Capstone Project

Mobile Price Range Prediction

Team members:

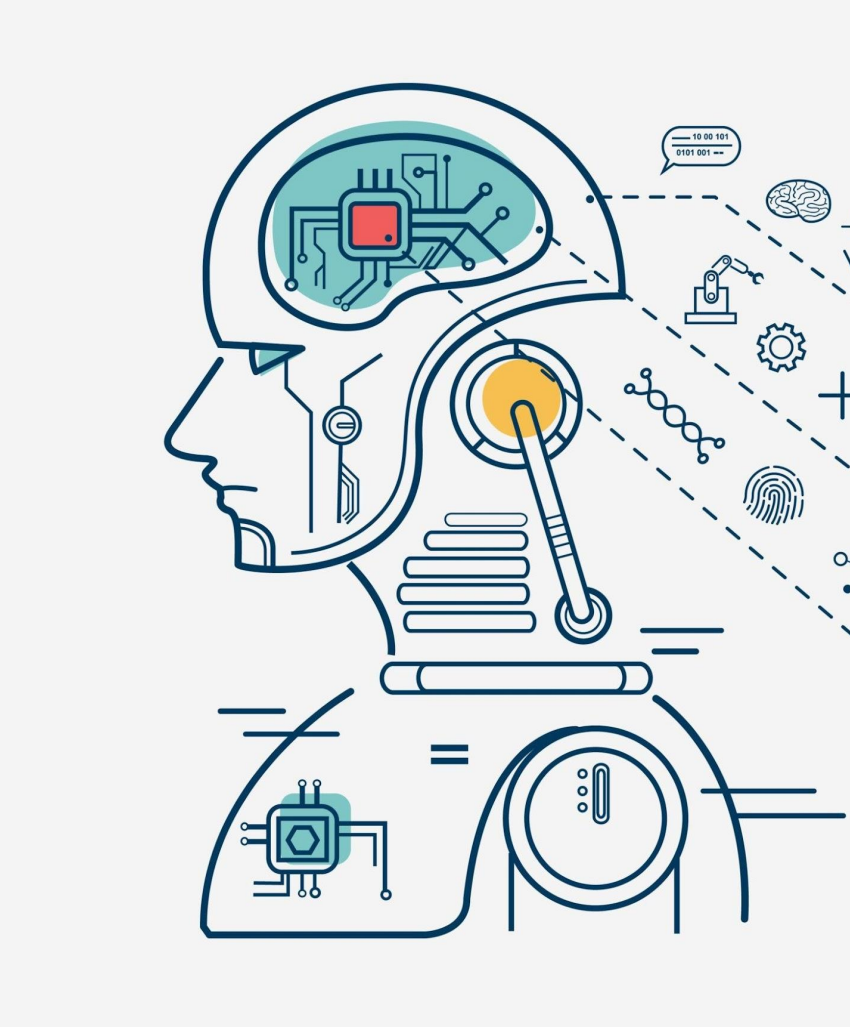
Prince Jain

Rishabh Patidar

Vikas Shrivas

Content

- **Problem Statement / Description**
- **Introduction**
- **Data preparation & cleaning**
- **EDA and Data Processing**
- **ML Model – Classification**
  - K-Nearest Neighbors.
  - Support Vector Machines.
  - Decision Tree Classifiers/Random Forests.
  - Naive Bayes.
  - Logistic Regression
- **Conclusion**



Problem Statement

Companies in the mobile phone market want to understand sales data and factors that drive prices. The objective is to find out some relation between features of a mobile phone and its selling price. In this problem, we do not have to predict the actual price but a price range indicating how high the price is.

Data Description

Data Description -

Battery_power - Total energy a battery can store in one time measured in mAh

Blue - Has bluetooth or not

Clock_speed - speed at which microprocessor executes instructions

Dual_sim - Has dual sim support or not

Fc - Front Camera mega pixels

Four_g - Has 4G or not

Int_memory - Internal Memory in Gigabytes

M_dep - Mobile Depth in cm

Mobile_wt - Weight of mobile phone

N_cores - Number of cores of processor

Pc - Primary Camera mega pixels

Px_height - Pixel Resolution Height

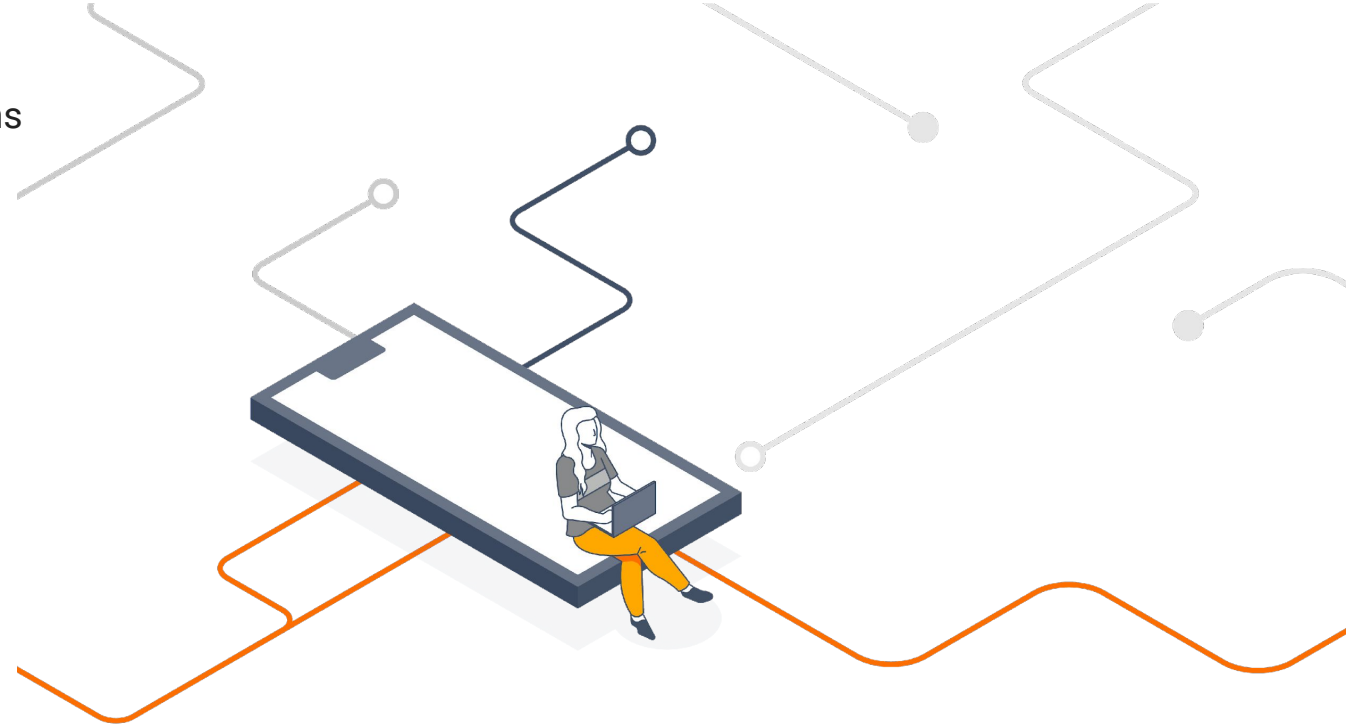
Px_width - Pixel Resolution Width

Ram - Random Access Memory in Mega Bytes

Sc_h - Screen Height of mobile in cm

Sc_w - Screen Width of mobile in cm

Talk_time - longest time that a single battery charge will last when you are



Introduction

Mobile phones come in all sorts of prices, features, specifications and all. Price estimation and prediction is an important part of consumer strategy. Deciding on the correct price of a product is very important for the market success of a product. A new product that has to be launched, must have the correct price so that consumers find it appropriate to buy the product.
The data contains information regarding mobile phone features, specifications etc and their price range. The various features and information can be used to predict the price range of a mobile phone.

In the competitive mobile phone market companies want to understand sales data of mobile phones and the factors which drive the prices. The objective is to find out some relation between the features of a mobile phone(eg:- RAM, Internal Memory, etc) and its selling price. In this problem, we do not have to predict the actual price but a price range indicating how high the price is.

Data preparation & cleaning

Import relevent data

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# mounting our drive path so that access the data
file_path= "/content/drive/MyDrive/data_mobile_price_range.csv"
df= pd.read_csv(file_path)
```

```
df.head()
```

Here we see that the data consist of 21 column and 2000 rows.

