KONYA FOOD AND AGRICULTURE UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

INDUSTRIAL ENGINEERING DEPARMENT

IENG 4206

INTRODUCTION TO DATA MINING

TERM PROJECT REPORT:
MOBILE PHONE PRICE RANGE PREDICTION

Submitted by:

NURSENA ERTUĞRUL - 182010020075

TABLE OF CONTENTS

1. Literature review

In a study, concluding that most feature selection algorithms and classifiers yielded similar results, except for the combination of WrapperattributEval and Decision Tree J48 classifier. This particular combination achieved the highest accuracy while selecting the minimum yet most relevant features. It is worth noting that in Forward selection, adding irrelevant or redundant features to the dataset reduces the effectiveness of both classifiers. Similarly, in backward selection, removing important features from the dataset leads to a decrease in efficiency. The primary reason for the low accuracy rate is the limited number of instances in the dataset. Additionally, it should be considered that converting a regression problem into a classification problem introduces more errors (Asim & Khan, 2018).

In another research, it was discovered that the LDA model achieved the highest accuracy (95%) in predicting mobile price classes. The study revealed that performing data preprocessing steps like normalization and standardization can enhance the accuracy of the models. The availability of feature selection and extraction algorithms proved useful in removing unsuitable and duplicate features, resulting in improved outcomes. The same approach used in this study, which involved utilizing archived data with characteristics such as cost and technical specifications for products like cars, bikes, and houses, can be applied to predict prices for other items. This application will contribute to the knowledge of both organizations and consumers, assisting them in making informed decisions regarding pricing (Varun Kiran, 2022).

In a different study, it was discovered that cost estimation played a crucial role in marketing and business. The technique employed in the study was found to be applicable in estimating the cost of various goods such as tools, food, medicines, laptops, and more. The key to an effective marketing strategy is to identify the optimal product that offers the minimum cost and maximum features. By considering factors such as product requirements, prices, and manufacturing company, the suitability of products can be evaluated. Through data mining and analysis, it was observed that a well-performing product could be recommended to customers by identifying the most suitable price range. In the specific use case of this study, a high-accuracy prediction of the price range for mobile phones was achieved by training a model on a dataset of two thousand samples with diverse features using the J48 decision tree learning model in the WEKA tool (Arora et al., 2020).

In a separate study, correlation analysis was conducted to determine the relationship between features, which was then visualized using a heatmap. The top 10 most relevant features were identified for further analysis. Three machine learning techniques were utilized, including Naïve Bayes, Decision Tree, and Random Forest. The Random Forest technique achieved the highest accuracy of 91%, while the Decision Tree technique achieved an accuracy of 84% due to the use of the 10 most related features. However, the other techniques used fewer features, resulting in lower accuracy (Sakib, 2020).

In a different study, an ANN-GWO-based model was introduced to predict housing prices using multiple variables. The ANN was trained to identify the best swarm particles or wolves from the GWO, which were then used to predict housing prices. In experiments, the ANN was initially trained with 4 neurons and the number of neurons was gradually increased to determine if better accuracy could be achieved. It was observed that the ANN's performance was significantly higher at 98% with 6 neurons in the hidden layer compared to 4, 5, and 5. However, since 6 neurons were not sufficient for the model to trace distributions, it was suggested to use a feature extraction algorithm such as CNN or RNN to extract the features of each particle (wolf) in future studies (Al-Gbury & Kurnaz, 2020).

2. Introduction

Due to the fast advancement of the internet and technology, online shopping has gained significant importance in people's everyday lives. It is increasingly becoming a popular choice due to its affordability, convenience, user-friendly nature, and other advantages (Bukvić et al., 2022). The cost of a product is a crucial attribute in both marketing and business. Customers are primarily concerned with the price and quality of the item they wish to purchase. Therefore, the primary objective of every customer is to estimate the price of the product before making a purchase. The recent advancements in Artificial Intelligence have provided computer science and automation industries with the ability to perform tasks more efficiently than humans, while still requiring human intelligence and discernment. This has enabled machines to answer questions intelligently and technically in a rapidly developing engineering field (Zehtab-Salmasi et al., 2021). Machine learning offers the best techniques and methods for artificial intelligence, including classification methods, regression techniques, supervised and unsupervised learning. Machine learning algorithms can be written using various tools such as Python, MATLAB, and WEKA. There are many methodologies and classifiers in machine learning, such as Decision Tree and Naïve Bayes. Feature selection algorithms can also be used to select the best features and minimize the dataset, reducing the computational complexity of the problem. Optimization techniques can also be used to reduce the dimensionality of the dataset. Mobile devices are the most important devices in today's world, and almost everyone owns one. Mobiles are the most selling and purchasing devices in the market, with new versions and features being launched every day. In this paper, we focused on predicting mobile prices, but similar predictions can be made for other products such as cars, bikes, and laptops. When analyzing mobile price prediction, the processor, battery timing, size, thickness, internal memory, camera pixels, and video quality are all important factors to consider. These features are used to classify whether a mobile device is economical or expensive (Mitta, 2021) (Ercüment GÜVENÇ et al., 2021). In a rapidly changing and competitive market, mobile companies need to set optimal prices to stay ahead of their rivals. The first step in determining a price is to estimate it based on the features of the mobile phone. The objective of this research is to develop a machine learning model capable of estimating the price of a mobile phone based on its features. Potential buyers can also use the model to estimate the price of a mobile phone by inputting the required features into the tool. This approach can be used to develop a price estimation model for most products that have similar independent variable parameters. The price of a mobile phone depends on various features such as the processor, battery capacity, camera quality, display size, and thickness. These features can be used to classify phones into different categories such as entry-level, mid-range, flagship, and premium (Varun Kiran, 2022).

Moreover, mobile applications generate significant income and experience high daily sales. New editions of mobile phones with improved features and a wide range of apps are launched on a daily basis. Thousands of different cell phone models are available for purchase every day. Therefore, it is important to estimate the price range when looking for top-quality products. Similar approaches can be used to predict the prices of various goods. There are several crucial factors for determining mobile phone prices, such as the mobile processors. In today's busy lifestyle, battery life is also a vital consideration. The size of the mobile phone is another important factor in the decision-making process. Memory capacity and camera quality should also be taken into account. Internet browsing is a significant feature in this technology-driven era of the 21st century. Therefore, the size of the mobile phone is determined by considering various functions related to these factors (Radhamani et al., 2022).

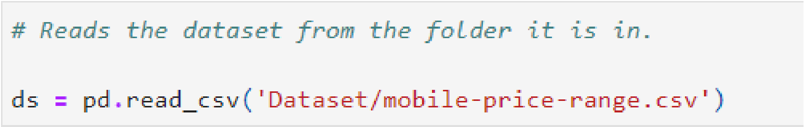
When launching a product into the market, many variables and factors are considered, especially in the case of mobile phones where features such as memory and specifications can impact the cost and competition in the market. Various constraints must be considered when determining the price, as the product should be economical and accessible while also meeting overall considerations. Mobile prices and specifications are crucial for selection and comparison, and different tools and classifiers are used to select the best features and datasets for comparison. Since thousands of mobile phones are released each year, collecting a complex dataset can be challenging. Therefore, selective features are used to reduce the complexity of the dataset and estimate the price to determine whether to release the product in the market. Multiple variables must be considered to obtain precise results for the price and other features of the mobile dataset. This will help buyers, marketers, and developers make informed decisions based on historical data of mobile phones (Kumuda et al., 2021).

In this project, it is aimed to reveal the relationship between the features of the new products to be produced by a mobile phone company that will enter the market and the prices of the products available in the market. Chapters describe how this project progressed.

