

Capstone Project

Mobile Price Range Prediction

By
Anurag
Taiskar

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.

Index

Discussion points

- ❑ Data description
- ❑ Data Cleaning
- ❑ Exploratory data analysis
- ❑ Correlation Analysis
- ❑ All models Evaluation Metrics
- ❑ Model Selection
- ❑ Conclusion



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 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
- **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 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



Data Overview

- There are 2000 observation
- There are 21 feature variable
- There is no null values
- Price Range is the target variable

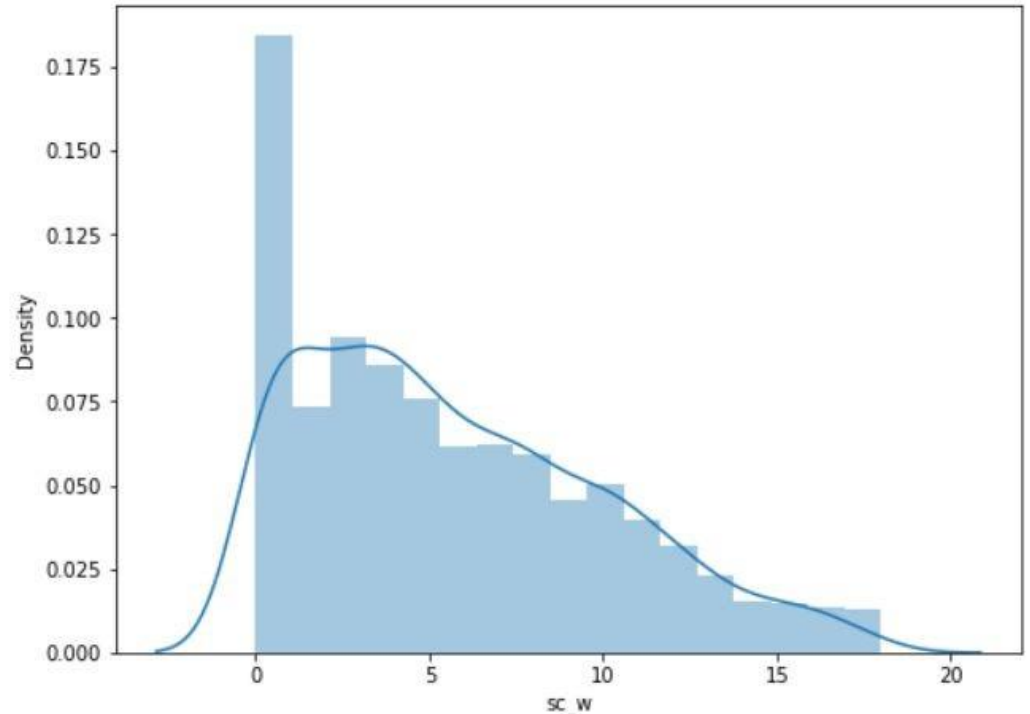
```
[ ] # overlook to data information using info function
```

```
df.info()
```

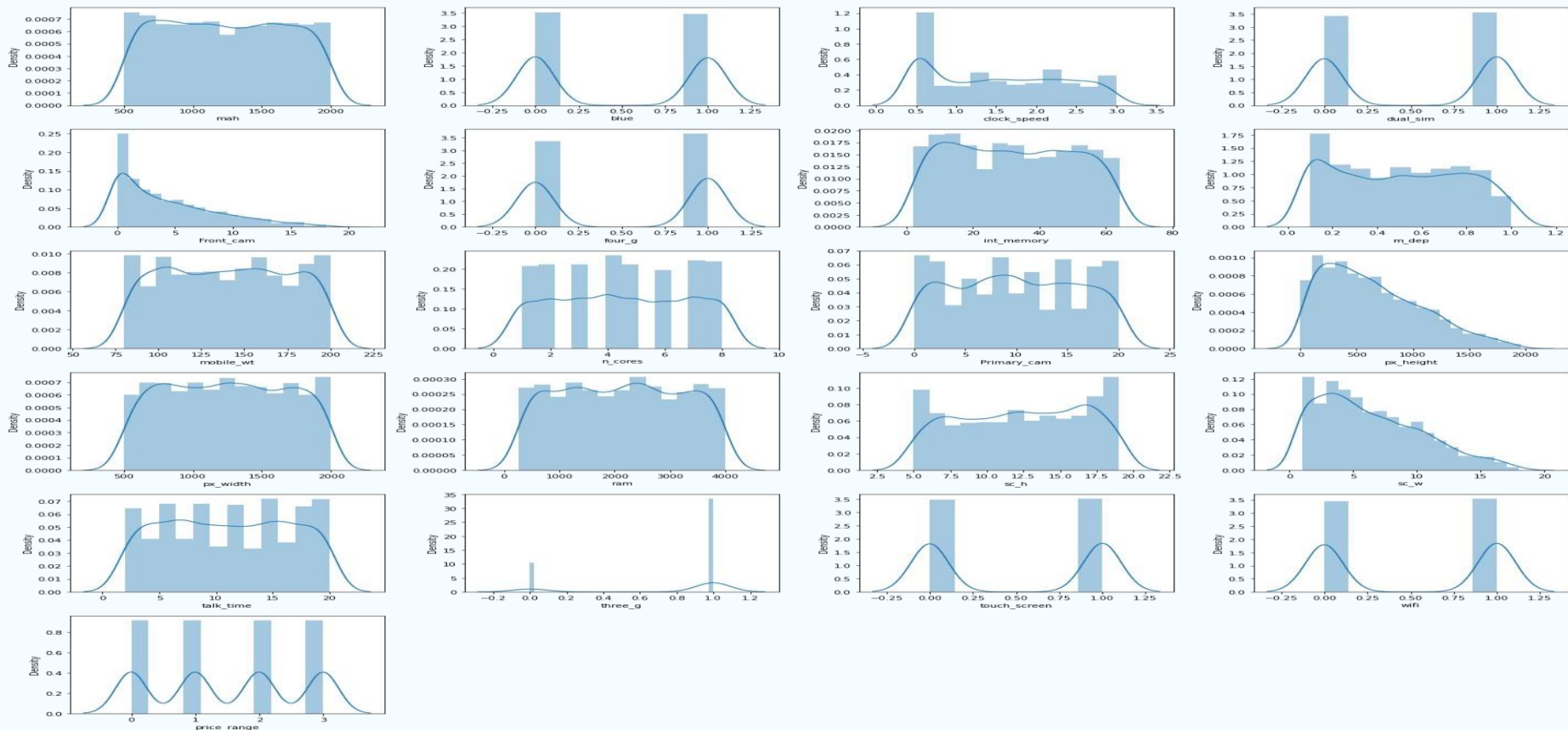
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   battery_power        2000 non-null   int64
1   blue                 2000 non-null   int64
2   clock_speed          2000 non-null   float64
3   dual_sim             2000 non-null   int64
4   fc                   2000 non-null   int64
5   four_g               2000 non-null   int64
6   int_memory           2000 non-null   int64
7   m_dep                2000 non-null   float64
8   mobile_wt            2000 non-null   int64
9   n_cores              2000 non-null   int64
10  pc                   2000 non-null   int64
11  px_height            2000 non-null   int64
12  px_width             2000 non-null   int64
13  ram                  2000 non-null   int64
14  sc_h                 2000 non-null   int64
15  sc_w                 2000 non-null   int64
16  talk_time            2000 non-null   int64
17  three_g              2000 non-null   int64
18  touch_screen         2000 non-null   int64
19  wifi                 2000 non-null   int64
20  price_range          2000 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