```
df.info()
```

```
<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
```

Now, we remove the data points with missing data.

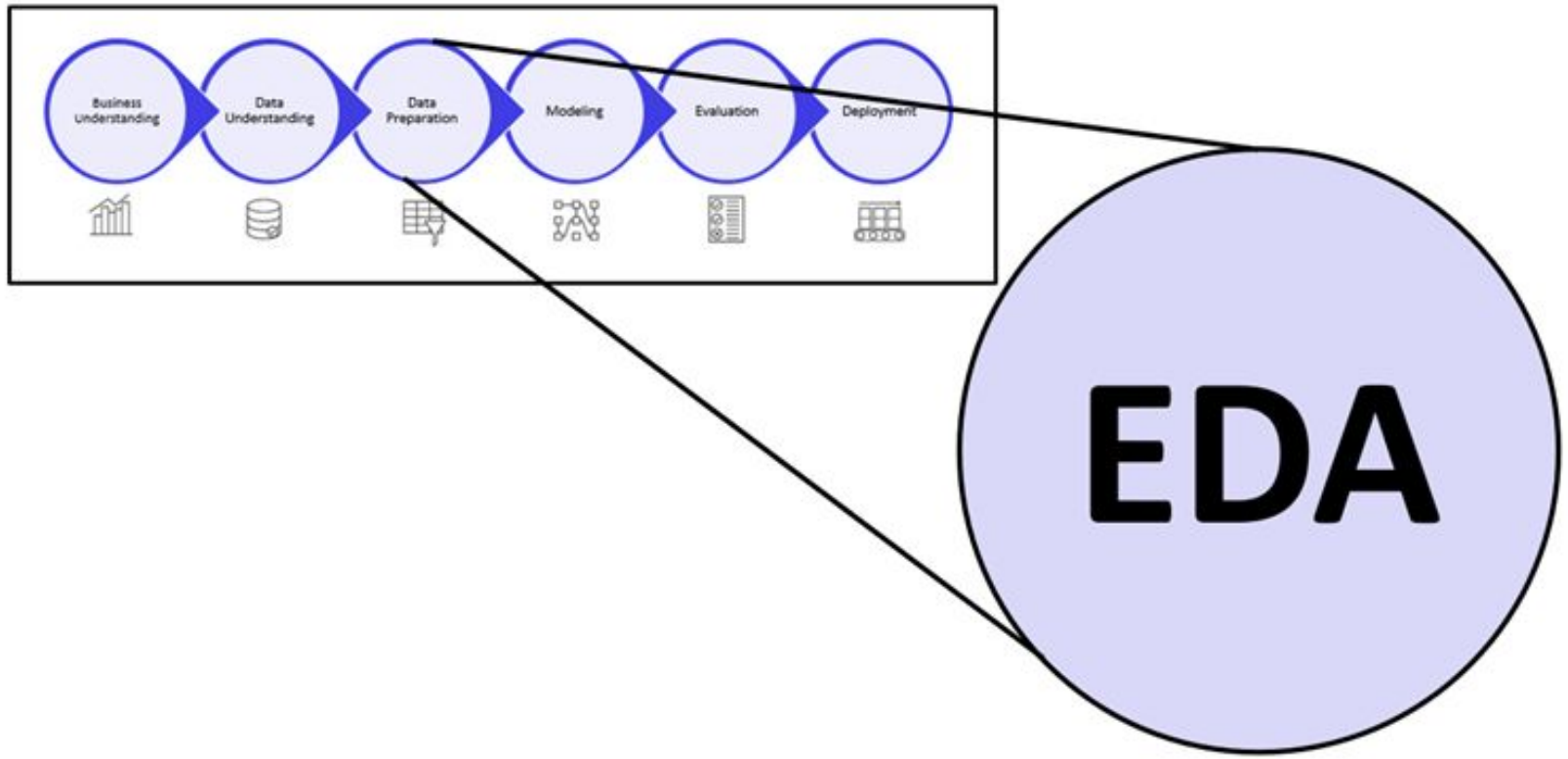
Like :- `data_f = df[df['sc_w'] != 0]`

After removing the missing values the data consist of 21 columns and 1820 rows

```
new_df.info()
```

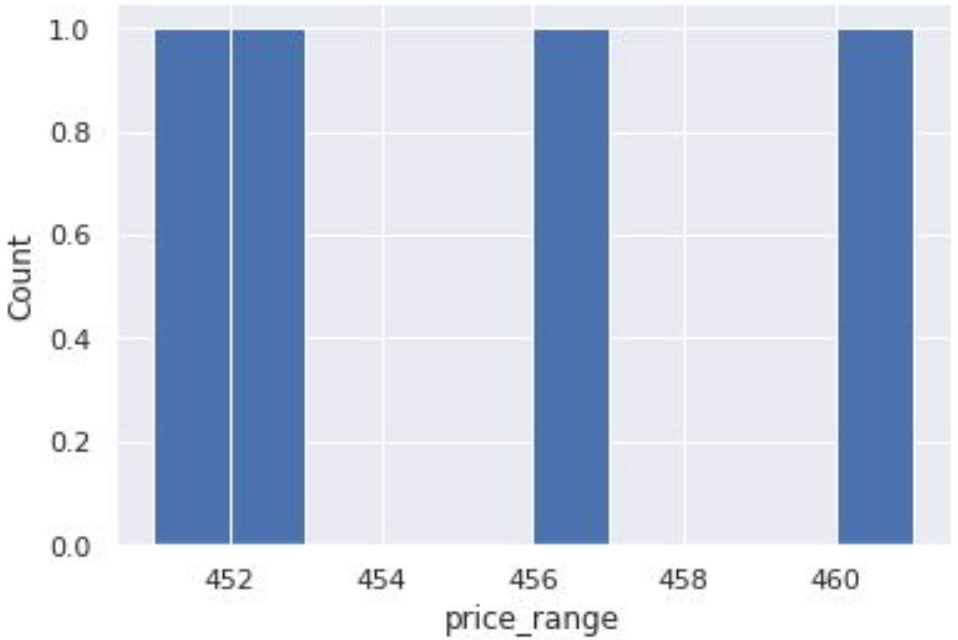
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1820 entries, 0 to 1999
Data columns (total 21 columns):
```

EDA and Data Processing



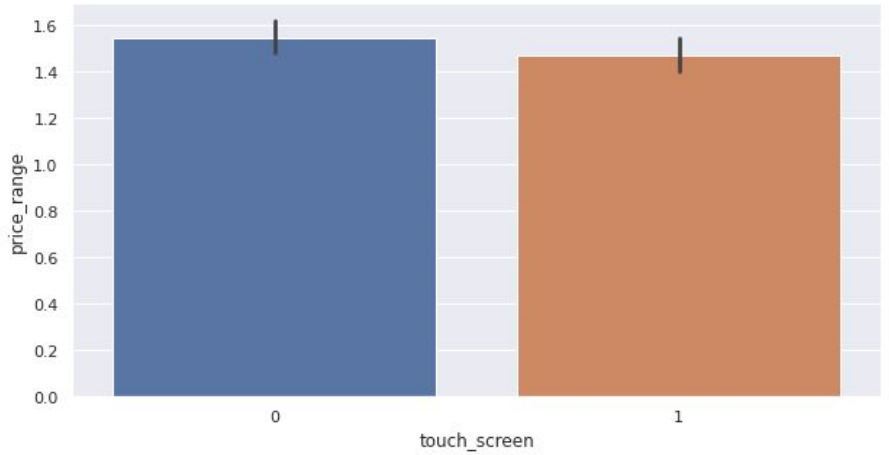
DATA PROCESSING

Price Segments



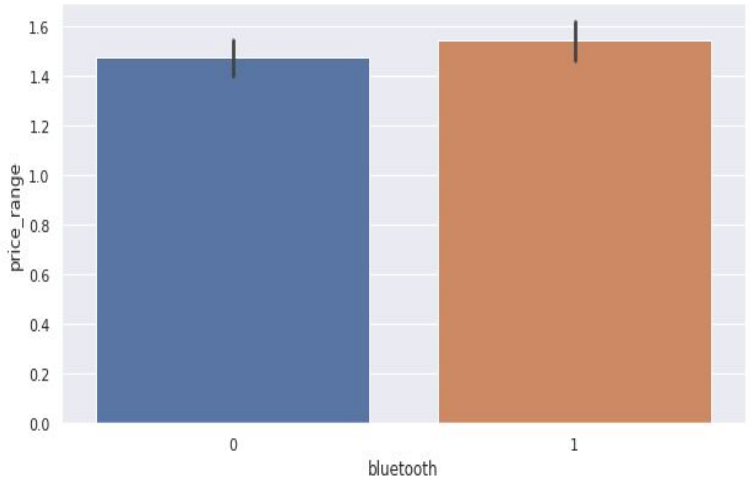
There are four price ranges for mobile phones. The number of elements is almost identical between the categories.

Screen types



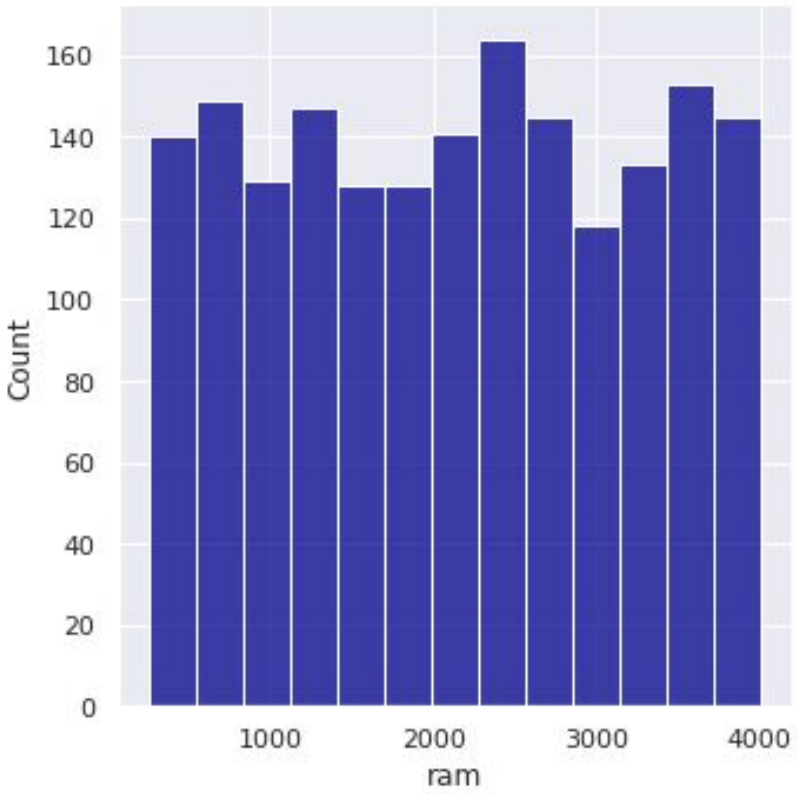
We see that half of all mobile phones have touch-screen features and the other half do not.

Bluetooth



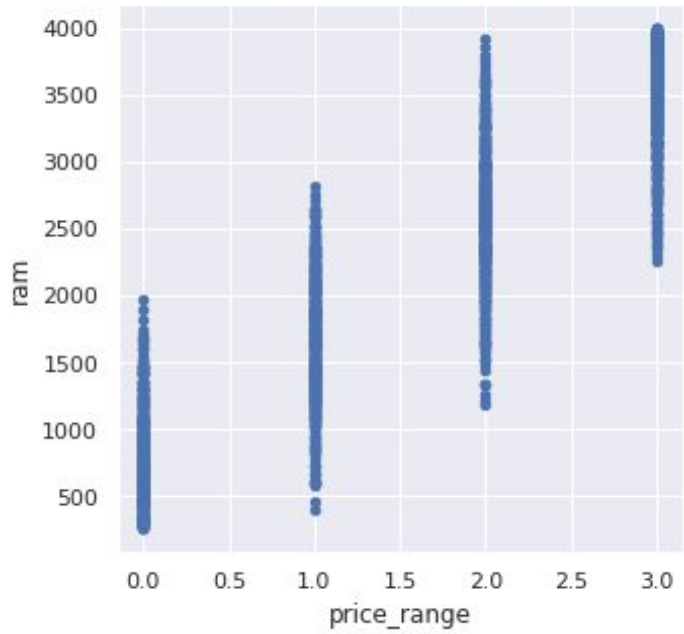
The survey found that half of the devices have Bluetooth, while half do not.

