulent customers.

Also, this research titled as “Credit Card Fraud Detection and is it worth Prediction using Machine Learning Algorithm “aims to check the credibility of various machine learning algorithms and also see the best performing algorithm. As per the dataset, it has around 30 features for prediction among which 28 are generated by the help of PCA and two others are amount and class. It was highly imbalanced dataset and dimensionality reduction technique and data balancing technique have to be used before data modelling. Hence SMOTE UP sampling technique has been used as part of data balancing and it was seen that after applying it, the accuracy was coming quite better. Also, Random Forest was the best performing algorithm among other traditional machine learning algorithms. It achieved a accuracy of around 99.98% which was quite good in compared to others. However ANN and Logistic was also performing good after SMOTE UP sampling.

We also performed exploratory data analysis to check the data with the help of some visualization like scatter plot and box plot.

As part of future enhancements, we can include a greater number of algorithms and other techniques to balance the dataset.

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| CHP NO. | TITLE | | PAGE NO. |
|  | **DECLARATION** | |  |
|  | **ACKNOWLEDGEMENTS** | |  |
|  | **ABSTRACT** | |  |
|  | **LIST OF ALGORITHMS** | |  |
| 1 | **INTRODUCTION** | |  |
|  | **1.1** | **OVERVIEW** |  |
|  | **1.2** | **RESEARCH QUESTION** |  |
|  | **1.3** | **AIM** |  |
|  | **1.4** | **OBJECTIVE** |  |
|  | **1.5** | **LIMITATIONS** |  |
|  | **1.6** | **PARAMETERS FOR PREDICTIVE MODELS** |  |
|  | **1.7** | **DISSERTATION ROADMAP** |  |
|  | **1.8** | **DISSERTATION CHALLENGES** |  |
| 2 | **LITERATURE REVIEW** | |  |
| 3 | **RESEARCH METHODOLOGY AND METHODS** | |  |
|  | **3.1** | **CRISP DM** |  |
| 4 | **DATA MODELLING** | |  |
|  | **4.1** | **K NEAREST NEIGBOURS (KNN)** |  |
|  | **4.2** | **DECISION TREE** |  |
|  | **4.3** | **RANDOM FOREST** |  |
|  | **4.4** | **SUPPORT VECTOR MACHINE** |  |
|  | **4.5** | **LOGISTIC REGRESSION** |  |
|  | **4.6** | **ARTIFICIAL NEUTRAL NETWORKS (ANN)** |  |
|  | **4.7** | **SMOTE UNSAMPLING** |  |
| 5 | **TESTING AND EVALUATION** | |  |
|  | **5.1** | **TOOLS USED** |  |
|  | **5.2** | **DATASET USED** |  |
|  | **5.3** | **DATASET URL** |  |
|  | **5.4** | **EVALUATION RESULTS** |  |
|  | **5.5** | **Final Output of All Model** |  |
| 6 | **DATA VISUALIZATION** | |  |
|  | **6.1** | **EXPLORATORY DATA ANALYSIS** |  |
| 7 | **CONCLUSIONS AND FURTURE WORK** | |  |
|  | **REFERENCES** | |  |

**LIST OF ALGORITHMS**

* Support Vector Machines (SVM)
* Logistic Regression
* Random Forest Classifier
* Artificial Neural Network

**Chapter 1**

**INTRODUCTION**

* 1. **OVERVIEW**

These days’ monetary organizations are getting lot of credit card frauds happening in the country. As lot of people are doing card transactions more in compared to cash transactions, henceforth these types of frauds are very commonly happening everywhere. Therefore, these monetary companies are looking for some credit card fraud detections software which can detect which is a genuine customer and which is a fraudulent customer. The main idea behind it is to save lots of customer money which is getting lost due to these frauds. Various algorithms have been built for these types of systems and various research has been also done on them. But still no successful software has been developed to identify fraudulent and non-fraudulent customers. Also, if credit card frauds won’t be identified then people will start using less credit cards and the benefit happening for these companies will decrease day by day. Therefore, there is a need to develop such types of credit card fraud detection system which can identify or distinguish between fraudulent and non-fraudulent customers.

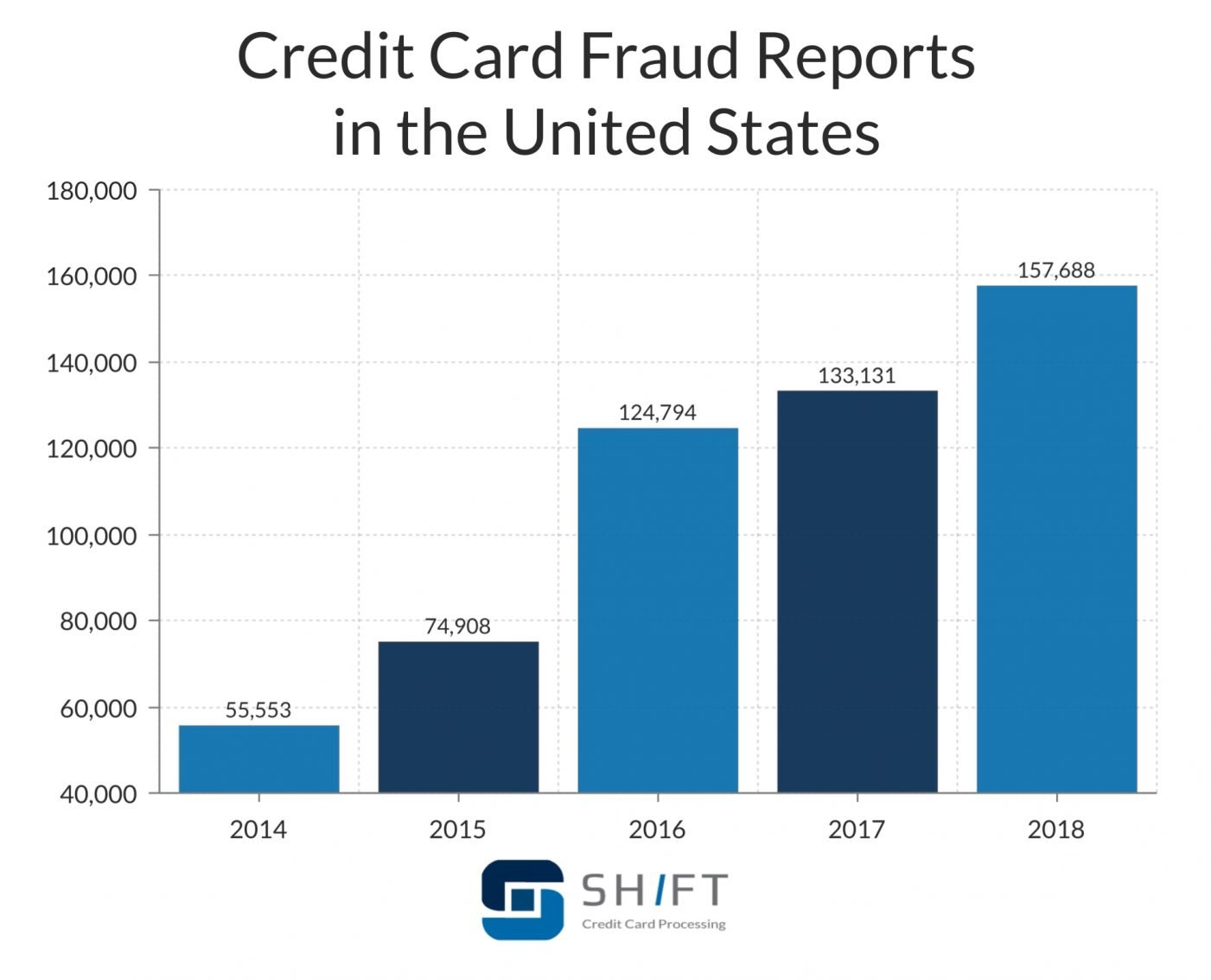


Fig: 1.1

The above picture shows the amount of credit cards frauds happened during various years. It is clear from the above picture the numbers of frauds are increasing day by day and if this is the case happening on then there will be less transactions through card and various monetary organizations may face various losses due to it.

* 1. **RESEARCH QUESTION**

Can Credit Card Fraud be detected with the help of various machine learning algorithms ?

* 1. **AIM**

Credit Card Fraud Detection with the help of various machine learning algorithms.

* 1. **OBJECTIVE**

Perform a study on credit card fraud detection by the help of various machine learning algorithms and check whether this kind of approach is successful in predict in credit card frauds.

* 1. **LIMITATIONS**
* The size of the dataset is too huge hence for performing machine learning algorithm I have to take a subset of data.
* Data is unbalanced hence I have to apply various down sampling and smote up sampling to balance the data.
* Data fields are not in correct name hence I have to perform data cleansing at the data preparation step.
* There are various limitations while generating plots in huge dataset.
  1. **PARAMETERS FOR PREDICITVE MODELS**

I choose following parameters while performing prediction of fraudulent customers.

1. **Credit History of Customer:** On the basis of credit history, I could easily identify customer behaviour and also check whether he is paying his bills or loans on time or not. If he does not have good score, then it may be the case that he is fraudulent customer.
2. **Credit Card Defaulter Customer:** If a person is credit card defaulter for many numbers of times then it may be the case that he is a fraudulent customer.
3. **Customer Purchase Behaviour:** If the purchase behaviour of customer is different from other regular customer then there may be a chance of fraudulent. For e.g. If a customer purchased lot of items in a single day till his credit limit and he does not do it frequently then there may a chance of fraud.
4. **Customer Income:** If the customer has a good and reliable income source then also there are less chance of frauds.

Hence there are various other parameters on which a credit card fraud customer can be identified based on various other parameters.

Below is a picture which shows various types of frauds of credit cards.

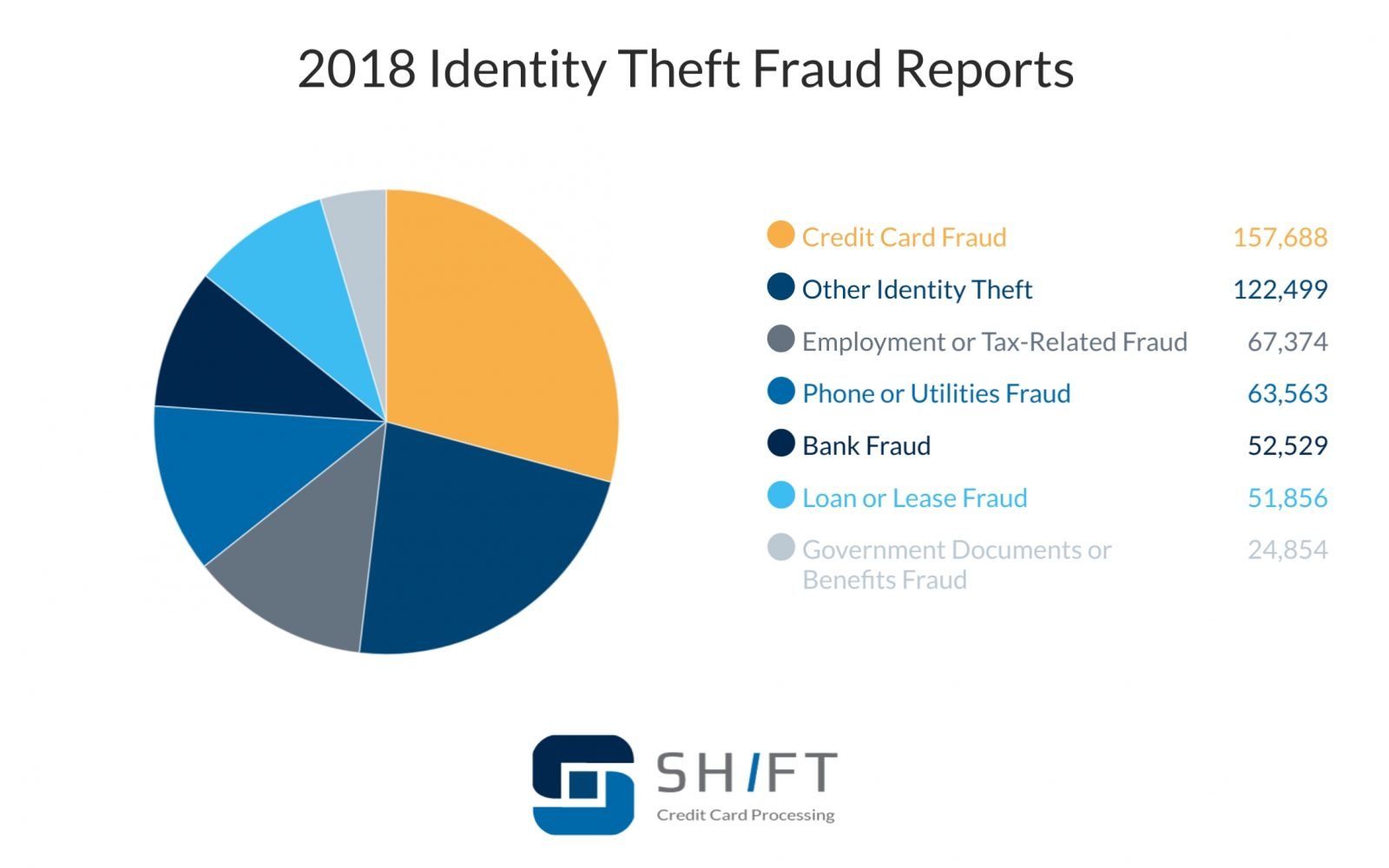


Fig: 1.2

Also, there is a case that certain age of customers is more affected by frauds as they are not much aware of these types of thefts as shown below:

