Link: GitHub

Smartphone Price Forecasting Model

Sonali Sharma

Date: 14th September, 2024

Smartphone Price Forecasting Model

Step 1: Prototype Selection

Problem Statement

Mobiles, first invented in 1992 and launched in 1994 by the techno-giant IBM, have become an integral part of the lives of human beings. Today, these technical devices serve a multitude of purposes — calling, video calls, texts, internet, mailing, playing games, taking pictures, shopping etc. Due to these very purposes, the buyers often take many parameters into consideration such as brand, processor, memory size (internal & external), camera, battery backup among others. However, one parameter that is generally not considered is the **price**. As such, the main objective of this report is to introduce a system to cross-validate the price of a mobile phone based on its features.

Market/Customer/Business Need Assessment

Price is the most important side of shopping. Customers are very often interested in knowing the price of the item they wish to buy. Likewise, they are also interested in knowing whether the item is worth the price or not given its features. Hence, the type of service proposed here will enable the common man to have an estimate of the price of a mobile before making a purchase.

Target Specifications

The service will be essential for almost everyone in predicting the mobile price by means of:

- → Brand
- → Front & Rear Camera Megapixels
- → RAM Capacity
- → Internal Memory (ROM)
- → Type of Android
- → 3/4/5 G Support
- → Number of Sim Card Support
- → Battery Support (in mAh)

External Search

- Dataset
- Linear Regression
- Lasso and Ridge Regression

Let's import the dataset and have a look at it!

Import Modules

```
[1] 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, Ridge, Lasso
    from sklearn.metrics import mean_absolute_error
```

Loading the Dataset

```
[10] df=pd.read_csv("ndtv_dataset.csv")
     df.head()
```

	3G	4g/ Lte	Battery capacity (mAh)	Battery capacity (mAh) (bin)	Bluetooth	Brand	F1	Front camera	GPS	Internal storage (GB)	...	Processor	Ram (Mb)	Rear camera	Resolution	Resolution x	Resolution y	Screen size (inches)
0	Yes	Yes	4000	4000	Yes	Realme	1	16.0	Yes	64	...	8	6000	64	1080X2400	1080	2400	6.50
1	Yes	Yes	3765	3500	Yes	Oppo	10	16.0	Yes	64	...	8	6000	16	1080X2340	1080	2340	6.50
2	Yes	Yes	3765	3500	Yes	Realme	11	16.0	Yes	128	...	8	4000	48	1080X2340	1080	2340	6.53
3	Yes	Yes	4045	4000	Yes	Realme	15	25.0	Yes	64	...	8	4000	16	1080X2340	1080	2340	6.30
4	Yes	Yes	4000	4000	Yes	Xiaomi	17	13.0	Yes	64	...	8	4000	48	1080X2340	1080	2340	6.30