RAM



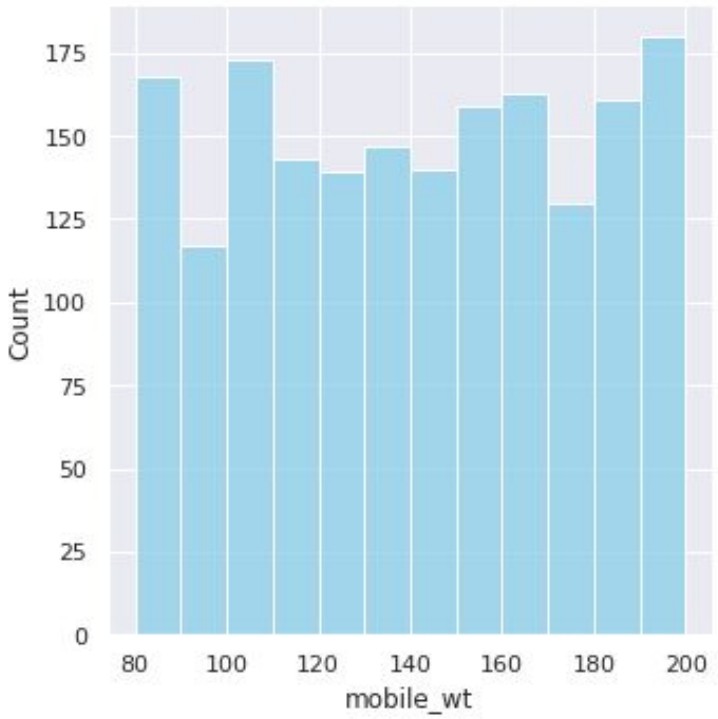
Ram varies from 256 MB to 4Gb, as shown in the chart above.

Random Access Memory in Mega Bytes



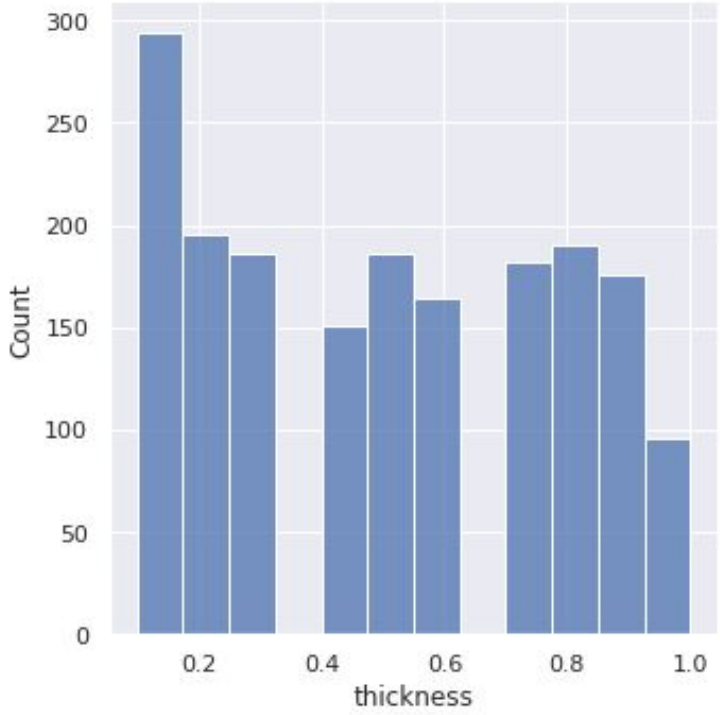
In this scatter chart we see that there are four price segments with increased ram and increased price of mobile phones.

Mobile weight



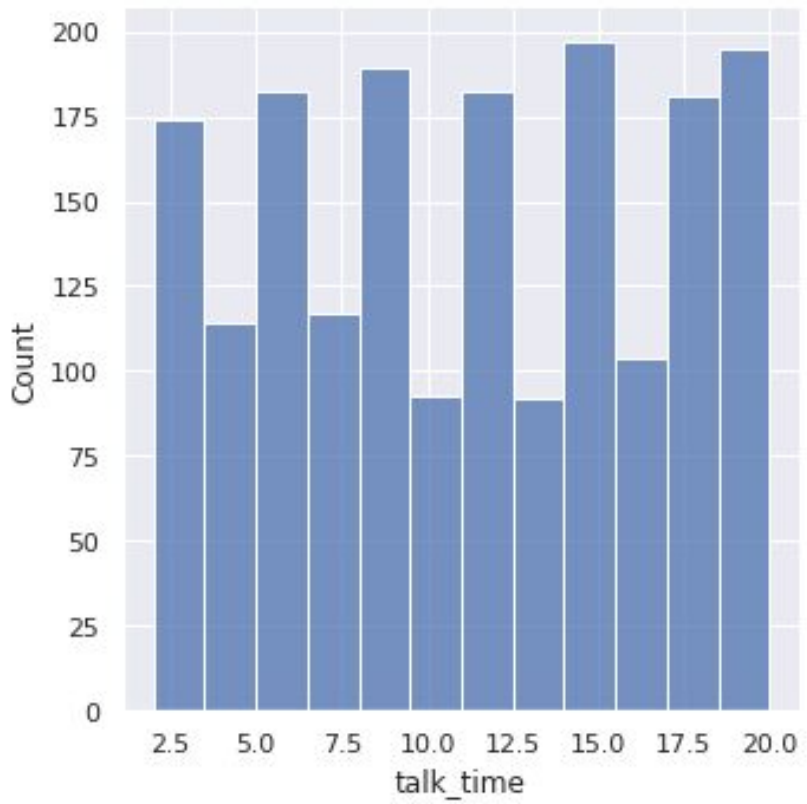
The data show that the average weight of mobile phones is over 80 grams and their maximum weight is less than 200 grams.

Thickness(in cm).



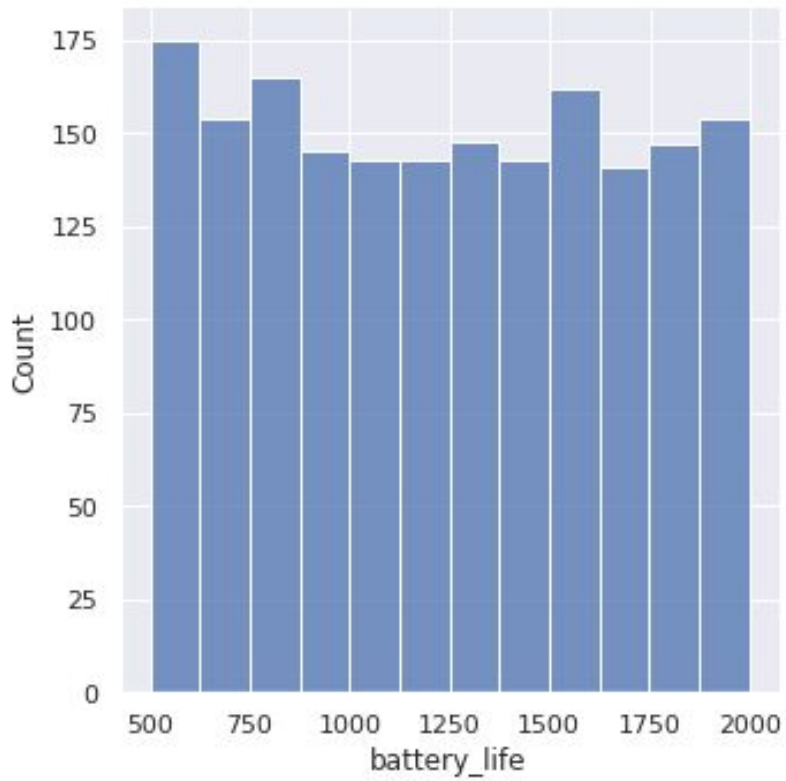
A few mobiles are very thin and some are almost a centimeter thick.

Talktime



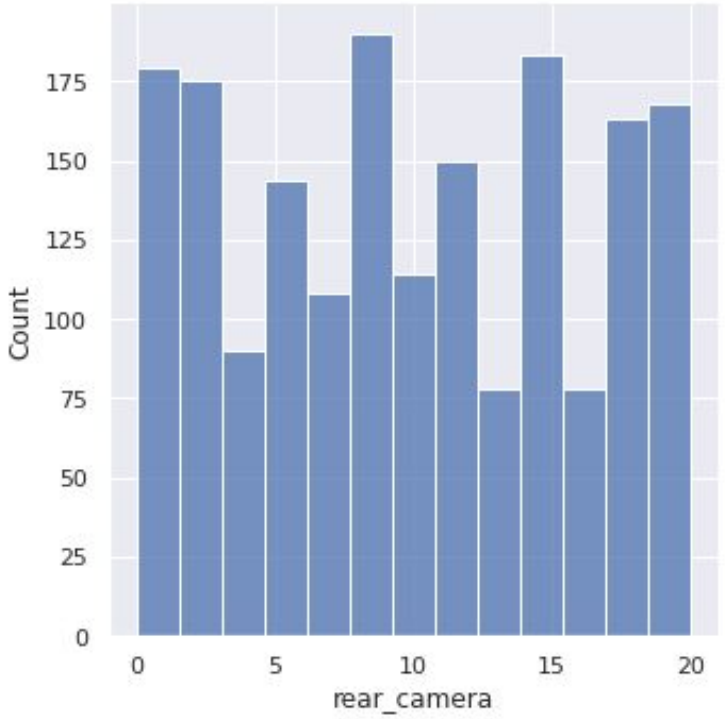
The bar chart above shows the range of talk time among the phones. The lowest range is 2.5 hours, and the longest is 20 hours.

