Mobile Price Range Prediction

By Komal Priyadarshini

Data Science Trainee

AlmaBetter, Bangalore

Abstract:

Various features like memory, display, battery, camera, etc., are considered when purchasing mobile phones. People fail to make correct decisions, due to the non-availability of necessary resources to cross-validate the price. To address this issue, a machine learning model is developed using the data related to the mobile phone's key features. The developed model is then used to predict the price range of the new mobile phone. Three machine learning algorithms namely Support Vector Machine (SVM), Random Forest Classifier (RFC), and Logistic Regression are used to train the model and predict the output as low, medium, high or very high. The dataset used in this study is taken from Kaggle platform. In order to improve the classification accuracy, Chi-Squared based feature selection method is used. Among 21 features available in the dataset, only top 10 features namely RAM, pixel height, battery power, pixel width, mobile weight, internal memory, screen width, talk time, front camera and screen height are selected and used to train the model. Before applying feature selection, the accuracy obtained using SVM, RFC and Logistic Regression is 95%, 83% and 76% respectively. After feature selection, the accuracy of SVM, RFC and Logistic Regression improved to 97%, 87% and 81% respectively. From the experiments conducted, it is found that SVM gave superior performance when compared to other two classifiers.

Keywords: EDA, Correlation, XGBoost, Random Forest, MultiClass Classification

Problem Statement:

In the competitive mobile phone market Companies want to understand sales data of mobile phones a ndfactors which drive the prices.
The objective is to find out some relation be tween features of a mobile phone (eg:- RAM, Internal Memory, etc) and its selling price. In thi s problem, we do not have to predict the actual price but a price range indicating how hig h the price is.

Introduction:

Price is the marketing and business attribute which is the most powerful. The costumer's very first question is about the price of the things. All the customers are worried first and wonder "whether he can buy something with the requirements given or not."So the basic aim of the research is to estimate price at home. This paper represents only the first step towards the destination described above. Artificial intelligence — which makes the computer intelligently capable of answering the questions — is now a very large field of engineering. Machine learning provides us with the latest artificial intelligence methods, such as classification, regression, supervised learning and unsupervised learning and much more. Different machine learning tools are available, such as MATLAB, Python, cygwin, WEKA etc. We may use one of Decision tree, Naïve Bayes and several other classifiers. Different types of algorithms are necessary for selecting only the best features and reducing the dataset. That will lower the problem's computational complexity. Because this is the problem of optimization, several optimization techniques are often used to reduce the dataset's dimensionality. Mobile now a day is one of the apps with the most sales and purchases. New mobiles are released each day with new version and more apps. Hundreds and thousands of cell phones are sold and bought every day. So here the prediction of the mobile price class is a case study for the given type of problem i.e. finding an optimal product. The same work can be done to estimate the real price of all goods, such as vehicles, motorcycles, generators, engines, food items, medication, etc. Several apps are very important for estimating mobile prices, for example Mobile Processor. In today's busy human life the timing of batteries is also very critical. Mobile size and thickness are also important determinants of decision. Internal memory, camera pixels and the consistency of the video must be remembered. Internet surfing is also one of the most significant limitations of this 21st-century technological period. And so is the list of several features dependent on those, it is decided on mobile size. So we're going to use all of the above features to determine whether the smart-phone will be very-economic, economical, expensive or very costly.

In this problem statement, we will be trying to create machine learning models that characterize and predict In addition to this, we will be trying to analyze and find all the features that are important for an email to not get ignored and based on that some recommendations are made.

Approach:

The approach followed here is to first check thesanctity of the data and then understand the features involved. The events followed were in our approach:

- **Understanding the Data**
- **Data cleaning and preprocessing-** finding null values and imputing themwith appropriate values.
- **Exploratory data analysis-** of categorical and continuous variablesagainst our target variable.
- **Data manipulation-** feature selectionand engineering, handling multicollinearity with the help of VIF scores, feature scaling and encoding.
- **Handling Class Imbalance-** our datasetwas highly imbalance with 80% majority, strategy was to splitting the stratified dataset and under sampling andoversampling with SMOTE on the train sets only so that our test set remains unknown to the models
- **Modeling-** worked on an evaluation code which was frequently used to evaluate the same models on under sampled and oversampled data in one go, logistic regression, random forest, and XGB were run to evaluate the results and then concluded on the basis ofmodel performance and some recommendations were made to improve the numbers of read and acknowledged emails.

