

# **Project Report**

## **Tweet Sentiment Analyser**

In the partial fulfillment of the requirement for the award of the degree of

**Bachelor of Technology**



Department of Computer Science

**Institute of Engineering and Rural Technology**

Prayagraj, (U.P.)

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University**

Lucknow( U.P.), INDIA

Submitted By - Ayush Kumar Singh

AKTU Roll No. - 1811022017

Under the guidance of - Dr. Rohit

# Project Report

By - Ayush Kumar Singh

Branch Roll No. - (62)

Aktu Roll No. - 1811022017

Team Member (1)

1. Ayush Kumar Singh

# Table of Contents

1. Introduction to Sentiment Analysis
2. Introduction to Python
3. Working (Step By Step)
4. Analysis
5. Result
6. Architecture Diagram
7. Conclusion
8. References

# 1. Introduction to Sentiment Analysis

The sentiment is a thought, an attitude, view, judgment, or opinion prompted by a feeling. Sentiment analysis studies users' sentiments towards certain objects. Internet Wets is a resourceful and capable place with respect to sentiment information.

From a user's perspective, people are able to post their own content through various social media, forums, blogs, E-commerce or online social networking sites.

From a researcher's perspective, many social media sites release their application programming interfaces (API) prompting data collection and analysis by researchers and developers.

## 1.1 Sentiment Analysis: A Fascinating Problem

Sentiment Analysis is a process of research and evaluation that analyzes people's emotions, opinions, thoughts, viewpoints, sentiments, and attitudes towards objects or entities such as products, services, organizations, individuals, events, topics, and their attributes.

There are also many different names with slight deviations in the task.

1. sentiment analysis
2. opinion extraction
3. sentiment mining
4. opinion mining Sappraisal extraction
6. subjectivity analysis
7. emotion analysis
8. Affect analysis
9. review mining

However, all of them fall under the umbrella of sentiment analysis or opinion.

Although natural language processing has been for years, little research has been done about people's viewpoints and sentiments. The past year 2000.

The Opinion mining field has become a very active research area. There are several reasons for this.

1. It has a wide arrange of applications in almost every domain. The industry adjoining sentiment analysis is flourishing due to the explosion of commercial applications. This provides a strong motivation for research.
2. It offers many challenging research problems, which had never been studied before.
3. For the first time in history, we have a huge volume of user opinionated data on the social media on the Web. A lot of research would not have been possible without this data. The rapid growth of sensitivity analysis coincides with that of social media. In fact, it is now right at the core of social media research.

"What other people think" has always been an important piece of information for most of us during the decision-making process. In the past before awareness of the Internet became widespread, we all used to ask our knowns (friends, relatives, colleagues) to recommend a car mechanic or to explain who and why they are planning to vote for in the public elections, consulted Consumer Reports to decide what product to buy or asked for reference letters regarding job applicants from colleagues. But the Internet and the Web have now (among other things) made it possible to find out about the experiences and opinions of those in the vast pool of users that are neither the personal acquaintances nor professional critics-that is, people we have never met or heard of. More and more people are making their views available to strangers via the Internet.

According to two surveys of more than 2000 American adults:

- 80% of Internet users have done online research on a product at least once;
- 20% do so on a typical day;
- among readers of online reviews of hotels, restaurants, and various services (e.g., travel or doctors), between 73% and 87% report that reviews had a vital influence on their purchase;
- consumers report that they are willing to pay from 20% to 99% more for a better 5-star-rated item than a 4-star-rated item.
- 32% have provided a rating on a service, product, brand, or person via an online rating medium, and 30% have posted an online comment or review regarding a brand, product, or service.

I would like to point out that consumption and usage of products and services are not the only inspiration behind people's looking for or expressing views or opinions online. A need for political and social information is another important factor.

The hunger and dependency of users upon online recommendations and advice that the data above discloses is one big reason behind the flood of interest in new systems that deal directly with opinions as a first-class object.

## 1.2 Sentiment Analysis Applications

With the speedy growth of social media that comprises posting reviews, forum discussions, writing blogs/microblogs, tweets, comments, likes, ratings, and postings on social network sites on the Web, users (individuals and organizations) have increasingly started using the content in these media for predictive analysis, forecasting and decision making. Today if a consumer wants to buy a service or product, they are no longer limited to requesting friends, colleagues, and family for views and ideas because there are many customer reviews, thoughts, opinions, and discussions in public forums available on the Internet about the service or product.

