DATA ANALYTICS AND PREDICTIVE MODELLING

SENTIMENT ANALYSIS PROJECT

MEGHA S RAO

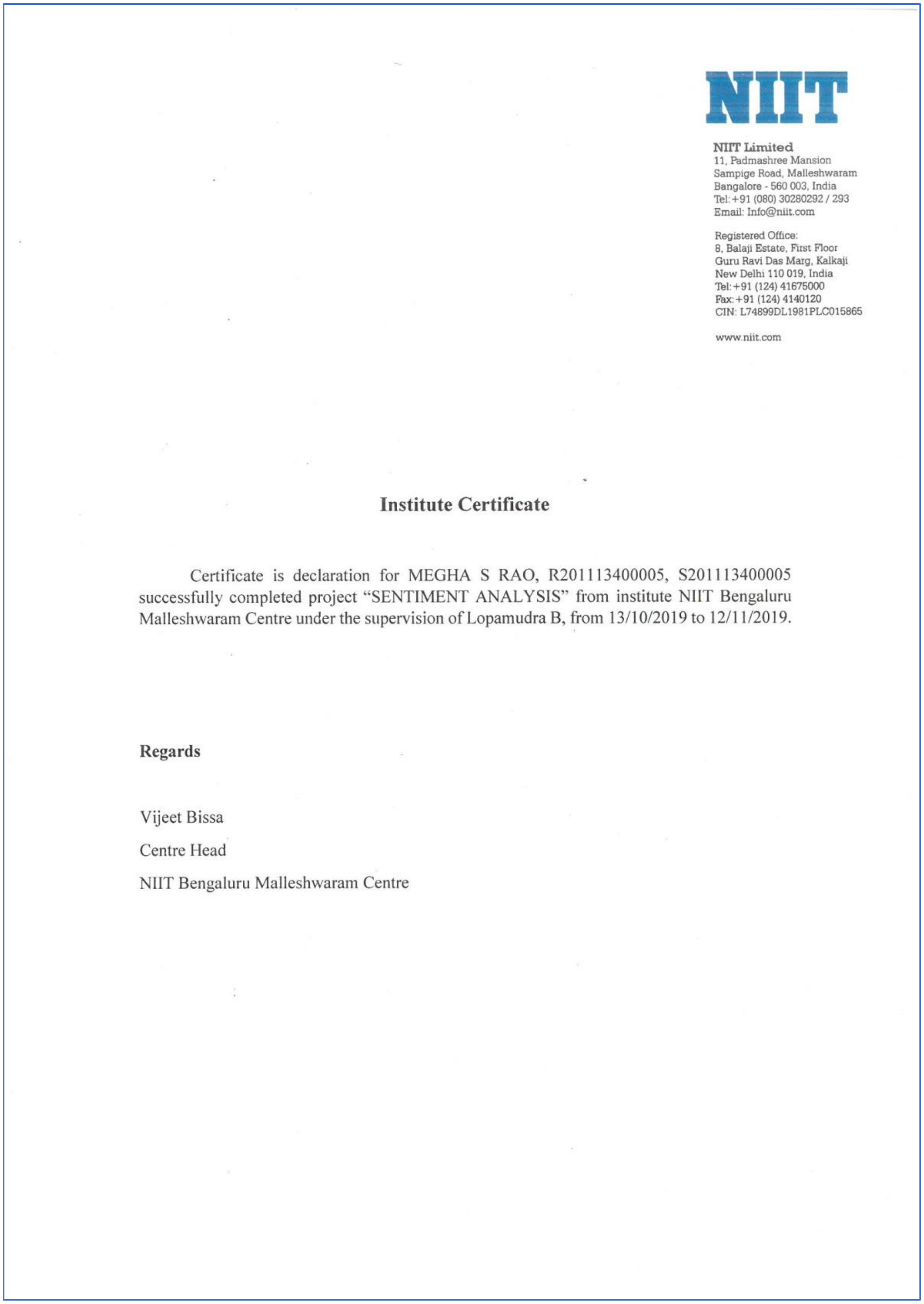
S201113400005

NIIT

BANGALORE

2019

CERTIFICATE



NIIT

NIIT Limited
11, Padmashree Mansion
Sampige Road, Malleshwaram
Bangalore - 560 003, India
Tel:+91 (080) 30280292 / 293
Email: Info@niit.com

Registered Office:
8, Balaji Estate, First Floor
Guru Ravi Das Marg, Kalkaji
New Delhi 110 019, India
Tel:+91 (124) 41675000
Fax:+91 (124) 4140120
CIN: L74899DL1981PLC015865

www.niit.com

Institute Certificate

Certificate is declaration for MEGHA S RAO, R201113400005, S201113400005 successfully completed project "SENTIMENT ANALYSIS" from institute NIIT Bengaluru Malleshwaram Centre under the supervision of Lopamudra B, from 13/10/2019 to 12/11/2019.

Regards

Vijeet Bissa

Centre Head

NIIT Bengaluru Malleshwaram Centre

ACKNOWLEDGEMENT

The satisfactory and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

I would like to proudly thank management of NIIT for providing such a healthy environment for the successful completion of project.

I would like to express my gratitude to Lopamudra Bera, Faculty Advisor for DAPM, NIIT, for her constant support and encouragement.

I would like to thank Vijeeth Bissa, IT Head, NIIT – Malleshwaram, for his constant support and encouragement

I would also like to thank Sudeep Tulpule, Trainer, NIIT - Malleshwaram, for his constant support and encouragement

Finally, I would hereby acknowledge and thank my family who has been a source of inspiration and instrument in the successful completion of the project work.

MEGHA S RAO

TABLE OF CONTENTS

INTRODUCTION

PROJECT BACKGROUND

Social media, as a platform for socializing and communicating, has evolved greatly over the past decade. It now serves as medium for people to express their views, displeasures and appreciation to people and companies about their services and products. Because of this openness and ease to share feedback, companies target social media to understand their customers better. This project will help us understand how consumer sentiment extracted from the tweets through machine learning can be used to generate insights regarding product acceptability and performance in the market.

WHAT IS SENTIMENT ANALYSIS?

Sentiment analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative or neutral.

Sentiment analysis is the process of determining whether a piece of writing is positive, negative or neutral. A sentiment analysis system for text analysis combines natural language processing (NLP) and machine learning techniques to assign weighted sentiment scores to the entities, topics, themes and categories within a sentence or phrase.

WHY DO WE NEED SENTIMENT ANALYSIS?

Sentiment analysis systems allows companies to derive information from unstructured texts by automating business processes, getting actionable insights and saving hours of manual data processing, in other words, by making teams more efficient.

Some of the advantages of sentiment analysis includes the following:

- **Scalability**:

There’s just too much data to process manually like, thousands of tweets, customer support conversations, or customer reviews. Sentiment analysis allows to process data at a faster rate in an efficient and cost-effective way.

- **Real-time analysis**:

We can use sentiment analysis to identify critical information that allows situational awareness during specific scenarios in real-time. Like, is there a PR crisis in social media about to burst? An angry customer that is about to churn? A sentiment analysis system can help you immediately identify these kinds of situations and take action.

- **Consistent criteria**:

Text analysis is a subjective task which is heavily influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data. This helps to reduce errors and improve data consistency.

TWITTER ANALYTICS WITH R

Text analysis in particular has become well established in R. There is a vast collection of dedicated text processing and text analysis packages, from low-level string operations to advanced text modeling techniques such as fitting Latent Dirichlet Allocation models, R provides it all. One of the main advantages of performing text analysis in R is that it is often possible, and relatively easy, to switch between different packages or to combine them. Recent efforts among the R text analysis developers’ community are designed to promote this interoperability to maximize flexibility and choice among users.4 As a result, learning the basics for text analysis in R provides access to a wide range of advanced text analysis features.

Twitter is an online microblogging tool that disseminates more than 400 million messages per day, including vast amounts of information about almost all industries from entertainment to sports, health to business etc. One of the best things about Twitter—indeed, perhaps its greatest appeal—is in its accessibility. It’s easy to use both for sharing information and for collecting it.

PROBLEM DEFINITION

In this project, the aim is to create insightful graphs that indicate consumer sentiment towards e-commerce websites such as Amazon, Flipkart and Myntra. In addition, training will be undertaken for the Naïve Bayes classifiers to classify the tweets according to their overall sentiment and check the accuracy of the results.

PART I : DATA AND PACKAGES

- Extracting around 1,000 tweets about Amazon, Flipkart and Myntra and prepare a data using various cleaning steps.
- The polarity and sentiment score for each tweet is calculated and used for creating individual and comparative plots.
- Various packages and data cleaning methods are used for data preparation.
- Separate data files (csv) are created to analyze positive, negative and neutral sentiments.

PART II : ALGORITHM

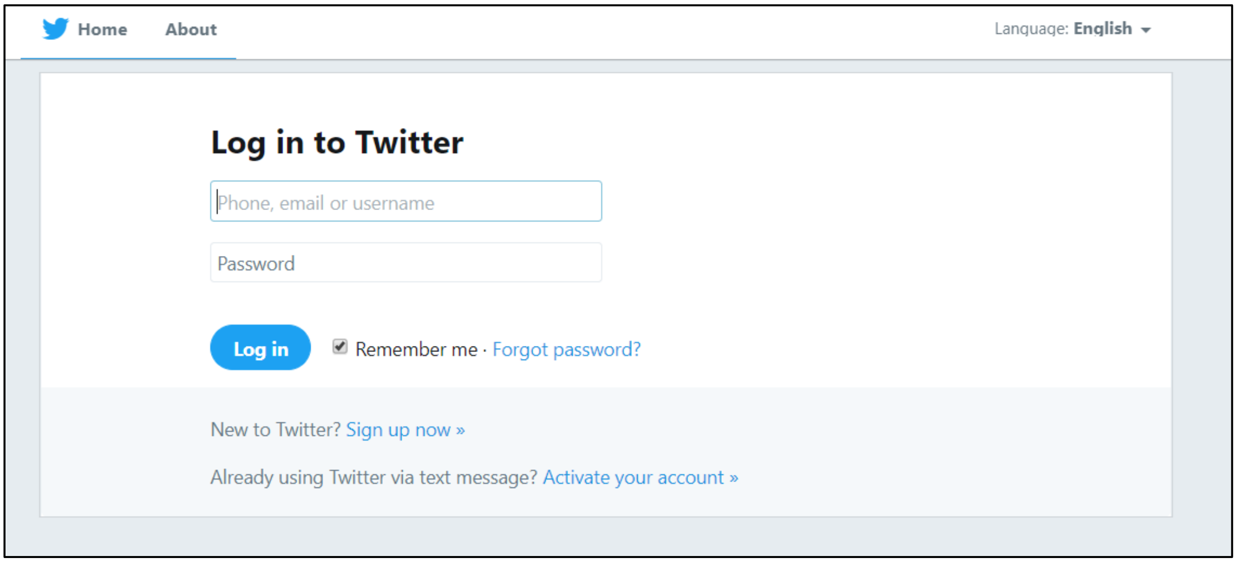
- Naïve Bayes (Supervised Algorithm) is used for the prediction of tweet sentiments as positive, negative and neutral.
- We will split the algorithm as follows

1. Data Collection
2. Exploring and Preparing data
    - Cleaning and Standardizing text data
    - Splitting text documents into words
    - Creating Training and test data sets
    - Visualizing text data – Word Clouds
    - Creating indicator features for frequent words
3. Training a model on the data
4. Evaluating Model Performance
5. Improving Model Performance

IMPLEMENTATION PREREQUISITES

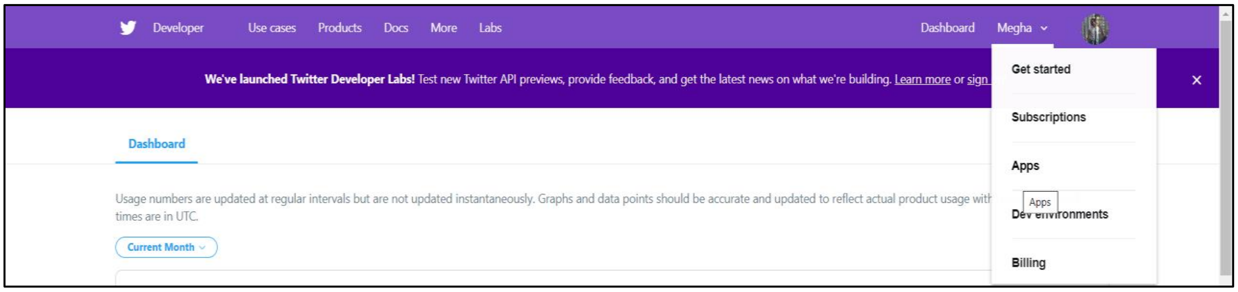
TWITTER APPLICATION

1. In order to create an application, we will need Twitter login ID.
2. Sign in at Twitter Developers



Home
About
Language: English
Log in to Twitter
Phone, email or username
Password
Log in
☑ Remember me · Forgot password?
New to Twitter? Sign up now »
Already using Twitter via text message? Activate your account »

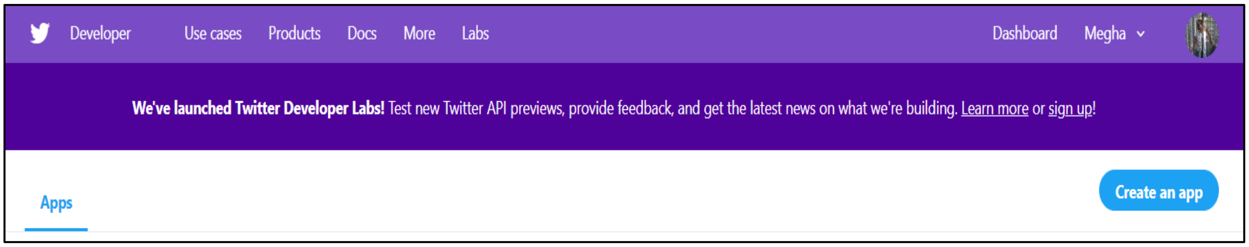
3. Navigate to Apps at the top right corner



We've launched Twitter Developer Labs! Test new Twitter API previews, provide feedback, and get the latest news on what we're building. Learn more or sign

Usage numbers are updated at regular intervals but are not updated instantaneously. Graphs and data points should be accurate and updated to reflect actual product usage wit times are in UTC.

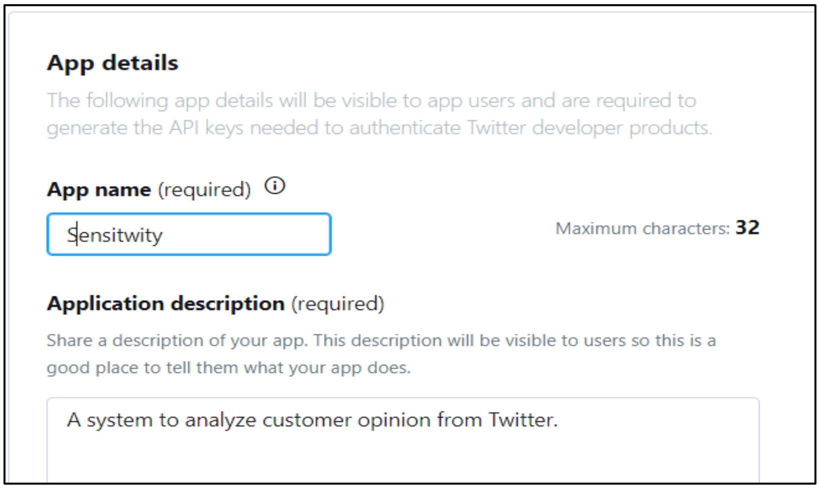
4. Click on Create an App at the right corner



We've launched Twitter Developer Labs! Test new Twitter API previews, provide feedback, and get the latest news on what we're building. Learn more or sign up!

5. Create an Application, enter the necessary details.

- *Fill out the new app form.*
- *Names should be unique, i.e., no one else should have used this name for their Twitter app.*



App details

The following app details will be visible to app users and are required to generate the API keys needed to authenticate Twitter developer products.

App name (required) ⓘ

Sensitwity

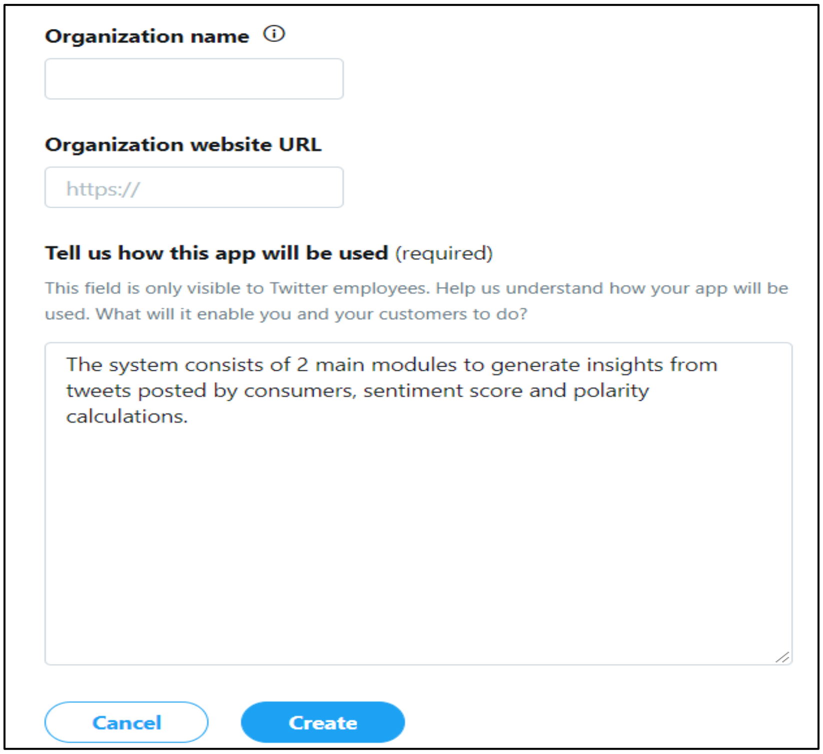
Maximum characters: **32**

Application description (required)

Share a description of your app. This description will be visible to users so this is a good place to tell them what your app does.

A system to analyze customer opinion from Twitter.

6. Give a brief description of the app. You can change this later on if needed.
7. Enter your website or blog address. Callback URL can be left blank.



