Class Assignment-1 of INT-254

Topic:- Social Media Sentiment Analysis Using Twitter Dataset

Professor:- Dr. Dhanpratap Singh

Submitted By:-

Purushottam

Purushottam Sahu
RegNo:-12015687
Section:- KM-118
RollNo:- RKM118A21

Student Declaration

I,Purushottam Sahu,12015687,hereby declare that the project entitled is an outcome of my own efforts with the help of Prof. Dr. Dhanpratap Singh for the partial fulfillment of the requirements to complete this project on Social Media Sentiment Analysis Using Twitter Dataset.

Sign:- Purushottam

Name:- Purushottam Sahu

Date:- 02/11/2022

Acknowledgement

I would like to express my special thanks of gratitude to the teacher and instructor of the course Machine Learning who provide me this project to learn new things.

I would like to also thank my own college Lovely Professional University for offering such a project which not only improve my technical skill but also taught me another new technology.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance to complete this task.

Purushottam Sahu
Reg no:- 12015687

Table of Contents

Introduction

Social media has opened a whole new world for people around the globe. People are just a click away from getting huge chunk of information. With information comes people's opinion and with this comes the positive and negative outlook of people regarding a topic. Sometimes this also results into bullying and passing on hate comments about someone or something.

So in this article we will use a data set containing a collection of tweets to detect the sentiment associated with a particular tweet and detect it as negative or positive accordingly using Machine Learning.

Data Set Description

Formally, given a training sample of tweets and labels, where **label '1'** denotes the tweet is **racist/ sexist** and **label '0'** denotes the tweet is **not racist/ sexist**,our objective is to predict the labels on the given test dataset.

- id : The id associated with the tweets in the given dataset.
- tweets : The tweets collected from various sources and having either positive or negative sentiments associated with it.
- label : A tweet with **label '0'** is of **positive sentiment** while a tweet with **label '1'** is of **negative sentiment.**

Importing the necessary packages

```
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import string
import nltk
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

%matplotlib inline
```

Reading the train.csv Pandas file

- In the first line we read the train.csv file using Pandas.

- In the second line as a safe backup we keep a copy of our original train.csv file. **We make a copy of train data so that even if we have to make any changes in this dataset we would not lose the original dataset.**

```
train = pd.read_csv('https://raw.githubusercontent.com/dD2405/Twitter_Sentiment_Analysis/master/train.csv')

train_original=train.copy()
train
```

Overview of the training dataset

1 to 25

index	id	label	tweet
0	1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in urð ±!!! ð ð ð ð ð ¦ð ¦ð ¦
4	5	0	factsguide: society now #motivation
5	6	0	[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get there. #allshowandnogo
6	7	0	@user camping tomorrow @user @user @user @user @user @user @user dannyâ€¦
7	8	0	the next school year is the year for exams.ð ¯ can't think about that ð #school #exams #hate #imagine #actorslife #revolutionschool #girl
8	9	0	we won!!! love the land!!! #allin #cavs #champions #cleveland #clevelandcavaliers â€¦
9	10	0	@user @user welcome here ! i'm it's so #gr8 !
10	11	0	â #ireland consumer price index (mom) climbed from previous 0.2% to 0.5% in may #blog #silver #gold #forex
11	12	0	we are so selfish. #orlando #standwithorlando #pulseshooting #orlandoshooting #biggerproblems #selfish #heabreaking #values #love #
12	13	0	i get to see my daddy today!! #80days #gettingfed
13	14	1	@user #cnn calls #michigan middle school 'build the wall' chant " #tcot
14	15	1	no comment! in #australia #opkillingbay #seashepherd #helpcovedolphins #thecove #helpcovedolphins
15	16	0	ouch...junior is angryð #got7 #junior #yugyoem #omg
16	17	0	i am thankful for having a paner. #thankful #positive
17	18	1	retweet if you agree!
18	19	0	its #friday! ð smiles all around via ig user: @user #cookies make people
19	20	0	as we all know, essential oils are not made of chemicals.
20	21	0	#euro2016 people blaming ha for conceded goal was it fat rooney who gave away free kick knowing bale can hit them from there.
21	22	0	sad little dude.. #badday #coneofshame #cats #pissed #funny #laughs
22	23	0	product of the day: happy man #wine tool who's it's the #weekend? time to open up & drink up!

