Capstone Project - 3

Mobile Price Range Prediction

By

Ishan Singh

Points to discuss

- Introduction
- Problem statement
- Data description and summary
- Exploratory data analysis
- Heat map
- Models
  - ○ Logistic regression
  - ○ Decision tree
  - ○ Random forest classifier
  - ○ Xgboost classifier
- Challenges faced
- Conclusion

Introduction

- Mobile phones are now like an essential commodity for us to connect to the world or our loved ones, resulting in the huge amount of mobile phone manufactured, hence; huge amount of data being generated.
- Mobile phone prediction helps in deciding the range of a mobile phone depending upon its specifications, as the most expensive mobile phone will be loaded with a lot more and better features than the cheap ones.
- This insight can help deciding the specification for a mobile phone at industry level.

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.

Data description and summary

The dataset contains information regarding mobile features , specification and their price range. The dataset contains 2000 rows and 21 columns.

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

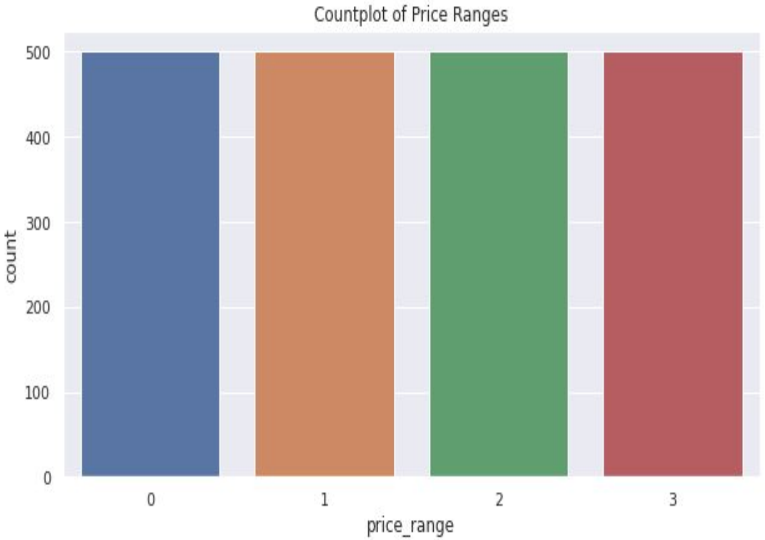
Data description and summary(contd.)

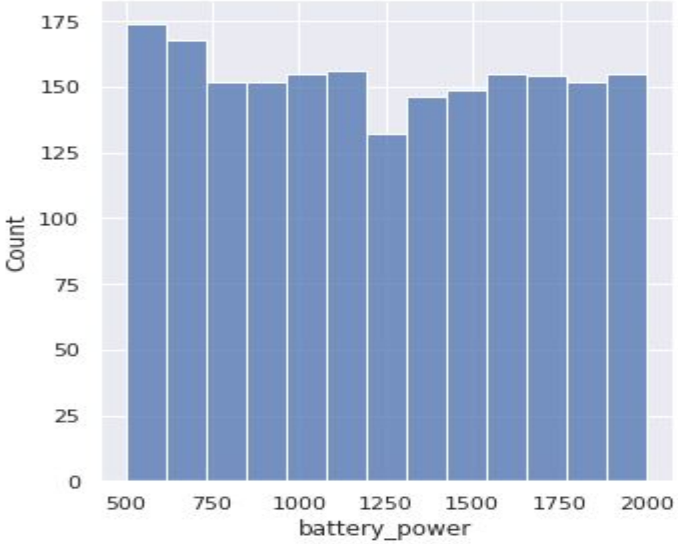
- Int_memory - Internal Memory in Gigabytes
- M_dep - Mobile Depth in cm
- Mobile_wt - Weight of mobile phone
- N_cores - Number of cores of processor
- Pc - Primary Camera megapixels
- Px_height - Pixel Resolution Height
- Px_width - Pixel Resolution Width
- Ram - Random Access Memory in Mega Bytes
- Sc_h - Screen Height of mobile in cm
- Sc_w - Screen Width of mobile in cm

Data description and summary(contd.)