3. EDA

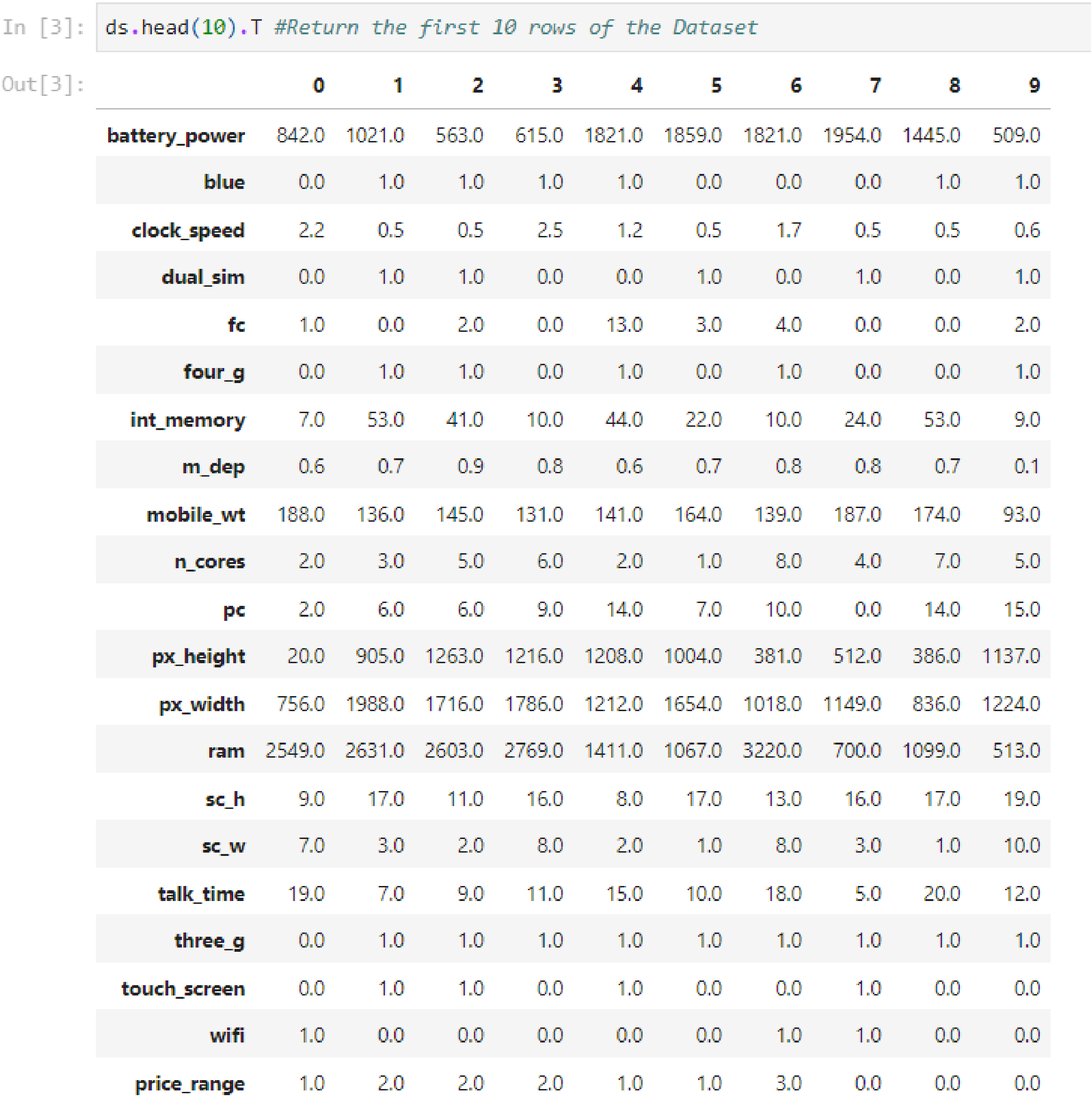
3.1. Data points

Prior to commencing work, it is necessary to establish the data points. A data point refers to the information acquired through observations within a specific timeframe, which is then utilized in data analysis and statistics to identify a specific unit. The data being examined in this project pertains to mobile phones produced from 1999 to 2017. Figure 1 shows how to read the dataset file with the read_csv function.



```
# Reads the dataset from the folder it is in.

ds = pd.read_csv('Dataset/mobile-price-range.csv')
```

***Fig*.1.** Dataset File Reading

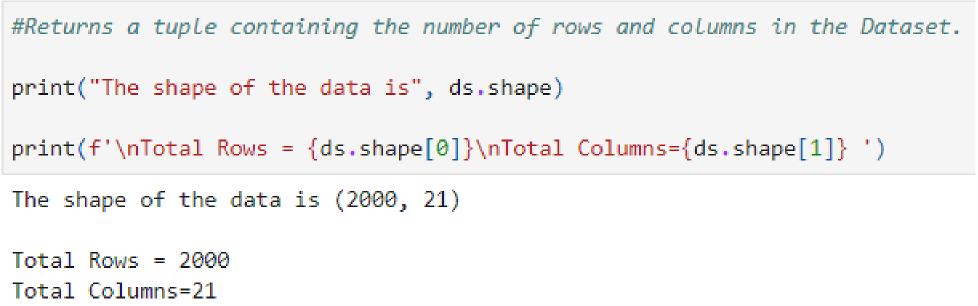


```
In [3]: ds.head(10).T #Return the first 10 rows of the Dataset

Out[3]:
```

	0	1	2	3	4	5	6	7	8	9
battery_power	842.0	1021.0	563.0	615.0	1821.0	1859.0	1821.0	1954.0	1445.0	509.0
blue	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0
clock_speed	2.2	0.5	0.5	2.5	1.2	0.5	1.7	0.5	0.5	0.6
dual_sim	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0
fc	1.0	0.0	2.0	0.0	13.0	3.0	4.0	0.0	0.0	2.0
four_g	0.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
int_memory	7.0	53.0	41.0	10.0	44.0	22.0	10.0	24.0	53.0	9.0
m_dep	0.6	0.7	0.9	0.8	0.6	0.7	0.8	0.8	0.7	0.1
mobile_wt	188.0	136.0	145.0	131.0	141.0	164.0	139.0	187.0	174.0	93.0
n_cores	2.0	3.0	5.0	6.0	2.0	1.0	8.0	4.0	7.0	5.0
pc	2.0	6.0	6.0	9.0	14.0	7.0	10.0	0.0	14.0	15.0
px_height	20.0	905.0	1263.0	1216.0	1208.0	1004.0	381.0	512.0	386.0	1137.0
px_width	756.0	1988.0	1716.0	1786.0	1212.0	1654.0	1018.0	1149.0	836.0	1224.0
ram	2549.0	2631.0	2603.0	2769.0	1411.0	1067.0	3220.0	700.0	1099.0	513.0
sc_h	9.0	17.0	11.0	16.0	8.0	17.0	13.0	16.0	17.0	19.0
sc_w	7.0	3.0	2.0	8.0	2.0	1.0	8.0	3.0	1.0	10.0
talk_time	19.0	7.0	9.0	11.0	15.0	10.0	18.0	5.0	20.0	12.0
three_g	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
touch_screen	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
wifi	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
price_range	1.0	2.0	2.0	2.0	1.0	1.0	3.0	0.0	0.0	0.0

***Fig*.2.** First 10 rows of the Dataset

The display of the first 10 rows in the dataset with the Head() function is shown in Figure 2. In addition, the features in the dataset and the first 10 data appear as the output of this function.

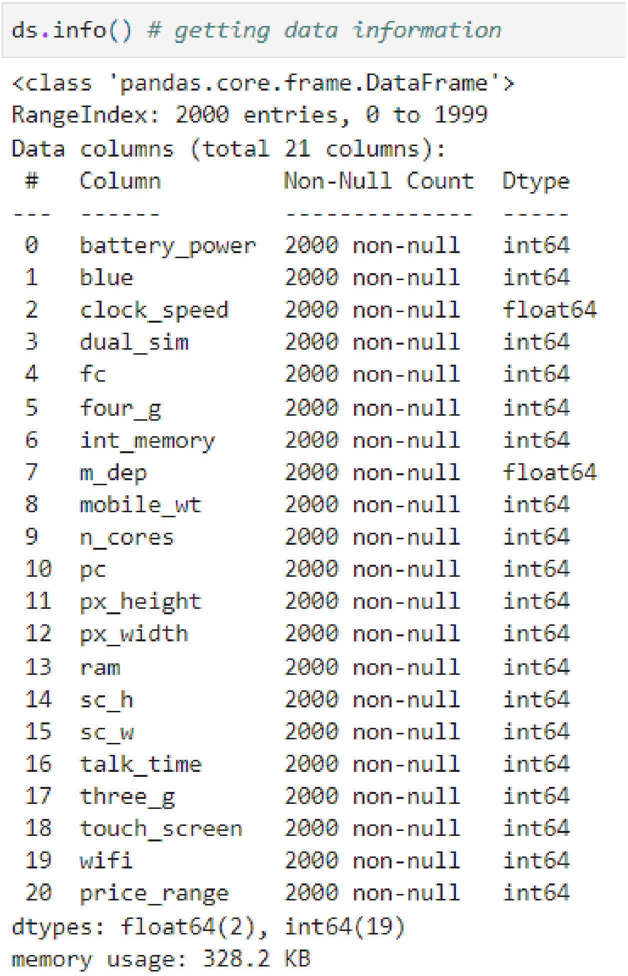


```
#Returns a tuple containing the number of rows and columns in the Dataset.

print("The shape of the data is", ds.shape)

print(f'\nTotal Rows = {ds.shape[0]}\nTotal Columns={ds.shape[1]} ')
```

```
The shape of the data is (2000, 21)

Total Rows = 2000
Total Columns=21
```