Battery in MAH



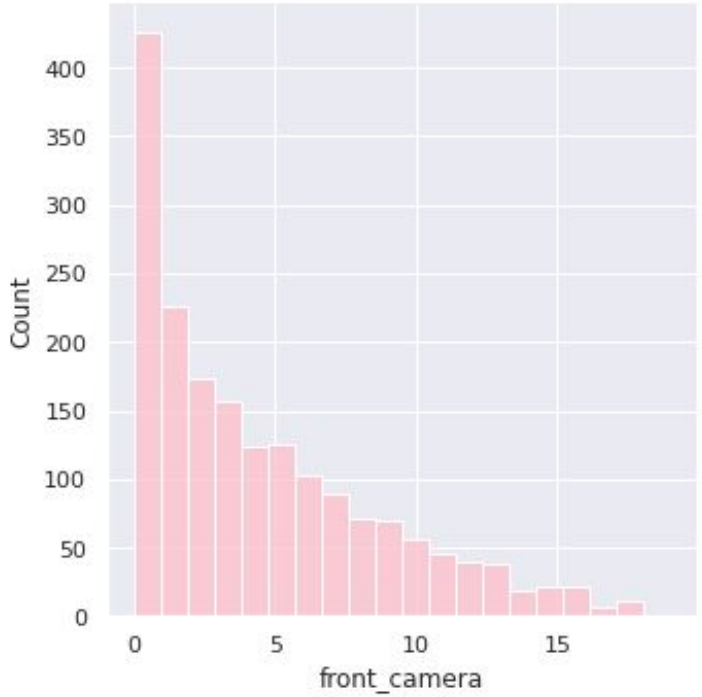
The battery life of a mobile phone typically ranges from 500 to 2000 mAh.The most frequently purchased mobile phone with a 500 mAh battery is the most popular.

PC (Primary camera Megapixels)



In this bar chart, we found that some mobile cameras do not contain a camera, and show zero. We also saw that the maximum mobile camera is 8MP and after it is 13MP.

Front Camera

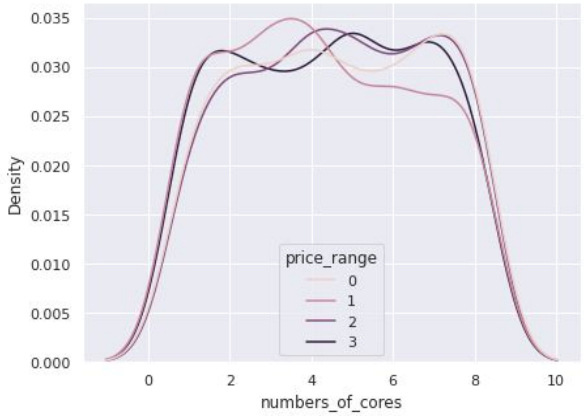


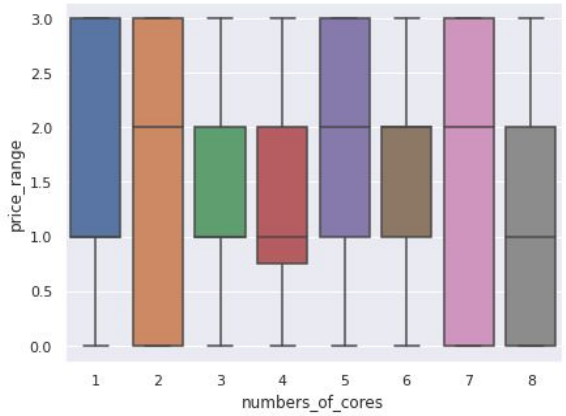
The majority of phones do not contain a front-facing camera, and the maximum number of phones currently on sale contains a 2mp camera.

Discuss various parameters and their relationship to price

This discussion will focus on some of the differences between parameters and prices.

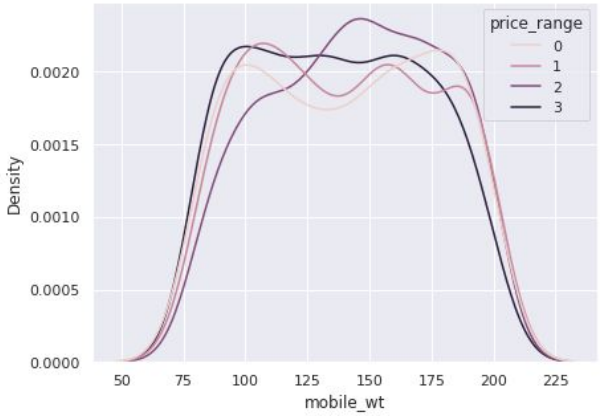
Numbers of core vs price

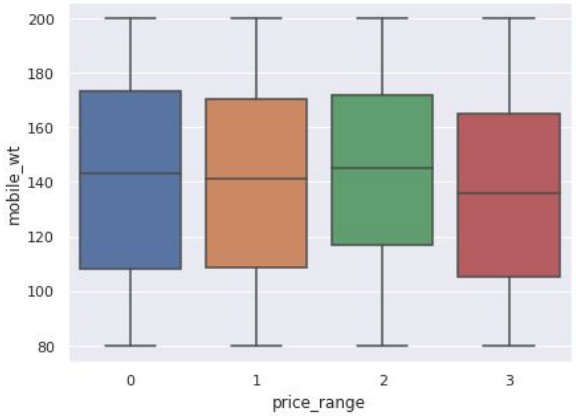




The above chart shows the number of core 2 and 7 available in the price range 0 to 3, whereas we saw that the number of course 8 is not available at a high price, and core number 1 is not available at a low price.

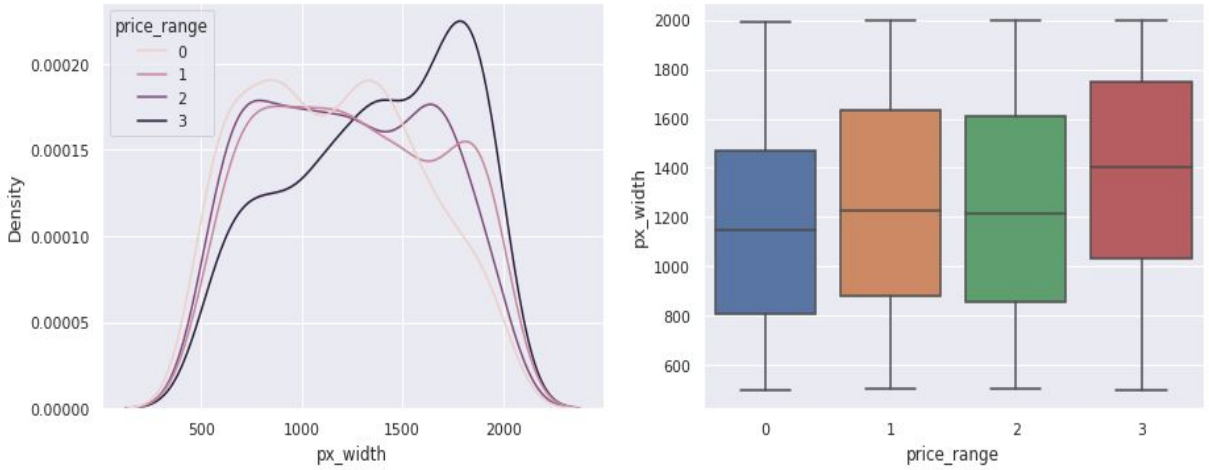
Weight vs price





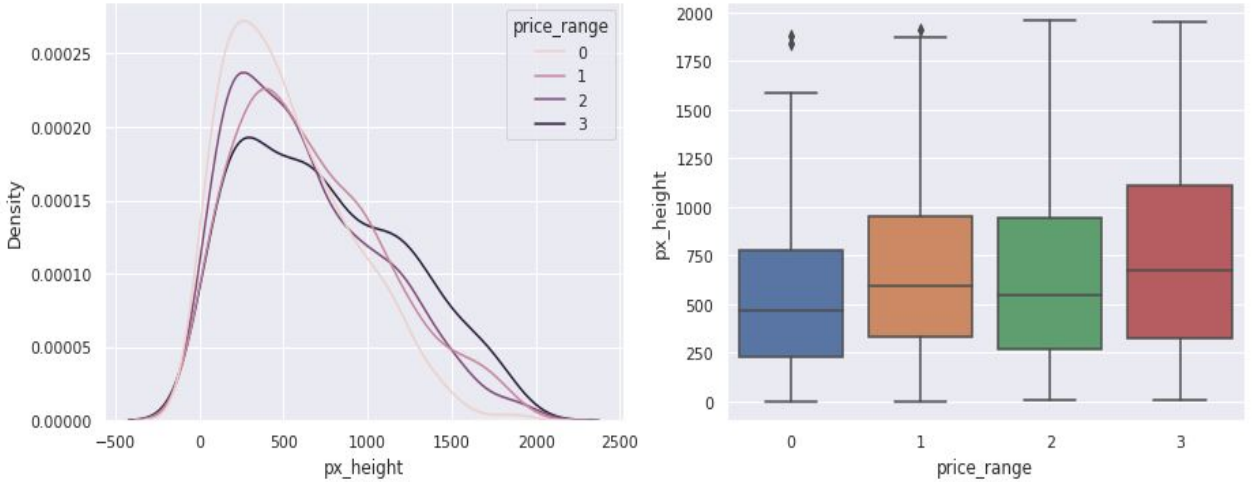
It is observed that the cost of a phone is directly proportional to its weight. The price of a cell phone rises as the weight of the phone decreases.

Pixel Resolution Width



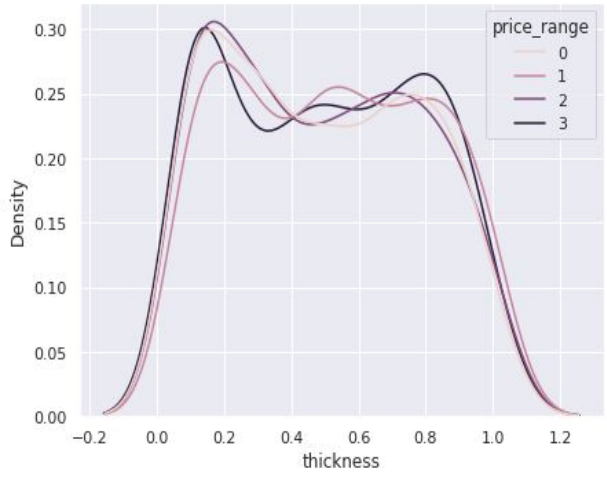
As we move from Low cost to Very high cost, the pixel widths of mobiles do not increase in absolute terms. However, mobile with 'Medium cost' and 'High cost' has almost equal pixel widths so we can say that it would be a driving factor in deciding price_range.

