Capstone Project 3

Machine Learning –Classification Project

Mobile Price Range Prediction

By

Anjali A Khillare

https://github.com/Anjalikhillare/Mobile-Price-Range-Prediction.git

Table of Content

Problem statement

In the competitive mobile phone market companies want to understand sales data of mobile phones and factors which drive the prices. The objective is to find out some relation between 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.

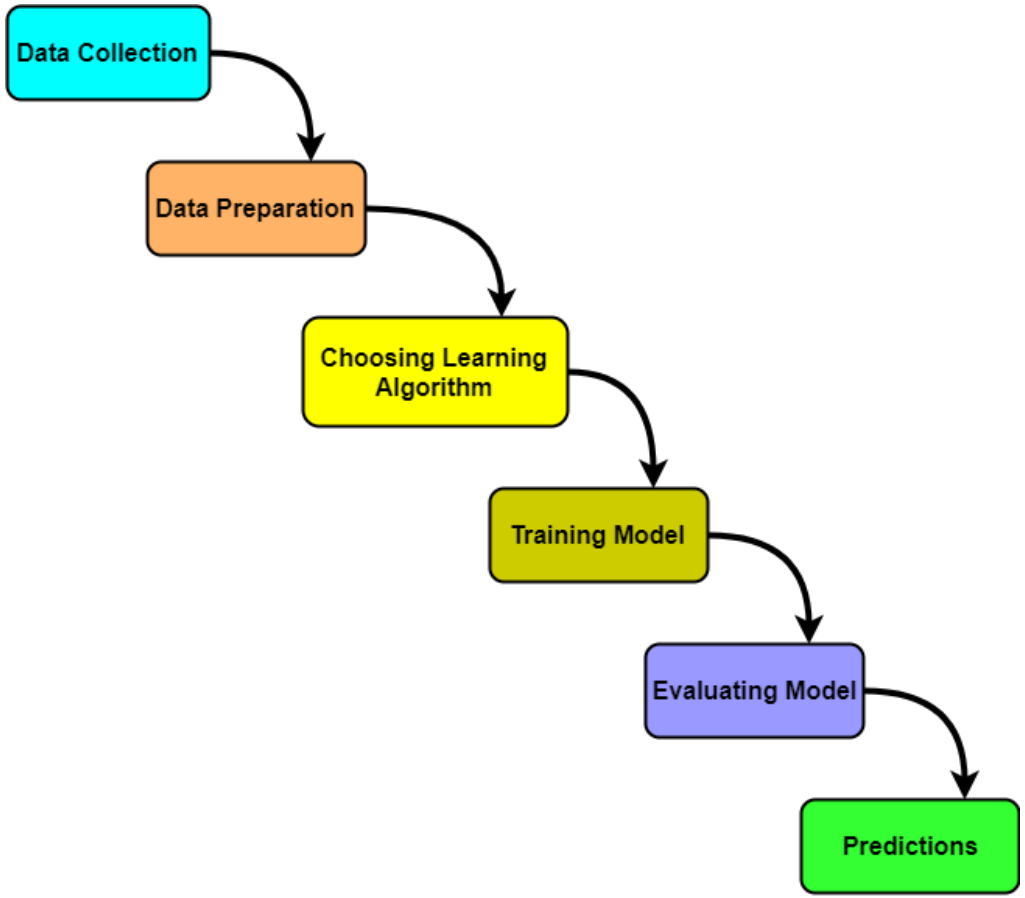
Introduction

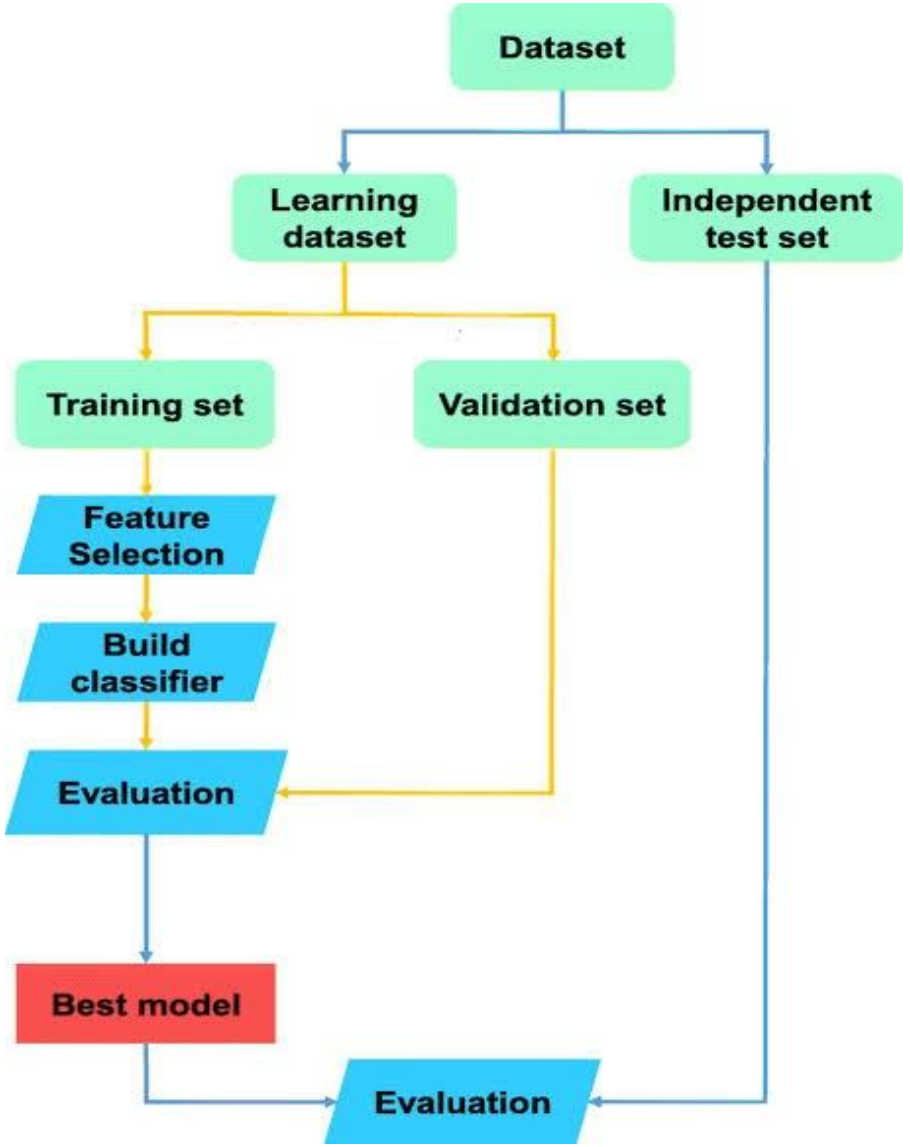
- The crucial component of marketing and business is cost prediction and same process can be used to estimate costs for other products.
- Finding the ideal product (with the lowest price and highest features) is the best marketing strategy. As a result, products can be contrasted based on factors like specification, price, manufacturer, etc.
- A decent product can be recommended to a customer by identifying their price range. Using several machine learning algorithm, mobile price prediction system is created in this project.

Objective

- To build a model which will categorize the mobile price range with the help of the ML algorithms based on several specifications.

Flow Chart





Data Information

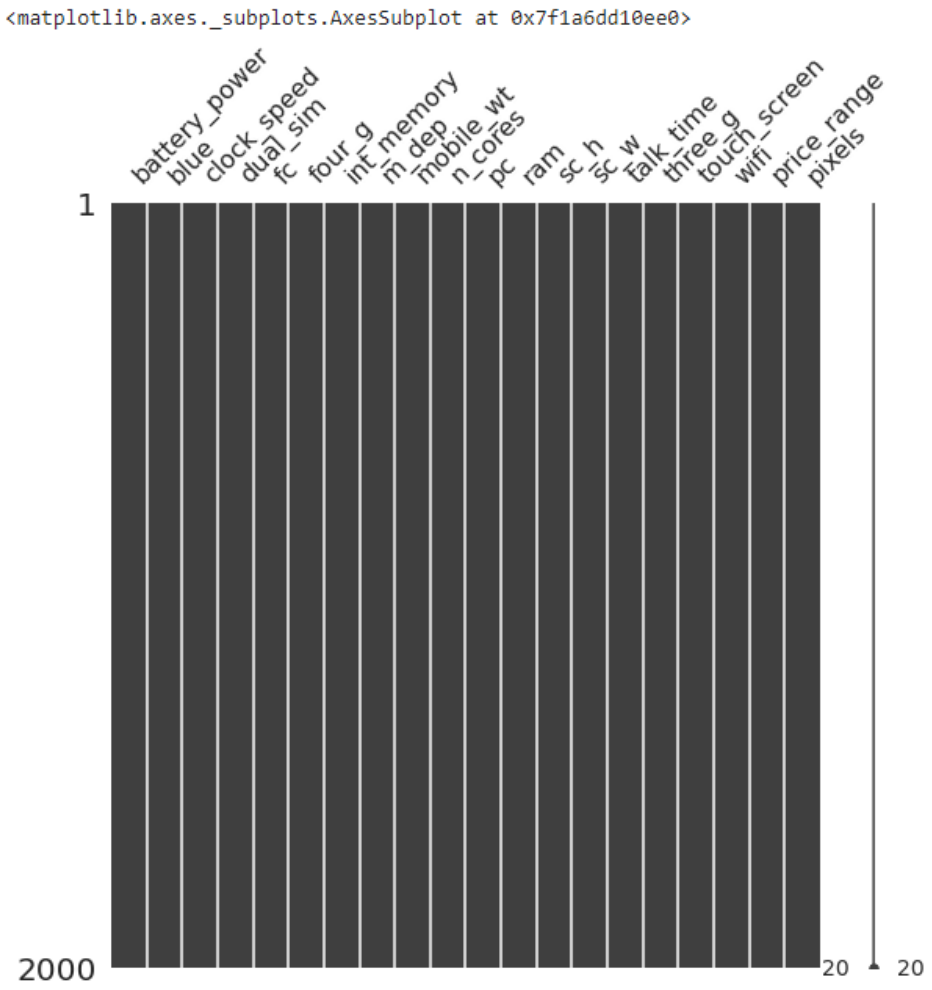
Information about the capabilities, attributes, and price range of mobile phones is included in the data. One can determine the price range of a mobile phone by using the numerous features of data.

- 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 can last over a call
- Three_g - Has 3G or not
- Touch_screen - Has touch screen or not
- Wifi - Has wifi or not
- Price_range - This is the target variable with value of 0(low cost), 1(medium cost),
- 2(high cost) and 3(very high cost).

Given dataset contains 2000 rows and 21 columns

Data Summarization



Missing values plot

Mismatch values

Given that many machine learning algorithms do not allow missing values, the management of missing data is crucial during the dataset's preprocessing.

