**Internship Report on**

Crash Course on Python

*Submitted by*

## Areddula Lohith

## 20691A3128

*In partial fulfillment of the requirements for the award of the degree of*

## BACHELOR OF TECHNOLOGY

**in**

## ARTIFICIAL INTELLIGENCE



**MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE (UGC – AUTONOMOUS)**

**(Affiliated to JNTUA, Ananthapuramu) Accredited by NBA, Approved by AICTE, New Delhi)**

**AN ISO 9001:2008 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle – 517325**

## 2022-2023

**2009-2013**



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**BONAFIDE CERTIFICATE**

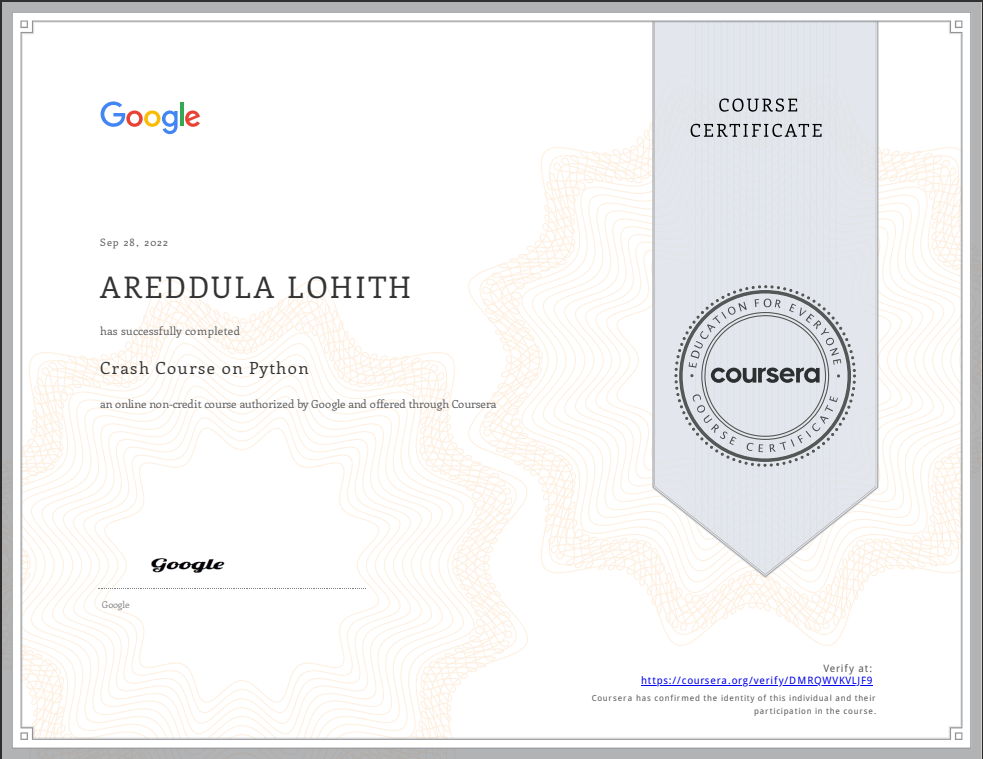
This is to certify that the internship work entitled **“Nlp With Tweet Emotions using Python”** is a bonafide work carried out by

## Areddula Lohith - 20691A3128

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the Department of **Artificial Intelligence or Data Science**, **Madanapalle Institute of Technology and Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022- 2023

|  |  |  |
| --- | --- | --- |
| **Mr. K. Durga Charan Assistant Professor, Department of AI & DS** | **Mr. Toralkar Pawan Assistant Professor Internship Coordinator**  **Department of AI & DS** | **Dr. K. Chokkanathan**  **Head & Asssociate Professor Department of AI & DS** |

## CERTIFICATE



**DECLARATION**

I, the undersigned hereby declare that the results embodied in this Internship **“NLP With Tweet Emotions”** is a bonafide record of the work done by me in partial fulfillment of the award of **Bachelor of Technology** in **Artificial Intelligence or Data Science** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu. The content of this report is** not submitted to any other University/Institute for award of any other degree.

|  |  |
| --- | --- |
| **Madanapalle-517325**  Date:25-11-2022 | **Name: Areddula Lohith**  Roll No: 20691A3128  Department of Artificial Intelligence & Data Science Madanapalle Institute of Technology & Science Madanapalle |

# ABSTRACT

Every social networking sites like facebook, twitter, instagram etc become one of the key sources of information. It is found that by extracting and analyzing data from social networking sites, a business entity can be benefited in their product marketing. Twitter is one of the most popular sites where people used to express their feelings and reviews for a particular product. In our work, we use twitter data to analyze public views towards a product. Firstly, we have developed a natural language processing (NLP) based pre-processed data framework to filter tweets. Secondly, we incorporate Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) model concept to analyze sentiment. This is an initiative to use BoW and TFIDF are used together to precisely classify positive and negative tweets. We have found that by exploiting TF-IDF vectorizer, the accuracy of sentiment analysis can be substantially improved and simulation results show the efficiency of our proposed system. We achieved 85.25% accuracy in sentiment analysis using NLP technique.

