Capstone Project - 3

Supervised ML - Classification

Topic - Mobile Price Range Prediction

By – Rahul Shah

CONTENTS OF THE PRESENTATION

- Problem Statement
- Data Summary
- Exploratory Data Analysis
- Data Wrangling
- Machine Learning models
- Model Explanation
- Challenges
- Conclusion

Problem Statement

- In modern times mobile phones have become a part and parcel of everyone’s life. People want smartphones with more features and best specifications in a phone and that too at affordable prices. The demand for smartphones is so high that there is a huge competition prevailing between mobile manufacturers. To stay ahead in the race, these companies should keep up with the competition and should rate their mobile devices well try to bring in new features and innovations so that people are lured towards buying their brand smartphones.

- Price of a mobile phone is influenced by various factors such as ram, battery size, primary camera megapixel, network connectivity, etc. are some of the important factors in determining the price. From a business perspective, it is necessary to analyze these factors from time to time and come up with best set of specifications and price ranges so that people buy their mobile phones.

- The problem statement is to build a machine learning model that could predict the price range values, given the other variables.

Data Summary

- The dataset contains information about the mobile phone features which is used to estimate the price range.
- The dataset contains 2000 non-null observations and 21 columns.

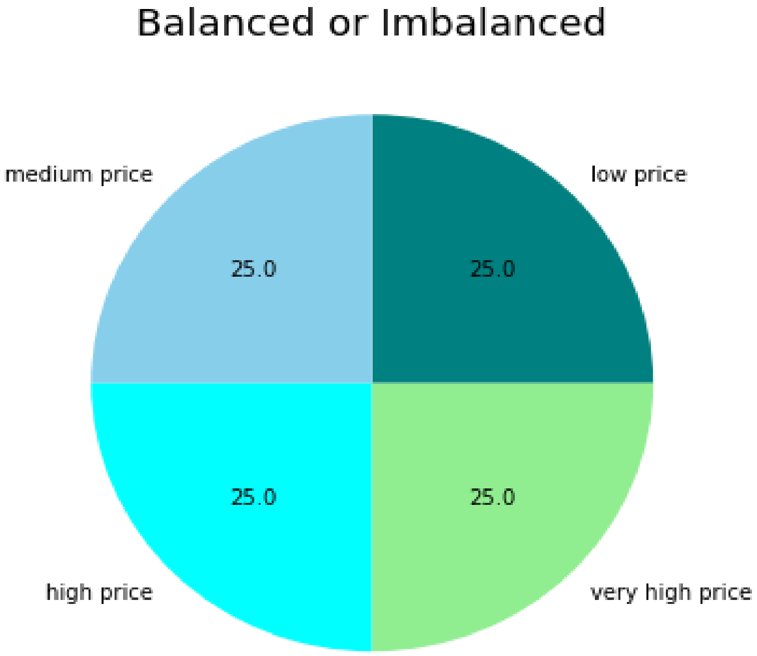
Data Attributes:

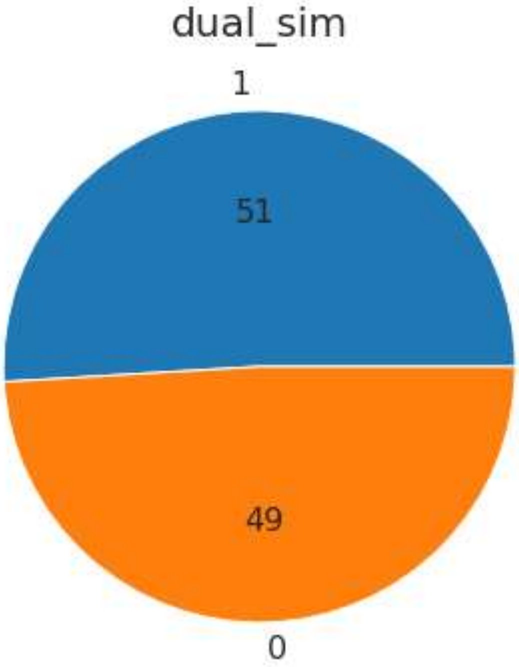
- **Battery_powe**r - Total energy a battery can store in one time measured in mAh
- **Blue** – Has bluetooth or not
- **Clock_speed** - speed at which microprocessor executes instructions
- **dual_sim** - Has dual sim support or not
- **Fc** – Front Camera mega pixels
- **Four_g** - Has 4G or not
- **Int_memory** - Internal Memory in Gigabytes
- **M_dep** - Mobile Depth in cm

Data Summary (continued)

- **Mobile_wt** - Weight of mobile phone
- **N_cores** - Number of cores of processor
- **Pc** - Primary Camera megapixels
- **Px_height, Px_width** - Pixel Resolution Height and 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 when you are on call
- **Three_g** - Has 3G or not
- **Touch_screen** - Has touch screen or not
- **Wifi** - has a 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).

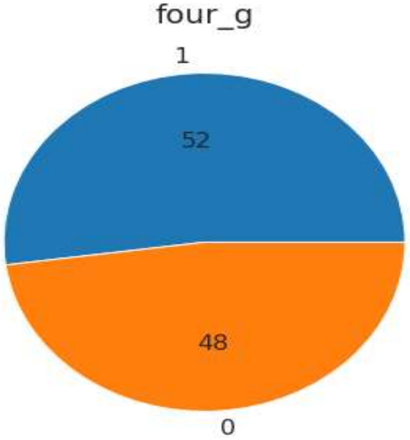
EDA - Univariate analysis

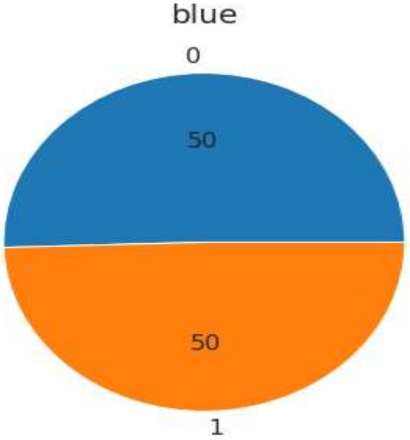


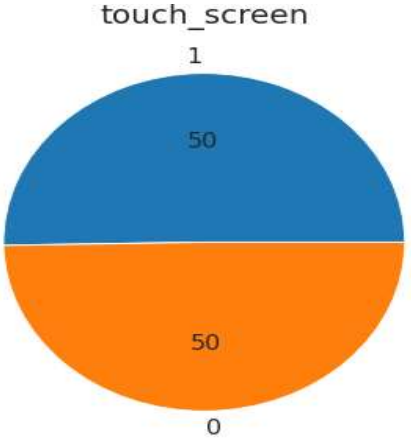


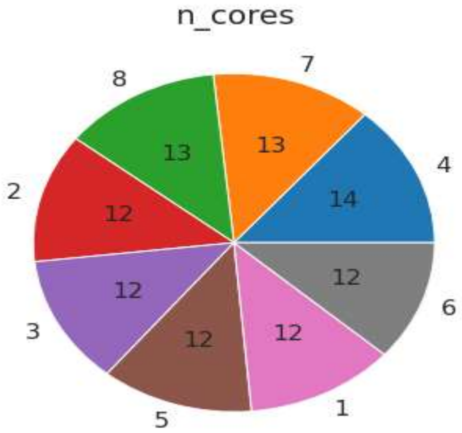
- Our dependent variable - Price range has equal no of observations in each bucket.
- Other dichotomous types have equal no of observations for each category, except for 3g

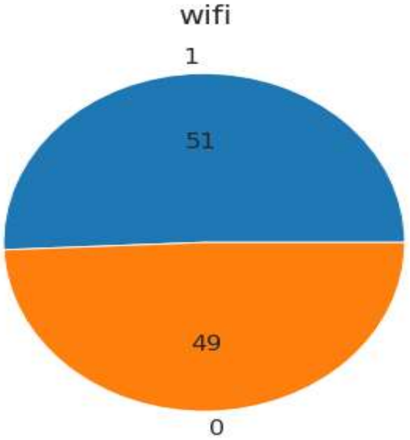
Continued..