Organization name ⓘ

Organization website URL

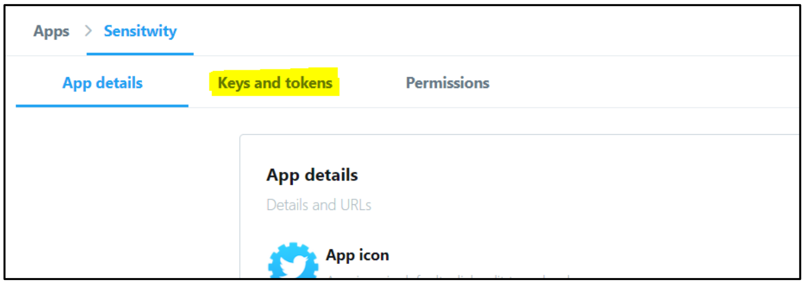
https://

Tell us how this app will be used (required)

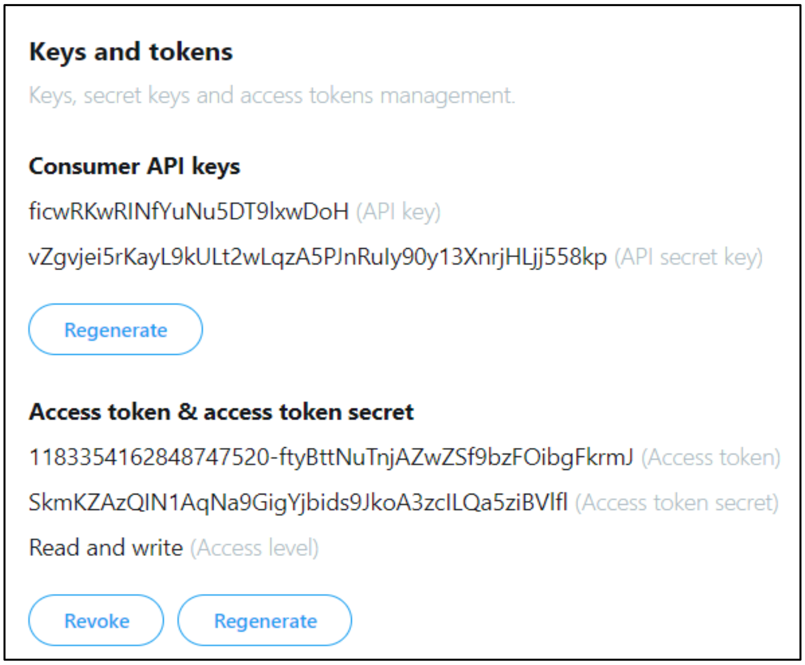
This field is only visible to Twitter employees. Help us understand how your app will be used. What will it enable you and your customers to do?

The system consists of 2 main modules to generate insights from tweets posted by consumers, sentiment score and polarity calculations.

8. Access Tokens by changing the tab ***App details*** to ***Keys and Tokens***



9. Click on **Generate** under Access Tokens to receive the same. This can be Regenerated when needed.



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

ficwRKwRINfYuNu5DT9lxwDoH (API key)

vZgvjei5rKayL9kULt2wLqzA5PJnRuly90y13XnrjHLjj558kp (API secret key)

Access token & access token secret

1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ (Access token)

SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl (Access token secret)

Read and write (Access level)

10. Use the API Key, API Secret, Access Token Key, Access Token Secret to extract data using R Studio.

PACKAGES

1. ***twitteR*** - The twitteR package is intended to provide access to the Twitter API within R, allowing users to grab interesting subsets of Twitter data for their analyses. Provides an interface to the Twitter web API.
2. ***Plyr*** - The plyr package is a set of clean and consistent tools that implement the split-apply-combine pattern in R. This is an extremely common pattern in data analysis: you solve a complex problem by breaking it down into small pieces, doing something to each piece and then combining the results back together again.
3. ***ROAuth*** - The OAuth class is currently implemented as a reference class. An instance of a generator for this class is provided as a convenience to the user as it is configured to handle most standard cases. To access this generator, use the object OAuthFactory.

4. ***Stringr* -** A consistent, simple and easy to use set of wrappers around the fantastic 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another.
5. ***ggplot2* -** A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.
6. ***e1071* -** The package e1071 offers an interface to the
   a. C- and ν-classification
   b. one-class-classification (novelty detection)
   c. ν-regression
   and includes:
   a. linear, polynomial, radial basis function, and sigmoidal kernels
   b. formula interface
   c. k-fold cross validation
7. ***tm*** - This vignette gives a short introduction to text mining in R utilizing the text mining framework provided by the tm package. We present methods for data import, corpus handling, preprocessing, metadata management, and creation of term-document matrices.
8. ***dplyr* -** It provides a flexible grammar of data manipulation. It's the next iteration of plyr, focused on tools for working with data frames (hence the *d* in the name).
9. ***caret* -** The **caret** package (short for Classification And REgression Training) contains functions to streamline the model training process for complex regression and classification problems. The package utilizes a number of R packages but tries not to load them all at package start-up (by removing formal package dependencies, the package startup time can be greatly decreased). The package “suggests” field includes 30 packages. **caret** loads packages as needed and assumes that they are installed.

FILES

.CSV files for positive, negative and neutral words.

SOLUTION STATEMENT

SENTIMENT ANALYSIS

Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

The rise of social media such as blogs and social networks has fueled interest in sentiment analysis. With the proliferation of reviews, ratings, recommendations and other forms of online expression, online opinion has turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities and manage their reputations. As businesses look to automate the process of filtering out the noise, understanding the conversations, identifying the relevant content and actioning it appropriately, many are now looking to the field of sentiment analysis. If web 2.0 was all about democratizing publishing, then the next stage of the web may well be based on democratizing data mining of all the content that is getting published.

Even though short text strings might be a problem, sentiment analysis within microblogging has shown that ***Twitter*** can be seen as a valid online indicator of e-commerce sentiment. Tweets' e-commerce sentiment demonstrates close correspondence to products indicating that the content of Twitter messages plausibly reflects the other (e-commerce websites like Amazon, Flipkart and Myntra).

NAÏVE BAYES CLASSIFICATION

Naive Bayes is a Supervised Machine Learning algorithm based on the Bayes Theorem that is used to solve classification problems by following a probabilistic approach. It is based on the idea that the predictor variables in a Machine Learning model are independent of each other. Meaning that the outcome of a model depends on a set of independent variables that have nothing to do with each other.

Here, we are building a Naïve Bayes model in order to classify future tweets by training the model to analyze the previously extracted data. The algorithm makes a very strong assumption about the data having features independent of each other while in reality, they may be dependent in some way. In other words, it assumes that the presence of one feature in a class is completely unrelated to the presence of all other features. If this assumption of independence holds, Naive Bayes performs extremely well and often better than other models. Naive Bayes can also be used with continuous features but is more suited to categorical variables. If all the input features are categorical, Naive Bayes is recommended. However, in case of numeric features, it makes another strong assumption which is that the numerical variable is normally distributed.

DATA AND CODE

PART I – Sentiment Analysis

1. Extracting and Analyzing Tweets

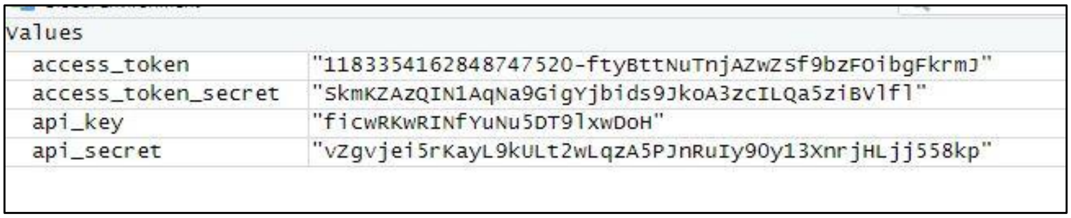
We can extract tweets containing a given # 'hashtag' or @ 'address' words or terms from a user's account or public tweets. Follow the codes below for creating the API keys:

a. Setting the Authorization for Extracting Tweets:
Run the following code in R-Studio to set the authorization for extracting tweets

Code:

```
api_key<-"ficwRKwRINfYuNu5DT9lxwDoH"
api_secret<-"vZgvjei5rKayL9kULt2wLqzA5PJnRuIy90y13XnrjHLjj558kp"
access_token<-"1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ"
access_token_secret<-"SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl"
```

Output:



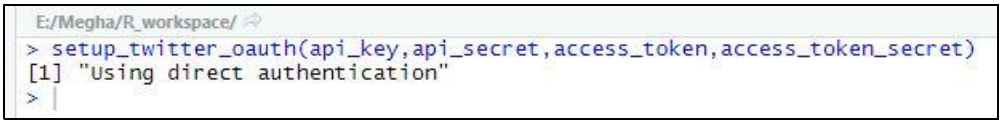
```
Values
  access_token         "1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ"
  access_token_secret  "SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl"
  api_key              "ficwRKwRINfYuNu5DT9lxwDoH"
  api_secret           "vZgvjei5rKayL9kULt2wLqzA5PJnRuIy90y13XnrjHLjj558kp"
```

b. Set up connection between the Twitter app and R:

Code:

```
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
```

Output:



```
E:/Megha/R_workspace/
> setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
[1] "Using direct authentication"
> 
```

2. Required Libraries and installation code

Code:

```
> install.packages("twitteR")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/megha/OneDrive/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
also installing the dependencies 'bit', 'bit64', 'rjson', 'DBI'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/bit_1.1-14.zip'
Content type 'application/zip' length 248571 bytes (242 KB)
```

```
downloaded 242 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/bit64_0.9-7.zip'
Content type 'application/zip' length 551687 bytes (538 KB)
downloaded 538 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/rjson_0.2.20.zip'
Content type 'application/zip' length 578301 bytes (564 KB)
downloaded 564 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/DBI_1.0.0.zip'
Content type 'application/zip' length 888710 bytes (867 KB)
downloaded 867 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/twitteR_1.1.9.zip'
Content type 'application/zip' length 537804 bytes (525 KB)
downloaded 525 KB

package ‘bit’ successfully unpacked and MD5 sums checked
package ‘bit64’ successfully unpacked and MD5 sums checked
package ‘rjson’ successfully unpacked and MD5 sums checked
package ‘DBI’ successfully unpacked and MD5 sums checked
package ‘twitteR’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(twitteR)
Warning message:
package ‘twitteR’ was built under R version 3.6.1

-----------------------------------------------------------------------------------------------
> install.packages("plyr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/plyr_1.8.4.zip'
Content type 'application/zip' length 1303750 bytes (1.2 MB)
downloaded 1.2 MB

package ‘plyr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(plyr)

Attaching package: ‘plyr’
```

```
The following object is masked from ‘package:twitteR’:

    id

Warning message:
package ‘plyr’ was built under R version 3.6.1
-----------------------------------------------------------------------------------------------
> install.packages("ROAuth")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
also installing the dependencies ‘bitops’, ‘RCurl’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/bitops_1.0-6.zip'
Content type 'application/zip' length 38469 bytes (37 KB)
downloaded 37 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/RCurl_1.95-4.12.zip'
Content type 'application/zip' length 2974210 bytes (2.8 MB)
downloaded 2.8 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/ROAuth_0.9.6.zip'
Content type 'application/zip' length 67983 bytes (66 KB)
downloaded 66 KB

package ‘bitops’ successfully unpacked and MD5 sums checked
package ‘RCurl’ successfully unpacked and MD5 sums checked
package ‘ROAuth’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(ROAuth)
Warning message:
package ‘ROAuth’ was built under R version 3.6.1
----------------------------------------------------------------------------------------------

> install.packages("stringr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/stringr_1.4.0.zip'
Content type 'application/zip' length 216967 bytes (211 KB)
downloaded 211 KB

package ‘stringr’ successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(stringr)
Warning message:
package ‘stringr’ was built under R version 3.6.1

-----------------------------------------------------------------------------------------------

> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/ggplot2_3.2.1.zip'
Content type 'application/zip' length 3976565 bytes (3.8 MB)
downloaded 3.8 MB

package ‘ggplot2’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(ggplot2)
Warning message:
package ‘ggplot2’ was built under R version 3.6.1

------------------------------------------------------------------------------------------------

> install.packages("e1071")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/e1071_1.7-2.zip'
Content type 'application/zip' length 1022097 bytes (998 KB)
downloaded 998 KB

package ‘e1071’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(e1071)
Warning message:
package ‘e1071’ was built under R version 3.6.1
```

```
-----------------------------------------------------------------------------------------------

> install.packages("tm")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
also installing the dependencies ‘NLP’, ‘slam’, ‘xml2’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/NLP_0.2-0.zip'
Content type 'application/zip' length 393252 bytes (384 KB)
downloaded 384 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/slam_0.1-46.zip'
Content type 'application/zip' length 211628 bytes (206 KB)
downloaded 206 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/xml2_1.2.2.zip'
Content type 'application/zip' length 3502775 bytes (3.3 MB)
downloaded 3.3 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/tm_0.7-6.zip'
Content type 'application/zip' length 1370304 bytes (1.3 MB)
downloaded 1.3 MB

package ‘NLP’ successfully unpacked and MD5 sums checked
package ‘slam’ successfully unpacked and MD5 sums checked
package ‘xml2’ successfully unpacked and MD5 sums checked
package ‘tm’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(tm)
Loading required package: NLP

Attaching package: ‘NLP’


The following object is masked from ‘package:ggplot2’:

    annotate

Warning message:
package ‘tm’ was built under R version 3.6.1

-----------------------------------------------------------------------------------------------
```

```
> install.packages("dplyr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/dplyr_0.8.3.zip'
Content type 'application/zip' length 3266031 bytes (3.1 MB)
downloaded 3.1 MB

package ‘dplyr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages

> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:plyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following objects are masked from ‘package:twitteR’:

    id, location

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

Warning message:
package ‘dplyr’ was built under R version 3.6.1

----------------------------------------------------------------------------------------------

> install.packages("caret")
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/megha/OneDrive/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
also installing the dependencies ‘SQUAREM’, ‘lava’, ‘prodlim’, ‘iterators’, ‘gower’, ‘ipred’,
‘timeDate’, ‘foreach’, ‘ModelMetrics’, ‘recipes’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/SQUAREM_2017.10-1.zip'
```

```
Content type 'application/zip' length 293374 bytes (286 KB)
downloaded 286 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/lava_1.6.6.zip'
Content type 'application/zip' length 2190774 bytes (2.1 MB)
downloaded 2.1 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/prodlim_2018.04.18.zip'
Content type 'application/zip' length 419855 bytes (410 KB)
downloaded 410 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/iterators_1.0.12.zip'
Content type 'application/zip' length 343802 bytes (335 KB)
downloaded 335 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/gower_0.2.1.zip'
Content type 'application/zip' length 246732 bytes (240 KB)
downloaded 240 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/ipred_0.9-9.zip'
Content type 'application/zip' length 399896 bytes (390 KB)
downloaded 390 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/timeDate_3043.102.zip'
Content type 'application/zip' length 1552467 bytes (1.5 MB)
downloaded 1.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/foreach_1.4.7.zip'
Content type 'application/zip' length 419983 bytes (410 KB)
downloaded 410 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/ModelMetrics_1.2.2.zip'
Content type 'application/zip' length 666096 bytes (650 KB)
downloaded 650 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/recipes_0.1.7.zip'
Content type 'application/zip' length 1564651 bytes (1.5 MB)
downloaded 1.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/caret_6.0-84.zip'
Content type 'application/zip' length 6237461 bytes (5.9 MB)
downloaded 5.9 MB

package ‘SQUAREM’ successfully unpacked and MD5 sums checked
package ‘lava’ successfully unpacked and MD5 sums checked
package ‘prodlim’ successfully unpacked and MD5 sums checked
package ‘iterators’ successfully unpacked and MD5 sums checked
package ‘gower’ successfully unpacked and MD5 sums checked
package ‘ipred’ successfully unpacked and MD5 sums checked
package ‘timeDate’ successfully unpacked and MD5 sums checked
package ‘foreach’ successfully unpacked and MD5 sums checked
package ‘ModelMetrics’ successfully unpacked and MD5 sums checked
package ‘recipes’ successfully unpacked and MD5 sums checked
package ‘caret’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
         C:\Users\megha\AppData\Local\Temp\RtmpsvTqlc\downloaded_packages


> library(caret)
Loading required package: lattice
Warning message:
package ‘caret’ was built under R version 3.6.1

-----------------------------------------------------------------------------------------------
```

3. Importing files:

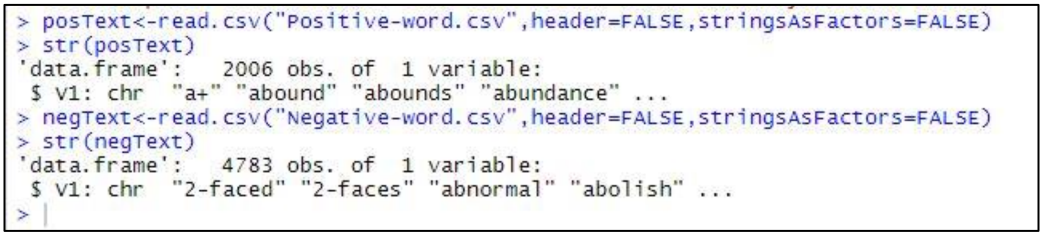
We have to now import files containing the dictionary of positive and negative words.

