Mobile Device Price Prediction Using Machine Learning Techniques

Parnika De
Department of Computer Science
San Jose State University
San Jose, USA
parnika.de@sjsu.edu

Lolitha Sresta Tupadha
Department of Computer Science
San Jose State University
San Jose, USA
lolithasresta.tupadha@sjsu.edu

***Abstract*—Mobile device price prediction is very important for consumers as well as marketers. Consumers need to know what they are paying for and the marketers should keep up with the competition and should rate their mobile device well. In this paper we will explore various machine learning techniques and compare their results. Along with machine learning techniques such as K-means, Random Forest, SVM and Logistic Regression we would also explore and use Feature Selection and Boosting techniques to make our prediction.**

Keywords—Feature Selection, Random Forest, Logistic Regression, SVM, Ada Boost, K-Means Clustering.

I. Introduction

In modern times mobile devices have become a part and parcel of everyone's life. Every person wants to get the most out of their mobile devices e.g. mobile phones, even if they don't want to pay much for their phone. When a phone is purchased there are a few things that people look at primarily besides its ability to make calls. These factors include primary camera megapixel, phone size, ram, battery power, network connectivity, etc. People in developed countries change phones every two years even sometimes every year. For them the price is not something that keeps them from buying phones, mostly it's the hype of getting new features on a phone or getting a better camera with the phone. In this paper, we would look into a few features and see what are the main features that can make prices of phones fall into which category.

The next most effective thing is the price of the mobile phone. Price is very important factor for marketing of anything. But how determine which product is worth the price, to do that we need to leverage the use of machine learning technique to predict the price of mobile devices. Mobile devices come with many features. Some are more important that the other when it comes to functionality of the mobile devices and therefore those features play an important role in predicting the price more than other non-usable features. In this paper we see a comparison of machine learning techniques with and without feature selection in prediction of Mobile device prices.

In this paper we will be using K-means clustering to see how the mobile devices have been clustered according to their features. We will also be using Random forest and SVM for classifying the mobile devices. Another prediction algorithm that we will be looking into is Linear Regression. In addition to that boosting will be performed and the boosting algorithm that would be used is AdaBoost. The random forest will be divided into one with feature selection and one without feature selection and then we will check and compare their result and accuracy. In today's world picking out the best feature for using and marketing that is very important. In the table next we will see all features that were present for the mobile device price prediction and the meaning of each of them.

Table 1: Features and their Meaning

Feature name	Description
id	ID
battery_power	Total energy a battery can store in one time measured in mAh
blue	Has Bluetooth or not
clock_speed	The speed at which microprocessor executes instructions
dual_sim	Has dual sim support or not
fc	Front Camera megapixels
four_g	Has 4G or not
int_memory	Internal Memory in Gigabytes
m_dep	Mobile Depth in cm
mobile_wt	Weight of the mobile phone
n_cores	Number of cores of the processor
pc	Primary Camera megapixels
px_height	Pixel Resolution Height
px_width	Pixel Resolution Width
ram	Random Access Memory in Megabytes
sc_h	Screen Height of mobile in cm
sc_w	Screen Width of mobile in cm

talk_time	The longest time that a single battery charge will last when you are
three_g	Has 3G or not
touch_screen	Has a touch screen or not
wifi	Has wifi or not

II. Related Work

Using previous data to predict price of available and new launching product is an interesting research background for machine learning researchers. Sameerchand-Pudaruth [1] predict the prices of secondhand cars in Mauritius. He implemented many techniques like Multiple linear regression, k-nearest neighbors (KNN), Decision Tree, and Naïve Bayes to predict the prices. Sameerchand-Pudaruth got Comparable results from all these techniques. During research it was found that most popular algorithms i.e Decision Tree and Naïve Bayes are unable to handle, classify and predict Numerical values. Number of instances for his research was only 97(47 Toyota+38 Nissan+12 Honda). Due to a smaller number of instances used, very poor prediction accuracies were recorded [1].

Shonda Kuiper [2] has also worked in the same field. Kuiper used multivariate regression model to predict price of 2005 General Motor cars. He collected the data from available online source www.pakwheels.com. The main part of this research work is "Introduction of suitable variable selection techniques, which helped to find that which variables are more suitable and relevant for inclusion in model. This (His research) helps students and future researchers in many fields to understand the conditions under which studies should be conducted and gives them the knowledge to discern when appropriate techniques should be used [2].

Support Vector Machine (SVM) concept is used by one another researcher Mariana Listiani [3] for the same work. Listiani predicted prices of leased cars using above mentioned technique. It was found in this research that SVM technique is far better and accurate for price prediction as compared to other like multiple linear regression when a very large data set is available. The researcher also showed that SVM also handles high dimensional data better and avoids both the under-fitting and over-fitting issues. To find important features for SVM Listiani used Genetic Algorithm. However, the technique failed to show in terms of variance and mean standard deviation why SVM is better than simple multiple regression [3].

Neural Networks (NN) are more better in estimating price of house, this was concluded in the research of Limsombunchai[4]. By comparing with hedonic method his method was more accurate.Operation of both the methods are same, but in NN the model is trained first and then tested for prediction. Using both the methods NN produced higher R-sq and smaller root mean square error (RMSE), while hedonic produced lower values. This research was limited because the actual house price were missing and only estimated prices were used for the research work [4].

