SCRAPING E-COMMERCE WEBSITE AND RATING PREDICTION USING MACHINE LEARNING

THARASREE A (19PCS017)

A MAJOR PROJECT REPORT SUBMITTED TO
AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND
HIGHER EDUCATION FOR WOMEN
COIMBATORE – 641043
DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF THE
MASTER'S DEGREE IN COMPUTER SCIENCE
APRIL – 2021

SCRAPING E-COMMERCE WEBSITE AND RATING PREDICTION USING MACHINE LEARNING

THARASREE A (19PCS017)

A MAJOR PROJECT REPORT SUBMITTED TO
AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND
HIGHER EDUCATION FOR WOMEN
COIMBATORE – 641043
DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF THE
MASTER'S DEGREE IN COMPUTER SCIENCE
APRIL – 2021

SIGNATURE OF THE SUPERVISOR **SIGNATURE OF THE HEAD OF THE DEPARTMENT**

VIVA-VOCE EXAMINATION HELD ON_____

SIGNATURE OF EXAMINER

DECLARATION

DECLARATION

We hereby declare that this project work entitled **"Scraping E-commerce Website And Rating Prediction Using Machine Learning".** submitted to **Avinashilingam Institute for Home Science and Higher Education for Women , Coimbatore** in partial fulfillment of the degree of Bachelor of Computer Science is a record of original work done by us under the guidance of **Dr.N.VALLIAMMAL Ph.D, Assistant professor(SS) Department of Computer Science,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore and this project has not formed the basis for the award of any Degree/Diploma/Associate Ship/Fellowship or similar title to any candidate of this University.

SIGNATURE OF THE GUIDE **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **God Almighty**, for his constant love and grace that he showered upon us.

We would like to express our deep sense of reverential gratitude and sincere thanks to **Prof. S.P. Thyagarajan, Chancellor,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of our study.

We owe our great deal of gratitude to **Dr.Premavathy Vijayan M.Sc., M.Ed., Dip.Spl.Edn., M.Phil., Ph.D., Vice Chancellor**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

We express our gratitude to **Dr. S. Kowsalya, M.Sc., M.Phil., Ph.D., Registrar**, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

We are also thankful to **Dr. K. Udaya Chandrika, M.Sc., M.Phil., Ph.D., Dean School of Physical Sciences & Computational Sciences,** Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for granting the facility required.

We wish to place our deep sense of gratitude to **Dr.Vasantha Kalyani Devid, M.Sc., M.Phil., Ph.D., Professor and Head, Department of Computer Science,** for providing all the facilities required to complete the project.

We express our honorable thanks to our project coordinator **Dr. G.Sudhamathy M.C.A.,Ph.D.Assistant Professor (SS), Department of Computer Science,** for her kind valuable advice and knowledgeable suggestions which helped us to complete our project successfully.

We heartily thank our esteemed project guide **Dr. N. Valliammal, M.Sc.,M.Phil., Ph.D., Assistant Professor (SS), Department of Computer Science**, for imparting the tremendous assistance and well-timed support for triumph of our project.

We would like to express our sincere gratitude to all the staff members of the Department of Computer Science, for their constant encouragement and for the opportunity to do our project in this esteemed university. Last yet importantly, we would like to thank our parents, family members, friends and all well-wishers for their kind inspiration, blessings and encouragement during the completion of the course of project.

SYNOPSIS

SYNOPSIS

The paper entitled as **"Scraping E-commerce Website And Rating Prediction Using Machine Learning".** In this project I have taken the flipkart websites and scrape that website(URL) the server sends the data and allows you to read the HTML or XML page and it parses and finds the data, and extracts it. Extraction can be done by scraping the website (URL), inspecting the page, and data will extract and store the data in required format. Here we can find product information, price details, specifications for electronic items. I will extract the Price in Rupees, Product name and rating. Finally merge all the features into a single data frame and it is stored in a required format. This format varies depending on your requirement. It will be converted into an Excel sheet and get the whole dataset in a CSV (Comma Separated Value) format. All the customers give more importance to the ratings. So Customers gives many ratings like 1 to 5. The dataset of Flipkart website products name, user ratings and products price are used. Making sentiment analysis by finding polarity. The ratings are classified into 8 catogories (2.6 to 4.6) then create the confusion matrix it measures the rating for each aspect category. We evaluate our approach by three criteria: precision, recall, and F1-Measure. In this paper we build a model to predict product ratings using Random forest algorithm and Naive Bayes algorithm and identify the best accuracy.

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

1.INTRODUCTION

1.1PROBLEM DEFINITION:

We would predicate how a user would rate for the products with the data set, which the dataset contains product name, product price and ratings. In this project I have taken the product named as Tablets. Tabs are most important for all purposes and it is used for daily use. The people have to go to the shops and buy the tab, here if we purchase directly the amount will be more. But if we purchase through online the amount will be less. If we purchase directly we do no whether the product is good or bad. People get bad products without seeing the ratings of the product. The problem can be rephrased as which kind of algorithm could provide the most accuracy for users. As the people don't view the ratings, they are going to face many difficulties. They are

- Expected features may not be available
- People may receive damage products
- Lack of picture clarity
- Online payment could not be done
- No Proper Battery

1.2 OVERVIEW OF THE PROJECT:

Web scraping is an automated method used to extract large amounts of data from websites. I have taken the flipkart websites and scrape that website (URL) the server sends the data and allows you to read the HTML or XML page and it parses and finds the data, and extracts it. After extracting the data it will save in an CSV format and the data will be converted into an excel sheet. Data collection can be done by web scraping the website and stored in a required format. Next process is preprocessing. Here preprocessing is used to clean the data which is in the excel sheet. The data can be cleaned and the commas can be removed. The exploratory data analysis is done for data visualization. The data can be visualized by using different charts, plots etc…The data like Name, price, ratings can be visualized using exploratory data analysis.

Here next the NLP concept is used for removing the data NLP means Natural language processing. This processing concept plays the vital role in this project. Next the TF-IDF vectorizer is used to extract the features. It is used to convert it to the vector form and apply two algorithms and the ratings can be predicted. The classifier models are used case study of product name for training data to classify comments as positive or negative called opinion mining. In addition, this classifier model has calculated probability that shows value of trend to give the rating using Naive bayes techniques and random forest classifier. Naive bayes which gives best accuracy by classifying when we compared with the random based classifier.

SYSTEM CONFIGURATION

2. SYSTEM CONFIGURATION

2.1 HARDWARE SPECIFICATION :

CPU Type : Intel I3 Processor
Clock Speed : 3.0 Ghz
Ram Size : 4 GB
Hard Disk Capacity : 500 GB
Monitor Type : 15 Inch Color Monitor
Keyboard Type : Standard Keyboard

2.2 SOFTWARE SPECIFICATION :

Operating System : Windows 7
Language : PYTHON 3.7

Libraries used:

- Requests
- Selenium
- Beautiful Soup
- Pandas
- Library
- Matplotlib
    - Pyplot
- Seaborn

2.3 ABOUT THE SOFTWARE:

PYTHON (PROGRAMMING LANGUAGE)

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

HISTORY:

- Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system.

- Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's *Benevolent Dictator For Life*, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.

- He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.

- Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.

- No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

- Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

SYNTAX AND SEMNATICS:

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than C or Pascal

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.

Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning. The recommended indent size is four spaces.

Python allows programmers to define their own types using classes, which are most often used for object-oriented programming.

New instances of classes are constructed by calling the classes like

- `SpamClass()`
- `EggsClass()`
- Metaclass

Before version 3.0, Python had two kinds of classes: *old-style* and *new-style* The syntax of both styles is the same, the difference being whether the class `object` is inherited from, directly or indirectly (all new-style classes inherit from `object` and are instances of `type`).

In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0.

The long-term plan is to support is **gradual typing** and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, **C Python**. An experimental optional static type checker named *numpy* supports compile-time type checking.

LIBRARIES USED:

Requests:

The requests library is the de facto standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.

The GET Request

- HTTP methods such as GET and POST, determine which action you're trying to perform when making an HTTP request. Besides GET and POST, there are several other common methods that you'll use later in this tutorial.
- One of the most common HTTP methods is GET. The GET method indicates that you're trying to get or retrieve data from a specified resource. To make a GET request, invoke requests.get().

The Response

- A Response is a powerful object for inspecting the results of the request.

Inspecting Your Request

- When you make a request, the requests library prepares the request before actually sending it to the destination server.
- Request preparation includes things like validating headers and serializing JSON content.
- Inspecting the Prepared Request gives you access to all kinds of information about the request being made such as payload, URL, headers, authentication, and more.

Table 1: Methods of response

Method	Description
response.headers	response.headers returns a dictionary of response headers.
response.encoding	response.encoding returns the encoding used to decode response.content.
response.elapsed	response.elapsed returns a time delta object with the time elapsed from sending the request to the arrival of the response.
response.close()	response.close() closes the connection to the server.

Selenium:

- ❖ Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e Python, Java, C#, etc, we will be working with Python.

- ❖ Selenium has four major components – Selenium IDE, Selenium RC, Selenium Web driver, Selenium GRID.

Uses of selenium:

- Selenium is an open-source web-based automation tool.
- Python language is used with Selenium for testing.
- It has far less verbose and easy to use than any other programming language
- The Python APIs empower you to connect with the browser through Selenium

Beautiful soup:

- ❖ Beautiful Soup is a Python library for getting data out of HTML, XML, and other markup languages. Say you've found some web pages that display data relevant to your research, such as date or address information, but that do not provide any way of downloading the data directly.

Table 2: Functions of Beautiful Soup

get () 	This function is absolutely essential since with it you will get to the certain web page you desire.
find () 	With the find() function, we are able to search for anything in our web page.
get text () 	As you can see in the previous function we used get_text() to extract the text part of the newly found elements.

strip () 	The strip() method returns a copy of the string with both leading and trailing characters removed (based on the string argument passed).
split () 	This function also has a general-purpose for Python but I found it very useful as well. It splits the string into different parts and we can use the parts that we desire.

Uses of beautiful soup:

- It is an Python package for parsing HTML and XML documents.
- It creates parse trees that is helpful to extract the data easily.

Matplotlib:

- **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
- Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license.
- Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012[5] and was further joined by Thomas Caswell.

Table 3: Colour code and symbols of Matplotlip

Colour	Symbol	Description
b –blue, k-black	**“s”**	square
r-red , w-white	**“p”**	pentagon
m-magenta	"**P**"	Plus(filled)
y-yellow, c-cyan	**“*”**	star

Pandas library:

- ❖ Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks.

Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

- Data cleansing
- Data fill
- Data normalization
- Merges and joins
- Data visualization
- Statistical analysis
- Data inspection
- Loading and saving data

Seaborn:

- ❖ Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- ❖ Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

- ❖ Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

- ❖ Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

API abstraction across visualizations

There is no universally best way to visualize data. Different questions are best answered by different plots. Seaborn makes it easy to switch between different visual representations by using a consistent dataset-oriented API.

Informative distributional summaries

Statistical analyses require knowledge about the distribution of variables in your dataset. The seaborn function **displot()** supports several approaches to visualizing distributions. These include classic techniques like histograms.