Code:

```
> posText<-read.csv(filename,header=FALSE,stringsAsFactors=FALSE)
str(posText)

** search for V1 feature, if there just refer this column/feature to the posText object.
> posText<-posText$V1

negText<-read.csv("Negative-word.csv",header=FALSE,stringsAsFactors=FALSE)
str(negText)

## search for V1 feature/column
posText<-posText$V1
negText<-negText$V1


posText<-unlist(lapply(posText, function(x){str_split(x,"\n")}))
negText<-unlist(lapply(negText, function(x){ str_split(x,"\n")}))

** add some more words into posText and negText
pos.word=c(posText,"upgrade")
neg.words=c(negText,"wtf","wait","waiting","epicfail","mechanical")
```

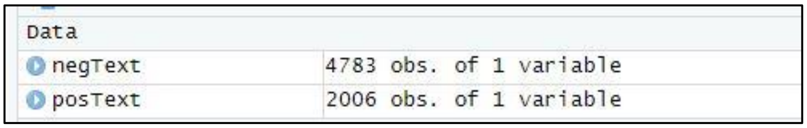
```
posText<-unlist(lapply(posText, function(x){str_split(x,"\n")}))
negText<-unlist(lapply(negText, function(x){ str_split(x,"\n")}))

** add some more words into posText and negText
pos.word=c(posText,"upgrade")
neg.words=c(negText,"wtf","wait","waiting","epicfail","mechanical”
```

Output 1:



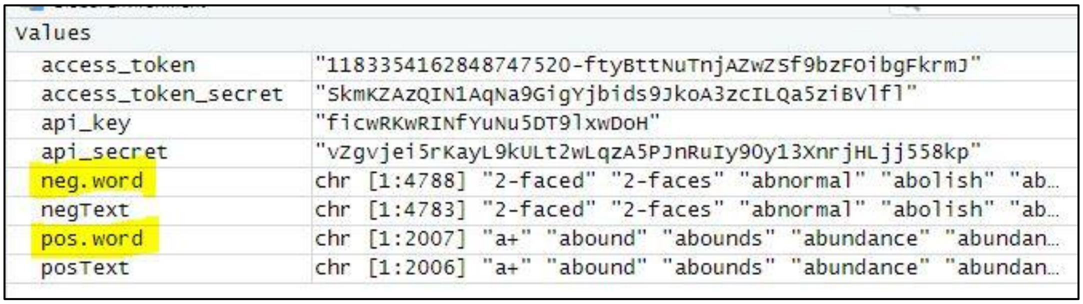
```
> posText<-read.csv("Positive-word.csv",header=FALSE,stringsAsFactors=FALSE)
> str(posText)
'data.frame':	2006 obs. of  1 variable:
 $ V1: chr  "a+" "abound" "abounds" "abundance" ...
> negText<-read.csv("Negative-word.csv",header=FALSE,stringsAsFactors=FALSE)
> str(negText)
'data.frame':	4783 obs. of  1 variable:
 $ V1: chr  "2-faced" "2-faces" "abnormal" "abolish" ...
> 
```

Output 2:



Data	
negText	4783 obs. of 1 variable
posText	2006 obs. of 1 variable

Output 3:



Values	
access_token	"1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ"
access_token_secret	"SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl"
api_key	"ficwRKwRINfYuNu5DT9lxwDoH"
api_secret	"vZgvjei5rKayL9kULt2wLqzA5PJnRuIy9Oy13XnrjHLjj558kp"
neg.word	chr [1:4788] "2-faced" "2-faces" "abnormal" "abolish" "ab…
negText	chr [1:4783] "2-faced" "2-faces" "abnormal" "abolish" "ab…
pos.word	chr [1:2007] "a+" "abound" "abounds" "abundance" "abundan…
posText	chr [1:2006] "a+" "abound" "abounds" "abundance" "abundan…

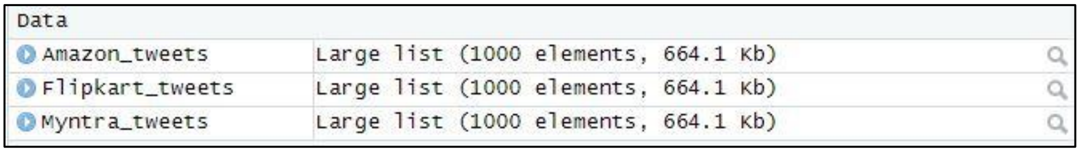
4. Extracting Tweets with hashtags:

To demonstrate sentiment analysis, we analyzed tweets relating to Amazon, Flipkart and Myntra.

Code:

```
Amazon_tweets=searchTwitter('@Amazon',n=1000)
Flipkart_tweets=searchTwitter('@Flipkart',n=1000)
Myntra_tweets=searchTwitter('@Myntra',n=1000)
```

Output:



Data	
Amazon_tweets	Large list (1000 elements, 664.1 Kb)
Flipkart_tweets	Large list (1000 elements, 664.1 Kb)
Myntra_tweets	Large list (1000 elements, 664.1 Kb)

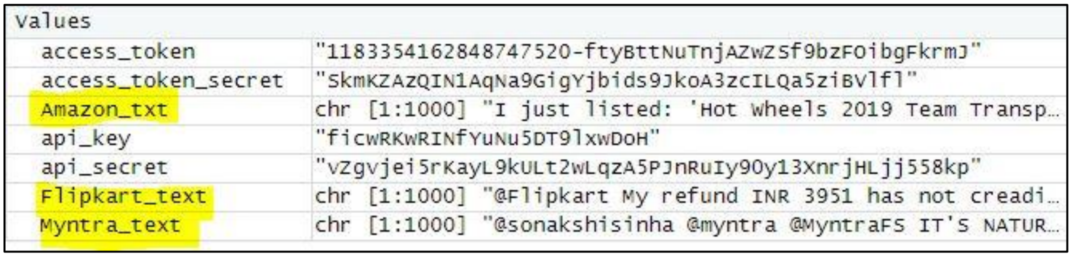
5. Processing Tweets:

a. Convert the tweets into text format:

Code:

```
Amazon_txt=sapply(Amazon_tweets,function(t) t$getText())
Flipkart_text=sapply(Flipkart_tweets,function(t) t$getText())
Myntra_text=sapply(Myntra_tweets,function(t) t$getText())
```

Output:



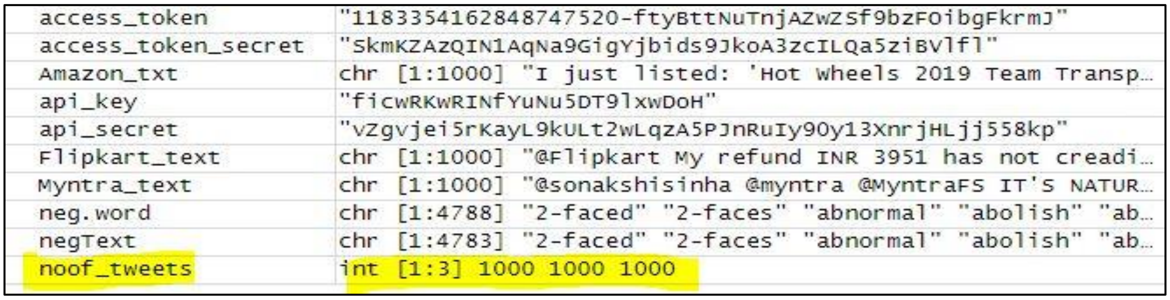
```
Values
  access_token          "1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ"
  access_token_secret   "SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl"
  Amazon_txt            chr [1:1000] "I just listed: 'Hot Wheels 2019 Team Transp…
  api_key               "ficwRKwRINfYuNu5DT9lxwDoH"
  api_secret            "vZgvjei5rKayL9kULt2wLqzA5PJnRuIy90y13XnrjHLjj558kp"
  Flipkart_text         chr [1:1000] "@Flipkart My refund INR 3951 has not creadi…
  Myntra_text           chr [1:1000] "@sonakshisinha @myntra @MyntraFS IT'S NATUR…
```

b. Calculate the number of tweets for each e-commerce company:

Code:

```
noof_tweets=c(length(Amazon_txt),length(Flipkart_txt),length(Myntra_txt))
```

Output:



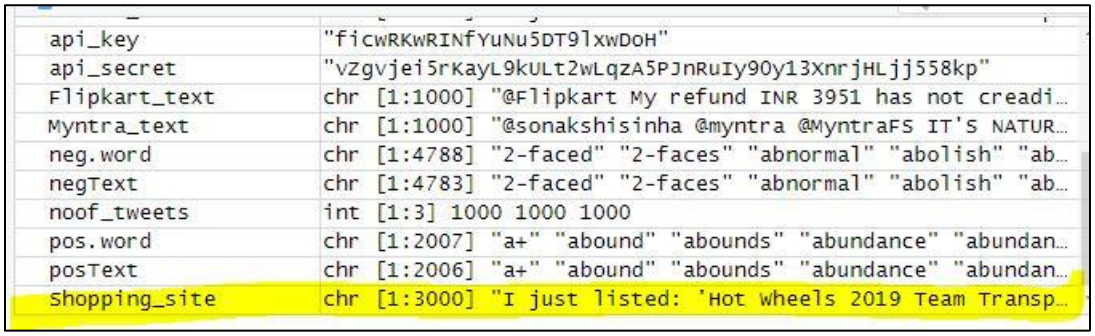
```
  access_token          "1183354162848747520-ftyBttNuTnjAZwZSf9bzFOibgFkrmJ"
  access_token_secret   "SkmKZAzQIN1AqNa9GigYjbids9JkoA3zcILQa5ziBVlfl"
  Amazon_txt            chr [1:1000] "I just listed: 'Hot Wheels 2019 Team Transp…
  api_key               "ficwRKwRINfYuNu5DT9lxwDoH"
  api_secret            "vZgvjei5rKayL9kULt2wLqzA5PJnRuIy90y13XnrjHLjj558kp"
  Flipkart_text         chr [1:1000] "@Flipkart My refund INR 3951 has not creadi…
  Myntra_text           chr [1:1000] "@sonakshisinha @myntra @MyntraFS IT'S NATUR…
  neg.word              chr [1:4788] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  negText               chr [1:4783] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  noof_tweets           int [1:3] 1000 1000 1000
```

c. Combining the text of all these e-commerce companies:

Code:

```
Shopping_site<- c(Amazon_txt,Flipkart_txt,Myntra_txt)
```

Output:



```
  api_key               "ficwRKwRINfYuNu5DT9lxwDoH"
  api_secret            "vZgvjei5rKayL9kULt2wLqzA5PJnRuIy90y13XnrjHLjj558kp"
  Flipkart_text         chr [1:1000] "@Flipkart My refund INR 3951 has not creadi…
  Myntra_text           chr [1:1000] "@sonakshisinha @myntra @MyntraFS IT'S NATUR…
  neg.word              chr [1:4788] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  negText               chr [1:4783] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  noof_tweets           int [1:3] 1000 1000 1000
  pos.word              chr [1:2007] "a+" "abound" "abounds" "abundance" "abundan…
  posText               chr [1:2006] "a+" "abound" "abounds" "abundance" "abundan…
  Shopping_site         chr [1:3000] "I just listed: 'Hot Wheels 2019 Team Transp…
```

6. Sentiment Analysis application code:

The code below will show how sentiment analysis is written and executed.

Before we proceed with sentiment analysis, a function needs to be defined, which will calculate the sentiment score.

parameters of function:

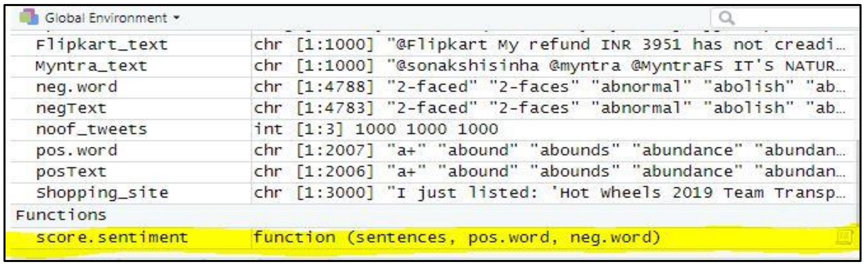
- *sentences* → vector of text to score
- *pos.word* → vector of words of positive sentiment

- *neg.word* → vector of words of negative sentiment
- *sent.score* → is the simple array with sapply()
- *#* → acts as comments which is not processed by R.

Code:

```
score.sentiment= function(sentences,pos.word,neg.word){
  sent.scores= sapply(sentences,function(sentence,pos.word,neg.word){
    #Removing Punctuations
    sentences=gsub("[[:punct:]]","",sentence)
    #Removing Control Characters
    sentences=gsub("[[:cntrl:]]","",sentence)
    #Removing digits
    sentences=gsub("//d+","",sentence)

    #Error handling function when trying to convert lower case
    tryTolower=function(x){
      y=NA
      try_error=tryCatch(tolower(x),error=function(e) e)
      if(!inherits(try_error,"error")){
         y=tolower(x)
        }

        return(y)
      }
    sentence=sapply(sentence,tryTolower)

    #split sentence into words with str_split
    word.list = str_split(sentence, "\\s+")
    words = unlist(word.list)

    #Compare words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.word)
    neg.matches = match(words, neg.word)

    # get the position of the matched term or NA
    # we just want a TRUE/FALSE
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # final score
    score = sum(pos.matches) - sum(neg.matches)
    return(score)
  }, pos.word, neg.word)
```

Output:



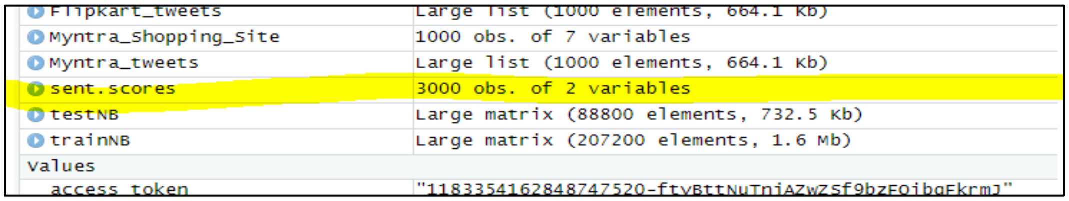
```
  Flipkart_text     chr [1:1000] "@Flipkart My refund INR 3951 has not creadi…
  Myntra_text       chr [1:1000] "@sonakshisinha @myntra @MyntraFS IT'S NATUR…
  neg.word          chr [1:4788] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  negText           chr [1:4783] "2-faced" "2-faces" "abnormal" "abolish" "ab…
  noof_tweets       int [1:3] 1000 1000 1000
  pos.word          chr [1:2007] "a+" "abound" "abounds" "abundance" "abundan…
  posText           chr [1:2006] "a+" "abound" "abounds" "abundance" "abundan…
  Shopping_site     chr [1:3000] "I just listed: 'Hot Wheels 2019 Team Transp…
Functions
  score.sentiment   function (sentences, pos.word, neg.word)
```

7. Start processing the tweets to calculate the sentiment score.

Code:

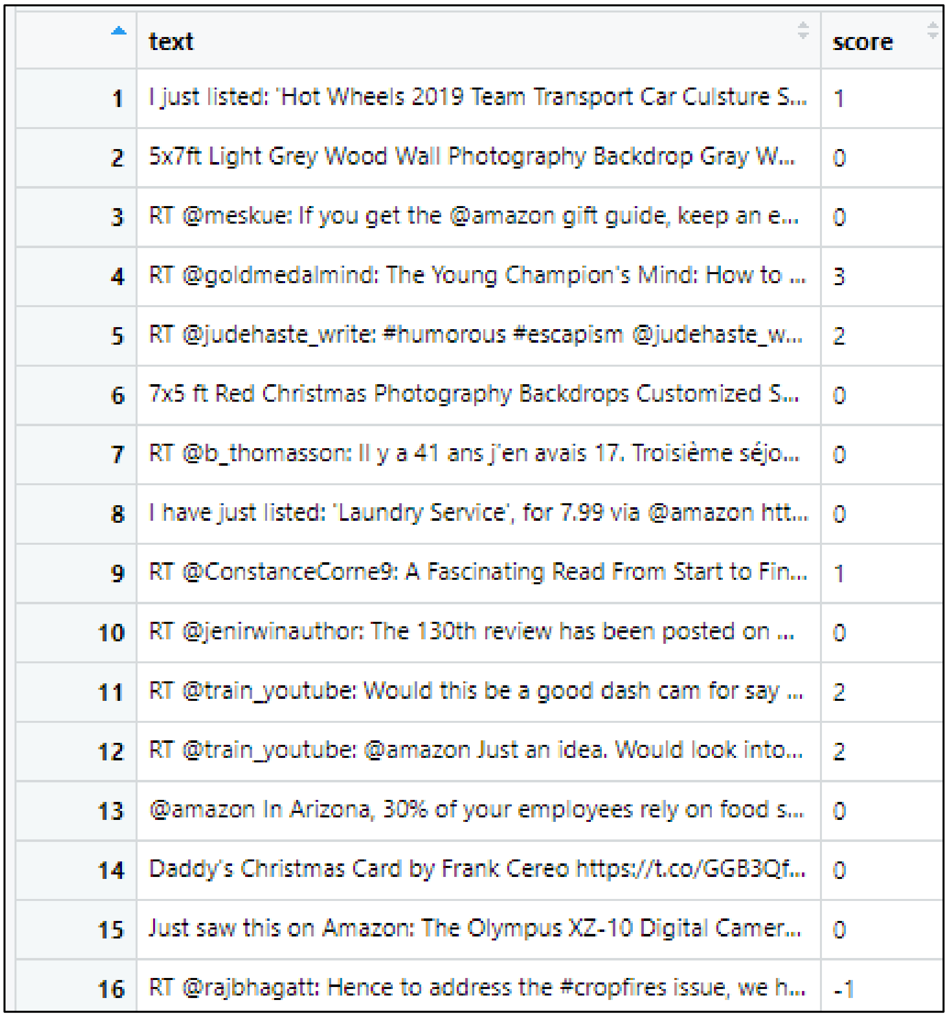
```
sent.scores = score.sentiment(Shopping_Site, pos.word,neg.word)
```

Output 1:



```
Flipkart_tweets          Large list (1000 elements, 664.1 Kb)
Myntra_Shopping_Site     1000 obs. of 7 variables
Myntra_tweets            Large list (1000 elements, 664.1 Kb)
sent.scores              3000 obs. of 2 variables
testNB                   Large matrix (88800 elements, 732.5 Kb)
trainNB                  Large matrix (207200 elements, 1.6 Mb)
Values
  access_token           "1183354162848747520-ftvBttNuTnjAZwZSf9bzFOibqFkrmJ"
```

Output 2:



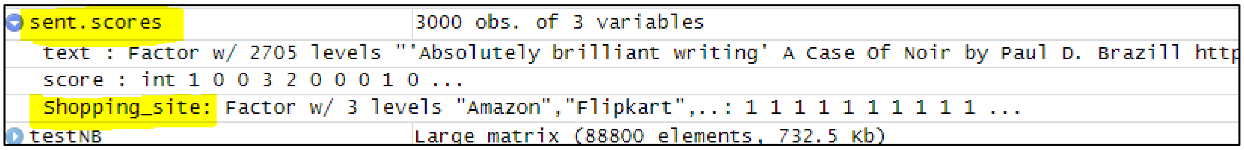
	text	score
1	I just listed: 'Hot Wheels 2019 Team Transport Car Culsture S...	1
2	5x7ft Light Grey Wood Wall Photography Backdrop Gray W...	0
3	RT @meskue: If you get the @amazon gift guide, keep an e...	0
4	RT @goldmedalmind: The Young Champion's Mind: How to ...	3
5	RT @judehaste_write: #humorous #escapism @judehaste_w...	2
6	7x5 ft Red Christmas Photography Backdrops Customized S...	0
7	RT @b_thomasson: Il y a 41 ans j'en avais 17. Troisième séjo...	0
8	I have just listed: 'Laundry Service', for 7.99 via @amazon htt...	0
9	RT @ConstanceCorne9: A Fascinating Read From Start to Fin...	1
10	RT @jenirwinauthor: The 130th review has been posted on ...	0
11	RT @train_youtube: Would this be a good dash cam for say ...	2
12	RT @train_youtube: @amazon Just an idea. Would look into...	2
13	@amazon In Arizona, 30% of your employees rely on food s...	0
14	Daddy's Christmas Card by Frank Cereo https://t.co/GGB3Qf...	0
15	Just saw this on Amazon: The Olympus XZ-10 Digital Camer...	0
16	RT @rajbhagatt: Hence to address the #cropfires issue, we h...	-1

a. Step 1 - Create a variable in the data frame.

