Prediction of Mobile Model Price Using Machine Learning Techniques

Manas Nayan Mukherjee

Cohort-Boston

# Abstract: - Mobile phone has 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 specification and new designs. So, the real question is prediction is that 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. Customer usually sees that they are able to buy with the specification with the given estimated price or not. So, to 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.

***Keywords: -*** *Data Collection, Correlation Analysis, Mobile price range Prediction, Machine Learning.*

# INTRODUCTION

Smartphone industry is booming at an exponential speed. Everyday new mobiles are launched in the market and deciding the price of these mobile phones plays a crucial role. The pricing affects sale and position of the company in market. As new features are added every day in the smartphones, finding the optimum product is challenging from a user’s perspective. Supervised learning algorithms can be used to predict the values based on labelled data. These can be used in prediction of mobile prices based on the features available in the smartphone. This prediction can help the manufacturers of smartphones to estimate mobile prices and at the same time it can assist them to assure that they are paying the best price for the mobile they are purchasing. This is extremely beneficial for emerging companies to set the prices in accordance to the market standards. The most essential factor to estimate the cost is the features and the preference of user. For instance, processor plays an important role in deciding the price. Similarly, screen size or camera megapixels play a vital role in differentiating between prices of the smartphones. Youth generally prefer handsets with good processor to play games. On the other hand, older people prefer phones with larger screen sizes. Females opt to choose mobiles with better camera pixels. Hence, the features of a smartphone are key consideration for predicting the prices. Machine learning provides us best techniques for artificial intelligence like classification, regression, supervised learning and unsupervised learning and many more.

During any product launch into the market, there is a lot of variables and factors are considered and especially in mobiles many features and specification like memory is considered and also the impacting of the cost also may have impact with the competition in the market place. Mobile prices and specification are mainly considered for selection and comparison. Different tools and classifiers are used select best features and select the dataset for comparison. Since thousands of mobiles are released each year so dataset is complex to collect. So with selective feature, it is used to reduce the complexity of the dataset and get the estimate price to get an idea to release the product in the market. In this project, we aim to visualize the attributes affecting the prices of smartphones and find the correlation among them. Also, we will analyse various models and compare their accuracy values. Finally, we will predict the class values for unlabelled samples when other features are provided with the model with best performance.

# RESEARCH METHODOLOGY

# 

The research was carried out in Google Colab’s Python kernel.

The general workflow diagram of supervised ML tasks

is as follows

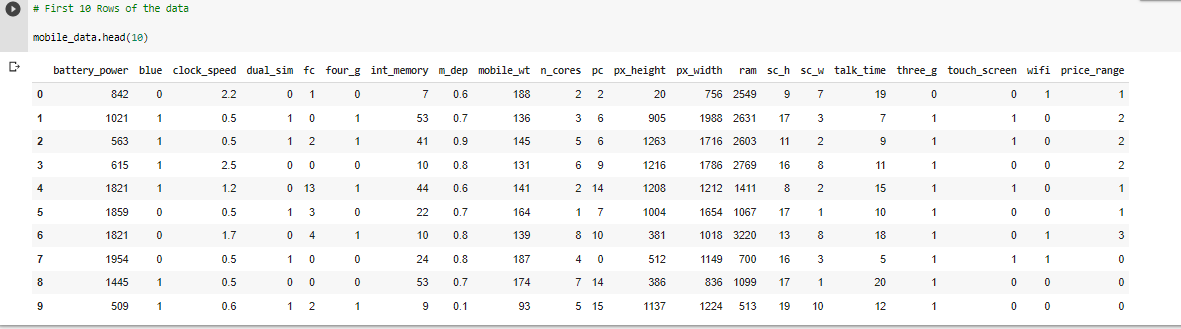


# The dataset is portioned into two – train for training the model and test for its evaluation. The computer tries to comprehend the logic behind the pricing of a mobile based on its features and uses it to forecast future instances as correctly as possible.

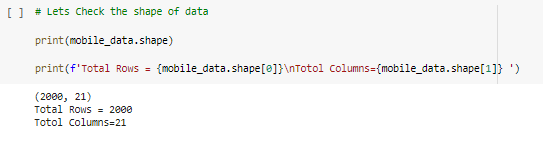
# UNDERSTANDING THE DATASET

The dataset contains 21 attributes in total – 20 features and a class label which is the price range. The features include battery capacity, RAM, weight, camera pixels, etc. 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. So, despite price traditionally being a numeric problem, the type of ML is classification (not regression) since there are discrete values in the class label. This is advantageous when using algorithms like Naive Bayes and Decision Tree as they normally don’t work well with numeric data.