SYSTEM STUDY AND ANALYSIS

3. SYSTEM STUDY AND ANALYSIS

3.1 EXISTING SYSTEM:

In the existing system, web scraping is done by scraping the flipkart website. Here I have taken the product named as tablets. Tabs are the important product for the customers. Here the flipkart website is scraped and select the product to inspect. After inspecting the page the data can be extracted and stored in a required format. The extracted data will be stored as a CSV file (Comma separated value) and it will convert into excel sheet. After collecting the data like Name, price, ratings these data can be visualized by using different charts and count the range of the price and ratings purchased by the customers.

Advantage:

1. Customers get to know the price of products
2. Get the counts of the products purchased by the customers
3. Gathering more information about the product

Drawbacks:

1. Only web scraping is done.
2. At a stretch, more pages cannot be scraped.
3. Lack of getting more information.
4. Only visualization could be done.

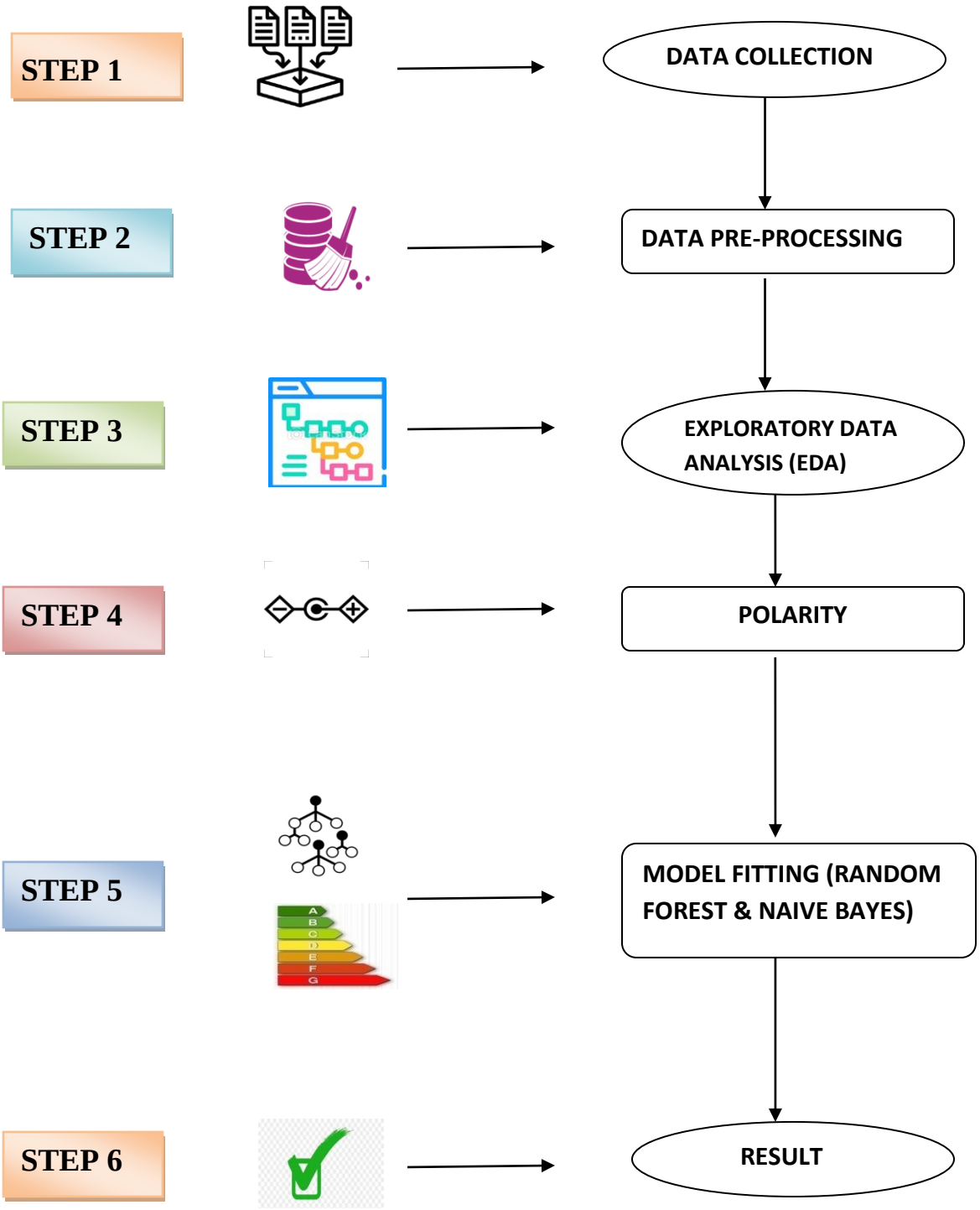
3.2 PROPOSED SYSTEM:

The proposed system overcomes the drawbacks in existing system. After web scraping, further processing is done. The ratings can be predicted by using the algorithms. Now a days people buy products through online by giving more importance for the ratings of the product. I have analyzed the product and classified by using two algorithms. Here the ratings are classified by using Random forest and Naive bayes classifier. The ratings are classified into 8 catagories and predict the rating by giving best accuracy.

Benefits:

- Multiple pages are scraped.
- Customers can purchase by viewing ratings.
- Visualization and classifications are done.

4.METHODOLOGY



SYSTEM DEVELOPMENT

5. SYSTEM DEVELOPMENT

5.1MODULES:

5.2.1 Web scraping

5.2.2 Dataset Collection

5.2.3 Data Pre-Processing

5.2.4 EDA (Exploratory Data Analysis)

5.2.5 Polarity

5.2.6 Model Fitting

Random Forest Algorithm

Navy base algorithm

5.2.7 Result

5.2MODULE DESCRIPTION:

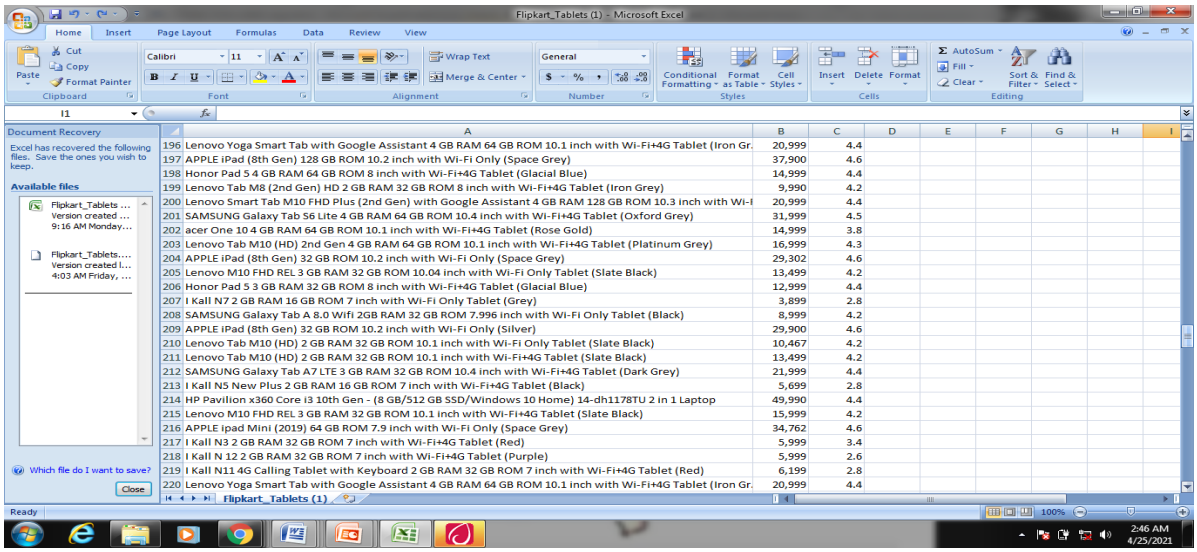
5.2.1 Web scraping:

- I have taken the flipkart websites and scrape that website(URL) the server sends the data and allows you to read the HTML or XML page and it parses and finds the data, and extracts it.
- Python library requests and beautifulsoup4 are used for performing web scraping. To parse html pages a python library Beautifulsoap4 is used.
- Here we scrape flipkart website and taken product is Tablets.

5.2.2 Dataset collection:

- We collect the data from e-commerce website. I will extract data like product name, price in rupees and rating for the website called flipkart.

- Datasets are collected from Web scraping, I have scraped multiple pages. Datasets are in Data frame format, Data frame means two dimensional- Two dimensional: x axis and y axis.
- Inputs are Product Information, Price, Ratings and Dataset is in .csv file format.



	A	B	C
196	Lenovo Yoga Smart Tab with Google Assistant 4 GB RAM 64 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Iron Gr.	20,999	4.4
197	APPLE iPad (8th Gen) 128 GB ROM 10.2 inch with Wi-Fi Only (Space Grey)	37,900	4.6
198	Honor Pad 5 4 GB RAM 64 GB ROM 8 inch with Wi-Fi+4G Tablet (Glacial Blue)	14,999	4.4
199	Lenovo Tab M8 (2nd Gen) HD 2 GB RAM 32 GB ROM 8 inch with Wi-Fi+4G Tablet (Iron Grey)	9,990	4.2
200	Lenovo Smart Tab M10 FHD Plus (2nd Gen) with Google Assistant 4 GB RAM 128 GB ROM 10.3 inch with Wi-I	20,999	4.4
201	SAMSUNG Galaxy Tab S6 Lite 4 GB RAM 64 GB ROM 10.4 inch with Wi-Fi+4G Tablet (Oxford Grey)	31,999	4.5
202	acer One 10 4 GB RAM 64 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Rose Gold)	14,999	3.8
203	Lenovo Tab M10 (HD) 2nd Gen 4 GB RAM 64 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Platinum Grey)	16,999	4.3
204	APPLE iPad (8th Gen) 32 GB ROM 10.2 inch with Wi-Fi Only (Space Grey)	29,302	4.6
205	Lenovo M10 FHD REL 3 GB RAM 32 GB ROM 10.04 inch with Wi-Fi Only Tablet (Slate Black)	13,499	4.2
206	Honor Pad 5 3 GB RAM 32 GB ROM 8 inch with Wi-Fi+4G Tablet (Glacial Blue)	12,999	4.4
207	I Kall N7 2 GB RAM 16 GB ROM 7 inch with Wi-Fi Only Tablet (Grey)	3,899	2.8
208	SAMSUNG Galaxy Tab A 8.0 Wifi 2GB RAM 32 GB ROM 7.996 inch with Wi-Fi Only Tablet (Black)	8,999	4.2
209	APPLE iPad (8th Gen) 32 GB ROM 10.2 inch with Wi-Fi Only (Silver)	29,900	4.6
210	Lenovo Tab M10 (HD) 2 GB RAM 32 GB ROM 10.1 inch with Wi-Fi Only Tablet (Slate Black)	10,467	4.2
211	Lenovo Tab M10 (HD) 2 GB RAM 32 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Slate Black)	13,499	4.2
212	SAMSUNG Galaxy Tab A7 LTE 3 GB RAM 32 GB ROM 10.4 inch with Wi-Fi+4G Tablet (Dark Grey)	21,999	4.4
213	I Kall N5 New Plus 2 GB RAM 16 GB ROM 7 inch with Wi-Fi+4G Tablet (Black)	5,699	2.8
214	HP Pavilion x360 Core i3 10th Gen - (8 GB/512 GB SSD/Windows 10 Home) 14-dh1178TU 2 in 1 Laptop	49,990	4.4
215	Lenovo M10 FHD REL 3 GB RAM 32 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Slate Black)	15,999	4.2
216	APPLE ipad Mini (2019) 64 GB ROM 7.9 inch with Wi-Fi Only (Space Grey)	34,762	4.6
217	I Kall N3 2 GB RAM 32 GB ROM 7 inch with Wi-Fi+4G Tablet (Red)	5,999	3.4
218	I Kall N 12 2 GB RAM 32 GB ROM 7 inch with Wi-Fi+4G Tablet (Purple)	5,999	2.6
219	I Kall N11 4G Calling Tablet with Keyboard 2 GB RAM 32 GB ROM 7 inch with Wi-Fi+4G Tablet (Red)	6,199	2.8
220	Lenovo Yoga Smart Tab with Google Assistant 4 GB RAM 64 GB ROM 10.1 inch with Wi-Fi+4G Tablet (Iron Gr.	20,999	4.4