Understanding the Data:

First step involved is understanding the data and getting answers to some basic questions like; What is the data about? How many rows or observations are there in it? How many features are there in it? What are the data types? Are there any missing values? And anything that could be relevant and useful to our investigation. Let's just understand the dataset first and the terms involved before proceeding further.
Our dataset consists of 2000 observations (i.e. rows) and 21 features (columns) about the emails. The data types were of integer, float and object innature.

Let's define the features involved:

- **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
- **Touch_screen** - Has touch screen or not
- **Wi-Fi** - Has Wi-Fi or not
- **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
- **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).

Data Cleaning and Preprocessing:

Handling missing values is an important skill inthe data analysis process. If there are very few missing values compared to the size of the dataset, we may choose to drop rows that have missing values. Otherwise, it is better to replacethem with appropriate values.
It is necessary to check and handle these values before feeding it to the models, so as to obtain good insights on what the data is trying to say and make great characterization and prediction which will in turn help improve the business's content.

The dataset had no nulls values.
For the rest of the continuous variables, the distribution was plotted to get an idea on how to impute these variables.
Here's the distribution plot of Total Past Communications:

The distribution plot of both Total Links and Total Images are skewed to the right. Right skewed distributions occur when the long tail is on the right side of the distribution also called aspositive skewed distribution which essentially suggests that there are positive outliers far alongwhich influences the mean.
It seems like most of the values of the Total Links in the column are between 0-10 and the number of images in most of the emails seems to be 0 or fewer than 3-4. Consequently, the longer tail in an asymmetrical distribution pulls the mean away from the most common values. The mean is greater than the median. The mean overestimates the most common values in the distribution and hence mode (value with highest frequency) is used in these cases, it is more robust to outlier effect and the same is done here.

Exploratory Data Analysis:

Exploratory data analysis is a crucial part of data analysis. It involves exploring and analyzing the dataset given to find out patterns, trends and conclusions to make better decisions related to the data, often using statistical graphics and other data visualization tools to summarize the results. The visualization tools involved in our investigation are python libraries- matplotlib and seaborn.
The goal here is to explore the relationships of different variables to see what factors might be contributing and then be able to correctly characterize them.

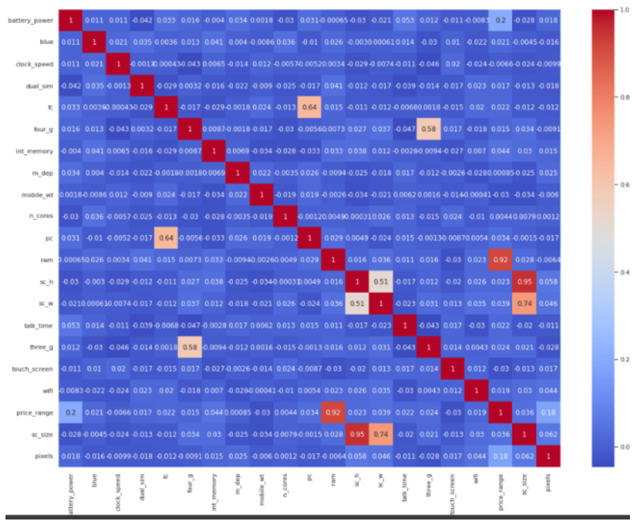
Approach:

There are two kinds of features in the dataset: Categorical and Non Categorical Variables.
Categorical- A categorical variable is a variable that can take on one of a limited, and usually fixed, number of possible values putting aa particular category to the observation.
Non Categorical- A non categorical or continuous variable is a variable whose value isobtained by measuring, i.e., one which can takeon an uncountable set of values.
Both of them are analyzed separately.
Categorical data is usually analyzed through

count plots in accordance with the target variable and that is what is done here too. On the other hand Numeric or Continuous variables were analyzed through distributionplots and box plots to get useful insights.
customers.