As you can see we have 3 attributes present in our dataset and a total of 31962 labeled tweets , ‘1’ standing for tweets with negative sentiment and ‘0’ for tweets with positive sentiments.

Reading the train.csv Pandas file

- In the first line we read the train.csv file using Pandas.

- In the second line as a safe backup we keep a copy of our original train.csv file. **We make a copy of train data so that even if we have**

to make any changes in this dataset we would not lose the original dataset.

```
test = pd.read_csv('https://raw.githubusercontent.com/dD2405/Twitter_Sentiment_Analysis/master/test.csv')

test_original=test.copy()
test
```

Overview of the training dataset

index	id	tweet
0	31963	#studiolife #aislife #requires #passion #dedication #willpower to find #newmaterialsâ€¦
1	31964	@user #white #supremacists want everyone to see the new â€ #birdsâ€ #movie â€ and hereâ€ s why
2	31965	safe ways to heal your #acne!! #altwaystoheal #healthy #healing!!
3	31966	is the hp and the cursed child book up for reservations already? if yes, where? if no, when? ð ð ð #harrypotter #pottermore #favorite
4	31967	3rd #bihday to my amazing, hilarious #nephew eli ahmir! uncle dave loves you and missesâ€¦
5	31968	choose to be :) #momtips
6	31969	something inside me dies ð ¦ð ¿â ¨ eyes ness #smokeyeyes #tired #lonely #sof #grungeâ ¦
7	31970	#finished#tattoo#inked#ink#loveitâ ¤ï¸ #â ¤ï¸ â ¤ï¸ â ¤ï¸ â ¤ï¸ #thanks#aleeee !!!
8	31971	@user @user @user i will never understand why my dad left me when i was so young.... :/ #deep #inthefeels
9	31972	#delicious #food #lovelife #capetown mannaepicure #resturantâ€¦
10	31973	1000dayswasted - narcosis infinite ep.. make me aware.. grinding neuro bass #lifestyle
11	31974	one of the world's greatest spoing events #lemans24 #teamaudi
12	31975	half way through the website now and #allgoingwell very
13	31976	good food, good life , #enjoy and ð ð ð ð ð this is called ~garlic bread~ ... #iloveitâ ¦
14	31977	i'll stand behind this #guncontrolplease #senselessshootings #taketheguns #comicrelief #stillsad
15	31978	i ate,i ate and i ate...ð ð #jamaisasthi #fish #curry #prawn #hilsa #foodfestival #foodies
16	31979	@user got my @user limited edition rain or shine set today!! ! @user @user @user @user
17	31980	& #love & #hugs & #kisses too! how to keep your #baby #parenting #healthcare
18	31981	ð ð ð #girls #sun #fave @ london, united kingdom
19	31982	thought factory: bbc neutrality on right wing fascism #politics #media #blm #brexit #trump #leadership >3
20	31983	hey guys tommorow is the last day of my exams i'm so happy yay
21	31984	@user @user @user #levyrroni #recuerdos memoriesð â ¤ð ð «ð #recuerdos #friends #life #triunfodelamor
22	31985	my mind is like ð ð ð ½ð but my body like ð ð ´ð µð ½.... #sleepy #stillallinð
23	31986	never been this down on myself in my entire life.

As we can see we have 2 **attributes** present here that is **'id'** and **'tweets'**. This is the dataset on which we are going to test our Machine Learning models so it is unlabeled

Data Pre-Processing

Data preprocessing can refer to manipulation or dropping of data before it is used in order to

ensure or enhance performance, and is an important step in the data mining process.

Let's begin with the pre-processing of our dataset.

STEP — 1 :
Combine the train.csv and test.csv files.

Pandas **dataframe.append()** function is used to append rows of other dataframe to the end of the given dataframe, returning a new dataframe object.

```
[7]  combine = train.append(test,ignore_index=True,sort=True)
```

STEP — 2
Removing Twitter Handles(@User)

In our analysis we can clearly see that the Twitter handles do not contribute anything significant to solve our problem. So it's better if we remove them in our dataset.

