Capstone Project

Mobile Price Range Prediction

By

Dinesh A Wagh.

Points To Discuss

- Problem Statement
- Introduction
- Abstract
- Data Description
- Data Pre-processing
- Exploratory Data Analysis
- Distribution of all Independent variables
- Looking for Outliers
- Correlation Analysis
- Machine learning algorithms

1.Logistic regression

2. Random forest classifier

3. Decision tree

4. SVM

- Conclusion

● Classification in supervised learning :

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.

● Difference of classification and regression :

The most significant difference between regression vs classification is that while regression helps predict a continuous quantity, classification predicts discrete class labels.

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(e.g.:- 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.

Introduction

- Price is the most important component in the marketing of any product and is often the definitive factor in its sale to a consumer. In a constantly evolving and volatile market, the price is often the factor that makes or breaks a product. Setting an optimal price before the release of a product is imperative for any company. A tool that gives the estimated price of a product after weighing in the features it provides can come in handy and can help the company in making an informed decision while setting the market price for a product. Such a tool can also be used by a consumer to get an estimated price based on the features they are looking for in the product.

- Nowadays, a cellphone is an essential accessory of a person. It is the fastest evolving and moving product in the technology market space. New mobiles with updated versions and new features are introduced into the market at a rapid pace. Thousands of mobiles are sold each day. In such a fast-paced and volatile market, a mobile company needs to set optimal prices to compete with its rivals. The first step in fixing a price is to estimate the price based on the features. The objective this research is to develop an ML model capable of estimating the price of a mobile phone based on its features. A potential buyer can also make use of the model to estimate the price of a mobile by inputting just the features they require into the tool. The same approach to create a prediction model can be used to develop a price estimation model for most products that have similar independent variable parameters. The price of a mobile is dependent on many features for example, the processor, battery capacity, camera quality, display size and thickness, etc. These features can be used to classify phones into various categories like entry-level, mid-range, flagship, premium, etc. Supervised ML algorithms are used in this paper as the dataset used has a definitive class label for price range.

Abstract

- Machine learning based classification techniques helps to solve the problem related to decision making. In many areas of price prediction are used like housing price prediction, stock price prediction different classification algorithm used. Some of them are used artificial neural network. In this study, four different classification techniques used for prediction the mobile price range. The first one is Logistic Regression, second one is Random Forest, third one is Decision Tree and fourth one is support vector Machines. The accuracy got by Random Forest and Decision trees are respectively 87.75% and 82.25%. As the accuracies these two techniques is lower compared to other two techniques so Random Forest and Decision Tree are not best suited for prediction. With logistic regression and Support Vector Machines we get accuracy up to 94.25% and 96% respectively after using hyperparameter tuning on both techniques.

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 wifi 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).

Data Pre-Processing

- Data preprocessing is a process of preparing the raw data and making it suitable for our analysis purpose, where we have to do a lot of Data Cleaning and handle the missing values by using appropriate imputation techniques and based on that variable nature i.e., either of categorical & numerical variable. Substitution/imputation of missing values using either mean, median, mode or zero according to the nature of those variables. Here, in this project, we have imputed zero values of px_height and sc_w column with mean values.
- 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). The pandas.unique() function returns the unique values present in a dataset.
- We will count total number of NaN data present in Mobile Price Range dataset and find out the number of NaN or missing values in each columns.

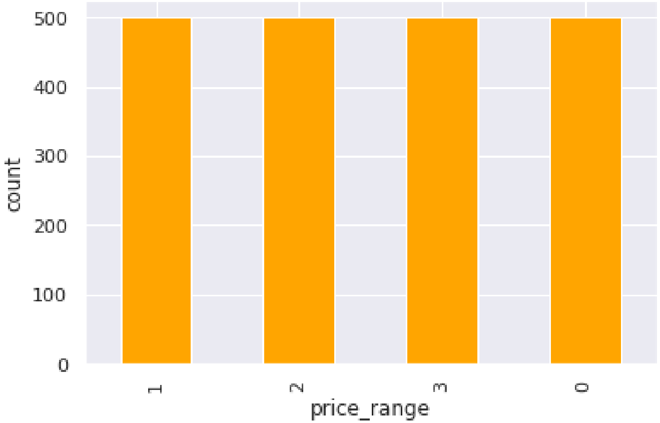
Data Preparation :

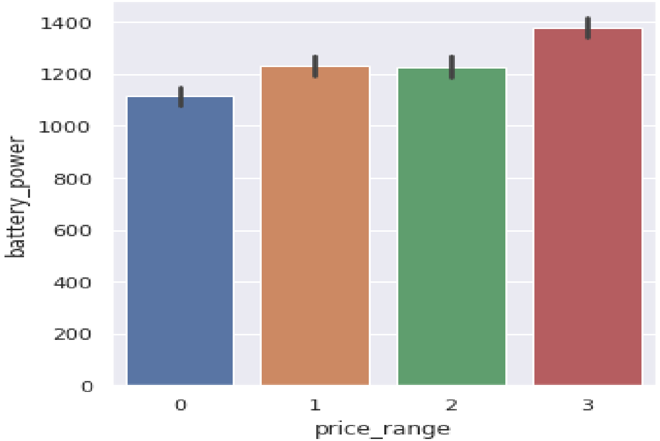
Data Preparation includes loading the dataset into a data frame, exploring the number of rows & columns, ranges of values, data types, descriptive summary of numerical features, correlation and distribution of features etc.

Data Cleaning :

Checking and cleaning if there are any null data. In our dataset we didn’t find any null values. There were zero values in pixel height and screen width columns. Values of this columns cannot be zero so this value has been replaced with mean values. Then check each attribute using histogram. Data cleaning or preprocessing is one of the important parts of research. Handling missing data is one of them. But our dataset doesn’t have any null values. Labeling the categorical data is also an important part of data preprocessing.

