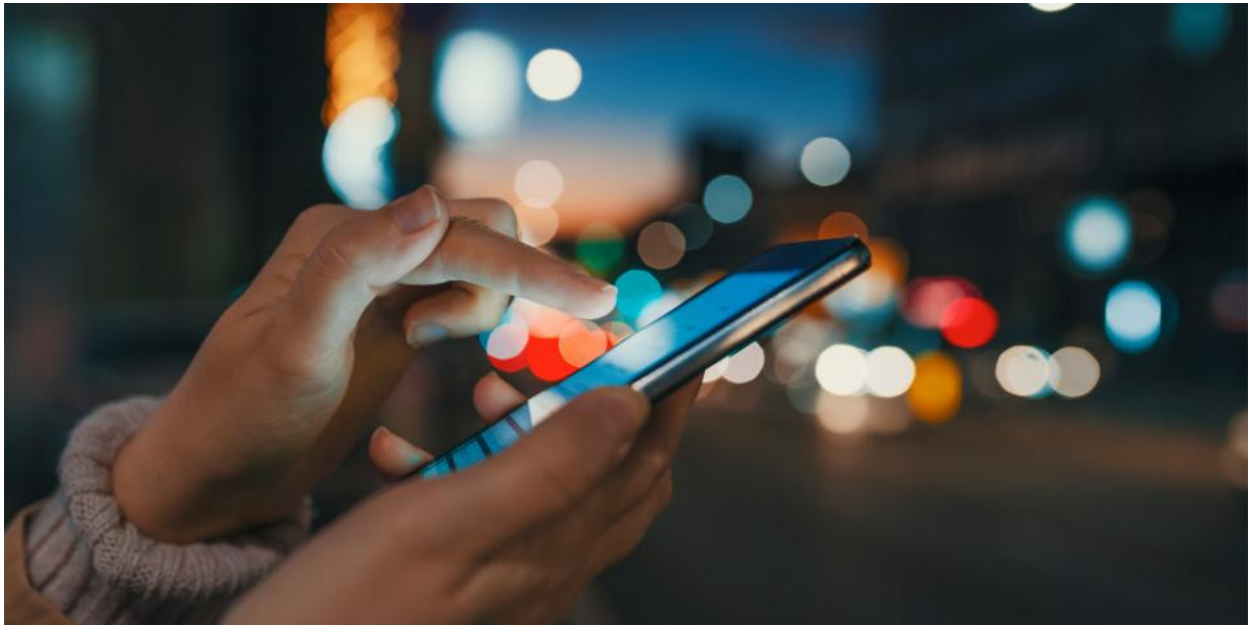


Mobile Price Range Prediction

Presented By

Tanay Tupe

Contents

* **Business domain**
* **Project Problem Description**
* **Dataset Summery**
* **Understanding Each Variable**
* **Modules in project**
    1. Importing necessary libraries, Loading and understanding data
    2. Exploratory Data Analysis
    3. Feature Engineering and Preparation for prediction model
    4. Implementing Classification models
* **Conclusion**

Business Domain

The mobile industry is a subset of the telecommunications industry focused on mobile phones, phone service, and peripheral devices. This industry has been steadily developing and growing, both in terms of market size and models. The number of smartphone subscriptions worldwide surpasses six billion and is expected to grow by several hundred million in the next few years further. China, India, and United States are the countries with the highest number of smartphone users.

Project Problem Description

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 and its selling price. In this problem, Our goal is to build a Classification ML model to predict the rank of price range for new models to launch.

Dataset Summery

The dataset we have used here is “Mobile Price Range Prediction”.This dataset in it’s raw form has around “2000” observations and “21” columns and it’s a mix between categorical, numeric and Ordinal values.

Also, we have "0" null values and no duplicated observations throughout our dataset. Target variable here is “Price_range” representing 4 different categories of range.

Understanding Each Variable

Let's Get Familiar With Each feature Present In The DataSet.

