Prediction of Mobile Model Price using Machine Learning Techniques

Sri Harish A

Data science trainees,
Alma Better, Bangalore

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 key challenge in prediction is what the true price of the mobile is and how to estimate the price of the mobile within the market for optimal marketing and a successful product launch. 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.

Keywords: Machine Learning, Data Collection, Forward Selection, Backward Selection.

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.

Data Description -

- **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 mega pixels
- **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 mega pixels
- **Px_height** - Pixel Resolution Height
- **Px_width** - Pixel Resolution Width
- **Ram** - Random Access Memory in Mega Bytes
- **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).
- Thus our target variable has 4 categories so basically it is a Multiclass classification problem.

Introduction

Price always has a important impact factor in the product buying aspect and also in the mindset of the buyer who would consider "what is the worth and is it good to buy within this range". 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. In Mobile there are many specification and features like camera, video, quality of processor, quality of the material. There is many constraints in consideration of the price, as the product should be economical and reachable with overall consideration. Mobile Prices and Specification is 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 Prediction, There are many multiple variables to be considered to get the précised results of the price and other features. Of the mobile dataset this will help the buyer and also the marketer and the developer to get precise information from historical data of mobile phones and help them to decide the price range which is fine and satisfactory.

Steps involved:

- **Exploratory Data Analysis**

  After loading the dataset we performed this method by comparing our target variable that is Price range with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

- **Null values Treatment**

  Our dataset had a few null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project in order to get a better result.

- **Treating mismatched values**

  There were some mismatched values in the data set which were

converted into NaN values and then imputed using KNN imputer where Euclidean distance is used to find the nearest neighbor.

- **Encoding of categorical columns**

  We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

- **Feature Selection**

  In these steps we used algorithms like SelectKbest to check the results of each feature i.e. which feature is more important compared to our model and which is of less importance.

  Next we used Chi2 for categorical features and ANOVA for numerical features to select the best feature which we will be using further in our model.

- **Standardization of features**

  Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it. The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

- **Fitting different models**

  For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **SVM Classifier**
3. **Random Forest Classifier**
4. **XG Boost classifier**
5. **K Neighbors classifier**

- **Tuning the hyper parameters for better accuracy**

  Tuning the hyper parameters of respective algorithms is necessary for getting better accuracy and to avoid over fitting in case of tree based models
  Like Random Forest Classifier and XG Boost classifier.

Algorithms:

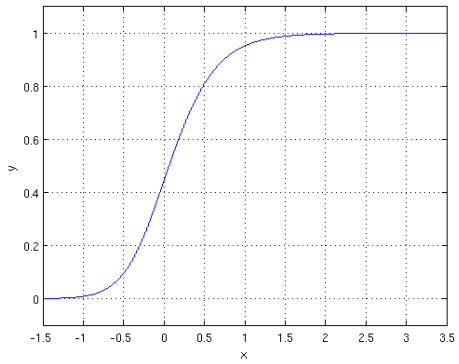
1. Logistic Regression:

Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical

formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

F(x) = 1/1+e ^ (-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:

$$\ln L(\mathbf{y},\beta)=\ln\prod_{i=1}^{n} f_i(y_i)=\sum_{i=1}^{n}\left[y_i \ln\left(\frac{\pi_i}{1-\pi_i}\right)\right]+\sum_{i=1}^{n}\ln(1-\pi_i)$$

.

2. Support Vector Machine Classifier:

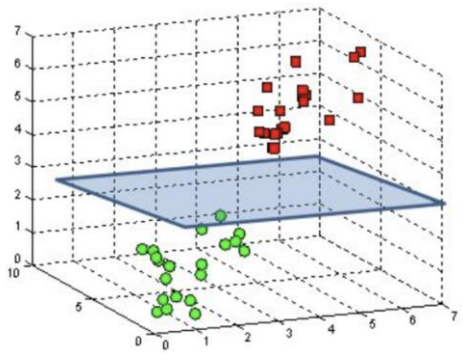
SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyper plane at a higher order dimension. In SVM we use the optimization algorithm as:

$$\min_{\xi,w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i,$$

$$\xi_i \geq 0;\ i = 1,\ldots,m.$$

where C is a cost parameter and ξ_i's are slack variables.

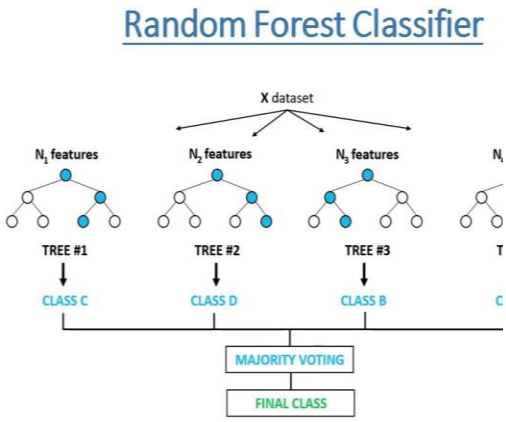


We use hinge loss to deal with the noise when the data isn't linearly separable.

Kernel functions can be used to map data to higher dimensions when there is inherent non linearity.

3. Random Forest Classifier:

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.

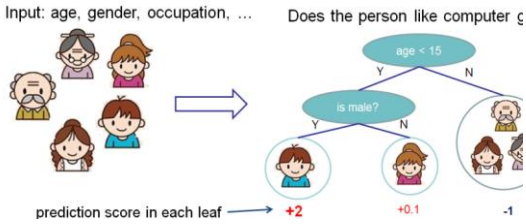


current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A 'greedy' way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

4. XG Boost-

To understand XG Boost we have to know gradient boosting beforehand.

