AN INTERNSHIP REPORT

ON

EXPLORATORY DATA ANALYSIS ON SMARTPHONES SALES FROM FLIPKART USING PYTHON

B.E. - III SEMESTER

in

INFORMATION TECHNOLOGY

(ARTIFICIAL INTELLIGENCE & DATA SCIENCE)

By

T. Mayukha Mohan (1601-21-771-021)
&
M. Gopi Prashanth Raju (1601-21-771-051)

Under the guidance

of

Mrs. K.H. Vijaya Kumari,

Assistant Professor

&

Dr. Ramu Kuchipudi

Associate Professor

DEPARTMENT OF INFORMATION TECHNOLOGY,

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)

(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)

GANDIPET, HYDERABAD – 500 075

Website: www.cbit.ac.in

2022-2023

CERTIFICATE

This is to certify that the project work entitled "**EXPLORATORY DATA ANALYSIS ON SMARTPHONES SALES FROM FLIPKART USING PYTHON**", submitted to **Chaitanya Bharathi Institute of Technology** during the academic year 2022-23 is a record of the original work carried out by **T. Mayukha Mohan (1601-21-771-021) & M. Gopi Prashanth Raju (1601-21-771-051)** during the period of study of 2021-25 in Department of IT, CBIT, Hyderabad, under our supervision and guidance.

Mentors

Dr. Ramu Kuchipudi
Associate Professor, Dept. of IT,
CBIT, Hyderabad.

Mrs. K.H. Vijaya Kumari
Assistant Professor, Dept. of IT,
CBIT, Hyderabad.

Head of the Department

Dr. K. Radhika
Professor, Dept. of IT,
CBIT, Hyderabad.

ACKNOWLEDGEMENTS

It is our privilege to acknowledge with a deep sense of gratitude and devotion for the keen personal interest and invaluable guidance rendered by our mentors, **Mrs. K.H. Vijaya Kumari** and **Dr. Ramu Kuchipudi,** and our project guide **Sri Saketh Kallepu**.

Our respects and regards to **Dr. K. Radhika**, Professor, Department of Information Technology, Chaitanya Bharathi Institute of Technology, for her invaluable suggestions that helped us successfully complete the project.

We are grateful to our Principal **Dr. P. Ravinder Reddy**, Chaitanya Bharathi Institute of Technology, for his cooperation and encouragement.

Finally, we also thank all the faculty of the Dept. of IT, CBIT, our friends, and all our family members who, with their valuable suggestions and support, directly or indirectly helped us complete this project.

ABSTRACT

There is huge competition among smartphone brands in India, where you can get the latest technology in a smartphone at half the price of other premium phones. We analyzed the prices and specifications of smartphones available in India with Flipkart data. Through this project, we make some graphical inferences on various smartphones available on Flipkart in the form of live data. We then fill in some missing values in the dataset. We then move to find insights into the data and then proceed to clean the data. We then tried to visualize the data using the various Python libraries which were imported. We can use Selenium to scrape live data from the Flipkart website. We performed web scrapping of the mobile sales data and collected the data and we processed the data by dividing the mobiles based on the prices. Using data processing techniques, we have divided the mobiles based on the brands. We have plotted visualization plots to give the user a clear idea about all the mobile brands present in the market so he can choose the better fit.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Overview

What is Web Scraping?

Web scraping is a technique to **automate the extraction process** of a large amount of data from a website. The data present on the websites will be in an unstructured format but with the help of Web scraping, you can scrape, access, and store the data in a much more structured and cleaner format for further analysis.

Web scraping can be performed only on the websites which provide permissions before scraping.

We have performed web scrapping on Flipkart and extracted data from it. We have performed data filtering and plotted the visualization plots.

1.2 Applications

Analyzing mobile phone sales is a classic case. This project attempts to examine the availability of data of mobile phones, comparing the various specifications provided by several leading mobile brands.

The applications of our project are

- We can get to know the most popular mobile in the market
- We can get to know which brand provides mobiles with good specifications
- We can also get to know which brand provides good specifications at a low cost
- We can get to know price v/s specifications ranges
- We can get to know which brand is the most loved
- Users can get to know which brand to prefer depending on the users' interest

1.3 Problem Statement

Many smartphones are available worldwide. There is huge competition among smartphone brands in India, where you can get the latest technology in a smartphone at half the price of other premium phones. Here we analyze the costs and specifications of smartphones.

