MOBILE PHONE PRICE PREDICTION SYSTEM

Rohan Suresh

15/10/2022

Problem Statement

Mobile Phones are an essential commodity of today's day and age. Nearly everyone owns one. Phone prices vary over a wide range from under ₹1,000 to over ₹170,000 in India. Prices vary over several aspects, mainly brand value and phone specifications (specs). Phone manufacturers in India are focussed on the 'value for money' component, making the market highly competitive. Majority of phones come with fairly good specs and are priced under ₹20,000 - ₹25,000. This brings about a need for a phone price prediction system, which would allow customers to have a benchmark for what is a fair price in the current market given their required phone specifications.

Market/Customer Need Assessment

Leaving brand value aside, there are only a given number of phone specifications (Camera Quality, RAM, ROM, Processor, Number of processor cores, etc) which largely vary the price of the phone. The different combinations of these features in phones is what varies the price. Looking at the market, you would find several phones with similar specs having huge price differences. A system that gives you an estimate of what the price would be given its features, would enable people to have good value for money.

Target Specification

- To build a model that takes phone specifications as input and predicts its price in the market with minimal error.
- Input features:
    - Brand
    - Front Camera Quality
    - Rear Camera Quality
    - Internal Storage (ROM)
    - RAM Capacity
    - Number of Processors (Cores)
    - Screen Size
    - Built-In Operating System
    - Phone Battery Capacity
    - Number of SIM Card Slots
- Output - Predicted Price

External Search

- Dataset
- Multiple Linear Regression
- Least Square Error for Multiple Linear Regression
- Pandas Documentation
- Box Plots with Tableau

Applicable Constraints

- Continuous updation of the dataset will be required since the mobile phone market actively changes.
- Most input features would be categorical, making the output price prediction subject to an error margin.

Business Model

This system allows customers to have an idea of the kind of price to expect from the market.
It can be made use of by phone manufacturers too. The estimated price given the input specs can determine the upper limit of the price to be set by the manufacturers in order to stay competitive in the market. Exceeding the estimate would draw customers to similar phones priced below the estimate. On the contrary, setting the price below the estimate could convince customers to opt for their product.

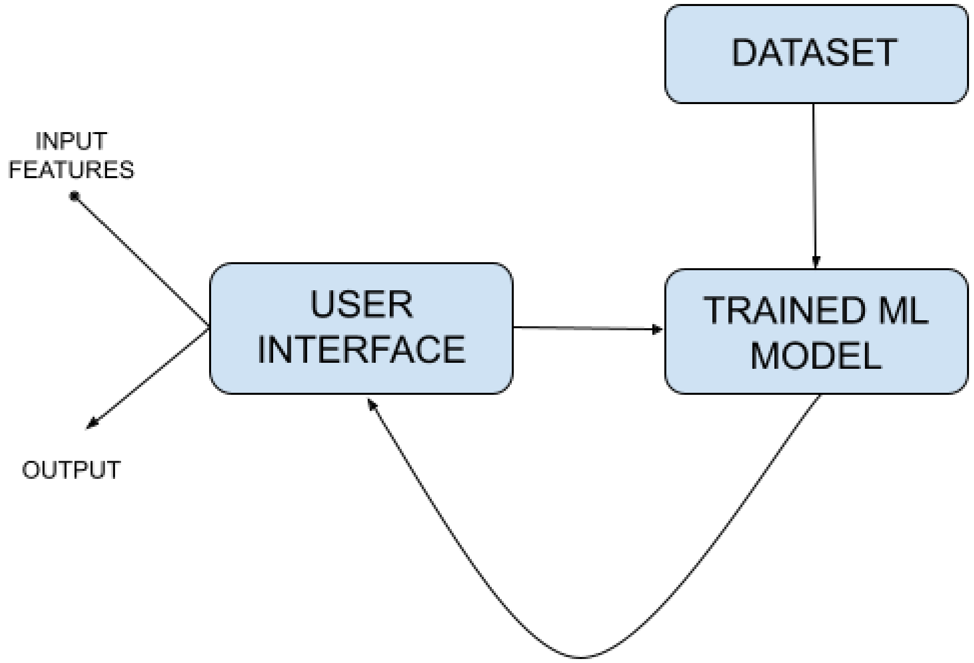
Concept Generation and Development

The dataset is analyzed and cleaned, removing outliers and categories which have very few values. The parameters to include in the model are decided upon. Binary variables except Number of SIMS are excluded since most of them have one-sided values. The important variables, in which the prices have reasonable variance in all categories are used for modeling.

Algorithm used: Linear Regression
Multiple Linear Regression, in specific. Most of the input features are categorical.
Brand and Operating System are treated as categorical. Front camera quality, Rear camera quality, Number of Processors, Number of SIMs are numerical variables but discrete. Hence they are treated as categorical too. ROM, RAM, Battery Capacity and Screen size are numerical variables treated as continuous since they follow a linear-like trend with increasing values.

The dataset is split for training and testing the model. The model is trained, then tested with the test split, and the test predictions are obtained. Finally, accuracy of the model is tested using the Root Mean Squared Error, and an error margin for the predicted prices is obtained.

