Mobile Price Range Prediction

Rajat Mishra
Data science trainee,
AlmaBetter, Bangalore

Abstract:

The mobile price range prediction project aims to develop a machine learning model that can accurately predict the price range of mobile phones based on their features. The project will involve data collection and preprocessing, feature engineering, model selection, and evaluation. The dataset will consist of mobile phone features such as brand, screen size, camera resolution, battery capacity, and RAM. Various machine learning algorithms such as logistic regression, XGboost, and random forests will be applied to the dataset to build the prediction model. The accuracy of the model will be evaluated using metrics such as mean absolute error, mean squared error, and R-squared. The project aims to help customers make informed decisions when purchasing mobile phones by providing them with accurate price range predictions based on phone features.

1. Problem Statement

In today's fiercely competitive mobile phone market, understanding sales data and identifying key factors that influence prices is critical for companies to stay ahead of the competition. To achieve this goal, companies can leverage advanced analytics techniques to identify the relationship between various features of a mobile phone, such as RAM, internal memory, camera quality, processor speed, and battery life, and their respective selling prices.

The primary objective of this analysis is not to predict the exact selling price of a mobile phone, but to identify a range that indicates how high or low the price might be based on its features. By using machine learning algorithms, companies can train models on a large dataset of mobile phones, including their features and selling prices, to identify patterns and correlations between different variables.

Once the model is trained, it can be used to predict the price range of a new mobile phone based on its features. This information can help manufacturers and retailers in several ways, such as optimizing pricing strategies, identifying product gaps, and making informed decisions about which features to prioritize in their product offerings.

Overall, by analyzing the relationship between mobile phone features and selling prices, companies can gain valuable insights that can help them stay competitive in the market and better serve their customers' needs.

2. Introduction

For a mobile price range prediction project, the goal is to build a predictive model that can help users estimate the price range of a mobile phone based on its features and specifications.

The model would take into account various factors such as the brand, model, screen size, camera quality, storage capacity, processor type, and other features that affect the price of a mobile phone. The model would analyze these features and use machine learning algorithms to predict the price range of the mobile phone.

Similar to the surge pricing algorithm, the mobile price range prediction model would also be real-time and dynamic, adjusting the predicted price range based on the latest market trends and demand. The output of the model would be a predicted price range for a specific mobile phone model, which could be communicated to the user.

The model would be trained on a dataset containing various mobile phone models and their corresponding prices, along with their features and specifications. This dataset would be used to train the model using supervised learning algorithms, such as linear regression, decision trees, or neural networks, to make accurate predictions of mobile phone price ranges.

Once the model is trained and validated, it can be deployed as an application or integrated into an existing mobile phone e-commerce platform, allowing users to estimate the price range of a mobile phone before making a purchase decision.

3.Types of Pricing

In the context of mobile price range prediction, companies may use static pricing or dynamic pricing strategies to set their prices.

Static pricing is a fixed pricing scheme that remains the same regardless of fluctuations in demand or supply. This pricing strategy is often used in traditional retail settings where the cost of goods sold is relatively stable.

Dynamic pricing, on the other hand, is a flexible pricing strategy that adjusts prices in real-time based on changes in supply and demand. This strategy is becoming increasingly popular in the mobile phone market as companies seek to optimize revenue and meet the needs of customers in real-time.

Similar to surge pricing in the taxi industry, dynamic pricing in mobile phone sales adjusts prices based on fluctuations in demand, supply, and other external factors such as promotions, competitors' pricing, and consumer behavior. By leveraging data and analytics to track market trends, companies can make strategic pricing decisions and optimize revenue.

Overall, understanding the pros and cons of static and dynamic pricing can help companies develop effective pricing strategies and stay competitive in a rapidly evolving market.

4. Mobile Price Drivers.

In the context of mobile price range prediction, there are several factors that can contribute to price fluctuations. These include:

Brand reputation: Established brands with a strong reputation for quality and innovation may command higher prices than lesser-known brands.

Features and specifications: The features and specifications of a mobile phone, such as screen size, camera quality, processor speed, battery life, and internal memory, can all influence the selling price.

Target audience: The target audience for a particular mobile phone can also impact its price range. For example, a phone designed for business professionals may be priced differently than a phone marketed towards students.

Market demand: Market demand for a particular mobile phone can fluctuate based on a variety of factors, such as consumer preferences, seasonal trends, and global events.

Competitor pricing: The pricing strategies of competitors can also impact the price range of a mobile phone. For example, if a competitor lowers their prices, it may force other companies to adjust their prices in response.

Overall, understanding the various factors that contribute to mobile phone prices can help companies make informed decisions about their product offerings and pricing strategies, and ultimately better serve the needs of their customers.

5. How Mobile pricing works

1.Managing Mobile Price Surges

In the context of mobile price range prediction, sudden changes in demand can also impact pricing. When demand for a particular mobile phone model suddenly increases, there may not be enough supply to meet the demand. This can lead to a surge in prices as consumers compete to purchase the limited supply of phones.

Factors that can contribute to sudden changes in demand for a mobile phone model may include the release of a highly anticipated new model, positive reviews from influential sources, or a sudden change in consumer preferences.

To mitigate the impact of sudden demand surges, companies may implement strategies such as limiting the number of phones sold per customer or increasing production capacity to meet the increased demand. Understanding demand patterns and being able to anticipate changes in consumer preferences can help companies better manage their pricing strategies and optimize their supply chain operations.

