King Saud University
College of Computer and Information Sciences
Department of Software Engineering

جـــامـــعـــة
الملك سعود
King Saud University
college of computer and information science

Cloud computing and data science

Costa coffee dataset

G#6 I section 54978

#	name	Student ID
1	Reem alshareef	[redacted]
2	Manar Alajmi	[redacted]
3	Lina altammami	[redacted]
4	Alkhansaa AlSultan	[redacted]
5	Reema almutairi	[redacted]
6	Shaden alayed	[redacted]

Supervised by: Dr. Afshan

Phase1 :

1. Project Description:

1.1 Introduction:

The story of Costa Coffee starts in 1971, brothers Sergio and Bruno Costa arrived in London with a passion for bringing great-tasting coffee to the masses. They set up a small roastery in Fenchurch Street, where they blind-tested 112 variations of coffee to create their signature blend, 'Mocha Italia' which remains the company's signature to this day.[1]

Since entering the Middle East coffee shop market in 1999, Costa Coffee has grown to more than 400 stores across 9 countries, Saudi Arabia's coffee shop market has tremendous potential, with a population of 34 million and a positive economic outlook. As consumer demand for premium coffee continues to grow, Costa Coffee is well-positioned to expand its presence in the country, The company currently operates 60 stores in partnership with Jawad Business Group and is pursuing further growth opportunities to meet the expected increase in demand.[2]

The increasing number of coffee-loving customers in Saudi Arabia led us to choose Costa Coffee as our project title. We are curious to know about the customers' opinions and whether there are any shortcomings in the current services.

1.2 initial Hypothesis:

H1: customers tend to love traditional coffee.

H2: customers loves The coffee environment .

H3: customers think that costa coffee prices are expensive.

1.3 Project Objectives:

The aim of this project is to quickly study and understand customer opinions, satisfaction levels, and needs, and satisfy our curiosity about the coffee industry and customers' choices.

2. Project plan

In this section, we will discuss how we plan to obtain the necessary data. We will also discuss which libraries and tools were utilized, and how the data will be stored.

Due to the fact that our goal is to analyze and explore people's opinions about Costa coffee. As a result, Twitter is the most appropriate platform to use, primarily because it allows people to express their thoughts clearly and directly without any doubt. Twitter also provides an easy way to search for relevant conversations and posts. Additionally, it has a wide reach and can be used to engage with a large audience. Furthermore, Twitter is a public platform, and the data is easily accessible. This makes it easy to collect and analyze the data needed for our project.

In order to begin the extraction process on Twitter, we need access to the developers' platform.

Data Extraction Process:

Generally, this is how we intend to describe our data collection process. This process begins with the extraction and collection of data from Twitter about people's opinions about Costa coffee. The following steps must be completed in order to complete the data extraction process:

I. Established a connection with the Twitter API.

II. Python libraries (Tweepy, Pandas, and Numpy) to collect Twitter data.

The collection of data begins after we have set up the Twitter API. Tweepy provides an easy way to connect and retrieve data from the Twitter API. We used the .search_tweets() search method which Tweepy provides, therefore we must use “q” query which will receive a string with the search operator we need, and the string we selected to collect data is “costa coffee”.

We then store the collected data in a Pandas data frame for analysis. Finally, we can visualize the results using Matplotlib to gain insights into Costa Coffee's popularity on Twitter.

III. The data extraction process will include examining the data to ensure that it is reliable and accurate, and most importantly, that it is stored correctly. We can plot the data and explore certain aspects to ensure it aligns with our vision and hypotheses. It is possible to quantify this by plotting tweets over time. Using this method, we will gain a deeper understanding of the data and additional knowledge.

IV. We can keep the data in an a.csv file once we are content with the amount we have collected.

3. Development environment
3.1 Language and IDE:

- **Python**

Python is a high-level, general-purpose programming language with a simple syntax like English.[9] Python’s syntax allows developers to write programs with fewer lines than other programming languages. Often, Python is used as a support language for software developers, build control and management, testing, etc. [11]

- **Jupyter Notebook**

A Project Jupyter document is a JSON file, a server-client application capable of editing and running a notebook. Project Jupyter is a project for open-source software development, for interactive computing services in multiple languages, and open standards development. On other hand, Project Jupyter provides a streamlined, and document-centric experience.[12]

- **Anaconda**

Anaconda is a Python and R programming languages distribution for scientific computing. Anaconda aims to have simple package management and deployment. In addition, anaconda distributions are data-science packages for appropriate Windows, macOS, and Linux.[10]

3.2 Tools:

- Tweepy

We will use Tweepy since our data is mainly collected through twitter. Tweepy is an open-sourced, easy-to-use Python library for accessing the Twitter API. It gives you an interface to access the API from your Python application.[3]

- Pandas

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.[4]

We need to use panda since it is used to analyze data during this project

. • Matplotlib

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. [5]

- NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. [6] It is a commonly used Python package for data analysis since it can speed up the workflow and interface with other packages in Python.

- NLTK

NLTK is a leading platform for building Python programs to work with human language data.[7]

- TextBlob

TextBlob has some predefined rules, or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity.[8]

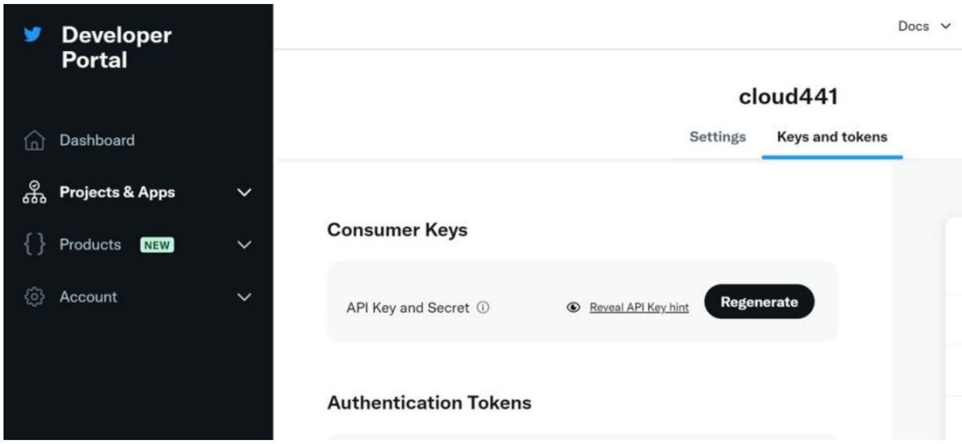
4. Data Collection

In this project, we will use the Twitter API to retrieve some recent public Tweets that match a search query. We have to take a few steps to do this.

1- Getting Access to the Twitter API

We must first create a Twitter developer account before we can use the Twitter API. As a result, we created one to gain access to the Twitter API to obtain credentials.

We created an app on the developer portal to acquire our API keys, Bearer Token, and authentication keys. These are required to connect to the Twitter API v2 endpoints, and we will keep them private.



2- Importing the required libraries:

We used Python 3.11.3 programming language and Jupyter Notebook open-source IDE. In this phase, we needed to import certain libraries which are (Tweepy, Pandas, and Numpy).

```python
import tweepy
import pandas as pd
import numpy as np
```

The first library Tweepy, will help us to access the Twitter API, and the Pandas library will be beneficial for Data analysis, lastly NumPy for handling Numerical values as it makes it easy to apply mathematical functions.

3- Connecting Twitter API to Twitter:

Using the tokens provided before in the Twitter developer portal, we generate a function that establishes a connection with Twitter's API.

```
def TWITTER_SETUP():
    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)

    # Obtain authenticated API
    api = tweepy.API(auth,wait_on_rate_limit= True)
    return api
```

4-Specifying Keywords and Extracting the Tweets:

The word we will use for the search query to retrieve tweets is “costa coffee”.

```
extractor = TWITTER_SETUP()

search_words= 'costa coffee'

tweets = tweepy.Cursor(extractor.search_tweets ,q=search_words,tweet_mode='extended').items(9000)

data = pd.DataFrame(data=[tweet.full_text for tweet in tweets], columns=['Tweets'])

display(data)
```

5- Search Results:

After running the code, we obtained the results.

	Tweets
36	@tonycarmelo15 I remember when a drive through Costa opened near me an…
37	TRADITIONAL / TRADICIONAL ICED COFFEE - World Famous in Costa Ric…
38	Calling All Students: Don't Miss the Youth Opportunities at Costa Coffee Riyad…
39	@Rylan The Costa Coffee song obvs
40	@Rylan I only wanna be with you (Costa coffee remix)
41	RT @AnneLouiseAvery: First trip out for ages and almost immediately a punct…
42	@Rylan The Costa coffee song
43	It's a mocha today cause I was feelin like something sweeter but w/ caffeine. I…
44	#FumingMan #boycotts #Costa #after #buying '#HalfEmpty' #coffee #and #get…
45	The "Repartee" on Twitter is essential. We have to test our views with others. …
46	I think he is in COSTA COFFEE at the moment https://t.co/u8pCSXNAyo
47	@TimChantler @eaglepeaknaod @fiona__jade @Greens4HS2 It isn't, is it? If …
48	Make your break with @CostaCoffee's new KitKat® range 🤩 Tag someone …
49	RT @CropTrust: Great visit from our team to @CATIEOficial headquarters in …
50	Fuming man boycotts Costa after buying 'half empty' coffee and gets apology …
51	First trip out for ages and almost immediately a puncture! Luckily managed to l…
52	RT @lalalolss: Costa Coffee tengah promo, guys! Hot Chocolate RM7.50. Sal…
53	@troublesomerini Felt but costa coffee is banging the cals r high tho
54	@TheQuadFather__ So, when I went to Costa Rica in 2015 I was converted i…
55	RT @lalalolss: Costa Coffee tengah promo, guys! Hot Chocolate RM7.50. Sal…
56	A huge thanks to everyone who has stopped with us, and to our awesome coll…
57	It was nice to meet a couple of friends this afternoon, to have a Costa coffee …
58	@arseblog Was it deliberate to have James's Costa coffee advert at the break…
59	RT @lalalolss: Costa Coffee tengah promo, guys! Hot Chocolate RM7.50. Sal…

6-Saved Retrieved Tweets to CSV file:

the final step was to save the Tweets into CSV file, we name it "Costatwt.csv"

```
data.to_csv("Costatwt.csv", index=False)
```

Phase 2 :

1. Data Exploration

After the data discovery phase, data is prepared and explored prior to modeling and analysis, which includes reformatting data and making corrections to data. It is regarded as the most crucial step. It provides us with an immense amount of information about the data we wish to process. It ensures that the data utilized in analytics gives trustworthy findings and identifies and corrects data errors that would otherwise go undetected.