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

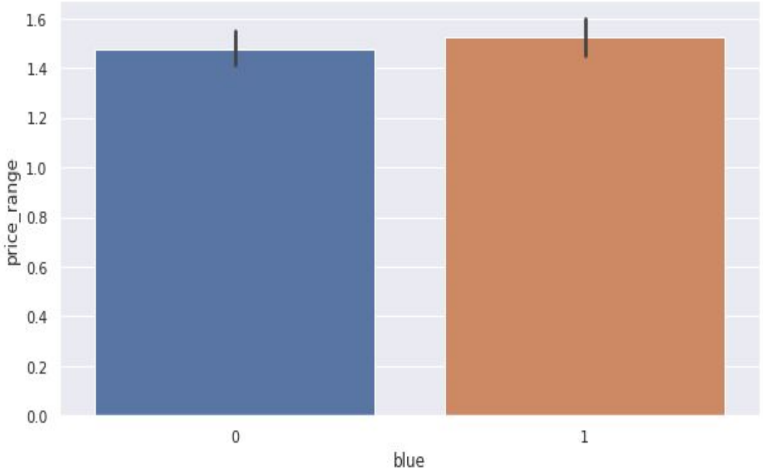
Exploratory Data Analysis(EDA)

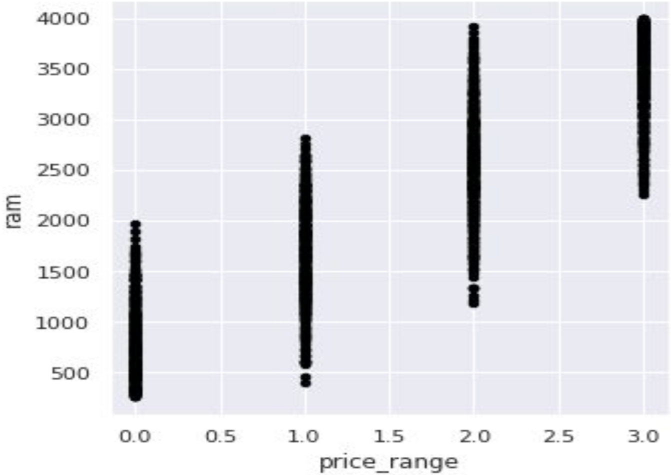




- There are mobile phones in 4 price ranges ,the number of elements is almost similar.
- We can see through graph that there is gradual increase as the price range increases.

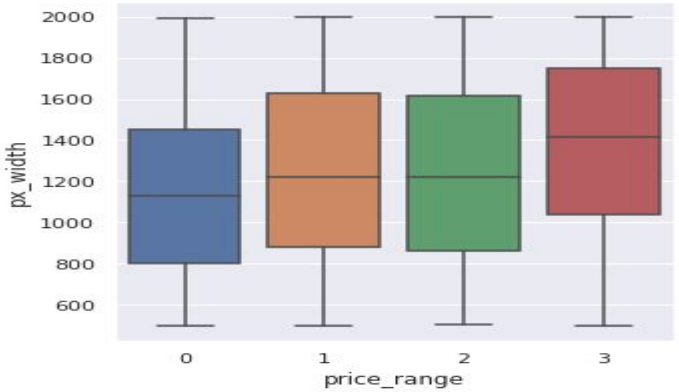
EDA(contd.)

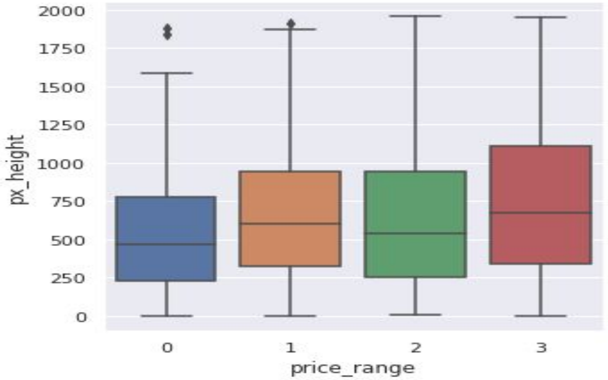




- Almost half of the devices have Bluetooth and half don’t
- Through scatter plot we can see that Ram has continuous increases with price range while moving from low to very high cost

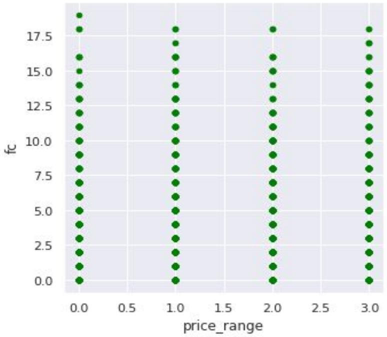
EDA(contd.)

