## Mobile Price Range Prediction

## Team Mate - Chetan Patil, Mrunal Badgujar, Nikhil Chaudhri, Rajesh Patil,

**Alma Better Capstone Project**

1. **Abstract:**

Mobile phones have become a common commodity and usually the most common purchased item. Thousands of types of mobiles are released every year with new features and new specifications and new designs. So, the real question is prediction is what is the real price of the mobile and to estimate the price of the mobile within the market for optimal marketing and successful launch of the product. Price has become a major factor for development of any product and its sustainability in the market. Mobile prices also impact the marketing of the mobile and also its popularity with other competitors. With the available specifications and desired designs, money is also an important factor to survive within the market. Customers usually see that they are able to buy with the specification with the given estimated price or not. So, estimating the price is an important factor before releasing the mobile and also to know about the market and competitors. In this Prediction, Dataset is collected from the existing market and different algorithms are applied to reduce the complexity and also identify the major selection features and get the best comparison within the data. This Tool is used to find the best price with maximum specifications.

1. **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.

1. **Data Summary:**

The dataset contains mobile features information (Battery power, Bluetooth, Dual sim, Front camera, Internal Memory, Mobile depth, Number of Cores, Primary camera etc) . By using these features we have to predict the mobile price range means mobile range from 0 to 3.

1. The dataset has a shape of (2000, 21) which indicates that it contains approximately 2000rows and 21columns.
2. Most of the values are in integer and some in float format
3. There are not null values in our dataset
4. The number of duplicate values is 0

* **Attribute Information:**
* **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
* ***Wi-Fi*** - 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).

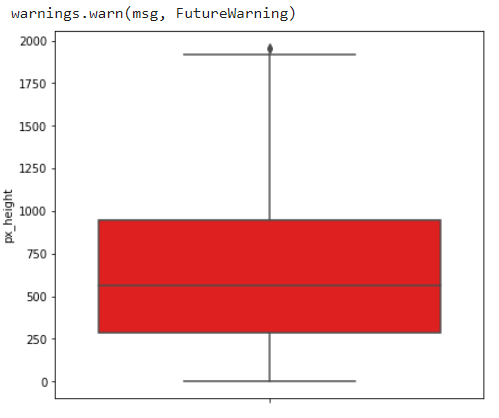
Thus, our target variable has 4 categories so basically it is a Multiclass classification problem.

1. **Pre-Processing**

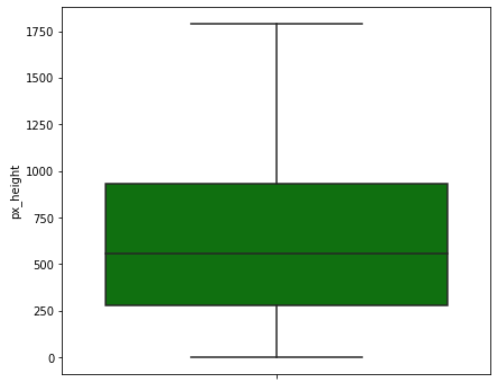
There is a need for data pre-processing because the data may be incomplete or inconsistent or noisy. There are many ways to deal with un-processed data viz:

1. **Data Cleaning:**

* By this term we mean filling in the missing values in data, identifying and removing outliers in the data, smoothening filling.
* We observed that sc\_width and px\_height have a minimum value 0. which is not possible on any mobile. We dropped the column



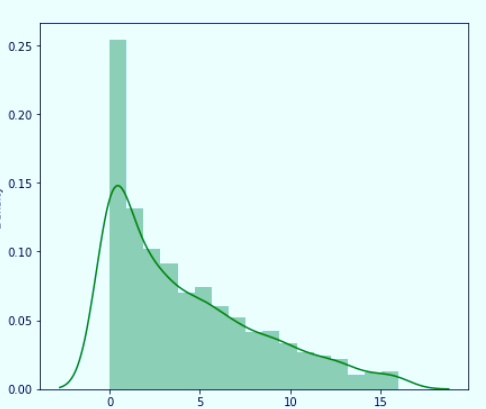
We can see the outliers present in our data need to clean data outliers removed by IQR method

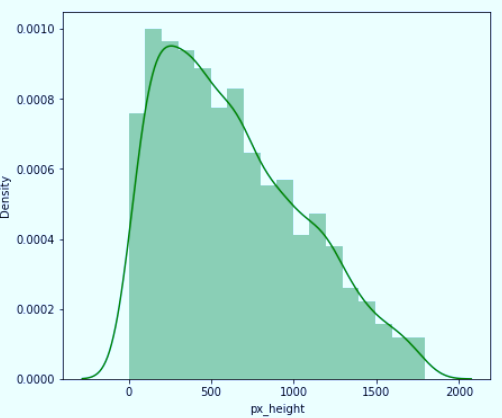
****

As we can see outliers are removed from our data so it will give us normalised data and easy to predict the prediction.

1. **Data Transformation**:

* In this stage operations like normalization and aggregation are performed.
* As we checked our dataset is rightly skewed so need to normalise it

****



Normalised the dataset.

1. **Data Reduction:**

* More number of features means the complexity in prediction we will keep only important features in this data reduction process.
* In this stage we will start with correlation matrix we will check the correlation between price with other features and keep only positive strong relation features
* Then we use Important feature score, by using this method we shortlisted 12 most important features

1. **Data Integration:**

* In this stage data is merged from different sources if needed, again redundancies are removed too.

1. **Exploratory Data Analysis:**

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

Exploratory data analysis is a statistical way of understanding the data which is usually done in a visual way. The graphs plotted in exploratory data analysis are for better understanding of data to the analyst.

### **There are various types of visualizations** –

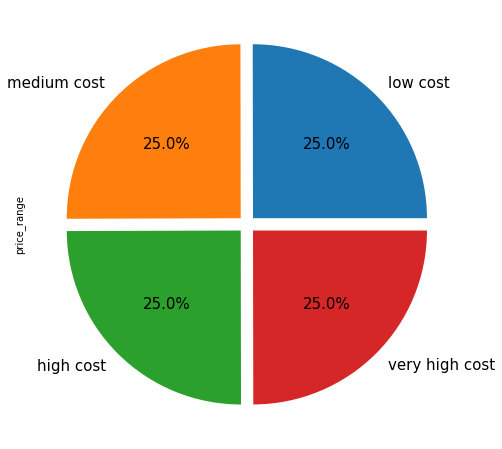
**Univariate analysis:** This type of data consists of only one variable. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.

**Bi-Variate analysis:** This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship between the two variables.

**Multivariate analysis:** When the data involves three or more variables, it is categorized under multivariate

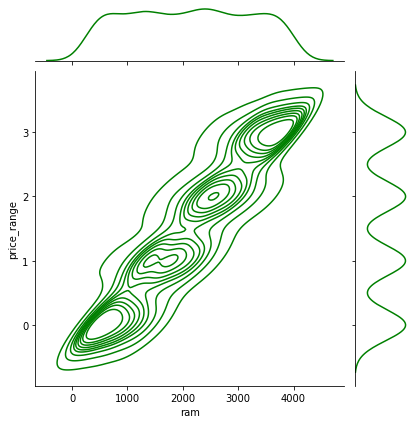
* **Let’s see the analysis by using data visualization**

# **Checking Data balance or imbalance**



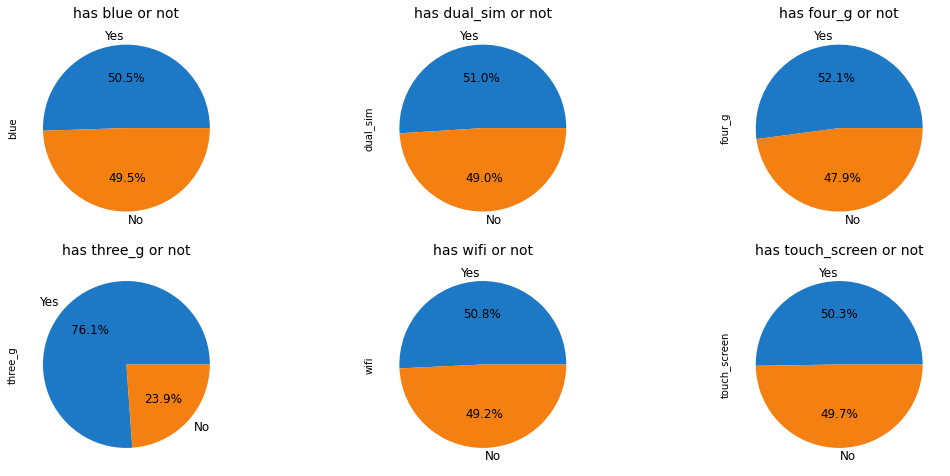
**Observation**: We can see that our target variable is equally distributed.