K Noor and Saddaqat J [5] also worked to predict the price of Vehicles using different techniques. The researchers achieved highest accuracy using multiple linear regression. This paper proposes a system where price is dependent variable which is predicted, and this price is derived from factors like vehicle's model, make, city, version, color, mileage, alloy rims and power steering [5].

A. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. Proposed Methods

In this section we would discussing all the machine learning models we will build along with ensemble methods and feature selection for some models and then in the next section we will discuss the results and compare them.

A. Feature Selection

This is a data pre-processing algorithm that takes that features that correlates the best with the identifier that would be predicted. Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that are used to train the machine learning models have a huge influence on the performance that can be achieved. Irrelevant or partially relevant features can negatively impact model performance. Feature selection and Data cleaning is the first and most important step for model designing. Feature Selection is the process where we automatically or manually select those features which contribute most to your prediction variable or output in which we are interested. The impacts of feature selection can be very beneficial if we are able to do it correctly. The positive impacts are:

- Reduce overfitting: Less significant features if kept will contribute to noise which as a result make our model overfitted.
- Improve Accuracy: Less misleading data means modeling accuracy improves.
- Reduce Training time: Fewer data points reduce algorithm complexity and algorithms train faster.

In this paper, we would be discussing 3 feature selection techniques to be sure which features contribute most. These methods are discussed below:

- Univariate Selection: In this method of selection we use the chi-square method to find which features contribute positively to the output variable and then rank them in descending order.
- Feature Importance: Feature importance gives you a score for each feature of your data, the higher the score

the more important or relevant is the feature towards your output variable.

- Correlation Heatmap: Correlation states how the features are related to each other or the target variable. Correlation can be positive, an increase in one value of feature increases the value of the target variable or negative, an increase in one value of feature decreases the value of the target variable.

B. K-Means Clustering

Clustering is one of the most common exploratory data analysis techniques used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

K-means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic means of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster. The way the k-means algorithm works is as follows:

- Specify the number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of all data points that belong to each cluster.

The algorithm starts by randomly selecting centroids. These centroids act as the beginning points of every clusters then it keeps on iterating over the clusters to find the optimal centroid for each cluster. It halts when the centroids have stabilized or when maximum number of iterations have been achieved. In this project we are using the same algorithm and the number of clusters used in this paper is four that is the optimal number of clusters to be used according to the silhouette score.

C. Random Forest Classification

It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from a randomly selected subset of the training set. It aggregates the votes from different decision trees to decide the final class of the test object. Ensemble algorithms are those which combine more than one algorithm of the same or different kind for classifying objects.

The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree.

The building block for Random forest is Decision Trees which is another classification algorithm. A decision Tree is like a series of yes/no questions asked about the data that eventually leads to the predicted class. But there is a problem just one decision tree as it is prone to overfitting if we don't limit the depth of the tree. Therefore, in this paper we won't be using decision tree instead would be using Random Forest which does not have this problem. A random forest uses two key concepts which earns it its name.

- Random sampling of training data points when building trees
- Random subsets of features considered when splitting nodes

Training a Random Forest makes it learn from a huge number of decision trees for a variety of data points. In this paper we would be using random forest for training with the full data set without preprocessing the data. Then we would do a feature selection which is a form of subset selection to classify the prices of the mobile devices.

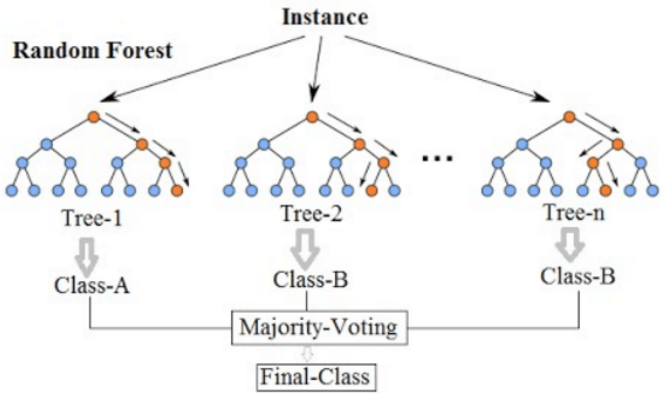

Fig 1: Structure of random forest

D. Boosting Using Ada Boost

Adaptive Boosting which is also known as AdaBoost can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms or weak learners is combined into weighted sum that represents final output of boosted classifier.

AdaBoosting is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Every learning algorithm tends to suit some problem types better than others, and typically has many different parameters and configurations to adjust before it achieves optimal performance on a dataset, AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

E. Logictic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Generally logistic regression means, binary logistic regression with binary target values but in this case, we cannot use binary logistic regression since our target variable is divided into four classes. Hence, we use another kind of logistic regression called multinomial logistic regression in order to classify our data.

F. Support Vector Machine

The objective of Support Vector Machine (SVM) is to find hyper plane in n-dimensional space to distinctly classify the data points. SVM is supervised machine learning algorithm that is trained on sets of labeled data for each category and then can categorize new text. The big ideas behind SVMs are the following.