Exploratory data analysis :





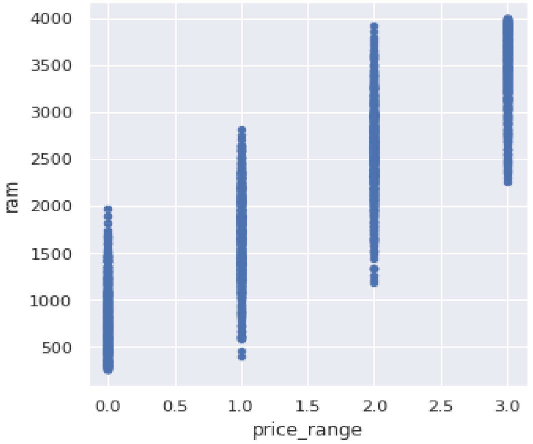
1. Price Range Count:

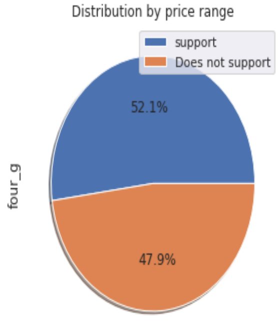
The class label is the price range. It has 4 kinds of values 0,1,2 and 3 which are of ordinal data type representing the increasing degree of price. Higher the value, higher is the price range the mobile falls under. These 4 values can be interpreted as economical, mid-range, flagship and premium. Also, distribution of price range is uniform for all four classes. Thus, we know that there's no class imbalance problem here.

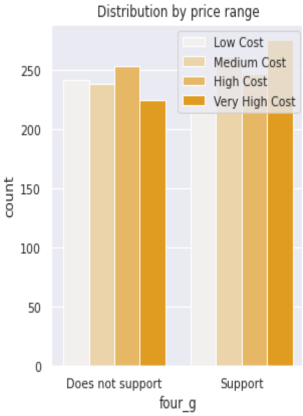
2. Price range vs Battery power:

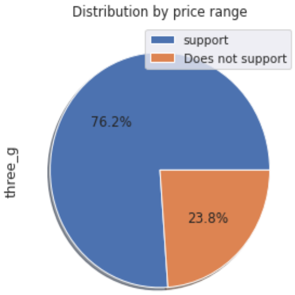
Batter power is of the most important feature in mobile selection. As we can see form bar graph below, with increase in price range batter power also increases. So, for premium mobiles batter power is higher.

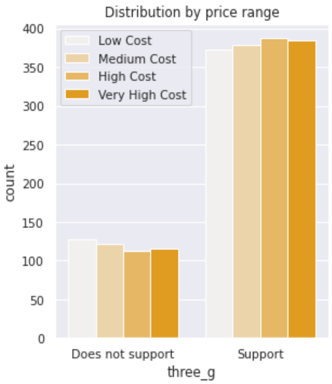
Exploratory data analysis :











3. Price Range vs RAM:

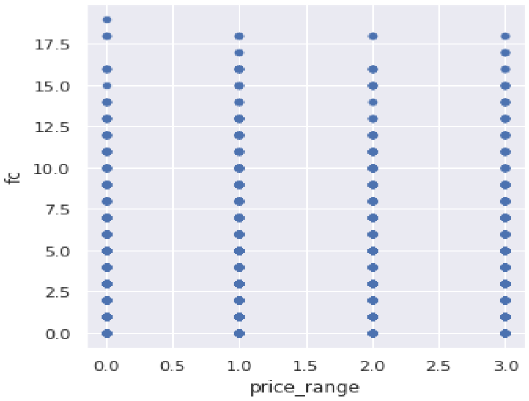
we can see from graph for higher price ranges ram is higher and for lower price ranges ram is low.

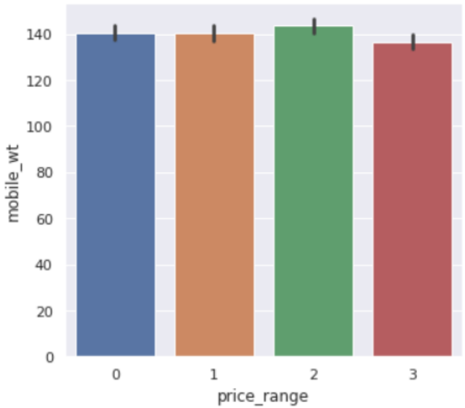
For premium and flagship mobiles ram is nearly same.

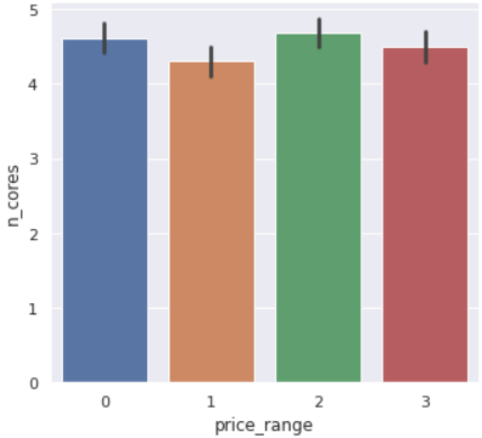
4. Price Range vs Three G and Four G:

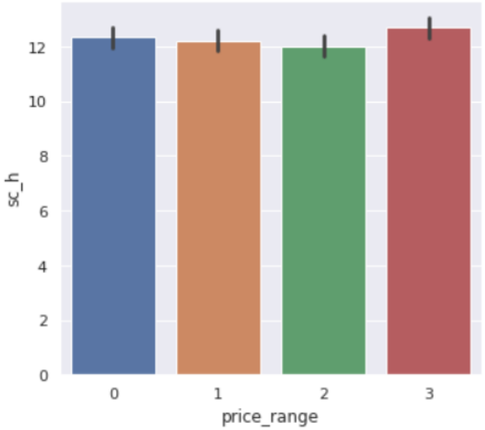
The distribution of the categorical features is almost similar to each other except the feature ‘three_g’ which contains very few mobile phones which do not have 3G access. We can infer that almost all phones have 3G access if not 4G.