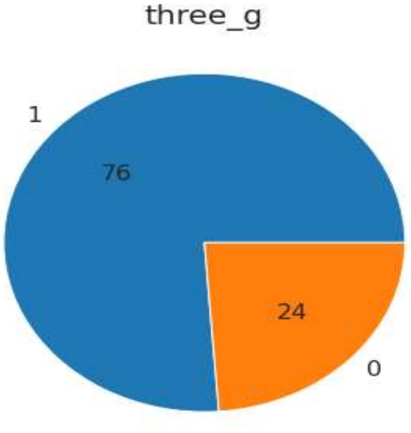

four_g


blue

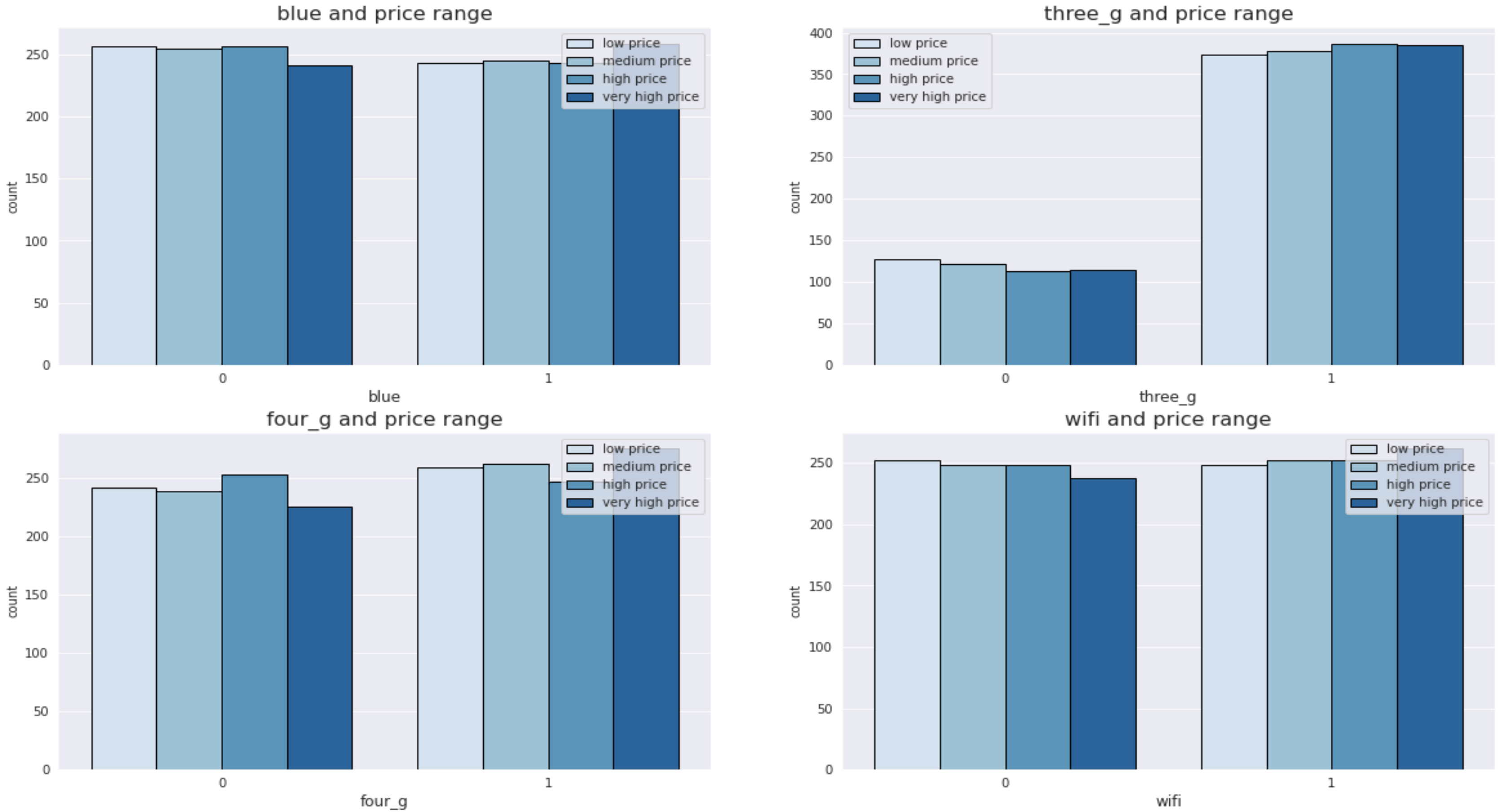

touch_screen


n_cores

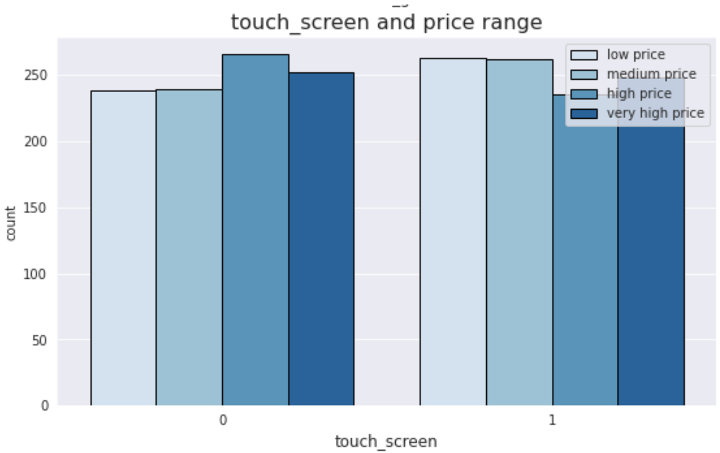

wifi

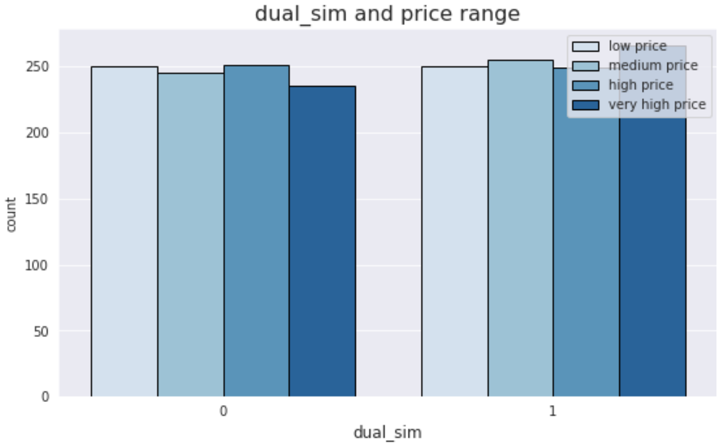

three_g

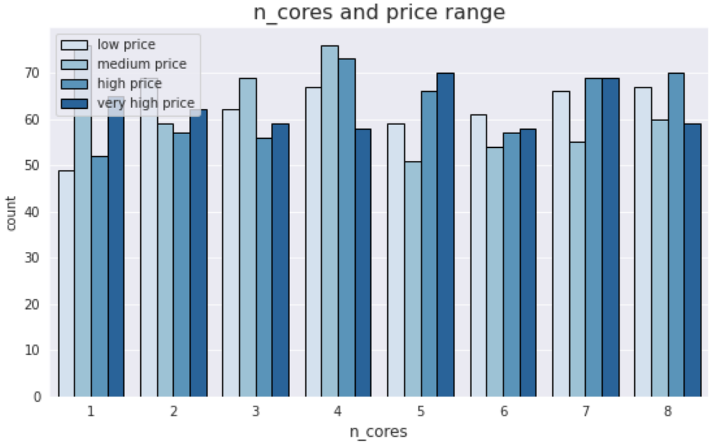