2. SYSTEM REQUIREMENTS

2.1. Functional Requirements

Functional requirements are the requirements that define specific behavior or function of the system

- The system should allow the user to enter a website to scrape.
- The system should crawl through and extract data from the desired website.
- The system should store data in a database
- The system should allow users to access and query the data from within the web application.

2.2. Non-Functional Requirements

- The system should be user-friendly and easy to navigate.
- The system should be easily maintained with minimal maintenance required.
- The system should be scalable and work efficiently under heavy workloads.
- The system should be easy to install and set up.
- The handling of user and scraped data should conform to the Data Protection Act 1998.

2.3. Software Requirements

- An Ubuntu 16.04.3 VM as a safe environment for compatibility and portability
- Programming Language: Python
- Microsoft Office
- Web browsers for testing, primarily Google Chrome, Firefox, and Microsoft Edge

2.4. Hardware Requirements

- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

3. PROPOSED METHODOLOGY

Today, the Internet is flooded with enormous data relative to what we had a decade ago. According to Forbes, the amount of data we produce every day is truly overwhelming. There are 2.5 quintillion bytes of data generated every day at our current pace, and the credit goes to the Internet of Things (IoT) devices. With access to this data, either in the form of audio, video, text, images or any format, most businesses are relying heavily on data to beat their competitors & succeed in their business. Unfortunately, most of this data is not open. Most websites do not provide the option to save the data which they display on their websites. This data can be extracted through web scrapping. From this, we will be able to plot interactive graphs to understand the market.

3.1 System Architecture

The conceptual model that defines the structure, behavior, and views of our system is:

Techniques to be used are:

- Web Scraping: It is the process of using bots to extract content and data from a website
- Data filtering: It is the process of examining a dataset to exclude, rearrange, or apportion data according to certain criteria
- Data Analysis: It is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data.

Firstly, we performed web scrapping on the Flipkart website and extracted the details of the different mobiles by performing scrapping of 70-80 pages

- We have extracted the RAM, ROM, and battery life from the specifications by performing the string operations
- We have converted all the RAM and ROM to GB
- After performing the scrapping we have to divide data based on prices below 5k,5k to 15k… to 100k.
- We perform data filtering on the above files and write all those to a single large file
- We then created CSV files based on brands and updated them in the file name with brandname.csv
- We plotted the visualization plots for the above .csv files of the brands
- We have also plotted visualization plots for the price vs RAM, ROM, etc…
- We have plotted graphs for price v/s rating to check which range-priced mobiles give the user more satisfaction