1. **Battery_power** - Total energy a battery can store in one time measured in mAh. (Continuous)

2. **Blue** - Has bluetooth or not. (Categorical)

3. **Clock_speed** - Speed at which microprocessor executes instructions. (Ordinal)

4. **Dual_sim** - Has dual sim support or not. (Categorical)

5. **Fc -** Front Camera mega pixels. (Ordinal)

6. **Four_g** - Has 4G or not. (Categorical)

7. **Int_memory** - Internal Memory in Gigabytes. (Ordinal)

Understanding Each Variable (Continued)

8. **M_dep** - Mobile Depth in cm. (Ordinal)

9. **Mobile_wt** - Weight of mobile phone. (Ordinal)

10. **N_cores** - Number of cores of processor. (Ordinal)

11. **Pc** - Primary Camera mega pixels. (Ordinal)

12. **Px_height** - Pixel Resolution Height. (Ordinal)

13. **Px_width** - Pixel Resolution Width. (Ordinal)

14. **Ram** - Random Access Memory in Mega Bytes. (Ordinal)

Understanding Each Variable (Continued)

15. **Sc_h** - Screen Height of mobile in cm. (Ordinal)

16. **Sc_w** - Screen Width of mobile in cm. (Ordinal)

17. **Talk_time** - Longest time that a single battery charge will last. (Ordinal)

18. **Three_g** - Has 3G or not. (Categorical)

19. **Touch_screen** - Has touch screen or not. (Categorical)

20. **Wifi** - Has wifi or not. (Categorical)

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

Module 1 Importing necessary libraries, Loading and understanding data.

Before anything, we'll be needing all relevant libraries as required in our analysis. Few universal libraries like Numpy, Pandas, Matplotlib, Seaborn are must needed, and are imported even before anything, but we also will be needing other libraries as we go through our analysis.

```
# importing all necessary libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# scaling, splitting, cross-validation
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, cross_validate, GridSearchCV

# feture extraction
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import chi2

# classification models
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier

# evaluation metrics
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import accuracy_score, make_scorer, f1_score
```

Module 1 Importing necessary libraries, Loading and understanding data.

As this analysis was performed on “Google Colab” we first need to mount the notebook to our drive, where we have our “**Mobile Price Range Prediction**” dataset in “CSV” format.

Checks were performed on dataset’s Size, Shape, general information on dataset with info() and describe() function. Where we observed number of rows(2000), features (21), the count of their null values, memory(328.2 KB) the dataset is occupying.

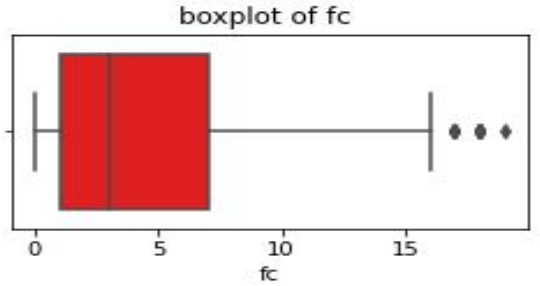
Also, we have a general overview of all numerical features and we can already make rough estimations over it. Like,

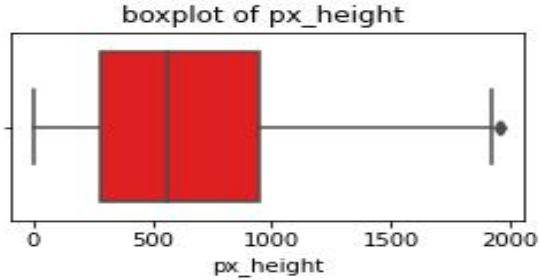
* Our target variable appears to uniform distribution.
* Main factors affecting could be "hour of the day", "Ram ", "Int_memory" etc.
* Median, mean, quartiles of each columns.

Module 2 Exploratory Data Analysis.

We began our EDA with univariate analysis. Where checks have been performed for type of distribution, skewness and outliers.