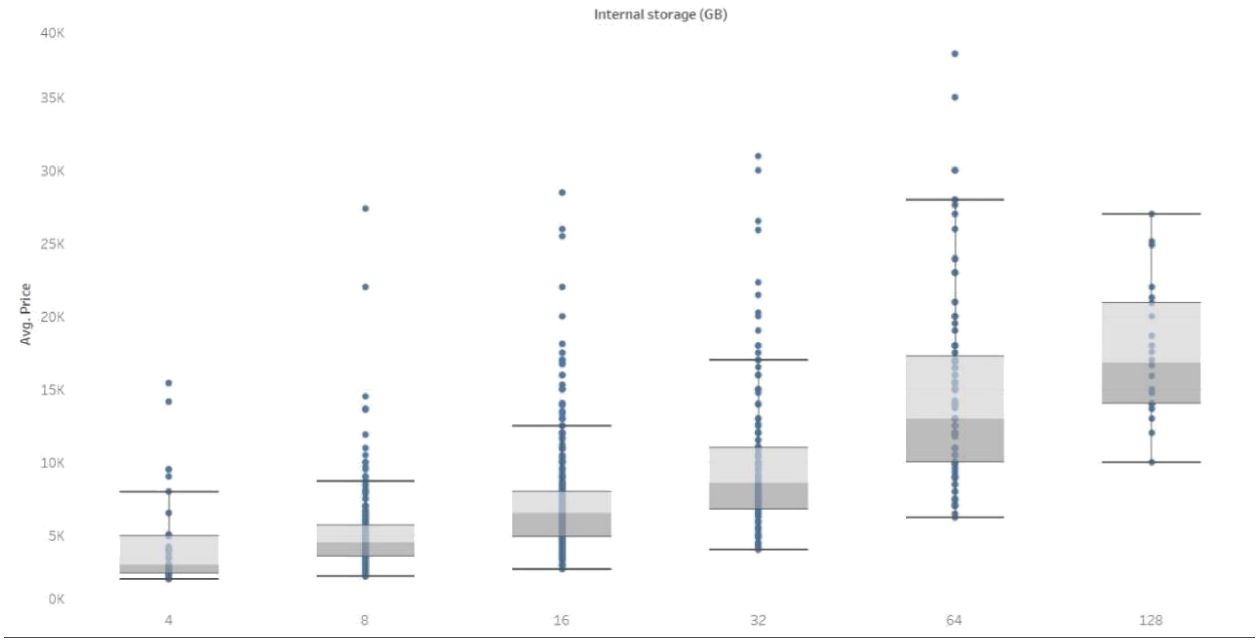
5 rows × 22 columns

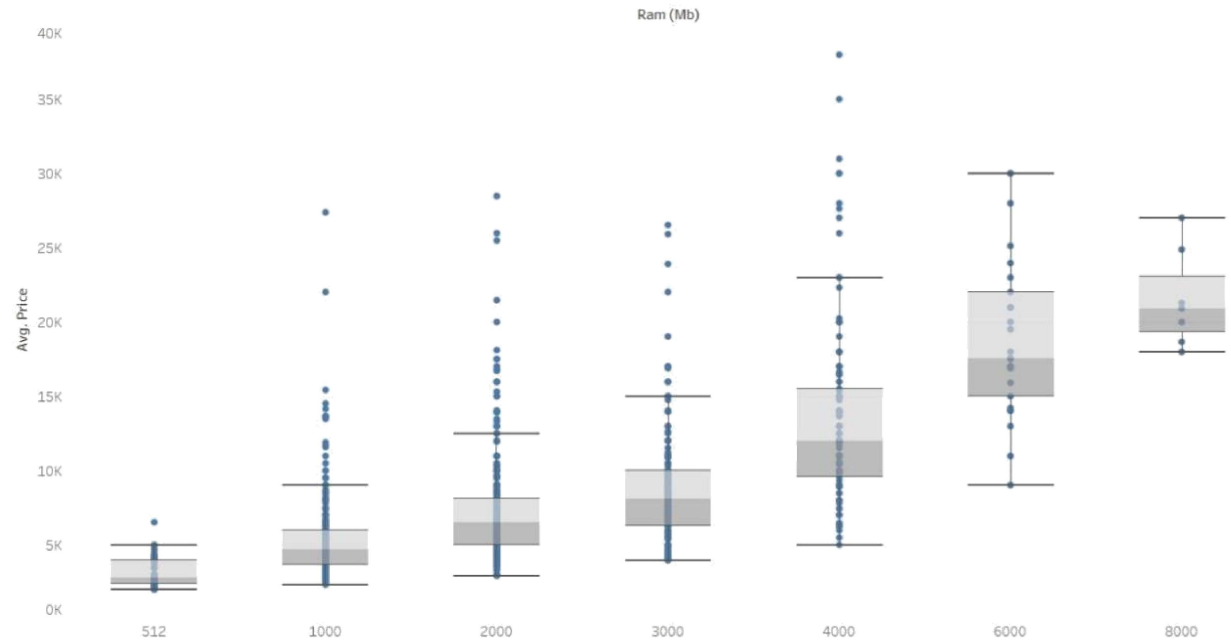
Getting Info of the Dataset

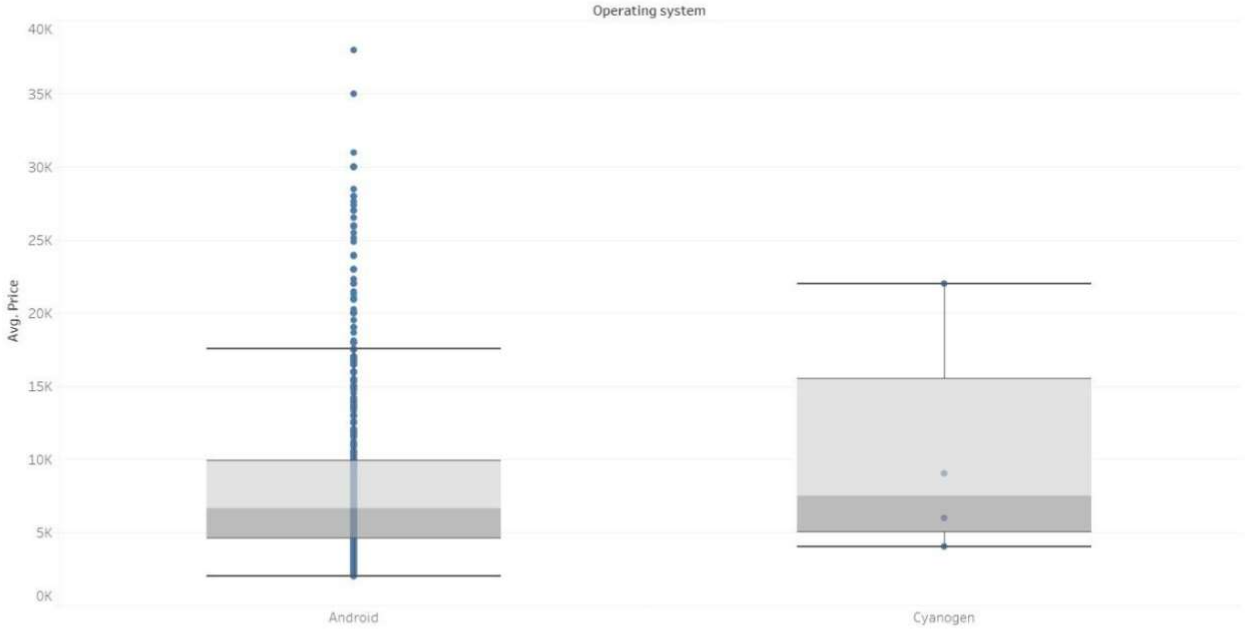
```
[11] df.info()
```

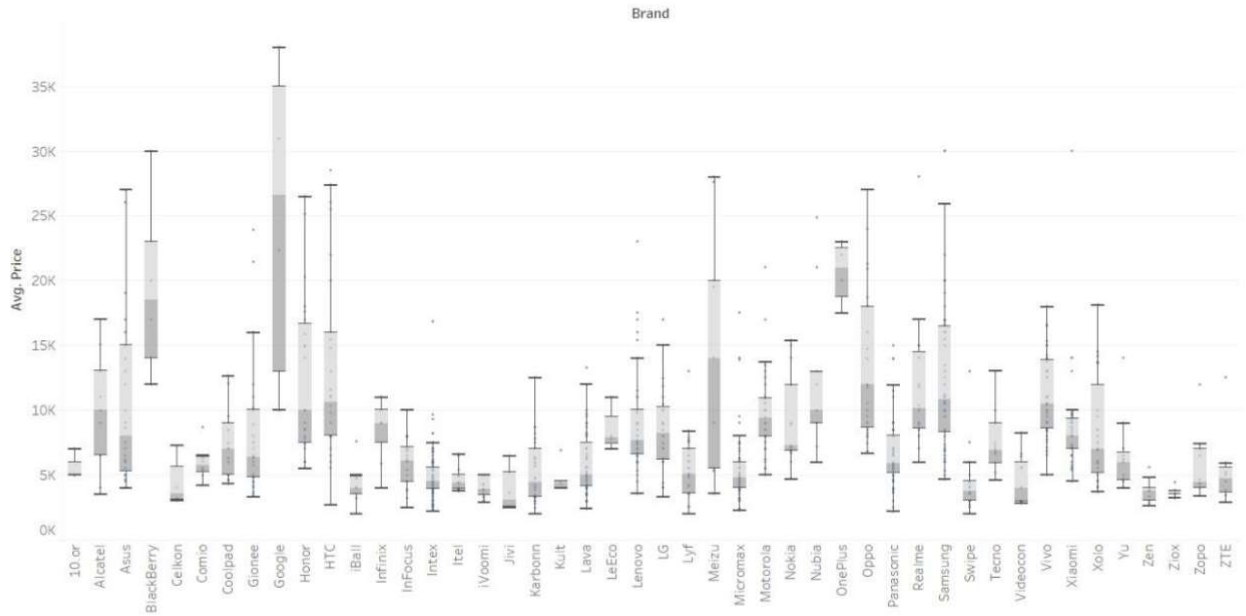
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 988 entries, 0 to 987
Data columns (total 22 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   3G                          988 non-null    object
 1   4g/ Lte                     988 non-null    object
 2   Battery capacity (mAh)      988 non-null    int64
 3   Battery capacity (mAh) (bin)  988 non-null    int64
 4   Bluetooth                   988 non-null    object
 5   Brand                       988 non-null    object
 6   F1                          988 non-null    int64
 7   Front camera                988 non-null    float64
 8   GPS                         988 non-null    object
 9   Internal storage (GB)       988 non-null    int64
 10  Number of SIMs              988 non-null    int64
 11  Operating system            988 non-null    object
 12  Processor                   988 non-null    int64
 13  Ram (Mb)                    988 non-null    int64
 14  Rear camera                 988 non-null    int64
 15  Resolution                  988 non-null    object
 16  Resolution x                988 non-null    int64
 17  Resolution y                988 non-null    int64
 18  Screen size (inches)        988 non-null    float64
 19  Touchscreen                 988 non-null    object
 20  Wi-Fi                       988 non-null    object
 21  Price                       988 non-null    int64
