Shri Vaishnav Vidyapeeth Vishwavidyalaya

DATA SCIENCE USING PYTHON

"MOBILE PHONE PRICE PREDICTION"

TRAINING PROJECT REPORT

SUBMITTED BY:
Yashasvi Mahajan
CS-B
IV Year
(1710DMBCSE01411)

A REPORT OF THREE WEEKS INDUSTRIAL

TRAINING AT

WebTek Labs Pvt. Ltd.

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD

OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING

JULY-AUGUST 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SVIIT,Indore

CANDIDATE'S DECLARATION

I hereby declare that I have undertaken industrial training at "**WEBTEK LABS PVT. LTD.**" during a period from **27 JULY to 14 AUGUST** in partial fulfilment of requirements for the award of degree of B.Tech (COMPUTER SCIENCE & ENGINEERING) at SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY, INDORE. The work which is being presented in the training report submitted to Department of COMPUTER SCIENCE & ENGINEERING at SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY, INDORE is an authentic record of training work.

Yashasvi Mahajan
(1710DMBCSE01411)
CS-B
IV Year

ACKNOWLEDGEMENT

It gives me great pleasure to acknowledge the guidance, assistance and support of Ms. Mousita Dhar in making the Project and this Project report successful, which has been structured under her valued suggestion.

She has helped us to accomplish the challenging task in a very short period of time.

Finally, I express the constant support of our friends, family and professors for inspiring us throughout and encouraging us.

Yashasvi Mahajan

(1) INTRODUCTION

1. PYTHON

About Python:

- Python is a high-level, general-purpose, open source, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale.
- Python was created By Guido van Rossum.
- The Python Software Foundation (PSF) is the organization behind Python.

Python versions:

- First released in 1991.
- Python 2.0 was released on 16 October 2000
- Python 3.0 was released on 3 December 2008

Current Versions:

- 3.6.3
- 2.7.14

Python features:

Some of the features of python include :-

- Easy to understand
- Dynamic
- Object oriented
- Multipurpose
- Strongly typed
- Open Sourced

Python is mainly used in many domains:

- Web Development
- Data Analysis
- Machine Learning

- Internet Of Things
- GUI Development
- Image processing
- Data visualisation
- Game Development

IDLE:

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

2. Anaconda

Anaconda is a open source Distribution for data science and machine learning using python. It includes hundreds of popular data science packages and the conda package and virtual environment manager for Windows, Linux, and MacOS. Conda makes it quick and easy to install, run, and upgrade complex data science and machine learning environments like scikit-learn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Service Machine Learning AMIs and Anaconda for Microsoft on Azure and Windows.

1.3 Packages

1. NumPy

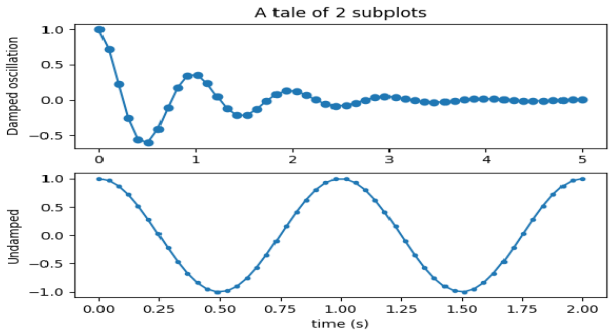
NumPy is the fundamental package for scientific computing with Python. It contains among other things:

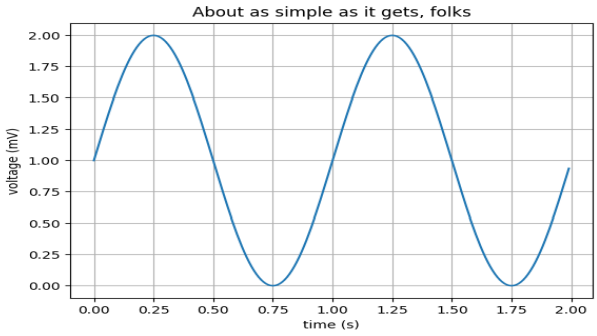
- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

