Capstone Project

Mobile Price Range Prediction

by

Lovejeet Singh

Contents

1. Problem Statement
2. Data Summary
3. Exploratory Data Analysis
4. Feature Engineering
5. Feature Selection
6. Model Implementation
7. Challenges Faced
8. Conclusion

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 (e.g.:- 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. The objective of the project is to come with a optimal machine learning model to predict sales.

Data Summary

In this Dataset we have 2000 number of observations and 21 features

Main Features

- **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

Contd…

- **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 charge will last when you are
- **Three_g** - Has 3G or not
- **Touch_screen** - Has touch screen or not
- **Wifi** - Has wifi or not

Our Target Variable.

- **Price_range** - This is the target variable with value of
  - 0(low cost),
  - 1(medium cost),
  - 2(high cost) and
  - 3(very high cost).

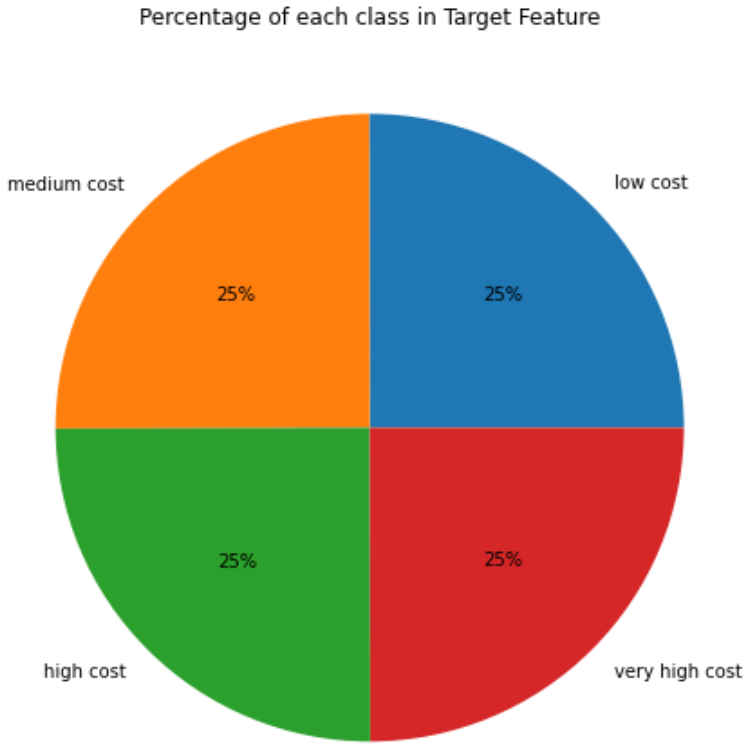
Null Values

In our dataset we have zero null values.

```
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
dtype: int64
```

Exploratory Data Analysis

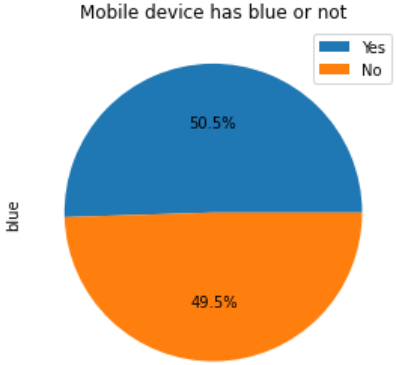
Checking Class ImBalanced

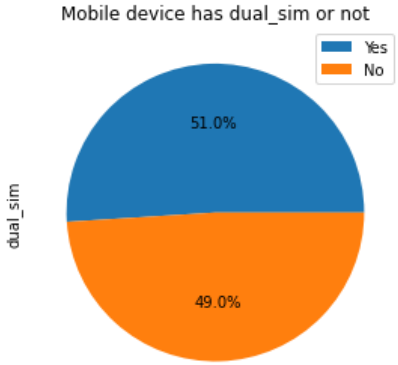


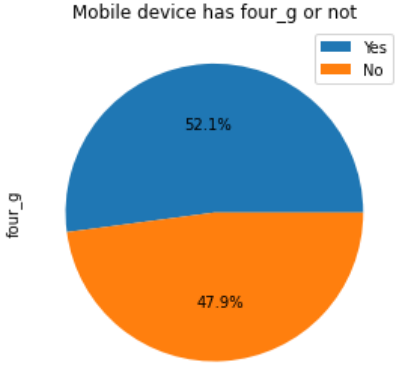
As we can see in this pie chart, we have very well class balanced dataset

