“Mobile Price Range Prediction”

Capstone Project

Submitted To

AlmaBetter

Submitted By:

Vikas panchal
Email id: - panchalvicky501@gmail.com

Naveen Kumar batta
Email id: - naveenbatta4587@gmail.com

Abstract

During the purchase of mobile phones, various features like memory, display, battery, camera, etc., are considered. People fail to make correct decisions, due to the non-availability of necessary resources to cross-validate the price. To address this issue, a machine learning model is developed using the data related to the key features of the mobile phone. The developed model is then used to predict the price range of the new mobile phone. Three machine learning algorithms namely Support Vector Machine (SVM), Random Forest Classifier (RFC), Logistic Regression are used to train the model and predict the output as low, medium, high or very high. The dataset used in this study is taken from Kaggle platform. In order to improve the classification accuracy, Chi-Squared based feature selection method is used. Among 21 features available in the dataset, only top 10 features namely RAM, pixel height, battery power, pixel width, mobile weight, internal memory, screen width, talk time, front camera and screen height are selected and used to train the model. Before applying feature selection, the accuracy obtained using SVM, RFC and Logistic Regression is 95%, 83% and 76% respectively. After feature selection, the accuracy of SVM, RFC and Logistic Regression improved to 97%, 87% and 81% respectively. From the experiments conducted, it is found that SVM gave superior performance when compared to other two classifiers.

Table of Contents

Chapter 1. Introduction:

1.0 Introduction:

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 orried and thinks “If he would be able to purchase something with given specifications or not”. Machine learning provides us best techniques for artificial intelligence like classification, regression, supervised learning, and unsupervised learning and many more Mobile now a days is one of the most selling and purchasing device. Every day new mobiles with new version and more features are launched. Hundreds and thousands of mobiles are sold and purchased on daily basis. So here the mobile price class prediction is a case study for the given type of problem i.e., finding optimal product. The same work can be done to estimate real price of all products like cars, bikes, generators, motors, food items, medicine etc. Many features are very important to be considered to estimate price of mobile. For example, Processor of the mobile. Battery timing is also very important in today’s busy schedule of human being. Size and thickness of the mobile are also important decision factors. Internal memory, Camera pixels, and video quality must be under consideration.

Internet browsing is also one of the most important constraints in this technological era of 21st century. And so is the list of many features based upon those, mobile price is decided. So, we will use many of above-mentioned features to classify whether the mobile would be very low, Medium, and High or very High.

1.2 Problem statement:

In the competitive mobile-phone market companies want to understand sales data of mobile-phones and factors which drive the prices.

The objective is to find out some relation between features of a mobile phone (eg: - RAM, Internal Memory, etc.) and its selling price. In this problem, we do not have to predict the actual price but a price range indicating how high the price is.

1.3 Objective:

- The main objective is to create a machine learning model to recommend relevant Mobile according to Price range.

- In addition to the ML Model prediction, we also have taken into Battery, Ram, camera Quality relevant recommend a best price.

Chapter 2

2. Data Summary –

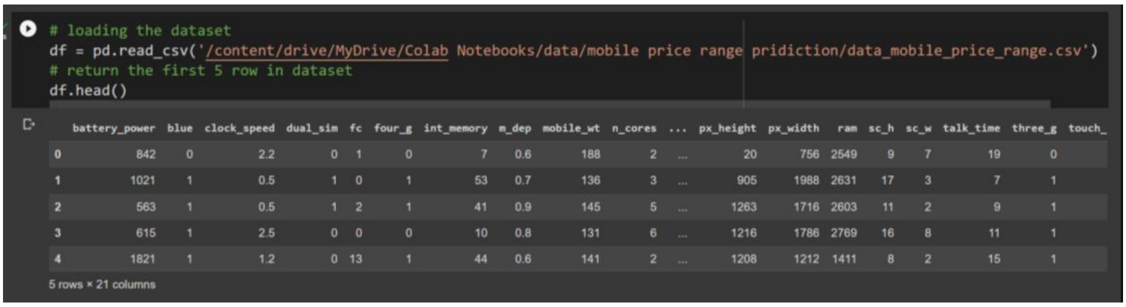
2.1 Import the Dataset –

Before building any machine learning model, it is vital to understand what the data is, and what are we trying to achieve. Data exploration reveals the hidden trends and insights and data pre-processing makes the data ready for use by ML algorithms.

So, let’s begin. . .

To proceed with the problem dealing first we will load our dataset that is given to us in a .csv file into a **data frame**.

Mount the drive and load the csv file into a data frame.



```
# loading the dataset
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/data/mobile price range pridiction/data_mobile_price_range.csv')
# return the first 5 row in dataset
df.head()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	

5 rows × 21 columns

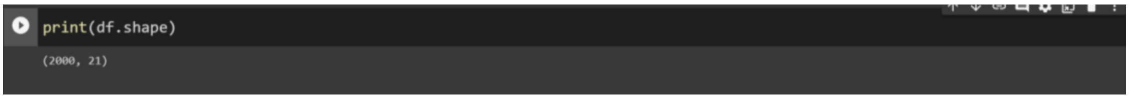
Fig 2. import dataset.

2.2 Exploratory Data Analysis (EDA) –

The primary goal of EDA is to support the analysis of data prior to making any conclusions. It may aid in the detection of apparent errors, as well as a deeper understanding of data patterns, the detection of outliers or anomalous events, and the discovery of interesting relationships between variables.

Duplication: In our further analysis we found that the dataset has no duplicate entries.

2.2.1 Dimension of dataset:



```
print(df.shape)
```

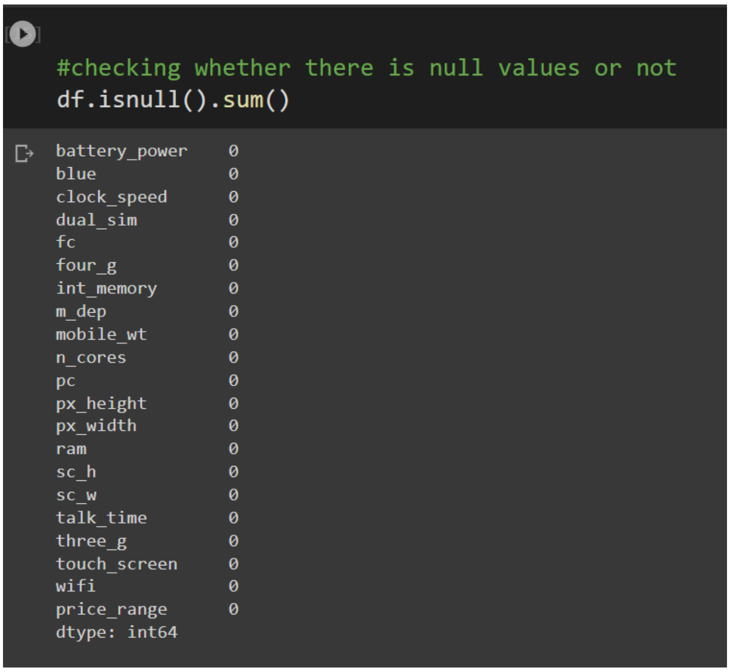
```
(2000, 21)
```

2.2.2 Data-Description:

The data contains information regarding mobile phone features, specifications etc. and their price range. The various features and information can be used to predict the price range of a mobile phone.

- Battery power - Total energy a battery can store in one time measured in mAh.
- Blue - Has Bluetooth or not
- Clock_speed - speed at which microprocessor executes instructions.
- Dual_sim - Has dual sim support or not
- Fc - Front Camera megapixels
- Four_g - Has 4G or not
- Int_memory - Internal Memory in Gigabytes
- M_dep - Mobile Depth in cm
- Mobile_wt - Weight of mobile phone
- N_cores - Number of cores of processor
- Pc - Primary Camera megapixels
- Px_height - Pixel Resolution Height
- Px_width - Pixel Resolution Width
- Ram - Random Access Memory in Megabytes
- Sc_h - Screen Height of mobile in cm
- Sc_w - Screen Width of mobile in cm
- Talk_time - longest time that a single battery charge will last when you are
- Three_g - Has 3G or not
- Touch_screen - Has touch screen or not
- Wifi - Has Wi-Fi or not
- Price_range - This is the target variable with value of 0(low cost), 1(medium cost),
- 2(high cost) and 3(very high cost).

Missing Values:- Find the missing values given dataset.

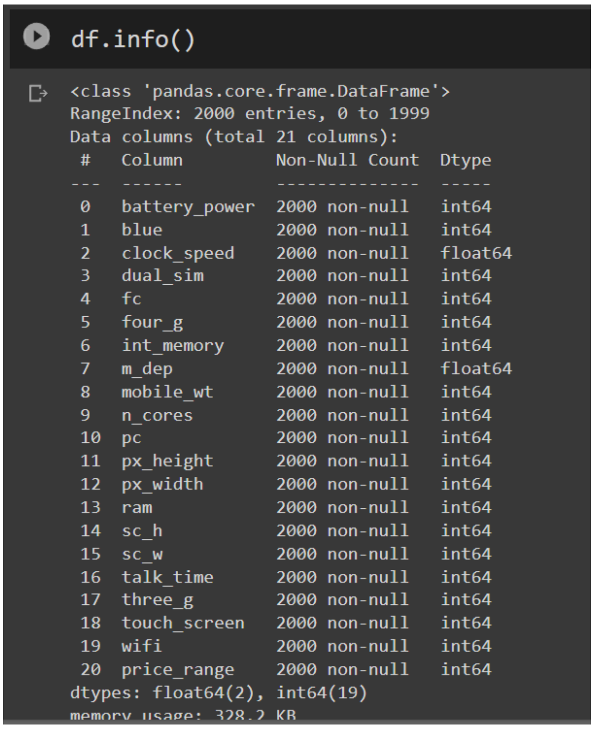