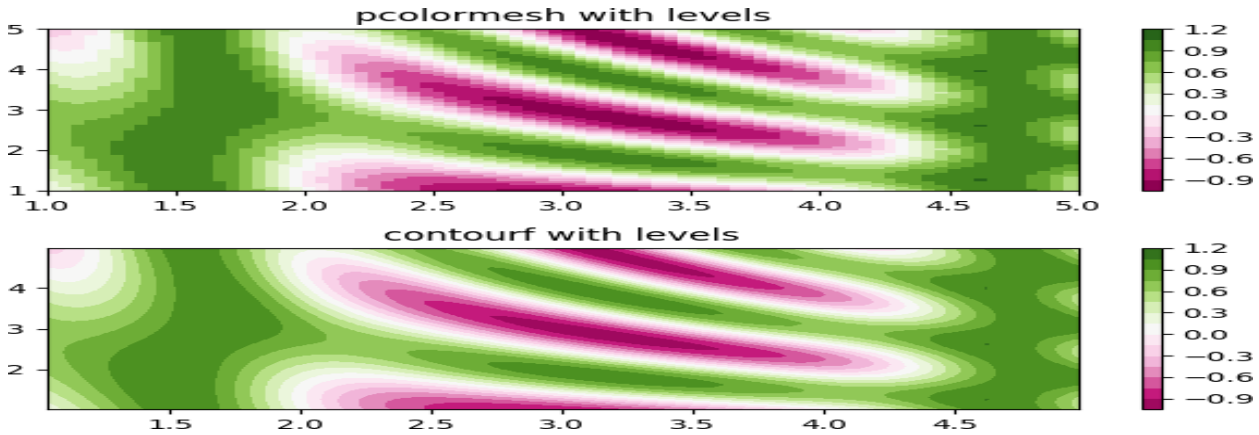
Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy toseamlessly and speedily integrate with a wide variety of databases.

2. Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.







Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

3. Scikit-learn

Scikit-learn provides machine learning libraries for python.Some of the features of Scikit-learn includes:

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

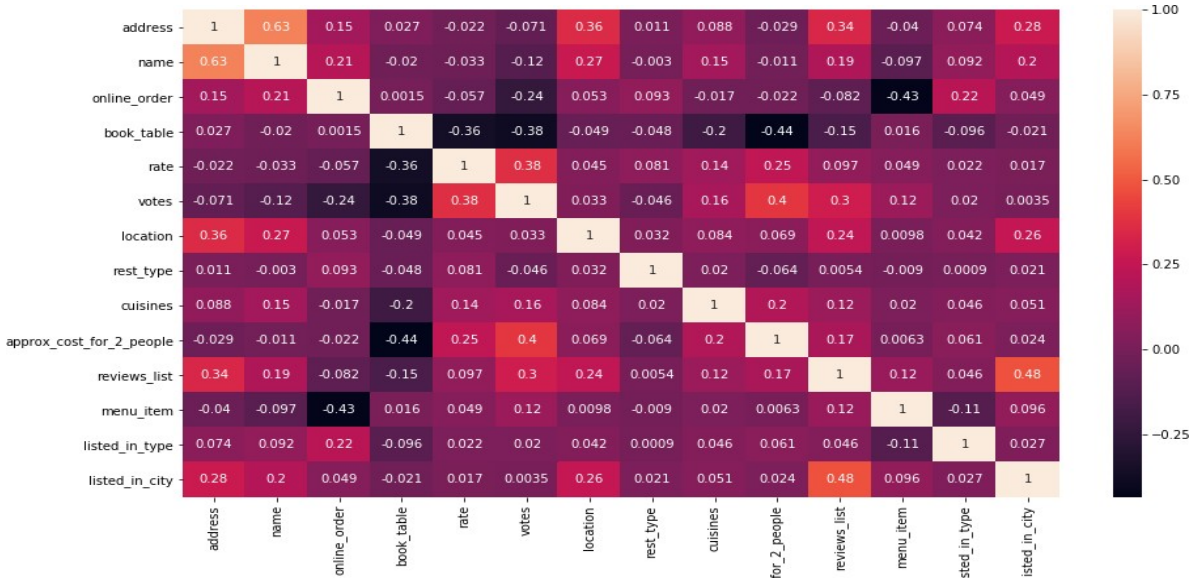
4. Pandas

Pandas is an open source, BSD-licensed library providing high- performance, easy-to-use data structures and data analysis tools for the Python programming language.

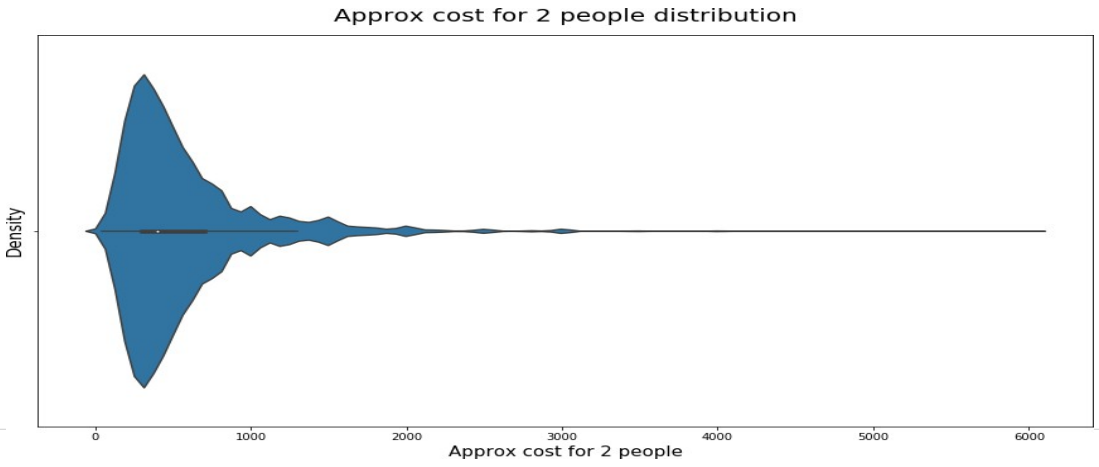
Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables
and time series.

5. Seaborn

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. E.g:-



Heatmap(above) & Violinplot(below)



(2) TRAINING WORK UNDERTAKEN

1. COLLECTING DATA FROM KAGGLE

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowd sourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. On 8 March 2017, Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

2. DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

Turing award winner JiGray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge. When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, business intelligence, predictive modeling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

3. SOURCE CODE & OUTPUT

(I) IMPORT PACKAGES

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression  # y=mx+c
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels
```

(II) LOAD THE DATASET

```
mydataset=pd.read_csv(r"C:\Users\admin\data_science\mobile_price_predict.csv")
```

```
print("Mobile price prediction dataset")
mydataset
```

Mobile price prediction dataset

	battery_power	bluetooth	clock_speed	dual_sim	front_cam	four_g	int_memory	m_dep	mobile_wt	n_cores	...	sc_h	sc_w	talk_time	three_g	touc
0	842	0	2.2	0	1	0	7	0.6	188.0	2	...	9	7	19	0	
1	1021	1	0.5	1	0	1	53	0.7	136.0	3	...	17	3	7	1	
2	563	1	0.5	1	2	1	41	0.9	145.0	5	...	11	2	9	1	
3	615	1	2.5	0	0	0	10	0.8	131.0	6	...	16	8	11	1	
4	1821	1	1.2	0	13	1	44	0.6	141.0	2	...	8	2	15	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1995	794	1	0.5	1	0	1	2	0.8	106.0	6	...	13	4	19	1	
1996	1965	1	2.6	1	0	0	39	0.2	187.0	4	...	11	10	16	1	
1997	1911	0	0.9	1	1	1	36	0.7	108.0	8	...	9	1	5	1	
1998	1512	0	0.9	0	4	1	46	0.1	145.0	5	...	18	10	19	1	
1999	510	1	2.0	1	5	1	45	0.9	168.0	6	...	19	4	2	1	

2000 rows × 24 columns

(III) DATA PRE-PROCESSING

è Counting missing values for different columns

