**Mobile Price Prediction Using LDA, QDA, KNN, and Random Forest**

Avanti Likhite, Ye Xu, Jingnan Yang, Weihao Yuan (Melvin)

**Introduction**

For any cell phone manufacturer, initiating a pricing strategy plays a crucial role because the market is price sensitive. While overpricing discourages the market demand, underpricing limits the potential revenue growth and undermines the brand image. Thus, striking the right balance between profit and the attractive/competitive price at the starting point becomes very critical while launching a new product. While a cell phone can have a variety of components that can together drive its price, typically a few factors can influence much more than others. In this research, we will study 14 such features, included Processor, RAM, Camera, etc., to analyze how varying those can impact the product price. The purpose of this case study is to empower the cell phone manufacturers in deciding the price range for their products.

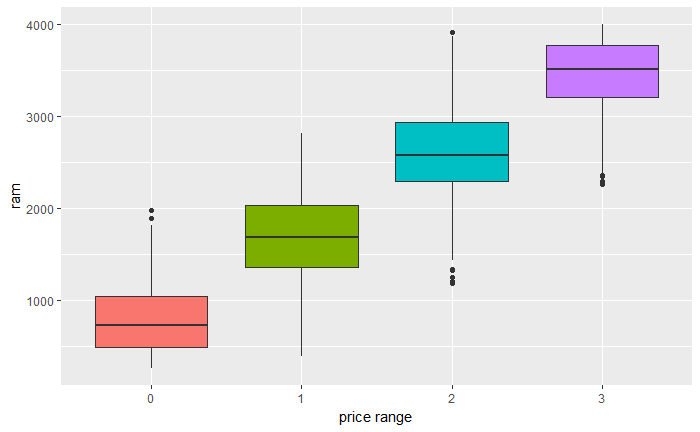
For the purpose of this study, we have segregated the phones into 4 market segments based on the price. There are several qualitative prediction approaches that can be used to determine the price range, including Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), and Classification Tree. Our study will use these four methods to create different pricing models and choose the one that can help cell phone manufacturers make the best decision on their pricing strategy.

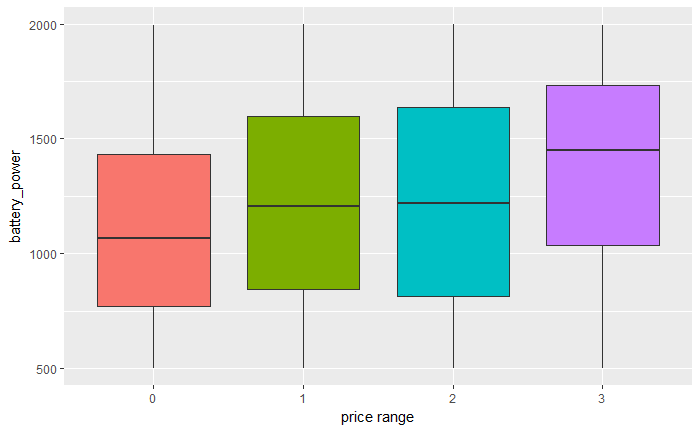
**Dataset**

In this study, we use the marketing research data collected by Kaggle. This dataset contains information on 2,000 cell phones in the marketplace, including product specification, features, and price range (market segments). More specifically, the data tabulation offers information of cell phones including battery power, Bluetooth, clock speed, dual sim support feature, front camera megapixels, primary camera megapixels, 4G, 3G, internal memory in gigabytes, mobile depth and width in cm, weight of mobile phone, number of cores of processor, pixel resolution height and width, random access memory in megabytes (RAM), screen height and width of mobile in cm, length of talk time, touch screen feature, wifi, and price range.

Among the twenty predictive variables, six of them are categorical variables, (Bluetooth, dual sim support feature, 4G, 3G, touch screen feature, and wifi), and the remaining fourteen are continuous variables. For the categorical variables, most of the observations are evenly distributed, except front camera megapixels, pixel resolution height and screen width are right-skewed. Similarly, most of the continuous variables are balanced in classes.