Figure 1: Sample Dataset

Dataset Description:

In this project taken data set we called as **Flipkart_Tablets.csv**

- This dataset has a shape of 360×3.
- This dataset has 3 rows and 360 columns.
- The first column identifies the product name , second column is price and third column identifies ratings of the product..

5.2.3 Data Pre-Processing:

- Data pre-processing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data.
- Data pre-processing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.

Steps in Data Pre-processing:

***Step 1*:** Acquire the dataset

***Step 2*:** Import all the crucial libraries

***Step 3*:** Handling Missing Values

***Step 4*:** Encoding the categorical data

***Step 5*:** Splitting the dataset

1.Acquire the dataset:

- To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset.
- For predicting the ratings from flipkart websites the dataset have been collected by web scraping the flipkart website.

2. Import all the crucial libraries

The Three core Python libraries used for this data pre-processing in Machine Learning are: **NumPy**,**Pandas** ,**Matplotlib.**

- **Import the dataset**
- Importing dataset in Python-Jupiter
- Using the libraries pandas, importing the flipkart dataset using pd.read_excel and importing the dataset path.
- flipcart_data=pd.read_csv("/content/drive/MyDrive/Rating_prediction/Rating_prediction/Flipkart_Tablets.csv")

3. Handling Missing Values

- In data pre-processing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data.
- Our dataset does not contain any missing values so this step is ignored.

4. Encoding the categorical data

Categorical data refers to the information that has specific categories within the dataset. As our dataset contain many categorical values, this step is ignored.

5. Splitting the dataset

Every dataset for Machine Learning model must be split into two separate sets.

5.2.4 Exploratory data analysis:

- Exploratory Data Analysis(EDA): Exploratory data analysis is a complement to inferential statistics, which tends to be fairly rigid with rules and formulas.
- At an advanced level, EDA involves looking at and describing the data set from different angles and then summarizing it..
- It is used to show historical data by using some analytics tools. It helps in drilling down the information, to transform metrics, facts, and figures into initiatives for improvement.

We will explore a Data set and perform the exploratory data analysis.

- Handle Missing value
- Removing duplicates
- Normalizing and Scaling(Numerical Variables)
- Encoding Categorical variables(Dummy Variables)

In this project, we visualize the dataset by using Distribution plot, Histograms and Bar plot.

Distribution Plot:

- The next visualization in line is the distribution plot. The distribution plot, as the name suggests is used for one purpose i.e. displaying the distribution and range of data values over a scale.
- In this project I have plotted the price and the ratings of the product.

Advantages: It can easily display a general distribution of a set of numeric values over a range.

Bar plot:

- A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent.
- The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories.
- One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.
- In this project I have plotted the product name, price and the ratings of the product.

Histogram:

- A histogram is a great tool for quickly assessing a probability distribution that is intuitively understood by almost any audience.
- Python offers a handful of different options for building and plotting histograms.
- In this project I have plotted the price and the ratings of the product.

Three major points in histogram

- Building histograms in pure Python, without use of third party libraries
- Constructing histograms with NumPy to summarize the underlying data
- Plotting the resulting histogram with Matplotlib, Pandas, and Seaborn

5.2.5 Polarity:

- Polarity analysis takes into account the amount of positive or negative terms that appear in a given sentence.
- By using machine learning methods and natural language processing, we can extract the personal information of a document and attempt to classify it according to its polarity, such as positive, neutral, or negative, making sentiment analysis.
- Typically, we quantify this sentiment with a positive or negative value, called polarity. The overall sentiment is often inferred as positive, neutral or negative from the sign of the polarity score.
- Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion.
- In polarity we use NLP concept for pre processing.

Natural language pre-proccessing:

- Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.
- Here in NLP nltk(Natural Language Tool Kit) tool is used to clean the data.
- In this project there are few steps in NLP are done they are:
    - Text Pre-Processing(Pre-processing is done for cleaning the raw data)
    - NLP concept is used to extract the main features.
    - Here I have removed the un-necessary data like RAM, ROM, inches, colour etc…
    - By using the Id and product name we can predict the ratings easily.

Tf-idf vectorizer:

- TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.
- Here the ratings are in numerical format, product information are in the string format, for that we have to convert the string into the vector form.
- For converting into vector form the TF-IDF vectorizer is used.
- Mainly this is used to extract the features.

After converting we will implement the algorithms and identify the accuracy.

Table 5: Methods in NLP

Tokenization	Tokenization refers to dividing text or a sentence into a sequence of tokens, which roughly correspond to "words". This is one of the basic tasks of NLP. To do this using TextBlob, follow the two steps: 1. Create a **textblob** object and pass a string with it. 2. Call **functions** of textblob in order to do a specific task.
Stop words	Stop-words being most commonly used in the English language; however, these words have no predictive power in reality. Words such as I, me, myself, he, she, they, our, mine, you, yours etc.
Stemming	Stemming algorithm is very useful in the field of text mining and helps to gain relevant information as it reduces all words with the same roots to a common form by removing suffixes such as -action, ing, -es and -ses. However, there can be problematic where there are spelling errors.
Removing Punctuations	Removing all punctuation marks from a sentence removes all punctuation marks from each word. For example, the sentence "Think and wonder, wonder and think." tokenized and removed of punctuation marks is ['Think', 'and', 'wonder', 'wonder', 'and', 'think'].

In our project I have used NLTK tool and TextBlob in NLP.

TextBlob:

TextBlob aims to provide access to common text-processing operations through a familiar interface. You can treat **TextBlob** objects as if they were Python strings that learned how to do Natural Language Processing.

Following steps in TextBlob

- Create a TextBlob
- Sentiment Analysis

5.2.6 Model fitting

Model fitting is the measure of how well a machine learning model generalizes data similar to that with which it was trained. A **good model fit** refers to a model that accurately approximates the output when it is provided with unseen inputs. **Fitting** refers to adjusting the parameters in the model to improve accuracy. The process involves running an algorithm on data for which the target variable ("labeled" data) is known to produce a machine learning model. Then, the model's outcomes are compared to the real, observed values of the target variable to determine the accuracy.Here in this project we use two classifiers in machine learning for predicting the ratings of the product. The two algorithms are

- ❖ Random Forest classifier
- ❖ Naive bayes classifier

Random Forest Algorithm:

- ❖ Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.
- ❖ Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- ❖ Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- o It takes less training time as compared to other algorithms.
- o It predicts output with high accuracy, even for the large dataset it runs efficiently.
- o It can also maintain accuracy when a large proportion of data is missing.

Python Implementation of Random Forest Algorithm

Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user_data.csv", which we have used in previous classification models.

Implementation Steps are given below:

- Data Pre-processing step
- Fitting the Random forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

1.Data Pre-Processing Step:

- importing libraries – pandas , numpy , matplotlib
- importing datasets-flipkart_tablets.csv
- Extracting Independent and dependent Variable
- Splitting the dataset into training and test set

Split the data as x and y into training and test sets. Here x as name, price and y as ratings of the product.

2.Fitting the Random Forest algorithm to the training set:

Now we will fit the Random forest algorithm to the training set. To fit it, we will import the **Random Forest Classifier** class from the **sklearn.ensemble** library.

- **n_estimators=** The required number of trees in the Random Forest. The default value is 10. We can choose any number but need to take care of the overfitting issue.
- **criterion=** It is a function to analyze the accuracy of the split.

3. Predicting the Test Set result

Since our model is fitted to the training set, so now we can predict the test result. Here we will predict the ratings and find the best accuracy.

4.Creating the Confusion Matrix

Each row in a confusion matrix represents an actual class, while each column represents a predicted class. For more info about the confusion matrix

These are calculated by using

True positive , True negative ,False positive , False negative

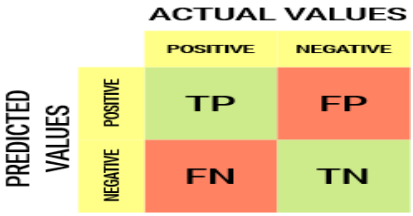


Figure 2: Polarity

True Positive (TP)

- The predicted value matches the actual value.
- The actual value was positive and the model predicted a positive value.

True Negative (TN)

- The predicted value matches the actual value.
- The actual value was negative and the model predicted a negative value.

False Positive (FP) – Type 1 error

- The predicted value was falsely predicted.
- The actual value was negative but the model predicted a positive value and also known as the Type 1 error.

False Negative (FN) – Type 2 error

- The predicted value was falsely predicted.
- The actual value was positive but the model predicted a negative value and also known as the Type 2 error.

- **Precision**

$$Precision = \frac{TP}{TP + FP}$$

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

TP is the number of true positives, and FP is the number of false positives. A trivial way to have perfect precision is to make one single positive prediction and ensure It is correct (precision = 1/1 = 100%). This would not be very useful since the classifier would ignore all but one positive instance.

- **Recall** $$Recall = \frac{TP}{TP + FN}$$

Recall goes another route. Instead of looking at the number of false positives the model predicted, recall looks at the number of **false negatives** that were thrown into the prediction.

- **F1 score** $$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Precision-Recall values can be very useful to understand the performance of a specific algorithm and also helps in producing results based on the requirements. But when it comes to comparing several algorithms trained on the same data, it becomes difficult to understand which algorithm suits the data better solely based on the Precision-Recall values.

- **Accuracy** $$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

The sum of true positives and true negatives divided by the total number of samples.

The accuracy is calculated by applying random forest and naïve bayes algorithm and give the accuracy. Here the naïve bayes is easy and it gives the best accuracy.

Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.

Naive Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.

- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

Why is it called Naive Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naive**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Python Implementation of the Naïve Bayes algorithm:

Now we will implement a Naive Bayes Algorithm using Python. So for this, we will use the "**flipkart_tablets**" **dataset**, which we have used in our other classification model. Therefore we can easily compare the Naive Bayes model with the other models.