- Separating hyperplane - We separate the labeled data into n classes based on a hyperplane.
- Maximize the margin - When constructing the separating hyperplane, we maximize the margin, which is the separation between the two classes in the training set. Intuitively, this seems like a good idea.
- Work in a higher dimensional space - We often try to reduce the dimensionality of the data we are working with because of the curse of dimensionality. However, in the context of SVMs, it's actually beneficial to work in a higher dimensional space. By moving the problem to a higher dimension, we have more space available, and hence there is a better chance of finding a separating hyperplane.
- Kernel trick - We'll use a kernel function to transform the data, with the goal of obtaining better separation. As the name suggests, this is the tricky part. While it's easy to understand the basic ideas behind SVMs from a few simple pictures, it's not so easy to understand how the kernel trick does its magic.

The goal when training an SVM is to find a separating hyperplane, where a hyperplane is defined as a subspace of one dimension less than the space in which we are working. For example, if our data lives in two-dimensional space, a hyperplane is just a line. And "separating" means exactly what it says, namely, that the hyperplane separates the two classes. If a separating hyperplane exists, we say the data is linearly separable. If our training data happens to be linearly separable, then any separating hyperplane could be used as the basis for subsequent classification. However, in this paper we deal with SVMs at higher dimension because of the dimensionality of our data.

IV. Experimental Evaluations

In this section of the paper, we would be discussing about the models which we have built and how they perform against one another. Firstly, the data was collected through Kaggle and then Feature selection algorithm was applied to the data to get most correlated features with the prices. The results from the feature selection shows that best features for mobile device price prediction are Ram, Pixel resolution height and width, internal memory and Weight of the mobile device.

	Specs	Score
13	ram	931267.519053
11	px_height	17363.569536
0	battery_power	14129.866576
12	px_width	9810.586750
8	mobile_wt	95.972863

Fig 2: Result showing correlation score (chi-square) with mobile device specification

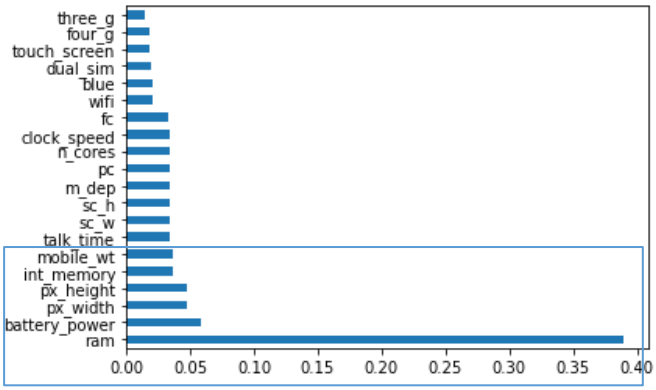

Fig 3: Feature importance score given to each feature towards the output feature i.e. the mobile device price.

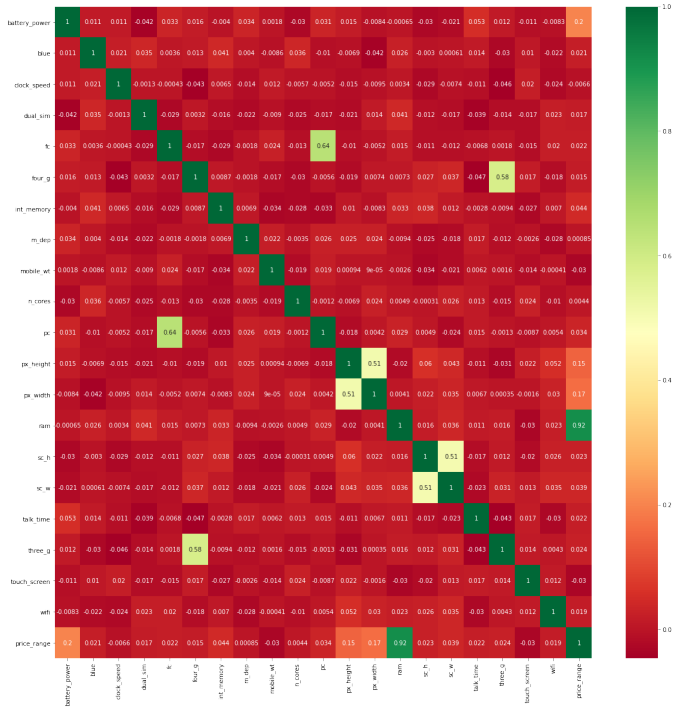

Fig 4: Correlation heatmap

Looking at the correlation heatmap, we can see that correlation between ram and price range is 0.92 which is very high. This means that price range increases if ram increases. We can plot a regression plot between ram and price range to see how price varies with increase in ram. Fig. 2. Shows regression plot. As discussed, we can see that price range increases with increase in ram size even though there is a bit of overlapping observed.

