# PREDICTION OF MOBILE PRICES USING MACHINE LEARNING TECHNIQUES

*A Project Report submitted in partial fulfilment of the requirements for the award of the degree of*

**Bachelor of Technology**

**in**

*Computer Science and Engineering*

**by**


| Name of Students | Roll No |
|---|---|
| Pratibha Dixit | 181500498 |
| Deeksha Mishra | 181500201 |
| Aditi Agarwal | 181500040 |
| Mayank Goyal | 181500376 |

Under the Guidance of

Dr. Pooja Pathak

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**




**GLA University**

**Mathura- 281406, INDIA**

**Dec, 2020**

**Department of Computer Engineering and Applications**
**GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,**
**Chaumuha, Mathura – 281406 U.P (India)**

# Declaration

I/we hereby declare that the work which is being presented in the B.Tech. Project **"Title of Project"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Name & Designation of supervisor(s).** The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *Pratibha Dixit*

Name of Candidate: Pratibha Dixit
University Roll No.: 181500498

Sign: *Deeksha Mishra*

Name of Candidate: Deeksha Mishra
University Roll No.: 181500201

Sign: *Aditi Agarwal*

Name of Candidate: Aditi Agarwal
University Roll No.: 181500040

Sign: *Mayank Goyal*

Name of Candidate: Mayank Goyal
University Roll No.: 181500376

# Certificate

This is to certify that the above statements made by the candidate/candidates are correct to the best of my/our knowledge and belief.

_____

**Supervisor**
Dr.(Prof). Anand Singh Jalal
Head of Dept., Department of CEA

_____ _____

**Project Coordinator** **Program Co-ordinator**

(Dr. Pooja Pathak) (Mr. Shashi Shekhar)

Date: 20 November, 2020

# ACKNOWLEDGEMENT

It gives us the immense pleasure to present the report of the B.Tech. Machine Learning Project undertaken during B.Tech. 3rd Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by mentors. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to **Dr.(Prof). Anand Singh Jalal, Head of Dept., Department of CEA** for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to **Dr. Pooja Pathak**, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance has been a constant source of inspiration for us. She has showered us with all her extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project.

Thanking You All,

Sign: *Pratibha Dixit*

Name of Candidate: Pratibha Dixit
University Roll No.: 181500498

Sign: *Deeksha Mishra*

Name of Candidate: Deeksha Mishra
University Roll No.: 181500201

Sign: *Aditi Agarwal*

Name of Candidate: Aditi Agarwal
University Roll No.: 181500040

Sign: *Mayank Goyal*

Name of Candidate: Mayank Goyal
University Roll No.: 181500376

# ABSTRACT

The key purpose of this project is to determine "If the mobile with given features would be under a certain price range." Specific feature selection algorithms are used to recognize and delete features that are less necessary and redundant, and have minimal complexity in computation. Different classifiers are used to achieve the best possible accuracy.

Results are measured in terms of achieving the maximum accuracy and choosing the minimum features. Statement is made based on the algorithm for best selection of features and best classifier for the given dataset. This work can be used to find the optimal product (with minimum cost and maximum features) in any form of marketing and industry. It is suggested that future work will extend this research and find a more sophisticated solution to the given problem and a more accurate tool for estimating prices.

In this modern era, smartphones are an integral part of the lives of human beings. When a smartphone is purchased, many factors like the display, processor, memory, camera, thickness, battery, connectivity and others are taken into account. One factor that people do not consider is whether the product is worth the cost. As there are no resources to cross-validate the price, people fail in taking the correct decision. This paper looks to solve the problem by taking the historical data pertaining to the key features of smartphones along with its cost and develop a model that will predict the approximate price of the new smartphone with a reasonable accuracy. The data set used for this purpose has taken into consideration 21 different parameters for predicting the price of the phone. Random forest classifier, support vector machine, decision trees, k-nearest neighbour and logistic regression have been used primarily. Based on the accuracy, the appropriate algorithm has been used to predict the prices of the smartphone. This not only helps the customers decide the right phone to purchase, it also helps the owners decide what should be the appropriate pricing of the phone for the features that they offer. This idea of predicting the price will help the people make informed choice when they are purchasing a phone in the future. Among all the classifiers chosen, logistic regression and support vector machine had the highest accuracy of 81%. Further, logistic regression was used to predict the prices of the phone.

# Contents

# CHAPTER 1 Introduction

## 1.1 Overview and Motivation:

On a daily basis, we encounter a lot of trade-off in life, for instance, what to opt tasty food versus healthy food, cost of product versus features of product, durability versus reliability and lot more. The complexity of such situations increases day by day and common man faces a lot of tough time to cope up with it. Moreover, taking correct decision in limited time has always been crucial in such a scenario.

Nowadays, in this modern digital age, social media is substantially evolving at a very fast pace. The growth has always been tremendous by connecting everyone across the globe in a quick, secure and convenient manner. Smartphones are one of the most readily accessible devices by every individual in this platform. It is one of the most common devices which many individuals possess and it is quite impossible for anyone to survive without it. With various customizations, features along with add-on plugin have enhanced its position in market.

Price is the most effective attribute of marketing and business. The very first question of costumer is about the price of items. All the costumers are first worried and thinks “If he would be able to purchase something with given specifications or not”. So to estimate price at home is the basic purpose of the work.

## 1.2 Objective:

The primary goal of our study is to determine the mobile price range based on attributes which are set of specifications for mobile phones.

The increase in demand for smartphones has simultaneously led to increase in manufacturers all over the world. The manufactures have started increasing features of their product to securely speed up their position in market. This arises problem for people as to which smartphone to purchase with appropriate features. Overall, purchase of smartphone has always been an issue encountered by all in some instance of time. People spent a lot of time thinking and cross-checking with their peers about product. People are often in dilemma whether the features provided by the manufacturer of the phone are really worth the cost of buying. The attempts to purchase a phone by people in dilemma led to disappointing results of spent significant amount of time as well as their money. “These above-mentioned factors fascinate up the thought of ‘Predicting the price of Mobile Phone”’. This helps people in making correct decision as well as for manufacturer for validating cost of phone with features provided to their customers. At the end of day, both customer and manufactures get satisfied with product on the whole with valid statistical proofs.

In social media, many people tend to post rating of product without any hesitation.

So, in this research, the historical sentiments of people are taken into consideration, i.e. their opinion of price whether it is high, medium or low. Apart from the important factors like display, processor and memory, other features like camera, thickness, battery and connectivity have also been taken into account to determine the approximate price for a phone.

For predicting the price, we are using data set from the popular data set platform Kaggle. The train and test data set are given as two different CSV files. So, to train the model and for predicting which model gives the most accuracy, the train data set is used. The trained model is then used on the test data set to predict the price.

## 1.3 Summary of Similar Application:

For this concept, due to unavailability of related specific papers, papers regarding recommender systems, sentiment analysis and other predicting system have been used for the purpose of literature survey.
In the paper, the authors have mined for historical data from many sources like IMDB and Rotten Tomatoes. After processing the data, the authors have implemented SVM and neural networks to find the accuracy of the prediction. For the data set chosen by the team, the neural network gave the most accurate predictions.

Mansouri et al. in the paper, have made prediction on the useful battery charge that can be used in a UAV. The authors have used a variation of SVM, an advanced tree-based algorithm, a linear sparse model and a multilayer perceptron to make the prediction. Using all these algorithms, a preliminary investigation was done.

In the paper, the authors look to predict the performance of Karachi Stock
Exchange. Various machine learning techniques like single-layer perceptron, SVM, radial basis function and multilayer perceptron are used on the data set to make the prediction. The performance of the multilayer perceptron was the best. The final conclusion drawn from the research was that KSE performance can be predicted using machine learning techniques.

In the paper, neural network is implemented and an accuracy of 75% is achieved. This sentiment analysis is very useful for mining useful knowledge from all the data available.

Paper aimed to recommend users the ingredients for different cuisines. The recipes were checked using classifiers like support vector machine and associative classification. The accuracy used in classifiers was used for comparison.

The paper predicts factors that will be affecting future usage of tags. Tags are essentially used to easily search for the answer in that particular domain. Machine learning techniques are used to automate as well as classify them popularity of tags based on structural and non-structural features. The classifiers used were logistic regression, SVM, random forest and AdaBoost. Random forest is the best among the other classifiers based on accuracy.

The paper helps in proper usage of correct algorithm in recommended system and identifying machine learning techniques, new research areas to invest upon. It also focuses on main and alternative performance metrics.

The paper attempted to research efficient models and compared their performance in predicting the direction of movement of daily Istanbul stock Exchange(ISE) National 100 index. Artificial neural network and support vector machine were used for classification. Among them, artificial neural network model (ANN) is a better performance model compared to support vector machine (SVM).

The paper helps in generating forecast of online content. Prediction methods of existing system are used in algorithm for real-time forecasting of popularity with minimal training information.

The paper helps in predicting severity of crash that is expected to occur in future. It uses multinomial logit (MNL), nearest neighbor classification (NNC), support vector machines and random forest (RF) for predicting the traffic crash severity. The proposed approach showed NNC as the best prediction performance in overall and is in more accurate in severe crashes.

Based on the literature, an efficient and accurate way for predicting the price of a smartphone has been determined.

# CHAPTER 2 Software Requirement Analysis

## 2.1 Hardware Requirements Specification:

- CPU: 2 x 64-bit 2.8 GHz 8.00 GT/s CPUs
- RAM: 32 GB (or 16 GB of 1600 MHz DDR3 RAM)
- Storage: 300 GB. (600 GB for air-gapped deployments.) Additional space recommended if the repository will be used to store packages built by the customer. With an empty repository, a base install requires 2 GB.
- Internet access to download the files from Anaconda Cloud or a USB drive containing all of the files you need with alternate instructions for air gapped installations.

## 2.2 Software Requirements Specification:

- RHEL/CentOS 6.5 to 7.4, Ubuntu 12.04+Ubuntu users may need to install cURL.
- Client environment may be Windows, macOS or Linux
- MongoDB 2.6 (provided)
- Anaconda Repository license file
- Cron entry to start the repo on reboot
- Linux system accountsmongod (RHEL) or mongodb (Ubuntu)anaconda-server

## 2.3 Security Requirements:

- Privileged access OR sudo capabilities
- Open HTTP(S) port
- SELinux policy edit privileges (SELinux does not have to be disabled for Anaconda Repository operation)
- Optional: Ability to make iptables modifications
- Optional: SSL certificate

## 2.4 Network Requirements (TCP ports):

- Inbound HTTP: TCP 8080, 8443 (Anaconda repository)
- Optional Inbound SSH: TCP 22 (SSH)

- Optional Outbound HTTPS:
  TCP http://443repo.anaconda.comanaconda.orgconda.anaconda.orgbinstar-cio-packages-prod.s3.amazonaws.com820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.rackcdn.com
- Optional Outbound SMTP: TCP 25 (if not using AD/LDAP) email notifications
- Optional Outbound LDAP(s): TCP 389/636 for authentication integration

## 3.5 Other Requirements:

- License file provided to you by Anaconda at the time of purchase
- Installation tokens for binstar and anaconda-server channels provided by Anaconda at the time of purchase. Not applicable for air gapped installs.
- Optional: Your Anaconda Cloud ( :: Anaconda Cloud ) account credentials. Not applicable for air gapped installs.

# CHAPTER 3 Software Design

## 3.1 Existing Software Design:

The existing system as we seen includes the use of classifier were they train dataset and perform operations on the dataset and then user get output value.

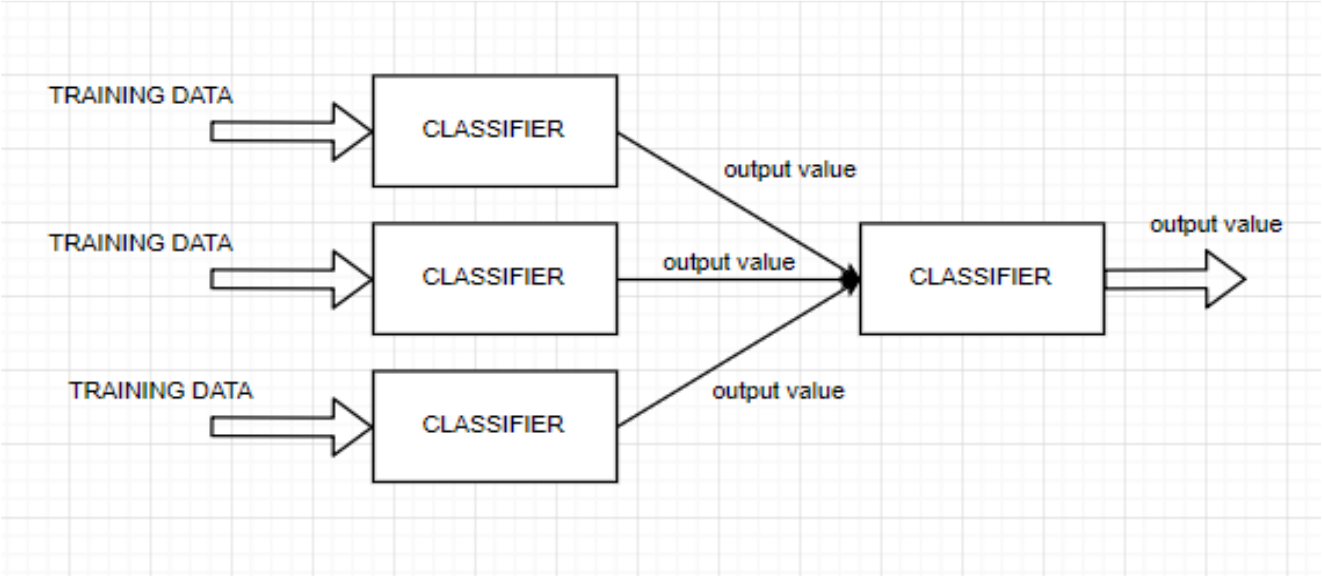



Fig 3.1: The presented System Architecture is using classifier and datasets

## 3.2 Proposed Software Design:

As discussed above, architecture does not have the accurate price prediction to overcome this we designed architecture where accuracy is achieved.