Given below is a user-defined function to remove unwanted text patterns from the tweets. It takes two arguments, one is the original string of text

and the other is the pattern of text that we want to remove from the string. The function returns the same input string but without the given pattern. We will use this function to remove the pattern '@user' from all the tweets in our data.

```
[8] def remove_pattern(text,pattern):

        # re.findall() finds the pattern i.e @user and puts it in a list for further task
        r = re.findall(pattern,text)

        # re.sub() removes @user from the sentences in the dataset
        for i in r:
            text = re.sub(i,"",text)

        return text
```

Here NumPy Vectorization **'np.vectorize()'** is used because it is much more faster than the conventional for loops when working on datasets of medium to large sizes.

```
[9] combine['Tidy_Tweets'] = np.vectorize(remove_pattern)(combine['tweet'], "@[\w]*")

    combine.head()
```

	id	label	tweet	Tidy_Tweets
0	1	0.0	@user when a father is dysfunctional and is s...	when a father is dysfunctional and is so sel...
1	2	0.0	@user @user thanks for #lyft credit i can't us...	thanks for #lyft credit i can't use cause th...
2	3	0.0	bihday your majesty	bihday your majesty
3	4	0.0	#model i love u take with u all the time in ...	#model i love u take with u all the time in ...
4	5	0.0	factsguide: society now #motivation	factsguide: society now #motivation

STEP — 3

Removing Punctuation, Numbers, and Special Characters

Punctuation, numbers and special characters do not help much. It is better to remove them from the text just as we removed the twitter handles. Here we will replace everything except characters and hashtags with spaces.

```
combine['Tidy_Tweets'] = combine['Tidy_Tweets'].str.replace("[^a-zA-Z#]", " ")

combine.head(10)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: The default value of regex w
  """Entry point for launching an IPython kernel.
```

	id	label	tweet	Tidy_Tweets
0	1	0.0	@user when a father is dysfunctional and is s...	when a father is dysfunctional and is so sel...
1	2	0.0	@user @user thanks for #lyft credit i can't us...	thanks for #lyft credit i can t use cause th...
2	3	0.0	bihday your majesty	bihday your majesty
3	4	0.0	#model i love u take with u all the time in ...	#model i love u take with u all the time in ...
4	5	0.0	factsguide: society now #motivation	factsguide society now #motivation
5	6	0.0	[2/2] huge fan fare and big talking before the...	huge fan fare and big talking before the...
6	7	0.0	@user camping tomorrow @user @user @user @use...	camping tomorrow danny
7	8	0.0	the next school year is the year for exams.ð ...	the next school year is the year for exams ...
8	9	0.0	we won!!! love the land!!! #allin #cavs #champ...	we won love the land #allin #cavs #champ...
9	10	0.0	@user @user welcome here ! i'm it's so #gr...	welcome here i m it s so #gr

STEP — 4

Removing Short Words

We have to be a little careful here in selecting the length of the words which we want to remove. So, I have decided to remove all the words having length 3 or less. These words are

also known as **Stop Words.**
For example, terms like “hmm”, “and”, “oh” are of very little use. It is better to get rid of them.

```
combine['Tidy_Tweets'] = combine['Tidy_Tweets'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))

combine.head(10)
```

	id	label	tweet	Tidy_Tweets
0	1	0.0	@user when a father is dysfunctional and is s...	when father dysfunctional selfish drags kids i...
1	2	0.0	@user @user thanks for #lyft credit i can't us...	thanks #lyft credit cause they offer wheelchai...
2	3	0.0	bihday your majesty	bihday your majesty
3	4	0.0	#model i love u take with u all the time in ...	#model love take with time
4	5	0.0	factsguide: society now #motivation	factsguide society #motivation
5	6	0.0	[2/2] huge fan fare and big talking before the...	huge fare talking before they leave chaos disp...
6	7	0.0	@user camping tomorrow @user @user @user @use...	camping tomorrow danny
7	8	0.0	the next school year is the year for exams.ð ...	next school year year exams think about that #...
8	9	0.0	we won!!! love the land!!! #allin #cavs #champ...	love land #allin #cavs #champions #cleveland #...
9	10	0.0	@user @user welcome here ! i'm it's so #gr...	welcome here

STEP — 5
Tokenization

Now we will tokenize all the cleaned tweets in our dataset. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens.

Here we tokenize our sentences because we will apply Stemming from the “NLTK” package in the next step.

```
[12] tokenized_tweet = combine['Tidy_Tweets'].apply(lambda x: x.split())

     tokenized_tweet.head()
```

```
0    [when, father, dysfunctional, selfish, drags, ...
1    [thanks, #lyft, credit, cause, they, offer, wh...
2                               [bihday, your, majesty]
3                     [#model, love, take, with, time]
4                   [factsguide, society, #motivation]
Name: Tidy_Tweets, dtype: object
```

STEP — 6

Stemming

Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word.