3.2 System Workflow

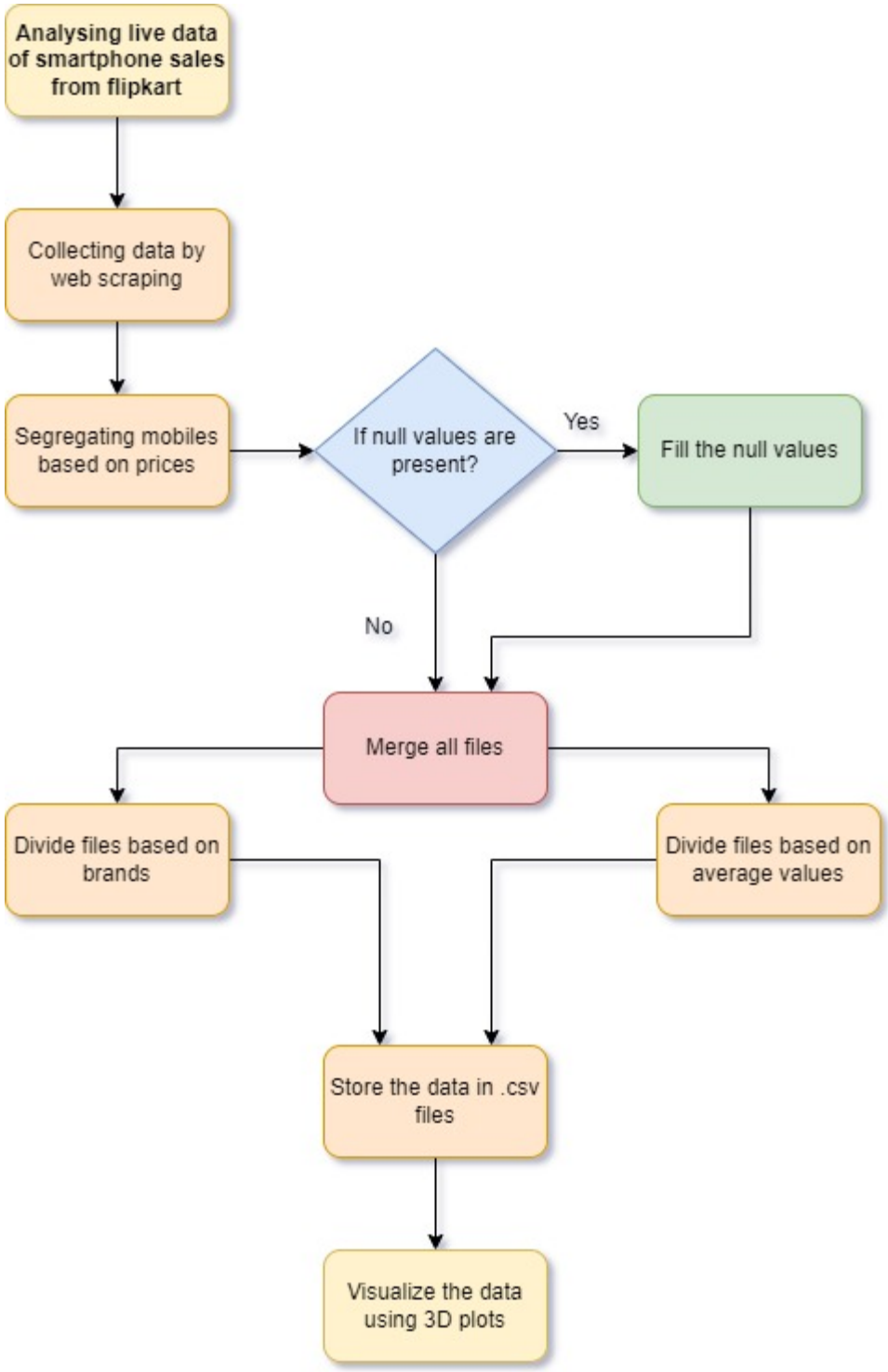

Fig- 3.1 Flow chart representing the project workflow

4. IMPLEMENTATION

We must extract the contents of the webpage from which we want to extract the data.

To do that we need to create a BeautifulSoup object which will parse the HTML content from the page source.

Create a soup object using **`driver.page_source`** to retrieve the HTML text and then we will use the default HTML parser to parse the HTML

Now we have identified that the above card/record indicated by the box contains all the information that we need for a mobile phone. So, let us find out all the tags for these boxes/cards which contain the information we want to extract.

The data is usually embedded in tags. So, we need to inspect the page to see, under which tag the data we want to scrape is embedded. To inspect the page, right-click on the element and select 'Inspect' and then check the attributes.

First, check the output by using the find function then use **`find_all()`** to extract all the data of that attribute from the page.

```
driver.get("https://www.flipkart.com/search?q=mobiles&as=on&as-show=on&otracker=AS_Query_TrendingAutoSuggest_1_0_na_na_na&otrack
source=bs4.BeautifulSoup(driver.page_source)
phone_names=source.find_all('div',attrs={'class':'_4rR01T'})
```

Fig 4.1 Code snippet showing find_all() method

Check the HTML tags of all the attributes like specifications, price, discount, and specifications and extract the data in the above manner adding multiple attributes in the class may also help us in extracting information effectively.