However, searching, reviewing, and monitoring opinion web pages on the Internet and cleansing the information contained in them remains a tough task because of the propagation of varied sites. Each site normally contains an enormous volume of judgment text that is not always easily decoded in long blog posts and forum postings. The average reader will struggle to identify relevant websites and extract, cleanse, and summarize the opinions in them. Automated opinion and sensitivity analysis systems are thus needed.

In recent years, we have witnessed that opinionated postings on social media have helped reshape businesses, and sway public sentiments and emotions which have profoundly impacted our social and political systems. Hence, it is critical to collect and study opinions data on the Web.

Due to these applications, industrial activities have flourished in recent years. Sensitivity analysis applications have spread to almost all possible areas, from consumer products, services, politics, health and welfare, banks, and financial services to cultural, social events, and political events and elections. There have been at least 50-70 start-up companies in the USA alone. Many big corporations have also built their own in-house capabilities, eg, Facebook, Amazon, Microsoft, Google, IBM, SAP, and HP. These practical implementations and industrial interests have provided strong inspirations for research and study in sentiment analysis.

Sentiment Analysis can be done on all kinds of texts, namely

- Newspaper texts
  1. Financial News
  2. Entertainment News
- Legal texts
- Novels E-mails
- SMS messages
- WhatsApp messages

- Customer Reviews
- Blog Posts Tweets
- Facebook Posts and so on.

A few of the Applications of Sentiment Analysis are

- Tracking sentiment toward politicians, movies, and products.
- Improving Customer Relationship Models.
- Identifying what evokes strong emotions in people.
- Detecting happiness and well-being.
- Measuring the impact of activist movements through text generated in social media.
- Improving Automatic Dialogue systems
- Improving Automatic Tutoring systems.
- Detecting how people use emotion-bearing words and metaphors to persuade and coerce others.

## 1.3 Different Levels of Sentiment Analysis

Overall, Sentiment Analysis has been explored mainly at the below levels:

1. **Document Level:** The task at this level is to classify whether a whole text opinion document states a positive or negative sentiment. For eg if given a consumer product review, this process analyzes and determines whether the review comment expresses an overall opinion in that content about the particular product as neutral, positive, or negative. This task is usually called document-level sentiment classification. This level of analysis assumes that each document expresses views on a single entity Le. a single product, Thus, it is not applicable to documents that assess or compare multiple objects and entities.

2. **Sentence Level:** This task goes to the level of sentences and concludes if each of the sentences expressed a positive, negative or neutral opinion. Neutral typically means no opinion. This level of analysis is related to subjectivity classification closely which separates sentences (called objective sentences) that express factual information from sentences (called subjective sentences) that express subjective views and opinions. However, it should be noted that subjectivity is not equivalent to sentiment as many objective sentences can imply opinions eg. "I bought the mobile yesterday and the screen has broken"

**3. Entity and Aspect Level:** The sentence level and document level analysis do not determine what exactly people liked and did not like. The aspect level performs an in-depth analysis. Aspect level was previously called feature level Le. feature-based opinion mining and summarization. Instead of looking at language constructs (documents, paragraphs, sentences, clauses, or phrases), the aspect level directly looks at the opinion itself. It is based on the idea that an opinion consists of a sentiment that is either positive or negative and a target. An opinion without identifying its target being is of partial or no use. We need to realize the importance of targets of opinions in order to understand the sentiment analysis problem better. For example, the sentence "although the manager is rude. I still love this hotel clearly has a positive tenor, we cannot say that this sentence is entirely positive. In fact, the sentence is positive about the hotel (emphasized) but negative about its manager (not emphasized). In many applications, opinion targets are described by objects or entities and/or their different aspects. Hence, the goal of this level of analysis is to discover sentiments on entities and/or their aspects.

## 1.4 Use Cases of Sentiment Analysis

### 1. Real-time Political Analysis

- Data-driven media and journalism.
- . PR management for political figures and parties

### 2. Financial Analysis

### 3. Radicalization Detection

### 4. Marketing and Advertising



## 2. Introduction to Python

### 2.1 What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source-level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

### 2.2 Advantages of Python

#### 1. Easy to Read, Learn and Write

Python is a high-level programming language that has an English-like syntax. This makes it easier to read and understand the code.

Python is really easy to pick up and learn, that is why a lot of people recommend Python to beginners. You need fewer lines of code to perform the same task as compared to other major languages like C/C++ and Java.