```
#checking whether there is null values or not
df.isnull().sum()
```

```
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

2.2.3 Data Preprocessing :

The info() method prints information about the Mobile Price Range Data Frame. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

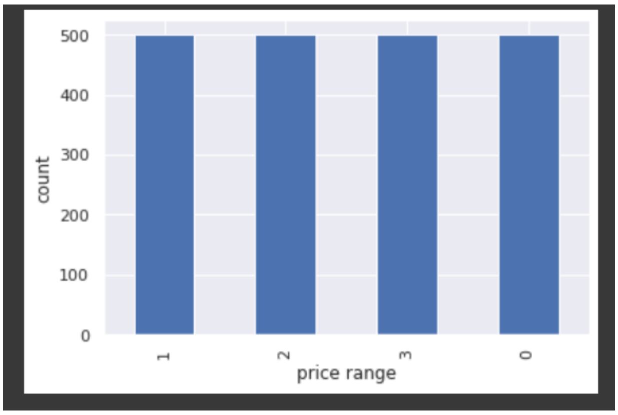


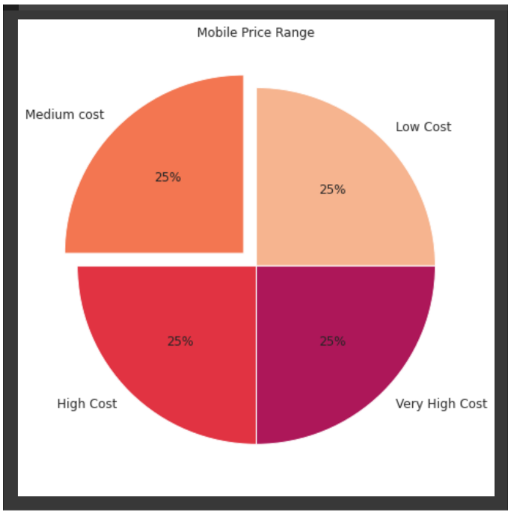
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

2.2.4 Price Range:

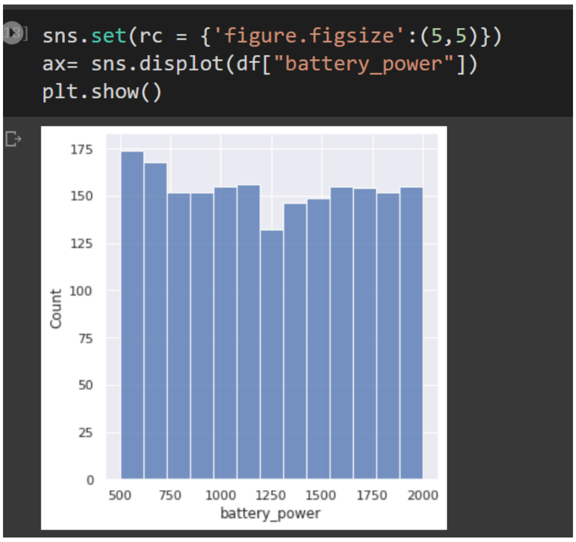
we can see that ,this pie chart there are mobile phones in 4 price ranges. the number of elements is almost similar.





2.2.5 Battery Count:

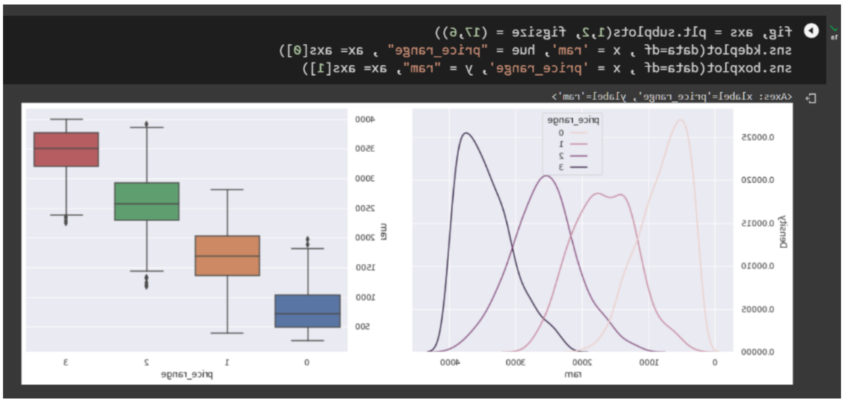
this plot shows how the battery mAh is spread. there is a gradual increase as the price range increases.



```
sns.set(rc = {'figure.figsize':(5,5)})
ax= sns.displot(df["battery_power"])
plt.show()
```

2.2.6 Ram Count:

Ram has continuous increase with price range while moving from Low cost to Very high cost.