The dataset contains 2000 records in total.





This is the numerical breakdown of the dataset:



# TRAINING THE PREDICATION MODEL

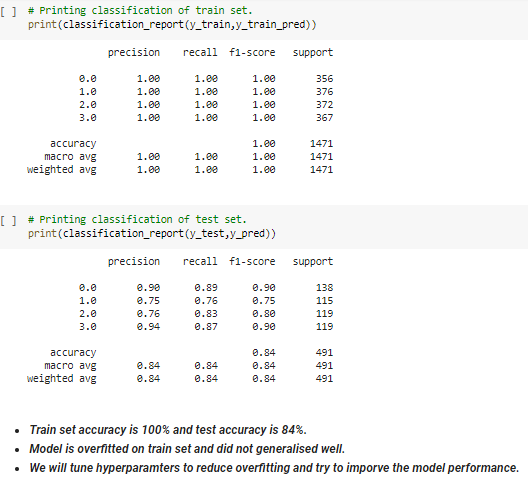
***Algorithms used for predictive modelling*:**

1) Decision Tree  
2) Random Forest classifier  
3) Gradient Boosting Classifier  
4) K-nearest Neighbors classifier  
5) XG Boost Classifier  
6) Support Vector Machine (SVM)

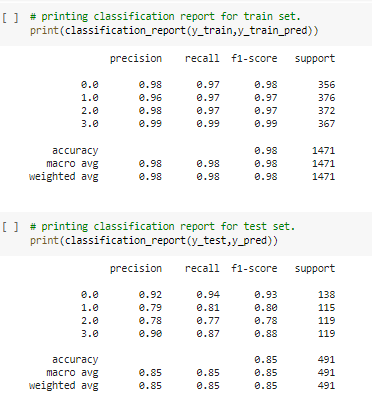
***Decision Tree:***

Decision trees and ensemble methods do not require feature scaling to be performed as they are not sensitive to the variance in the data. So here we will use X\_train,X\_test,y\_test and Y\_train which are not scaled.

**With default hyperparameters**:



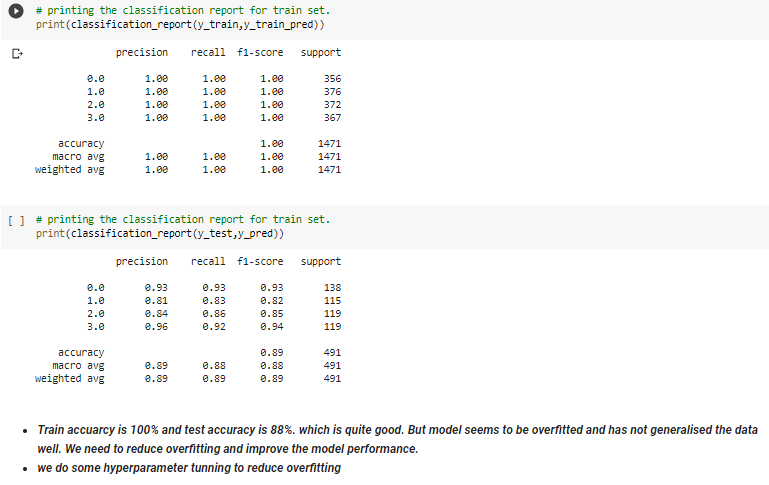
**Let's tune some hyperparameters of Decision Tree classifier:**



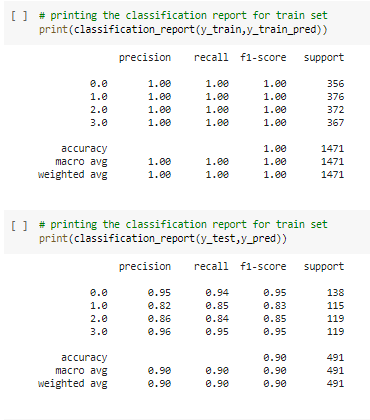
* Decision Tree Classifier-Observations:
* Train accuracy has been reduced to 98% from 100% and test accuracy is increased by 1% . Thus we somewhat reduced the over fitting by reducing the training accuracy . However this will not be good model for us.
* RAM, battery power, px\_height and width came out to be the most important features
* This model classified the class 0 and class 3 very nicely as we can see the AUC is almost 0.96 for both classes, whereas for class 1 and class 2 it is 0.88.