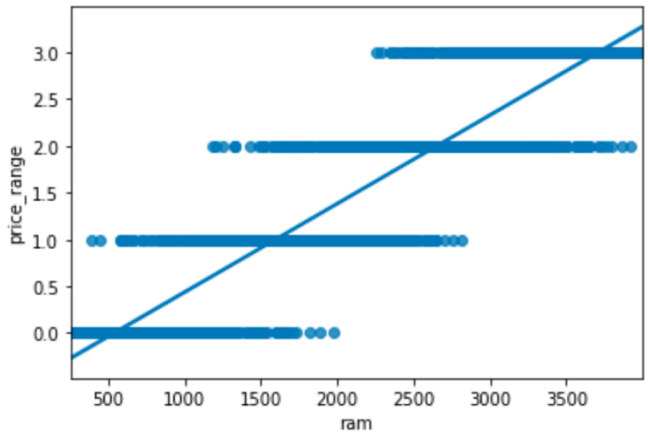

Fig 5: Regression plot between ram and price

Next we build k-means clustering model to see the features and the mobile device prices are being clustered. The score for the clustering algorithm is not as good as that of random forest. The R-squared score for K-means algorithm is 0.76. For clustering with price, we have taken the highest correlated feature Ram and clustered it with price. The result was as expected the higher the Ram value the higher is the price of the mobile device.

