Capstone Project-3

Mobile Price range Prediction

(Supervised Machine Learning Regression)

By
Chand Kamble.

Points To Discuss.

- Problem Statement
- Data Overview
- Insights For Dataset
- Data wrangling
- EDA
- Model selection
- Hyperparameter Tuning
- Feature Importance
- Conclusions

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 mobile phones (eg: Ram,internal memory,etc) and its selling price, in this problem we don't have to predict the actual price but the price range indicating that how high the price is.

- 0- Low cost phones
- 1- medium cost phones
- 2- high cost phones
- 3- very high cost phones

This will basically help companies to estimate price of mobiles to give tough competition to other mobile manufacturer.

Also, it will be useful for consumers to verify that they are paying best price for a mobile.

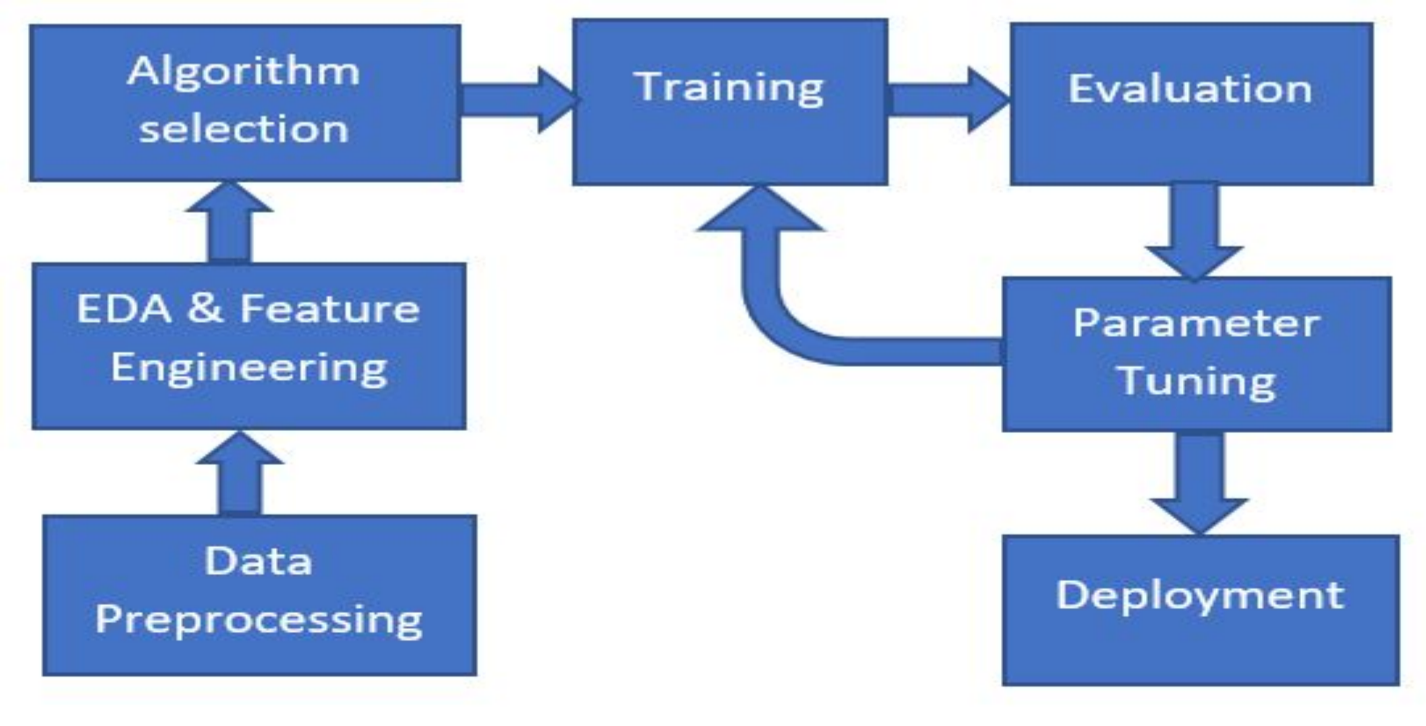
Data Overview.

- **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 megapixels
- **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 megapixels

Data Overview (Cont..)

- **Px_height** - Pixel Resolution Height
- **Px_width** - Pixel Resolution Width
- **Ram** - Random Access Memory in MegaBytes
- **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
- **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).

Workflow


Data Preprocessing
EDA & Feature Engineering
Algorithm selection
Training
Evaluation
Parameter Tuning
Deployment

Insights For Dataset

- Dataset Contains 2000 Rows and 20 Columns.

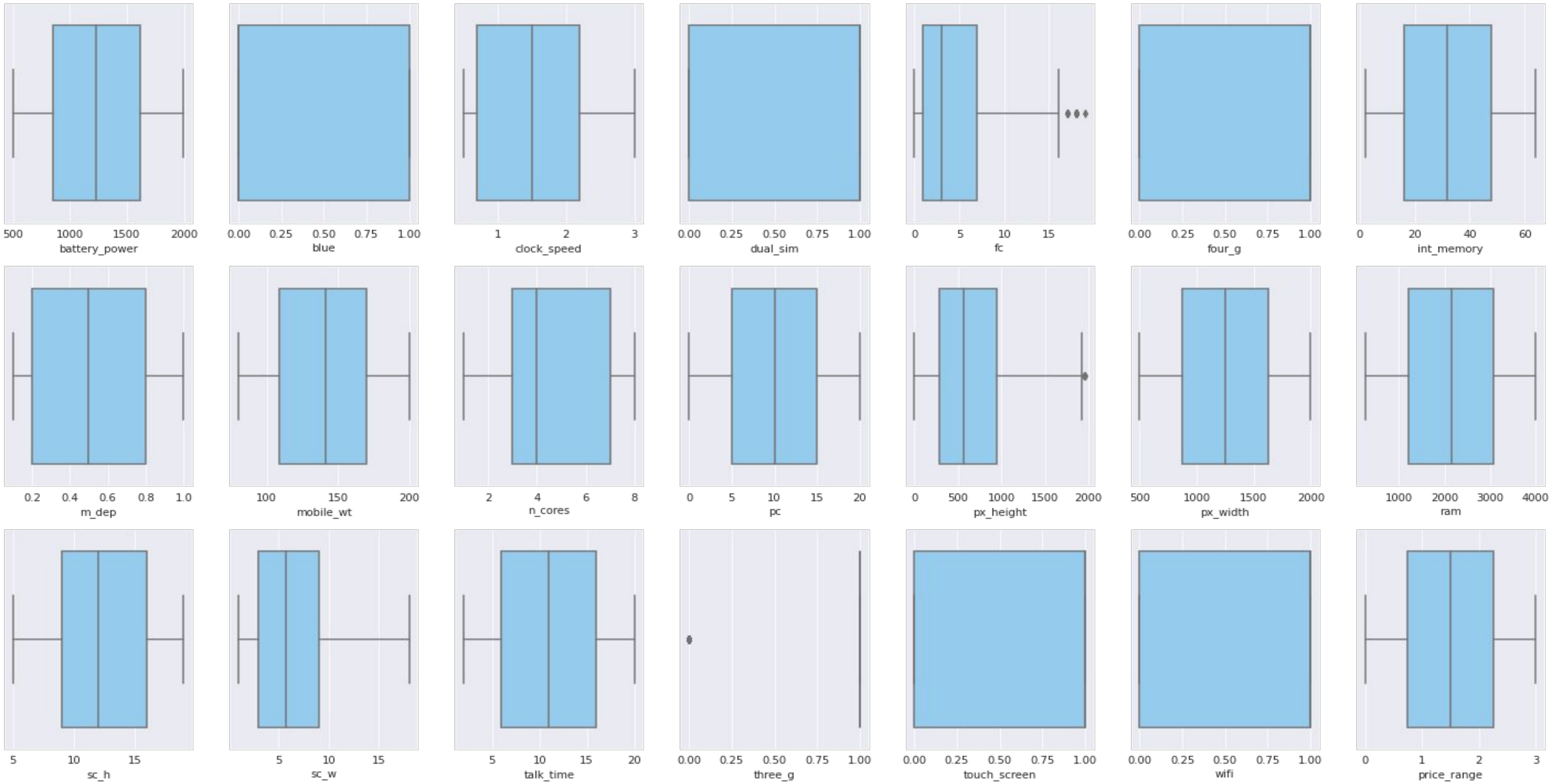
- **Categorical Feature**: Blue, Dual Sim, Four-g,Three-g,Touch-Screen,Wifi,Price Range.
- **Numerical Feature**: Battery Power, Clock Speed, Int-Memory, Fc ,M-Depth, Mobile Weight,N-Cores,Pc, Px-Height, Px-Width, Ram, Sc-h, Sc-w, Talktime.

- There Are No Missing Values Present And Duplicate Value Present.

- The Px-Height and Sc-Weight Had Some Zero Values Which We Had To Remove Before Proceeding Further As These Values Cannot Be Zero In Real Life.

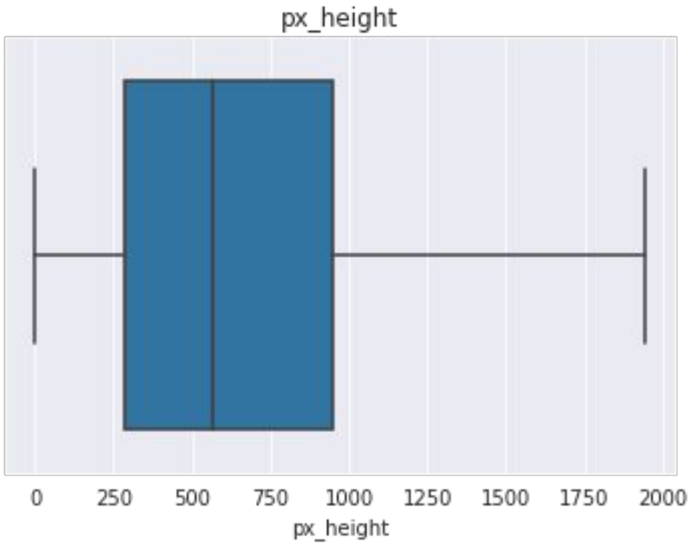
Data wrangling.