#### 2. Improved Productivity

Python is a very productive language. Due to the simplicity of Python, developers can focus on solving the problem. They don't need to spend too much time in understanding the syntax or behavior of the programming language. You write less code and get more things done.

### **3. Interpreted Language**

Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred.

Python shows only one error even if the program has multiple errors. This makes debugging easier.

### **4. Dynamically Typed**

Python doesn't know the type of variable until we run the code. It automatically assigns the data type during execution. The programmer doesn't need to worry about declaring variables and their data types.

### **5. Free and Open-Source**

Python comes under the OSI-approved open-source license. This makes it free to use and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behavior and use their version for development.

### **6. Vast Libraries Support**

The standard library of Python is huge, you can find almost all the functions needed for your task. So, you don't have to depend on external libraries.

But even if you do, a Python package manager (pip) makes things easier to import other great packages from the Python package index (PyPi). It consists of over 200,000 packages.

### **7. Portability**

In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

However, you should be careful not to include any system-dependent features.

## **2.3 Python Modules Used In This Project**

1. Pandas
2. Plotly
3. SNS Scraper
4. Vader Sentiment Analyser
5. Streamlit

## **2.3.1 Pandas**

### **2.3.1.1 What is Pandas?**

Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data-wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.

### **2.3.1.2 Application and Use Case of Pandas**

1. Recommendation systems on Online Platforms
2. Economics
3. Neuroscience
4. Prediction of stocks
5. Advertising
6. Statistical analysis
7. Natural Language Processing (NLP)
8. Analytics
9. Data science
10. Big Data

### **2.3.1.3 Why Pandas Dataframe is used in this project?**

#### **1. Data representation**

Pandas provide extremely streamlined forms of data representation. This helps to analyze and understand data better. Simpler data representation facilitates better results for data science projects.

#### **2. Less writing and more work done**

It is one of the best advantages of Pandas. What would have taken multiple lines in Python without any support libraries, can simply be achieved through 1-2 lines with the use of Pandas. Thus, using Pandas helps to shorten the procedure of handling data. With the time saved, we can focus more on data analysis algorithms.

#### **3. An extensive set of features**

Pandas are really powerful. They provide you with a huge set of important commands and features which are used to easily analyze your data. We can use Pandas to perform variously

tasks like filtering your data according to certain conditions, segmenting and segregating the data according to preference, etc.

#### **4. Efficiently handles large data**

Wes McKinney, the creator of Pandas, made the python library mainly handle large datasets efficiently. Pandas help to save a lot of time by importing large amounts of data very fast.

#### **5. Makes data flexible and customizable**

Pandas provide a huge feature set to apply to the data you have so that you can customize, edit and pivot it according to your own will and desire. This helps to bring the most out of your data.

#### **6. Made for Python**

Python programming has become one of the most sought-after programming languages in the world, with its extensive amount of features and the sheer amount of productivity it provides. Therefore, being able to code Pandas in Python, enables you to tap into the power of the various other features and libraries which will use with Python. Some of these libraries are NumPy, SciPy, Matplotlib, etc.

### **2.3.2 Plotly**

#### **2.3.2.1 What is Plotly?**

The Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

Built on top of the Plotly JavaScript library (plotly.js), plotly enables Python users to create beautiful interactive web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash. The Plotly Python library is sometimes referred to as "plotly.py" to differentiate it from the JavaScript library.

Thanks to deep integration with our Kaleido image export utility, plotly also provides great support for non-web contexts including desktop editors (e.g. QtConsole, Spyder, PyCharm) and static document publishing (e.g. exporting notebooks to PDF with high-quality vector images).

#### **2.3.2.2 Plotly Generates**

- **Bar charts:** Bar charts are used when we want to compare different groups of data and make inferences of which groups are highest and which groups are common and compare how one group is performing compared to others.
- **Scatter Plot:** Scatter plot represents values for two different numeric variables. They are mainly used for the representation of the relationship between two variables.

- **Pie chart:** A pie chart represents the distribution of different variables in total. In the pie chart, each slice shows its contribution to the total amount.
- **Histogram:** A histogram plots the continuous distribution of a variable as a series of bars and each bar indicates the frequency of the occurring value in a variable. In order to use a histogram, we simply require a variable that takes continuous numeric values

### 2.3.2.3 Why Plotly is used in this project?

The main benefits of Plotly are ease of use, advanced analytics, reduced costs, scalability, and total customization.

- **Ease of Use**

Plotly is a user-friendly data visualization software that offers highly advanced visualization tools. No advanced training and knowledge are necessary to fully utilize all tools and features. It even has an open development process that seamlessly allows users to fully customize functions. Beginners will not be intimidated with its clean and simple dashboard.