The total collected reviews is 1049, we run the following code snippets to explore the dataset details and generate general information about:

1. The total costa coffee reviews in the dataset:

```
[4]: len(data)

[4]: 1049
```

Figure 1: total reviews

2. Showing the columns' names included in the data to demonstrate what types of information are included in the data using columns:

```
[17]: data.columns

[17]: Index(['Tweets'], dtype='object')
```

Figure 2:columns name

3. The data type of each column:

```
[21]: data.dtypes

[21]: Tweets    object
      dtype: object
```

Figure 3:data type

4. Showing the count of non-null values contained in each column using count(). Which gives an indication of what attributes are mostly present and which are less present:

```
[6]: data.count()

[6]: Tweets    1049
     dtype: int64
```

Figure 4: count of non-null values

5. Information about the dataset: The function info () in the figure below used to check if there's null values in the columns, figure 5 shows that no null values in the columns.

```
[8]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1049 entries, 0 to 1048
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Tweets  1049 non-null   object
dtypes: object(1)
memory usage: 8.3+ KB
```

Figure 5: info of dataset

2. Data Issues (6 Issues)

Issue1. Duplicate Records

During the collection phase, we discovered that Twitter's API associates each retweeted tweet with the user who retweeted it and treats it as a new tweet with a unique tweet id. This leads in meaningless duplicate tweets. Figures 6 and 7 show an example of duplicated tweets as well as the total number of duplicated tweets in our dataset.

```
[9]: display(sum(data.duplicated()))

151
```

Figure 6: number of duplicated

```
[10]: #find dublicated row with the same value in 'Tweets'
      display(data[data.duplicated(subset='Tweets')].head())
```

	Tweets
4	RT @MaximMag: The breathtaking seaside escape ...
6	RT @MaximMag: The breathtaking seaside escape ...
7	RT @MaximMag: The breathtaking seaside escape ...
9	RT @MaximMag: The breathtaking seaside escape ...
10	RT @MaximMag: The breathtaking seaside escape ...

Figure 7: example of duplicated

We notice that retweeted tweets begin with “RT” so in figure 8 below we are removing tweets that start with “RT”.

```
[25]: data=data[data['Tweets'].str.contains('RT')==False]

      display(data[data['Tweets'].str.contains('RT')])
```

Tweets

Figure 8: remove RT from tweets

Although we removed tweets that begin with "RT" the dataset still shows that there are more duplicates. As figures 9 show

```
[12]: display(sum(data.duplicated()))

10
```

Figure 9: more duplicates

We removed duplicated tweets using drop_duplicates function from pandas library and we can see in figure 10 that the count of duplicated tweets has become 0.

```
[13]: data.drop_duplicates(subset='Tweets',inplace=True)

[14]: display(sum(data.duplicated()))

0
```

Figure 10: fully removal of duplicates

Issue2. (Links, Mentions, Hashtags):

re (Regular expression) Libraries is used in solving this issue

During preprocessing, hyperlinks, mentions, and hashtags are removed from tweets

Using regular expressions.

1) The first step is to remove hashtags from tweets and convert hashtag names into normal text. As an example, if a tweet contains the following text(#Costa), the returned value will be (Costa).

2) Secondly, tweets should be free of hyperlinks. For example, (textbody) will be returned as the value of a tweet that contains (https://www.costa.com/ textbody)

3) In the third step, mentions are removed and the content of tweets is converted into normal text. It will return the value (textbody) if a tweet contains (@costa textbody).

```
[122]: def cleantwt1 (twt):

         twt = re.sub('#[A-Za-z0-9]+', '', twt) # remove the '#' from the tweets
         twt = re.sub('https?:\/\/\S+', '', twt) # remove the hyperlinks
         twt = re.sub('@[\S]*', '', twt) # remove @mentions
         return twt

[123]: data['CleanedTweets'] = data['Tweets'].apply(cleantwt1)

[124]: data.head()
```

[124]:

	Tweets	CleanedTweets
0	@chrisdysonHT @SafeSENCOSaeed @BrightLeadChris...	We should launch a kindness c...
1	#Costa use my link to treat us both xx\n\nUse ...	use my link to treat us both xx\n\nUse the Co...
5	After the release of a new video of costa coff...	After the release of a new video of costa coff...
8	@ahyani99 😍😍 مع هشام باشا مبسوط أوي (@ Costa C...	مع هشام باشا مبسوط أوي 😍😍 (Costa Coffee | Dr...
36	@ChronicleLive Never use Costa or any of the o...	Never use Costa or any of the other coffee sh...

Figure 11: the method of removing hashtags, hyperlinks and mentions and the result.

Issue3. (emojis):

The Python library was utilized to enable the use of emojis in our project. Emojis have been used in several tweets to convey users' emotions. However, they can sometimes be misleading and give inaccurate indications. Therefore, we made the decision to remove them.

Figure 12 shows the the function we used to remove Emojis, a random sample before removing the emojis, and the same sample after removing the emojis.

```
[27]: def remove_Emojis(twt):
          twt = re.sub('[/\W+/g]',' ', twt)
          return twt

[30]: data['remove_Emojis'] = data['CleanedTweets'].apply(remove_Emojis)

[31]: data.head()
```

[31]:

	Tweets	CleanedTweets	remove_Emojis
0	@chrisdysonHT @SafeSENCOSaeed @BrightLeadChris...	We should launch a kindness c...	We should launch a kindness c...
1	#Costa use my link to treat us both xx\n\nUse ...	use my link to treat us both xx\n\nUse the Co...	use my link to treat us both xx Use the Cost...
5	After the release of a new video of costa coff...	After the release of a new video of costa coff...	After the release of a new video of costa coff...
8	@ahyani99 😍😍 مع هشام باشا مبسوط أوي (@ Costa C...	مع هشام باشا مبسوط أوي 😍😍 (Costa Coffee | Dr...	مع هشام باشا مبسوط أوي Costa Coffee Dr...
36	@ChronicleLive Never use Costa or any of the o...	Never use Costa or any of the other coffee sh...	Never use Costa or any of the other coffee sh...

Figure 12: the method of removing Emojis and the result.

Issue4. (Foreign words)

First, we need to download the re Python library and import it to support removing foreign words. Then, we can use it with a function that takes a review as input and returns the review after removing foreign words as shown in the following figures.

In [62]:
```
#issue 4 remove non-english world
# Define a regular expression pattern to match English letters
english_pattern = re.compile('[a-z A-Z]+')

# Define a function to remove non-English letters from a string
def remove_non_english_letters(text):
    return ''.join([char for char in text if english_pattern.match(char)])

# Apply the function to all elements in the DataFrame
data['remove_nonEnglish'] = data['remove_Emojis'].apply(remove_non_english_letters)
```

In [63]:
```
data.head()
```

Out[63]:

	Tweets	CleanedTweets	remove_Emojis	remove_nonEnglish
0	@chrisdysonHT @SafeSENCOSaeed @BrightLeadChris...	We should launch a kindness c...	We should launch a kindness c...	We should launch a kindness c...
1	#Costa use my link to treat us both xx\n\nUse ...	use my link to treat us both xx\n\nUse the Co...	use my link to treat us both xx Use the Cost...	use my link to treat us both xx Use the Cost...
5	After the release of a new video of costa coff...	After the release of a new video of costa coff...	After the release of a new video of costa coff...	After the release of a new video of costa coff...
8	مع هشام باشا مبسوط اوي @ahyani99 😍😍 (@ Costa C...	مع هشام باشا مبسوط اوي 😍😍 (Costa Coffee | Dr...	مع هشام باشا مبسوط اوي Costa Coffee Dr...	Costa Coffee Drive Thru in Mecc...
36	@ChronicleLive Never use Costa or any of the o...	Never use Costa or any of the other coffee sh...	Never use Costa or any of the other coffee sh...	Never use Costa or any of the other coffee sh...

Figure 13: the method of removing foreign word (non-English) and the result.