***Random Forest classifier:***

**With default hyperparameters**:



### **Let's do some Hyperparameter Tunning of the Random Forest model**

******

Before Tuning

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

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

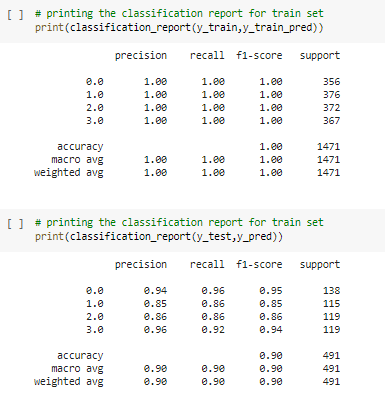
After tuning:

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

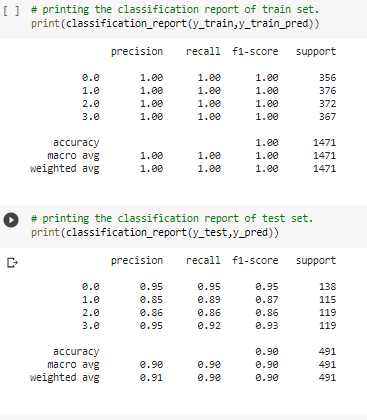
we have slightly improved the model and overfitting is reduced slightly. From roc curve it’s clear that model has poorly performed to classify class 1 and class 2.

***Gradient Boosting Classifier:***

**With default hyperparameters**:

****

**Let's tune some hyperparameters of *Gradient Boosting Classifier :***

****

Before tunning:

* Train accuracy score= 100%.
* Test accuracy score= 89%

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

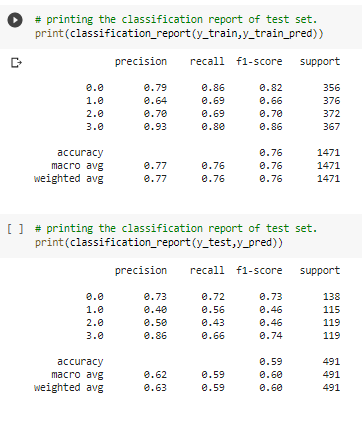
After Hyperparameter Tuning

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

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

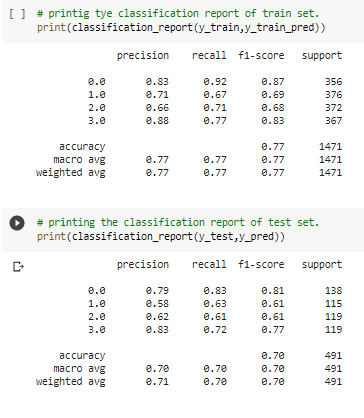
From ROC curve it's clear that model was good to classify the class 0 and class 3.From the classification report its clear that recall for class 0 and class 3 is also good which is 96% and 90% respectively.

# *K Nearest Neighbors:* With default hyperparameters:

****

**Let's tune some hyperparameters of *Gradient Boosting***

***Classifier :***

****

Before hyperparameters tuning:

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

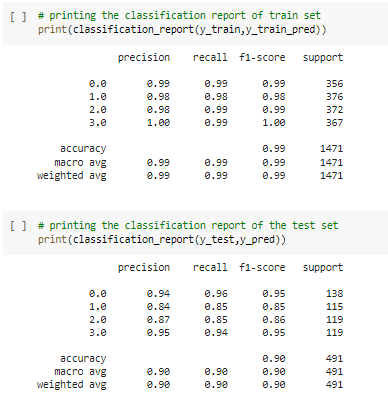
Clearly Model has performed very worst. We did hyperparameter tuning

After Hyperparameter Tuning:

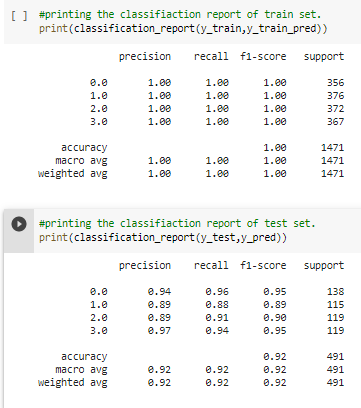
* Train Accuracy : 77%
* Test Accuracy : 70%