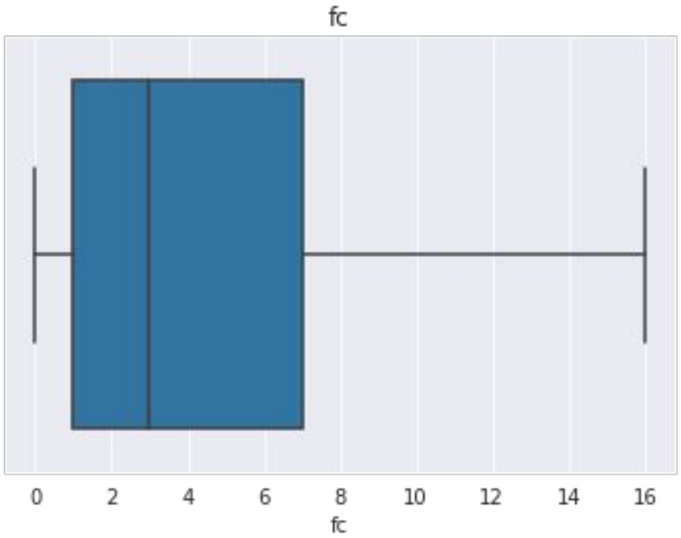
- **Checking outliers in numerical variables.**



Data Wrangling.

- After removing outliers.

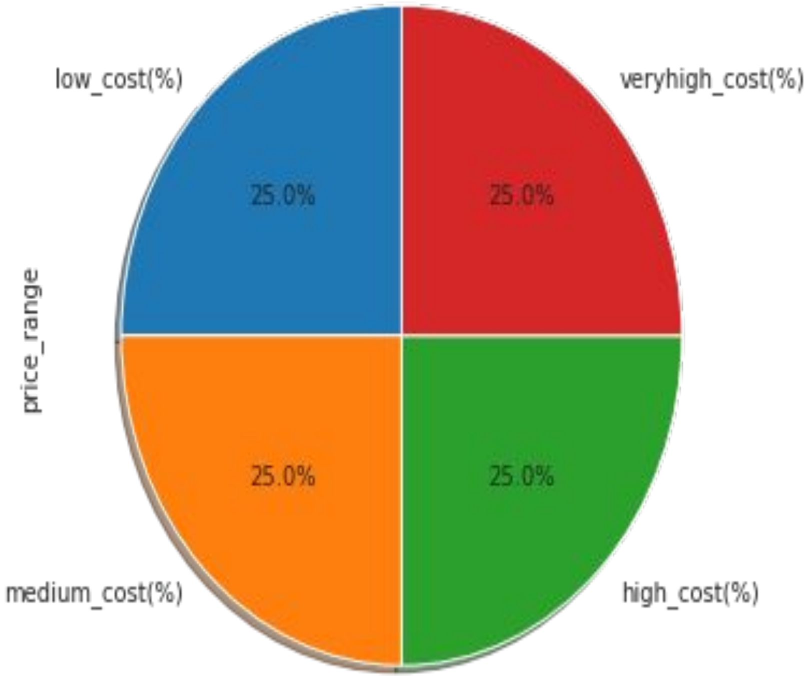




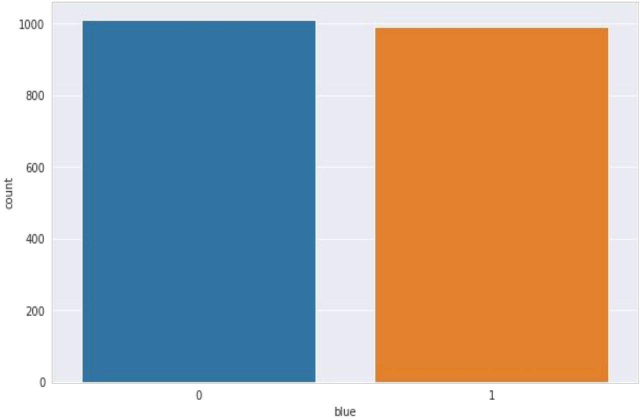
EDA

Target variable/Dependent variable.

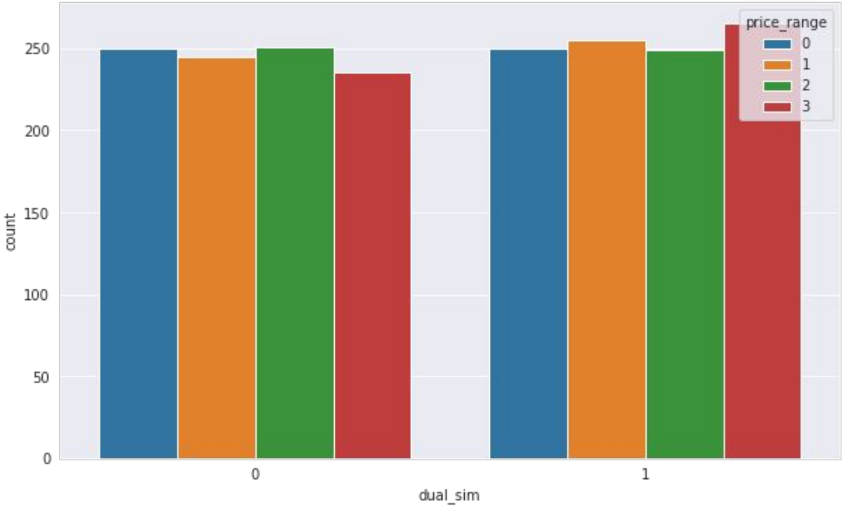
- **we have records of 2000 mobiles with 20 columns/features.**
- **we have perfectly balanced dataset with 500 observations in each class.**



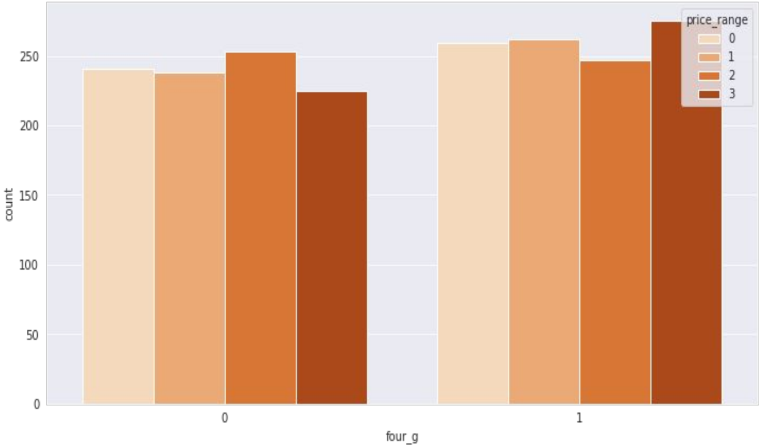
EDA



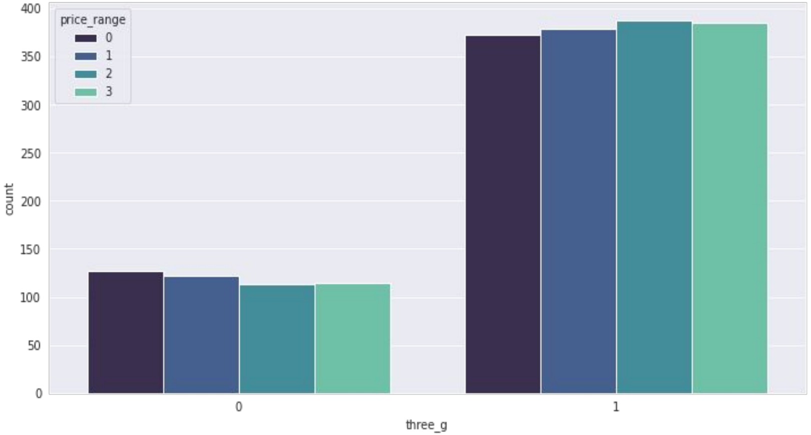
1000
800
600
400
200
0
count
0
1
blue



250
200
150
100
50
0
count
0
1
dual_sim
price_range
0
1
2
3



250
200
150
100
50
0
count
0
1
four_g
price_range
0
1
2
3

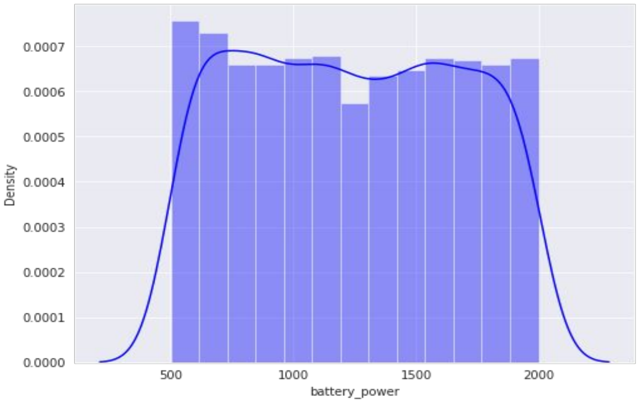


400
350
300
250
200
150
100
50
0
count
0
1
three_g
price_range
0
1
2
3

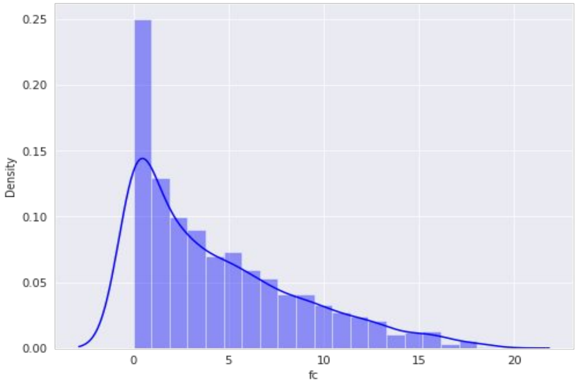
Observations:

- Half the devices have bluetooth and half don’t.
- Half of the phones here have a dual sim card slot the other half doesn’t.
- There are slightly more phone with 4g line compared to non 4g line .
- Phone with price range 3 has the most 4g line among all group
- Three quarter of the phones in this data set has a 3g line,while quarter of it doesn’t even have 3g.