## ACKNOWLEDGEMENT

I sincerely thank the **Management** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

I sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D., Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

I express our deep sense of gratitude to **Dr. K. Chokkanathan, M. Tech., Ph.D.,** Associate Professor & Head, Department of AI & DS for his continuous support in making necessary arrangements for the successful completion of the project.

I express our sincere thanks to the **Internship Coordinator**, **Mr. Toralkar Pawan, M.E.,** Assistant Professor, Department of AI & DS for his tremendous support for the successful completion of the internship Project.

.

I express our deep gratitude to **Mr.K.Durga Charan,** Assistant Professor, Department of AI & DS for his guidance and encouragement that helped us to complete this internship project.

I also wish to place on record my gratefulness to other **Faculty members of Department of AI & DS** and also to our friends and our parents for their help and cooperation during our internship.

|  |  |
| --- | --- |
| **CONTENTS** | |
| **CHAPTER** | **PAGENO** |
| Abstract | v |
| Acknowledgement | Vi |
| **1. INTRODUCTION TO PYTHON** | **8** |
| 1.1.1Future Technologies Counting on Python | |
| **2. TOOLS AND TECHNIQUES** | **17** |
| 2.1 PLATFORM USED | |
| 2.1.1 Hardware Used | |
| 2.1.2 Software Used | |
| **3.PROJECT ANALYSIS** | **19** |
| 3.1 Overview | |
| 3.1.1 Proposed work | |
| 3.2 Project Architecture | |
| 3.3 Algorithms | |
| 3.3.1 Sentiment Analysis  3.3.2 Simulation and Performance Evaluation | |
| **CONCLUSION** | | **24** |
| **BIBLOGRAPHY** | | **25** |
| **APPENDICES** | | **26** |
|  |  |
|  |  |
|  |  |

# CHAPTER - 1: - INTRODUCTION TO PYTHON

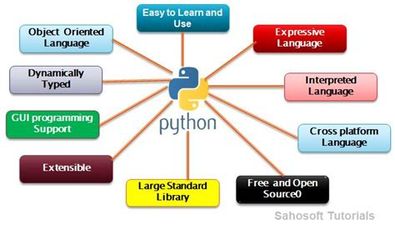
## PYTHON LANGUAGE INTRODUCTION

Python is widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that integrates systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted **-** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP. Python is Interactive**.** We can sit at a Python prompt and interact with the interpreter directly to write your programs. Python is Object-Oriented**.** It Python supports Object- Oriented style or techniques of programming that encapsulates code within objects.

## Python Features

Features of python includes Simple Syntax, Easy to Learn, Free & Open Source, High- Level Language, Portal, Interpreted, Memory Allocation, Object Oriented, Procedure Oriented, Classes, Extensible, Embedded.



## Easy to Code

It is a developer-friendly language which implies that anyone and everyone can learn to code and as compared to other object-oriented programming languages like Java, C, C++, and C# it is one of the easiest to learn.

## Open Source and Free

It is an open-source programming language meaning that anyone can create and contribute to its development. It has an online forum where thousands of coders gather daily to improve this language further. Along with this, it is free to download and use in any operating system, be it Windows, Mac, or Linux.

## Support for GUI

GUI or Graphical User Interface is one of the key aspects of any programming language because it has the ability to add flair to code and make the results more visual.It supports a wide array of GUIs which can easily be imported to the interpreter, thus making this one of the most favourite languages for developers**.** PyQt5 is the most popular option for creating graphical apps with Python.

## Object-Oriented

One of the key aspects of Python is its object-oriented approach. This basically means that Python recognizes the concept of class and object encapsulation thus allowing programs to be efficient in the long run.

9

## High-Level Language

It has been designed to be a high-level programming language, which means that when you code in Python you don’t need to be aware of the coding structure, architecture as well as memory management.