```
mydataset.isnull().sum()
```

```
To find null values in columns

battery_power      0
bluetooth          0
clock_speed       14
dual_sim           0
front_cam          0
four_g             0
int_memory         0
m_dep              0
mobile_wt          2
n_cores            0
primary_Cam        0
px_height         54
px_width           0
ram               17
sc_h               0
sc_w               0
talk_time          0
three_g            0
touch_screen       0
wifi              33
price_range        0
default_app        0
imei_num         606
Price              0
dtype: int64
```

è Information on original price prediction data

```
print("To find information of columns")
mydataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1998 entries, 0 to 1999
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  1998 non-null   int64
 1   bluetooth      1998 non-null   int64
 2   int_memory     1998 non-null   int64
 3   mobile_wt      1998 non-null   float64
 4   px_height      1998 non-null   float64
 5   px_width       1998 non-null   int64
 6   ram            1998 non-null   float64
 7   sc_h           1998 non-null   int64
 8   sc_w           1998 non-null   int64
 9   three_g        1998 non-null   int64
 10  wifi           1998 non-null   float64
 11  Price          1998 non-null   int64
dtypes: float64(4), int64(8)
memory usage: 202.9 KB
```

è Dropping columns

```
print("The imei number column has most of the values as null. So dropping that column")
mydataset.drop(['imei_num','price_range'],axis=1,inplace=True)
```

```
The imei number column has most of the values as null. So dropping that column
```

```
mydataset.shape
```

```
(2000, 22)
```

```
print("The number of default apps does not affect price of a mobile phone. Therefore deleting this column as well")
mydataset.drop(['default_app'],axis=1,inplace=True)
```

```
The number of default apps does not affect price of a mobile phone. Therefore deleting this column as well
```

```
mydataset.shape
```

```
(2000, 21)
```

```
print("Mobile weight column has only 2 values missing. Therefore deleting those 2 rows itself.")
mydataset=mydataset.drop([29],axis=0)
mydataset=mydataset.drop([282],axis=0)
```

```
Mobile weight column has only 2 values missing. Therefore deleting those 2 rows itself.
```

è Imputing Values

```
print("Imputing missing values columns by using mean()")
mydataset['px_height'].fillna(value=mydataset['px_height'].mean(), axis=0, inplace=True)
mydataset['ram'].fillna(value=mydataset['ram'].mean(), axis=0, inplace=True)
mydataset['wifi'].fillna(value=mydataset['wifi'].mean(), axis=0, inplace=True)
mydataset['clock_speed'].fillna(value=mydataset['clock_speed'].mean(), axis=0, inplace=True)
```