***Fig*.3.** Number of Rows and Columns

Figure 3 above shows the total amount of rows and columns in the data set. In this data set, the total rows are 2000 and the total columns are 21.



```
ds.info() # getting data information
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

***Fig*.4.** Information of Features

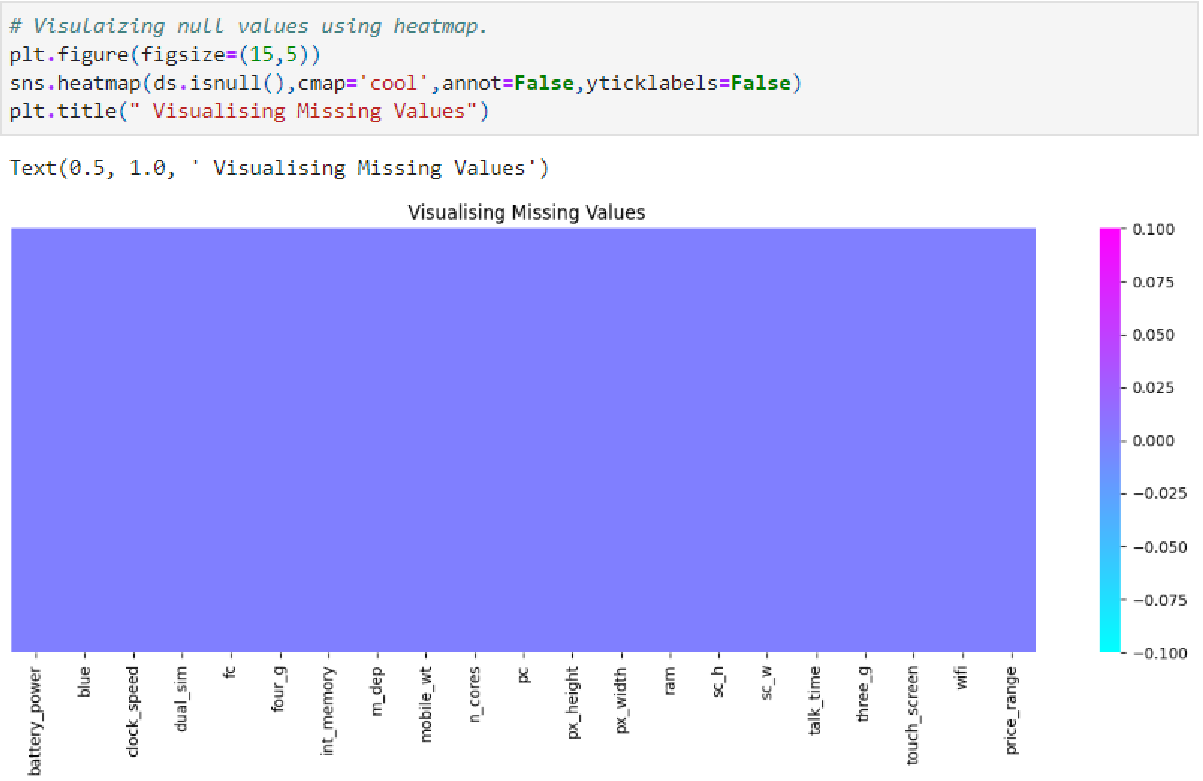
Figure 4 contains information on the previously mentioned features. As seen in the picture, there are 21 features in total. Only two of these properties are float64 Dtype. Also, there are 2000 non-null data as stated here. That is, there are no null values in this dataset.

3.2. Features and Labels

Ensuring accurate identification of features and labels in a project lays a strong groundwork. Features can be defined as the components or elements within the system that contribute towards the label. The label, in turn, encompasses the representation formed by combining all the features.

In this project, the label is determined as price range. Features are: 'battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g', 'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height ', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g', 'touch_screen', 'wifi', 'price_range'. The descriptions of the features are as follows:

battery_power: Total energy of a battery, blue: bluetooth, clock_speed: microprocessor executes instructions, dual_sim: dual sim support, fc: Front Camera mega pixels, four_g: 4G, int_memory: Internal Memory, m_dep: Mobile Depth in cm, mobile_wt: Weight of mobile phone, n_cores: Number of cores, pc: Primary Camera mega pixels, px_height: Pixel Resolution Height, px_width: Pixel Resolution Width, ram: Random Access Memory, sc_h: Screen Height, sc_w: Screen Width, talk_time: longest time in a single battery charge, three_g: 3G, touch_screen: touch screen, wifi: wifi, price_range: This is the target variable with value of 0 (low cost), 1 (medium cost), 2 (high cost) and 3 (very high) cost).



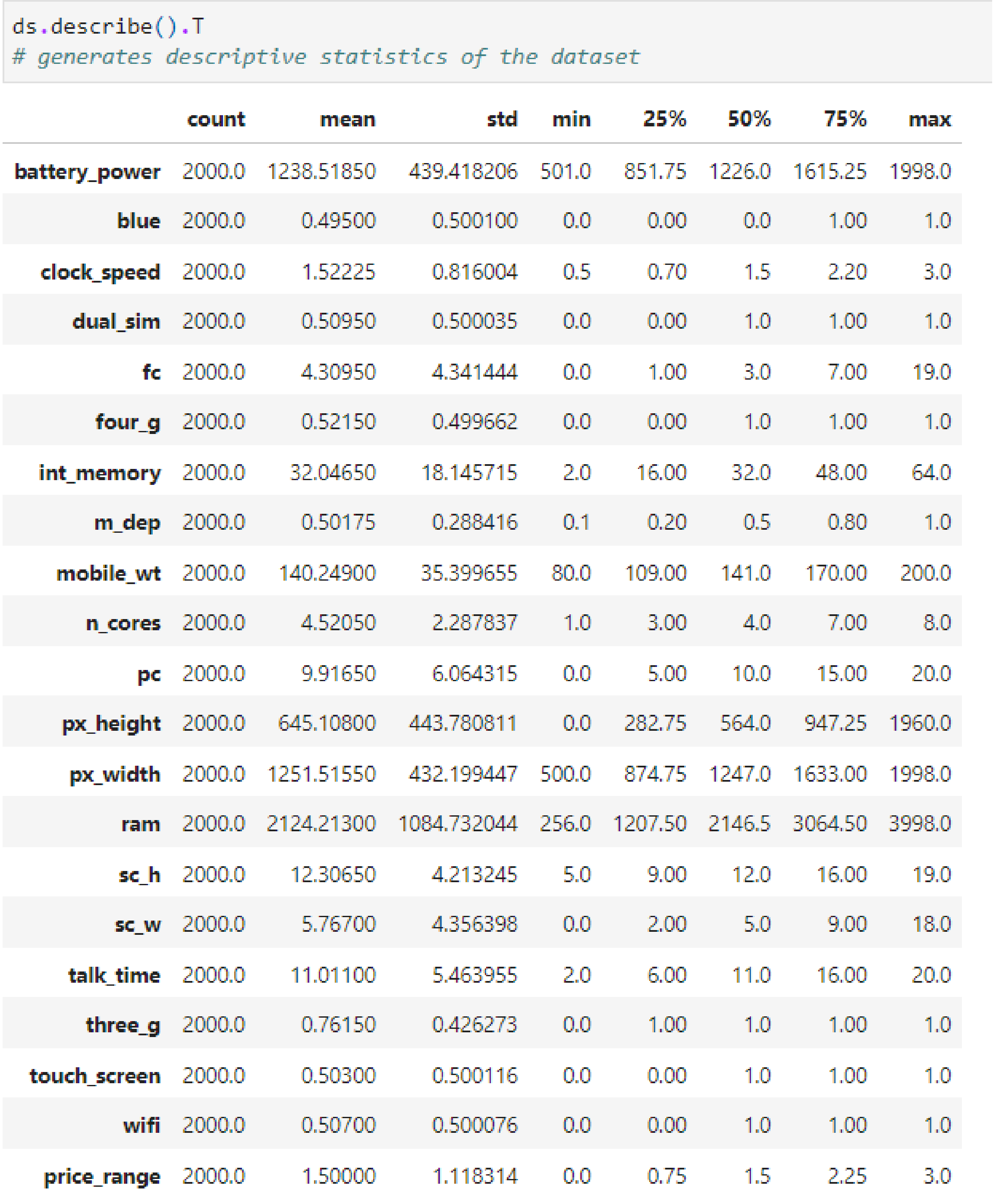
```
# Visulaizing null values using heatmap.
plt.figure(figsize=(15,5))
sns.heatmap(ds.isnull(),cmap='cool',annot=False,yticklabels=False)
plt.title(" Visualising Missing Values")
```