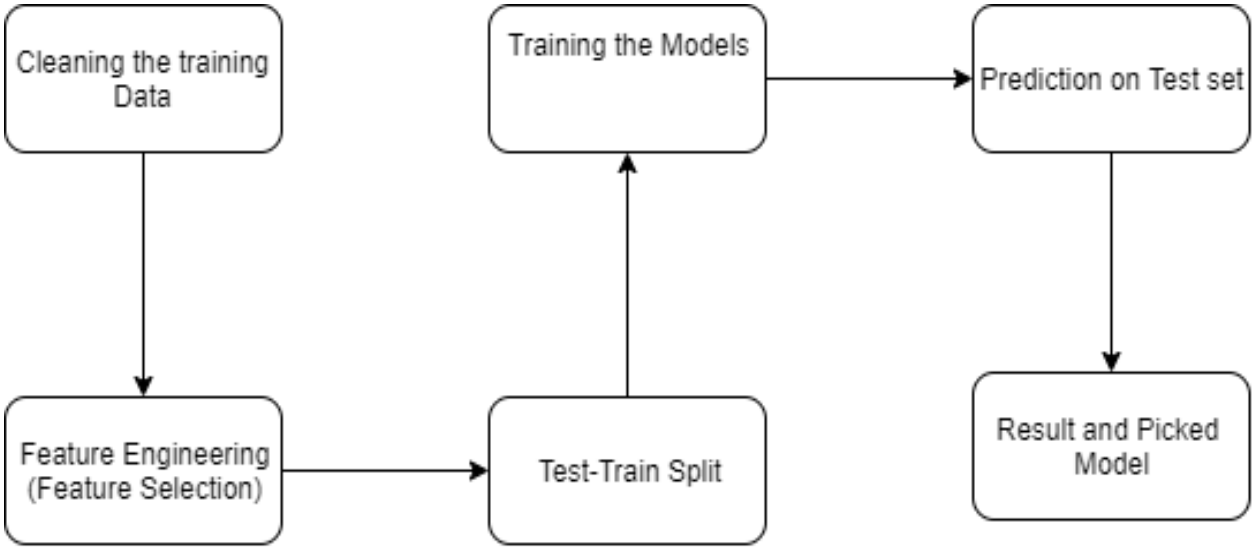



Fig 3.2: The presented System Architecture when accuracy is achieved

# CHAPTER 4 Implementation

## 4.1 Methodology:

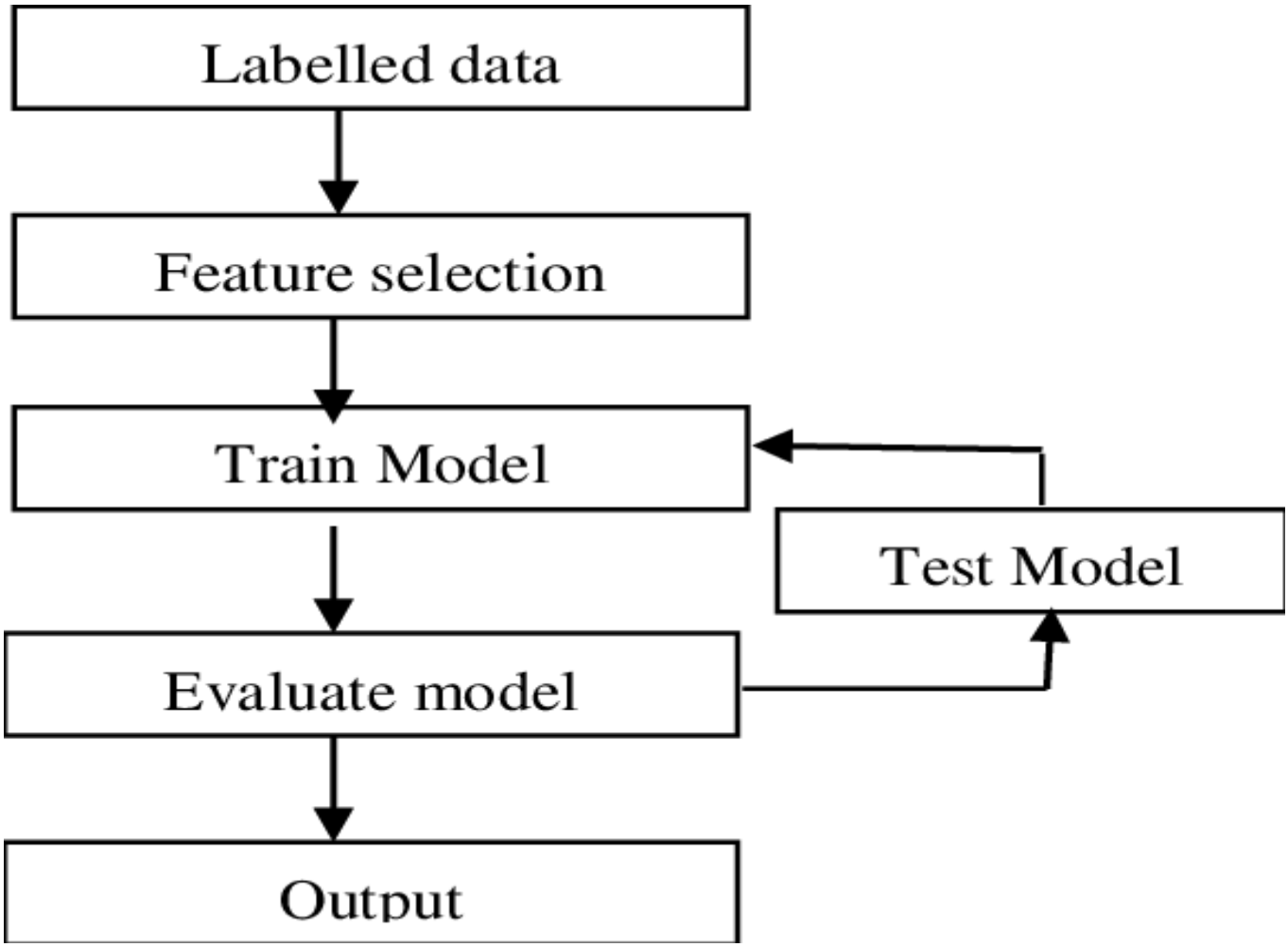



Fig 4.1: Methodology for development of our project

## 4.2 Data Description:

The data set was downloaded from Kaggle. As the pre-processed data were already available no pre-processing has been done on the data set. The data set has the following columns. Each of this is an attribute associated with phone. The test data set that has the same attributes as the train dataset except the price column is not available. This data set takes into account all the parameters associated with a smartphone as they are all a deciding factor on the price of the phone. Listed below is the various names of the column in the table.

- Battery_power: Total energy that a battery can store at a time (measured in mAh)
- Blue: if phone as Bluetooth or not
- Clock_speed: speed at which the microprocessor in the phone executes the instructions
- Dual_sim: If dual-sim support is offered or not
- Fc: Front camera focus in megapixels
- Four_g: If 4G is available or not
- Int_memory: Internal memory of phone in gigabytes
- M_dep: Mobile depth in cm
- Mobile_wt: Weight of the phone
- N_cores: Number of cores in the processor
- Pc: Primary camera focus in megapixels
- Px_height: Pixel resolution height
- Px_width: Pixel resolution width
- Ram: Random access memory (megabytes)
- Sc_h: Screen height of phone in cm
- Sc_w: Screen width of phone in cm
- Talk_time: maximum time that a single battery charge will last when used for talking
- Three_g: Has 3G or not
- Touch_screen: is the display touch screen or not
- Wi-Fi: Has Wi-Fi connectivity or not
- Price_range: The target variable of this data set. This has four discrete values of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).

Here, the approximate price range is predicted. We use discrete value for the price in the place of exact price. The price of the column is predicted based on 21 different parameters. The data set contains data of close to 2000 odd phones.

## 4.3 Classifiers:

### 4.3.1 Linear Regression:

Before knowing what is linear regression, let us get ourselves accustomed to regression. Regression is a method of modelling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

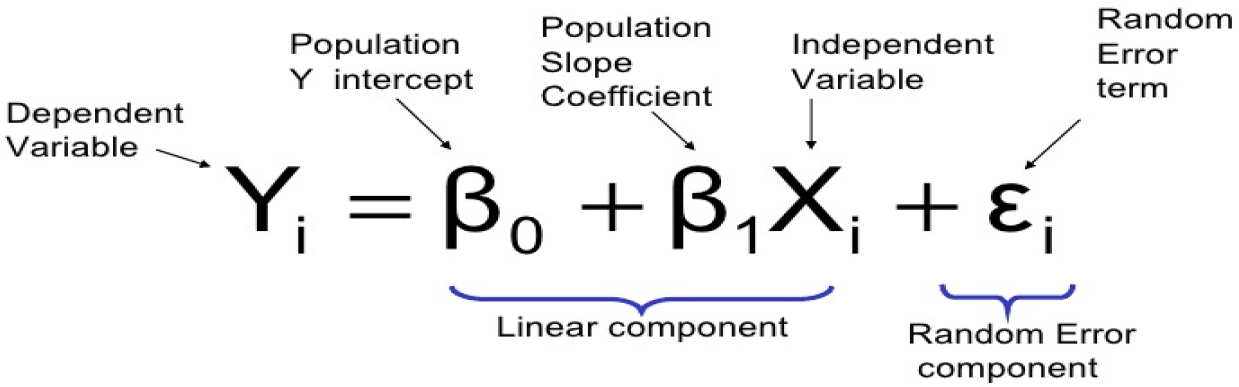



Fig 4.2: A line can be modelled based on the linear equation shown above.

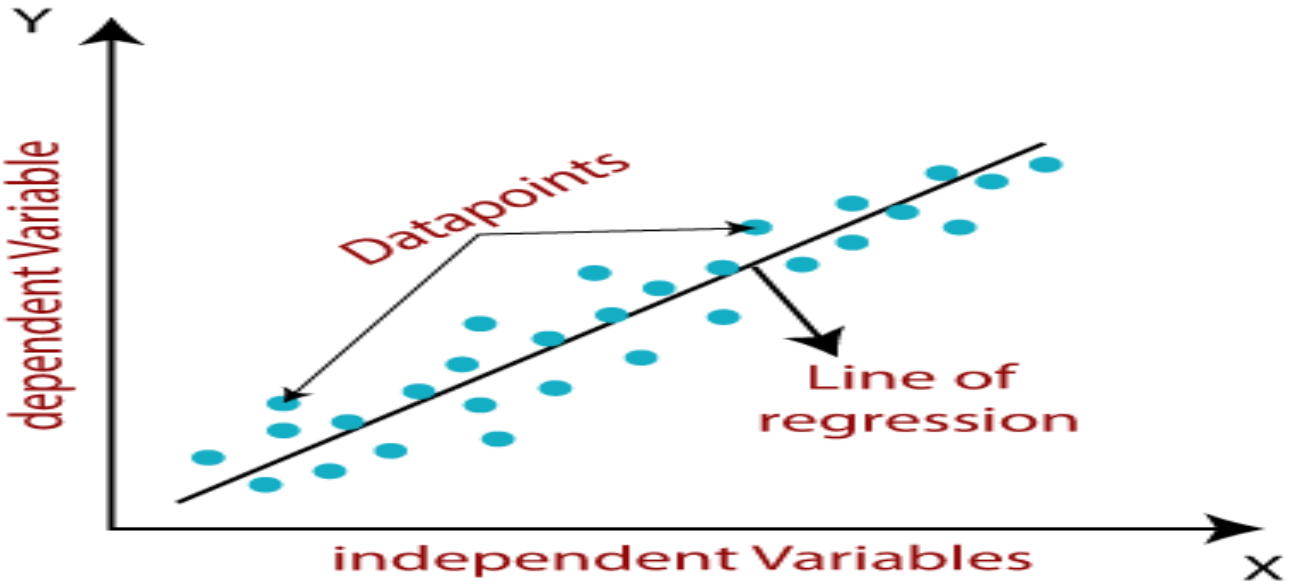



Fig 4.3: Graphical Representation of Linear Regression

## 4.3.2 Logistic Regression:

The name regression for this algorithm is a misnomer. This is a classification algorithm. By using a given set of independent variables, this algorithm predicts discrete values as output. Mathematically, the logarithm of probability of outcome is modeled as a combination of various predictor variables. This combination is linear in nature. Here, the algorithm chooses parameters in such a way that odds of observing the sample quantity is increased. Mathematical equation is as follows:

Linear Equation:

$$y = a + bx$$

$$\text{Outcome probability} = \frac{P}{(1 - P)}$$

$$\log e\left(\frac{p}{(1-p)}\right) = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

Fig 4.4: The function is a step function, and so, logarithmic value is taken to make replication easier.

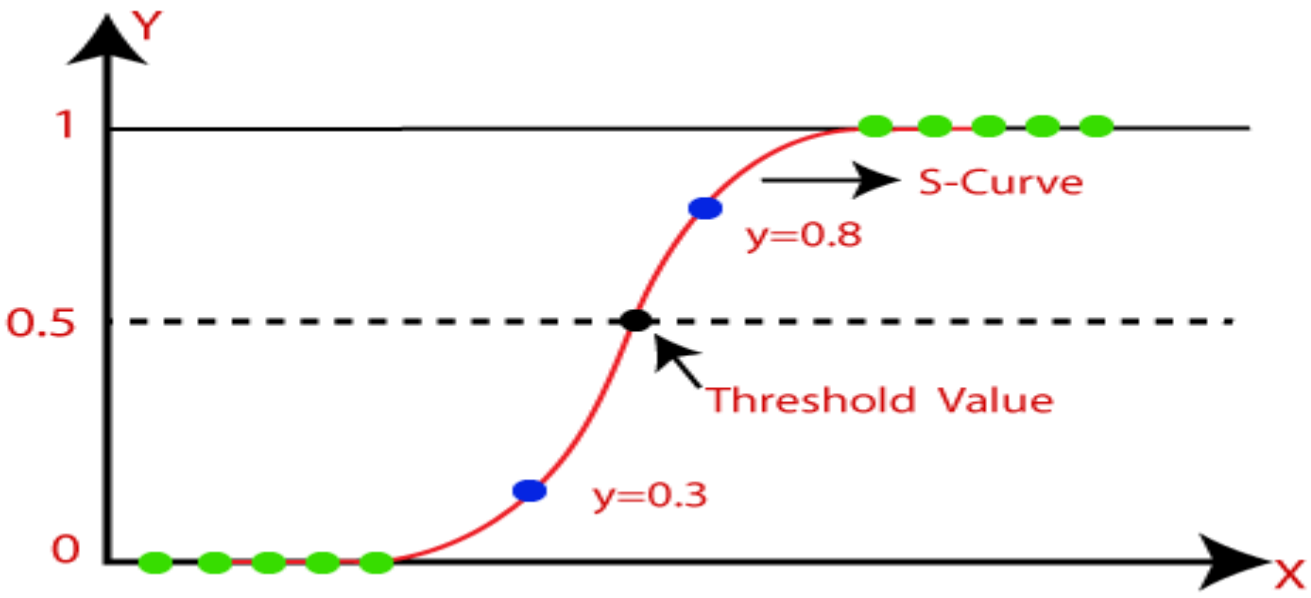



Fig 4.5: Graphical Representation of Logistic Regression

### 4.3.3 K-Nearest Neighbor (or knn):

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

KNN's main disadvantage of becoming significantly slower as the volume of data increases makes it an impractical choice in environments where predictions need to be made rapidly. Moreover, there are faster algorithms that can produce more accurate classification and regression results.

However, provided you have sufficient computing resources to speedily handle the data you are using to make predictions, KNN can still be useful in solving problems that have solutions that depend on identifying similar objects. An example of this is using the KNN algorithm in recommender systems, an application of KNN-search.