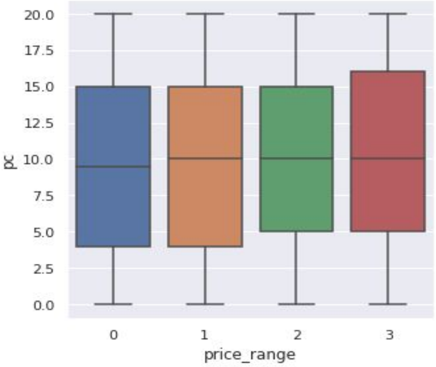


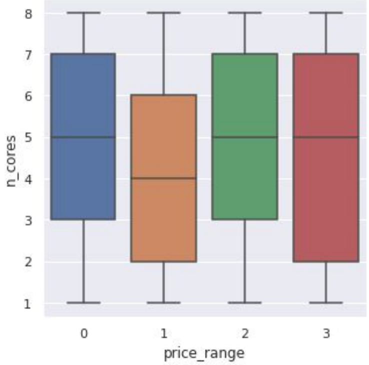


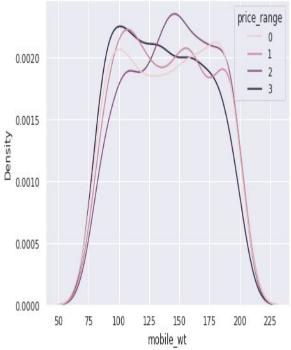
- There is no continuous increase in pixel width as we move from low to very high cost .Mobiles with medium cost and high cost has almost equal pixel width so we can say that it would be a best factor in deciding the price range.
- Pixel height is almost similar as we move from low cost to very high cost and there’s little variation in pixel height.

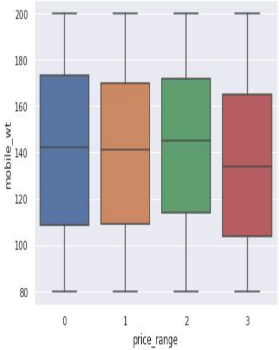
EDA(contd.)





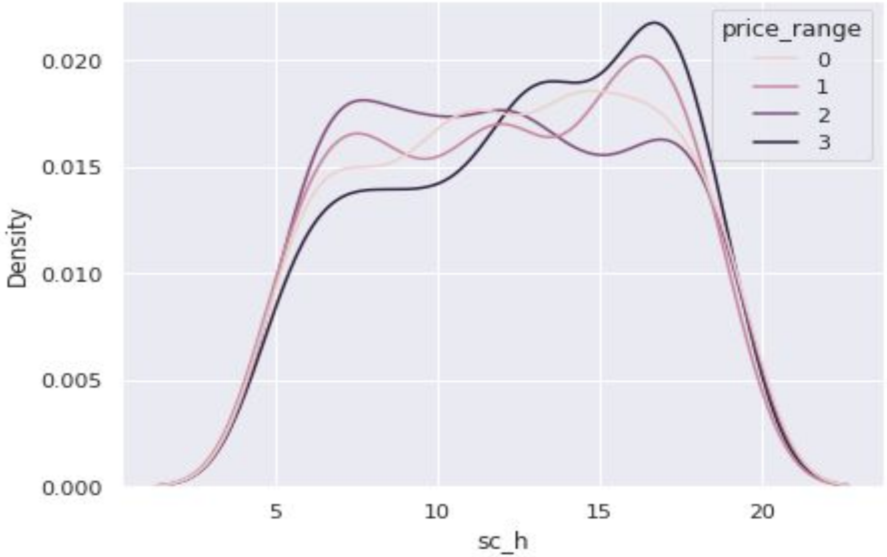


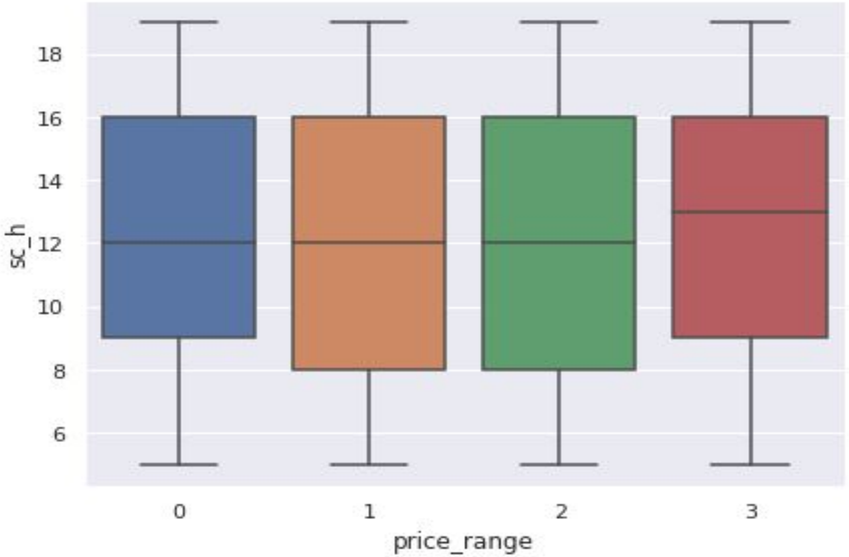




- This feature distribution is almost similar along with all the price range variable ,it may not be helpful for prediction.
- Primary camera megapixel are showing a little variation along with target categories ,which is good for prediction.
- Costly phone are light weight.