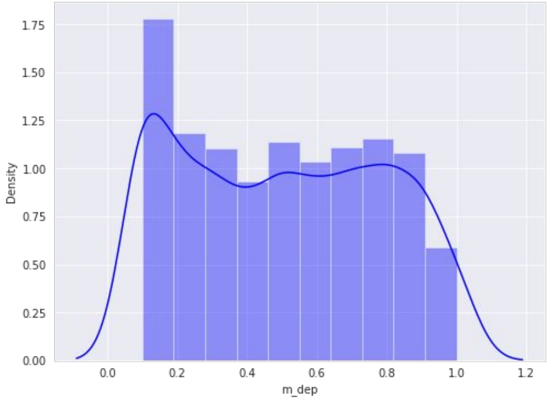
EDA



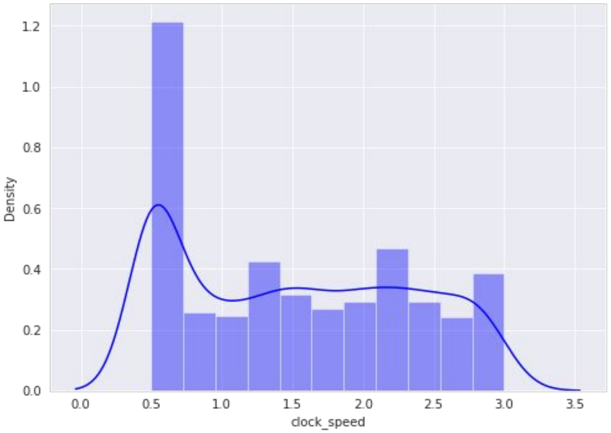
Density
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
500
1000
1500
2000
battery_power



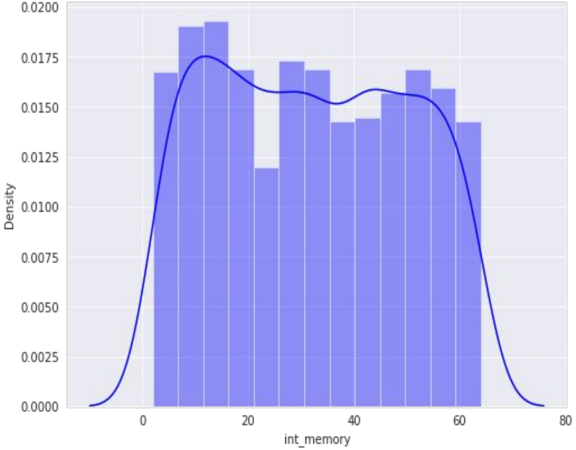
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
fc



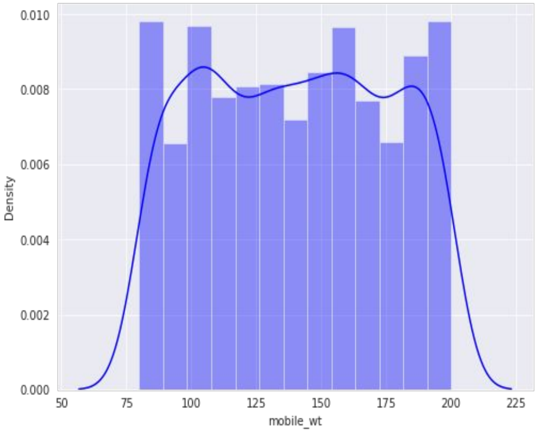
Density
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
1.2
m_dep



Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
clock_speed

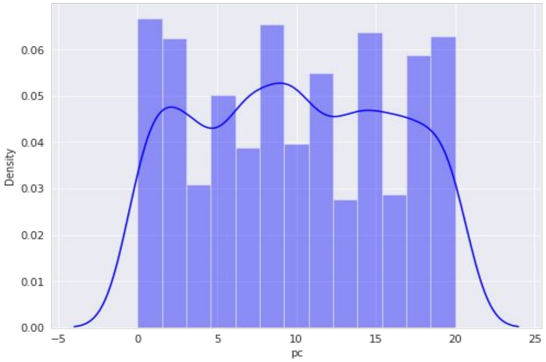


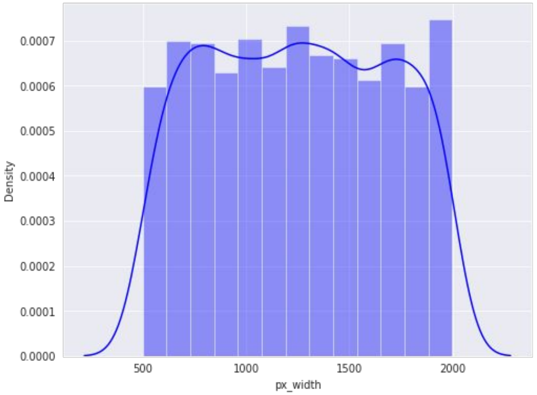
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
20
40
60
80
int_memory

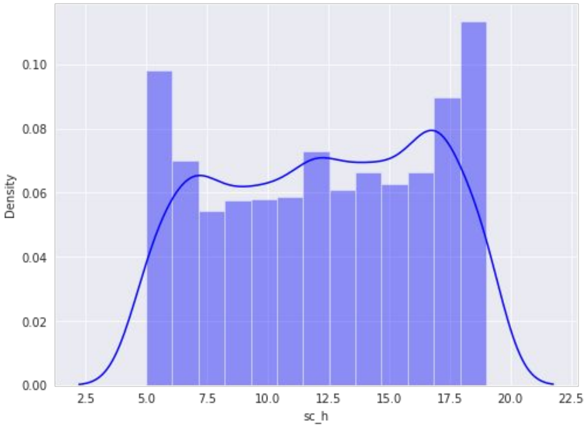


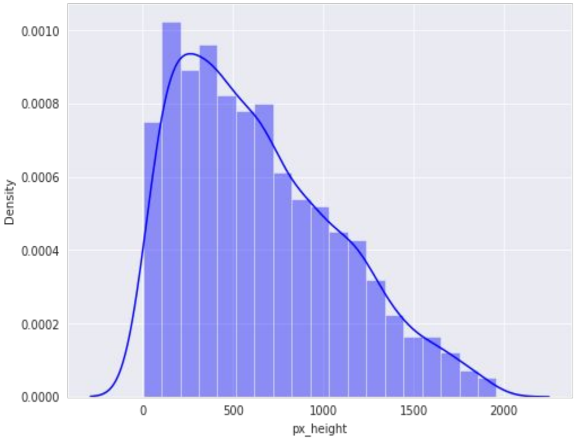
Density
0.010
0.008
0.006
0.004
0.002
0.000
50
75
100
125
150
175
200
225
mobile_wt

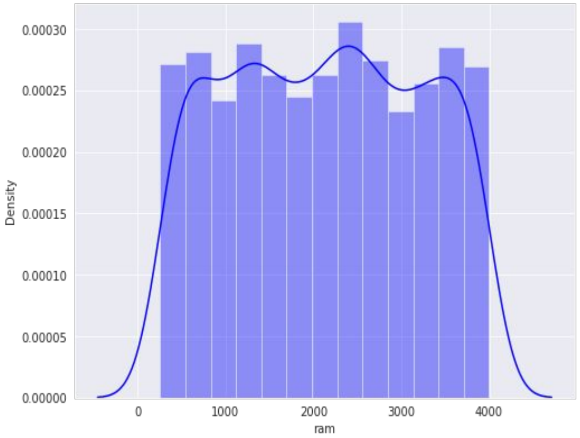
EDA

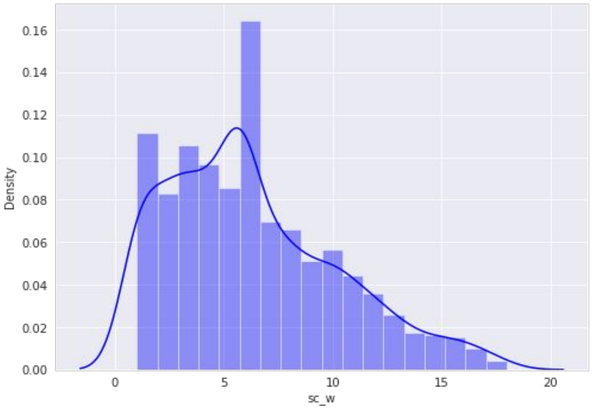

Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
−5
0
5
10
15
20
25
pc


Density
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
0.0006
0.0007
500
1000
1500
2000
px_width

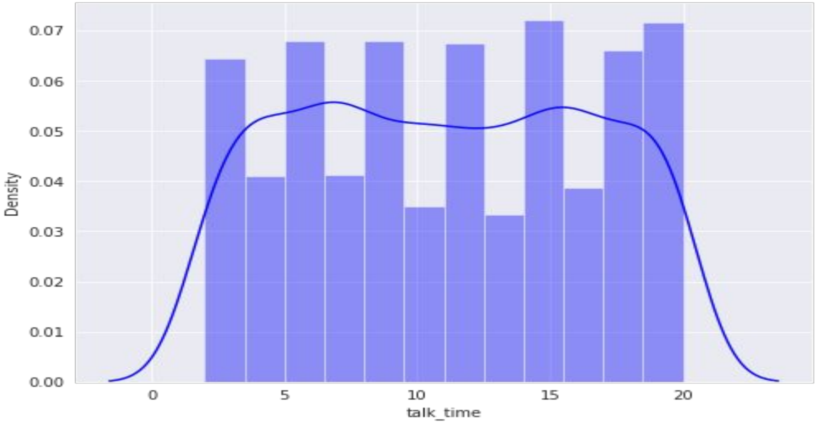

Density
0.00
0.02
0.04
0.06
0.08
0.10
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
sc_h