Handling inappropriate values

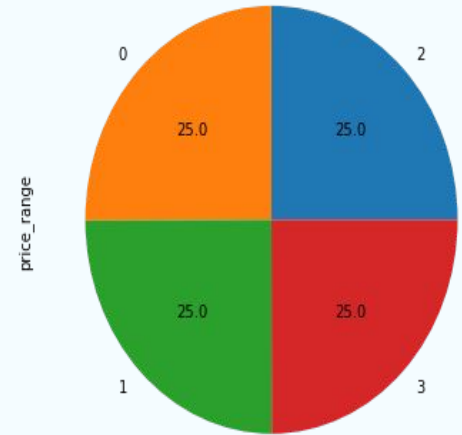
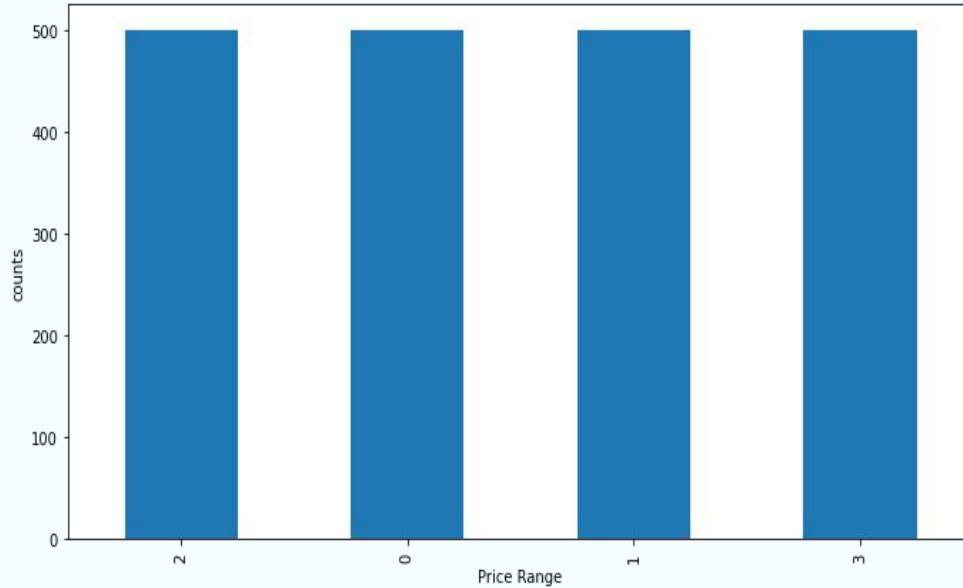
- In this screen width column, there are 179 observations whose width is zero, which is not possible
- 179 is a big part of the dataset
Filling is the only option so
- KNN Imputer with one neighbour is the best option to fill these values



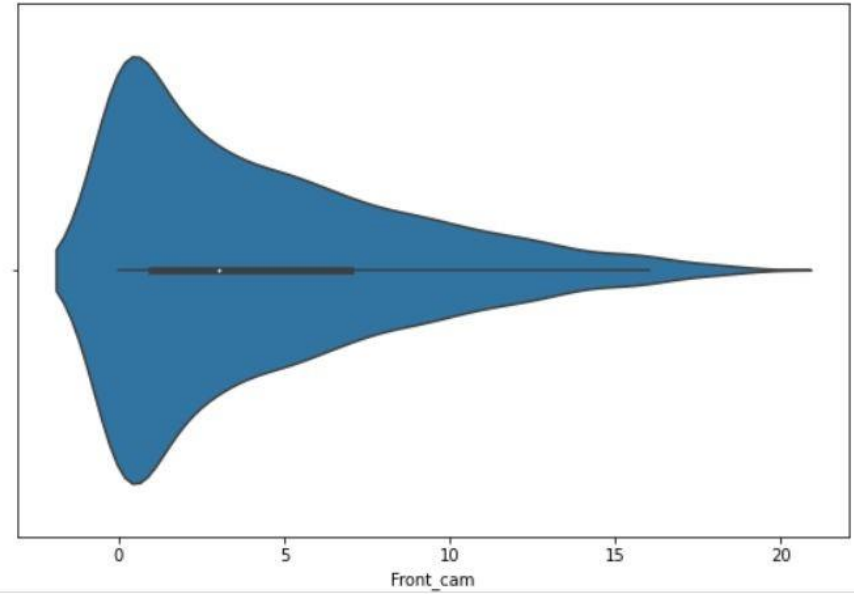
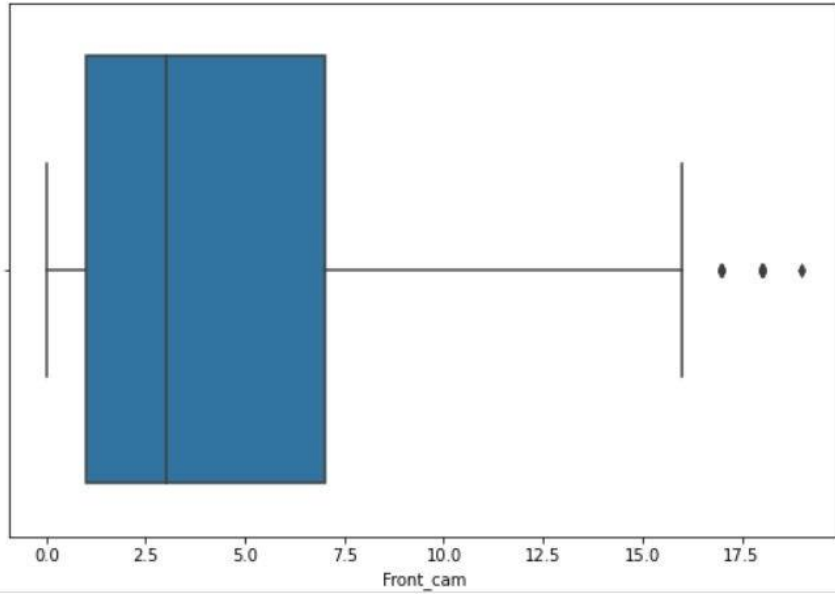
Distribution of Features