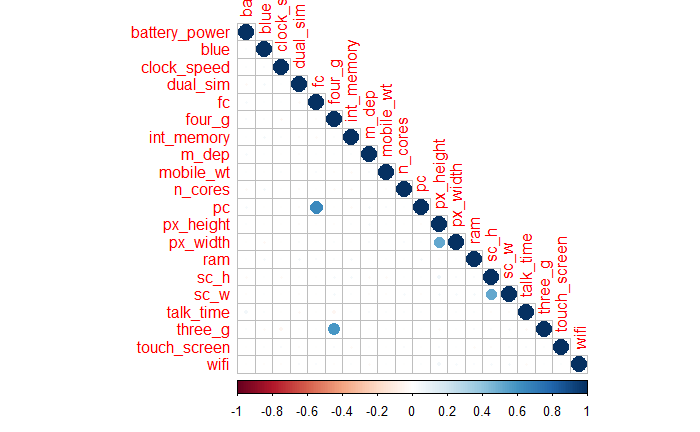
The dependent variable price range is categorical in nature and has 4 levels: discount, medium, high-end, and premium. Each class contains 500 observations. Our data is balanced that is we have an even distribution of our dependent variable across all its levels. After conducting the exploratory data analysis, it appears that the specification of RAM and battery power are the most important factors that distinguish the market segments between different cell phones.





(Figure 1. Above:: The price range and RAM of sample observations. Below: The price range and battery power of sample observations)

There are no obvious correlations (correlations that higher than 0.7 or lower than -0.7) between each two of the independent variables. Though correlations between the primary camera (pc) and the front camera (fc), 3G and 4G are relatively high (within -0.4 and -0.6), they should not cause severe multicollinearity problems.



(Figure 2. Correlation Heat Map)

In order to better test the prediction accuracy and prevent potential overfitting issue, we partition the whole dataset (2,000 observations) into a training set (1,600 observation, 80% of total dataset), and a test set (400 observations, 20% of total dataset).

**Methodology**

Since we are trying to categorize our potential productions into various market segments, we are using four different predictive methods to create different classification models, including linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbor (KNN), and classification tree. To compare and select the final model, we implement 10-folds cross-validation resampling method to find the model that generates the lowest mean misclassification rate.

1. **Resampling**

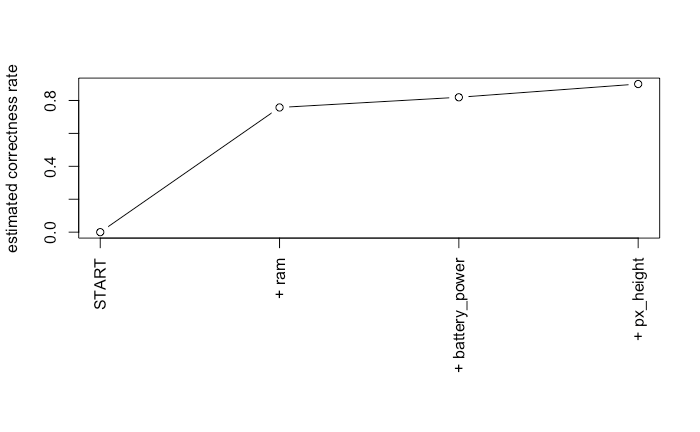
As the best model should not overfit the training data, we use the resampling method to find the model that yields the lowest cross-validation testing error, instead of a low training error. There are 3 resampling methods: validation, cross-validation, and bootstrap. The validation set approach randomly divide the available set of observations into two parts, a training set, and a validation set. Cross-validation method includes k-fold cross-validation and leave-one-out cross-validation (LOOCV). K-fold cross-validation is implemented by randomly divide the data into K equal sizes, then for each 1 to Kth dataset, train the model with all data except for the Kth dataset and use the Kth data sets for validation to get the MSE. At last, average all the MSE to get the mean cross-validation error. LOOCV is a special case of k-fold cross-validation in which K= n, in fact, it divides each observation into separate folds. Bootstrap creates new datasets by repeatedly sampling observations from the original data set with replacement. In our analysis, we use 10-fold cross validation because compared to the other methods, 10-fold cross validation has a lot of advantages. In general, k-fold cross validation has less bias compared with the validation set approach and is far less computational expensive compared with LOOCV and bootstrap. However, K-fold suffers from higher bias compared to LOOCV, although it has better variance as training data is reduced by 1/K. So, there is a bias-variance trade-off associated with the choice of K in K-fold cross-validation. Typically using K=5 or K=10 yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. In our analysis, we choose 10-fold cross validation for LDA, QDA, KNN and random forest models.

