**Mobile price range prediction**

**Akhilesh Kumar**

**Abstract:** A predictive model was built from the mobile price range dataset to predict the range of a mobile phone based on different features. The model was built using different algorithms such as linear regression, random forest, decision trees, XG boost and KNN. The performance parameters of these model were closely checked in order to figure out the best model for the mobile price range prediction. KNN seemed to be the predict the price range more closely than the other algorithms for all the classes as the performance parameters for the KNN were the most consistent than other algorithms. Also, XG boost seemed to overfit the data.

**Keywords:** *predictive, linear regression, random forest, decision trees, XG boost, KNN, overfit*

**Introduction:** Price is the most effective attribute of marketing and business. It is the most important factor that decides the sales of that product. Mobile technology is a technology where users go, this technology also goes. The mobile phone is stimulating one of the most important technological revolutions in human history. This statement is not hyperbole. There are more mobile phones in use today than there are people. This portable technology consists of two-way communication, computing and networking technology (Mobile Technology | IBM, n.d.). The number of mobile users in the world in 2019 is about 3.2 billion and increasingly in 2020 is about 3.5 billion (29+ Smartphone Usage Statistics: Around the World in 2020, n.d.). Different commercial activities, university courses, entertainment, communications are also done by a phone. Different organizational tasks, meetings also maintained and held virtually in this pandemic situation. As well as the use of mobile phones is increasing day by day and the prices also vary by their different attributes. Nowadays, mobile phones are selling in a huge number and within a short timespan new version with new features are launched to market. There are many features which are important to consider a mobile price like brand, display, resolution, ram, camera, processor, chipset etc. So, it becomes very important for a company to decide on which features it should focus upon to maximize the sales of that particular mobile phone in the market. The current paper aims to figure out which of the attributes are the most important ones in predicting the price of the mobile phone in the market based on the data provided by AlmaBetter. The dataset contained a list of columns or features including total energy a battery can store in one time measured in mAh, presence of bluetooth or not, speed at which microprocessor executes instructions,has dual sim support or not, front Camera mega pixels etc. The model was trained and validated using supervised ML models such as Linear regression, random forest, XG boost and KNN.

**Linear regression:** Linear regression is used to study the linear relationship between a dependent or target variable Y (price range) and one or more independent variables X (primary camera megapixels, ram, weight etc.). The dependent variable Y must be continuous, while the independent variables may be either continuous (battery power, clock speed, internal memory etc.), binary (Wi-Fi, 4G, touch screen etc.), or categorical. The initial judgment of a possible relationship between two continuous variables should always be made based on a scatter plot (scatter graph). This type of plot will show whether the relationship is linear or nonlinear. Performing a linear regression makes sense only if the relationship is linear. Other methods must be used to study nonlinear relationships.

**Random forest:** Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. As the name suggests, a Random Forest is a tree-based ensemble with each tree depending on a collection of random variables. More formally, for a p-dimensional random vector X = representing the real-valued input or predictor variables and a random variable Y representing the real-valued response, we assume an unknown joint distribution . The goal is to find a prediction function for predicting Y. The prediction function is determined by a loss function and defined to minimize the expected value of the loss .

**XG boost:** Extreme Gradient Boosting (XGBoost) can be defined as an improved version of the Gradient boosting algorithm, and this algorithm considers one of the machine learning techniques\tools that applied for classification and regression problems. The idea behind its concept is to boost the weak learner to become stronger using the decision trees mechanism. This improved version utilized a more regularized model in order to reduce and control the overfitting of the model to improve its performance. Basically, the XGBoost adopted the three main techniques of the gradient boosting, which are Regularized, Gradient and Stochastic boosting to enhance and tune the model. Moreover, it has the ability to decrease the time consumption alongside using the optimal resources of memory, parallel execution and handling the missing values while generating the tree construction.

**KNN:** KNN is a method which is used for classifying objects based on closest training examples in the feature space. KNN is the most basic type of instance-based learning or lazy learning. It assumes all instances are points in n-dimensional space. A distance measure is needed to determine the “closeness” of instances. KNN classifies an instance by finding its nearest neighbors and picking the most popular class among the neighbors.

**Model deployment:** The machine learning modeling process comprises of various operations performed from collection of raw data to the implementation of the algorithms to learn. The various sub-divisions are listed below:

* Collection of raw data
* Feature Extraction
* Feature Selection
* Exploratory Data analysis
* Data Visualization
* Feature extraction
* Implementation of ML algorithms
* Tuning of hyperparameters
* Comparing errors different algorithms on test dataset

The data was provided by AlmaBetter which contained our target variable Price\_range and the independent variables such as ram, Int\_memory, Pc, Mobile\_wt etc. The feature selection and extraction were already done for the dataset.The data was also perfect in terms of missing or null values. So, there was no need of cleaning of the dataset in regard to missing values and outliers.