Exploratory data analysis :





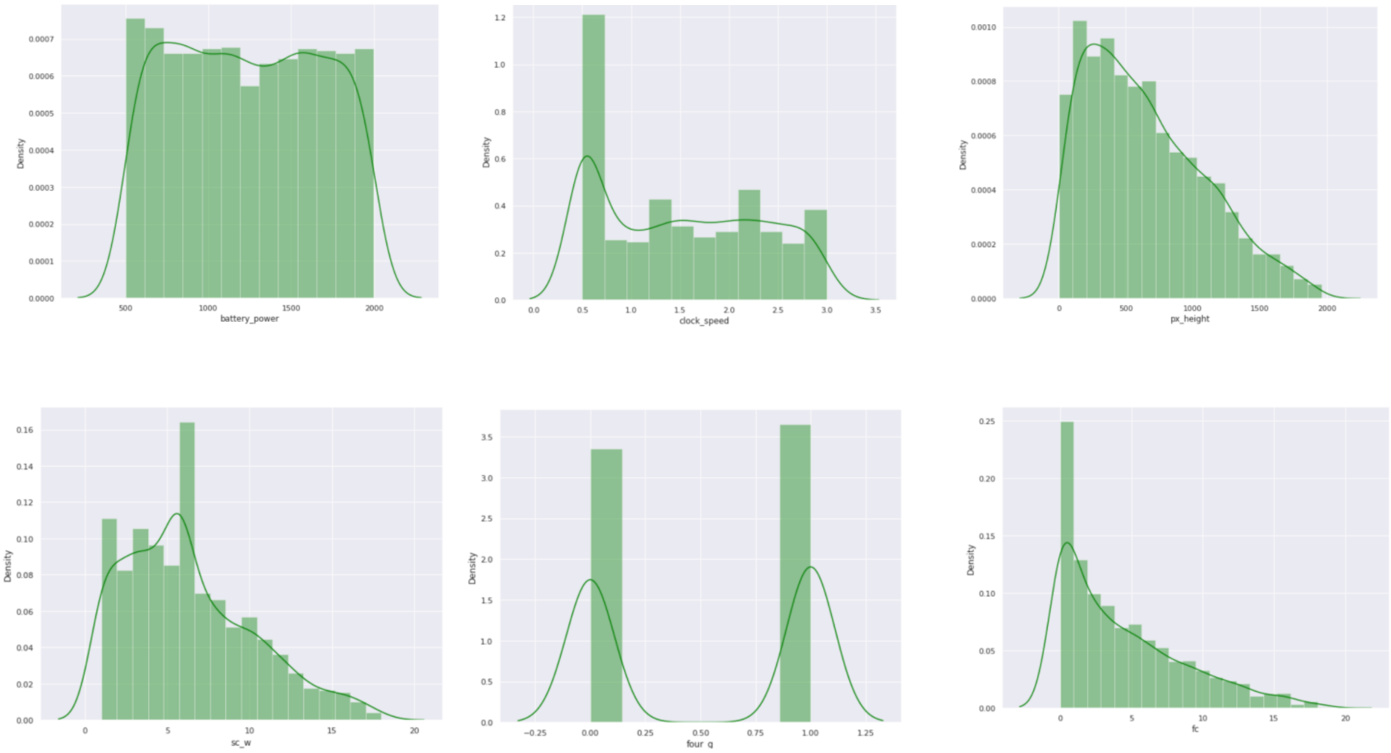




5.Price Range vs other features:

The distribution of the other features across different price ranges shows us that only the features RAM, battery power, px_height and px_width increase with an increase in price. These features are the most influential in determining the price ranges

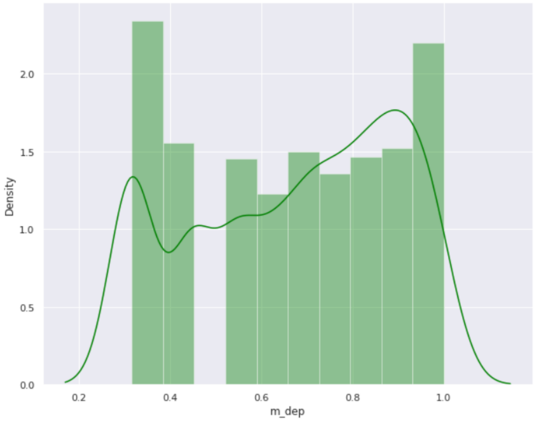
Distribution of all the independent variables :

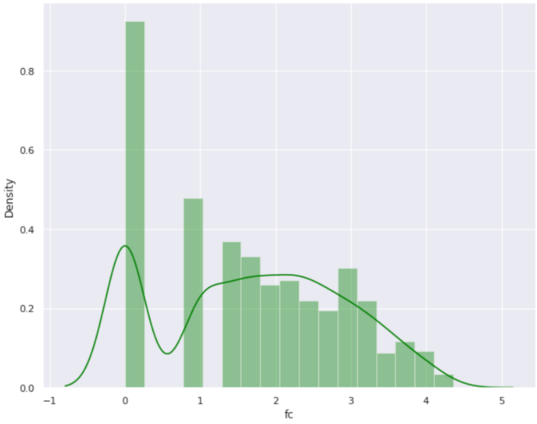


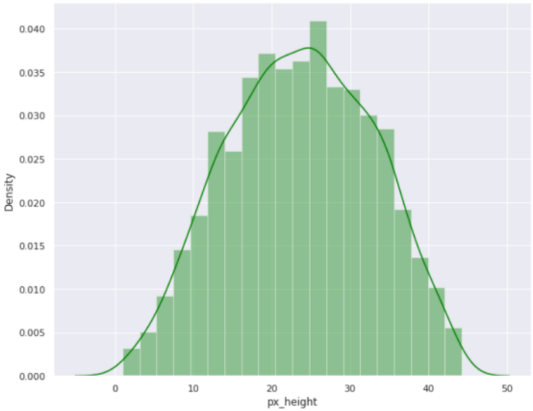
6. Checking the distribution of Independent variables :

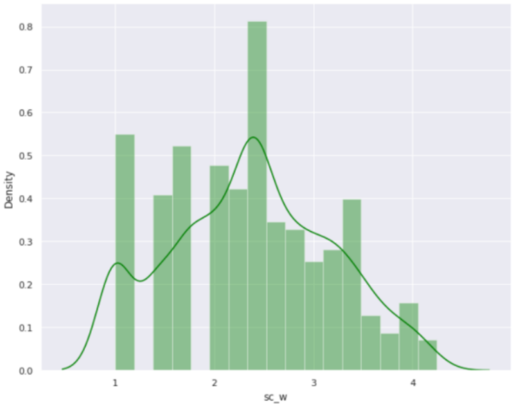
Most of the numerical features follow a uniform distribution except a few features like fc, px_height, and sc_w follows a right skewed distribution.