Multivariate Analysis



continued …

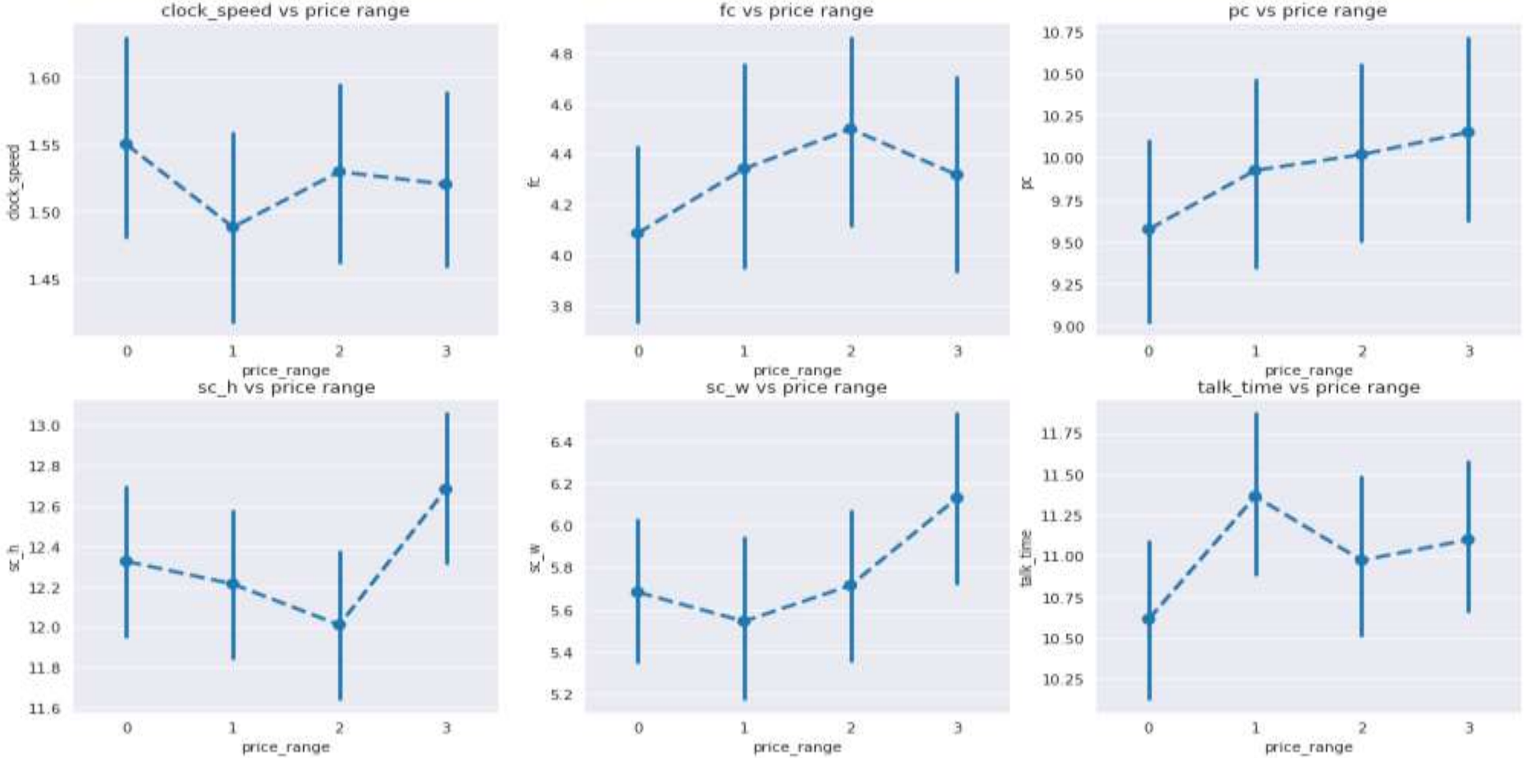

touch_screen and price range


dual_sim and price range


n_cores and price range

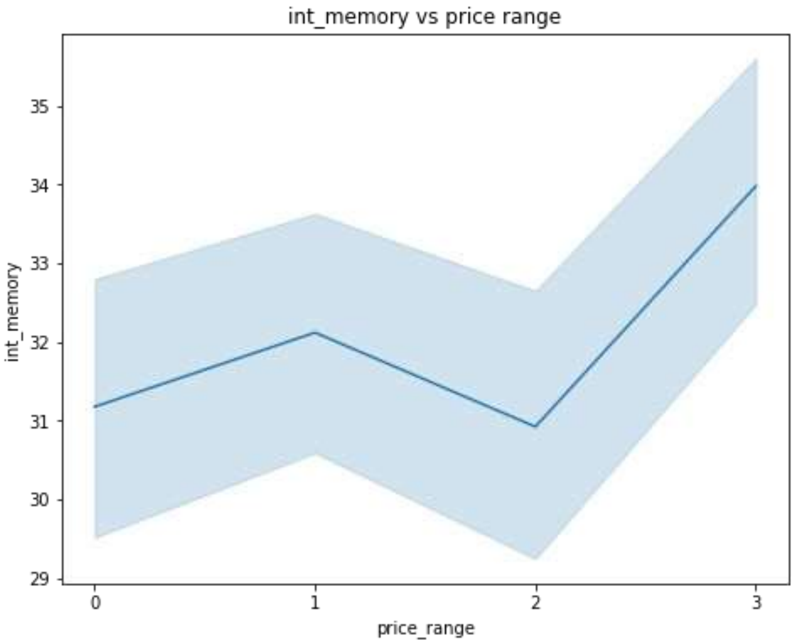
Multivariate analysis - Numerical variables