## Integrated by Nature (Interpreted Language)

It is an integrated language by nature meaning that the python interpreter executes codes one line at a time. Unlike other object-oriented programming languages, you don’t need to compile Python code thus making the debugging process much easier and efficient. Another advantage of this is, that upon execution the Python code is immediately converted into an intermediate form also known as bytecode which makes it easier to execute and also saves runtime in the long run.

## Highly Portable

Suppose you are running Python on Windows and you need to shift the same to either a Mac or a Linux system, then you can easily achieve the same in Python without having to worry about changing the code. This is not possible in other programming languages, thus making Python one of the most portable languages available in the industry.

## Support for Other Languages

Being coded in C, Python by default supports the execution of code written in other programming languages such as Java, C, and C#, thus making it one of the versatile in the industry.

## Future Technologies counting on Python

According to many pieces of research, Python is continuously raising its rank surpassing many languages. It is also a highly secure and reliable language and is gaining popularity since its release. This proves that Python surely has a bright future scope ahead. With the increasing demand for all these fields, there is a linearly increasing scope for Python. Many developers are making use of Python in countless applications such as gaming, industrial, scientific, financial, etc. These technologies count on Python for its expansion. Let’s discuss some of this

## Python in Artificial Intelligence (AI) and Machine Learning (ML):

In our day-to-day life, we all are using the products of ML and AI in one or the other form, Voice Assistant to give an example. AI has its hand in the development of almost every industry, with Information Technology being among the first. IT uses AI for improving efficiency, enhancing productivity, increasing security, and assuring quality. Also, the percentage of machine learning jobs is the most in Python with about 0.16%. We are also close to a new generation where advanced applications will be introduced in the forms of self-driving cars, medical diagnosis, etc. Python is a language that is most certainly to be used with its hundreds of diverse libraries allowing developers and programmers to automate tasks with a focus on the core requirements. And a whopping 69% of ML developers and data scientists now are using Python.

With more than Python’s libraries, such as Scikit, Pandas, NumPy, and Tensorflow, provide a flexible and dynamic foundation for effective and minimal cost implementation. Think of, for instance, how the Voice Assistants are able to understand our commands and give appropriate responses: this is made possible with the help of Python implementing deep learning and patterns analysis in data. Also, its integration capability with other languages makes it easy to convert programs written in C or C++. With the scope for more advancements in applications of AI and ML, like Robots, BrainChips, etc., Python would be expected to have more deployments and the future.

## Python for Big Data:

The increase the global business and the requirement for complex analysis offered the advancement of big data. The future of Python is promising in the field of Big Data through its frameworks and libraries. With built in tools like Numpy, Matplotlib, Pandas, SciPy, etc., particularly for complex calculation, data visualization, etc, data analysts are getting attracted towards Python. According to surveys in 2018 and 2020, Python developers are working more on data science rather than web or app development. In the data analytics industry, in which Python is being used, the market value is over $10 billion. With the companies desiring data analytics for more profits, faster turnover cycles, etc. more and more seats are availing to hire Python Developers. This is already observable in the ones that use Python such as Amazon Web Services, Google, etc. Thus, Python’s usage in Data Science is a symbol for its future scope.

## Networking and Automation

Another field where python has a brighter scope is networking. With the advancements of the Internet of Things (IoT) ranging from home appliances to public appliances like restaurants, deployment of the next “G” for communication, expansion of AI, etc. thrives for more networking and automation. Reading, writing, and configuring routers and switches and performing other networking automation can be performed in a cost-effective and secure manner using Python. There are many dedicating libraries like Ansible, Netmiko, PySNMP, etc. for networking. Python also helps in automating a range of tasks such as automating test cases in the software industry, scraping content from websites, and also repetitive office tasks. It can save both work and time by quickly accessing files. Thus, this could be an important python trend in this fast-moving era. And coming to the libraries part, they include PyUnit, Selenium, PyTest, etc.