Issue5. (Special Characters)

We used Python library to remove the most common special characters such as underscore, hashtags, and "@". This library should be downloaded first. After downloading the library, we import it. Also, we use "replace" to remove lines, tabs, and underscore and prefix on string, and replace it with empty string.

```
[64]: def remove_unrelated_chars(Tweets):
          Tweets = re.sub('{-}','',Tweets)
          Tweets = re.sub('#([^\s]+)',r'\\u003c\\1',Tweets)
          Tweets = re.sub('@([^\s]+)',r' ',Tweets)
          Tweets = re.sub('\n',' ',Tweets)
          Tweets = re.sub('\t',' ',Tweets)
          Tweets = re.sub('\r',' ',Tweets)
          return Tweets
```

```
[65]: data['remove_unrelated_chars'] = data['remove_nonEnglish'].apply(remove_unrelated_chars)
```

```
[66]: data.sample(20)
```

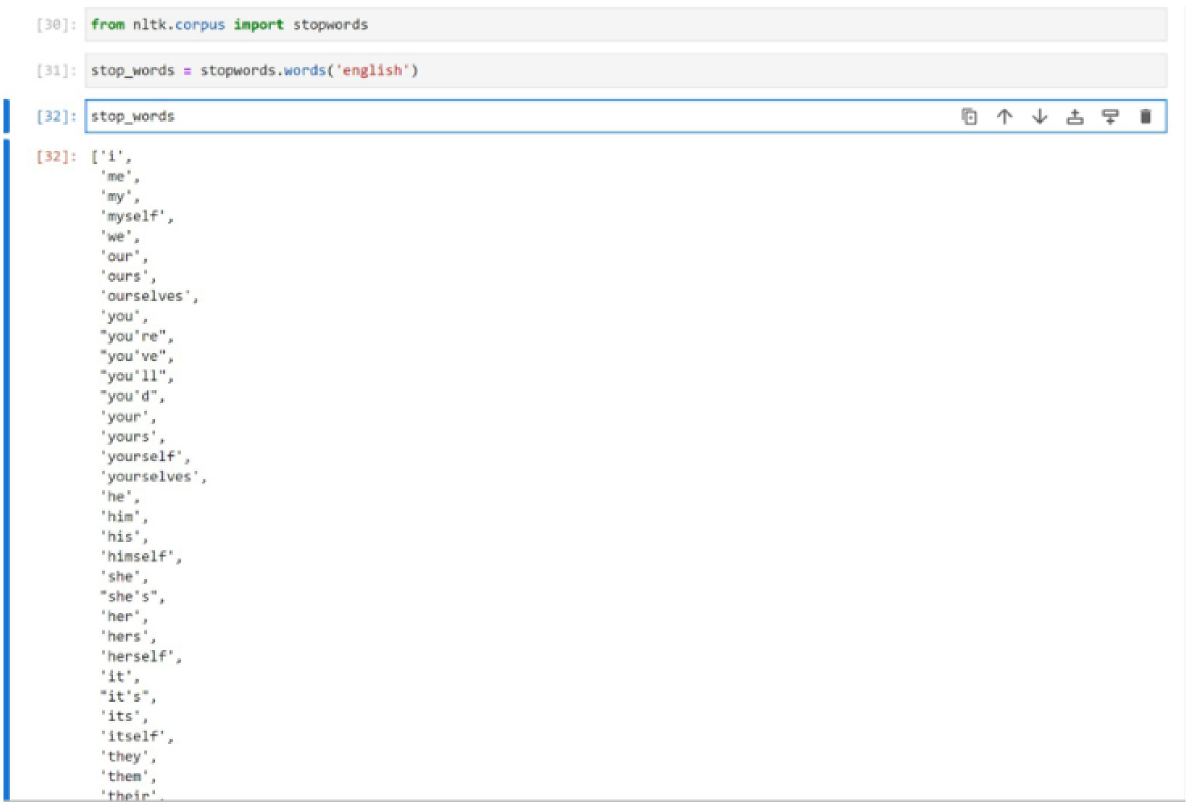
[66]:

	Tweets	CleanedTweets	remove_Emojis	remove_nonEnglish	remove_unrelated_chars
278	The menu is available now 😋 \nhttps://t.co/4YUz...	The menu is available now 😋\n	The menu is available now	The menu is available now	The menu is available now
202	@flloydthecat Hey, I want to assure you animal...	Hey, I want to assure you animal welfare is a...	Hey I want to assure you animal welfare is a...	Hey I want to assure you animal welfare is a...	Hey I want to assure you animal welfare is a...
667	Costa Coconut Coffee / Tea - Zhou Ye PH GO\n\n...	Costa Coconut Coffee / Tea - Zhou Ye PH GO\n\n...	Costa Coconut Coffee Tea Zhou Ye PH GO ...	Costa Coconut Coffee Tea Zhou Ye PH GO ...	Costa Coconut Coffee Tea Zhou Ye PH GO ...
702	PEANUT BUTTER ICED COFFEE - World Famous in Co...	PEANUT BUTTER ICED COFFEE - World Famous in Co...	PEANUT BUTTER ICED COFFEE World Famous in Co...	PEANUT BUTTER ICED COFFEE World Famous in Co...	PEANUT BUTTER ICED COFFEE World Famous in Co...
140	@sploshmedia @CostaCoffee @Customerservace bui...	building, not owned by Costa, has leaking r...	buildin not owned by Costa has leakin r...	buildin not owned by Costa has leakin r...	buildin not owned by Costa has leakin r...

Figure 13: the method of removing special character and the dataset before and after removing

Issue6. (Stop Words)

The stop words are a list of words that are very common but don't provide helpful information for most text analysis procedures. For example, some NLP tasks do not offer additional or valuable information to the text containing them. In addition, the stop words in **NLTK** are the most common in data.
Here, while writing a list of stop words provided by the **NLTK** library. By default, we won't have to define every stop word manually.
Since the stop words don't give a piece of valuable information, we need to define a function to remove stop words from the tweets, as you can see in Figure 14.



```
[30]: from nltk.corpus import stopwords

[31]: stop_words = stopwords.words('english')

[32]: stop_words

[32]: ['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
```

Figure 14: identifying stop of words with NLTK library.

```
[37]: data['remove_stopwords'] = data['remove_unrelated_chars'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop_words)]))

[39]: data.sample(10)
```

[39]:

	Tweets	CleanedTweets	remove_Emojis	remove_nonEnglish	remove_unrelated_chars	remove_stopwords
684	セブンイレブン(@711SEJ)様より\n\n当選したCOSTA COFFEE を引き換えま...	セブンイレブン(\n\n当選したCOSTA COFFEE を引き換えました！♥\n\nプレ...	セブンイレブン 当選した COSTA COFFEE を引き換えました プレミアムラ...	COSTA COFFEE	COSTA COFFEE	COSTA COFFEE
604	Hello free coffee! \n\nGet a quick buildi...	Hello free coffee! \n\nGet a quick buildi...	Hello free coffee Get a quick buildin ...	Hello free coffee Get a quick buildin ...	Hello free coffee Get a quick buildin ...	Hello free coffee Get quick buildin contents i...
493	i will say he's so real for the costa coffee tho!	i will say he's so real for the costa coffee tho!	i will say he s so real for the costa coffee tho	i will say he s so real for the costa coffee tho	i will say he s so real for the costa coffee tho	say real costa coffee tho
653	Developers have withdrawn their planning appli...	Developers have withdrawn their planning appli...	Developers have withdrawn their plannin appli...	Developers have withdrawn their plannin appli...	Developers have withdrawn their plannin appli...	Developers withdrawn plannin application witho...
514	Felt like having a coffee asked my older sis i...	Felt like having a coffee asked my older sis i...	Felt like havin a coffee asked my older sis i...	Felt like havin a coffee asked my older sis i...	Felt like havin a coffee asked my older sis i...	Felt like havin coffee asked older sis wanted ...
127	Now till 15 May 2023: Costa Coffee London Almo...	Now till 15 May 2023: Costa Coffee London Almo...	Now till 15 May 2023 Costa Coffee London Almo...	Now till May Costa Coffee London Almond Dri...	Now till May Costa Coffee London Almond Dri...	Now till May Costa Coffee London Almond Drinks...
440	【ラウンジTIMEサウス】福岡空港国内線カードラウンジ COSTA COFFEE 保安検査場...	【ラウンジTIMEサウス】福岡空港国内線カードラウンジ COSTA COFFEE 保安検査場...	ラウンジTIMEサウス 福岡空港国内線カードラウンジ COSTA COFFEE 保安検査場...	TIME COSTA COFFEE	TIME COSTA COFFEE	TIME COSTA COFFEE
979	https://t.co/Duus5XsORA\nThe Costa Rica Good N...	\nThe Costa Rica Good News Report\nEnjoy some ...	The Costa Rica Good News Report Enjoy some GO...	The Costa Rica Good News Report Enjoy some GO...	The Costa Rica Good News Report Enjoy some GO...	The Costa Rica Good News Report Enjoy GOOD NEW...
54	@AgingWhiteGay Starbucks UK rewards system is ...	Starbucks UK rewards system is as crappy as t...	Starbucks UK rewards system is as crappy as t...	Starbucks UK rewards system is as crappy as t...	Starbucks UK rewards system is as crappy as t...	Starbucks UK rewards system crappy

Figure 15 shows the previous tweet's vs the new ones by using removing stop words method.