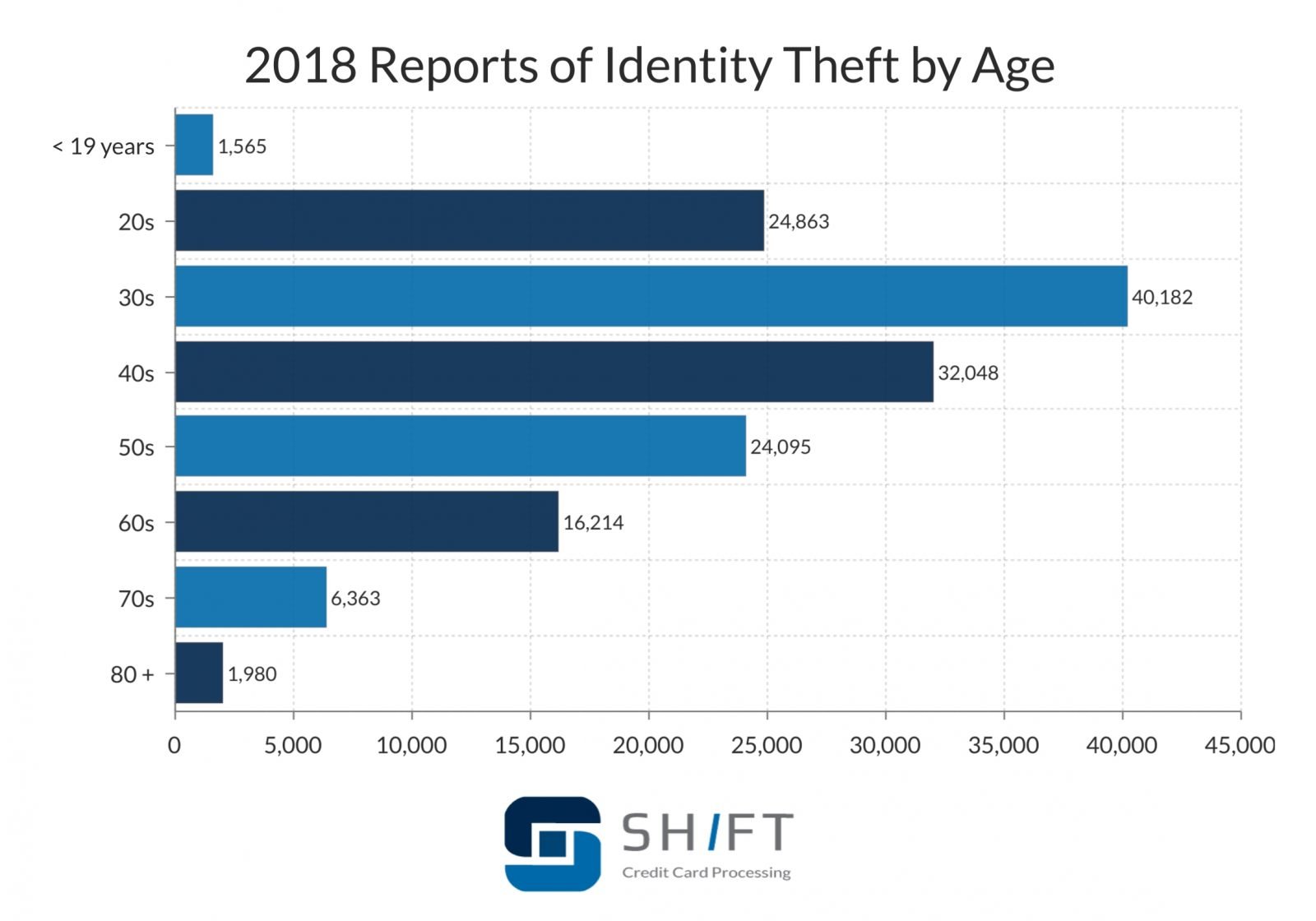


Fig: 1.3

* 1. **DISSERTATION ROADMAP**

Dissertation roadmap has been used which in further detail has been explained.

1. **Introduction:** This chapter includes the problem definition, research question, aim and objectives and the hypothesis to be tested.
2. **Literature** Review: This chapter highlights the existing research for time series forecasting with the use of research journals and books which include the theories, concepts and models of forecasting.
3. **Methodology:** This chapter utilizes the CRISP-DM approach to conduct the research, each of the six phases of the methodology are tailored for this research.
4. **Data Analysis:** The aim of this chapter is to compare the finding and the performance of each algorithms without concluding the finding.
5. **Discussions:** This chapter includes the interpretation of the algorithms' results, discussion of the findings and answering the research question.
6. **Conclusions:** This chapter will summarize the finding of the research to come to a conclusion.

I have covered all the above in my thesis in further in detail manner.

* 1. **DISSERTATION CHALLENGES**

Building a fraud detection system isn't as simple because it appearance. The professional person must determine that learning strategy to use (e.g., supervised learning or unsupervised learning), which algorithms to use (e.g., supplying regression, call trees, etc.), that options to use, and most

importantly, a way to wear down the category imbalance drawback (fraudulent cases ar very distributed as compared to the legitimate cases) Category imbalance isn't solely the most important concern in fraud detection system. Overlapping of the real and deceitful categories because of restricted info about the dealing records is another drawback within the classification task and most machine learning algorithms underneath perform under these situations [JS02].

In a real-life state of affairs, a fraud detection model predicts the character of sophistication (genuine or fraudulent) and gives the alert for the foremost suspicious dealing to the investigators. Investigators then perform an additional investigation and supply feedback to the fraud detection system to enhance its performance. However, this method will be AN overhead for the investigators because of that solely a few transactions ar valid on time by the investigators. In such a case, simply a couple of feedbacks are provided to the prognosticative model, that usually leads to a lesser correct model .Lastly, as money institutes terribly seldom disclose the client information to the general public because of confidentiality problems, the $64000 money datasets are terribly onerous to search out. this is often one in all the most important challenges in fraud detection analysis work.

**Chapter 2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION OF LITERATURE REVIEW**

There are various research which are performed on the credit card dataset and they use statistical machine learning algorithms like Random Forest(RF),Logistic Regression, K Nearest Neighbour (KNN),Decision Tree(DT). Also, some of the researches has been using Multilayer Perceptron (MLP) and ADA Boost kind of algorithms as well. Some of the research papers uses up sampling and under sampling approaches to perform research. I have gone through 10-12 research papers and I have found that none of the paper has done a comparison and checked whether machine learning algorithms are successful in predicting fraudulent customers.

**2.2 RESEARCH PAPER OVERVIEW**

**(Philip K. Chan et al, March 2019)** This paper used multi-classifier meta-learning approach to skewed class distributions, they used transactions from the first 8 months (10/95 - 5/96) for training, the ninth month (6/96) for validating, and the twelfth month (9/96) for testing. They used four algorithms (C4.5, CART, RIPPER, and BAYES) to each subset and generated 128 base classifiers. The results from this study showed that raining class distribution affects the performance of the learned classifiers and the natural distribution can be different from the desired training distribution that maximizes performance. Results also showed that that multi-classifier meta-learning approach using a 50:50 distribution in the data subsets for training can significantly reduce the amount of loss due to illegitimate transactions. Instead of learning one classifier using incremental learning, our modular multi classifier approach facilitates adaptation over time and removal of out-of-date knowledge.

**(Vaishnavi Nath Dornadula et al, Feb 2020)** This paper main aim was to develop a novel scam detection methodology for Streaming Transfer of Data, with an aim, to check the past transfer details of the consumers and elaborate the shape. Here customers are grouped using clustering techniques based on the amount of transaction done by them. Then using sliding window plan to average the transfer created by the customers from various clusters so that the shapes of the groups can be extracted respectively. Later different classifiers were trained over the cluster differently. Furthermore, the differentiator with better review run can be chosen to be one of the best methods to estimates scams. They used European credit card fraud dataset to perform this experiment.

**(S P Maniraj et al, Sep 2019)** In this paper, they mainly used analysis and data pre-processing steps and performed multiple anomaly detection using various algorithms such as Local Outlier Factor and Isolation Forest algorithm on the Principal Component Analysis (PCA) transformed credit card dataset. They used the latest machine learning algorithms to detect anomalous activities known as outliers. In their dataset, they had around 31 columns, out of which 28 holder sensitive data. Other columns represented such as Amount, Time and class of transaction. Time showed the gap between first and another transaction.

Amount is the amount of money transacted. Whereas Class 0 represents a valid transaction and 1 represents a fraudulent transaction. The results of this algorithms showed that, other algorithm does reach over 99.6% accuracy, its precision remains only at 28% because only 10% of data was used for it. However, when the entire dataset was fed into the algorithm, the precision rises to 33%.This high percentage of accuracy was expected due to the huge imbalance between the number of valid and number of genuine transactions.

**(Massimiliano Zanin et al, May 2018)** This paper used hybrid data mining techniques and network classification algorithms to detect illegal instances in a real card transaction data set. It is based on a recently proposed network reconstruction algorithm that allows creating representations of the deviation of one instance from a reference group. They showed he inclusion of features extracted from the network data representation improves the score obtained by a standard, neural network-based classification algorithm and additionally how this combined approach can outperform a commercial fraud detection system in specific operation niches. Beyond these specific results, this contribution represents a new example on how complex networks and data mining can be integrated as complementary tools, with the former providing a view to data beyond the capabilities. Results from the experiment showed that features extracted from a network-based representation of data, leveraging on a recently proposed par enclitic approach can play an important role: while not effective in themselves, such features can improve the score obtained by a standard ANN classification model. Furthermore, they illustrated that the network-based model is able to yield better results than a commercial fraud detection system.

**(Amir Hassan Monadjemi et al, March 2020)** This paper used two approaches:

Fraud Analysis(Misuse Detection) and user behaviour analysis(Anomaly Detection).In the first approach they used supervised machine learning approach to create classification models which can predict the state (normal or fraud) of new records. There are numerous model creation methods for a typical two class classification task such as rule induction, decision trees and neural networks .In the Second approach they used unsupervised machine learning approach which are based on account behaviour. In this method a transaction is detected fraudulent if it is in contrast with user’s normal behaviour. For this they extracted the legitimate user behavioural model (e.g., user profile)for each account and then detected fraudulent activities according to it. Comparing new behaviours with this model, different enough activities were distinguished as frauds. The profiles may contain the activity information of the account, such as merchant types, amount, location and time of transactions, [6].This method is known as anomaly detection. In This technique they used ANN (Artificial Neural Network Technique) for fraud detection. They also used Hidden Markov Model and Genetic Algorithm in this case. Varieties of measure were used to check the performance of models. The aim of all algorithms and techniques was to minimize FP and FN rate and maximize TP and TN rate and with a good detection rate at the same time.

**(Devika S P et al, July 2019)** This paper used online shopping transaction fraud detection to extract various BPs behaviour based on user transaction records. They used Markov Model and Genetic Algorithm for Fraud Detection. They proposed novel fraud detection method and it composed of four stages. In first stage they utilize the cardholder's transaction data, and they divided each cardholder's into different teams as the behaviour members in the same team are similar. And they proposed a strategy of window-sliding in each team to aggregate the transaction. In second stage they extract the collection of some special behavioural patterns based on transaction for each cardholder. In third stage they train set of classifiers for each team based on all their behavioural patterns. In the last stage, they used classifiers to identify online fraud, if any fresh transactions were fake, the procedure for detecting the drift idea and its issues will be adopted. Here they proposed a technique for solving an adaptive capacity to adjust its parameters to cardholder's timely actions.

This paper also used a Hybrid approach, named qualitative case-based reasoning and learning (QCBRL) .It combined three AI methods,

1. Case-based reasoning
2. Reinforcement learning
3. Qualitative spatial reasoning

In experimental results it showed that Some of the classifiers evaluated when model creation yielded higher accuracy for random tree and decision tree of around 94.32% and 93.50%.

**(Thulasyammal Ramiah Pillai et al, July 2019)** This paper uses multilayer perceptron for credit card fraud detection. They used various feature in multi-layer perceptron to compare the performance of MLP. In Compared to other papers used other statistical algorithms, they used deep learning methodology for credit card fraud detection. They found that logistic and hyperbolic tangent activation function offer good performance in detecting the credit card fraud. The logistic activation function performed better when there were 10 nodes, the sensitivity achieved was 82% and when there were 100 nodes, the sensitivity was 83% respectively in the 3 hidden layer model. However, hyperbolic tangent activation function performed better when there were 1000 nodes, the sensitivity was 82% in all the number (1, 2 and 3) of hidden layers.

**(S. Benson Edwin Raj et al, March 2011)** This paper presented a survey of various techniques which can be used in credit card fraud detection and performed evaluation of each methodology based on various criteria. They made use of methods like Fusion of Dempster Shafer and Bayesian Learning technique. They also used ANN (Artificial Neural Network, Bayesian Approach and Hidden Markov model BLAST, Fuzzy Darwinian System approached in credit card fraud detection. After application of them, they performed a comparative study on them. They compared models on parameters like Accuracy, Fraud Detection rate in FP (False Positives) and TP(True Positives) and cost of training. From the experimental results it was clear that Fuzzy Darwinian, Dempster and Bayesian theory has far greater accuracy in terms of FP and TP. In terms of processing speed, ANN performed much better in compared to other ones. Also fraud detection done by Hidden Markov Model was very less accurate In compared to other ones whereas Fuzzy Detection gave 100 % accurate results. Also the processing speed of BLAST-SSAHA was very good. ANN and BNN could easily detect Network illusion and phone fraud.

**(Hassan Najadat et al, Apr 2020)** This paper used various classification techniques such as Voting, Ada Boost, Naïve Bayes, Decision Tree (DT),Logistic Regression on IEEE-CIS Fraud Detection Dataset. Their dataset contains four files, training set of transactional data with 394 columns, train set with identity data having 41 columns, Test set with transactional data having 393 columns and test set with identity data containing of 41 columns. They merged the files of identity and transaction into single one which consisted of around 433 features in around 590540 records. However, they ignore those features which contained major null values (around 378 features). The dataset was highly imbalanced hence they used Random Up sampling, SMOTE up sampling and Random Under sampling to balance it. They made use of various evaluation metrics such as AUC (Area Under Curve), Precision, Recall and F1-Score to determine best model. As from the experimental results it showed that Best AUC was 80 % and 81 % which could be achieved by hard voting technique with over sampling and under sampling. Also, they used deep learning further to achieve more better results in compared to other models. They also made use of Bidirectional long Short-Term Memory (BiLSTM) with maximum layers of pooling. After following this approach, they further achieved AUC of 91.37%.

**(Wen Fang Yu et al, Apr 2009)** This paper used outlier detection technique on the basis of distance sum to detect credit card frauds and decide whether the fraud is genuine or not . It also worked in accordance with infrequency and unconventionality of credit card data fraud. They also made use of outlier mining for fraud detection. The results from this experiment also revealed that outlier detection was much more efficient in compared to other statistical algorithms who didn’t used it. They used real credit card data of a commercial bank for research. Among 16,584 transactions,1449 were detected as fraudulent and around 15135 were non fraudulent. Amon 58 attributes,28 attributes were decided to choose for detection based on feature selection technique. They achieved accuracy of around 89.4 %. Experiment results also showed that this algorithm was more effective to predict malevolent frauds. Also, it was seen that outlier mining could detect fraud much more in a better way in compared to anomaly detection clustering technique.