The heatmap represents the linear correlation between different variables that are present in our dataset. The correlation plot shows that there is a positive correlation present between fc, pc and four\_g, three\_g. Furthermore, it was seen that Battery power, clock\_speed, dual\_sim, m\_dep, mobile\_wt, px\_height, px\_width, ram, sc\_h and talk\_time got linear relationship with our dependent variable price range.

**Results and Discussion:**

The model was developed by using the linear regression, random forest, decision trees, XG boost and KNN algorithm. The training of these models was done on 70% of the data and testing was done on rest 30% of the data. The following tables shows the performance parameters of different algorithms on train and test dataset respectively. The performance parameters of the different algorithms for both training and testing are provided in table 2 and table 3 respectively.

**Table 1: Performance parameters of different algorithms for the training dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Class label | Performance parameters | | | |
| Accuracy | Precision | Recall | F1 score |
| Linear regression | 0 | 0.9175 | 1.0000 | 0.8842 | 0.9385 |
| 1 | 0.8558 | 0.9674 | 0.9082 |
| 2 | 0.8407 | 0.9596 | 0.8962 |
| 3 | 1.0000 | 0.8684 | 0.9296 |
| Random forest | 0 | 0.9485 | 0.9630 | 0.9734 | 0.9671 |
| 1 | 0.9071 | 0.9095 | 0.9083 |
| 2 | 0.9385 | 0.9197 | 0.9290 |
| 3 | 0.9882 | 0.9970 | 0.9926 |
| Decision trees | 0 | 0.8921 | 0.9266 | 0.9970 | 0.9235 |
| 1 | 0.7968 | 0.7555 | 0.7756 |
| 2 | 0.7517 | 0.7019 | 0.7260 |
| 3 | 0.8287 | 0.9202 | 0.8720 |
| XG boost | 0 | 0.9200 | 0.9662 | 0.9470 | 0.9565 |
| 1 | 0.8726 | 0.9383 | 0.9043 |
| 2 | 0.9065 | 0.8513 | 0.8780 |
| 3 | 0.9359 | 0.9419 | 0.9389 |
| KNN | 0 | 0.9433 | 0.9677 | 0.9934 | 0.9804 |
| 1 | 0.9403 | 0.9333 | 0.9368 |
| 2 | 0.9127 | 0.9007 | 0.9067 |
| 3 | 0.9506 | 0.9477 | 0.9477 |

**Table 2: Performance parameters of different algorithms for the test dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Class label | Performance parameters | | | |
| Accuracy | Precision | Recall | F1 score |
| Linear regression | 0 | 0.9175 | 1.0000 | 0.8842 | 0.9385 |
| 1 | 0.8558 | 0.9674 | 0.9082 |
| 2 | 0.8407 | 0.9596 | 0.8962 |
| 3 | 1.0000 | 0.8684 | 0.9296 |
| Random forest | 0 | 0.9686 | 0.9801 | 0.9470 | 0.9502 |
| 1 | 0.8800 | 0.9041 | 0.8919 |
| 2 | 0.8523 | 0.8581 | 0.8552 |
| 3 | 0.9271 | 0.9032 | 0.9150 |
| Decision trees | 0 | 0.8921 | 0.9316 | 0.9369 | 0.9342 |
| 1 | 0.8793 | 0.8383 | 0.8583 |
| 2 | 0.8588 | 0.8366 | 0.8476 |
| 3 | 0.8975 | 0.9614 | 0.9283 |
| XG boost | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1 | 1.0000 | 1.0000 | 1.0000 |
| 2 | 1.0000 | 1.0000 | 1.0000 |
| 3 | 1.0000 | 1.0000 | 1.0000 |
| KNN | 0 | 0.9450 | 0.9687 | 0.9770 | 0.9728 |
| 1 | 0.9274 | 0.9452 | 0.9362 |
| 2 | 0.9193 | 0.9140 | 0.9166 |
| 3 | 0.9665 | 0.9436 | 0.9549 |

The KNN model seems to perform the best while predicting the output as the performance parameters for both the train and test set are consistent for all the classes of the price range. The XG boost seems to overfit the data.

**Conclusion:**

Different features like Battery power, clock speed, dual sim, mobile depth, mobile weight, pixel height, pixel width, ram, secondary camera, talk time got linear relationship with our dependent variable price range. It was seen that ram has the highest impact on the price of the mobile. Surprisingly, Linear regression performed well in this classification problem. But there was heteroscedasticity present, hence non-linear models were preferred than the linear ones. Using decision tree, decent performance was gained after tuning the hyperparameters. It was also found that there's some overfitting which is a usual problem with the decision tree. Also, different bagging and boosting models were tested which performed well as compared to the above models. The XG boost model was also getting overfitted. The KNN model performed the best and most consistently while predicting the different classes of the output. Hence, it was concluded that the KNN model was the best among all.