Hello Could you please tell me if there is a complaints procedure for Costa I…	Hello Could please tell complaints procedure Costa I prepared bel…
Take a coffee journey and discover Costa Rica s most valuable export with on…	Take coffee journey discover Costa Rica valuable export one bari…
And there better not be one of them Costa coffee machines in there or it s ov…	And better one Costa coffee machines
Thank you notes Costa Rican coffee Malaysian tea and now this beautiful E…	Thank notes Costa Rican coffee Malaysian tea beautiful E yptian …
Summer has arrived at Costa Coffee with their new whipped coffee ran e Ch…	Summer arrived Costa Coffee new whipped coffee ran e Choose …
To say the least We never know how one wants to be treat like better find a…	To say least We never know one wants treat like better find educa…
Costa Rica superb wild life and coffee	Costa Rica superb wild life coffee
Ripenin coffee cherries in the Tarraz rowin re ion of Costa Rica ori in of our…	Ripenin coffee cherries Tarraz rowin ion Costa Rica ori Guanacas…
For otten just how nice Costa soya mocha coffee with tiffin is Well it s almost…	For otten nice Costa soya mocha coffee tiffin Well almost birthday…
Get a quote for Police Mutual Car Insurance by May and you can rab a trea…	Get quote Police Mutual Car Insurance May rab treat us Choose …
Starbucks coffee is awful as is costa and Nero Prefer smaller coffee shops …	Starbucks coffee awful costa Nero Prefer smaller coffee shops sel…
Cant wait to land in birmin ham and buy an overpriced coffee at the airport co…	Cant wait land birmin ham buy overpriced coffee airport costa like…
Well Lobby overnment instead of sycophantic social media Government m…	Well Lobby overnment instead sycophantic social media Govern…
Day amp in Paradise Our roup experienced Selvatura Park Cloud Forest…	Day amp Paradise Our roup experienced Selvatura Park Cloud F…
my actual lastname is costa so it always tricks me rememberin that it s a wh…	actual lastname costa always tricks rememberin whole coffee chain
costas creamin money in new delhi bbc do a pod cast please	costas creamin money new delhi bbc pod cast please
every office worker thinks woo a bank holiday reat i ll o to the open super…	every office worker thinks woo bank holiday reat open supermark…
I suspect a lot of it is nepotism Alberta and unscrupulous climbers who fou…	I suspect lot nepotism Alberta unscrupulous climbers found easier…
Midday snack at Costa Coffee in Yate	Midday snack Costa Coffee Yate
Use the Costa app like me and et bean to start then more with your first pu…	Use Costa app like et bean start first purchase Costa handcrafted…
The worst thin to ever happen to me was to move near a costa drive thru Ju…	The worst thin ever happen move near costa drive thru Just took …
my dau hter just placed an order on the app only to arrive at the Costa Coffe…	dau hter placed order app arrive Costa Coffee shop closed How e…
Summer has arrived at Costa Coffee with our new whipped coffee ran e Cho…	Summer arrived Costa Coffee new whipped coffee ran e Choose …
My local pharmacy was just replaced with a Costa coffee shop	My local pharmacy replaced Costa coffee shop

Figure 16: After removing stope words.

Phase 3:

3.1. sentiment analysis:

Sentiment analysis (also known as opinion mining) is a type of natural language processing (NLP) approach that determines whether data is positive, negative, or neutral. Sentiment analysis on textual data is frequently used to assist organizations in monitoring brand and product sentiment in consumer feedback and understanding customer demands.

Using the TextBlob library, which is an open-source Python library for text processing. It provides a straightforward API for accessing its operations and doing basic NLP activities. TextBlob is a textual data processing tool that we made use of along with nltk sentiment vader to perform sentiment analysis on the data we cleaned.

TextBlob returns a sentence's polarity and subjectivity. Polarity is defined as [-1,1], where -1 represents a negative sentiment and 1 represents a positive sentiment.

The first step was to implement the required libraries

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
import nltk
```

Figure 17: libraries used.

The next step was to define a function that calculates subjectivity, polarity and classify weather the sentence is negative, positive or neutral

```python
#Calculating Negative, Positive, Neutral and Compound values

data[['polarity', 'subjectivity']] = data['remove_stopwords'].apply(lambda Text: pd.Series(TextBlob(Text).sentiment))
for index, row in data['remove_stopwords'].iteritems():
    score = SentimentIntensityAnalyzer().polarity_scores(row)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']
    comp = score['compound']
    if comp <= -0.05:
        data.loc[index, 'sentiment'] = "negative"
    elif comp >= 0.05:
        data.loc[index, 'sentiment'] = "positive"
    else:
        data.loc[index, 'sentiment'] = "neutral"
    data.loc[index, 'neg'] = neg
    data.loc[index, 'neu'] = neu
    data.loc[index, 'pos'] = pos
    data.loc[index, 'compound'] = comp

data.head(20)
```

Figure 18: defining sentences.

After running the code, we obtained the results

remove_stopwords	polarity	subjectivity	sentiment	neg	neu	pos
We launch kindness campai n John Ma ee try et ...	0.400000	0.800000	positive	0.000	0.690	0.310
use link treat us xx Use Costa app like et bea...	0.325000	0.566667	positive	0.000	0.691	0.309
After release new video costa coffee empoyin c...	0.136364	0.454545	neutral	0.000	1.000	0.000
Costa Coffee Drive Thru Mecca	0.000000	0.000000	neutral	0.000	1.000	0.000
Never use Costa coffee shop always use local c...	0.200000	0.500000	positive	0.000	0.654	0.346
Eh need say orderin coffee Costa West coasters...	0.000000	0.000000	neutral	0.000	1.000	0.000
COSTA COFFEE	0.000000	0.000000	neutral	0.000	1.000	0.000
A Costa coffee shop shuttin town woolworths sh...	-0.400000	0.700000	positive	0.071	0.827	0.102
customer costa rican coffee know read customer...	-0.400000	0.850000	negative	0.174	0.725	0.101
If people want coffee pub Nescafe Instant noth...	0.000000	0.666667	positive	0.000	0.874	0.126

Figure 19: results of the code.

3.2 descriptive analysis:

Descriptive analysis attempts to characterize or summarize past and present data, hence assisting in the creation of data insights. It is the act of utilizing statistical methods to clarify, illustrate, or summarize data points in such a way that patterns emerge that satisfy all the data's conditions. It also allows us to detect commonalities between variables, preparing us for further statistical analysis.
-Importing libraries: We imported the important libraries.

```
import pandas as pd
import numpy as np
```

Figure 20: libraries used.

-Identify shape, column, and summary of the data frame
The shape of the data frame has 797 rows and 5 column which are cleanTweets, sentiment which is the textual classification, and the neg, neu, pos for the numeric values.

```
data2.shape
```

```
(797, 5)
```

```
data2.columns
```

```
Index(['cleanTweets', 'sentiment', 'neg', 'neu', 'pos'], dtype='object')
```

Figure 21: shape and columns of the data.

- For the summary we used info() function to display a summary of the Data Frame that contains number dtypes and columns number and info:

```
data2.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 797 entries, 0 to 1048
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   cleanTweets  797 non-null    object
 1   sentiment    797 non-null    object
 2   neg          797 non-null    float64
 3   neu          797 non-null    float64
 4   pos          797 non-null    float64
dtypes: float64(3), object(2)
memory usage: 69.6+ KB
```

Figure 22: summary of the data frame .

-Use describe method
To calculate mean, IQR values and std for the numeric columns we will use. describe () function from pandas:

```
data2.describe()
```

	neg	neu	pos
count	797.000000	797.000000	797.000000
mean	0.050105	0.815262	0.125842
std	0.113575	0.196446	0.159201
min	0.000000	0.000000	0.000000
25%	0.000000	0.688000	0.000000
50%	0.000000	0.850000	0.048000
75%	0.000000	1.000000	0.231000
max	0.649000	1.000000	0.730000

Figure 23: results of the function describe() for numeric data.

```
data2.describe(include=['object'])
```

	cleanTweets	sentiment
count	797	797
unique	696	3
top	COSTA COFFEE	positive
freq	25	350

Figure 24: results of the describe() function for the objects data .

- Use count value in column() method
We used the function "count value in column" which is a pandas function which returns objects containing counts of classification values "sentiment", and from this measurement we found that positive is the most frequently occurring element, then neutral and the least frequently occurring element is negative.

```
#Function for count_values_in single columns

def count_values_in_column(data,feature):
    total=data.loc[:,feature].value_counts(dropna=False)
    percentage=round(data.loc[:,feature].value_counts(dropna=False,normalize=True)*100,2)
    return pd.concat([total,percentage],axis=1,keys=['Total','Percentage'])
```