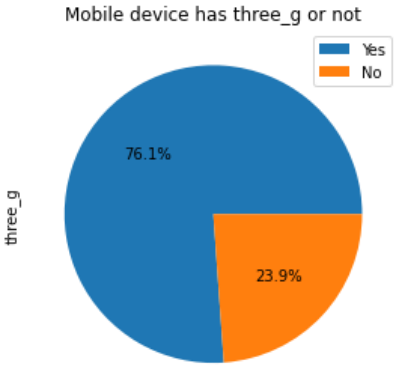
Accuracy Score will be the best evaluation metric to check our model performance

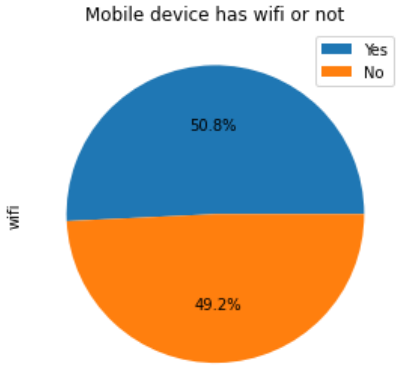
Univariate Analysis

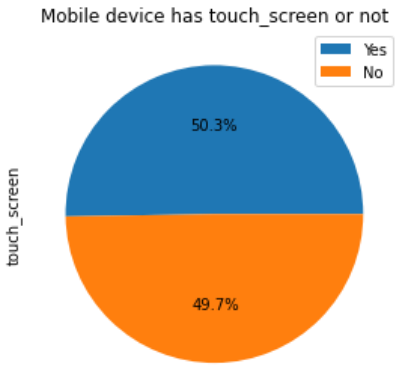




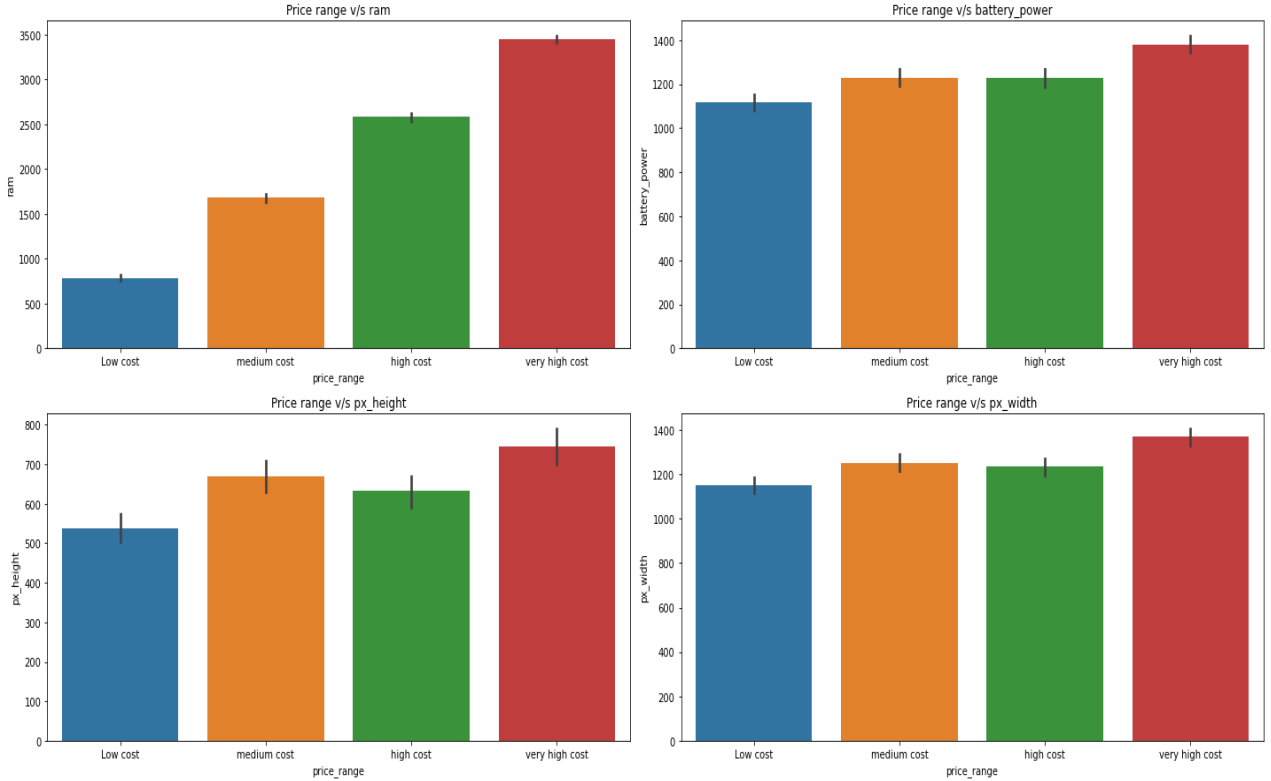




