MINI PROJECT

(2021-22)

"IMDb And Twitter Sentiment Analysis"

Project Report

Institute of Engineering & Technology

Submitted By -

Saksham Sabhani (191500697)

Sanskar Gupta (191500716)

Tushar Sharma (191500864)

Utkarsh Parihar(191500883)

Under the Supervision Of

Md. Farmanul Haque

Technical Trainer

Department of Computer Engineering & Applications

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Declaration

We hereby declare that the work which is being presented in the Bachelor of technology. Project **"IMDb And Twitter Sentiment Analysis"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Md. Farmanul Haque, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *SakshamSabhani*

Name of Candidate: Saksham Sabhani

University Roll No.:191500697

Sign: *TusharSharma*

Name of Candidate: Tushar Sharma

University Roll No.:191500864

Sign: *SanskarGupta*

Name of Candidate: Sanskar Gupta

University Roll No.:191500716

Sign: *UtkarshParihar*

Name of Candidate: Utkarsh Parihar

University Roll No.:191500883

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Certificate

This is to certify that the project entitled "IMDb And Twitter Sentiment Analysis", carried out in Mini Project – I Lab, is a bonafide work by Saksham Sabhani, Sanskar Gupta, Tushar Sharma, and Utkarsh Parihar and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Md. Farmanul Haque

Date:

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place our deep gratitude to GLA University for providing us the instructor Md. Farmanul Haque, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also our colleagues who helped us find resources during the training.

Thanking You

Sign: *SakshamSabhani*

Name of Candidate: Saksham Sabhani

University Roll No.:191500697

Sign: *TusharSharma*

Name of Candidate: Tushar Sharma

University Roll No.:191500864

Sign: *SanskarGupta*

Name of Candidate: Sanskar Gupta

University Roll No.:191500716

Sign: *UtkarshParihar*

Name of Candidate: Utkarsh Parihar

University Roll No.:191500883

ABSTRACT

Sentiment Analysis or Opinion Mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviours. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms.

TABLE OF CONTENTS

Cover Page …… i

Declaration ……ii

Certificate ……..iii

Acknowledgement ……..……...iv

Abstract .....……...….v

Table of Contents ...……...……..vi

1. Introduction

- 1.1 Motivation.....……...1
- 1.2 Overview...…….....2
- 1.3 Objective……...…….3

2. Software Requirement Analysis

- 2.1 Problem Statement.....…….…….....4
- 2.2 Modules and their functionalities in SRS format..……..5

3. Software Design

- 3.1 Data Flow Diagrams…….….7
- 3.2 UML Diagrams.....…….……..8

4. Testing

- 4.1 Black Box Testing……...……...….9
- 4.2 White Box Testing……...……...….9

5. Implementation and User Interface

- 5.1 Descriptions.....……...……10
- 5.2 Algorithms Implementation...……..15
- 5.3 Comparisions and Outputs...……....18

6. Challenges and Future steps…….....20

7. Conclusion….....…….……22

8. References….....…….……23

9. Appendices......…….……24

MOTIVATION

- In the past decade, new forms of communication, such as microblogging and text messaging have emerged and become ubiquitous. While there is no limit to the range of information conveyed by tweets and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them.
- Tweets and texts are short: a sentence or a headline rather than a document. The language used is very informal, with creative spelling and punctuation, misspellings, slang, new words, URLs, and genre-specific terminology and abbreviations, such as, RT for "re-tweet" and # hashtags, which are a type of tagging for twitter messages.
- Another aspect of social media data such as Twitter messages is that it includes rich structured information about the individuals involved in the communication. For example, Twitter maintains information of who follows whom and re-tweets and tags inside of tweets provide discourse information.

OVERVIEW

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. Social media monitoring tools like Bandwidth Analytics make that process quicker and easier than ever before, thanks to real-time monitoring capabilities.

The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organizations across the world.

Voice of customer (VoC), we combine and evaluate all of your customer feedback: from the web, customer surveys, chats, call centers, and emails. Sentiment analysis allows you to categorize and structure this data to identify patterns and discover recurring topics and concerns.

Listening to the voice of your customers, and learning how to communicate with them – what works and what doesn't – will help you create a personalized customer experience.