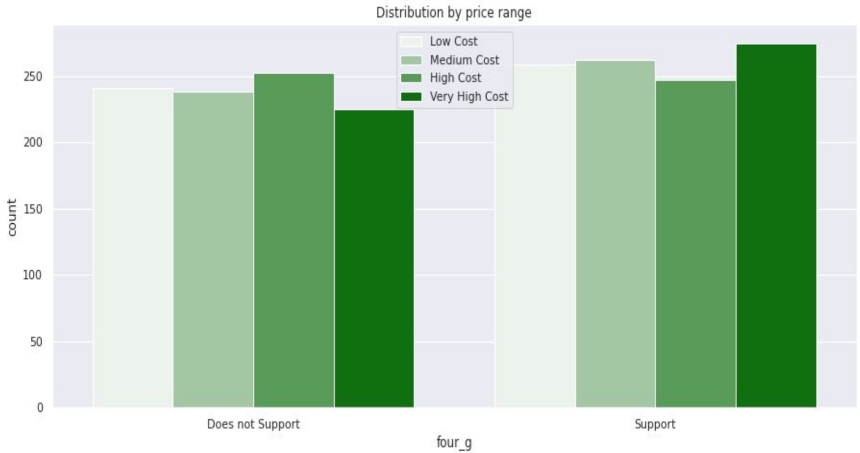
EDA(contd.)

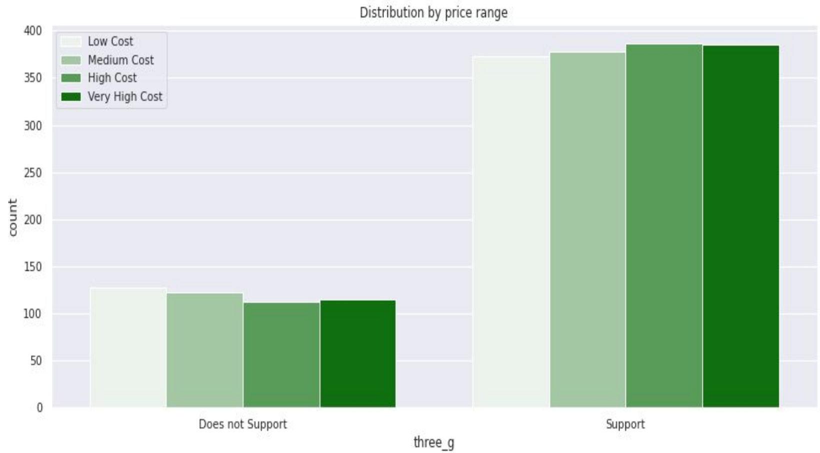




- Screen height shows little variation along with the target variables that is price range . This can be helpful to predict the target categories.

EDA(contd.)

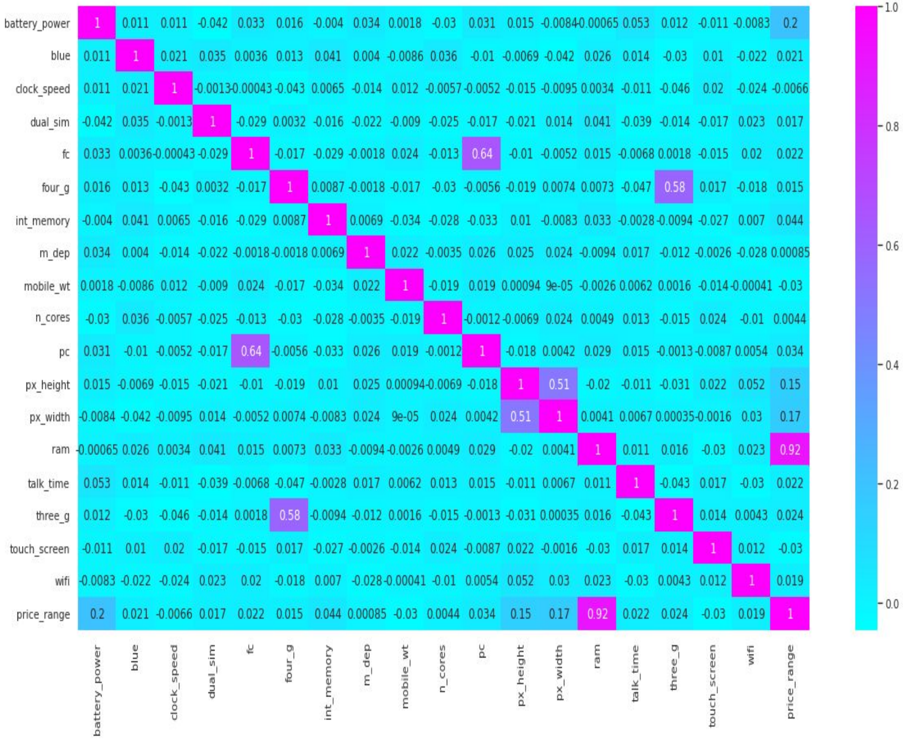




- We can see through graph that most of the mobile phones support 3G features and it would be play an important role in predictions. Most of the high cost phones are 3G.
- We can see through graph that around 50% mobile phones support 4G features . Also, very high cost phones are 4G.