# **RAM EFFECT ON PRICE**



**Observation**: As Ram size increases price also increase in mobile

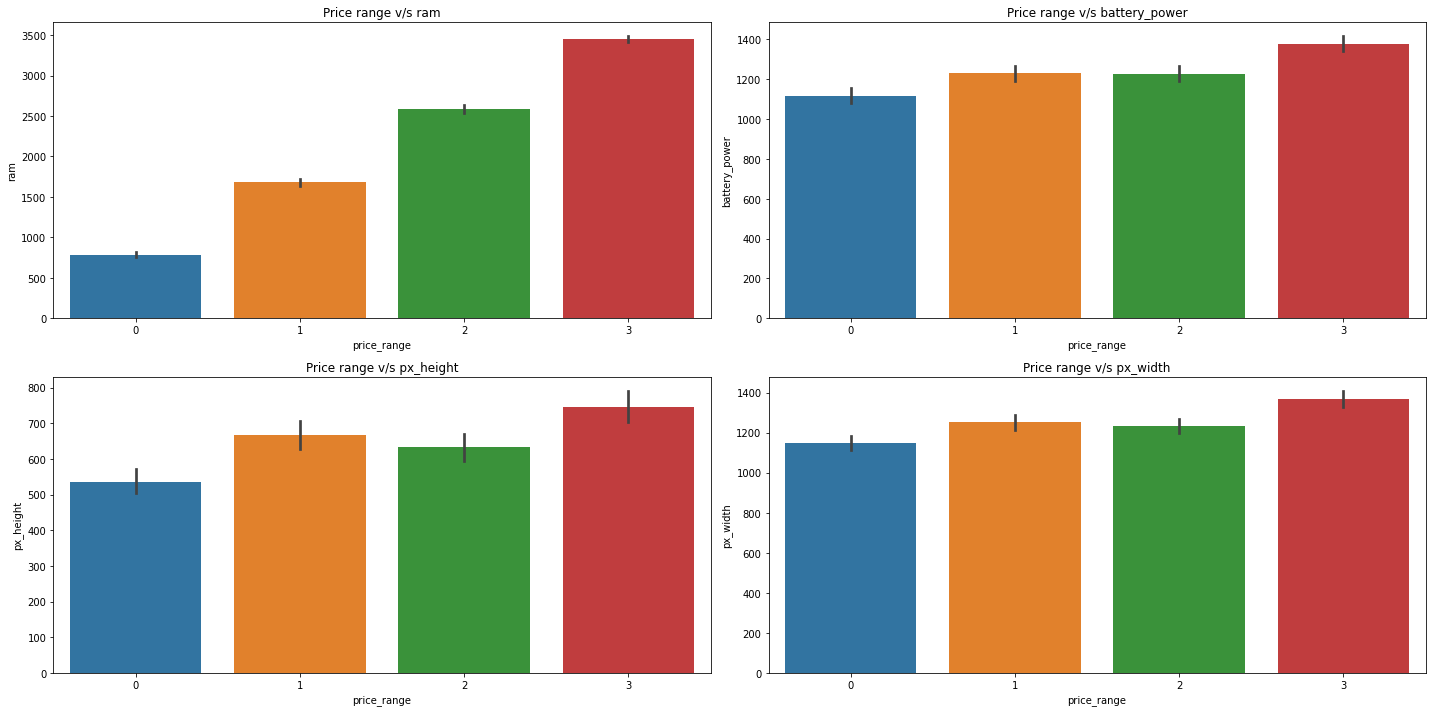
## Binary categorical variables



**Observation:**

* 1 means it has the specifications.
* 0 means it does not have the specifications.
* Percentage Distribution of Mobiles having Bluetooth, dual sim, 4G,wifi and touchscreen are almost 50%
* very few mobiles (23.8%) do not have Three\_g.

## Price range vs other features



### **Observations:**

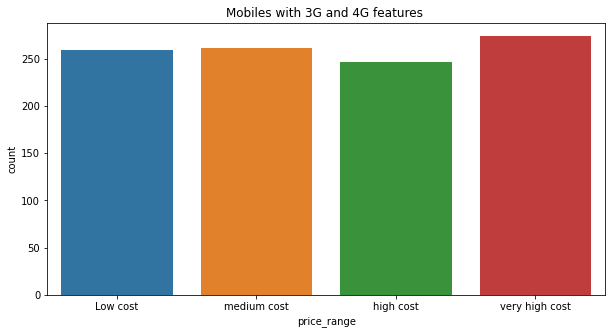
* Mobiles having RAM more than 3000MB falls under the Very high-cost category. As RAM increases, the price range also increases.
* Mobiles having RAM less than 1000 MB falls under the low-cost category.
* Mobiles with battery power more than 1300 mAh have a very high cost. And Mobiles with battery power between 1200 and 1300 mAH falls under medium and high-cost category.
* Mobiles with more than 700-pixel height and width more than 1300 have very high cost.

## Box plot price vs other features

## 

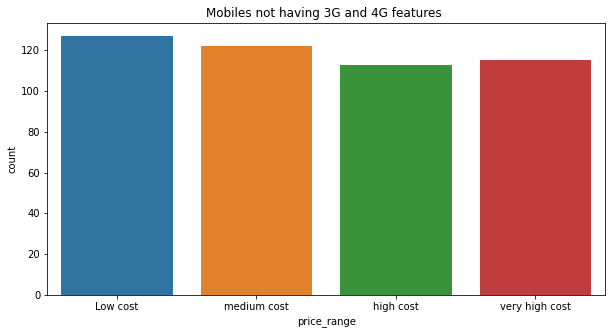
**Observation:** Ram has the strongest positive correlation with price range

# **Mobiles with both 3G and 4G.**



**Observation**:As we can see from low cost to very high-cost mobiles have both features**.**

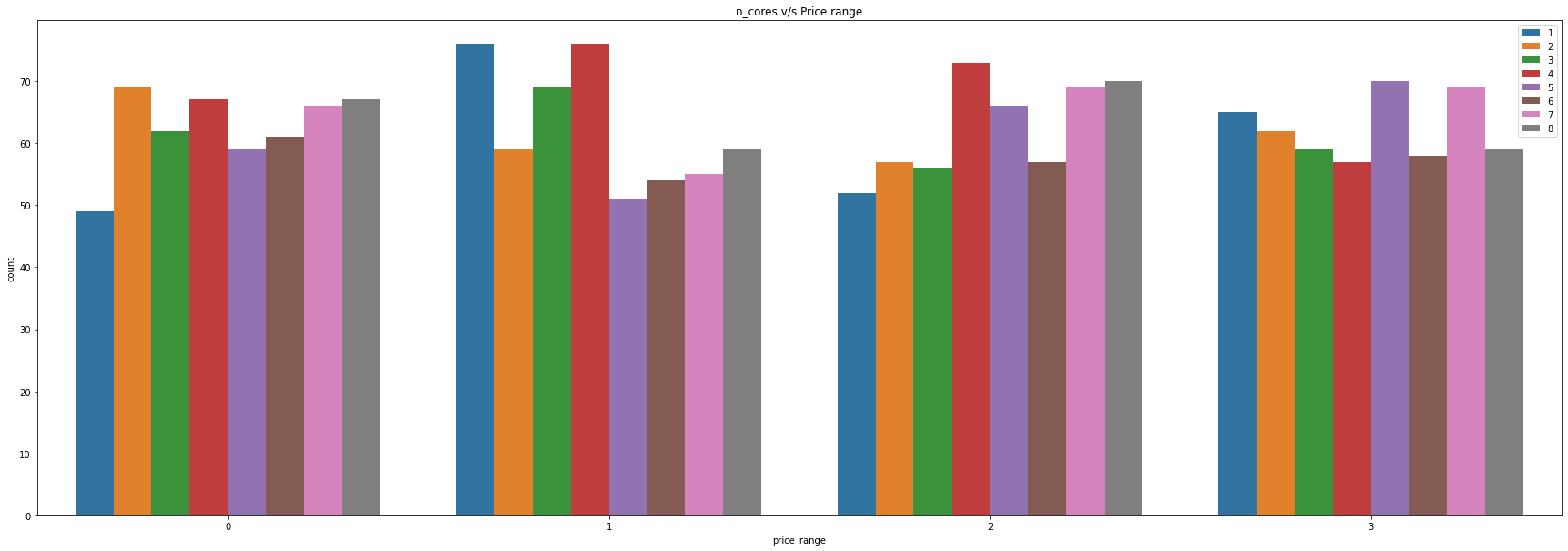
# **Mobiles not having no 3G and 4G.**



**Observation:**

* It’s very obvious that low-cost mobiles will not have 3G and 4G.
* Mobiles with very high cost may have 5G. As we know, technology changes every time.