1. **Feature selection**

There are 20 variables in our dataset, but not all of them are useful to predict the price range. Thus, we use feature selection method to choose the most influential features and remove irrelevant ones, so that we can obtain a model that is accurate and interpretable. In specific, we choose forward stepwise selection, which is a very simple and logical approach. Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model. In particular, at each step, the variable that gives the greatest additional improvement to the t is added to the model. In the end, we build 20 models with different numbers of predictors and we need to select the best model among this collection of models. Because the best model is the one with smallest test error, not the smallest training error, we need to adopt a method to estimate the test error for each model. As we know, there are two methods: direct and indirect. As their names indicate, indirect method indirectly estimate a test error by making an adjustment to the training error to account for the bias due to overfitting, examples include Mallow’s Cp, AIC, BIC and Adjusted R2, while direct method directly estimate the test error, using either a validation set approach or a cross-validation approach. Because the direct method doesn’t require an estimate of the error variance, it’s more accurate and can be used in a wider range of model selection tasks. In our analysis, we use caret package to conduct a 10-fold cross-validation to pick a single model from all the 20 “best” models.

1. **Linear Discriminant Analysis (LDA)**

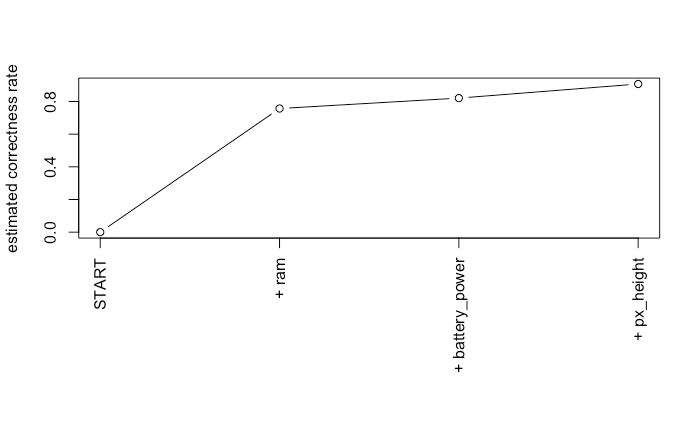
Linear discriminant analysis (LDA) is one of the most commonly used classification models when having more than two response classes. LDA treats a grouping variable as the response variable and categorizes it based on its relationship with a set of independent predictors. LDA assumes that observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all four classes in our case. In addition, the rule of thumb for using LDA is satisfied since the number of observations in our case is greater than five times the number of predictors.  
  
In this paper, we use forward stepwise model selection along with 10-folds cross-validation as dimension reduction tool that searches for a subset of predictors that yields the best prediction accuracy. As shown in Figure 4, the best model contains three predictors -- RAM, battery power, and pixel resolution height. This confirms our assumption in the dataset section. By using RAM as a single predictor, the LDA model can correctly categorize approximately 80% of the training response variables. By combining all three variables, the accurate prediction rate for the training set increases to 90.44%, as shown in the table below. In other words, nine out of ten products can be correctly categorized into the market segments they belong to.

****

(Figure 3: Average estimated correntness rate from the best LDA model identified by using forward stepwise model selection)

1. **Quadratic Discriminant Analysis (QDA)**

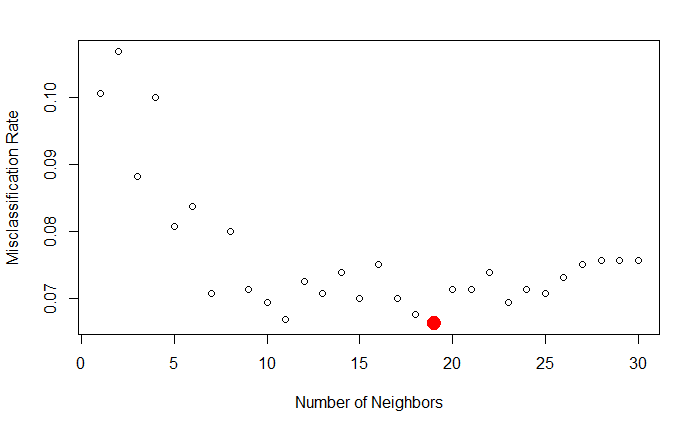
Quadratics Discriminate Analysis (QDA) is an extension of LDA. It also assumes that observations within each class are drawn from a multivariate Gaussian distribution, and plugs estimates for the parameters into Bayes’ theorem to perform prediction. But QDA assumes that each class has its own covariance matrix and the predictors are not strongly correlated. Similar to LDA, we use forward stepwise model selection along with 10-folds cross-validation as dimension reduction tool. QDA with stepwise feature selection method also yields a very similar model to that of LDA. The best QDA model also uses RAM, battery power, and pixel resolution height as predictors. The average accurate prediction rate for the training set is 90.94% which is slightly more accurate than that of the LDA model.