Transformation of independent variables :

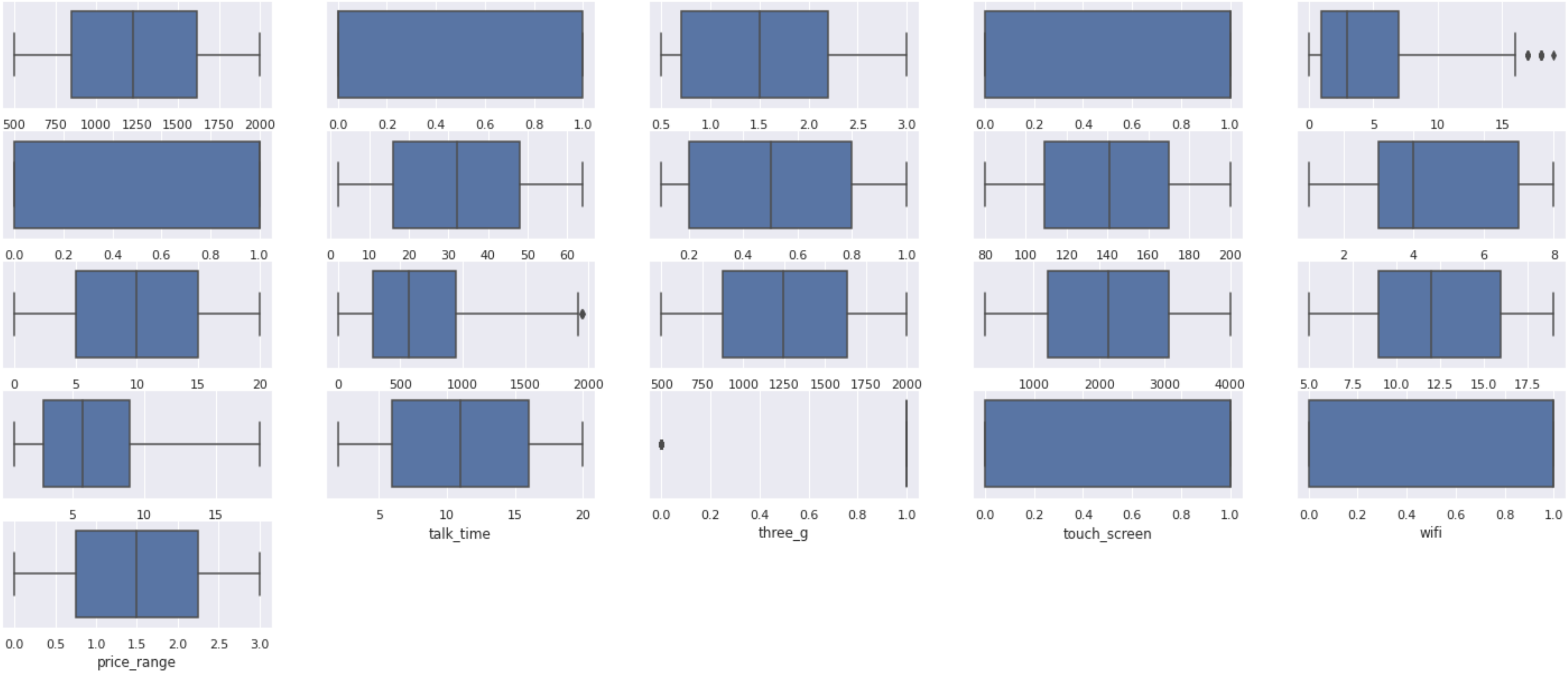






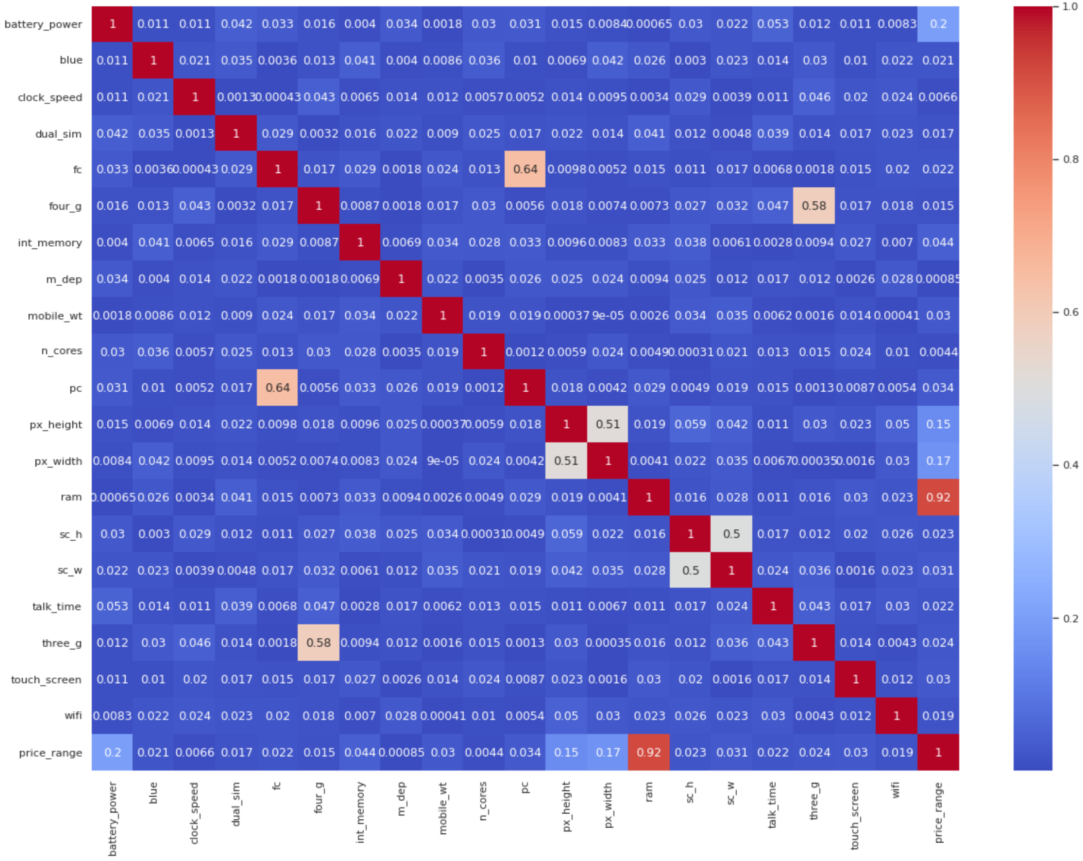


Removing Outliers from the dataset:



After visualizing these box plots, we can clearly see that there are not too much outliers in independent features. So, no need to modify any columns in the dataset.