Heat Map

- RAM and price range shows high correlation which is a good sign it signifies that RAM will be a major factor to estimate the price range.
- There is some collinearity in feature pairs “pc , fc” and “px_width , px_height” . Both correlations are justified since there are good chances that if fc of phone is good , the back camera would also be good.
- Also if px_height increase pixel_width also increase. We can replace these two features with one feature. FC megapixel are different entities despite of showing collinearity.



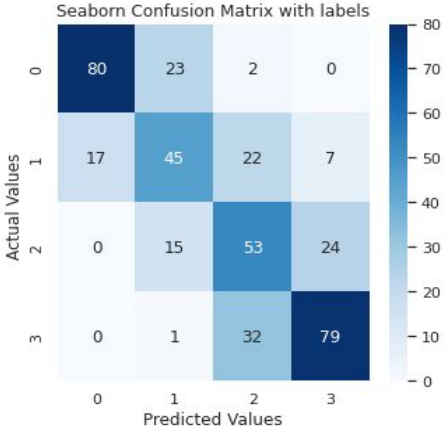
	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	talk_time	three_g	touch_screen	wifi	price_range
battery_power	1	0.011	0.011	-0.042	0.033	0.016	-0.004	0.034	0.0018	-0.03	0.031	0.015	-0.0084	-0.00065	0.053	0.012	-0.011	-0.0083	0.2
blue	0.011	1	0.021	0.035	0.0036	0.013	0.041	0.004	-0.0086	0.036	-0.01	-0.0069	-0.042	0.026	0.014	-0.03	0.01	-0.022	0.021
clock_speed	0.011	0.021	1	-0.0013	-0.00043	-0.043	0.0065	-0.014	0.012	-0.0057	-0.0052	-0.015	-0.0095	0.0034	-0.011	-0.046	0.02	-0.024	-0.0066
dual_sim	-0.042	0.035	-0.0013	1	-0.029	0.0032	-0.016	-0.022	-0.009	-0.025	-0.017	-0.021	0.014	0.041	-0.039	-0.014	-0.017	0.023	0.017
fc	0.033	0.0036	-0.00043	-0.029	1	-0.017	-0.029	-0.0018	0.024	-0.013	0.64	-0.01	-0.0052	0.015	-0.0068	0.0018	-0.015	0.02	0.022
four_g	0.016	0.013	-0.043	0.0032	-0.017	1	0.0087	-0.0018	-0.017	-0.03	-0.0056	-0.019	0.0074	0.0073	-0.047	0.58	0.017	-0.018	0.015
int_memory	-0.004	0.041	0.0065	-0.016	-0.029	0.0087	1	0.0069	-0.034	-0.028	-0.033	0.01	-0.0083	0.033	-0.0028	-0.0094	-0.027	0.007	0.044
m_dep	0.034	0.004	-0.014	-0.022	-0.0018	-0.0018	0.0069	1	0.022	-0.0035	0.026	0.025	0.024	-0.0094	0.017	-0.012	-0.0026	-0.028	0.00085
mobile_wt	0.0018	-0.0086	0.012	-0.009	0.024	-0.017	-0.034	0.022	1	-0.019	0.019	0.00094	9e-05	-0.0026	0.0062	0.0016	-0.014	-0.00041	-0.03
n_cores	-0.03	0.036	-0.0057	-0.025	-0.013	-0.03	-0.028	-0.0035	-0.019	1	-0.0012	-0.0069	0.024	0.0049	0.013	-0.015	0.024	-0.01	0.0044
pc	0.031	-0.01	-0.0052	-0.017	0.64	-0.0056	-0.033	0.026	0.019	-0.0012	1	-0.018	0.0042	0.029	0.015	-0.0013	-0.0087	0.0054	0.034
px_height	0.015	-0.0069	-0.015	-0.021	-0.01	-0.019	0.01	0.025	0.00094	-0.0069	-0.018	1	0.51	-0.02	-0.011	-0.031	0.022	0.052	0.15
px_width	-0.0084	-0.042	-0.0095	0.014	-0.0052	0.0074	-0.0083	0.024	9e-05	0.024	0.0042	0.51	1	0.0041	0.0067	0.00035	-0.0016	0.03	0.17
ram	-0.00065	0.026	0.0034	0.041	0.015	0.0073	0.033	-0.0094	-0.0026	0.0049	0.029	-0.02	0.0041	1	0.011	0.016	-0.03	0.023	0.92
talk_time	0.053	0.014	-0.011	-0.039	-0.0068	-0.047	-0.0028	0.017	0.0062	0.013	0.015	-0.011	0.0067	0.011	1	-0.043	0.017	-0.03	0.022
three_g	0.012	-0.03	-0.046	-0.014	0.0018	0.58	-0.0094	-0.012	0.0016	-0.015	-0.0013	-0.031	0.00035	0.016	-0.043	1	0.014	0.0043	0.024
touch_screen	-0.011	0.01	0.02	-0.017	-0.015	0.017	-0.027	-0.0026	-0.014	0.024	-0.0087	0.022	-0.0016	-0.03	0.017	0.014	1	0.012	-0.03
wifi	-0.0083	-0.022	-0.024	0.023	0.02	-0.018	0.007	-0.028	-0.00041	-0.01	0.0054	0.052	0.03	0.023	-0.03	0.0043	0.012	1	0.019
price_range	0.2	0.021	-0.0066	0.017	0.022	0.015	0.044	0.00085	-0.03	0.0044	0.034	0.15	0.17	0.92	0.022	0.024	-0.03	0.019	1