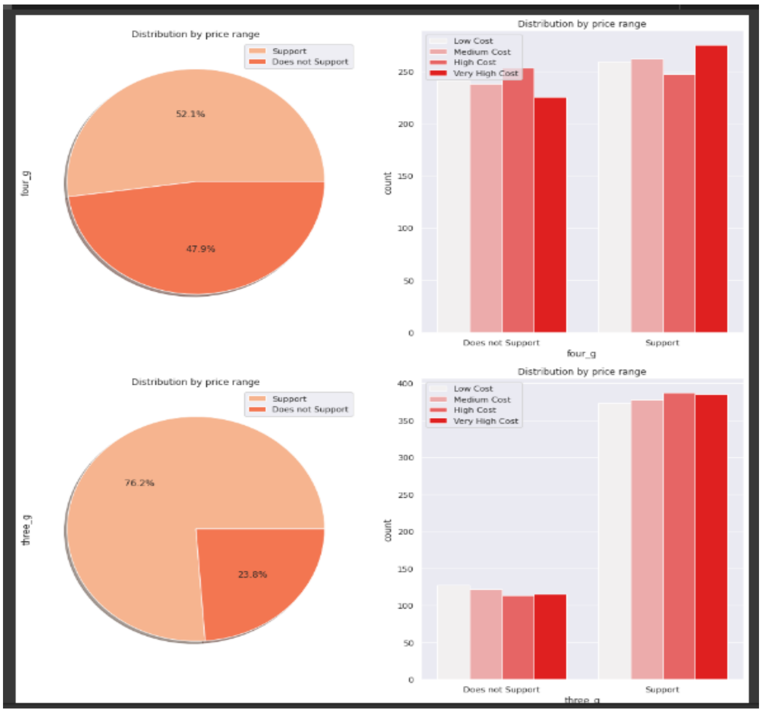


```
fig, axs = plt.subplots(1,2, figsize = (17,6))
sns.kdeplot(data=df, x = 'ram', hue = "price_range", ax= axs[0])
sns.boxplot(data=df, x = 'price_range', y = "ram", ax= axs[1])
```

```
<Axes: xlabel='price_range', ylabel='ram'>
```

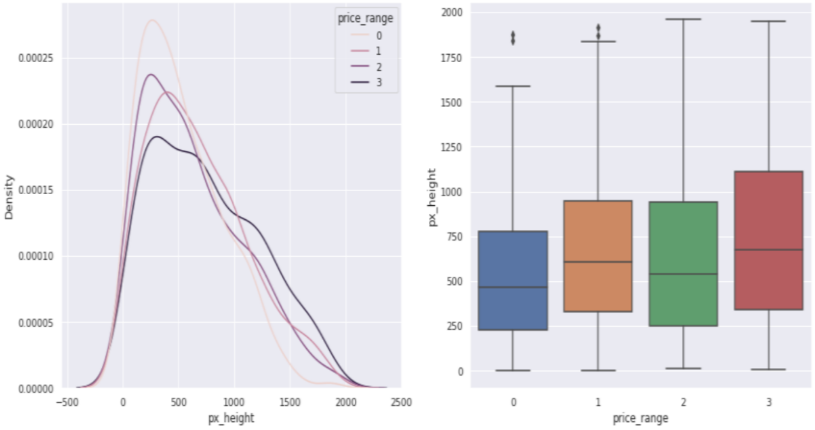
2.2.7 Mobile Network (3G & 4G):

- 50% of the phones support 4_g and 76% of phones support 3_g
- Distribution of price range almost similar of supported and unsupported feature in 4G . So that is not used full of prediction.
- feature 'Three G' play an important.



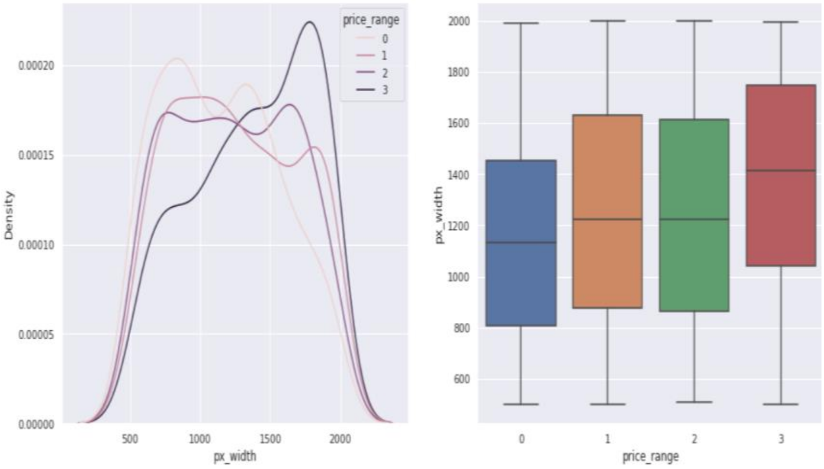
2.2.8 Screen height:

There is not a continuous increase in pixel width as we move from Low cost to Very high cost. Mobiles with 'Medium cost 'and 'High cost' has almost equal pixel width. so we can say that it would be a driving factor in deciding price range.



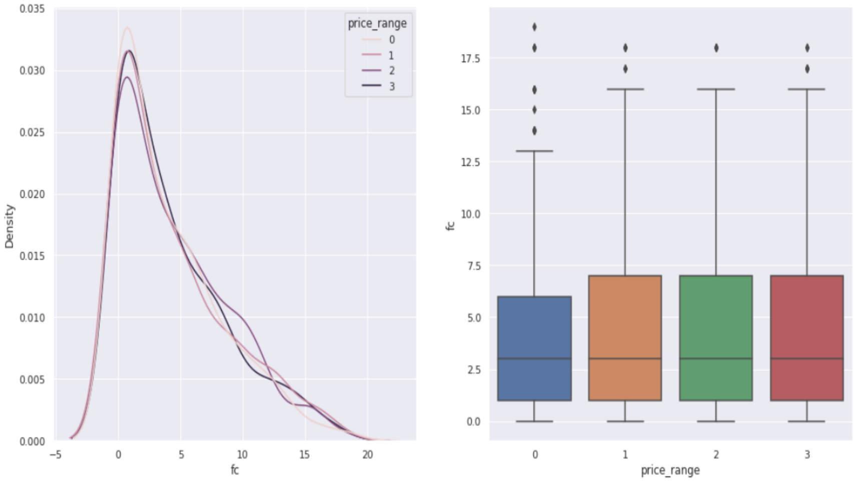
2.2.9 Screen width:

Pixel height is almost similar as we move from Low cost to Very high cost. Little variation in pixel height.



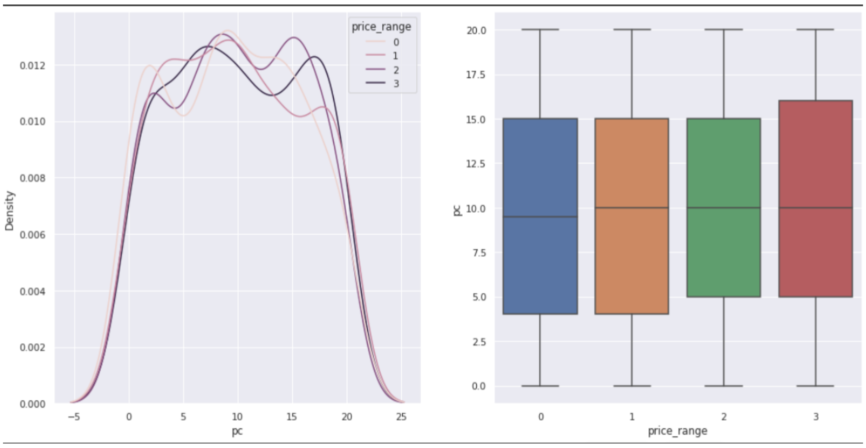
2.2.10 Front camera :

This features distribution is almost similar along all the price ranges variable, it may not be helpful in making predictions.



2.2.11 Primary camera :

Primary camera megapixels are showing a little variation along the target categories, which is a good sign for prediction.