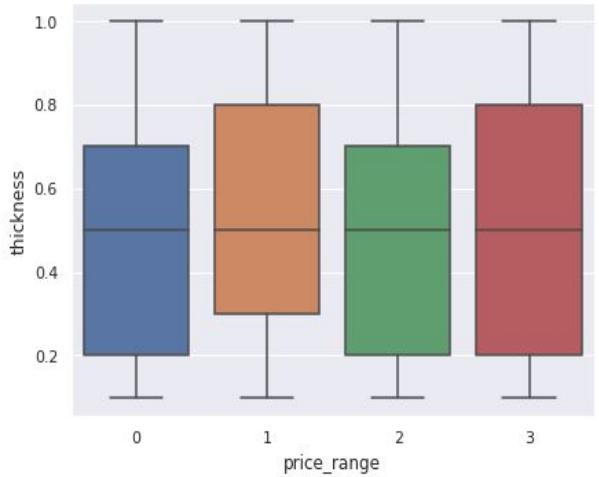
Pixel Resolution height



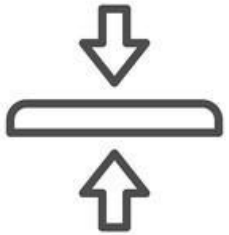
Pixel height, as we move from low cost to very high cost, remains relatively consistent.

Thickness vs price

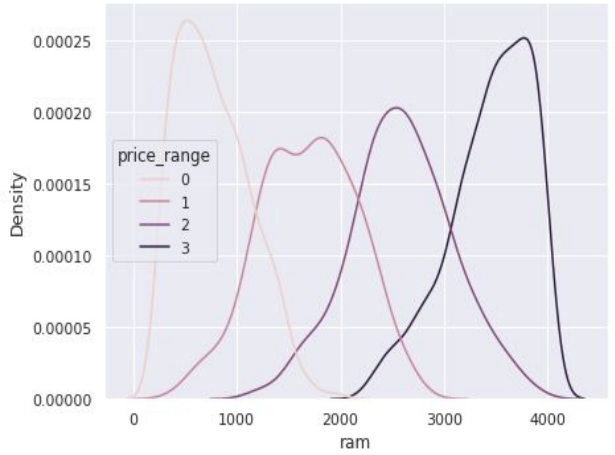


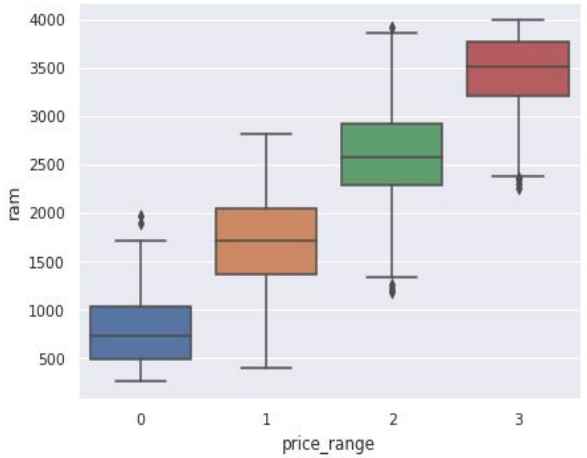


Thick phones are available at the lowest price. The thickness of mobiles ranges from 0.2 cm to almost one centimeter thick. The cost of thick phones is low or may be high.



Ram vs price

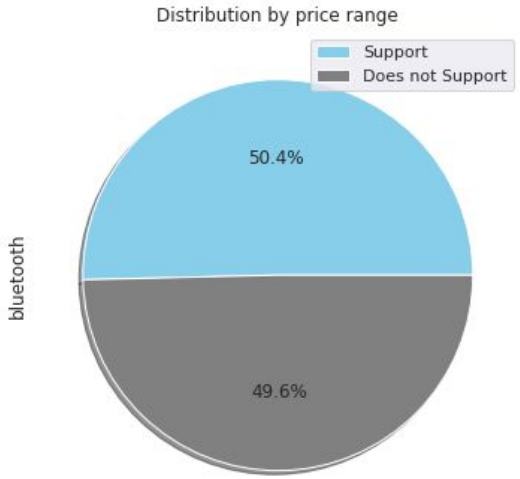


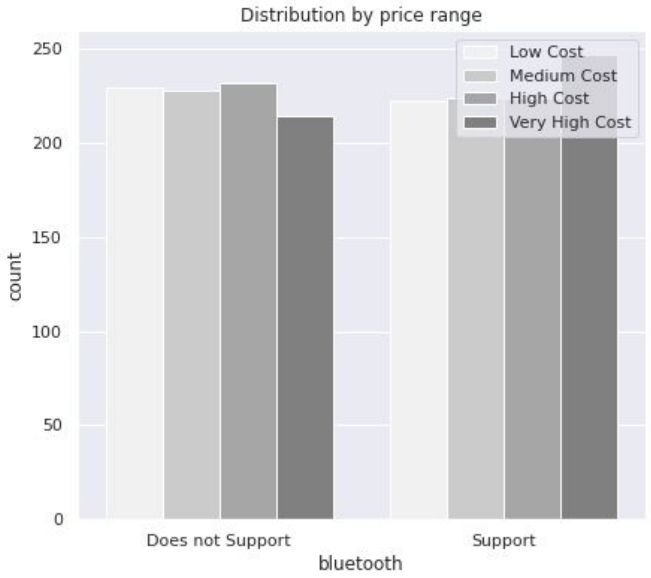


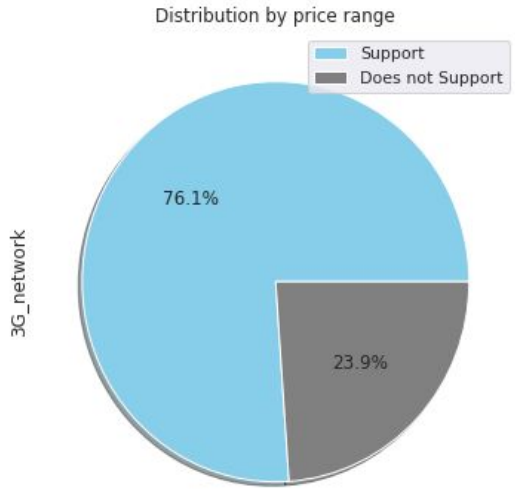
The higher the RAM, the more expensive a smartphone is likely to be.

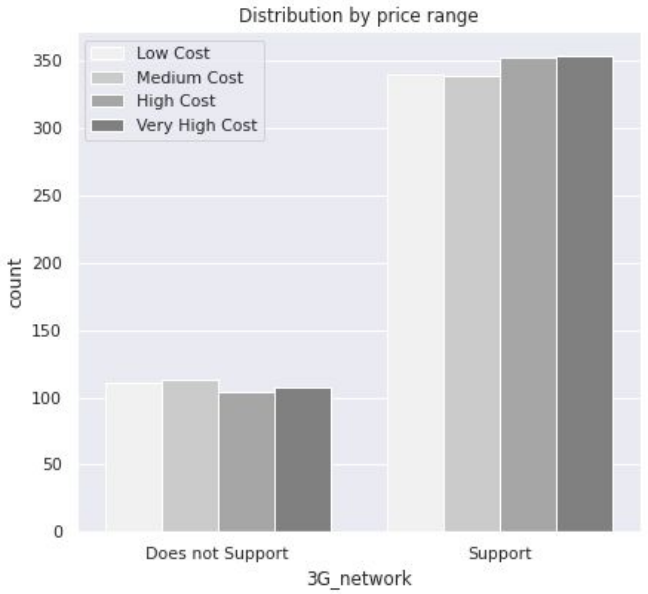
RAM ∝ MOBILE PRICE

Connectivity

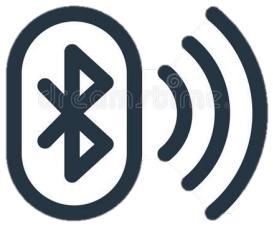


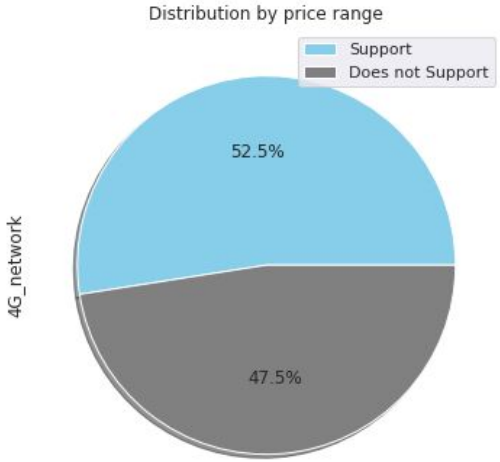


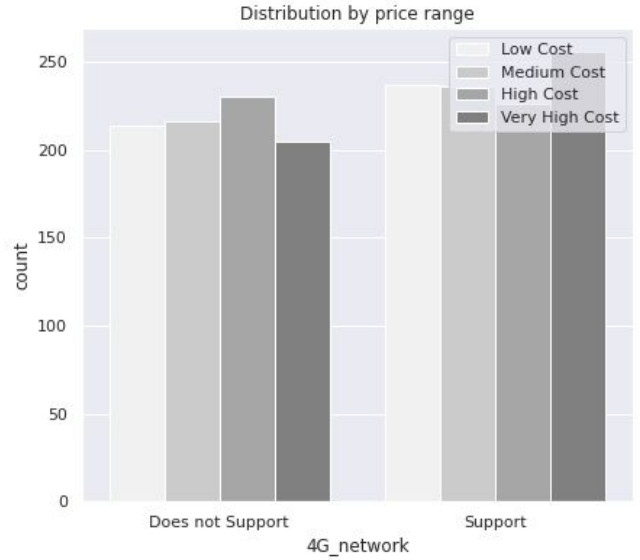




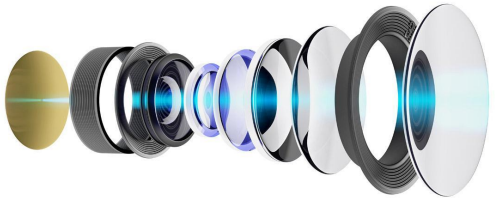
A study by **Mobile Price Range Prediction** has shown that 76.1% of mobile phone users have 3G connectivity, while 23.9% do not support 3G. There are 52.5% of mobile phones with 4G connectivity and 43.5% do not support 4G. According to TRAI data, 50.4% of the users have Bluetooth connectivity and 49.6% do not have it. All these mobile phones are expensive, though.