Dynamic Pricing in Mobile Sale

In the context of mobile price range prediction, when there is a surge in demand for a particular mobile phone model, prices may increase to ensure that supply is available to those who are willing to pay a premium price.

This dynamic pricing system is designed to balance supply and demand in real-time, and it allows companies to optimize revenue while still meeting the needs of customers. When prices are raised due to a surge in demand, customers are typically notified via marketing campaigns, advertisements, or email newsletters.

Some customers may choose to pay the premium price to ensure they get the phone they want, while others may choose to wait until prices come back down. Companies may also choose to offer discounts or incentives to encourage customers to purchase at a higher price point, or they may adjust production and supply to better meet demand.

Overall, understanding the dynamics of surge pricing and customer behavior can help companies optimize their pricing strategies, improve customer satisfaction, and drive revenue growth.

6. Steps involved:

- **Exploratory Data Analysis**
  After loading the dataset we performed this method by comparing our target variable that is Surge_Pricing_Type 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 contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.

- **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 ExtraTree classifier 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. **Random Forest Classifier**
3. **XGBoost classifier**

- **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models like Random Forest Classifier and XGBoost classifier.

- **SHAP Values for features**
  We have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn't put much weight on the performance of our model.

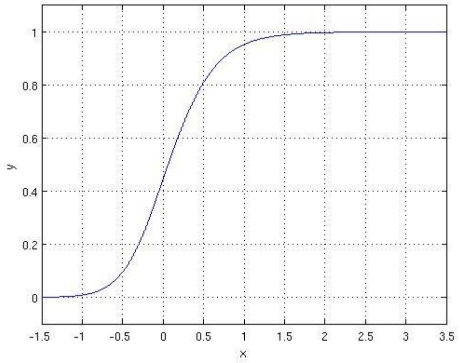
Algorithms:

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)

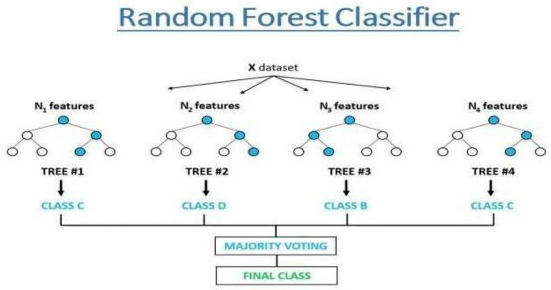


The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:

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

2. Random Forest Classifier:

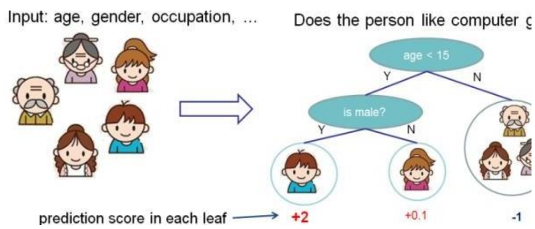
Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of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



3. XGBoost-

To understand XGBoost we have to know gradient boosting beforehand.

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


Input: age, gender, occupation, ...
Does the person like computer g
age < 15
Y
N
is male?
Y
N
prediction score in each leaf
+2
+0.1
-1

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 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.

XGBoost

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). XGBoost 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.

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 modelis 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:

Hyperparameters 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 hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

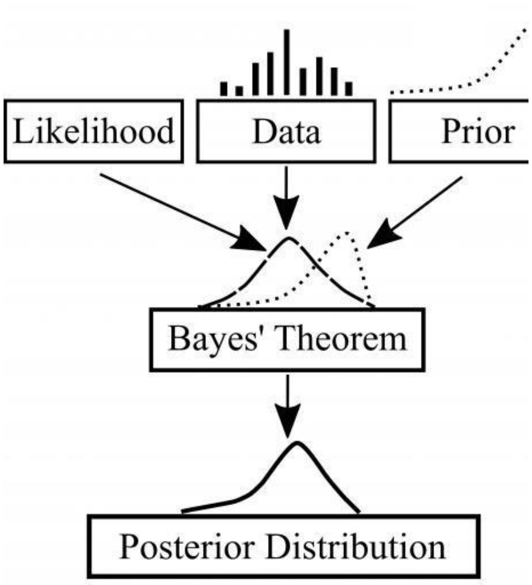
We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV-**Grid Search combines a selection of hyperparameters 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 hyperparameters 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 hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control

3. **Bayesian Optimization-** Bayesian Hyperparameter optimization is a very efficient and interesting way to find good hyperparameters. In this approach, in naive interpretation way is to use a support model to find the best hyperparameters.A hyperparameter 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 hyperparameters.



As it is a Bayesian process at each iteration, the distribution of the model's performance in relation to the hyperparameters 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.

8. Conclusion:

Based on the evaluation metrics provided for each model, it seems like the XGBoost model performed better than the Logistic Regression and Random Forest models. The XGBoost model achieved a very high accuracy score, precision, recall, and F1-score for all classes, which indicates that the model is performing very well on the training set and able to generalize well to all classes.

On the other hand, the Logistic Regression model achieved a decent accuracy score and precision for class 0, but its recall and F1-score for class 0 were slightly lower compared to the XGBoost model. Similarly, the Random Forest model had moderate performance, with accuracy, precision, recall, and F1-score ranging from 0.63 to 0.92 depending on the class being predicted.

References-

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya
4. Flowing Data
5. KDnuggets