## Web Development

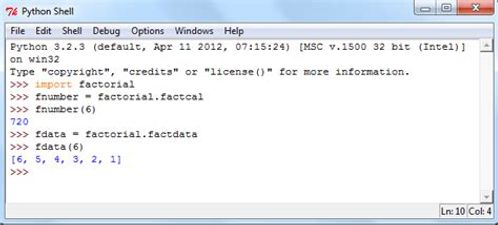
We all spend most of our time on the Internet. Do you know there are more than 1.92 billion websites online, over 5.1 billion active internet users, and about 7 million blog posts get published per day? It demands more technological innovations and tools to deliver the applications themselves. Previously companies hired people mainly for Python web development services. But today, Companies and App Developers are searching for who is called the DevOps Managers. And the obvious reasons would include its features are Its smooth

integration with other programming languages for embedded applications. Consisting of frameworks, particularly for web development such as Django, Pyramid, and Flask makes the development process faster by availing the facility to utilize built-in elements. Reduced development cost, that Python is open source and free. Thus, providing these developer- friendly facilities, a survey conducted got a response with nearly 85% of developers using Python as their primary programming language. Also, a large number of students show interest in Python. The demand for Python is expanding exponentially with many large-scale organizations are also making use of Python like Google, Facebook, Amazon, LinkedIn, Google, Uber, NASA, etc. Also, scholars are looking for their research work.

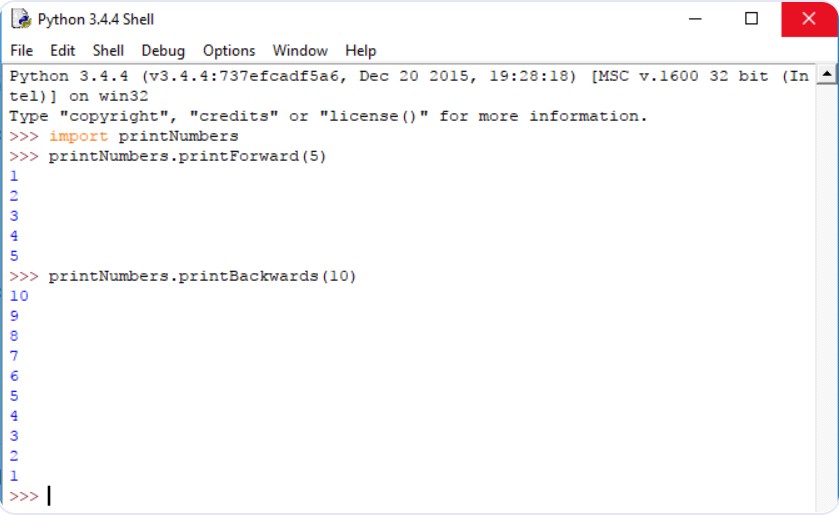
## PYTHON MODULES

Python Module is essentially a python script file that can contain variables, functions, and classes. Python modules help us in organizing our code and then referencing them in other classes or python scripts. A file containing Python definitions and statements is called a python module. So naturally, the file name is the module name which is appended with the suffix **.py**.

For better understanding let’s create a python module to explore it completely. First create a file named **printNumbers.py** with the following contents.



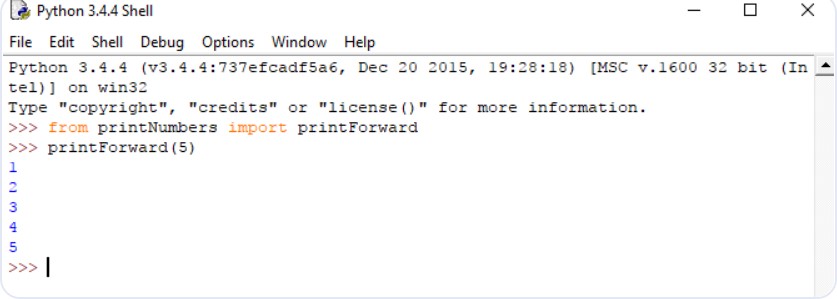
Now in the python interpreter import this module with the following command.This import command will look for **printNumbers.py** file in the current directory and PATH variable locations.



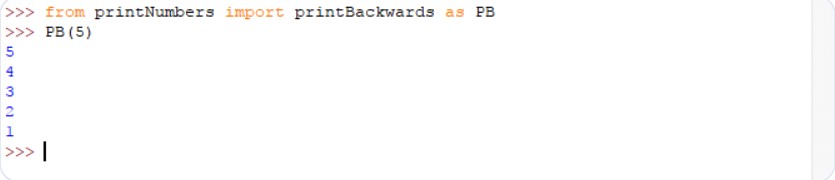
Once the file is found, the code in the file will be available for us to use. Now to access the module’s function we need to use the module name like below:

## Importing specific function of a Python Module

Sometimes it’s unnecessary to import all the functions of a python module. We may need only one or two functions. In that case, we can use the following variant of the import statement.



One thing to notice here, as we import **printForward**, it is included in the current symbol table. So, we don’t need to call the function like – **printNumbers.printForward()**. Another variant can be useful sometimes. Here we used to rename as we did previously to ease our use of the function.