Observation-
It can be seen that there is no presence of missing values in the given dataset.

Handling mismatch values

Here it is observed that, minimum value of sc_w and px_height cannot be zero.Let us see the count of those values ie, which equals to zero.

```
[13] # Count of mobiles with sc_w = 0
     print(len(df[df.sc_w == 0]))
     # Count of mobiles with px_height = 0
     print(len(df[df.px_height == 0]))
```

```
180
2
```

```
[14] # Assigning mean values to sc_W and px_height for their zero values
     df['sc_w'][df[df.sc_w == 0].index] = df.sc_w.mean()
     df['px_height'][df[df.px_height == 0].index] = df.px_height.mean()
```

```
[15] # Revised count of mobiles for sc_w and px_height
     print(len(df[df.sc_w == 0]))
     print(len(df[df.px_height == 0]))
```

```
0
0
```

AImaBetter

Graphs-Skewness plot-Boxplot-Probability plot

Skewness Plot

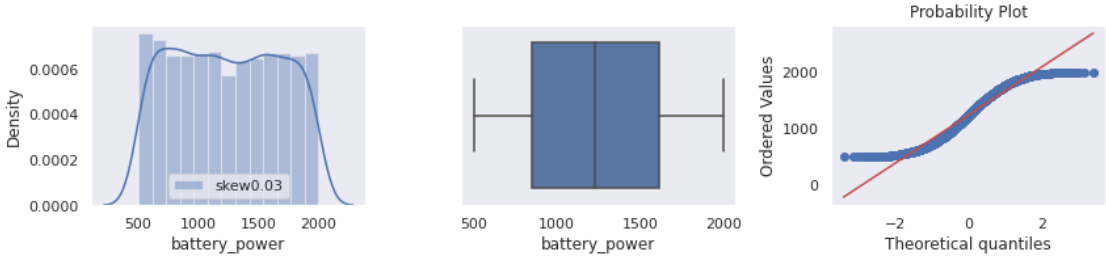
Skewed data is defined as data that causes an uneven, distorted curve on a graph. In statistics, a data set with a normal distribution has a bell shaped, symmetrical graph. The tail of skewed data, however, is on either side of the graph.

Boxplot

Data distribution in a data collection is assessed using boxplots. Three quartiles are used to split the data collection. The first quartile and third quartile of the data set, as well as the median, are represented in this graph. By creating boxplots for each data collection, it is also useful for comparing how data are distributed among them.

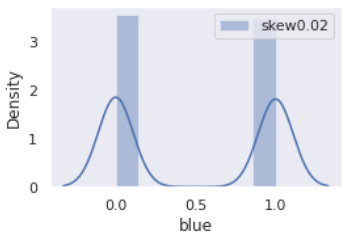
Probability Plot

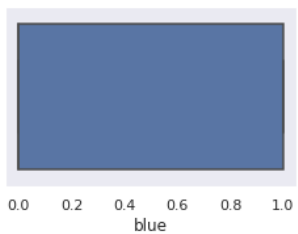
Probability plot is a visual method for determining whether a data set fits a specific distribution, like the normal. Plotting the data in relation to a theoretical distribution should result in the dots roughly forming a straight line. This line's deviations signify deviations from the expected distribution. An indicator of how well the data is fitted linearly is the correlation coefficient linked with the probability plot.

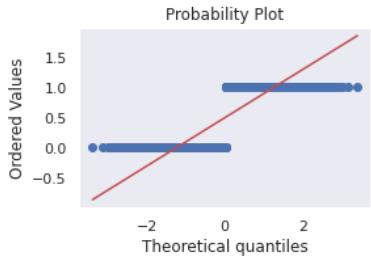

skew0.03
Density
battery_power

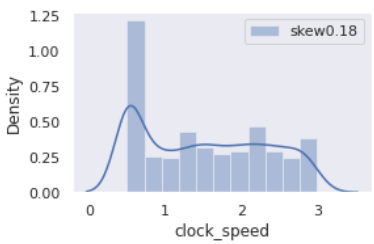
battery_power

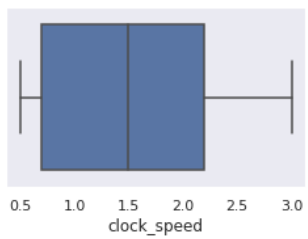
Probability Plot
Ordered Values
Theoretical quantiles

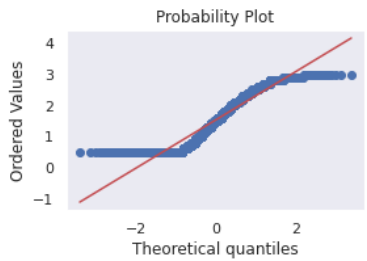

skew0.02
Density
blue

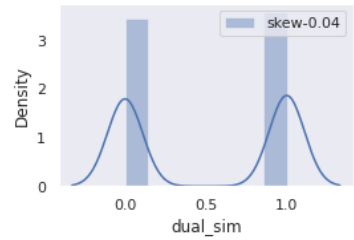

blue

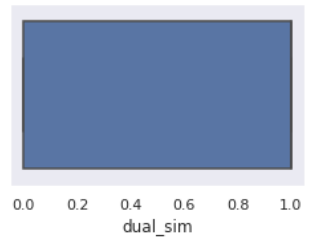

Probability Plot
Ordered Values
Theoretical quantiles

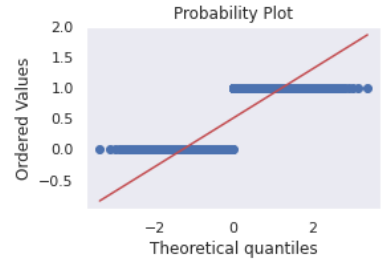

skew0.18
Density
clock_speed

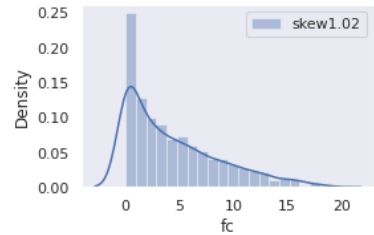

clock_speed

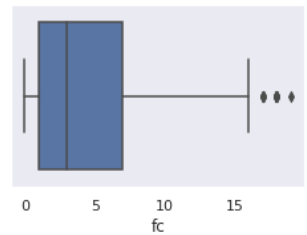

Probability Plot
Ordered Values
Theoretical quantiles

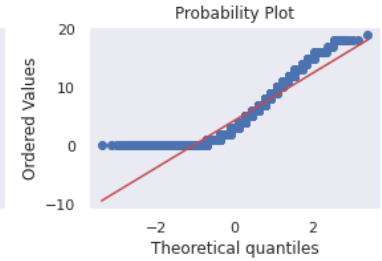

skew-0.04
Density
dual_sim

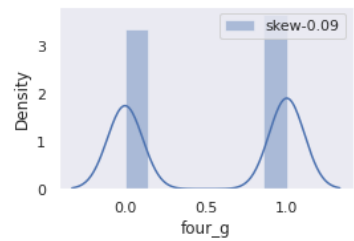

dual_sim

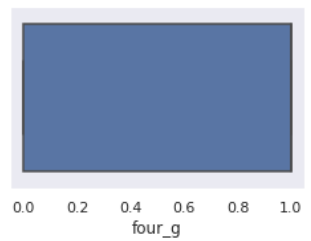

Probability Plot
Ordered Values
Theoretical quantiles

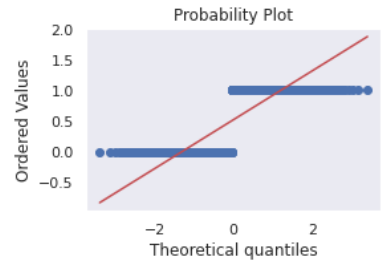

skew1.02
Density
fc

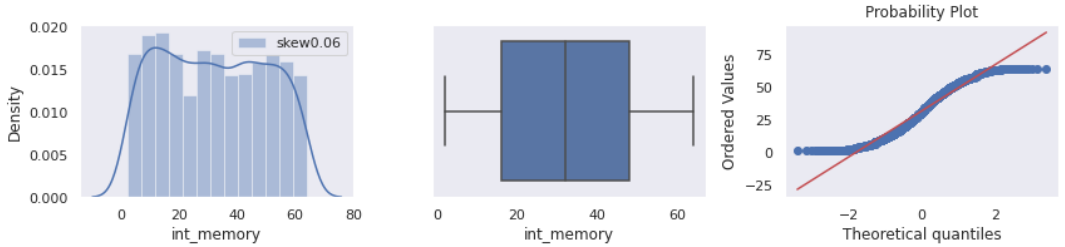

fc


Probability Plot
Ordered Values
Theoretical quantiles


skew-0.09
Density
four_g

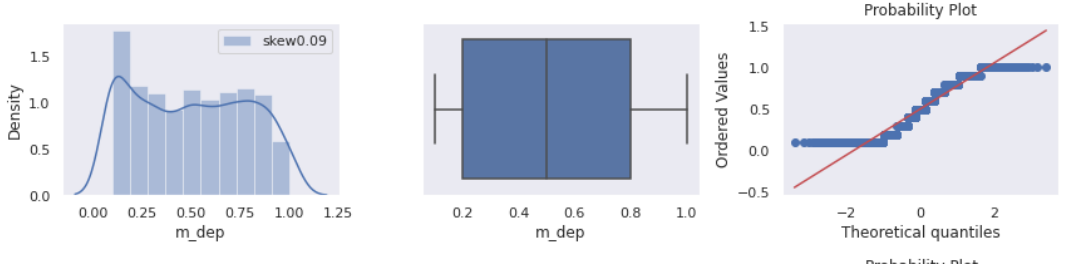

four_g


Probability Plot
Ordered Values
Theoretical quantiles


skew0.06
Density
int_memory

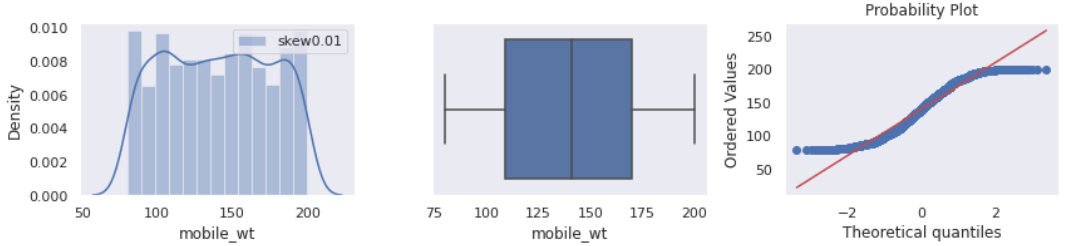
int_memory

Probability Plot
Ordered Values
Theoretical quantiles


skew0.09
Density
m_dep

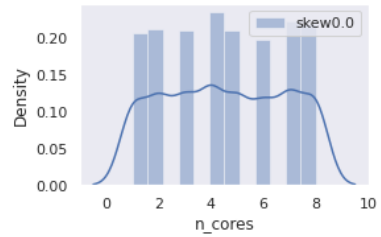
m_dep

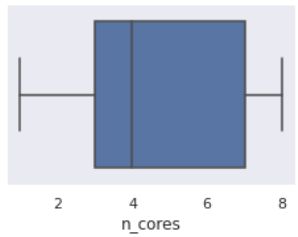
Probability Plot
Ordered Values
Theoretical quantiles