- Now, repeat the same for other attributes. As the scrapping must be done for multiple pages place the above block inside a for loop and iterate it to scrape data for all pages and extract all items by performing string operations on specifications

```
for i in range (2,30):
  url="https://www.flipkart.com/search?q=mobiles&as=on&as-show=on&otracker=AS_Query_TrendingAutoSuggest_1_0_na_na_na&otracker1
  driver.get(url)
  s=driver.page_source
  source=bs4.BeautifulSoup(driver.page_source)
  print(s)
  phone_names=source.find_all('div',attrs={'class':'_4rR01T'})
  Name+=[i.text for i in phone_names]
  phone_prices=source.find_all('div',attrs={'class':'_30jeq3 _1_WHN1'})
  Price+=[i.text for i in phone_prices]
  spec=source.find_all('ul',attrs={'class':'_1xgFaf'})
  specifications+=[i.text for i in spec]
  discount=source.find_all('div',attrs={'class':'_3Ay6Sb'})
  disc+=[discount[i].text for i in range(len(discount))]
  rat=source.find_all('div',attrs={'class':'_3LWZlK'})
  Rating+=[rat[i].text for i in range(len(rat))]
  siz=source.find_all('li',attrs={'class':'rgWa7D'})
  for i in siz:
    m=i.text
    if 'cm' in i.text:
      size.append(i.text[:i.text.index('cm')])
  cb=0
  for i in siz:
    if 'mAh' in i.text:
      Battery.append(i.text[:i.text.index('mAh')])

  for i in siz:
    if 'Rear' not in i.text and '|' in i.text:
      d=i.text[:i.text.index('|')][0:2]
      Rear.append(d)
    elif 'Rear' in i.text:
      Rear.append(i.text[:i.text.index('Rear')])
    else:
      Rear.append(np.nan)
  for i in siz:
    if '|' not in i.text:
      Front.append(np.nan)
```

Fig-4.2 Code for String Operations

Now after collecting the data, write it into a CSV file

The output of the CSV file –

Name	Brand	Price	Discount	Rating	ram	rom	batterylife	front cam	Rear
Nokia 105	Nokia	1299	21	4.4	3.13E-05	0.032	800	0	0
POCO C31	POCO	9499	20	4.3	4	64	5000	5	13
POCO M4	POCO	16999	15	4.2	6	128	5000	16	50
APPLE iPh	APPLE	43999	16	4.2	6	128	4500	12	12
APPLE iPh	APPLE	69999	12	4.7	8	128	4270	12	12
APPLE iPh	APPLE	129900	30	4.2	8	256	6000	12	48
Nokia 105	Nokia	1299	10	4.2	3.13E-05	0.032	800	0	0
REDMI 9i S	REDMI	7099	29	4.3	4	64	5000	5	13
vivo T1 44'	vivo	15999	35	4.4	6	128	5000	16	50

Fig-4.3 CSV File

We must sort the mobile phones based on the prices as we cannot use the same value to fill the column of all the mobile phones. As there are 1000 rupees mobile to 1 lakh mobile we cannot fill the data of all the mobiles in the same way we have to divide mobiles based on the prices and store them in CSV files.

```
import csv
import pandas as pd
file=pd.read_csv('asses.csv')
file.describe()
def new(file,lowerlimit,higherlimit):
      l=[]
      for i in range(len(file)):
        if int(file.iloc[i].Price)<=higherlimit and int(file.iloc[i].Price)>lowerlimit:
          try:
            l.append(file.iloc[i])
          except:
            l.append(file.iloc[i])
      return l
def create(name,file,lowerlimit,higherlimit):
      m=new(file,lowerlimit,higherlimit)
      header=m[0].index[:-1]
      file1=open(name,'w')
      writer=csv.writer(file1)
      writer.writerow(header)
      for i in m:
            writer.writerow(i[:-1])
      file1.close()
create('below5k.csv',file,0,5000)
create('below15k.csv',file,5000,15000)
create('below30k.csv',file,15000,30000)
create('below50k.csv',file,30000,50000)
create('below100k.csv',file,50000,100000)
create('above100k.csv',file,100000,1000000000)
```