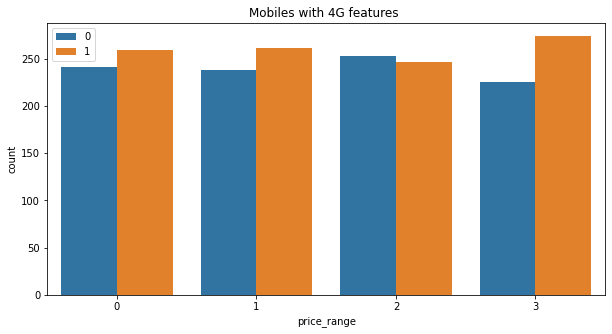
# **n\_cores v/s price range**



**Observations:**

* Price range 0 has majority of phones with 2 core processors
* Price range 1 has majority of phones with 1 and 4 core processors
* Price range 2 has majority of phones with 4 core processors
* Price range 3 has majority of phones with 5 and 7 core processors

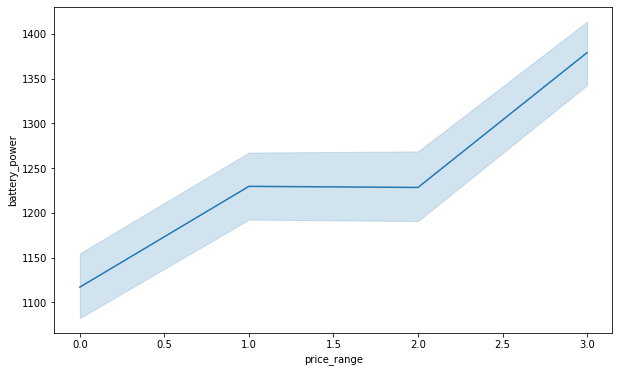
# **Mobile with 4G Features**



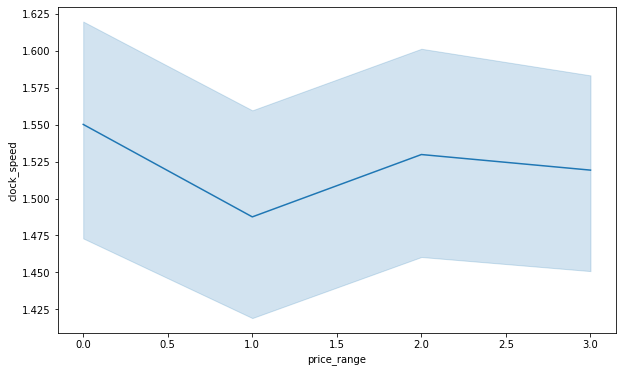
**Observation**

Majority of phones of only price range 2 dont have 4G service.

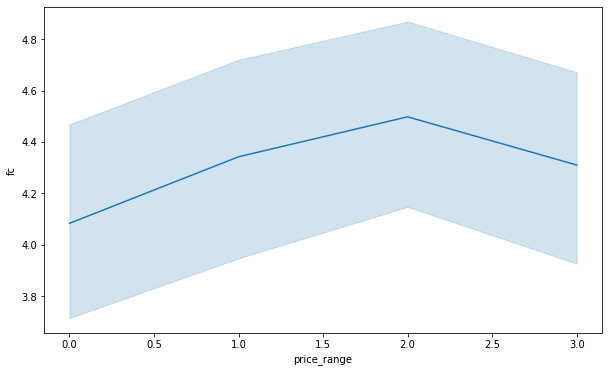
# **Let's Check which numerical feature is driving the price range most**.



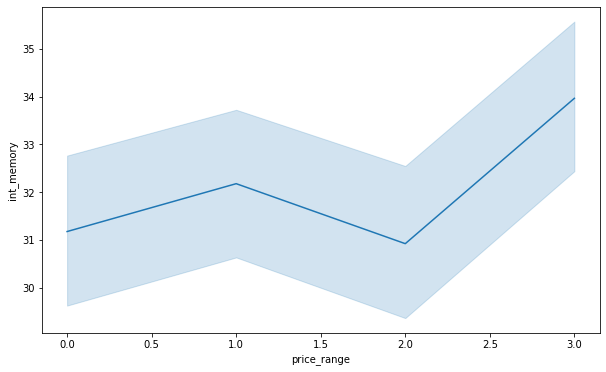
As we can see in the above graph the relationship between price and battery power is positive,As battery power increases price also increases.But in ranges 1 and 2, price is stable, meaning battery size has no effect on price.

****

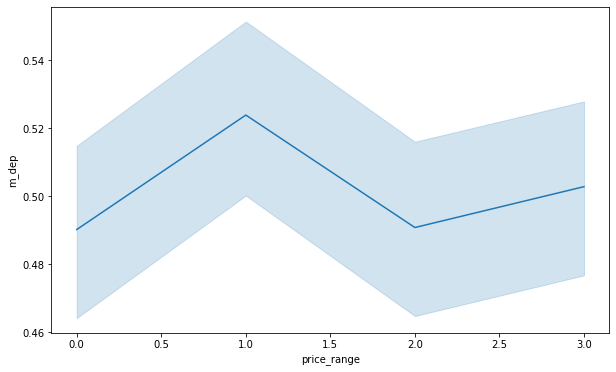
As per the graph, we see the clock speed not much affecting the price, but in the range of 1 is affected by clock speed.

****

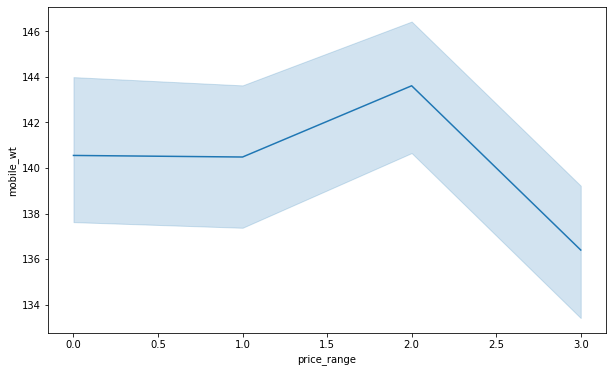
As per the graph we see the front camera affecting the price in range o to 2 but not to range 3 mobiles



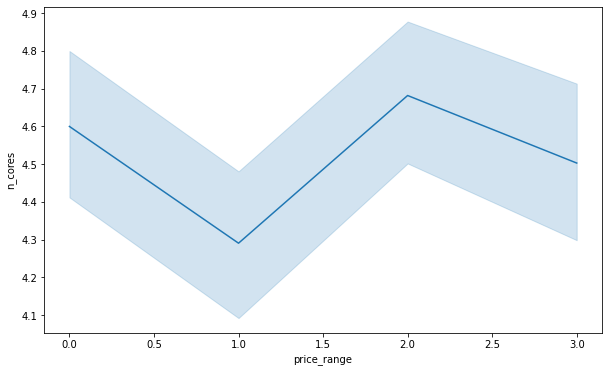
Internal memory features affecting range 0 and 1 but not range 2 and range 3 are also affected by int memory.



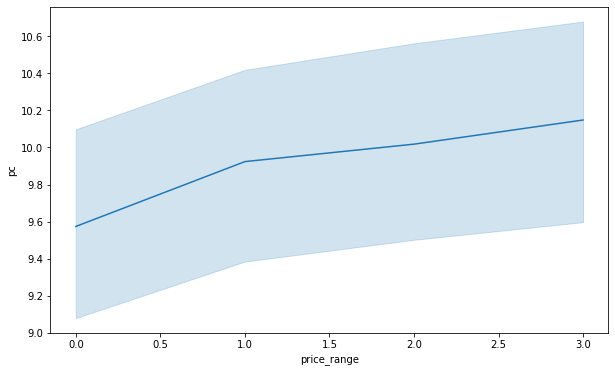
Range 1 and 3 is affected by mobile depth



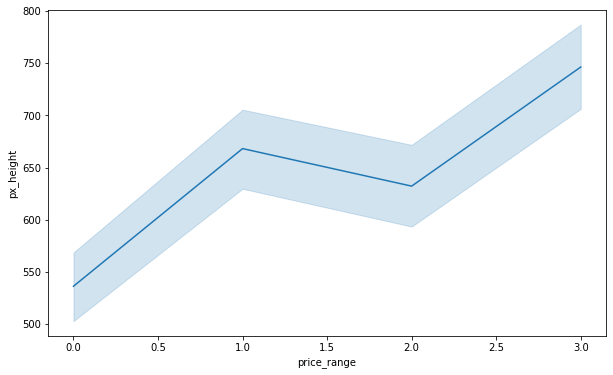
Range 2 mobiles mostly affected by mobile width



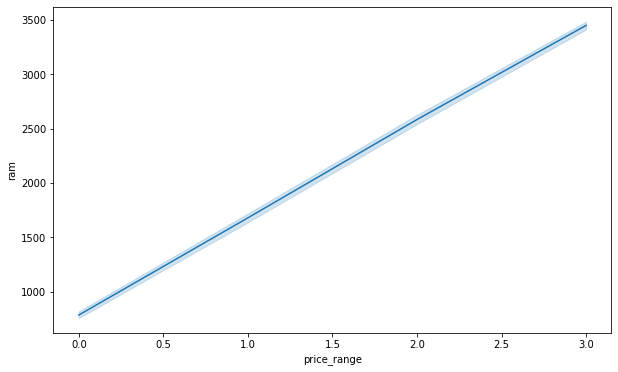
Range 0 and range 2 mobiles price affected by number of cores.