OBJECTIVE

The objective of this project is to analyze IMDb reviews and turned it into overall polarity which tells us that the overall sentiment of the review of a movie is positive, negative or neutral. Which can be used for multiple purpose like other people can watch a particular movie or not, where all the comments are processed and turned into overall result? In Decision making, the opinions of others have a significant effect on customers ease, making choices with regards to online shopping, choosing events, products, and entities.

PROBLEM STATEMENT

A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry", "sad", and "happy".

SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a structured collection of information that that embodies the requirements of the system. Basically it is a comprehensive description of the intended purpose and environment for the proposed work under development. A software requirements specification (SRS) is an illustration of the intentional motivation and environment for the software under development. How software will work and what software is going to do is described by the SRS. The intention of SRS is to minimize the time required to achieve goals by the developers and also it reduces the cost of development. SRS will show or define that how a particular application will be interacting with the hardware of the system, human users and other programs. Through SRS operating speed, availability, portability, maintainability, and speed of recovery is evaluated.

Product Functions:

This model is basically used for Product Rating system that detects hidden sentiments in comments and rates the product accordingly. The system uses sentiment analysis methodology in order to achieve desired functionality.

This project is an E-Commerce web application where the registered user will view the product and product features and will comment about the product. System will analyze the comments of various users and will rank product.

We use a database of sentiment based keywords along with positivity or negativity weight in database and then based on these sentiment keywords mined in user comment is ranked. Comment will be analyzed by comparing the comment with the keywords stored in database. The System takes comments of various users, based on the comment, system will specify whether the product is good, bad, or worst.

Functional Requirements:

Functional Requirements are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

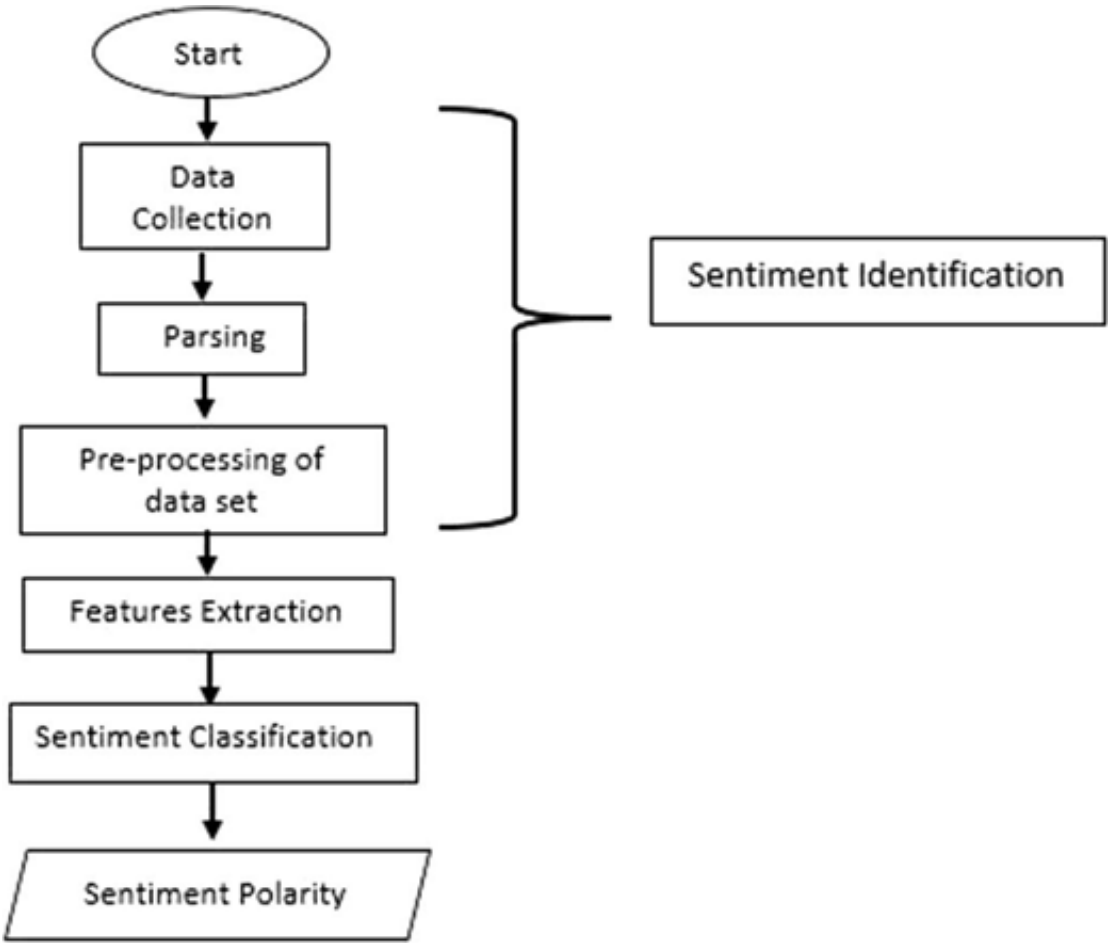
- It should be able to process new images.
- It should be able to analyze data and classify each type with decent accuracy.

