Mobile Price Prediction

MET CS 777 – Final Project

**Overview**

The goal of this project is to predict price of a mobile device based on following features –

1. Battery Power
2. Processor Clock Speed
3. Front Camera Mega Pixels
4. Internal Memory
5. Mobile Depth in cm
6. Mobile Weight
7. Number of cores
8. Primary Camera Mega Pixels
9. Screen Height in pixels
10. Screen Width in pixels

A mobile device is expected to have a price based on the above features.

**Data Set**

Mobile data public data set is available at <https://www.kaggle.com/iabhishekofficial/mobile-price-classification>. The data set contains the above fields. The data set contains 2000 records, and 21 columns. Out of the 21 columns we have selected the most important features based on the understanding that certain features tend to impact the price of the mobile phone.

The data is split into training and testing models.

**Research Question**

For this given dataset, we are predicting the price as high or low, based on the above stated features. The data set contains the classification column as Low, High, and very high. These classes have been reduced to Low and High.

Logistic Regression and Support Vector Machine models will be used for this binary classification. Gaussian Mixture Model was applied for a multi class classification, but the results were not as good as Logistic Regression and SVM models.

**After this implementation …**

We wish to predict the cost of a mobile phone based on the above parameters. This could be helpful for price estimation of a mobile phone seller.

**Correctness of the model**

F1 score of the model was evaluated for Logistic Regression and SVM models, and both the models performed similar.

Below are the results from each of the models –

Logistic Regression –

F1 = 0.6728599867285998 , Precision = 0.507 , Recall = 1.0

SVM –

F1 = 0.6728599867285998 , Precision = 0.507 , Recall = 1.0

The models are not accurate, but the prediction results are better than a flip of a coin. A regularization factor of 0.2 was used for tuning the model. Other model hyper parameters were also tried, but the model performed best with the regularization factor of 0.2

Spark’s implementation of Logistic Regression and SVM were used for this prediction.

**Spark History**

Logistic Regression

**Timeline

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

Support Vector Machine

Graphical user interface, timeline

Description automatically generated

Table

Description automatically generated