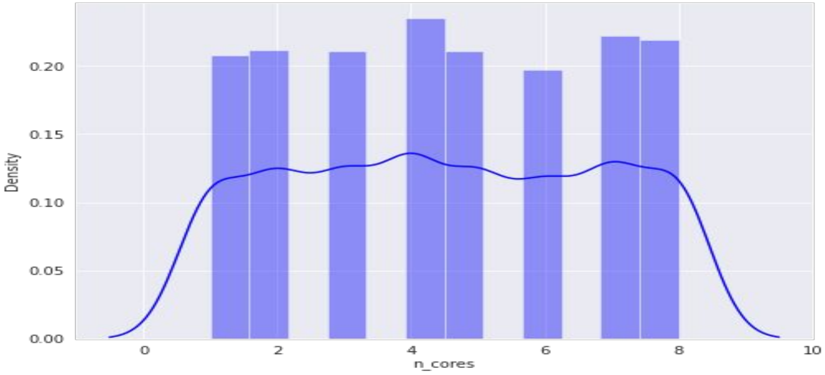

Density
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0
500
1000
1500
2000
px_height


Density
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025
0.00030
0
1000
2000
3000
4000
ram


Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0
5
10
15
20
sc_w

EDA

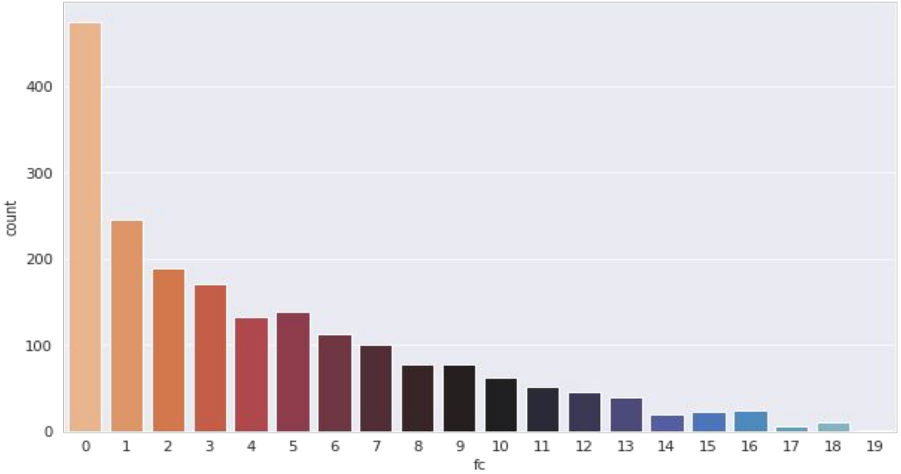


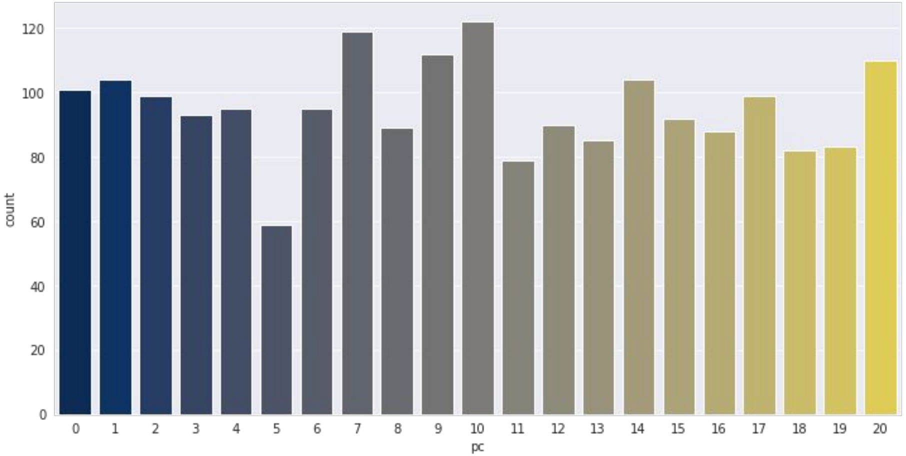


- Data is well distributed.
- fc and px_height has some outliers.
- Many features have an almost uniform distribution which means there are products available almost equally in all sizes/values.
- Battery power, internal_memory, Ram are almost normally distributed with respect to their set of feature values and also around 2.5gb has most count in ram similarly others.
- In second row pixel_height is slightly positively skewed, pixel_width is normally distributed.
- In third row screen_height and screen_size are normally distributed unlike screen_width which is slightly positive skewed.
- we can see that internal memory sizes doesn't have many difference when it comes to distribution.

EDA

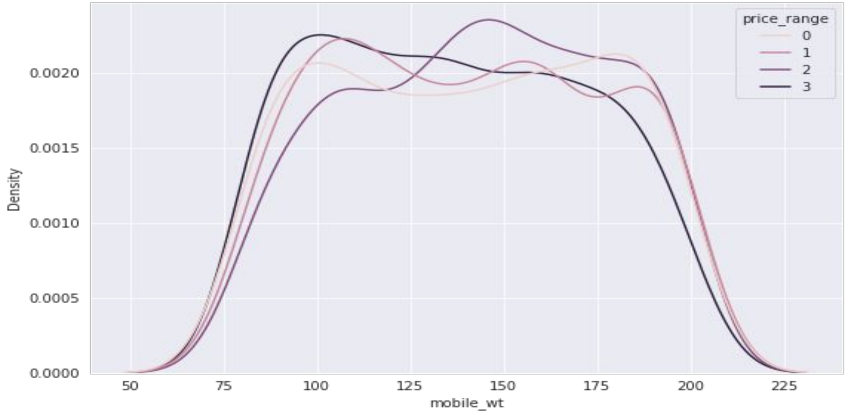
- Majority of phone in this dataset have a front facing camera,almost a quarter of the phone in this dataset doesn't have front facing camera .

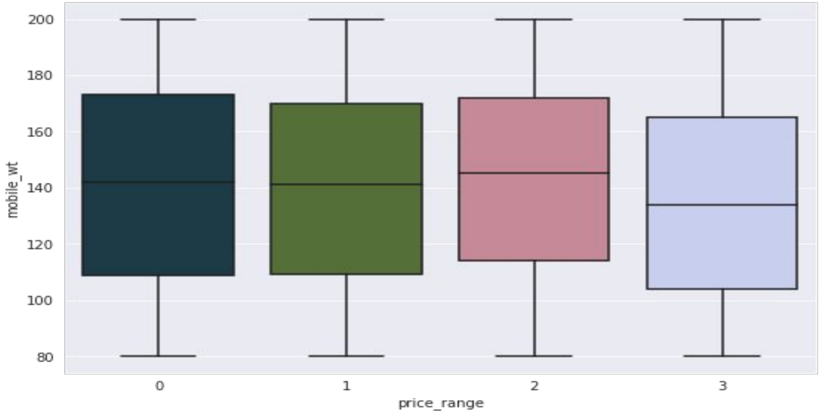


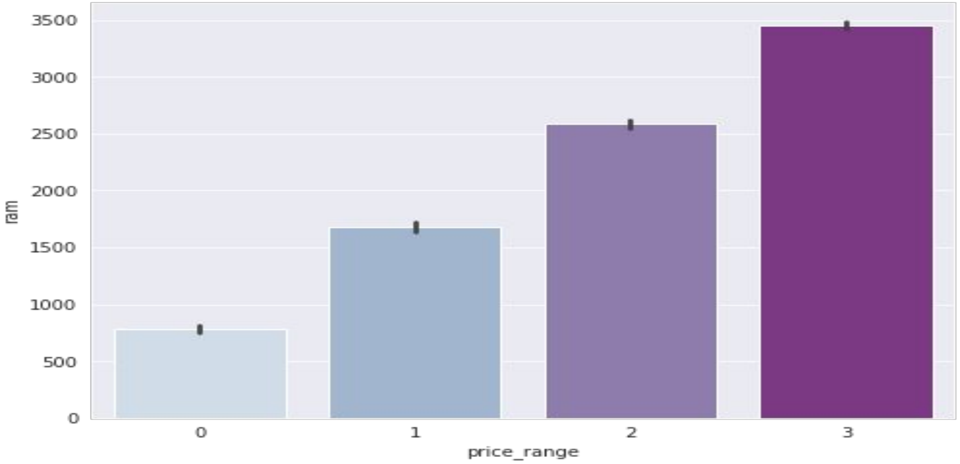


- 95% phone in this dataset has a primary camera (back camera)only 5 % of phones in this dataset does not have camera.

EDA (Price range vs Ram and Mobile weight).