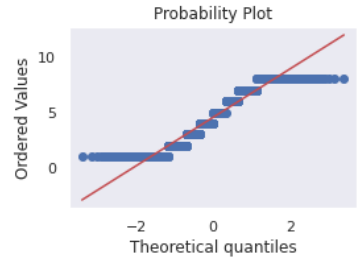

skew0.01
Density
mobile_wt

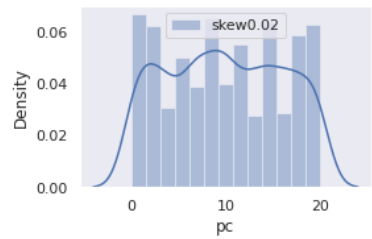
mobile_wt

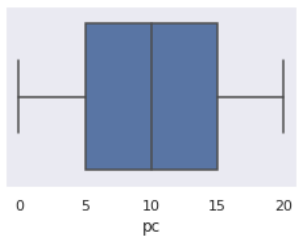
Probability Plot
Ordered Values
Theoretical quantiles

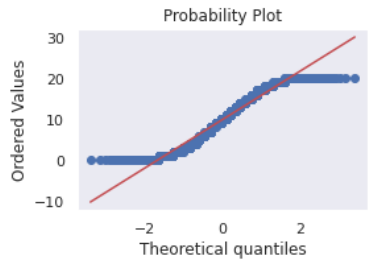

skew0.0
Density
n_cores

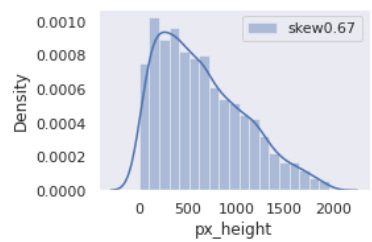

n_cores

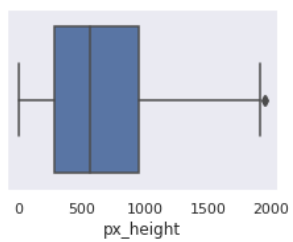

Probability Plot
Ordered Values
Theoretical quantiles

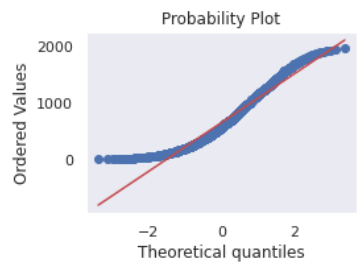

skew0.02
Density
pc

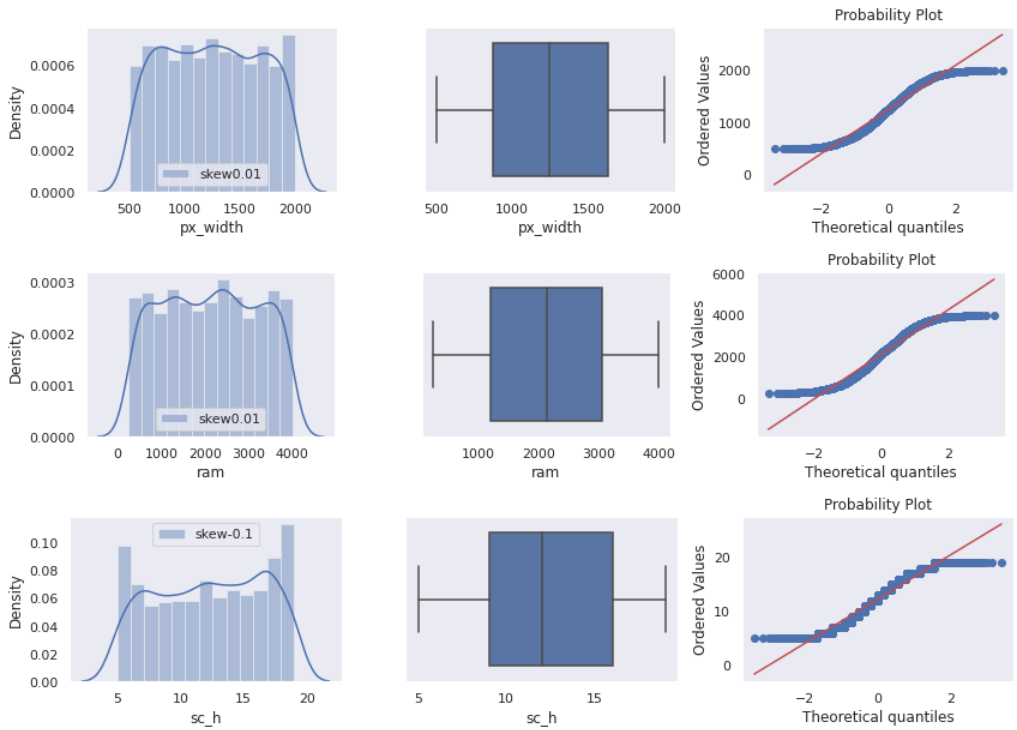

pc

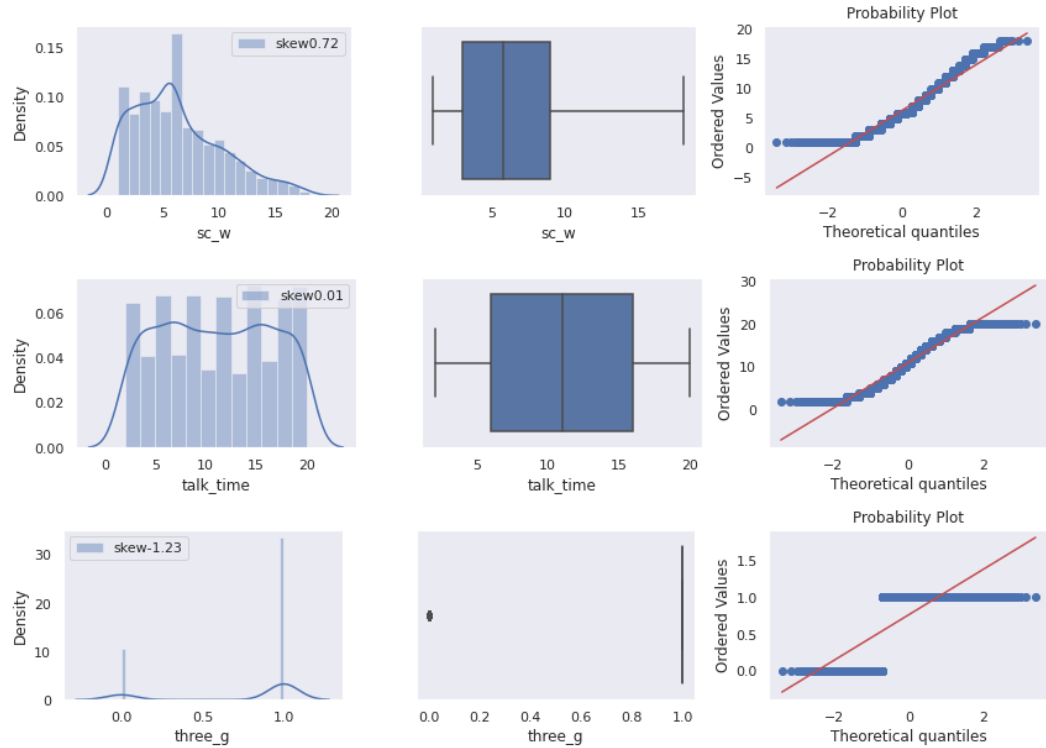

Probability Plot
Ordered Values
Theoretical quantiles

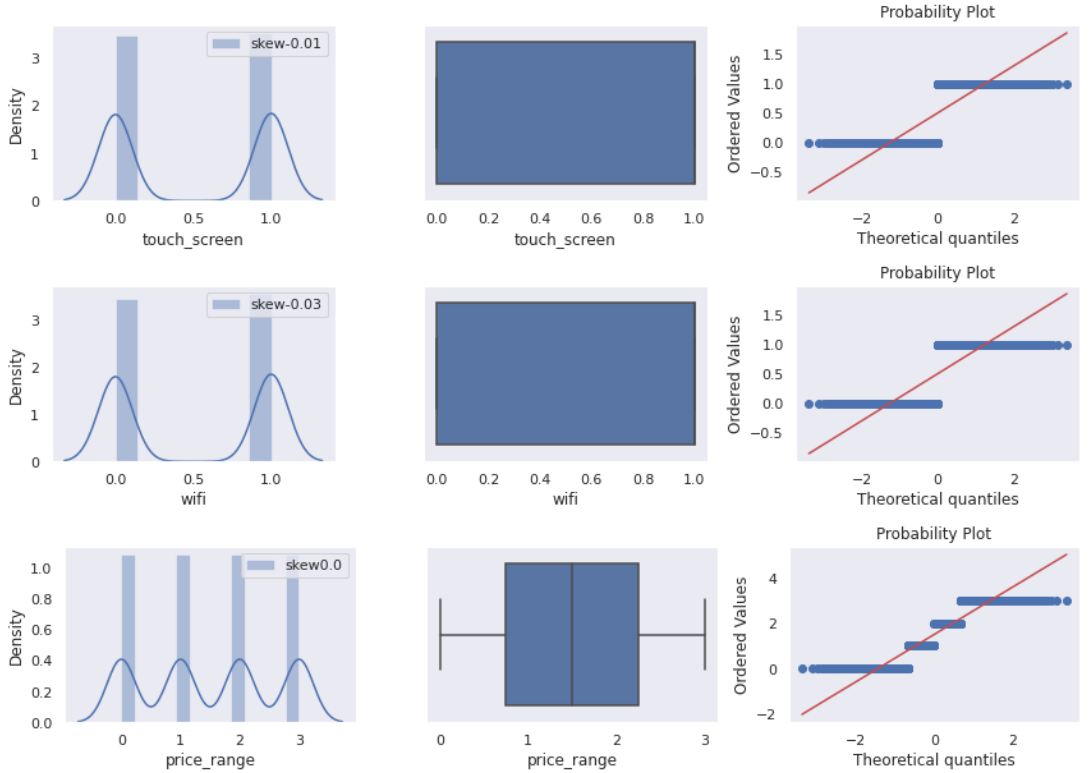

skew0.67
Density
px_height


px_height


Probability Plot
Ordered Values
Theoretical quantiles


skew0.01
Density
px_width
Probability Plot
Ordered Values
Theoretical quantiles
skew0.01
Density
ram
Probability Plot
Ordered Values
Theoretical quantiles
skew-0.1
Density
sc_h
Probability Plot
Ordered Values
Theoretical quantiles


skew0.72
Density
sc_w
Probability Plot
Ordered Values
Theoretical quantiles
skew0.01
Density
talk_time
Probability Plot
Ordered Values
Theoretical quantiles
skew-1.23
Density
three_g
Probability Plot
Ordered Values
Theoretical quantiles