Code:

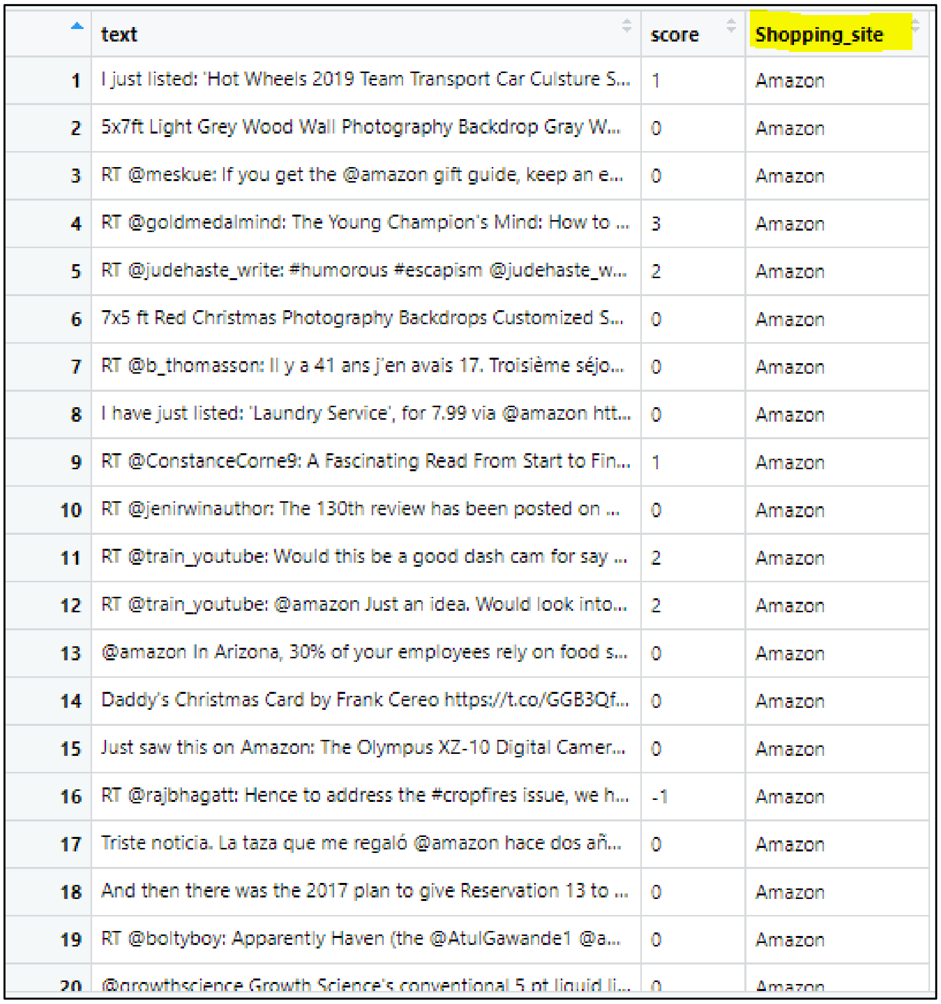
```
sent.scores$Shopping_Site = factor(rep(c("Amazon", "Flipkart","Myntra"), noof_tweets))
```

Output 1:



```
sent.scores                      3000 obs. of 3 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A Case Of Noir by Paul D. Brazill http
  score : int 1 0 0 3 2 0 0 0 1 0 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 1 1 1 1 1 1 1 1 1 1 ...
testNB                           Large matrix (88800 elements, 732.5 Kb)
```

Output 2:



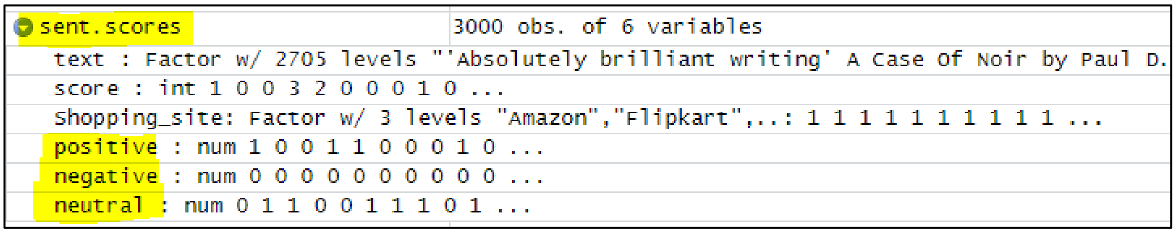
	text	score	Shopping_site
1	I just listed: 'Hot Wheels 2019 Team Transport Car Culsture S...	1	Amazon
2	5x7ft Light Grey Wood Wall Photography Backdrop Gray W...	0	Amazon
3	RT @meskue: If you get the @amazon gift guide, keep an e...	0	Amazon
4	RT @goldmedalmind: The Young Champion's Mind: How to ...	3	Amazon
5	RT @judehaste_write: #humorous #escapism @judehaste_w...	2	Amazon
6	7x5 ft Red Christmas Photography Backdrops Customized S...	0	Amazon
7	RT @b_thomasson: Il y a 41 ans j'en avais 17. Troisième séjo...	0	Amazon
8	I have just listed: 'Laundry Service', for 7.99 via @amazon htt...	0	Amazon
9	RT @ConstanceCorne9: A Fascinating Read From Start to Fin...	1	Amazon
10	RT @jenirwinauthor: The 130th review has been posted on ...	0	Amazon
11	RT @train_youtube: Would this be a good dash cam for say ...	2	Amazon
12	RT @train_youtube: @amazon Just an idea. Would look into...	2	Amazon
13	@amazon In Arizona, 30% of your employees rely on food s...	0	Amazon
14	Daddy's Christmas Card by Frank Cereo https://t.co/GGB3Qf...	0	Amazon
15	Just saw this on Amazon: The Olympus XZ-10 Digital Camer...	0	Amazon
16	RT @rajbhagatt: Hence to address the #cropfires issue, we h...	-1	Amazon
17	Triste noticia. La taza que me regaló @amazon hace dos añ...	0	Amazon
18	And then there was the 2017 plan to give Reservation 13 to ...	0	Amazon
19	RT @boltyboy: Apparently Haven (the @AtulGawande1 @a...	0	Amazon
20	@growthscience Growth Science's conventional 5 pt liquid li...	0	Amazon

b. Step 2 - Calculate positive, negative and neutral sentiments.

Code:

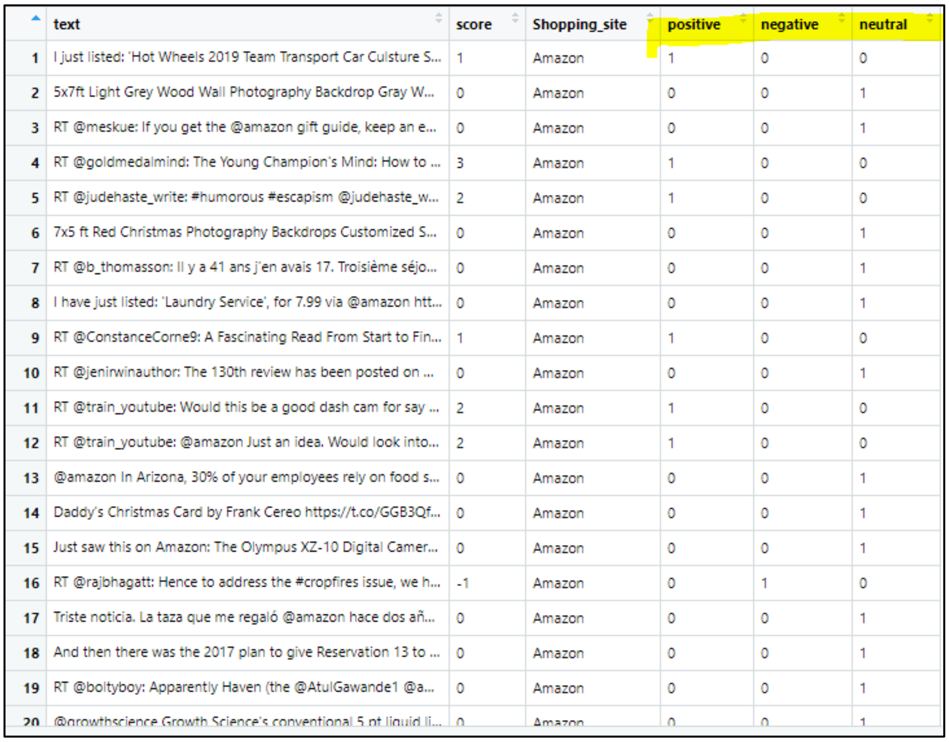
```
sent.scores$positive <- as.numeric(sent.scores$score >0)
sent.scores$negative <- as.numeric(sent.scores$score <0)
sent.scores$neutral <- as.numeric(sent.scores$score==0)
```

Output 1:



```
sent.scores                        3000 obs. of 6 variables
   text : Factor w/ 2705 levels "'Absolutely brilliant writing' A Case Of Noir by Paul D.
   score : int 1 0 0 3 2 0 0 0 1 0 ...
   Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 1 1 1 1 1 1 1 1 1 1 ...
   positive : num 1 0 0 1 1 0 0 0 1 0 ...
   negative : num 0 0 0 0 0 0 0 0 0 0 ...
   neutral : num 0 1 1 0 0 1 1 1 0 1 ...
```

Output 2:



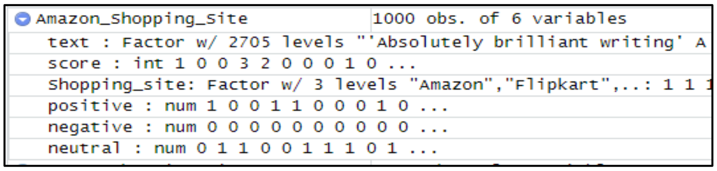
	text	score	Shopping_site	positive	negative	neutral
1	I just listed: 'Hot Wheels 2019 Team Transport Car Culsture S...	1	Amazon	1	0	0
2	5x7ft Light Grey Wood Wall Photography Backdrop Gray W...	0	Amazon	0	0	1
3	RT @meskue: If you get the @amazon gift guide, keep an e...	0	Amazon	0	0	1
4	RT @goldmedalmind: The Young Champion's Mind: How to ...	3	Amazon	1	0	0
5	RT @judehaste_write: #humorous #escapism @judehaste_w...	2	Amazon	1	0	0
6	7x5 ft Red Christmas Photography Backdrops Customized S...	0	Amazon	0	0	1
7	RT @b_thomasson: Il y a 41 ans j'en avais 17. Troisième séjo...	0	Amazon	0	0	1
8	I have just listed: 'Laundry Service', for 7.99 via @amazon htt...	0	Amazon	0	0	1
9	RT @ConstanceCorne9: A Fascinating Read From Start to Fin...	1	Amazon	1	0	0
10	RT @jenirwinauthor: The 130th review has been posted on ...	0	Amazon	0	0	1
11	RT @train_youtube: Would this be a good dash cam for say ...	2	Amazon	1	0	0
12	RT @train_youtube: @amazon Just an idea. Would look into...	2	Amazon	1	0	0
13	@amazon In Arizona, 30% of your employees rely on food s...	0	Amazon	0	0	1
14	Daddy's Christmas Card by Frank Cereo https://t.co/GGB3Qf...	0	Amazon	0	0	1
15	Just saw this on Amazon: The Olympus XZ-10 Digital Camer...	0	Amazon	0	0	1
16	RT @rajbhagatt: Hence to address the #cropfires issue, we h...	-1	Amazon	0	1	0
17	Triste noticia. La taza que me regaló @amazon hace dos añ...	0	Amazon	0	0	1
18	And then there was the 2017 plan to give Reservation 13 to ...	0	Amazon	0	0	1
19	RT @boltyboy: Apparently Haven (the @AtulGawande1 @a...	0	Amazon	0	0	1
20	@growthscience Growth Science's conventional 5 pt liquid li...	0	Amazon	0	0	1

c. Step 3 - Split the data frame into individual datasets for each Shopping Site.

Code:

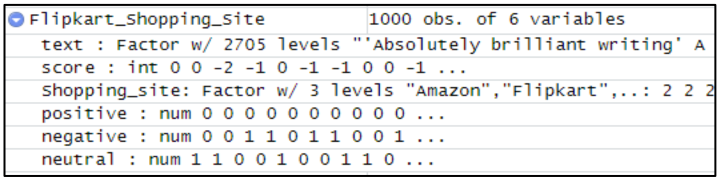
```
Amazon_Shopping_Site <- subset(sent.scores, sent.scores$Shopping_Site=="Amazon")
Flipkart_Shopping_Site <- subset(sent.scores,sent.scores$Shopping_Site=="Flipkart")
Myntra_Shopping_Site <- subset(sent.scores,sent.scores$Shopping_Site=="Myntra")
```

Output 1:



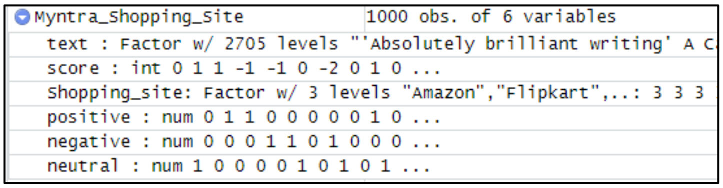
```
Amazon_Shopping_Site              1000 obs. of 6 variables
   text : Factor w/ 2705 levels "'Absolutely brilliant writing' A
   score : int 1 0 0 3 2 0 0 0 1 0 ...
   Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 1 1 1
   positive : num 1 0 0 1 1 0 0 0 1 0 ...
   negative : num 0 0 0 0 0 0 0 0 0 0 ...
   neutral : num 0 1 1 0 0 1 1 1 0 1 ...
```

Output 2:



```
Flipkart_Shopping_Site        1000 obs. of 6 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A
  score : int 0 0 -2 -1 0 -1 -1 0 0 -1 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 2 2 2
  positive : num 0 0 0 0 0 0 0 0 0 0 ...
  negative : num 0 0 1 1 0 1 1 0 0 1 ...
  neutral : num 1 1 0 0 1 0 0 1 1 0 ...
```

Output 3:



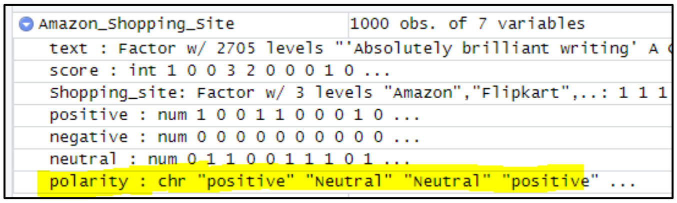
```
Myntra_Shopping_Site          1000 obs. of 6 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A C
  score : int 0 1 1 -1 -1 0 -2 0 1 0 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 3 3 3
  positive : num 0 1 1 0 0 0 0 0 1 0 ...
  negative : num 0 0 0 1 1 0 1 0 0 0 ...
  neutral : num 1 0 0 0 0 1 0 1 0 1 ...
```

d. Step 4 - Create polarity variable for each data frame.