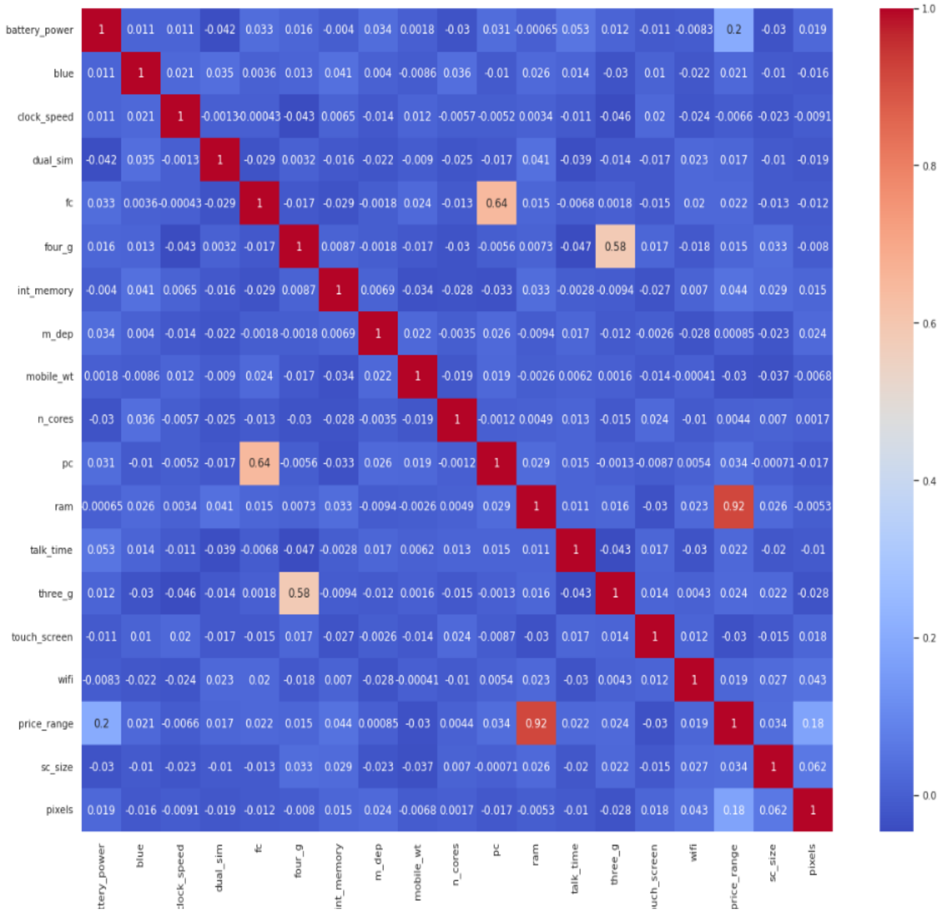


2.2.12 Heat Map:–

- RAM and price range shows high correlation which is a good sign, it signifies that RAM will play major deciding factor in estimating the price range.

- There is some collinearity in feature pairs ('pc', 'fc') and ('px_width', 'px_height'). Both correlations are justified since there are good chances that if front camera of a phone is good, the back camera would also be good.

- Also, if px_height increases, pixel width also increases, that means the overall pixels in the screen. We can replace these two features with one feature. Front Camera megapixels and Primary camera megapixels are different entities despite of showing collinearity. So we'll be keeping them as they are.


battery_power 1 0.011 0.011 -0.042 0.033 0.016 -0.004 0.034 0.0018 -0.03 0.031 -0.00065 0.053 0.012 -0.011 -0.0083 0.2 -0.03 0.019
blue 0.011 1 0.021 0.035 0.0036 0.013 0.041 0.004 -0.0086 0.036 -0.01 0.026 0.014 -0.03 0.01 -0.022 0.021 -0.01 -0.016
clock_speed 0.011 0.021 1 -0.0013 -0.00043 -0.043 0.0065 -0.014 0.012 -0.0057 -0.0052 0.0034 -0.011 -0.046 0.02 -0.024 -0.0066 -0.023 -0.0091
dual_sim -0.042 0.035 -0.0013 1 -0.029 0.0032 -0.016 -0.022 -0.009 -0.025 -0.017 0.041 -0.039 -0.014 -0.017 0.023 0.017 -0.01 -0.019
fc 0.033 0.0036 -0.00043 -0.029 1 -0.017 -0.029 -0.0018 0.024 -0.013 0.64 0.015 -0.0068 0.0018 -0.015 0.02 0.022 -0.013 -0.012
four_g 0.016 0.013 -0.043 0.0032 -0.017 1 0.0087 -0.0018 -0.017 -0.03 -0.0056 0.0073 -0.047 0.58 0.017 -0.018 0.015 0.033 -0.008
int_memory -0.004 0.041 0.0065 -0.016 -0.029 0.0087 1 0.0069 -0.034 -0.028 -0.033 0.033 -0.0028 -0.0094 -0.027 0.007 0.044 0.029 0.015
m_dep 0.034 0.004 -0.014 -0.022 -0.0018 -0.0018 0.0069 1 0.022 -0.0035 0.026 -0.0094 0.017 -0.012 -0.0026 -0.028 0.00085 -0.023 0.024
mobile_wt 0.0018 -0.0086 0.012 -0.009 0.024 -0.017 -0.034 0.022 1 -0.019 0.019 -0.0026 0.0062 0.0016 -0.014 -0.00041 -0.03 -0.037 -0.0068
n_cores -0.03 0.036 -0.0057 -0.025 -0.013 -0.03 -0.028 -0.0035 -0.019 1 -0.0012 0.0049 0.013 -0.015 0.024 -0.01 0.0044 0.007 0.0017
pc 0.031 -0.01 -0.0052 -0.017 0.64 -0.0056 -0.033 0.026 0.019 -0.0012 1 0.029 0.015 -0.0013 -0.0087 0.0054 0.034 -0.00071 -0.017
ram -0.00065 0.026 0.0034 0.041 0.015 0.0073 0.033 -0.0094 -0.0026 0.0049 0.029 1 0.011 0.016 -0.03 0.023 0.92 0.026 -0.0053
talk_time 0.053 0.014 -0.011 -0.039 -0.0068 -0.047 -0.0028 0.017 0.0062 0.013 0.015 0.011 1 -0.043 0.017 -0.03 0.022 -0.02 -0.01
three_g 0.012 -0.03 -0.046 -0.014 0.0018 0.58 -0.0094 -0.012 0.0016 -0.015 -0.0013 0.016 -0.043 1 0.014 0.0043 0.024 0.022 -0.028
touch_screen -0.011 0.01 0.02 -0.017 -0.015 0.017 -0.027 -0.0026 -0.014 0.024 -0.0087 -0.03 0.017 0.014 1 0.012 -0.03 -0.015 0.018
wifi -0.0083 -0.022 -0.024 0.023 0.02 -0.018 0.007 -0.028 -0.00041 -0.01 0.0054 0.023 -0.03 0.0043 0.012 1 0.019 0.027 0.043
price_range 0.2 0.021 -0.0066 0.017 0.022 0.015 0.044 0.00085 -0.03 0.0044 0.034 0.92 0.022 0.024 -0.03 0.019 1 0.034 0.18
sc_size -0.03 -0.01 -0.023 -0.01 -0.013 0.033 0.029 -0.023 -0.037 0.007 -0.00071 0.026 -0.02 0.022 -0.015 0.027 0.034 1 0.062
pixels 0.019 -0.016 -0.0091 -0.019 -0.012 -0.008 0.015 0.024 -0.0068 0.0017 -0.017 -0.0053 -0.01 -0.028 0.018 0.043 0.18 0.062 1
ttery_power
blue
lock_speed
dual_sim
fc
four_g
int_memory
m_dep
mobile_wt
n_cores
pc
ram
talk_time
three_g
uch_screen
wifi
price_range
sc_size
pixels
1.0
0.8
0.6
0.4
0.2
0.0

➢

Chapter 3

Technology Used

3.1. Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

3.2. Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

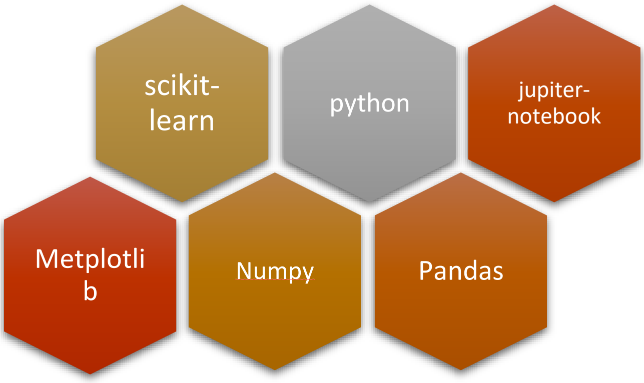


Fig 3. Technology used.