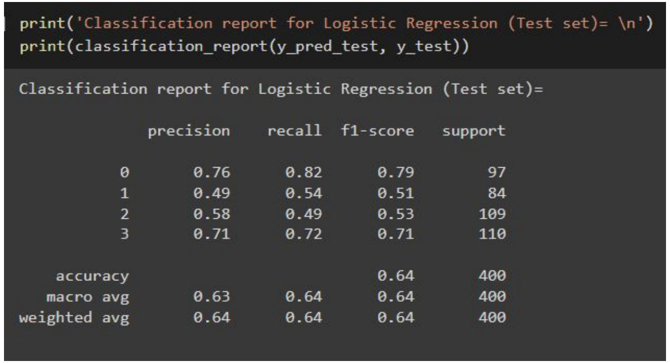
Models

- Logistic regression

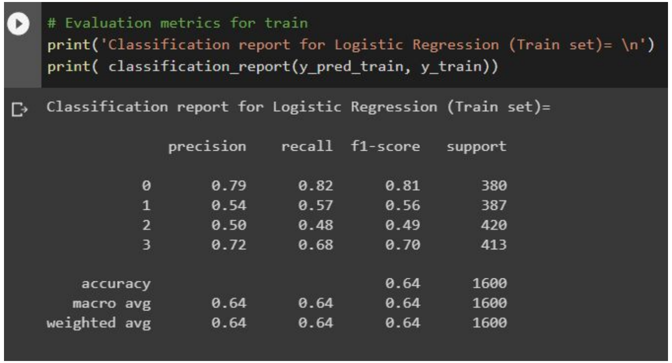
  Train accuracy : 64%

  Test accuracy : 64%





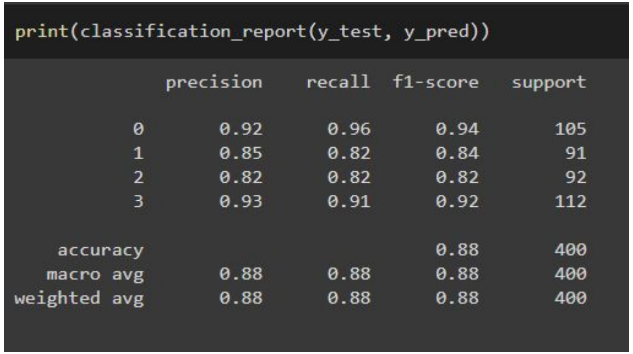
```
print('Classification report for Logistic Regression (Test set)= \n')
print(classification_report(y_pred_test, y_test))

Classification report for Logistic Regression (Test set)=

              precision    recall  f1-score   support

           0       0.76      0.82      0.79        97
           1       0.49      0.54      0.51        84
           2       0.58      0.49      0.53       109
           3       0.71      0.72      0.71       110

    accuracy                           0.64       400
   macro avg       0.63      0.64      0.64       400
weighted avg       0.64      0.64      0.64       400
```