Checking Class Imbalance



Outliers



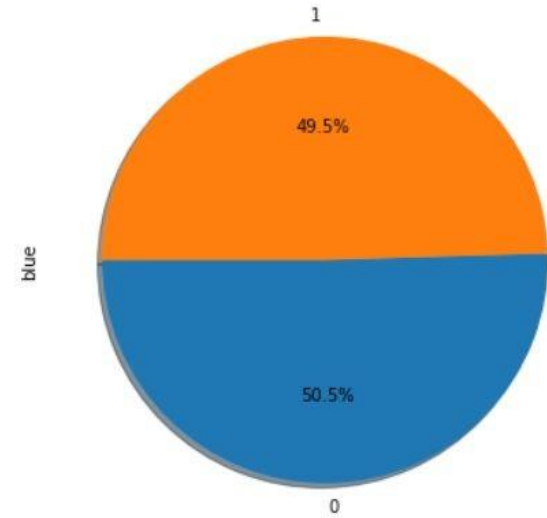
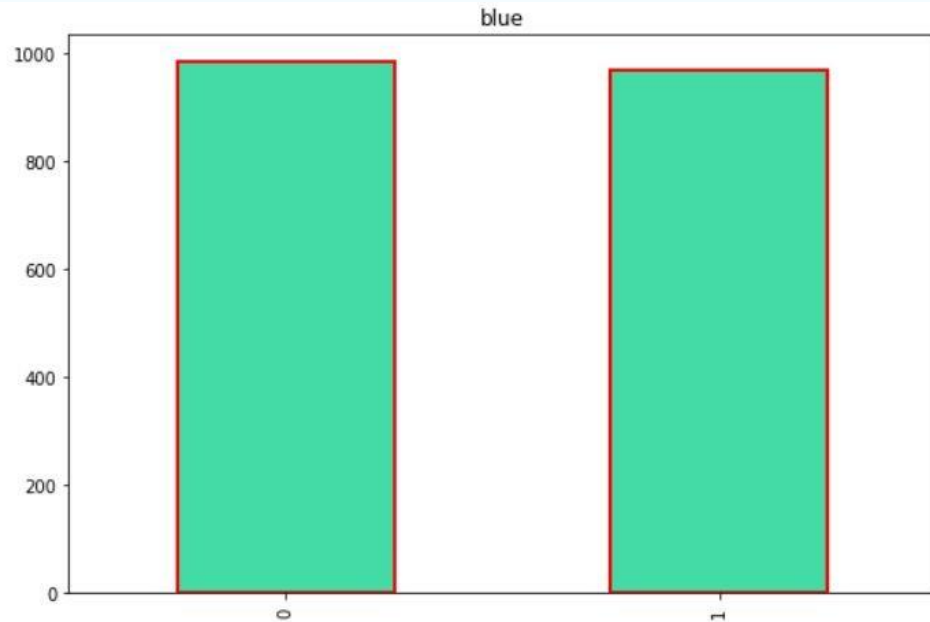
- There are 20 outliers in front camera column

EDA

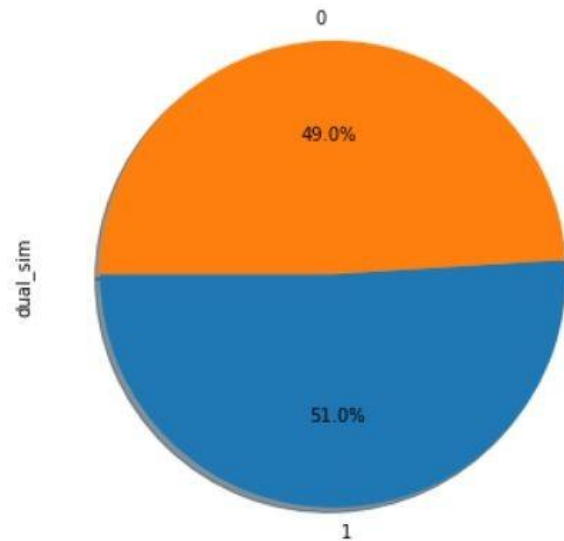
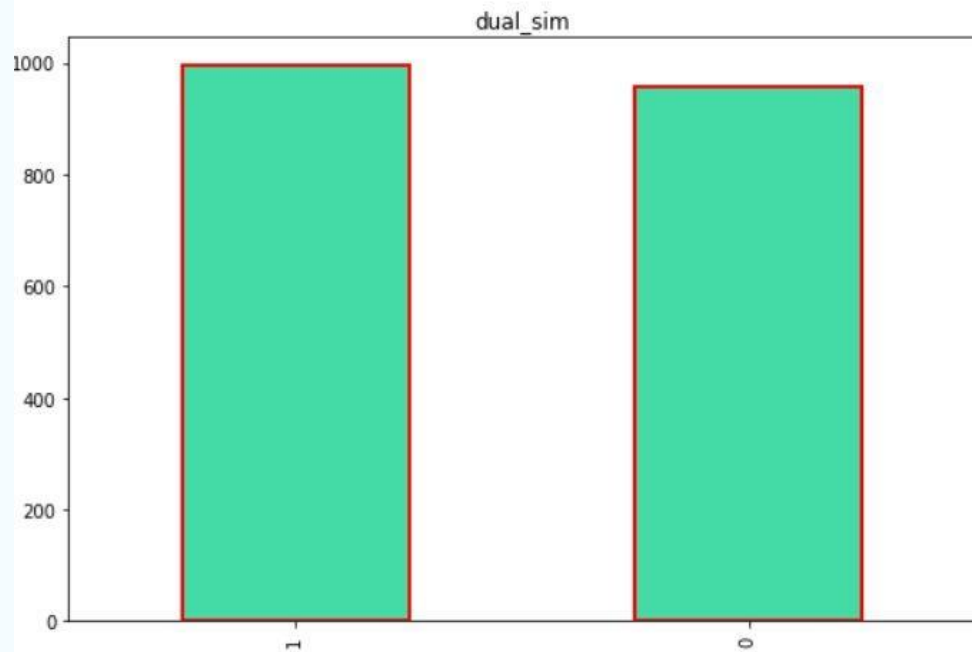
EXPLORATORY DATA ANALYSIS



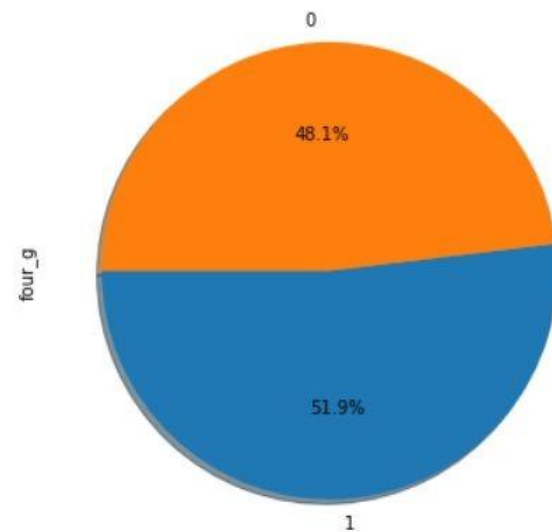
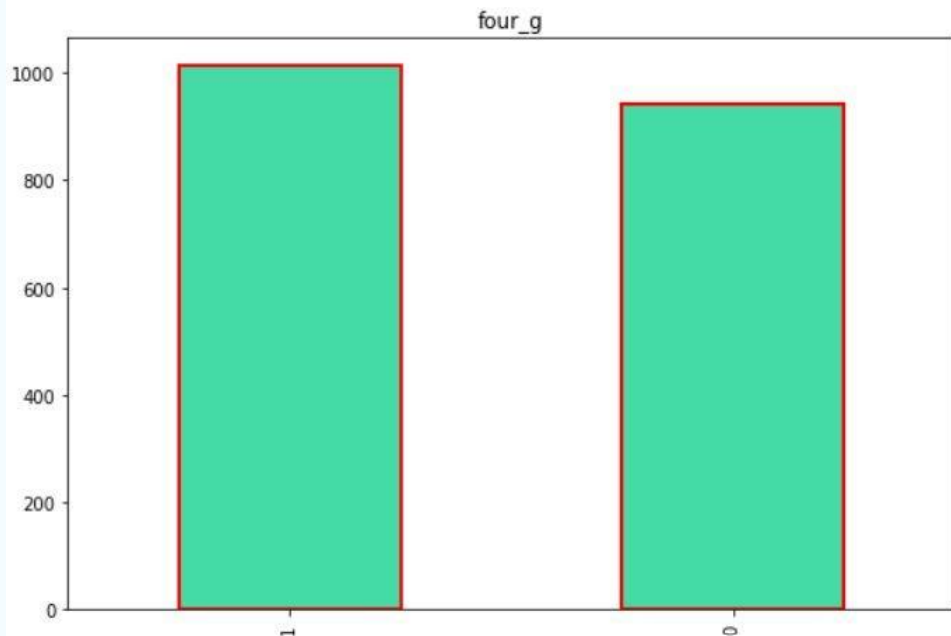
Values Counts on bluetooth