Code:

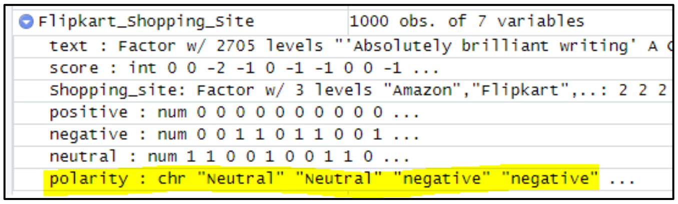
```
Amazon_Shopping_Site$polarity <- ifelse(Amazon_Shopping_Site$score
>0,"positive",ifelse(Amazon_Shopping_Site$score <
0,"negative",ifelse(Amazon_Shopping_Site$score==0,"Neutral",0)))

Flipkart_Shopping_Site$polarity <- ifelse(Flipkart_Shopping_Site$score
>0,"positive",ifelse(Flipkart_Shopping_Site$score <
0,"negative",ifelse(Flipkart_Shopping_Site$score==0,"Neutral",0)))

Myntra_Shopping_Site$polarity <- ifelse(Myntra_Shopping_Site$score
>0,"positive",ifelse(Myntra_Shopping_Site$score <
0,"negative",ifelse(Myntra_Shopping_Site$score==0,"Neutral",0)))
```

Output 1:



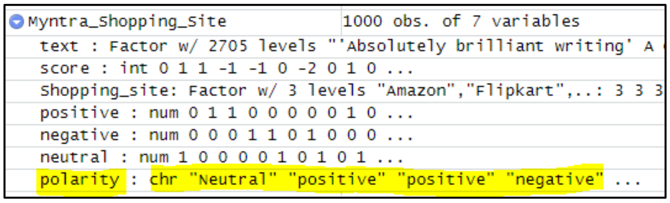
```
Amazon_Shopping_Site          1000 obs. of 7 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A
  score : int 1 0 0 3 2 0 0 0 1 0 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 1 1 1
  positive : num 1 0 0 1 1 0 0 0 1 0 ...
  negative : num 0 0 0 0 0 0 0 0 0 0 ...
  neutral : num 0 1 1 0 0 1 1 1 0 1 ...
  polarity : chr "positive" "Neutral" "Neutral" "positive" ...
```

Output 2:



```
Flipkart_Shopping_Site        1000 obs. of 7 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A
  score : int 0 0 -2 -1 0 -1 -1 0 0 -1 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 2 2 2
  positive : num 0 0 0 0 0 0 0 0 0 0 ...
  negative : num 0 0 1 1 0 1 1 0 0 1 ...
  neutral : num 1 1 0 0 1 0 0 1 1 0 ...
  polarity : chr "Neutral" "Neutral" "negative" "negative" ...
```

Output 3:



```
Myntra_Shopping_Site          1000 obs. of 7 variables
  text : Factor w/ 2705 levels "'Absolutely brilliant writing' A
  score : int 0 1 1 -1 -1 0 -2 0 1 0 ...
  Shopping_site: Factor w/ 3 levels "Amazon","Flipkart",..: 3 3 3
  positive : num 0 1 1 0 0 0 0 0 1 0 ...
  negative : num 0 0 0 1 1 0 1 0 0 0 ...
  neutral : num 1 0 0 0 0 1 0 1 0 1 ...
  polarity : chr "Neutral" "positive" "positive" "negative" ...
```

8. Generating Graphs

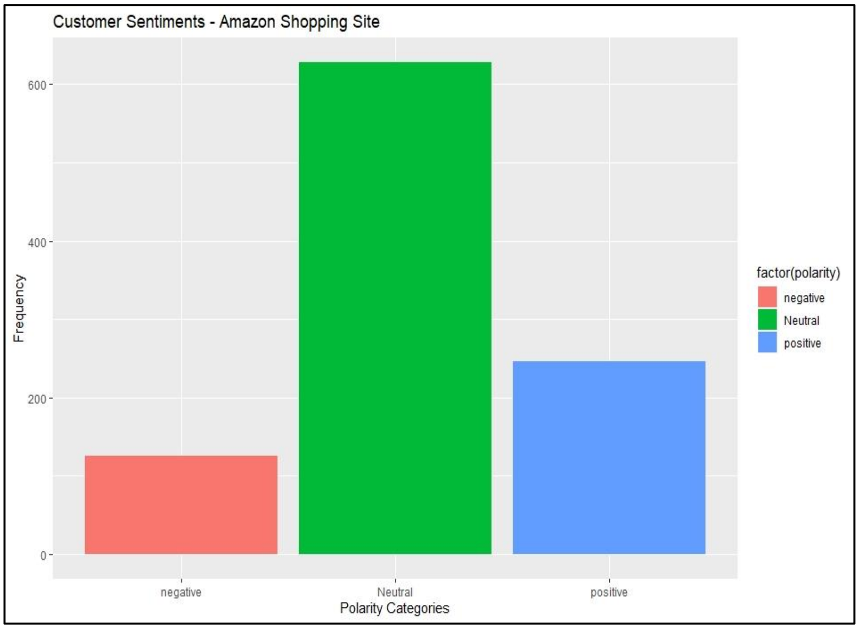
After the execution of the above steps, we will go ahead and create insightful graphs. The steps below outline the process to create graphs.

Plot 1- Polarity Plot – Customer Sentiments (Amazon)

Code:

```
qplot(factor(polarity), data=Amazon_Shopping_Site, geom="bar",
fill=factor(polarity))+xlab("Polarity Categories") +ylab("Frequency") +ggtitle("Customer
Sentiments - Amazon Shopping Site")
```

Output:



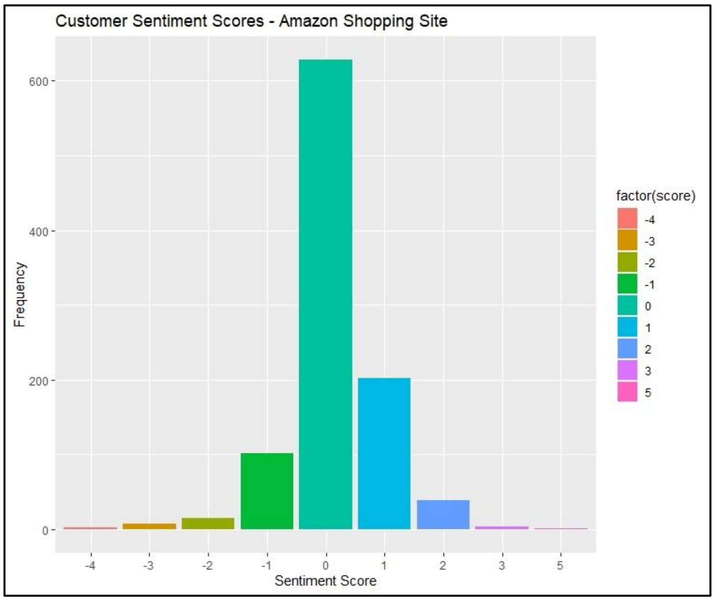
The bar graph above depicts polarity if we closely analyze the graph. It reveals that out of 1,000 Twitter users, 100 users have commented in a negative way while 660 users are neutral. However, 240 users are pretty positive about Amazon.

Plot 2- Customer Sentiment Scores (Amazon Shopping Site)

Code:

```
qplot(factor(score), data=Amazon_Shopping_Site, geom="bar", fill=factor(score))+xlab("Sentiment
Score") + ylab("Frequency") + ggtitle("Customer Sentiment Scores - Amazon Shopping Site")
```

Output:



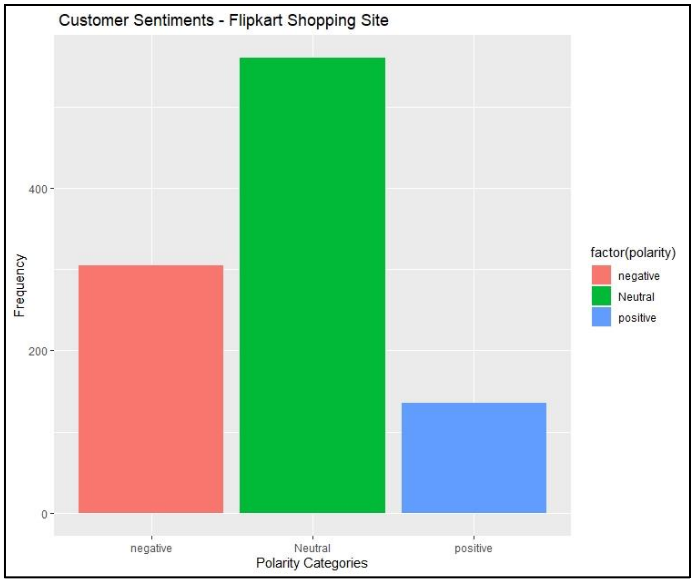
The bar graph above depicts a Twitter user's sentiment score, the negative score denoted by the (-) symbol, indicates the unhappiness of users with Amazon, and the positive score denotes that users are happy with Amazon. Zero represents that Twitter users are neutral.

Plot 3 - Polarity Plot – Customer Sentiments (Flipkart)

Code:

```
qplot(factor(polarity), data=Flipkart_Shopping_Site, geom="bar",
fill=factor(polarity))+xlab("Polarity Categories") +
ylab("Frequency") + ggtitle(" Customer Sentiments - Flipkart
Shopping Site ")
```

Output:



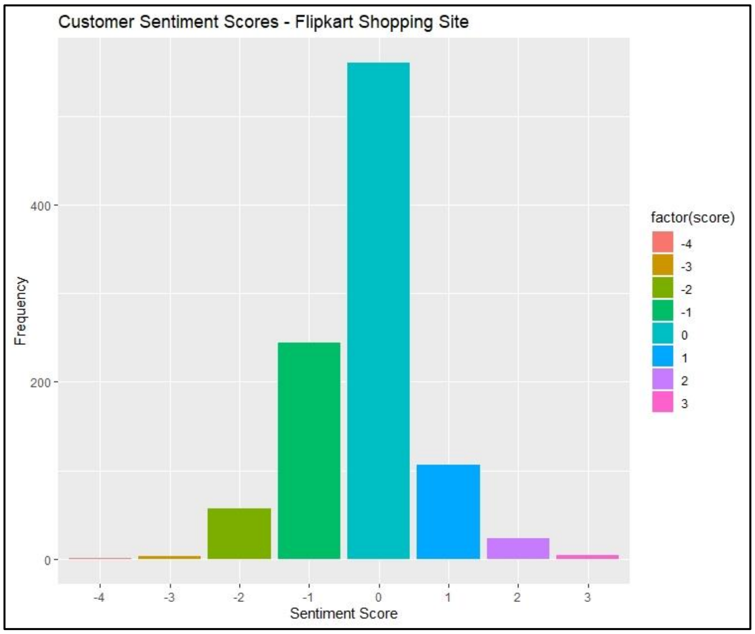
The bar graph above represents polarity. In this case, out of the 1,000 Twitter users, 240 users have commented negatively, 460 users remain neutral, and 300 users are positive about Flipkart.

Plot 4 - Customer Sentiment Scores (Flipkart)

Code:

```
qplot(factor(score), data=Flipkart_Shopping_Site, geom="bar",
fill=factor(score))+xlab("Sentiment Score") + ylab("Frequency")
+ ggtitle("Customer Sentiment Scores - Flipkart Shopping Site")
```

Output:



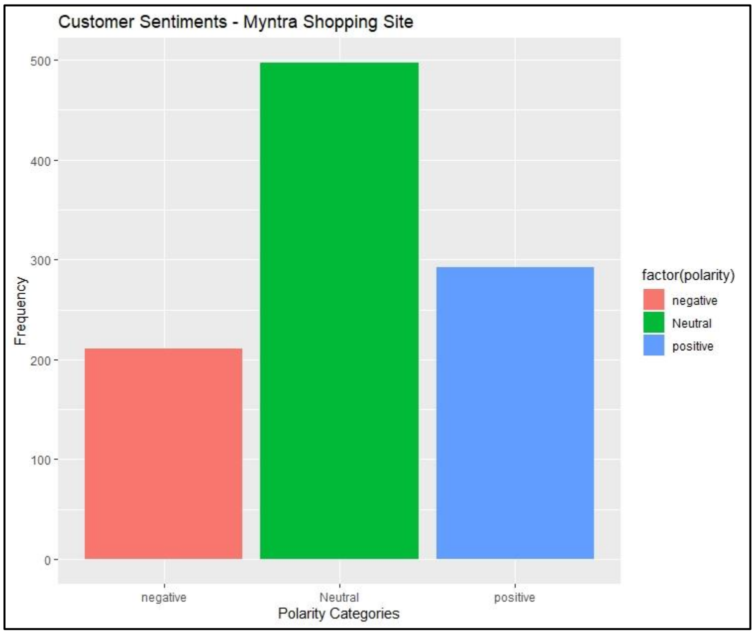
The bar graph above depicts a Twitter user's sentiment score. The negative score, denoted by the (-) symbol, indicates unhappiness with the Flipkart Shopping Site and the positive score denotes that users are quite happy. The zero here represents that users are neutral.

Plot 5 - Polarity Plot – Customer Sentiments (Myntra)

Code:

```
qplot(factor(polarity), data=Myntra_Shopping_Site, geom="bar",
fill=factor(polarity))+xlab("Polarity Categories") + ylab("Frequency") + ggtitle("Customer
Sentiments - Myntra Shopping Site")
```

Output:



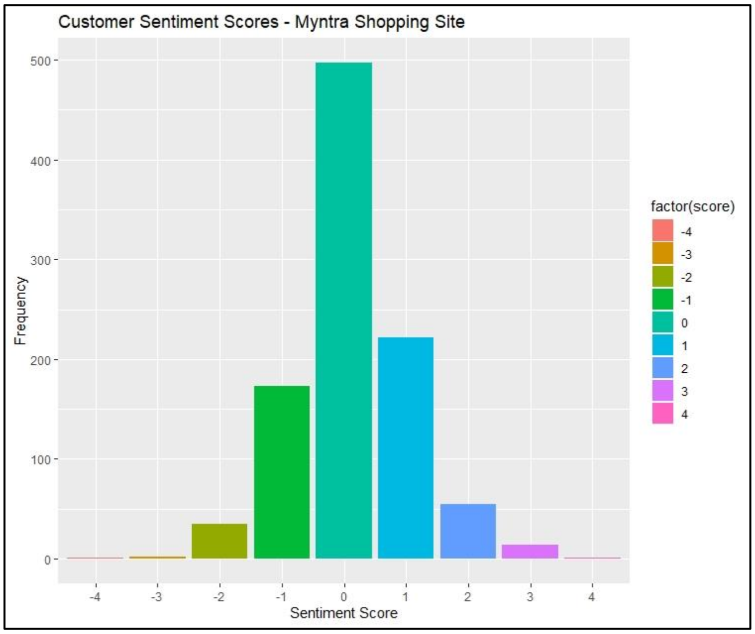
The bar graph above represents polarity. In this case, out of the 1,000 Twitter users, 80 users have commented negatively, 380 users are neutral, and the remaining 540 users remain positive about the e-commerce company.

Plot 6 - Customer Sentiment Scores (Myntra)

Code:

```
qplot(factor(score), data=Myntra_Shopping_Site, geom="bar",
fill=factor(score))+xlab("Sentiment Score") + ylab("Frequency")
+ ggtitle("Customer Sentiment Scores - Myntra Shopping Site ")
```

Output:



The bar graph above depicts the Twitter user's sentiment score. The negative score denoted by the (-) symbol indicates the unhappiness of users with the e-commerce company while the positive score denotes that users are quite happy. Zero represents that users are neutral about their opinion.

9. Summarizing Scores

The code below will help us to summarize the overall positive, negative, and neutral scores. To put it in another way, we will create the total count by adding the positive, negative, and neutral sums. Additionally, we will calculate the positive, negative, and neutral percentages using the code below:

Code:

```
datafrm = ddply(sent.scores, c("Shopping_site"),summarise,pos_count=sum( positive
),neg_count=sum( negative ),neu_count=sum(neutral))

####Total Count
datafrm$total_count = datafrm$pos_count +datafrm$neg_count + datafrm$neu_count

####Percentage
datafrm$pos_percent_score = round( 100*datafrm$pos_count/datafrm$total_count )

datafrm$neg_percent_score = round( 100*datafrm$neg_count/datafrm$total_count )

datafrm$neu_percent_score = round( 100*datafrm$neu_count/datafrm$total_count )
```

10. Comparison Charts

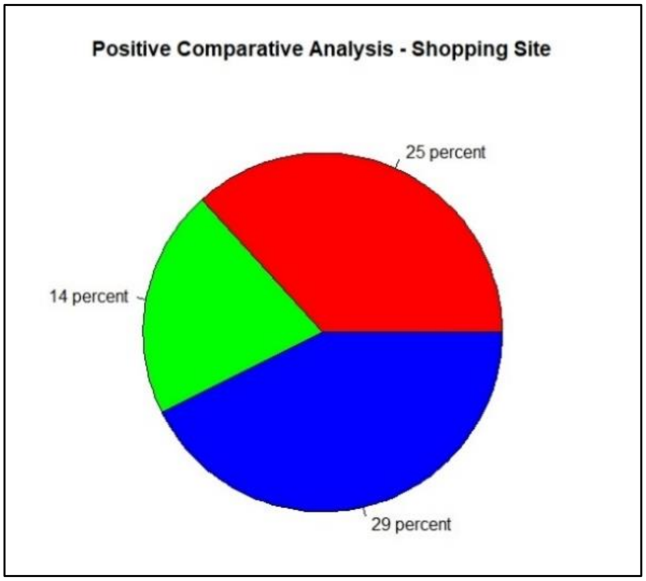
Comparison 1 - Positive Comparative Analysis

Here is the code to create a positive comparison pie chart for these three ecommerce companies:

Code:

```
> attach(datafrm)
> pie.chart.lbls <-paste(datafrm$Shopping_Site,datafrm$pos_percent_score)
> pie.chart.lbls <- paste(pie.chart.lbls,"percent",sep="")
> pie(pos_percent_score, labels = pie.chart.lbls, col = rainbow(length(pie.chart.lbls)),main =
"Positive Comparative Analysis - Shopping Site")
```

The pie chart below represents the positive percentage score of these companies:

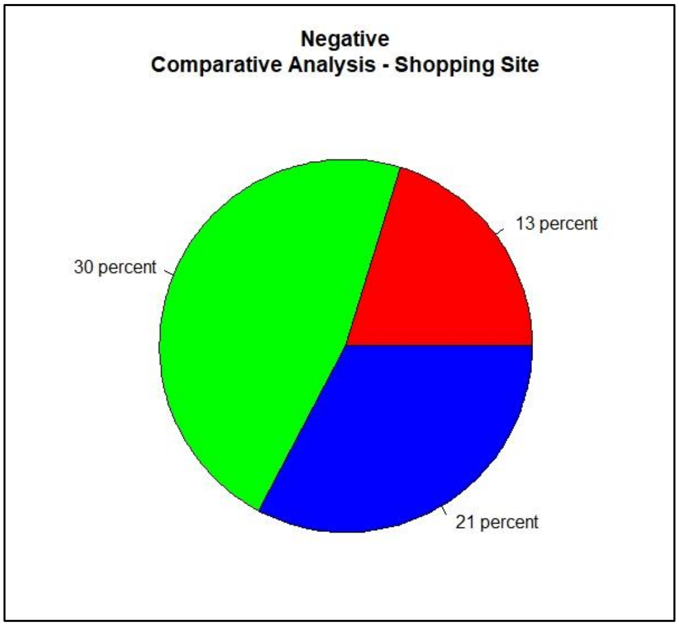


Comparison 2 - Negative Comparative Analysis

Here is the code to create a negative comparison pie chart for these three ecommerce companies:

```
pie.chart.lbls <-paste(datafrm$Shopping_Site,datafrm$neg_percent_score)
pie.chart.lbls <- paste(pie.chart.lbls,"percent",sep=" ")
pie(neg_percent_score, labels = pie.chart.lbls, col = rainbow(length(pie.chart.lbls)),
main = "Negative Comparative Analysis - Shopping Site")
```