- **Advanced Analytics**

Plotly is packed with high-powered tools for analytics that can take on computes vision, ML, forecasting, NLP, and more. Data scientists can freely work with familiar Python, Julia, and R languages.

- **Improved Productivity**

Users can easily speed up work and avoid bottlenecks and delays through the use of centralized project dashboards. Teams can collaborate and share files seamlessly.

- **Reduced Costs**

Plotly is powerful enough to have users dispense of the need for a dedicated team of developers and IT experts. It can take on tasks that require an IT team, front-end developers, and back-end developers. It also offers numerous pricing plans that can cater to the on-premise and cloud-based needs of users.

- **Scalability**

Plotly can work with solo researchers, SMBS, startups, and even enterprises. All enterprise-grade tools are available even for solo practitioners and small teams.

- **Total Customization**

Plotly has an open API that can be used to fully customize any user's experience. It can easily integrate with third-party apps and work with already existing workflow structures.

## 2.3.3 SNS Scraper

### 2.3.3.1 What is SNS Scraper?

snscape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts.

The following services are currently supported:

- Facebook: user profiles, groups, and communities (aka visitor posts)
- Instagram: user profiles, hashtags, and locations
- Mastodon: user profiles and toots (single or thread)
- Reddit: users, subreddits, and searches (via Pushshift)
- Telegram: channels
- Twitter: users, user profiles, hashtags, searches, tweets (single or surrounding thread), list posts, and trends
- VKontakte: user profiles
- Weibo (Sina Weibo): user profiles

### 2.3.3.2 SNS Scraper Advantages

Why are we even talking just about snscape? There should be more options to do the job. Of course, there are. (or rather there were). And below are the problems with other tools or packages which leads us to snscape

#### 1. Tweepy

Tweepy is one of the most popular Python libraries to set up access with Twitter. You create a Twitter Developer Account, get credentials and start scraping. It is a great tool for simple automation, creating Twitter bots, or a small school project. However, Tweepy has a scraping limit of 3200 tweets and the farthest time you can go in a week. There is no access to historical data.

#### 2. GetOldTweets3

Twitter has removed the endpoint the GetOldTweets3 uses and that makes GOT no longer useful. You will find a LOT of projects on GitHub that used GetOldTweets but as of Nov 27, 2019, there have been no updates to the package to adhere to the revised Twitter guidelines.

### **3. TWINT**

Twint is an advanced tool written in Python but Twitter apparently has a more strict device + IP-ban after a certain amount of queries. It's a hit or miss. A lot of people including me, have issues installing. The author of the library recommends using a Dockerfile temporarily while the solution becomes clearer.

### **4. Octoparse**

Octoparse is a paid software that allows you to get data from the web without having to code by automatically extracting content from the webpage. Basically, you select a frame on the webpage and scroll down to a point where you see an end to your scraping. I personally found it very time-consuming and the learning curve is tough, apart from it not being free.

#### **2.3.3.3 Why SNS Scraper is used in this project**

It is being used to scrape tweets from twitter based on the user input.

### **2.3.4 Vader**

#### **2.3.4.1 What is Vader?**

VADER ( Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data.

VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

For example- Words like 'love', 'enjoy', 'happy', 'like' all convey a positive sentiment. Also VADER is intelligent enough to understand the basic context of these words, such as "did not love" as a negative statement. It also understands the emphasis of capitalization and punctuation, such as "ENJOY"

#### **2.3.4.2 Why Use Vader for this project?**

VADER is the superior library for performing sentiment analysis on social media data. Natural Language Processing in general is a highly experimental sector of machine learning, and it's all about finding the right packages that can adapt to your dataset. In this case, with the Twitter data, we needed a lexicon that was trained on dirty, nuanced text data.

## 2.3.5 Streamlit

### 2.3.5.1 What is Streamlit?

It is the app framework specifically for Machine Learning and Data Science teams. You can rapidly build the tools you need. Build apps in a dozen lines of Python with a simple API.

Streamlit is a tool in the Machine Learning Tools category of a tech stack. Streamlit is an open source tool with 10.4K GitHub stars and 892 GitHub forks. Here's a link to Streamlit's open source repository on GitHub.