****

(Figure 4: Average estimated correntness rate from the best QDA model identified by using forward stepwise model selection)

1. **K-Nearest Neighbor**

KNN is an easy-used learning algorithm. It takes a completely different approach from the above classifiers. Neither does it make any assumption about the underlying data distribution, nor requires any training phase. KNN can also perform a just-in-time calculation. In order to make a prediction for an observation X = x, the K training observations that are closest to x are identified. Then X is assigned to the class to which the plurality of these observations belongs. We also use 10 fold cross-validation method to find the optimal K from 1 to 100 producing the lowest mean test error. Our findings show that using 19 nearest neighbors yields the lowest mean cross-validation error rate of 6.63%.



(Figure 5: Average estimated misclassification rate based on number of neighbors)

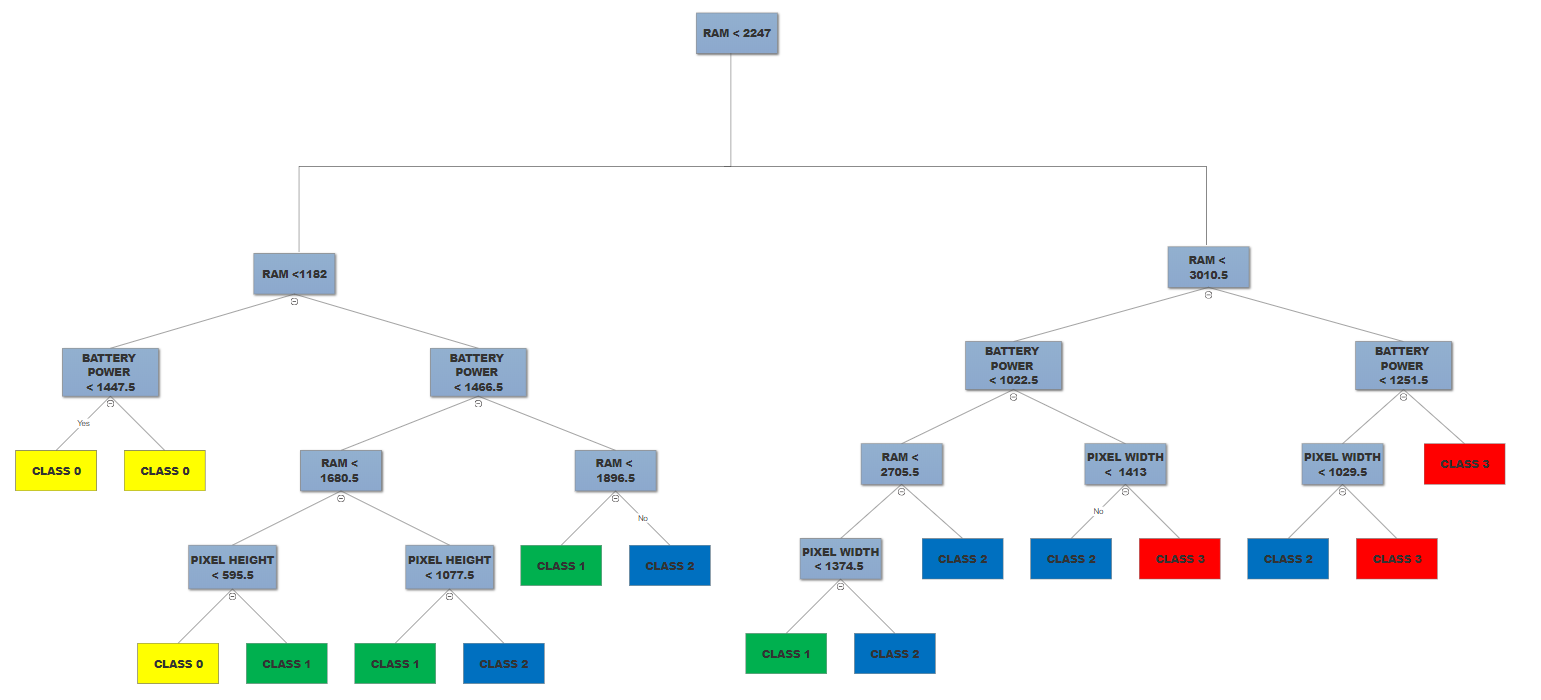
1. **DECISION TREE:**

Decision trees can be applied to both regression and classification problems. Here we focus on classification trees since our dependent variable is categorical in nature. The main advantages of using decision trees are:

* Classification trees are easy to interpret and are relatively robust to outliers and misclassified points in the training set.
* They require less data cleaning compared to some other modeling techniques.
* They do not calculate any average or anything else from the data points themselves.

However the most important drawback of decision trees is that while working with continuous numerical variables, information can be lost when decision trees categorize variables in different categories.

For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. Decision trees follow the recursive binary splitting for classification. Below is the classification tree that we obtained using our training data.