**(Anuruddha Thennakoon et al, Jan 2019)** This paper focuses on four main fraud criteria in real world applications. Each and every fraud can be identified by a series of various statistical machine learning models and they can be chosen in the evaluation stage. In their evaluation study, they provided a efficient guide to choose best optimal algorithm on the basis of various performance measure. They also performed real-time credit card fraud detection. In this, they took help of predictive analytics and also used an API module to decide on whether a transaction done is fraudulent or genuine. They also developed a strategy to check the data skewness distribution. They used data from a financial institution for their experiments.

**(Dejan Varmedja et al, March 2019)** This paper used Random Forest, Logistic Regression, Naïve Bayes and Multilayer Perceptron algorithms. Since the dataset was highly imbalanced, hence they applied Smote Up sampling for balancing of the data. The results showed that all the algorithms can be used for detection of fraud customers with high accuracy. The algorithms can be used for other irregularities as well.

**2.3 SUMMARY**

Below is a table which shows year of research paper and author name and algorithms used.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sn No.** | **Author Name** | **Year of Publish** | **Algorithms Used** |
| 1 | **Dejan Varmedja et al** | March 2019 | Random Forest, Logistic Regression, Naïve Bayes, MLP |
| 2 | **Anuruddha Thennakoon et al** | Jan 2019 | API Module |
| 3 | **Wen Fang Yu et al,** | Apr 2009 | Outlier Detection Technique |
| 4 | **Hassan Najadat et al** | Apr 2020 | Voting, Ada Boost, Naïve Bayes, Decision Tree(DT),Logistic Regression |
| 5 | **S. Benson Edwin Raj et al** | March 2011 | Dempster Shafer and Bayesian Learning |
| 6 | [**Thulasyammal Ramiah Pillai**](https://ieeexplore.ieee.org/author/37085585238) **et al** | July 2019 | Multilayer Perceptron(MLP) |
| 7 | **Devika S P et al** | July 2019 | Markov Model and Genetic Algorithm. |
| 8 | **Amir Hassan Monadjemi et al** | March 2020 | Rule induction, decision trees and neural networks |
| 9 | **Massimiliano Zanin et al** | May 2018 | Neural network-based classification algorithm |
| 10 | **S P Maniraj et al** | Sep 2019 | Local Outlier Factor and Isolation Forest algorithm on the Principal Component Analysis(PCA) |
| 11 | **Vaishnavi Nath Dornadula et al** | Feb 2020 | Sliding window strategy |
| 12 | **Philip K. Chan et al** | Mar 2019 | (C4.5, CART, RIPPER, and BAYES) |

**Chapter 3**

**EXPERIMENTAL METHODS**

There are various machine learning algorithms which are used for credit card determination for fraudulent and non-fraudulent customers. I am using few supervised and unsupervised algorithms in my thesis Detailed explanation of each algorithm is provided as below:

The research was done by the help of CRISP DM (Cross-Industry Standard Process for Data Mining) process. It is a standard process which has six phases which is described further in detail. It helps to guide and see the business requirements, set objectives, get optimized results and perform testing afterwards to see whether the model is performing well or not.

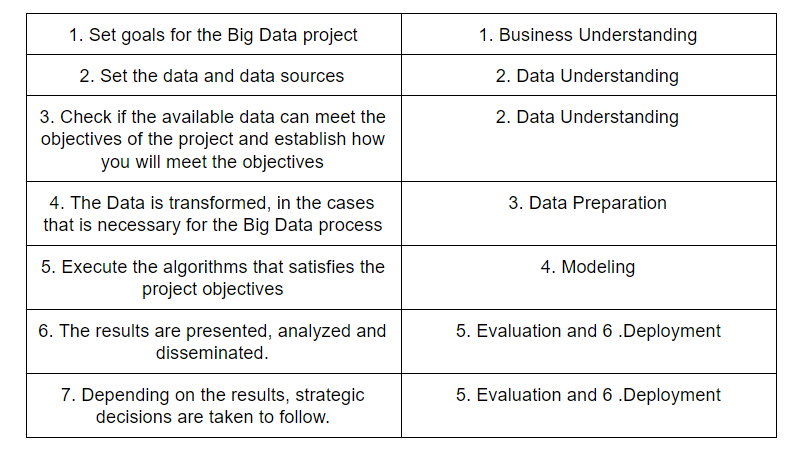
**3.1 CRISP DM PROCESS**



Fig: 3.1

This model is associate degree perfect sequence of events. In observe several of the tasks usually will be performed in a very totally different order and it'll often be necessary to double back to previous tasks and repeat bound actions. The model doesn't try and capture all attainable routes through the information mining method.

CRISP DM Process in details is explained as below:



The Initial stage is Business Understanding, and its objective is to provide context to the goals and to the info in order that the developer connects to knowledge in this explicit model.

It is composition of conferences, on-line conferences, document reading, specific field learning, and a protracted list of the way they assist the event group, create query on related context.

The consequence of this stage is that the event group gets the circumstance of the project. The vision of the project thought to be outlined before the initial project begins. as an example, developer team thought to understand by currently that target is to extend sale, and when the step is completed, perceive what's the consumer merchandising and the way they sell it.

The second stage is Data and its sources, and its objective is to grasp what is predicted and got from the info. It confirms the standard of the info, in many ways, like knowledge achieved, values segregation, knowledge compliance.

It’s an important a part of the project as a result of it tells however practical and trustable is the ultimate outcomes. During this stage, group people conceptualize on a way to separate the simplest price of the items of knowledge. In case, the employment or connection of some piece of knowledge is unclear to the event team, they'll momentarily step back, to grasp the business and the way it advantages from that piece of knowledge.

On account of this stage knowledge somebody currently power, on terms of knowledge, the result ought to fulfil the objectives of the project, what algorithmic program and method bring that result, however, is that the present stage of the info, and the way it ought to be, so as to be helpful to the algorithmic program and method concerned.

The third step is Data Preparation and includes the ETLs or ELTs method that transforms the items of knowledge into one thing helpful by algorithms and method.

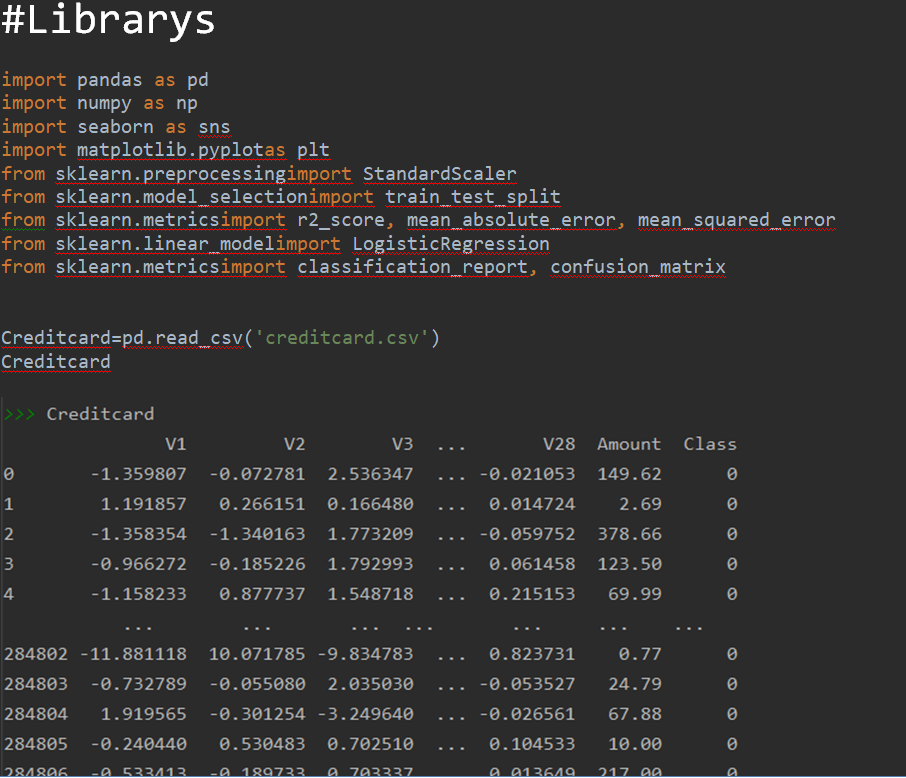
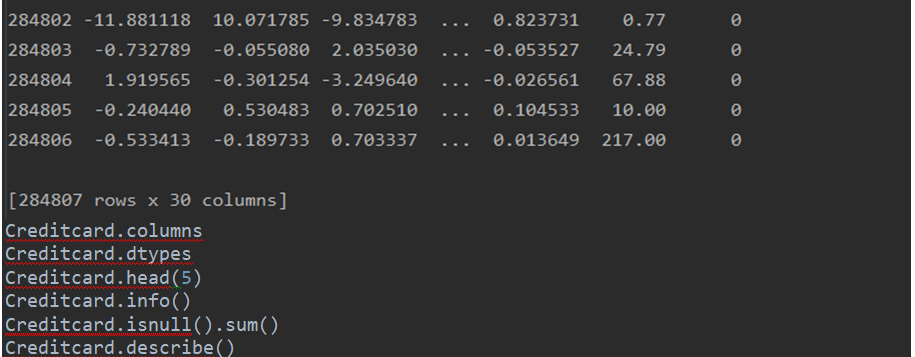
At times knowledge administrative strategics don't seem to be revered or set in a company, and so as to provide true aspiring to knowledge, it becomes knowledge engineers and knowledge scientists’ job to standardize the knowledge.

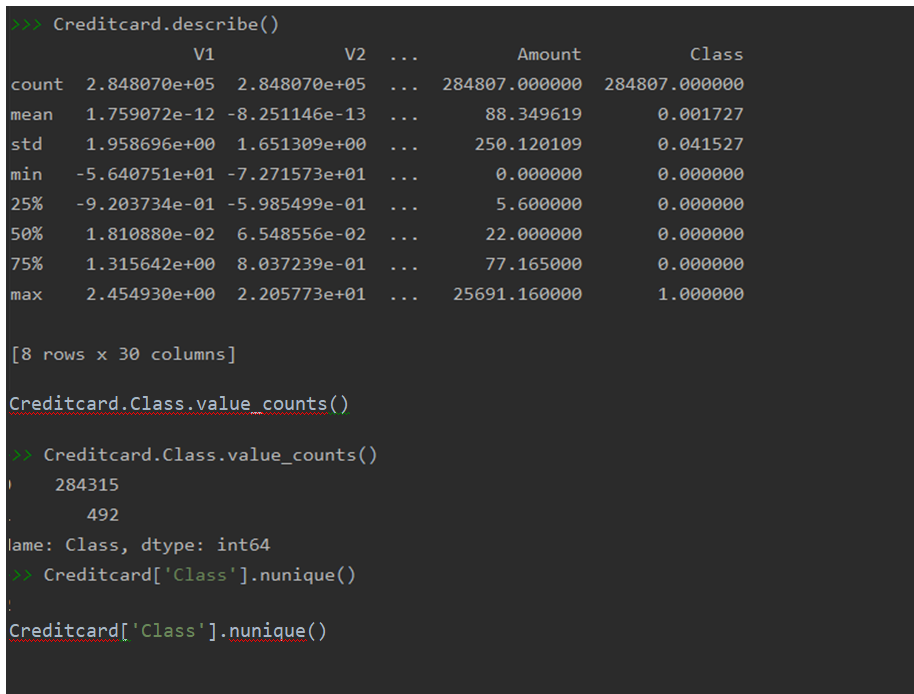
In like manner, few set of programs perform higher beneath sure boundaries, somebody doesn’t settle someone doesn't agree to no-mathematical qualities, others don’t work comfortable with an oversized variance on values. Then again, it's up to the event team to standardize data..

The greater part of the comes invested the majority of their energy in this progression, there’s Associate in Nursing IT profile decision knowledge engineer. As is long, that may get very complicated once operating with giant amounts of knowledge, IT departments might realize a bonus in devoting assets to explicitly play out these obligations.

Following is the example at data preparation step:

We are trying to understand the nature of credit card dataset by the use of various functions as describe etc.



Here we are checking that the dataset has 284807 rows and features are 30 to perform prediction.

Also we are doing a check on the data types of the columns, checking if there are any nulls values by the help of is null function. Using Describe function to describe the columns.

We can also see the mean, median, standard deviation types of statistics in various features.

All these V1-V28 features are generated by the help of PCA.

**PCA (Principal Component Analysis):** It is unsupervised technique majorly used for dimensionality reduction and help us to perform prediction in a efficient manner.

A picture of PCA can be shown as below:

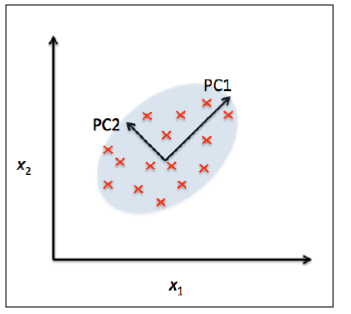


Fig: 3.2

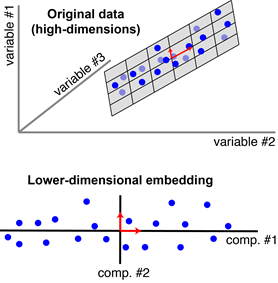


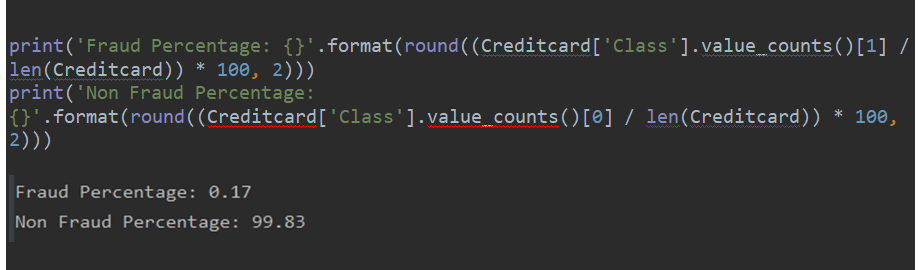
Fig: 3.3

Each Blue point is the observation, and the principal components dimension has been reduced from three to two. In our dataset V1-V28 features were generated and downloaded from Kaggle after PCA.

**PCA Limitations:**

1. **Model performance:** PCA will result in a discount in model performance on datasets with no or low feature correlation or doesn't meet the assumptions of dimensionality.
2. **Classification accuracy:** Variance primarily based PCA framework doesn't contemplate the differentiating characteristics of the categories. Also, the knowledge that distinguishes one category from another could be within the low variance parts and will be discarded.
3. **Outliers:** PCA is additionally stricken by outliers, and standardization of the info has to be an important part of any advancement.
4. **Interpretability:** every principal part could be a combination of original options and doesn't leave the individual feature importance to be recognized.