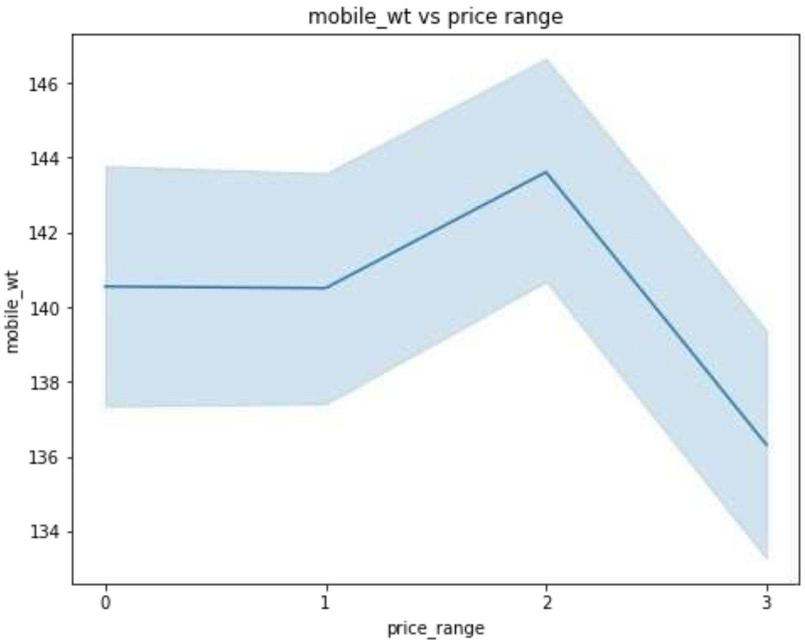
- Clock speed is high for low price range phones, Talk time is less.
- Pc, fc , sc_w are in increasing trend.



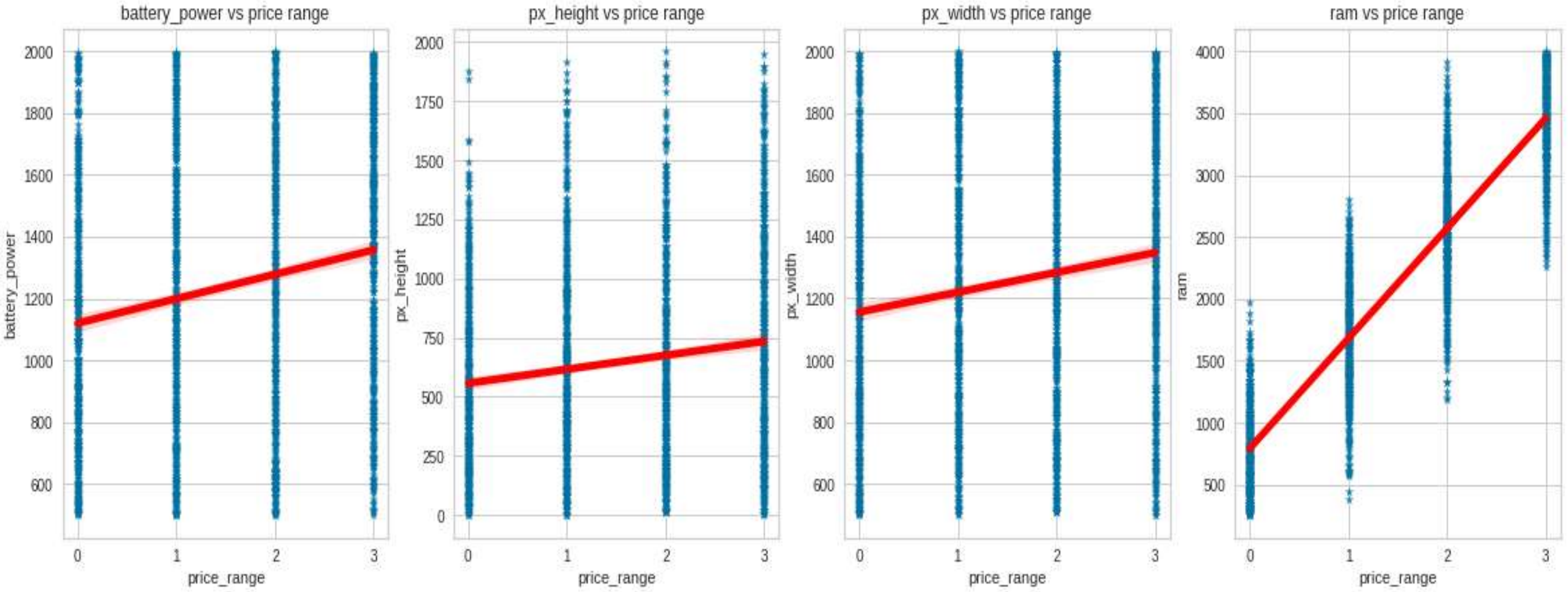
Multivariate analysis - int_memory, mobile_wt

1. We can observe drastic increase in internal memory for very high prices.
2. Also there is drastic decrease in mobile weight for very high prices.