dtypes: float64(2), int64(11), object(9)
memory usage: 169.9+ KB
```

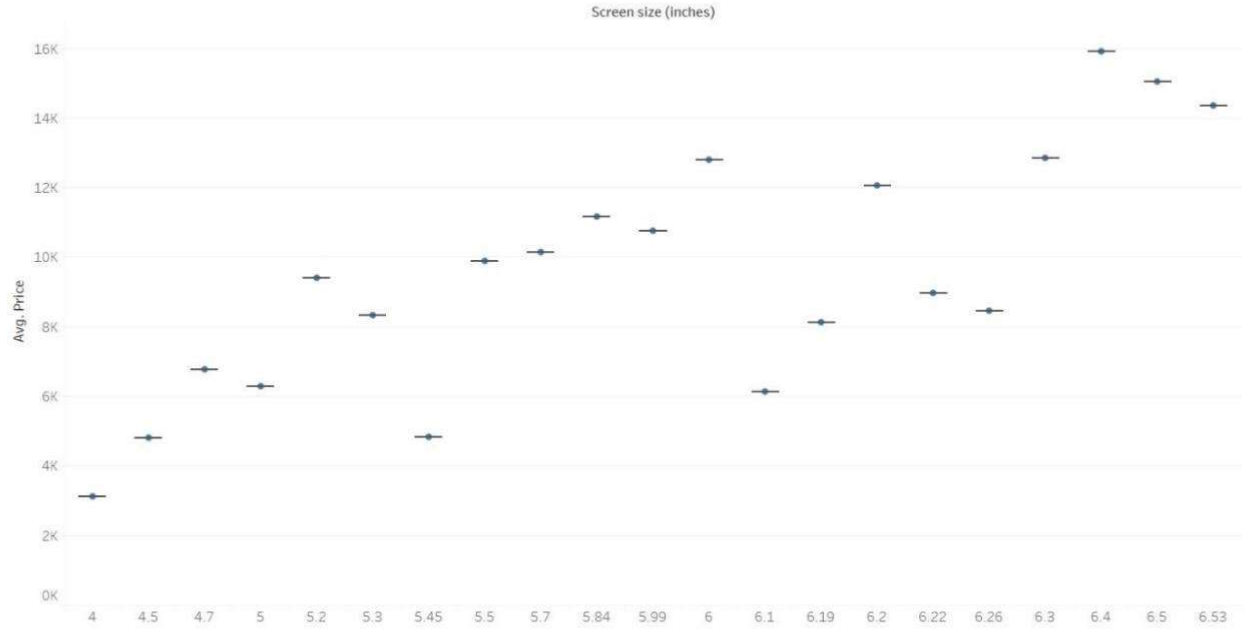
Exploratory Data Analysis (EDA)

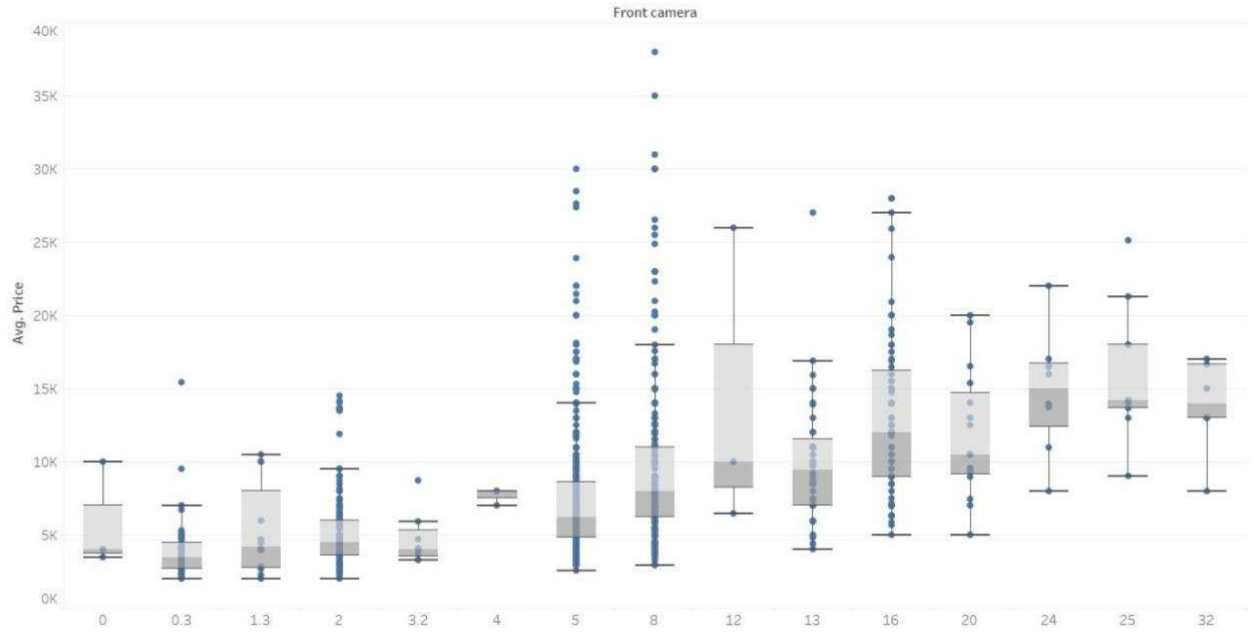


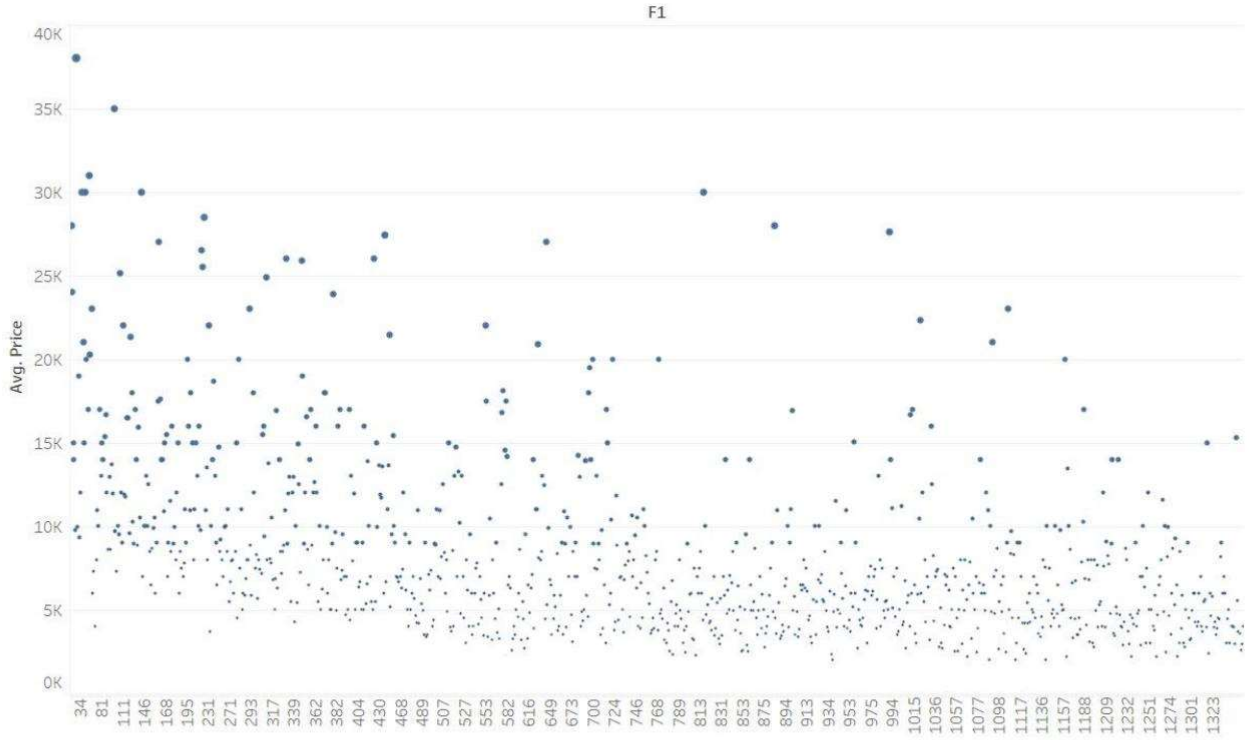


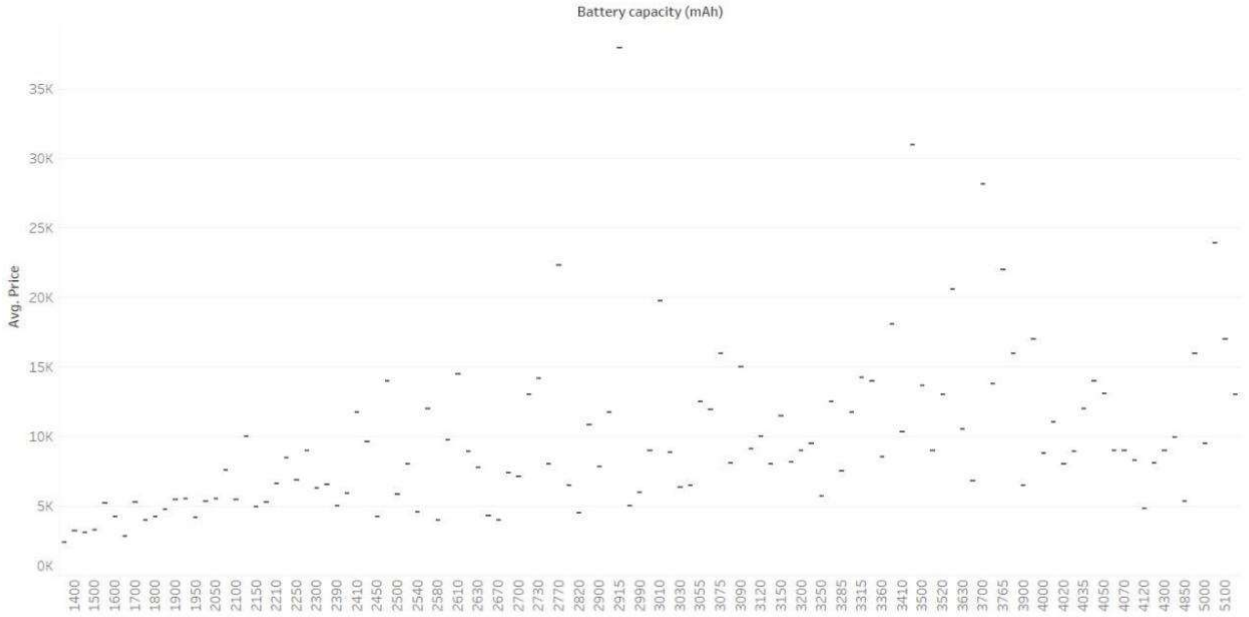

Operating system
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
Android
Cyanogen

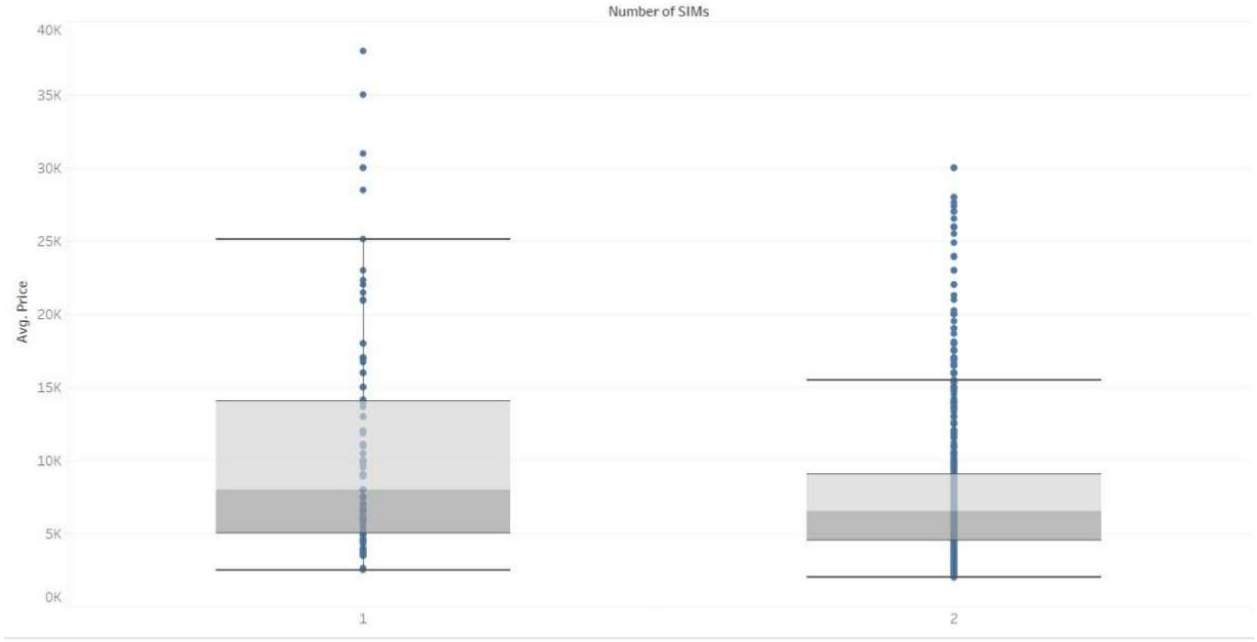

Brand
Avg. Price
35K
30K
25K
20K
15K
10K
5K
0K
10.or
Alcatel
Asus
BlackBerry
Celkon
Comio
Coolpad
Gionee
Google
Honor
HTC
iBall
Infinix
InFocus
Intex
Itel
iVoomi
Jivi
Karbonn
Kult
Lava
LeEco
Lenovo
LG
Lyf
Meizu
Micromax
Motorola
Nokia
Nubia
OnePlus
Oppo
Panasonic
Realme
Samsung
Swipe
Tecno
Videocon
Vivo
Xiaomi
Xolo
Yu
Zen
Ziox
Zopo
ZTE

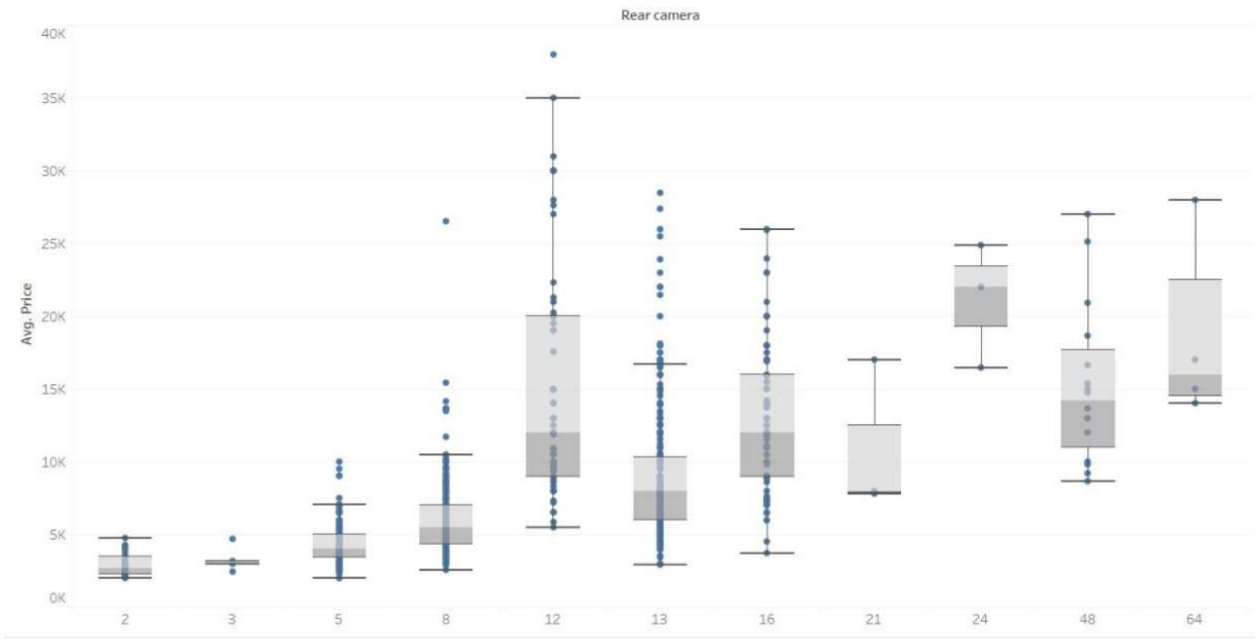

Screen size (inches)
Avg. Price
16K
14K
12K
10K
8K
6K
4K
2K
0K
4
4.5
4.7
5
5.2
5.3
5.45
5.5
5.7
5.84
5.99
6
6.1
6.19
6.2
6.22
6.26
6.3
6.4
6.5
6.53


Front camera
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
0
0.3
1.3
2
3.2
4
5
8
12
13
16
20
24
25
32


F1
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
34
81
111
146
168
195
231
271
293
317
339
362
382
404
430
468
489
507
527
553
582
616
649
673
700
724
746
768
789
813
831
853
875
894
913
934
953
975
994
1015
1036
1057
1077
1098
1117
1136
1157
1188
1209
1232
1251
1274
1301
1323


Battery capacity (mAh)
Avg. Price
35K
30K
25K
20K
15K
10K
5K
0K
1400
1500
1600
1700
1800
1900
1950
2050
2100
2150
2210
2250
2300
2390
2410
2450
2500
2540
2580
2610
2630
2670
2700
2730
2770
2820
2900
2915
2990
3010
3030
3055
3075
3090
3120
3150
3200
3250