Also, if we want to import all the names that a module defines there is another variant for importing. This imports all names except those beginning with an underscore (\_). But this is not ideal practice as this introduces an unknown set of names into the interpreter.



# CHAPTER 2: - TOOLS AND TECHNIQUES

## Platform Used

This section is to present a detailed description of the Platforms used for this project.

It explains the hardware and software requirements for developing the application and its interface, tested features of the program

## Hardware Used

Processors: Any two or higher core processor including AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz

RAM: 8GB of system memory from any decent manufacturer Disk space: 1-2 GB of Solid-State Drive (SSD)

Operating System: Windows 11 Home Single Language Version 21H2 System Type: 64-bit operating system, x64-based processor

## Software Used Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web and data science development. PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features. PyCharm offers some of the best features to its users and developers in the following aspects are Advanced debugging, Support for web programming and frameworks such as Django and Flask, Code completion and inspection

## Deta

Deta is a free cloud crafted with the developer and user experience at heart. Its mission

is to dramatically reduce the gap between ideas and working cloud applications. Deta provides easy to use tools for your development needs such as:

* + - * [Deta Base - Instantly usable database with a feature-rich API](https://docs.deta.sh/docs/home/#deta-base)
      * [Deta Micros - Deploy scalable Node & Python apps in seconds](https://docs.deta.sh/docs/home/#deta-micros)
      * [Deta Drive - Upload, host and serve images and files](https://docs.deta.sh/docs/drive/about)

Deta is accessible, lightweight, and scalable. In this project, Deta Micros are used. Deta Micros(servers) are a lightweight but scalable cloud runtime tied to an HTTP endpoint. Currently Node.js and Python Micros are supported.

## Windows PowerShell

Windows PowerShell is a command-line shell and scripting language designed especially for system administration. Its analogue in Linux is called as Bash Scripting. Built on the .NET Framework, Windows PowerShell helps IT professionals to control and automate the administration of the Windows operating system and applications that run on Windows Server environment. Windows PowerShell commands, called cmdlets, let you manage the computers from the command line. Windows PowerShell providers let you access data stores, such as the Registry and Certificate Store, as easily as you access the file system.

## Features

**Cmdlets** − Cmdlets perform common system administration tasks, for example managing the registry, services, processes, event logs, and using Windows Management Instrumentation (WMI).

**Task oriented** − PowerShell scripting language is task based and provide supports for existing scripts and command-line tools.

**Consistent design** − As cmdlets and system data stores use common syntax and have common naming conventions, data sharing is easy. The output from one cmdlet can be pipelined to another cmdlet without any manipulation.

**Simple to Use** − Simplified, command-based navigation lets users navigate the registry and other data stores similar to the file system navigation.

**Object based** − PowerShell possesses powerful object manipulation capabilities. Objects can be sent to other tools or databases directly.

**Extensible interface.** − PowerShell is customizable as independent software vendors and enterprise developers can build custom tools and utilities using PowerShell to administer their software.

## CHAPTER 3: PROJECT ANALYSIS

## NLP WITH TWEET EMOTIONS

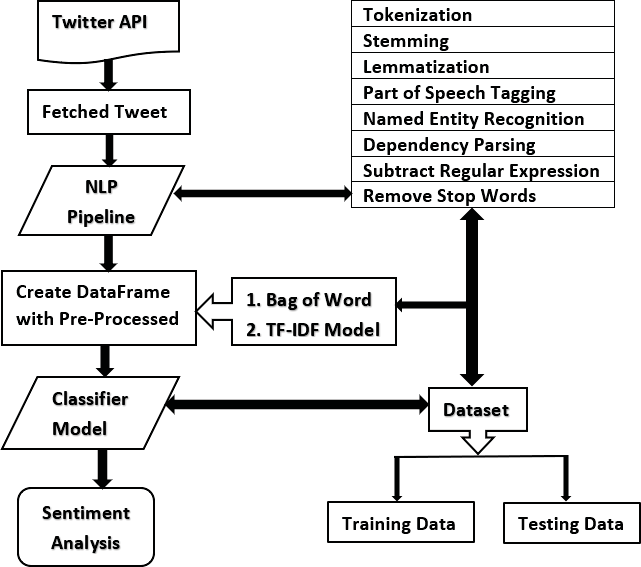
* 1. **Overview**

The main aim is to identify the sentiment of the tweet by defining positive and negative polarity where tweets are collected by using Twitter streaming API from Twitter. We utilize these tweets as crude data. At that point, we use the proposed technique that gives the assessment of tweet. From the sentiment analysis, the client will understand the feedback of the services before creating a buying deal.