Final Product Prototype



Product Details

- **Working**

  The Phone Prediction System would make use of a front end graphical user interface. This would receive the input from the user, allowing them to specify whichever features they want to, and get back the predicted price. The user need not specify all the features involved in modeling. A minimum of 1 feature would suffice, leaving the rest as null. The model would treat this as a multiple prediction request from the user, returning predictions for phones prices by fixing the specified features and iterating through all possible combinations of the remaining features.

For example, suppose the model involves only 3 features entirely: Brand, Operating System, and RAM. Assume that given below are all the available distinct categories in the dataset:

BRAND	OS	RAM
Xiaomi	Android	4 GB
Apple	IOS	6 GB
Samsung		8 GB
.		
.		
.		

Say the user specifies only 1 feature - RAM - as 6 GB. Then, the model would return predictions for phones having 6 GB RAM, Android OS and of all different brands, along with phones having 6 GB RAM, IOS and of all different brands.

The feature specifications given as input are then used to obtain a predicted price by the Multiple Linear Regressor. The GUI displays the result to the user.

- **Software Requirement**

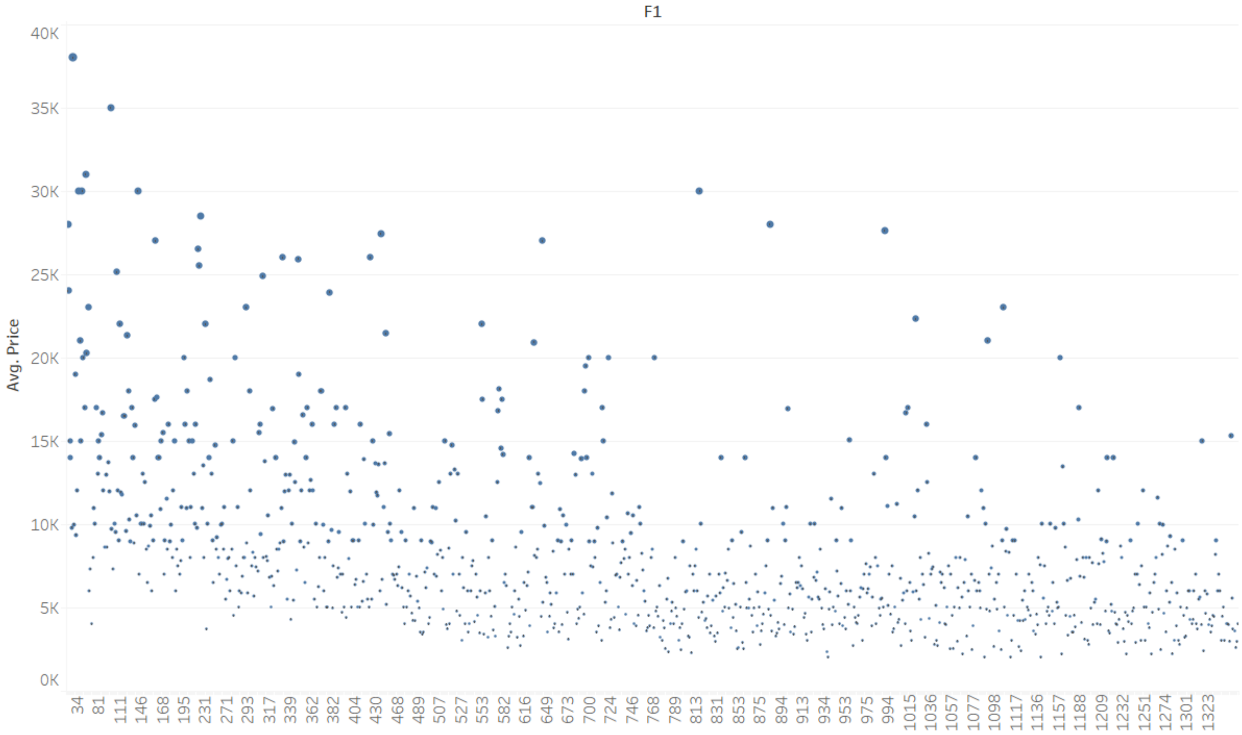
  - Python 3.7
  - Pandas
  - Sklearn
  - Web scraping tools
  - GUI Application

- **Team Required**
  1. Data Extraction Team - To scrape the web and update the dataset
  2. Data Analysts
  3. Data Scientists

Code Implementation (Small Scale)

Data Visualization and EDA (Done using Tableau)