Steps to implement:

- Fitting Naive Bayes to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

1.Fitting the Random Forest algorithm to the training set:

After the pre-processing step, now we will fit the Naive Bayes model to the Training set.

2. Predicting the Test Set result

Now we will predict the test set result. For this, we will create a new predictor variable **y_pred**, and will use the predict function to make the predictions.

3. Creating Confusion Matrix:

Now we will check the accuracy of the Naive Bayes classifier using the Confusion matrix. Confusion matrix are created by using precision, recall, F1 score

True positive, True negative, False positive, False negative

- **Precision**

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**

$$Recall = \frac{TP}{TP + FN}$$

- **F1 score**

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

- **Accuracy**

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It performs well in Multi-class predictions as compared to the other Algorithms.

5.2.7 Result:

To evaluate the accuracy score using confusion matrix method. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. Here the ratings can be classified into 8 catogories and the ratings can be predicted by using two classifiers. By comparing these two algorithms check whether which gives the best. Here we can predict ratings by using many algorithms but Naïve Bayes algorithm is easy and it gives the best accuracy.

CONCLUSION

6. CONCLUSION

Web scraping and techniques confronting numerous difficulties as the extraction of the information are not excessively simple. In this work the ecommerce websites are scraped to extract the product details and the extracted data is stored in CSV format.

Mostly all the customers give more importance to the ratings of the products. It can quickly predict the ratings of the products by using Random forest algorithm and Naive bayes. By classifying Random forest classifier gives the accuracy. The ratings are classified into 8 catogories (2.6 – 4.6). Naïve bayes algorithm gives 1.0 accuracy and Random forest algorithm gives 0.96 accuracy. Naïve bayes gives the best accuracy compared to random forest algorithm.

SCOPE FOR FUTURE DEVELOPMENT

7. SCOPE FOR FUTURE DEVELOPMENT

- In future we will scrape two e commerce websites like Amazon, snapdeal etc…
- Extract the entire features like specifications of different products like laptops, mobile phones etc…
- Predict the ratings for both the different e commerce websites.
- Use many algorithms to predict and give the best accuracy.

8. BIBLIOGRAPHY

REFERENCE BOOKS:

1. Li, Q. Yang and X. Xue, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.[C]", *IJCAI*, pp. 2052-2057, 2009.

2.NN. Liu, EW. Xiang, M. Zhao et al., "Unifying explicit and implicit feedback for collaborative filtering[C]", *ACM International Conference on Information and Knowledge Management*, pp. 1445-1448, 2010.

3.Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems[J]", *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

REFERENCE WEBSITES:

https://medium.com/edureka/web-scraping-with-python-d9e6506007bf

https://stackoverflow.com/questions/43134940/how-to-get-flipkart-product-data-using-web-scraping/51753794

https://ieeexplore.ieee.org/document/7965709/references#references

https://ieeexplore.ieee.org/abstract/document/8228161

9.APPENDIX

SCREENSHOTS

Web scraping

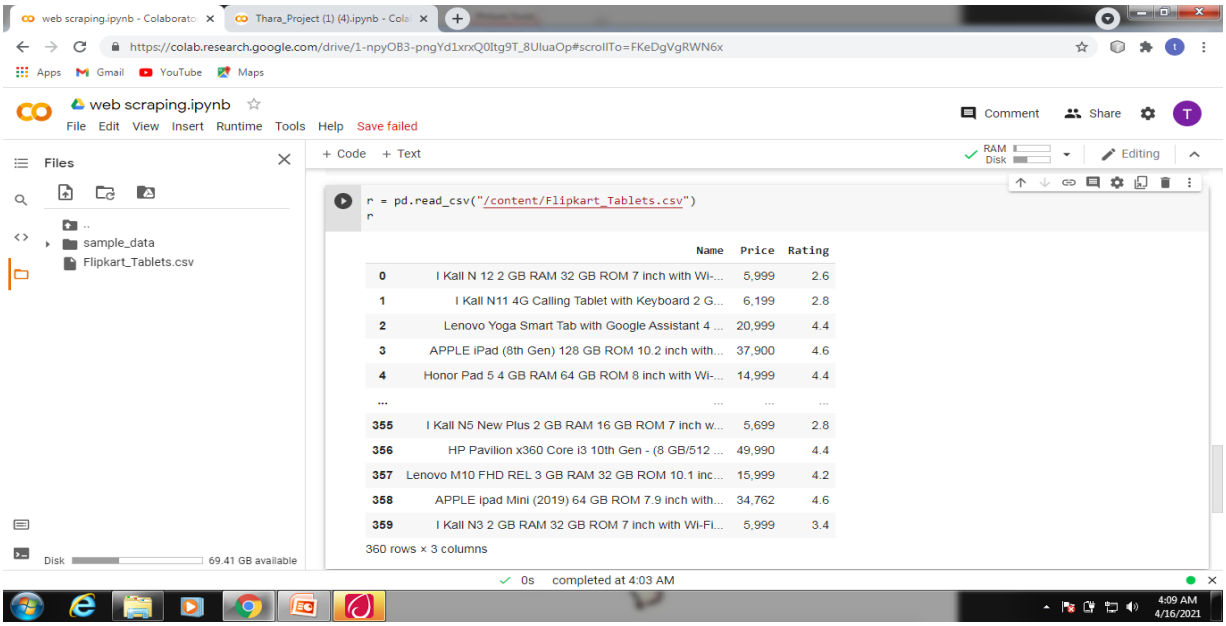


Figure 3: Web scraping from flipkart website

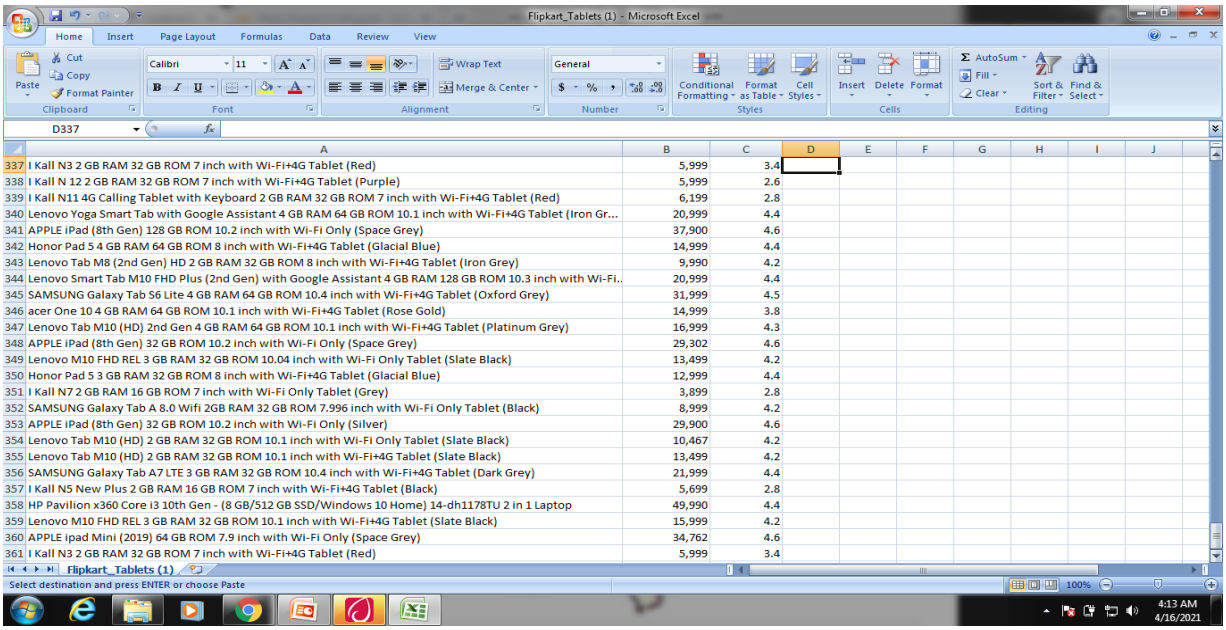


Figure 4: Dataset collected by scraping the website

Data pre-processing:

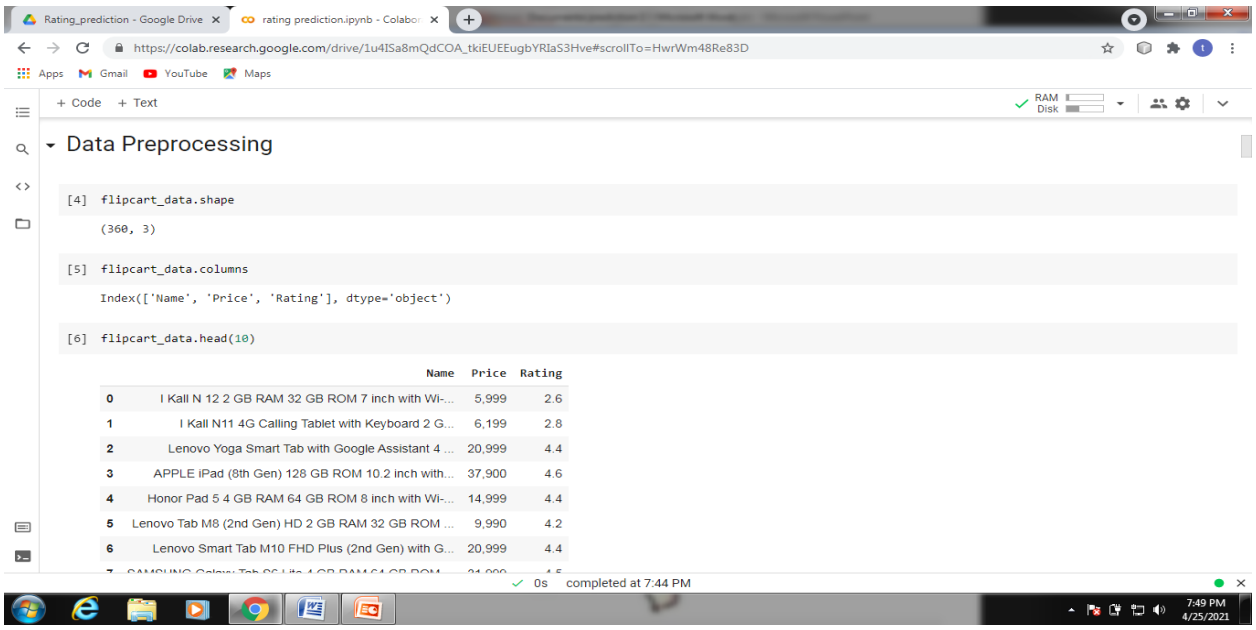


Figure 5: Data preprocessing

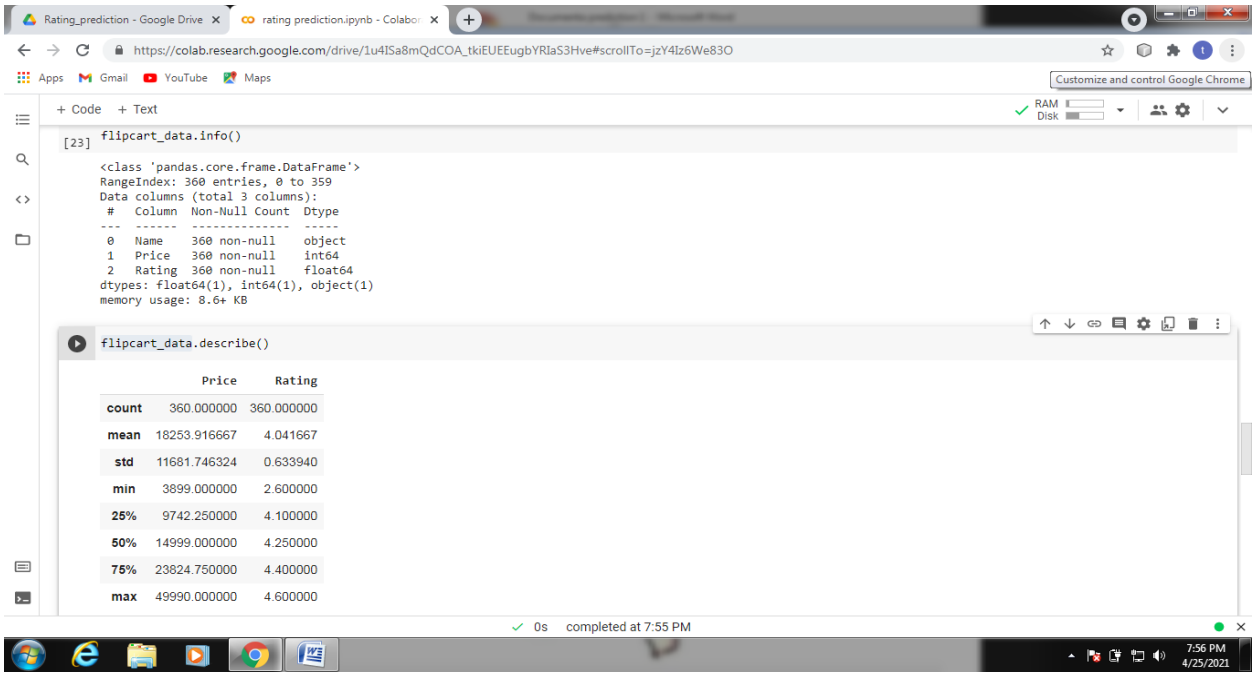


Figure 6: Null value checking

Exploratory data Analysis:

```
[19] plt.hist(flipcart_data.head(10).Price)

(array([1., 1., 2., 0., 1., 2., 0., 1., 1., 1.]),
 array([0. , 0.7, 1.4, 2.1, 2.8, 3.5, 4.2, 4.9, 5.6, 6.3, 7. ]),
 <a list of 10 Patch objects>)
```

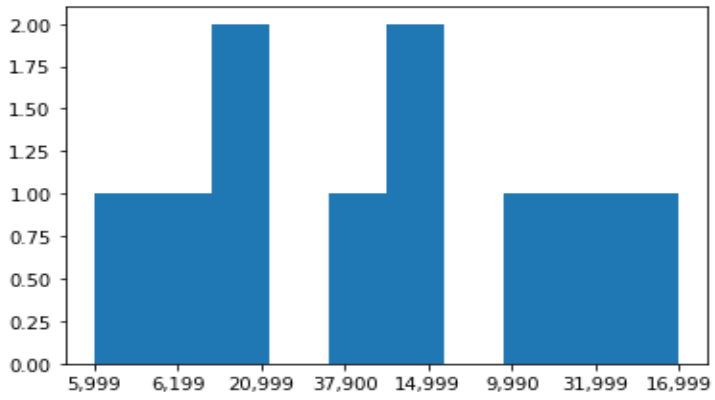


Figure 7: Visualization of price

```
[ ] plt.hist(flipcart_data.Rating)

(array([ 15.,  45.,   0.,   0.,  15.,   0.,  15.,   0., 105., 165.]),
 array([2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. , 4.2, 4.4, 4.6]),
 <a list of 10 Patch objects>)
```

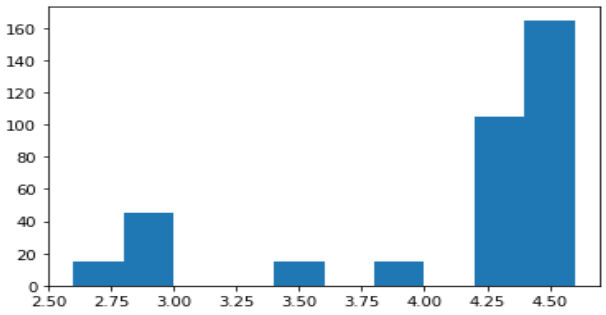


Figure 8: Visualization of ratings

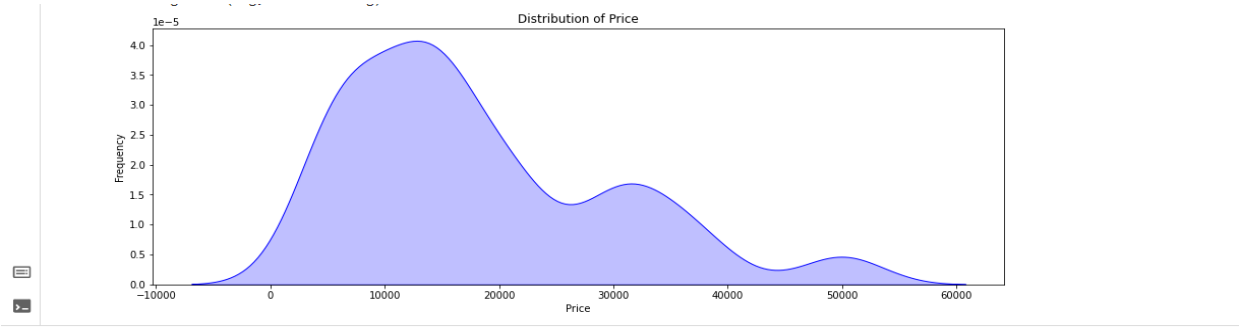


Figure 9: Distribution plots of price and ratings

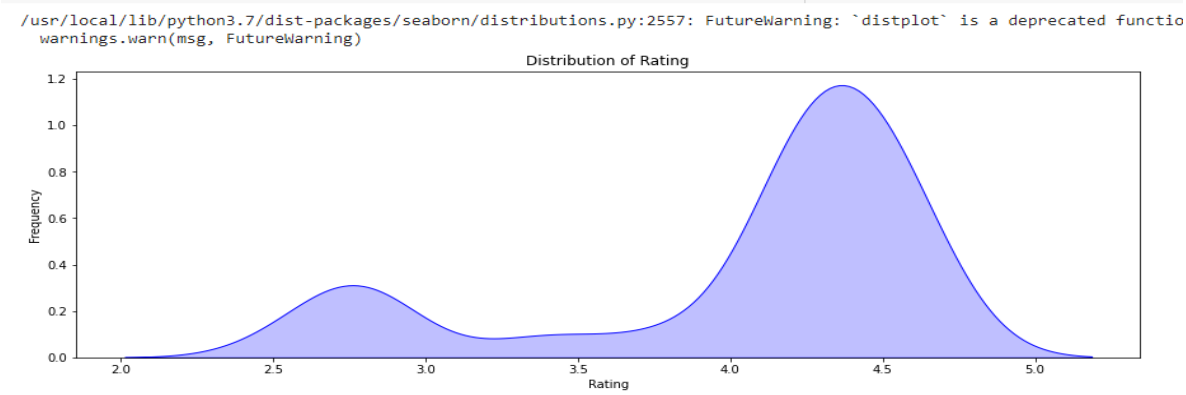


Figure 10: Distribution plots of price and ratings

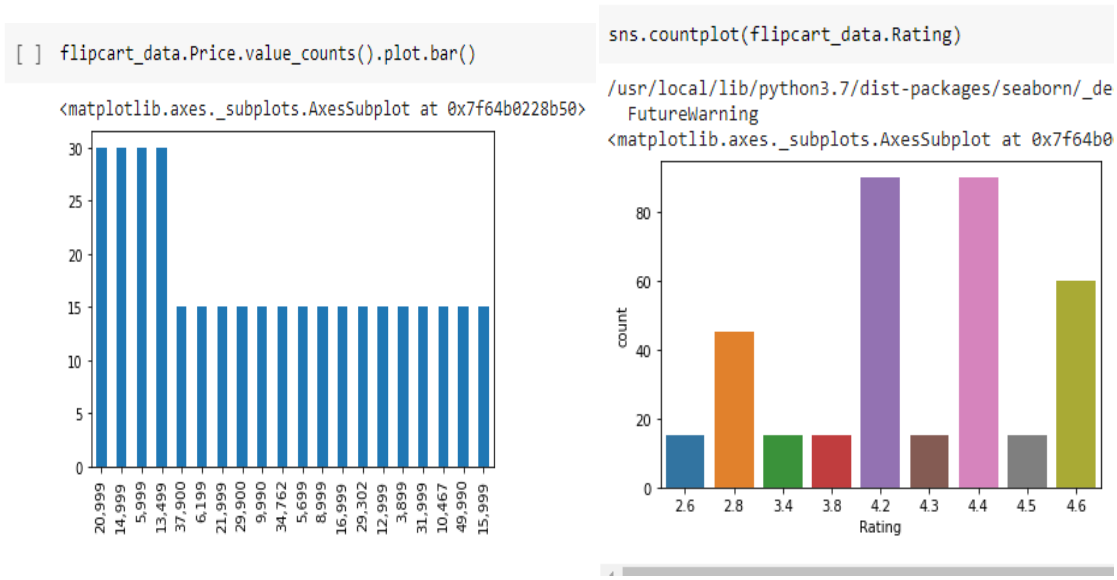
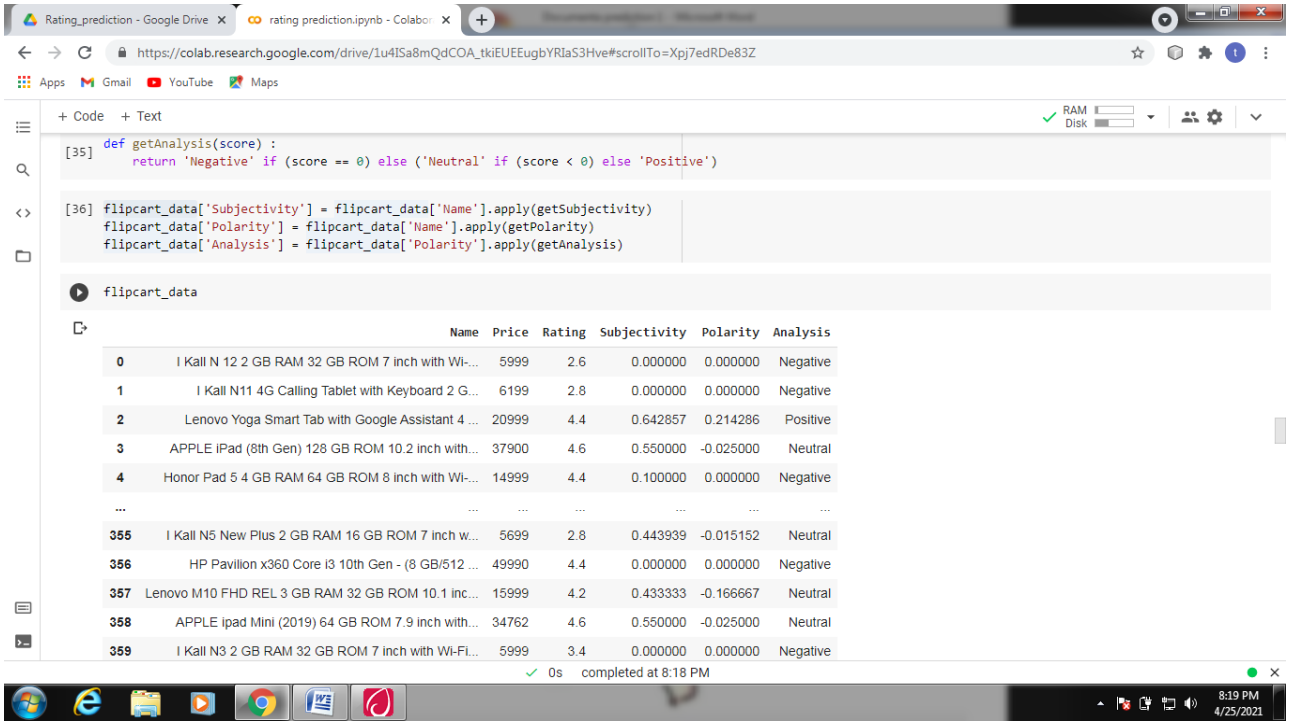