```
Imputing missing values columns by using mean()
```

èFinding Correlation in data

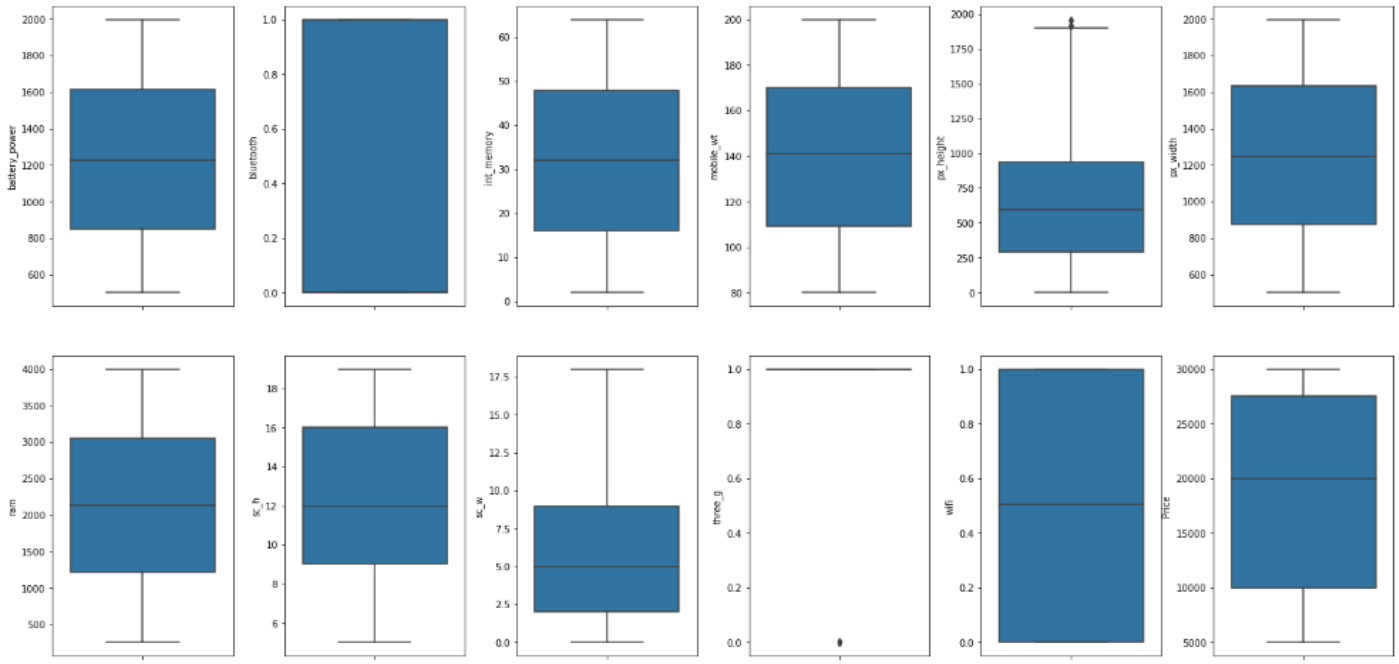
```
print("Finding correlation between data")
mydataset.corr()
```

	battery_power	bluetooth	clock_speed	dual_sim	front_cam	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width
battery_power	1.000000	0.010136	0.013253	-0.041649	0.032494	0.014566	-0.005105	0.033957	0.001438	-0.030405	...	0.024183	-0.008779
bluetooth	0.010136	1.000000	0.021610	0.035244	0.002900	0.012455	0.040303	0.003880	-0.008638	0.035746	...	0.000512	-0.041827
clock_speed	0.013253	0.021610	1.000000	-0.001847	-0.000338	-0.044970	0.009973	-0.012774	0.012452	-0.006881	...	-0.017793	-0.011757
dual_sim	-0.041649	0.035244	-0.001847	1.000000	-0.028749	0.003146	-0.015387	-0.022337	-0.007271	-0.024018	...	-0.021971	0.014598
front_cam	0.032494	0.002900	-0.000338	-0.028749	1.000000	-0.017249	-0.029902	-0.001842	0.023000	-0.013919	...	-0.005038	-0.005476
four_g	0.014566	0.012455	-0.044970	0.003146	-0.017249	1.000000	0.007800	-0.002006	-0.016496	-0.030140	...	-0.006182	0.007199
int_memory	-0.005105	0.040303	0.009973	-0.015387	-0.029902	0.007800	1.000000	0.006785	-0.034788	-0.028925	...	0.016527	-0.008670