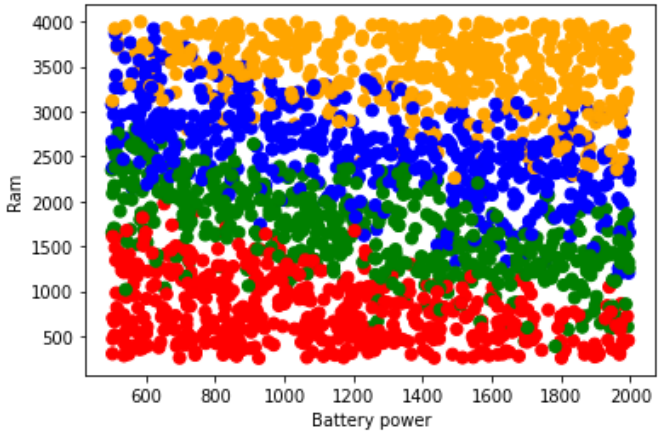

Fig 4: Clustering using Ram vs Battery power

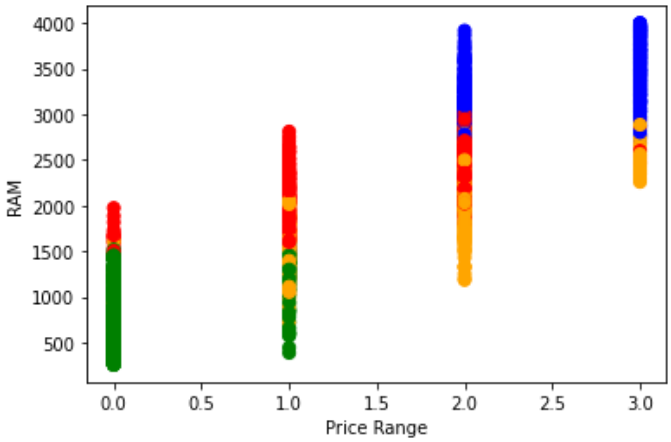

Fig 5: Clustering using Ram vs Price

The random forest was built with and without feature selection and their results did vary a bit. The R-squared score of Random forest without feature selection is 0.86 which better than K-means clustering algorithm. The R-squared score for Random Forest classification algorithm with feature selection was 0.92 which is better than regular Random Forest. The results for both Random Forests will be shown next and some of the differences in prediction will be highlighted. There are a few differences present because in the Random Forest the features with low correlation score was removed.

```
[3, 3, 2, 3, 1, 3, 3, 1, 3, 0, 3, 3, 0, 0, 2, 0, 2, 1, 3, 2, 1, 3,
 1, 1, 3, 0, 2, 0, 2, 0, 2, 0, 3, 0, 0, 1, 3, 1, 2, 1, 1, 2, 0, 0,
 0, 1, 0, 3, 1, 2, 2, 0, 3, 0, 3, 1, 3, 1, 1, 3, 3, 2, 0, 1, 1, 1,
 1, 3, 1, 2, 1, 2, 2, 3, 3, 0, 2, 0, 2, 3, 0, 3, 3, 0, 3, 0, 3, 1,
 3, 0, 1, 2, 2, 0, 2, 2, 0, 2, 1, 2, 1, 0, 0, 3, 1, 2, 0, 1, 2, 3,
 3, 3, 1, 3, 3, 3, 3, 1, 3, 0, 0, 3, 2, 1, 1, 0, 3, 2, 3, 1, 0, 2,
 1, 1, 3, 1, 2, 0, 3, 2, 1, 3, 1, 2, 2, 3, 3, 2, 2, 3, 2, 3, 0, 0,
 3, 2, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 1, 0, 3, 0, 0, 0, 1, 1, 0, 1,
 0, 0, 1, 2, 1, 0, 0, 1, 2, 2, 2, 1, 0, 0, 0, 1, 0, 3, 1, 0, 2, 2,
 2, 3, 1, 2, 3, 3, 3, 1, 2, 1, 0, 0, 1, 3, 0, 2, 3, 3, 0, 2, 0, 3,
 2, 3, 3, 0, 0, 1, 0, 3, 0, 1, 0, 2, 2, 1, 3, 0, 2, 0, 3, 1, 2, 0,
 0, 2, 1, 3, 3, 3, 1, 1, 3, 0, 0, 2, 3, 3, 1, 3, 1, 1, 3, 2, 1, 2,
 3, 3, 3, 1, 0, 1, 2, 3, 1, 1, 3, 2, 0, 3, 0, 1, 2, 0, 0, 3, 2, 3,
 3, 2, 1, 3, 3, 2, 3, 2, 2, 1, 1, 0, 2, 3, 1, 0, 0, 3, 0, 3, 0, 1,
 2, 0, 2, 3, 1, 3, 2, 2, 1, 2, 0, 0, 0, 1, 3, 2, 0, 0, 0, 3, 1, 0,
 3, 3, 1, 2, 3, 2, 3, 1, 3, 3, 2, 2, 3, 3, 3, 0, 3, 0, 3, 1, 3, 1,
 3, 3, 0, 1, 1, 3, 1, 3, 2, 3, 0, 0, 0, 0, 2, 0, 0, 2, 1, 1, 2, 2,
 2, 0, 1, 0, 0, 3, 2, 0, 3, 1, 2, 2, 1, 2, 3, 1, 1, 2, 2, 1, 2, 0,
 1, 1, 0, 3, 2, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 2, 2, 3, 2, 3, 0, 2,
 0, 3, 0, 1, 1, 1, 1, 0, 3, 2, 3, 3, 1, 3, 1, 3, 1, 2, 2, 1, 2, 2,
 1, 1, 0, 0, 0, 1, 2, 1, 0, 3, 2, 1, 2, 3, 0, 0, 3, 1, 1, 0, 3, 3,
 3, 0, 3, 0, 2, 3, 3, 3, 0, 2, 0, 2, 3, 0, 1, 1, 0, 0, 1, 1, 1, 3,
 3, 3, 2, 3, 1, 1, 2, 3, 3, 3, 1, 0, 2, 2, 2, 2, 1, 0, 2, 2, 0, 0,
 0, 3, 1, 1, 2, 2, 2, 0, 3, 0, 2, 2, 0, 3, 0, 2, 3, 0, 1, 1, 3, 3,
 1, 1, 2, 3, 2, 0, 2, 1, 2, 0, 3, 3, 1, 2, 2, 2, 3, 0, 1, 2, 3, 1,
 3, 2, 3, 1, 1, 0, 0, 3, 1, 0, 3, 2, 3, 2, 0, 3, 3, 3, 2, 3, 3, 1,
 2, 0, 2, 3, 3, 1, 0, 1, 1, 2, 2, 1, 0, 0, 2, 2, 3, 2, 0, 2, 1, 3,
 3, 0, 1, 3, 1, 2, 1, 0, 0, 0, 2, 1, 0, 1, 1, 2, 2, 0, 2, 2, 1, 0,
 3, 0, 0, 3, 2, 0, 0, 0, 0, 0, 3, 0, 3, 1, 3, 2, 1, 3, 2, 0, 1, 1,
 3, 2, 3, 1, 0, 3, 0, 2, 0, 2, 0, 0, 1, 1, 1, 2, 1, 3, 1, 3, 2, 2,
 1, 3, 2, 0, 1, 2, 0, 3, 3, 0, 2, 1, 1, 2, 0, 3, 2, 0, 3, 2, 3, 0,
 0, 3, 0, 1, 2, 3, 2, 2, 2, 2, 1, 2, 3, 0, 1, 1, 2, 1, 0, 2, 0, 2,
 2, 2, 1, 2, 0, 2, 1, 3, 0, 0, 3, 1, 3, 0, 0, 2, 3, 2, 1, 3, 2, 1,
 0, 0, 2, 3, 0, 3, 0, 0, 0, 2, 2, 1, 2, 0, 3, 2, 1, 2, 3, 3, 0, 1,
 1, 2, 1, 2, 2, 0, 1, 3, 1, 1, 3, 1, 2, 3, 1, 1, 1, 2, 3, 3, 0, 2,
 3, 0, 2, 3, 2, 2, 2, 3, 2, 0, 1, 2, 0, 2, 1, 1, 2, 2, 2, 1, 2, 0,
 0, 1, 3, 1, 0, 1, 1, 3, 1, 0, 0, 3, 2, 2, 3, 0, 3, 2, 2, 1, 3, 0,
 1, 3, 1, 2, 1, 2, 2, 2, 0, 3, 0, 2, 3, 0, 3, 1, 2, 3, 1, 0, 2, 3,
 1, 0, 1, 1, 2, 1, 3, 0, 2, 2, 0, 2, 3, 2, 3, 0, 2, 1, 1, 2, 2, 3,
 3, 0, 2, 1, 2, 1, 3, 0, 1, 3, 0, 1, 0, 0, 3, 2, 2, 0, 0, 0, 0, 3,
 2, 3, 3, 0, 0, 2, 1, 0, 2, 2])
```