Correlation analysis:



	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
battery_power	1	0.011	0.011	0.042	0.033	0.016	0.004	0.034	0.0018	0.03	0.031	0.015	0.0084	0.00065	0.03	0.022	0.053	0.012	0.011	0.0083	0.2
blue	0.011	1	0.021	0.035	0.0036	0.013	0.041	0.004	0.0086	0.036	0.01	0.0069	0.042	0.026	0.003	0.023	0.014	0.03	0.01	0.022	0.021
clock_speed	0.011	0.021	1	0.0013	0.00043	0.043	0.0065	0.014	0.012	0.0057	0.0052	0.014	0.0095	0.0034	0.029	0.0039	0.011	0.046	0.02	0.024	0.0066
dual_sim	0.042	0.035	0.0013	1	0.029	0.0032	0.016	0.022	0.009	0.025	0.017	0.022	0.014	0.041	0.012	0.0048	0.039	0.014	0.017	0.023	0.017
fc	0.033	0.0036	0.00043	0.029	1	0.017	0.029	0.0018	0.024	0.013	0.64	0.0098	0.0052	0.015	0.011	0.017	0.0068	0.0018	0.015	0.02	0.022
four_g	0.016	0.013	0.043	0.0032	0.017	1	0.0087	0.0018	0.017	0.03	0.0056	0.018	0.0074	0.0073	0.027	0.032	0.047	0.58	0.017	0.018	0.015
int_memory	0.004	0.041	0.0065	0.016	0.029	0.0087	1	0.0069	0.034	0.028	0.033	0.0096	0.0083	0.033	0.038	0.0061	0.0028	0.0094	0.027	0.007	0.044
m_dep	0.034	0.004	0.014	0.022	0.0018	0.0018	0.0069	1	0.022	0.0035	0.026	0.025	0.024	0.0094	0.025	0.012	0.017	0.012	0.0026	0.028	0.00085
mobile_wt	0.0018	0.0086	0.012	0.009	0.024	0.017	0.034	0.022	1	0.019	0.019	0.00037	9e-05	0.0026	0.034	0.035	0.0062	0.0016	0.014	0.00041	0.03
n_cores	0.03	0.036	0.0057	0.025	0.013	0.03	0.028	0.0035	0.019	1	0.0012	0.0059	0.024	0.0049	0.00031	0.021	0.013	0.015	0.024	0.01	0.0044
pc	0.031	0.01	0.0052	0.017	0.64	0.0056	0.033	0.026	0.019	0.0012	1	0.018	0.0042	0.029	0.0049	0.019	0.015	0.0013	0.0087	0.0054	0.034
px_height	0.015	0.0069	0.014	0.022	0.0098	0.018	0.0096	0.025	0.00037	0.0059	0.018	1	0.51	0.019	0.059	0.042	0.011	0.03	0.023	0.05	0.15
px_width	0.0084	0.042	0.0095	0.014	0.0052	0.0074	0.0083	0.024	9e-05	0.024	0.0042	0.51	1	0.0041	0.022	0.035	0.0067	0.00035	0.0016	0.03	0.17
ram	0.00065	0.026	0.0034	0.041	0.015	0.0073	0.033	0.0094	0.0026	0.0049	0.029	0.019	0.0041	1	0.016	0.028	0.011	0.016	0.03	0.023	0.92
sc_h	0.03	0.003	0.029	0.012	0.011	0.027	0.038	0.025	0.034	0.00031	0.0049	0.059	0.022	0.016	1	0.5	0.017	0.012	0.02	0.026	0.023
sc_w	0.022	0.023	0.0039	0.0048	0.017	0.032	0.0061	0.012	0.035	0.021	0.019	0.042	0.035	0.028	0.5	1	0.024	0.036	0.0016	0.023	0.031
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	0.0067	0.011	0.017	0.024	1	0.043	0.017	0.03	0.022
three_g	0.012	0.03	0.046	0.014	0.0018	0.58	0.0094	0.012	0.0016	0.015	0.0013	0.03	0.00035	0.016	0.012	0.036	0.043	1	0.014	0.0043	0.024
touch_screen	0.011	0.01	0.02	0.017	0.015	0.017	0.027	0.0026	0.014	0.024	0.0087	0.023	0.0016	0.03	0.02	0.0016	0.017	0.014	1	0.012	0.03
wifi	0.0083	0.022	0.024	0.023	0.02	0.018	0.007	0.028	0.00041	0.01	0.0054	0.05	0.03	0.023	0.026	0.023	0.03	0.0043	0.012	1	0.019
price_range	0.2	0.021	0.0066	0.017	0.022	0.015	0.044	0.00085	0.03	0.0044	0.034	0.15	0.17	0.92	0.023	0.031	0.022	0.024	0.03	0.019	1

This is a measure of the mutual relationship between attributes. By this measure the impact of an attribute depends on another attribute. Suppose, when it is summer people use to buy ice-cream (Correlation for Data Science | Towards Data Science, n.d.). Correlation gives a better understanding to a dataset (Correlation Towards Data Science, n.d.). From above correlation plot we can see that, Ram and price_range shows high correlation which is 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 phone is good, the back camera would also be good.