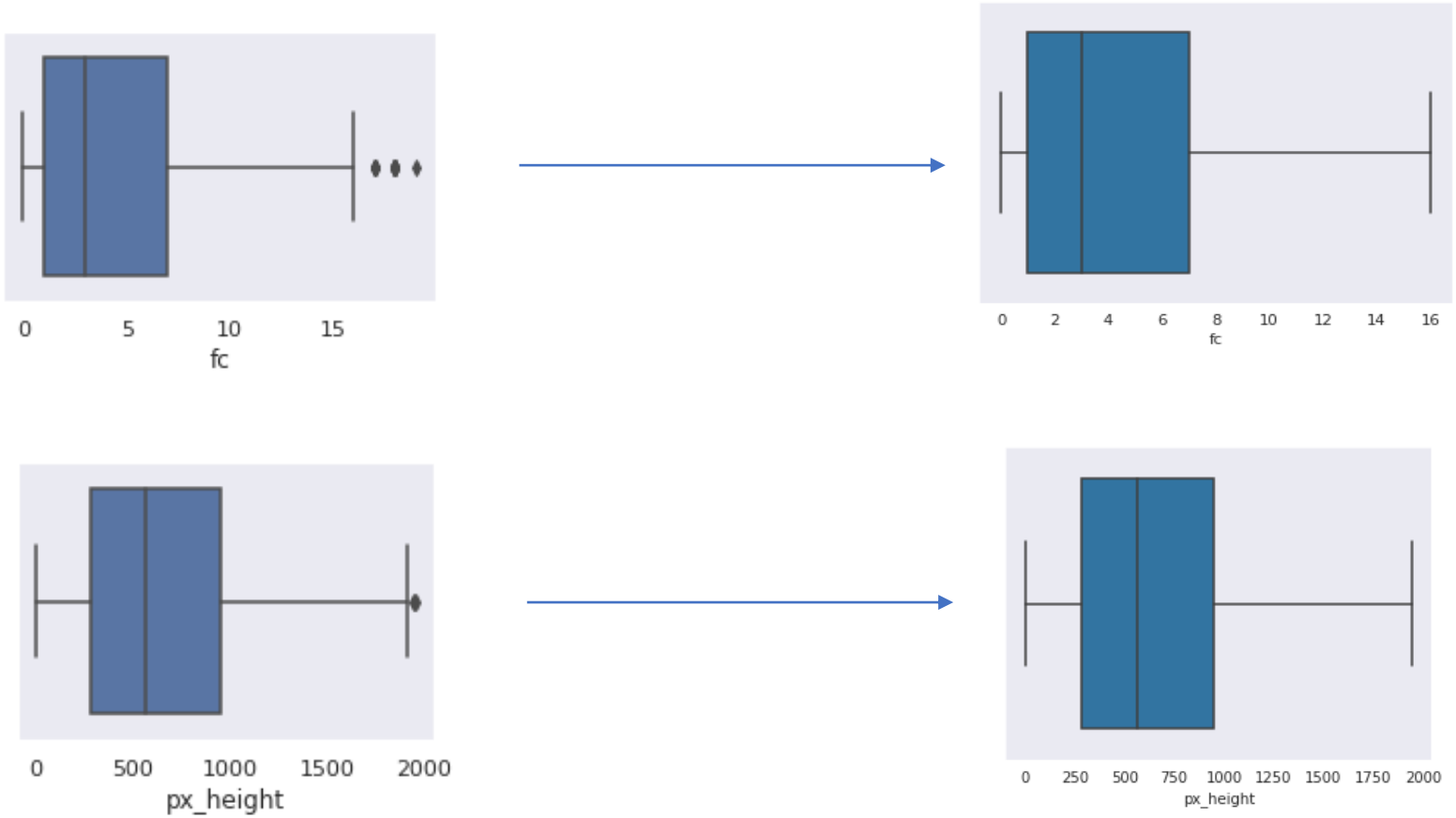
Observations-

Right skewness is observed in fc, px_height and sc_w

Presence of outliers is seen in fc, px_height

Sample data that is somewhat normally distributed is of battery_power,int_memory, mobile_wt, px_width, ram.

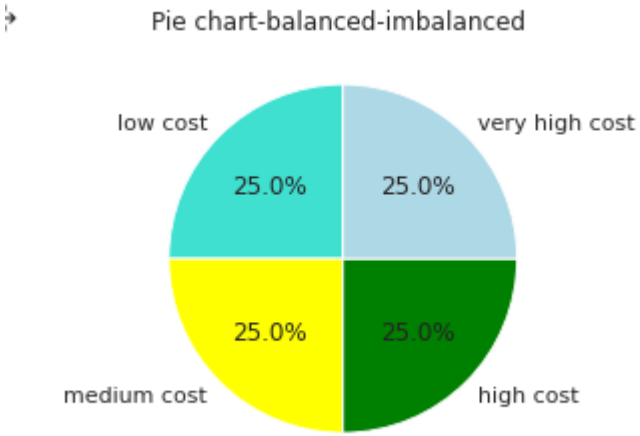
Boxplot after removal of outliers



From the boxplot it can be observed that outliers are removed

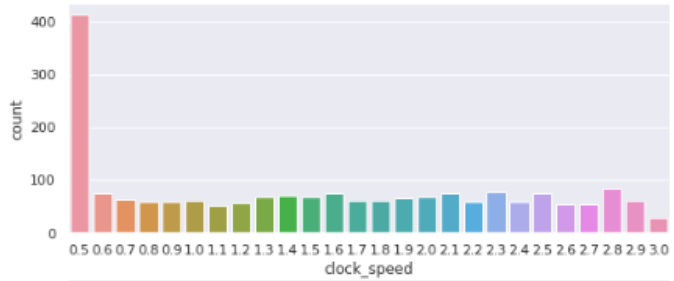
EDA

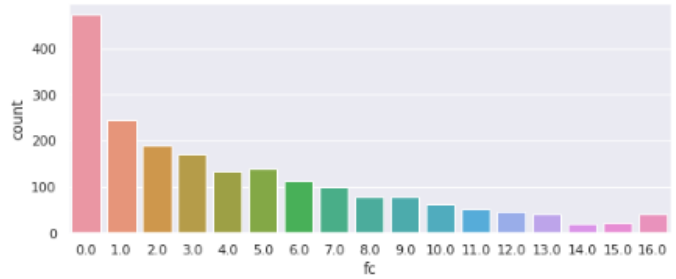
A. Univariate Analysis

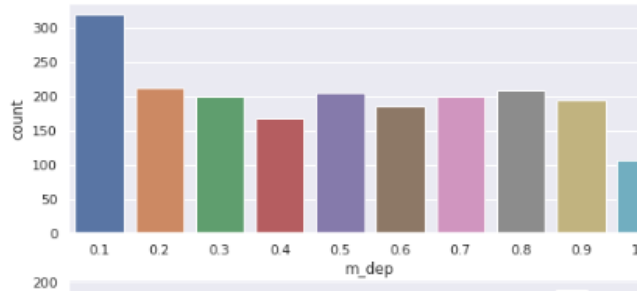


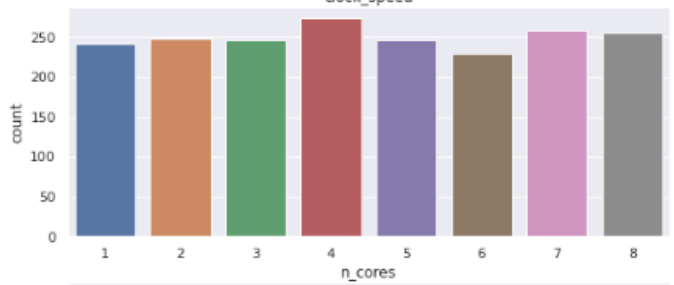
Target variable is equally distributed and have equal number of observations in each category