Non-Functional Requirements:

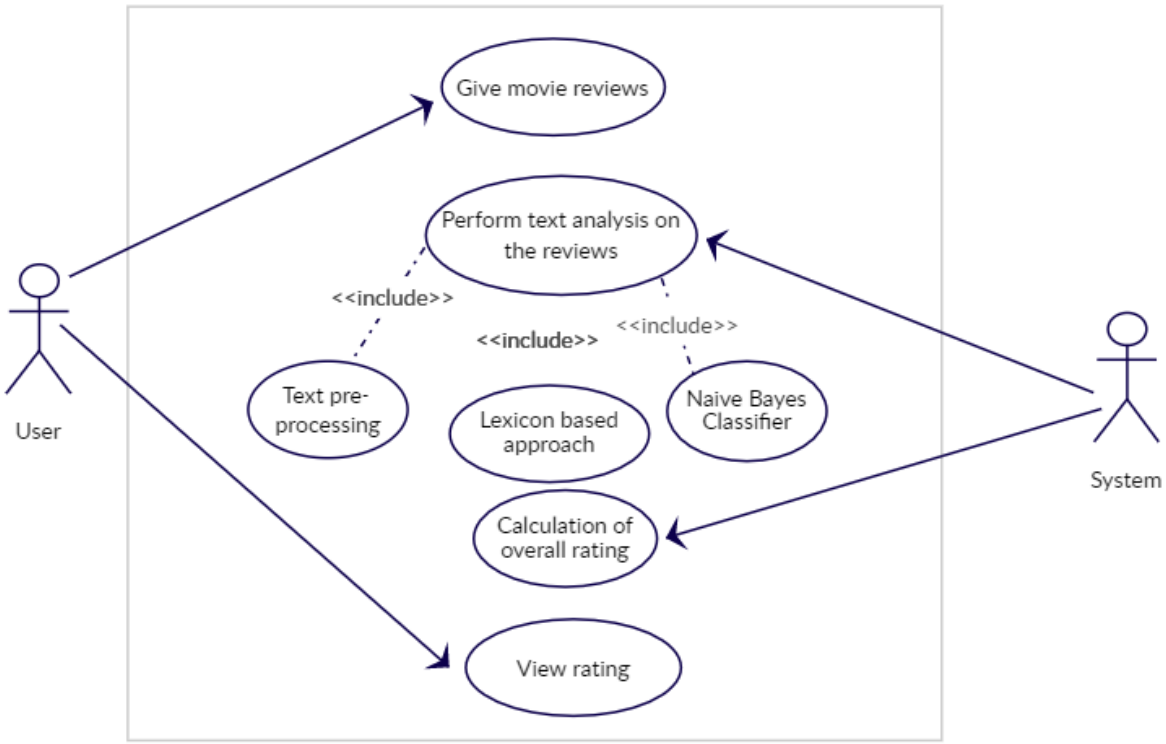
Non-Functional Requirements is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Some non-functional requirements are as follows:

- User-friendly
- System should provide better accuracy.
- To perform with efficient throughput and reponse time.

DATA FLOW DIAGRAM



USE CASE DIAGRAM



TESTING

Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results i.e. testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

Testing Methods:

- **Black Box Testing –**
  The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

- **White Box Testing –**
  White box testing is the detailed investigation of internal logic and structure of the code. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit of the code is behaving inappropriately.