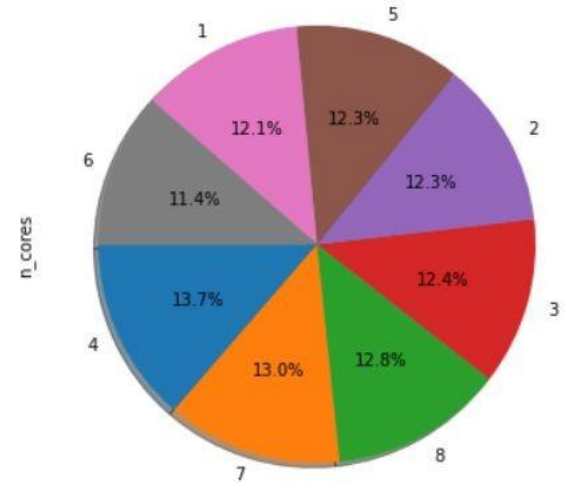
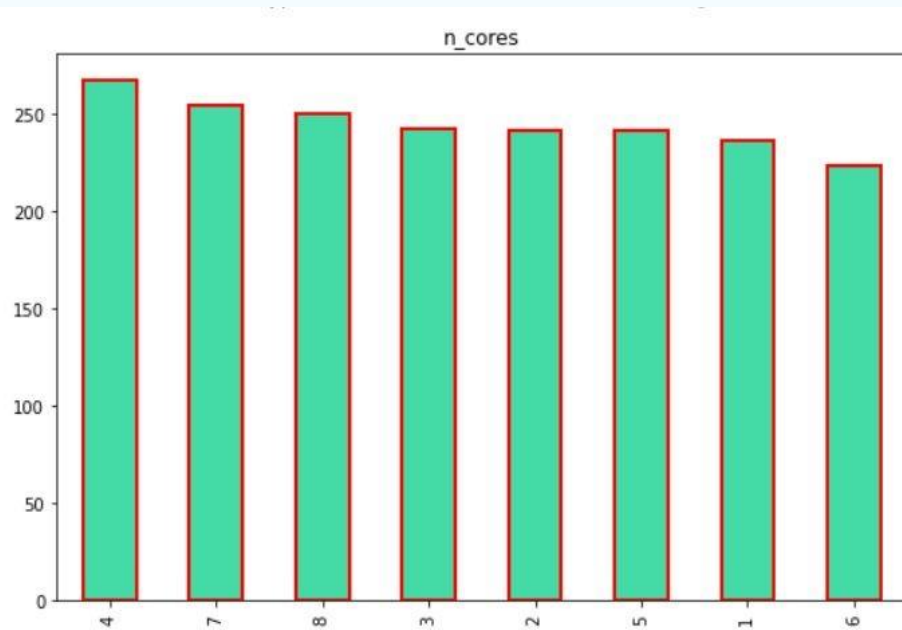
Value Counts on Dual Sim



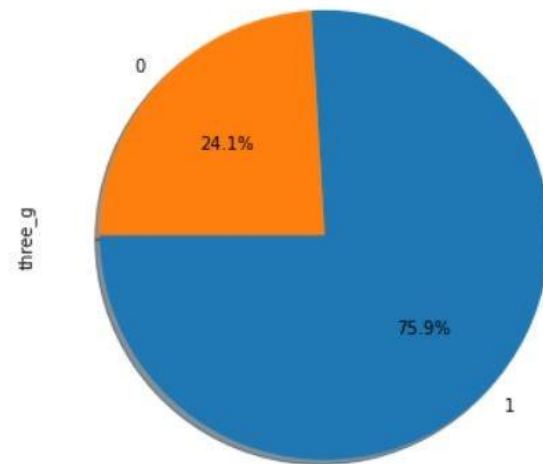
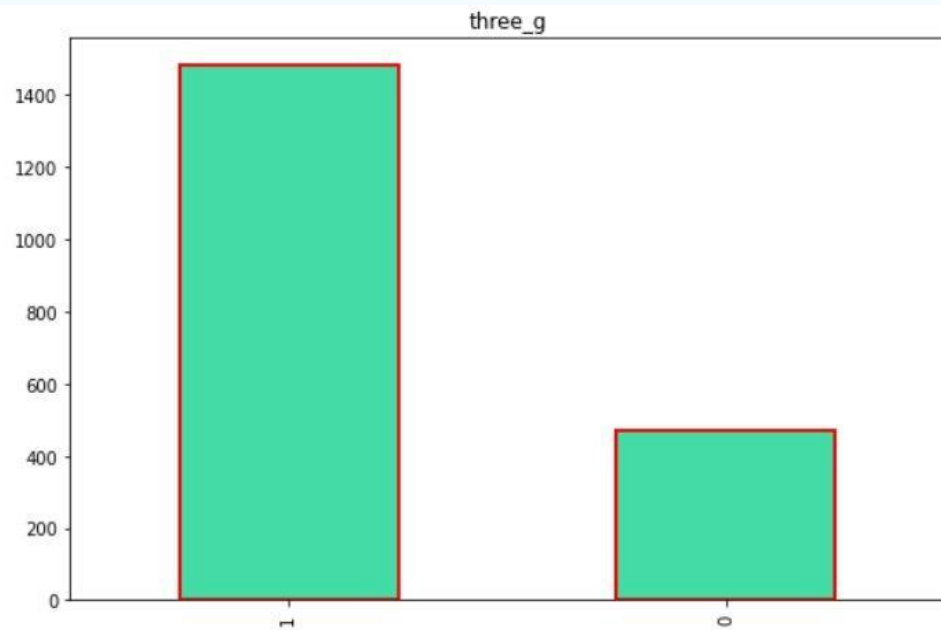
Values Counts on Four G



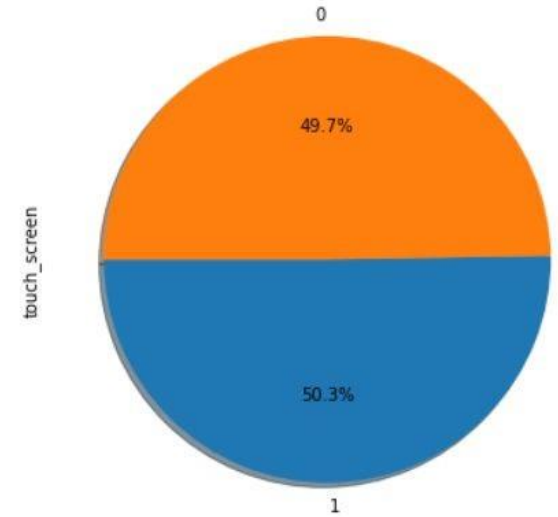
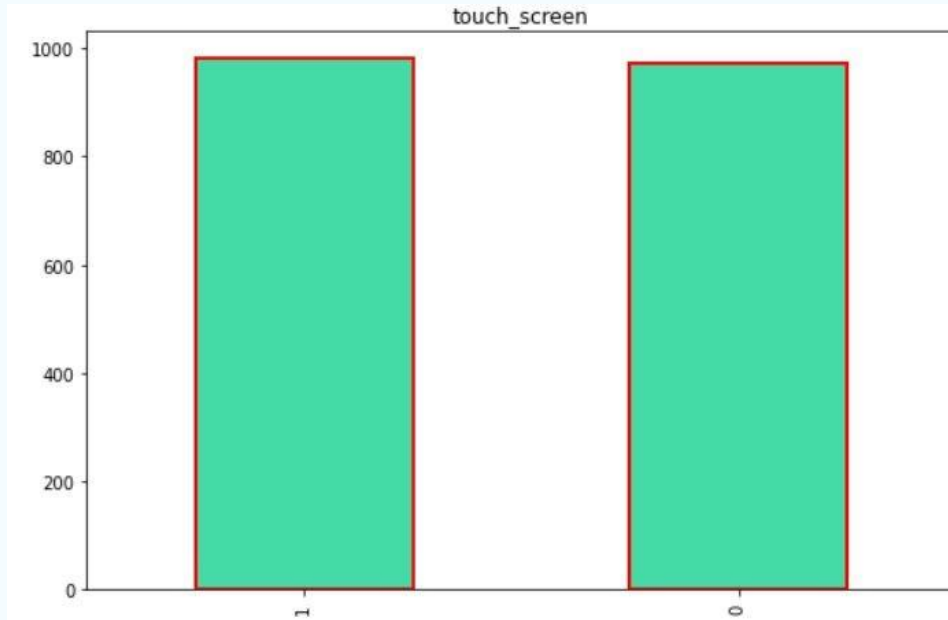
Values Counts on ncores