```
Text(0.5, 1.0, ' Visualising Missing Values')
```

***Fig*.5.** Visualizing Missing Values

The graph shown in Figure 5 shows whether there are any null values in the data set. If there were any null values, the graph would appear on the chart. Since there is no null value, there is no change in the graph.

3.3. Statistic Analysis



```
ds.describe().T
# generates descriptive statistics of the dataset
```

	count	mean	std	min	25%	50%	75%	max
battery_power	2000.0	1238.51850	439.418206	501.0	851.75	1226.0	1615.25	1998.0
blue	2000.0	0.49500	0.500100	0.0	0.00	0.0	1.00	1.0
clock_speed	2000.0	1.52225	0.816004	0.5	0.70	1.5	2.20	3.0
dual_sim	2000.0	0.50950	0.500035	0.0	0.00	1.0	1.00	1.0
fc	2000.0	4.30950	4.341444	0.0	1.00	3.0	7.00	19.0
four_g	2000.0	0.52150	0.499662	0.0	0.00	1.0	1.00	1.0
int_memory	2000.0	32.04650	18.145715	2.0	16.00	32.0	48.00	64.0
m_dep	2000.0	0.50175	0.288416	0.1	0.20	0.5	0.80	1.0
mobile_wt	2000.0	140.24900	35.399655	80.0	109.00	141.0	170.00	200.0
n_cores	2000.0	4.52050	2.287837	1.0	3.00	4.0	7.00	8.0
pc	2000.0	9.91650	6.064315	0.0	5.00	10.0	15.00	20.0
px_height	2000.0	645.10800	443.780811	0.0	282.75	564.0	947.25	1960.0
px_width	2000.0	1251.51550	432.199447	500.0	874.75	1247.0	1633.00	1998.0
ram	2000.0	2124.21300	1084.732044	256.0	1207.50	2146.5	3064.50	3998.0
sc_h	2000.0	12.30650	4.213245	5.0	9.00	12.0	16.00	19.0
sc_w	2000.0	5.76700	4.356398	0.0	2.00	5.0	9.00	18.0
talk_time	2000.0	11.01100	5.463955	2.0	6.00	11.0	16.00	20.0
three_g	2000.0	0.76150	0.426273	0.0	1.00	1.0	1.00	1.0
touch_screen	2000.0	0.50300	0.500116	0.0	0.00	1.0	1.00	1.0
wifi	2000.0	0.50700	0.500076	0.0	0.00	1.0	1.00	1.0
price_range	2000.0	1.50000	1.118314	0.0	0.75	1.5	2.25	3.0

***Fig*.6** Statistical Anlaysis

The values in the figure above are the number of data, the mean of the data, the standard deviation of the data, the minimum value of the data, the 25%, 50% and 75% quantiles of the data, and the maximum value of the data. The sc_v and px_height properties cannot be zero for a phone, but as seen in the table above, the min value of these two properties has been obtained as zero.

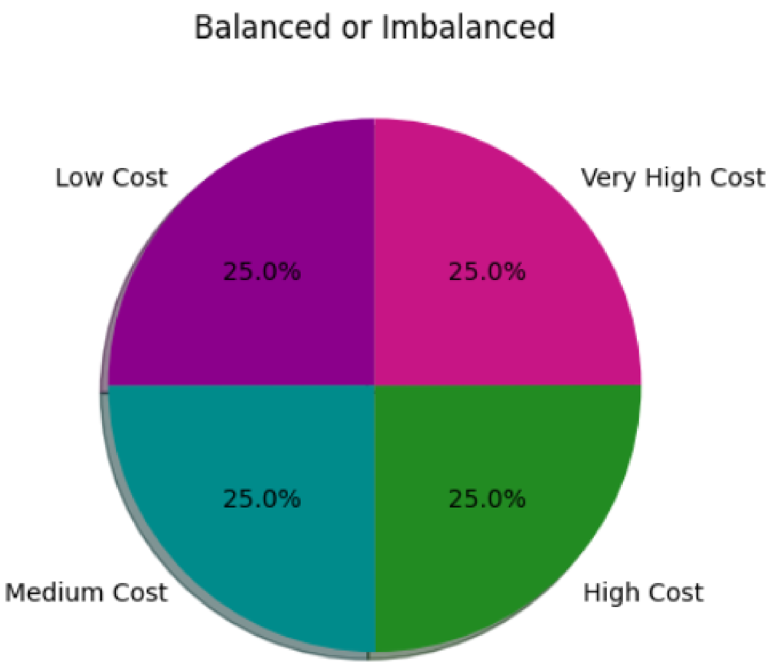


***Fig*.7.** Examinig Dataset

The pie chart in Figure 7 shows whether the dataset is balanced or unbalanced. As can be seen from this chart, the data set is in balance.

3.4. Generating New Features

In the study, new features need to be obtained in order to achieve better results. There were two features available. These are sc_width and sc_height. Using these two features, sc_size, ie screen size, was added. In addition, the pixels property was created using the px_height and px_width properties. Thus, the number of features increased from 21 to 23.

3.5. Data Visualization

Data visualization aims to make data more accessible and understandable. Provides an easy way to see and understand trends, outliers, and patterns in data.