DESCRIPTIONS

Data Set Description:

We have used the “Twitter” dataset with given attributes:

1. Id
2. Label
3. Tweet

Libraries used:-

Loading Libraries

In [1]:
```
import nltk # for text manipulation
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

%matplotlib inline
```

Let's read train and test datasets.

Uploading Data:-

Loading Dataset

In [14]:
```
import pandas as pd
import re # for regular expressions
pd.set_option("display.max_colwidth", 200)
import numpy as np
import seaborn as sns
import string
import matplotlib.pyplot as plt
```

In [3]:
```
train  = pd.read_csv('train_tweets.csv')
test = pd.read_csv('test_tweets.csv')
```

Getting Insights from Data:

Data Inspection

Let's check out a few non racist/sexist tweets.

In [4]:
```
train[train['label'] == 0].head(10)
```

Out[4]:

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

Now check out a few racist/sexist tweets.

In [5]:
```
train[train['label'] == 1].head(10)
```

Out[5]:

	id	label	tweet
13	14	1	@user #cnn calls #michigan middle school 'buil...
14	15	1	no comment! in #australia #opkillingbay #se...
17	18	1	retweet if you agree!
23	24	1	@user @user lumpy says i am a . prove it lumpy.
34	35	1	it's unbelievable that in the 21st century we'...
56	57	1	@user lets fight against #love #peace
68	69	1	ð□□©the white establishment can't have blk fol...
77	78	1	@user hey, white people: you can call people '...
82	83	1	how the #altright uses & insecurity to lu...
111	112	1	@user i'm not interested in a #linguistics tha...

Dimension of Training and Test Dataset:

Let's check dimensions of the train and test dataset.

In [6]:
```
train.shape, test.shape
```

Out[6]:
```
((31962, 3), (17197, 2))
```

Train set has 31,962 tweets and test set has 17,197 tweets.

Let's have a glimpse at label-distribution in the train dataset.

In [7]:
```
train["label"].value_counts()
```

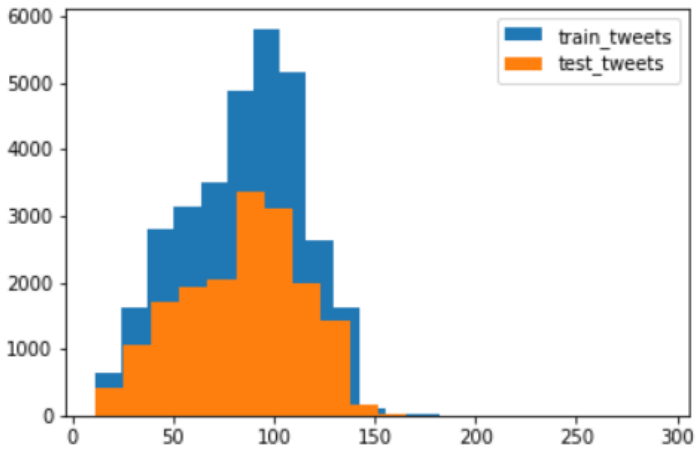
Out[7]:
```
0    29720
1     2242
Name: label, dtype: int64
```

In the train dataset, we have 2,242 (~7%) tweets labeled as racist or sexist, and 29,720 (~93%) tweets labeled as non racist/sexist. So, it is an imbalanced classification challenge.

Data Distribution:

Now we will check the distribution of length of the tweets, in terms of words, in both train and test data.

In [10]:
```
length_train = train['tweet'].str.len()
length_test = test['tweet'].str.len()

plt.hist(length_train, bins=20, label="train_tweets")
plt.hist(length_test, bins=20, label="test_tweets")
plt.legend()
plt.show()
```



The tweet-length distribution is more or less the same in both train and test data.

Data Pre-Processing:

```
In [12]: def remove_pattern(input_txt, pattern):
             r = re.findall(pattern, input_txt)
             for i in r:
                 input_txt = re.sub(i, '', input_txt)

             return input_txt
```

A.

1. Removing Twitter Handles (@user)

```
In [15]: combi['tidy_tweet'] = np.vectorize(remove_pattern)(combi['tweet'], "@[\w]*")
         combi.head()
```

Out[15]:

	id	label	tweet	tidy_tweet
0	1	0.0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run	when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0.0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked	thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
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 urð□□±!!! ð□□□ð□□□ð□□□ð□□□ ð□□¦ð□□¦ð□□¦	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ ð□□¦ð□□¦ð□□¦
4	5	0.0	factsguide: society now #motivation	factsguide: society now #motivation

B.