Correlation:

Correlation is a statistical term used to measure the degree in which two variables move in relation to each other. A perfect positive correlation means that the correlation coefficientis exactly 1. This implies that as one variable moves, either up or down, the other moves in thesame direction. A perfect negative correlation means that two variables move in opposite directions, while a zero correlation implies no linear relationship at all.



A correlation matrix is a table showing correlatio n coefficients between variables. Each cell in the t able shows the correlation between two variables. A correlation matrix is used to summarize data, a s an input into a more advanced analysis, and as a diagnostic for advanced analyses. The range of c orrelation is [-1,1].

Thus to know the correlation between all the varia bles along with the correlation coeficients, i used correlation heatmap.

RAM and price_range shows high correlation which i s a good sign, it signifies that RAM will play major deciding factor in estimat ing the price range.

There is some collinearity in feature pairs ('pc', 'fc') and ('px_width', 'px_height'). Both correlations a re justified since there are good chances that if fro nt camera of a phone is good, the back camera wo uld also be good.

Also, if px_height increases, pixel width also incr eases, that means the overall pixels in the screen. We can replace these two features with one featur e.
Front Camera megapixels and Primary camera me gapixels are different entities despite of showing c olinearity. So we'll be keeping them as they are.

Data Manipulation:

Data manipulation involves manipulating and changing our dataset before feeding it to various classification machine learning models. This involves keeping important features handling multicollinearity in the dataset, outlier treatmentand creating dummy variables if necessary.

Multicollinearity:

Multicollinearity occurs when two or more independent continuous variables are highly correlated with one another in classification or regression models. This means that an independent variable can be predicted by another independent variable and they both alsopredict the dependent variable. Multicollinearitymakes it harder for the models to interpret the coefficients of individual variables or the role ofthem in predicting and hence in turn can exaggerate their roles and misclassify sometimes as well.
We can quantify multicollinearity using Variance Inflation Factors (VIF).
VIF determines the strength of the correlation between the independent variables. It is predicted by taking a variable and regressing itagainst every other variable. VIF score of an independent variable represents how well the variable is explained by other variables.

VIF = (1/(1-R^2))

R-squared is a statistical measure of how close the data are to the fitted regression line. It is alsoknown as the coefficient of determination, or thecoefficient of multiple determination for multiple regression. In general, the higher the R-squared, the better the model fits your data.Themore the value of R^2 is closer to 1 the more, VIF score tends to infinity. VIF starts with 1 and denotes that the variable has no correlation at all. VIF more than 5-10 can be considered as a serious case of multicollinearity and can affect prediction models.

The VIF results showed a high value for Total links and upon creating a scatter plot of Total Links and Total Images, it gave a linear relationship with some outliers, the essential stepwould be to combine both of these up and then check the VIF scores again and it was under check.

Outliers:

With the help of box-plots, we earlier saw that besides Word Count all our other continuous variables have outliers, but deleting them wouldlead to loss of information as our target variableis highly imbalanced we need to make sure that we aren't deleting more than 5% of information or data related to the minority class.

It is more than 5% information of our minority class. This dataset already has a high class imbalance issue, and if deleted this much amount of information from minority class will lead to a lack of information issue for predictingmodels and hence these were not deleted. They are going to affect the models either way.

Feature Scaling:

.Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is done to prevent biased nature of machine learning algorithms towards featureswith greater values and scale. The two techniques are: Normalization: is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling. [0,1]

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization: is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. [-1,1]

$$X' = \frac{X - \mu}{\sigma}$$

Normalization of the continuous variables was done here.

One hot encoding:

For categorical variables where no such ordinal relationship exists, the integer encoding is not enough. We have categorical data integers encoded with us, but assuming a natural order andallowing this data to the model may result in poor performance. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form in order to present them or use them in some application.

Handling Class Imbalance:

In the exploratory data analysis, we saw clearly that the number of emails being ignored was a lot more than being read and acknowledged. This imbalance in the class can lead to biased classification towards ignored emails.

Only 3% of observations are classified as acknowledged emails and 80% are ignored emails.