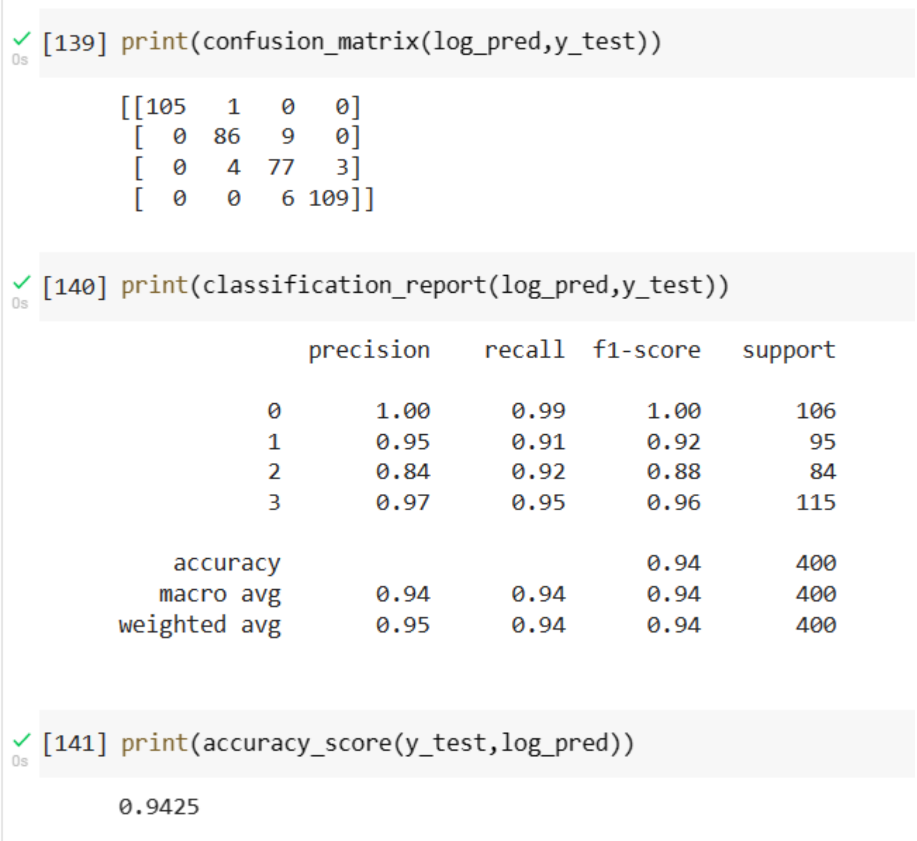
Model Building & Predictions :

1. Logistic Regression:

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome, something that can take two values such as true/false, yes/no and so on. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as 1. Just like linear regression assumes that the data follows a linear function, logistic regression models the data using the sigmoid function.

The results obtained by implementing logistic regression to predict mobile price range are:

1. Logistic Regression:



```
[139] print(confusion_matrix(log_pred,y_test))

      [[105   1   0   0]
       [  0  86   9   0]
       [  0   4  77   3]
       [  0   0   6 109]]
```

```
[140] print(classification_report(log_pred,y_test))

                    precision    recall  f1-score   support

                 0       1.00      0.99      1.00       106
                 1       0.95      0.91      0.92        95
                 2       0.84      0.92      0.88        84
                 3       0.97      0.95      0.96       115

          accuracy                           0.94       400
         macro avg       0.94      0.94      0.94       400
      weighted avg       0.95      0.94      0.94       400
```

```
[141] print(accuracy_score(y_test,log_pred))

      0.9425
```

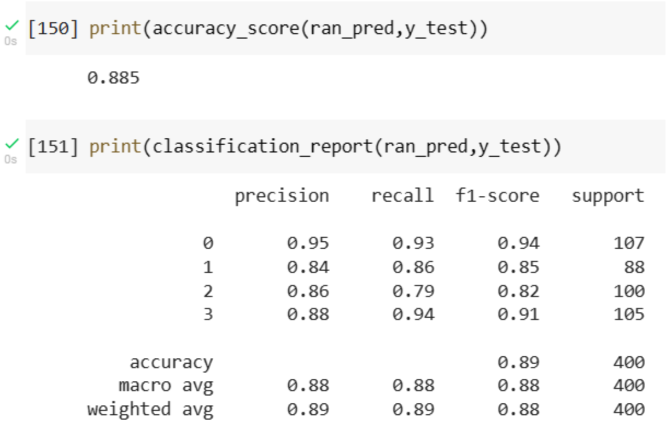
Logistic regression was found to be able to correctly forecast the classes with a accuracy of 94.25 %.

The accuracy of this model is good and we can use this model for prediction of mobile price range.

2. Random Forest Regressor :

Random forest is an ensemble of decision trees. Decision trees are great for obtaining non-linear relationships between input features and the target variable.