2. Removing Punctuations, Numbers, and Special Characters

```
In [16]: combi['tidy_tweet'] = combi['tidy_tweet'].str.replace("[^a-zA-Z#]", " ")
         combi.head(10)
```

Out[16]:

	id	label	tweet	tidy_tweet
0	1	0.0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run	when a father is dysfunctional and is so selfish he drags his kids into his dysfunction #run
1	2	0.0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked	thanks for #lyft credit i can t use cause they don t offer wheelchair vans in pdx #disapointed #getthanked
2	3	0.0	bihday your majesty	bihday your majesty

C.

3. Removing Short Words

In [17]:
```
combi['tidy_tweet'] = combi['tidy_tweet'].apply(lambda x: ' '.join([w fo
r w in x.split() if len(w)>3]))
```

Let's take another look at the first few rows of the combined dataframe.

In [18]:
```
combi.head()
```

Out[18]:

	id	label	tweet	tidy_tweet
0	1	0.0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run	when father dysfunctional selfish drags kids into dysfunction #run
1	2	0.0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked	thanks #lyft credit cause they offer wheelchair vans #disapointed #getthanked
2	3	0.0	bihday your majesty	bihday your majesty

D. Text Normalization

Here we will use nltk's PorterStemmer() function to normalize the tweets. But before that we will have to tokenize the tweets. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens.

In [19]:
```
tokenized_tweet = combi['tidy_tweet'].apply(lambda x: x.split()) # token
izing
```

In [20]:
```
tokenized_tweet.head()
```

Out[20]:
```
0               [when, father, dysfunctional, selfish, drags, kids, int
o, dysfunction, #run]
1    [thanks, #lyft, credit, cause, they, offer, wheelchair, vans, #disap
ointed, #getthanked]
2                                                                  [bi
hday, your, majesty]
3                                                    [#model, lov
e, take, with, time]
4                                                  [factsguide, s
ociety, #motivation]
Name: tidy_tweet, dtype: object
```

Now we can normalize the tokenized tweets.

Story Generation and Visualization from Tweets

A) Understanding the common words used in the tweets: WordCloud

Now We want to see how well the given sentiments are distributed across the train dataset. One way to accomplish this task is by understanding the common words by plotting wordclouds.

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

Let's visualize all the words our data using the wordcloud plot.

In [42]:
```
all_words = ' '.join([text for text in combi['tidy_tweet']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_s
ize=110).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

We can see most of the words are positive or neutral. Words like love, great, friend, life are the most frequent ones. It doesn't give us any idea about the words associated with the racist/sexist tweets. Hence, we will plot separate wordclouds for both the classes (racist/sexist or not) in our train data.

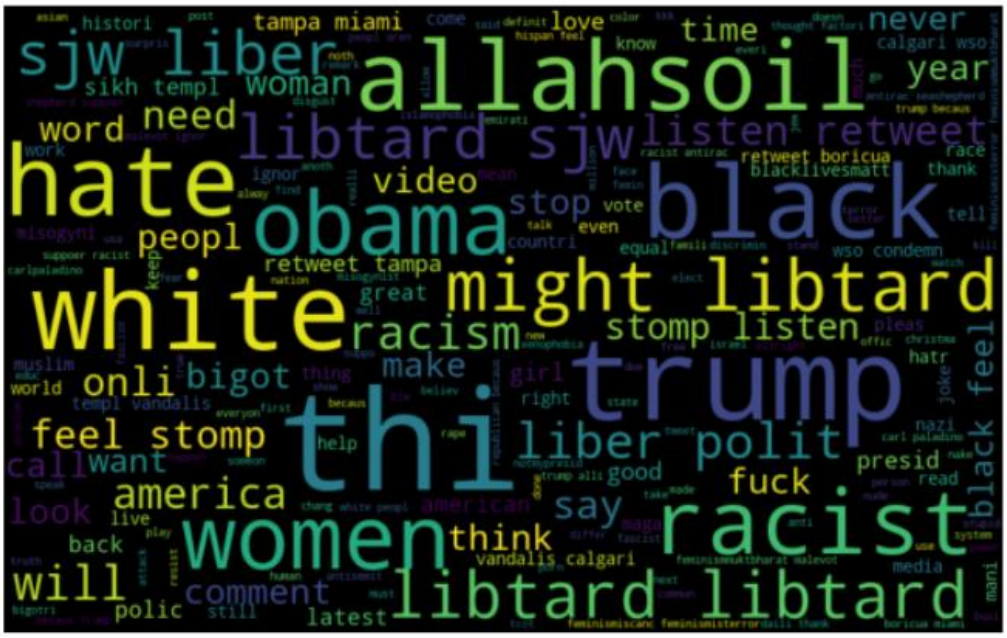
B) Words in non racist/sexist tweets