The pie chart below represents the negative percentage score of these three companies:

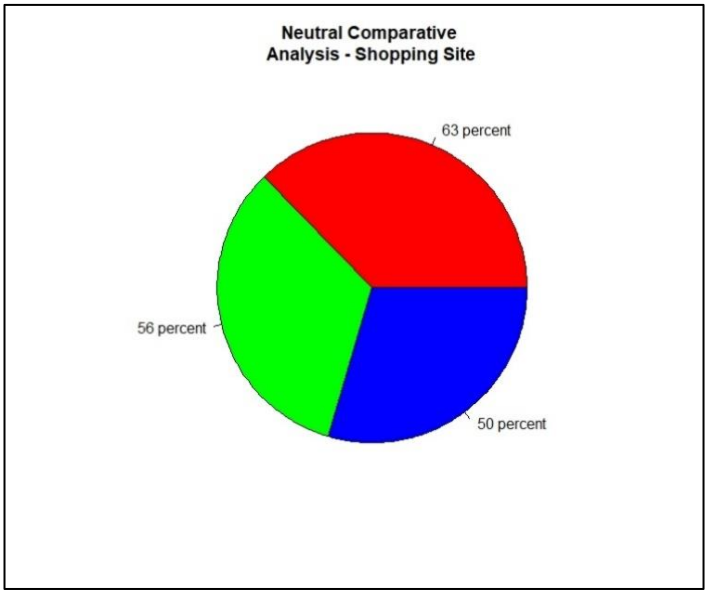


Comparison 3 - Neutral Comparative Analysis

Here is the code to create a neutral comparison pie chart:

```
pie.chart.lbls <-paste(datafrm$Shopping_Site,datafrm$neu_percent_score)
pie.chart.lbls <- paste(pie.chart.lbls,"percent",sep="")
pie(neu_percent_score, labels = pie.chart.lbls, col = rainbow(length(pie.chart.lbls)),
main = "Neutral Comparative Analysis - Shopping Site")
```

The pie chart below represents the neutral percentage score of these three companies:



Part II : Naïve Bayes Algorithm

1. Data Preprocessing

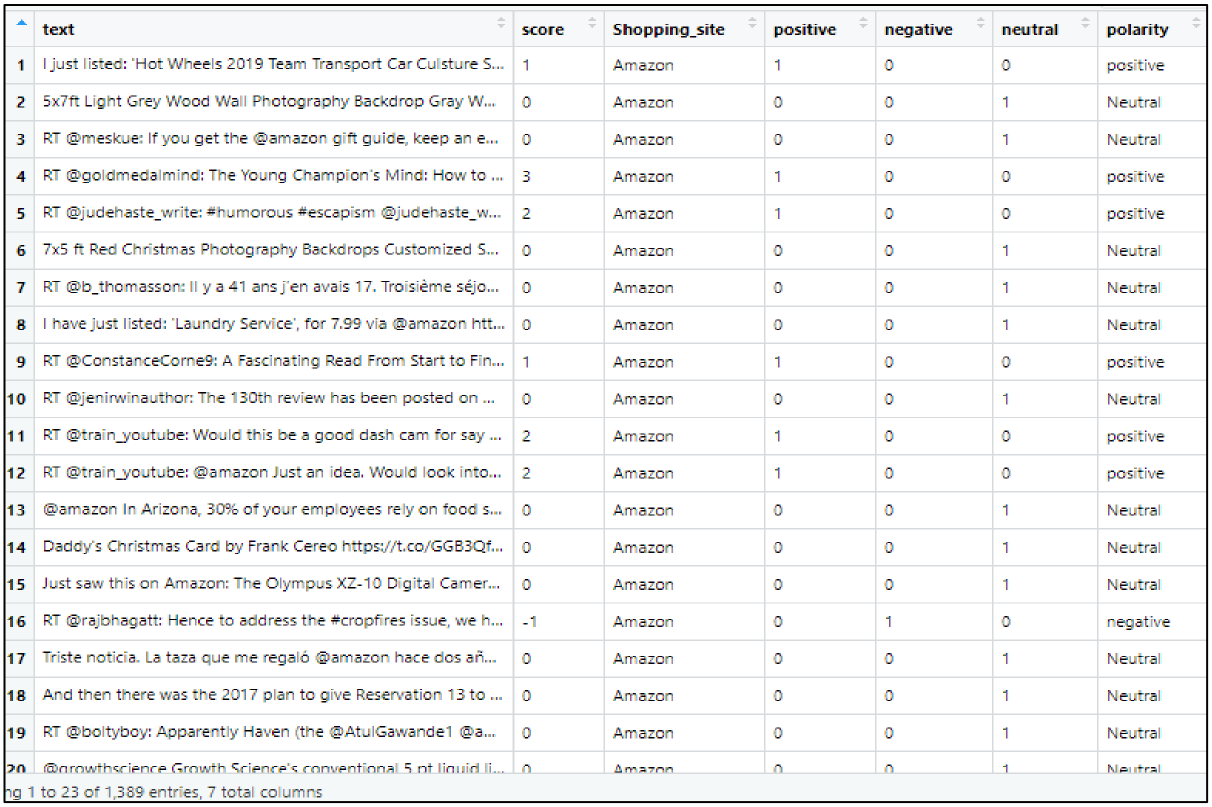
We will first load all the required libraries (packages).

a. Writing and Reading the Data as 'Amazon_Shopping_Site'

Code:

```
write.table(Amazon_Shopping_Site,"your R_Workspace directory/Amazon_Shopping_Site.csv",
append=T, row.names=F, col.names=T,sep=",",)

Amazon_Shopping_Site_csv <-read.csv("your R_Workspace directory/Amazon_Shopping_Site.csv",
header = TRUE, encoding = "UCS-2LE")

View(Amazon_Shopping_Site_csv)
```

Output:



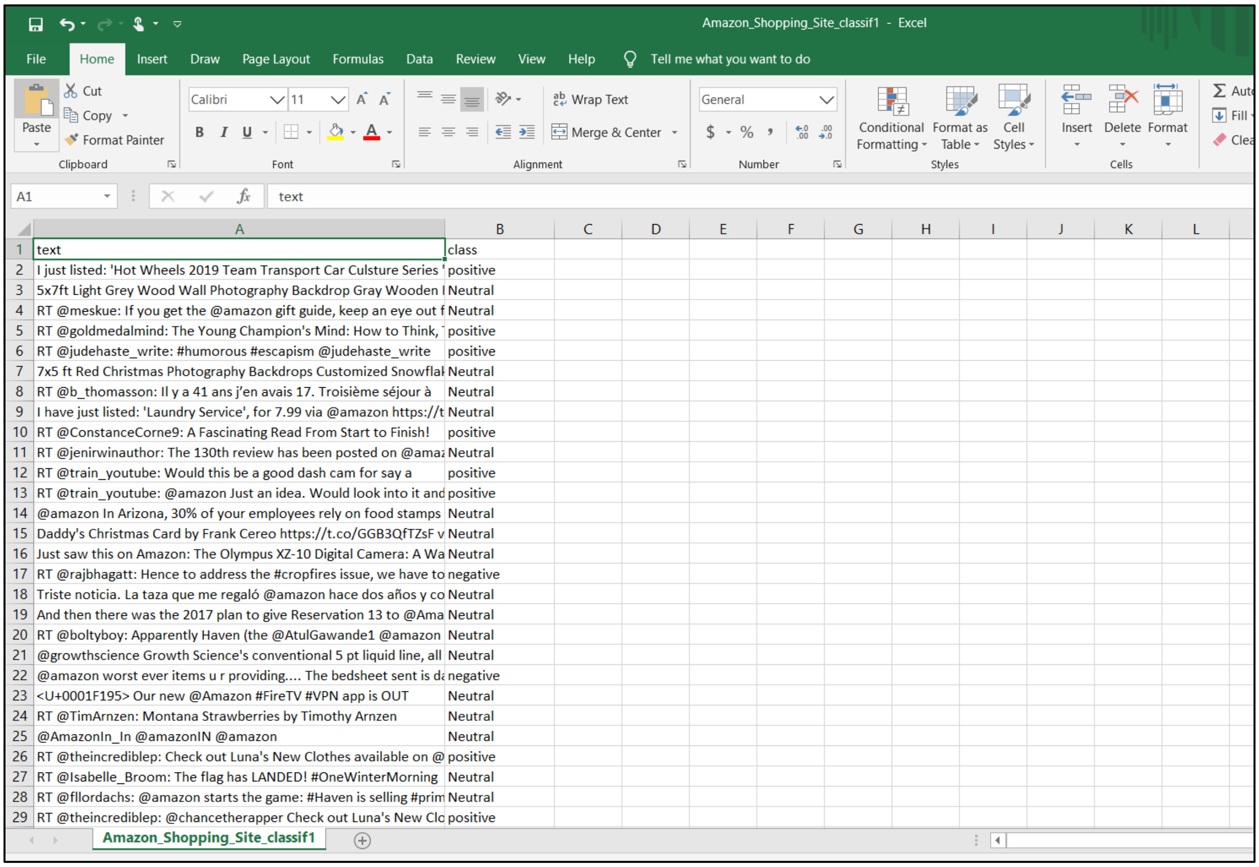
	text	score	Shopping_site	positive	negative	neutral	polarity
1	I just listed: 'Hot Wheels 2019 Team Transport Car Culsture S...	1	Amazon	1	0	0	positive
2	5x7ft Light Grey Wood Wall Photography Backdrop Gray W...	0	Amazon	0	0	1	Neutral
3	RT @meskue: If you get the @amazon gift guide, keep an e...	0	Amazon	0	0	1	Neutral
4	RT @goldmedalmind: The Young Champion's Mind: How to ...	3	Amazon	1	0	0	positive
5	RT @judehaste_write: #humorous #escapism @judehaste_w...	2	Amazon	1	0	0	positive
6	7x5 ft Red Christmas Photography Backdrops Customized S...	0	Amazon	0	0	1	Neutral
7	RT @b_thomasson: Il y a 41 ans j'en avais 17. Troisième séjo...	0	Amazon	0	0	1	Neutral
8	I have just listed: 'Laundry Service', for 7.99 via @amazon htt...	0	Amazon	0	0	1	Neutral
9	RT @ConstanceCorne9: A Fascinating Read From Start to Fin...	1	Amazon	1	0	0	positive
10	RT @jeninwinauthor: The 130th review has been posted on ...	0	Amazon	0	0	1	Neutral
11	RT @train_youtube: Would this be a good dash cam for say ...	2	Amazon	1	0	0	positive
12	RT @train_youtube: @amazon Just an idea. Would look into...	2	Amazon	1	0	0	positive
13	@amazon In Arizona, 30% of your employees rely on food s...	0	Amazon	0	0	1	Neutral
14	Daddy's Christmas Card by Frank Cereo https://t.co/GGB3Qf...	0	Amazon	0	0	1	Neutral
15	Just saw this on Amazon: The Olympus XZ-10 Digital Camer...	0	Amazon	0	0	1	Neutral
16	RT @rajbhagatt: Hence to address the #cropfires issue, we h...	-1	Amazon	0	1	0	negative
17	Triste noticia. La taza que me regaló @amazon hace dos añ...	0	Amazon	0	0	1	Neutral
18	And then there was the 2017 plan to give Reservation 13 to ...	0	Amazon	0	0	1	Neutral
19	RT @boltyboy: Apparently Haven (the @AtulGawande1 @a...	0	Amazon	0	0	1	Neutral
20	@growthscience Growth Science's conventional 5 pt liquid li...	0	Amazon	0	0	1	Neutral

ng 1 to 23 of 1,389 entries, 7 total columns

Note:

Create a ***.csv*** file with only one column "class" (Open Amazon_Shopping_Site.csv and create a new ***.csv*** file and save it as "Amazon_Shopping_Site_classif1.csv" file; open this file in excel and delete all the columns except polarity; now, change the column name polarity to "class" and select the filter to delete all the rows other than positive, negative, and neutral tweets and save it.)

Output:



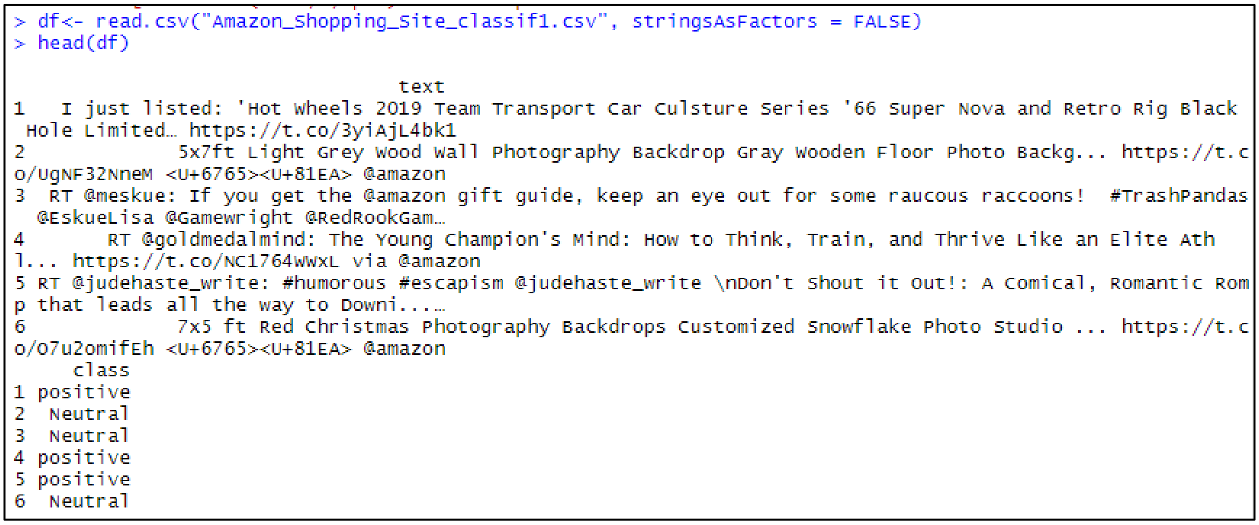
	A	B
1	text	class
2	I just listed: 'Hot Wheels 2019 Team Transport Car Culsture Series '	positive
3	5x7ft Light Grey Wood Wall Photography Backdrop Gray Wooden I	Neutral
4	RT @meskue: If you get the @amazon gift guide, keep an eye out f	Neutral
5	RT @goldmedalmind: The Young Champion's Mind: How to Think, '	positive
6	RT @judehaste_write: #humorous #escapism @judehaste_write	positive
7	7x5 ft Red Christmas Photography Backdrops Customized Snowflal	Neutral
8	RT @b_thomasson: Il y a 41 ans j'en avais 17. Troisième séjour à	Neutral
9	I have just listed: 'Laundry Service', for 7.99 via @amazon https://t	Neutral
10	RT @ConstanceCorne9: A Fascinating Read From Start to Finish!	positive
11	RT @jenirwinauthor: The 130th review has been posted on @amaz	Neutral
12	RT @train_youtube: Would this be a good dash cam for say a	positive
13	RT @train_youtube: @amazon Just an idea. Would look into it and	positive
14	@amazon In Arizona, 30% of your employees rely on food stamps	Neutral
15	Daddy's Christmas Card by Frank Cereo https://t.co/GGB3QfTZsF v	Neutral
16	Just saw this on Amazon: The Olympus XZ-10 Digital Camera: A Wa	Neutral
17	RT @rajbhagatt: Hence to address the #cropfires issue, we have to	negative
18	Triste noticia. La taza que me regaló @amazon hace dos años y co	Neutral
19	And then there was the 2017 plan to give Reservation 13 to @Ama	Neutral
20	RT @boltyboy: Apparently Haven (the @AtulGawande1 @amazon	Neutral
21	@growthscience Growth Science's conventional 5 pt liquid line, all	Neutral
22	@amazon worst ever items u r providing.... The bedsheet sent is da	negative
23	<U+0001F195> Our new @Amazon #FireTV #VPN app is OUT	Neutral
24	RT @TimArnzen: Montana Strawberries by Timothy Arnzen	Neutral
25	@AmazonIn_In @amazonIN @amazon	Neutral
26	RT @theincrediblep: Check out Luna's New Clothes available on @	positive
27	RT @Isabelle_Broom: The flag has LANDED! #OneWinterMorning	Neutral
28	RT @fllordachs: @amazon starts the game: #Haven is selling #prim	Neutral
29	RT @theincrediblep: @chancetherapper Check out Luna's New Clo	positive

Now, read the new file, "Amazon_Shopping_Site_classif1.csv".