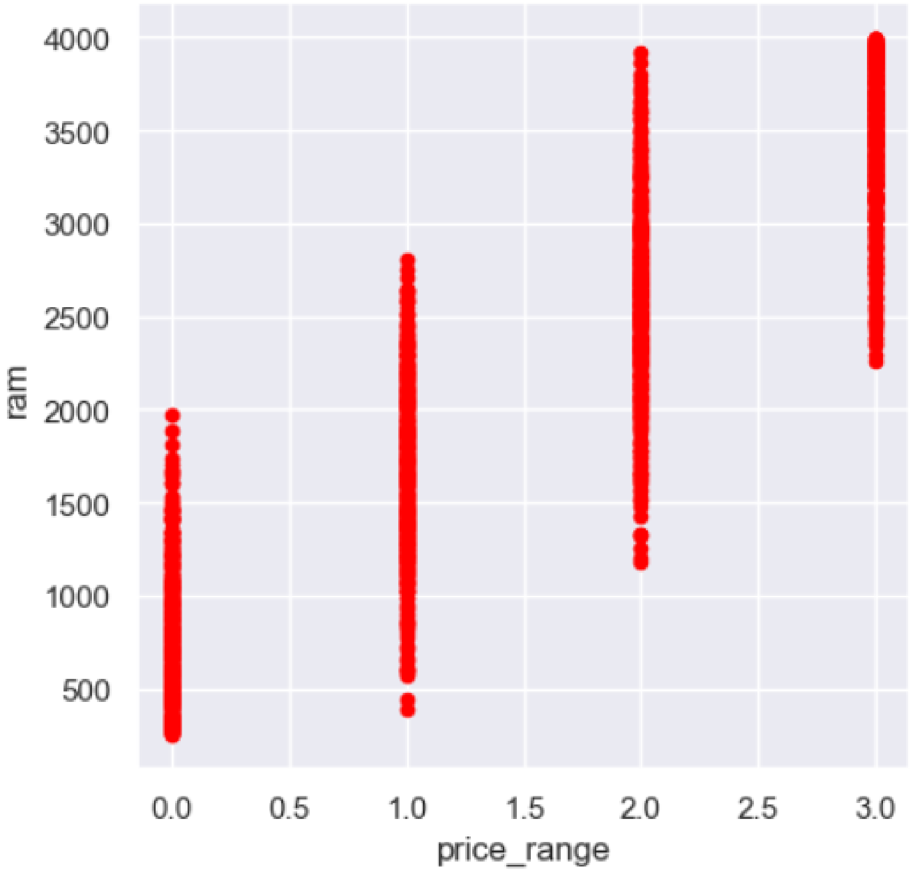


***Fig*.8.** Ram versus Price Range Chart

As shown in the figure above, ram has continuous increase with price range while moving from low cost to very high cost.

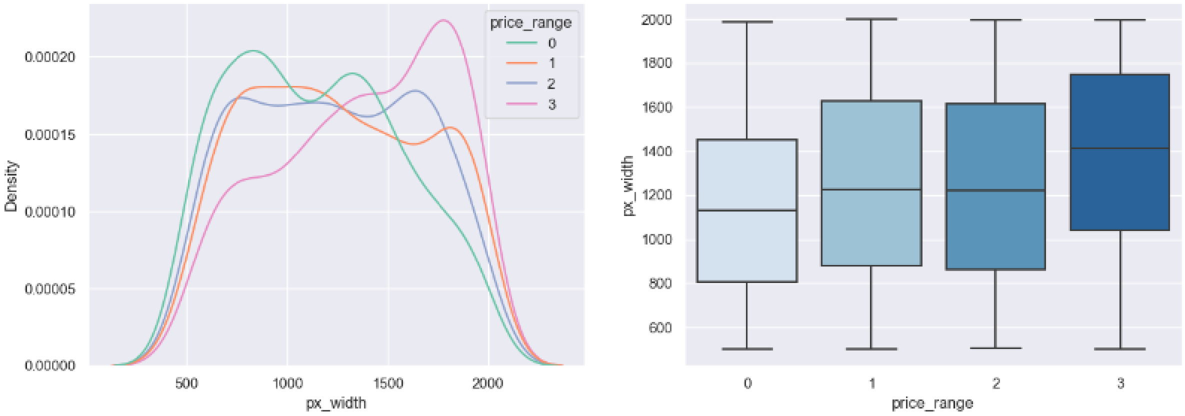


***Fig*.9.** Pixel width versus Price Range Chart

As indicated in Figure 9, there is not a continuous increase in pixel width as we move from Low cost to Very high cost. Mobiles with 'Medium cost' and 'High cost' has almost equal pixel width. So it can said be that it would be a driving factor in deciding price_range.

3.6. Correlation Analysis

Correlation analysis is a statistical technique used to assess and quantify the linear association between two variables in a dataset. When there is a high correlation between two variables, it indicates a strong relationship between them. Conversely, a low correlation coefficient suggests a weak relationship between the variables.

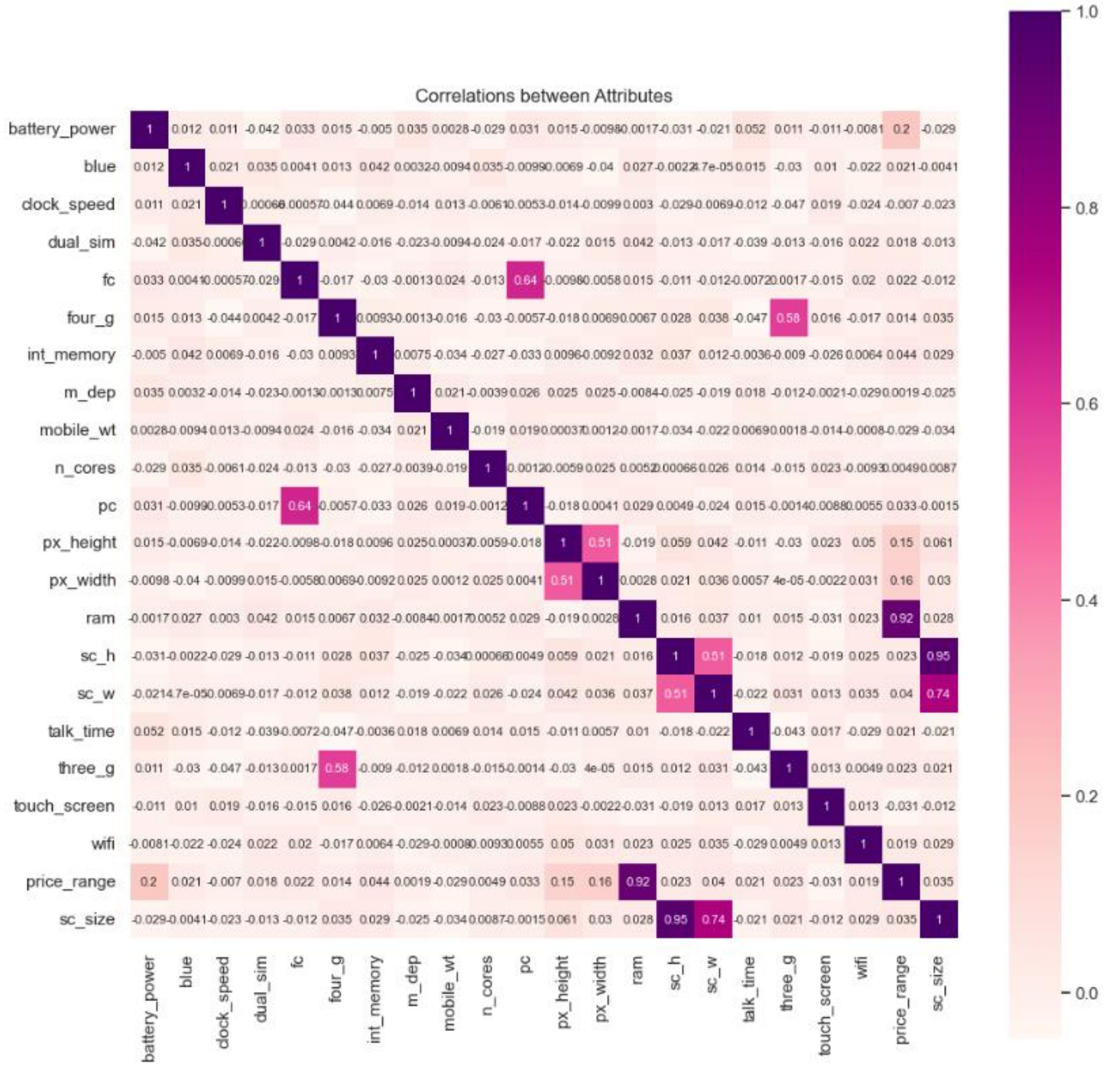


***Fig*.10.** Correlations Between Features

RAM and price_range exhibit a strong positive correlation, indicating that RAM will have a significant impact on determining the price range.

There is a certain degree of collinearity present in the feature pairs ('pc', 'fc') and ('px_width', 'px_height'). These correlations are reasonable as it is likely that if the front camera of a phone is of high quality, the back camera will also be of good quality.

Furthermore, when the pixel height (px_height) increases, the pixel width (px_width) also tends to increase, resulting in an overall increase in screen resolution. Consequently, it is possible to consolidate these two features into a single feature. Despite exhibiting collinearity, it is important to note that Front Camera megapixels and Primary camera megapixels represent distinct entities, so we will retain them in their original form.

The presence of both 3G and 4G technologies shows a moderate correlation.

Most of the variables have a weak correlation with the price range.

Our dataset does not contain highly correlated inputs, indicating the absence of multicollinearity issues.

3.7. Outlier Tretament and Data Normalization

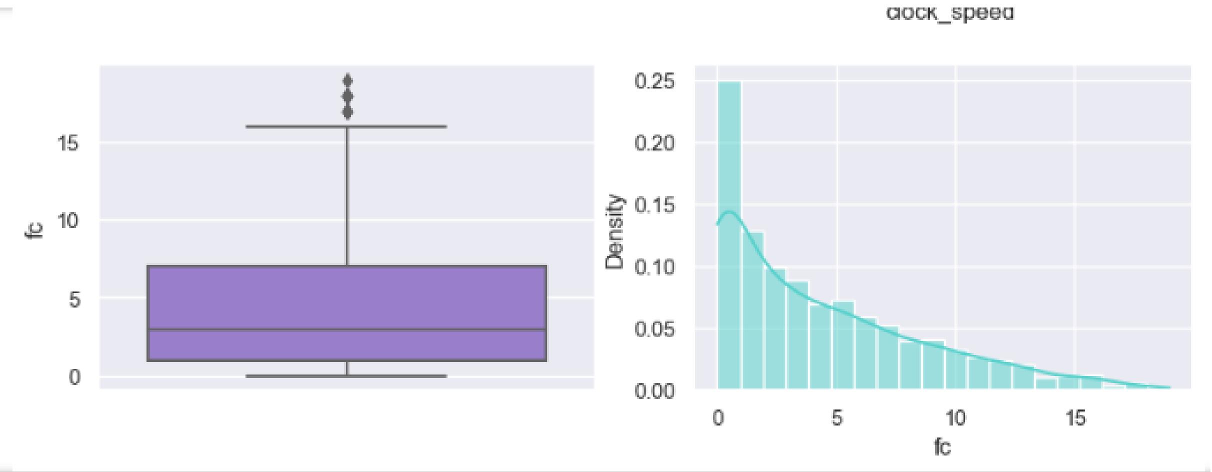


Fig.11. Front Camera Meagapixel Graph

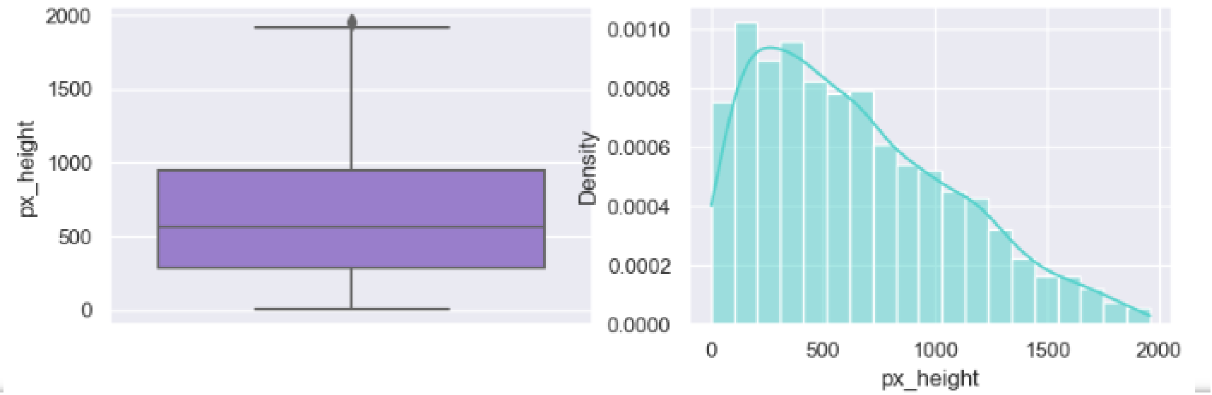
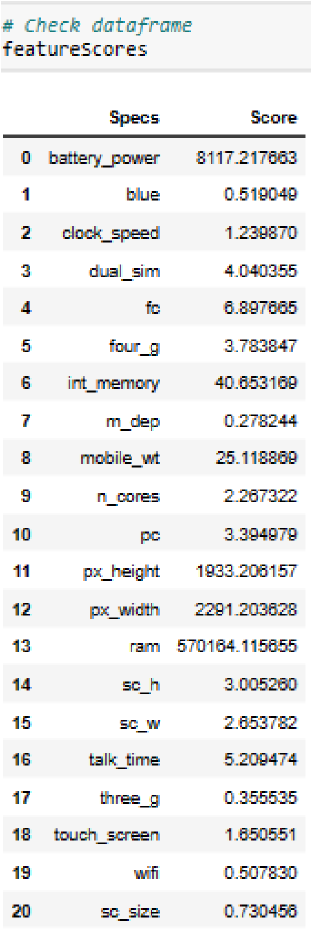


***Fig*.12.** Pixel Height Graph

There are some outliers in the fc and px_height properties as seen in the graphs. Outliers are values that differ from other values in the dataset and distort the distribution. This was determined by the boxplot method as seen above.

3.8. Future Selection

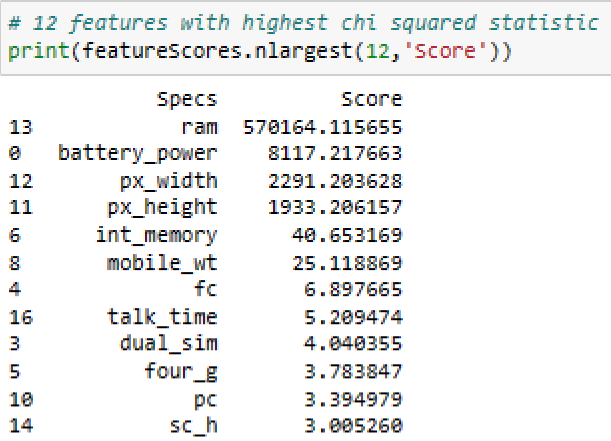
Feature selection was used to select a subset of the features in the dataset used. This technique was used to increase the accuracy of the model while also reducing the size of the data set.