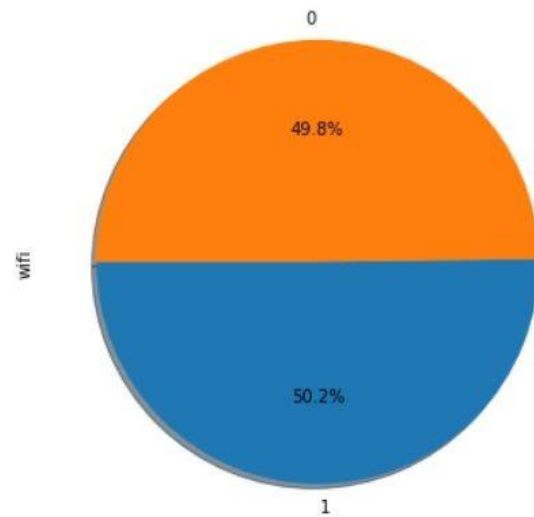
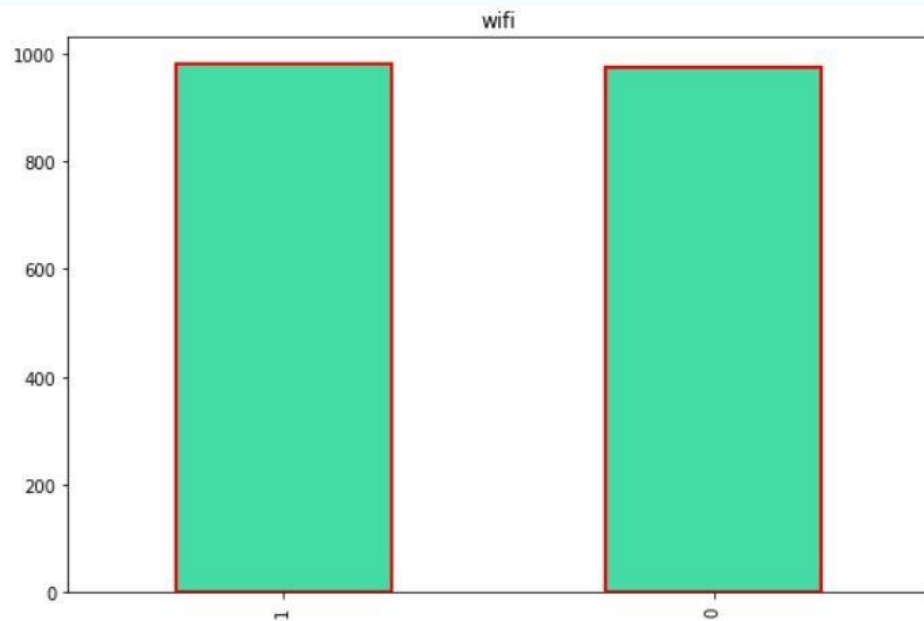
Value Counts on Three G



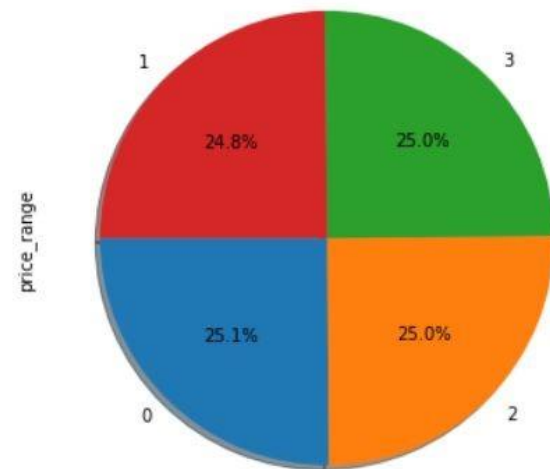
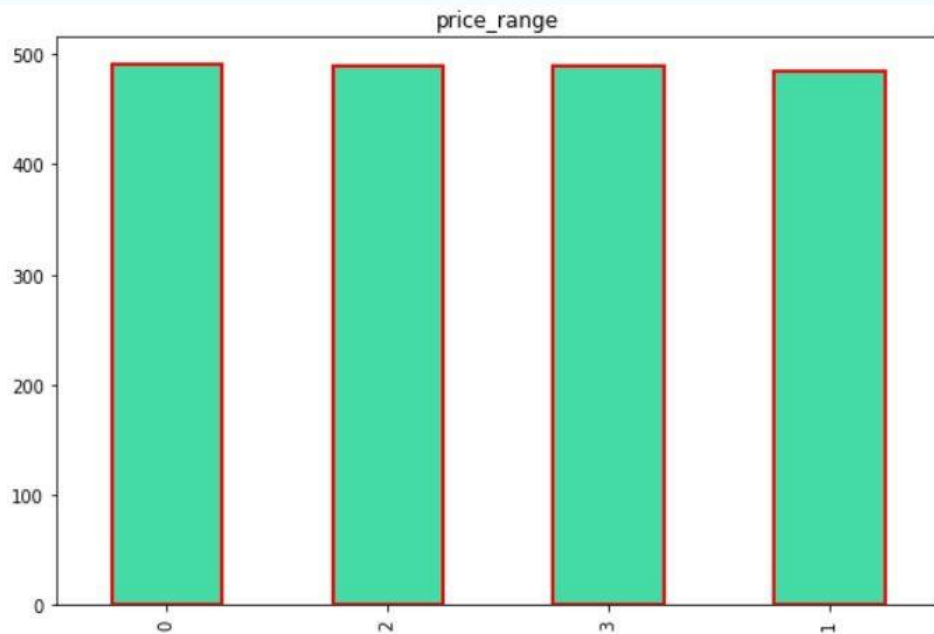
Value Count on Touch Screen