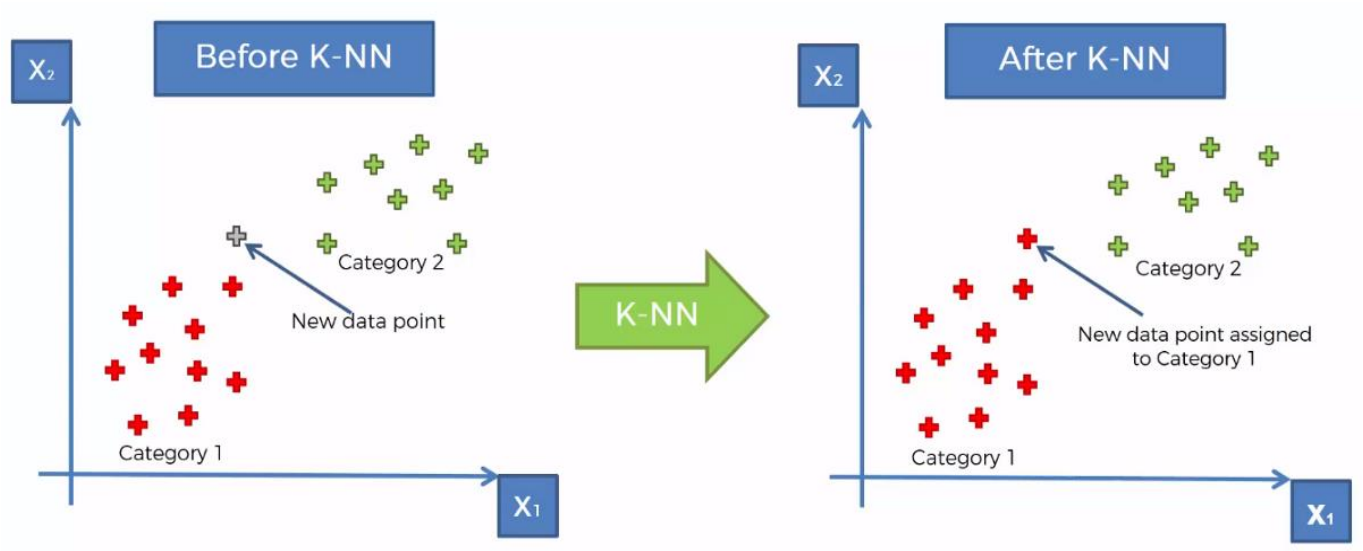



Fig 4.6: Graphical Representation of K-Nearest Neighbor (or knn).

### 4.3.4 Decision Trees:

Decision trees are constructed via an algorithm approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are non-parametric supervised learning method used for both classification and regression tasks.

Tree models where the target variable can take a discrete set of values are called classification trees. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

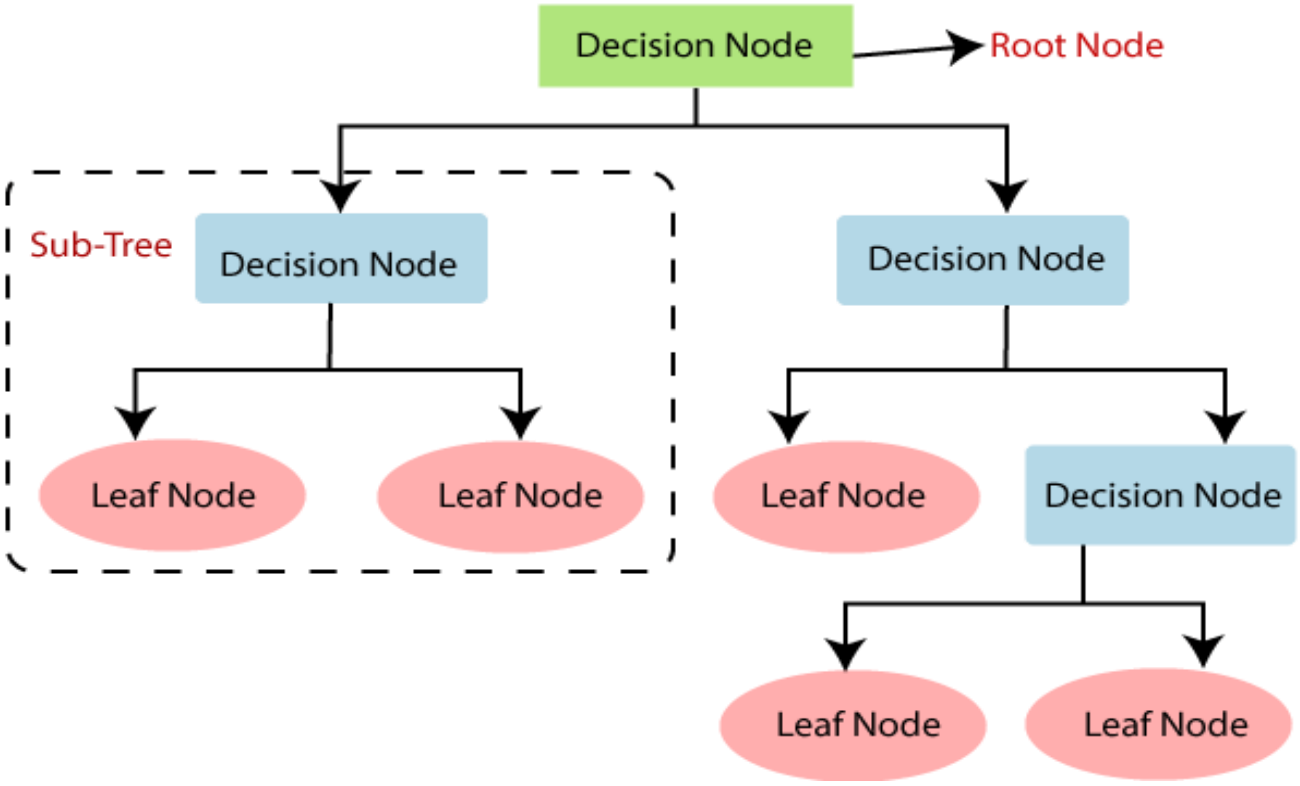



Fig 4.7: Graphical Representation of Decision Trees.

### 4.3.5 Random Forest Classifier or Random Decision Forests:

Random Forest Classifier is one of the decision tree algorithms. As the name forest suggests, the name forest refers to the fact that the algorithm is a collection of many decision trees. When a number of cases are present in the training set, the equivalent number of samples is taken at random and is used for training the growing tree. Each of the decision tree is let to grow for the maximum extent. Pruning is not done.

Random forest is an aggregate of decision trees, supervised classifier. Each decision tree is in turn a subset of forest which is trained on different training data set with some random selected subset of features. The final result is collection of results of different subset model.

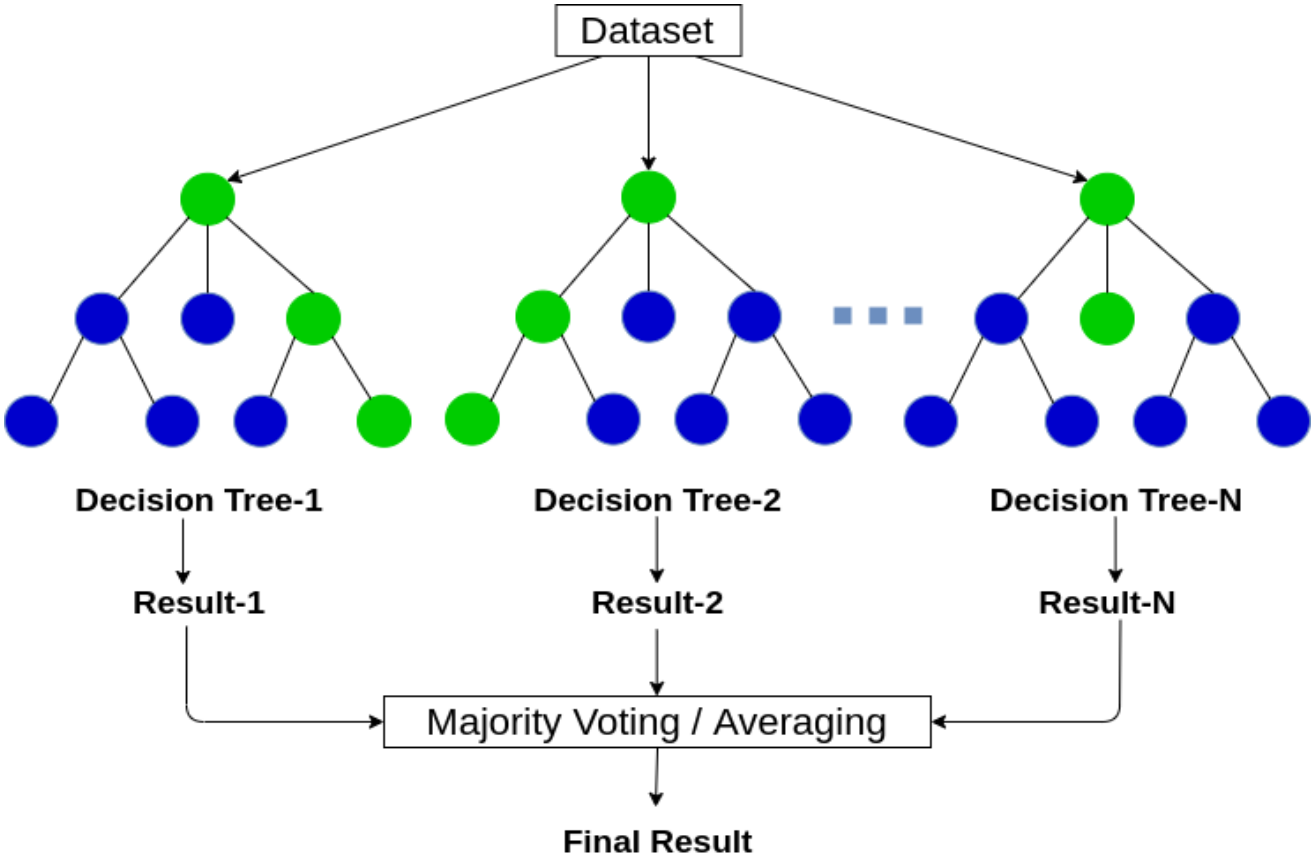



Fig 4.8: Graphical Representation of Random Forest Classifier or Random Decision Forests.

## 4.3.6 Support Vector Machine Classifier (or SVM):

Among the various classification algorithms available, this is a very powerful algorithm. Here, the data are first plotted as a scatter plot. Then, multiple lines are drawn separating the various data points. Then, a line is modeled in such a way that it splits the data points into two distinct groups.
Support vector machine is primarily used for modeling binary classifiers, which classifies data into two categories by developing hyper plane in multidimensional space. Moreover, its decision of predicting is based on linear function.

$$y = f(x) = a + bx$$

Fig 4.9: The above equation shows linear relationship of SVM equation.

It is supervised model wherein it learns from training data and predicts the output. The best hyper plane is known by name called margin hyper plane which maximizes sum of distance on either.
The Rbf parameter in support vector machine means radial basis function. It is widely used in the algorithms that use kernels for execution. One such algorithm that uses Rbf is the SVM algorithm. The equation representing the equation is given below.

$$\exp\left(-\frac{1}{2}|x-x'|^2\right) = \sum_{j=0}^{\infty}\frac{1}{j!}(x^T x')\exp\left|\frac{-1}{2}\right|x^2|\exp|\frac{-1}{2}|x^2|$$

Fig 4.10: The above equation shows the equation of radial basis function kernel.

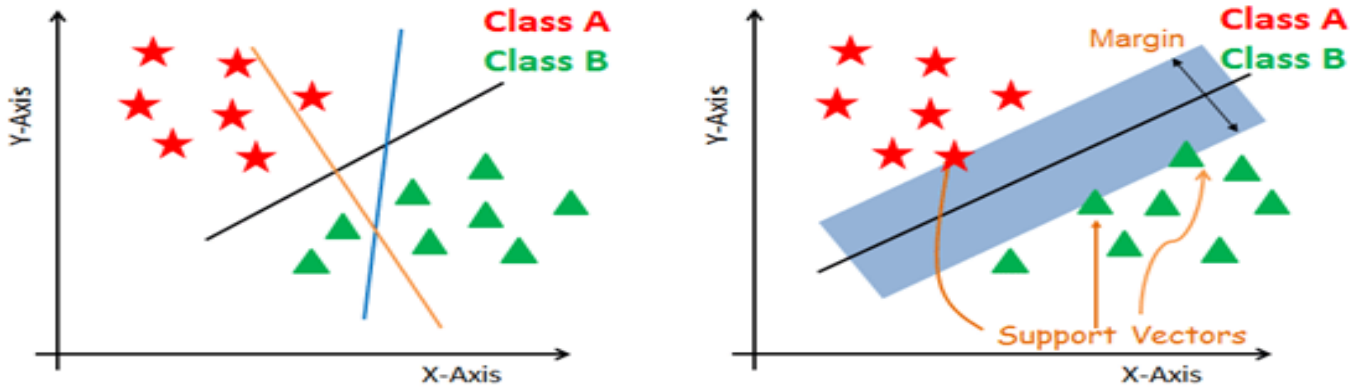



Fig 4.11: Graphical Representation of Support Vector Machine Classifier (or SVM)

## 4.4 Performance Measure and Analysis:

### 1. Binary Class Classification:

#### 1.1 Linear Regression:

- Accuracy: 64.00%
- Scatter Plot Graphical Representation:

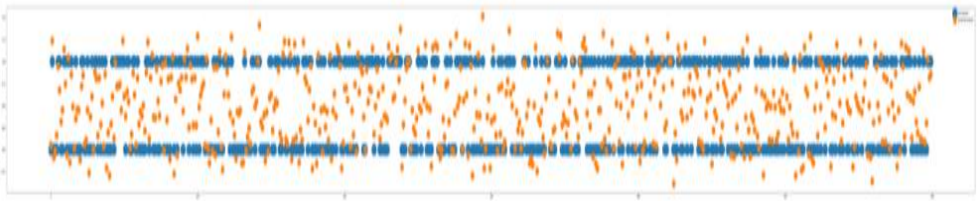


Fig 4.12: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Linear Regression model.

- Line Plot Graphical Representation:

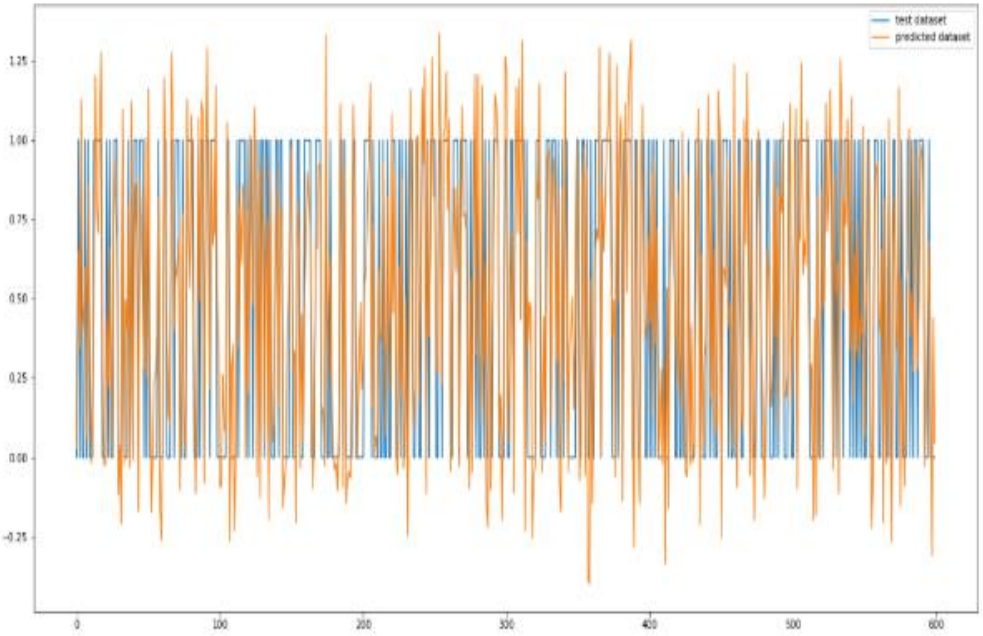



Fig 4.13: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Linear Regression model.