Multivariate analysis - Continued

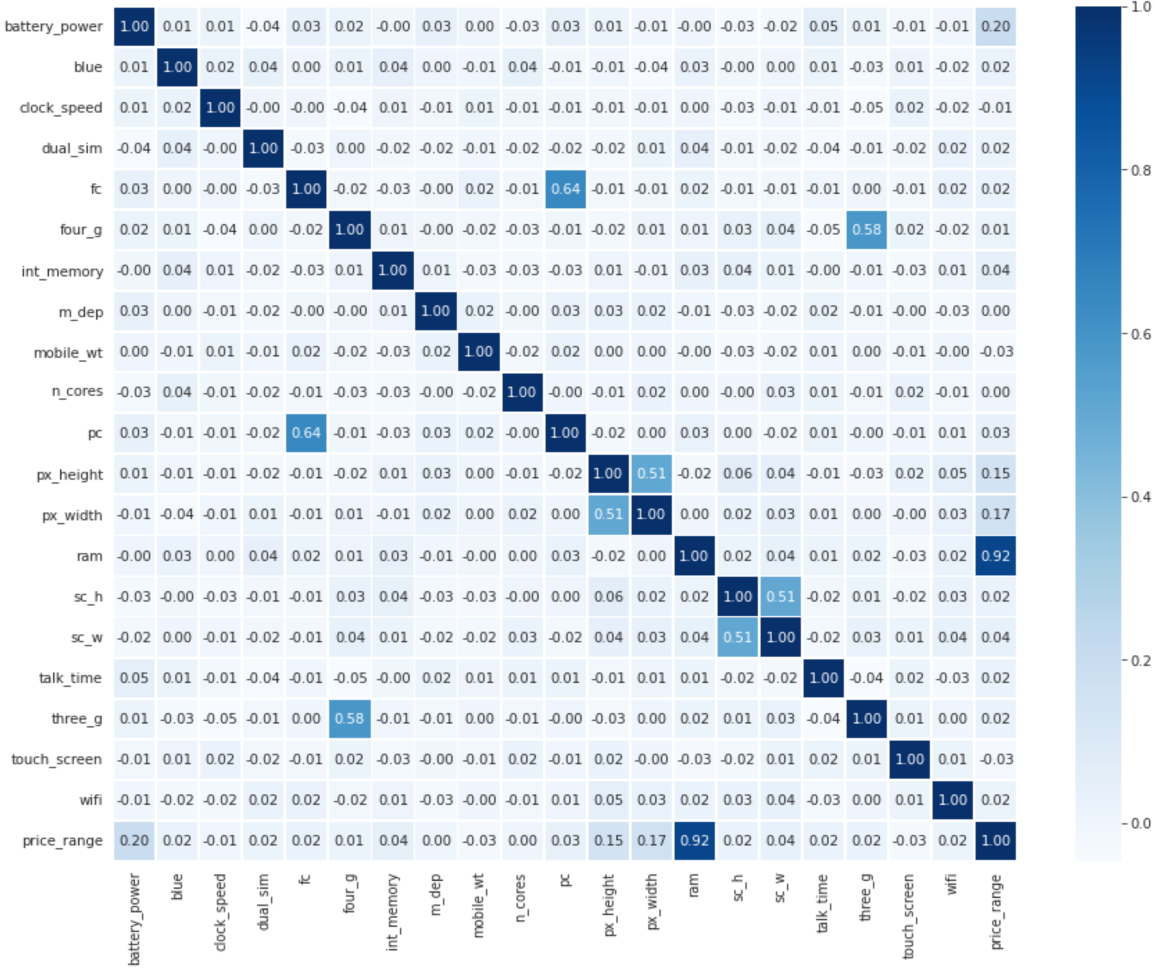


Correlation

- Ram and price_range are highly correlated, ram has more impact on the dependant variable compared to other variables.

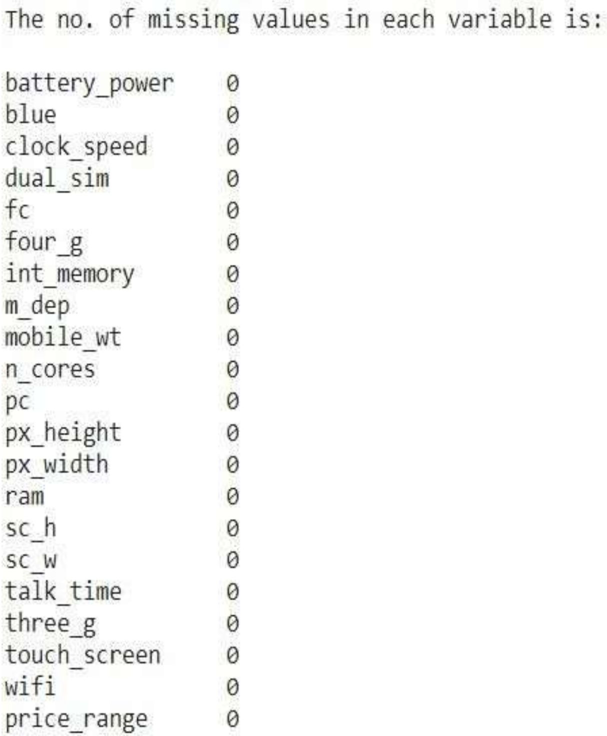
- pc and fc are moderately correlated. Similarly 3g and 4g are also moderately correlated with each other.

- px_height & px_width are moderately correlated. Similarly sc_h and sc_w are moderately correlated with each other.