3285
3315
3360
3410
3500
3520
3630
3700
3765
3900
4000
4020
4035
4050
4070
4120
4300
4850
5000
5100


Number of SIMs
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
1
2


Rear camera
Avg. Price
40K
35K
30K
25K
20K
15K
10K
5K
0K
2
3
5
8
12
13
16
21
24
48
64

Applicable Regulations

Many mobile manufacturing companies don't allow to scrap data from its official websites which could be a possible hindrance in data collection.

Applicable Constraints

Since the mobile market is always changing, continuous data collection and its updation is extremely necessary as lack of quality data is likely to reduce the accuracy of the model.

Business Opportunity

This way of predicting the price of phones has been floating around the internet but there seem NO service in place to achieve the same. Therefore, there is a greater chance of the service being useful to not only the customers but also the sellers as the service will allow the sellers to assess what the customers are looking for.

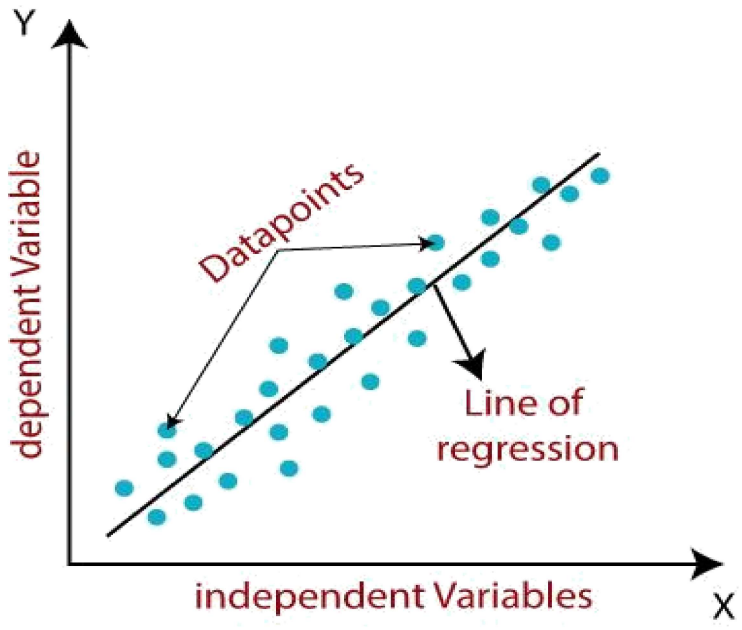
Concept Generation

For successful implementation, the proposed service will require the following algorithms, tools and experts.

Algorithms:

Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. It makes predictions for continuous/numeric variables such as sales, salary, age, product price, etc.



Lasso and Ridge Regression