Value Counts on WiFi

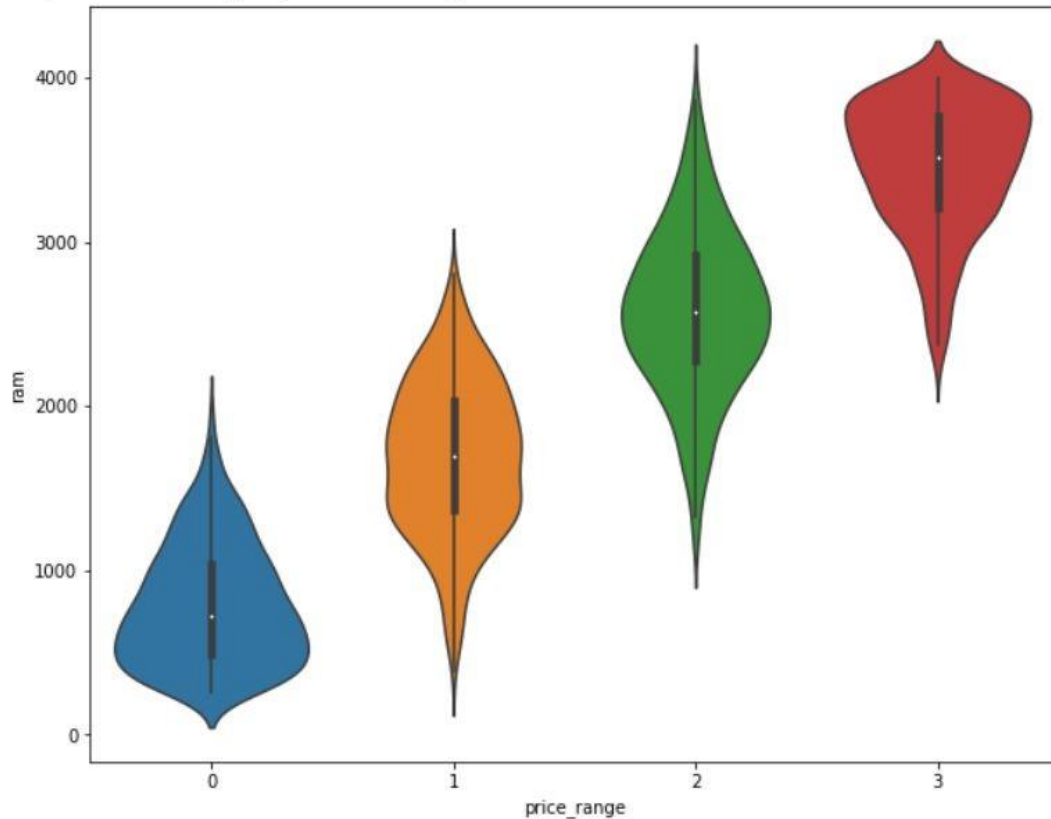


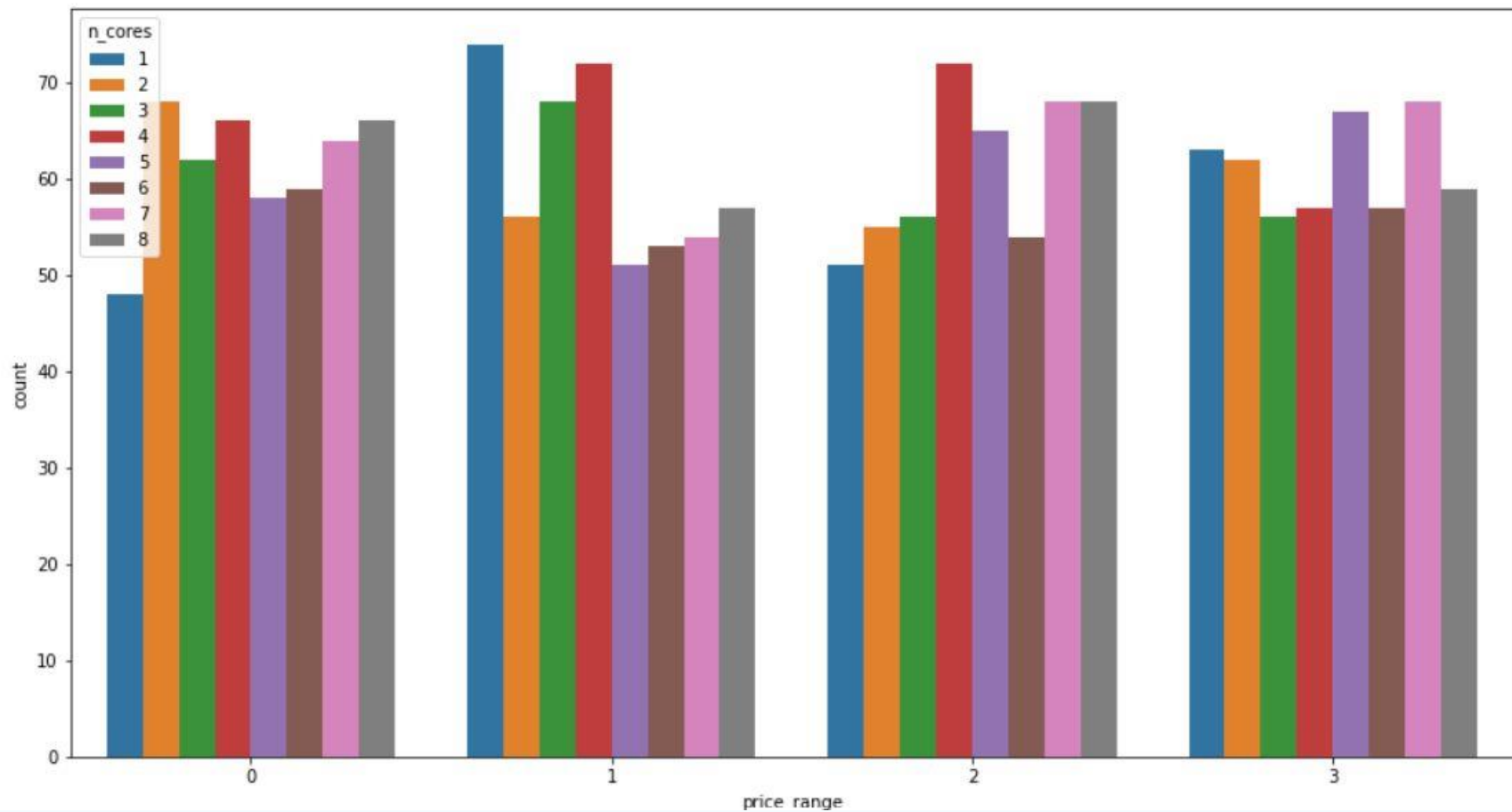
Values Counts on Price Range Prediction



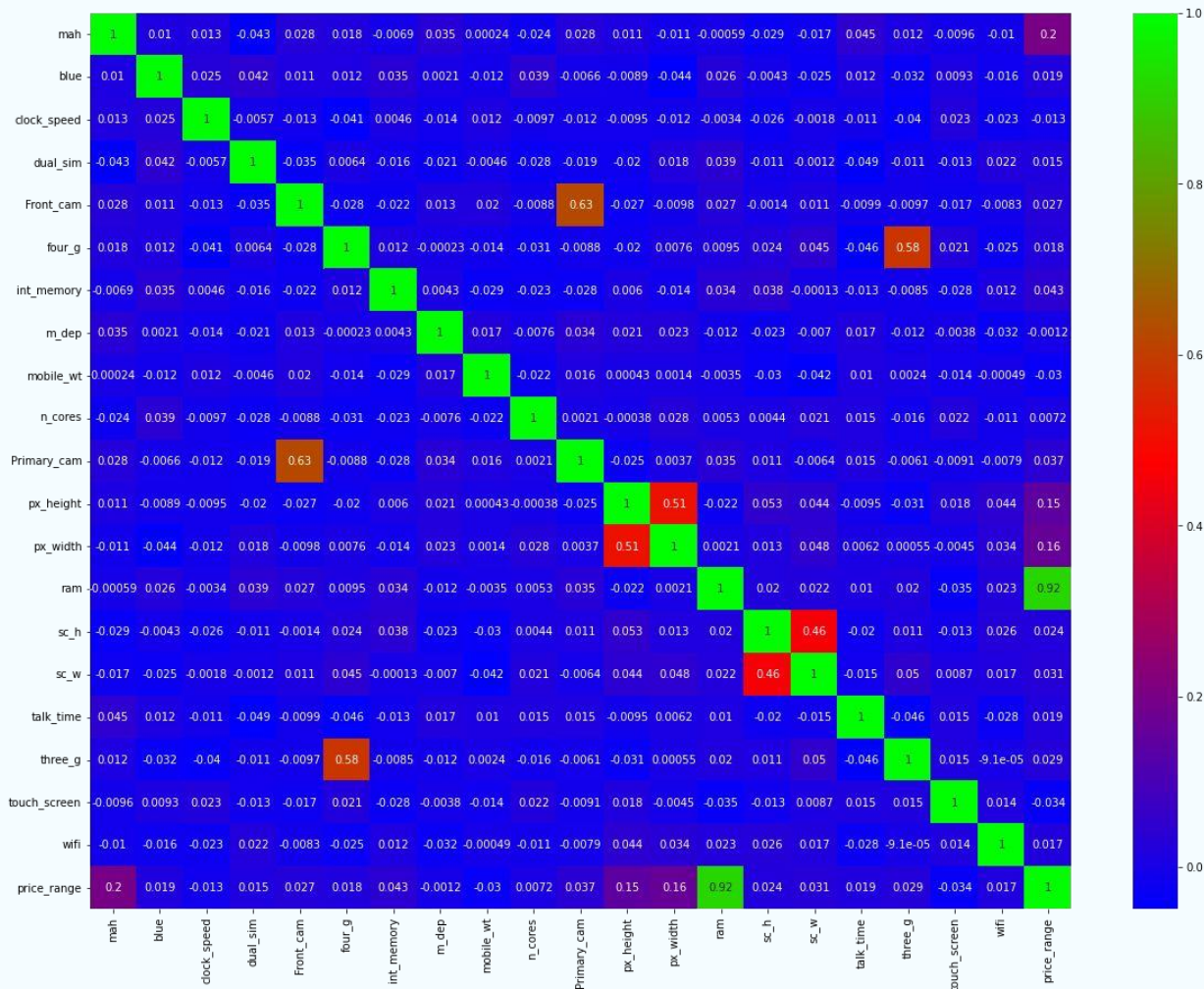
Bivariate Analysis

<matplotlib.axes._subplots.AxesSubplot at 0x7f6bd82e3910>





Correlation Matrix



Feature Selection

- There are 21 column but all are not Contributed equally so, I pick up $\frac{3}{4}$ Part of the dataset columns which is Highly contributed in Mobile price