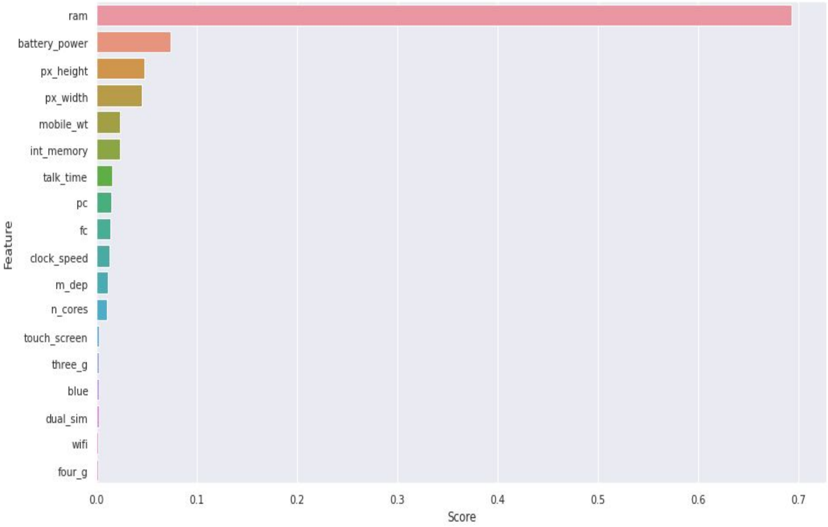
```
# Evaluation metrics for train
print('Classification report for Logistic Regression (Train set)= \n')
print( classification_report(y_pred_train, y_train))

Classification report for Logistic Regression (Train set)=

              precision    recall  f1-score   support

           0       0.79      0.82      0.81       380
           1       0.54      0.57      0.56       387
           2       0.50      0.48      0.49       420
           3       0.72      0.68      0.70       413

    accuracy                           0.64      1600
   macro avg       0.64      0.64      0.64      1600
weighted avg       0.64      0.64      0.64      1600
```

Models(contd.)

- Random Forest classifier with hyper parameter tuning
- Test accuracy : 88%
- We can see that the top 4 important features of our dataset are: RAM , battery_power , px_height and px_width.



```
print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.92      0.96      0.94       105
           1       0.85      0.82      0.84        91
           2       0.82      0.82      0.82        92
           3       0.93      0.91      0.92       112

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



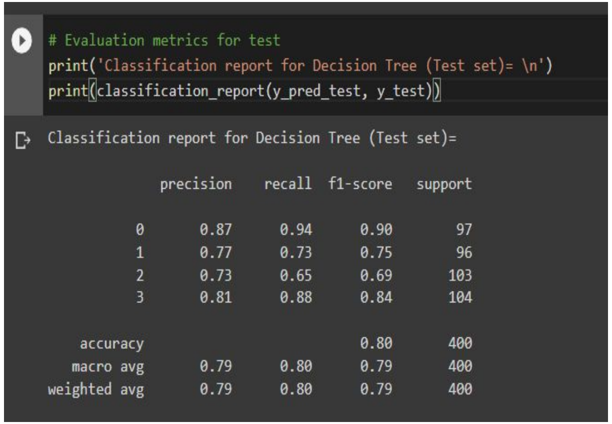
Models(contd.)

- Decision Tree

    Test accuracy :84%

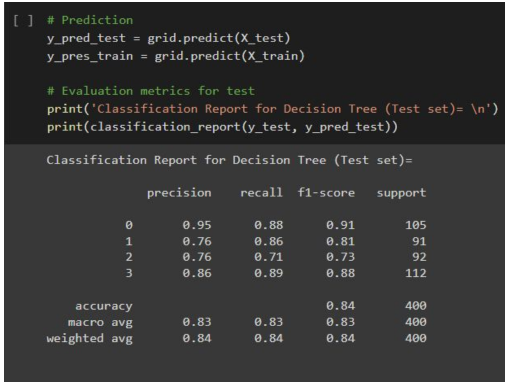
- Decision Tree with hyperparameter tuning

    Test accuracy : 80%



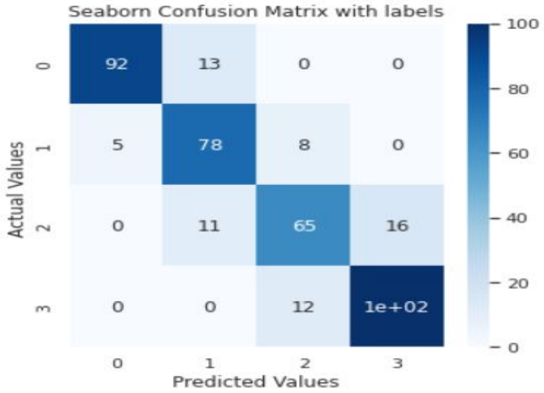
```
# Evaluation metrics for test
print('Classification report for Decision Tree (Test set)= \n')
print(classification_report(y_pred_test, y_test))
```

```
Classification report for Decision Tree (Test set)=

              precision    recall  f1-score   support

           0       0.87      0.94      0.90        97
           1       0.77      0.73      0.75        96
           2       0.73      0.65      0.69       103
           3       0.81      0.88      0.84       104

    accuracy                           0.80       400
   macro avg       0.79      0.80      0.79       400
weighted avg       0.79      0.80      0.79       400
```



```
[ ]  # Prediction
     y_pred_test = grid.predict(X_test)
     y_pres_train = grid.predict(X_train)