Considering outliers, we encountered very few outliers in features like “Front Camera megapixels” and “pixel height”.



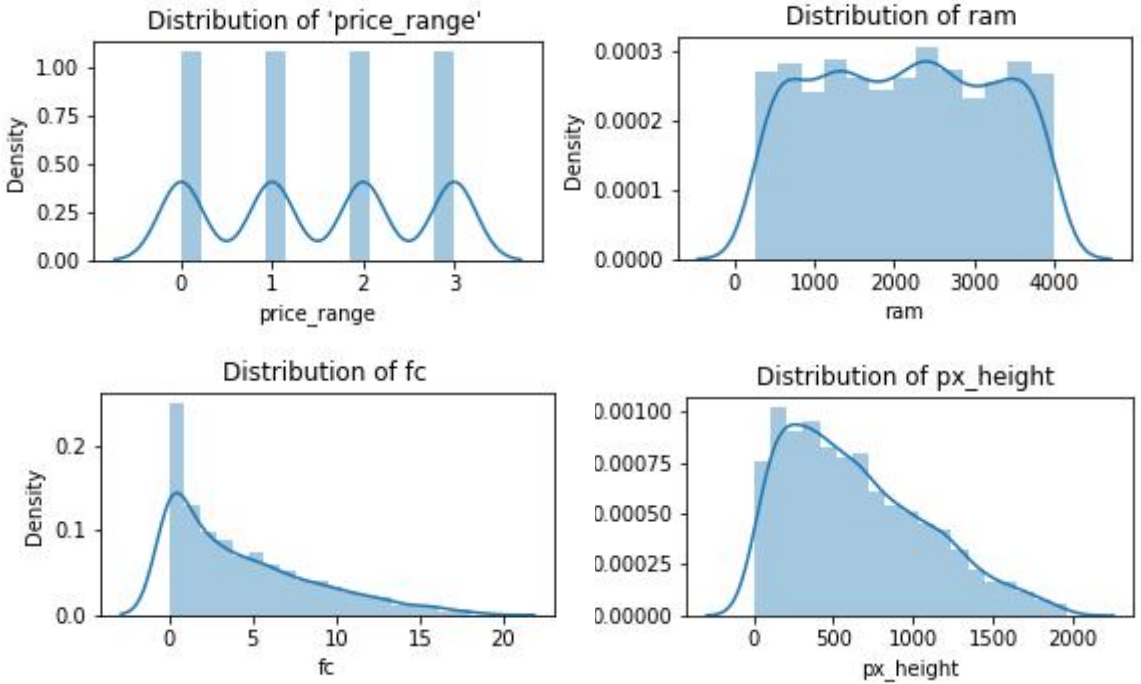


Our dataset is fairly clean from outliers. We have very few of them which won't disturb our analysis.

Reason we are considering outliers is because few important variables like "Front Camera megapixels" have a maximum value of about 20 Megapixels from dataset. But, mobiles in market today do have camera having 20+ Mp. It might be that those are legit observations.

Module 2 Exploratory Data Analysis.

Analyzing distributions of variables.