This bias in the training dataset can influence many machine learning algorithms, leading some to ignore the minority class entirely.One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called undersampling, and to duplicate examples fromthe minority class, called oversampling. This project involves both of these techniquesand compares the end result.

- Random undersampling deletes examples from the majority class andcan result in losing information invaluable to a model.
- Oversampling is achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. One technique for this isSynthetic Minority Oversampling Technique, or SMOTE for short. Specifically, a random example from theminority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the twoexamples in feature space.

The train test split was done before applying any resampling technique so that the test-datasetremains unknown to the models. Resampling of the train dataset was first done by Random undersampling and then by SMOTE.

Before balancing, it was made sure the train split has class distribution as same as the main dataset by using stratify while splitting.The strategy here is to develop a model evaluation function which takes in both undersampled and oversampled data to evaluateand predict results and visualize model evaluation metrics for both of them.

Modeling:

Logistic Regression: Logistic Regression is a classification algorithm that predicts the probability of an outcome that can have only twovalues. Multinomial logistic regression is an extension of logistic regression that adds native support for multi-class classification problems.

Instead, the multinomial logistic regression algorithm is a model that involves changing the loss function to cross-entropy loss and predicting probability distribution to a multinomial probability distribution to natively support multi-class classification problems.

Random Forest: Random forests are an ensemble learning method for classification andregression that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

To prevent overfitting, a random forest model was built. Random forest builds multiple decision trees and merges them together to get amore accurate and stable prediction.

We got better results with SMOTE and decided to get a hyperparameter tuned model as well andthen the tuned SMOTE version gave the best results till now with a good F1 score and AUC ROC.

```
[{'Model_Name': 'Random Forest RUS',
  'Test_AUC': 0.7634729938199679,
  'Test_Accuracy': 0.6280447662936142,
  'Test_F1score': 0.6841713877588803,
  'Test_Precision': 0.7754233629597981,
  'Test_Recall': 0.6280447662936142,
  'Train_AUC': 0.7537853426026989,
  'Train_Accuracy': 0.565331928345627,
  'Train_F1score': 0.5427419799819592,
  'Train_Precision': 0.5595483093444947,
  'Train_Recall': 0.565331928345627},
 {'Model_Name': 'Random Forest SMOTE',
  'Test_AUC': 0.7634355208741295,
  'Test_Accuracy': 0.6804184039207081,
  'Test_F1score': 0.7154587562726483,
  'Test_Precision': 0.7705208259350318,
  'Test_Recall': 0.6804184039207081,
  'Train_AUC': 0.7580678479928142,
  'Train_Accuracy': 0.5571709174193646,
  'Train_F1score': 0.5270095256083359,
  'Train_Precision': 0.5376834641549533,
  'Train_Recall': 0.5571709174193646}]
```

Random Forest Hyperparameter Tuned:

```
[{'Model_Name': 'RandomF Tuned RUS',
  'Test_AUC': 0.7576589665070962,
  'Test_Accuracy': 0.6148050618096701,
  'Test_F1score': 0.6762096575189265,
  'Test_Precision': 0.779784883659752,
  'Test_Recall': 0.6148050618096701,
  'Train_AUC': 0.9134095361504893,
  'Train_Accuracy': 0.7448191078328065,
  'Train_F1score': 0.743932385070194,
  'Train_Precision': 0.748197339042421,
  'Train_Recall': 0.7448191078328065},
 {'Model_Name': 'RandomF Tuned SMOTE',
  'Test_AUC': 0.7573239964478374,
  'Test_Accuracy': 0.7427401067954064,
  'Test_F1score': 0.7508011216581711,
  'Test_Precision': 0.7596556084261982,
  'Test_Recall': 0.7427401067954064,
  'Train_AUC': 0.9835795130710663,
  'Train_Accuracy': 0.9063924343427449,
  'Train_F1score': 0.9057915850497917,
  'Train_Precision': 0.9063248715670915,
  'Train_Recall': 0.9063924343427449}]
```

XG Boost: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. The two reasons to use XGBoost are also the two goals of the project:

- Execution Speed.
- Model Performance.

Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made.