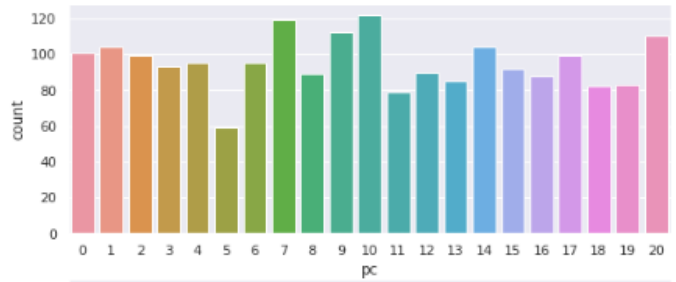
Numerical Features

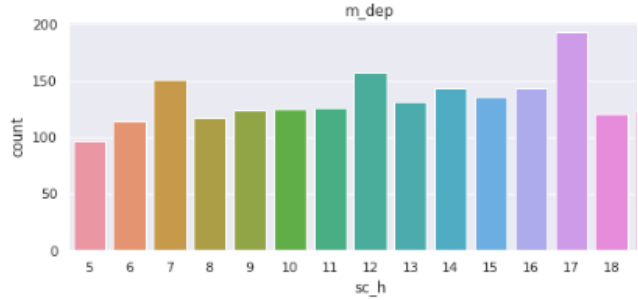


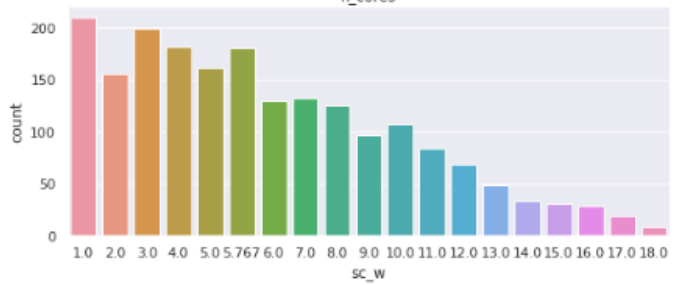


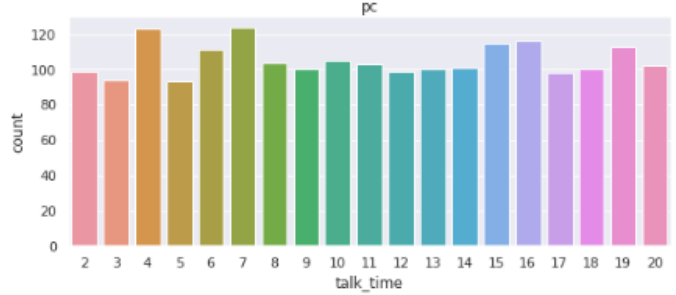






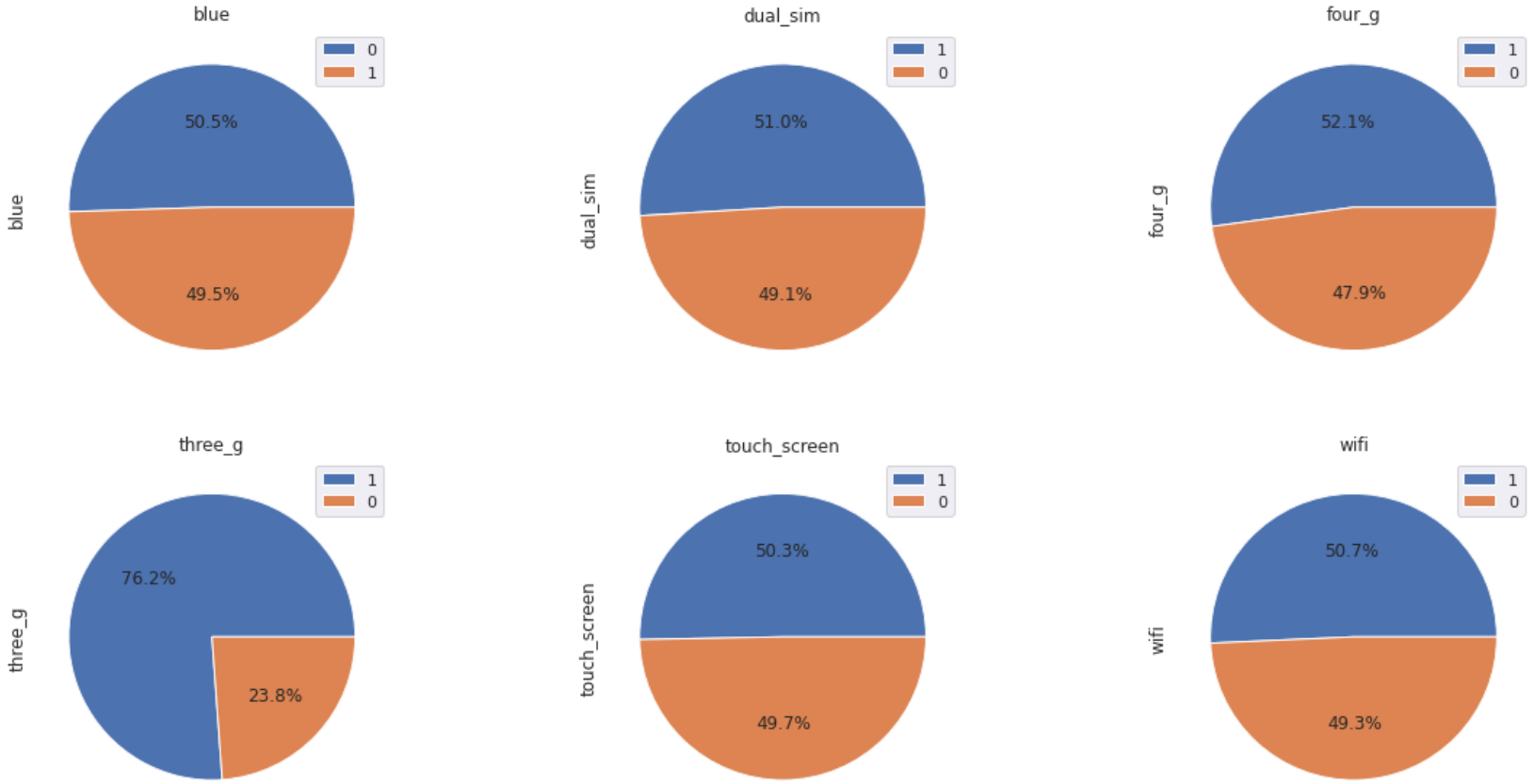






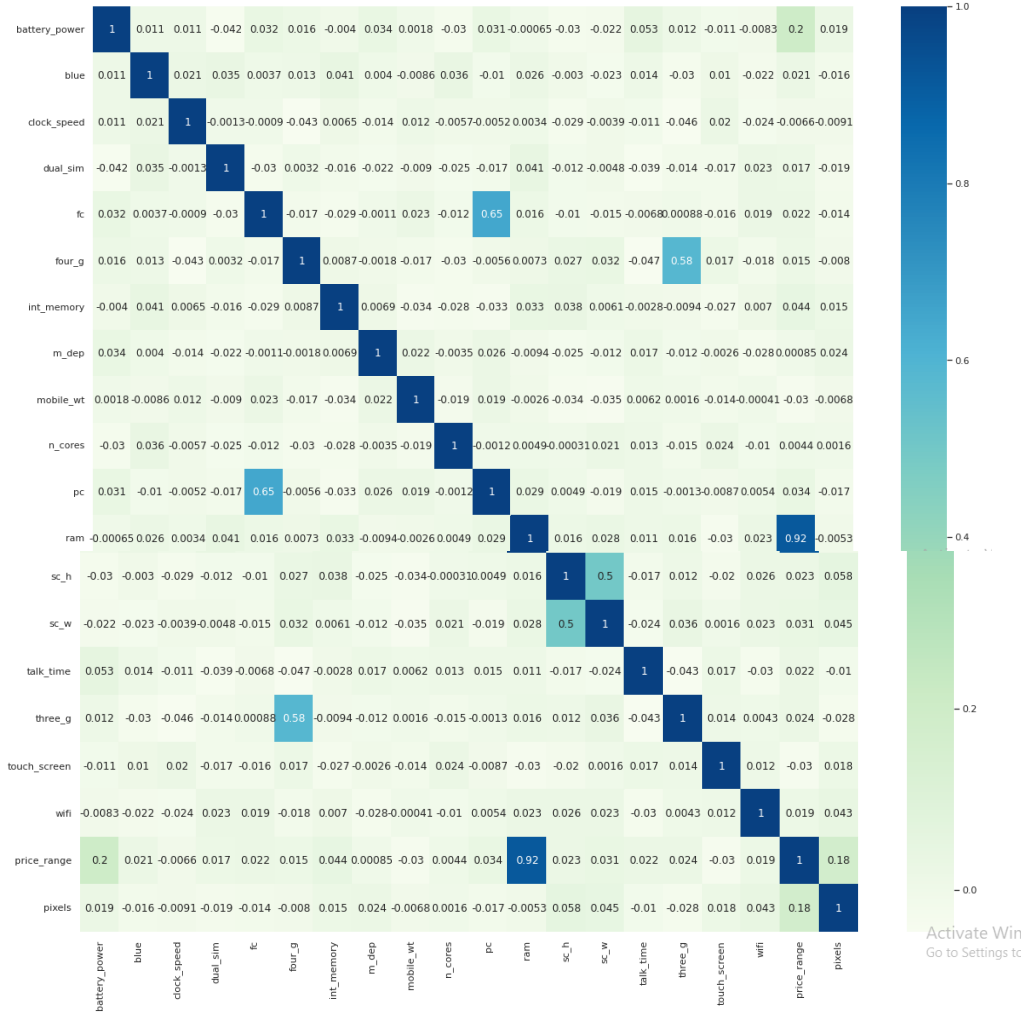
AlmaBetter

Categorical Features



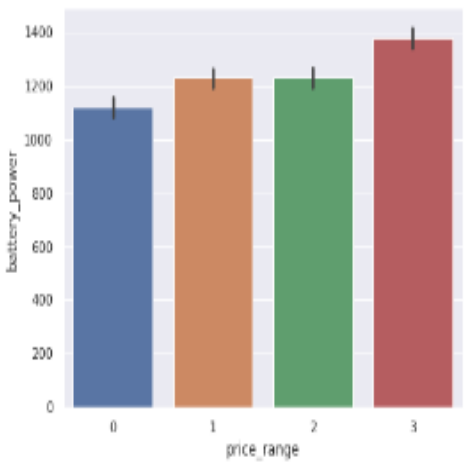
B. Bivariate Analysis

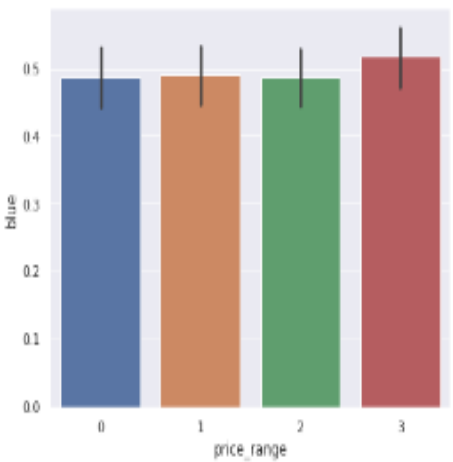
Heatmap

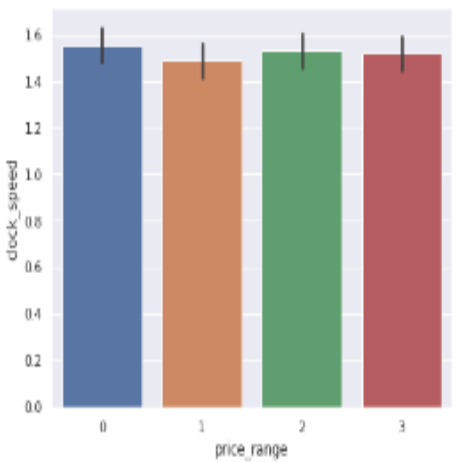


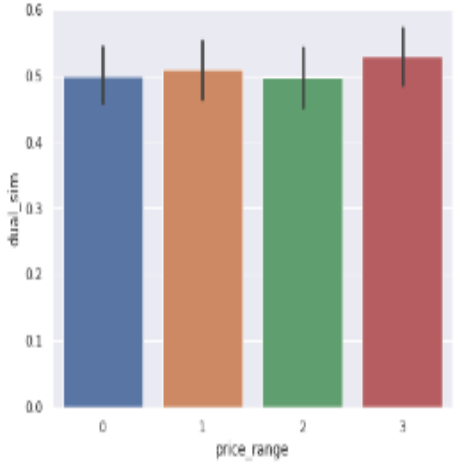
- Ram is the variable that has the most influence, as can be seen from the heatmap, while most other variables have very little relationship to price range.
- There is a correlation between the primary camera's mega pixels and the front camera's mega pixels, the price range is unaffected.
- px_height and px_width these two features can be swapped out for a single feature and both are positively correlated.
- 3G and 4G are moderately associated.