## 1.2 Logistic Regression:

- Accuracy: 98.30%

- ROC- curve and PR-curve:

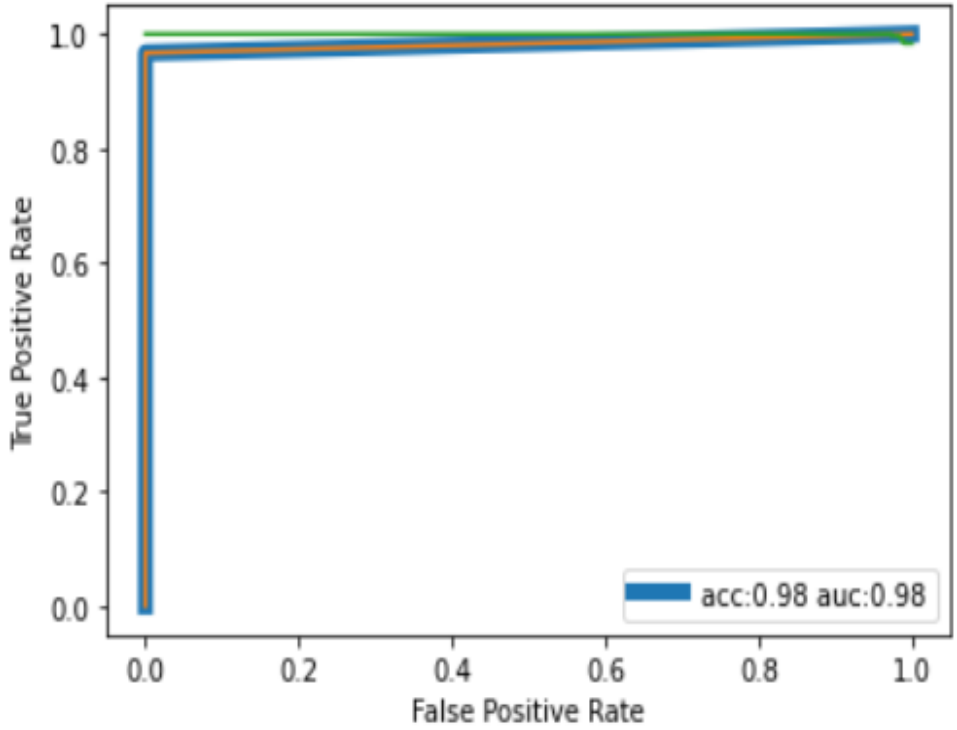



Fig 4.14: ROC curve and PR curve for Logistic Regression model.

- Scatter Plot Graphical Representation:

Fig 4.15: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Logistic Regression model.

- Line Plot Graphical Representation:

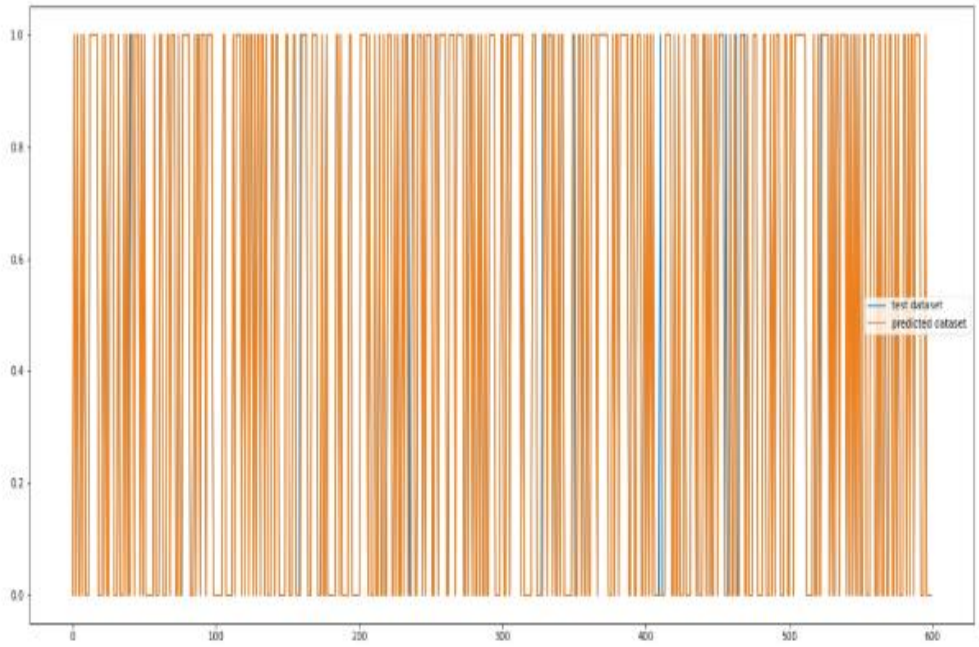



Fig 4.16: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Logistic Regression model.

## 1.3 K-Nearest Neighbor (or knn):

- Accuracy: 81.30%
- Error Rate Graphical Representation:

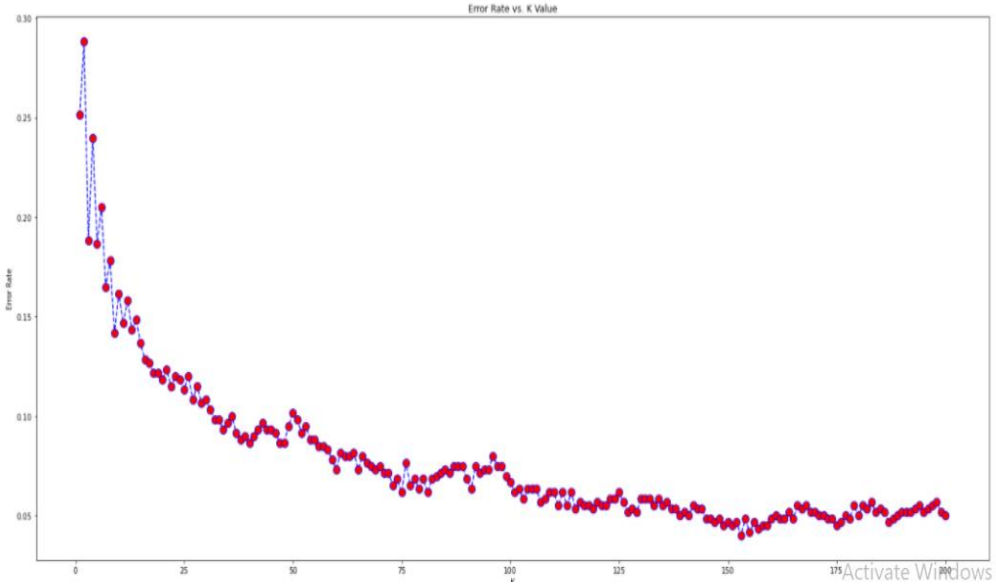



Fig 4.17: Error rate graph for KNN Model.

- Testing and Training Accuracy:

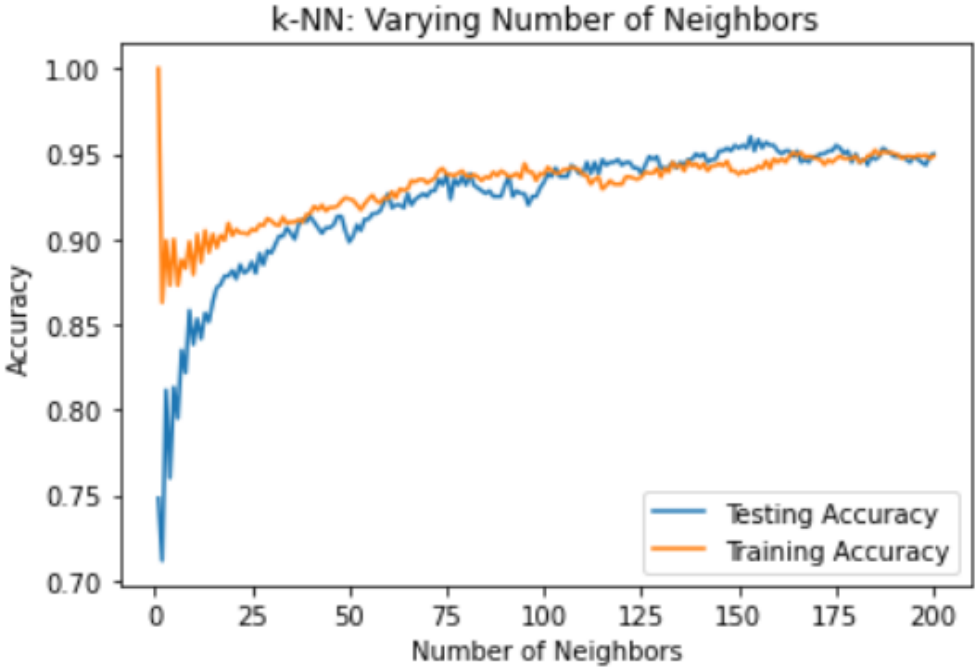



Fig 4.18: Testing and Training accuracy graph for KNN model.

- Scatter Plot Graphical Representation:

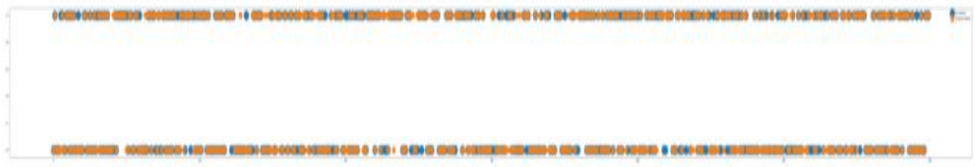


Fig 4.19: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of KNN model.

- Line Plot Graphical Representation:

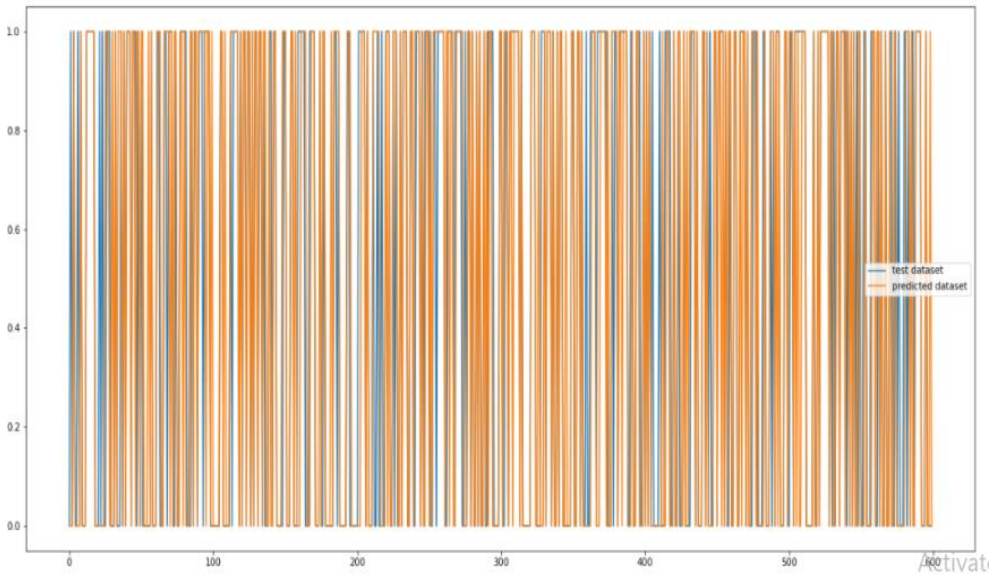



Fig 4.20: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of KNN model.

## 1.4 Decision Trees:

- Accuracy: 93.50%

- Tree Text Representation:


Fig 4.21: Textual representation of Decision Tree.

- Tree Graphical Representation:

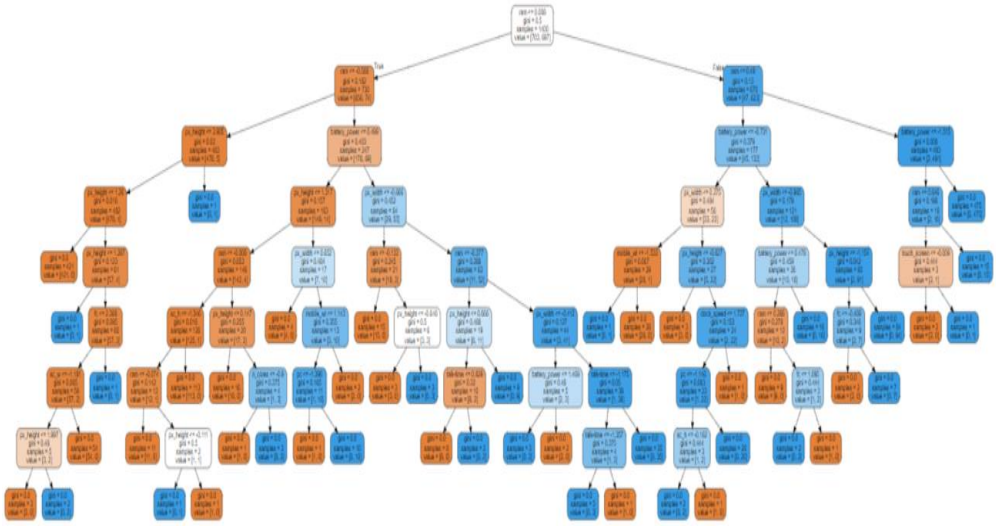


Fig 4.22: Graphical representation of Decision Tree.

- Scatter Plot Graphical Representation:

Fig 4.23: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Decision Tree model.