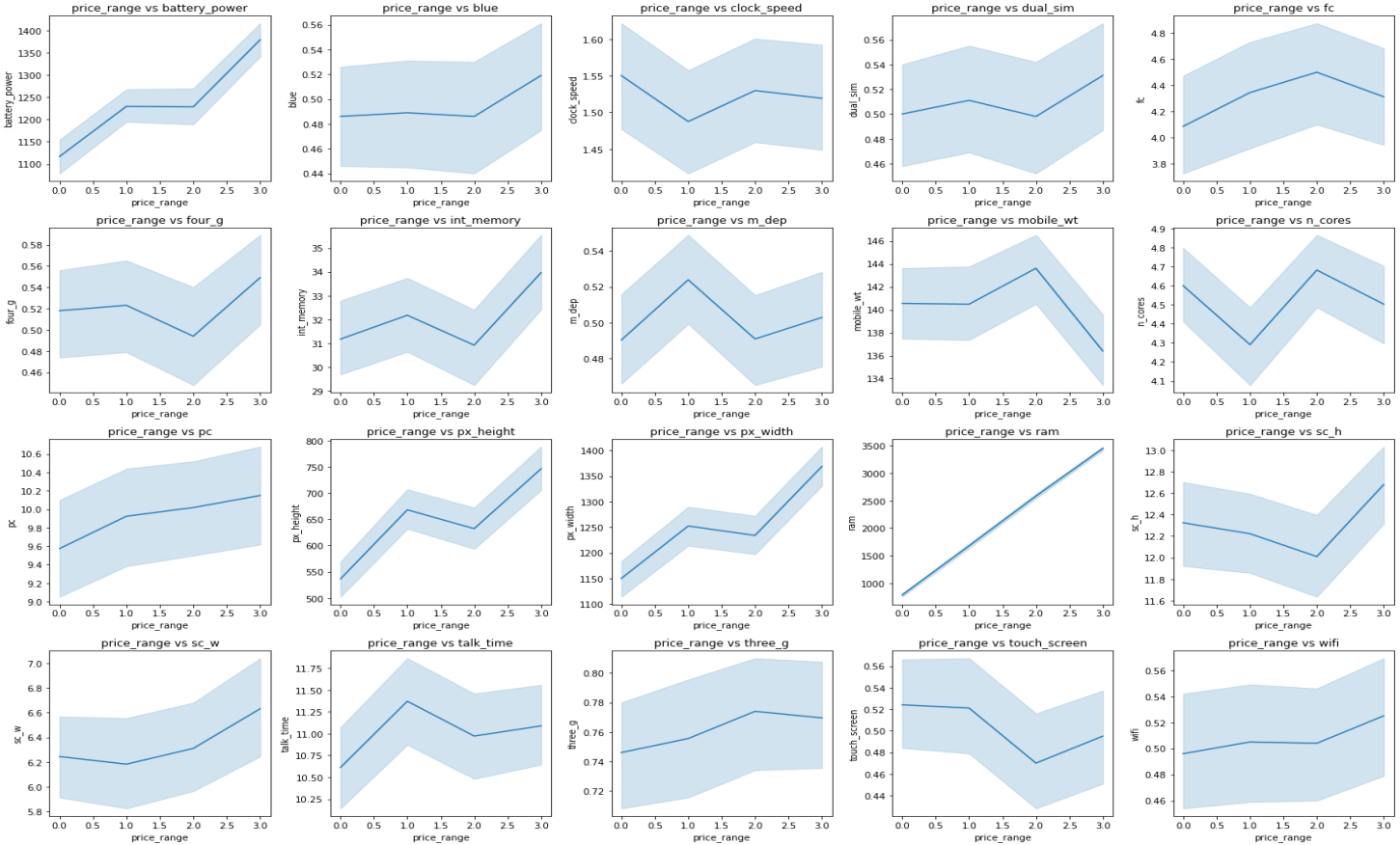




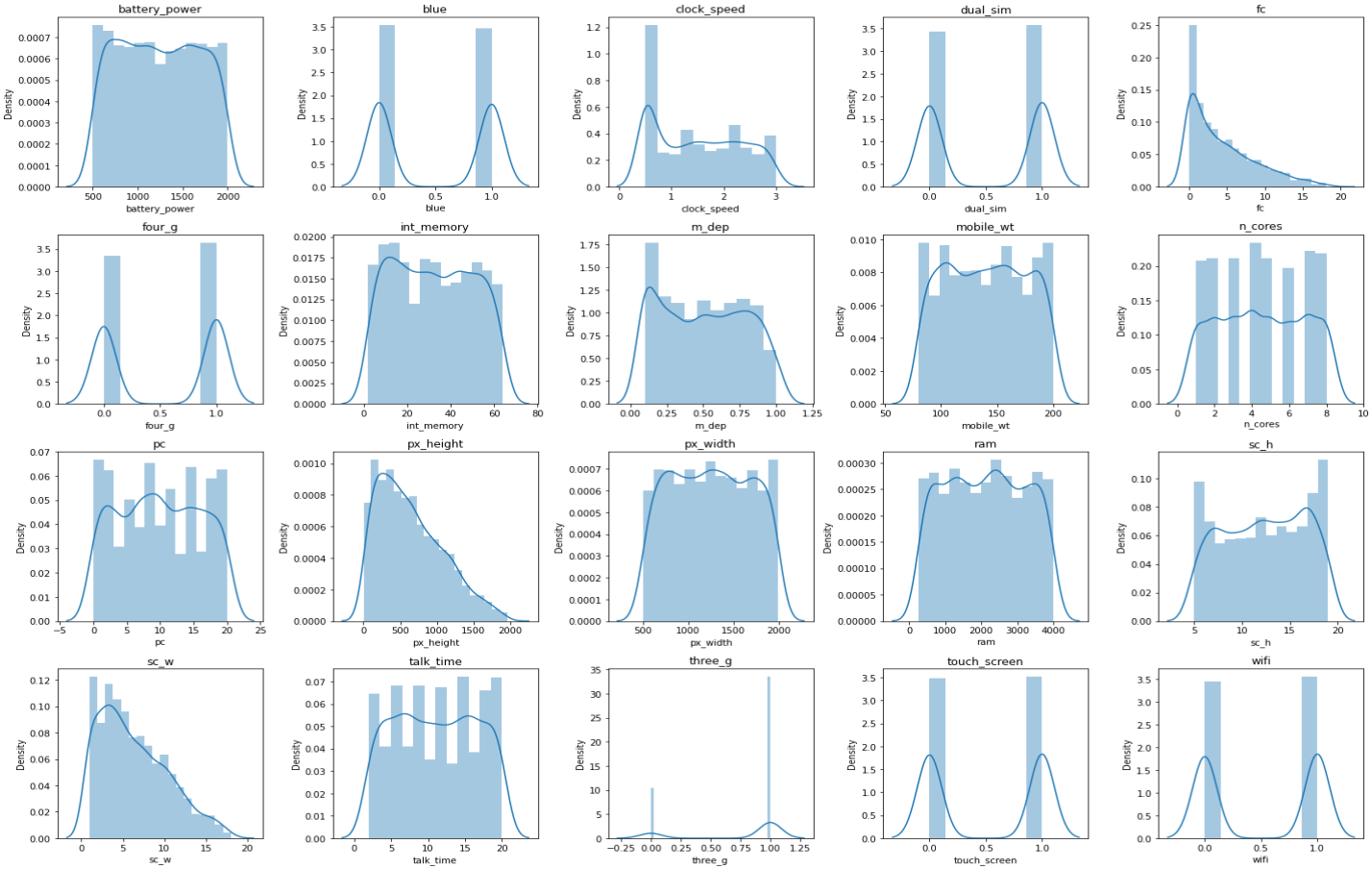
How Price Range Related to Numerical features