Figure 11: Bar plots of price and ratings

Polarity

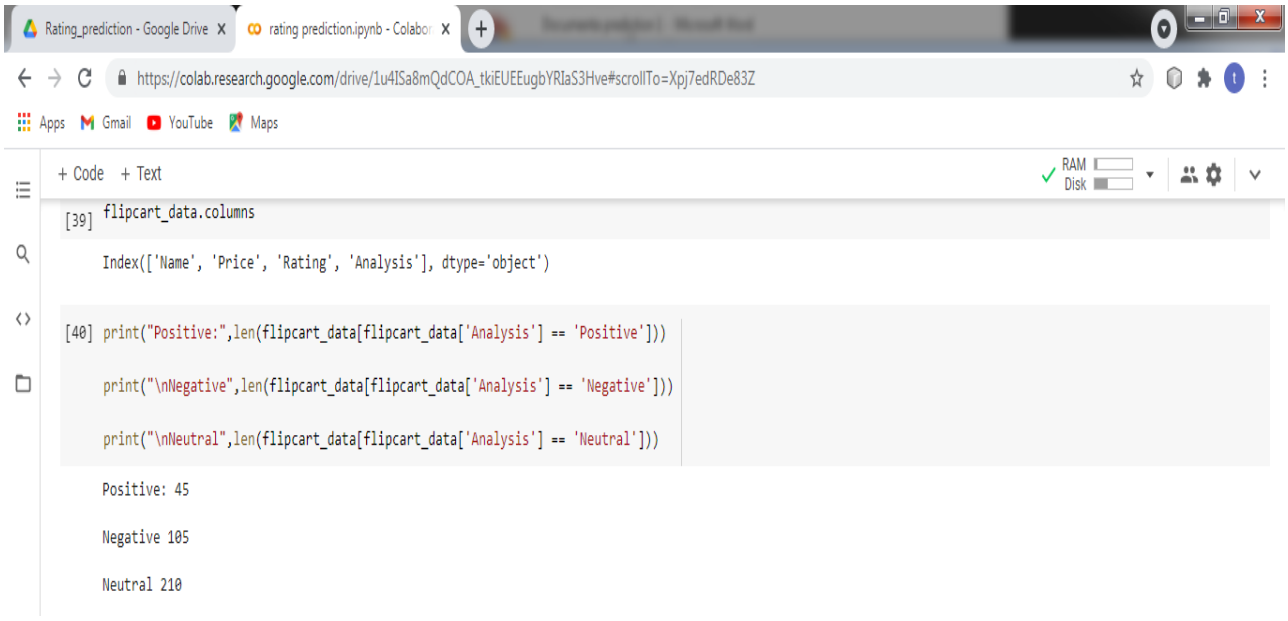


```
[35] def getAnalysis(score) :
         return 'Negative' if (score == 0) else ('Neutral' if (score < 0) else 'Positive')

[36] flipcart_data['Subjectivity'] = flipcart_data['Name'].apply(getSubjectivity)
     flipcart_data['Polarity'] = flipcart_data['Name'].apply(getPolarity)
     flipcart_data['Analysis'] = flipcart_data['Polarity'].apply(getAnalysis)

     flipcart_data
```

	Name	Price	Rating	Subjectivity	Polarity	Analysis
0	I Kall N 12 2 GB RAM 32 GB ROM 7 inch with Wi-...	5999	2.6	0.000000	0.000000	Negative
1	I Kall N11 4G Calling Tablet with Keyboard 2 G...	6199	2.8	0.000000	0.000000	Negative
2	Lenovo Yoga Smart Tab with Google Assistant 4 ...	20999	4.4	0.642857	0.214286	Positive
3	APPLE iPad (8th Gen) 128 GB ROM 10.2 inch with...	37900	4.6	0.550000	-0.025000	Neutral
4	Honor Pad 5 4 GB RAM 64 GB ROM 8 inch with Wi-...	14999	4.4	0.100000	0.000000	Negative
...	...	...	...	...	...	...
355	I Kall N5 New Plus 2 GB RAM 16 GB ROM 7 inch w...	5699	2.8	0.443939	-0.015152	Neutral
356	HP Pavilion x360 Core i3 10th Gen - (8 GB/512 ...	49990	4.4	0.000000	0.000000	Negative
357	Lenovo M10 FHD REL 3 GB RAM 32 GB ROM 10.1 inc...	15999	4.2	0.433333	-0.166667	Neutral
358	APPLE ipad Mini (2019) 64 GB ROM 7.9 inch with...	34762	4.6	0.550000	-0.025000	Neutral
359	I Kall N3 2 GB RAM 32 GB ROM 7 inch with Wi-Fi...	5999	3.4	0.000000	0.000000	Negative

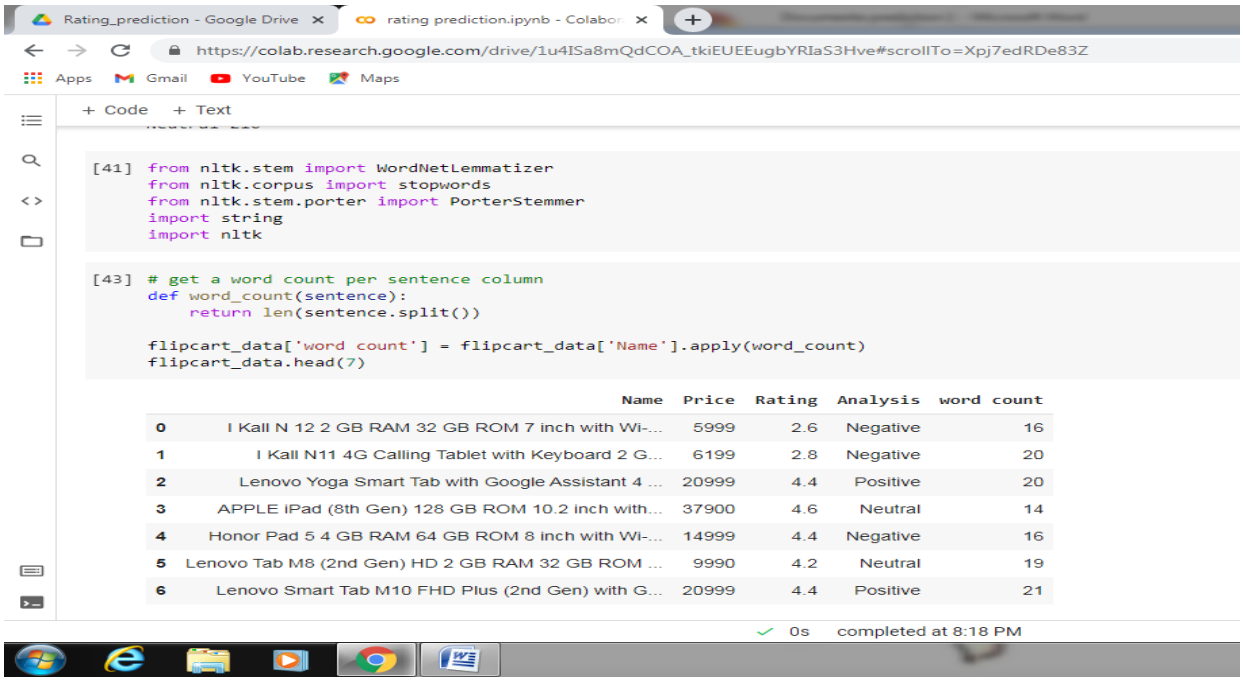
0s completed at 8:18 PM

Figure 12: Sentiment Analysis



```
[39] flipcart_data.columns

     Index(['Name', 'Price', 'Rating', 'Analysis'], dtype='object')

[40] print("Positive:",len(flipcart_data[flipcart_data['Analysis'] == 'Positive']))

     print("\nNegative",len(flipcart_data[flipcart_data['Analysis'] == 'Negative']))

     print("\nNeutral",len(flipcart_data[flipcart_data['Analysis'] == 'Neutral']))

     Positive: 45

     Negative 105

     Neutral 210
```