For example — “play”, “player”, “played”, “plays” and “playing” are the different variations of the word — “play”

```
from nltk import PorterStemmer

ps = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda x: [ps.stem(i) for i in x])

tokenized_tweet.head()
```

```
0    [when, father, dysfunct, selfish, drag, kid, i...
1    [thank, #lyft, credit, caus, they, offer, whee...
2                               [bihday, your, majesti]
3                     [#model, love, take, with, time]
4                         [factsguid, societi, #motiv]
Name: Tidy_Tweets, dtype: object
```

Now let’s stitch these tokens back together

```
[14] for i in range(len(tokenized_tweet)):
        tokenized_tweet[i] = ' '.join(tokenized_tweet[i])

    combine['Tidy_Tweets'] = tokenized_tweet
    combine.head()
```

	id	label	tweet	Tidy_Tweets
0	1	0.0	@user when a father is dysfunctional and is s...	when father dysfunct selfish drag kid into dys...
1	2	0.0	@user @user thanks for #lyft credit i can't us...	thank #lyft credit caus they offer wheelchair ...
2	3	0.0	bihday your majesty	bihday your majesti
3	4	0.0	#model i love u take with u all the time in ...	#model love take with time
4	5	0.0	factsguide: society now #motivation	factsguid societi #motiv

So finally these are the basic steps to follow when we have to Pre-Process a dataset containing textual data.

OK, so now we are done with our Data Pre-Processing stages.

Let's move on to our next step that is **Data Visualisation.**

Data Visualisation

So Data Visualisation is one of the most important steps in Machine Learning projects because it gives us an approximate idea about the dataset and what it is all about before

proceeding to apply different machine learning models.

So, let's dive in.

WordCloud

One of the popular visualisation techniques is **WordCloud.**

A WordCloud is a visualisation wherein the most frequent words appear in large size and the less frequent words appear in smaller sizes.

So, in Python we have a package for generating **WordCloud**.

Let's dive into the code to see how can we generate a **WordCloud.**

Importing packages necessary for generating a WordCloud

```
from wordcloud import WordCloud,ImageColorGenerator
from PIL import Image
import urllib
import requests
```

Generating WordCloud for tweets with label '0'.

Store all the words from the dataset which are non-racist/sexist.

```
[16] all_words_positive = ' '.join(text for text in combine['Tidy_Tweets'][combine['label']==0])
```

The code to generate the required **WordCloud**.

```
[17] # combining the image with the dataset
Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-PNG-Image.png', stream=True).raw))

# We use the ImageColorGenerator library from Wordcloud
# Here we take the color of the image and impose it over our wordcloud
image_colors = ImageColorGenerator(Mask)

# Now we use the WordCloud function from the wordcloud library
wc = WordCloud(background_color='black', height=1500, width=4000,mask=Mask).generate(all_words_positive)
```

```
plt.figure(figsize=(10,20))

# Here we recolor the words from the dataset to the image's color
# recolor just recolors the default colors to the image's blue color
# interpolation is used to smooth the image generated
plt.imshow(wc.recolor(color_func=image_colors),interpolation="hamming")

plt.axis('off')
plt.show()
```

We can see most of the words are positive or neutral. With happy, smile, and love being the most frequent ones. Hence, most of the frequent words are compatible with tweets in positive sentiment.