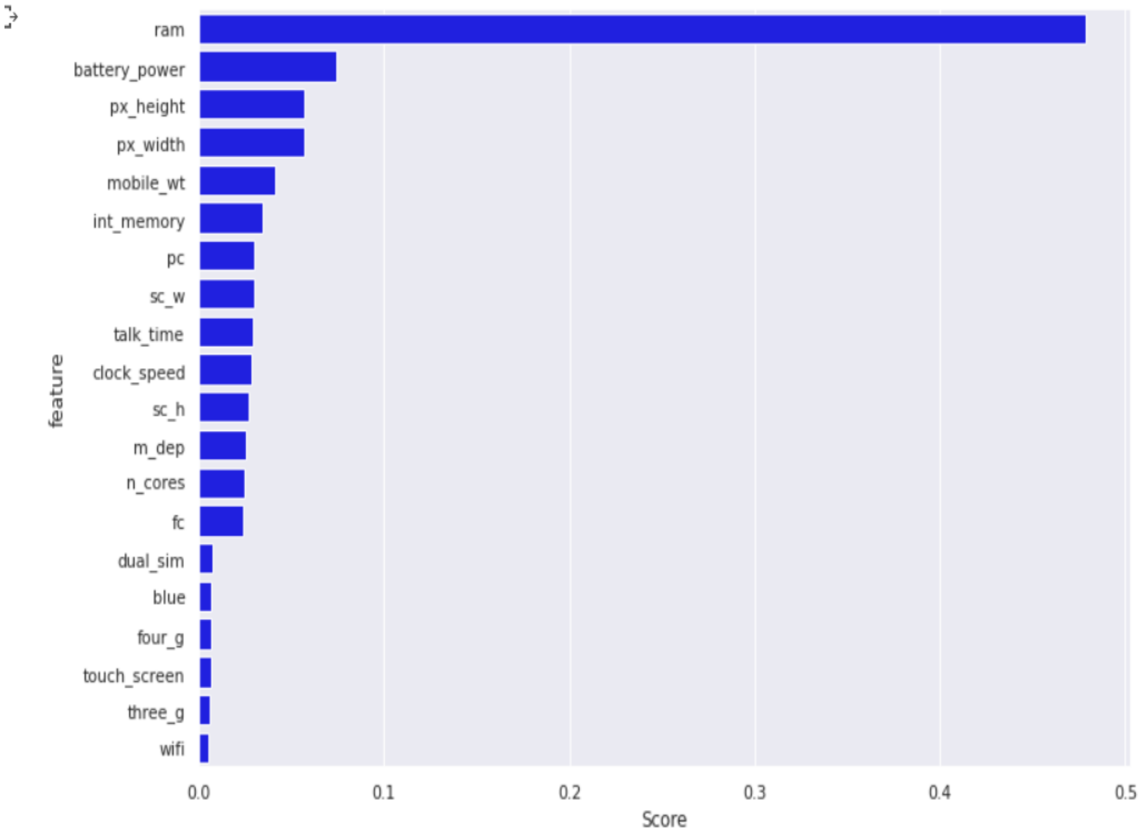
It starts at the very top with one node which then splits into a left and right node known as decision nodes. These nodes then split into their respective left and right nodes. At the end of the leaf node, the average of the observation that occurs within that area is computed. The most bottom nodes are referred to as leaves or terminal nodes. The results obtained by implementing Random Forest Regressor are:



```
[150] print(accuracy_score(ran_pred,y_test))

      0.885

[151] print(classification_report(ran_pred,y_test))

                    precision    recall  f1-score   support

                 0       0.95      0.93      0.94       107
                 1       0.84      0.86      0.85        88
                 2       0.86      0.79      0.82       100
                 3       0.88      0.94      0.91       105

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

The efficiency of the model trained using the Random Forest algorithm was found to be 88.5%.

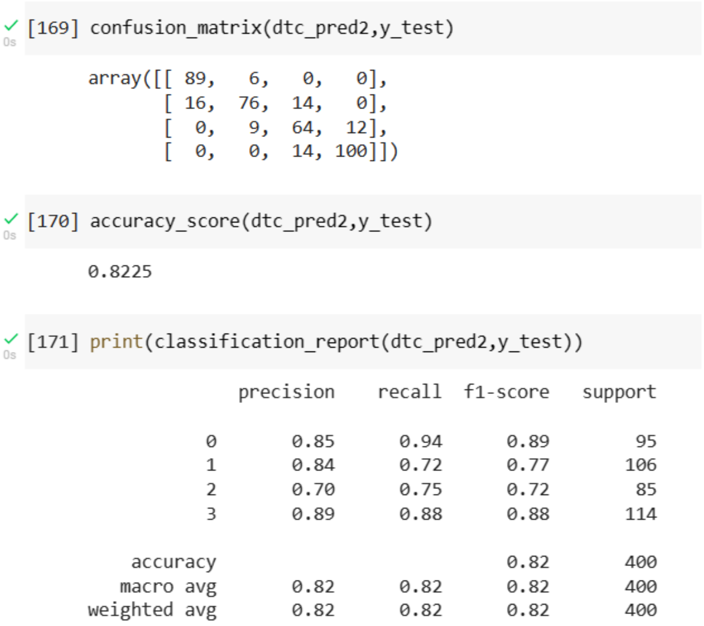
2. Random Forest Regressor :



From the graph, we can identify the importance of the feature to predict the price range of mobile.

3. Decision Tree :

Third ML Classification model we used for prediction is Decision tree. Decision Tree was found to be able to correctly forecast the classes with a certainty of 83.25%. The reason for the average level of accuracy obtained is that the Decision Tree is not suited for handling numeric data.



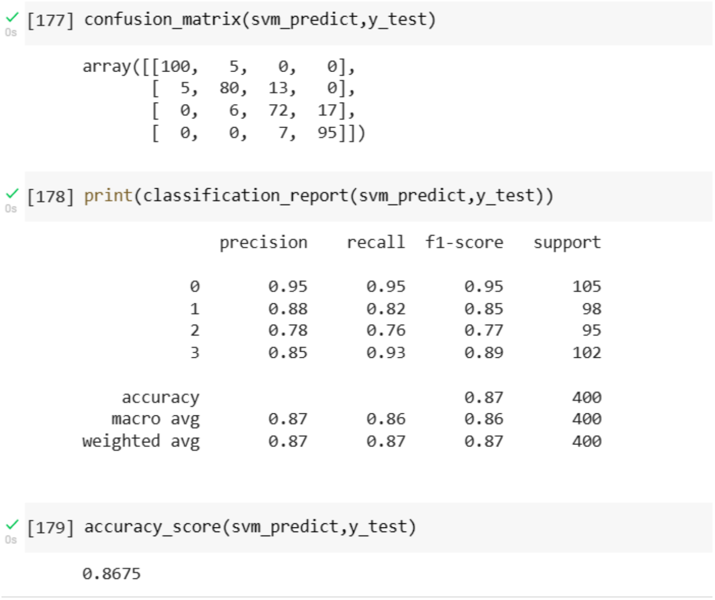
```
[169] confusion_matrix(dtc_pred2,y_test)

      array([[ 89,   6,   0,   0],
             [ 16,  76,  14,   0],
             [  0,   9,  64,  12],
             [  0,   0,  14, 100]])

[170] accuracy_score(dtc_pred2,y_test)

      0.8225

[171] print(classification_report(dtc_pred2,y_test))

                    precision    recall  f1-score   support

                 0       0.85      0.94      0.89        95
                 1       0.84      0.72      0.77       106
                 2       0.70      0.75      0.72        85
                 3       0.89      0.88      0.88       114

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

After using hyper parameter tuning, we got accuracy of 82.25 %. So, this model is not suitable for prediction of mobile price range.

4. Support Vector Machine :

Support vector machines are a set of supervised learning methods used for classification, regression and outlier detection. SVMs are different from the other classification algorithms because of the way they choose the decision boundary that maximizes the distance from the nearest data points of all the classes. The decision boundary created by SVMs is called the maximum margin classifier or the maximum margin hyperplane.