Surely we improved the model performance and reduced over fitting but however this is not good model for us***.***

# *XG Boost Classifier:* With default hyperparameters:

****

**Let's tune some hyperparameters of *XG Boost Classifier:***

****

Before hyperparameter Tuning

* Train Accuracy = 98%
* Test Accuracy = 90%

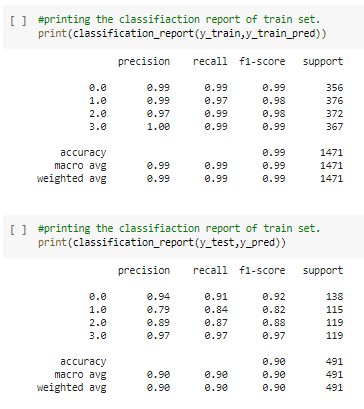
After hyperparameter Tuning

* Train Accuracy = 1%
* Test Accuracy = 92%

we have improved the model performance by Hyper parameter tuning. Test accuracy is 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 over fitted

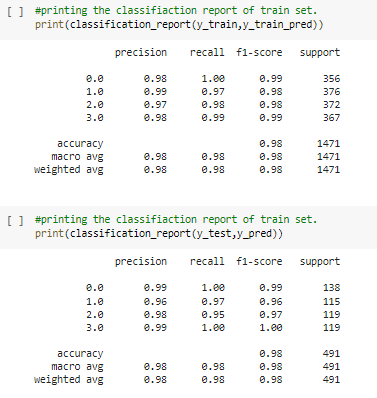
From AUC-ROC curve its clear that model has almost correctly predicted the class 0 and class 3.

# *Support Vector Machine (SVM):* With default hyperparameters:

****

**Let's tune some hyperparameters of *Gradient Boosting***

***Classifier:***

****

* Accuracy score on train set is 98.5% and Test score is 89%.
* Model seems to be overfitted as the difference between train and test accuracy score is almost 10%.\*
* After Hyperparameter tuning train accuracy remained almost same 98.3% and test accuracy score increased to 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 curve for each class prediction is also almost 1.

# CHALLENGES

Below mentioned challenges were considered while predicting the mobile price range model:

1. **Data preparation:** One of the most frequently overlooked challenges of predictive modeling is acquiring the correct amount of data and sorting out the right data to use when developing algorithms. Once the data has been sorted, one must be careful to avoid over fitting. Over-testing on training data can result in a model that appears very accurate but has memorized the key points in the data set rather than learned how to generalize.

2. **Technical and cultural barriers**: While predictive modeling is often considered to be primarily a mathematical problem, users must plan for the technical and organizational barriers that might prevent them from getting the data they need. Often, systems that store useful data are not connected directly to centralized data warehouses. Also, some lines of business may feel that the data they manage is their asset, and they may not share it freely with data science teams.

3. **Choosing the right business case:** Another potential obstacle for predictive modeling initiatives is making sure projects address significant business challenges. Predictive modeling initiatives need to have a solid foundation of business relevance.

Hence, we Implemented various classification algorithms, out of which the SVM(Support vector machine) algorithm gave the best performance after hyper-parameter tuning with 98.3% train accuracy and 97 % test accuracy.

# VI. CONCLUSION

The model trained using LDA was found to predict mobile price classes most accurately (95%). The accuracy of the models can be improved by doing some data preprocessing steps like normalization and standardization. Feature selection and extraction algorithms can be used to remove unsuitable and duplicative features to get better results. The same procedure used in this paper can be applied to predict the prices of other products like cars, bikes, houses, etc. using the archival data containing features like cost, specifications, etc. This would help organizations and consumers alike to make more educated decisions when it comes to price.

This study is focused on utilizing the efficiency of machine learning models in prediction of prices for smartphones. Various experiments were carried out to determine the performance of various supervised machine learning algorithms. The algorithms used are Logistic Regression, Decision Tree, Random Forest and K- Nearest Neighbour. For the given dataset, KNN performed very worst as compared with other models. The best model came out to be SVM after hyper-parameter tuning with an accuracy of 98% was obtained. Moreover, the hyper-parameter tuning matrix was plotted for all the models that were used in the study and the classification report was generated that contains the values for precision, recall, F1 score and support. The chosen algorithm could successfully predict the class for these samples and would be helpful when it comes to predict the price.