```
# Check dataframe
featureScores
```

	Specs	Score
0	battery_power	8117.217663
1	blue	0.519049
2	clock_speed	1.239870
3	dual_sim	4.040355
4	fc	6.897665
5	four_g	3.783847
6	int_memory	40.653169
7	m_dep	0.278244
8	mobile_wt	25.118869
9	n_cores	2.267322
10	pc	3.394979
11	px_height	1933.206157
12	px_width	2291.203628
13	ram	570164.115655
14	sc_h	3.005260
15	sc_w	2.653782
16	talk_time	5.209474
17	three_g	0.355535
18	touch_screen	1.650551
19	wifi	0.507830
20	sc_size	0.730456

***Fig*.13.** Feature Scores



```
# 12 features with highest chi squared statistic
print(featureScores.nlargest(12,'Score'))
```

```
            Specs          Score
13            ram  570164.115655
0   battery_power    8117.217663
12       px_width    2291.203628
11      px_height    1933.206157
6      int_memory      40.653169
8       mobile_wt      25.118869
4              fc       6.897665
16      talk_time       5.209474
3        dual_sim       4.040355
5          four_g       3.783847
10             pc       3.394979
14           sc_h       3.005260
```

***Fig*.14.** 12 Features

4. Model Selection and Validation

In this project, four different algorithms were used: Logistic Regression, Decision Tree, K-nearest Neighbor Classifier, and Random Forest.

Logistic Regression is a classification algorithm used to predict the probability of data belonging to a class. This algorithm determines an output class based on the features of the data.

Decision Tree is a classification or regression algorithm that creates a decision tree based on the features of the data. This tree is used to classify or predict the data.

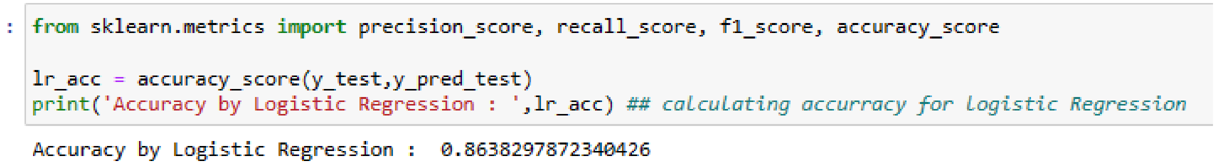
K-nearest Neighbor Classifier is a classification algorithm that looks at the classes of the nearest neighbors to determine the class of a new data point. This algorithm classifies data based on their similarities.

Random Forest is a classification or regression algorithm that consists of many decision trees. This algorithm creates many decision trees based on the features of the data and combines the results of the majority of the trees to produce a final result.

These four algorithms provide different advantages for different datasets and problems, and they were used in this project to produce the best results for classifying or predicting the data.

4.1. Logistic Regression

The multiclass logistic regression model is a popular choice for multiclass classification problems because it is relatively simple to implement and can handle a large number of features. By using this model, we hope to accurately classify the discrete target variables in our dataset and make predictions based on the probabilities of each possible outcome. Overall, we believe that the multiclass logistic regression model is the best approach for analyzing our dataset and achieving our research goals.

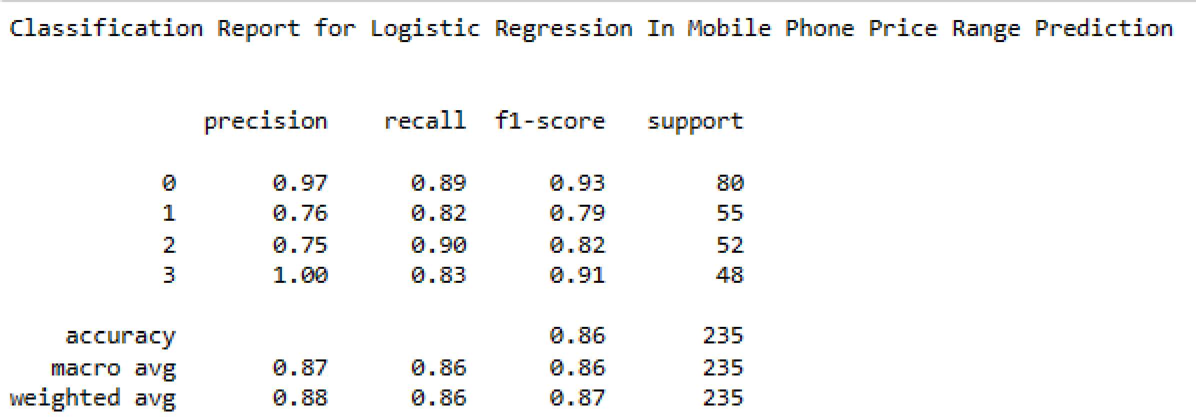