The results obtained by implementing Support Vector Machine are:



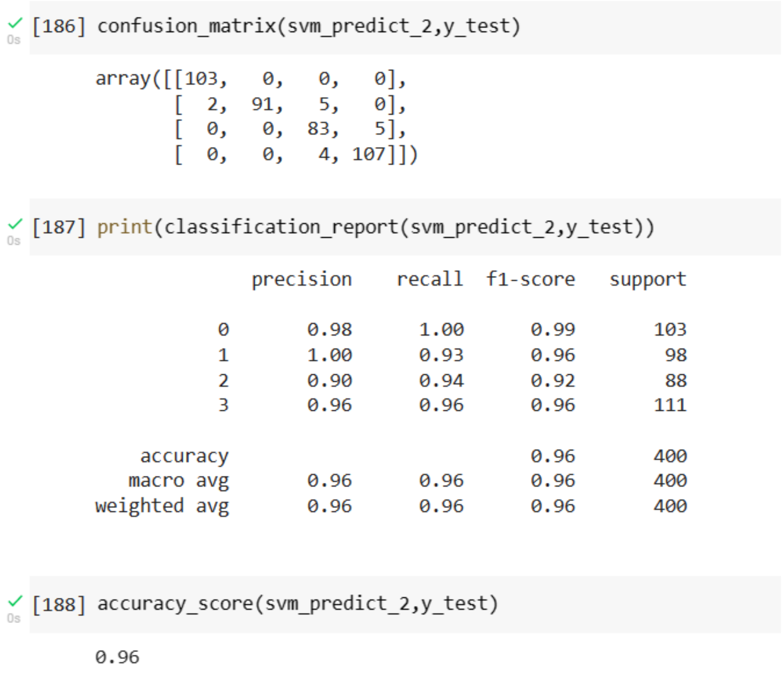
```
[177] confusion_matrix(svm_predict,y_test)

      array([[100,   5,   0,   0],
             [  5,  80,  13,   0],
             [  0,   6,  72,  17],
             [  0,   0,   7,  95]])

[178] print(classification_report(svm_predict,y_test))

                    precision    recall  f1-score   support

                 0       0.95      0.95      0.95       105
                 1       0.88      0.82      0.85        98
                 2       0.78      0.76      0.77        95
                 3       0.85      0.93      0.89       102

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

[179] accuracy_score(svm_predict,y_test)

      0.8675
```

The efficiency of the model trained using the Support vector machine algorithm was found to be 86.75%.

Hyperparameter Tunning on Support Vector Machine :

The efficiency of the model trained using the Support vector machine algorithm was found to be 86.75%.

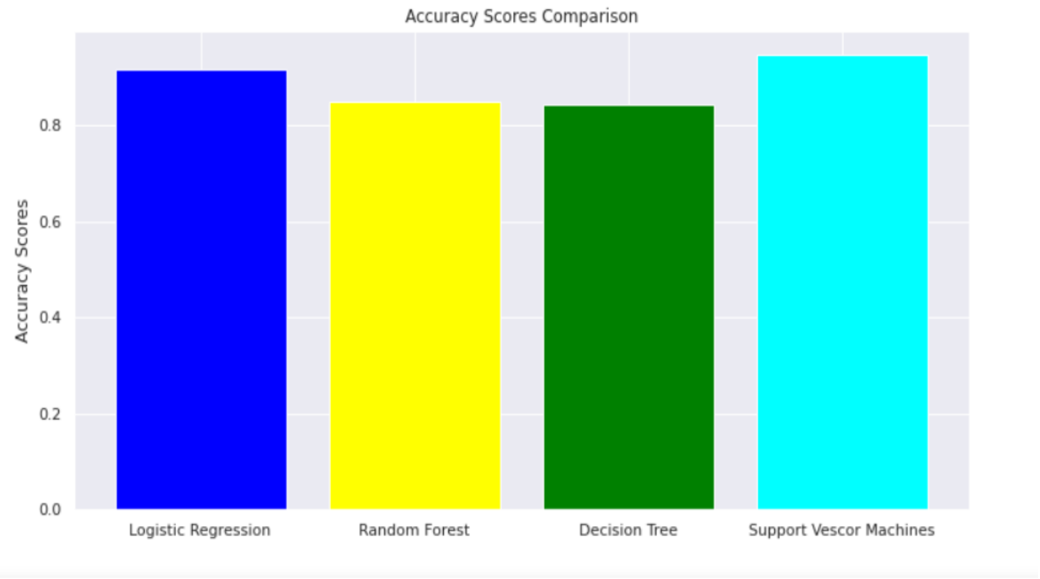
So, we perform hyper parameter tuning on SVM. The accuracy achieved after tuning the parameter is 96% which is highest among all models used.



```
✓ [186] confusion_matrix(svm_predict_2,y_test)
0s
        array([[103,   0,   0,   0],
               [  2,  91,   5,   0],
               [  0,   0,  83,   5],
               [  0,   0,   4, 107]])

✓ [187] print(classification_report(svm_predict_2,y_test))
0s
                      precision    recall  f1-score   support

                   0       0.98      1.00      0.99       103
                   1       1.00      0.93      0.96        98
                   2       0.90      0.94      0.92        88
                   3       0.96      0.96      0.96       111

            accuracy                           0.96       400
           macro avg       0.96      0.96      0.96       400
        weighted avg       0.96      0.96      0.96       400

✓ [188] accuracy_score(svm_predict_2,y_test)
0s
        0.96
```

The accuracy achieved after tuning the parameter is 96% which is highest among all models used.

Comparison of accuracy Scores of all the models being used:



The algorithm that is found to be able to classify instances the most accurately among the ones tested is SVM using hyper parameter tuning with an accuracy of 96%, followed closely by Logistic Regression that was able to predict instances with an accuracy of 94.25%. The Random Forest Classifier and Decision Tree gives accuracy scores of 87.75% and 82.25% respectively.

After Training our dataset with four different models, We conclude that support vector machine model & Logistic Regression model are the best model for our dataset.

Conclusion:

1. From EDA we can see that there are mobile phones in 4 price ranges. The number of elements are almost same.

2. Half of devices have Bluetooth connectivity and another half of devices doesn't have Bluetooth connectivity.

3. There is gradual increase in battery power as the price increases.

4. Ram has continuous increase in price range while moving from low cost to vey high cost.

5. Costly phones are lighter in weight.

6. RAM, Battery power, pixels and connectivity features 'three_g' and 'four_g' plays more significant role in deciding the price range of mobile phones.

7. From above used classification models, we can conclude that Logistic Regression and SVM with using Hyperparameter tuning we can achieve best results.

Thank You