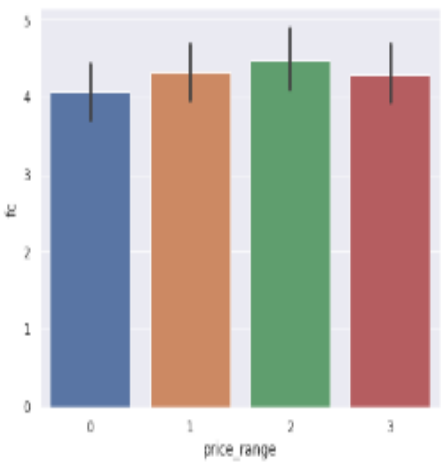
Price Range vs Independent variables

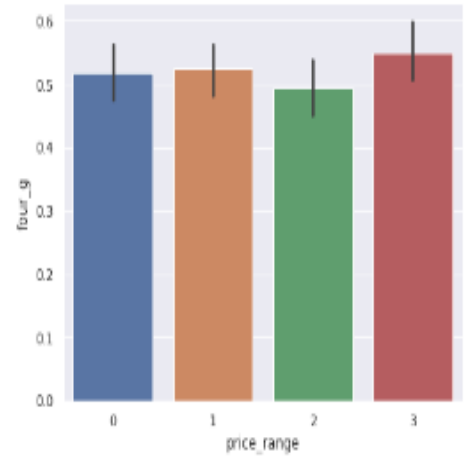


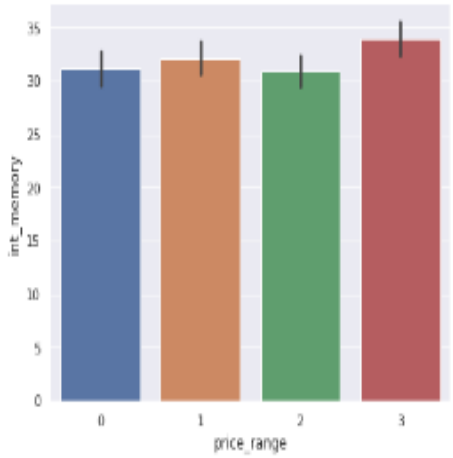


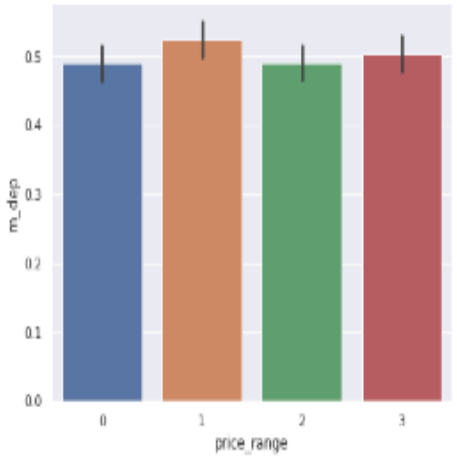


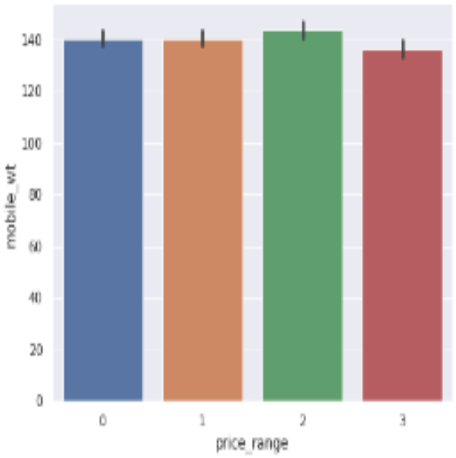


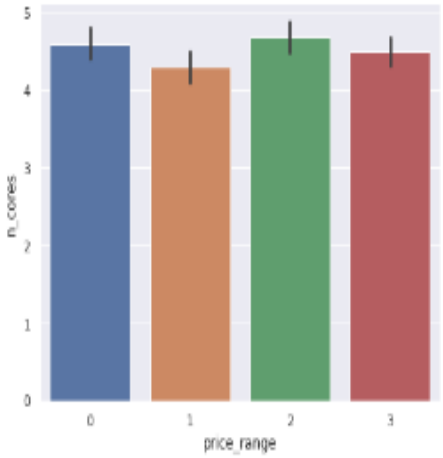




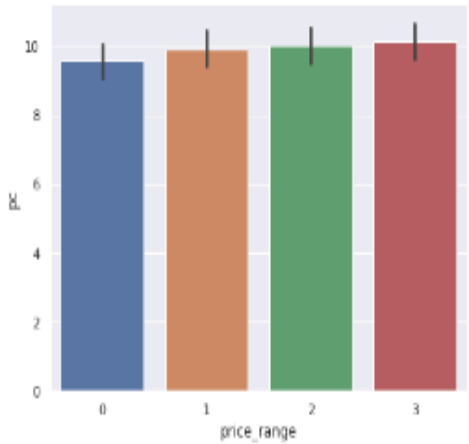


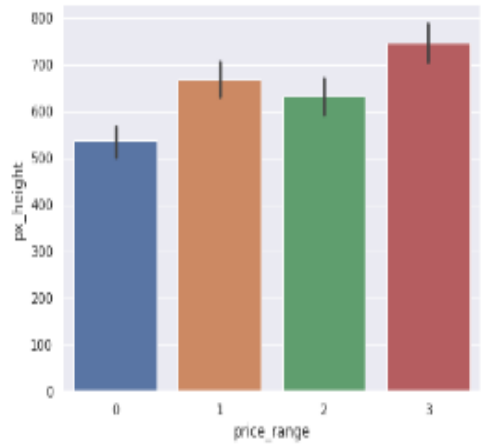


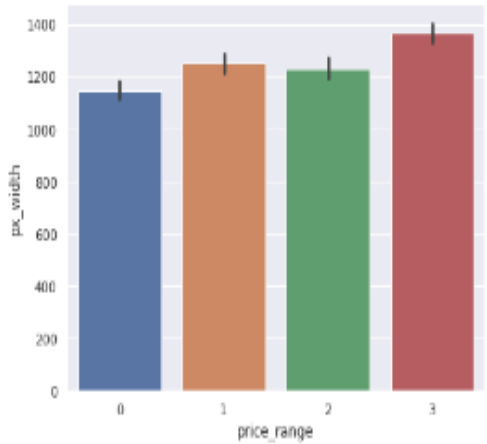


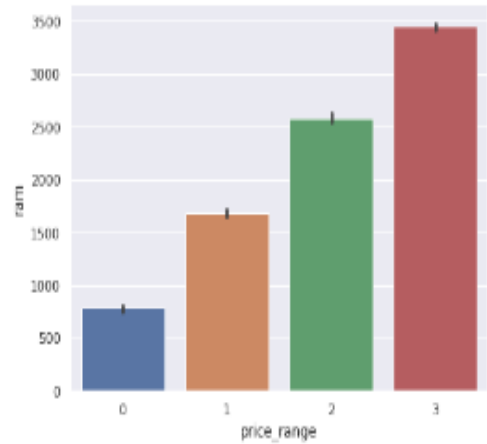


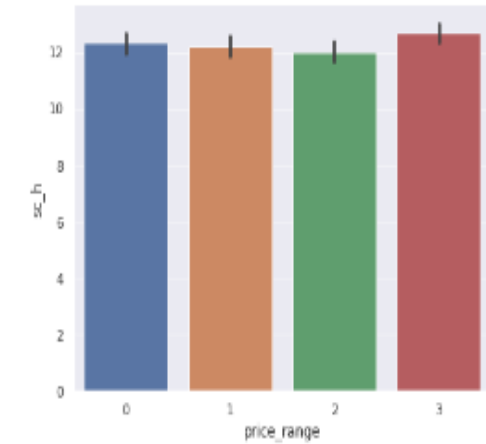
AlmaBetter

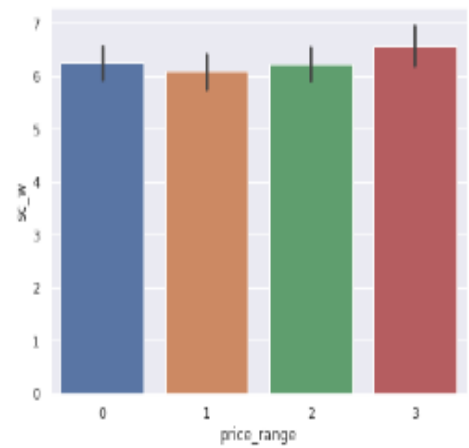

pc
price_range

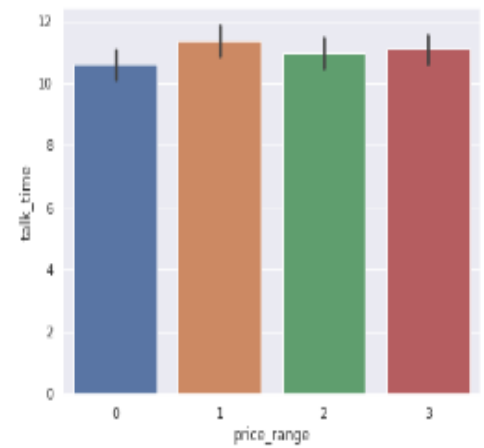

px_height
price_range

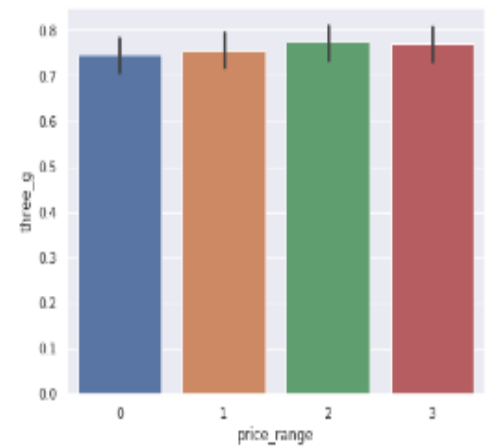

px_width
price_range

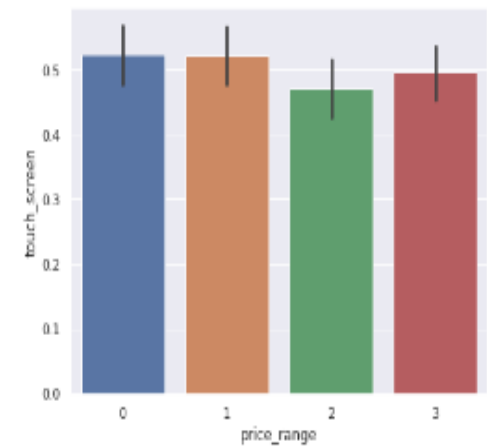

ram
price_range

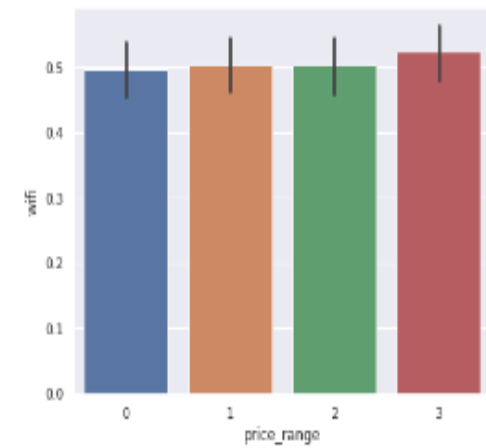

sc_h
price_range


sc_w
price_range


talk_time
price_range


three_g
price_range


touch_screen
price_range


wifi
price_range

Feature Engineering

- One of the fundamental ideas in machine learning is feature selection. It significantly affects the performance of model.
- Ability of model to perform well depends greatly on the data features used to train machine learning models.
- Reduces Overfitting: Less duplicated data reduces the chance that decisions will be based on noise.
- Enhances Accuracy: When there are fewer misleading data, modelling accuracy increases.
- Less data points result in less algorithm complexity and quicker training of algorithms, which reduces training time.