```
In [43]: normal_words =' '.join([text for text in combi['tidy_tweet'][combi['label'] == 0]])

         wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(normal_words)
         plt.figure(figsize=(10, 7))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis('off')
         plt.show()
```

Most of the frequent words are compatible with the sentiment, i.e., non-racist/sexists tweets. Similarly, we will plot the word cloud for the other sentiment. Expect to see negative, racist, and sexist terms.

C) Racist/Sexist Tweets

```
In [44]: negative_words = ' '.join([text for text in combi['tidy_tweet'][combi['l
         abel'] == 1]])
         wordcloud = WordCloud(width=800, height=500,
         random_state=21, max_font_size=110).generate(negative_words)
         plt.figure(figsize=(10, 7))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis('off')
         plt.show()
```



As we can clearly see, most of the words have negative connotations. So, it seems we have a pretty good text data to work on. Next we will the hashtags/trends in our twitter data.

D) Understanding the impact of Hashtags on tweets sentiment.

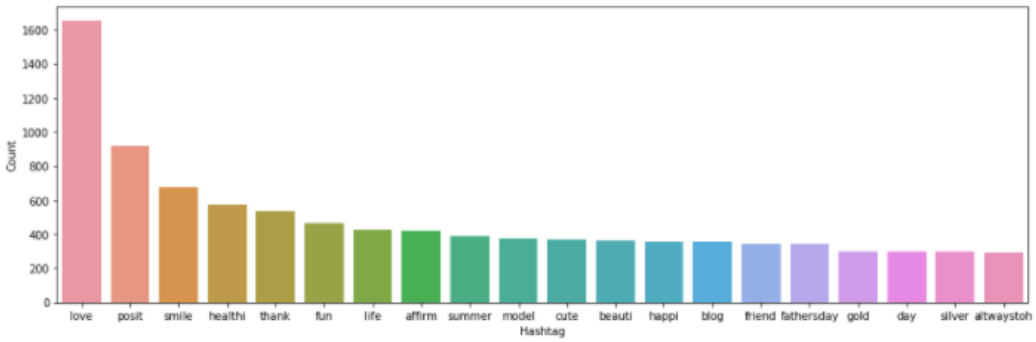
In [46]:
```
# function to collect hashtags
def hashtag_extract(x):
    hashtags = []
    # Loop over the words in the tweet
    for i in x:
        ht = re.findall(r"#(\w+)", i)
        hashtags.append(ht)

    return hashtags
```

In [47]:
```
# extracting hashtags from non racist/sexist tweets

HT_regular = hashtag_extract(combi['tidy_tweet'][combi['label'] == 0])