Fig-4.4 Dividing into CSV files

Now we have to individually fill all the data into the CSV files and perform data cleaning and filling.

```
file=pd.read_csv("below5k.csv")
file['front camera'].fillna(0,inplace=True)
file['Rear'].fillna(file['Rear'].mode()[0],inplace=True)
file['ram'].fillna(file['ram'].mode()[0],inplace=True)
file['rom'].fillna(file['rom'].mode()[0],inplace=True)
file['batterylife'].fillna(file['batterylife'].mode()[0],inplace=True)
file.to_csv('below5k.csv',index=False)
```

```
file=pd.read_csv("below15k.csv",encoding= 'unicode_escape')
file['front camera'].fillna(file['front camera'].mode()[0],inplace=True)
file['Rear'].fillna(file['Rear'].mode()[0],inplace=True)
file['ram'].fillna(file['ram'].mode()[0],inplace=True)
file['rom'].fillna(file['rom'].mode()[0],inplace=True)
file['batterylife'].fillna(file['batterylife'].mode()[0],inplace=True)
file.to_csv('below15k.csv',index=False)
```

```
file=pd.read_csv("below30k.csv",encoding= 'unicode_escape')
file['front camera'].fillna(file['front camera'].mode(),inplace=True)
file['Rear'].fillna(file['Rear'].mode()[0],inplace=True)
file['ram'].fillna(file['ram'].mode()[0],inplace=True)
file['rom'].fillna(file['rom'].mode()[0],inplace=True)
file['batterylife'].fillna(file['batterylife'].mode()[0],inplace=True)
file.to_csv('below30k.csv',index=False)
```

```
file=pd.read_csv("below50k.csv",encoding= 'unicode_escape')
file['front camera'].fillna(file['front camera'].mode(),inplace=True)
file['Rear'].fillna(file['Rear'].mode()[0],inplace=True)
file['ram'].fillna(file['ram'].mode()[0],inplace=True)
file['rom'].fillna(128,inplace=True)
file['batterylife'].fillna(file['batterylife'].mode()[0],inplace=True)
file.to_csv('below50k.csv',index=False)
```

Fig-4.5 Code to fill data into CSV file

Now write them all to the old existing file. This file doesn't contain any empty or null or improper values as it is written after performing data cleaning and filling.

Now make CSV files for different brands and write all the mobiles info of that brands in that as it is written after data filtering the file is clean and ready to be plotted and we can plot user interactive plots from it.

Now make a CSV file that contains the average price of the mobiles and the average rating and RAM and ROM of the mobiles in a certain price range and plot the graph.

5. RESULTS

Graphs Plotted:

The below graph shows the distribution of the share of different brands in the market

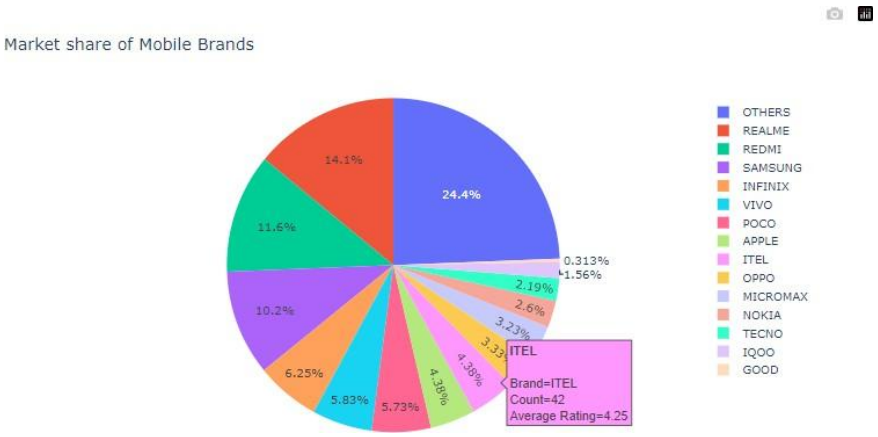

Fig 5.1 Market share of Mobile Brands

The below graph shows us the average rating and availability of different mobile brands in the market