	Specifications	Score
13	ram	931267.519053
11	px_height	17589.263347
0	battery_power	14129.866576
12	px_width	9810.586750
8	mobile_wt	95.972863
6	int_memory	89.839124
16	talk_time	13.236400
4	fc	10.454588
15	sc_w	10.047181
14	sc_h	9.614878
10	pc	9.186054
9	n_cores	9.097556
18	touch_screen	1.928429
5	four_g	1.521572
7	m_dep	0.745820

```
[32] #  Defining new variable for pixels

     df['pixels'] = df['px_height']*df['px_width']
     # Dropping px_height and px_width

     df.drop(['px_height', 'px_width'], axis = 1, inplace = True)
```

```
[33] # Defining X and y

     X = df.drop(['price_range'], axis = 1)
     y = df['price_range']
```

Scaling data and Train-test split

- **Scaling Data**

The process of feature scaling is used to uniformly scale the variety of independent variables or features in data. It is also known as data normalization or standardization in the context of data processing. Before utilizing machine learning methods to train models, feature scaling is typically done.

- **Train test split**

When using machine learning algorithms to make predictions on data that was not used to train the model, the train-test split technique is used to measure how well they perform.. In essence, it is a method of splitting the training data into two parts so that you may test your algorithm on one part and assess the results on the rest of the part.

Model Selection-ML algorithms

1. Logistic regression
2. Decision tree
3. Random Forest
4. SVM-Support Vector Machine
5. KNN-K Nearest Neighbour

1.Logistic Regression

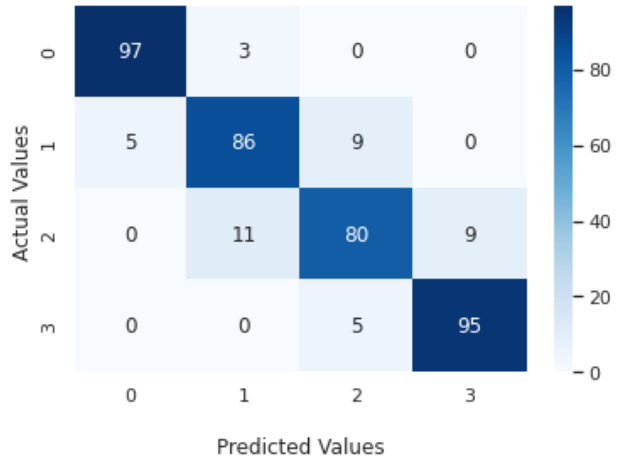
Optimized L.R

Classification report for Logistic Regression (Test set)=

	precision	recall	f1-score	support
0	0.97	0.95	0.96	102
1	0.86	0.86	0.86	100
2	0.80	0.85	0.82	94
3	0.95	0.91	0.93	104
accuracy			0.90	400
macro avg	0.90	0.89	0.89	400
weighted avg	0.90	0.90	0.90	400

```
[[97  3  0  0]
 [ 5 86  9  0]
 [ 0 11 80  9]
 [ 0  0  5 95]]
```

Seaborn Confusion Matrix with labels

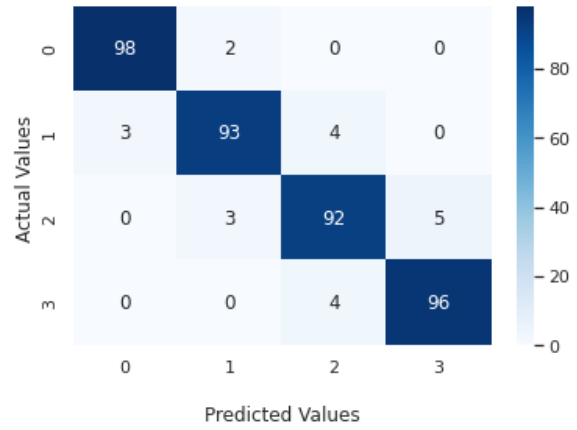


Classification report for Logistic Regression GridSearch (Test set)=

	precision	recall	f1-score	support
0	0.98	0.97	0.98	101
1	0.93	0.95	0.94	98
2	0.92	0.92	0.92	100
3	0.96	0.95	0.96	101
accuracy			0.95	400
macro avg	0.95	0.95	0.95	400
weighted avg	0.95	0.95	0.95	400

```
[[98  2  0  0]
 [ 3 93  4  0]
 [ 0  3 92  5]
 [ 0  0  4 96]]
```

Seaborn Confusion Matrix with labels



AlmaBetter

2.Decision Tree

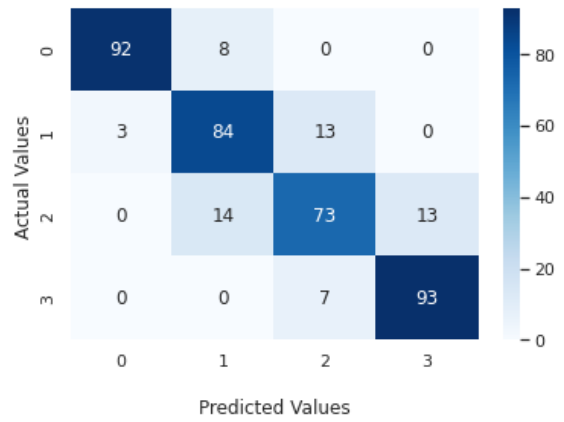
Optimized D.T

Classification report for Decision Tree (Test set)=

	precision	recall	f1-score	support
0	0.92	0.97	0.94	95
1	0.84	0.79	0.82	106
2	0.73	0.78	0.76	93
3	0.93	0.88	0.90	106
accuracy			0.85	400
macro avg	0.86	0.86	0.85	400
weighted avg	0.86	0.85	0.86	400

```
[[92  8  0  0]
 [ 3 84 13  0]
 [ 0 14 73 13]
 [ 0  0  7 93]]
```

Seaborn Confusion Matrix with labels

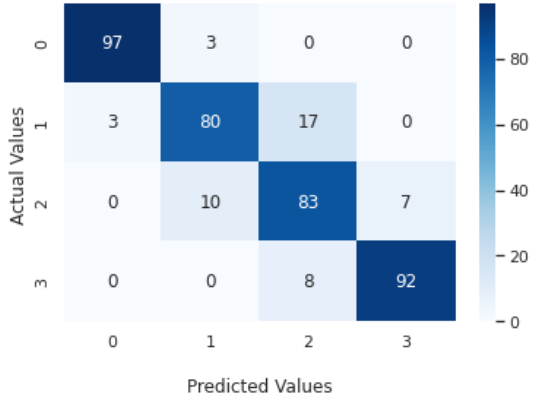


Classification Report for Decision Tree Opti. (Test set)=

	precision	recall	f1-score	support
0	0.97	0.97	0.97	100
1	0.86	0.80	0.83	100
2	0.77	0.83	0.80	100
3	0.93	0.92	0.92	100
accuracy			0.88	400
macro avg	0.88	0.88	0.88	400
weighted avg	0.88	0.88	0.88	400

```
[[97  3  0  0]
 [ 3 80 17  0]
 [ 0 10 83  7]
 [ 0  0  8 92]]
```

Seaborn Confusion Matrix with labels



AlmaBetter

Random Forest

Optimized R.F

Classification Report for Random Forest (Test set)=

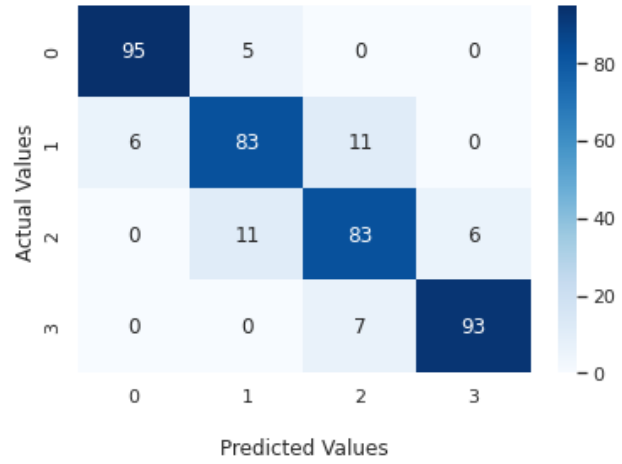
	precision	recall	f1-score	support
0	0.94	0.95	0.95	100
1	0.84	0.83	0.83	100
2	0.82	0.83	0.83	100
3	0.94	0.93	0.93	100
accuracy			0.89	400
macro avg	0.89	0.89	0.88	400
weighted avg	0.89	0.89	0.88	400

Classification Report for Random Forest GridSearch (Test set)=

	precision	recall	f1-score	support
0	0.91	0.97	0.94	100
1	0.77	0.77	0.77	100
2	0.80	0.67	0.73	100
3	0.88	0.96	0.92	100
accuracy			0.84	400
macro avg	0.84	0.84	0.84	400
weighted avg	0.84	0.84	0.84	400

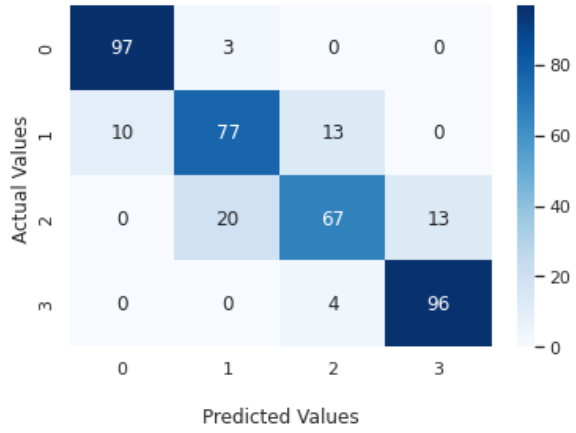
```
[[95  5  0  0]
 [ 6 83 11  0]
 [ 0 11 83  6]
 [ 0  0  7 93]]
```