     # Evaluation metrics for test
     print('Classification Report for Decision Tree (Test set)= \n')
     print(classification_report(y_test, y_pred_test))
```

```
Classification Report for Decision Tree (Test set)=

              precision    recall  f1-score   support

           0       0.95      0.88      0.91       105
           1       0.76      0.86      0.81        91
           2       0.76      0.71      0.73        92
           3       0.86      0.89      0.88       112

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



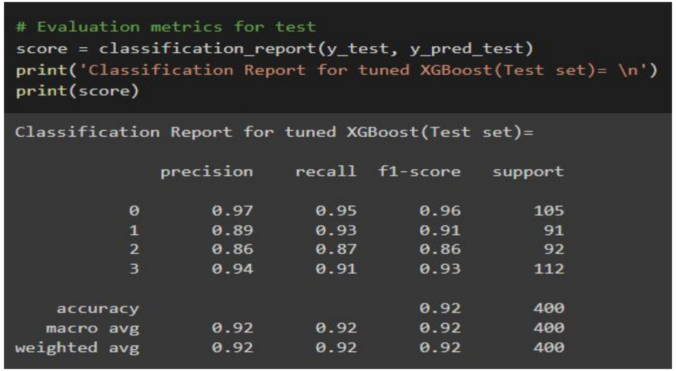
Models(contd.)

- XgBoost

  Test accuracy :91%

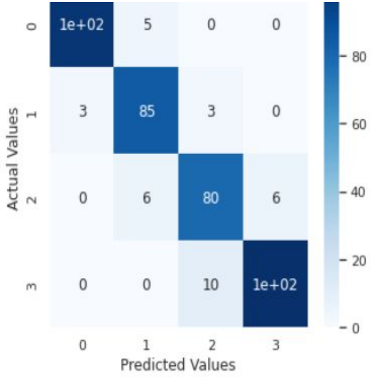
- XgBoost with hyperparameter tuning

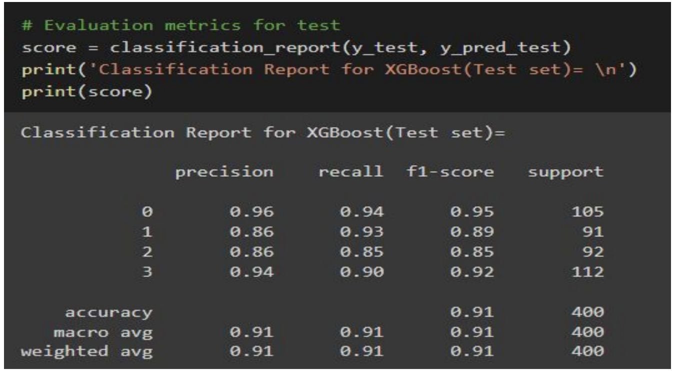
  Test accuracy : 92%



```
# Evaluation metrics for test
score = classification_report(y_test, y_pred_test)
print('Classification Report for tuned XGBoost(Test set)= \n')
print(score)
```

```
Classification Report for tuned XGBoost(Test set)=

              precision    recall  f1-score   support

           0       0.97      0.95      0.96       105
           1       0.89      0.93      0.91        91
           2       0.86      0.87      0.86        92
           3       0.94      0.91      0.93       112

    accuracy                           0.92       400
   macro avg       0.92      0.92      0.92       400
weighted avg       0.92      0.92      0.92       400
```





```
# Evaluation metrics for test
score = classification_report(y_test, y_pred_test)
print('Classification Report for XGBoost(Test set)= \n')
print(score)
```

```
Classification Report for XGBoost(Test set)=

              precision    recall  f1-score   support

           0       0.96      0.94      0.95       105
           1       0.86      0.93      0.89        91
           2       0.86      0.85      0.85        92
           3       0.94      0.90      0.92       112

    accuracy                           0.91       400
   macro avg       0.91      0.91      0.91       400
weighted avg       0.91      0.91      0.91       400
```

Challenges faced

- Comprehending the problem statement, and understanding the business implication
- Feature engineering deciding on which features to be dropped which to be kept and transformed
- Choosing the best visualization to show the trends among different features clearly in the EDA phase
- Deciding how to handle outliers
- Choosing the ML models to make predictions
- Deciding the evaluation metric to evaluate the models
- Choosing the best hyperparameters, which prevents overfitting

Conclusion

- From EDA we conclude that there are mobile phones in four price ranges.
- The number of elements is almost similar.
- Half the mobile phones have Bluetooth and half don’t.
- There is gradual increase in battery power 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 , px_height , px_width played more significant role to decide the price range.
- From all the above experiments we conclude that Random forest classifier with hyperparameter tuning and XgBoosting with hyperparameter tuning we got the best accuracy score.
- The accuracy and performance of the model is evaluated by using confusion matrix.

Thank You