FeyNN Labs: Project 3
Link: < GitHub >

Mobile Price Prediction System

Contributors

Shubham Navghare, Rohit Rannavre, Rohan Suresh, K. Subhaashini, Chandra SekharPusarla

Date: @@th November, 2022

Mobile Phone Price Prediction System

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
```

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.

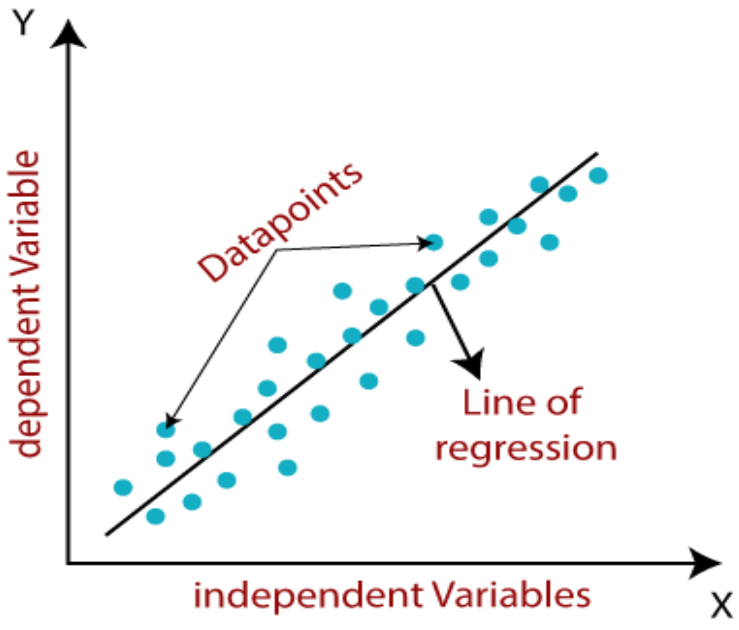
Concept Generation

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.
- → BeautifulSoup: 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.