Code:

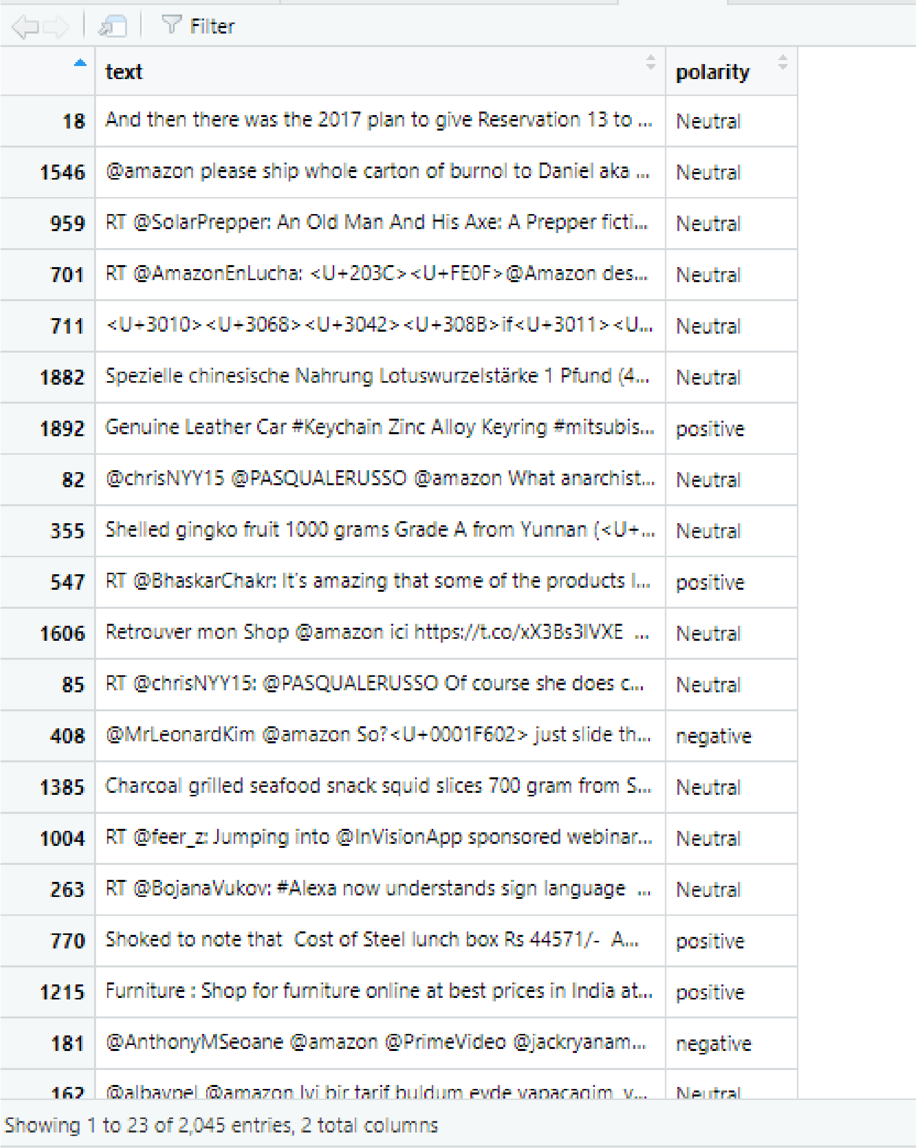
```
df<- read.csv("Amazon_Shopping_Site_classif1.csv", stringsAsFactors = FALSE)
head(df)
```

Output 1:



```
> df<- read.csv("Amazon_Shopping_Site_classif1.csv", stringsAsFactors = FALSE)
> head(df)

                                text
1   I just listed: 'Hot wheels 2019 Team Transport Car Culsture Series '66 Super Nova and Retro Rig Black
 Hole Limited… https://t.co/3yiAjL4bk1
2          5x7ft Light Grey Wood Wall Photography Backdrop Gray Wooden Floor Photo Backg... https://t.c
o/UgNF32NneM <U+6765><U+81EA> @amazon
3  RT @meskue: If you get the @amazon gift guide, keep an eye out for some raucous raccoons!  #TrashPandas
  @EskueLisa @Gamewright @RedRookGam…
4       RT @goldmedalmind: The Young Champion's Mind: How to Think, Train, and Thrive Like an Elite Ath
l... https://t.co/NC1764wWxL via @amazon
5 RT @judehaste_write: #humorous #escapism @judehaste_write \nDon't Shout it Out!: A Comical, Romantic Rom
p that leads all the way to Downi....…
6          7x5 ft Red Christmas Photography Backdrops Customized Snowflake Photo Studio ... https://t.c
o/O7u2omifEh <U+6765><U+81EA> @amazon
     class
1 positive
2  Neutral
3  Neutral
4 positive
5 positive
6  Neutral
```

Output 2:



	text	polarity
18	And then there was the 2017 plan to give Reservation 13 to ...	Neutral
1546	@amazon please ship whole carton of burnol to Daniel aka ...	Neutral
959	RT @SolarPrepper: An Old Man And His Axe: A Prepper ficti...	Neutral
701	RT @AmazonEnLucha: <U+203C><U+FE0F>@Amazon des...	Neutral
711	<U+3010><U+3068><U+3042><U+308B>if<U+3011><U...	Neutral
1882	Spezielle chinesische Nahrung Lotuswurzelstärke 1 Pfund (4...	Neutral
1892	Genuine Leather Car #Keychain Zinc Alloy Keyring #mitsubis...	positive
82	@chrisNYY15 @PASQUALERUSSO @amazon What anarchist...	Neutral
355	Shelled gingko fruit 1000 grams Grade A from Yunnan (<U+...	Neutral
547	RT @BhaskarChakr: It's amazing that some of the products I...	positive
1606	Retrouver mon Shop @amazon ici https://t.co/xX3Bs3lVXE ...	Neutral
85	RT @chrisNYY15: @PASQUALERUSSO Of course she does c...	Neutral
408	@MrLeonardKim @amazon So?<U+0001F602> just slide th...	negative
1385	Charcoal grilled seafood snack squid slices 700 gram from S...	Neutral
1004	RT @feer_z: Jumping into @InVisionApp sponsored webinar...	Neutral
263	RT @BojanaVukov: #Alexa now understands sign language ...	Neutral
770	Shoked to note that Cost of Steel lunch box Rs 44571/- A...	positive
1215	Furniture : Shop for furniture online at best prices in India at...	positive
181	@AnthonyMSeoane @amazon @PrimeVideo @jackryanam...	negative
162	@albaypel @amazon Iyi bir tarif buldum evde yapacagim v...	Neutral

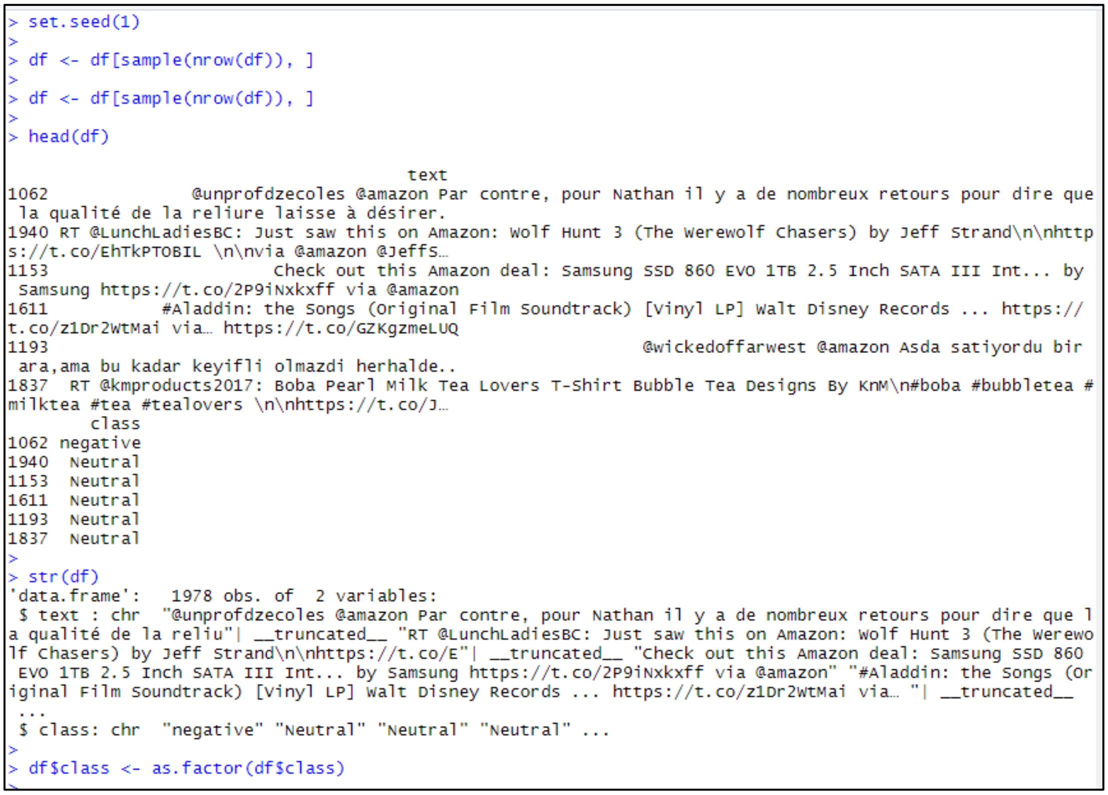
Showing 1 to 23 of 2,045 entries, 2 total columns

b. Randomize the dataset and convert the 'class' variable from character to factor.

Code:

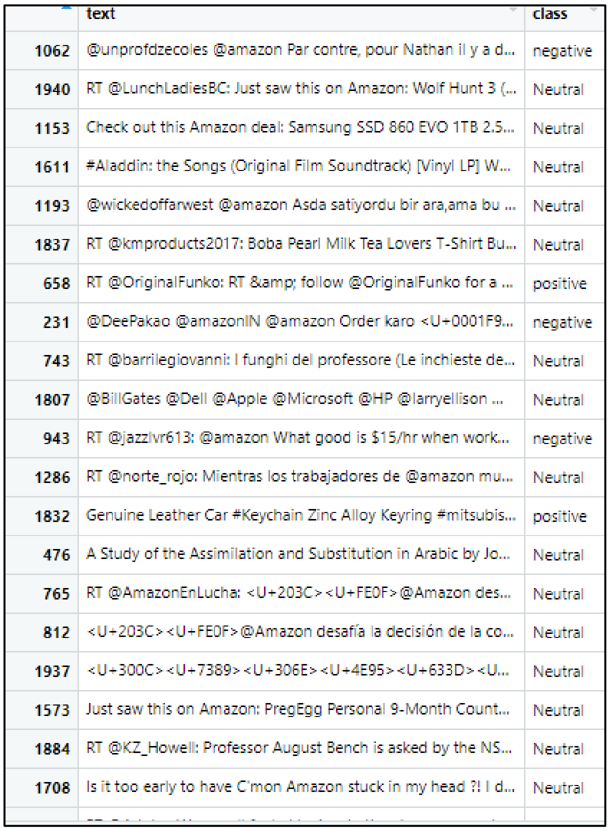
```
set.seed(1)

df <- df[sample(nrow(df)),]
df <- df[sample(nrow(df)),]

head(df)
str(df)

df$class <- as.factor(df$class)
```

Output 1:



```
> set.seed(1)
>
> df <- df[sample(nrow(df)), ]
>
> df <- df[sample(nrow(df)), ]
>
> head(df)

                                      text
1062              @unprofdzecoles @amazon Par contre, pour Nathan il y a de nombreux retours pour dire que
 la qualité de la reliure laisse à désirer.
1940 RT @LunchLadiesBC: Just saw this on Amazon: Wolf Hunt 3 (The Werewolf Chasers) by Jeff Strand\n\nhttp
s://t.co/EhTkPTOBIL \n\nvia @amazon @JeffS…
1153                      Check out this Amazon deal: Samsung SSD 860 EVO 1TB 2.5 Inch SATA III Int... by
 Samsung https://t.co/2P9iNxkxff via @amazon
1611           #Aladdin: the Songs (Original Film Soundtrack) [Vinyl LP] Walt Disney Records ... https://
t.co/z1Dr2WtMai via… https://t.co/GZKgzmeLUQ
1193                                                          @wickedoffarwest @amazon Asda satiyordu bir
 ara,ama bu kadar keyifli olmazdi herhalde..
1837  RT @kmproducts2017: Boba Pearl Milk Tea Lovers T-Shirt Bubble Tea Designs By KnM\n#boba #bubbletea #
milktea #tea #tealovers \n\nhttps://t.co/J…
        class
1062 negative
1940  Neutral
1153  Neutral
1611  Neutral
1193  Neutral
1837  Neutral
>
> str(df)
'data.frame':	1978 obs. of  2 variables:
 $ text : chr  "@unprofdzecoles @amazon Par contre, pour Nathan il y a de nombreux retours pour dire que l
a qualité de la reliu"| __truncated__ "RT @LunchLadiesBC: Just saw this on Amazon: Wolf Hunt 3 (The Werewo
lf Chasers) by Jeff Strand\n\nhttps://t.co/E"| __truncated__ "Check out this Amazon deal: Samsung SSD 860
 EVO 1TB 2.5 Inch SATA III Int... by Samsung https://t.co/2P9iNxkxff via @amazon" "#Aladdin: the Songs (Or
iginal Film Soundtrack) [Vinyl LP] Walt Disney Records ... https://t.co/z1Dr2WtMai via… "| __truncated__
 ...
 $ class: chr  "negative" "Neutral" "Neutral" "Neutral" ...
>
> df$class <- as.factor(df$class)
```

Output 2:



	text	class
1062	@unprofdzecoles @amazon Par contre, pour Nathan il y a d...	negative
1940	RT @LunchLadiesBC: Just saw this on Amazon: Wolf Hunt 3 (...	Neutral
1153	Check out this Amazon deal: Samsung SSD 860 EVO 1TB 2.5...	Neutral
1611	#Aladdin: the Songs (Original Film Soundtrack) [Vinyl LP] W...	Neutral
1193	@wickedoffarwest @amazon Asda satiyordu bir ara,ama bu ...	Neutral
1837	RT @kmproducts2017: Boba Pearl Milk Tea Lovers T-Shirt Bu...	Neutral
658	RT @OriginalFunko: RT &amp; follow @OriginalFunko for a ...	positive
231	@DeePakao @amazonIN @amazon Order karo <U+0001F9...	negative
743	RT @barrilegiovanni: I funghi del professore (Le inchieste de...	Neutral
1807	@BillGates @Dell @Apple @Microsoft @HP @larryellison ...	Neutral
943	RT @jazzlvr613: @amazon What good is $15/hr when work...	negative
1286	RT @norte_rojo: Mientras los trabajadores de @amazon mu...	Neutral
1832	Genuine Leather Car #Keychain Zinc Alloy Keyring #mitsubis...	positive
476	A Study of the Assimilation and Substitution in Arabic by Jo...	Neutral
765	RT @AmazonEnLucha: <U+203C><U+FE0F>@Amazon des...	Neutral
812	<U+203C><U+FE0F>@Amazon desafía la decisión de la co...	Neutral
1937	<U+300C><U+7389><U+306E><U+4E95><U+633D><U...	Neutral
1573	Just saw this on Amazon: PregEgg Personal 9-Month Count...	Neutral
1884	RT @KZ_Howell: Professor August Bench is asked by the NS...	Neutral
1708	Is it too early to have C'mon Amazon stuck in my head ?! I d...	Neutral

c. Bag of Words Tokenization

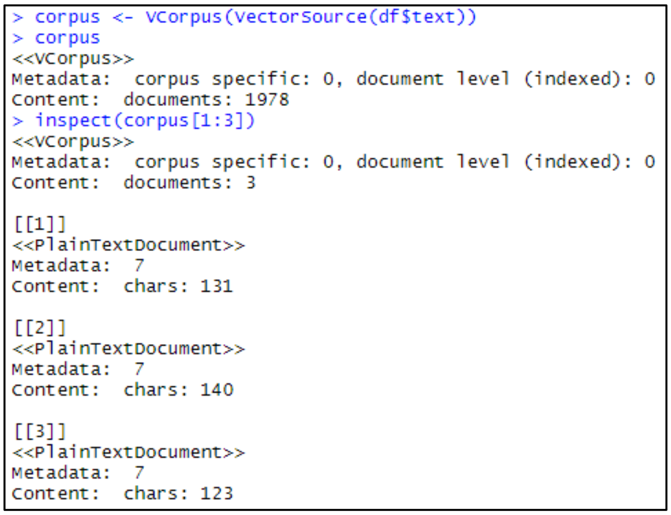
In this approach, we represent each word in a document as a token (or feature) and each document as a vector of features. In addition, for simplicity, we disregard word order and focus only on the number of occurrences of each word, which means that we represent each document as a multi-set ‘bag’ of words.

We first prepare a corpus of all the documents in the dataframe.

Code:

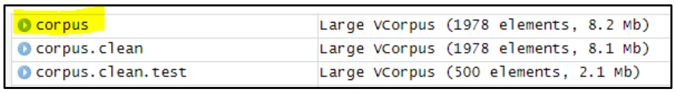
```
corpus <- VCorpus(VectorSource(df$text))

corpus
inspect(corpus[1:3])
```

Output 1:



```
> corpus <- VCorpus(VectorSource(df$text))
> corpus
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 1978
> inspect(corpus[1:3])
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 3

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 131

[[2]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 140

[[3]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 123
```

Output 2:



```
corpus                      Large VCorpus (1978 elements, 8.2 Mb)
corpus.clean                Large VCorpus (1978 elements, 8.1 Mb)
corpus.clean.test           Large VCorpus (500 elements, 2.1 Mb)
```

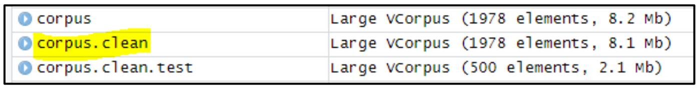
d. Data Cleanup

We clean up the corpus by eliminating numbers, punctuation, and white space and by converting to lowercase. In addition, we discard common stop words, such as “his”, “our”, “hadn’t”, couldn’t“, etc.

Code:

```
>corpus.clean <- corpus %>% tm_map(content_transformer(tolower)) %>% tm_map(removePunctuation)
%>% tm_map(removeNumbers) %>% tm_map(removeWords, stopwords(kind="en"))
%>%tm_map(stripWhitespace)
```

Output:



```
corpus                      Large VCorpus (1978 elements, 8.2 Mb)
corpus.clean                Large VCorpus (1978 elements, 8.1 Mb)
corpus.clean.test           Large VCorpus (500 elements, 2.1 Mb)
```

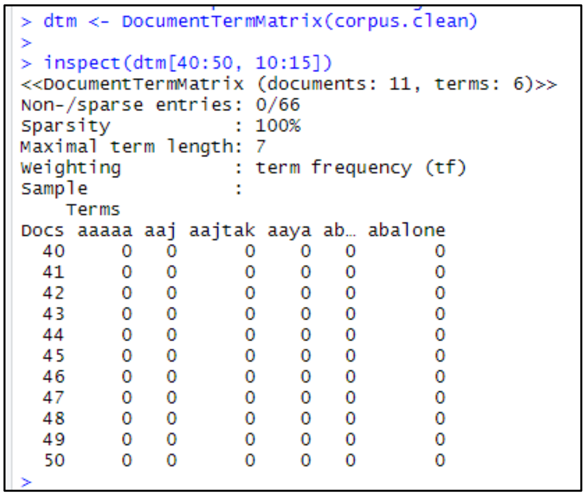
e. Matrix representation of Bag of Words: The Document Term Matrix (DTM)

We represent the bag of words tokens with a document term matrix (DTM). The rows of the DTM correspond to the documents in the collection, the columns correspond to the terms, and its elements are the term frequencies.