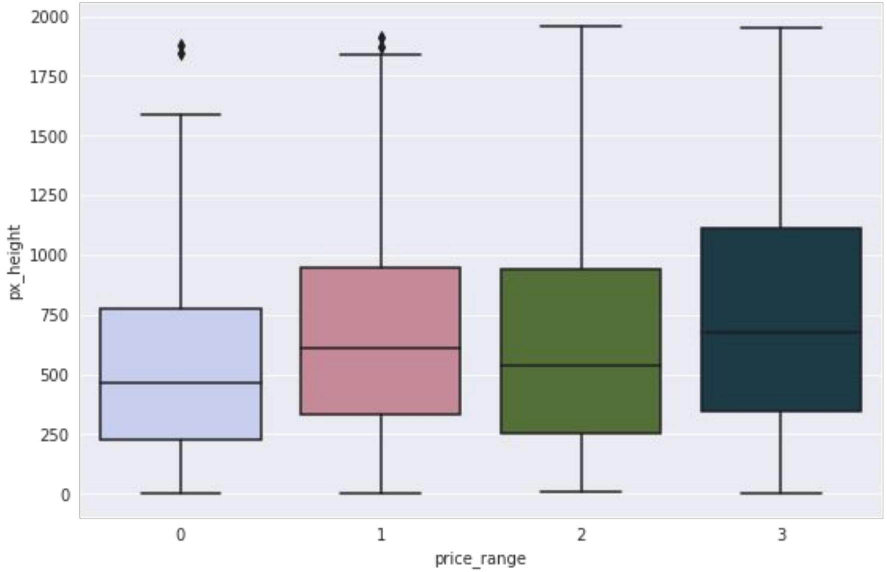


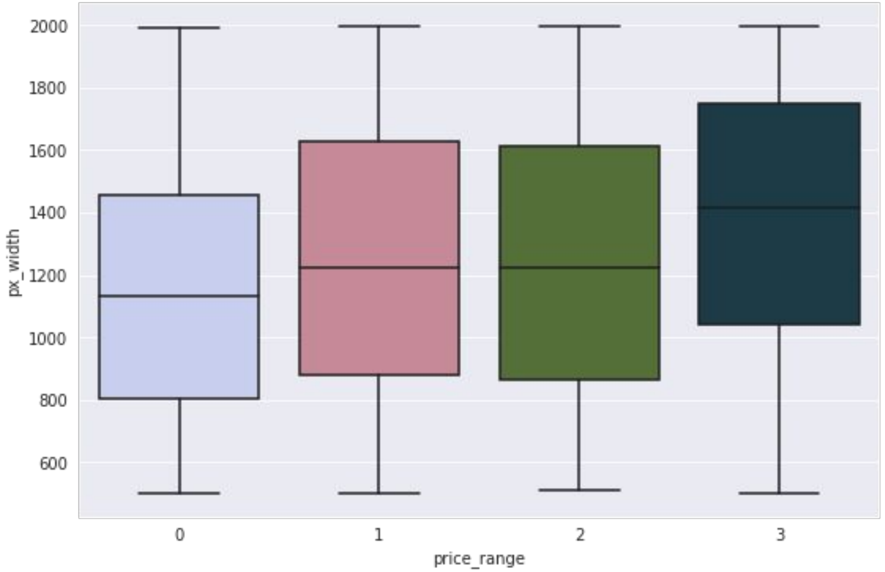




- Costly phones are lighter in weight.
- As from our analysis before that ramsize is highly correlated with the price range. the more expensive the price range the higher the ram.
- A phone have Ram has continuous increase with price range while moving from Low cost to Very high cost.

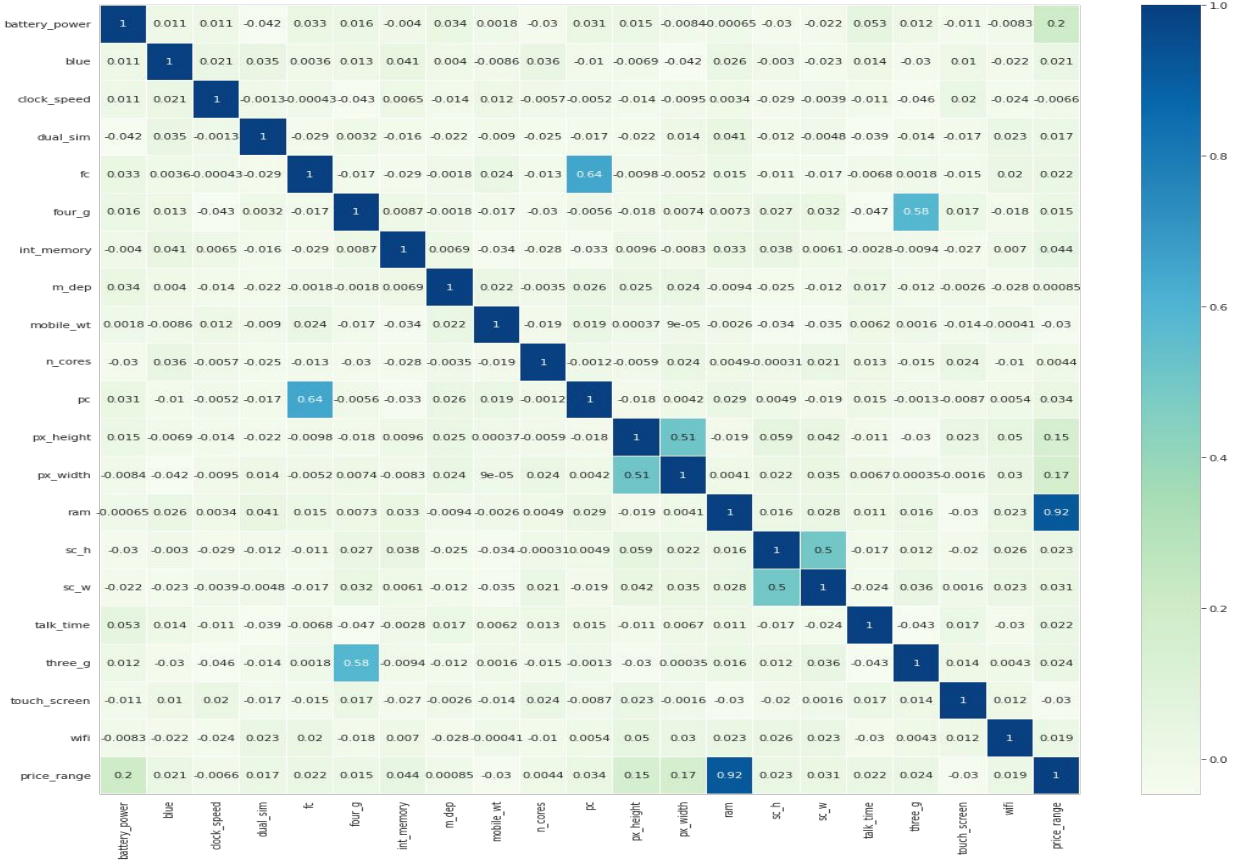
EDA (Price range vs px-height and px-width)





- There is not a continuous increase in pixel height as we move from Low cost to Very high cost. Mobiles with 'Medium cost' and 'High cost' has almost equal pixel width. so we can say that it would be a driving factor in deciding price_range.
- Pixel height is almost similar as we move from Low cost to Very high cost.little variation in pixel_height.

EDA (HEATMAP).



- From the graph RAM and price_range is highly correlated.
- There is some collinearity in feature pairs ('pc', 'fc') and ('px_width', 'px_height').
- Also, if px_height increases, pixel width also increases, that means the overall pixels in the screen.
- We can replace these two features with one feature. Front Camera megapixels and Primary camera megapixels are different entities despite of showing colinearity. So we'll be keeping them as they are.

Model Selection and Evaluation.

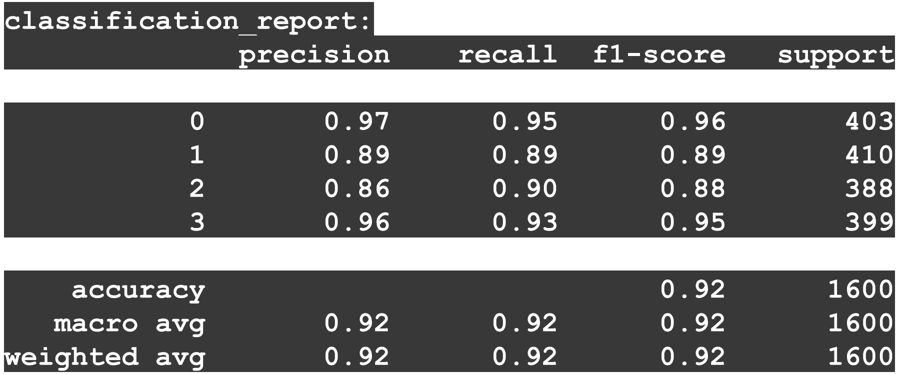
- Before Building The Model We Performed The Train Test Split .We Split The Training Data In 80% And Test Data In 20%.

- We Compared four Algorithms And Evaluated On Basis Of Overall Accuracy And The Recall Value Of Individual Classes.

1. Accuracy Score Is The Ratio Of Total Number Of Correct Predictions And Total Predictions.
2. The Recall Is The Measure Of Our Model Correctly Identifying True Positives.

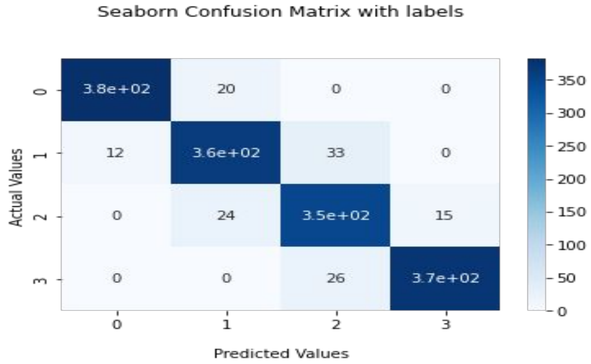
- Logistic Regression .
- Random Forest .
- support vector machine classifier.
- Decision Tree Classifier.