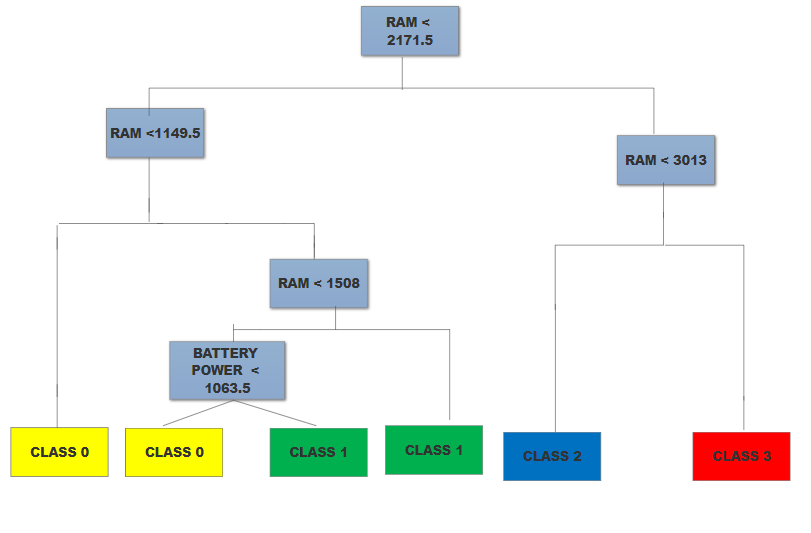
(Figure 6: Basic Decision Tree)

As shown in Figure 6, the decision tree is obtained without specifying the number of nodes. There are 16 terminal nodes and the we have used 4 features(RAM, battery Power, pixel width and pixel height) to construct the tree. We get high accuracy of 85% and misclassification error rate of 14.86% when we try to predict the price range for training data since it overfits the data. However, when we try to use the model to predict the price range for unknown data (in our case the test dataset) the accuracy is reduced to 80%.

We have carried out Cross Validation to determine the optimal size of the tree. Figure 7 below shows the result of our cross validation.

(Figure 7: Cross Validation Result for Choosing Size of The Tree)

From the above figure we can see that the optimal number of nodes in our pruned tree would be 3. By pruning we are trying to create a smaller tree that might have lower variance and which would be easier to interpret but might have higher bias.



(Figure 8: Pruned Tree)

As shown in Figure 8, a pruned decision tree is obtained with 6 terminal nodes and the we have used 2 features(RAM, battery Power) to construct the tree. We get accuracy of 76.5% and misclassification error rate of 23.5% when we try to predict the price range for training data since it overfits the data.

However, when we try to use the model to predict the price range for unknown data (in our case the test dataset) the accuracy is reduced to 75%

Bagging is a technique used to reduce the variance of our predictions by combining the result of multiple classifiers modeled on different sub-samples of the same data set. There are various implementations of bagging models and random forest is the main amongst them.