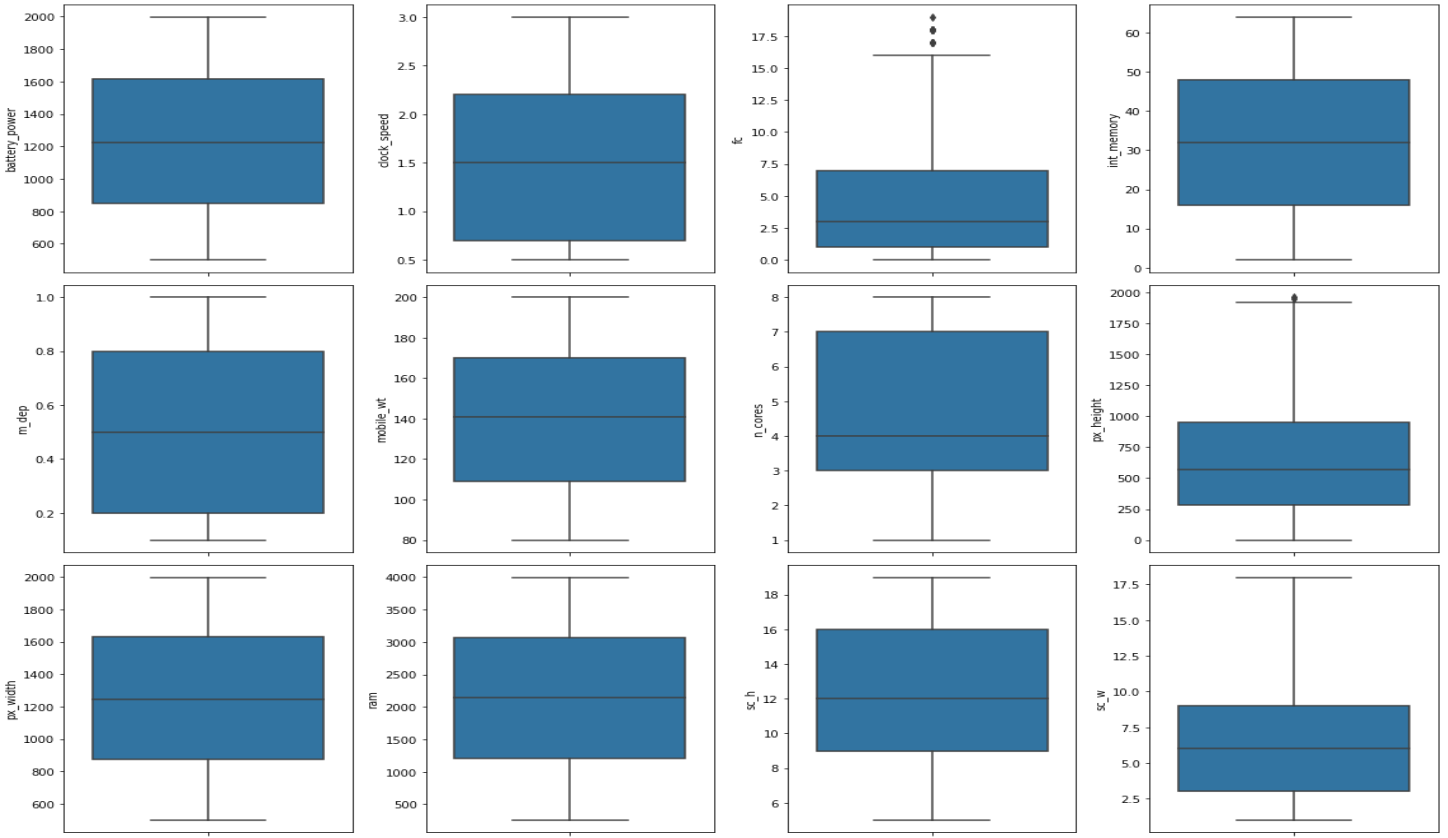
How features Drive Price Range



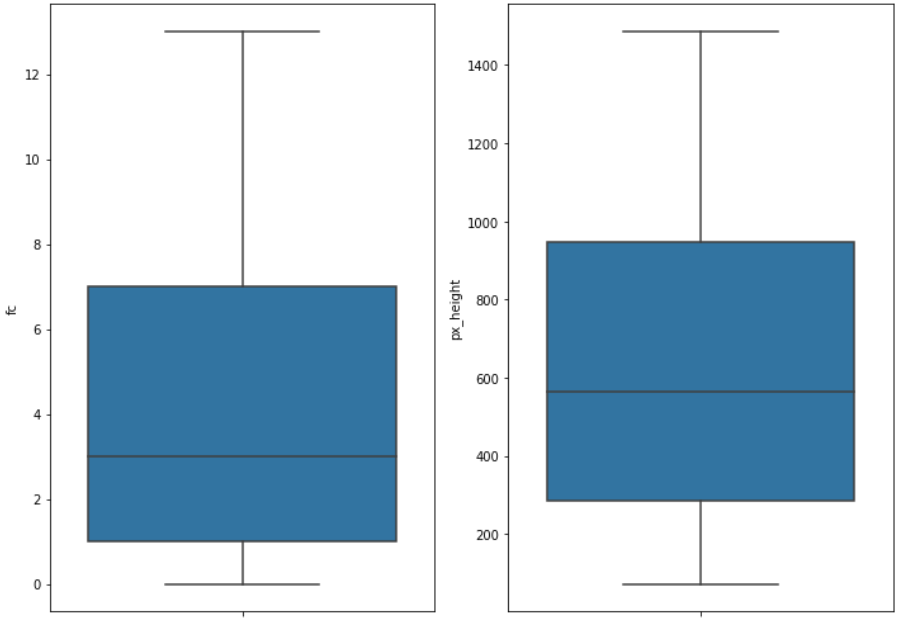
Checking Distribution



Detecting Outlier



Outlier Treatment



We have successfully treated Outlier by using IQR(Interquartile Range) formula

Lower Outlier = Q1 – (1.5 x IQR)