- Line Plot Graphical Representation:

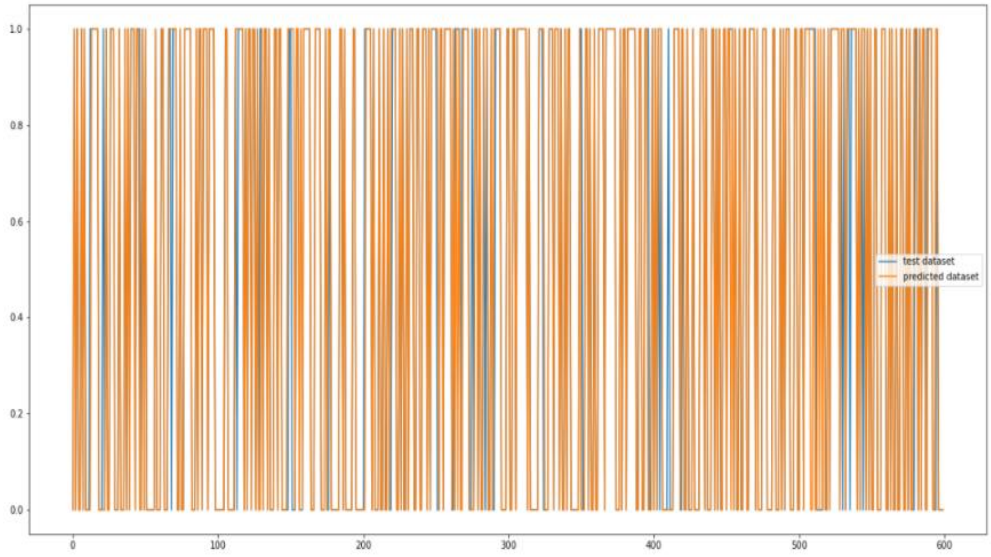



Fig 4.24: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Decision Tree model.

## 1.5 Random Forest Classifier or Random Decision Forest:

- Accuracy: 96.00%

- ROC Curve:

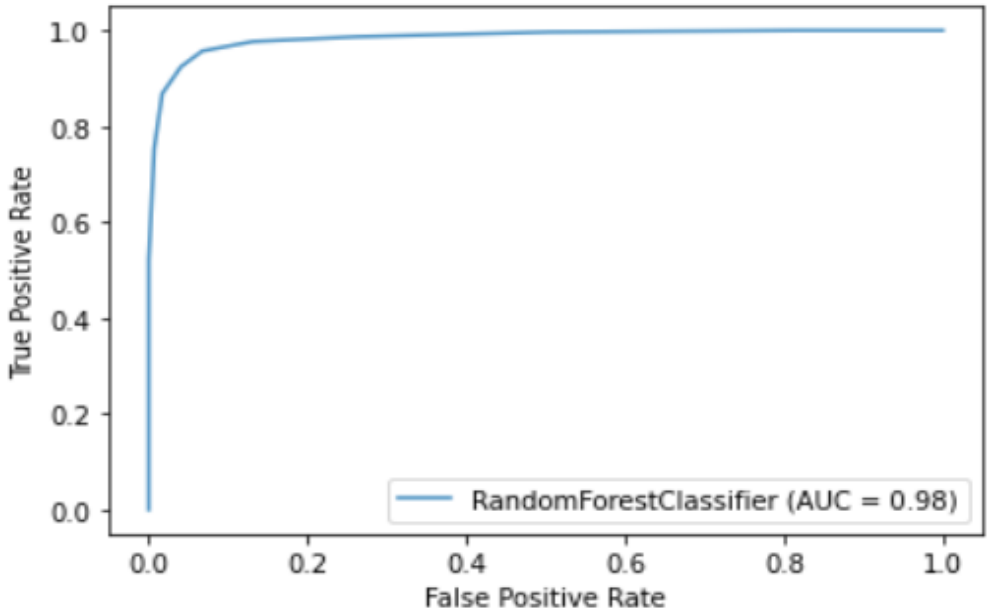



Fig 4.25: ROC curve for Random Forest.

- Scatter Plot Graphical Representation:

Fig 4.26: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Random Forest model.

- Line Plot Graphical Representation:

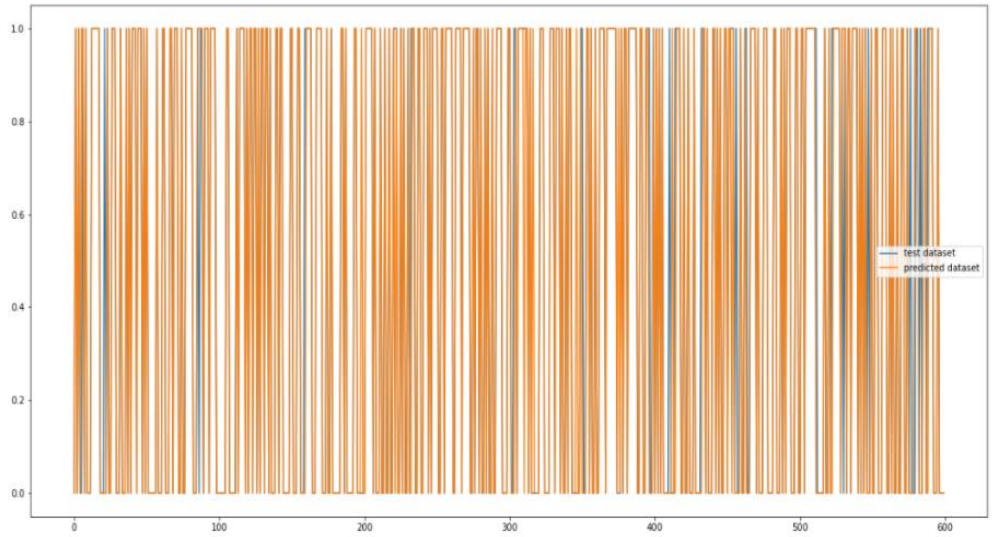



Fig 4.27: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Random Forest model.

## 1.6 Support Vector Machine:

- Accuracy: 98.50%

- Scatter Plot Graphical Representation:

Fig 4.28: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of SVM model.

- Line Plot Graphical Representation:

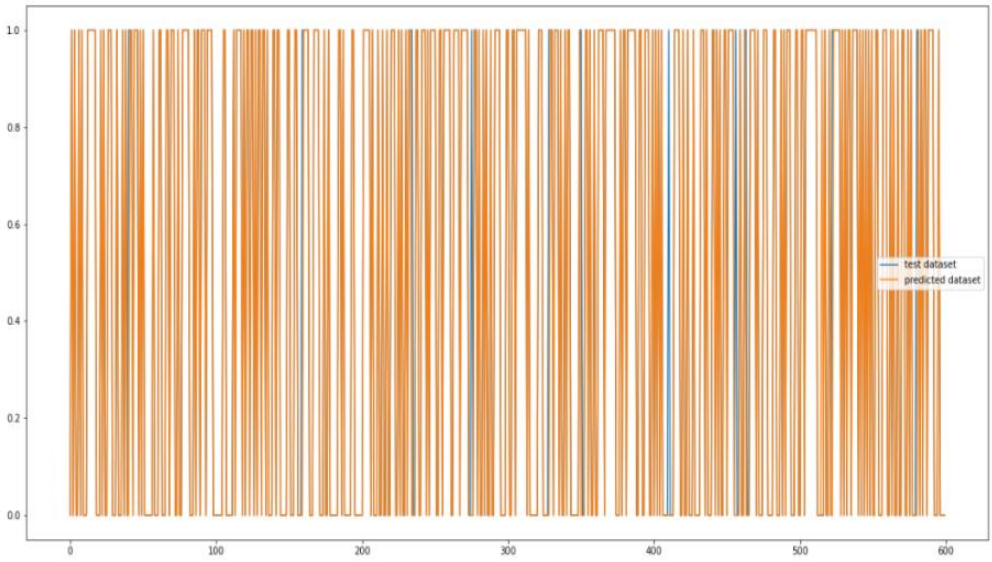



Fig 4.29: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of SVM model.

## 2. Multi Class Classification:

### 2.1 Logistic Regression:

- Accuracy: 91.50%

- Confusion Matrix:

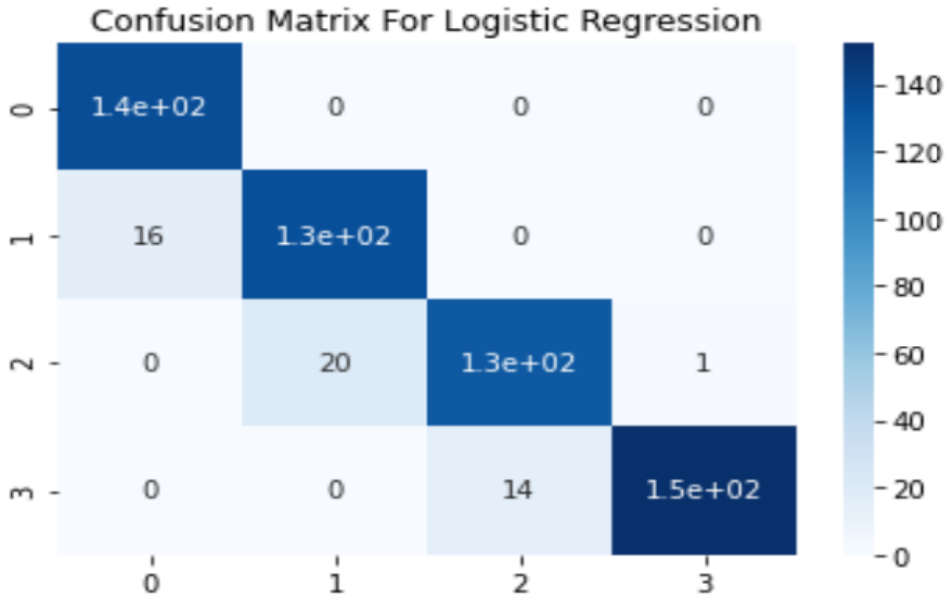



Fig 4.30: Confusion Matrix for Logistic model.

- Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.89 | 1.00 | 0.94 | 135 |
| 1 | 0.87 | 0.89 | 0.88 | 150 |
| 2 | 0.90 | 0.86 | 0.88 | 149 |
| 3 | 0.99 | 0.92 | 0.95 | 166 |
| accuracy | | | 0.92 | 600 |
| macro avg | 0.91 | 0.92 | 0.91 | 600 |
| weighted avg | 0.92 | 0.92 | 0.91 | 600 |

Fig 4.31: Classification Report for Logistic model.

- Scatter Plot Graphical Representation:

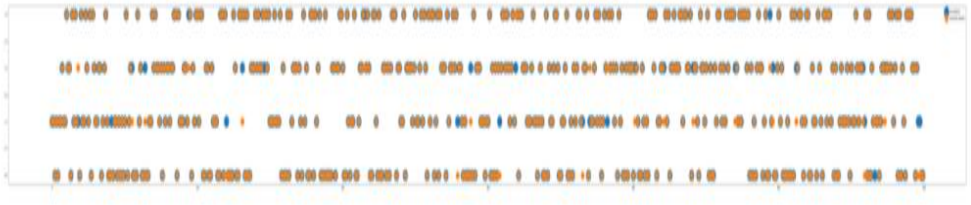


Fig 4.32: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Logistic Regression model.

- Line Plot Graphical Representation:

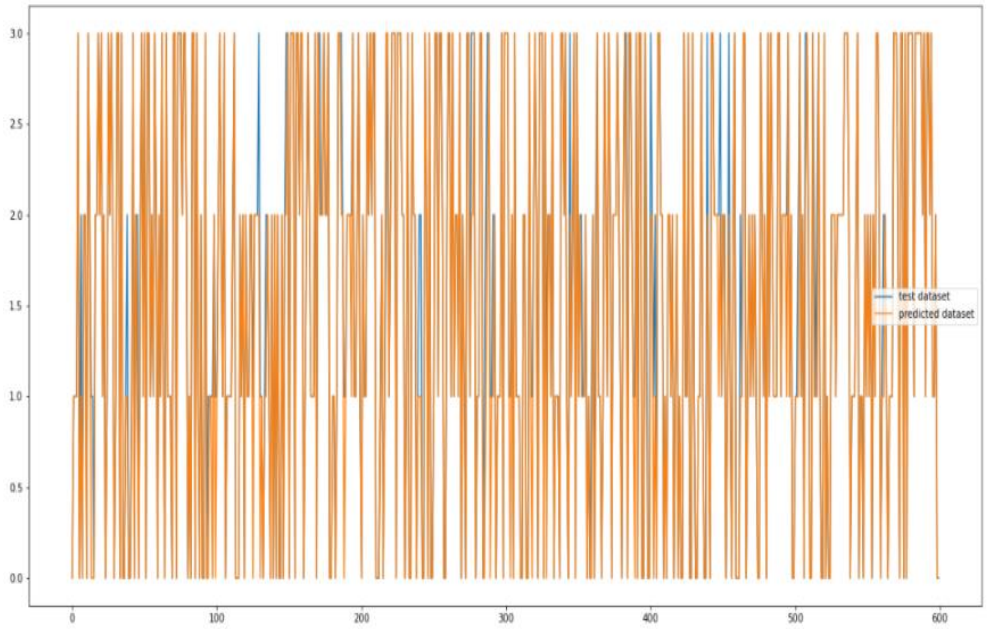



Fig 4.33: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Logistic Regression model.

## 2.2 K-Nearest Neighbor (or knn):

- Accuracy: 48.30%

- Error Rate Graphical Representation:

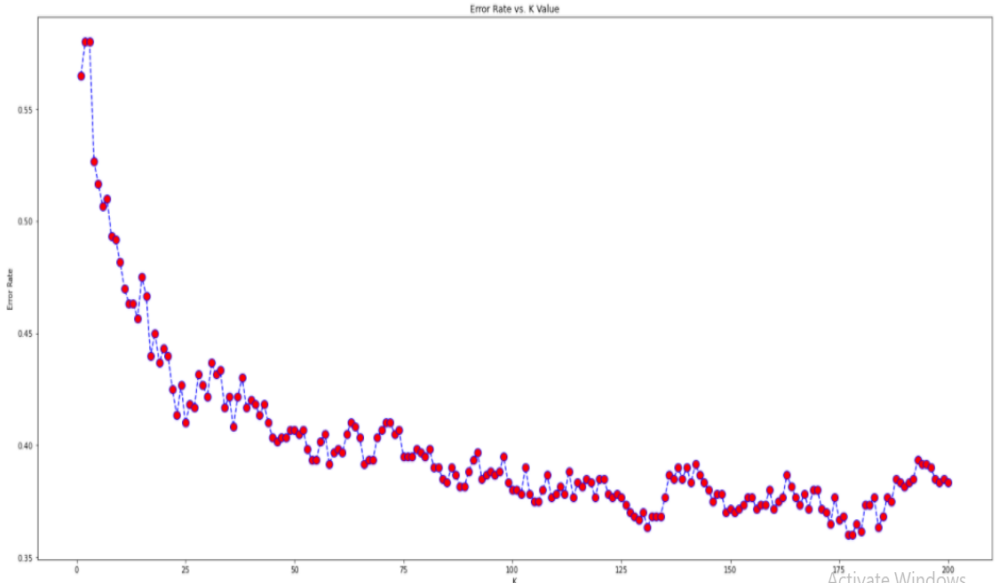



Fig 4.34: Error rate graph for KNN Model.