m_dep	0.033957	0.003880	-0.012774	-0.022337	-0.001842	-0.002006	0.006785	1.000000	0.022092	-0.003465	...	0.025593	0.023575
mobile_wt	0.001438	-0.008638	0.012452	-0.007271	0.023000	-0.016496	-0.034788	0.022092	1.000000	-0.020117	...	-0.005986	-0.000406
n_cores	-0.030405	0.035746	-0.006881	-0.024018	-0.013919	-0.030140	-0.028925	-0.003465	-0.020117	1.000000	...	-0.004706	0.024183
primary_Cam	0.031129	-0.010286	-0.003722	-0.017307	0.644672	-0.005938	-0.033546	0.026201	0.019128	-0.001239	...	-0.016795	0.004153
px_height	0.024183	0.000512	-0.017793	-0.021971	-0.005038	-0.006182	0.016527	0.025593	-0.005986	-0.004706	...	1.000000	0.507489
px_width	-0.008779	-0.041827	-0.011757	0.014598	-0.005476	0.007199	-0.008670	0.023575	-0.000406	0.024183	...	0.507489	1.000000
ram	-0.001813	0.027162	0.010033	0.036218	0.014542	0.012553	0.039893	-0.007255	-0.000952	0.003541	...	-0.018211	-0.001658
sc_h	-0.031117	-0.003795	-0.028559	-0.011189	-0.011895	0.026396	0.036827	-0.025373	-0.035209	-0.001170	...	0.055708	0.021170

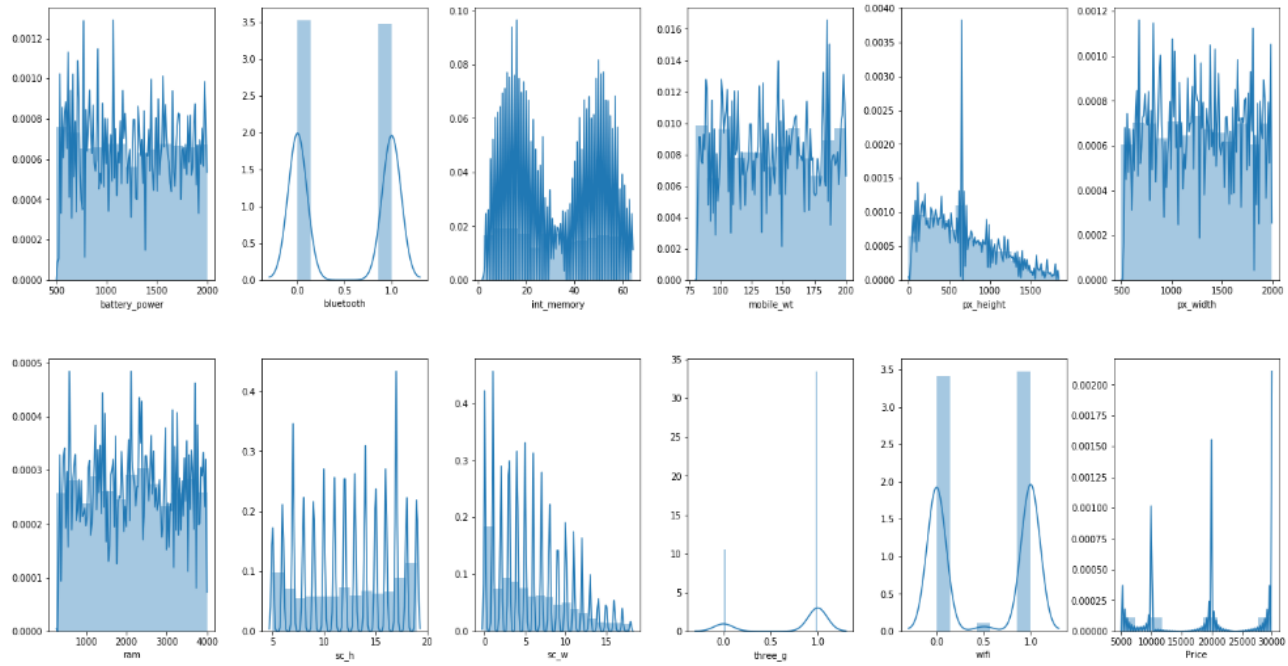
(IV) DATA VISUALISATION

BOXPLOT

Finding outliers by using boxplot



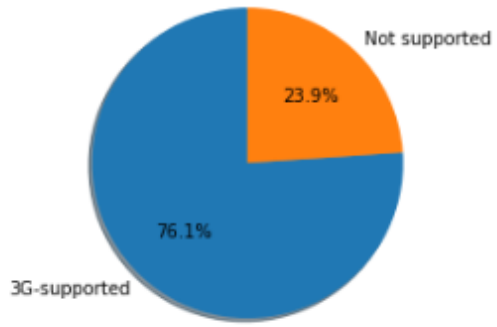
DISTPLOT



HEATMAP

PIECHART

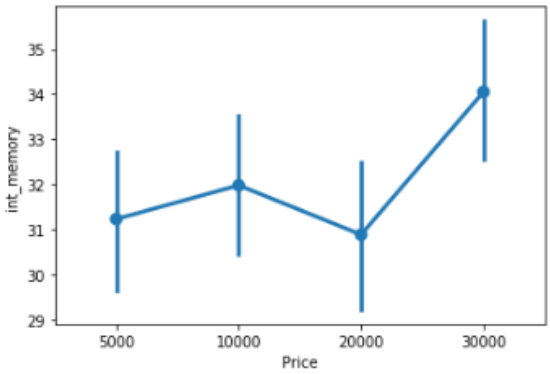
```
3G supported specs
```



POINTPLOT