```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

lr_acc = accuracy_score(y_test,y_pred_test)
print('Accuracy by Logistic Regression : ',lr_acc) ## calculating accurracy for Logistic Regression
```

```
Accuracy by Logistic Regression :  0.8638297872340426
```

***Fig*.15.** Accuracy of Logistic Regression

As is known, accuracy is used to measure the success of a model. The value must be between 0 and 1. The accuracy value obtained for the logistic regression model is 0.86 as seen in the figure.



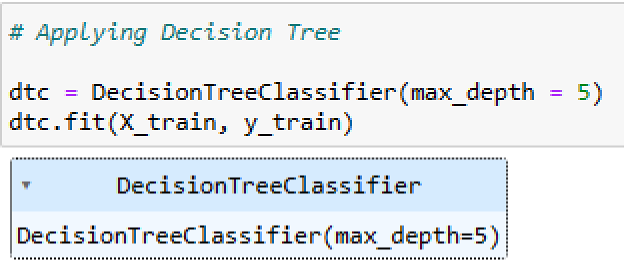
```
Classification Report for Logistic Regression In Mobile Phone Price Range Prediction

              precision    recall  f1-score   support

           0       0.97      0.89      0.93        80
           1       0.76      0.82      0.79        55
           2       0.75      0.90      0.82        52
           3       1.00      0.83      0.91        48

    accuracy                           0.86       235
   macro avg       0.87      0.86      0.86       235
weighted avg       0.88      0.86      0.87       235
```

***Fig*.16.** Classficiation Report

As seen in the Classification report and logistic regression, the accuracy values seem to be compatible with each other.

4.2. Decision Tree

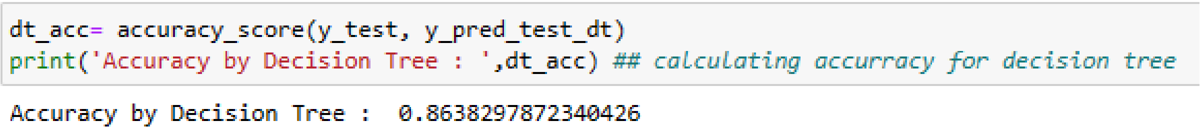
One important aspect of decision trees is that they are easy to interpret and visualize, making them a popular choice for data analysis. Additionally, decision trees can handle both categorical and numerical data, making them a versatile tool for classification and regression problems.



```
# Applying Decision Tree

dtc = DecisionTreeClassifier(max_depth = 5)
dtc.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5)
```

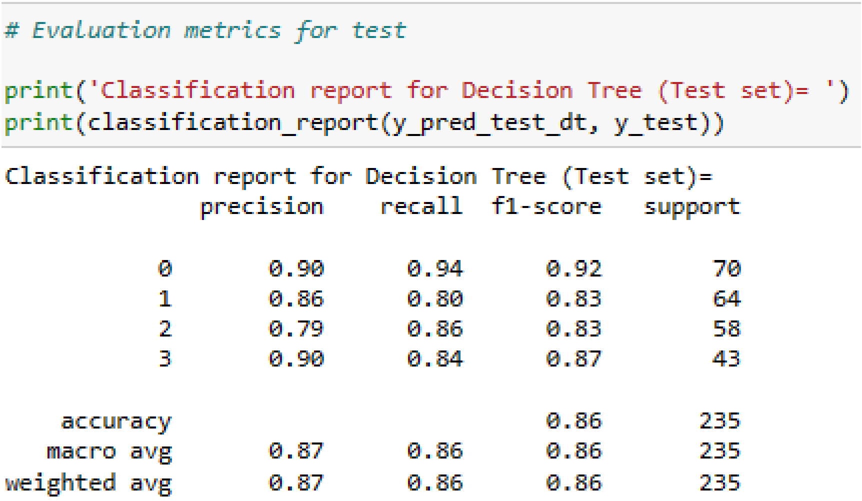
***Fig*.17**. Decision Tree Depth



```
dt_acc= accuracy_score(y_test, y_pred_test_dt)
print('Accuracy by Decision Tree : ',dt_acc) ## calculating accurracy for decision tree
```

```
Accuracy by Decision Tree :  0.8638297872340426
```

***Fig*.18.** Accuracy by Decision Tree

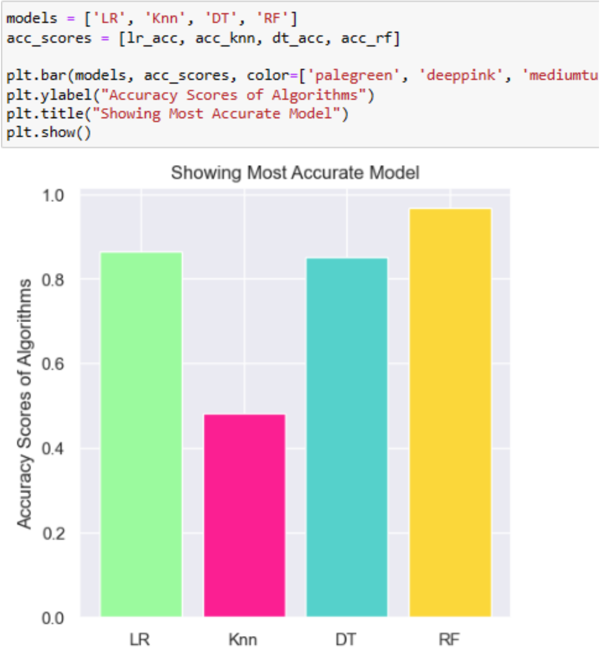


```
# Evaluation metrics for test

print('Classification report for Decision Tree (Test set)= ')
print(classification_report(y_pred_test_dt, y_test))
```

```
Classification report for Decision Tree (Test set)=
              precision    recall  f1-score   support

           0       0.90      0.94      0.92        70
           1       0.86      0.80      0.83        64
           2       0.79      0.86      0.83        58
           3       0.90      0.84      0.87        43

    accuracy                           0.86       235
   macro avg       0.87      0.86      0.86       235
weighted avg       0.87      0.86      0.86       235
```

***Fig*.19.** Classification Report for Decision Tree

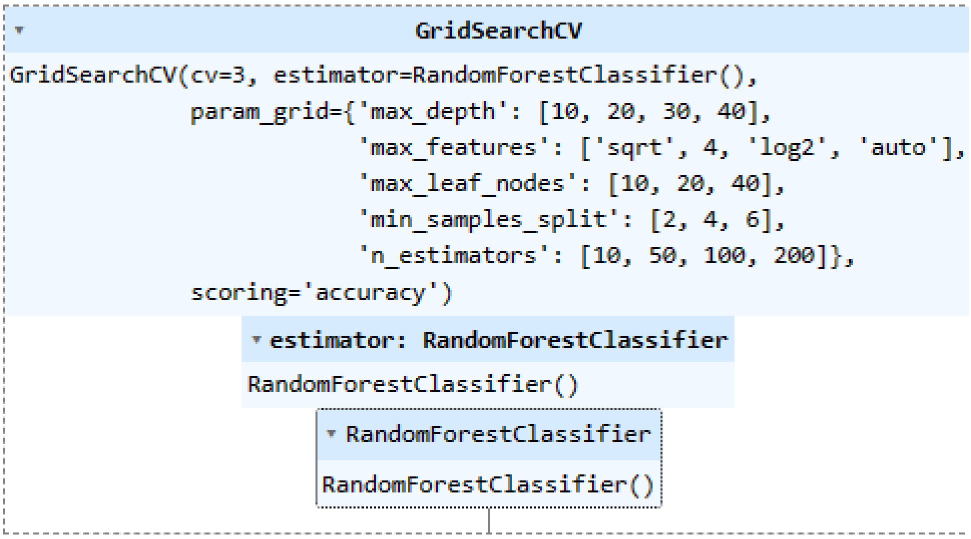
As is known, decision tree is used to measure the success of a model. The value must be between 0 and 1. The accuracy value obtained for the logistic regression model is 0.86 as seen in the figure. As seen in the Classification report and logistic regression, the accuracy values seem to be compatible with each other.