- Testing and Training Accuracy:

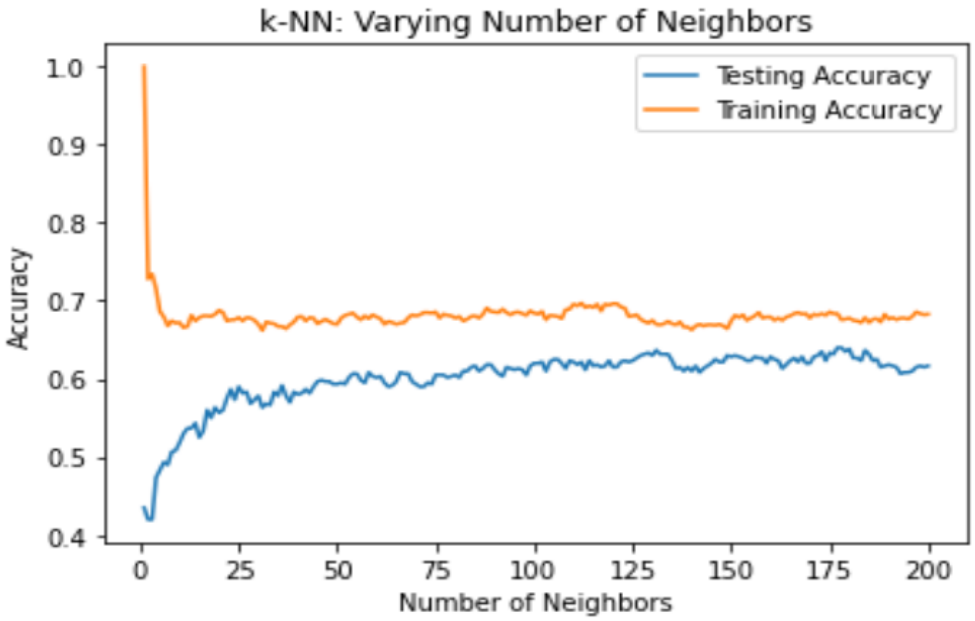



Fig 4.35: Testing and Training accuracy graph for KNN model.

- Confusion Matrix:

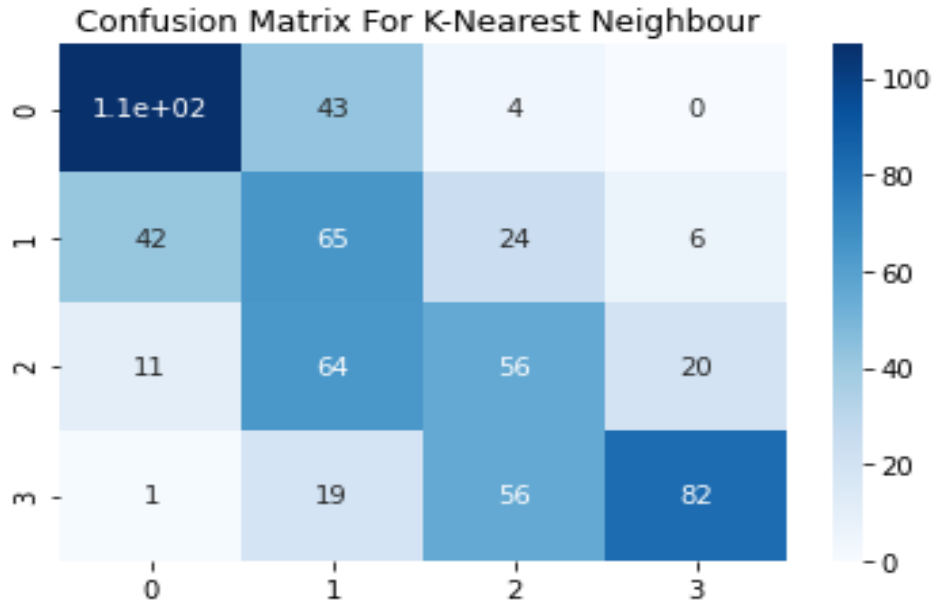



Fig 4.36: Confusion Matrix for KNN model.

- Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.66 | 0.69 | 0.68 | 154 |
| 1 | 0.34 | 0.47 | 0.40 | 137 |
| 2 | 0.40 | 0.37 | 0.38 | 151 |
| 3 | 0.76 | 0.52 | 0.62 | 158 |
| accuracy | | | 0.52 | 600 |
| macro avg | 0.54 | 0.51 | 0.52 | 600 |
| weighted avg | 0.55 | 0.52 | 0.52 | 600 |

Fig 4.37: Classification Report for KNN model.

- Scatter Plot Graphical Representation:

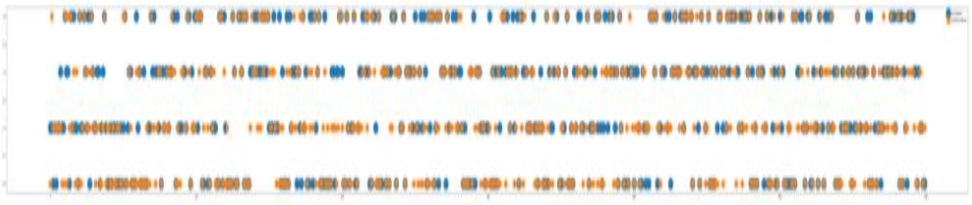


Fig 4.38: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of KNN model.

- Line Plot Graphical Representation:

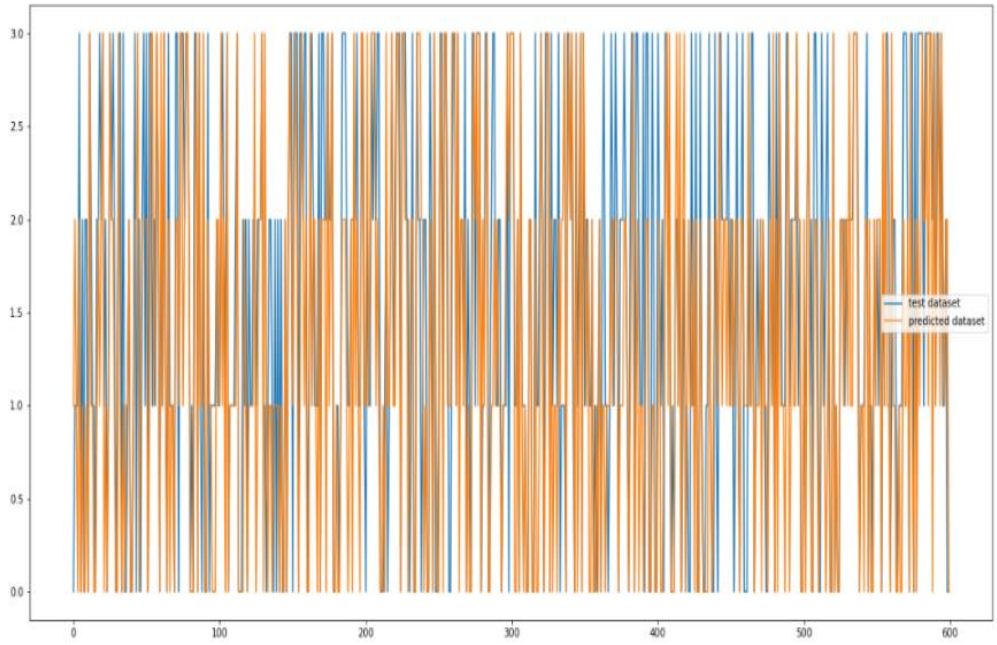



Fig 4.39: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of KNN model.

## 2.3 Decision Trees:

- Accuracy: 79.50%

- Tree Text Representation:

```
|--- feature_13 <= 0.13
|   |--- feature_13 <= -0.91
|   |   |--- feature_11 <= 1.82
|   |   |   |--- feature_0 <= 0.85
|   |   |   |   |--- feature_13 <= -0.96
|   |   |   |   |   |--- feature_6 <= -1.59
|   |   |   |   |   |   |--- feature_9 <= -1.11
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_9 >  -1.11
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_6 >  -1.59
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_13 >  -0.96
|   |   |   |   |   |--- feature_13 <= -0.95
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_13 >  -0.95
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- feature_0 >  0.85
|   |   |   |   |--- feature_11 <= 0.60
|   |   |   |   |   |--- feature_13 <= -1.14
|   |   |   |   |   |   |--- feature_15 <= 2.61
|   |   |   |   |   |   |   |--- feature_11 <= 0.18
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_11 >  0.18
|   |   |   |   |   |   |   |   |--- feature_13 <= -1.27
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_13 >  -1.27
|   |   |   |   |   |   |   |   |   |--- feature_12 <= -0.34
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_12 >  -0.34
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_15 >  2.61
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_13 >  -1.14
|   |   |   |   |   |   |--- feature_14 <= -1.28
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_14 >  -1.28
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_11 >  0.60
|   |   |   |   |   |--- feature_13 <= -1.59
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_13 >  -1.59
|   |   |   |   |   |   |--- feature_15 <= 0.02
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_15 >  0.02
|   |   |   |   |   |   |   |--- class: 0
|   |   |--- feature_11 >  1.82
|   |   |   |--- feature_10 <= -1.21
|   |   |   |   |--- feature_7 <= -0.51
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_7 >  -0.51
|   |   |   |   |   |--- class: 1
|   |   |   |--- feature_10 >  -1.21
|   |   |   |   |--- class: 1
|   |--- feature_13 >  -0.91
|   |   |--- feature_0 <= -0.42
|   |   |   |--- feature_13 <= -0.52
|   |   |   |   |--- feature_12 <= 1.11
|   |   |   |   |   |--- feature_11 <= 1.06
|   |   |   |   |   |   |--- feature_11 <= 0.46
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_11 >  0.46
|   |   |   |   |   |   |   |--- feature_15 <= 0.25
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_15 >  0.25
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_11 >  1.06
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_12 >  1.11
|   |   |   |   |   |--- class: 1
|   |   |   |--- feature_13 >  -0.52
|   |   |   |   |--- feature_12 <= -0.79
|   |   |   |   |   |--- feature_13 <= -0.24
|   |   |   |   |   |   |--- feature_0 <= -1.13
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_0 >  -1.13
|   |   |   |   |   |   |   |--- feature_4 <= 0.04
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_4 >  0.04
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_13 >  -0.24
|   |   |   |   |   |   |--- feature_14 <= 1.45
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_14 >  1.45
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_12 >  -0.79
|   |   |   |   |   |--- feature_8 <= -1.67
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- feature_8 >  -1.67
|   |   |   |   |   |   |--- feature_0 <= -1.66
|   |   |   |   |   |   |   |--- feature_10 <= 0.09
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_10 >  0.09
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_0 >  -1.66
|   |   |   |   |   |   |   |--- feature_0 <= -0.67
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_0 >  -0.67
|   |   |   |   |   |   |   |   |--- feature_0 <= -0.65
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- feature_0 >  -0.65
|   |   |   |   |   |   |   |   |   |--- feature_13 <= -0.08
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_13 >  -0.08
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= -0.80
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
```

Fig 4.40: Textual representation of Decision Tree.

- Tree Graphical Representation:

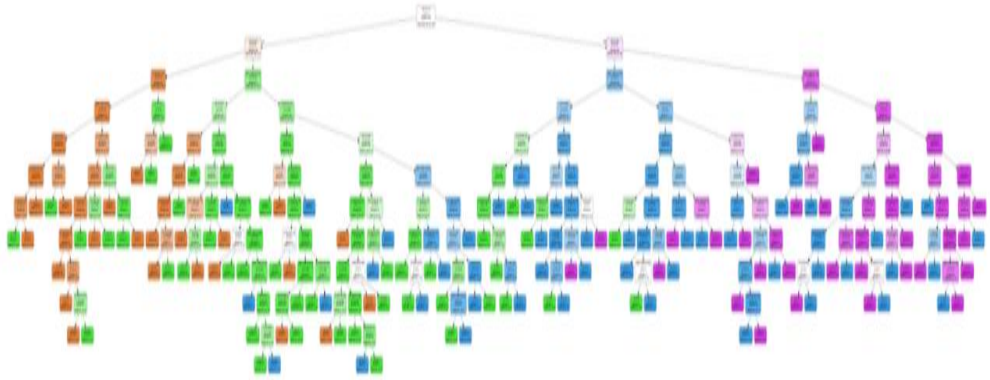


Fig 4.41: Graphical representation of Decision Tree.

- Confusion Matrix:

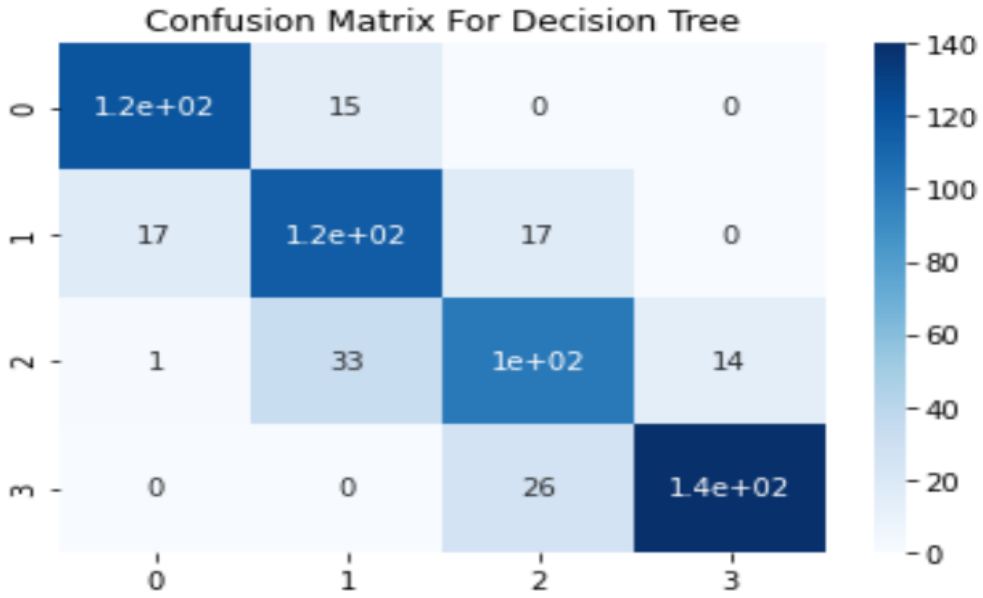



Fig 4.42: Confusion Matrix for Decision Tree model.

- Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.87 | 0.89 | 0.88 | 135 |
| 1 | 0.71 | 0.77 | 0.74 | 150 |
| 2 | 0.70 | 0.68 | 0.69 | 149 |
| 3 | 0.91 | 0.84 | 0.88 | 166 |
| accuracy | | | 0.80 | 600 |
| macro avg | 0.80 | 0.80 | 0.80 | 600 |
| weighted avg | 0.80 | 0.80 | 0.80 | 600 |

Fig 4.43: Classification Report for Decision Tree model.