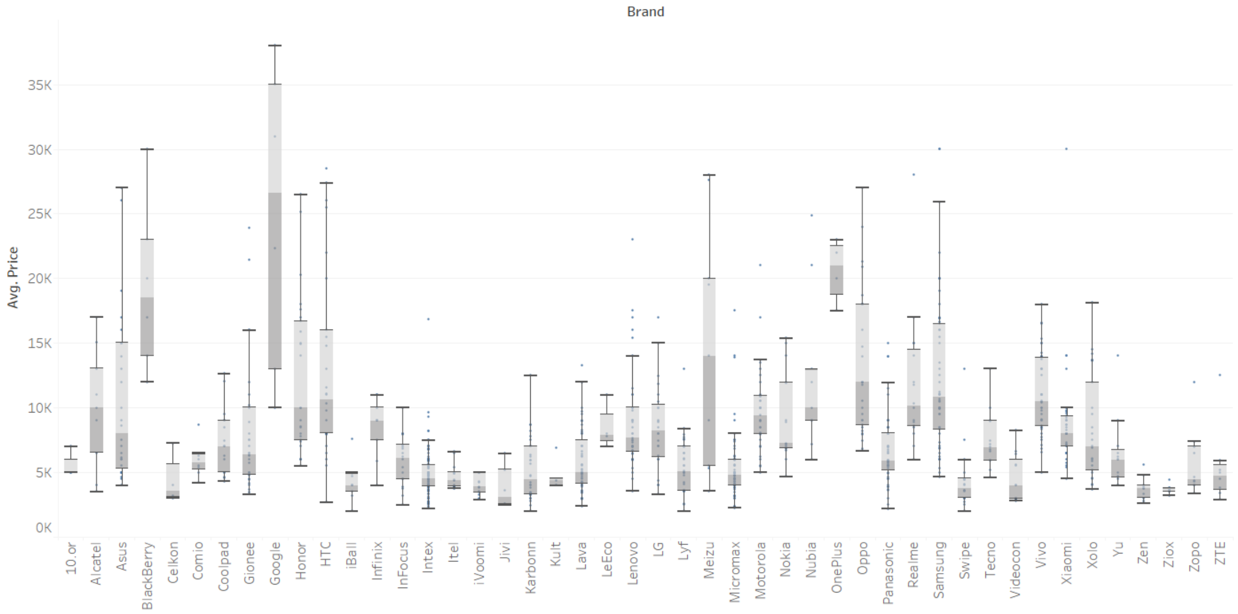
Scatter plot of phone prices



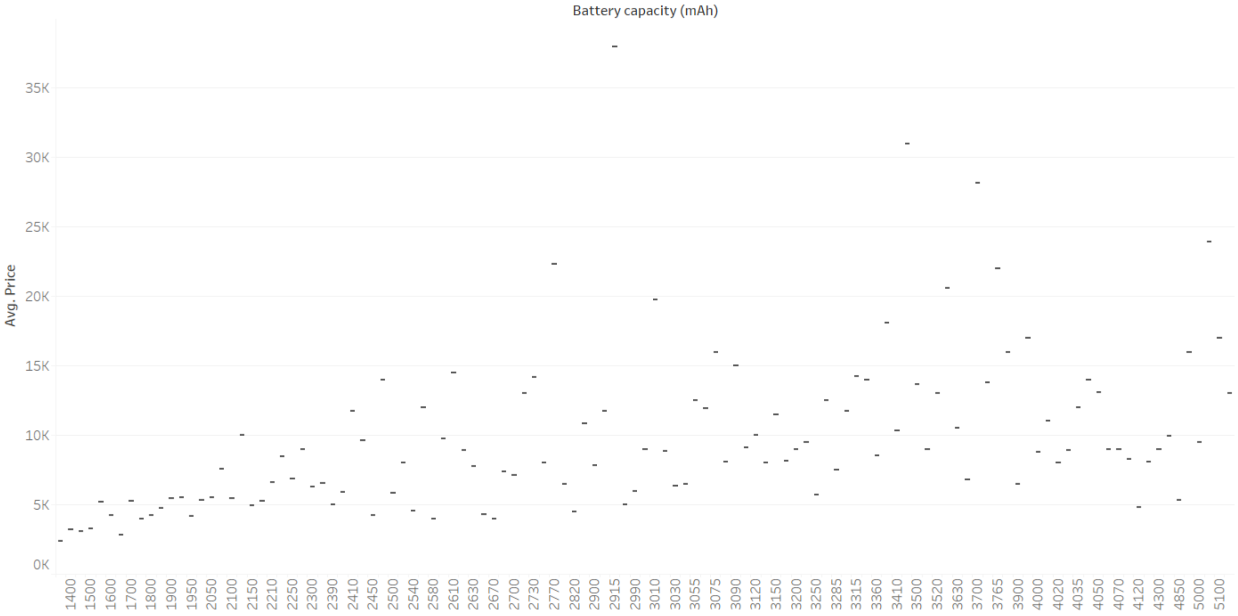
Clearly, the majority of prices are under ₹20,000.

Variation of price with:

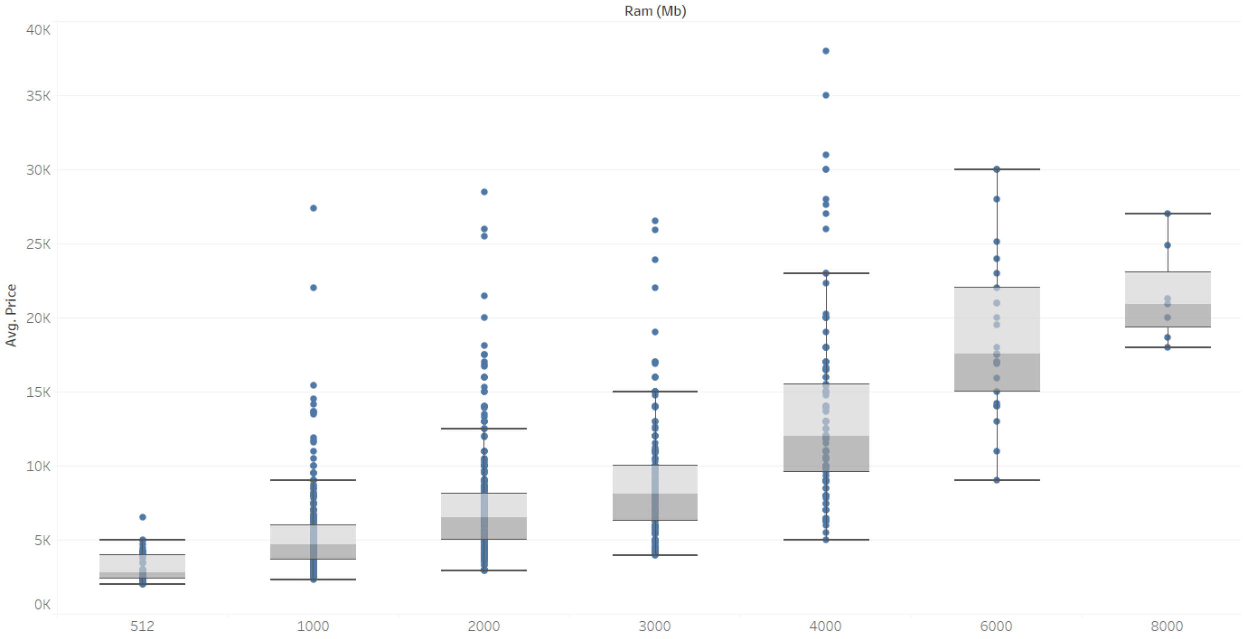
1. Brand



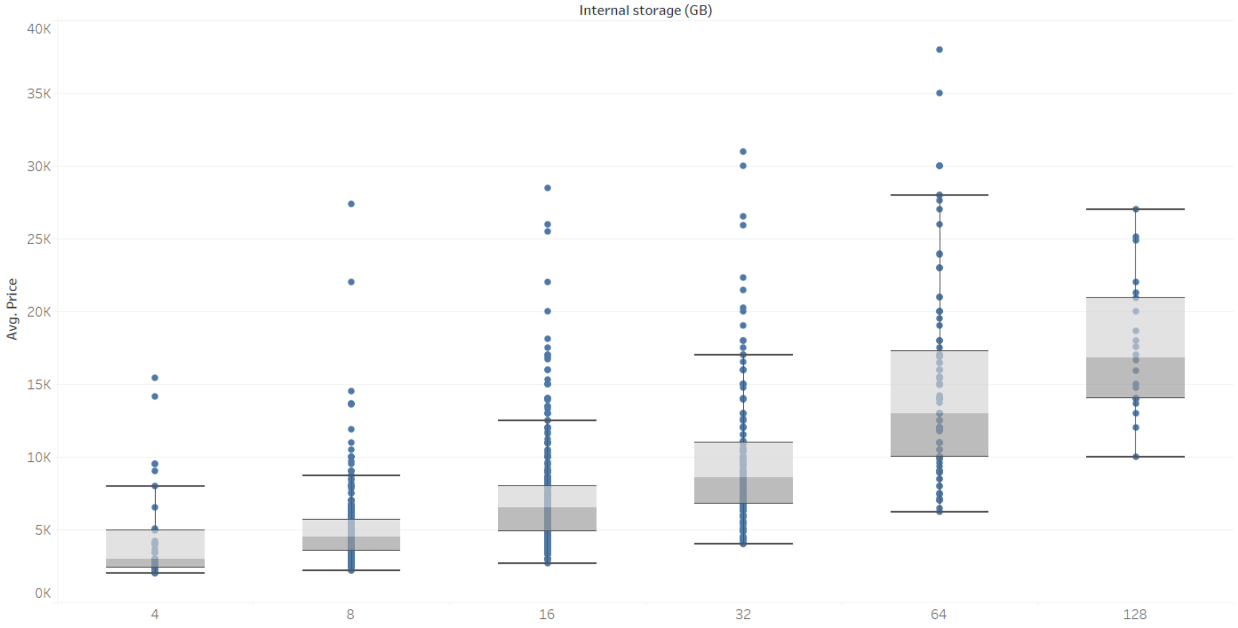
2. Battery Capacity



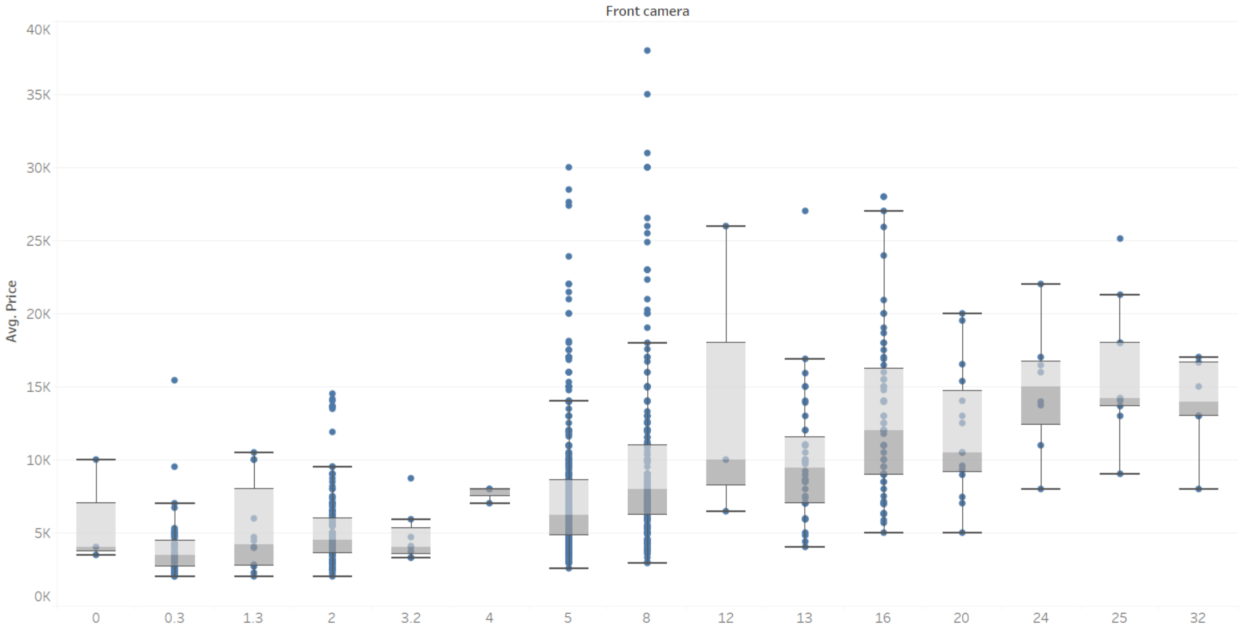
3. RAM


Ram (Mb)
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