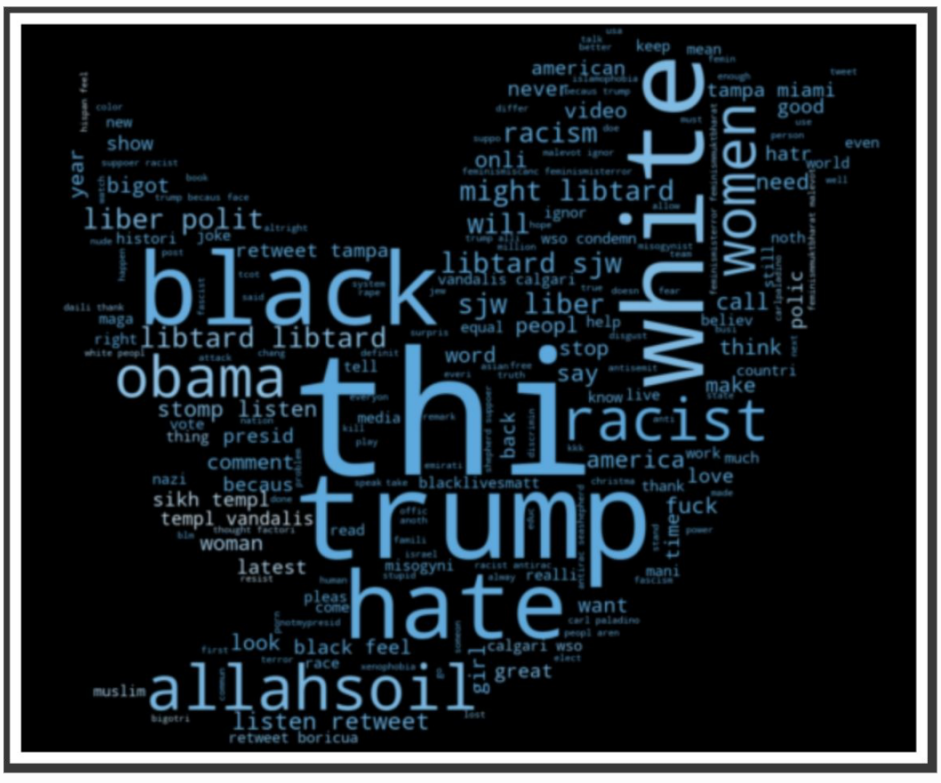
Generating WordCloud for tweets with label ‘1‘.

Store all the words from the dataset which are non-racist/sexist.

```
[21] all_words_negative = ' '.join(text for text in combine['Tidy_Tweets'][combine['label']==1])
```

The code to generate the required **WordCloud**.

```
[22] # combining the image with the dataset
     Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-PNG-Image.png', stream=True).raw))

     # We use the ImageColorGenerator library from Wordcloud
     # Here we take the color of the image and impose it over our wordcloud
     image_colors = ImageColorGenerator(Mask)

     # Now we use the WordCloud function from the wordcloud library
     wc = WordCloud(background_color='black', height=1500, width=4000,mask=Mask).generate(all_words_negative)

     # Size of the image generated
     plt.figure(figsize=(10,20))

     # Here we recolor the words from the dataset to the image's color
     # recolor just recolors the default colors to the image's blue color
     # interpolation is used to smooth the image generated
     plt.imshow(wc.recolor(color_func=image_colors),interpolation="gaussian")

     plt.axis('off')
     plt.show()
```



We can clearly see, most of the words have negative connotations. So, it seems we have a pretty good text data to work on.

Understanding the impact of Hashtags on tweets sentiment

Hash-tagging on Twitter can have a major impact when it comes to your follower count by using general and non-specific hashtags. If you hashtag general words, like **#creative,** or events, like **#TIFF,** that are going on, it is more likely that your tweet will reach beyond your follower list.

So we will look how we can extract the hashtags

and see which hashtags fall into which category.

Function to extract hashtags from tweets

```
[24] def Hashtags_Extract(x):
         hashtags=[]

         # Loop over the words in the tweet
         for i in x:
             ht = re.findall(r'#(\w+)',i)
             hashtags.append(ht)

         return hashtags
```

A nested list of all the hashtags from the positive reviews from the dataset.

```
ht_positive = Hashtags_Extract(combine['Tidy_Tweets'][combine['label']==0])

ht_positive
```

```
[['run'],
 ['lyft', 'disapoint', 'getthank'],
 [],
 ['model'],
 ['motiv'],
 ['allshowandnogo'],
 [],
 ['school', 'exam', 'hate', 'imagin', 'actorslif', 'revolutionschool', 'girl'],
 ['allin', 'cav', 'champion', 'cleveland', 'clevelandcavali'],
 [],
 ['ireland', 'blog', 'silver', 'gold', 'forex'],
 ['orlando',
  'standwithorlando',
  'pulseshoot',
  'orlandoshoot',
  'biggerproblem',
  'selfish',
  'heabreak',
  'valu',
  'love'],
```

Here we unnest the list

```
ht_positive_unnest = sum(ht_positive,[])
ht_positive_unnest
```

```
['run',
 'lyft',
 'disapoint',
 'getthank',
 'model',
 'motiv',
 'allshowandnogo',
 'school',
 'exam',
 'hate',
 'imagin',
 'actorslif',
 'revolutionschool',
 'girl',
 'allin',
 'cav',
 'champion',
 'cleveland',
 'clevelandcavali',
 'ireland',
 'blog',
 'silver',
 'gold',
 'forex',
```