Furthermore, we are also performing a quick analysis of data by checking the fraud and non-fraud percentage as shown below:



From above we can see that the number of fraud transactions are just 17% while non fraud are 99.83 %. Hence after doing a initial analysis of data, it looks clearly that it is highly imbalanced dataset. Hence in further steps we applied balancing techniques like SMOTE up sampling etc.

The fourth step is Modelling and is that the core of any machine learning project. This step is accountable for the results that ought to satisfy or facilitate glad the project goals.

despite is that the exciting a branch of the project, it's also the most limited as expected, as though everything past is finished appropriately, there's very small to regulate. In this condition, the results area unit corrigible, the method is ready to stay back to knowledge readiness and increase the obtainable knowledge.

Related set of programs like k-means, hierarchic clump, statistic, simple regression, k-nearest neighbours, Associate in Nursing quantity several many others, area unit the inner code lines of this stage within the method. Modelling has been explained in detail further.

The next step is Testing and validation wherever it's up to check that the venture area unit limited and proper. just in way the results area unit might not be right, the method performs the rating back to the primary stage, so as to grasp why the results area unit misguided.

actually, on an information research project, the info somebody, separate the info between coaching and test. On this step the testing knowledge is employed, its main objective is to validate that the model (product of the modelling stage) is correct to the truth.

Contingent on the assignment and therefore the thusly, there are a unit numerous technique. as an unit various strategy, on the context of supervised learning, with the task of classifying things, a method to verify the results is with the confusion matrix. For unattended learning, to form analysis becomes tougher, as there's no static price to separate “correct” from “incorrect”, as an example, the task of classifying things would be evaluated by calculative the lay and intra distance between components in a(some) cluster(s).

In any case, it's vital to specify some supply of wrong live. This error live tells the user however they will need confidence within the results, either for: “for positive this can work” or “for positive it won’t”. If somehow the error live happens to by zero or none for all cases, it'd indicate that the model it’s over fit, and reality may perform otherwise. It has been explained in detail further in it.

The next step is Deploy, and it consists of gift the ends up in a helpful and comprehensible way, and by achieving this, the project ought to attain its targets. It's the sole stage not happiness to a cycle.

Depends on the ultimate user a helpful and comprehensible manner may vary. as an example, if the ultimate user is different piece of software package, as within the sales web site set of steps asking its recommendation system what to recommend for a customer, a helpful manner would be a JSON carrying the response to a selected question. In another case, sort of a big businessman UN agency needs projected info for deciding, the simplest manner to gift the findings is to store then in Associate in Nursing analytical information and gift them as a dashboard on a business intelligence resolution.

**CHAPTER 4**

**DATA MODELLING**

There are various algorithms which are applied; few of them are explained in detail as below:

**4.1 K NEAREST NEIGBOUR (KNN)**

K- It is one of the only machines learning algorithm which comparable with supervised learning.

KNN algorithm predicts the common between the new data & accessible data and put new data into accessible classes.

K-NN stores all accessible data and classifies in different category and replace with dataset.

This creates a new data set and classifies into a well class by victimization of K-NN algorithm

It’s often use for regression, it’s may be non- parametric algorithm which suggest it does not create any prediction on dataset, it’s also known as lazy learner .

KNN algorithm simply store the information set and once get new data then its create and replace with new data like the new data set.

Example: A example is shown as, suppose there are two categories as A and B . And we have to identify the new data point at which category it belongs to, then it can be easily identified with the help of KNN algorithms. As it checks the nearest neighbour based on Euclidean distance feature as shown below:



Fig: 4.1

**Euclidean Distance:** It can be derived by following formulae :



Fig: 4.2

**Advantages:**

* It is simple to implement and build.
* It is good for noisy data.
* For larger datasets its is more effective to build and simple.

**Disadvantages:**

* We need to identify value of K all the time which is a time taking activity.
* Implementation cost is high because of calculation of Euclidean distance calculation.

**4.2 DECISION TREE**

There are various classification technique to make classification model from input dataset.

Examples,

* Call tree classifier
* Rule base classifier
* Neural networks
* SVM support vector machine
* Naïve mathematician classifiers square

Above all the technique adopts a learning rules programme to identify a perfect model that most perfect for our requirements and relationship between attribute set and label data set.

So the we train our model and predict for choose right model and give accurate result and predict the category label. Here , representation

Nodes- options of the dataset

Branches- choice rules

Leaf node- result

Decision tree have two types of nodes

1. **Leaf node** : output of the decision nodes
2. **Decision node**: to make any decision or final result or come to conclusion based on define criteria

It’s a graphical example for obtaining all the potential solution to our problem in given condition

That’s why it’s known as choice tree,

Its start with the root then expands to branch and built the tree structure.

It is known as a choice tree because of, like a tree, it starts with the root node that expands on any branches and constructs a tree-like structure.

To make a tree, we tend to use the CART rule that stands for Classification and Regression Tree rule.

A decision tree merely asks a matter, and supported the solution (Yes/No), it any split the tree into sub trees.

General Structure of Decision tree with decision and leaf node is as below:



Fig: 4.3

Decision Tree Classifier may be a easy and wide used classification technique. It applies a plan to resolve the classification problem statements. Decision Tree Classifier poses a series of fastidiously crafted questions about the attributes of the record. Every time it receives a solution, a follow-up question is asked till a conclusion concerning the class(Y or N) label of the record is reached.

Build a optimum decision tree is essential drawback in decision tree classifier. In general, decision trees are created from a given set of attributes. Whereas a number of the trees area unit additional correct than others, finding the decision tree is computationally unfeasible due to the exponential size of the search area.

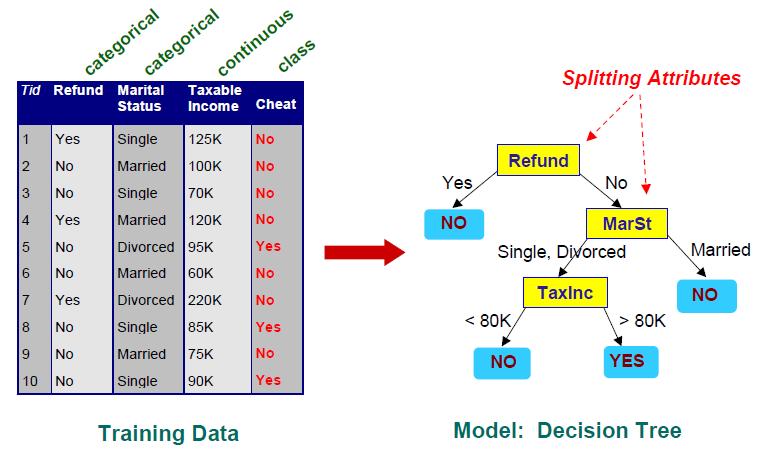


Fig: 4.4

**Advantages:**

* It is simple to understand and build structure wise. It follows the same human like decisions.
* It can be used to think about all the possible values as output based on some criteria.
* Less data cleansing is required in compared to other algorithms.
* Less Computational cost

**Disadvantages:**

* It has data over fitting issue which can be resolved at random forest.
* Complex to build as multiple layers required.
* If there are more number of labels then complexity of computation may increase.

**4.3 RANDOM FOREST**

Random forest also called as supervised learning that is employed for each classification and in final we get regression of that. Its specially used for classification issues. Also in this lots of tress are created and combine to gather and make a sturdy forest.

Based on the knowledge sample random forest formula create tree.so that prediction from every result became easy and choose most effective result and choose among them.. It’s Associate in Nursing ensemble methodology that is healthier than one call tree as a result of it reduces the over-fitting by averaging the result.



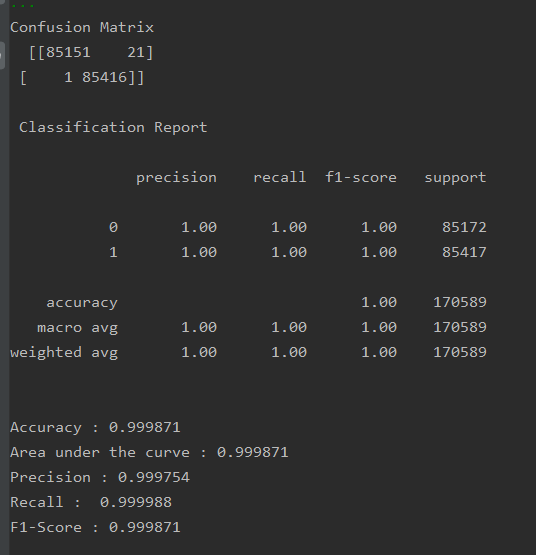
**Advantages:**

* Random Forest can be used for both Regression and Classification problems.
* It can handle large datasets with good accuracy.
* It does not have over fitting issue and also improves performance of model.
* It takes less training time in compared to other algorithms.
* It can also predict more accuracy even if more data is missing.

**Disadvantages:**

* It can be used for both classification and regression problems, but it does not suit More for regression problems.
* It is complex to build.
* It takes more computation complexity and have performance issue.

Random forest has been applied as below :



**4.4 SUPPORT VECTOR MACHINE**

Support Vector Machine or SVM is one in every of the foremost in supervised Learning algorithms, that is employed for Classification in addition as Regression issues. However, primarily, it's used for Classification issues in Machine Learning.

The goal of the SVM algorithmic program is to make the most effective line or call boundary that may segregate n-dimensional area into categories in order that we will simply place the new points of data within the correct class within the future. This best call boundary is termed as hyper plane.

SVM chooses the intense points/vectors that facilitate in making the hyper plane. These extreme cases square measure referred to as as support vectors, and thence algorithmic program is termed as Support Vector Machine. Below diagram within which their square measure 2 totally different classes that square measure classified employing a call boundary or hyper plane:



Fig: 4.5

This algorithm can be used for various applications such as Text Categorization, Face Recognition etc.

There are two types of SVM as shown below:

**Linear SVM:** Linear SVM is employed for linearly severable datasets, which suggests if a dataset is classified into 2 categories by employing a single line, then such knowledge is termed as linearly severable datasets, and classifier is employed known as Linear SVM classifier.

**Non-linear SVM:** Non-Linear SVM is employed for non-linearly separated datasets, which suggests if a dataset can't be classified by employing a line, then such knowledge is termed as non-linear knowledge and classifier used is named as Non-linear SVM classifier.

Example of SVM is as below:

 Fig: 4.6

Here we first train the model with lots of images of cats and dogs to the system and further test it with different creature image. As SVM has a decision boundary set between these two sets of cats and dogs ,where it chooses extreme cases which is known as support vectors . On the basis of these extreme vector values it decides on whether the image is a cat or dog and identifies that it is cat.

**Hyper plane:** There will be multiple lines/decision boundaries to segregate the categories in n-dimensional house; however we want to seek out out the simplest call boundary that helps to classify the information points. This best boundary is thought because the hyper plane of SVM.

The dimensions of the hyper plane rely upon the options gift within the dataset, which implies if there are a pair of options (as shown in image), then hyper plane are going to be a line. And if there are three options, then hyper plane are going to be a 2-dimension plane.

We forever produce a hyper plane that encompasses a most margin, which implies the most distance between the information points.

**Support Vectors:** The data points or vectors that are the highest to the hyper plane and that have an effect on the position of the hyper plane are termed as Support Vector. Since these vectors support the hyper plane, therefore referred to as a Support vector.

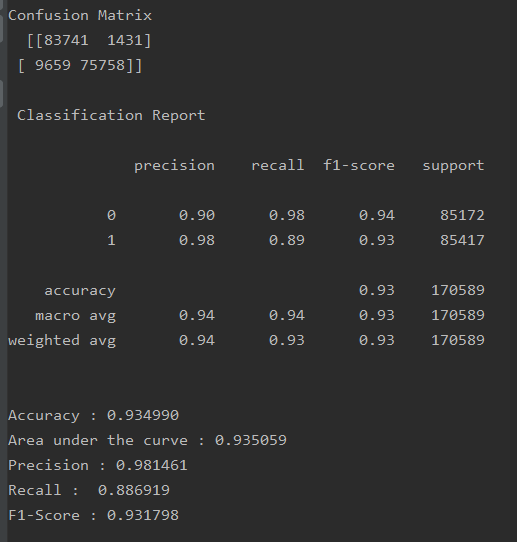
**Advantages:**

* It can be used for both classification and regression problems.

**Disadvantages:**

* Training data is required for the system as it is supervised learning model.

SVM has been applied on credit card dataset as below:



**4.5 LOGISTIC REGRESSION**

Logistic Regression is one of the most famous supervised learning techniques. It uses a set of independent features to predict the output feature. It is used for classification whereas liner regression is used for regression problems.

It analyses the outcome of a categoric value quantity. So the outcome should be a categoric or segregate. It is often binary like, 0 or 1, true or not true, etc. however giving the precise worth as zero and one, it offers the probability values that comes under zero and one.

It is way totally like the regression toward the mean except that however they're used.

In provision regression, rather than fitting a curve, we have a tendency to match AN "S" formed provision operate, that predicts 2 most values (0 or 1).

The graph from the provision operate tells the chance of one thing like climate or not the cells area unit danger or not, a mouse is weighty or not defined its heavyness, etc.

Logistic Regression could be a important machine learning algorithmic program as a result of it's the flexibility to exchange chances and derieve new knowledge victimisation continue and separate datasets.

Logistic Regression are often wont to classify the observations victimisation differing kinds of information and may simply verify the foremost effective variables used for the classification.

Here is a image which shows Logistic function :



Fig: 4.7

Logistic regression uses prediction technique for regression. It is further used to classify the sample data hence it is known as classification technique too.

**Sigmoid Function :**

* It is also known as logistic function. It generally map the probability of values to its actual values,
* It converts one value to another value in 0, 1 range.
* Its value can’t go beyond 0, 1 limit. It forms a S curve which is known as Sigmoid Function.
* Here we work on the basis of threshold values. If value is below threshold value then it is 0 otherwise its value is 1.

**Assumptions:**

* The dependent value should be categorical in nature and independent value should not be multi collinear.