Fig 6: Random Forest without Feature Selection

```
[3, 3, 3, 3, 1, 3, 3, 1, 3, 0, 3, 3, 0, 0, 2, 0, 2, 1, 3, 2, 1, 3,
 1, 1, 3, 0, 2, 0, 3, 0, 2, 0, 3, 0, 0, 1, 3, 1, 2, 1, 1, 2, 0, 0,
 0, 1, 0, 3, 1, 2, 1, 0, 3, 0, 3, 1, 3, 1, 1, 3, 3, 2, 0, 2, 1, 1,
 1, 3, 1, 2, 1, 2, 2, 3, 3, 0, 2, 0, 2, 3, 1, 3, 3, 0, 3, 0, 3, 1,
 3, 0, 1, 2, 2, 0, 2, 2, 0, 2, 1, 2, 1, 0, 0, 3, 0, 2, 0, 1, 2, 3,
 3, 3, 1, 3, 3, 3, 3, 2, 3, 0, 0, 3, 2, 1, 2, 0, 3, 2, 3, 1, 0, 2,
 1, 1, 3, 1, 1, 0, 3, 2, 1, 3, 1, 3, 2, 3, 3, 2, 2, 3, 2, 3, 1, 0,
 3, 2, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 1, 0, 3, 0, 0, 0, 1, 1, 0, 1,
 0, 0, 1, 2, 1, 0, 0, 1, 2, 2, 2, 1, 0, 0, 0, 1, 0, 3, 1, 0, 2, 2,
 2, 3, 1, 2, 3, 2, 3, 2, 2, 1, 0, 0, 1, 2, 0, 3, 3, 3, 0, 2, 0, 3,
 2, 3, 3, 1, 0, 1, 0, 3, 0, 1, 0, 2, 2, 1, 2, 1, 3, 0, 3, 1, 2, 0,
 0, 2, 1, 3, 3, 3, 1, 1, 3, 0, 0, 2, 3, 3, 1, 3, 1, 1, 3, 2, 1, 2,
 3, 3, 3, 1, 0, 1, 2, 3, 2, 1, 3, 2, 0, 3, 0, 1, 2, 0, 0, 3, 2, 3,
 3, 2, 1, 3, 3, 2, 3, 2, 2, 1, 2, 0, 2, 3, 1, 0, 0, 3, 0, 3, 0, 1,
 2, 0, 2, 3, 1, 3, 2, 2, 1, 2, 0, 0, 0, 1, 3, 2, 0, 0, 0, 3, 2, 0,
 3, 3, 1, 2, 2, 2, 3, 1, 3, 3, 2, 2, 2, 3, 3, 0, 3, 0, 3, 1, 3, 1,
 3, 3, 0, 1, 1, 3, 1, 3, 2, 3, 0, 0, 0, 0, 2, 0, 0, 2, 2, 1, 2, 2,
 2, 0, 1, 0, 0, 3, 2, 0, 3, 1, 2, 2, 1, 2, 3, 1, 1, 2, 2, 1, 2, 0,
 1, 1, 0, 3, 2, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 2, 2, 3, 2, 3, 0, 3,
 0, 3, 0, 1, 1, 1, 1, 0, 3, 2, 3, 3, 1, 3, 1, 3, 1, 3, 2, 1, 2, 2,
 1, 1, 0, 0, 0, 1, 2, 1, 0, 3, 2, 0, 2, 3, 0, 0, 3, 1, 1, 0, 3, 3,
 3, 0, 3, 0, 2, 3, 3, 3, 0, 2, 0, 2, 3, 0, 1, 1, 0, 0, 1, 1, 1, 3,
 3, 3, 2, 3, 1, 2, 2, 2, 3, 3, 2, 0, 2, 1, 2, 2, 1, 0, 2, 2, 0, 0,
 0, 3, 1, 0, 2, 2, 2, 0, 3, 0, 2, 2, 0, 3, 0, 2, 3, 0, 1, 1, 3, 3,
 1, 1, 2, 3, 2, 0, 2, 1, 3, 0, 3, 3, 1, 2, 2, 2, 3, 0, 1, 2, 3, 1,
 3, 2, 3, 1, 1, 1, 0, 3, 1, 0, 3, 2, 3, 2, 0, 3, 3, 3, 2, 3, 3, 1,
 2, 1, 2, 3, 3, 1, 0, 1, 1, 2, 2, 1, 0, 0, 2, 2, 3, 2, 0, 2, 1, 3,
 3, 0, 1, 3, 0, 2, 1, 1, 0, 0, 2, 1, 0, 1, 1, 2, 2, 0, 2, 2, 1, 0,
 3, 0, 0, 3, 2, 0, 0, 0, 0, 0, 3, 0, 3, 1, 3, 2, 1, 3, 2, 0, 1, 0,
 3, 2, 2, 2, 0, 3, 0, 2, 0, 2, 0, 1, 1, 1, 1, 2, 1, 3, 1, 3, 2, 2,
 1, 3, 2, 0, 1, 2, 0, 3, 3, 0, 2, 1, 1, 2, 0, 3, 2, 0, 3, 2, 3, 0,
 0, 3, 0, 2, 2, 3, 2, 2, 2, 2, 1, 2, 3, 0, 1, 1, 1, 2, 1, 0, 0, 1,
 0, 0, 3, 0, 1, 2, 0, 0, 1, 1, 3, 0, 3, 2, 3, 0, 0, 1, 2, 2, 1, 0,
 1, 1, 0, 1, 1, 0, 0, 3, 3, 0, 3, 1, 2, 3, 0, 1, 0, 2, 2, 0, 3, 1,
 0, 3, 0, 1, 0, 3, 3, 3, 2, 3, 0, 3, 2, 0, 0, 0, 2, 3, 2, 0, 2, 1,
 3, 1, 0, 3, 2, 0, 3, 1, 2, 1, 1, 1, 3, 1, 1, 1, 2, 0, 0, 1, 2, 0,
 2, 0, 0, 0, 0, 3, 3, 3, 3, 0, 1, 2, 2, 1, 0, 0, 2, 1, 0, 2, 0, 3,
 2, 2, 1, 2, 0, 2, 1, 3, 0, 0, 3, 1, 3, 0, 0, 2, 3, 3, 1, 2, 2, 1,
 0, 0, 2, 3, 0, 3, 0, 0, 0, 2, 2, 1, 2, 0, 3, 2, 1, 2, 3, 3, 0, 1,
 1, 2, 1, 2, 2, 0, 1, 3, 1, 1, 3, 1, 2, 3, 2, 1, 1, 2, 3, 3, 0, 2,
 3, 0, 2, 3, 2, 2, 2, 3, 2, 0, 1, 2, 0, 2, 1, 1, 2, 2, 2, 1, 2, 1,
 1, 1, 3, 1, 0, 1, 2, 3, 1, 0, 0, 2, 2, 2, 3, 0, 3, 3, 2, 1, 3, 0,
 1, 3, 1, 2, 1, 1, 3, 2, 0, 3, 0, 2, 3, 0, 2, 2, 2, 3, 1, 0, 2, 3,
 1, 0, 2, 1, 2, 1, 3, 0, 2, 2, 0, 2, 3, 2, 3, 0, 2, 1, 1, 2, 2, 3,
 3, 0, 2, 1, 2, 1, 3, 0, 1, 3, 0, 1, 0, 0, 3, 2, 2, 0, 0, 0, 0, 3,
 2, 3, 3, 0, 0, 2, 1, 0, 2, 2])
```

