SENTIMENT ANALYSIS ON SCRAPED TWEETS

A Project report submitted in partial fulfillment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

B.Chandrika(317126511065) P.Sai Kiran(317126511101)

L.Rajini Piya(317126511090) K.Anu Srihitha(317126611083)

Under the guidance of

Dr. P.Padmaja

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with ‘A’ Grade*)

Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

2019-2020

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr.P.Padmaja.** Professor & HOD, Department of Information Technology, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr.P.Padmaja**, Head of the Department, Information Technology, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of IT, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of IT, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

B.Chandrika(317126511065),
P.Sai Kiran(317126511101),
L.Rajini Priya(317126511090),
K.Anu Srihitha(317126511083),

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

CERTIFICATE

This is to certify that the project reported entitled **"Sentiment Analysis on Scraped Tweets"** submitted by **B.Chandrika(317126511065), P. Sai Kiran(317126511101), L.Rajini Priya (317126511090), K.Anu Srihitha(317126511087)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** of Anil Neerukonda Institute of technology and sciences, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide	Head of the Department
Dr.P.Padmaja	**Dr.P.Padmaja**
Department of IT	Department of IT
ANITS	ANITS

CONTENTS

ABSTRACT

Social networks are the main resources to gather information about people's opinion and sentiments towards different topics as they spend hours daily on social medias and share their opinion. This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users - out of which 100 million are active users and half of them log on twitter on a daily basis - generating nearly 250 million tweets per day.

Due to this large amount of usage we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets. Analysing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socio economic phenomena like stock exchange. In this project, we show the application of sentimental analysis and how to connect to Twitter and run sentimental analysis queries. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

CHAPTER 1 – INTRODUCTION

In the past few years, there has been a huge growth in the use of microblogging platforms such as Twitter. Spurred by that growth, companies and media organizations are increasingly seeking ways to mine Twitter for information about what people think and feel about their products and services. Companies such as Twitratr (twitrratr.com), tweetfeel (www.tweetfeel.com), and Social Mention (www.socialmention.com) are just a few who advertise Twitter sentiment analysis as one of their services.

The online medium has become a significant way for people to express their opinions and with social media, there is an abundance of opinion information available. Using sentiment analysis, the polarity of opinions can be found, such as positive, negative, or neutral by analyzing the text of the opinion. Sentiment analysis has been useful for companies to get their customer's opinions on their products predicting outcomes of elections , and getting opinions from movie reviews. The information gained from sentiment analysis is useful for companies making future decisions.

Many traditional approaches in sentiment analysis uses the bag of words method. The bag of words technique does not consider language morphology, and it could incorrectly classify two phrases of having the same meaning because it could have the same bag of words . The relationship between the collection of words is considered instead of the relationship between individual words . When determining the overall sentiment, the sentiment of each word is determined and combined using a function . Bag of words also ignores word order, which leads to phrases with negation in them to be incorrectly classified. Other techniques discussed in sentiment analysis include Naive Bayes, Maximum Entropy, and Support Vector Machines.

Sentiment analysis refers to the broad area of natural language processing which deals with the computational study of opinions, sentiments and emotions expressed in text. Sentiment Analysis (SA) or Opinion Mining (OM) aims at learning people's opinions, attitudes and emotions towards an entity. The entity can represent individuals, events or topics. An immense amount of research has been performed in the area of sentiment analysis. But most of them focused on classifying formal and larger pieces of text data like reviews.

Project Objective:

The objective is to find the opinionative data and classify it according to its polarity, i.e. positive, negative or neutral feedback, known as sentiment classification and then analysing it which is known as sentiment analysis. However, before performing sentiment examination, the information is exposed to different pre-processing procedures which finally give the desired optimized output. This allows us to get to know about the public's mood or opinion about a particular topic.

Project Outline:

Sentiment analysis refers to the class of natural language processing based techniques used to identify, extract or characterize subjective information, such as opinions, expressed in a given piece of text. The main purpose of sentiment analysis is to classify a people attitude towards various topics into positive, negative or neutral categories. Sentiment analysis has many applications in different domains including, but not limited to, business intelligence, politics, sociology, etc.