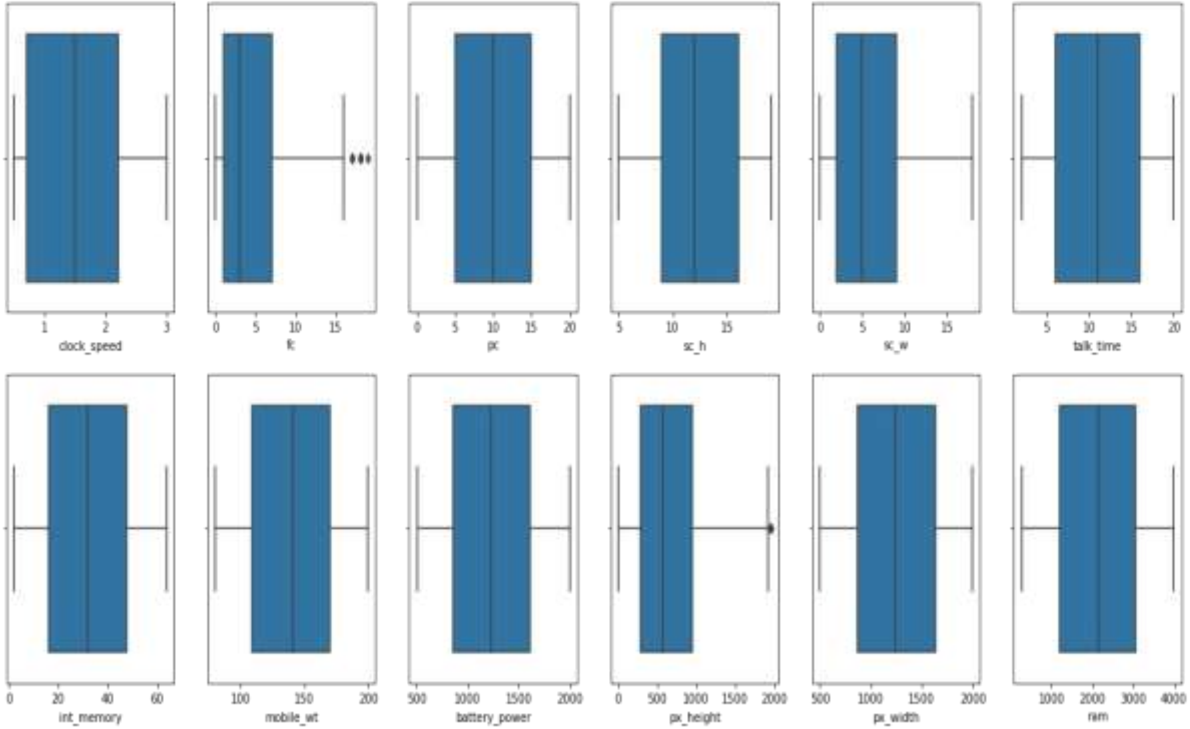


	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
battery_power	1.00	0.01	0.01	-0.04	0.03	0.02	-0.00	0.03	0.00	-0.03	0.03	0.01	-0.01	-0.00	-0.03	-0.02	0.05	0.01	-0.01	-0.01	0.20
blue	0.01	1.00	0.02	0.04	0.00	0.01	0.04	0.00	-0.01	0.04	-0.01	-0.01	-0.04	0.03	-0.00	0.00	0.01	-0.03	0.01	-0.02	0.02
clock_speed	0.01	0.02	1.00	-0.00	-0.00	-0.04	0.01	-0.01	0.01	-0.01	-0.01	-0.01	-0.01	0.00	-0.03	-0.01	-0.01	-0.05	0.02	-0.02	-0.01
dual_sim	-0.04	0.04	-0.00	1.00	-0.03	0.00	-0.02	-0.02	-0.01	-0.02	-0.02	-0.02	0.01	0.04	-0.01	-0.02	-0.04	-0.01	-0.02	0.02	0.02
fc	0.03	0.00	-0.00	-0.03	1.00	-0.02	-0.03	-0.00	0.02	-0.01	0.64	-0.01	-0.01	0.02	-0.01	-0.01	-0.01	0.00	-0.01	0.02	0.02
four_g	0.02	0.01	-0.04	0.00	-0.02	1.00	0.01	-0.00	-0.02	-0.03	-0.01	-0.02	0.01	0.01	0.03	0.04	-0.05	0.58	0.02	-0.02	0.01
int_memory	-0.00	0.04	0.01	-0.02	-0.03	0.01	1.00	0.01	-0.03	-0.03	-0.03	0.01	-0.01	0.03	0.04	0.01	-0.00	-0.01	-0.03	0.01	0.04
m_dep	0.03	0.00	-0.01	-0.02	-0.00	-0.00	0.01	1.00	0.02	-0.00	0.03	0.03	0.02	-0.01	-0.03	-0.02	0.02	-0.01	-0.00	-0.03	0.00
mobile_wt	0.00	-0.01	0.01	-0.01	0.02	-0.02	-0.03	0.02	1.00	-0.02	0.02	0.00	0.00	-0.00	-0.03	-0.02	0.01	0.00	-0.01	-0.00	-0.03
n_cores	-0.03	0.04	-0.01	-0.02	-0.01	-0.03	-0.03	-0.00	-0.02	1.00	-0.00	-0.01	0.02	0.00	-0.00	0.03	0.01	-0.01	0.02	-0.01	0.00
pc	0.03	-0.01	-0.01	-0.02	0.64	-0.01	-0.03	0.03	0.02	-0.00	1.00	-0.02	0.00	0.03	0.00	-0.02	0.01	-0.00	-0.01	0.01	0.03
px_height	0.01	-0.01	-0.01	-0.02	-0.01	-0.02	0.01	0.03	0.00	-0.01	-0.02	1.00	0.51	-0.02	0.06	0.04	-0.01	-0.03	0.02	0.05	0.15
px_width	-0.01	-0.04	-0.01	0.01	-0.01	0.01	-0.01	0.02	0.00	0.02	0.00	0.51	1.00	0.00	0.02	0.03	0.01	0.00	-0.00	0.03	0.17
ram	-0.00	0.03	0.00	0.04	0.02	0.01	0.03	-0.01	-0.00	0.00	0.03	-0.02	0.00	1.00	0.02	0.04	0.01	0.02	-0.03	0.02	0.92
sc_h	-0.03	-0.00	-0.03	-0.01	-0.01	0.03	0.04	-0.03	-0.03	-0.00	0.00	0.06	0.02	0.02	1.00	0.51	-0.02	0.01	-0.02	0.03	0.02
sc_w	-0.02	0.00	-0.01	-0.02	-0.01	0.04	0.01	-0.02	-0.02	0.03	-0.02	0.04	0.03	0.04	0.51	1.00	-0.02	0.03	0.01	0.04	0.04
talk_time	0.05	0.01	-0.01	-0.04	-0.01	-0.05	-0.00	0.02	0.01	0.01	0.01	-0.01	0.01	0.01	-0.02	-0.02	1.00	-0.04	0.02	-0.03	0.02
three_g	0.01	-0.03	-0.05	-0.01	0.00	0.58	-0.01	-0.01	0.00	-0.01	-0.00	-0.03	0.00	0.02	0.01	0.03	-0.04	1.00	0.01	0.00	0.02
touch_screen	-0.01	0.01	0.02	-0.02	-0.01	0.02	-0.03	-0.00	-0.01	0.02	-0.01	0.02	-0.00	-0.03	-0.02	0.01	0.02	0.01	1.00	0.01	-0.03
wifi	-0.01	-0.02	-0.02	0.02	0.02	-0.02	0.01	-0.03	-0.00	-0.01	0.01	0.05	0.03	0.02	0.03	0.04	-0.03	0.00	0.01	1.00	0.02
price_range	0.20	0.02	-0.01	0.02	0.02	0.01	0.04	0.00	-0.03	0.00	0.03	0.15	0.17	0.92	0.02	0.04	0.02	0.02	-0.03	0.02	1.00

Data Wrangling



```
The no. of missing values in each variable is:

battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
```

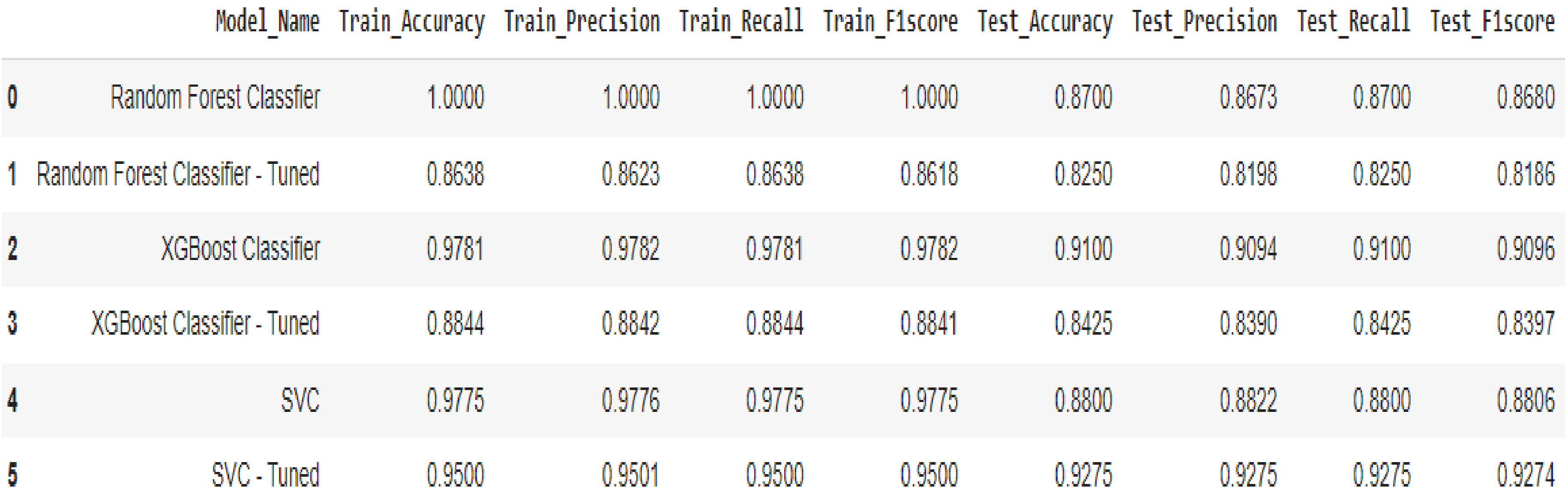


Data Wrangling

- There are no null values in our dataset.
- There is no duplicate observation present in our dataset.
- The boxplot clearly shows there are no outliers except in fc, which can be considered unimportant because they are not far away from the maximum value.
- Some values in the screen width and pixel height feature were 0 which is impossible in real life. So these 0 values were replaced with the mean values based on their price range.
- screen width and screen height converted into one variable as "screen_size“.
- three_g and four_g features merged into one variable as "network“.
- px_height and px_width merged into one variable as "pixels“.