- **Gradient Boosting-**
  Gradient boosted trees consider the special case where the simple model is a decision tree



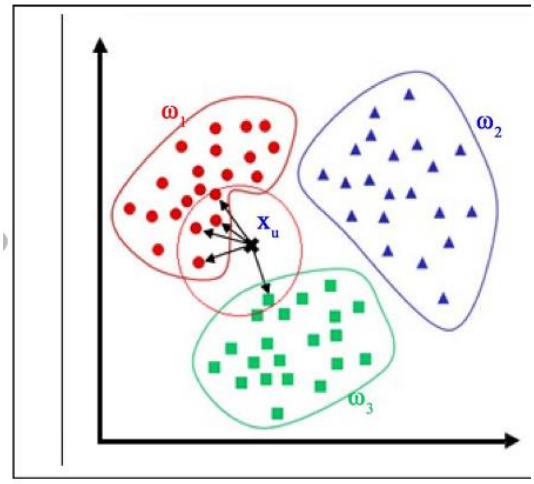
In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).
When building a decision tree, a challenge is to decide how to split a

XG Boost is one of the fastest implementations of gradient boosting trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XG Boost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

5. K Neighbors classifier

It is one of the simplest and widely used classification algorithms in which a new data point is classified based on similarity in the specific group of neighboring data points.

For a given data point in the set, the algorithms find the distances between this and all their **K** numbers of data point in the dataset close to the initial point and votes for that category that has the most frequency. Usually, **Euclidean distance** is taking as a measure of distance. Thus the end resultant model is just the labeled data placed in a space. This algorithm is popularly known for various applications like **genetics**, **forecasting**, etc.



KNN reducing over fitting is a fact. On the other hand, there is a need to choose the best value for K. So now how do we choose K? Generally we use the Square root of the number of samples in the dataset as value for K. An optimal value has to be found out since lower value may lead to over fitting and higher value may require high computational complication in distance. So using an error plot may help. Another method is the elbow method. You can prefer to take root else can also follow the elbow method.

Model performance:

Model can be evaluated by various metrics such as:

1. **Confusion Matrix**-
The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

2. **Precision/Recall**-
Precision is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP
Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

3. **Accuracy**-
Accuracy is given by the number of correctly classified examples divided by the total number
of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

4. **Area under ROC Curve(AUC)**-
ROC curves use a combination of the true positive rate (the proportion of positive examples

predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

Hyper parameter tuning:

Hyper parameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyper parameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyper parameters grid that can be adjusted according to the business problem. Hyper parameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

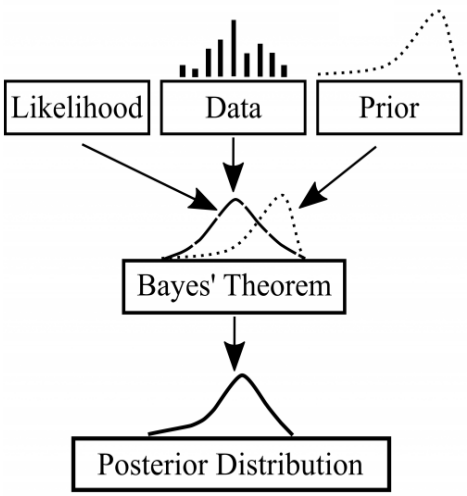
We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyper parameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV-**Grid Search combines a selection of hyper parameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

2. **Randomized Search CV-** In Random Search, the hyper parameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyper parameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyper parameters is beyond the scientist's control

3. **Bayesian Optimization-** Bayesian Hyper parameter optimization is a very efficient and interesting way to find good hyper parameters. In this approach, in naive interpretation way is to use a support model to find the best hyper parameters. A hyper parameter optimization process based on a probabilistic model, often Gaussian Process, will be

used to find data from data observed in the later distribution of the performance of the given models or set of tested hyper parameters.



As it is a Bayesian process at each iteration, the distribution of the model's performance in relation to the hyper parameters used is evaluated and a new probability distribution is generated. With this distribution it is possible to make a more appropriate choice of the set of values that we will use so that our algorithm learns in the best possible way.

Conclusion:

This Project deals with the predication of the price range and the features of the mobile .It uses Feature selection to give precise features to be selected and get maximum accuracy results.
To find the optimal model, the accuracy score was chosen as the best statistic.
The test accuracy for the tree based classification models hovered around 89%.
For this dataset, KNN produced the lowest accuracy score.
Linear classification models like SVM and Logistic regression gave high accuracy scores.
After performing hyper parameter tuning

SVM
Train accuracy: 98%
Test accuracy: 97%

Logistic Regression
Train accuracy: 98%
Test accuracy: 97%

Feature importance showed that Ram and battery power were the top two most significant features.
AUC scores for all four classes were almost close to 1 for both SVM and Logistic regression.

OUTCOMES OF THE WORK

- Cost prediction is the very important factor of marketing and business. To predict the cost same procedure can be performed for all types of products for example Cars, Foods, Medicine, and Laptops etc.

- Best marketing strategy is to find optimal product (with minimum cost and maximum specifications). So products can be compared in terms of their specifications, cost, manufacturing company etc.

- By specifying economic range a good product can be suggested to a costumer

FUTURE WORK EXTENSION

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

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

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