```
Internal memory vs Price

<matplotlib.axes._subplots.AxesSubplot at 0x97b4472748>
```



(V) PREDICTION

DATA

```
ds.head()
```

	battery_power	bluetooth	int_memory	mobile_wt	px_width	ram	sc_w	three_g	wifi	Price
0	842	0	7	188.0	756	2549.0	7	0	1.0	10000
1	1021	1	53	136.0	1988	2631.0	3	1	0.0	20000
2	563	1	41	145.0	1716	2603.0	2	1	0.0	20000
3	615	1	10	131.0	1786	2769.0	8	1	0.0	20000
4	1821	1	44	141.0	1212	1411.0	2	1	0.0	10000

Prediction by KNN Classifier

Accuracy Score: 67.33%

```
print("Finding accuracy by KNN Classifier")
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train,y_train)
knn.score(X_test,y_test)
```

```
Finding accuracy by KNN Classifier

0.6733668341708543
```

Prediction by Linear Regression

Accuracy Score : 87.8%

```
r2_score(y_test,y_pred) #unknown, or prediction accuray
```

```
0.8780490444745961
```

```
lr.score(X_train,y_train) # known
```

```
0.8791713332862752
```

The accuracy by Linear Regression Classifier is more i.e. 87.8% whereas the accuracy by KNN Classifier is less i.e. 67.3%. Therefore this problem is a linear regression problem.

(3) CONCLUSION

- The main goal of this project is to **predict the price of mobile phone** given its features so that you can compare the actual price of mobile phone with the price being offered to you by the supplier.
- Another goal was to highlight the importance of features of mobile phone which are the important factors in deciding the price of mobile phone.
- We do that by finding correlation between features and finding major features determining the rate of mobile phone and then predicting the price as per the important features.
- Predicting the price of mobile phone is going to help people to get an idea about the price of mobile phone comprising of the features which he/she is looking for.
- She/He can do it just by mentioning the features they need.
- Price determines how well the features are enabled.
- This project involved two supervised learning algorithms i.e. **Linear Regression** and **K Nearest Neighbors**. Both of them have different accuracy score. I got most score in Linear Regression but it may vary depending upon datasets.
- I concluded that the prediction of price was most accurate using Linear **Regression**.

(4) REFERENCES

https://www.kaggle.com/ https://www.python.org/
https://anaconda.org/anaconda/python/
http://www.numpy.org/ https://
matplotlib.org/
http://scikit-learn.org/
https://pandas.pydata.org/