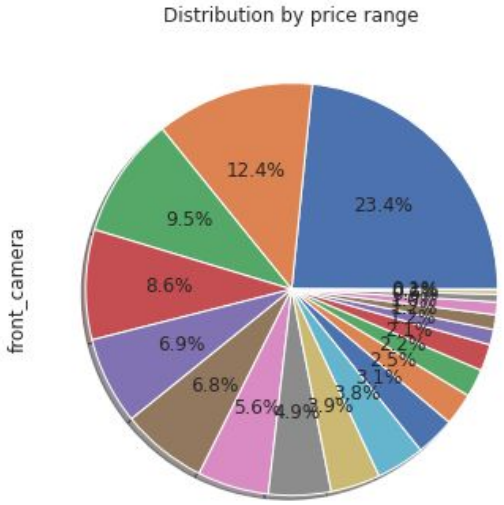


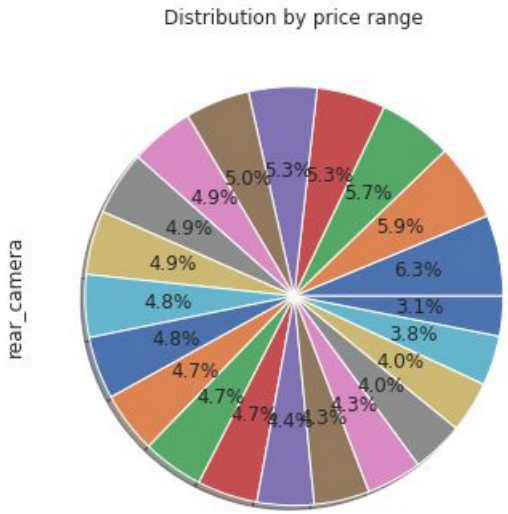
Camera and Price

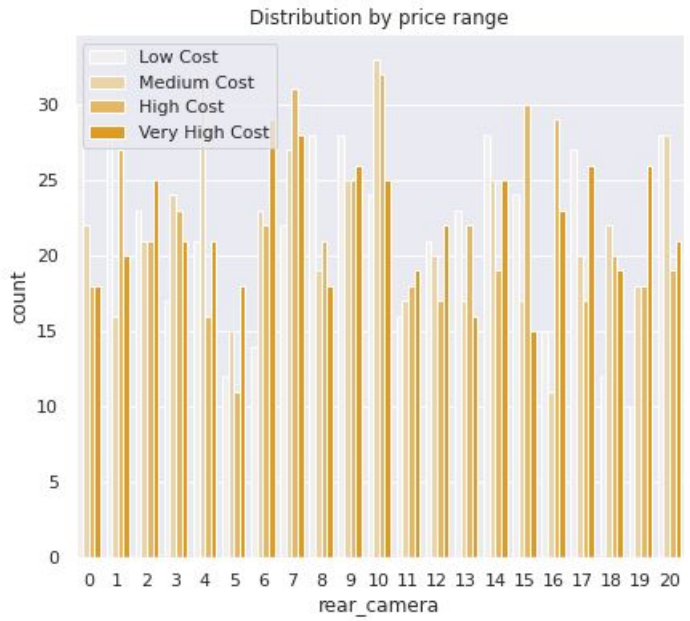


In this distribution, we can see that 23.4% of mobiles do not have a front camera. This is the highest percentage and there are 19 types of front cameras.

If we look at rear cameras, there are 20 different models, with a maximum percentage of 6.3%. The price range for these mobiles is from 0 to 4

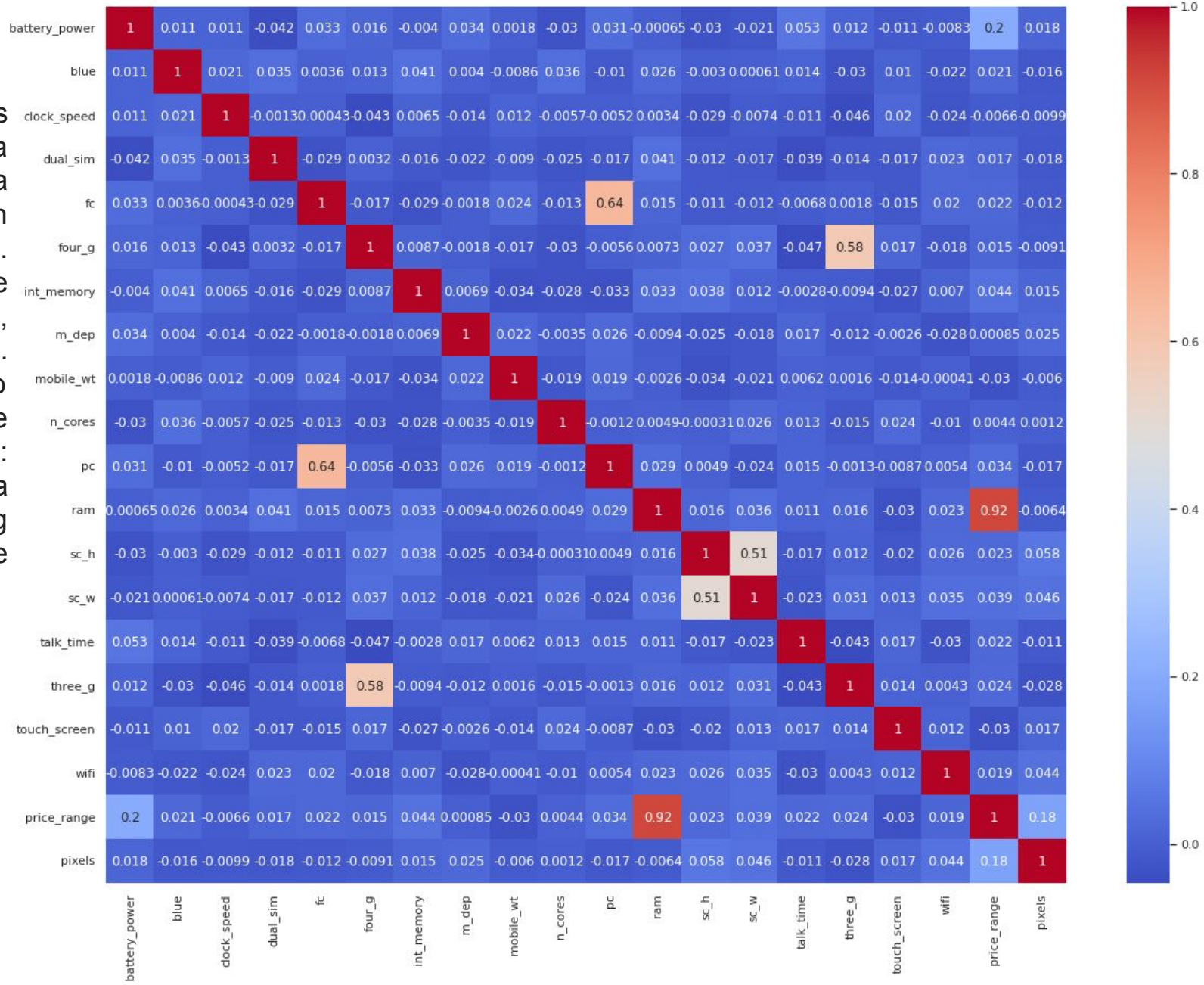






Heat map

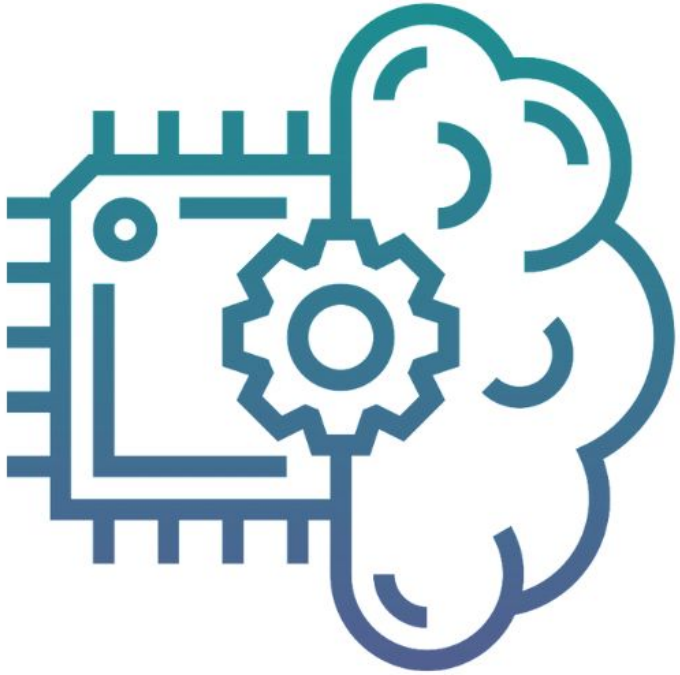
RAM and price_range are highly correlated, as expected. This signifies that RAM will play a significant role in predicting the price range of a smartphone. There is some collinearity between feature pairs ('pc', 'fc') and ('px_width', 'px_height'). Both correlations are justified because there are good chances that if front camera quality is good, back camera quality would be good too. Additionally, if px_height increases, pixel width also increases, which means overall screen pixels. We can replace these two features with one feature: Front Camera megapixels and Primary Camera megapixels are different entities despite being correlated; therefore we will keep them as they are for now.