The main problem with decision trees is that they are greedy. They choose which variable to split on using a greedy algorithm that minimizes error. As such, even with Bagging, the decision trees can have a lot of structural similarities and in turn have high correlation in their predictions. Random forests reduces the variance when we average the trees.

In Random Forest, we grow multiple trees as opposed to a single tree in CART models. To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

Since we have a lot of variables but most of our models (LDA, QDA and KNN are suggesting that only a few are important), random forest can help us identify the most significant variables acting as a dimensionality reduction method. Further, the model outputs **Importance of variables**.

**Conclusion**

In this research, several tests have been performed using LDA, QDA, KNN, and Classification Trees to predict the price ranges for mobile phone. Our findings indicate that KNN is the best model for our case scenario since it yields the lowest mean cross-validation error of 6.63% for the training set. The testing misclassification rate for the KNN model is 7%, which indicates our model has a solid prediction power. Although we only use a small dataset with 1,600 observations to train our model, the final model can still yield a low misclassification rate without suffering from the overfitting problem.

Although we choose KNN as our final model, we are still able to identify two key features that have a strong influence on phone price -- RAM and battery power. From the Apple Battery Scandal, we can learn that cell phone users expect their phone to have high speed and long-lasting battery. They expect their phone to run as fast as possible even though the battery power is low and they also want to have long-lasting battery power for a longer usage time. According to Thomas, a technical writer at Gadget Hacks, “RAM is where bits of data are stored as they wait in line to be processed by your CPU”. It is the key for multitasking, “as recently-active apps are stored in your phone's RAM just in case you might want to switch back to them sometime soon.” (Thomas). Besides, battery power is another crucial factor since modern people are always with their phones. Therefore, we recommend the company to invest in Research and Development to develop a more powerful RAM and a more endurant battery in order to maintain the competency in the cell phone industry.

However, our research may also suffer from several potential limitations. Firstly, we do realize that brand awareness plays a very important role in determining the price of mobile phones, but our dataset does not have any field that can help us relate the prices to brand awareness. The prediction models overly concern about product features, does not pay enough attention to the customer and their preference. In addition, the dataset did not provide a clear definition of the price range. The ambiguity adds an additional difficulty in making a decision on the final pricing.

**Citation**

**Kaggle.** 2018. Mobile Phone Classification.

https://www.kaggle.com/iabhishekofficial/mobile-price-classification. (accessed June 10, 2018)

**Seutin, Alex, and Ian Jones.** 2016. Using Machine Learning to Predict Housing Prices Given Multivariate Input.

https://web.stanford.edu/class/cs221/2017/restricted/p-final/ianjones/final.pdf. (accessed June 10, 2018)

**Thomas, Dallas.** 2016. The Specs That Really Count When Buying a Phone.

https://smartphones.gadgethacks.com/how-to/specs-really-count-when-buying-phone-0171678/. (accessed June 10, 2018)

**Socialcops.** 2017. 6 Sampling Techniques: How to Choose a Representative Subset of the Population.

https://blog.socialcops.com/academy/resources/6-sampling-techniques-choose-representative-subset/. (accessed June 10, 2018)

**Appendix. Metadata**

id: ID (Numeric)

battery\_power: Total energy a battery can store in one time measured in mAh (Numeric)

blue: Has bluetooth or not (Boolean)

clock\_speed: speed at which microprocessor executes instructions (Numeric)

dual\_sim: Has dual sim support or not (Boolean)

fc: Front Camera mega pixels (Numeric)

four\_g: Has 4G or not (Boolean)

int\_memory: Internal Memory in Gigabytes (Numeric)

m\_dep: Mobile Depth in cm (Numeric)

mobile\_wt: Weight of mobile phone (Numeric)

n\_cores: Number of cores of processor (Numeric)

pc: Primary Camera mega pixels (Numeric)

px\_height: Pixel Resolution Height (Numeric)

px\_width: Pixel Resolution Width (Numeric)

ram: Random Access Memory in Megabytes (Numeric)

sc\_h: Screen Height of mobile in cm (Numeric)

sc\_w: Screen Width of mobile in cm (Numeric)

talk\_time: longest time that a single battery charge will last when you are (Numeric)

three\_g: Has 3G or not (Boolean)

touch\_screen: Has touch screen or not (Boolean)

wifi: Has wifi or not (Numeric)

price\_range: Price range indicating how high the mobile phone price is (Boolean)