Range 1 mobile price affected by Primary camera



Range 1 and range 3 mobile prices affected by mobile height



Ram affects on all the range of mobiles as ram increases the price also increases.

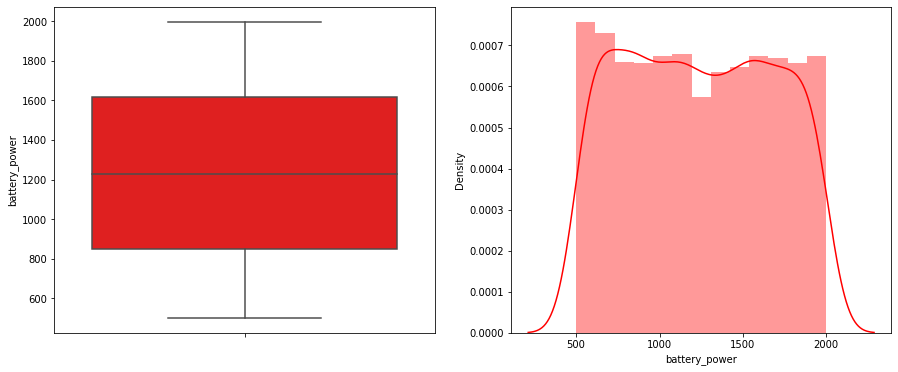
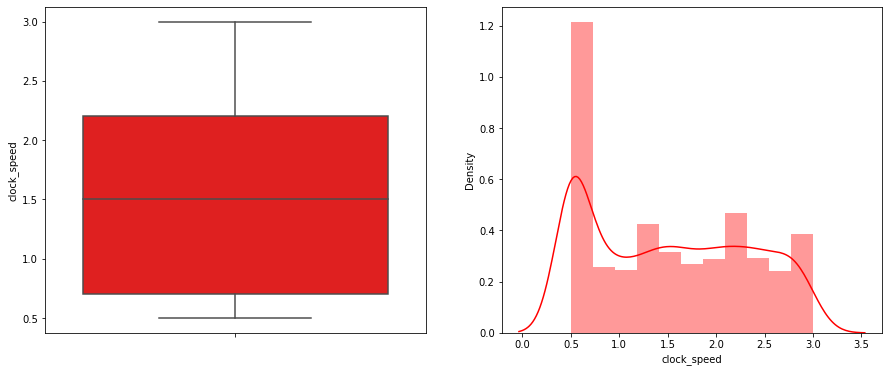
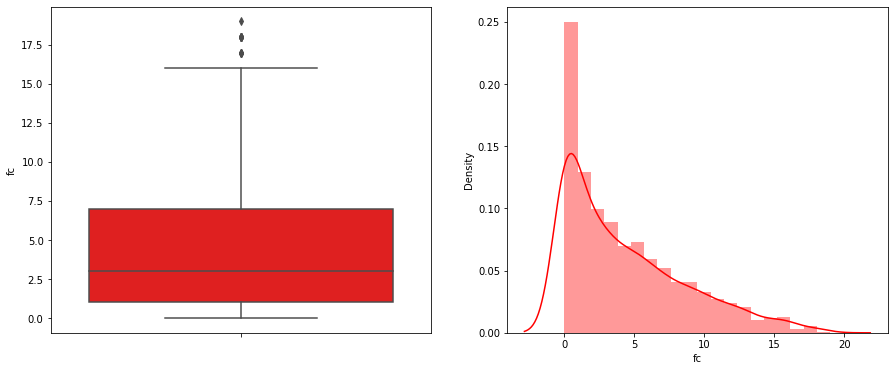
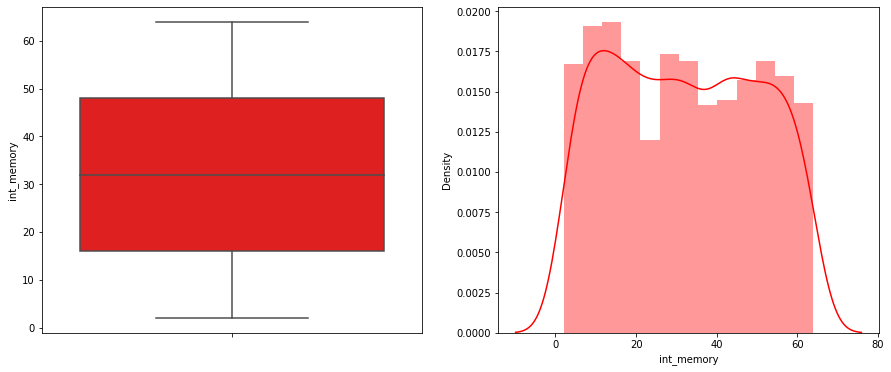
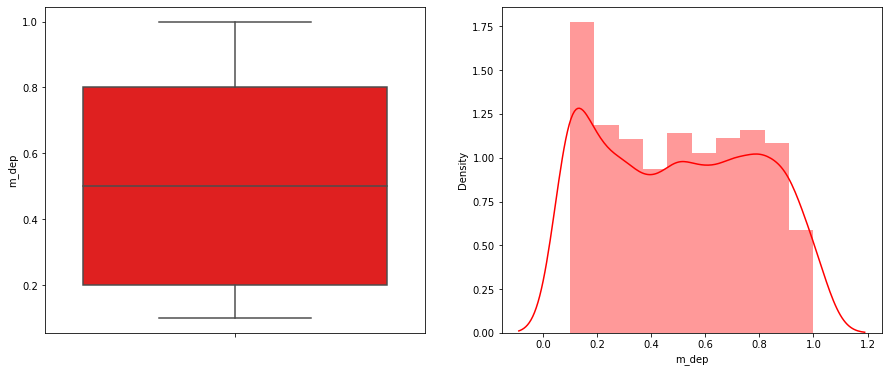
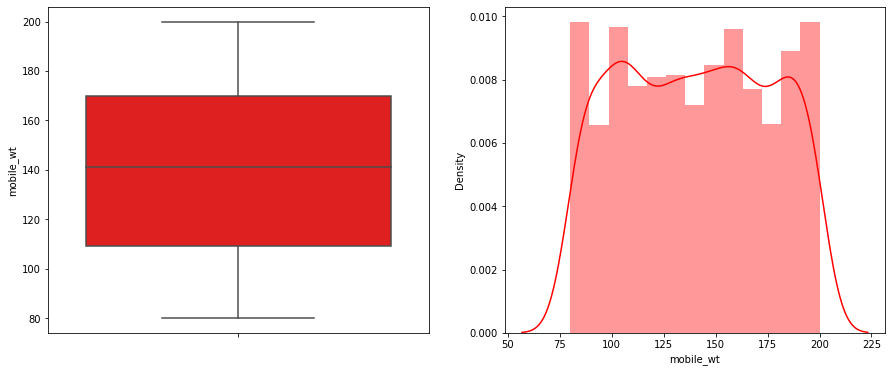
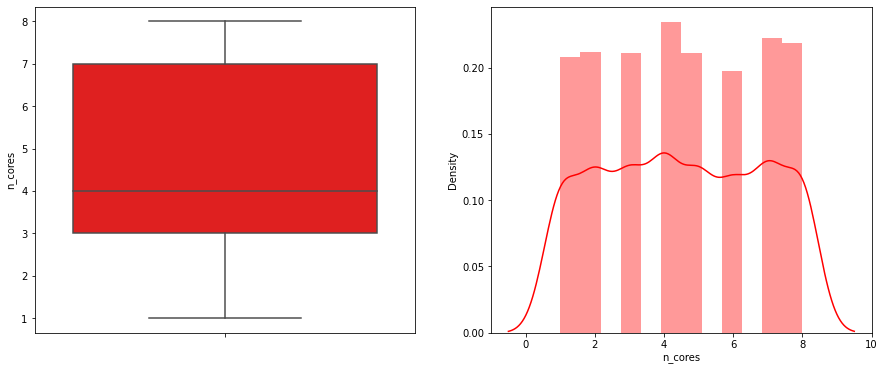
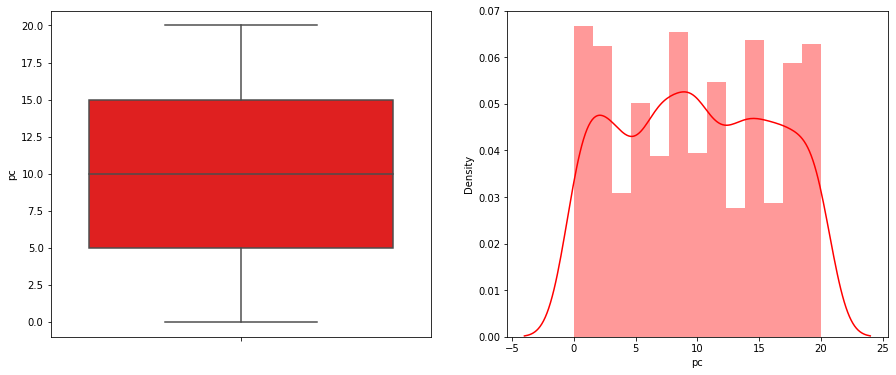
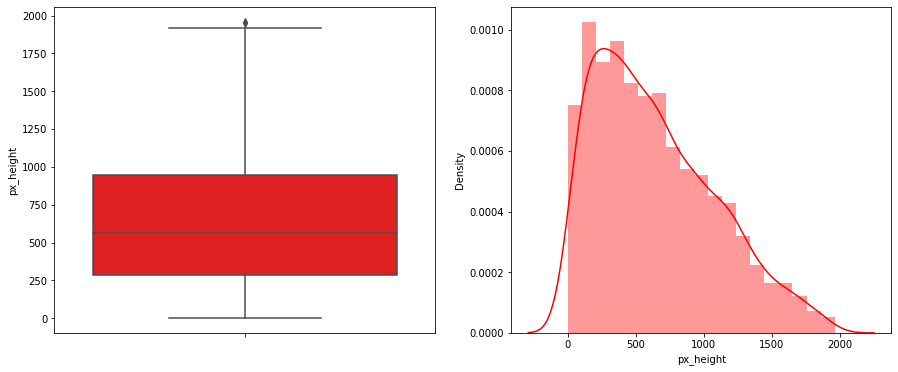
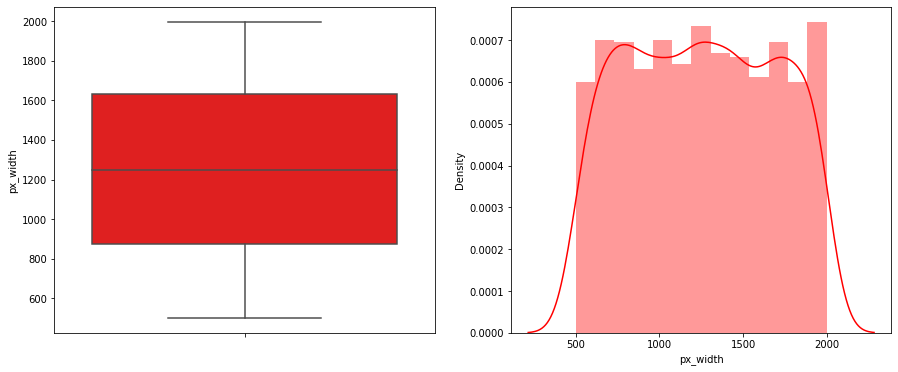
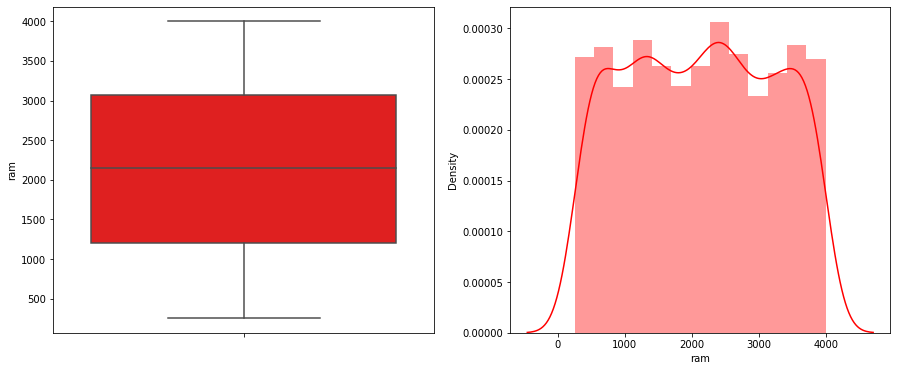
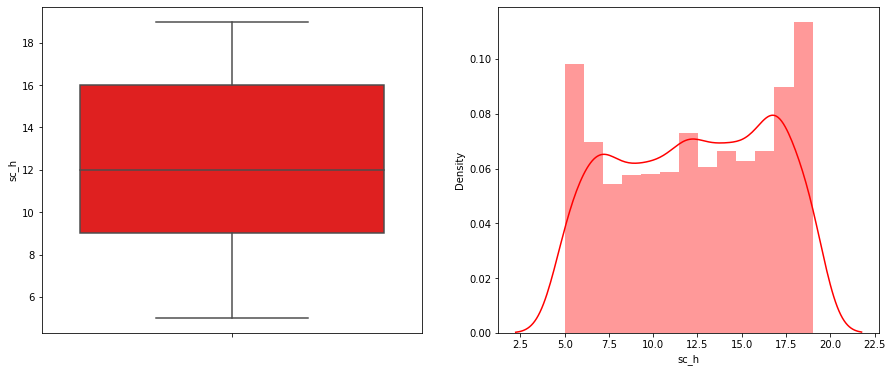
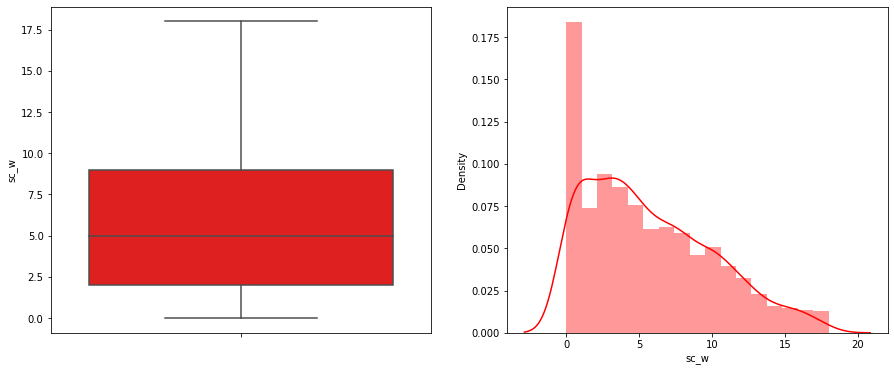