CHAPTER 2 – DOMAIN INTRODUCTION

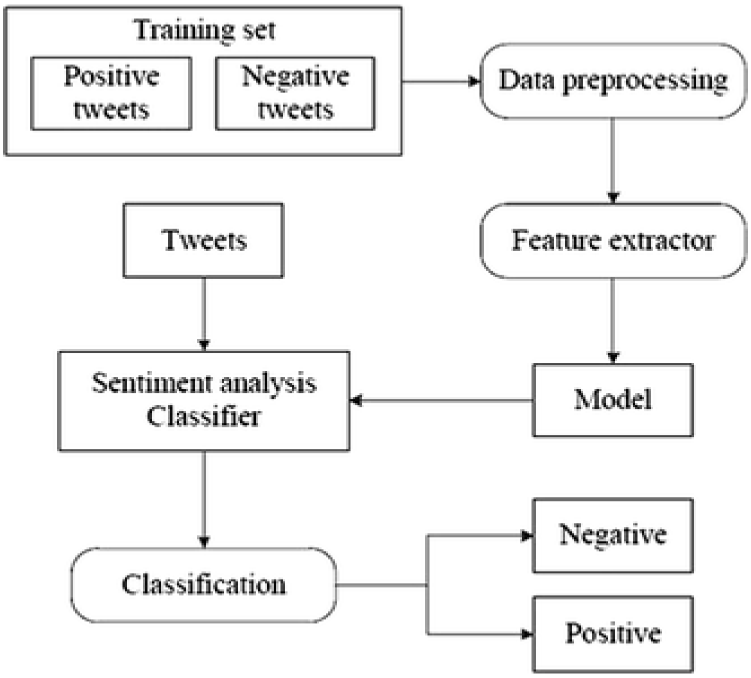
This project of analyzing sentiments of tweets comes under the domain of "Pattern Classification" and "Data Mining". Both of these terms are very closely related and intertwined, and they can be formally defined as the process of discovering "useful" patterns in large set of data, either automatically (unsupervised) or semiautomatically (supervised). The project would heavily rely on techniques of "Natural Language Processing" in extracting significant patterns and features from the large data set of tweets and on "Machine Learning" techniques for accurately classifying individual unlabelled data samples (tweets) according to whichever pattern model best describes them.

The features that can be used for modeling patterns and classification can be divided into two main groups: formal language based and informal blogging based. Language based features are those that deal with formal linguistics and include prior sentiment polarity of individual words and phrases, and parts of speech tagging of the sentence. Prior sentiment polarity means that some words and phrases have a natural innate tendency for expressing particular and specific sentiments in general. For example the word "excellent" has a strong positive connotation while the word "evil" possesses a strong negative connotation. So whenever a word with positive connotation is used in a sentence, chances are that the entire sentence would be expressing a positive sentiment. Parts of Speech tagging, on the other hand, is a syntactical approach to the problem. It means to automatically identify which part of speech each individual word of a sentence belongs to: noun, pronoun, adverb, adjective, verb, interjection, etc. Patterns can be extracted from analyzing the frequency distribution of these parts of speech (ether individually or collectively with some other part of speech) in a particular class of labeled tweets. Twitter based features are more informal and relate with how people express themselves on online social platforms and compress their sentiments in the limited space of 140 characters offered by twitter. They include twitter hashtags, retweets, word capitalization, word lengthening [13], question marks, presence of url in tweets, exclamation marks, internet emoticons and internet shorthand/slangs.

Classification techniques can also be divided into a two categories: Supervised vs. unsupervised and non-adaptive vs. adaptive/reinforcement techniques. Supervised approach is when we have pre-labeled data samples available and we use them to train our classifier. Training the classifier means to use the pre-labeled to extract features that best model the patterns and differences between each of the individual classes, and then classifying an unlabeled data sample according to whichever pattern best describes it.

Unsupervised classification is when we do not have any labeled data for training. In addition to this adaptive classification techniques deal with feedback from the environment. In our case feedback from the environment can be in form of a human telling the classifier whether it has done a good or poor job in classifying a particular tweet and the classifier needs to learn from this feedback. There are two further types of adaptive techniques: Passive and active. Passive techniques are the ones which use the feedback only to learn about the environment (in this case this could mean improving our models for tweets belonging to each of the three classes) but not using this improved learning in our current classification algorithm, while the active approach continuously keeps changing its classification algorithm according to what it learns at real-time.

