## 

**TITLE**

**Sentiment analysis using twitter dataset**

**Submitted to: Mr. Sagar Pandey**

|  |  |  |
| --- | --- | --- |
| Name | Registration Number | Roll Number |
| Km Arisha Pathak | 11810128 | A03 |
| Mann Bansal | 11804971 | A35 |

**Table of contents**

[ Introduction: 4](#_Toc36060926)

[ Objective: 4](#_Toc36060927)

[ Motivation: 4](#_Toc36060928)

[ Implementation of the project 5](#_Toc36060929)

[ Output: 7](#_Toc36060930)

[ Scope: 8](#_Toc36060931)

[ Work Distribution: 8](#_Toc36060932)

[ Libraries Used: 9](#_Toc36060933)

[ GitHub link: 9](#_Toc36060934)

[ References: 9](#_Toc36060935)

**Abstract**

Sentiment analysis or opinion mining is the computational study of people’s sentiments ,attitude and emotions expressed in written language. It is one of the most active research areas in natural language processing (NLP) 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 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.It is thus no surprise that the inception and the rapid growth of the field coincide with those of the social media on the web.In fact,the research has also scored outside of computer science to management sciences and social sciences due to its importance to business and society as a whole in this talk,discussion of the mainstream sentiment analysis .

# Introduction:

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.

# Objective:

This project aims on sentiment analysis which is classifying the polarity of a given text at the document, sentence, or featurewhether the expressed opinion in a document, a sentence or an entity feature is positive, negative, or neutral. The sentiment classification looks like, for instance, at emotional states such as "angry", "sad", and "happy" or the type of the text such as “Positive”, “Negative”.

# Motivation:

The rise of social media such as blogs and social networks has fuelled interest in sentiment analysis. With the compulsion 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. Further complicating the matter ,is the rise of anonymous social media platforms such as 4chan and Reddit.

# Implementation of the project

To run the project make sure you run the following lines of code:

* import nltk
* nltk.download('wordnet')
* nltk.download('averaged\_perceptron\_tagger')
* nltk.download('stopwords')
* nltk.download('twitter\_samples')
* nltk.download('punkt')

Following is the code:

from nltk.stem.wordnet import WordNetLemmatizer

from nltk.corpus import twitter\_samples, stopwords

from nltk.tag import pos\_tag

from nltk.tokenize import word\_tokenize

from nltk import FreqDist, classify, NaiveBayesClassifier

import re, string, random

def remove\_noise(tweet\_tokens, stop\_words = ()):

    cleaned\_tokens = []

    for token, tag in pos\_tag(tweet\_tokens):

        token = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+#]|[!\*\(\),]|'\

                       '(?:%[0-9a-fA-F][0-9a-fA-F]))+','', token)

        token = re.sub("(@[A-Za-z0-9\_]+)","", token)

        if tag.startswith("NN"):

            pos = 'n'

        elif tag.startswith('VB'):

            pos = 'v'

        else:

            pos = 'a'

        lemmatizer = WordNetLemmatizer()

        token = lemmatizer.lemmatize(token, pos)

        if len(token) > 0 and token not in string.punctuation and token.lower() not in stop\_words:

            cleaned\_tokens.append(token.lower())

    return cleaned\_tokens

def get\_all\_words(cleaned\_tokens\_list):

    for tokens in cleaned\_tokens\_list:

        for token in tokens:

            yield token

def get\_tweets\_for\_model(cleaned\_tokens\_list):

    for tweet\_tokens in cleaned\_tokens\_list:

        yield dict([token, True] for token in tweet\_tokens)

def sentiment\_analysis(sample\_tweet\_in):

    #custom\_tweet = "I hate the product"

    custom\_tokens = remove\_noise(word\_tokenize(sample\_tweet\_in))

    print(sample\_tweet\_in, classifier.classify(dict([token, True] for token in custom\_tokens)))

if \_\_name\_\_ == "\_\_main\_\_":

    positive\_tweets = twitter\_samples.strings('positive\_tweets.json')

    negative\_tweets = twitter\_samples.strings('negative\_tweets.json')

    text = twitter\_samples.strings('tweets.20150430-223406.json')

    tweet\_tokens = twitter\_samples.tokenized('positive\_tweets.json')[0]

    stop\_words = stopwords.words('english')

    positive\_tweet\_tokens = twitter\_samples.tokenized('positive\_tweets.json')

    negative\_tweet\_tokens = twitter\_samples.tokenized('negative\_tweets.json')

    positive\_cleaned\_tokens\_list = []

    negative\_cleaned\_tokens\_list = []

    for tokens in positive\_tweet\_tokens:

        positive\_cleaned\_tokens\_list.append(remove\_noise(tokens, stop\_words))

    for tokens in negative\_tweet\_tokens:

        negative\_cleaned\_tokens\_list.append(remove\_noise(tokens, stop\_words))

    all\_pos\_words = get\_all\_words(positive\_cleaned\_tokens\_list)

    freq\_dist\_pos = FreqDist(all\_pos\_words)

    print(freq\_dist\_pos.most\_common(10))

    positive\_tokens\_for\_model = get\_tweets\_for\_model(positive\_cleaned\_tokens\_list)

    negative\_tokens\_for\_model = get\_tweets\_for\_model(negative\_cleaned\_tokens\_list)

    positive\_dataset = [(tweet\_dict, "Positive")

                         for tweet\_dict in positive\_tokens\_for\_model]

    negative\_dataset = [(tweet\_dict, "Negative")

                         for tweet\_dict in negative\_tokens\_for\_model]

    dataset = positive\_dataset + negative\_dataset

    random.shuffle(dataset)

    train\_data = dataset[:7000]

    test\_data = dataset[7000:]

    classifier = NaiveBayesClassifier.train(train\_data)

    #print("Accuracy is:", classify.accuracy(classifier, test\_data))

    #print(classifier.show\_most\_informative\_features(10))\*/

    custom\_tweet = "@Naveen I love it"

    #sentiment\_analysis(custom\_tweet)

    custom\_tokens = remove\_noise(word\_tokenize(custom\_tweet))

    print(custom\_tweet, classifier.classify(dict([token, True] for token in custom\_tokens)))

# Output:





# Scope:

With the day by day increase in technology and thereby increasing the reliance of the humankind on it for getting the things done, it is becoming an important job to keep a check on peoples’ opinion about everything and it can be done by analysing what is the intent behind every comment and review posted online.

The scope of Sentiment analysis is broadening day by day with the increasing involvement of people in technology.

# Work Distribution:

Km Arisha Pathak-

1. Categorization of tweets as positive ornegative .
2. Classification of dataset using naïve bayes classifier.

Mann Bansal-

1. Training Dataset
2. Categorization of tweets as positive or negative .

# Libraries Used:

1. Nltk
2. WordNetLemmatizer
3. twitter\_samples, stopwords
4. pos\_tag
5. FreqDist
6. classify
7. NaiveBayesClassifier
8. re
9. string
10. random

# GitHub link:

<https://github.com/mann-bansal/sentiment-analysis>

# References:

<https://en.wikipedia.org/wiki/Sentiment_analysis>

<https://monkeylearn.com/sentiment-analysis>

<https://nevonprojects.com/artificial-intelligence>