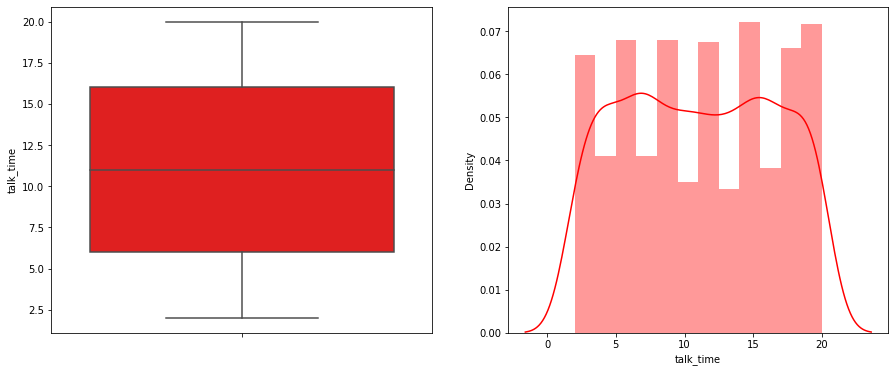
### **Observations**

* For class 1 and class2 battery power range is almost similar. As battery power increases the price also increases which is quite obvious.
* Mobiles in a very high price range (Class 3) have less weight compared to other classes. That means as the weight of mobiles decreases, price increases.
* Mobiles having max screen height and width falls in a very high price category. We can see in the line chart of Sc width and Sc height from class 2 screen width and height starts increasing with price. Similar case is with px\_height and px\_width. As resolution of screen increases the price also increases
* RAM has a clear relationship with the price range we saw in the correlation matrix also.