Code:

```
>dtm <- DocumentTermMatrix(corpus.clean)
>inspect(dtm[40:50, 10:15])
```

Output:



```
> dtm <- DocumentTermMatrix(corpus.clean)
>
> inspect(dtm[40:50, 10:15])
<<DocumentTermMatrix (documents: 11, terms: 6)>>
Non-/sparse entries: 0/66
Sparsity           : 100%
Maximal term length: 7
Weighting          : term frequency (tf)
Sample             :
    Terms
Docs aaaaa aaj aajtak aaya ab… abalone
  40     0   0      0    0   0       0
  41     0   0      0    0   0       0
  42     0   0      0    0   0       0
  43     0   0      0    0   0       0
  44     0   0      0    0   0       0
  45     0   0      0    0   0       0
  46     0   0      0    0   0       0
  47     0   0      0    0   0       0
  48     0   0      0    0   0       0
  49     0   0      0    0   0       0
  50     0   0      0    0   0       0
>
```

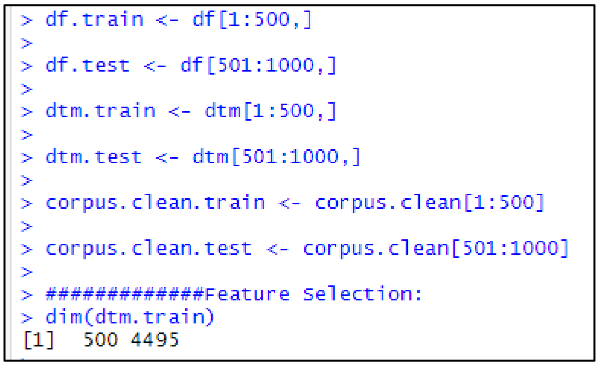
2. Partitioning the Data

We create 70:30 partitions of the dataframe, corpus, and DTM for training and testing purposes.

Code:

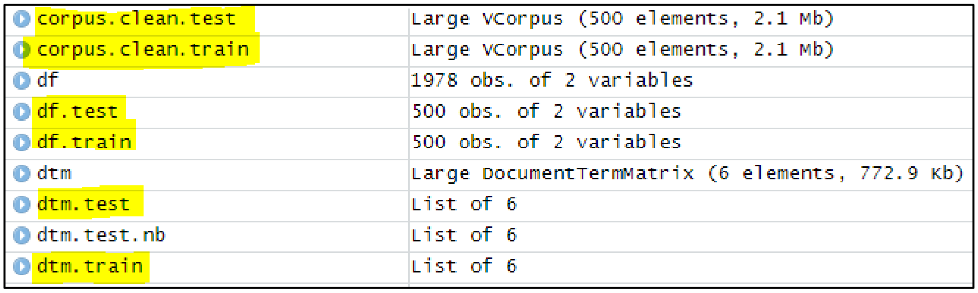
```
>df.train <- df[1:1200,]
>df.test <- df[1201:1554,]
>dtm.train <- dtm[1:1200,]
>dtm.test <- dtm[1201:1554,]
>corpus.clean.train <- corpus.clean[1:1200]
>corpus.clean.test <- corpus.clean[1201:1554]
>dim(dtm.train)
```

Output:



```
> df.train <- df[1:500,]
>
> df.test <- df[501:1000,]
>
> dtm.train <- dtm[1:500,]
>
> dtm.test <- dtm[501:1000,]
>
> corpus.clean.train <- corpus.clean[1:500]
>
> corpus.clean.test <- corpus.clean[501:1000]
>
> #############Feature Selection:
> dim(dtm.train)
[1]  500 4495
```

Output:



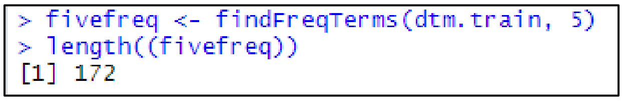
corpus.clean.test	Large VCorpus (500 elements, 2.1 Mb)
corpus.clean.train	Large VCorpus (500 elements, 2.1 Mb)
df	1978 obs. of 2 variables
df.test	500 obs. of 2 variables
df.train	500 obs. of 2 variables
dtm	Large DocumentTermMatrix (6 elements, 772.9 Kb)
dtm.test	List of 6
dtm.test.nb	List of 6
dtm.train	List of 6

The DTM contains many features, but not all of them are useful for classification. We reduce the number of features by ignoring the words that appear in less than five reviews. To do this, we use the ‘findFreqTerms’ function to indentify frequent words, and then we restrict the DTM to use only the frequent words using the ‘dictionary’ option.

Code:

```
fivefreq <- findFreqTerms(dtm.train, 5)
length((fivefreq))
```

Output:

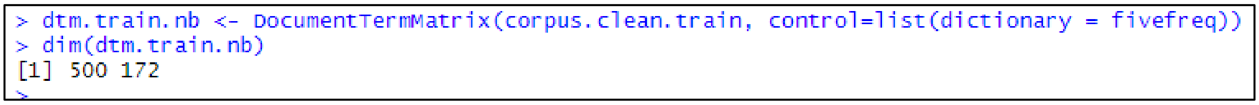


```
> fivefreq <- findFreqTerms(dtm.train, 5)
> length((fivefreq))
[1] 172
```

Code:

```
dtm.train.nb <- DocumentTermMatrix(corpus.clean.train, control=list(dictionary = fivefreq))
dim(dtm.train.nb)
```

Output:

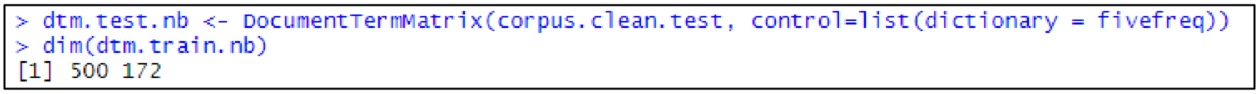


```
> dtm.train.nb <- DocumentTermMatrix(corpus.clean.train, control=list(dictionary = fivefreq))
> dim(dtm.train.nb)
[1] 500 172
>
```

Code:

```
dtm.test.nb <- DocumentTermMatrix(corpus.clean.test, control=list(dictionary = fivefreq))
dim(dtm.train.nb)
```

Output:



```
> dtm.test.nb <- DocumentTermMatrix(corpus.clean.test, control=list(dictionary = fivefreq))
> dim(dtm.train.nb)
[1] 500 172
```

Naïve Bayes algorithm

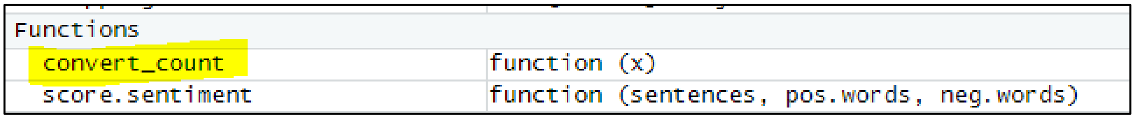
The Naive Bayes text classification algorithm is essentially an application of Bayes theorem (with a strong independence assumption) to documents and classes. In this method, the term frequencies are replaced by Boolean presence/absence features. The logic behind this is that for sentiment classification, word occurrence matters more than word frequency.

a. Function to convert the word frequencies to yes (presence) and no (absence)labels:

Code:

```
convert_count <- function(x) {
         y <- ifelse(x > 0, 1,0)
         y <- factor(y, levels=c(0,1), labels=c("No", "Yes"))
         y
}
```

Output:



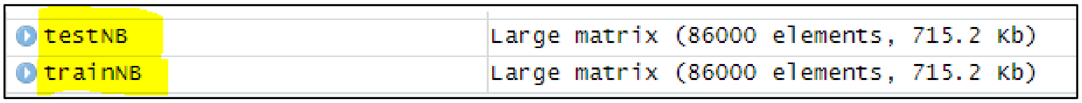
```
Functions
  convert_count                     function (x)
  score.sentiment                   function (sentences, pos.words, neg.words)
```

b. Applying the convert_count function to get the final training and testing DTMs:

Code:

```
trainNB <- apply(dtm.train.nb, 2, convert_count)
testNB <- apply(dtm.test.nb, 2, convert_count)
```

Output:



```
testNB                              Large matrix (86000 elements, 715.2 Kb)
trainNB                             Large matrix (86000 elements, 715.2 Kb)
```

3. Training the Naive Bayes Model

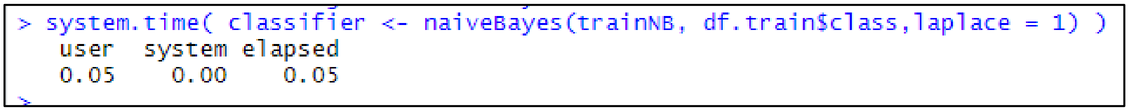
To train the model, we use the Naive Bayes function from the ‘e1071’ package. Since Naive Bayes evaluates the products of probabilities, we need some way of assigning nonzero probabilities to words that do not appear in the sample.

Train the classifier.

Code:

```
system.time( classifier <- naiveBayes(trainNB, df.train$class,laplace = 1) )
```

Output:



```
> system.time( classifier <- naiveBayes(trainNB, df.train$class,laplace = 1) )
   user  system elapsed 
   0.05    0.00    0.05 
```

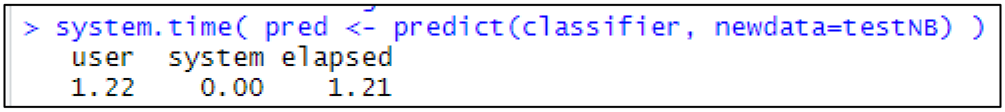
4. Testing the Predictions

Use the NB classifier we built to make predictions on the test set:

Code:

```
system.time( pred <- predict(classifier, newdata=testNB) )
```

Output:



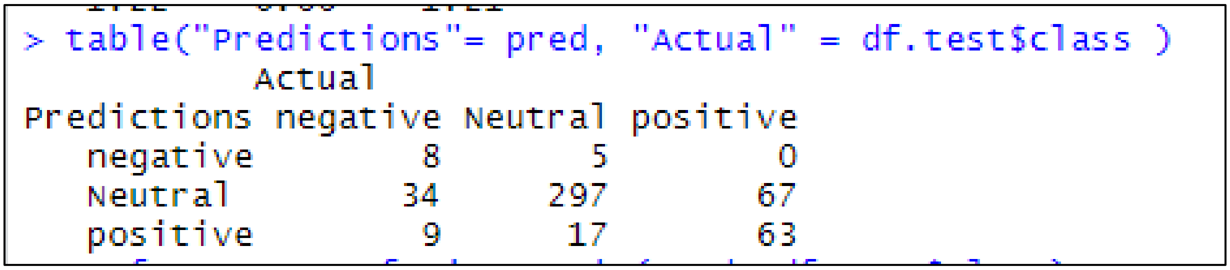
```
> system.time( pred <- predict(classifier, newdata=testNB) )
   user  system elapsed 
   1.22    0.00    1.21 
```

Create a truth table by tabulating the predicted class labels with the actual class labels:

Code:

```
table("Predictions"= pred, "Actual" = df.test$class )
```

Output:



```
> table("Predictions"= pred, "Actual" = df.test$class )
           Actual
Predictions negative Neutral positive
   negative        8       5        0
   Neutral        34     297       67
   positive        9      17       63
```

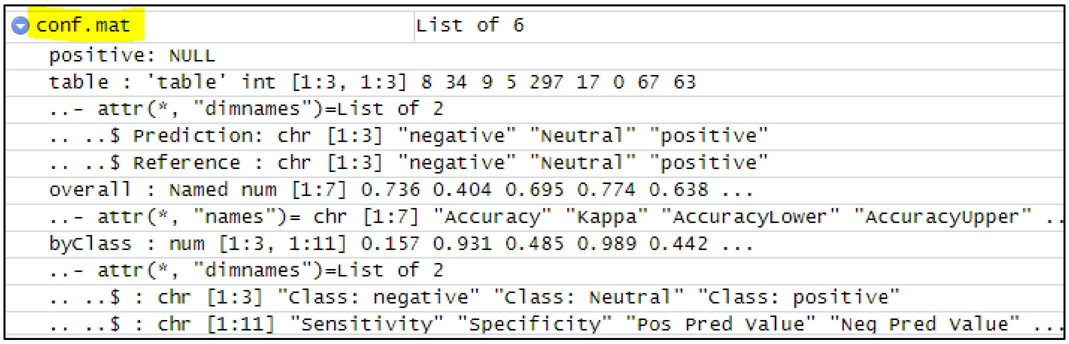
5. Confusion Matrix

Prepare the confusion matrix:

Code:

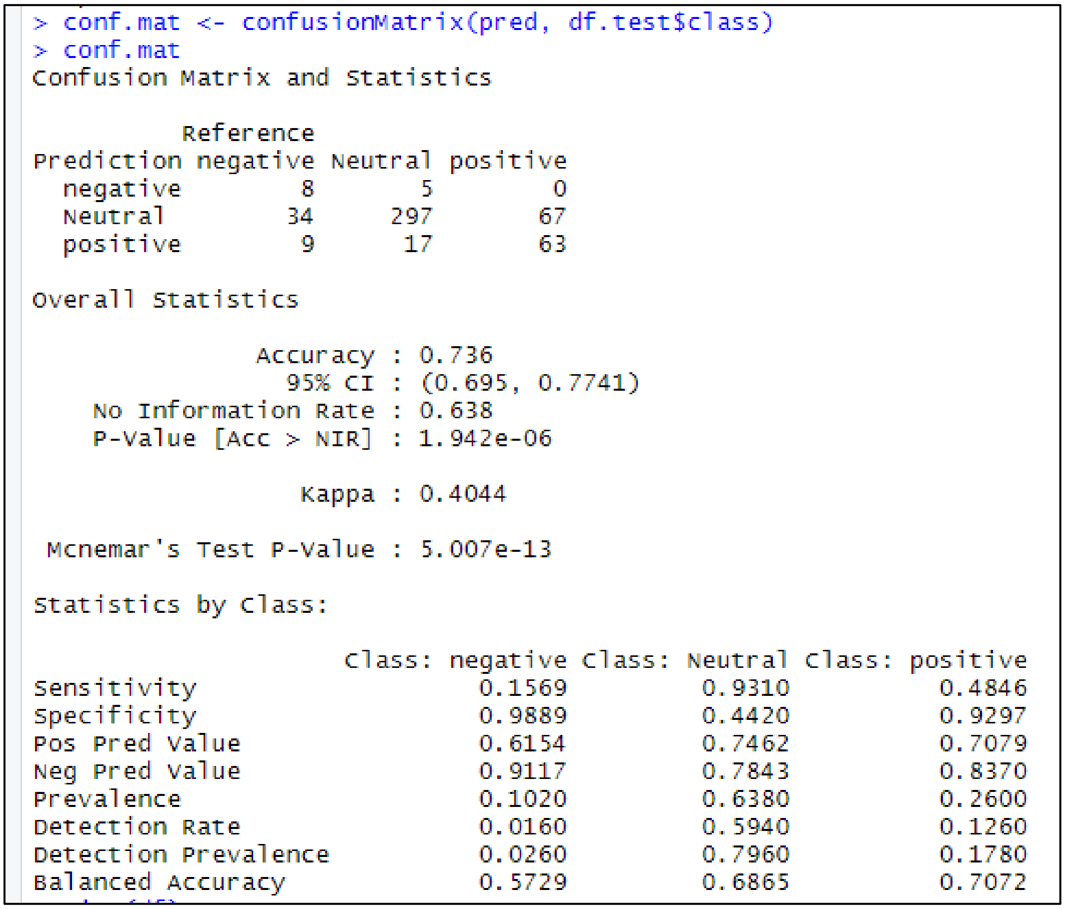
```
conf.mat <- confusionMatrix(pred, df.test$class)
conf.mat
```

Output 1:



```
conf.mat                                List of 6
  positive: NULL
  table : 'table' int [1:3, 1:3] 8 34 9 5 297 17 0 67 63
  ..- attr(*, "dimnames")=List of 2
  .. ..$ Prediction: chr [1:3] "negative" "Neutral" "positive"
  .. ..$ Reference : chr [1:3] "negative" "Neutral" "positive"
  overall : Named num [1:7] 0.736 0.404 0.695 0.774 0.638 ...
  ..- attr(*, "names")= chr [1:7] "Accuracy" "Kappa" "AccuracyLower" "AccuracyUpper" ..
  byClass : num [1:3, 1:11] 0.157 0.931 0.485 0.989 0.442 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "Class: negative" "Class: Neutral" "Class: positive"
  .. ..$ : chr [1:11] "Sensitivity" "Specificity" "Pos Pred Value" "Neg Pred Value" ...
```

Output 2:



```
> conf.mat <- confusionMatrix(pred, df.test$class)
> conf.mat
Confusion Matrix and Statistics

          Reference
Prediction negative Neutral positive
  negative        8       5        0
  Neutral        34     297       67
  positive        9      17       63

Overall Statistics

               Accuracy : 0.736
                 95% CI : (0.695, 0.7741)
    No Information Rate : 0.638
    P-Value [Acc > NIR] : 1.942e-06

                  Kappa : 0.4044

 Mcnemar's Test P-Value : 5.007e-13

Statistics by Class:

                     Class: negative Class: Neutral Class: positive
Sensitivity                   0.1569         0.9310          0.4846
Specificity                   0.9889         0.4420          0.9297
Pos Pred Value                0.6154         0.7462          0.7079
Neg Pred Value                0.9117         0.7843          0.8370
Prevalence                    0.1020         0.6380          0.2600
Detection Rate                0.0160         0.5940          0.1260
Detection Prevalence          0.0260         0.7960          0.1780
Balanced Accuracy             0.5729         0.6865          0.7072
```

CONCLUSION

In the first part, we analysed tweets for competing e-commerce brands and characterised the sentiment score for each tweet as positive, negative, and neutral. With this polarity data, we have created a variety of charts to enable a comparative study of brand value, in terms of the customer's response on Twitter. Our analysis shows that Myntra is the most-liked brand out of the three brands (Amazon, Myntra, and Flipkart) we analyzed for this project . Customer tweets for Myntra were mostly of positive sentiment as opposed to Flipkart, which had tweets mostly of negative sentiment and Amazon, which had tweets mostly of neutral sentiment.

In the second part, we trained the Naïve Bayes algorithm, using the tweet and polarity data from part one of the sentiment analysis for the prediction of new tweets. Our results show an accuracy of 73.73%; higher accuracy can be achieved with more training on a larger dataset. We also calculated sensitivity, specificity, and the P-Value of test data through confusion matrix for better insights.