A nested list of all the hashtags from the negative reviews from the dataset.

```
ht_negative = Hashtags_Extract(combine['Tidy_Tweets'][combine['label']==1])

ht_negative
```

```
[['cnn', 'michigan', 'tcot'],
 ['australia',
  'opkillingbay',
  'seashepherd',
  'helpcovedolphin',
  'thecov',
  'helpcovedolphin'],
 [],
 [],
 ['neverump', 'xenophobia'],
 ['love', 'peac'],
 [],
 ['race', 'ident', 'med'],
 ['altright', 'whitesupremaci'],
 ['linguist', 'race', 'power', 'raciolinguist'],
 ['brexit'],
 ['peopl', 'trump', 'republican'],
 ['michelleobama'],
 ['knick', 'golf'],
 ['jewishsupremacist'],
 ['libtard', 'sjw', 'liber', 'polit'],
 ['trash', 'hate'],
 [],
 [],
```

Here we unnest the list

```
ht_negative_unnest = sum(ht_negative,[])
ht_negative_unnest
```

```
['cnn',
 'michigan',
 'tcot',
 'australia',
 'opkillingbay',
 'seashepherd',
 'helpcovedolphin',
 'thecov',
 'helpcovedolphin',
 'neverump',
 'xenophobia',
 'love',
 'peac',
 'race',
 'ident',
 'med',
 'altright',
 'whitesupremaci',
 'linguist',
 'race',
 'power',
 'raciolinguist',
 'brexit',
 'peopl',
 'trump',
 'republican',
 'michelleobama',
```

Plotting Bar-plots

For Positive Tweets in the dataset

Counting the frequency of the words having Positive Sentiment.

```
word_freq_positive = nltk.FreqDist(ht_positive_unnest)

word_freq_positive
```

```
FreqDist({'love': 1654, 'posit': 917, 'smile': 676, 'healthi': 573, 'thank': 534, 'fun': 463, 'life': 425, 'affirm': 423, 'summer': 390,
'model': 375, ...})
```

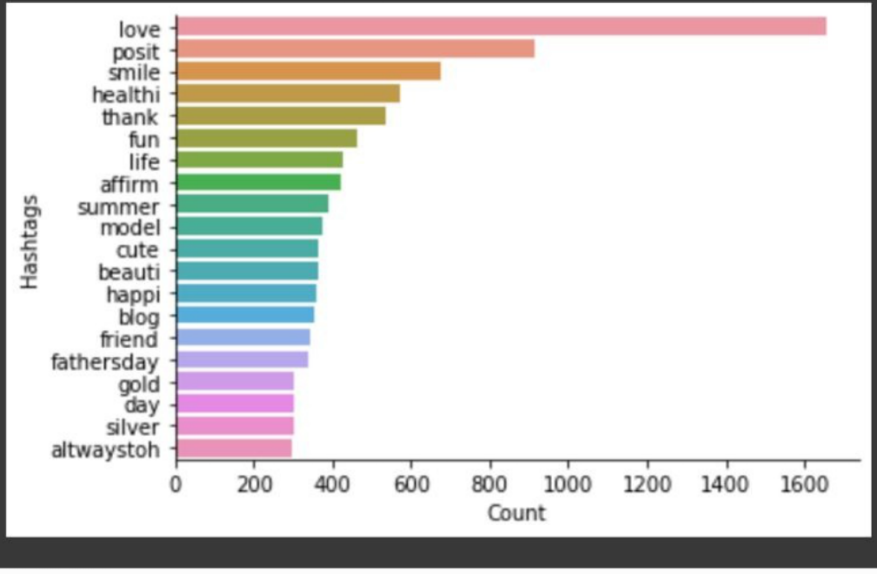
Creating a dataframe for the most frequently used words in hashtags

```
df_positive = pd.DataFrame({'Hashtags':list(word_freq_positive.keys()),'Count':list(word_freq_positive.values())})

df_positive.head(10)
```

	Hashtags	Count
0	run	72
1	lyft	2
2	disapoint	1
3	getthank	2
4	model	375
5	motiv	202
6	allshowandnogo	1
7	school	30
8	exam	9
9	hate	27