## Correlation between parameters

## 

* Ram has the strongest positive correlation with mobile price
* **Check the distribution of dataset**

* Data is well distrusted.
* fc and px\_height have some outliers.

# **Predictive Modelling:**

Algorithms used for predictive modelling:

1) Decision Tree

2) Random Forest classifier

3) Gradient Boosting Classifier

4) K-nearest Neighbour classifier

5) XG Boost Classifier

6) Support Vector Machine (SVM)

7) Logistic regression

## Logistic Regression

## Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features. Contrary to popular belief, logistic regression IS a regression model. 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.

## Lightbox

Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

|  |
| --- |
| accuracy of validation set 0.95 |
| accuracy of the training set 0.97 |
| accuracy of the test set 0.97 |

## Decision Tree Classifier

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.



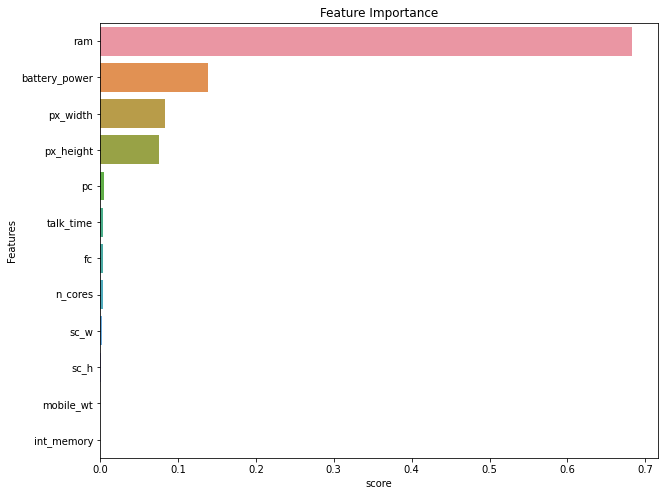
Before hyper tuning

|  |
| --- |
| accuracy of validation set 0.85 |
| accuracy of the training set 1.0 |
| accuracy of the test set 0.83 |

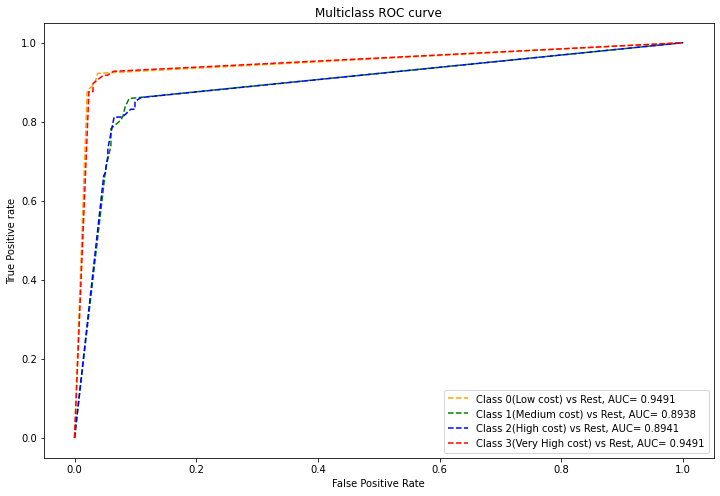
After hyper tuning

|  |
| --- |
| accuracy of the training set 0.96 |
| accuracy of the test set 0.84 |

Important Features



As we can see the Ram is the most important feature for mobile price prediction

****

**OBSERVATION**

* Train accuracy has been reduced to 96% from 100% and test accuracy has increased by 1%. Thus, we somewhat reduced the overfitting by reducing the training accuracy. However, this will not be a good model for us.
* RAM, battery power, px\_height and width came out to be the most important features
* This model classified class 0 and class 3 very nicely as we can see the AUC is almost 0.94 for both classes, whereas for class 1 and class 2 it is 0.84.

## Random Forest Classifier

## Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

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

## Random Forest Algorithm

## Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

<="" li="">

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

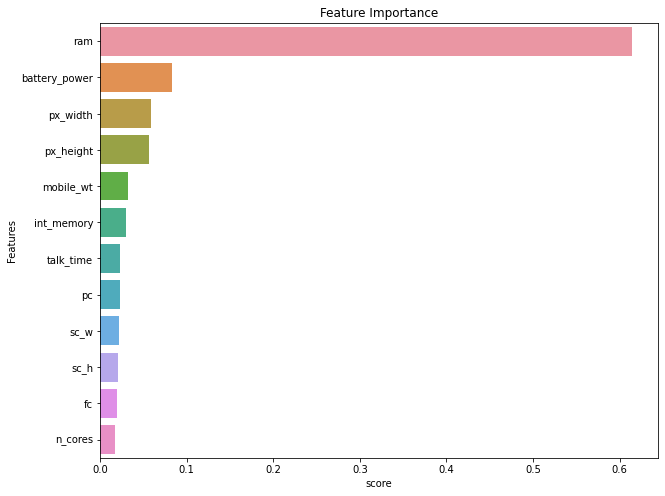
Before hyper tuning

|  |
| --- |
| **accuracy of validation set: 0.8961213650515862** |
| **accuracy of the training set : 1.0** |
| **accuracy of the testset: 0.8854961832061069** |

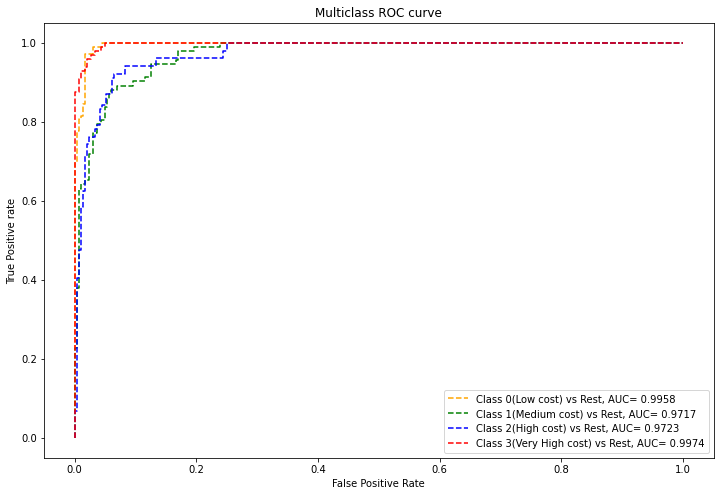
After hyper tuning

|  |
| --- |
| **accuracy of the training set: 1.0** |
| **accuracy of the testset: 0.9007633587786259** |

Important Features:



As we can Ram is the most important feature



### **Observations of Random Forest:**

Before Tuning

|  |
| --- |
| **training accuracy = 100%** |
| **test accuracy = 88%** |

Model is overfitted the data and does not generalise well. So, we tuned the hyperparameters.

After tuning:

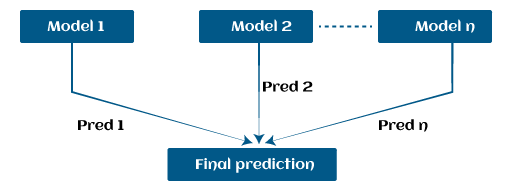
|  |
| --- |
| **Training accuracy= 100%** |
| **Test accuracy = 90%** |

From the roc curve it’s clear that the model has poorly performed to classify class 1 and class 2.

# **Gradient Boosting Classifier:**

Machine learning is one of the most popular technologies to build predictive models for various complex regression and classification tasks. **Gradient Boosting Machine** (GBM) is considered one of the most powerful boosting algorithms.

Boosting is one of the popular learning ensemble modelling techniques used to build strong classifiers from various weak classifiers. It starts with building a primary model from available training data sets then it identifies the errors present in the base model. After identifying the error, a secondary model is built, and further, a third model is introduced in this process. In this way, this process of introducing more models is continued until we get a complete training data set by which the model predicts correctly.