Higher Outlier = Q3 + (1.5 x IQR)

Feature Engineering

Creating New Feature

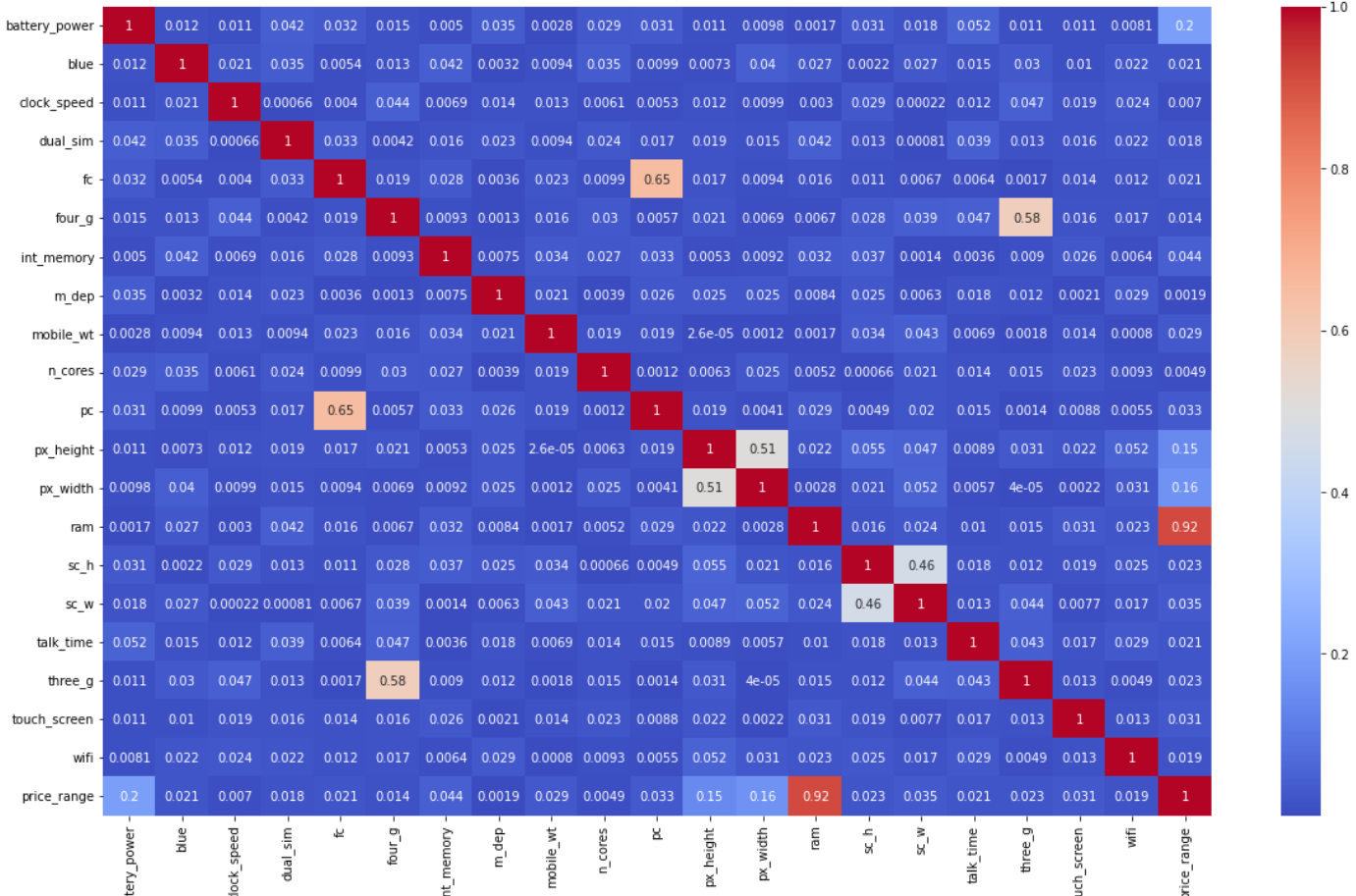
- We can make a new feature by simply multiple sc_h which is screen height and sc_w which is screen width to create a new feature called screen_area.
- After creating new feature called screen area we have to drop sc_h and sc_w feature from our dataset.