Figure 13: Finding polarity



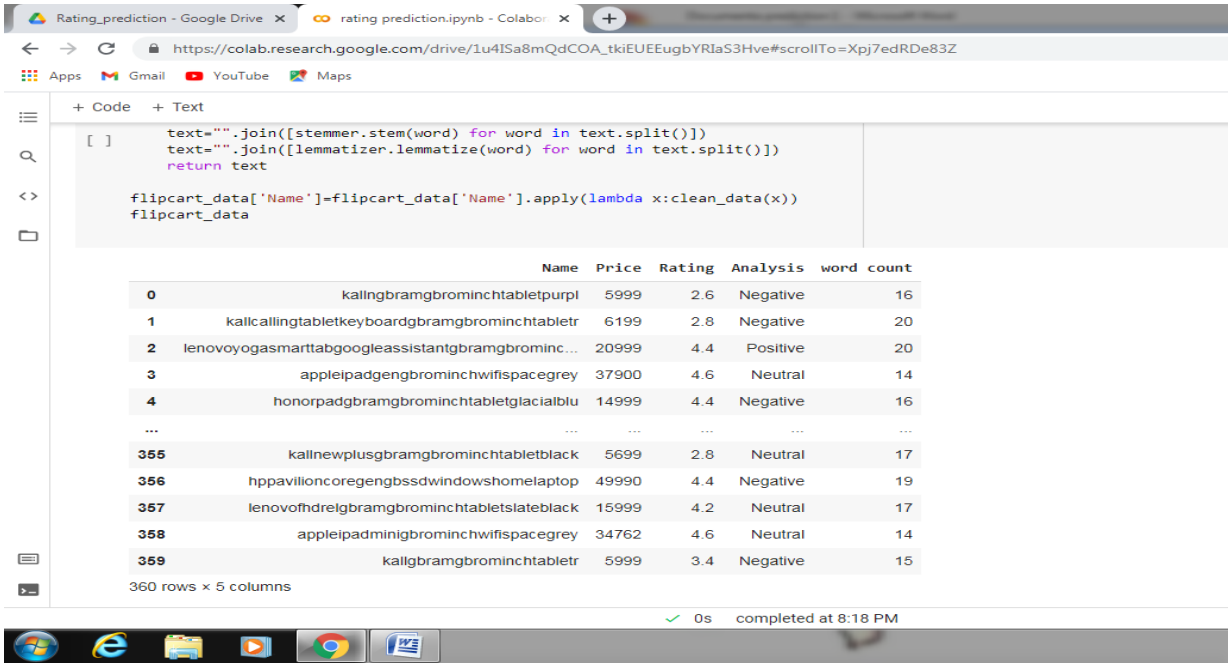
```
[41] from nltk.stem import WordNetLemmatizer
     from nltk.corpus import stopwords
     from nltk.stem.porter import PorterStemmer
     import string
     import nltk
```

```
[43] # get a word count per sentence column
     def word_count(sentence):
         return len(sentence.split())

     flipcart_data['word count'] = flipcart_data['Name'].apply(word_count)
     flipcart_data.head(7)
```

	Name	Price	Rating	Analysis	word count
0	I Kall N 12 2 GB RAM 32 GB ROM 7 inch with Wi-...	5999	2.6	Negative	16
1	I Kall N11 4G Calling Tablet with Keyboard 2 G...	6199	2.8	Negative	20
2	Lenovo Yoga Smart Tab with Google Assistant 4 ...	20999	4.4	Positive	20
3	APPLE iPad (8th Gen) 128 GB ROM 10.2 inch with...	37900	4.6	Neutral	14
4	Honor Pad 5 4 GB RAM 64 GB ROM 8 inch with Wi-...	14999	4.4	Negative	16
5	Lenovo Tab M8 (2nd Gen) HD 2 GB RAM 32 GB ROM ...	9990	4.2	Neutral	19
6	Lenovo Smart Tab M10 FHD Plus (2nd Gen) with G...	20999	4.4	Positive	21

0s completed at 8:18 PM

Figure 14: Before using NLP



```
[ ]     text="".join([stemmer.stem(word) for word in text.split()])
        text="".join([lemmatizer.lemmatize(word) for word in text.split()])
        return text

    flipcart_data['Name']=flipcart_data['Name'].apply(lambda x:clean_data(x))
    flipcart_data
```

	Name	Price	Rating	Analysis	word count
0	kallngbramgbrominchtabletpurpl	5999	2.6	Negative	16
1	kallcallingtabletkeyboardgbramgbrominchtabletr	6199	2.8	Negative	20
2	lenovoyogasmarttabgoogleassistantgbramgbrominc...	20999	4.4	Positive	20
3	appleipadgengbrominchwifispacegrey	37900	4.6	Neutral	14
4	honorpadgbramgbrominchtabletglacialblu	14999	4.4	Negative	16
...	...	...	...	...	...
355	kallnewplusgbramgbrominchtabletblack	5699	2.8	Neutral	17
356	hppavilioncoregengbssdwindowshomelaptop	49990	4.4	Negative	19
357	lenovofhdrelgbramgbrominchtabletslateblack	15999	4.2	Neutral	17
358	appleipadminigbrominchwifispacegrey	34762	4.6	Neutral	14
359	kallgbramgbrominchtabletr	5999	3.4	Negative	15

360 rows × 5 columns

0s completed at 8:18 PM

Figure 15: After using NLP

Model implementation:

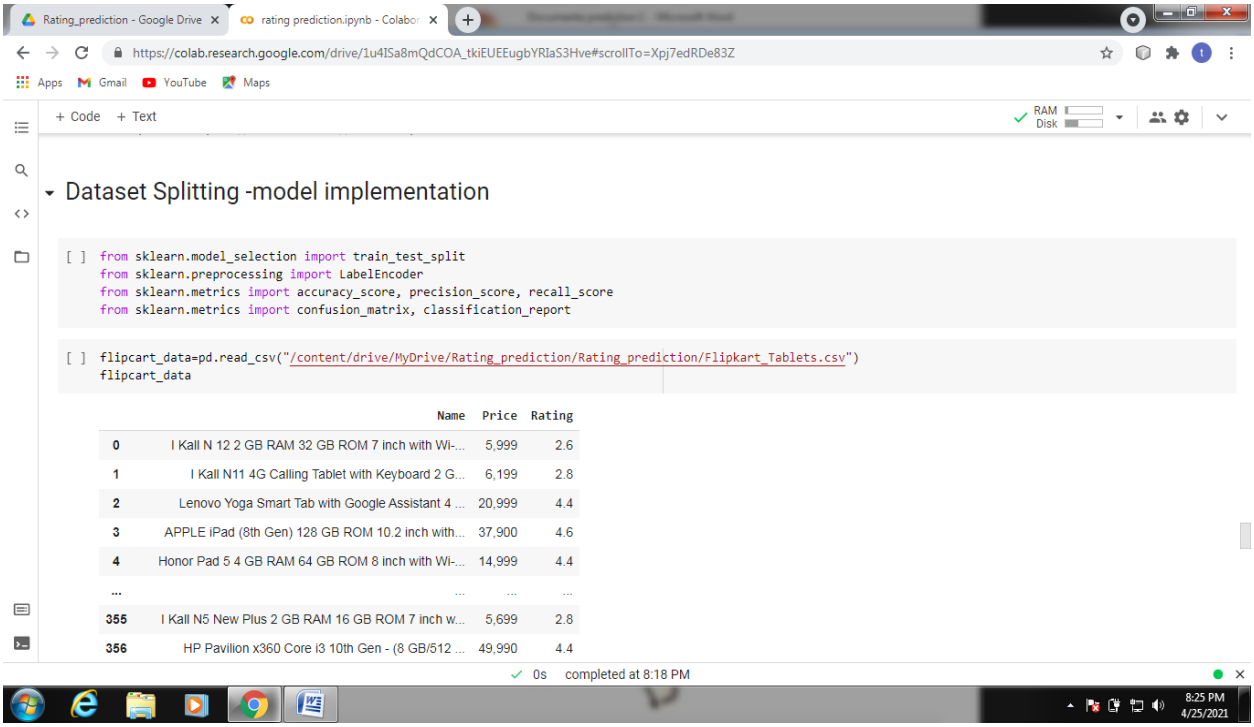


Figure 16: Splitting the dataset

Naive bayes:

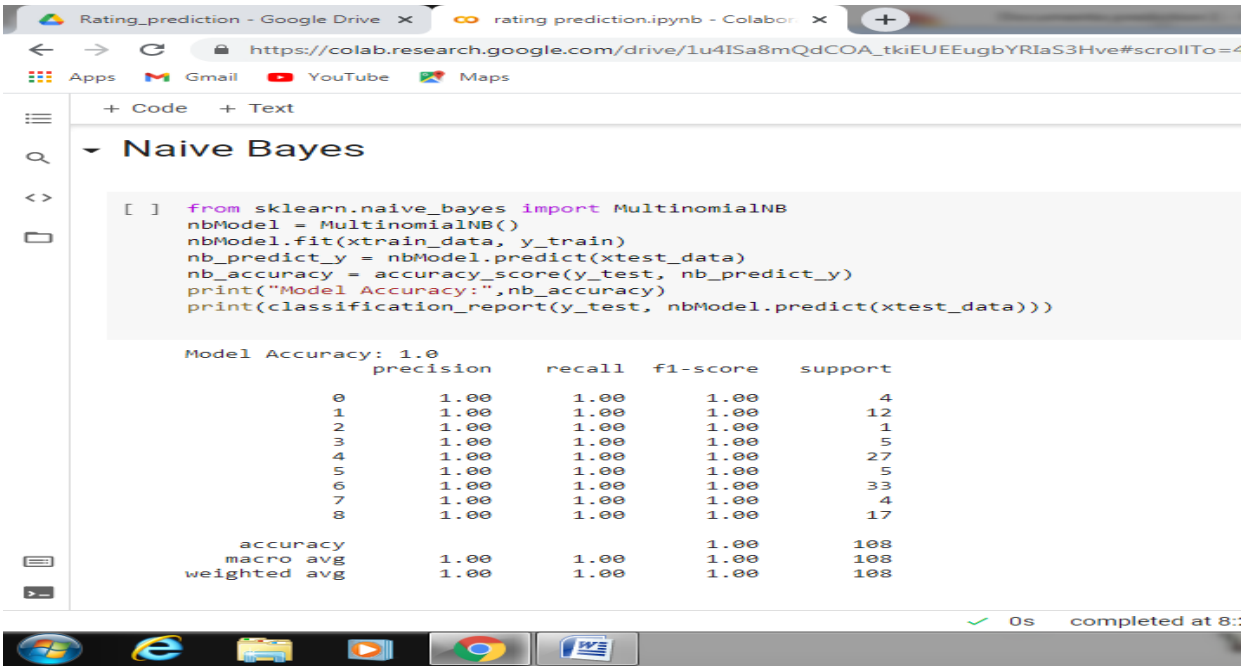


Figure 17: Classification report for Naive bayes accuracy

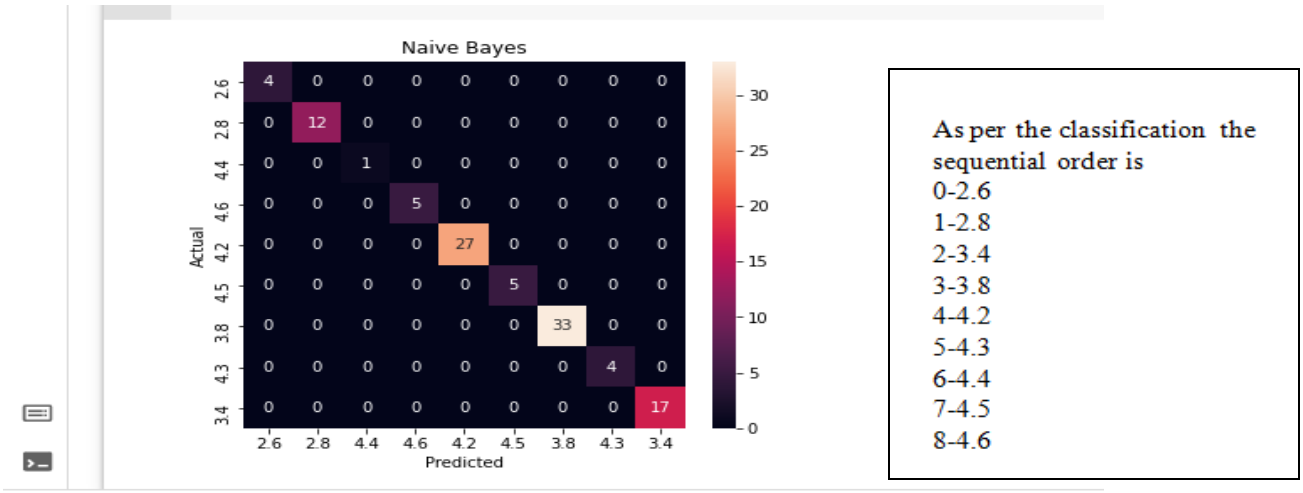


Figure 18: Confusion matrix

Figure 19: Naive bayes prediction

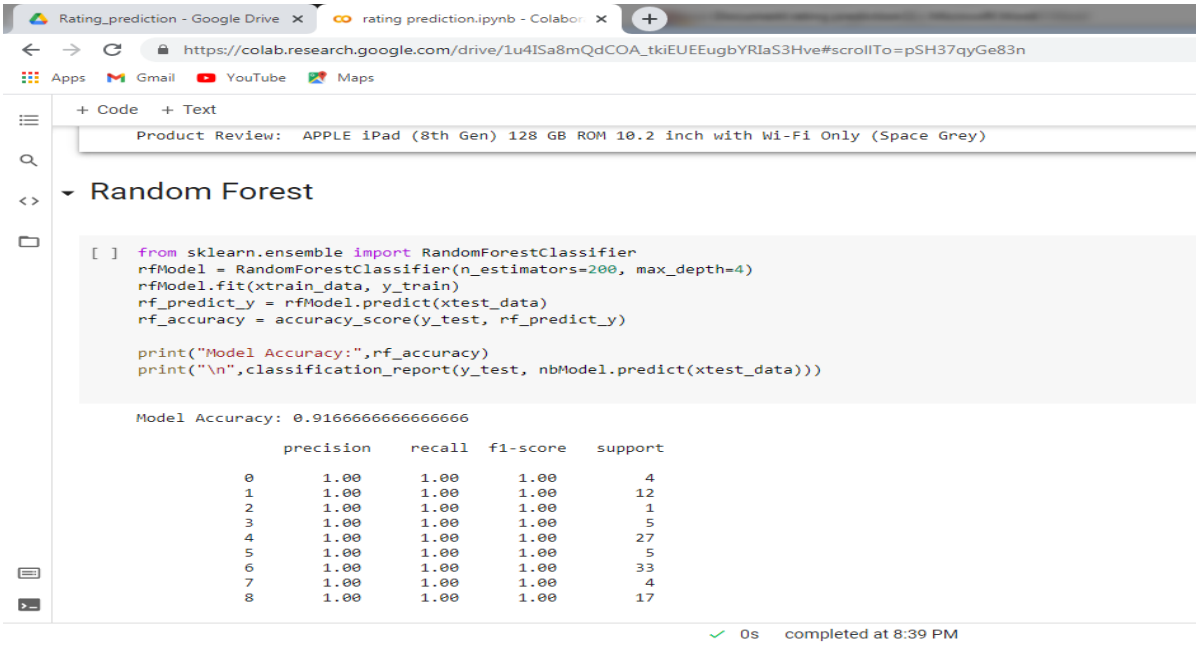


Figure 20: Classification for Random forest accuracy

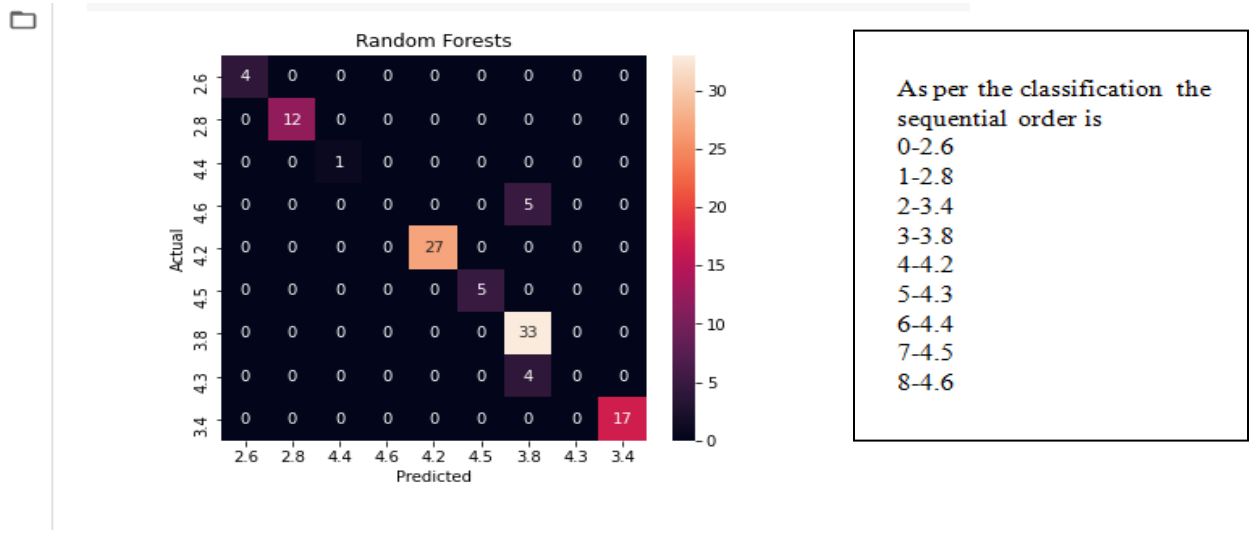


Figure 21: Confusion matrix

Figure 22: Random forest prediction

Sample coding:

```
import requests
from bs4 import BeautifulSoup as bs
URL = 'https://www.flipkart.com/search?q=%20laptop&otracker=search&otracker1=search&ma
rketplace=FLIPKART&as-show=on&as=off'
for page in range(1,10):
 req = requests.get(URL + str(page) + '/')
soup = bs(req.text, 'html.parser')
 main_box = soup.find_all('div',attrs={'class','_2kHMtA'})
 data_list=[]
 for i in range(4,19):
 if page>1:
       for box in main_box:
         temp_dict={}
         temp_dict['Name'] =box.find('div', attrs={'class':"_4rR01T"}).text.strip()
```

```
        temp_dict['Price']=box.find('div', attrs={'class':"_30jeq3 _1_WHN1"}).text.replace("₹","").
strip()
        temp_dict['Rating']=box.find('div', attrs={'class':"_3LWZlK"}).text.strip()
        data_list.append(temp_dict)
mobil=pd.DataFrame(data_list)
mobil =mobil.to_csv('Flipkart_Tablets.csv', index=False, encoding='utf-8')
flipcart_data= pd.read_csv("/content/Flipkart_Laptop.csv")
flipcart_data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
flipcart_data=pd.read_csv("/content/drive/MyDrive/Rating_prediction/Rating_prediction/Flipkar
t_Tablets.csv")
flipcart_data["Price"] =flipcart_data["Price"].astype(int)
type(flipcart_data["Price"])
#polarity calculation
from textblob import TextBlob
def getSubjectivity(tweet) :
   return TextBlob(tweet).sentiment.subjectivity
def getPolarity(tweet) :
   return TextBlob(tweet).sentiment.polarity
def getAnalysis(score) :
 return 'Negative' if (score == 0) else ('Neutral' if (score < 0) else 'Positive')
flipcart_data['Subjectivity'] = flipcart_data['Name'].apply(getSubjectivity)
flipcart_data['Polarity'] = flipcart_data['Name'].apply(getPolarity)
flipcart_data['Analysis'] = flipcart_data['Polarity'].apply(getAnalysis)
flipcart_data
flipcart_data=flipcart_data.drop(['Subjectivity','Polarity'], axis = 1)
flipcart_data.columns
print("Positive:",len(flipcart_data[flipcart_data['Analysis'] == 'Positive']))
print("\nNegative",len(flipcart_data[flipcart_data['Analysis'] == 'Negative']))
print("\nNeutral",len(flipcart_data[flipcart_data['Analysis'] == 'Neutral']))
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import string
import nltk
def word_count(sentence):
 return len(sentence.split())
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import confusion_matrix, classification_report
```

```
x_tain,x_test,y_train,y_test=train_test_split(flipcart_data['Name'], flipcart_data['Rating'], test_size=0.3, random_state=42)
encoder = LabelEncoder()
from sklearn.feature_extraction.text import TfidfVectorizer
data_tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=6000)
data_tfidf_vect.fit(flipcart_data['Name'])
xtrain_data =  data_tfidf_vect.transform(x_tain)
xtest_data =  data_tfidf_vect.transform(x_test)
from sklearn.naive_bayes import MultinomialNB
nbModel = MultinomialNB()
nbModel.fit(xtrain_data, y_train)
nb_predict_y = nbModel.predict(xtest_data)
nb_accuracy = accuracy_score(y_test, nb_predict_y)
print("Model Accuracy:",nb_accuracy)
print(classification_report(y_test, nbModel.predict(xtest_data)))
flipcart_data['Price'] = flipcart_data['Rating'].factorize()[0]
category_id_df = flipcart_data[['Rating', 'Price']].drop_duplicates().sort_values('Price')
category_to_id = dict(category_id_df.values)
id_to_category = dict(category_id_df[['Price', 'Rating']].values)
nb = confusion_matrix(y_test, nbModel.predict(xtest_data))
sns.heatmap(nb, annot=True, xticklabels=category_id_df.Rating.values, yticklabels=category_id_df.Rating.values)
print('Actual Rating:', flipcart_data.iloc[3]["Rating"])
new_text = [flipcart_data.iloc[3]["Name"]]
text_features = data_tfidf_vect.transform(new_text)
nbPredict = nbModel.predict(text_features)
print('\nNaive Bayes Prediction: ', nbPredict[0])
print('\nProduct Review: ', new_text[0])
from sklearn.ensemble import RandomForestClassifier
rfModel = RandomForestClassifier(n_estimators=200, max_depth=4)
rfModel.fit(xtrain_data, y_train)
rf_predict_y = rfModel.predict(xtest_data)
rf_accuracy = accuracy_score(y_test, rf_predict_y)
print("Model Accuracy:",rf_accuracy)
print("\n",classification_report(y_test, nbModel.predict(xtest_data)))
flipcart_data['Price'] = flipcart_data['Rating'].factorize()[0]
category_id_df = flipcart_data[['Rating', 'Price']].drop_duplicates().sort_values('Price')
category_to_id = dict(category_id_df.values)
id_to_category = dict(category_id_df[['Price', 'Rating']].values)
rf_data = confusion_matrix(y_test, rfModel.predict(xtest_data))
plt.title('Random Forests')
plt.ylabel('Actual')
plt.xlabel('Predicted')
```