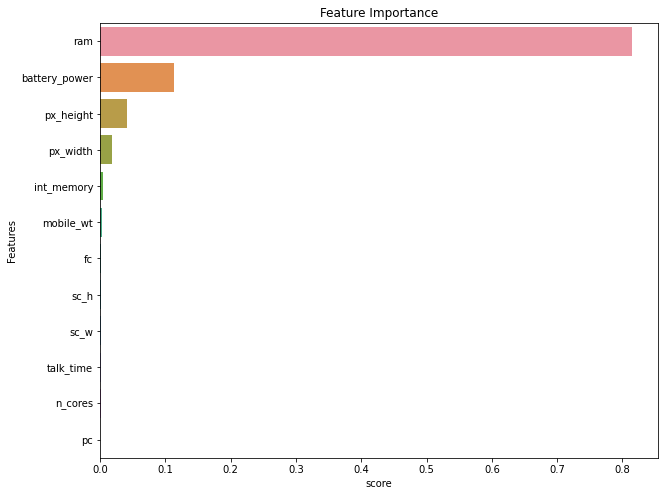
Before Tuning

|  |
| --- |
| **accuracy score of train data 1.0** |
| **accuracy score of test data 0.9134860050890585** |

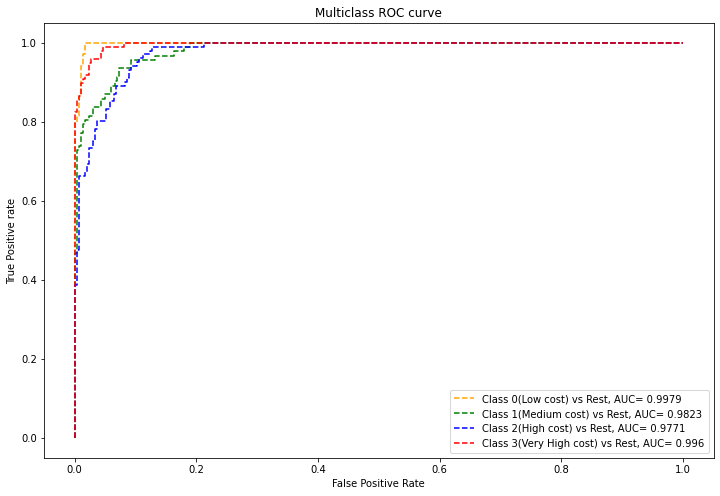
After tuning

|  |
| --- |
| **accuracy score of train data 1.0** |
| **accuracy score of test data 0.9007633587786259** |

Important Features



As we can see Ram is the most important feature for price prediction



### Observations of Gradient Boost Classifiers:

Before running

|  |
| --- |
| Train accuracy score= 100%. |
| Test accuracy score= 91% |

Model did not generalise well and overfitted the training data. so, we tuned the hyperparameters of the model.

After Hyperparameter Tuning

|  |
| --- |
| **Train accuracy score= 100%** |
| ***Test accuracy score=90%*** |

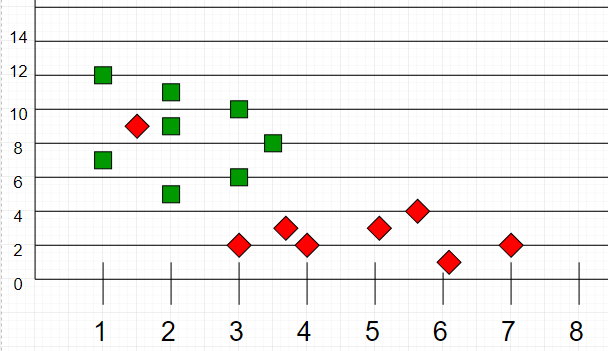
Thus, we slightly improved the model performance. However, the model is not the best.

From the ROC curve it's clear that the model was good to classify class 0 and class 3. From the classification report it's clear that recall for class 0 and class 3 is also good which is 96% and 90% respectively.

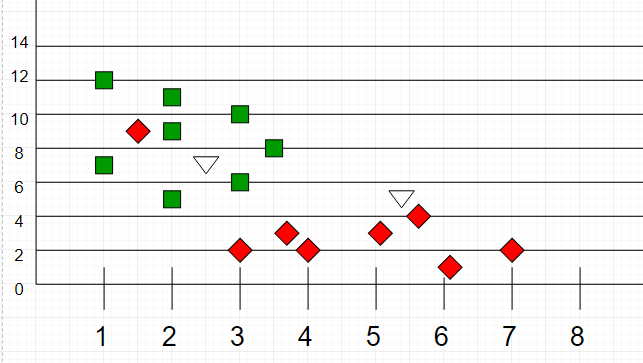
# **K Nearest Neighbour**

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as [GMM](https://en.wikipedia.org/wiki/Mixture_model), which assume a Gaussian distribution of the given data).



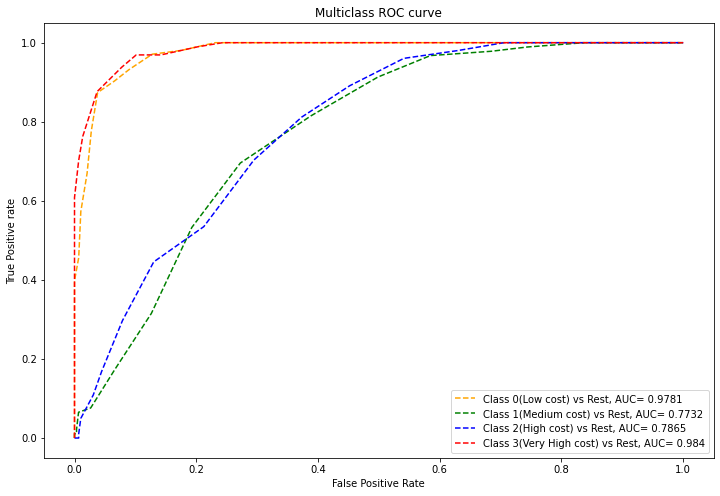
Now, given another set of data points (also called testing data), allocate these points to a group by analysing the training set. Note that the unclassified points are marked as ‘White’.



**Intuition**

If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbours belong to This means a point close to a cluster of points classified as ‘Red’ has a higher probability of getting classified as red.

Intuitively, we can see that the first point 1 should be classified as ‘Green’ and the second point should be classified as ‘Red’.



### **Observations:**

Before hyperparameters tuning:

|  |
| --- |
| **Train Accuracy:75 %** |
| **Test Accuracy:59 %** |

Clearly the Model has performed very poorly. We did hyperparameter tuning

After Hyperparameter Tuning:

|  |
| --- |
| **Train Accuracy: 75%** |
| **Test Accuracy: 70%** |

Surely, we improved the model performance and reduced overfitting but however this is not a good model for us**.**

# **XGBooster**

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

Before understanding the XGBoost, we first need to understand the trees especially the decision tree**:**

### **Decision Tree:**

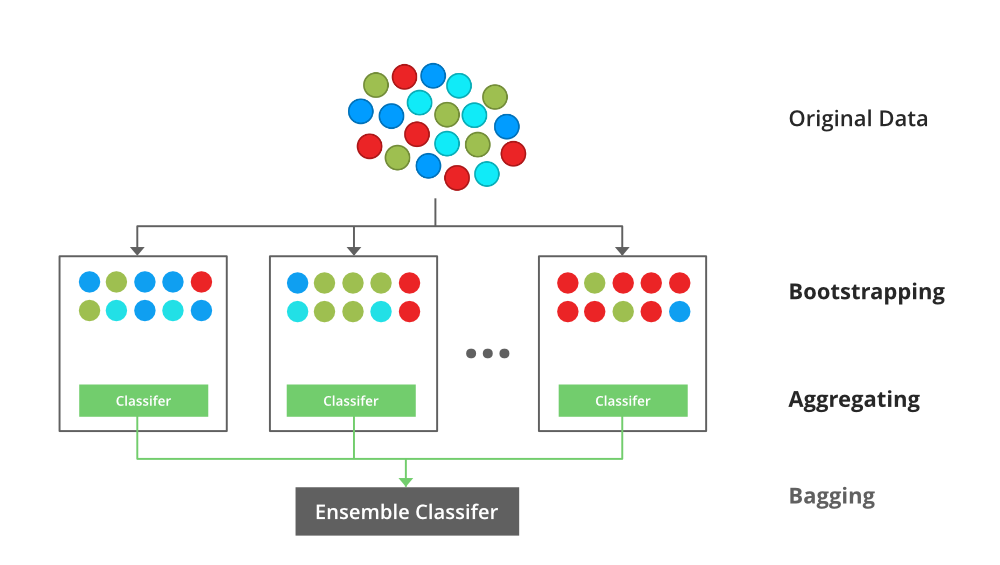
A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be *“learned”* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner calledrecursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

### **Bagging**:

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.  
Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples (or data) from the original training dataset, where N is the size of the original training set. The training set for each of the base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

Bagging reduces overfitting (variance) by averaging or voting, however, this leads to an increase in bias, which is compensated by the reduction in variance though



### **Boosting**:

Boosting is an ensemble modelling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.