Logistic Regression determination of value:

Logistic Regression in Machine Learning

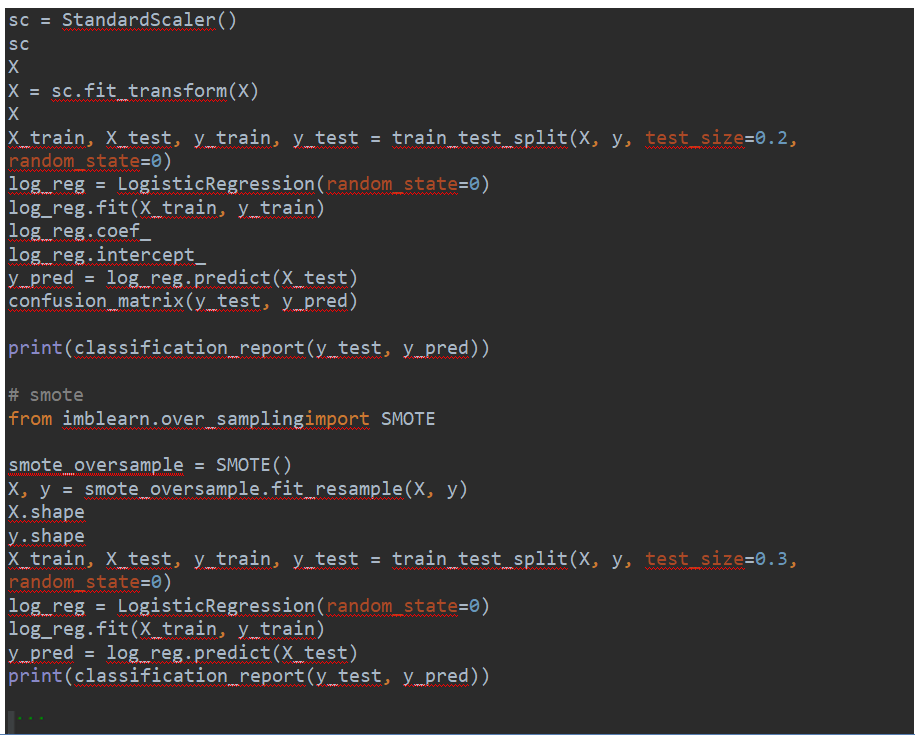
Types:

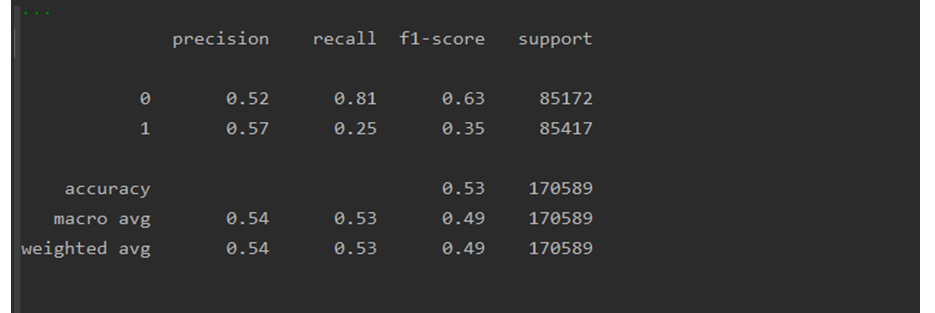
Binomial: Output value should be binary in nature either 0 or 1.

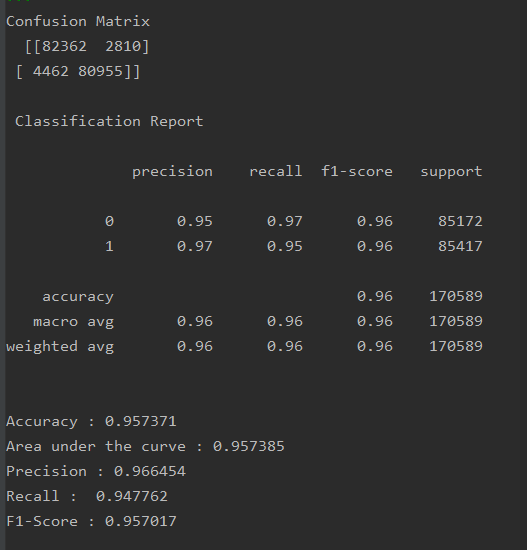
Multinomial: Output value can be multinomial in nature like types of animals etc and value can be three or more unordered dependent features.

Ordinal: Here the value can be three or more ordered dependent features.

Logistic regression on credit card dataset has been applied as below:







**4.5 ARTIFICIAL NEURAL NETWORKS (ANN)**

Artificial neural networks are some sort of model whose functionality is similar to human brain. It has more complex structures and also uses hidden layer technique. There are various of ANN such as feed forward ANN, Radial Based ANN etc. They are calculated by the help of mathematical equations and some values of parameters to determine the output.

ANN’s are around for years, however their study and use has been usually restricted to only one or even a pair of hidden layers. A neural network with multiple hidden layers is named a Deep Network and learning with a deep network is named, Deep Learning. Since hidden layers provide additional complicated issues and therefore additional advanced learning.

A picture of ANN is shown below:

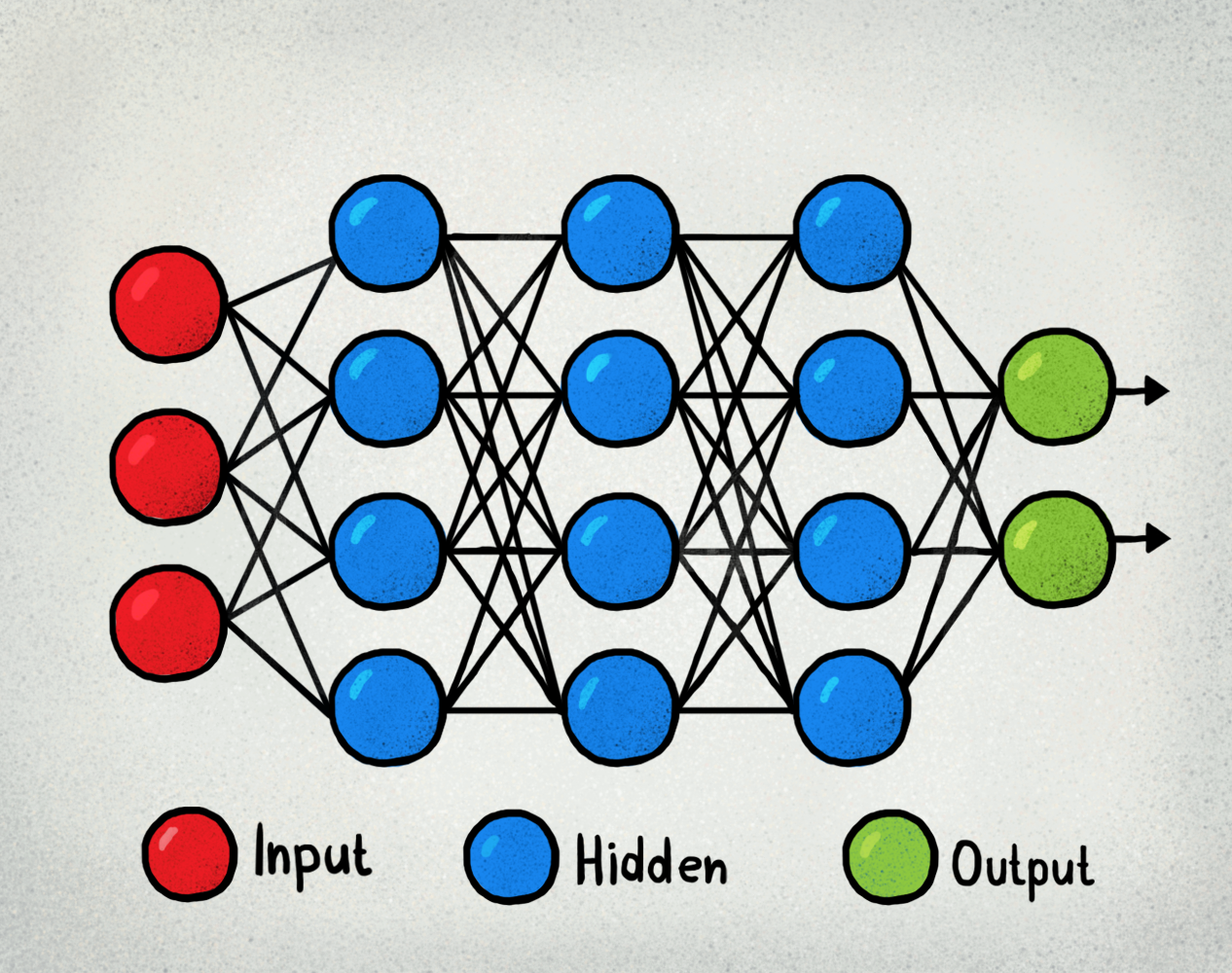


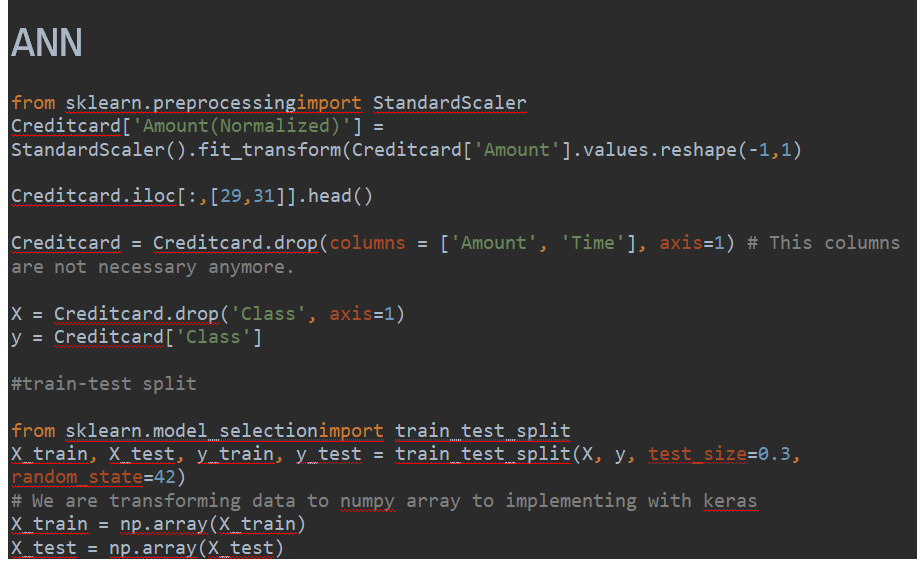
Fig: 4.8

ANN has been applied in credit card dataset as below:

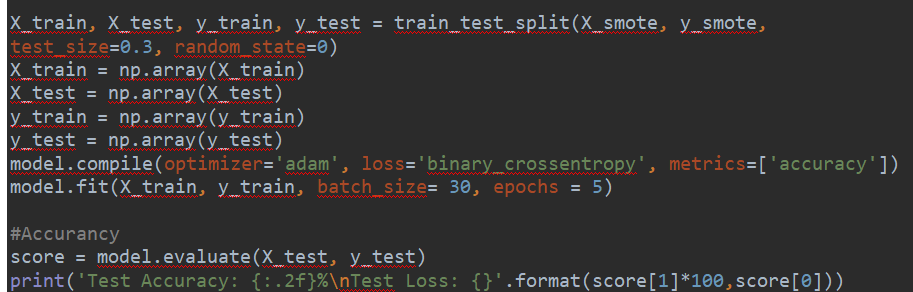
We have used standard scalar library for it.

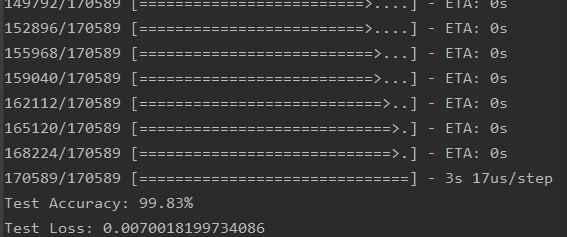
Few of the columns like Amount and Time does not correspond in the prediction hence we are dropping these columns.

Afterwards splitting data between training and test data set in the ratio of 70-30.

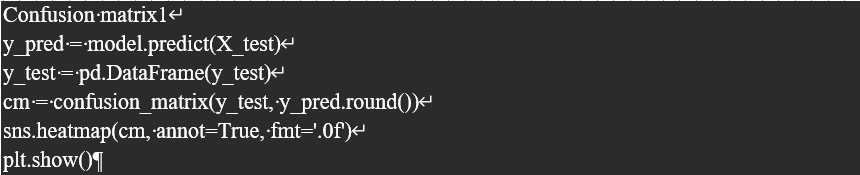
****



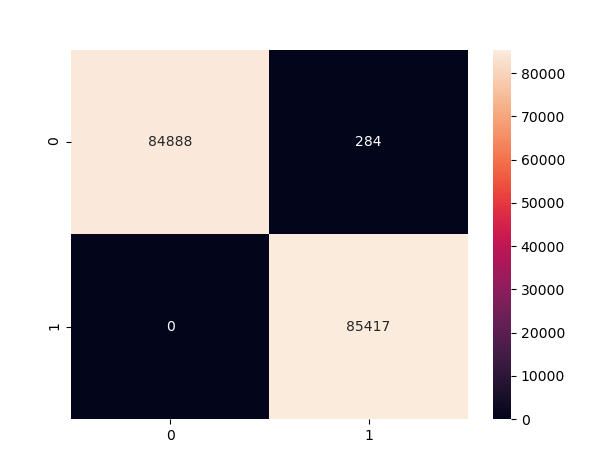
****

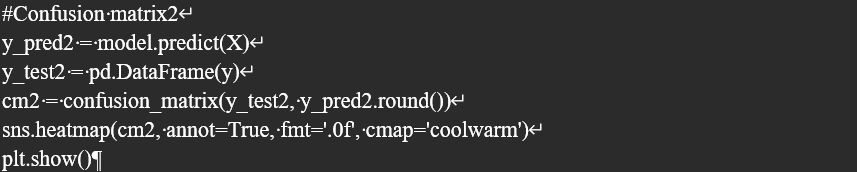
****

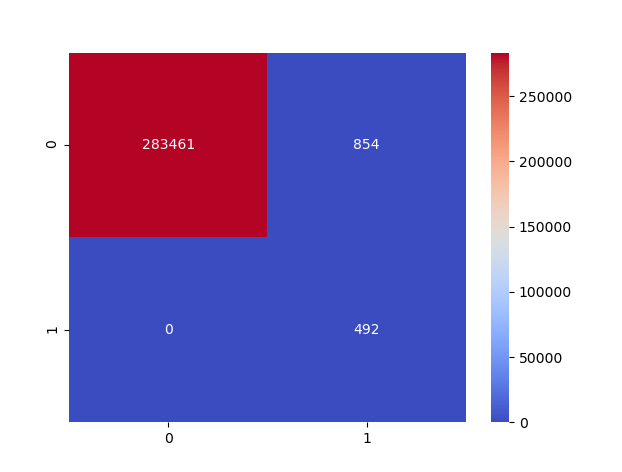
ANN Accuracy:

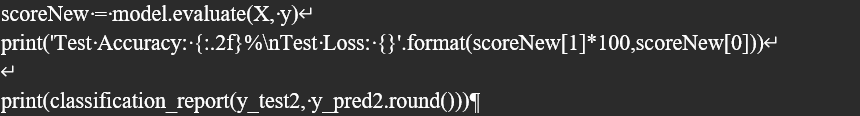


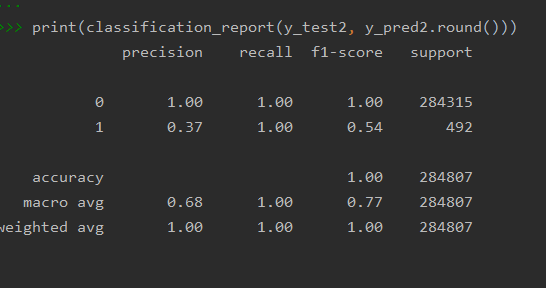
ANN Confusion Matrix:









****

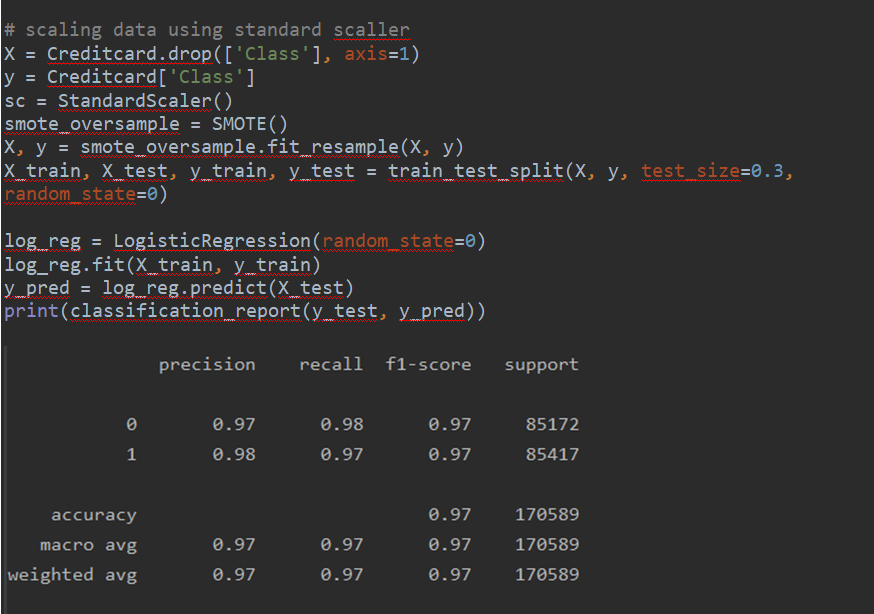
**4.6 SMOTE UNSAMPLING**

SMOTE (Synthetic Minority Oversampling Technique) – Oversampling

SMOTE (synthetic minority oversampling technique) is one in all the foremost normally used oversampling strategies to unravel the imbalance downside.

It aims to balance category distribution by arbitrarily increasing minority category examples by replicating them.

SMOTE synthesises new minority instances between existing minority instances. It generates the virtual coaching records by linear interpolation for the minority category. These artificial coaching records square measure generated by arbitrarily choosing one or additional of the k-nearest neighbours for every example within the minority category. Once the oversampling method, information |the info |the information} is reconstructed and a number of other classification models is applied for the processed data. The dataset has 492 fraud transactions out of total 284,807 number of transactions, from this it can be concluded that dataset was highly unbalanced hence it need some kind of balancing of dataset. Below is a example of it where I applied SMOTE up sampling technique.



**CHAPTER 5**

**EVALUATION AND TESTING OF MODELS**

Here we will compare the results obtained fro various models and decide which is the best performing model among all based on various criteria such as Model Accuracy, sensitivity ,specificity ,Recall etc.

**5.1 TOOLS USED**

PyCharm: Used for programming in dataset, data cleansing, data preparation and checking accuracy.

**5.2 DATASET USED**

I have used credit card dataset available in kaggle which holds data of number of transactions made by European cardholders in the month of Sep 2013.It generally contains transactions of around two days where total around 284,807 transactions were made within two days, among which 492 were reported as fraud transactions. This is highly imbalanced dataset hence I have to apply smote up sampling technique to balance it first and then perform machine learning on it. Frauds are around .0172 % of total number of transactions made within two days.

It generally contains data numeric in nature only and this result is obtained from PCA ( Principal Component Analysis).All the features correspond to principal component values which we get after PCS transformation. Some features like Amount and Time is not transferred after PCA. Amount holds the value of the transaction amount. Output variable is 1 or 0 which corresponds to fraudulent transaction or not.

**5.3 DATASET URL**

The dataset URL is as below:

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

**5.4 EVALUATION OF RESULTS**

We will be performing various results obtained from various algorithms here and decide on which algorithm is performing best in terms of accuracy, Recall Feature etc.

* Accuracy : It is the amount of accuracy which can be calculated as below :

Image for post

As we can see that accuracy can be calculated by the division of Number of Correct Predictions by Total number of Predictions made.

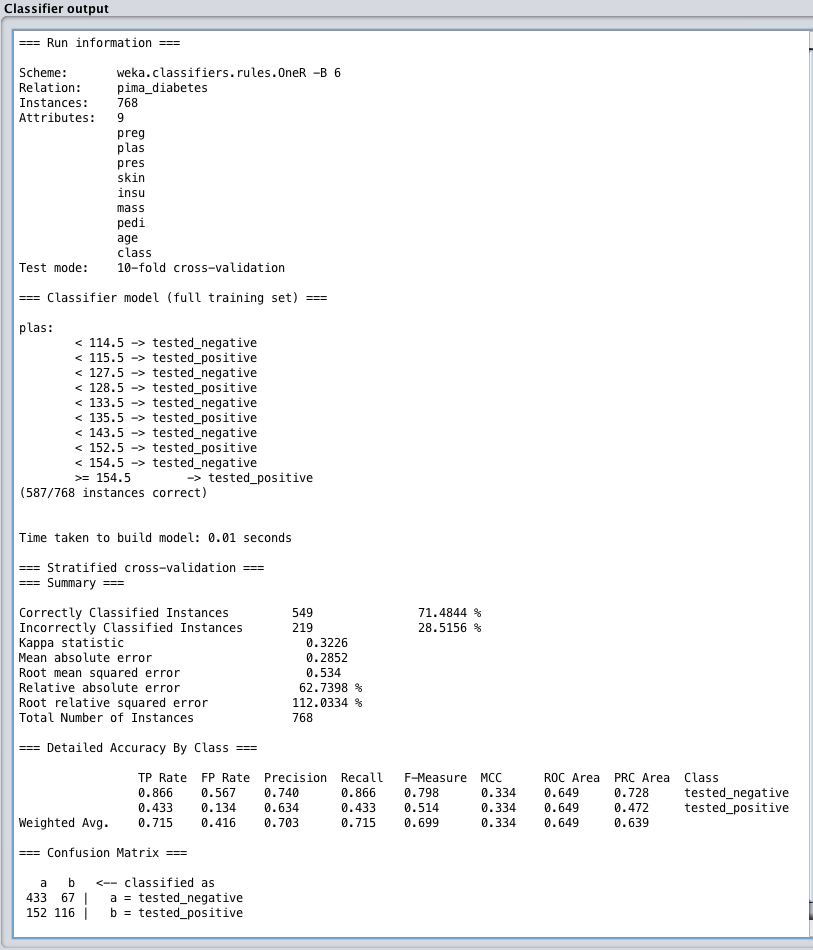
The more accurate the algorithm is more is its accuracy.

Classification accuracy is most famous these days. We divide our dataset into training and test dataset then perform machine learning in it and check the classifications accuracy if it’s a classifications problem. If accuracy is good then we can choose it, if not then we have to choose another algorithm. It can be achieved as 100 % or can be less depending on the size and complexity or balancing of dataset. If dataset is unbalanced, then we have to balance it first before modelling.

Confusion Matrix: It is another important parameter to check the algorithm.

It is a table which shows the number of predictions verses Actual ones.

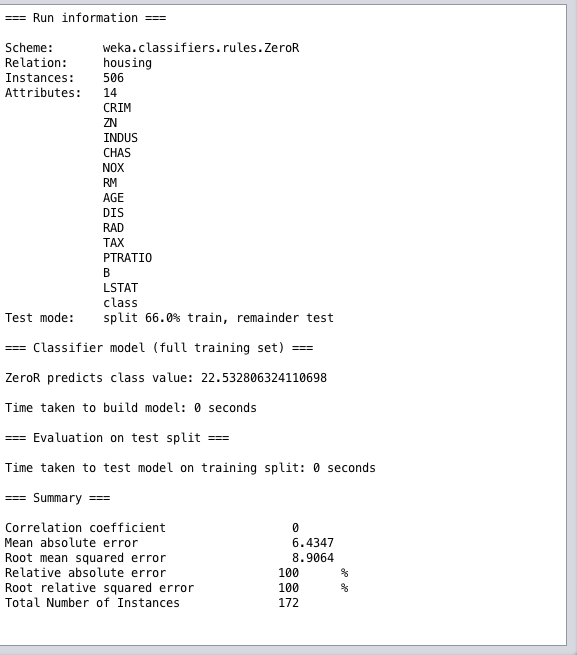
A example of it is as below:



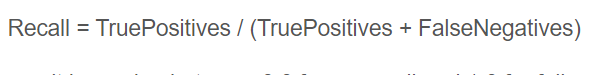
**Correlation Coefficient:** It tells how good features are correlated between each other. Value ‘0’ shows that it is least correlated whereas ‘1’ shows that it is highly correlated. It is used in regressions as it shows which features to take on for prediction of output variable.

**Root Mean Squared Error:** It is also used for regression as it shows the amount of error made on the testing dataset. It tells how the predictions has went good or bad. If RMSE if big then predictions didn’t want good and if it’s less then predictions went good.

An example of it as below:



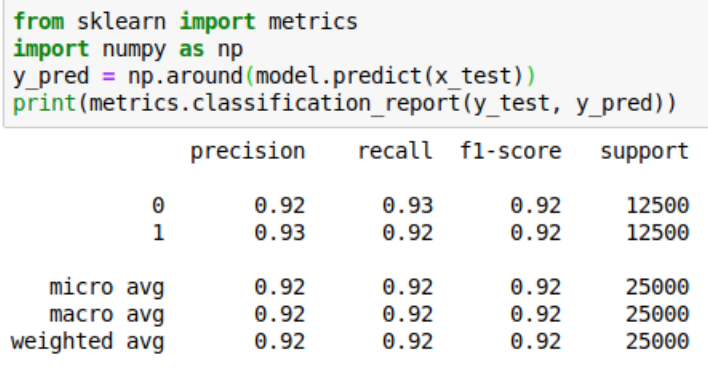
Recall : It can be calculated as below :



It is number of True positive divided by the total number of TP and FN. Recall value can vary between 0 to 1.0

Classification report It displays the feature values such as F1,Recall,Precision for the model.

An example of it as below:



Precision: It is also another important parameter to check the model. Precision is that the ability of a classifier to not label it as positive in Nursing instance positive that's really negative. For every category, it's outlined because the magnitude relation of true positives to the aid of a real positive and false positive. It can be calculated as below:



F1 Score: It is another important parameter to check the model performance.

The F1 score could be a weighted mean value of preciseness and recall such the simplest score is one.0 and therefore the worst is zero.0. F1 scores are below accuracy measures as they imbed preciseness and recall into their computation. As a rule of thumb, the weighted average of F1 ought to be wont to compare classifier models, not world accuracy.

It can be calculated as below :



**5.5 Final Output of All Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Area under the curve** | **Precision** | **Recall** | **F1-Score** |
| Logistic Regression | 0.957371 | 0.957385 | 0.966454 | 0.947762 | 0.957071 |
| Support Vector Machine | 0.934990 | 0.935059 | 0.981461 | 0.886919 | 0.931798 |
| Random Forest | 0.999871 | 0.999871 | 0.999754 | 0.999988 | 0.999871 |
| Artificial Neural Network | 99.83 | 99.83 | 1.00 | 1.00 | 0.54 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Area under the curve** | **Precision** | **Recall** | **F1-Score** |
| Logistic Regression | 0.957371 | 0.957385 | 0.966454 | 0.947762 | 0.957071 |
| Support Vector Machine | 0.934990 | 0.935059 | 0.981461 | 0.886919 | 0.931798 |
| Random Forest | 0.999871 | 0.999871 | 0.999754 | 0.999988 | 0.999871 |
| Artificial Neural Network | 99.83 | 99.83 | 1.00 | 1.00 | 0.54 |

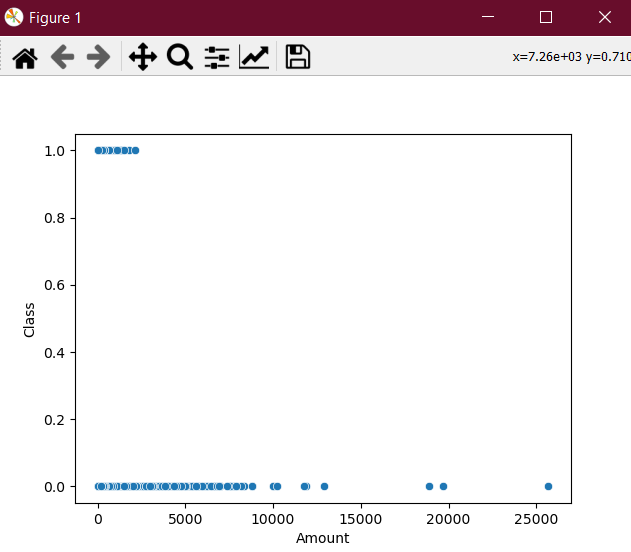
**CHAPTER 6**

**DATA VISUALIZATION**

6.1 Exploratory Data Analysis

Before starting with machine learning I wanted to do some exploratory data analysis for which I created some visualizations to check the quality of data. Below are some of visualizations:

1. Below is a scatter plot designed between Amount and Class features.

**** Fig: 6.1

From the above scatter plot it is clear that the number of customers who belong to class ‘0’ are far greater than the number of customers with class ‘1’.