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

XGB SMOTE gave the best results till now, with good Test Recall, F1 score and AUC ROC.

```
[{'Model_Name': 'KNN Tuned RUS',
  'Test_AUC': 0.7120568798168657,
  'Test_Accuracy': 0.5780118499012509,
  'Test_F1score': 0.645603496972498,
  'Test_Precision': 0.7695832996291636,
  'Test_Recall': 0.5780118499012509,
  'Train_AUC': 0.8181196963657232,
  'Train_Accuracy': 0.6243414120126449,
  'Train_F1score': 0.623237047408161,
  'Train_Precision': 0.6234362654846187,
  'Train_Recall': 0.6243414120126449},
 {'Model_Name': 'KNN Tuned SMOTE',
  'Test_AUC': 0.6820543345467238,
  'Test_Accuracy': 0.627313290907761,
  'Test_F1score': 0.6769157196720779,
  'Test_Precision': 0.754417249103387,
  'Test_Recall': 0.627313290907761,
  'Train_AUC': 0.9791208368560474,
  'Train_Accuracy': 0.880819663428359,
  'Train_F1score': 0.8791915206050119,
  'Train_Precision': 0.8846573457260328,
  'Train_Recall': 0.880819663428359}]
```

```
[{'Model_Name': 'XGB RUS',
  'Test_AUC': 0.7317300414945984,
  'Test_Accuracy': 0.5676980469607198,
  'Test_F1score': 0.6396494731848822,
  'Test_Precision': 0.7711408031941839,
  'Test_Recall': 0.5676980469607198,
  'Train_AUC': 0.9995195338076092,
  'Train_Accuracy': 0.9866526167896031,
  'Train_F1score': 0.986648578883858,
  'Train_Precision': 0.9867723820780556,
  'Train_Recall': 0.9866526167896031},
 {'Model_Name': 'XGB SMOTE',
  'Test_AUC': 0.7632622116922312,
  'Test_Accuracy': 0.789920269182942,
  'Test_F1score': 0.7623546221491548,
  'Test_Precision': 0.7474644778985376,
  'Test_Recall': 0.789920269182942,
  'Train_AUC': 0.9841138760763947,
  'Train_Accuracy': 0.9117542223132286,
  'Train_F1score': 0.910068254199571,
  'Train_Precision': 0.914848981706282,
  'Train_Recall': 0.9117542223132286}]
```

Conclusion:

Challenges:

- We had a highly imbalanced target variable with 80% data of the majority class, 16% and 4% data of the minority classes respectively.
- The second challenge we faced was of outliers. Almost all the continuous variables had a good number of outliers. Upon calculating we came to know of the fact that more than 5% of outliers were in minority classes itself.
- This made the classifiers a bit confused, when there was a low number of data observations related to minority class and with outliers.
- The last challenge was to resample the unbalanced data. Many a times we see resampling on the whole dataset and then splitting it into train test set so that we don't have an unbalanced validation set too but it may create a bias in the models and they might cheat towards synthetically created data in the validation set, so in this project resampling was done only on the training set and validation set was kept unknown to the models to predict on, which obviously gave lower results than the other way. Both ways were tried.

Evaluation Metrics:

There are a number of model evaluation metricsto choose from but since our dataset was highlyimbalanced, it is critical to understand which metric should be evaluated to understand the model performance.

Accuracy- Accuracy simply measures how oftenthe classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions. Accuracy is useful when the target class is well balanced but is not a good choice for the unbalanced classes, because if the model poorly predicts every observation as of the majority class, we are going to get a pretty high accuracy.

Confusion Matrix - It is a performance measurement criteria for the machine learning classification problems where we get a table with a combination of predicted and actual values.

Precision - Precision for a label is defined as the number of true positives divided by the number of predicted positives.

Recall - Recall for a label is defined as the number of true positives divided by the total number of actual positives. Recall explains how many of the actual positive cases we were able to Predict correctly with our model.

F1 Score - It's actually the harmonic mean of Precision and Recall. It is maximum when Precision is equal to Recall.

References:

- Machine Learning Mastery
- GeeksforGeeks
- Analytics Vidhya Blogs
- Towards Data Science Blogs
- Built in Data Science Blogs
- Statistics by Jim