**A PROJECT REPORT**

**ON**

**SENTIMENT ANALYSIS USING PYTHON**

Submitted in partial fulfillment of the requirement for the III semester

**Bachelor of Technology**

**By**

**ARYAN RAWAT**

**2016680**



**DEPARTMENT OF COMPUTER SCIENCE &** **ENGINEERING**

**GRAPHIC ERA DEEMED UNIVERSITY**

**DEHRADUN**

**2021-2022**

**ACKNOWLEDGEMENT**

Here by I am submitting the project report on **“Sentiment Analysis using Python”** as per the scheme of Graphic Era Deemed University, Dehradun.

I would like to express our sincere gratitude to Ms. Parul Madan**,** for providing a congenial environment to work in and carry out our project.

Finally, I am very much thankful to all the faculty members of the Department of Computer Science and Technology, friends and our parents for their constant encouragement, support and help throughout the period of project conduction.

NAME:ARYAN RAWAT

ROLL NO:2016680

MOTIVATION

 In the first and third semester ,I was learning C and C++,then I got my mini project which was very interesting and complicated, I always have used social media apps and wanted to see how do different people of the world have their attitude towards life whether it is positive or negative. Since, the burden of academic studies and the need to perform well in my college, I took a toll on my extra learning hours, I decided to opt for this project thereby indirectly making it a part of my syllabus and hence it led me to know how to efficiently work on large amount of data.

**PROBLEM STATEMENT**

**Sentiment Analysis using Python**

**SOFTWARE USED**

* Jupyter notebook,Anaconda

**TOOLS USED**

* Jypter notebook libararies like tweepy,textblob,pandas,matplotlib

**LANGUAGE**

* Python

**INTRODUCTION**

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Sentiment analysis is the automated process of identifying and classifying subjective information in text data. This might be an opinion, a judgment, or a feeling about a particular topic or product feature.

The most common type of sentiment analysis is ‘polarity detection’ and involves classifying statements as *Positive*, *Negative* or *Neutral*.

It is commonly used to analyze customer feedback, survey responses, and product reviews. Social media monitoring, reputation management, and customer experience are just a few areas that can benefit from sentiment analysis.

Sentiment analysis helps data analysts within large enterprises gauge public opinion, conduct nuanced market research, monitor brand and product reputation, and understand customer experiences. In addition, data analytics companies often integrate third-party [sentiment analysis APIs](https://www.lexalytics.com/semantria) into their own customer experience management, social media monitoring, or workforce analytics platform, in order to deliver useful insights to their own customers.

**METHDOLOGY**

In our program, we follow these 3 main steps:

* + Twitter API Client Authorization.
  + o Make a Twitter API GET request for a particular query to fetch tweets.
  + Just parse your tweets. Classify as positive, negative or neutral for each tweet.

Also, we use these different libraries provided by python to use these different kinds of functions for our code.

TextBlob ,Tweepy,pandas etc are the libraries which are used throughout the code.

**TextBlob** is actually a high level library built over top of [NLTK](http://www.nltk.org/) library. First we call clean the tweet method to remove links, special characters, etc. from the tweet using some simple regex.

Then, as we pass tweet to create a TextBlob object, following processing is done over text by textblob library.

* Tokenize the tweet, i.e split words from body of text.
* Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
* Do POS( part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.

**Tweepy** is an open source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter’s models and API endpoints, and it transparently handles various implementation details, such as:

* Data encoding and decoding
* HTTP requests
* Results pagination
* OAuth authentication
* Rate limits
* Streams

If I wasn’t using Tweepy, then you would have to deal with low-level details having to do with HTTP requests, data serialization, authentication, and rate limits. This could be time consuming and prone to error. Instead, thanks to Tweepy, you can focus on the functionality you want to build.

**RESULT**

Finally ,we are able to see the total positive or negative tweets of the user and also see a plotted graph which lets us see the total percentage of positive tweets with respect to negative tweets and vice versa.