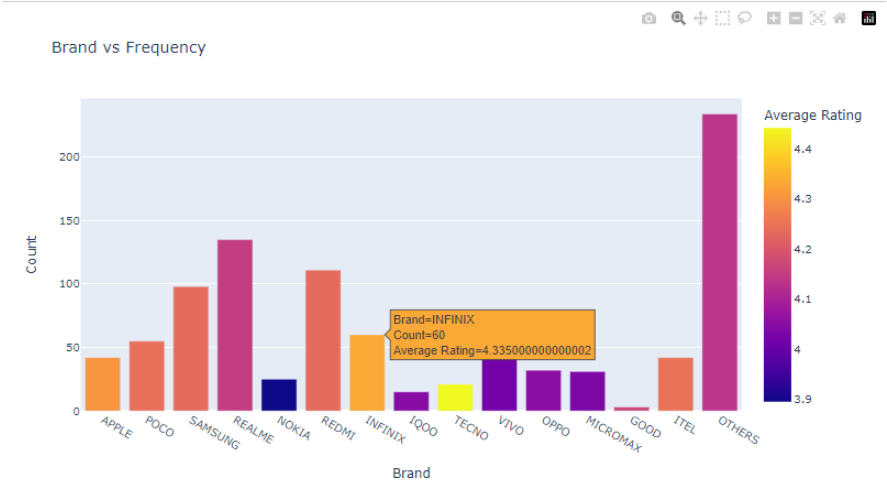

Fig 5.2 Brand v/s Frequency

The below plot shows us identify which brand fits in the budget

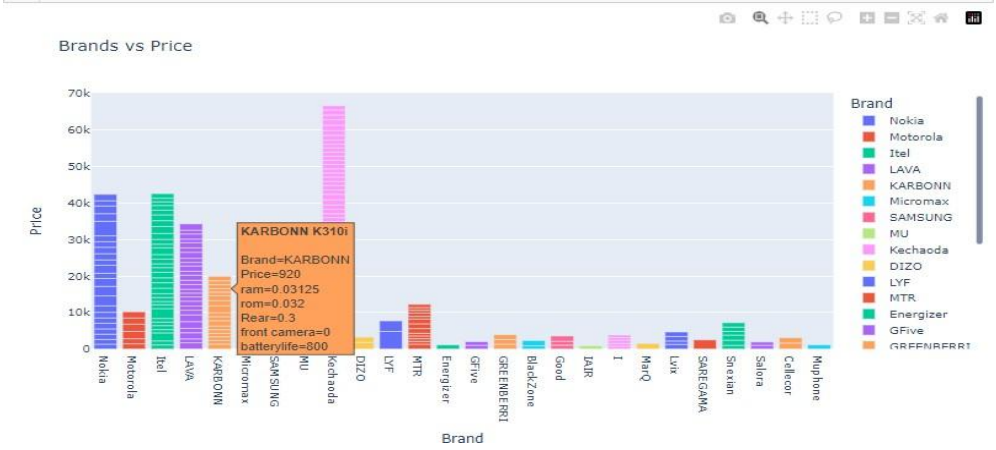

Fig 5.3 Brand v/s Price

Given below are the distribution of different brands of mobiles on comparison with price and rating.