CHAPTER-3: ARCHITECTURAL FLOW



In this method we uses textblob as a method to find the polarity of the text (positive text, negative text). The tweets are imported from the Twitter using the (API) provided by the Twitter Developer. From these API various fields like tweets, source, re-tweets, likes, language, user etc. can be scrapped. After collecting these data, we can analyses the various thoughts on an event or occasion.

The Above figure explains the extraction of tweets id from twitter through API, then preprocess the data that are extracted. Preprocessing includes exclusion of unwanted fields, segregating the fields important for analysis. Once the fields are extracted, a model will be built which classify the tweets whether positive or negative .

CHAPTER 4 – PROJECT MODULES

Sentiment analysis on scraped tweets project consists of scraping tweets, identifying sentiments and model building. We divided the complete project into five modules . They are as follows:

Module 1: Scraping Tweets

Module 2: Identifying sentiments

Module 3: Text Pre-processing

Module 4: Feature Extraction

Module 5: Model Building

MODULE 1: SCRAPING TWEETS

A process of getting small fragments of something. In our case, it is web scraping, so here we are taking fragments of information available on a website. We can use Selenium, tweepy python libraries to scrap tweets from the twitter, because of dynamic and progressive generation of tweets. Selenium is the browser mocking tool usually used for testing web pages and tweepy as I mentioned, a python library which provides access for various twitter APIs. But in our we used tweepy because it is fast and more reliable.

Twitter is a great tool to gather tons of quality data. Twitter makes it really easy to gather publicly available data using its APIs.

The following are the steps to create API:

1. Sign up for our Twitter.

2. Create New Application :

Enter our Application Name, Description and our website address. Copy the consumer key (API key) and consumer secret from the screen into our application.

3. Create an Application.

4. After creating application you can copy your secret key.

Now, we will using Tweepy library to extract data from twitter using the above keys.

MODULE 2: IDENTIFYING SENTIMENTS

Sentiment analysis is the process of analyzing views or opinions of people on any subject . Sentiment type is nothing but the overall reaction, it can be positive, negative or neutral. In our case, we are only going to consider positive (includes neutral) and negative. we will be training a model which should be capable of classifying negative and positive sentiments on tweets. For this classification, we will be using some supervised learning model, so we need to have a target variable. Sentiment type is going to be our target variable.We use textblob for identifying sentiments because it gives better categorization.

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral.

The sentiment function of textblob returns two properties, **polarity**, and **subjectivity**.

Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

Example: Checking the sentiment of some text.

>print(blob)

>blob.sentiment

>>Sara is the good platform to learn data science.

Sentiment(polarity=0.8, subjectivity=0.75)

We can see that polarity is **0.8**, which means that the statement is positive and **0.75** subjectivity refers that mostly it is a public opinion and not a factual information.

MODULE 3: TEXT PRE-PROCESSING

A tweet contains a lot of opinions about the data which are expressed in different ways by different users. Raw tweets scraped from twitter generally result in a noisy and obscure data. This is due to the casual and ingenious nature of people's usage of social media. Tweets have certain special characteristics such as retweets, emoticons, user mentions, etc. which should be suitably extracted.

The twitter data used in this survey work is already labeled into two classes viz. negative and positive polarity and thus the sentiment analysis of the data becomes easy to observe the effect of various features. The raw data having polarity is highly susceptible to inconsistency and redundancy. Text obtained from tweets is not clean enough to be used for model training So it needs to be pre-processed first. We have applied an extensive number of pre-processing steps to standardize the data.

We first do some general preprocessing on tweets which is as follows:

a. **Removing '@names':**
   All the '@anyname' are of no use since they don't convey any meaning.
b. **Removing links (http | https):**
   Links in the text are of no use because they don't convey any useful information as well.
c. **Dropping duplicate row:**
   We may have duplicate tweets in our data-frame, so we need to remove them.
d. **Removing Punctuations, Numbers and Special characters:**
   We also want to do sentiment analysis on __key phrases__ as well, because semantic meaning in a sentence needs to be present. So here we will create one additional column 'absolute_tidy_tweets' which will contain absolute tidy words which can be further used for sentiment analysis on __key words__.
e. **Removing Stop Words:**
   Stop words are the words in that are used just for the sake of correct sentence formations. They don't have any meaning full information. So it needs to be removed to make our text record cleaner.