512
1000
2000
3000
4000
6000
8000

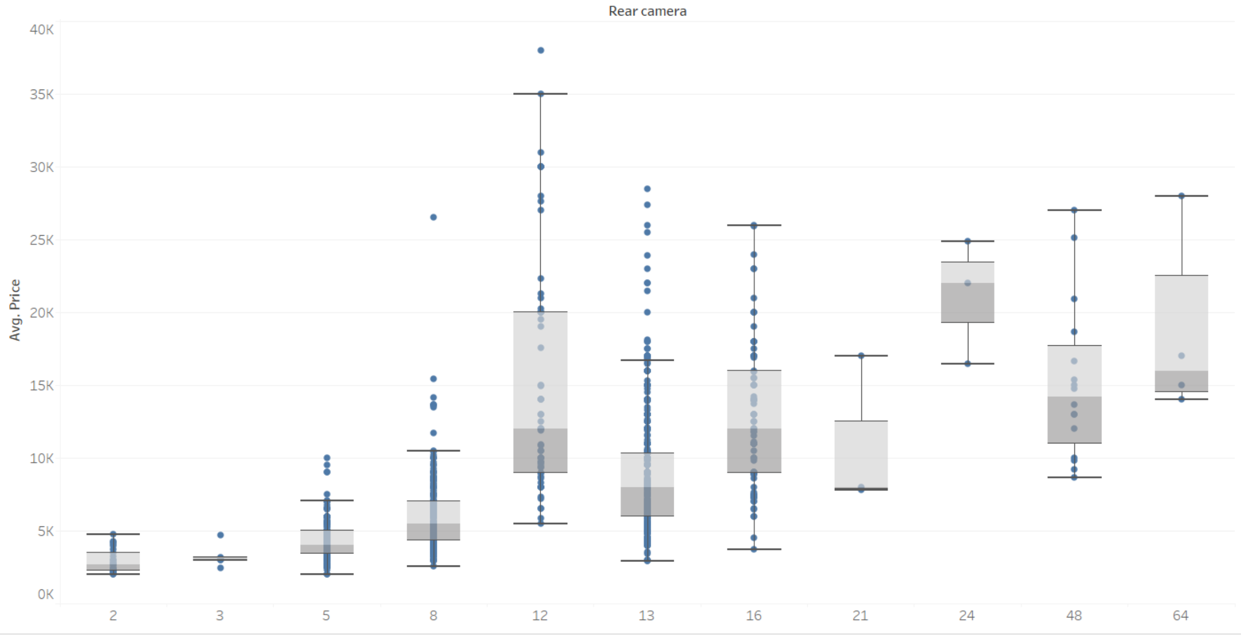
4. ROM


Internal storage (GB)
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
4
8
16
32
64
128