Null Values Treatment

	sc_w	px_height
count	2000.000000	2000.000000
mean	5.767000	645.108000
std	4.356398	443.780811
min	0.000000	0.000000
25%	2.000000	282.750000
50%	5.000000	564.000000
75%	9.000000	947.250000
max	18.000000	1960.000000

- We observed that sc_w and px_height has some observation which holds value as zero which is not possible like how screen width has 0 measurement.
- px_height has only 2 observation which has value as zero so we simply drop these observation/Rows.
- Using kNNImputer we are able to replace those values with their nearest neighbor.

Correlation Heatmap



Feature Selection

Multicollinearity

```
calc_vif(mobile_df[[i for i in mobile_df.describe().columns if i not in ['price_range']]])
```

	variables	VIF
0	battery_power	7.999994
1	blue	1.979866
2	clock_speed	4.243132
3	dual_sim	2.012806
4	fc	3.585027
5	four_g	3.188527
6	int_memory	3.911078
7	m_dep	3.905595
8	mobile_wt	12.615744
9	n_cores	4.610833
10	pc	6.261243
11	px_height	4.502284
12	px_width	11.674273
13	ram	4.656402
14	talk_time	4.826325
15	three_g	6.168841
16	touch_screen	1.985014
17	wifi	2.017588
18	screen_area	2.324745

- Collinearity of mobile_weight and px_width is little bit high. So we have to drop that columns only for Logistic Regression Classifier algorithm.
- Rest of all algorithm like decision tree, xgboost we are going to use all features.

Contd...

```
calc_vif(mobile_df[[i for i in mobile_df.describe().columns if i not in ['price_range','mobile_wt','px_width']]])
```

	variables	VIF
0	battery_power	7.534949
1	blue	1.973960
2	clock_speed	4.093746
3	dual_sim	1.981151
4	fc	3.583534
5	four_g	3.186915
6	int_memory	3.841244
7	m_dep	3.794567
8	n_cores	4.437792
9	pc	6.182835
10	px_height	3.169918
11	ram	4.535454
12	talk_time	4.655756
13	three_g	6.022200
14	touch_screen	1.975593
15	wifi	2.001209
16	screen_area	2.310417

- After removing mobile_weight and px_width. Now these are our final feature only for Logistic Regression Classifier algorithm.

Model Implementation

Algorithm used

Following are the Classification algorithm used.

1. Logistic Regression Classifier
2. Decision Tree Classifier
3. Random Forest Classifier
4. XGBoost Classifier
5. K Nearest Neighbour
6. Gradient Boosting Classifier
7. Support Vector Machine(SVM)

Evaluation Metrics For Classification

Following are the evaluation metrics for classification.

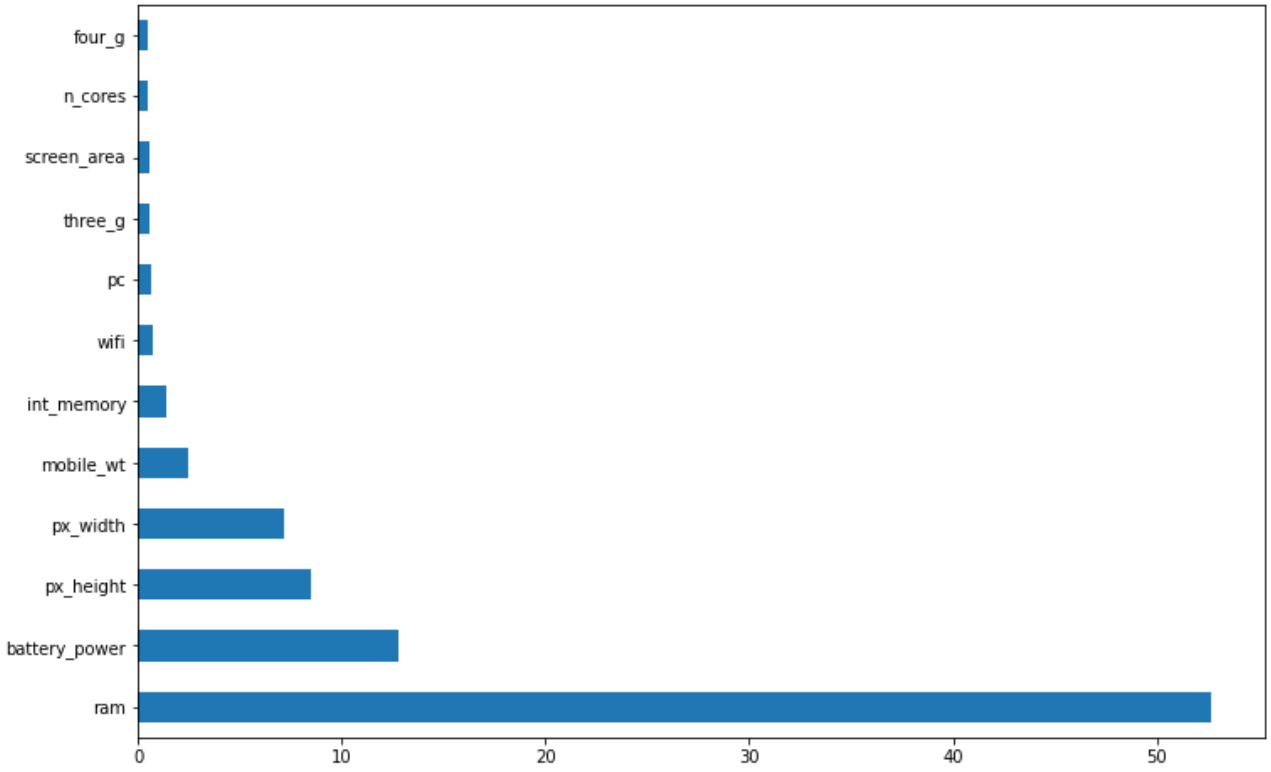
1. Accuracy
2. Confusion Metrics
    1. Precision
    2. Recall(Sensitivity)
3. Auc-Roc