```
models = ['LR', 'Knn', 'DT', 'RF']
acc_scores = [lr_acc, acc_knn, dt_acc, acc_rf]

plt.bar(models, acc_scores, color=['palegreen', 'deeppink', 'mediumtu
plt.ylabel("Accuracy Scores of Algorithms")
plt.title("Showing Most Accurate Model")
plt.show()
```

Fig.20. Best Model

As can be seen from this graph, it is understood that the best model for the dataset is the rainforest algorithm.

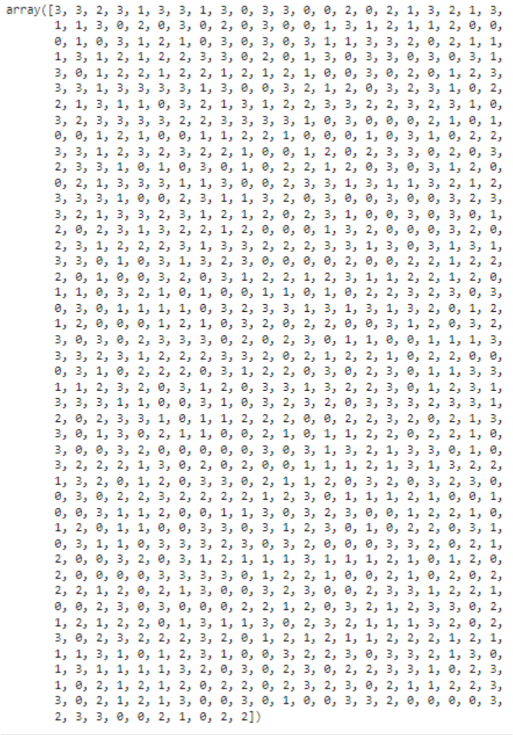


```
GridSearchCV
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [10, 20, 30, 40],
                         'max_features': ['sqrt', 4, 'log2', 'auto'],
                         'max_leaf_nodes': [10, 20, 40],
                         'min_samples_split': [2, 4, 6],
                         'n_estimators': [10, 50, 100, 200]},
             scoring='accuracy')
estimator: RandomForestClassifier
RandomForestClassifier()
RandomForestClassifier
RandomForestClassifier()
```

Fig.21. Hyper Parameter Tuning

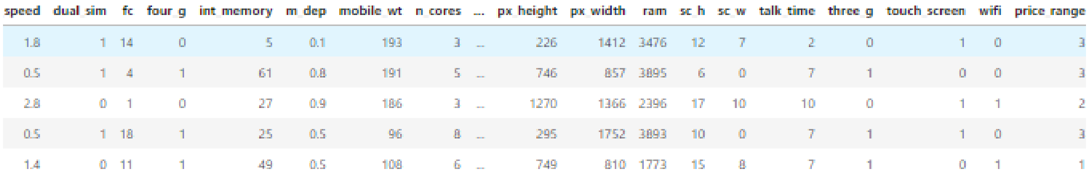
5. Results

Predicted price ranges are shown in the figure below.



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

Fig.22. Predicted Price Rnges



speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
1.8	1	14	0	5	0.1	193	3	...	226	1412	3476	12	7	2	0	1	0	3
0.5	1	4	1	61	0.8	191	5	...	746	857	3895	6	0	7	1	0	0	3
2.8	0	1	0	27	0.9	186	3	...	1270	1366	2396	17	10	10	0	1	1	2
0.5	1	18	1	25	0.5	96	8	...	295	1752	3893	10	0	7	1	1	0	3
1.4	0	11	1	49	0.5	108	6	...	749	810	1773	15	8	7	1	0	1	1

Fig.23. Predicted Price Ranges

The figure above is the predicted version of the price range in the leather set.

As a result, a newly established phone company can learn the estimated price range of the features of the phones in the market by using this project.

6. Future Works

One potential area for future work is to explore the use of more advanced machine learning algorithms, such as deep learning models, to improve the accuracy of the price range predictions. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promising results in various fields, including image and speech recognition, and may be able to capture more complex patterns and relationships in the mobile phone data. However, these models require a large amount of data and computational resources, and may be more difficult to interpret and explain than traditional machine learning models.

Another potential area for improvement is to collect more data and features related to mobile phones, such as camera quality, battery life, and screen size, to further enhance the predictive power of the model. This could involve scraping data from additional sources, such as user reviews or manufacturer specifications, or conducting surveys or experiments to collect more detailed information. Additionally, it may be useful to preprocess the data more carefully, such as by handling missing values or outliers, or by transforming the features to better capture their relationships with the target variable. By improving the quality and quantity of the data, we can potentially improve the accuracy and robustness of the model, and make more accurate predictions for mobile phone price ranges.

7. References

Al-Gbury, O., & Kurnaz, S. (2020). Real Estate Price Range Prediction Using Artificial Neural Network and Grey Wolf Optimizer. *4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings*, 7–11. https://doi.org/10.1109/ISMSIT50672.2020.9254972

Arora, P., Srivastava, S., & Garg, B. (2020). Mobile Price Prediction Using Weka. *IJSDR2004057 International Journal of Scientific Development and Research*, *5*(4), 330–333. www.ijsdr.org

Asim, M., & Khan, Z. (2018). Mobile Price Class prediction using Machine Learning Techniques. *International Journal of Computer Applications*, *179*(29), 6–11. https://doi.org/10.5120/ijca2018916555

Bukvić, L., Pašagić Škrinjar, J., Fratrović, T., & Abramović, B. (2022). Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning. *Sustainability (Switzerland)*, *14*(24). https://doi.org/10.3390/su142417034

Ercüment GÜVENÇ, ÇETİN, G., & KOÇAK, H. (2021). Comparison of KNN and DNN Classifiers Performance in Predicting Mobile Phone Price Ranges. *Advances in Artificial Intelligence Research*, *1*(1), 19–28. www.dergipark.com/aair/

Kumuda, Karur, V., & S E., K. B. (2021). Prediction of Mobile Model Price us ing Machine Learning Techniques. *International Journal of Engineering and Advanced Technology*, *11*(1), 273–275. https://doi.org/10.35940/ijeat.a3219.1011121

Mitta, S. (2021). *MOBILE PRICE PREDICTION USING FEATURE SELECTION AND CLASSIFIER ALGORITHMS OF MACHINE LEARNING*. *06*, 1040–1044.

Radhamani, V., Manju, D., Bobby, P. M., Javagar, M., Nivetha, V., & Rinubha, P. (2022). Gold Price Prediction Using Ml Algorithms. *Ymer*, *21*(7), 183–192.

Sakib, A. H. (2020). *Predicting Mobile Price Range Using Classification Techniques*. http://dspace.daffodilvarsity.edu.bd:8080/handle/123456789/5685%0Ahttp://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5685/171-35-1838 %2813_%29.pdf?sequence=1&isAllowed=y

Varun Kiran, A. (2022). Prediction of Mobile Phone Price Class using Supervised Machine Learning Techniques. *International Journal of Innovative Science and Research Technology*, *7*(1), 248–251. www.ijisrt.com248

Zehtab-Salmasi, A., Feizi-Derakhshi, A. R., Nikzad-Khasmakhi, N., Asgari-Chenaghlu, M., & Nabipour, S. (2021). Multimodal Price Prediction. *Annals of Data Science*, 0–2. https://doi.org/10.1007/s40745-021-00326-z