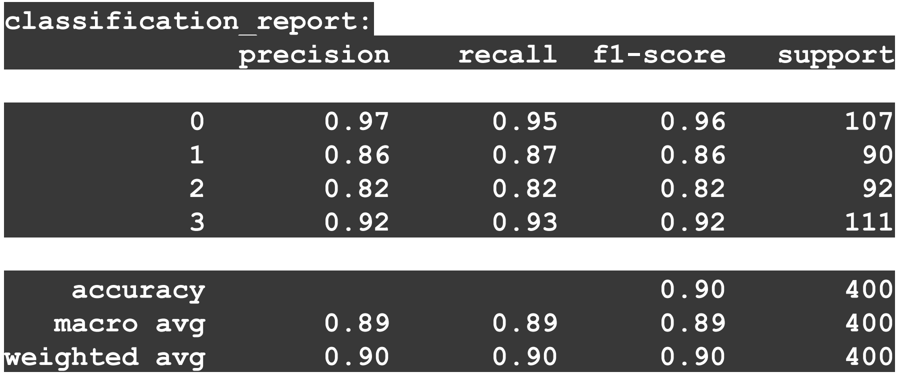
Implementing Logistic regression.



```
classification_report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       403
           1       0.89      0.89      0.89       410
           2       0.86      0.90      0.88       388
           3       0.96      0.93      0.95       399

    accuracy                           0.92      1600
   macro avg       0.92      0.92      0.92      1600
weighted avg       0.92      0.92      0.92      1600
```

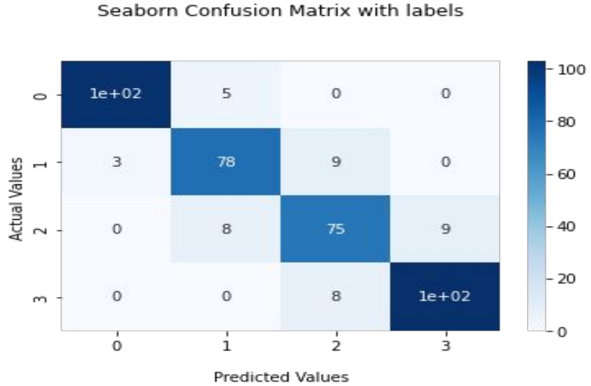
Train metrics





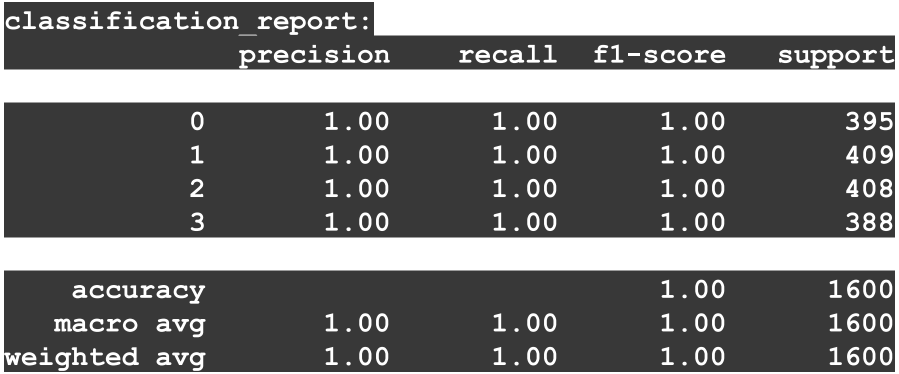
```
classification_report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       107
           1       0.86      0.87      0.86        90
           2       0.82      0.82      0.82        92
           3       0.92      0.93      0.92       111

    accuracy                           0.90       400
   macro avg       0.89      0.89      0.89       400
weighted avg       0.90      0.90      0.90       400
```

Test metrics



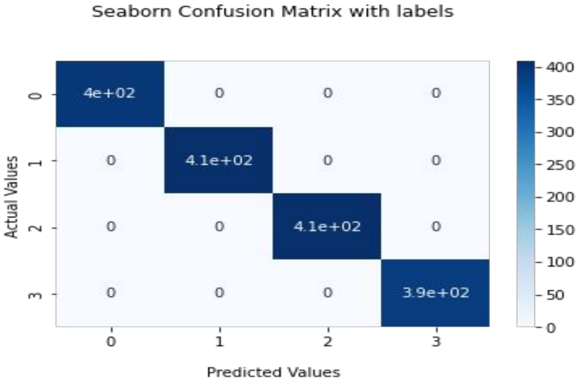
- Diagonal labels are true predicted labels all other labels are false predicted variables.

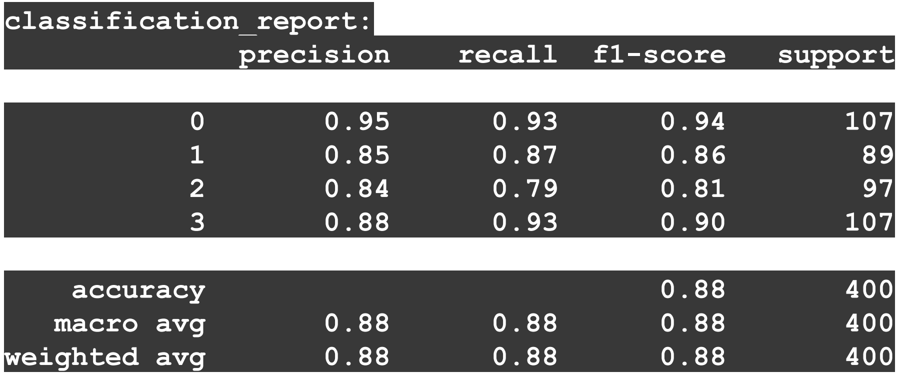
Implementing Random Forest.



```
classification_report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       395
           1       1.00      1.00      1.00       409
           2       1.00      1.00      1.00       408
           3       1.00      1.00      1.00       388

    accuracy                           1.00      1600
   macro avg       1.00      1.00      1.00      1600
weighted avg       1.00      1.00      1.00      1600
```

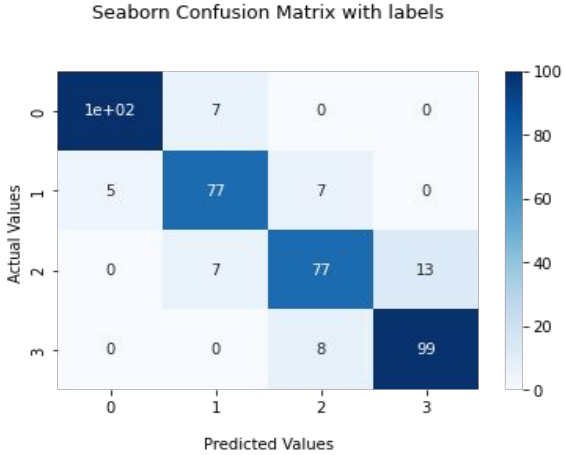
Train metrics



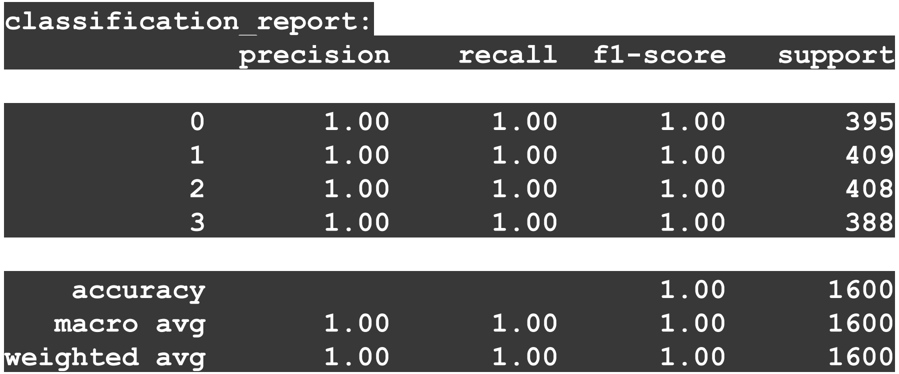


```
classification_report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       107
           1       0.85      0.87      0.86        89
           2       0.84      0.79      0.81        97
           3       0.88      0.93      0.90       107

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

Test metrics

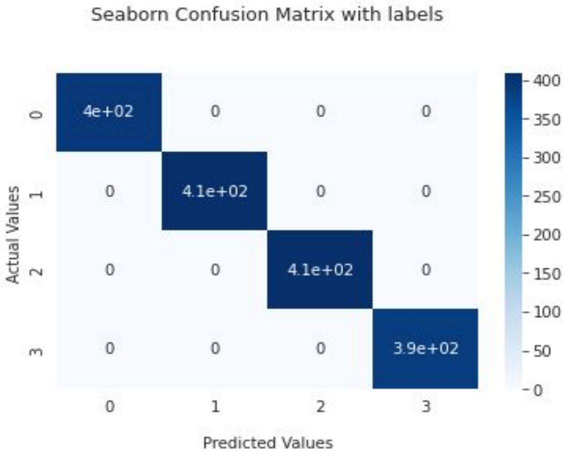


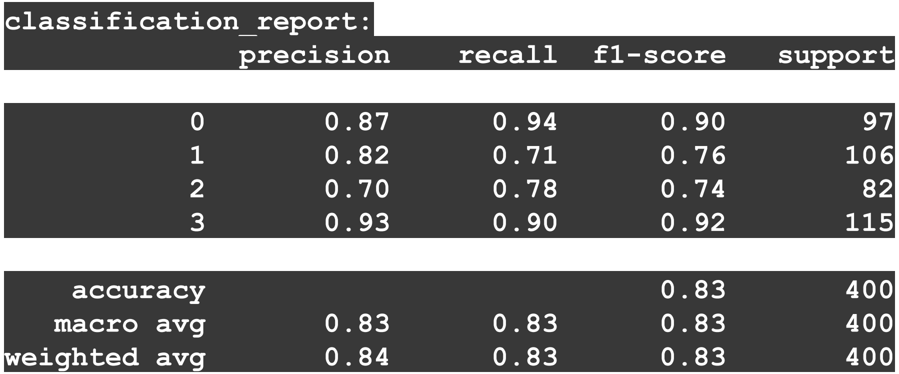
Implementing Decision Tree classifier.