MODULE 4: FEATURE EXTRACTION

We need to convert textual representation in the form of numeric features. Machine learning techniques require representing the key features of text or documents for processing. These key features are considered as feature vectors which are used for the classification task.

Some examples features that have been reported are:

1.Words And Their Frequencies:

Unigrams, bigrams and n-gram models with their frequency counts are considered as features. There has been more research on using word presence rather than frequencies to better describe this feature.

2. Parts Of Speech Tags:

Parts of speech like adjectives, adverbs and some groups of verbs and nouns are good indicators of subjectivity and sentiment. We can generate syntactic dependency patterns by parsing or dependency trees.

3. Opinion Words And Phrases:

Apart from specific words, some phrases and idioms which convey sentiments can be used as features. e.g. cost someone an arm and leg.

4. Position Of Terms:

The position of a term with in a text can affect on how much the term makes difference in overall sentiment of the text.

5. Negation:

Negation is an important but difficult feature to interpret. The presence of a negation usually changes the polarity of the opinion.

6. Syntax:

Syntactic patterns like collocations are used as features to learn subjectivity patterns by many of the researchers.

MODULE 5: MODEL BUILDING

Training Supervised learning is an important technique for solving classification problems. Training the classifier makes it easier for future predictions for unknown data.

Model building involves five steps:

1. Defining the model
2. Compiling the model
3. Fitting the model
4. Evaluating the model
5. Making predictions with the model

In our project we use Naïve bayes classifier to build the model.

Naive Bayes: It is a probabilistic classifier and can learn the pattern of examining a set of documents that has been categorized . It compares the contents with the list of words to classify the documents to their right category or class. To train and classify using Naïve Bayes Machine Learning technique ,we can use the Python NLTK library .

This model applies Bayes theorem with a Naive assumption of no relationship between different features. According to Bayes theorem:

Posterior = likelihood * proposition/evidence (or)

$P(A|B) = P(B|A) * P(A)/P(B)$

Naive Bayes Model works particularly well with text classification and spam filtering. **Advantages** of working with NB algorithm are:

- Requires a small amount of training data to learn the parameters
- Can be trained relatively fast compared to sophisticated models

CHAPTER 5 – EXPERIMENTAL RESULTS

5.1 SYSTEM CONFIGURATION

5.1.1 Software Configuration

These are the Software Configurations that are required.

- Operating System: Windows 10/8/7 (incl. 64-bit), Mac OS, Linux
- Language: Python 3
- IDE: Jupyter Notebook

5.1.2 Hardware Configuration

These are minimum Hardware configurations that are required.

- Processor: Intel core 2 duo or higher.
- RAM: 1 GB or higher
- HDD: 256 GB or higher
- Monitor: 1024 x 768 minimum screen resolution.
- Keyboard: US en Standard Keyboard.