Machine Learning Models

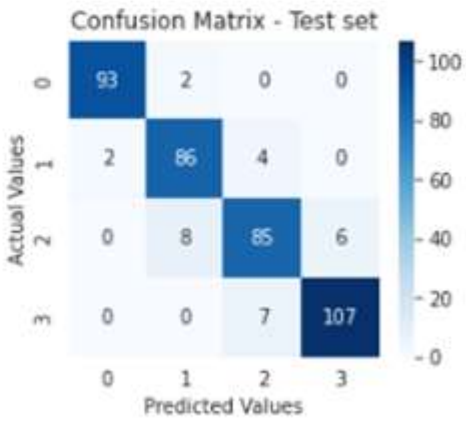
- 3 Models were implemented: Random Forest Classifier, XGBoost and SVC
- The evaluation results are:
- Training set shape – 1600, 17 and Test set shape – 400, 17



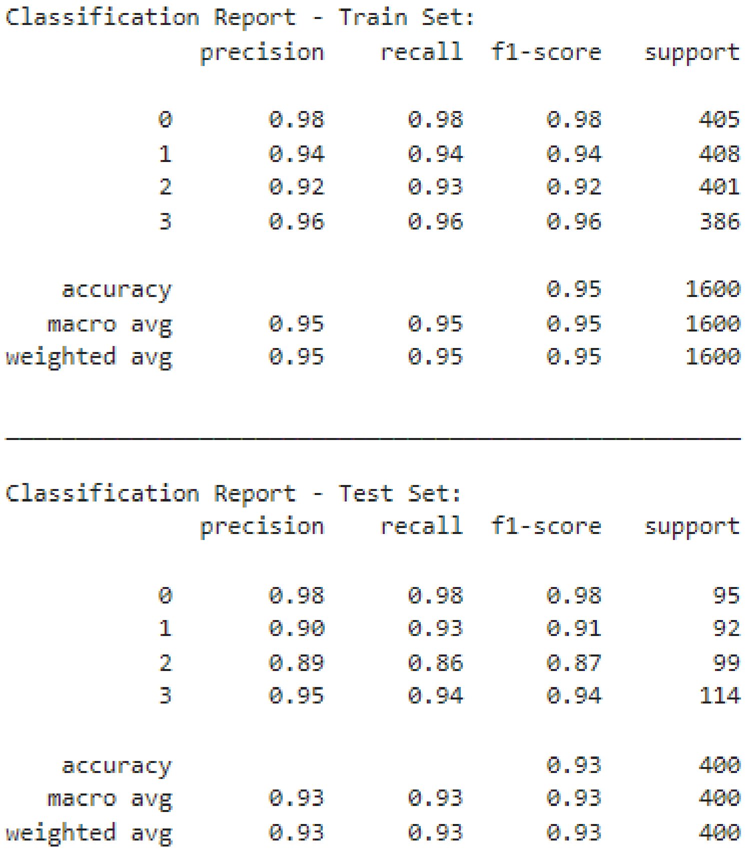
	Model_Name	Train_Accuracy	Train_Precision	Train_Recall	Train_F1score	Test_Accuracy	Test_Precision	Test_Recall	Test_F1score
0	Random Forest Classfier	1.0000	1.0000	1.0000	1.0000	0.8700	0.8673	0.8700	0.8680
1	Random Forest Classifier - Tuned	0.8638	0.8623	0.8638	0.8618	0.8250	0.8198	0.8250	0.8186
2	XGBoost Classifier	0.9781	0.9782	0.9781	0.9782	0.9100	0.9094	0.9100	0.9096
3	XGBoost Classifier - Tuned	0.8844	0.8842	0.8844	0.8841	0.8425	0.8390	0.8425	0.8397
4	SVC	0.9775	0.9776	0.9775	0.9775	0.8800	0.8822	0.8800	0.8806
5	SVC - Tuned	0.9500	0.9501	0.9500	0.9500	0.9275	0.9275	0.9275	0.9274

Model Selection and Validation

- Random Forest and XGBoost initially overfitted.
- Overfitting was tackled using hyperparameter tuning.
- The best performance was given by Support Vector Classifier model.
- **Support Vector Classifier**
  - train accuracy – 0.95
  - test accuracy – 0.93



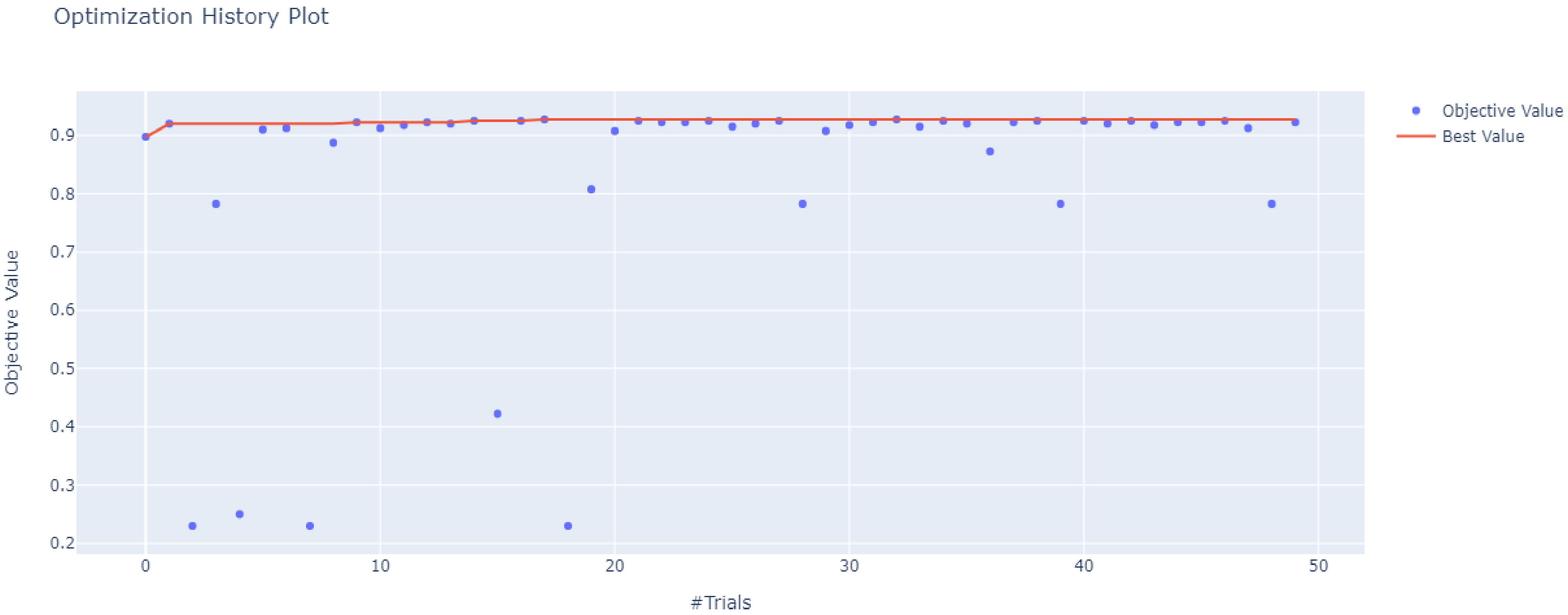
	0	1	2	3
0	93	2	0	0
1	2	86	4	0
2	0	8	85	6
3	0	0	7	107