```
#Count_values for sentiment
count_values_in_column(data,"sentiment")
```

	Total	Percentage
positive	350	43.91
neutral	320	40.15
negative	127	15.93

Figure 25: count of sentiment values classification.

3.3 predictive analysis:

We have chosen these two models, Naïve Baye which Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems

and Logistic Regression
which is A classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud

We will explain the details of the implementation of each model below, and we chose a set of techniques that helped us evaluate them, which are
Ten-Folds Cross Validation
ROC Curve
Confusion Matrix

Evaluation techniques:

Ten-Folds Cross Validation:

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.
It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.
The general procedure is as follows:

6. Shuffle the dataset randomly.
7. Split the dataset into k groups
8. For each unique group:
    a. Take the group as a hold out or test data set
    b. Take the remaining groups as a training data set
    c. Fit a model on the training set and evaluate it on the test set
    d. Retain the evaluation score and discard the model
9. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

ROC Curve:

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.
The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

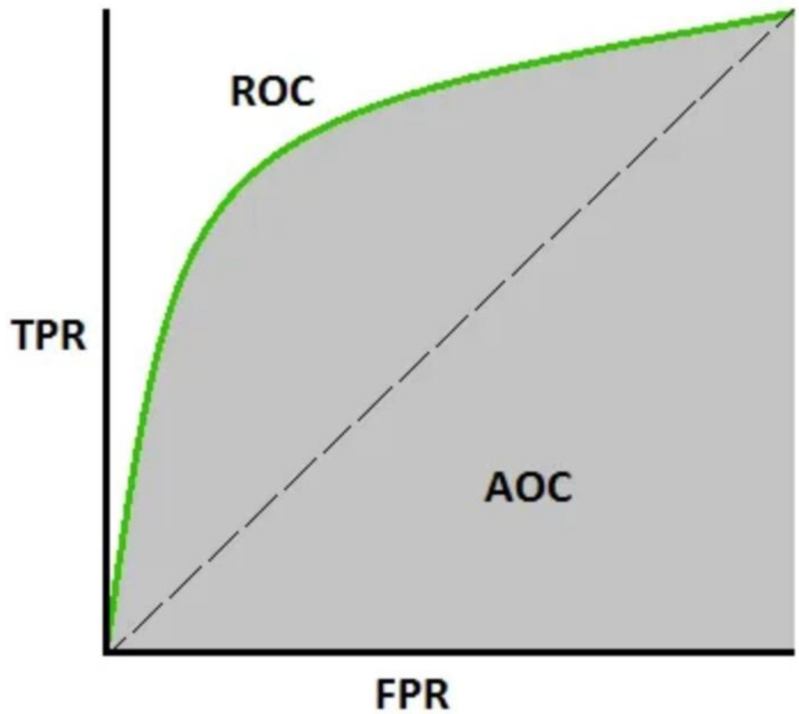


Figure 26: ROC curve.

TPR (True Positive Rate) / Recall /Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Image 3

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

Image 4

FPR

$$\text{FPR} = 1 - \text{Specificity}$$
$$= \frac{FP}{TN + FP}$$

Image 5

Figure 27: Formulas of TPR, specificity and FPR.

Confusion Matrix:

Confusion matrix is a very popular measure used while solving classification problems. it was utilized for the performance evaluations of the methods used after the classification. Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated based on the above-stated TP, TN, FP, and FN. Accuracy of an algorithm is represented as the ratio of correctly classified patients (TP+TN) to the total number of patients (TP+TN+FP+FN). Accuracy=(TP+TN) (TP+FP+FN+TN). Precision of an algorithm is represented as the ratio of correctly classified patients with the disease (TP) to the total patients predicted to have the disease (TP+FP). Recall metric is defined as the ratio of correctly classified diseased patients (TP) divided by total number of patients who have the disease. Recall=TPTP+FN

3.3.1 model 1 (Naïve bayes model)

Several libraries were used, including pandas, numpy, sklearn, and matplotlib

A Naive Bayes model is a type of machine learning algorithm that focuses on classification. In a naive Bayes model, the predictor variables are assumed to be independent of one another based on a statistical classification technique called the Bayes Theorem. The simplicity of these models makes it very easy for a novice to build accurate models with very good performance. In other words, the presence of one feature in a dataset has no link to the presence of another feature.

Balanced and unbalanced dataset

We start by dropping data with NAN values along with the "Neutral" class sentiment.

```
len (data_df1 [data_df1['cleanTweets']=='negative'])
0

len (data_df1 [data_df1['cleanTweets']=='positive'])
0

data_df1.drop(data_df1.columns[data_df1.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True

# remove data with NAN sentiment
data_df1=data_df1[~data_df1["sentiment"].isna()]
data_df1=data_df1[~data_df1["cleanTweets"].isna()]

# remove the "Neutral" class
data_df1=data_df1[data_df1['sentiment'] != "neutral"]
```

Figure28: dropping NAN and Natural records.

Then, we converted the sentiment column from categorical to numerical data type; where 0 represents "Negative" and 1 represents "Positive".

```
# change values to numeric
data_df1['sentiment'] = data_df1['sentiment'].map({'negative' : 0, 'positive' : 1})
```

	cleanTweets	sentiment
1	It effin awful Welp uess money saved Cause I NOT onna est hot arba e servin I ot authentic bold Costa Rican coffee peekin cupboard	0
2	The initial application last year refused	0
3	I need et Costa Coffee I try bubble tea inspired frappes	1
5	Like beans et low hopper order Costa Rica Tarrazu San Die beans Quinnin Coffee	0
8	Costa app invite code free coffee cake	1

Figure29: changing type of data from categorial to numerical.

Then, we separated the dataset into features and targets, which contain sentiments and reviews, respectively. X represents Features whereas Y represents the target value which is the classification we are predicting.

```
# idneitfy the data and the labels
X= data_df1['cleanTweets']
y= data_df1['sentiment']
```

Figure30: separating the data into features and target.

To digest and deal with data, models require a certain type of data. Our textual data must therefore be vectorized. The unbalance in our classes most likely lead to overfitting despite all of the preprocessing we have done. Our dataset appears biased because the negative class is underrepresented. As a result, our models will perform poorly. To illustrate the impact of balanced data and the impact it makes on the models' performance, we worked on both cases – balanced and unbalanced - each with its own training and testing data.
The problem would not be entirely solved even after resampling the unbalanced dataset. For now, we have converted and split our unbalanced dataset:

```
# Use TfidfVectorizer for feature extraction (TFIDF to convert textual data to numeric form):
# Convert to a vector representation
unbalanced_tfidf = TfidfVectorizer()
unbalanced_X = unbalanced_tfidf.fit_transform(X)
unbalanced_X.shape

(476, 2931)

unbalanced_X

<476x2931 sparse matrix of type '<class 'numpy.float64'>'
	with 7648 stored elements in Compressed Sparse Row format>
```

Figure31: transform textual data.

preparing the dataset for training, by splitting it into 70% Training – 30% Testing.

```
anced, X_test_balanced, y_train_balanced, y_test_balanced = train_test_split(X,y, test_size=0.3, random_state=

print("Training set has {} samples.".format(X_train_balanced.shape[0]))
print("Testing set has {} samples.".format(X_test_balanced.shape[0]))

Training set has 333 samples.
Testing set has 143 samples.
```

Figure32: Split raw training and testing records

. Then as we did to unbalanced, we did to balancing the other training and testing dataset. We started with textual data conversion. Afterwards, we split the dataset into training and testing sets. Then, we performed a dataset transformation.

```
balanced_tfidf = TfidfVectorizer()
balanced_tfidf.fit(X)

▾ TfidfVectorizer
TfidfVectorizer()

anced, X_test_balanced, y_train_balanced, y_test_balanced = train_test_split(X,y, test_size=0.3, random_state=

print("Training set has {} samples.".format(X_train_balanced.shape[0]))
print("Testing set has {} samples.".format(X_test_balanced.shape[0]))

Training set has 333 samples.
Testing set has 143 samples.

X_train_balanced =balanced_tfidf.transform(X_train_balanced)
X_test_balanced = balanced_tfidf.transform(X_test_balanced)
balanced_X = balanced_tfidf.transform(X)

X_train_balanced

<333x2931 sparse matrix of type '<class 'numpy.float64'>'
	with 5315 stored elements in Compressed Sparse Row format>
```

Figure33: Transform textual data and split the dataset

To show the difference between unbalanced and balanced datasets, the following outputs of the following code segments.

```
print("Data before balence: {} samples.".format(y_train_unbalanced.value_counts()[0]))
print("Data before balance: {} samples.".format(y_train_unbalanced.value_counts()[1]))

Data before balence: 83 samples.
Data before balance: 250 samples.

print("Data after balance: {} samples.".format(y_train_balanced.value_counts()[0]))
print("Data after balance: {} samples.".format(y_train_balanced.value_counts()[1]))

Data after balance: 250 samples.
Data after balance: 250 samples.
```

Figure34: Compare dataset count before and after balancing.

To avoid code duplication, we created a training pipeline that evaluates balanced and unbalanced Naïve Bayes model classifiers.

```
def train_predict_pipeline(model, X_train, y_train, X_test, y_test, X, y):
    print("            {} Training            ".format(model.__class__.__name__))
    results = {}
    # Training start
    start1 = time()
    # Train the model
    model1 = model.fit(X_train, y_train)
    # Training end
    end1 = time()
    # Store the time
    results['training_time'] = end1 - start1
    # Prediction start
    start1 = time()
    # Predict
    predictions_test = model.predict(X_test)
    predictions_train = model.predict(X_train)
    # Prediction end
    end1 = time()
    # Store the time
    results['prediction_time'] = end1 - start1
    # Overall accuracy
    results['model_accuracy'] = model1.score(X_train, y_train)
    # Cross validation score
    cross_validation_scores = cross_val_score(model,X, y, cv=10)
    results['model_cross_validation'] = np.mean(cross_validation_scores)
    # Accuracy scores - for plotting
    results['accuracy_train'] = accuracy_score(y_train, predictions_train)
    results['accuracy_test'] = accuracy_score(y_test, predictions_test)
    # F-scores
    results['fbeta_train'] = fbeta_score(y_train, predictions_train, beta=0.5)
    results['fbeta_test'] = fbeta_score(y_test, predictions_test, beta=0.5)
    # Print the report
    print('            Accuracy Report            ')
    print('Model Accuracy: %.2f' % results['model_accuracy'])
    print('10-Fold Cross Validation: %.2f' % results['model_cross_validation'])
    print('Training Score: %.2f' % results['fbeta_train'])
    print('Testing Score: %.2f' % results['fbeta_test'])
    print('            Confusion Matrix            ')
    print(confusion_matrix(y_test, predictions_test))
    print(classification_report(y_test, predictions_test))
    display = ConfusionMatrixDisplay.from_estimator(model1, X_test, y_test, display_labels=['negative', 'posi
    display.ax_.set_title('Confusion Matrix ')
    plt.show()
    # Return the results and the classifier
    return results, model1
```

Figure35: Training pipeline method implementation.

Now, we created the models and use them in our training pipeline

```
Naivebayes_classifier = MultinomialNB()
```

```
results_raw = {}

for classifier in [ Naivebayes_classifier]:
    classifier_name = classifier.__class__.__name__
    results_raw[classifier_name] = {}
    results_raw[classifier_name], classifier = train_predict_pipeline(
classifier, X_train_unbalanced, y_train_unbalanced, X_test_unbalanced, y_test_unbalanced, unbalanced_X, y)
```

Figure36: print pipeline method implementation.

Output for Naïve Bayes

```
             MultinomialNB Training
             Accuracy Report
Model Accuracy: 0.80
10-Fold Cross Validation: 0.74
Training Score: 0.83
Testing Score: 0.74
             Confusion Matrix
[[  0  43]
 [  0 100]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        43
           1       0.70      1.00      0.82       100

    accuracy                           0.70       143
   macro avg       0.35      0.50      0.41       143
weighted avg       0.49      0.70      0.58       143
```

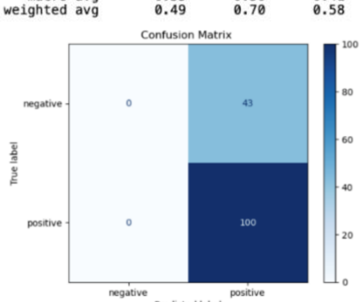


Figure37: Naïve Bayes accuracy report of unbalanced.