Lasso and Ridge regressions, aka L1 and L2 regularization respectively, are some of the simple techniques often used to tackle what is known as 'overfitting' (model complexity) which may result from linear regression. Although they both work towards a common goal by penalizing the magnitude of coefficients of features along with minimizing the error (difference between actual and predicted values), there is a slight difference between the penalties they add to the cost function. Lasso adds penalty equal to absolute value of the magnitude of coefficients while Ridge adds penalty equal to square of the magnitude of coefficients.

Tools:

- Python: It's a programming language that will be used for building the service.
- Pandas: Pandas is a library mainly used for handling, manipulating and transforming data.
- Beautiful Soup: It is a web-scraping tool which will be used for fetching data from different sources (webpages).
- Scikit-learn: It is the gold standard library for machine learning which comes with plenty of algorithms to perform different tasks such as regression, classification etc.
- Matplotlib and Seaborn: Both of these libraries are used for visualization purposes.

Team:

Data Scientists who are good at web scraping, data analysis and ML algorithms.

Business Model

There are many applications of the Mobile Phone Price Predictor, but the major ones lie in helping manufacturer sales and ensuring customers' value for money. This product can make use of several business models, but the one we shall be using is:

→ Fee for Service Business Model

Instead of selling products, fee-for-service business models are centered around labor and providing services. A fee-for-service business model may charge by an hourly rate or a fixed cost for a specific agreement. Fee-for-service companies are often specialized, offering insight that may not be common knowledge or may require specific training.