5.2 SAMPLE CODE:

MODULE-1:SCRAPED TWEETS SAMLE CODE:

```
import numpy as np
import pandas as pd
import tweepy
from tweepy import OAuthHandler
#Keys and tokens of twitter console
consumer_key = 'Sec3MvclRIx2RVlgu9l0SJX6D'
consumer_secret='ayoPNWtBm7fWpMBoK6EwRmegu3SW8Rw9mzJkottkv97
quPe941'
access_token = '736550752760406018-so5CPJrEbJKb3c3Pq8va3VFr0yk4S0E'
access_token_secret='Cgr8tz0h6FTU7kxAjDzpHnjffNTHxWsBytXnu4Ihd1TF
b'
#Function for fetching random tweets from twitter
#Create tweepy API object to fetch tweets
class TwitterClient(object):
    def _init_(self):
        try:
            auth = OAuthHandler(consumer_key, consumer_secret)
            auth.set_access_token(access_token, access_token_secret)

self.api=tweepy.API(auth,wait_on_rate_limit=True,wait_on_rate_limit_notify=
True)

        except tweepy.TweepError as e:
            print(f"Error: Tweeter Authentication Failed - \n{str(e)}")

    def get_tweets(self, query, maxTweets = 1000):
        tweets = []
```

```
sinceId = None
max_id = -1
tweetCount = 0
tweetsPerQry = 100

while tweetCount < maxTweets:
    try:
        if (max_id <= 0):
            if (not sinceId):
                new_tweets = self.api.search(q=query, count=tweetsPerQry)
            else:
                new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                        since_id=sinceId)
        else:
            if (not sinceId):
                new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                        max_id=str(max_id - 1))
            else:
                new_tweets = self.api.search(q=query, count=tweetsPerQry,
                                        max_id=str(max_id - 1),
                                        since_id=sinceId)
        if not new_tweets:
            print("No more tweets found")
            break

        for tweet in new_tweets:
            parsed_tweet = {}
```

```
                parsed_tweet['tweets'] = tweet.text
                if tweet.retweet_count > 0:
                    if parsed_tweet not in tweets:
                        tweets.append(parsed_tweet)
                else:
                    tweets.append(parsed_tweet)

            tweetCount += len(new_tweets)
            print("Downloaded {0} tweets".format(tweetCount))
            max_id = new_tweets[-1].id

        except tweepy.TweepError as e:
            print("Tweepy error : " + str(e))
            break

    return pd.DataFrame(tweets)
#Calling function to get tweets
twitter_client = TwitterClient()
tweets_df = twitter_client.get_tweets('AI and Deep learning', maxTweets=1000)
print(f'tweets_df Shape - {tweets_df.shape}')
tweets_df.head(10)
```

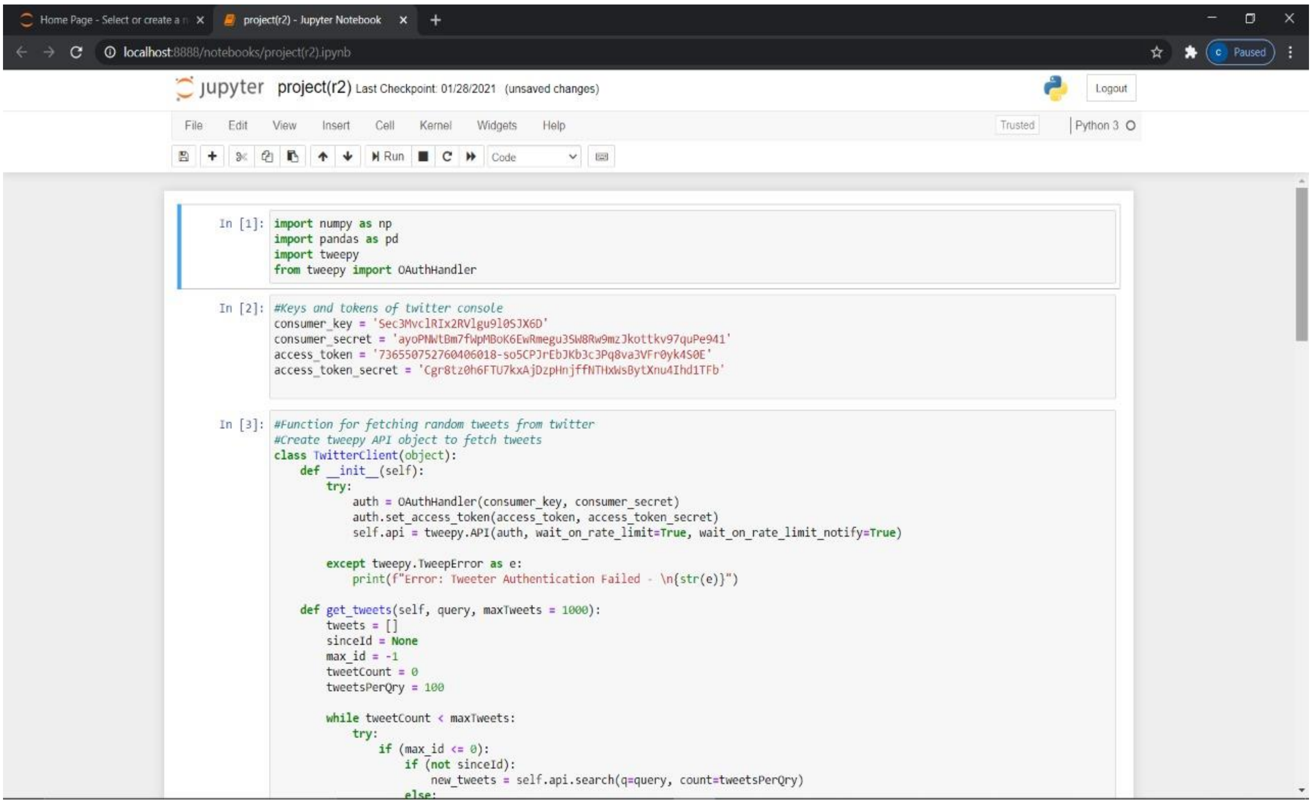
MODULE-1:IDENTIFYING SENTIMENTS SAMLE CODE:

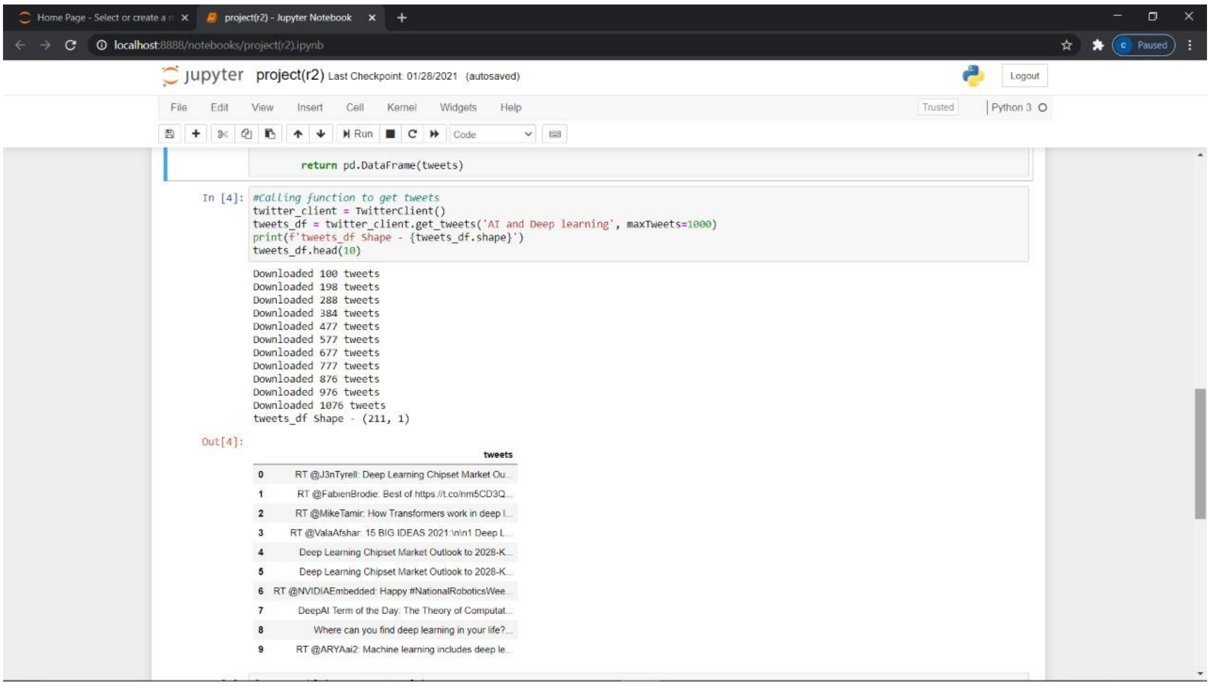
```
from textblob import TextBlob
from textblob.np_extractors import ConllExtractor
#Fetch sentiments using textblob
def fetch_sentiment_using_textblob(text):
    analysis = TextBlob(text)
```

```
    return 'pos' if analysis.sentiment.polarity >= 0 else 'neg'
#Counts The Number Of Positive And Negative Tweets
sentiments_using_textblob=tweets_df.tweets.apply(lambdatweet:
fetch_sentiment_using_textblob(tweet))
pd.DataFrame(sentiments_using_textblob.value_counts())
#Show The Polarity Of the Fetched Tweets Using Textblob
tweets_df['sentiment'] = sentiments_using_textblob
tweets_df.head()
```