Fig 7: Random Forest with Feature Selection

We trained an SVM using different kernels to evaluate which kernel would give the best results. We achieved an accuracy of 93.48 % using polynomial kernel and an accuracy of 94.69 % using gaussian kernel. In this case, best accuracy of 96.7% is achieved by linear kernel whereas sigmoid kernel gave the least accuracy of 19.24 %.

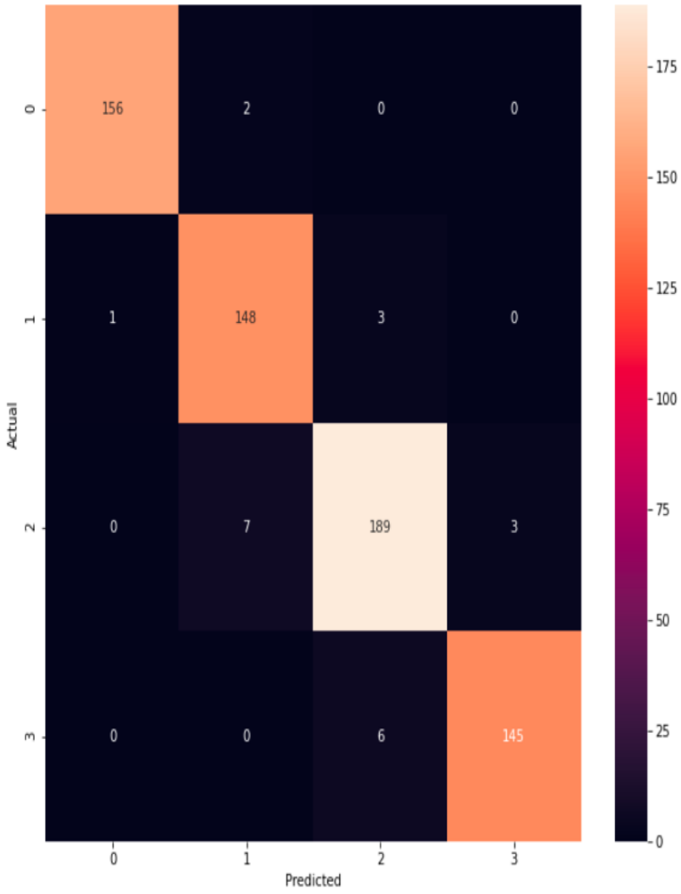

Fig 8: Confusion matrix for SVM with linear kernel

We trained a regression model using logistic regression. Since we do not want with binary classification in this paper, we trained multinomial logistic regression model and achieved an accuracy of 68.94 %. The confusion matrix generated using logistic regression model is given in Fig. 9.