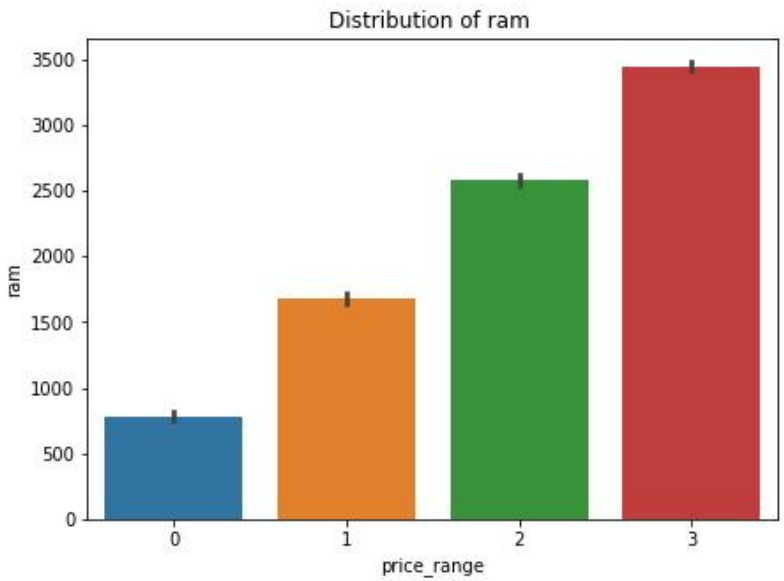


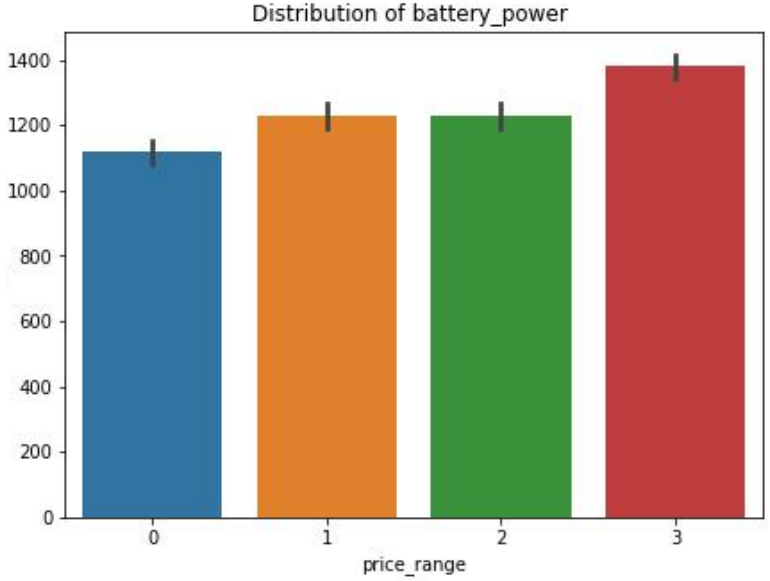
Most of the features have uniform distribution including our target variable. But, we have some skewed variables like “Front camera MP” and “Pixel Height”.

It’s difficult to choose a algorithm in this case, but as it’s a multi-class classifier we’ll stick with KNN or Tree based models.

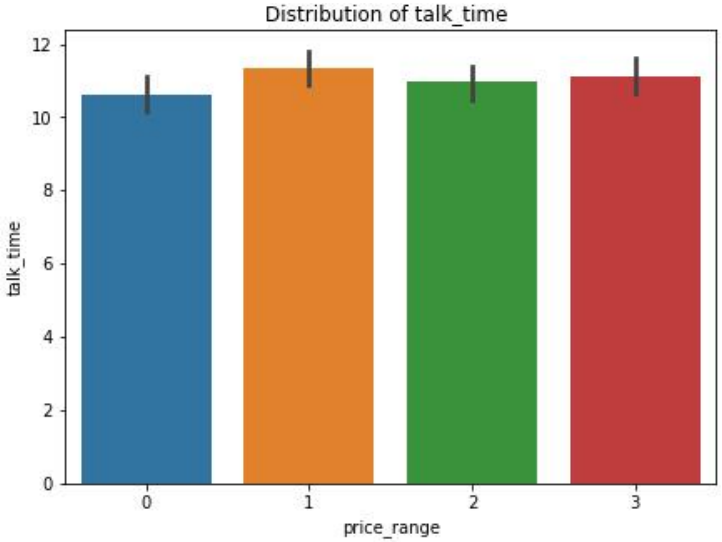
Module 2 Exploratory Data Analysis.