1. Below is a box plot developed between Amount and Class Features.

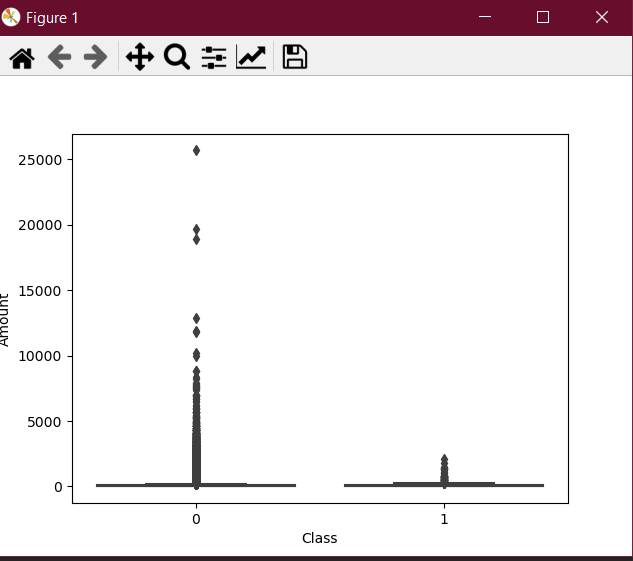


Fig: 6.2

From above box plot it is clear that most amount of data belongs to class ‘0’ and less amount of data belongs to class ‘1’.

1. Below is a scatter plot between features class and credit card.

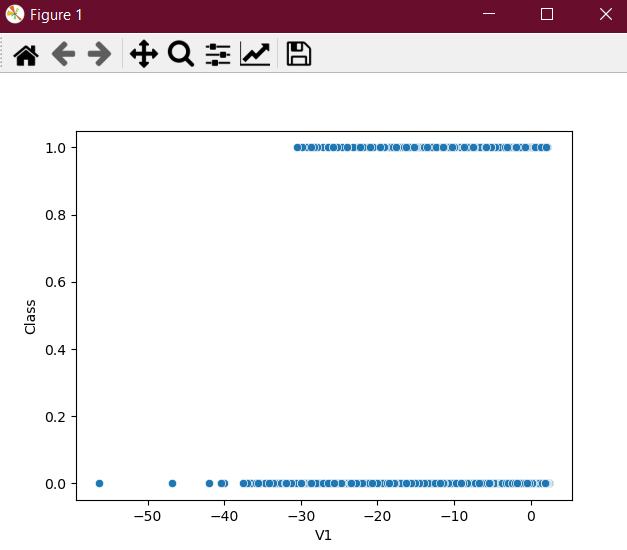


Fig: 6.3

From the above scatter plot it is clear that that number of credit card transactions which occurred in class ‘0’ are more in compared to other transactions that belong to class ‘1’. Hence from this we can understand that most of the transactions were genuine, but there were few fraud transactions as well.

1. Below is a scatter plot showing the plot between number of credit card transactions and Amount.

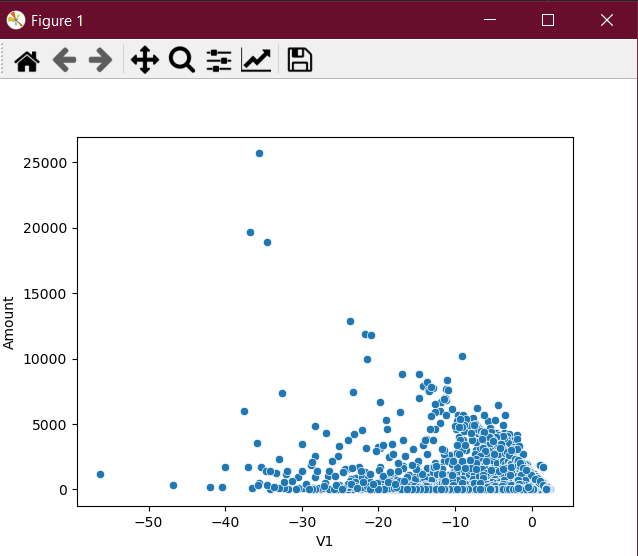


Fig: 6.4

From the above scatter plot it is clear that the number of credit card transactions for more amount was very less in compared to the number of transactions where the amount was less.

It can be also said that most of the number of transactions lied between 0 to 5000 amounts and comparatively less between 5000 to 10,000 and very less between 10,000 to 15000. Furthermore, it can be decided that only one transaction happened of around 25000.

It tells the customer behaviour that due to the more frauds happening, the number of transactions has been decreased a lot these days.

1. After seeing the diagram, it is clearly visible that most of the values are near to zero and almost unconnected. Moreover, a strong connection associated with bright square.

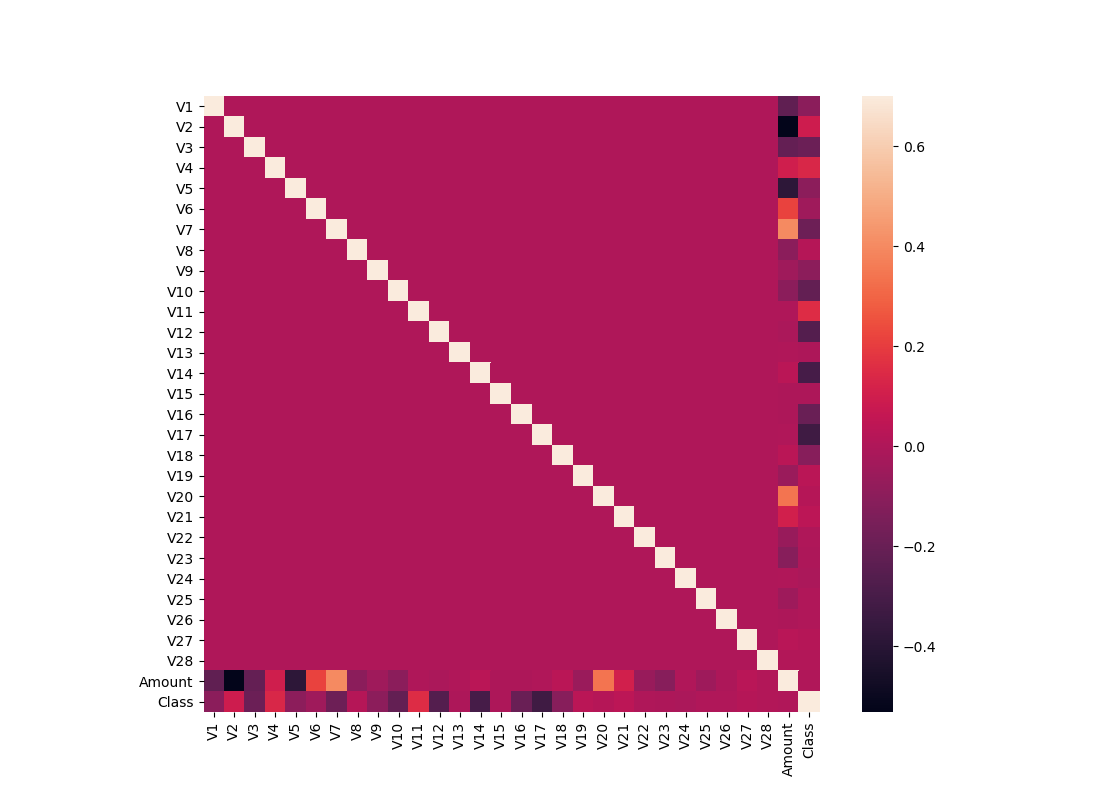


Fig: 6.5

6) The chart demonstrates that almost V’s are assemble around zero with few or no outlines. In our class histogram, it is remarkable that we have founded mostly logical cases rather than fraud.

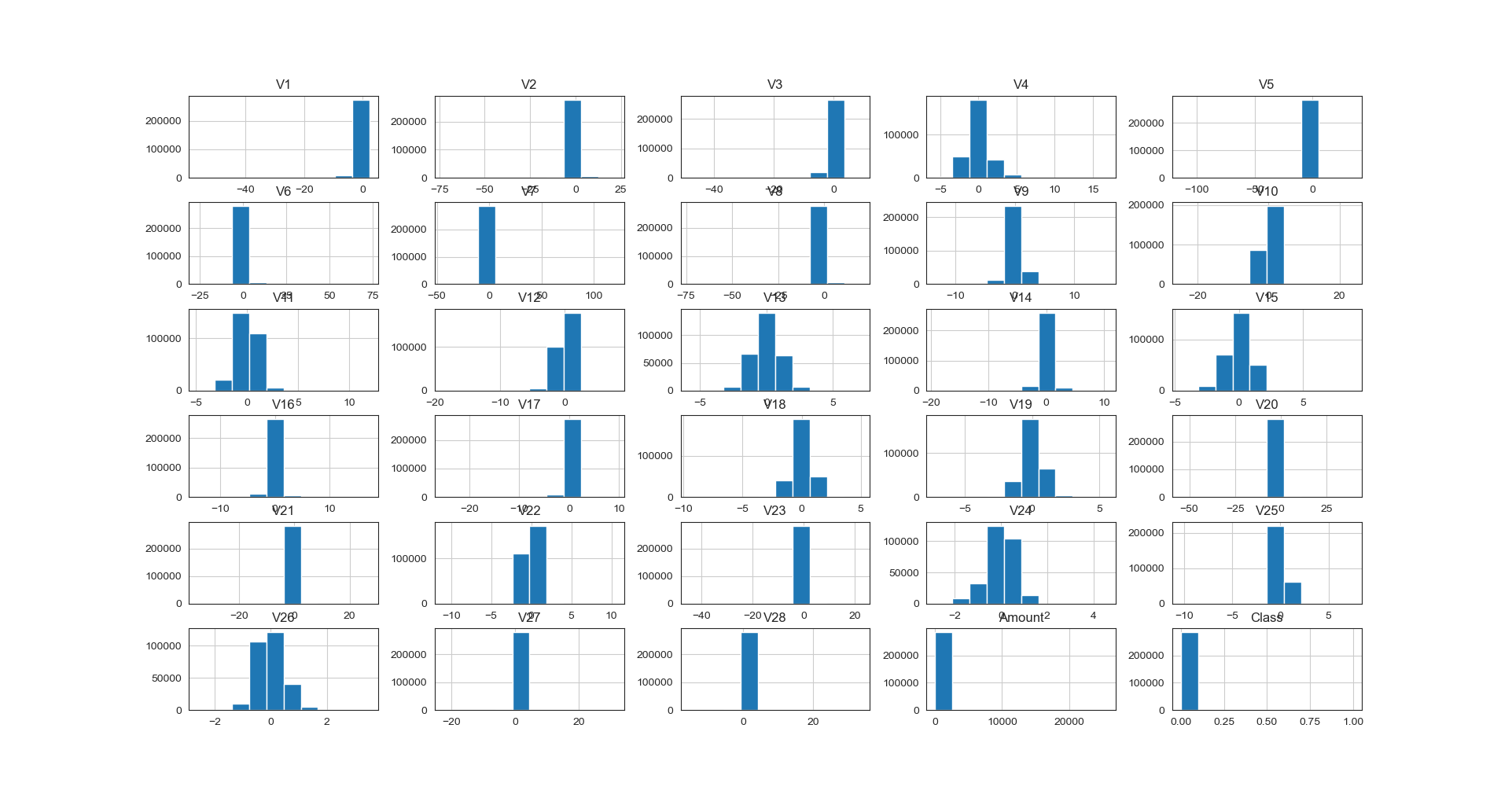


Fig: 6.6

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

**7.1 CONCLUSION**

After Comparing all algorithms, we checked that machine learning is performing good and we can detect fraudulent customers by the use of various machine learning algorithms .However there are few challenges as data is highly unbalanced and also the dataset is very huge. We need a big query to evaluate it properly or we need some king of data balancing to balance the dataset before performing machine learning. However, we made use of SMOTE UP sampling to balance the dataset and also took a subset of data to perform machine learning. However, we also dropped few of the columns which didn’t made contribute in machine learning.

When we checked various algorithms accuracy then we saw that Random Forest is performing much better in compared to other algorithms. Random Forest achieved a accuracy of about 99.98 % which is far better However Logistic algorithm and ANN also performed quite good but the best accuracy was achieved by Random Forest.

From the Confusion Matrix of ANN also it was clear that Class ‘0’ was identified correctly 84888 times while incorrectly identified around 284 times which is quite less in compared to correctly identification.

After performing this research, it can be concluded that working with highly imbalanced dataset, after balancing the data, Random Forrest was the best performing algorithm in terms of accuracy. It can be also concluded that machine learning was performing very good to predict fraudulent customers from credit card dataset.

**7.2 FUTURE WORK**

As part of future work enhancements, we can include a greater number of algorithms like Deep Learning etc which will further let me to see how deep learning is performing in credit card dataset. As we can see that ANN was a good performing algorithm hence Deep Learning can be included as a future enhancement. Also, I have limited the number of records to perform machine learning. We can make use of big data technologies in it and perform machine learning in a more efficient manner.

Below are the future enhancements which can be made in it.

* Include more algorithms like Deep Learning, Gradient Boosted Trees etc .
* Include Big Query and make use of Big data technologies to handle a big data.
* Add More visualizations to understand more about the data.
* Can include Rapid Miner kind of tools to perform data mining.

Some steps can be used while trying to make a good prediction system as below:

* Accurately find the number of fraudulent customers based on the available features like Amount, Class and other features obtained after performing PCA.
* Then test dataset on the testing set.
* Check the best performing model in terms of accuracy, Recall and other parameters.
* Also before performing machine learning, balance the dataset by the use of Up sampling and Under sampling methods.

**REFERENCES**

**Dataset :** [**https://www.kaggle.com/mlg-ulb/creditcardfraud/notebooks**](https://www.kaggle.com/mlg-ulb/creditcardfraud/notebooks)

**Research Papers :**

D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, "Credit Card Fraud Detection - Machine Learning methods," 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2019, pp. 1-5, doi: 10.1109/INFOTEH.2019.8717766.

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. and Kuruwitaarachchi, N., 2019, January. Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE.

R. Sailusha, V. Gnaneswar, R. Ramesh and G. R. Rao, "Credit Card Fraud Detection Using Machine Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 1264-1270, doi: 10.1109/ICICCS48265.2020.9121114.

W. Yu and N. Wang, "Research on Credit Card Fraud Detection Model Based on Distance Sum," 2009 International Joint Conference on Artificial Intelligence, Hainan Island, 2009, pp. 353-356, doi: 10.1109/JCAI.2009.146.

H. Najadat, O. Altiti, A. A. Aqouleh and M. Younes, "Credit Card Fraud Detection Based on Machine and Deep Learning," 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2020, pp. 204-208, doi: 10.1109/ICICS49469.2020.239524.