- Scatter Plot Graphical Representation:

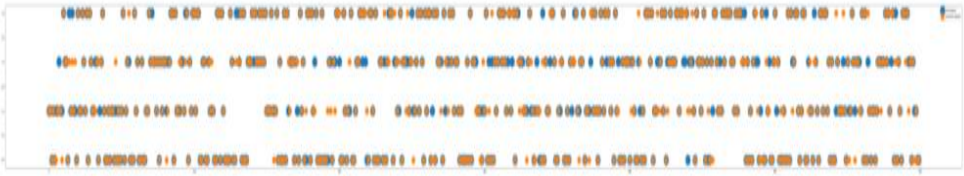


Fig 4.44: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Decision Tree model.

- Line Plot Graphical Representation:

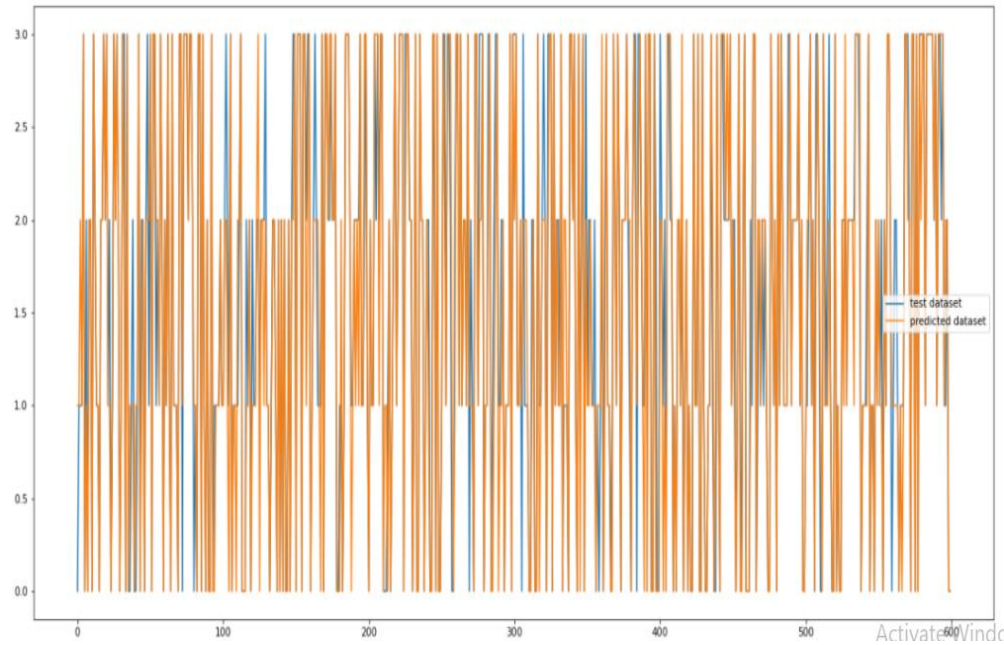



Fig 4.45: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Decision Tree model.

## 2.4 Random Forest Classifier or Random Decision Forest:

- Accuracy: 86.80%

- Confusion Matrix:

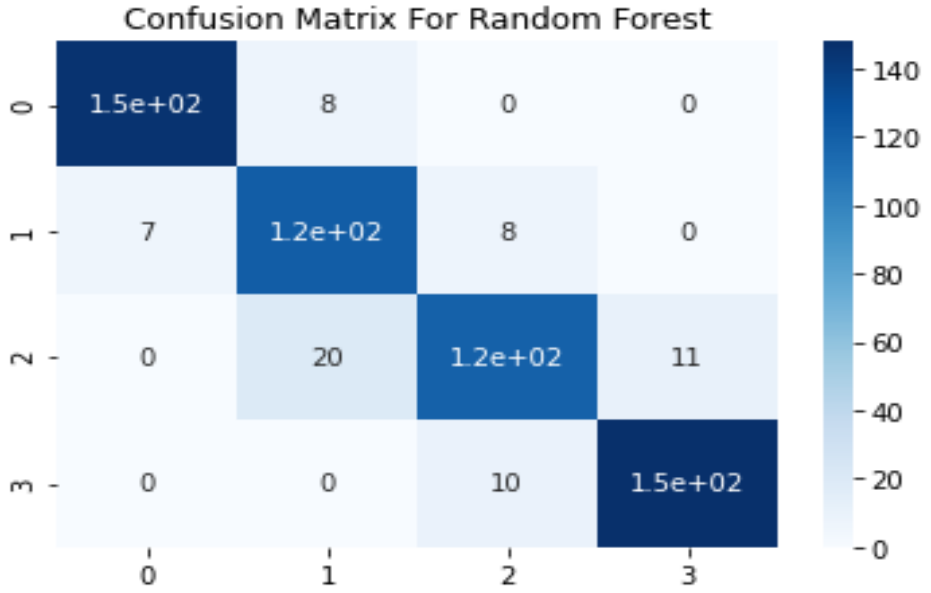



Fig 4.46: Confusion Matrix for Random Forest model.

- Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.95 | 0.95 | 154 |
| 1 | 0.81 | 0.89 | 0.85 | 137 |
| 2 | 0.87 | 0.79 | 0.83 | 151 |
| 3 | 0.93 | 0.94 | 0.93 | 158 |
| accuracy | | | 0.89 | 600 |
| macro avg | 0.89 | 0.89 | 0.89 | 600 |
| weighted avg | 0.89 | 0.89 | 0.89 | 600 |

Fig 4.47: Classification Report for Random Forest model.

- Scatter Plot Graphical Representation:

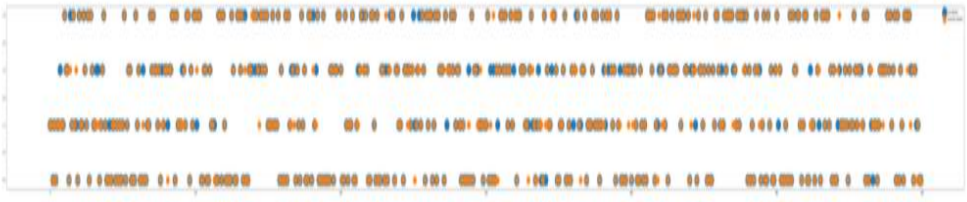


Fig 4.48: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Random Forest model.

- Line Plot Graphical Representation:

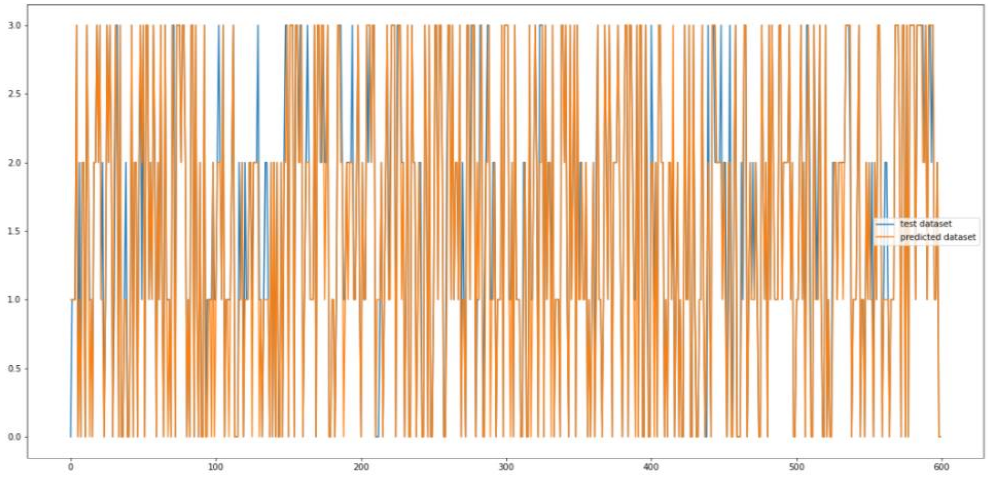



Fig 4.49: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of Random Forest model.

## 2.5 Support Vector Machine:

- Accuracy: 84.10%
- Confusion Matrix:

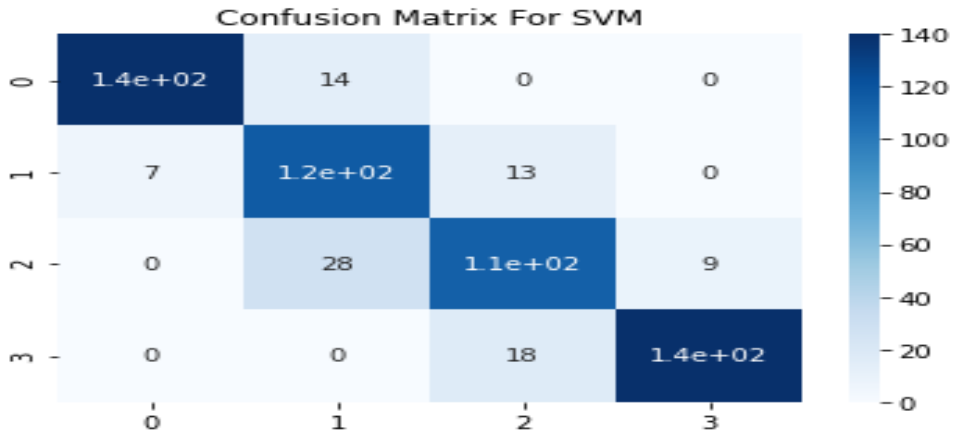



Fig 4.50: Confusion Matrix for SVM model.

- Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.91 | 0.93 | 154 |
| 1 | 0.74 | 0.85 | 0.79 | 137 |
| 2 | 0.79 | 0.75 | 0.77 | 151 |
| 3 | 0.94 | 0.89 | 0.91 | 158 |
| accuracy | | | 0.85 | 600 |
| macro avg | 0.85 | 0.85 | 0.85 | 600 |
| weighted avg | 0.86 | 0.85 | 0.85 | 600 |

Fig 4.51: Classification Report for SVM model.

- Scatter Plot Graphical Representation:

Fig 4.52: Scatter plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of SVM model.

- Line Plot Graphical Representation:

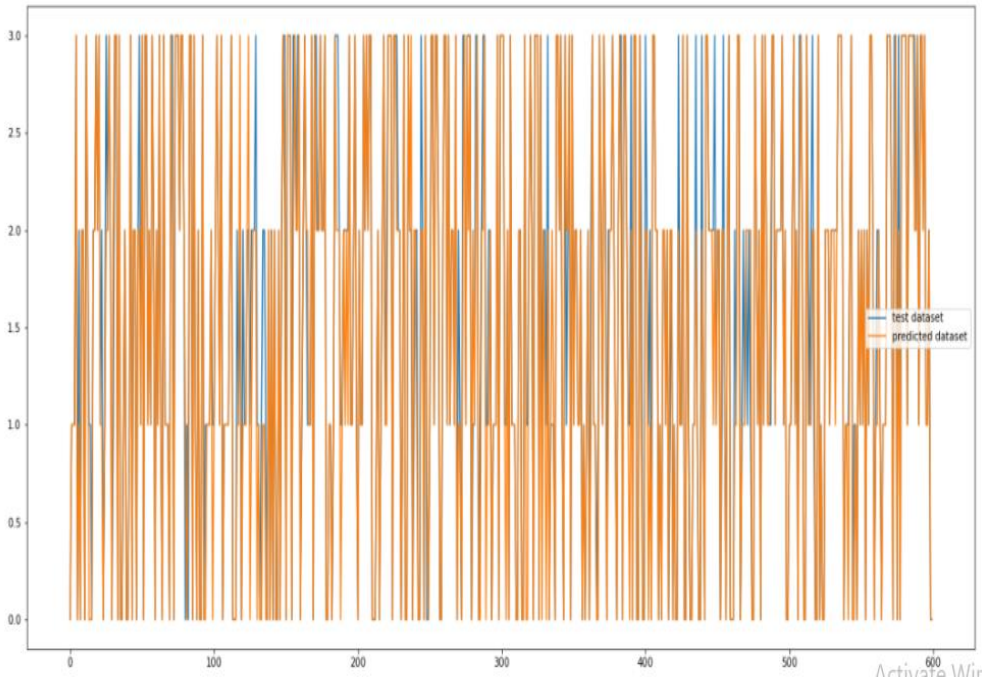
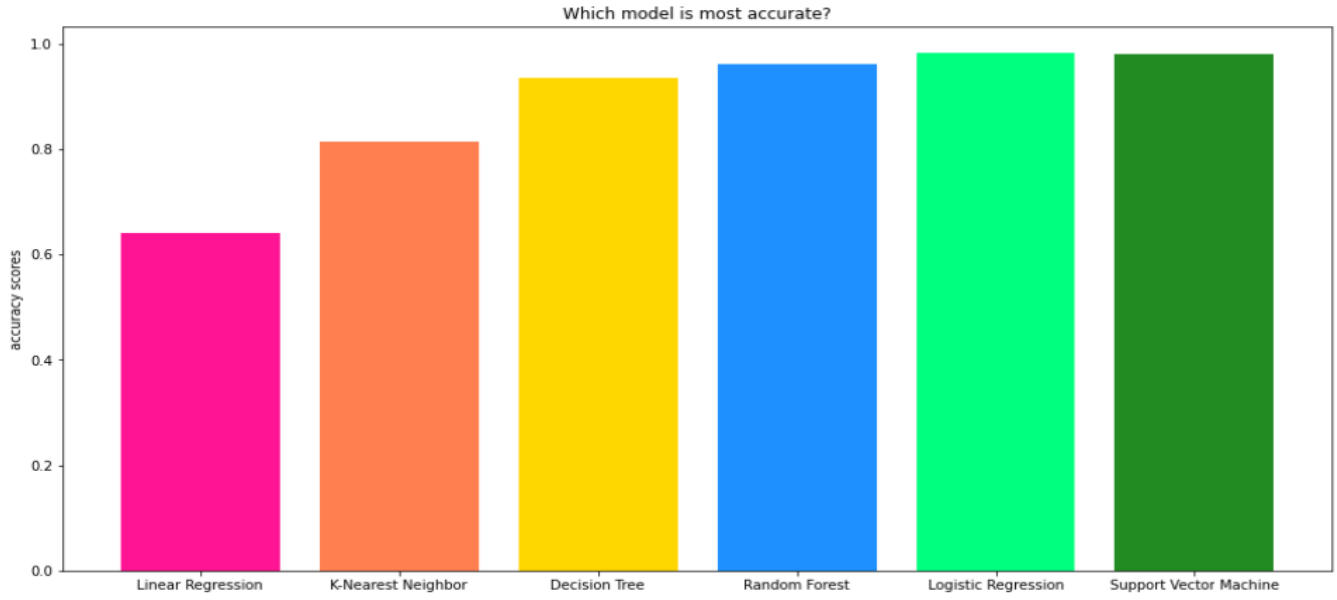



Fig 4.53: Line plot representation for showing the predicted output(in orange) and actual output(in blue) to visualize accuracy and error of SVM model.