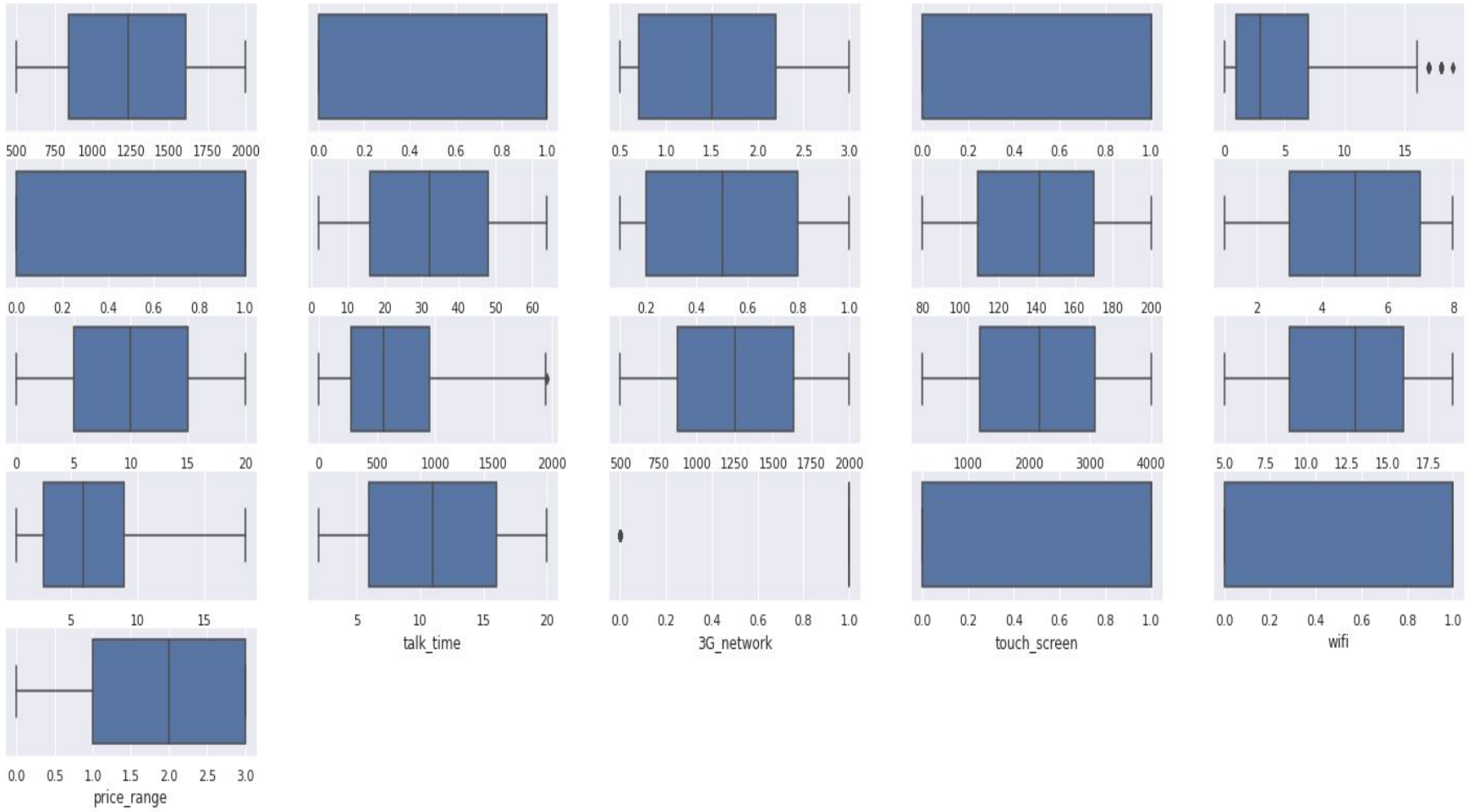
Machine Learning Model – Classification

Types of Classifiers Used :

K-Nearest Neighbors.
Support Vector Machines.
Decision Tree Classifiers/Random Forests.
Naive Bayes.
Logistic Regression.



looking for outliers using box plot

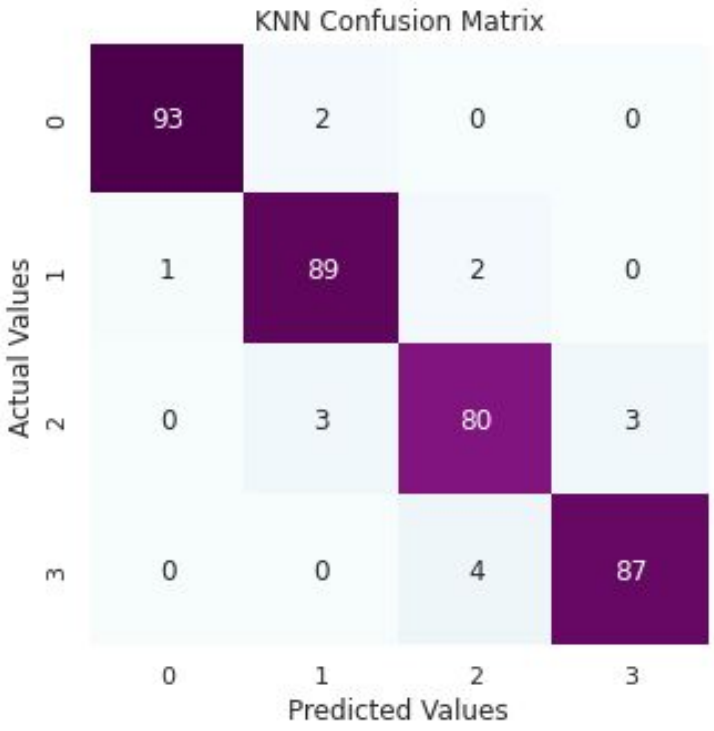


missing values or missing data

```
X=data_f.drop(['price_range'], axis=1)
y=data_f['price_range']
#missing values
X.isna().any()
```

```
battery_life        False
bluetooth           False
clock_speed         False
dual_sim            False
front_camera        False
4G_network          False
int_memory          False
thickness           False
mobile_wt           False
numbers_of_cores    False
rear_camera         False
px_height           False
px_width            False
ram                 False
sc_h                False
sc_w                False
talk_time           False
3G_network          False
touch_screen        False
wifi                False
dtype: bool
```

K-Nearest Neighbors



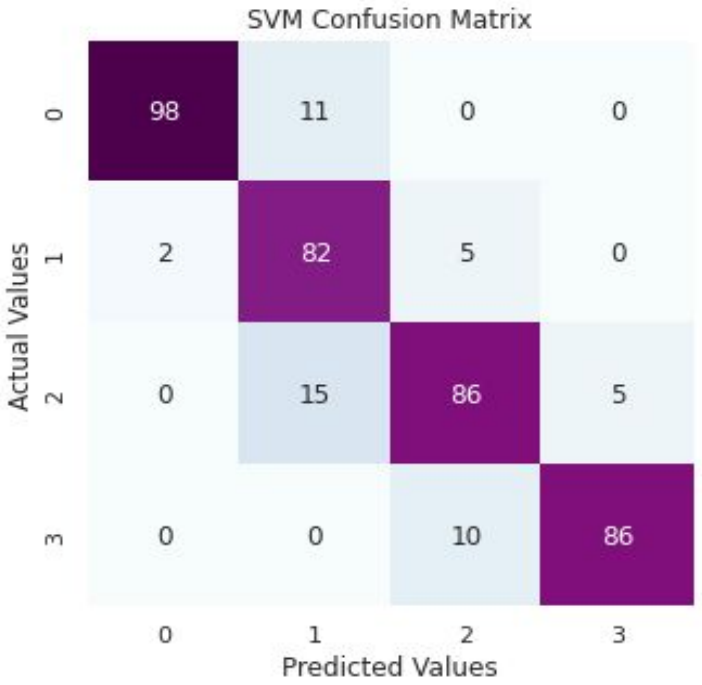
Accuracy_ Score : 95.87%

whereas the kNN method was to produce a accuracy score is 95.87 **%**

KNN Classifier Accuracy Score: 0.9587912087912088

	precision	recall	f1-score	support
0	0.99	0.98	0.98	95
1	0.95	0.97	0.96	92
2	0.93	0.93	0.93	86
3	0.97	0.96	0.96	91
accuracy			0.96	364
macro avg	0.96	0.96	0.96	364
weighted avg	0.96	0.96	0.96	364

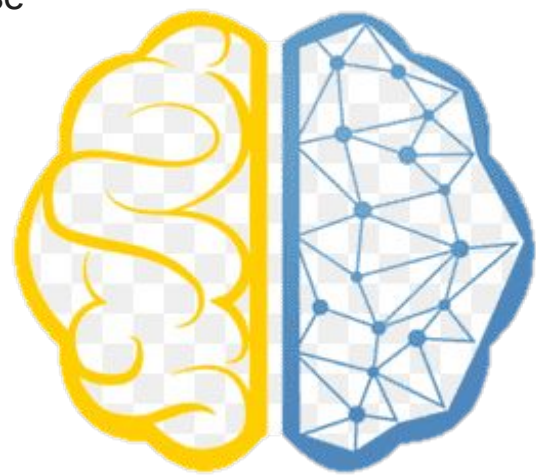
Support Vector Machines



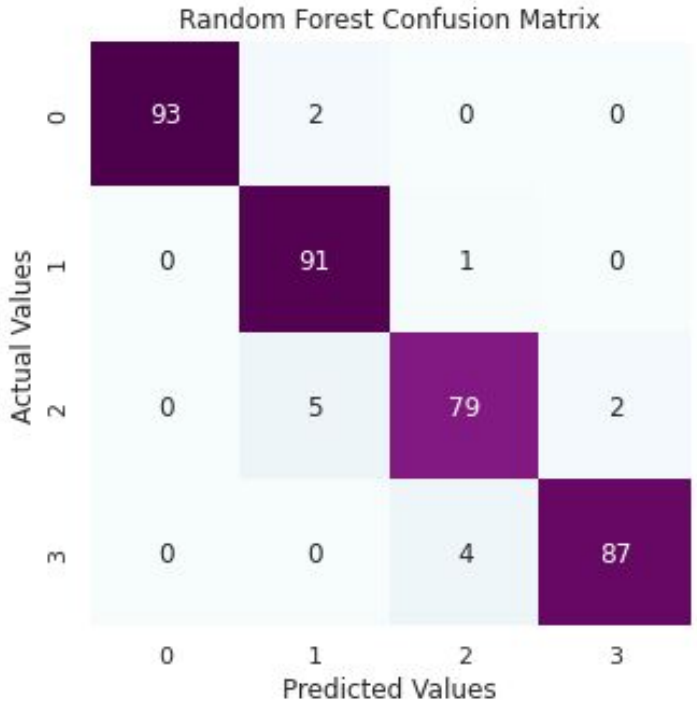

```
SVM Classifier Accuracy Score:  0.88
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       109
           1       0.76      0.92      0.83        89
           2       0.85      0.81      0.83       106
           3       0.95      0.90      0.92        96

    accuracy                           0.88       400
   macro avg       0.88      0.88      0.88       400
weighted avg       0.89      0.88      0.88       400
```

Accuracy_ Score : 88%

The linear SVM model had a classification accuracy of **88%** with those transcript variables, four fewer variables than logistic regression