3.3. Pandas

Pandas is a Python library for data analysis. **Started by Wes McKinney in 2008** out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries.

3.4. NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. **NumPy was created in 2005 by Travis Oliphant**. It is an open-source project, and you can use it freely.

3.5. Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

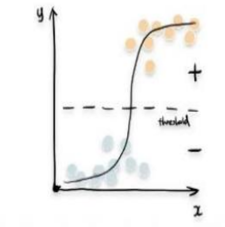
3.6. Scikit-learn

Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects. Scikit-learn is focused on machine learning tools including mathematical, statistical and general purpose algorithms that form the basis for many machine learning technologies. As a free tool, Scikit-learn is tremendously important in many different types of algorithm development for machine learning and related technologies.

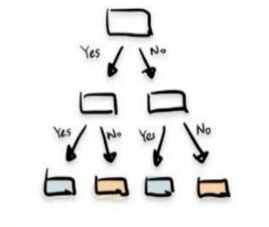
Chapter 4

Machine Learning algorithms:

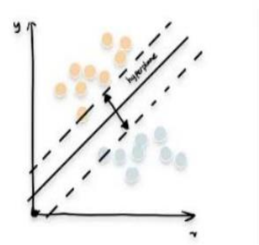
1.Logistic regression



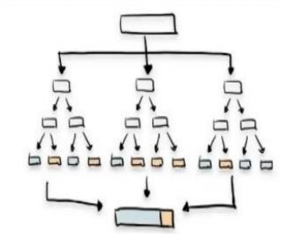
2. Decision Tree



3. Support Vector Machine



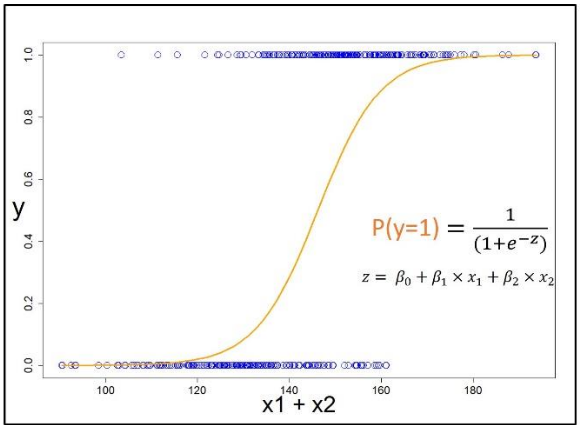
4. Random Forest

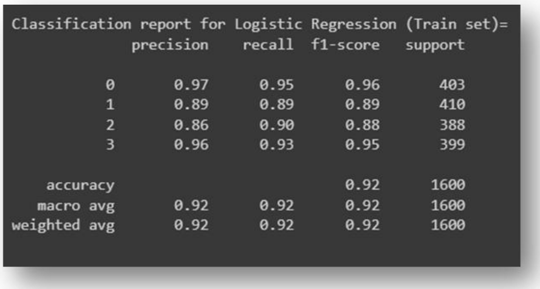


4.1 Logistic Regression classifier:

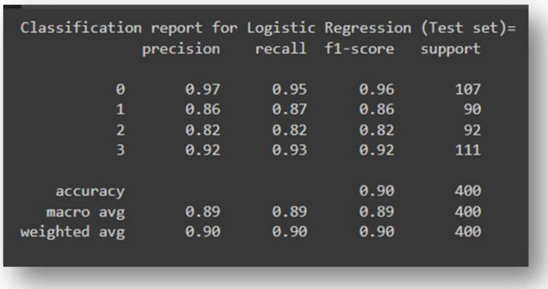
Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).





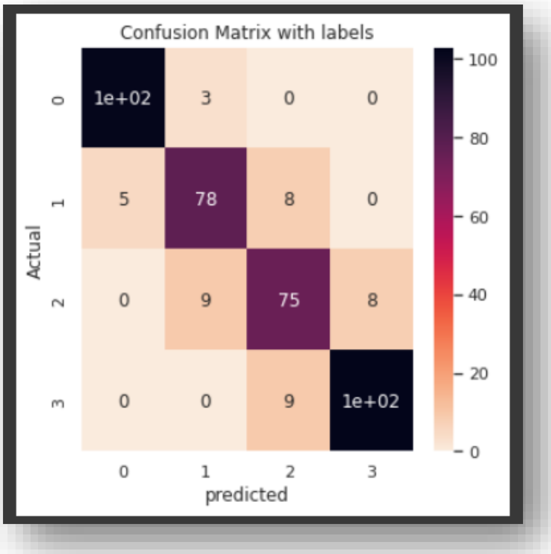
```
Classification report for Logistic Regression (Train set)=
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       403
           1       0.89      0.89      0.89       410
           2       0.86      0.90      0.88       388
           3       0.96      0.93      0.95       399

    accuracy                           0.92      1600
   macro avg       0.92      0.92      0.92      1600
weighted avg       0.92      0.92      0.92      1600
```



```
Classification report for Logistic Regression (Test set)=
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       107
           1       0.86      0.87      0.86        90
           2       0.82      0.82      0.82        92
           3       0.92      0.93      0.92       111

    accuracy                           0.90       400
   macro avg       0.89      0.89      0.89       400
weighted avg       0.90      0.90      0.90       400
```

TRAIN ACCURACY : 92%

TEST ACCURACY : 88%



4.2 Random Forest Classifier :

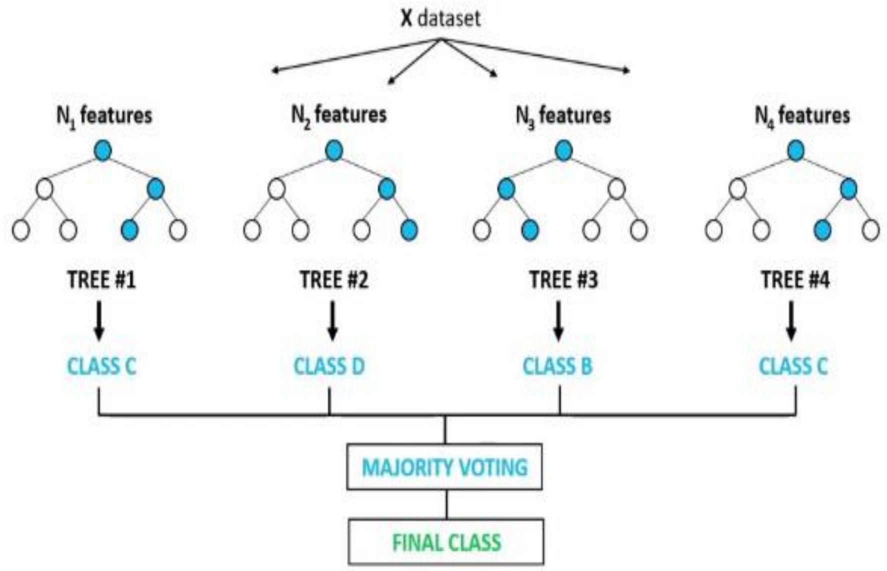
A random forest is a supervised machine learning method built from decision tree techniques. This algorithm is used to anticipate behavior and results in a variety of sectors, including banking and e-commerce.

A random forest is a machine learning approach for solving regression and classification issues. It makes use of ensemble learning, which is a technique that combines multiple classifiers to solve complicated problems.

A random forest method is made up of a large number of decision trees. The random forest algorithm's 'forest' is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm ensemble that increases the accuracy of machine learning algorithms.