**3.1.1 Proposed Work**

This section represents our proposed work and focuses on a strategy to sentiment analysis on Twitter data. The architectural overview describing an overall process design of sentiment analysis . The developed method is based on three important parts that are Data Extraction from a particular project or product, pre-processed the extracted Tweet using Natural language toolkit (NLTK) and Classifier model that calculates the sentiment of each Tweet.

* 1. **Project Architecture**



* 1. **Algorithm**

**Algorithm 1** Pseudocode To build classiﬁer Model

**Input**: A Dataset

**Output**: Create TFIDF model and Classifer Model

# Method

1. Importing dataset
2. Serialization and de-serialization
3. save to a ﬁle and then load again
4. FOR each data in dataset :
5. extract from and append required data into dataset
6. Create a simple binary bag of words model
7. Create a simple TF-IDF model
8. Create training and testing dataset :
9. Create classiﬁer model by using logistic regression
10. Pickle the classiﬁer
11. vectorizer TFIDFVectorizer ()
12. Unpickle the classiﬁer and vectorizer for read operation

**Algorithm 2** Pseudocode for Sentiment Analysis with NLP Technique on Twitter Data

**Input**: Fetching Tweets From Twitter

**Output**: Given Sentiment Polarity of Each Tweet

# Method

1. function twitterAPI setup():

Authenticating and accessing twitter keys

1. set argument ’iPhone’
2. return api
3. end function
4. list tweets = []
5. Create a dataframe list tweets by using pandas dataframe
6. Loading TF-IDF model and classiﬁer
7. retrieve data into matrix form TF-IDF model
8. load TF-IDF model
9. load classiﬁer
10. Importing re library
11. Initialize p🡪0 , n🡪0 ;
12. FOR tweet in list tweets :
13. extract required tweet from list tweet
14. Using vectorizer and classiﬁer model to predict sentiment
15. Print tweet with sentiment
16. IF sentiment[0] = = 1 :
17. increase positive tweet
18. ELSE :
19. increase negative tweet
20. Total positive *←* p

# Total negative *←* n

* + 1. **Sentiment Analysis**

Our developed method for sentiment analysis from twitter on twitter data is presented in this research. As an example, we show the comparison between two the most popular device ”iPhone” and ”Samsung”. We fetched 100 tweets on iPhone from Twitter. By using NLP pipeline we have cleaned redundant information and stop words are removed from tweets to sentiment analysis which has already discussed in the previous section. To predict sentiment now we used ”vectorizer” and ”classiﬁer” which is ”clf” object which is created from the ”tﬁdfmodel.pickle” and ”classiﬁer.pickle”. Then we show the prediction of sentiment according to each tweet for iPhone device .

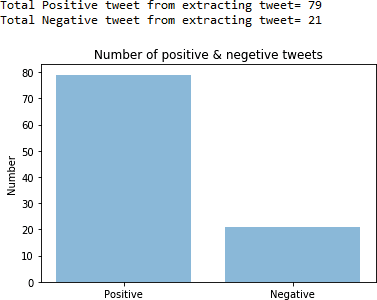


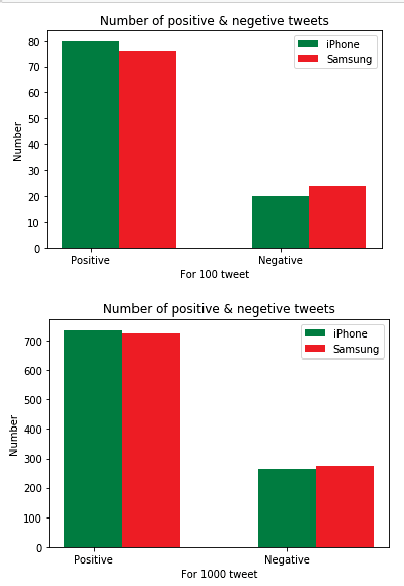
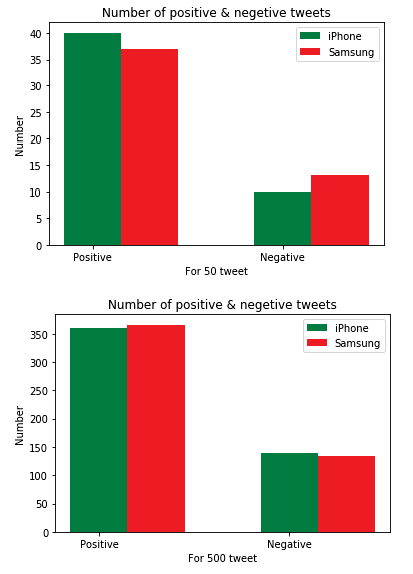
TABLE I: Sentiment evaluation for 1000 tweet

|  |  |  |
| --- | --- | --- |
| Output | Positive | Negative |
| Total Tweet for iPhone | 726 | 274 |
| Total Positive Tweet for Samsung | 712 | 282 |