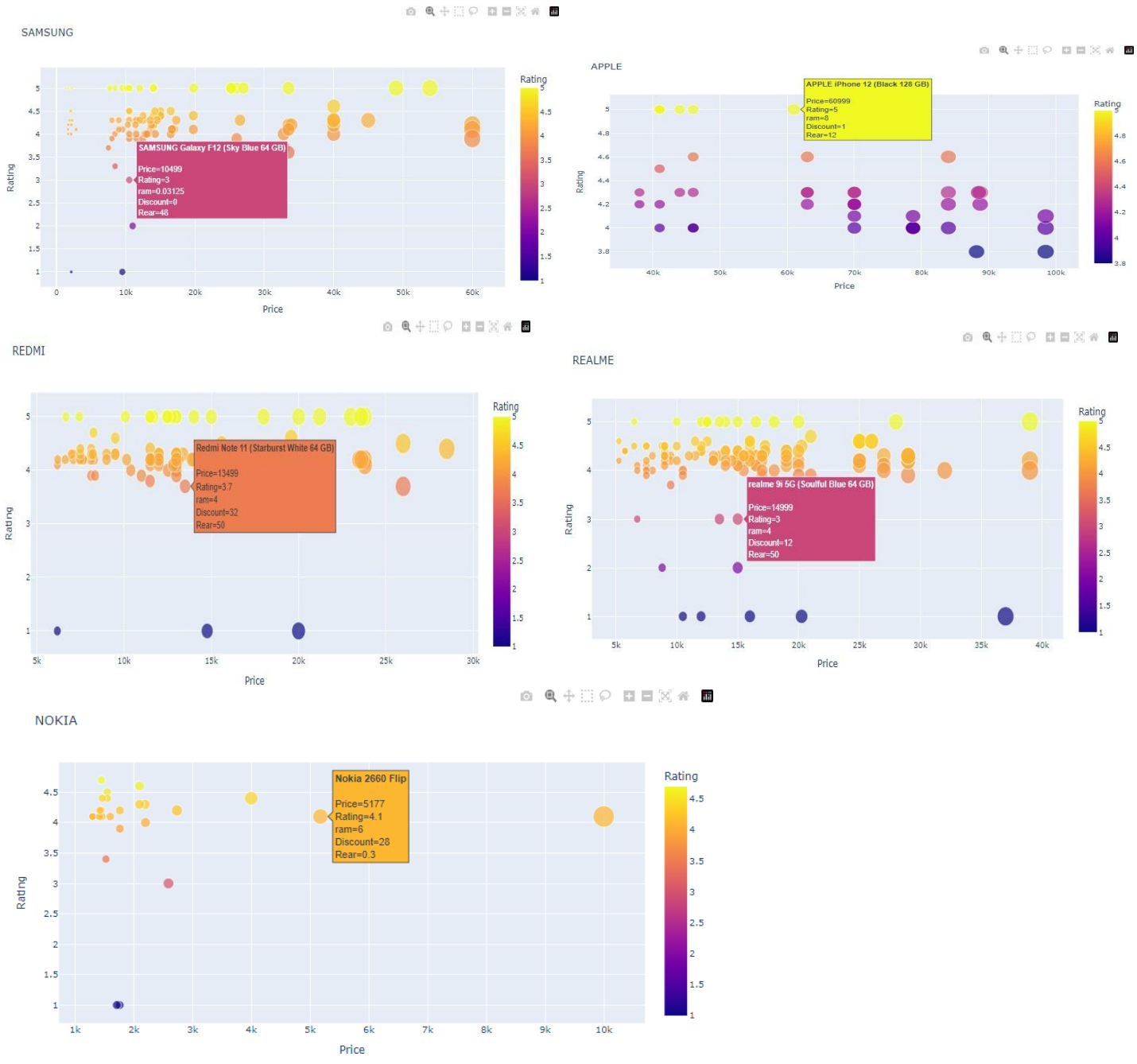

Fig 5.4(a - e) Brand-wise Comparison of Mobiles w.r.t Price and Rating

The below plot depicts the distribution of the different brands' prices and discount