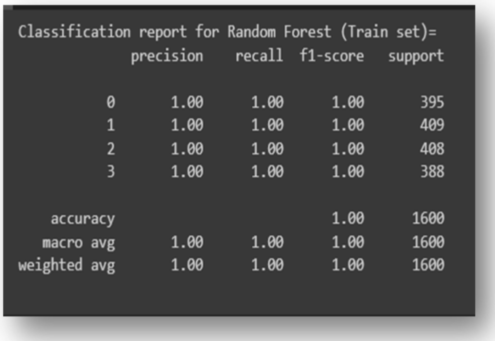
The outcome is determined by the (random forest) algorithm based on the predictions of the decision trees. It forecasts by averaging or averaging the output of several trees. The precision of the outcome improves as the number of trees grows.

Random Forest Classifier

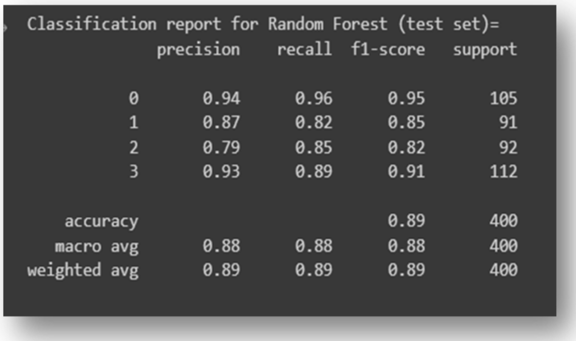


A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning.

We know that a forest comprises numerous trees, and the more trees more it will be robust.



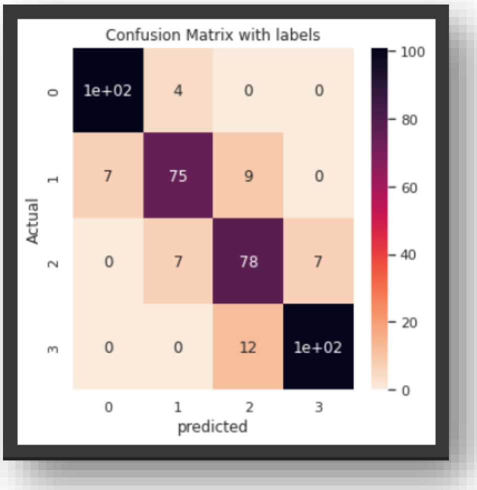
```
Classification report for Random Forest (Train set)=
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       395
           1       1.00      1.00      1.00       409
           2       1.00      1.00      1.00       408
           3       1.00      1.00      1.00       388

    accuracy                           1.00      1600
   macro avg       1.00      1.00      1.00      1600
weighted avg       1.00      1.00      1.00      1600
```



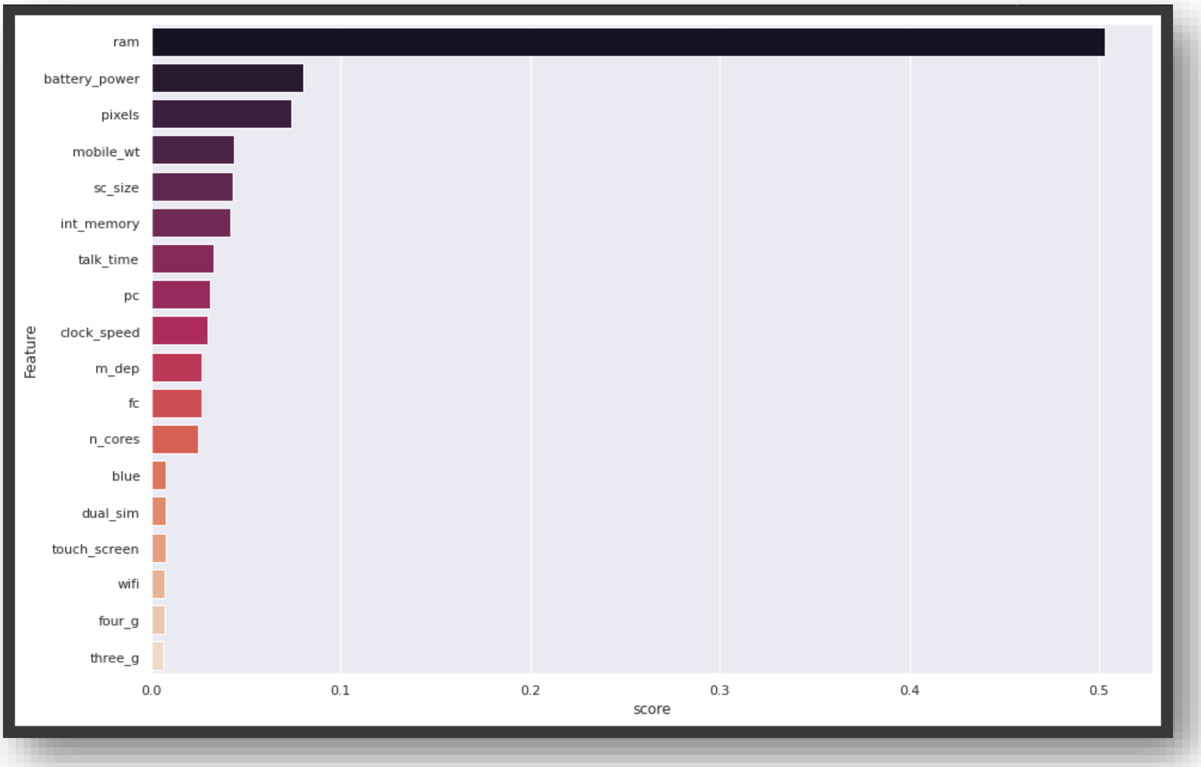
```
Classification report for Random Forest (test set)=
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       105
           1       0.87      0.82      0.85        91
           2       0.79      0.85      0.82        92
           3       0.93      0.89      0.91       112

    accuracy                           0.89       400
   macro avg       0.88      0.88      0.88       400
weighted avg       0.89      0.89      0.89       400
```

TRAIN ACCURACY : 100%

TEST ACCURACY : 88%



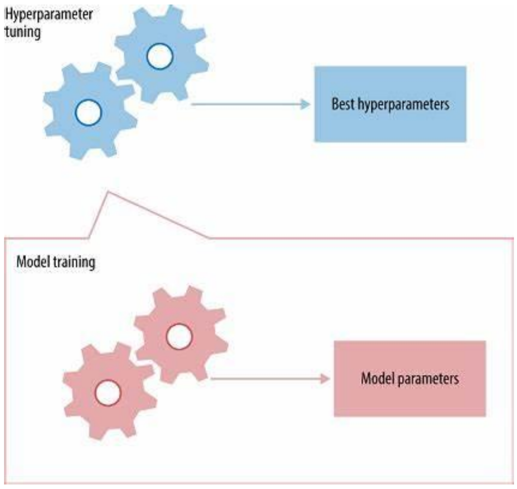
Feature Important Plots: Feature importance are provided by the fitted attribute feature_im portances_ and they are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree.

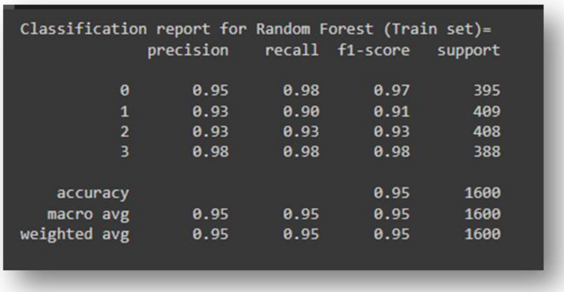


So, we can see that the random forest algorithm has good accuracy in prediction.