TABLE II: Sentiment evaluation for 100 tweet

|  |  |  |
| --- | --- | --- |
| Output | Positive | Negative |
| Total Tweet for iPhone | 66 | 34 |
| Total Positive Tweet for Samsung | 70 | 30 |

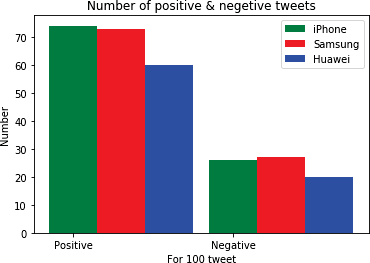
It just gives an inference about the sentiment analysis. To get better understand and visualization, we have shown different sentiment according to a number of collecting tweet such as 50, 100, 500, 1000 respectively which is shown in Fig. 7. Two companies have launched many phones within a year where some phones have taken more popularity than other company’s phone and it varies phone to phone. So, it is difﬁcult to say one company is more popular than other company because the ratio of positive and negative tweet among two phones isn’t signiﬁcant. But we can say both two phone company are more popular than other phone company.



**3.3.2 Simulation and Performance Evaluation**

We fetched 1000 tweet for iPhone and Samsung then calculate the total positive and negative sentiment for both devices . Here, we have seen that the iPhone phones are more popular than Samsung Phone. For 100 tweet for iPhone and Samsung, the comparison.

Here, we can see that the Samsung phones are more popular than iPhone. But The result doesn’t create any major inﬂiction,



**Performance Evaluation**

From TABLE III, IV, V, we have got maximum accuracy of 85.25% according to max features, minimum document frequency (min df), maximum document frequency (max df). We compare our proposed method with support vector machine (SVM) , maximum entropy , naive bayes algorithm and k-nearest neighbor classiﬁer . Therefore, we have taken the accuracy of 2000 features for each technique and evaluated the performance of the techniques which are more efﬁcient for the sentiment analysis.

we see that our proposed technique achieved 85.25% accuracy which outperforms other techniques.

TABLE III: Accuracy for min df=1

|  |  |  |  |
| --- | --- | --- | --- |
| Max features | min df | max df | Accuracy(%) |
| 2000 | 1 | 0.1 | 78 |
| 2000 | 1 | 0.2 | 79 |
| 2000 | 1 | 0.3 | 81.5 |
| 2000 | 1 | 0.4 | 82.75 |
| 2000 | 1 | 0.5 | 84.75 |
| 2000 | 1 | 0.6 | 84.75 |
| 2000 | 1 | 0.7 | 84.75 |
| 2000 | 1 | 0.8 | 85.25 |
| 2000 | 1 | 0.9 | 84.75 |
| 2000 | 1 | 1.0 | 84.25 |

TABLE IV: Accuracy for min df=2

|  |  |  |  |
| --- | --- | --- | --- |
| Max features | min df | max df | Accuracy(%) |
| 2000 | 2 | 0.1 | 78 |
| 2000 | 2 | 0.2 | 79 |
| 2000 | 2 | 0.3 | 80.75 |
| 2000 | 2 | 0.4 | 82.5 |
| 2000 | 2 | 0.5 | 84.75 |
| 2000 | 2 | 0.6 | 84.5 |
| 2000 | 2 | 0.7 | 84.75 |
| 2000 | 2 | 0.8 | 85.25 |
| 2000 | 2 | 0.9 | 85 |
| 2000 | 2 | 1.0 | 84.25 |

TABLE V: Accuracy for min df=3

|  |  |  |  |
| --- | --- | --- | --- |
| Max features | min df | max df | Accuracy(%) |
| 2000 | 3 | 0.1 | 78.25 |
| 2000 | 3 | 0.2 | 79.25 |
| 2000 | 3 | 0.3 | 81.25 |
| 2000 | 3 | 0.4 | 82.75 |
| 2000 | 3 | 0.5 | 84.75 |
| 2000 | 3 | 0.6 | 84.75 |
| 2000 | 3 | 0.7 | 85 |
| 2000 | 3 | 0.8 | 85 |
| 2000 | 3 | 0.9 | 85 |
| 2000 | 3 | 1.0 | 84.25 |