5.3 SCREENSHOTS AND OUTPUTS :

These are the screenshots of project outputs.





project(r2) Last Checkpoint: 01/28/2021 (autosaved)

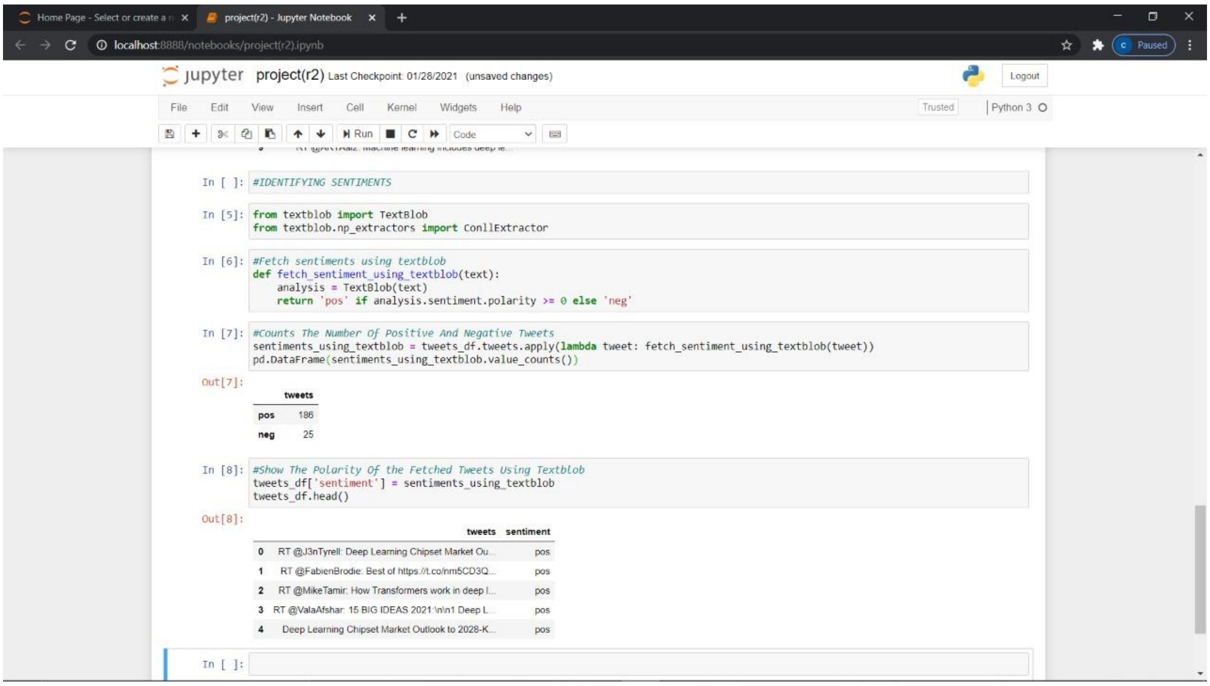
```
        return pd.DataFrame(tweets)
```

In [4]:
```
#Calling function to get tweets
twitter_client = TwitterClient()
tweets_df = twitter_client.get_tweets('AI and Deep learning', maxTweets=1000)
print(f'tweets_df Shape - {tweets_df.shape}')
tweets_df.head(10)
```

```
Downloaded 100 tweets
Downloaded 198 tweets
Downloaded 288 tweets
Downloaded 384 tweets
Downloaded 477 tweets
Downloaded 577 tweets
Downloaded 677 tweets
Downloaded 777 tweets
Downloaded 876 tweets
Downloaded 976 tweets
Downloaded 1076 tweets
tweets_df Shape - (211, 1)
```