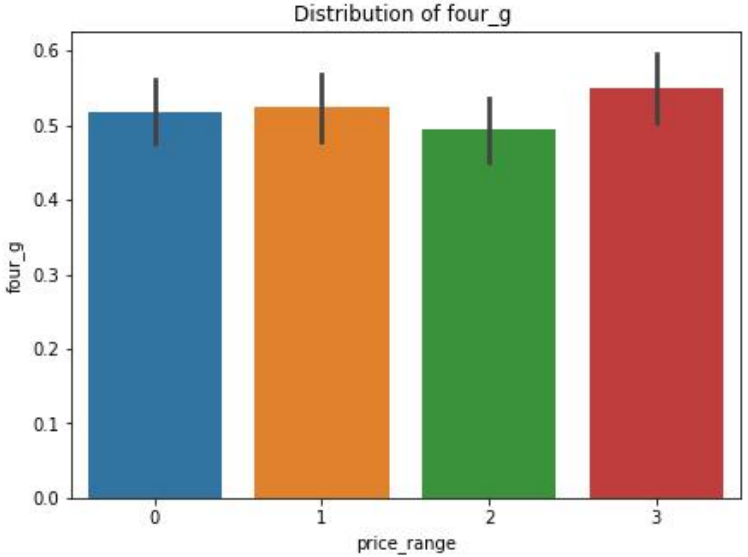
Bivariate analysis. Checking for relationship between dependent and independent variables.





Module 2 Exploratory Data Analysis.





All other features are mostly uniformly distributed considering price range. Above we can see two examples of “Talk Time” and weather the model has 4G or not.

Module 3 Feature Engineering and Preparation for prediction model

Feature engineering is the addition and construction of additional variables, or features, to your dataset to improve machine learning model performance and accuracy.

Feature engineering facilitates the machine learning process and increases the predictive power of machine learning algorithms by creating features from raw data.

New Features created.

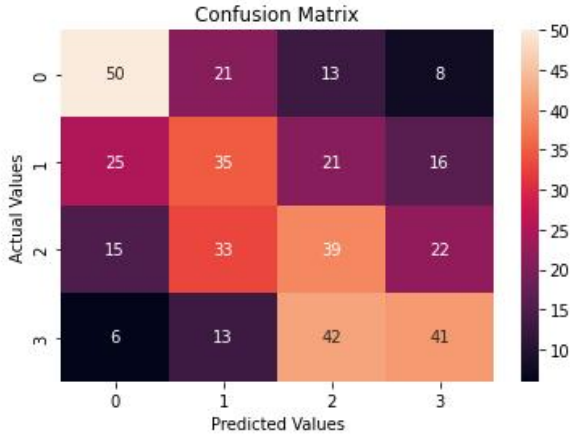
* We introduced new feature **"px_area"** which will be combination of "px_width" and "px_height". We multiplied these two to represent area of particular pixel.

* Also, **"sc_area"** was introduced, taking place of "sc_h" and "sc_w". Here again we multiplied these two to get screen area.

Module 4 Implementing classification models

Implementing KNN Classifier.

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It's a non-parametric algorithm, which means it does not make any assumption on underlying data. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

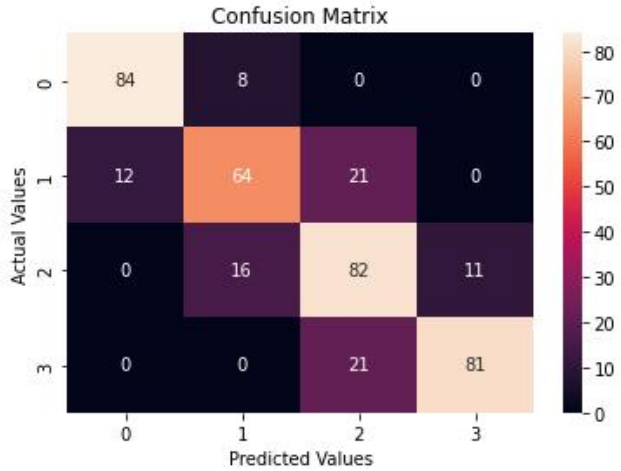


	precision	recall	f1-score	support
0	0.52	0.54	0.53	92
1	0.34	0.36	0.35	97
2	0.34	0.36	0.35	109
3	0.47	0.40	0.43	102
accuracy			0.41	400
macro avg	0.42	0.42	0.42	400
weighted avg	0.42	0.41	0.41	400

Module 4 Implementing classification models

Implementing Decision Tree Classifier.