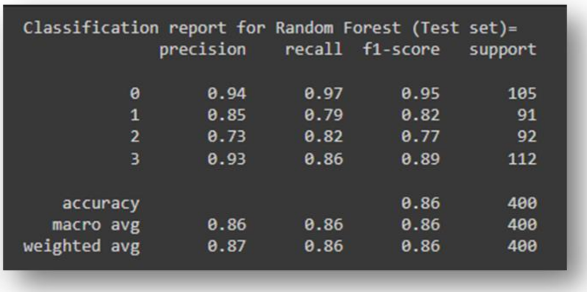
4.3 Hyperparameter tuning for Random Forest :

In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. (The parameters of a random forest are the variables and thresholds used to split each node learned during training)





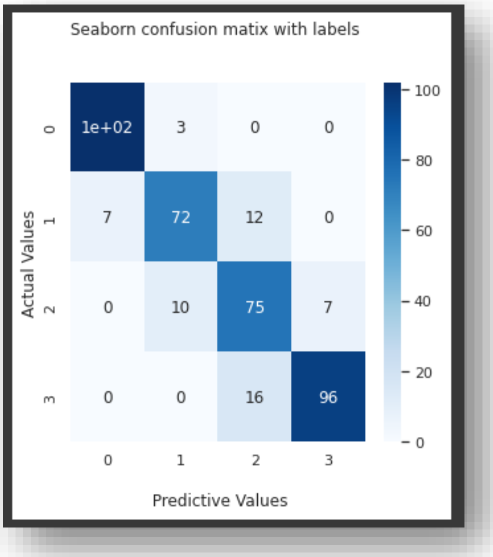
```
Classification report for Random Forest (Train set)=
              precision    recall  f1-score   support

           0       0.95      0.98      0.97       395
           1       0.93      0.90      0.91       409
           2       0.93      0.93      0.93       408
           3       0.98      0.98      0.98       388

    accuracy                           0.95      1600
   macro avg       0.95      0.95      0.95      1600
weighted avg       0.95      0.95      0.95      1600
```



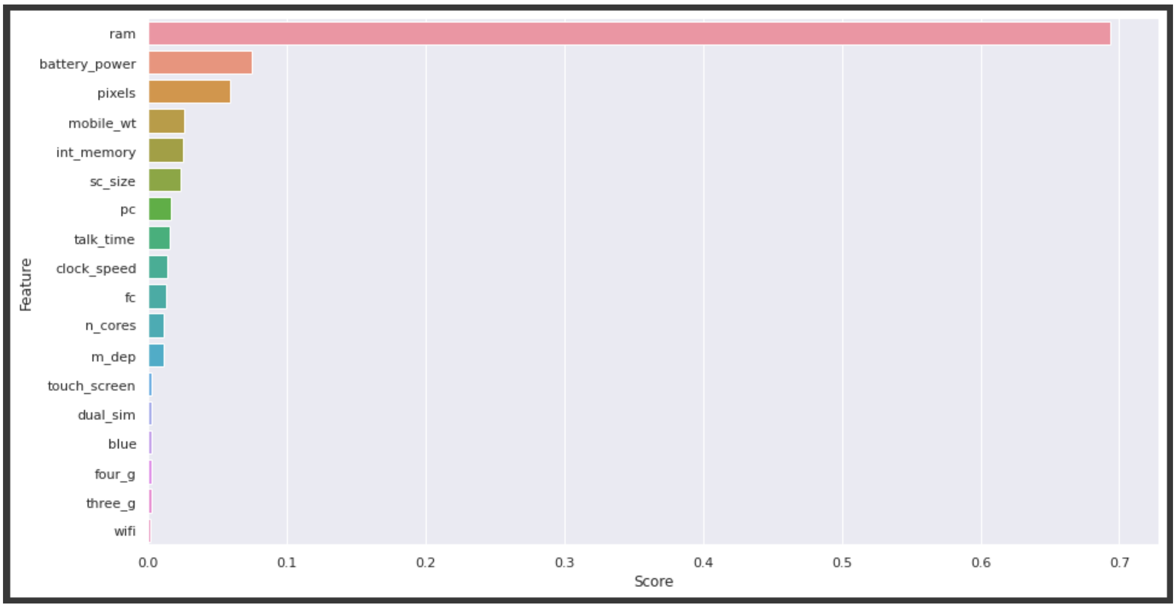
```
Classification report for Random Forest (Test set)=
              precision    recall  f1-score   support

           0       0.94      0.97      0.95       105
           1       0.85      0.79      0.82        91
           2       0.73      0.82      0.77        92
           3       0.93      0.86      0.89       112

    accuracy                           0.86       400
   macro avg       0.86      0.86      0.86       400
weighted avg       0.87      0.86      0.86       400
```

TRAIN ACCURACY : 95%

TEST ACCURACY : 87%



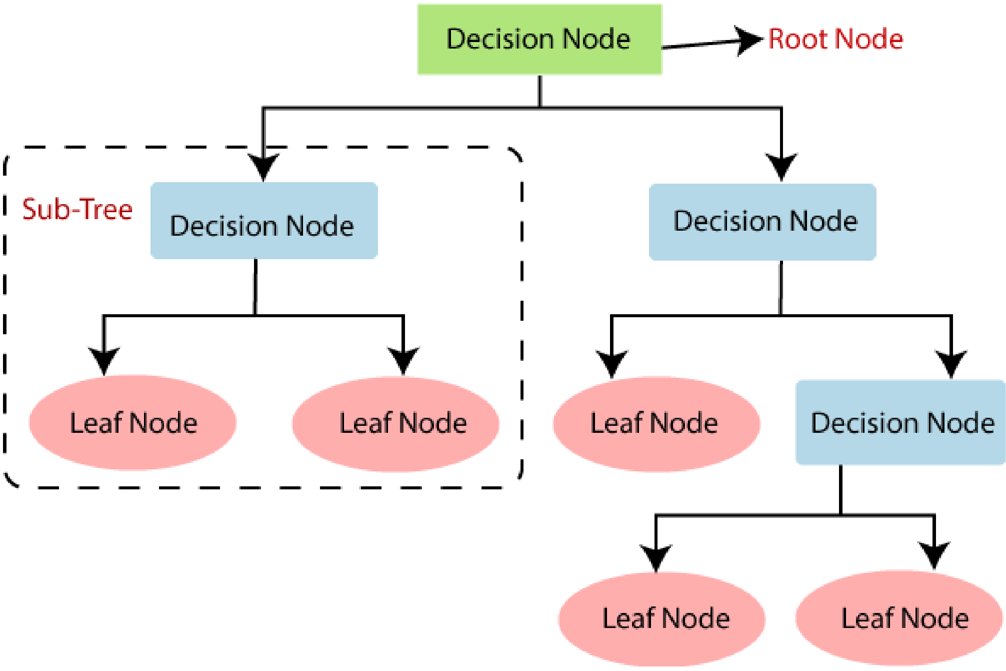
Feature importance for Hyperparameter tuning for Random Forest:

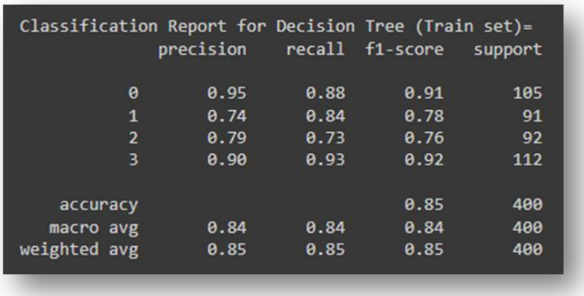


4.4 Decision Tree Classifier Algorithm:

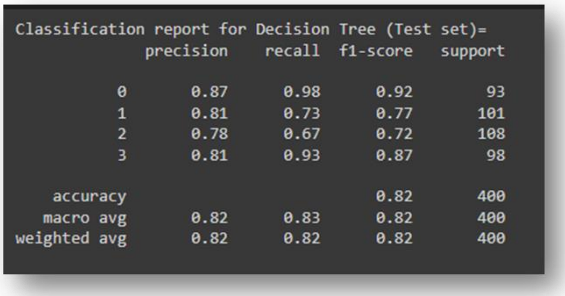
A Decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.