# CONCLUSION

In This Project we have developed a model to sentiment analysis which allows the processing of Twitter API streaming feed in real time and to classify its polarity to provide valuable insight in industry and users. Our built classiﬁer can be utilized as data analysis tools in NLTK. Therefore, in general we can use our proposal technique to sentiment analysis for any device, public ﬁgure or sports team that is better than.

any other existing model with high accuracy performance.

we tried to show the basic way of classifying tweets into positive or negative category using Naive Bayes as baseline and how language models are related to the Naive Bayes and can produce better results. We could further improve our classifier by trying to extract more features from the tweets, trying different kinds of features, tuning the parameters of the naïve Bayes classifier, or trying another classifier all together.

**BIBLIOGRAPHY**

● Alexander Pak, Patrick Paroubek. 2010, Twitter as a Corpus for Sentiment Analysis and Opinion Mining.

● Alec Go, Richa Bhayani, Lei Huang. Twitter Sentiment Classification using Distant Supervision.

● Jin Bai, Jian­Yun Nie. Using Language Models for Text Classification.

● Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. Sentiment Analysis of Twitter Data.

● Fuchun Peng. 2003, Augmenting Naive Bayes Classifiers with Statistical Language Models.

**APPENDICIES**

**CODE**

**Tweet\_Emotions.py**

from google.colab import drive

drive.mount('/content/drive')

**#Data Prepartion**

### load pkgs

import pandas as pd

import numpy as np

df=pd.read\_csv("/content/drive/MyDrive/Dataset/tweet\_emotions.csv")

# Dataset is now stored in a Pandas Dataframe

# load data viz pkgs

import matplotlib.pyplot as plt

import seaborn as sns

#value counts of emotions

df['sentiment'].value\_counts()

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

sns.countplot(x='sentiment',data=df)

plt.show()

#value counts of emotions

df['sentiment'].value\_counts().plot(kind='bar')

import re

import nltk

from nltk.corpus import stopwords

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import MultinomialNB

from sklearn.linear\_model import LogisticRegression

from textblob import Word

sw = stopwords.words("english")

training=df

training["sentiment"].value\_counts()

total = training.isnull().sum().sort\_values(ascending=False)

percent = (training.isnull().sum()/training.isnull().count()).sort\_values(ascending=False)

missing\_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

missing\_data.head(20)

def hapus\_url(text):

return re.sub(r'http\S+','', text)

def remove\_special\_characters(text, remove\_digits=True):

pattern=r'[^a-zA-Z0-9\s]'

text=re.sub(pattern,'',text)

return text

def stemmer(text):

ps=nltk.porter.PorterStemmer()

text=' '.join([ps.stem(word) for word in text.split()])

return text

def final\_clean(text):

final\_text= []

for i in text.split():

if i.strip().lower() not in sw and i.strip().lower().isalpha():

final\_text.append(i.strip().lower())

return " ". join (final\_text)

def clean(text):

text = hapus\_url(text)

text = remove\_special\_characters(text, remove\_digits=True)

text = stemmer(text)

text = final\_clean(text)

return text

training['content'] = training['content'].apply(clean)

%matplotlib inline

training.columns

training.drop(['tweet\_id'], axis=1, inplace=True)

**#Training**

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf\_vect=TfidfVectorizer()

text1 = tfidf\_vect.fit\_transform(training["content"])

y = training["sentiment"]

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test,y\_train,y\_test = train\_test\_split(text1, y, test\_size=0.3, random\_state=123)

from sklearn.svm import SVC

from sklearn.metrics import classification\_report

print(X\_train.shape)

model = SVC()

model.fit(text1,y)

pred\_svm = model.predict(X\_test)

print(classification\_report(pred\_svm, y\_test))

from sklearn import svm, datasets

from sklearn.metrics import roc\_curve, auc

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import label\_binarize

from sklearn.multiclass import OneVsRestClassifier

from sklearn.metrics import roc\_auc\_score

classifier = OneVsRestClassifier(

svm.SVC(kernel="linear", probability=True, random\_state=random\_state)

y=df['sentiment'].value\_counts()

y\_score = classifier.fit(X\_train, y\_train).decision\_function(X\_test)

**OUTPUT:**