```
            MultinomialNB Training
            Accuracy Report
Model Accuracy: 1.00
10-Fold Cross Validation: 0.74
Training Score: 1.00
Testing Score: 0.79
            Confusion Matrix
[[21 22]
 [19 81]]
              precision    recall  f1-score   support

           0       0.53      0.49      0.51        43
           1       0.79      0.81      0.80       100

    accuracy                           0.71       143
   macro avg       0.66      0.65      0.65       143
weighted avg       0.71      0.71      0.71       143
```

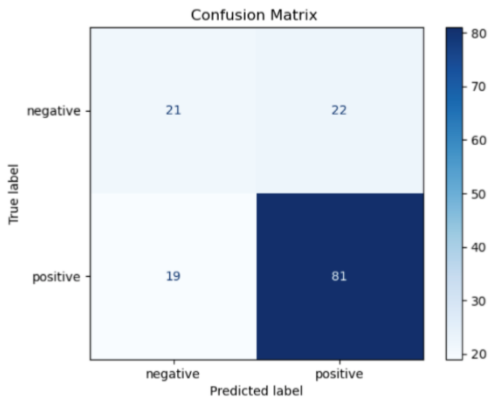


Figure38: Naïve Bayes accuracy report of balanced.

As the figure shows, we plotted the True Positive Rate (TPR) and False Positive Rate (FPR) using the ROC Curve.

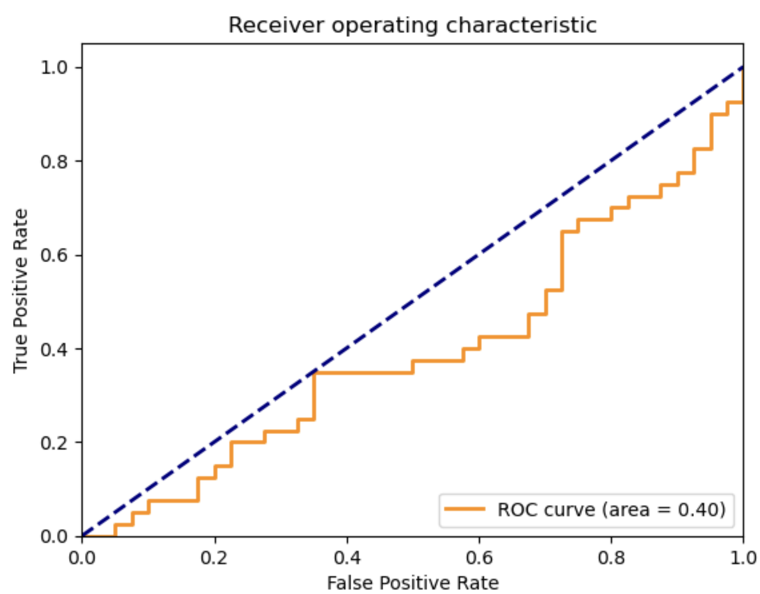


Figure39 : ROC Curve report of unbalanced approach.

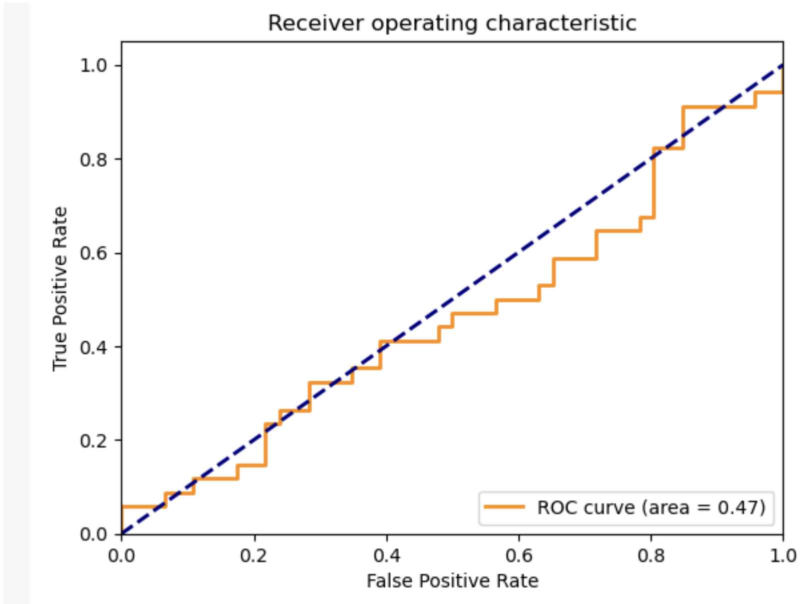


Figure 40: ROC Curve report of balanced approach

.

Questions discussion:

	Unbalanced Dataset	Balanced Dataset
Does the model appear valid and accurate on the test data?	The naive bayes model's accuracy is 0.8, which we consider high, and that would result in it being valid and accurate enough.	We have balanced the data in order to improve the erroneous values and raise accuracy. The accuracy of this model has increased to 1. by balancing the data.
Does the model output/behavior make sense to the domain experts?	For both approaches, the outcome for balanced and unbalanced data is (0.4-0.47). It is seen as a sign that a model's output/behavior makes sense by domain experts.	
Do the parameter values make sense in the context of the domain?	The parameters of both strategies make sense in the context of the domain. Since we thoroughly cleansed all the parameters before classification and endured them inside the scope.	
Is the model sufficiently accurate to meet the goals?	We may say that our models are precise enough to achieve our goals. Since we want to understand how people feel about Costa in order to put our information to use in a practical way and, if possible, for economic purposes. Also, since the model's purpose is not to accurately predict crucial data, any minor deviation or decrease in accuracy is entirely acceptable. Further, our models achieved good accuracy outcomes.	
Are more data or inputs needed?	We constantly require more data to make up for the data that is lost throughout the balancing process in order to improve the quality and accuracy of the data. Considering that we're taking the least-frequented class's number. The remaining members of the other class will be discarded, which will result in a significant loss of data.	

3.3.2 model 2 (logistic regression)

We chose to use the Logistic Regression technique as it is widely used for classification tasks, and our dataset consists of categorical data. Our objective is to build a binary classification model with two classes: Positive and Negative. To achieve this, we removed the Neutral class from the dataset, as shown in Figure41.

```
]: #dropp the neutral and iirelevant classes
data=data[data['sentiment'] != 'natural']
data=data[data['sentiment'] != 'irrelevant']
```

Figure 41: Dropped the neutral and irrelevant classes

```
In [195]: #change values to numeric
data['sentiment'] = data['sentiment'].map({'positive': 0, 'negative':1})
data=data.drop(columns=['Tweets','CleanedTweets','remove_Emojis','remove_nonEnglish','remove_unrelated_chars','remove_stopwords'
data.head(10)
```

Also as shown in figure 42, we changed the values of the sentiment column to binary positive class to 0 and the negative class to 1

Figure 42: change values to numeric values 0,1

```
data.head(10)
```

	cleanTweets	polarity	subjectivity	sentiment	neg	neu	pos	compound
0	We launch kindness campai n John Ma ee try st...	0.400000	0.900000	0.0	0.000	0.690	0.310	0.7430
1	use link treat us so Use Costa app like at bes...	0.325000	0.566667	0.0	0.000	0.691	0.309	0.8176
6	After release new video costa coffee employin c...	0.136364	0.454545	NaN	0.000	1.000	0.000	0.0000
8	Costa Coffee Drive Thru Mecca	0.000000	0.000000	NaN	0.000	1.000	0.000	0.0000
36	Never use Costa coffee shop always use local r...	0.200000	0.500000	0.0	0.000	0.654	0.346	0.6486
38	[illegible] need say [illegible] coffee Costa West coasters...	0.000000	0.000000	NaN	0.000	1.000	0.000	0.0000
40	COSTA COFFEE	0.000000	0.000000	NaN	0.000	1.000	0.000	0.0000
41	A Costa coffee shop shuttin town woolworths sh...	-0.400000	0.700000	0.0	0.071	0.827	0.102	0.2023
42	customer costa [illegible] coffee know road customer...	-0.400000	0.850000	1.0	0.174	0.725	0.101	-0.3612
43	If people want coffee pub Nescafe instant noth...	0.000000	0.666667	0.0	0.000	0.874	0.126	0.0772

Figure 43: sample data output after modifications

As shown in the following Figure 44, we used the isna() method to remove any NaN (Not a Number) cells, if present, from the dataset. Next, we identified the column that contains the tweets for which we want to predict the sentiment, as well as the column that contains the target sentiment class label. To convert the textual data into a meaningful numerical representation, we used the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer.