# extracting hashtags from racist/sexist tweets
HT_negative = hashtag_extract(combi['tidy_tweet'][combi['label'] == 1])

# unnesting list
HT_regular = sum(HT_regular,[])
HT_negative = sum(HT_negative,[])
```

Now that we have prepared our lists of hashtags for both the sentiments, we can plot the top 'n' hashtags. So, first let's check the hashtags in the non-racist/sexist tweets.

- **Non-Racist/Sexist Tweets**

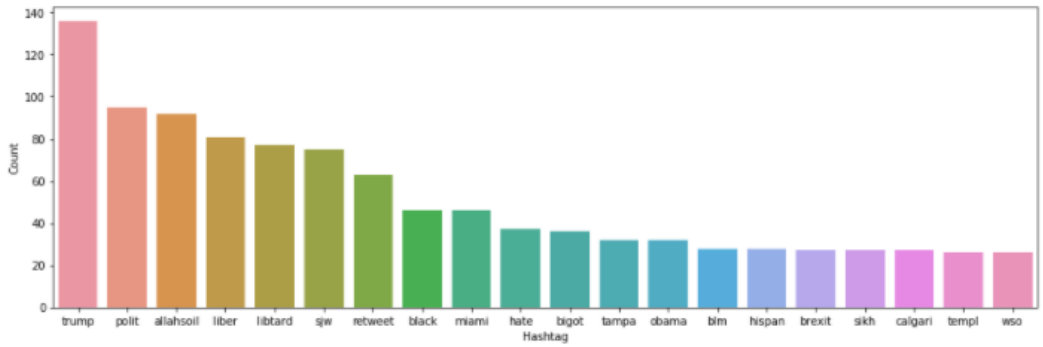
In [49]:
```
a = nltk.FreqDist(HT_regular)
d = pd.DataFrame({'Hashtag': list(a.keys()),
                  'Count': list(a.values())})

# selecting top 20 most frequent hashtags
d = d.nlargest(columns="Count", n = 20)
plt.figure(figsize=(16,5))
ax = sns.barplot(data=d, x= "Hashtag", y = "Count")
ax.set(ylabel = 'Count')
plt.show()
```



All these hashtags are positive and it makes sense. We am expecting negative terms in the plot of the second list. Let's check the most frequent hashtags appearing in the racist/sexist tweets.

- **Racist/Sexist Tweets**

```
In [51]: b = nltk.FreqDist(HT_negative)
         e = pd.DataFrame({'Hashtag': list(b.keys()), 'Count': list(b.values())})

         # selecting top 20 most frequent hashtags
         e = e.nlargest(columns="Count", n = 20)
         plt.figure(figsize=(16,5))
         ax = sns.barplot(data=e, x= "Hashtag", y = "Count")
```



As expected, most of the terms are negative with a few neutral terms as well. So, it's not a bad idea to keep these hashtags in our data as they contain useful information. Next, we will try to extract features from the tokenized tweets.

CHALLENGES AND FUTURE STEPS

Today, text data is everywhere. As humans, we can easily understand this information, but for computers it's a complicated task. The science of understanding and learning from text data is called natural language processing (NLP). Programmers encounter many common challenges when trying to teach computers to understand natural language text data.

1) Unstructured data and Big Data: Most common challenges we face in NLP are around unstructured data and Big Data. Data generated from online conversations, comments, tweets, etc. is "big" and highly unstructured. It's a huge challenge to process that data and get useful information out of it.

2) Semantic meaning of words: Another common challenge is the semantic meaning of words. The vocabulary of any given language is very vast, and many words have similar meanings. So, machines need to find those words. While training a model for NLP, words not present in the training data commonly appear in the test data. Because of this, predictions made using test data may not be correct. To solve this problem, machines need to capture the semantic meaning of words. Using the semantic meaning of words it already knows as a base, the model can understand the meanings of words it doesn't know that appear in test data.

3) Information extraction: Another major challenge is extracting useful information from data. With the increase in data availability, extracting important information is quite challenging, like searching for a needle in a haystack.

4) Real-time data: Datasets are expanding at breakneck speed; new data is being generated every second, and old information is updated in real time. It's difficult to retrain models frequently from scratch for new data.

Building a better future with NLP:-

Data is the new oil. It creates new prospects and challenges every day. Established and emerging companies alike are putting their efforts into creating platforms and apps that understand natural language the way humans do. In the future, we'll simply talk to all of our devices to get them to do what we want, and techniques like these are part of the foundation of that future.

CONCLUSION

In this, We describe our approach to analyze all tweets and turned it into conclusion that either the overall tweet is positive, negative or neutral. This is done using textblob, nltk library.

REFERENCES

- https://towardsdatascience.com/
- https://kaggle.com/
- https://github.com/
- Alec Go, Richa Bhayani, Lei Huang. Twitter Sentiment Classification using Distant Supervision.
- Fuchun Peng. 2003, Augmenting Naive Bayes Classifiers with Statistical Language Models

APPENDICES

Dataset used:

https://www.kaggle.com/lakshmi25npathi/IMDb-dataset-of-50k-movie-reviews/download

Code Repository:

https://github.com/sakshamsabhani/SentimentAnalysisProjects