Decision Tree Classifier is a structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.

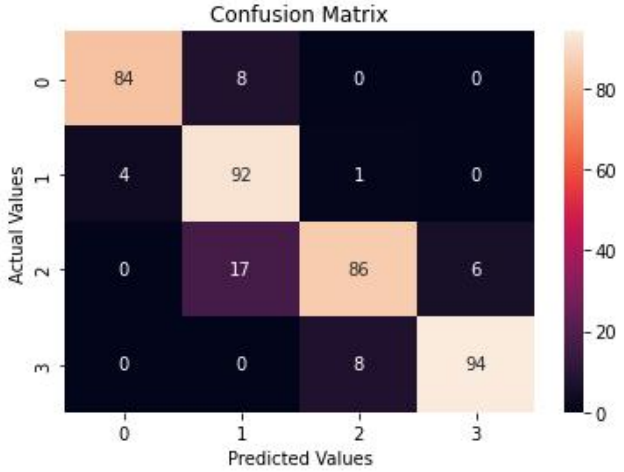


	precision	recall	f1-score	support
0	0.88	0.91	0.89	92
1	0.73	0.66	0.69	97
2	0.66	0.75	0.70	109
3	0.88	0.79	0.84	102
accuracy			0.78	400
macro avg	0.79	0.78	0.78	400
weighted avg	0.78	0.78	0.78	400

Module 4 Implementing classification models

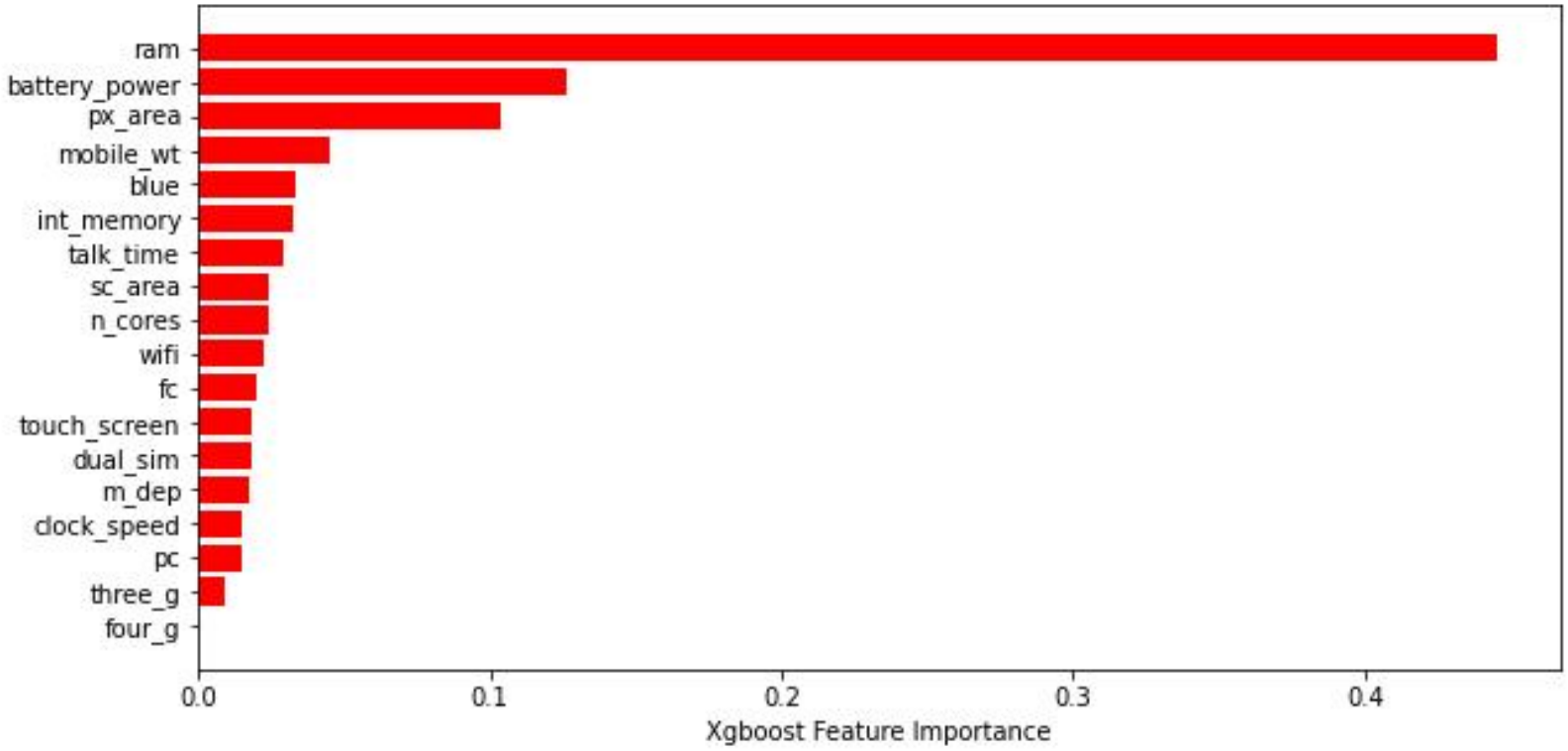
Implementing XGBoost Classifier.