# 3. Best Model Conclusion:

## 3.1 Binary Class Classification:

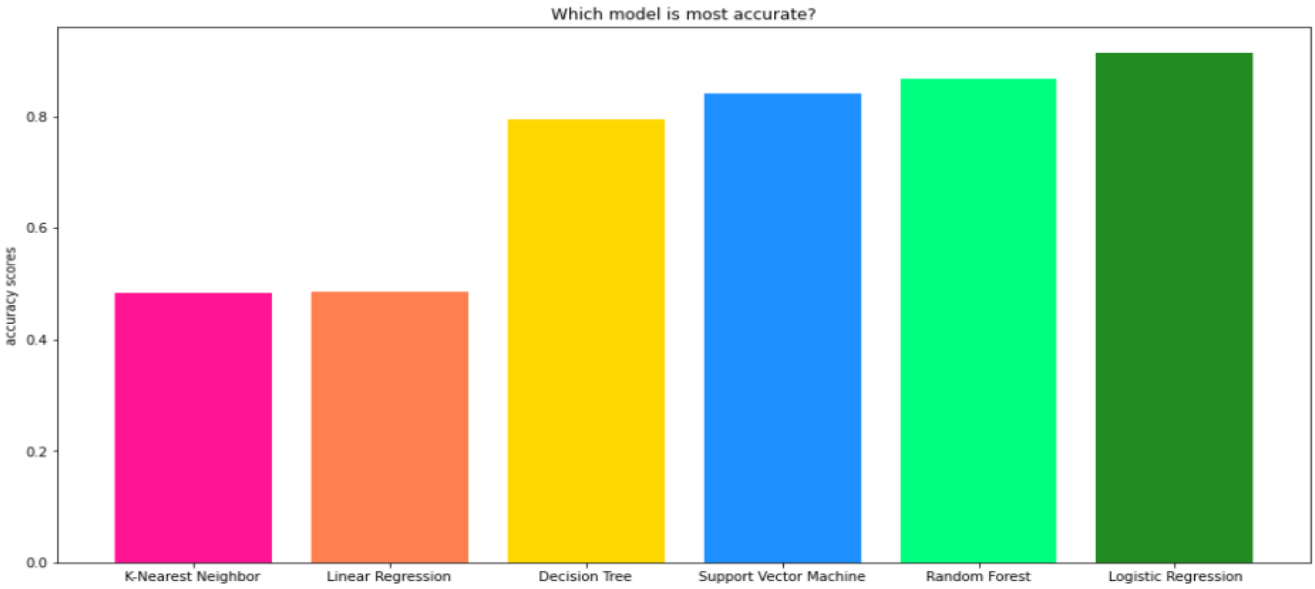



From the above bar graph we can conclude that:

| S.No. | Name Of Model | Accuracy (in percentage) |
|---|---|---|
| 1. | Linear Regression | 64.00 |
| 2. | K-Nearest Neighbor | 81.30 |
| 3. | Decision Tree | 93.50 |
| 4. | Random Forest | 96.00 |
| 5. | Logistic Regression | 98.30 |
| 6. | Support Vector Machine | 98.50 |

So, from the above we can conclude that Support Vector Machine performed with highest accuracy. So, this model will be used by us to predict outputs on test dataset for binary class classification.

## 3.2 Multi Class Classification:




From the above bar graph we can conclude that:

| S.No. | Name Of Model | Accuracy (in percentage) |
|---|---|---|
| 1. | K-Nearest Neighbor | 48.30 |
| 2. | Decision Tree | 79.50 |
| 3. | Support Vector Machine | 84.10 |
| 4. | Random Forest | 86.80 |
| 5. | Logistic Regression | 91.50 |

So, from the above we can conclude that Logistic Regression performed with highest accuracy. So, this model will be used by us to predict outputs on test dataset for multi-class classification.

# CHAPTER 5 Software Testing

## 5.1 Introduction:

In traditional software systems, humans write the logic which interacts with data to produce a desired behavior. Our software tests help ensure that this **written logic** aligns with the actual expected behavior.

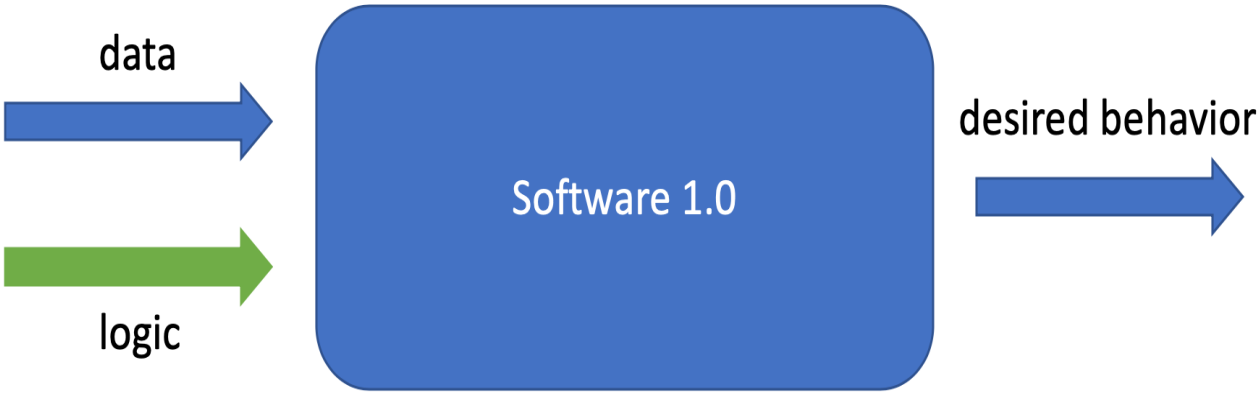



Fig 5.1: Traditional Software Systems

However, in machine learning systems, humans provide desired behavior as examples during training and the model optimization process produces the logic of the system. How do we ensure this **learned logic** is going to consistently produce our desired behavior?


Fig 5.2: Machine Learning Software Systems

## 5.2 Terms in Testing Fundamentals:

Let's start by looking at the best prices for testing traditional software systems and developing high-quality software.

A typical software testing suite will include:

- Unit tests which operate on atomic pieces of the codebase and can be run quickly during development,
- Regression tests replicate bugs that we've previously encountered and fixed,
- Integration tests which are typically longer-running tests that observe higher-level behaviors that leverage multiple components in the codebase,

And follow conventions such as:

- Don't merge code unless all tests are passing,
- Always write tests for newly introduced logic when contributing code,
- When contributing a bug fix, be sure to write a test to capture the bug and prevent future regressions.

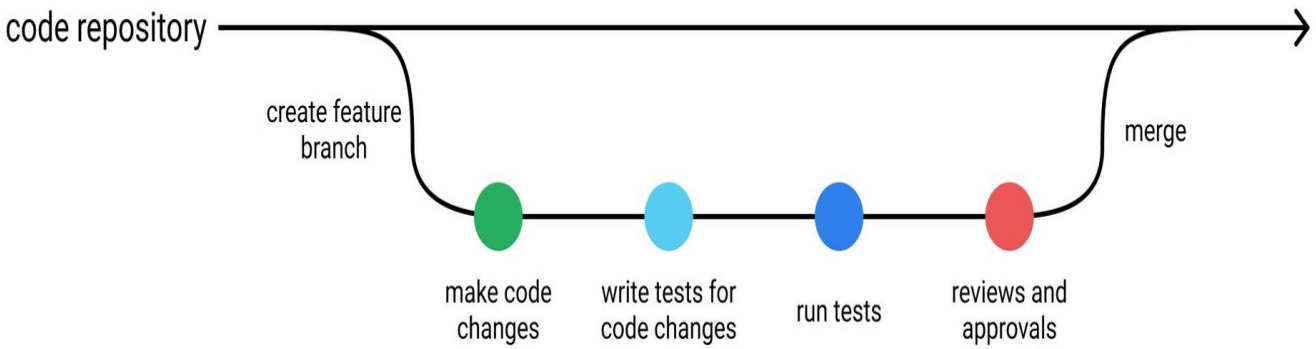



Fig 5.3: A typical workflow for software development.

When we run our testing suite against the new code, we'll get a report of the specific behaviors that we've written tests around and verify that our code changes don't affect the expected behaviour of the system. If a test fails, we'll know which specific behaviour is no longer aligned with our expected output.

Let's contrast this with a typical workflow for developing machine learning systems. After training a new model, we'll typically produce an evaluation report including:

- Performance of an established metric on a validation dataset,
- Plots such as precision-recall curves,

- Operational statistics such as inference speed,
- Example where the model was most confidently incorrect, and follow conventions,
- Save all of the hyper-parameters used to train the model,
- Only promote models which offer an improvement over the existing model (or baseline) when evaluated on the same dataset.

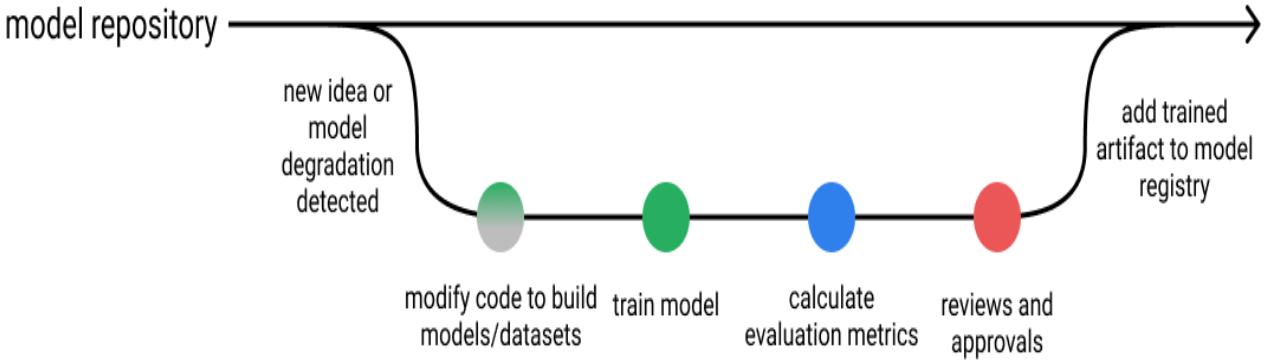



Fig 5.4: A typical workflow for model development.

When reviewing a new machine learning model, we'll inspect metrics and plots which summarize model performance over a validation dataset. We're able to compare performance between multiple models and make relative judgements, but we're not immediately able to characterize specific model behaviors. For example, figuring out where the model is failing usually requires additional investigative work; one common practice here is to look through a list of the top most egregious model errors on the validation dataset and manually categorize these failure modes.

While traditional software tests have metrics such as the lines of code covered when running tests, this becomes harder to quantify when you shift your application logic from lines of code to parameters of a machine learning model.

## 5.3 Future Scope of the Project:

More sophisticated artificial intelligence techniques can be used to maximized the accuracy and predict the accurate price of the products.

Software or Mobile app can be developed that will predict the market price of any new launched product.

To achieve maximum accuracy and predict more accurate, more and more instances should be added to the data set. And selecting more appropriate features can also increase the accuracy. So data set should be large and more appropriate features should be selected to achieve higher accuracy.

More sophisticated artificial intelligence techniques can be used to maximized the accuracy and predict the accurate price of the products.

Software or Mobile app can be developed that will predict the market price of any new launched product. To achieve maximum accuracy and predict more accurate, more and more instances should be added to the data set. And selecting more appropriate features can also increase the accuracy. So data set should be large and more appropriate features should be selected to achieve higher accuracy.

# CHAPTER 6 Conclusion

This work can be concluded with the comparable results of both Feature selection algorithms and classifier. This combination has achieved maximum accuracy and selected minimum but most appropriate features. It is important to note that in Forward selection by adding irrelevant or redundant features to the data set decreases the efficiency of both classifiers. While in backward selection if we remove any important feature from the data set, its efficiency decreases. The main reason of low accuracy rate is low number of instances in the data set. One more thing should also be considered while working that converting a regression problem into classification problem introduces more error.

The logistic regression algorithm and SVM were found to give the most accuracy of 81%. Then, the prediction of the price is done using logistic regression. This is just a preliminary paper. The future work to be done on this paper lies in predicting the approximate price of a second-hand phone sale. The same project can further be extended to predict the exact price of the phone instead of the discrete values that is being predicted in this paper. By further pursuing this project, we will be able to help people to spend money wisely and also mine other data like the how the price is affected by the brand, how long a phone can work, etc.

# CHAPTER 7 Summary

This project is based on the current market price of various mobile phones, their wear resistance, drop resistance, charging time, battery life, communication stability, camera effect, design, memory size, whether to buy again as dependent variables, they are, input variables. The sales forecast levels of different mobile phones are the output variables in this article. This article also applies SVM to establish the forecasting model of various mobile phone sales prospects. The predicted value is basically consistent with the actual sales value announced by each mobile phone manufacturer. This model can provide guidance for product configuration and sales for various mobile phone manufacturers, which has certain practical value.

This kind of prediction will help companies estimate price of mobiles to give tough competition to other mobile manufacturer. Also it will be useful for Consumers to verify that they are paying best price for a mobile.

# CHAPTER 8 References


1. https://www.kaggle.com/iabhishekofficial/mobile-price-classification#train.csv
2. Quader, N., Ganim M.O., Chaki, D., Ali, M.H.: A machine learning approach to predict movie box-office success. In: 2017 20th International Conference of Computer and Information Technology (ICCIT), pp. 1–7. IEEE (2017)
3. Mansouri, S.S., Karvelis, P., Georgoulas, G., Nikolakopoulos, G.: Remaining useful battery life prediction for UAVs based on machine learning. IFAC-Papers OnLine 50(1), 4727–4732 (2017)
4. Usmani, M., Adil, S.H., Raza, K., Ali, S.S.: Stock market prediction using machine learning techniques. In: 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), pp. 322–327. IEEE (2016)
5. Ramadhani, A.M., Goo, H.S.: Twitter sentiment analysis using deep learning methods. In: 7th International Annual Engineering Seminar (InAES) pp. 1–4. IEEE (2017)
6. Jayaraman, S., Choudhury, T., Kumar, P.: Analysis of classification models based on cuisine prediction using machine learning. In: International Conference on Smart Technologies For Smart Nation (SmartTechCon) pp. 1485–1490. IEEE (2017)
7. Fu, C., Zheng, Y., Li, S., Xuan, Q., Ruan, Z.: Predicting the popularity of tags in StackExchange QA communities. In: 2017 International Workshop on 2017 Complex Systems and Networks (IWCSN) pp. 90–95. IEEE (2017)
8. Portugal, I., Alencar, P., Cowan, D.: The use of machine learning algorithms in recommender systems: a systematic review. Expert Syst. Appl. 1(97), 205–227 (2018)
9. Kara, Y., Boyacioglu, M.A., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. Expert Syst. Appl. 38(5), 5311–5319 (2011)
10. Lymperopoulos, I.N.: Predicting the popularity growth of online content: model and algorithm. Inf. Sci. 10(369), 585–613 (2016)
11. Iranitalab, A., Khattak, A.: Comparison of four statistical and machine learning methods for crash severity prediction. Accid. Anal. Prev. 30(108), 27–36 (2017)
12. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
13. https://thedatalass.com/2018/04/17/random-forest/
14. https://www.researchgate.net/figure/Figure-3-SVM-classification-scheme-H-is-theclassification-hyperplane-W-is-the-normal_fig3_286268965
15. https://en.wikipedia.org/wiki/Radial_basis_function_kernel