### **Gradient Boosting**

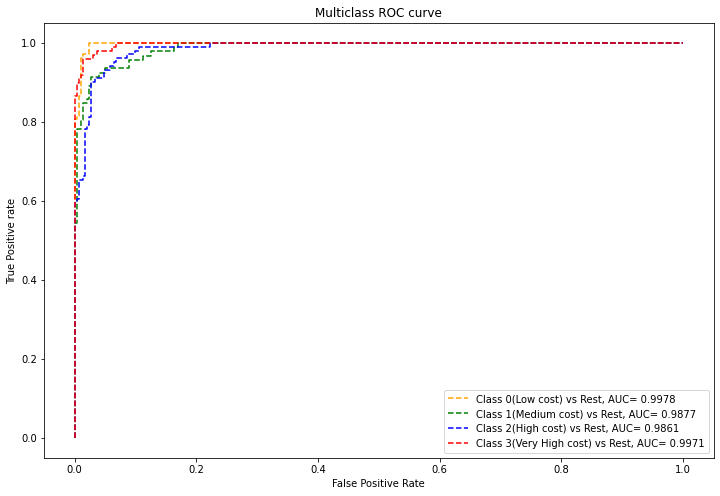
Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor’s error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of the predecessor as labels.

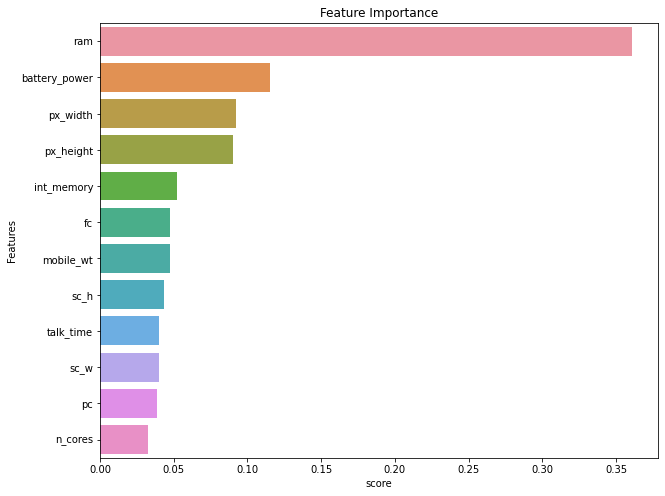
There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

### **XGBoost**

XGBoost is an implementation of Gradient Boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions.

In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.





### **Observations**

Before hyperparameter Tuning

|  |
| --- |
| Train Accuracy = 98% |
| Test Accuracy = 89% |

After hyperparameter Tuning

|  |
| --- |
| Train Accuracy = 100% |
| Test Accuracy = 92% |

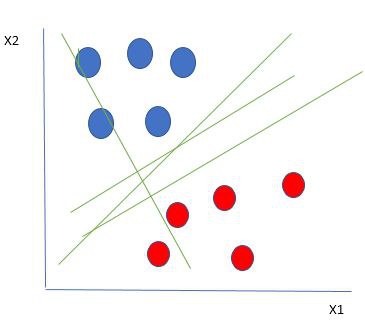
We have improved the model performance by Hyperparameter tuning. Test accuracy has increased to 92%. But still the difference of accuracy score between train and test is more than 5%. We can say model is very slightly overfitted

From the AUC-ROC curve it’s clear that the model has almost correctly predicted the class 0 and class 3.

## SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well, it's best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

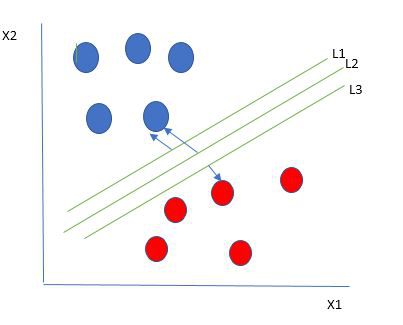
Let’s consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle.



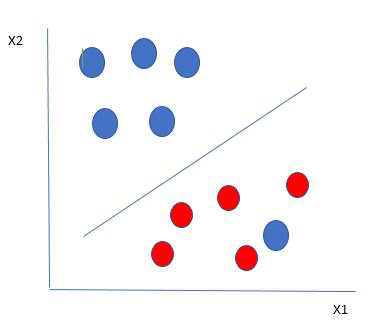
From the figure above it's very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points.

**Selecting the best hyper-plane:**

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

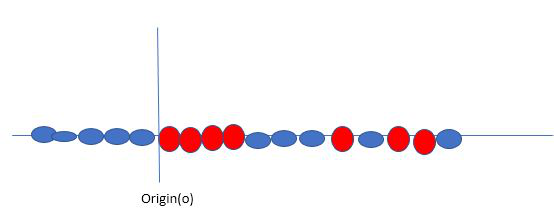


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

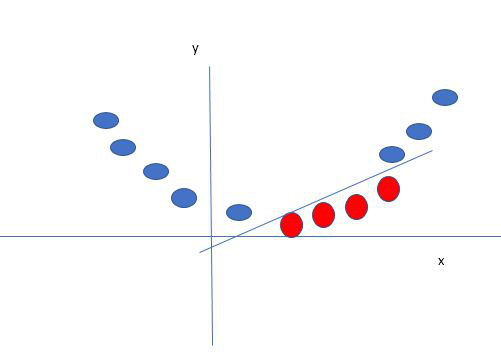


So in this type of data points what SVM does is, it finds maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called soft margins. When there is a soft margin to the data set, the SVM tries to minimize *(1/margins∧(∑penalty))*. Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is like shown in the figure above solves this by creating a new variable using a kernel. We call a point xion the line and we create a new variable yi as a function of distance from origin o.so if we plot this, we get something like as shown below

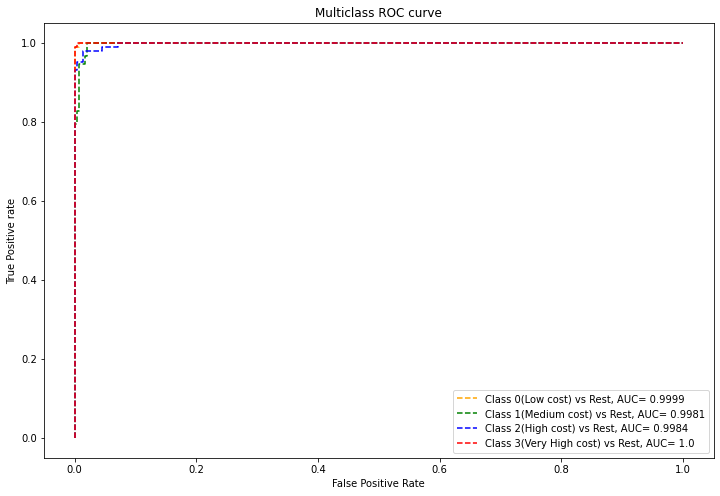


Before hyper tuning

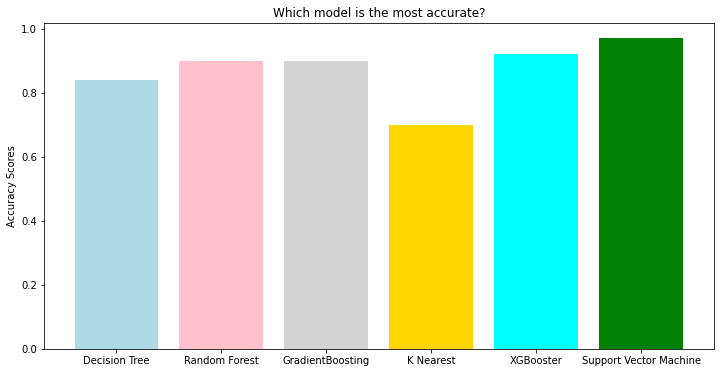
|  |
| --- |
| **accuracy of validation set : 0.959848191937486** |
| **accuracy of the training set : 0.9783301465901848** |
| **accuracy of the testset: 0.9745547073791349** |

After hyper tuning

|  |
| --- |
| **accuracy of the training set 0.982791586998088** |
| **accuracy of the testset 0.9745547073791349** |



# Conclusion



**AS we can see SVM algorithm has highest accuracy which is 97%**

* **SVM performed very well as compared to other algorithms.**
* **In terms of feature importance RAM, Battery power, px\_height and px\_weight are the important features.**
* **f1 score for individual classes is also very good. Area under the curve for each class prediction is also almost 1.**