Out[4]:

	tweets
0	RT @J3nTyrell: Deep Learning Chipset Market Ou...
1	RT @FabienBrodie: Best of https://t.co/nm5CD3Q...
2	RT @MikeTamir: How Transformers work in deep l...
3	RT @ValaAfshar: 15 BIG IDEAS 2021:\n\n1 Deep L...
4	Deep Learning Chipset Market Outlook to 2028-K...
5	Deep Learning Chipset Market Outlook to 2028-K...
6	RT @NVIDIAEmbedded: Happy #NationalRoboticsWee...
7	DeepAI Term of the Day: The Theory of Computat...
8	Where can you find deep learning in your life?...
9	RT @ARYAai2: Machine learning includes deep le...



project(r2) Last Checkpoint: 01/28/2021 (unsaved changes)

In [ ]:
```
#IDENTIFYING SENTIMENTS
```

In [5]:
```
from textblob import TextBlob
from textblob.np_extractors import ConllExtractor
```

In [6]:
```
#Fetch sentiments using textblob
def fetch_sentiment_using_textblob(text):
    analysis = TextBlob(text)
    return 'pos' if analysis.sentiment.polarity >= 0 else 'neg'
```

In [7]:
```
#Counts The Number Of Positive And Negative Tweets
sentiments_using_textblob = tweets_df.tweets.apply(lambda tweet: fetch_sentiment_using_textblob(tweet))
pd.DataFrame(sentiments_using_textblob.value_counts())
```

Out[7]:

	tweets
pos	186
neg	25

In [8]:
```
#Show The Polarity Of the Fetched Tweets Using Textblob
tweets_df['sentiment'] = sentiments_using_textblob
tweets_df.head()
```

Out[8]:

	tweets	sentiment
0	RT @J3nTyrell: Deep Learning Chipset Market Ou...	pos
1	RT @FabienBrodie: Best of https://t.co/nm5CD3Q...	pos
2	RT @MikeTamir: How Transformers work in deep l...	pos
3	RT @ValaAfshar: 15 BIG IDEAS 2021:\n\n1 Deep L...	pos
4	Deep Learning Chipset Market Outlook to 2028-K...	pos

In [ ]:

CHAPTER 6 – CONCLUSION

The experimental studies performed through the chapters, successfully show that the existing machine learning analysis and lexical analysis techniques for sentiment classification yield comparatively outperforming accurate results. The first method that we approached for our problem is Naïve Bayes. It is mainly based on the independence assumption .Training is very easy and fast In this approach each attribute in each class is considered separately. Testing is straightforward, calculating the conditional probabilities from the data available. One of the major task is to find the sentiment polarities which is very important in this approach to obtain desired output. In this Naïve Bayes approach we only considered the words that are available in our dataset and calculated their conditional probabilities. We have obtained successful results after applying this approach to our problem. Clearly from the success of Naive Bayes, it can positively be applied over other related sentiment analysis applications like financial sentiment analysis (stock market, opinion mining), customer feedback services, and etc.

CHAPTER 6 – REFERENCES

[1] Albert Biffet and Eibe Frank. Sentiment Knowledge Discovery in Twitter Streaming Data. Discovery Science, Lecture Notes in Computer Science,2010,Volume 6332/2010, 1-15, DOI: 10.1007/978-3-642-16184-1_1

[2] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[3] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[4] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2010.

[5] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002.

[6] Chenhao Tan, Lilian Lee, Jie Tang, Long Jiang, Ming Zhou and Ping Li. User Level Sentiment Analysis Incorporating Social Networks. In Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), 2011.

[7] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[8] Hatzivassiloglou, V., & McKeown, K.R.. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL, 2009.

[9] Johann Bollen, Alberto Pepe and Huina Mao. Modelling Public Mood and Emotion: Twitter Sentiment and socio-economic phenomena. In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[10] Luciano Barbosa and Junlan Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In Proceedings of the international conference on Computational Linguistics (COLING), 2010.

[11] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002. [12] Rudy Prabowo and Mike Thelwall. Sentiment Analysis: A Combined Approach. Journal of Infometrics, Volume 3, Issue 2, April 2009, Pages 143-157, 2009

[12] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In Proceedings of International Conference on Computational Linguistics (ICCL), 2004.

[13] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[14] Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT), 2005.

[15] Ian H. Witten, Eibe Frank & Mark A. Hall. Data Mining – Practical Machine Learning Tools and Techniques.