```
[ ] print(featureScores.nlargest(15, 'Score'))
```

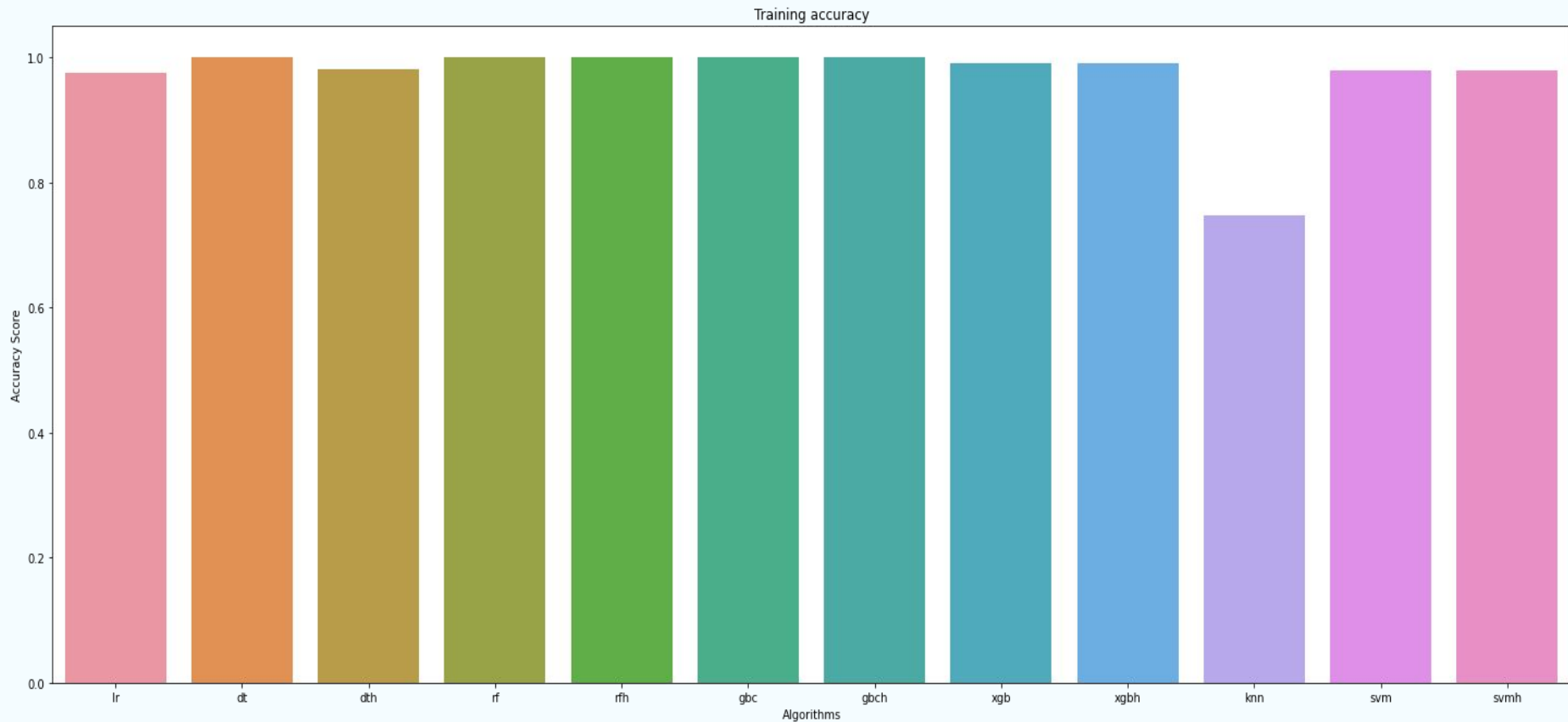
	Feature	Score
13	ram	909561.357223
11	px_height	15908.164022
0	mah	13505.022467
12	px_width	9079.507773
8	mobile_wt	93.737357
6	int_memory	79.957546
16	talk_time	12.131473
4	Front_cam	10.834975
10	Primary_cam	10.221820
14	sc_h	9.168885
9	n_cores	8.379480
15	sc_w	8.374349
18	touch_screen	1.884086
5	four_g	1.698998
2	clock_speed	0.823457