```
In [196]: data=data[~data['sentiment'].isna()]

In [202]: from sklearn.feature_extraction.text import TfidfVectorizer
          tweet = data['cleanTweets']
          target = data['sentiment']
          tf_vec = TfidfVectorizer()
          x = tf_vec.fit_transform(tweet)
          x.shape

Out[202]: (477, 2958)
```

```
In [203]: x_train , x_test , y_train , y_test , = train_test_split(x , target , test_size=0.30 , random_state=0)
          print(x_train.shape , x_test.shape , y_train.shape , y_test.shape)

          (333, 2958) (144, 2958) (333,) (144,)
```

Figure 44: Identified the columns and used IF-IDF vectorizer

The following figure shown the training phase, during that phase we randomly split our data to 70% for training and 30% for testing

```
In [206]: from sklearn.linear_model import LogisticRegression
          def logistic_classifer (x_train , y_train , x_test , y_test , _C=1.0):
              model = LogisticRegression(C=_C , max_iter=3000).fit (x_train , y_train)
              score = model.score (x_test , y_test)
              print('test score with tf-id features', score)
              return model

          model_tfidf = logistic_classifer(x_train , y_train , x_test , y_test)

          test score with tf-id features 0.75
```

Figure 45: training phase

then we build the logistic regression classifier by entering our training and testing data to the model, as a result we got accuracy 75% as shown in figures 46, 47,48

```
]: from sklearn.linear_model import LogisticRegression
   from sklearn.datasets import load_iris
   from sklearn.model_selection import cross_val_score
   from sklearn.metrics import confusion_matrix
   from sklearn.metrics import classification_report
   from sklearn.metrics import f1_score

   classifier_log = LogisticRegression().fit(x_train , y_train)
   print('logistic accuracy: %.2F'%classifier_log.score(x_test,y_test))
   result_log = cross_val_score (classifier_log , x, target , cv=10)
   print('_'*10)
   print('\n10-fold cross-validation:')

   print(result_log)

   print('_'*100)

   print('the average accuracy of the logistic classifier is : %.2d'% np.mean (result_log))
   print('_'*100)
   print('nConfusion matrix of the logistic classifier:')

   predicted_log= classifier_log.predict(x_test)
   print(confusion_matrix(y_test , predicted_log))
   print('_'*100)

   print('\nclassification_report of logistic classifier :')

   print(classification_report(y_test , predicted_log, zero_division=1))

   print('_'*100)

   logistic accuracy: 0.75
   __________
```

Figure 46: build logistic regression classifier

```
In [206]: from sklearn.linear_model import LogisticRegression
          def logistic_classifer (x_train , y_train , x_test , y_test , _C=1.0):
              model = LogisticRegression(C=_C , max_iter=3000).fit (x_train , y_train)
              score = model.score (x_test , y_test)
              print('test score with tf-id features', score)
              return model

          model_tfidf = logistic_classifer(x_train , y_train , x_test , y_test)

          test score with tf-id features 0.75
```

Figure 47: build logistic regression classifier

```
logistic accuracy: 0.75
__________

10-fold cross-validation:
[0.72916667 0.72916667 0.72916667 0.72916667 0.72916667 0.72916667
 0.72916667 0.74468085 0.74468085 0.74468085]
______________________________________________________________________________________
the average accuracy of the logistic classifier is : 00
______________________________________________________________________________________
nConfusion matrix of the logistic classifier:
[[108   0]
 [ 36   0]]
______________________________________________________________________________________

classification_report of logistic classifier :
              precision    recall  f1-score   support

         0.0       0.75      1.00      0.86       108
         1.0       1.00      0.00      0.00        36

    accuracy                           0.75       144
   macro avg       0.88      0.50      0.43       144
weighted avg       0.81      0.75      0.64       144

______________________________________________________________________________________
```

Figure 48: Logistic Regression Classifier Report

Finally, As shown in figure 49 , we plotted the True Positive Rate (TPR) and False Positive Rate (FPR).

```
In [264]: from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
import numpy as np

# y_true are the true binary labels, and y_score are the predicted scores/probabilities
y_true = np.random.randint(2, size=80)
y_score = np.random.rand(20)

# Interpolate y_score to match the length of y_true
y_score_interp = np.interp(np.linspace(0, 1, num=len(y_true)), np.linspace(0, 1, num=len(y_score)), y_score)

# Compute the ROC curve
fpr, tpr, thresholds = roc_curve(y_true, y_score_interp)
roc_auc = auc(fpr, tpr)

# Plot the ROC curve
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

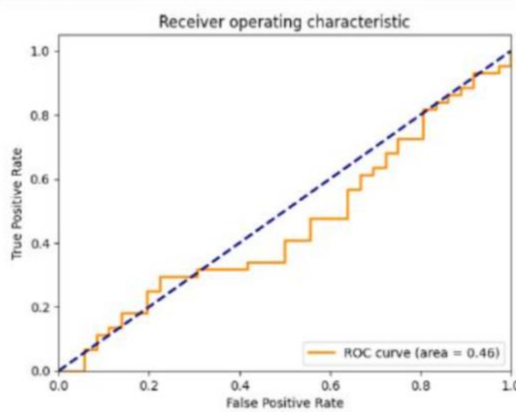


Figure 49: Logistic Regression ROC Curve

Questions discussion:

Does the model appear valid and accurate on the test data?	While applying the model, we have considered that our accuracy is 75% which is a high result. As well we have concluded that the model is valid and accurate.
Does the model output/behavior make sense to the domain experts?	Yes, the model output/behavior makes sense to the domain experts. Because we have a value of the ROC curve that is near (1), the confusion matrix results are good.
Do the parameter values make sense in the context of the domain?	We were mentioned doing sentiment analysis by using the TextBlob tool, which provides a straightforward API and led us to have a clean dataset. So, the answer is yes, a parameter's values do make sense in the domain's context.
Is the model sufficiently accurate to meet the goals?	Yes, according to our classification results, the positive class is higher than the negative class. We got 75% accuracy, which is good enough to meet the goals and reasonably sufficient.
Are more data or inputs needed?	We are looking forward to gaining more data for higher accuracy as well since our a few data.

Phase 4:

Communicate Results

One of the most important abilities for data scientists to have is the ability to clearly convey results to various stakeholders. Because data projects are typically collaborative across functions, the genuine value of a data scientist's work is determined by how well others interpret their insights in order to take additional action.

1. People's Opinions on Costa coffee taste

Our first data visualization aims to test the first hypothesis mentioned in the phase one report "Customers tend to love traditional coffee. ". If the hypothesis was approved, it is going to give Costa an insight into their customers' opinions on whether the customers love the taste of their original traditional coffee in order to create special offers and discounts to attract customers. Conversely, if we could not approve the hypothesis, this may give an indication that customers are not satisfied with the taste of their original coffee.

Findings:

Our hypothesis essentially states that people love the taste of costa original coffee. So, the first step was to visualize people's opinion about the drink's taste as an overview.

```
# importing pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# reading csv file
df = pd.read_csv("CostatwtClean2.csv")

df = df[df['cleanTweets'].str.contains('taste', na=False)]
print(df)
```

```
                                           cleanTweets sentiment    neg  \
12   Costa sort burned caramel coffee ish flavour h...   neutral  0.000
16   I put petrol mistake I otta diesel mp litre I ...  negative  0.185
131  man costa ordered iced espresso immediately ca...  positive  0.123
164  Don Starbucks coffee crap always tastes burnt ...  negative  0.476
180  look photos people coffees costa starbucks wan...  negative  0.393
222  Cos none taste like coffee Nah havin barfed ca...  negative  0.262
232  Lol I I work chief taster Costa round shops te...  positive  0.000
585  Costa Rican coffee experience miss Learn taste...  positive  0.101
683       Thanks Costa Rican coffee Best I ever tasted  positive  0.000
719                    Costa coffee tastes like faeces  positive  0.000
724  I ot kitchen service mornin coffee like busine...  negative  0.205
738  Costa people like weak coffee lar e shot mediu...  positive  0.112
786  Kecewa ah dapat costa coffee ive tasted previo...   neutral  0.000

       neu    pos
12   1.000  0.000
16   0.815  0.000
131  0.672  0.206
164  0.524  0.000
180  0.426  0.180
222  0.738  0.000
232  0.763  0.237
585  0.633  0.266
683  0.413  0.587
```

Figure 50: extracting tweets containing "taste"

Then we printed the following bar chart

```
[52]: positive_df = df[df.sentiment == 'positive']

[53]: negative_df = df[df.sentiment == 'negative']

[57]: charData = {'positive':len(positive_df), 'negative':len(negative_df)}
      label = list(charData.keys())
      value = list(charData.values())

[58]: plt.bar(label,value)

[58]: <BarContainer object of 2 artists>
```

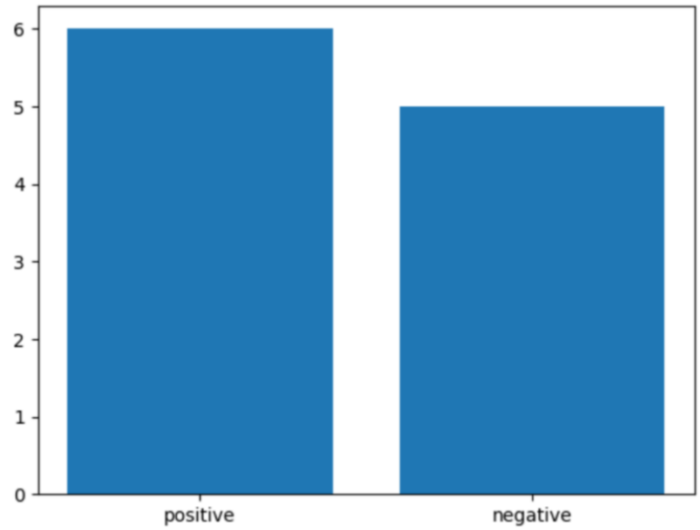


Figure 51: bar chart

We got insight from the first diagram that people seem happy about costa coffee taste. Based on our data analysis, the number of positive tweets about costa coffee taste are higher than the negative. We can say that customers tend to love costa coffee taste which approve to our hypotheses.