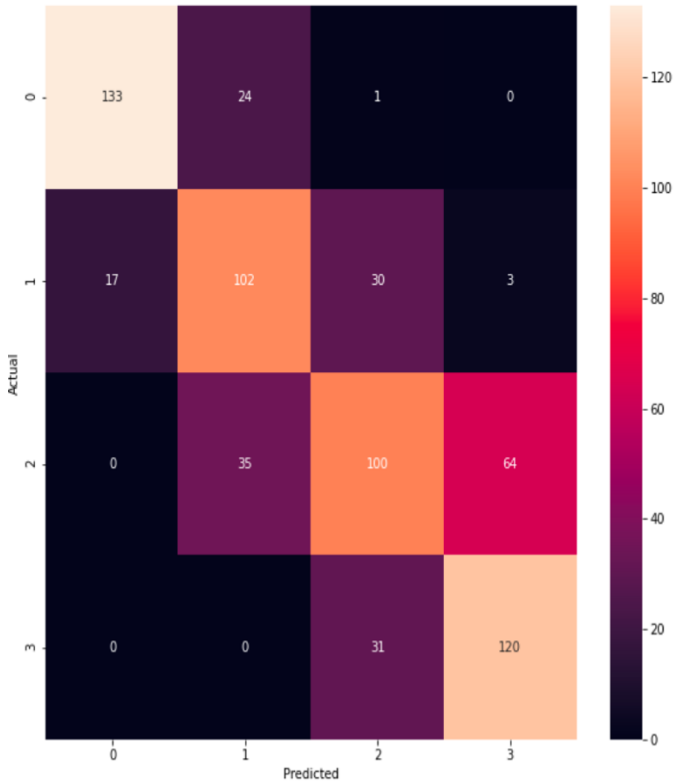

Fig 9: Confusion matrix for logistic regression

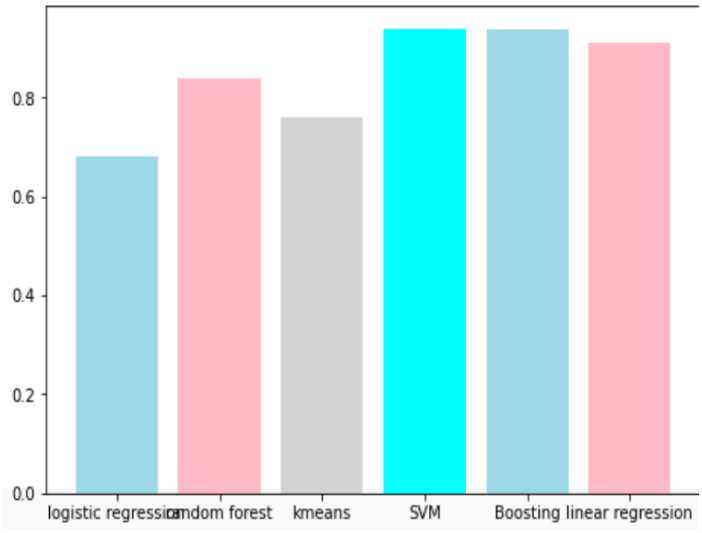

Fig 10: Accuracy of models

We evaluated and compared different algorithms and we concluded that best performance is given by SVM with linear kernel. Fig 10 shows the performance of each algorithm

V. Conclusion and Discussions

As we know mobiles have become inevitable part of human lives and pricing them at the right price according to the market gives market advantage for mobile companies and analyzing the mobile features and knowing under which price range a mobile falls into helps user to make better decision in choosing his phone. In this paper, we have done feature analysis, model development, model evaluation using data we obtained from Kaggle. We have also looked into various algorithms and obtained the best one which is SVM using linear kernel to have best accuracy among all. The accuracy of SVM is 96.7%. SVM in general have declared itself as the winner as with any kernel type it's above all the models created. Random Forest with feature selection is also good having the second best accuracy of 92.6%.

References

[1] G Sameerchand Pudaruth . "Predicting the Price of Used Cars using Machine Learning Techniques", International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 753- 764

[2] Shonda Kuiper, "Introduction to Multiple Regression: How Much Is Your Car Worth? " , Journal of Statistics Education · November 2008

[3] Mariana Listiani , 2009. "Support Vector Regression Analysis for Price Prediction in a Car Leasing Application". Master Thesis. Hamburg University of Technology.

[4] Limsombunchai, V. 2004. "House Price Prediction: Hedonic Price Model vs. Artificial Neural Network", New Zealand Agricultural and Resource Economics Society Conference, New Zealand, pp. 25-26. 2004

[5] Kanwal Noor and Sadaqat Jan, "Vehicle Price Prediction System using Machine Learning Techniques" , International Journal of Computer Applications (0975 – 8887) Volume 167 – No.9, June 2017.

[6] Mobile data and specifications online available from https://www.gsmarena.com/ (Last Accessed on Friday, December 22, 2017, 6:14:54 PM)

[7] Introduction to dimensionality reduction, A computer science portal for Geeks. https://www.geeksforgeeks.org/dimensionality-reduction/ (Last Accessed on Monday , Jan 2018 22, 3 PM)

[8] Ethem Alpaydın, 2004. Introduction to Machine Learning, Third Edition. The MIT Press Cambridge, Massachusetts London, England