Algorithms for Machine Learning

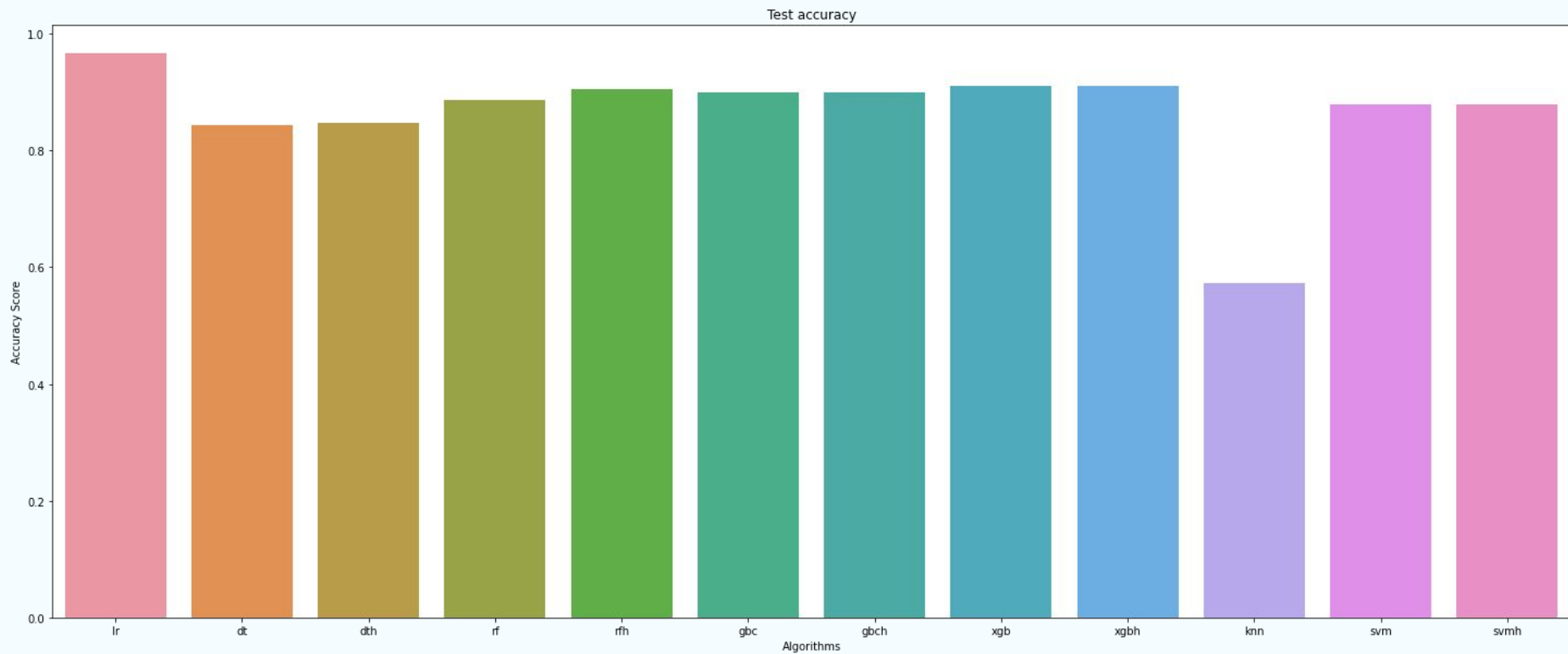
- Logistic Regression
- Decision tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- XGBoost Classifier
- K-Nearest-Neighbour Classifier
- Support Vector Classifier



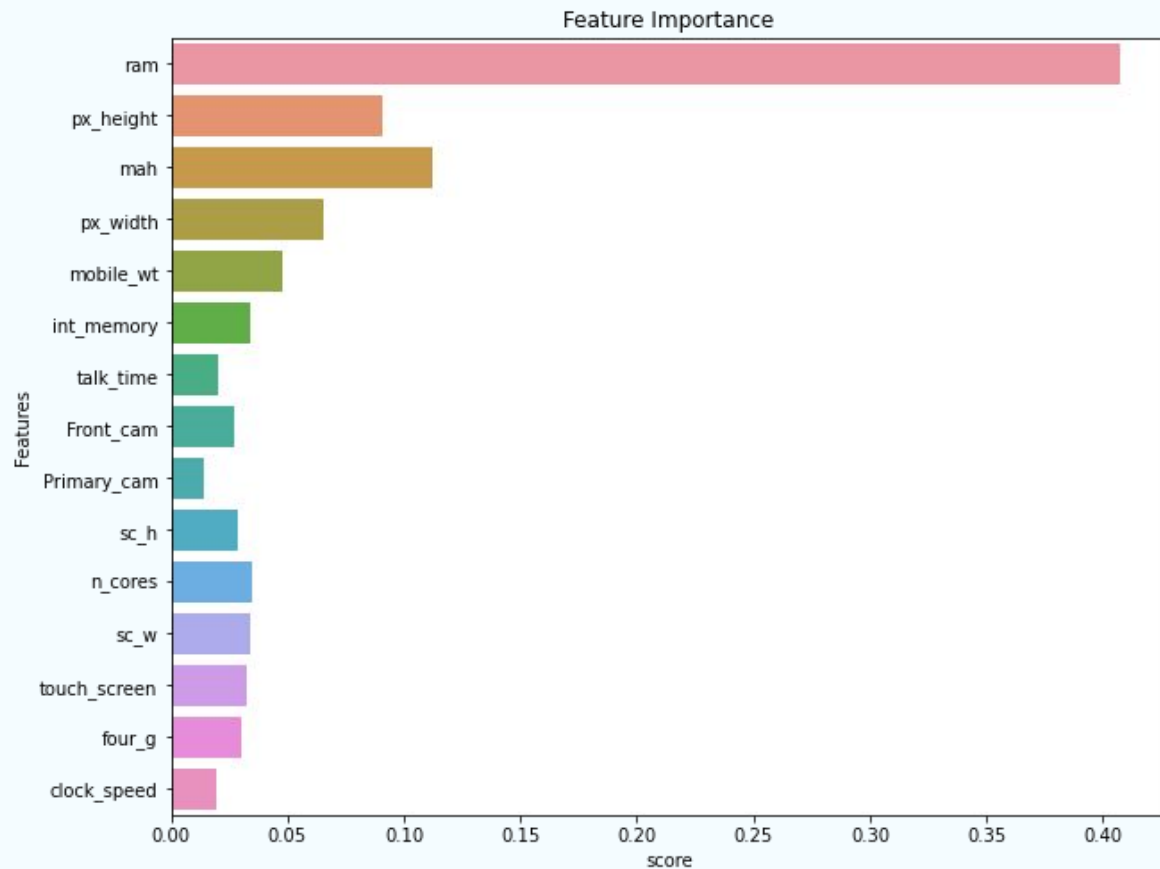
Performance On Training Dataset



Performance On Test Dataset



Feature Importance



Model Report



- **Decision Tree Classifier , Random Forest Classifier , Gradient Boosting Classifier , XGBoost Classifier and Support vector Classifier all are tends to Overfit on Training Data with Accuracy of 100%**
- **These above algorithms after hyperparameter tuning gives almost same result as they give in default condition**
- **These above algorithms gives almost 90% accuracy on test data because of overfitting issue**
- **K-Nearest-Neighbour algorithm performance is poorest as compare to others because it gives accuracy of 75% on training data and gives only 58% accuracy on test data. which is not good**
- **Logistic Regression performance is very good on this dataset because it gives 97.5% accuracy on training data and**
96.7% accuracy on test dataset

Conclusion

- RAM is responsible for price of mobile phone Range
- After RAM, Battery Power and Pixel are play a crucial role in price of mobile.
- Mobile Weight is negatively correlated with price
- More features increases price little bit.
- Logistic Regression algorithm gives accuracy of 97.5% on training data and 96.7% on test data which is highest in all the algorithms
So Logistic Regression
- Is the best Algorithm to predict Mobile price range

THANK
YOU