S. Benson Edwin Raj and A. Annie Portia, "Analysis on credit card fraud detection methods," 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), Tirunelveli, India, 2011, pp. 152-156, doi: 10.1109/ICCCET.2011.5762457.

https://www.engpaper.com/credit-card-fraud-detection.htm

https://www.sciencedirect.com/science/article/pii/S187705092030065X

https://www.aaai.org/Papers/KDD/1998/KDD98-026.pdf

https://www.researchgate.net/publication/336800562\_Credit\_Card\_Fraud\_Detection\_using\_Machine\_Learning\_and\_Data\_Science --used same as this dataset

https://www.hindawi.com/journals/complexity/2018/5764370/

https://arxiv.org/ftp/arxiv/papers/1611/1611.06439.pdf

<https://www.ijrte.org/wp-content/uploads/papers/v8i2/B1083078219.pdf>

T. R. Pillai, I. A. T. Hashem, S. N. Brohi, S. Kaur and M. Marjani, "Credit Card Fraud Detection Using Deep Learning Technique," 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, 2018, pp. 1-6, doi: 10.1109/ICACCAF.2018.8776797.

**CRISP DM :**

<https://www.datascience-pm.com/crisp-dm-2/>

**What is CRISP DM :**

https://towardsdatascience.com/crisp-dm-methodology-leader-in-data-mining-and-big-data-467efd3d3781

**Credit Card Fraud Statistics available at :**

https://shiftprocessing.com/credit-card-fraud-statistics/

**Metrics for Machine learning algorithm available at :**

https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

**Performance of Algorithms :**

https://machinelearningmastery.com/estimate-performance-machine-learning-algorithms-weka/

**Handling imbalanced dataset :**

https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

**SMOTE UPSAMPLING :**

https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/

**Random Forest :**

<https://www.javatpoint.com/machine-learning-random-forest-algorithm>

https://www.tutorialspoint.com/machine\_learning\_with\_python/machine\_learning\_with\_python\_classification\_algorithms\_random\_forest.htm

**SVM :**

<https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>ss

**PYTHON CODE**

import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import StandardScaler  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import r2\_score, mean\_absolute\_error, mean\_squared\_error  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import classification\_report, confusion\_matrix  
Creditcard=pd.read\_csv('creditcard.csv')  
Creditcard  
Creditcard.columns  
Creditcard.dtypes  
Creditcard.head(5)  
Creditcard.info()  
Creditcard.isnull().sum()  
Creditcard.describe()  
Creditcard.Class.value\_counts()  
Creditcard['Class'].nunique()  
  
print('Fraud Percentage: {}'.format(round((Creditcard['Class'].value\_counts()[1] / len(Creditcard)) \* 100, 2)))  
print('Non Fraud Percentage: {}'.format(round((Creditcard['Class'].value\_counts()[0] / len(Creditcard)) \* 100, 2)))  
Creditcard.drop("Time", axis=1, inplace=True)  
Creditcard.shape  
  
# visulization  
  
# correlation matrix  
corrmat = Creditcard.corr()  
fig = plt.figure(figsize = (11, 8))  
  
sns.heatmap(corrmat, vmax = .7, square = True)  
plt.show()  
  
  
  
sns.scatterplot(x='Amount', y='Class', data=Creditcard)  
plt.show()  
sns.boxplot(x="Class", y="Amount", data=Creditcard)  
plt.ylim(0, 5000)  
plt.show()  
plt.close();  
  
# Plot histograms of each parameter  
Creditcard.hist(figsize = (30, 30))  
plt.show()  
  
plt.close()  
Creditcard  
sns.scatterplot(x='V1', y='Class', data=Creditcard)  
sns.scatterplot(x='V2', y='Amount', data=Creditcard)  
sns.scatterplot(x='V3', y='Amount', data=Creditcard)  
sns.scatterplot(x='V4', y='Amount', data=Creditcard)  
sns.scatterplot(x='V5', y='Amount', data=Creditcard)  
sns.scatterplot(x='V6', y='Amount', data=Creditcard)  
sns.scatterplot(x='V7', y='Amount', data=Creditcard)  
sns.scatterplot(x='V8', y='Amount', data=Creditcard)  
sns.scatterplot(x='V9', y='Amount', data=Creditcard)  
sns.scatterplot(x='V10', y='Amount', data=Creditcard)  
sns.scatterplot(x='V11', y='Amount', data=Creditcard)  
sns.scatterplot(x='V12', y='Amount', data=Creditcard)  
sns.scatterplot(x='V13', y='Amount', data=Creditcard)  
sns.scatterplot(x='V14', y='Amount', data=Creditcard)  
sns.scatterplot(x='V15', y='Amount', data=Creditcard)  
sns.scatterplot(x='V16', y='Amount', data=Creditcard)  
sns.scatterplot(x='V17', y='Amount', data=Creditcard)  
sns.scatterplot(x='V18', y='Amount', data=Creditcard)  
sns.scatterplot(x='V19', y='Amount', data=Creditcard)  
sns.scatterplot(x='V20', y='Amount', data=Creditcard)  
sns.scatterplot(x='V21', y='Amount', data=Creditcard)  
sns.scatterplot(x='V22', y='Amount', data=Creditcard)  
sns.scatterplot(x='V23', y='Amount', data=Creditcard)  
sns.scatterplot(x='V24', y='Amount', data=Creditcard)  
sns.scatterplot(x='V25', y='Amount', data=Creditcard)  
sns.scatterplot(x='V26', y='Amount', data=Creditcard)  
sns.scatterplot(x='V27', y='Amount', data=Creditcard)  
sns.scatterplot(x='V28', y='Amount', data=Creditcard)  
sns.scatterplot(x='V27', y='Amount', data=Creditcard)  
sns.scatterplot(x='V1', y='V2', data=Creditcard)  
sns.scatterplot(x='V1', y='V3', data=Creditcard)  
sns.scatterplot(x='V1', y='V4', data=Creditcard)  
sns.scatterplot(x='V1', y='V5', data=Creditcard)  
sns.scatterplot(x='V1', y='V6', data=Creditcard)  
sns.scatterplot(x='V1', y='V7', data=Creditcard)  
sns.scatterplot(x='V1', y='V8', data=Creditcard)  
sns.scatterplot(x='V1', y='V9', data=Creditcard)  
sns.scatterplot(x='V1', y='V10', data=Creditcard)  
sns.scatterplot(x='V1', y='V11', data=Creditcard)  
sns.scatterplot(x='V1', y='V12', data=Creditcard)  
sns.scatterplot(x='V1', y='V13', data=Creditcard)  
sns.scatterplot(x='V1', y='V14', data=Creditcard)  
sns.scatterplot(x='V1', y='V15', data=Creditcard)  
sns.scatterplot(x='V1', y='V16', data=Creditcard)  
sns.scatterplot(x='V1', y='V17', data=Creditcard)  
sns.scatterplot(x='V1', y='V18', data=Creditcard)  
sns.scatterplot(x='V1', y='V19', data=Creditcard)  
sns.scatterplot(x='V1', y='V20', data=Creditcard)  
sns.scatterplot(x='V1', y='V21', data=Creditcard)  
sns.scatterplot(x='V1', y='V22', data=Creditcard)  
sns.scatterplot(x='V1', y='V23', data=Creditcard)  
sns.scatterplot(x='V1', y='V24', data=Creditcard)  
sns.scatterplot(x='V1', y='V25', data=Creditcard)  
sns.scatterplot(x='V1', y='V26', data=Creditcard)  
sns.scatterplot(x='V1', y='V27', data=Creditcard)  
sns.scatterplot(x='V1', y='V28', data=Creditcard)  
  
# model Building  
# LogisticRegression  
X = Creditcard[['Amount']]  
y = Creditcard['Class']  
sc = StandardScaler()  
sc  
X  
X = sc.fit\_transform(X)  
X  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)  
log\_reg = LogisticRegression(random\_state=0)  
log\_reg.fit(X\_train, y\_train)  
log\_reg.coef\_  
log\_reg.intercept\_  
y\_pred = log\_reg.predict(X\_test)  
confusion\_matrix(y\_test, y\_pred)  
  
print(classification\_report(y\_test, y\_pred))  
  
# smote  
from imblearn.over\_sampling import SMOTE  
  
smote\_oversample = SMOTE()  
X, y = smote\_oversample.fit\_resample(X, y)  
X.shape  
y.shape  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)  
log\_reg = LogisticRegression(random\_state=0)  
log\_reg.fit(X\_train, y\_train)  
y\_pred = log\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
# LinearSVC  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.svm import LinearSVC  
  
svc\_reg = LinearSVC(random\_state=0)  
svc\_reg.fit(X\_train, y\_train)  
y\_pred = svc\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
# RandomForest  
rf\_reg = RandomForestClassifier(random\_state=0)  
rf\_reg.fit(X\_train, y\_train)  
y\_pred = rf\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
# scaling data using standard scaller  
X = Creditcard.drop(['Class'], axis=1)  
y = Creditcard['Class']  
sc = StandardScaler()  
smote\_oversample = SMOTE()  
X, y = smote\_oversample.fit\_resample(X, y)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)  
  
log\_reg = LogisticRegression(random\_state=0)  
log\_reg.fit(X\_train, y\_train)  
y\_pred = log\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
#Evaluating The Model Random Forest  
from sklearn import metrics  
  
print('Confusion Matrix\n ',metrics.confusion\_matrix(y\_test, y\_pred))  
print('\n Classification Report\n ')  
print(metrics.classification\_report(y\_test, y\_pred))  
  
print('\nAccuracy : %f' % (metrics.accuracy\_score(y\_test, y\_pred)))  
print('Area under the curve : %f' % (metrics.roc\_auc\_score(y\_test, y\_pred)))  
print("Precision : %f" % (metrics.precision\_score(y\_test,y\_pred)))  
print("Recall : %f" % (metrics.recall\_score(y\_test,y\_pred)))  
print("F1-Score : %f" % (metrics.f1\_score(y\_test,y\_pred)))  
  
  
X = sc.fit\_transform(X)  
svc\_reg = LinearSVC(random\_state=0)  
svc\_reg.fit(X\_train, y\_train)  
y\_pred = svc\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
#Evaluating The Model Random Forest  
from sklearn import metrics  
  
print('Confusion Matrix\n ',metrics.confusion\_matrix(y\_test, y\_pred))  
print('\n Classification Report\n ')  
print(metrics.classification\_report(y\_test, y\_pred))  
  
print('\nAccuracy : %f' % (metrics.accuracy\_score(y\_test, y\_pred)))  
print('Area under the curve : %f' % (metrics.roc\_auc\_score(y\_test, y\_pred)))  
print("Precision : %f" % (metrics.precision\_score(y\_test,y\_pred)))  
print("Recall : %f" % (metrics.recall\_score(y\_test,y\_pred)))  
print("F1-Score : %f" % (metrics.f1\_score(y\_test,y\_pred)))  
  
  
rf\_reg = RandomForestClassifier(random\_state=0)  
rf\_reg.fit(X\_train, y\_train)  
y\_pred = rf\_reg.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
  
 #Evaluating The Model Random Forest  
from sklearn import metrics  
  
print('Confusion Matrix\n ',metrics.confusion\_matrix(y\_test, y\_pred))  
print('\n Classification Report\n ')  
print(metrics.classification\_report(y\_test, y\_pred))  
  
print('\nAccuracy : %f' % (metrics.accuracy\_score(y\_test, y\_pred)))  
print('Area under the curve : %f' % (metrics.roc\_auc\_score(y\_test, y\_pred)))  
print("Precision : %f" % (metrics.precision\_score(y\_test,y\_pred)))  
print("Recall : %f" % (metrics.recall\_score(y\_test,y\_pred)))  
print("F1-Score : %f" % (metrics.f1\_score(y\_test,y\_pred)))  
  
from sklearn.preprocessing import StandardScaler  
Creditcard['Amount(Normalized)'] = StandardScaler().fit\_transform(Creditcard['Amount'].values.reshape(-1,1)  
  
Creditcard.iloc[:,[29,31]].head()  
  
Creditcard = Creditcard.drop(columns = ['Amount', 'Time'], axis=1) # This columns are not necessary anymore.  
  
X = Creditcard.drop('Class', axis=1)  
y = Creditcard['Class']  
  
#train-test split  
  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)  
# We are transforming data to numpy array to implementing with keras  
X\_train = np.array(X\_train)  
X\_test = np.array(X\_test)  
y\_train = np.array(y\_train)  
y\_test = np.array(y\_test)  
  
#ANN  
from keras.models import Sequential  
from keras.layers import Dense, Dropout  
model = Sequential([  
 Dense(units=20, input\_dim = X\_train.shape[1], activation='relu'),  
 Dense(units=24,activation='relu'),  
 Dropout(0.5),  
 Dense(units=20,activation='relu'),  
 Dense(units=24,activation='relu'),  
 Dense(1, activation='sigmoid')  
])  
model.summary()  
  
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])  
model.fit(X\_train, y\_train, batch\_size=30, epochs=5)  
  
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])  
model.fit(X\_train, y\_train, batch\_size=30, epochs=5)  
  
score = model.evaluate(X\_test, y\_test)  
print('Test Accuracy: {:.2f}%\nTest Loss: {}'.format(score[1]\*100,score[0]))  
  
#SMOTE  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_smote, y\_smote, test\_size=0.3, random\_state=0)  
X\_train = np.array(X\_train)  
X\_test = np.array(X\_test)  
y\_train = np.array(y\_train)  
from imblearn.over\_sampling import SMOTE  
X\_smote, y\_smote = SMOTE().fit\_sample(X, y)  
X\_smote = pd.DataFrame(X\_smote)  
y\_smote = pd.DataFrame(y\_smote)  
y\_smote.iloc[:,0].value\_counts()  
  
y\_test = np.array(y\_test)  
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])  
model.fit(X\_train, y\_train, batch\_size = 30, epochs = 5)  
  
#Accurancy  
score = model.evaluate(X\_test, y\_test)  
print('Test Accuracy: {:.2f}%\nTest Loss: {}'.format(score[1]\*100,score[0]))  
  
  
  
  
#Confusion matrix1  
y\_pred = model.predict(X\_test)  
y\_test = pd.DataFrame(y\_test)  
cm = confusion\_matrix(y\_test, y\_pred.round())  
sns.heatmap(cm, annot=True, fmt='.0f')  
plt.show()  
#Confusion matrix2  
y\_pred2 = model.predict(X)  
y\_test2 = pd.DataFrame(y)  
cm2 = confusion\_matrix(y\_test2, y\_pred2.round())  
sns.heatmap(cm2, annot=True, fmt='.0f', cmap='coolwarm')  
plt.show()  
  
scoreNew = model.evaluate(X, y)  
print('Test Accuracy: {:.2f}%\nTest Loss: {}'.format(scoreNew[1]\*100,scoreNew[0]))  
  
print(classification\_report(y\_test2, y\_pred2.round()))