Seaborn Confusion Matrix with labels




```
[[97  3  0  0]
 [10 77 13  0]
 [ 0 20 67 13]
 [ 0  0  4 96]]
```

Seaborn Confusion Matrix with labels



AlmaBetter

SVM-Support Vector Machine

SVM (kernel)

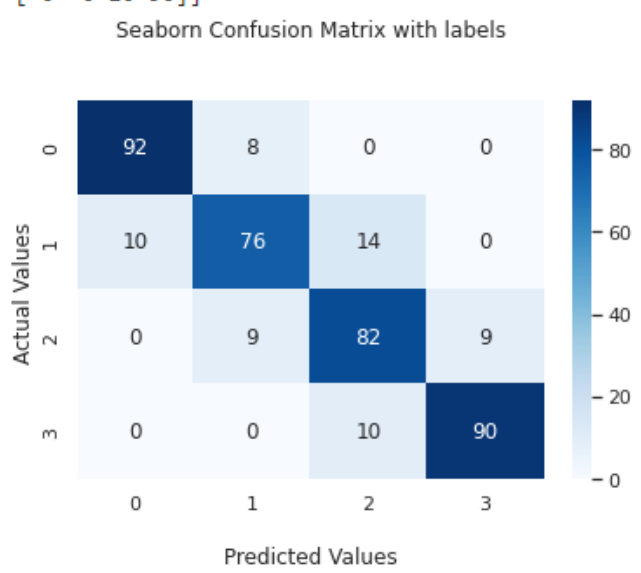
Classification Report for SVM (Test set)=

	precision	recall	f1-score	support
0	0.90	0.92	0.91	100
1	0.82	0.76	0.79	100
2	0.77	0.82	0.80	100
3	0.91	0.90	0.90	100
accuracy			0.85	400
macro avg	0.85	0.85	0.85	400
weighted avg	0.85	0.85	0.85	400

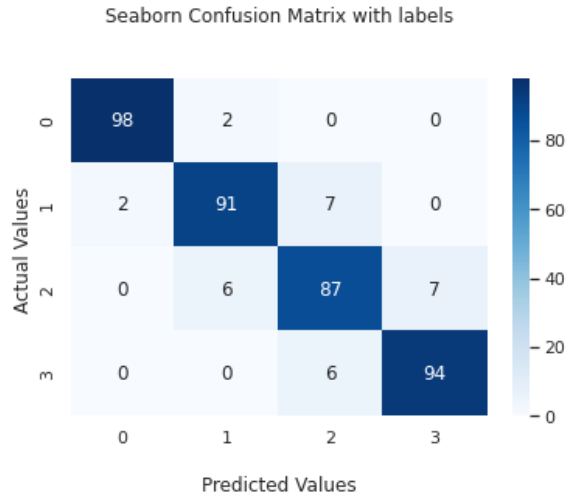
Classification Report for SVM Opti. (Test set)=

	precision	recall	f1-score	support
0	0.98	0.98	0.98	100
1	0.92	0.91	0.91	100
2	0.87	0.87	0.87	100
3	0.93	0.94	0.94	100
accuracy			0.93	400
macro avg	0.92	0.93	0.92	400
weighted avg	0.92	0.93	0.92	400

```
[[92  8  0  0]
 [10 76 14  0]
 [ 0  9 82  9]
 [ 0  0 10 90]]
```




```
[[98  2  0  0]
 [ 2 91  7  0]
 [ 0  6 87  7]
 [ 0  0  6 94]]
```



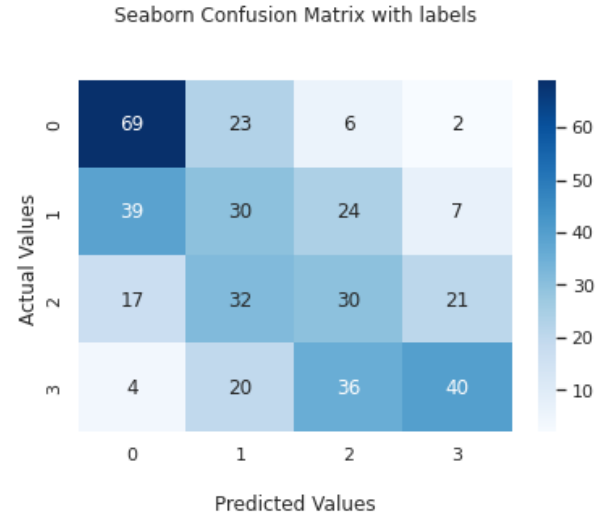
AlmaBetter

KNN-K Nearest Neighbour

Classification report for KNN (Test)=

	precision	recall	f1-score	support
0	0.53	0.69	0.60	100
1	0.29	0.30	0.29	100
2	0.31	0.30	0.31	100
3	0.57	0.40	0.47	100
accuracy			0.42	400
macro avg	0.43	0.42	0.42	400
weighted avg	0.43	0.42	0.42	400

```
[[69 23  6  2]
 [39 30 24  7]
 [17 32 30 21]
 [ 4 20 36 40]]
```

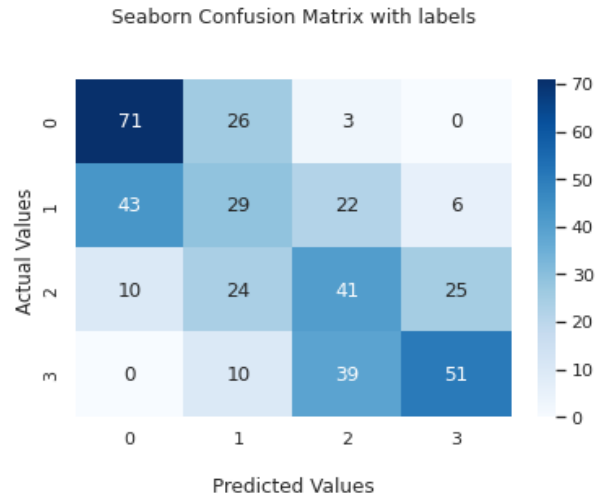


Optimized KNN

Classification report for KNN Opti.(Test)=

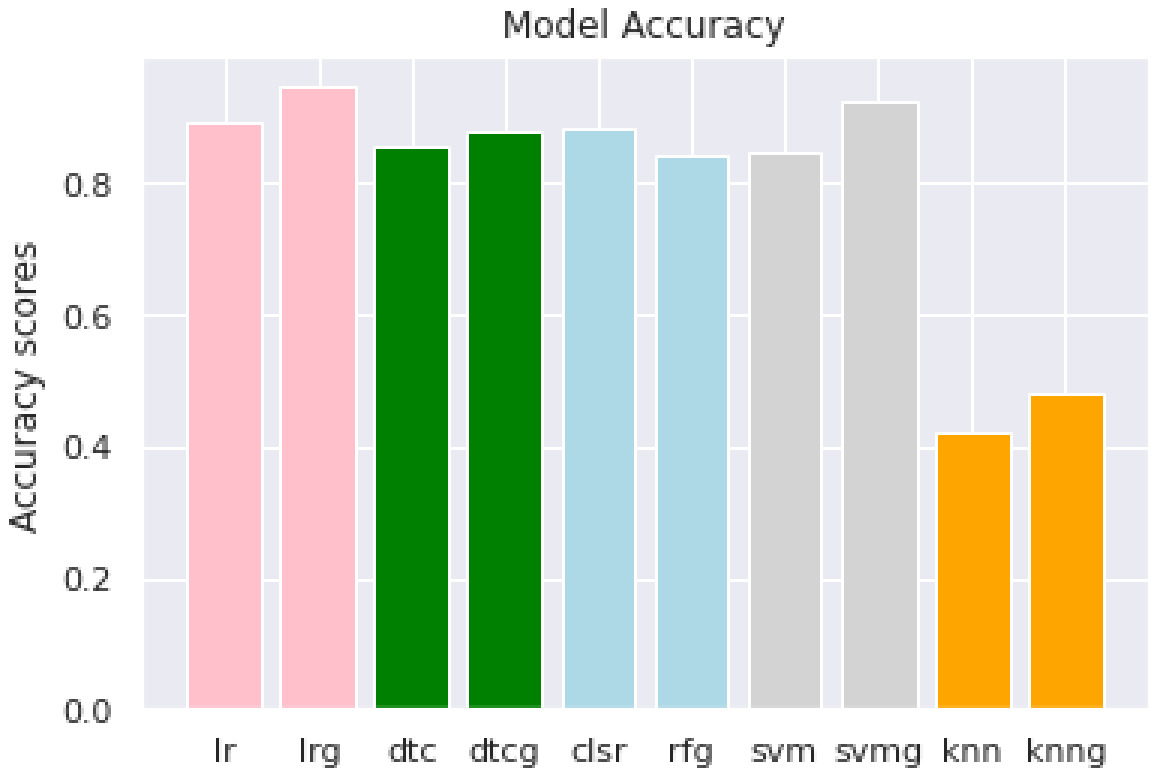
	precision	recall	f1-score	support
0	0.57	0.71	0.63	100
1	0.33	0.29	0.31	100
2	0.39	0.41	0.40	100
3	0.62	0.51	0.56	100
accuracy			0.48	400
macro avg	0.48	0.48	0.48	400
weighted avg	0.48	0.48	0.48	400

```
[[71 26  3  0]
 [43 29 22  6]
 [10 24 41 25]
 [ 0 10 39 51]]
```



AlmaBetter

Results



- Among the models, the optimization-based logistic regression model with an accuracy score of 94.75% and the SVM optimization model with an accuracy score of 92.5% both performed well.
- KNN was the model that did the poorest. The accuracy score improved from 42% to 48% even after optimization.

- Decision tree and random forest classifiers' performance with optimization was adequate.
- Logistic regression is effective with variables that have already been determined to be independent while SVM is effective with unstructured and semi-structured data, such as text and images.
- Logistic regression is based on statistical methods whereas SVM is based on the geometrical characteristics of the data.
- Logistic regression and SVM with a linear kernel both were effective, however depending on features, logistic regression model proved to be more effective than the others.

Conclusion

- According to EDA, there are mobile phones available in four pricing groups.
- Ram's is the deciding factor for price range and it continuously rises as it moves from low to high cost
- In determining the pricing range of a mobile phone, RAM, battery life and pixels were important factors.
- Around 50% of the mobiles have Bluetooth. As the price range widens, battery capacity gradually increases. Expensive phones now weigh less.
- Based on the results of the aforementioned trials , it has been concluded that logistic regression model and SVM optimized model performed effectively. Confusion matrix evaluation is used to assess model performance and accuracy.

Thank You