Random Forests



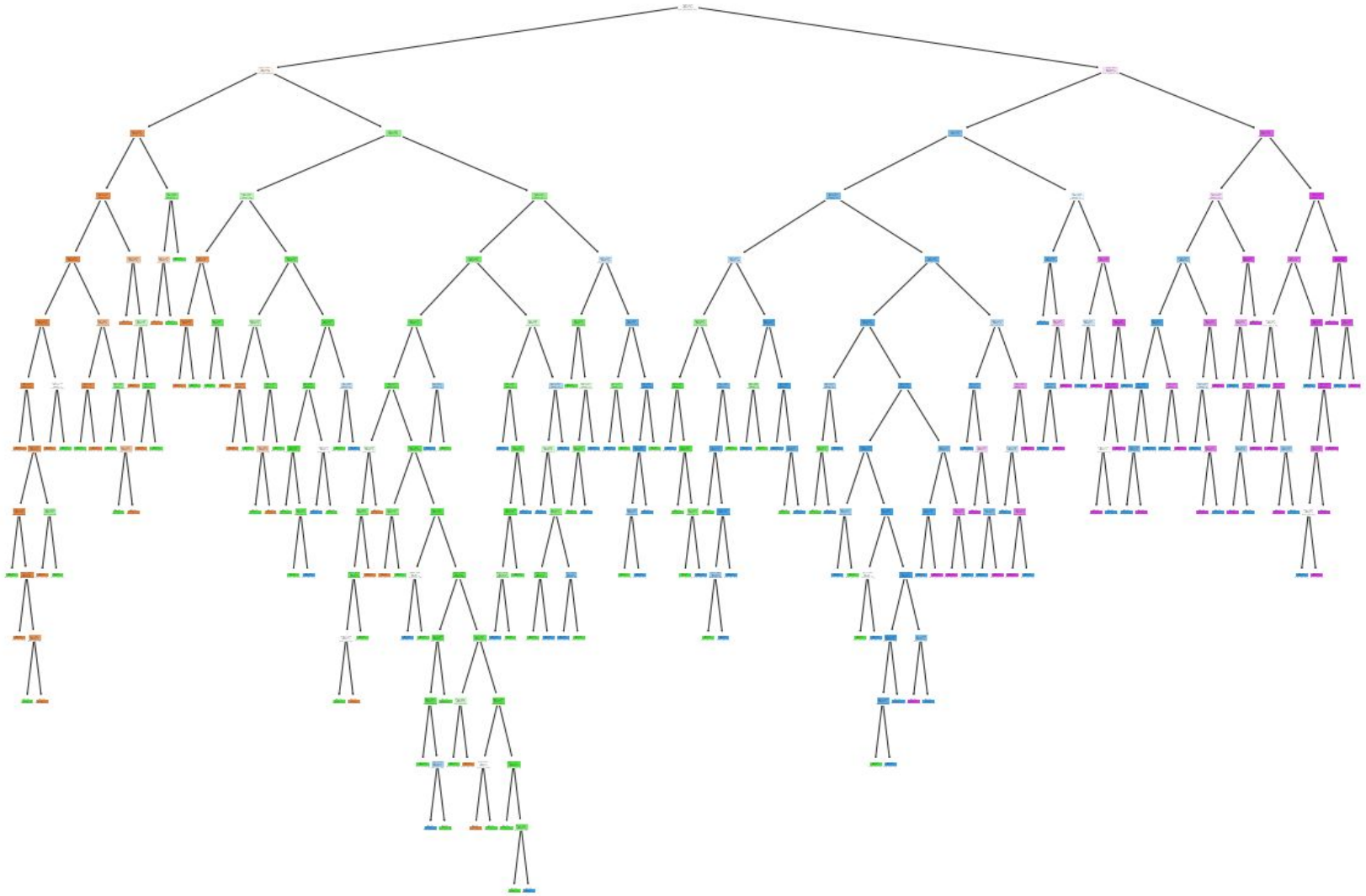
A random forest is built on a variety of decision trees. Every decision tree is made up of nodes that represent decisions, leaf nodes and a root node. The leaf nodes of each tree represent the decisions in the decision tree, and the root node represents the final result of that decision tree. The final product can be determined using a majority-voting procedure. Let us now implement our random forest algorithm.

Accuracy_ Score : 96.15%

Random Forest Classifier Accuracy Score: 0.9615384615384616

	precision	recall	f1-score	support
0	1.00	0.98	0.99	95
1	0.93	0.99	0.96	92
2	0.94	0.92	0.93	86
3	0.98	0.96	0.97	91
accuracy			0.96	364
macro avg	0.96	0.96	0.96	364
weighted avg	0.96	0.96	0.96	364

Decision Tree Classifiers



The model overfit because the training dataset accuracy was 100% for all 5 different folds of the cross validation, while the average testing dataset accuracy was 83.14%. An 83.14% accuracy for the testing set is pretty good, but I believe this model could do better. I will try to tune this model by adding more layers with smaller number of neurons per layer.

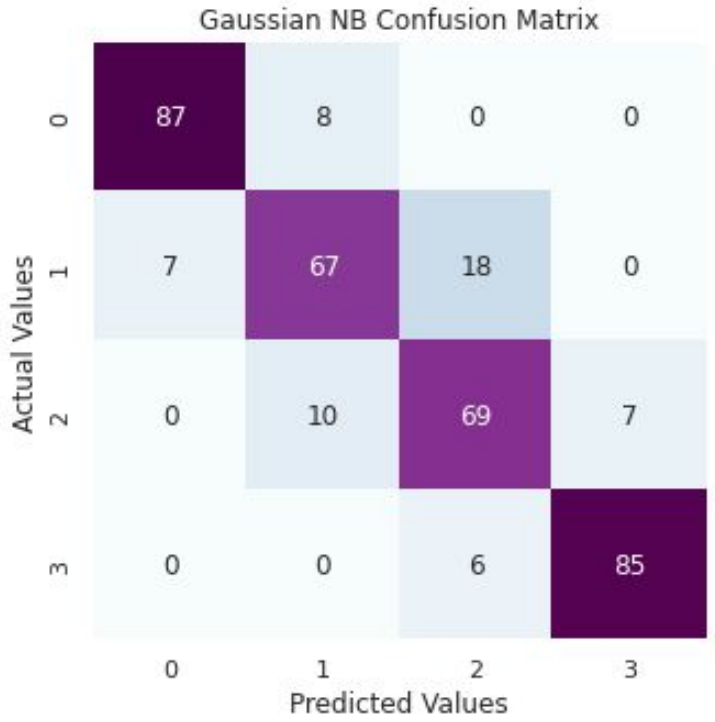
```
clf_tree.get_depth()= 14
clf_tree.get_n_leaves() =154
```

```
The average training set accuracy for the Decision Tree model is: 1.0
The average testing set accuracy for the Decision Tree model is: 0.8314999999999999
```

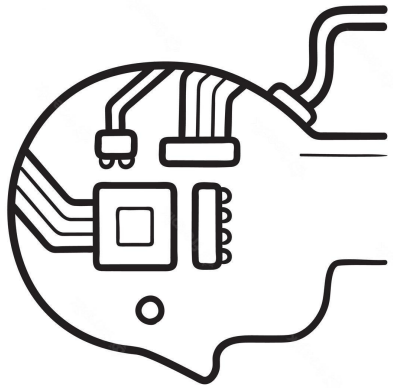
```
clf_tree.get_params()
```

```
{'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': None,
 'max_leaf_nodes': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'random_state': 42,
 'splitter': 'best'}
```

Naive Bayes



By comparing the actual and predicted values of the Naïve Bayes method, an accuracy of **84.61%** was achieved.



```
Gaussian NB Classifier Accuracy Score:  0.8461538461538461
              precision    recall  f1-score   support

           0       0.93      0.92      0.92        95
           1       0.79      0.73      0.76        92
           2       0.74      0.80      0.77        86
           3       0.92      0.93      0.93        91

    accuracy                           0.85       364
   macro avg       0.84      0.85      0.84       364
weighted avg       0.85      0.85      0.85       364
```

Accuracy_ Score : 84.61%

Logistic Regression

```
print('The accuracy of the training set is: ', clf.score(X_train, y_train))
```

```
The accuracy of the training set is:  0.975625
```

```
print('The accuracy of the testing set is: ', clf.score(X_test, y_test))
```

```
The accuracy of the testing set is:  0.9775
```

```
confusion_matrix(y_test, clf.predict(X_test))
```

```
array([[102,   3,   0,   0],
       [  0,  91,   0,   0],
       [  0,   2,  87,   3],
       [  0,   0,   1, 111]])
```

```
The average training set accuracy for the Logistic Regression model is: 0.977125
The average testing set accuracy for the Logistic Regression model is: 0.9625
```

Overall, some good baseline statistics. I will use KFold Cross Validation to ensure that the model is not overfitting and get a more realistic accuracy for the training and testing datasets. For future scenarios, I will use 5 folds for KFold Cross Validation. Note that StratifiedKFold Cross Validation isn't needed as the price_range data is spread out (equal number of 0, 1, 2, and 3)

Train_accuracy : 97.71%
Test_accuracy : 96.25%

Conclusion

Classifiers are a set of mathematical algorithms that organize data into groups. They are used for problem-solving, decision-making, and marketing activities. Classifier designers must consider many factors when creating an algorithm, including the nature of the data being analyzed, as well as the goals of the classifier's user.

- There are 4 price segments are available having similar number of devices on each.
- Ram are important parameter which varies from **256MB to 4GB** and the price is increases as Ram is increase
- Most of phones Front Camera are not available, and maximum phones contain **2MP front camera**.
- In this bar chart we found that some mobile does not contain camera and shows zero and also seen that the maximum mobile having **8MP camara and after it is followed by 13MP**
- The lowest talk time is 2.5 hour and longest talk time is 20 hours.
- The mobile weight are more than **80 grams and maximum wt. is under 200 grams.**
- Battery life starts from **500 MAH and goes to the 2000** MAH.
- Costly phones are lighter and the as the weight is increases the price is decreases.
- There are **76.1% mobile phone having 3G** connectivity and 23.9 % does not support 3G.
- There are **52.5% mobile phone having 4G** connectivity and 43.5 % does not support 4G.
- There are **50.4 % mobile phone having Bluetooth** connectivity and 49.6% does not Bluetooth
- Phone having all these connectivity is very expensive.

Conclusion

Models Results:

- **K-Nearest Neighbors -Accuracy_ Score: 95.87%.**
- **Support Vector Machines accuracy of 88%.**
- **Random Forests: Accuracy_ Score: 96.15%**
- **Decision Tree Classifiers: The average testing dataset accuracy was 83.14%.**
- **Naive Bayes: Naïve Bayes method, an accuracy of 84.61% was achieved.**
- **Logistic Regression: Train accuracy: 97.71% -----Test accuracy: 96.25%.**
- Among all the models **Logistic Regression:** gives best results.


THANK
YOU!