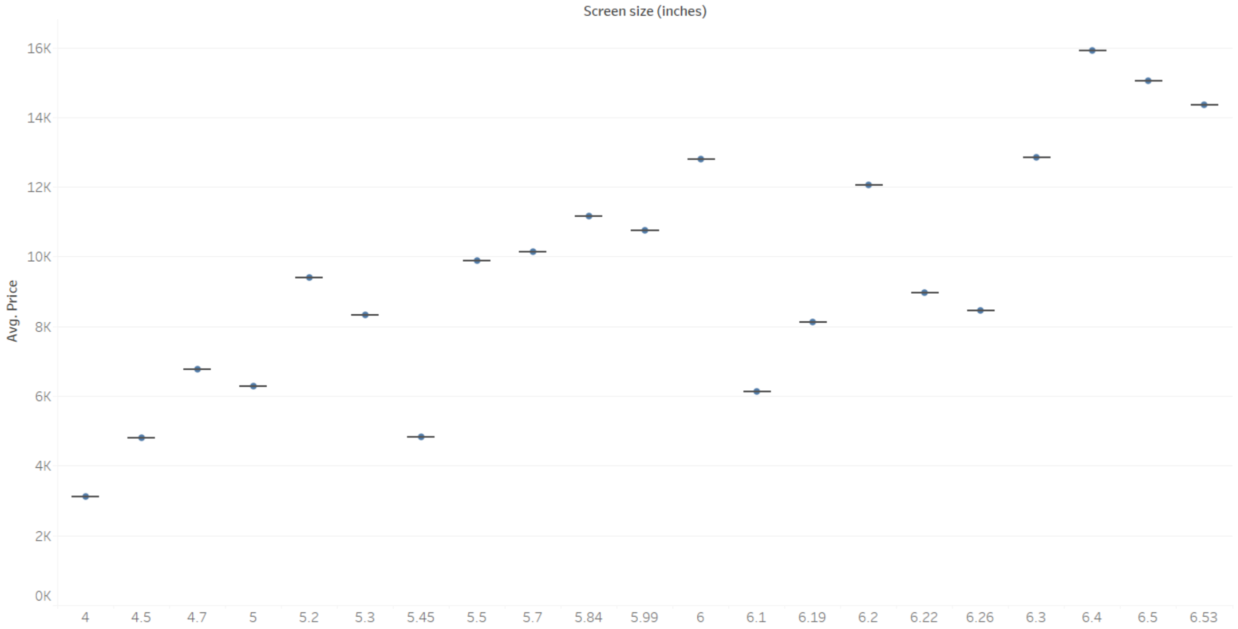
5. Front Camera Quality



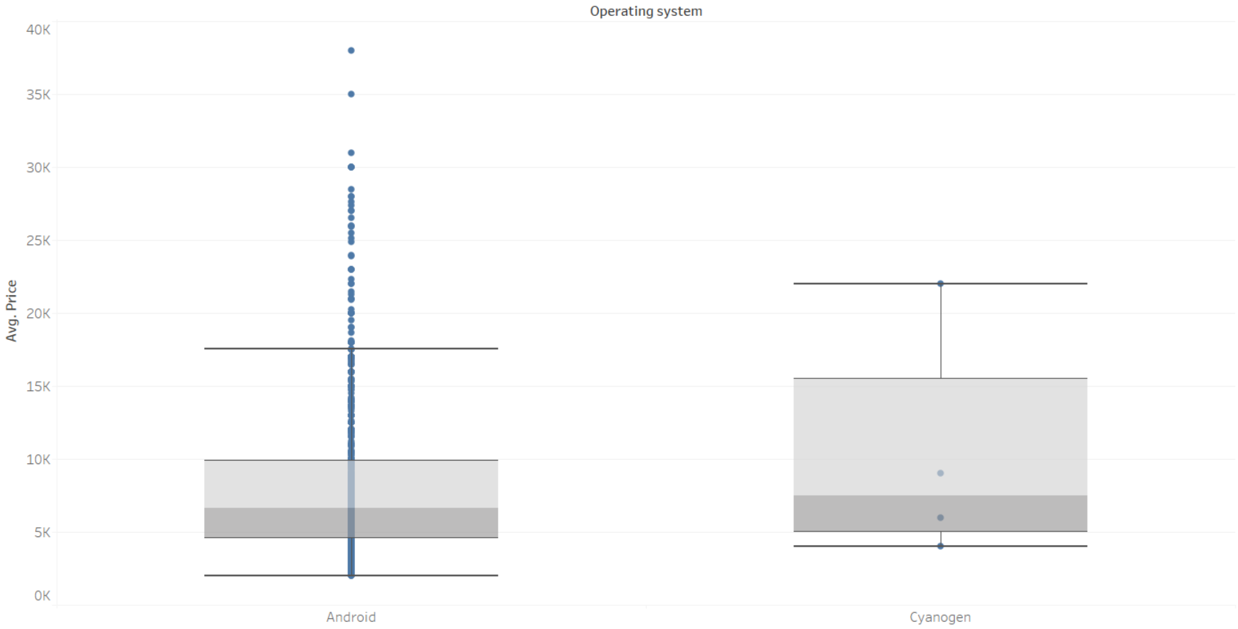
6. Rear Camera Quality



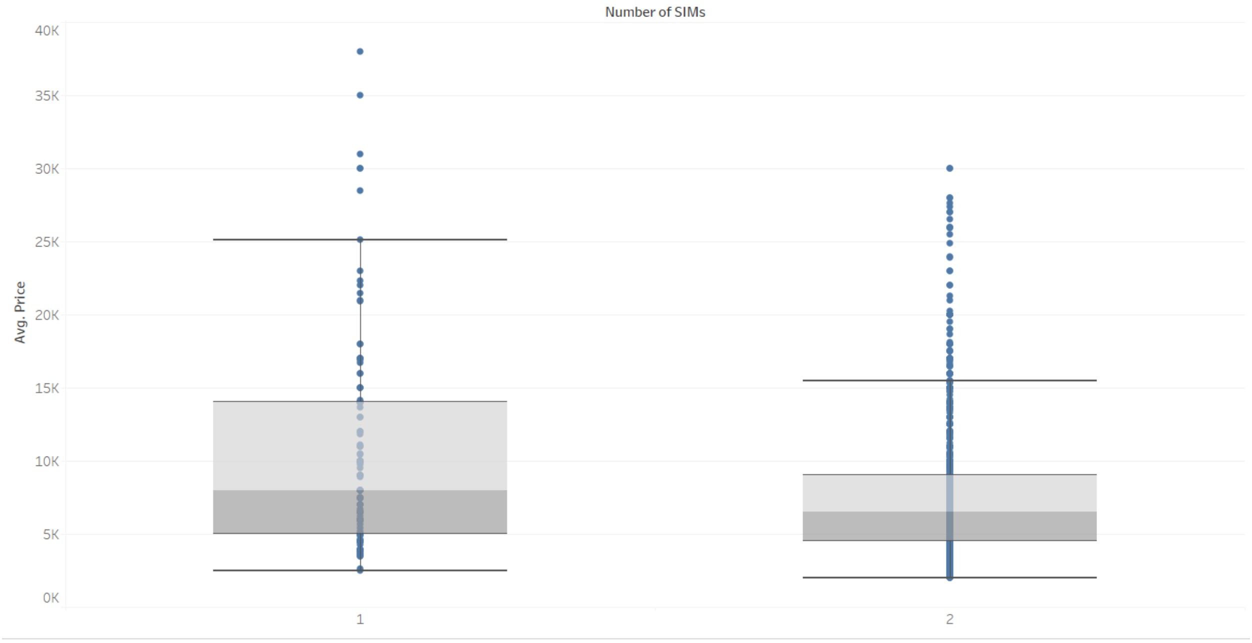
7. Screen Size



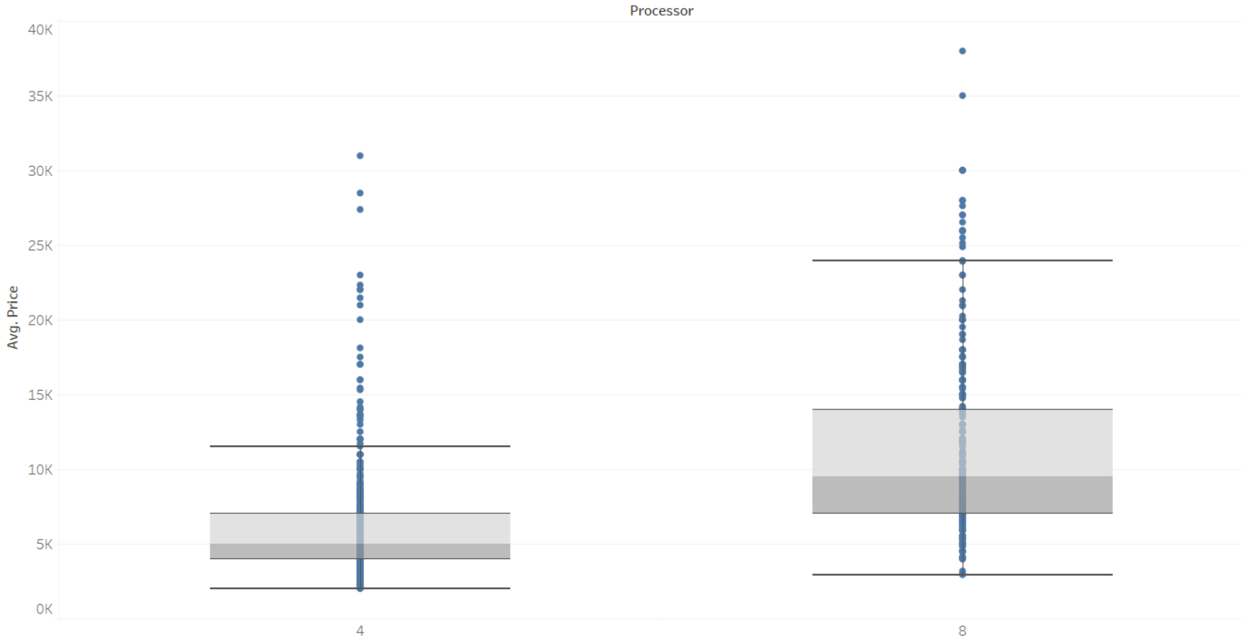
8. Operating System



9. Number of SIMs

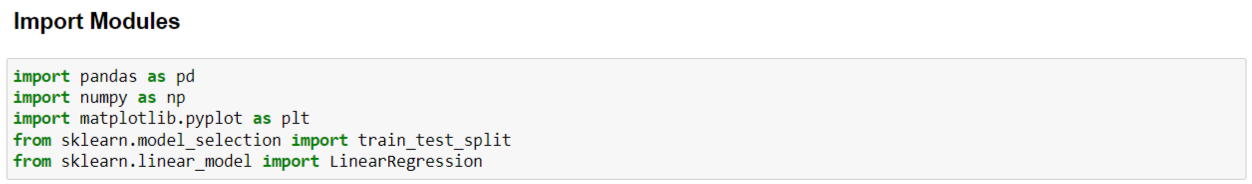


10. Number of Processor Cores



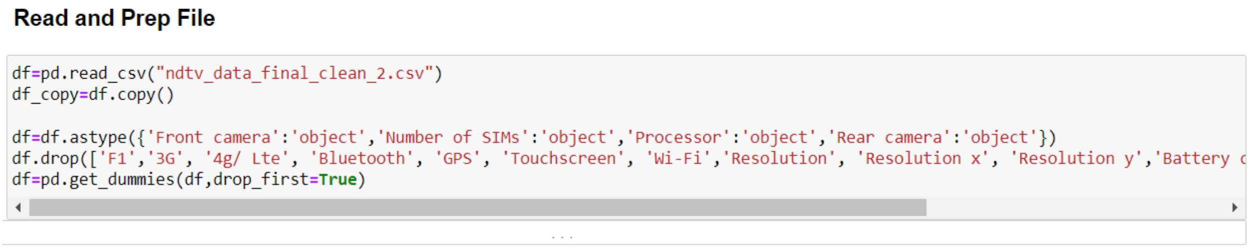
Since the graphs of Battery Capacity, RAM, ROM, and Screen Size follow linear-like trends, they are treated as continuous variables.

ML Modeling



Import Modules

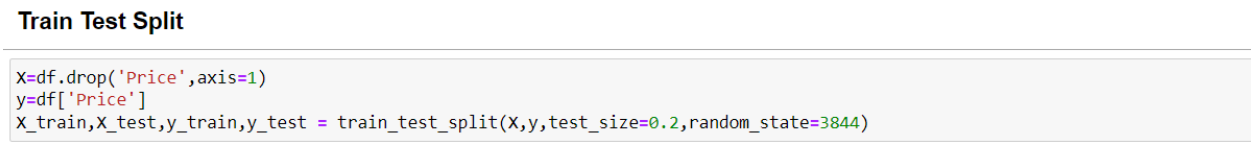
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```



Read and Prep File