The XGBoost algorithm is effective for a wide range of regression and classification predictive modeling problems. It is an efficient implementation of the stochastic gradient boosting algorithm and offers a range of hyperparameters that give fine-grained control over the model training procedure.



	precision	recall	f1-score	support
0	0.95	0.91	0.93	92
1	0.79	0.95	0.86	97
2	0.91	0.79	0.84	109
3	0.94	0.92	0.93	102
accuracy			0.89	400
macro avg	0.90	0.89	0.89	400
weighted avg	0.90	0.89	0.89	400

Feature Importance



Conclusion

Considering multi class classification, Logistic Classification won't fit this dataset well enough. Therefore, we used KNN, Decision Tree and XGBoost Classifier here.

Considering outliers, our dataset is fairly clean. We have very few of them which won't disturb our analysis. We encountered outliers in features like "Front Camera megapixels" and "pixel height".

While implementing KNN Classifier, even using best parameters we observed.
accuracy score was - 0.41
Precision - 0.52, 0.34, 0.34, 0.47
Recall - 0.54, 0.36, 0.36, 0.40
F1 Score - 0.53, 0.35, 0.35, 0.43
which is really not acceptable.

Conclusion

While implementing Decision Tree Classifier, tuning it's hyperparameters and using cross validation we observed metrics were way better than KNN. Here the model fit's well to test data and can be used to predict actual price range.

accuracy score was - 0.78
Precision - 0.88, 0.73, 0.66, 0.88
Recall - 0.91, 0.66, 0.75, 0.79
F1 Score - 0.89, 0.69, 0.70, 0.84

Implementing XGBoost Classifier, this is where we observed the highest values of all metrics compared to Decision Tree and KNN.

accuracy score was - 0.89
Precision - 0.95, 0.79, 0.91, 0.94
Recall - 0.91, 0.95, 0.79, 0.92
F1 Score - 0.92, 0.86, 0.84, 0.93

Conclusion

Top 5 features extracted from while fitting XGB model are.

1. Ram
2. Battery power
3. Pixel Area
4. Weight of mobile phone
5. Weather it has bluetooth or not