```
classification_report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       395
           1       1.00      1.00      1.00       409
           2       1.00      1.00      1.00       408
           3       1.00      1.00      1.00       388

    accuracy                           1.00      1600
   macro avg       1.00      1.00      1.00      1600
weighted avg       1.00      1.00      1.00      1600
```

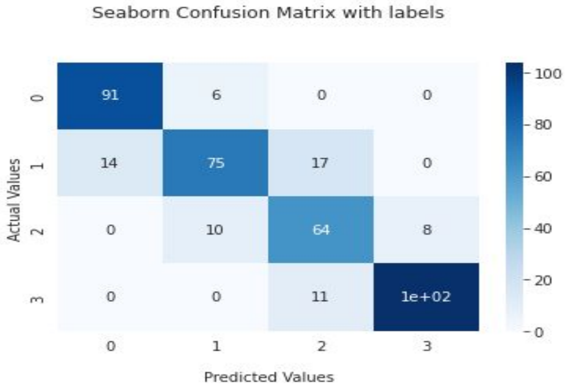
Train metrics



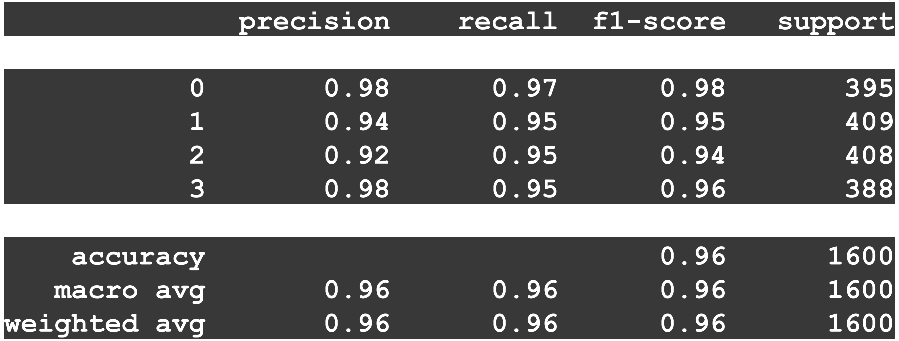


```
classification_report:
              precision    recall  f1-score   support

           0       0.87      0.94      0.90        97
           1       0.82      0.71      0.76       106
           2       0.70      0.78      0.74        82
           3       0.93      0.90      0.92       115

    accuracy                           0.83       400
   macro avg       0.83      0.83      0.83       400
weighted avg       0.84      0.83      0.83       400
```

Test metrics

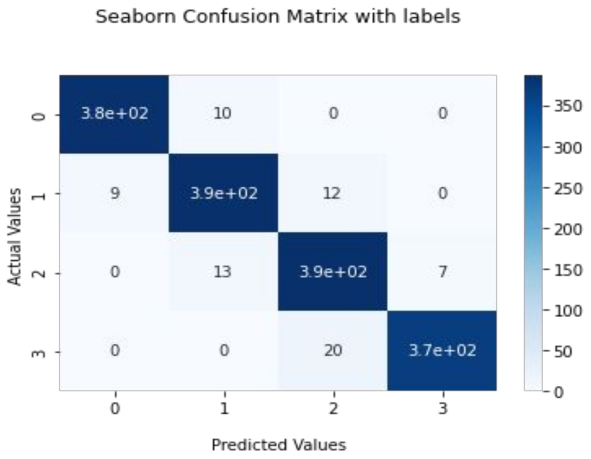


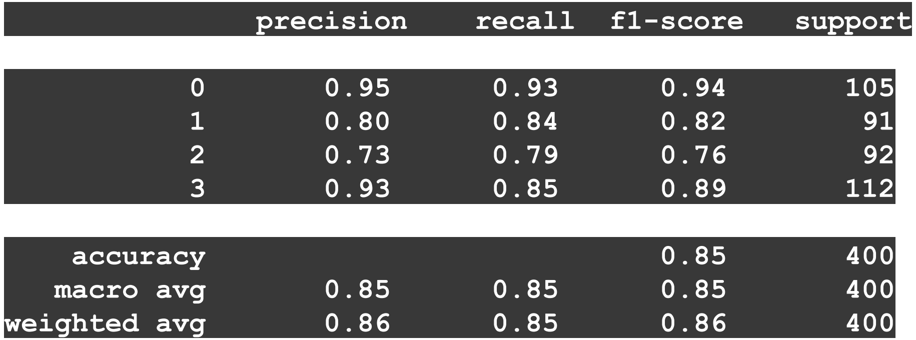
Implementing support vector machine classifier.



```
              precision    recall  f1-score   support

           0       0.98      0.97      0.98       395
           1       0.94      0.95      0.95       409
           2       0.92      0.95      0.94       408
           3       0.98      0.95      0.96       388

    accuracy                           0.96      1600
   macro avg       0.96      0.96      0.96      1600
weighted avg       0.96      0.96      0.96      1600
```

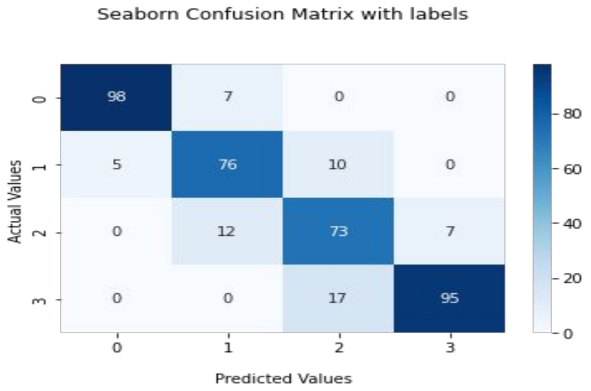
Train metrics



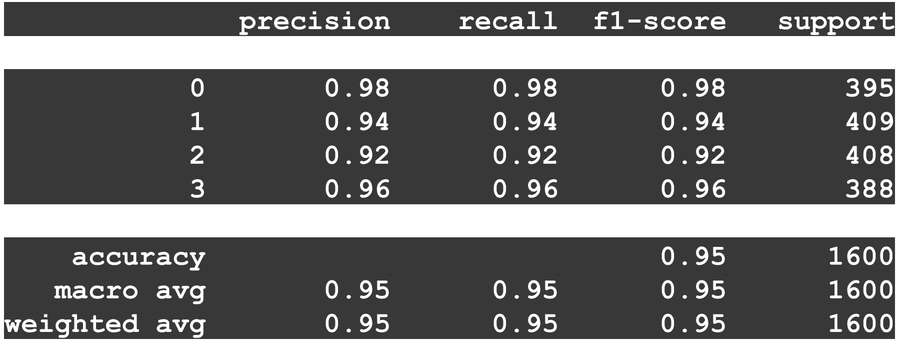


```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       105
           1       0.80      0.84      0.82        91
           2       0.73      0.79      0.76        92
           3       0.93      0.85      0.89       112

    accuracy                           0.85       400
   macro avg       0.85      0.85      0.85       400
weighted avg       0.86      0.85      0.86       400
```

Test metrics

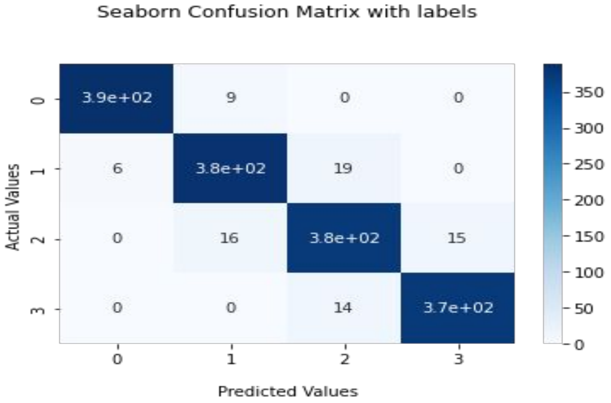


Best Model after Hyperparameter Tuning. ie (logistic regression).



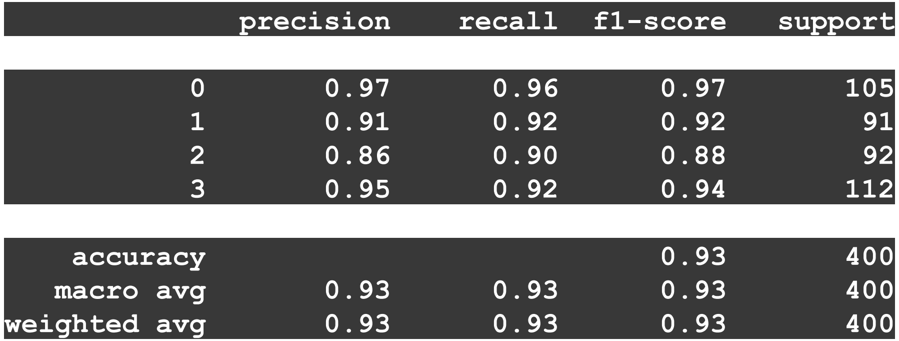
```
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       395
           1       0.94      0.94      0.94       409
           2       0.92      0.92      0.92       408
           3       0.96      0.96      0.96       388

    accuracy                           0.95      1600
   macro avg       0.95      0.95      0.95      1600
weighted avg       0.95      0.95      0.95      1600
```

Train metrics



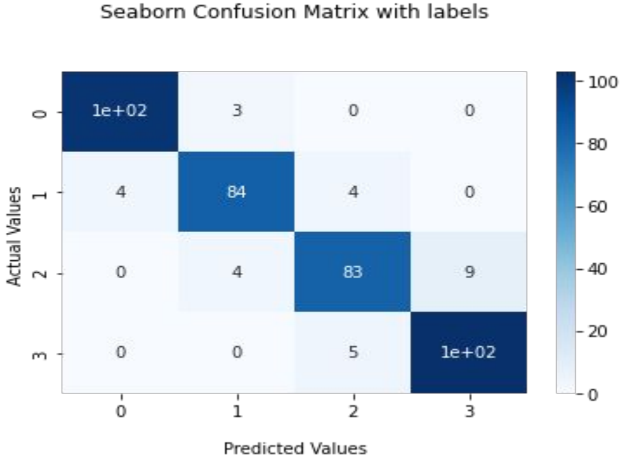
Seaborn Confusion Matrix with labels

Actual Values \ Predicted Values	0	1	2	3
0	3.9e+02	9	0	0
1	6	3.8e+02	19	0
2	0	16	3.8e+02	15
3	0	0	14	3.7e+02