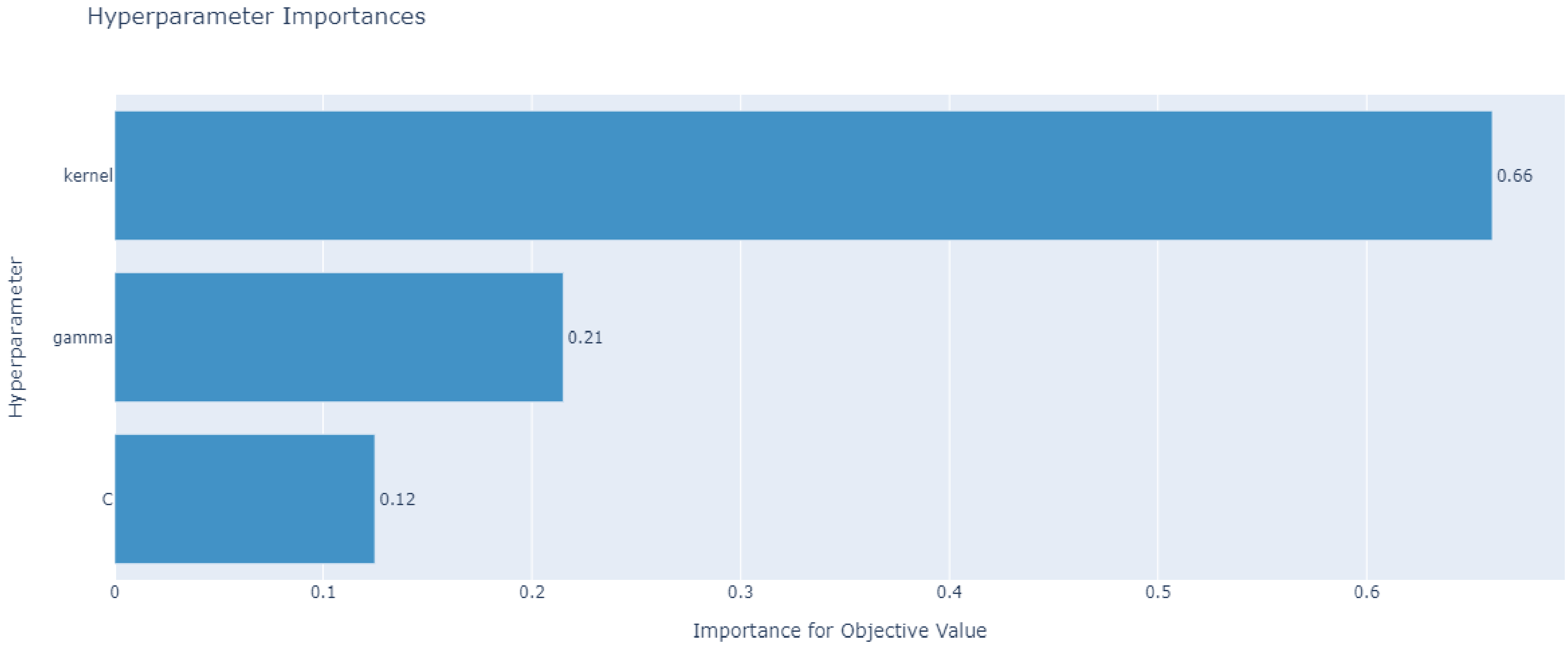
```
Classification Report - Train Set:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       405
           1       0.94      0.94      0.94       408
           2       0.92      0.93      0.92       401
           3       0.96      0.96      0.96       386

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


_____________________________________________________


Classification Report - Test Set:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        95
           1       0.90      0.93      0.91        92
           2       0.89      0.86      0.87        99
           3       0.95      0.94      0.94       114

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

SVC – Optuna Hyper-parameter Tuning

Optuna – Optimization History Plot



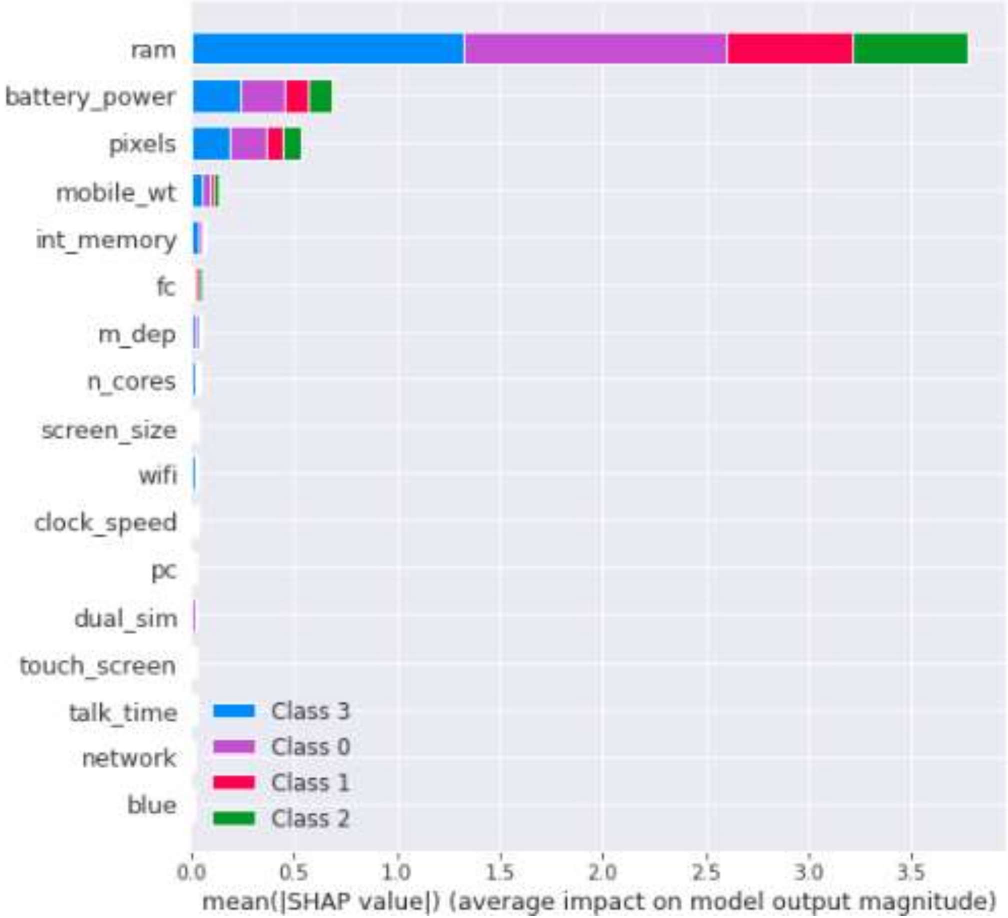
Optuna – Hyper-parameter Importances Plot



Model Explanation

- Shap techniques were implemented to understand the working of the best model.

- The most important features were ram, battery_power and pixels.



Challenges

- The most challenging part in this exercise was to find the optimal set of parameters that could give us the best performance.

- It took hours to try every combination and finally selecting the best values.

- The shap instance for SVM model using KernelShap was taking a lot of time to produce.

Conclusion

We performed various steps to determine our predictions for the mobile price range. We started with simple EDA where we analyzed our dependent variable as well as other independent variables. We found out the correlation, count, relationships with the dependent variable. We looked for missing values and outliers and did some feature modifications.

Finally we implemented 3 machine learning algorithms namely; RandomForest, XGBoost and SVM. We tried hyperparameter tuning to reduce overfitting and increase model performance. The best performance was given by SVM model.

We also implemented shap techniques to identify the important features impacting our model predictions. We saw ram, battery power and pixels were the major contributors. Higher the values of these led to higher predicted values.

SVM performed much better than all other models because it has more accurate results and also able to generalize the features much better than others. The accuracy of our best model was 0.95 and 0.93 for training and test set respectively.

THANK YOU