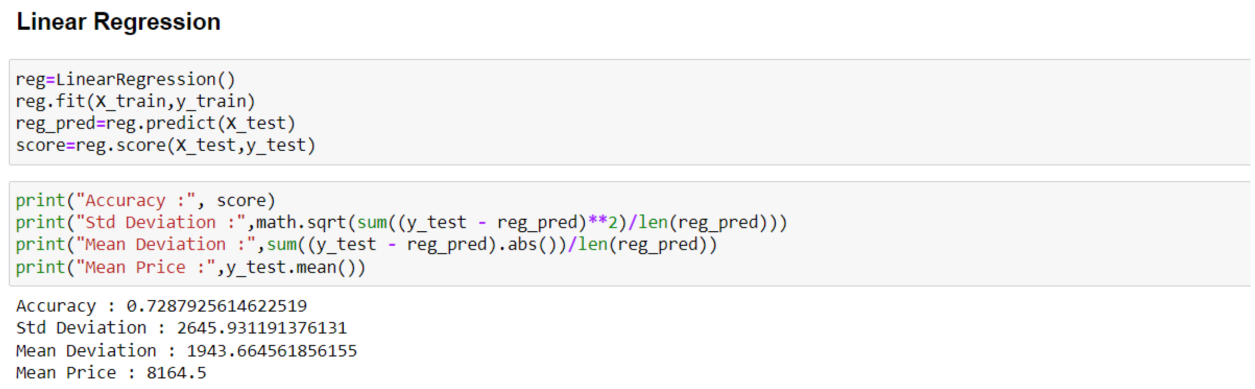
```python
df=pd.read_csv("ndtv_data_final_clean_2.csv")
df_copy=df.copy()

df=df.astype({'Front camera':'object','Number of SIMs':'object','Processor':'object','Rear camera':'object'})
df.drop(['F1','3G', '4g/ Lte', 'Bluetooth', 'GPS', 'Touchscreen', 'Wi-Fi','Resolution', 'Resolution x', 'Resolution y','Battery c
df=pd.get_dummies(df,drop_first=True)
```



Train Test Split

```python
X=df.drop('Price',axis=1)
y=df['Price']
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=3844)
```



Linear Regression

```python
reg=LinearRegression()
reg.fit(X_train,y_train)
reg_pred=reg.predict(X_test)
score=reg.score(X_test,y_test)
```

```python
print("Accuracy :", score)
print("Std Deviation :",math.sqrt(sum((y_test - reg_pred)**2)/len(reg_pred)))
print("Mean Deviation :",sum((y_test - reg_pred).abs())/len(reg_pred))
print("Mean Price :",y_test.mean())
```

```
Accuracy : 0.7287925614622519
Std Deviation : 2645.931191376131
Mean Deviation : 1943.664561856155
Mean Price : 8164.5
```

[GitHub Link to Code]

Conclusion

Phone prices can be estimated based on their specifications by making use of Machine Learning. Linear Regression resulted in a RMSE error margin of ₹2645.9311 and a Mean error margin of ₹1943.6645. Note that the above implementation is done on a small scale and use of more features and data could result in improved accuracy and error margin.