Our product is service based, but is helpful to both mobile phone manufacturers as well as mobile phone purchasers. A manufacturer is responsible for sourcing raw materials and producing finished products by leveraging internal labor, machinery, and equipment. A manufacturer may make custom goods or highly replicated, mass-produced products. A manufacturer can also sell goods to distributors, retailers, or directly to customers. The manufacturers in discussion here shall be Mobile Phone Manufacturers.

The service we provide aims to set a benchmark price for the mobile phone manufacturers, ensuring that they don't over-price or under price their products. The Price Predictor works based on the input specifications of the mobile phone, giving a calculated output of a workable price for any desired mobile. Since the Price Predictor uses the current market trends for prices of mobile phones with similar specifications, the output benchmark price would be in sync with the current market, assuring a fair price and increasing mobile phone sales. As for mobile phone purchasers, i.e., customers, the Price Predictor suggests them to purchase their desired mobile in and around the output price's range. Again, this would require the customer to input the specifications of the mobile they are looking for, or just enter the specifications they would like in their mobile and our product would output the price of such a mobile in the current market.

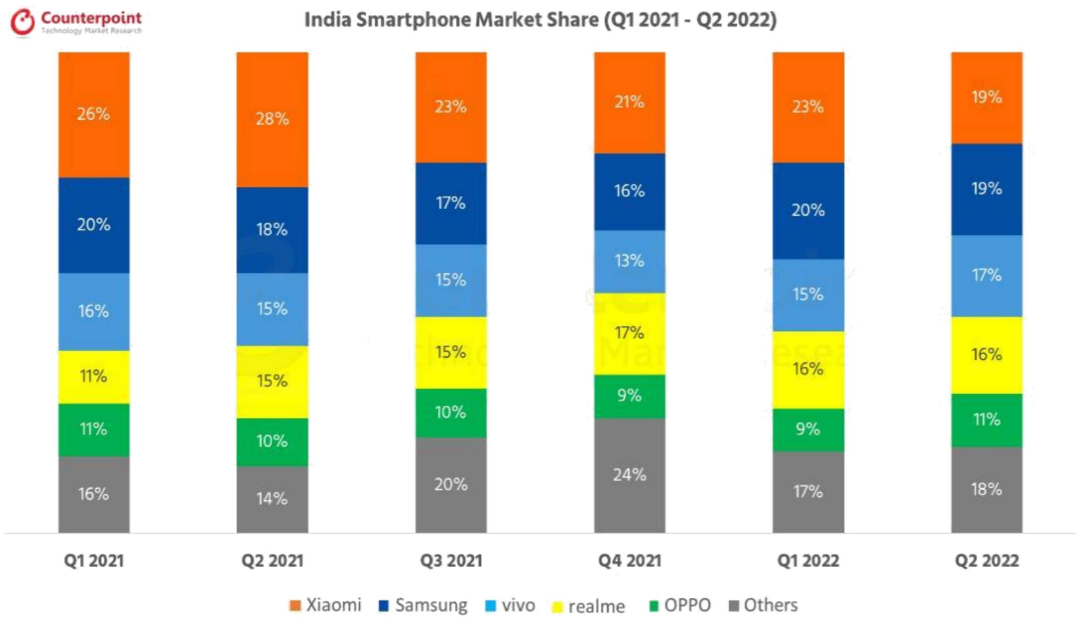