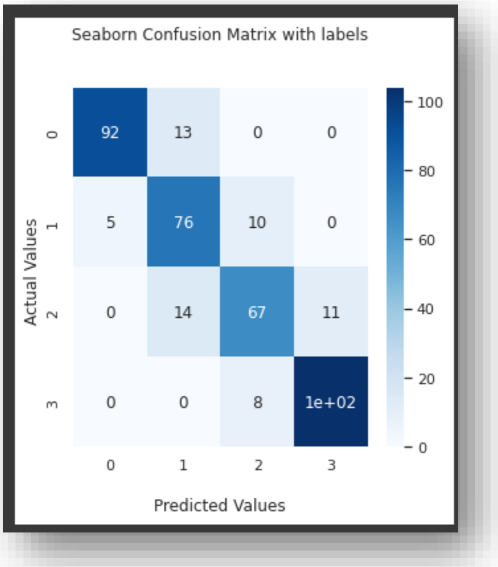
```
Classification Report for Decision Tree (Train set)=
              precision    recall  f1-score   support

           0       0.95      0.88      0.91       105
           1       0.74      0.84      0.78        91
           2       0.79      0.73      0.76        92
           3       0.90      0.93      0.92       112

    accuracy                           0.85       400
   macro avg       0.84      0.84      0.84       400
weighted avg       0.85      0.85      0.85       400
```



```
Classification report for Decision Tree (Test set)=
              precision    recall  f1-score   support

           0       0.87      0.98      0.92        93
           1       0.81      0.73      0.77       101
           2       0.78      0.67      0.72       108
           3       0.81      0.93      0.87        98

    accuracy                           0.82       400
   macro avg       0.82      0.83      0.82       400
weighted avg       0.82      0.82      0.82       400
```

TRAIN ACCURACY : 85%

TEST ACCURACY : 82%

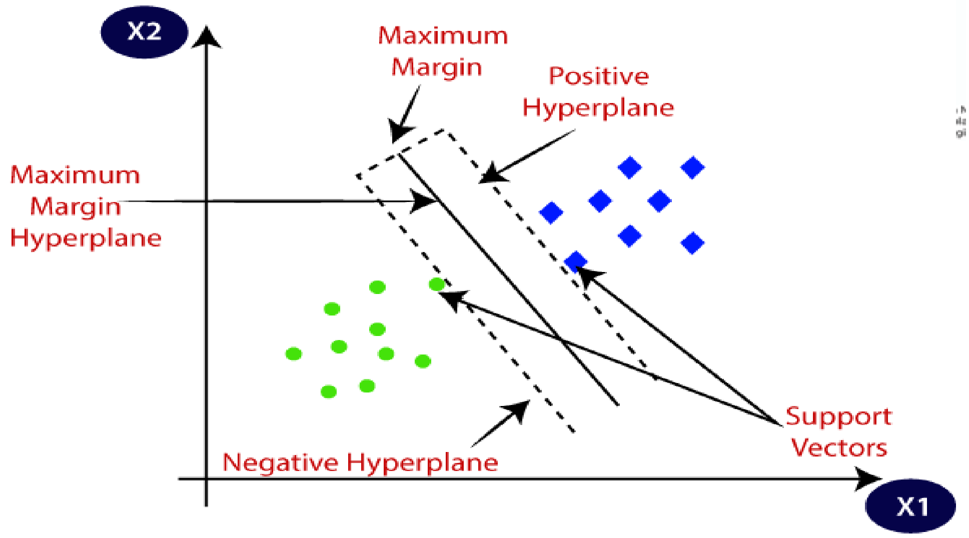


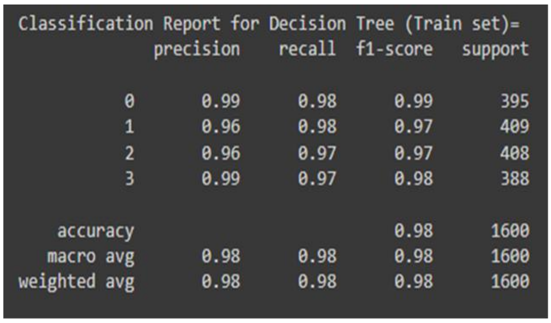
4.5 Support Vector Machine:

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

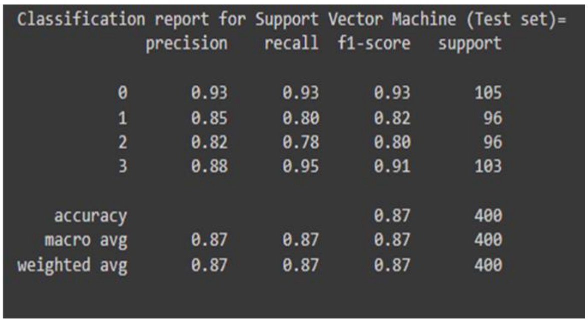
Support Vector Machine, or SVM, is a prominent Supervised Learning technique that is used for both classification and regression issues. However, it is mostly utilized in Machine Learning for Classification purposes.

The SVM algorithm's purpose is to find the optimum line or decision boundary for categorizing n-dimensional space so that we may simply place fresh data points in the proper category in the future. A hyperplane is the optimal choice boundary.





```
Classification Report for Decision Tree (Train set)=
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       395
           1       0.96      0.98      0.97       409
           2       0.96      0.97      0.97       408
           3       0.99      0.97      0.98       388

    accuracy                           0.98      1600
   macro avg       0.98      0.98      0.98      1600
weighted avg       0.98      0.98      0.98      1600
```



```
Classification report for Support Vector Machine (Test set)=
              precision    recall  f1-score   support

           0       0.93      0.93      0.93       105
           1       0.85      0.80      0.82        96
           2       0.82      0.78      0.80        96
           3       0.88      0.95      0.91       103

    accuracy                           0.87       400
   macro avg       0.87      0.87      0.87       400
weighted avg       0.87      0.87      0.87       400
```

TRAIN ACCURACY : 98%

TEST ACCURACY : 87%

We can see that the SVM classifier is giving the best accuracy.

Chapter 5

5.1 Challenges Faced:

- Reading the dataset and understanding the meaning of some columns.
- For answering some of the questions we had to understand
- Mobile Price Range System model that how they work.
- Designing multiple visualizations to summarize the information in the dataset and successfully communicate the results and trends to the reader.

5.2 Summary:

- From EDA we can see that there are mobile phones in 4 price ranges. The number of elements is almost similar.
- half the devices have Bluetooth, and half don’t.
- There is a gradual increase in battery as the price range increases Ram has continuous increase with price range while moving from Low cost to Very high cost.
- costly phones are lighter.
- RAM, battery power, pixels played more significant role in deciding the price range of mobile phones.
- form all the above experiments we can conclude that logistic regression, SVM and Hyperparameter tuning for Random Forest we got the best results

5.4 Conclusion:

- From EDA we can see that here are mobile phones in 4 price ranges. The number of elements is almost similar.
- half the devices have Bluetooth, and half don’t
- There is a gradual increase in battery as the price range increases Ram has continuous increase with price range while moving from Low cost to Very high cost.
- costly phones are lighter

- RAM, battery power, pixels played more significant role in deciding the price range of mobile phone.
- form all the above experiments we can conclude that logistic regression , SVM and Hyperparameter tuning for Random Forest we got the best Results.
- This project model could be improved by developing software that could predict by selecting features so that it could be used while launching the new product.