Plotting the barplot for the 20 most frequent words used for hashtags

```
df_positive_plot = df_positive.nlargest(20,columns='Count')

sns.barplot(data=df_positive_plot,y='Hashtags',x='Count')
sns.despine()
```



For Negative Tweets in the dataset

Counting the frequency of the words having Negative Sentiment

```
[34] word_freq_negative = nltk.FreqDist(ht_negative_unnest)

     word_freq_negative
```

```
FreqDist({'trump': 136, 'polit': 95, 'allahsoil': 92, 'liber': 81, 'libtard': 77, 'sjw': 75, 'retweet': 63, 'black': 46, 'miami': 46, 'hate': 37, ...})
```

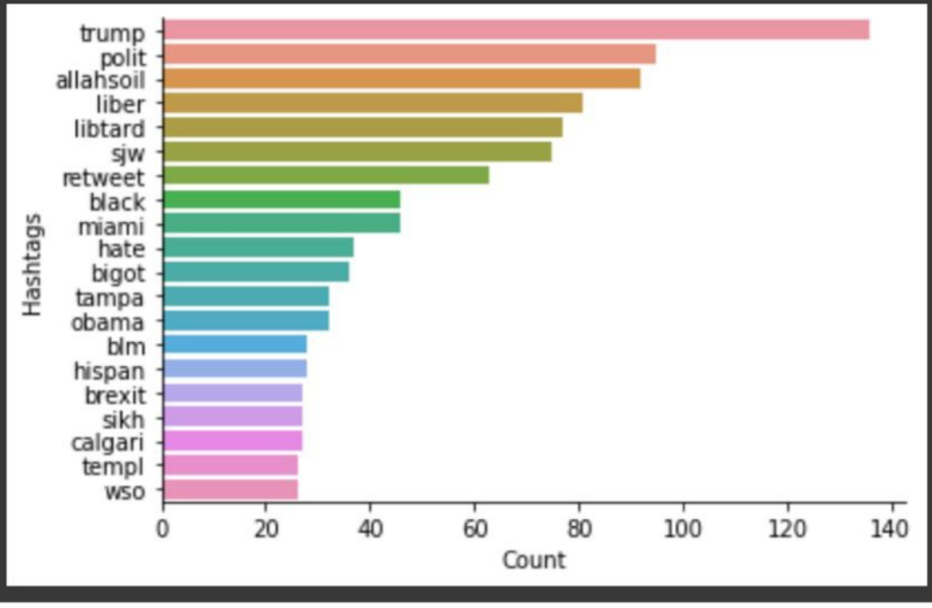
Creating a dataframe for the most frequently used words in hashtags

```
df_negative = pd.DataFrame({'Hashtags':list(word_freq_negative.keys()),'Count':list(word_freq_negative.values())})

df_negative.head(10)
```

	Hashtags	Count
0	cnn	10
1	michigan	2
2	tcot	14
3	australia	6
4	opkillingbay	5
5	seashepherd	22
6	helpcovedolphin	3
7	thecov	4
8	neverump	8
9	xenophobia	12

Plotting the barplot for the 20 most frequent words used for hashtags

```
df_negative_plot = df_negative.nlargest(20,columns='Count')

sns.barplot(data=df_negative_plot,y='Hashtags',x='Count')
sns.despine()
```



Conclusion

Upon evaluating all the models we can conclude the following details i.e.

Accuracy: As far as the accuracy of the model is concerned Logistic Regression performs better than SVM which in turn performs better than Bernoulli Naive Bayes.

F1-score: The F1 Scores for class 0 and class 1 are :
(a) For class 0: Bernoulli Naive Bayes(accuracy = 0.90) < SVM (accuracy =0.91) < Logistic Regression (accuracy = 0.92)
(b) For class 1: Bernoulli Naive Bayes (accuracy = 0.66) < SVM (accuracy = 0.68) < Logistic Regression (accuracy = 0.69)

AUC Score: All three models have the same ROC-AUC score.

We, therefore, conclude that the Logistic Regression is the best model for the above-given

dataset.

In our problem statement, Logistic Regression is following the principle of Occam's Razor which defines that for a particular problem statement if the data has no assumption, then the simplest model works the best. Since our dataset does not have any assumptions and Logistic Regression is a simple model, therefore the concept holds true for the above-mentioned dataset.

Reference

- NPTEL
- Google
- Github
- Class Notes


Thank
You