- On the basis of Accuracy Score we evaluate our model performance on both train or test set.

Comparing different models

	Model	Accuracy Score On Train Set	Accuracy Score On Test Set
0	Logestic regression classifier	0.915	0.895
1	Logestic regression classifier with ElasticNet...	0.917	0.892
2	Decsion Tree classifier	1.000	0.800
3	Decsion Tree classifier with GridSearchCV	0.993	0.865
4	Random Forest classifier	1.000	0.895
5	Random Forest classifier With GridSearchCV	0.991	0.872
6	Gradient Boosting Classifier	0.999	0.902
7	Gradient Boosting Classifier With GridSearchCV	1.000	0.915
8	XGBoost Classifier	0.989	0.895
9	XGBoost Classifier With GridSearchCV	0.998	0.912
10	K Nearest Neighbors	0.704	0.515
11	K Nearest Neighbors With GridSearchCv	0.759	0.665
12	Support Vector Machine(SVM)	0.986	0.868
13	Support Vector Machine(SVM) With GridSearchCV	0.983	0.960

Features Importance



Model Selection

- By looking evaluation metric on both train and test set. We decided to go with Support Vector Machine Model in which we got 0.98 Accuracy score on train and 0.96 Accuracy score on test set.
- The Logistic Regression Classifier with ElasticNet penalty model also perform very well and we got 0.91 accuracy Score on train and 0.89 on test set.
- there was a bit overfitting seen but after tuning hyperparameter we were able to reduce overfitting.

Challenges

- sc_w features has some observations has value as zero .But when we use isnull function we got nothing but when we use describe method, we able to see these values.
- Handling outlier using IQR.
- Some algorithm seems overfit so choosing the best parameter to reduce the over fitting its an challenge for us.

Conclusions

- Our target variable is well balanced. There is no class imbalance seen.
- The analysis of the catagorical features like blue(bluetooth), dual_sim, touch_screen and four_g we saw that 50% of the devices has these feature and 50% hasn't.
- The 75% of the devices has feature called three_g.
- when we did analysis of the numerical features like ram, battery_power, px_height and px_width h. we found that these features directly proportional to price_range.
- The distribution of the most features look like normal distribution.
- we found two feature has outlier, so we must treat them using IQR formula.
- The accuracy score for logistic regression we got 0.91 on train set and 0.89 for test set by tuning hyperparameter. which is great.
- The Decision Tree Classifier model perform good but over fiitng seen. after tuning some hyper parameter, the model slightly improved but not that much..

Contd...

- The Random forest classifier model perform good but there was some overfitting seen. after tuning hyper parameter, the algorithm could not be able to reduce overfitting that much.
- The Gradient Boosting Classifier and XGBoost models performs same as random forest classifier.
- The K Nearest Neighbor algorithm perform worst among all the algorithms.
- The Support Vector Machine algorithm perform very well after tuning some hyper parameter using optimal algorithm search tool like GridSearchCV.
- The Support Vector Machine with Hyper parameter tuning gives accuracy score on train set is 0.98 and 0.96 on test set which is great. the F1 score is 0.96 on test set.
- the most important features are 'Ram', 'battery_power', 'px_height', 'px_weight' for all the algorithm.
- We can say that SVM model is ready to deploy.

THANK YOU