```
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       105
           1       0.91      0.92      0.92        91
           2       0.86      0.90      0.88        92
           3       0.95      0.92      0.94       112

    accuracy                           0.93       400
   macro avg       0.93      0.93      0.93       400
weighted avg       0.93      0.93      0.93       400
```

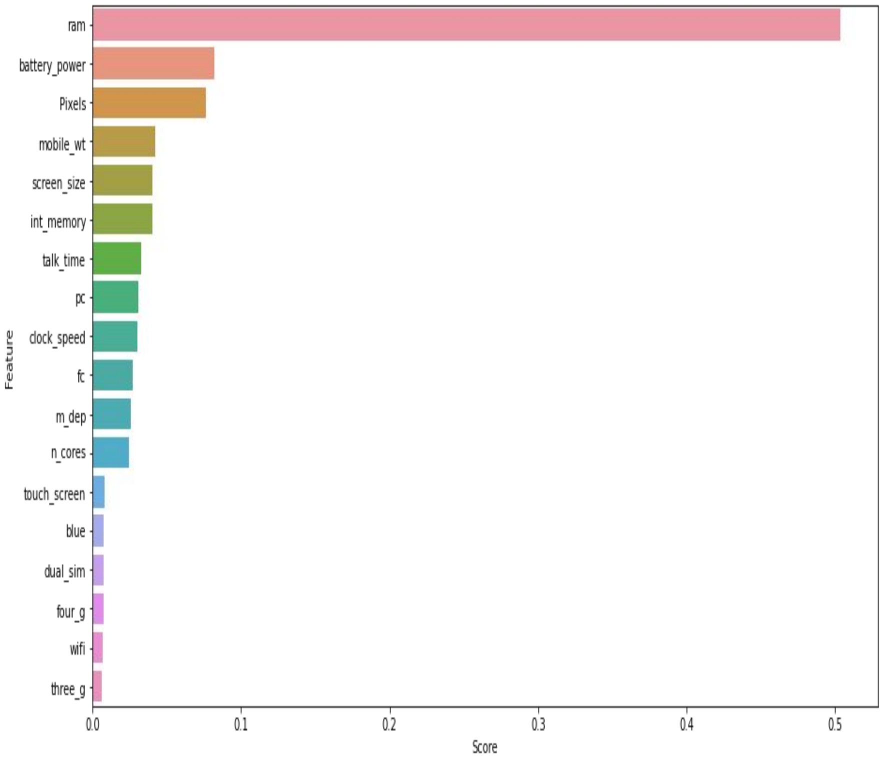
Train metrics



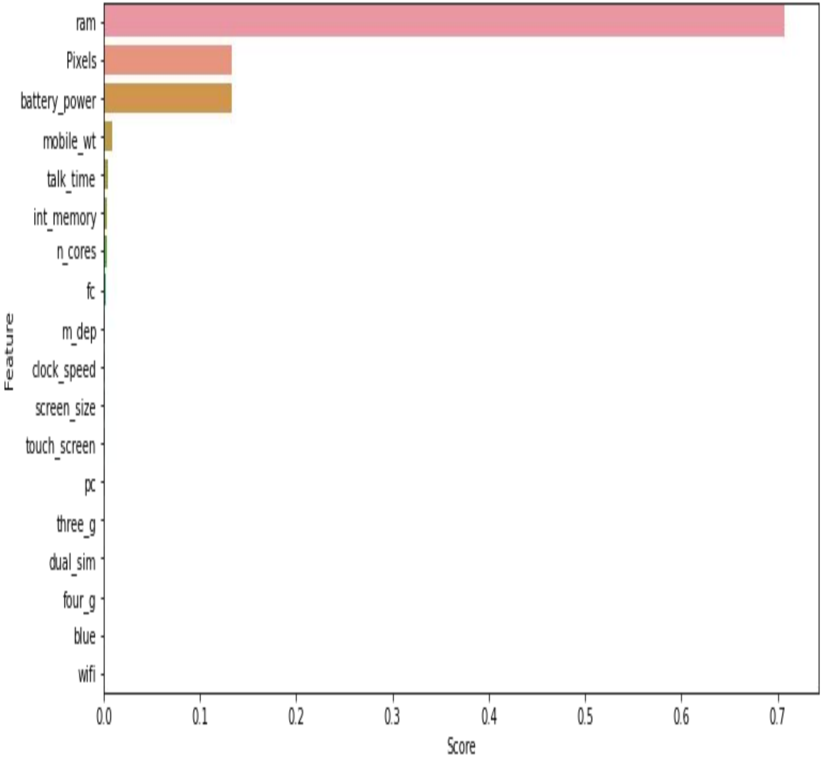
Seaborn Confusion Matrix with labels

Actual Values \ Predicted Values	0	1	2	3
0	1e+02	3	0	0
1	4	84	4	0
2	0	4	83	9
3	0	0	5	1e+02

Feature Importance.

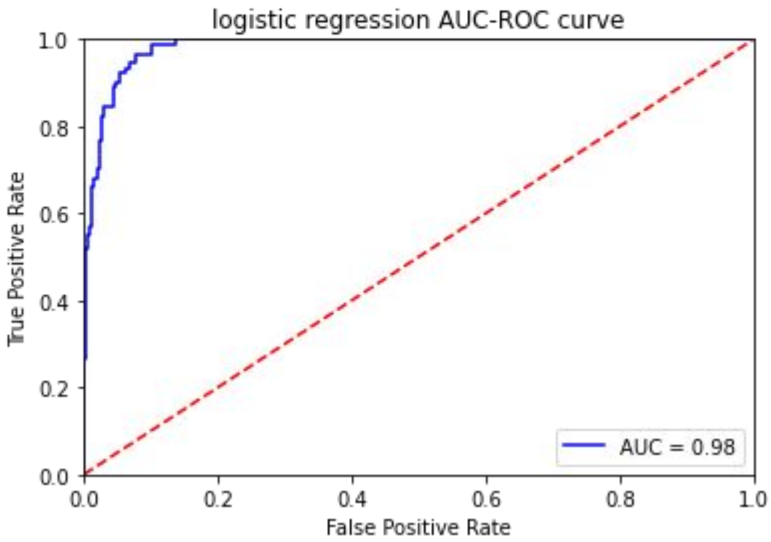


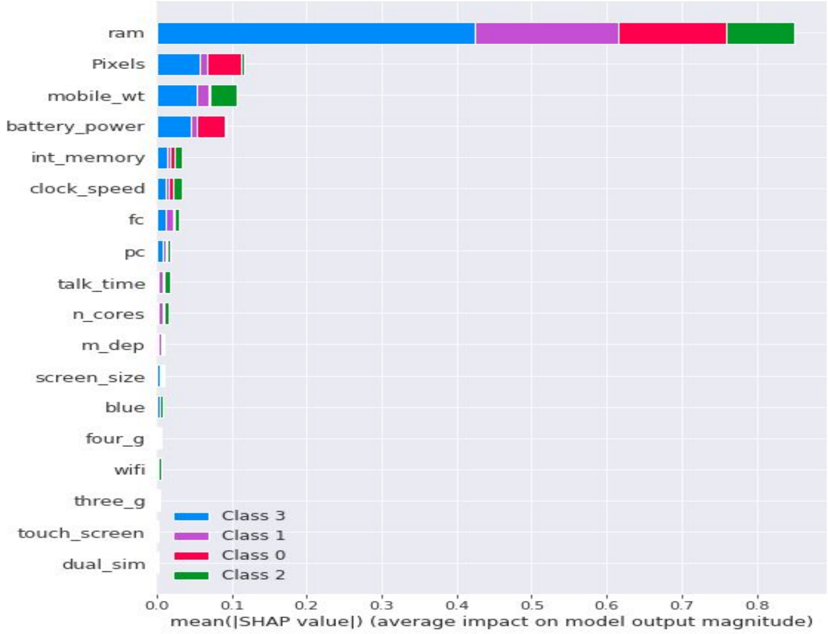
Random Forest classifier.



Decision Tree classifier.

Model Explanation.





- ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.An AUC of 0.98 would actually mean that lets say we take two data points belonging to separate classes then there is 0.98% chance model would be able to segregate them or rank order them correctly.
- In this above plot shows that ram is the most important features in predicting price range.
- Among all the classes of price range class 3 gets most affected by the ram features.

Conclusions :

- We Started with Data understanding, data wrangling, basic EDA where we found the relationships, trends between price range and other independent variables.
- From EDA we can see that here are mobile phones in 4 price ranges. The number of elements is almost similar.
- Half the devices have Bluetooth, and half don’t.
- there is a gradual increase in battery as the price range increases.
- Ram has continuous increase with price range while moving from Low cost to Very high cost.
- Costly phones are lighter.
- RAM, battery power, pixels played more significant role in deciding the price range of mobile phone.
- Implemented various classification algorithms, out of which the logistic regression algorithm gave the best performance after hyper-parameter tuning with 95% train accuracy and 93 % test accuracy.
- SVM is the second best good model which gave good performance after hyper-parameter tuning with 94% train accuracy and 92% test accuracy score.
- We checked for the feature importance of each model. RAM, Battery Power, Px_height and px_width contributed the most while predicting the price range.
- From all the above experiments we can conclude that in logistic regression we got the best results .

Thank you