2. people opinion on costa coffee environment

In our second data visualization, our goal is to test the second hypothesis that was mentioned during the first phase: ' customers loves The coffee environment .' If the hypothesis is supported, it may suggest that the coffee environment is attractive to customers .

Findings :

First, we have specified the word that might be related to the environment of Starbucks such as " environment". Then we used the contains() method to return all the tweets that contain the related word which were a total of 3 tweets.

```
In [39]: df = pd.read_csv("CostatwtClean2.csv")
         df = df[df['cleanTweets'].str.contains('environment', na=False)]
         print(df)
         print('number of related word :',len(df))

                                        cleanTweets sentiment  neg    neu  \
359  For environmental reasons use k cups Recyclin ...   neutral  0.0  1.000
567  One shot espresso please Workin Costa coffee m...  positive  0.0  0.929
774  Exclusive interview Costa Rican President Rodr...  positive  0.0  0.812

       pos
359  0.000
567  0.071
774  0.188
number of related word : 3
```

Figure 52 :Extracting tweets containing 'enviroment'

Then , we have printed the bar chart based on the sentiment analysis of each tweet "positive" or "negative".

```
In [33]: positive_df = df[df.sentiment == 'positive']
         negative_df = df[df.sentiment == 'negative']
         charData = {'positive':len(positive_df), 'negative':len(negative_df)}
         label = list(charData.keys())
         value = list(charData.values())
         plt.bar(label,value)

Out[33]: <BarContainer object of 2 artists>
```

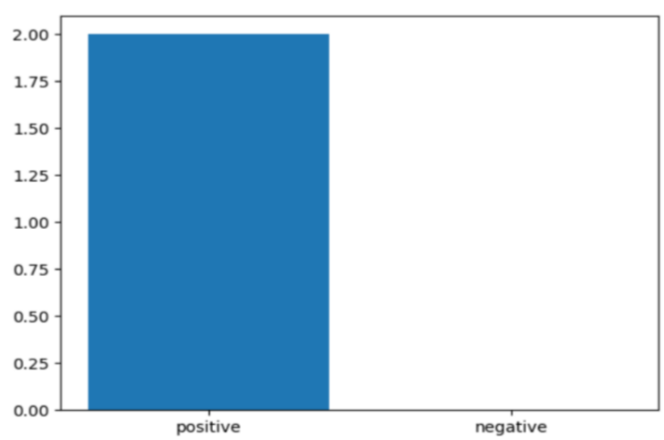


Figure 53 : bar plot chart

3. People’s Opinions on costa prices

According to the phase one report, “Customers believe Costa coffee prices are too expensive.", our third data visualization aims to test that hypothesis. As a result, Costa will have an insight into their customers' opinions on the prices of drinks if the hypothesis is approved, which will allow them to make special offers and discounts to encourage customers to buy. Conversely, if the hypothesis could not be approved, it could indicate that customers are satisfied with the price of the drink.

Findings:

We hypothesize that people think Costa drinks are expensive. First, we visualized people's opinions about the drink's prices.

```
In [132]: # importing pandas as pd
          import pandas as pa
          import matplotlib.pyplot as plt
          import numpy as np
          # reading csv file
          df = pd.read_csv("CostatwtClean2.csv")
          df = df [df ['cleanTweets'].str.contains('price', na=False) ]
          print(df)
```

```
cleanTweets  \
38                                                   Wtf happened costa coffee prices
KD cold brew ridiculous Coffee industry profit mar ins absolute dayli ht robbery
92                                                             Cant wait land birmin
ham buy overpriced coffee airport costa like feel like make happier many ways rn
144
Don Starbucks coffee crap always tastes burnt Don worry I Costa priced shit Brew
323  With profits revenue billion Costa profits billion Why think okay put price coffe
e p When makin billion profits I endin Costa habit today Dirty stinkin rich folks
359                                                                    Jeeso bou ht s
mall black coffee Costa price went Apparently staff havin say rise minimal Really
402                                                                 fxxx milk hot wat
er coffee everyone I spoke piss come Costa ive better deals You cakes priced well
426
I relate overpriced coffee Costa hospital
463          Wow I bi Costa flat white fan increase yet price takin coffee moved luxur
y item ran e alon Heinz ketchup I cut back often I buy invest coffee machine home
472
What rip COSTA put prices AGAIN medium coffee My days definitely
483                                                   The recent surprise price hike m
eans visit Costa Coffee soon every day event rather part routine Bad choice Costa
542
Costa Coffee puttin price coffee p villain ori story
551                                                                              Taru
n Jain replaces Navin Gurnaney Tim Horton India CEO CMP INR DMA w HL TTM price cn
725
parents drink coffee lame ass cappuccinos like Costa overpriced coffee place

    sentiment     neg     neu     pos
38   negative 0.26000 0.62000 0.12000
92   positive 0.00000 0.64100 0.35900
144  negative 0.47600 0.52400 0.00000
323  positive 0.08000 0.52600 0.39300
359   neutral 0.00000 1.00000 0.00000
402  positive 0.12400 0.64500 0.23000
426   neutral 0.00000 1.00000 0.00000
463  positive 0.06500 0.67700 0.25800
472  positive 0.00000 0.78700 0.21300
483  negative 0.15500 0.75200 0.09300
542  negative 0.34000 0.66000 0.00000
551   neutral 0.00000 1.00000 0.00000
725  negative 0.37500 0.47600 0.14900
```

Figure 54 : Extracting tweets containing 'price’.

Then we printed the following bar chart:

```
In [92]: len(df)
         positive_df = df[df.sentiment == 'positive']
         print(len(positive_df))

         5

In [93]: len(df)
         negative_df = df[df.sentiment == 'negative']
         print(len(negative_df))

         5

In [94]: charData = {'positive' :len(positive_df), 'negative': len (negative_df)}
         label = list(charData.keys ())
         value = list(charData.values ())

In [95]: plt.bar (label, value)

Out[95]: <BarContainer object of 2 artists>
```

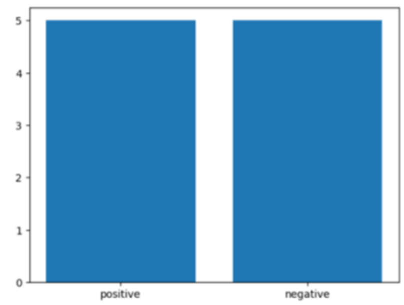


Figure 55: bar plot chart.

Based on the diagram, it appears that people are happy with Costa's prices. Analyzing our data, we found that there were just as many negative tweets about Costa prices as positive tweets.

Our hypotheses are disproved by customers who think Costa prices are good.

Recommendation

The following recommendations are provided to advance the project:

- In order to find new items that people like and to increase the precision of the model by collecting more data.
- It is recommended that we collect data from various sources in order to enhance our analysis.
- New libraries and tools are recommended to improve our analysis.

References:

[1]Costa Coffee, “Our history - Our story | Behind the beans | Costa Coffee,” *Costa.co.uk*, 2022. https://www.costa.co.uk/behind-the-beans/our-story/history .

[2]World Coffee Portal, “Costa Coffee – 3 key Middle East market dynamics,” *World Coffee Portal*, Sep. 18, 2018. https://www.worldcoffeeportal.com/Latest/InsightAnalysis/2018/Costa-Coffee-%E2%80%93-3-key-Middle-East-market-dynamics .

[3] Tyagi, A., 2022. How to Make a Twitter Bot in Python using Tweepy. [online] Auth0.com. Available at: <https://auth0.com/blog/amp/how-to-make-a-twitter-bot-in-python-using- tweepy/>

[4] Pandas.pydata.org. 2022. pandas - Python Data Analysis Library. [online] Available at: <https://pandas.pydata.org/>

[5] Tutorialspoint.com. 2022. Python - Matplotlib. [online] Available at: <https://www.tutorialspoint.com/python_data_science/python_matplotlib.htm>

[6] W3schools.com. 2022. Introduction to NumPy. [online] Available at: <https://www.w3schools.com/python/numpy/numpy_intro.asp>

[7] Nltk.org. 2022. NLTK :: Natural Language Toolkit. [online] Available at: <https://www.nltk.org/>

[8] Barai, M., 2022. Sentiment Analysis with Textblob and Vader in Python. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/10/sentiment- analysis-with-textblob-and-vader/>

[9] *Introduction to Python*. (n.d.).https://www.w3schools.com/python/python_intro.asp#:~:text=Python%20has%20a%20simple%20syntax,prototyping%20can%20be%20very%20quick.

[10]Wikipedia contributors. (2023a). Anaconda (Python distribution). *Wikipedia*. https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)

[11]Wikipedia contributors. (2023b). Python (programming language). *Wikipedia*. https://en.wikipedia.org/wiki/Python_(programming_language)

[12]Wikipedia contributors. (2023c). Project Jupyter. *Wikipedia*. https://en.wikipedia.org/wiki/Project