Operational Procedure

Our operating plan consists of several steps targeting manufacturers and customers alike to successfully deploy the Price Predictor in the service market. We also plan to employ different marketing strategies to reach out to mobile phone manufacturers and customers. Charged at a certain rate for each use of the Price Predictor, manufacturers will be allowed to use our product and receive a suggested pricing of their mobile. Depending on the sales of the manufacturer after pricing their mobile based on our Predictor, they may reach out to us for further use of our product. As for customers, we aim to rely on their feedback of our product to improve the algorithm used for prediction since the market constantly changes and we may need to tweak our product from time to time. For this reason, our product shall be made available as a web-based online application since the product would need to be updated every now and then. This is easier than following a downloadable offline form of our product, since that would require users to download different versions of our product from time to time. We plan to make our service available for purchase in different forms. This includes purchase for one time use, purchase by subscription (monthly or annually), and purchase for infinite usage. Rates would differ for the different modes of purchase.

People who use the prediction

- This kind of prediction will help companies estimate price of mobiles to give tough competition to other mobile manufacturer
- Also, it will be useful for Consumers to verify that they are paying best price for a mobile.

Statistics of smartphone market in India



- India's smartphone shipments grew 9% YoY but fell 5% QoQ to reach almost 37 million units in Q2 2022 (April-June)
- Xiaomi led the market in Q2 2022 with a 19% shipment share, closely followed by Samsung.
- India's smartphone installed base crossed 600 million during the quarter.
- 5G smartphones contributed to 29% of overall shipments. Samsung led the 5G smartphone segment with a 25% share, followed by vivo and OnePlus.
- Three out of the top five smartphone models in Q2 2022 were from realme.
- Samsung led the premium smartphone market (>INR 30,000), closely followed by Apple.

Financial Equation

The above diagram indicates how the smartphone companies in India have been growing for the past 2 years. Even with the impact of COVID, the industry has continued to grow. So if we are following the above trend it would be advisable to price our service around Rs.5000.
Once the customer base increases, we can either increase the price or reduce the duration for which our product will be available. Let's assume that the duration of developing the ML model takes about 1 to 3 weeks and the cost for producing the model is the salary of the members the team.

Let there be two ML engineers and one full stack web developer. Let the salary of the ML engineers be 'ml' 18 and the full stack web developer be 'fs'. So, the total cost c = 2*ml + fs.

So, the profit or financial equation will look like this y = 5000*x(t) – (2*ml+fs)

Here x(t) is a function that represents the growth of the customer base and y is the profit.