The goals of this project is to find which mobile features influence the pricing of mobile phones the most and to predict the price of a phone based on those features. The data set used for this project contains a variety features pertaining to the mobile phone market, such as Ram, camera quality number of n-core processors etc.

The first thing that was done to the data was to try and get an idea of what the data set looked like and what information the data set contains. This was done with the code and out put show below. The data contains two csv files one for raining and one for testing.The training set is implemented to build up the models, while a test (or validation) set is to validate the models built.

A picture containing table

Description automatically generated

The picture above shows that the training set has 2000 rows and 21 columns whilst the test set has 1000 rows and 21 columns. After this to further get an idea of what features the data set contained the operations in the picture below were performed.

Table

Description automatically generated

Furthermore, to further understand if any feature engineering was needed. The code shown in the below picture was performed resulting in the output that is in the same picture.

Table

Description automatically generated

The information in the output shows the data type , the column names and the number of null values in each column. The fact that there are no null values shown in the data set shows that no feature engineering is required. However, to further verify from a visual point of view that there are truly no null values the code shown below as performed resulting in the subsequent output, which verifies ghat they are indeed no null values.

Chart, bar chart

Description automatically generated

The last thing to be checked was the nature of the values for the price range to be predicted. As shown below the values are categorical, with zero being the lowest price and 3 being the highest price range.

Graphical user interface, text

Description automatically generated

For the second step, the goal will be trying to find which mobile features influence the pricing of mobile phones the most. To reach the goal, the correlation between price range and all the other features is going to be conducted using the code below.

Graphical user interface, text, application

Description automatically generated

The result of the above code is shown in the picture below. The picture shows a heatmap with correlation between the features. When looking to see which features have the strongest correlations with price is clear that battery power, ram and pixel height and width have the highest correlation.

Chart, scatter chart

Description automatically generated

To further understand the relationship between price range and features with strong correlations. The visualisations below give further insight into one of the features with strong relationships with price range and ram and at least two feature that many would’ve thought has a strong relationship with price range such as internal memory and battery power.

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

In the above pictures it is discernible that the relationships with a correlations are linear and the ones with vey low or no correlation are not as linear.

The third step was to see if it was possible to accurately predict the price ranges of phones using a few machine learning algorithms. The algorithms used for this data set are linear regression, K- nearest Neighbour, scale vector machine and decision trees. The following paragraphs are going to briefly explain each algorithm and show the code and results associated with the data set in mind.

**Linear regression** refers to linear relationship between two or more variables. Meaning that if this relationship between two or more variables is drawn, it will be a straight line. The regression technique finds out linear relationship between (input) and (output), hence, the name Linear Regression. It performs task to predict a dependent variable value based on a given independent variable The linear regression shown below has an accuracy of 91%.

Text

Description automatically generated

**K Nearest Neighbor** is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbors are classified. The KNN shown below has an accuracy of 93%.

Table

Description automatically generated

The goal of using a **Decision Tree** is to create a training model that can use to predict the class or value of the target variable (in this case price range) by learning simple decision rules inferred from prior data (training data). The accuracy of the decision tree in this data set is 82%.

Table

Description automatically generated with low confidence

SVM or Support Vector Machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. The accuracy of the prediction is 90%. The confusion matrix seeks to further find the accuracy through classification.

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

The final step is meant to classify the test data set and then compare the results with the results from the training, to make sure that there are no bias and or variance errors. As a result of a lack time to run all the above algorithms, I chose to only do classification of the test data set using the SVM model as shown in the pictures below. The accuracy of the test data set is 88% which is very similar to the 90% of the training data set using the SVM model. Meaning that the data does not have significant bias or variance errors.

Background pattern

Description automatically generated

Graphical user interface, application, table

Description automatically generated

In conclusion, I would’ve loved to run the algorithms on the test set and see if there was any significant errors. I would’ve also wanted to run some cross validation on all models in both the test and training data set and lastly it would have been nice to visualize all the algorithms used in this project. In the future I plan to learn and understand more about validation and evaluation of algorithms, so I can better implement them.