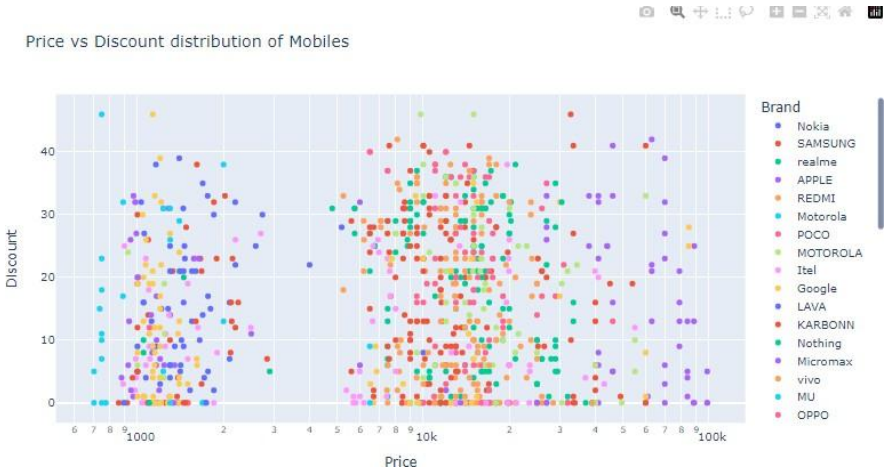

Fig 5.5 Price v/s Distribution of mobiles

The below graph depicts the types of mobiles available in different brands

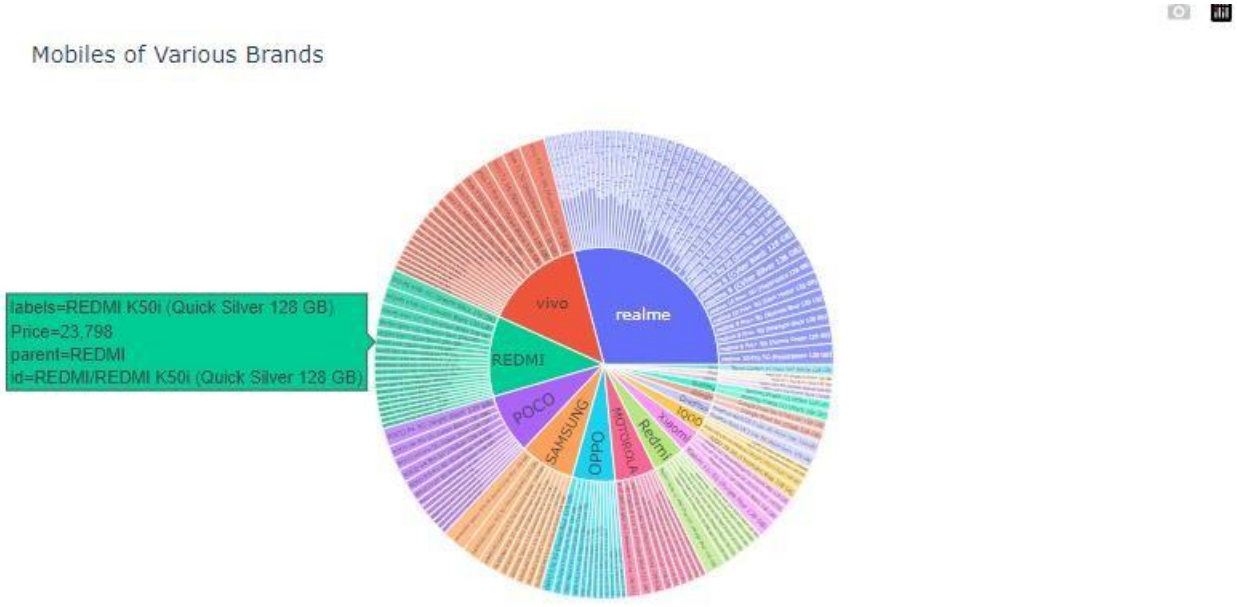

Fig 5.6 Sunburst plot to display Models of various Brands

The below graph depicts the price v/s average RAM as RAM has a greater impact on the systems speed and systems performance availability of devices with good ram at lower or moderate prices will be great

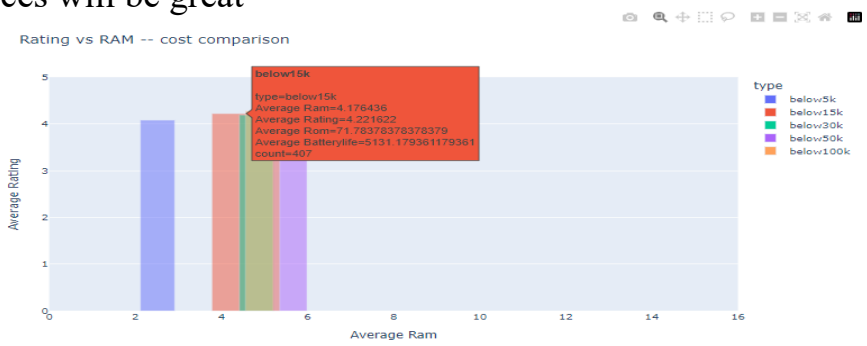

Fig 5.7 Rating vs RAM – cost comparison

6. CONCLUSION

Analyzing a dynamic eCommerce website is a challenging task due to consistently changing information which is dependent on multiple parameters and forms complex patterns. The historical dataset available on the eCommerce websites consists of very few features due to the existence of several types of mobile phones etc., which are not sufficient. To obtain higher accuracy in the predicted price value new variables have been created using the existing variables.

BIBLIOGRAPHY

1. https://pypi.org/project/requests/
2. https://pypi.org/project/beautifulsoup4/
3. https://pandas.pydata.org/
4. https://numpy.org/
5. https://pypi.org/project/selenium/
6. https://plotly.com/python/