#### Streamlit's Features

- Free and open-source
- Build apps in a dozen lines of Python with a simple API
- No callbacks
- No hidden state
- Works with TensorFlow, Keras, PyTorch, Pandas, Numpy, Matplotlib, Seaborn, Altair, Plotly, Bokeh, Vega-Lite, and more

### 2.3.5.2 Advantages of Streamlit?

- In minutes, Streamlit transforms data scripts into shareable web apps.
- Everything is written in Python. It's all free.
- There is no requirement for prior front-end experience.
- To easily share, maintain, and collaborate on your apps, use Streamlit's invite-only sharing tool.
- With Streamlit Components, you can create HTML components fast.



## 3. Working (Step By Step)

### 3.1 Data Collection

The data collection is being done in this project is done using sns scraper a module that is used to scrape Twitter.

Before the collection of data two inputs are being taken.

1. Query is being taken from the user. In default, we have "Ukraine War" set to the input.
2. Amount of tweets. Default is set to 100 tweets.

This input is then provided to the sns scraper as a search term and then the amount of tweets is the number of times our loop runs.

This entire process happens to collect data from Twitter.

Every tweet scraped right now has 4 data points.

1. User Id
2. Date & Time of tweet
3. Tweet Content
4. Username of the person who tweeted

### 3.2 Sentimental Analysis

In the for loop as soon as a tweet is scraped and sentimental analysis is performed on it using Vader before appending it into an array.

Each data appended in the array has the tweet plus the sentimental analysis data which includes a month of tweets, and positive, negative, and neutral sentiments.

The month of a tweet is being appended here to be used in-line chart that is displayed further.

### 3.3 Dataframe Conversion

The appended array is then converted into dataframe. This is done by passing the value of the array in which data was appended before. This conversion takes place using pandas dataframe.

## 3.4 Visual Charts

Two charts are constructed using Plotly bypassing the data we collected above.

**1. Pie Chart** - This chart contains the percentage of positive, negative, and neutral tweets in all of the tweets.

**2. Line Chart** - This line shows the sentimental change of data per month.

## 3.5 Filtering Dataframe

The dataframe is filtered to display three more types of data.

**Most Positive Tweet** - Tweets with the highest positive sentiment

**Most Negative Tweet** - Tweets with the highest negative sentiment

**Most Neutral Tweet** - Tweets with the highest positive sentiment

## 4. Analysis

### 4.1 Importing Python Modules

```
import snsrape.modules.twitter as sntwitter
import pandas as pd
import json
import streamlit as st
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import plotly.express as px
from datetime import datetime
```

### 4.2 Array and Putting Vader into a variable

```
analyzer = SentimentIntensityAnalyzer()

tweets_list1 = []
tcontent = []
tusername = []
tid = []
tdate = []
tvolume = []
ee = []
volume = []
ee = []
```

### 4.3 Getting Query and Amount of Tweets from User

```
title = st.text_input('Find Sentiment', 'Ukraine War')
heading = "Sentimental Analysis Of " + title + " Tweets"
option = st.selectbox(
    'How would tweets do you want to analyze?',
    (100, 1000, 10000, 100000, 1000000))
```

## 4.4 Scraping Tweets, Performing Sentimental Analysis, and appending them into array

```
dev = int(option/100)
my_bar = st.progress(0)
# Using TwitterSearchScrapper to scrape data and append tweets to list
for i,tweet in enumerate(sntwitter.TwitterSearchScrapper(title +' since:2021-01-01
until:2021-12-31').get_items()): #declare a username
    if i>option:
        break
    # data to be printed
    #st.write(tweet.content)
    #st.write(tweet.user.username)
    #st.write(tweet.id)
    #st.write(tweet.date)
    tcontent.append(tweet.content)
    tusername.append(tweet.user.username)
    tid.append(tweet.id)
    d = datetime.strptime(str(tweet.date), '%Y-%m-%d %H:%M:%S+00:00')
    tdate.append(d.day)

    vs = analyzer.polarity_scores(tweet.content)
    pos = vs['pos'] * 100
    neg = vs['neg'] * 100
    neu = vs['neu'] * 100

    volume.append(1)
    if vs['compound'] >= 0.05 :

        ccc = "Positive"

    elif vs['compound'] <= - 0.05 :

        ccc = "Negative"

    else :

        ccc = "Neutral"

    ee.append(ccc)
    if(i % dev == 0):
        my_bar.progress(int(i/dev))
    tweets_list1.append([tweet.date, tweet.id, tweet.content, tweet.user.username, ccc,
d.month,pos, neg, neu])
# printing array
#st.write(tweets_list1)
my_bar.empty()
```

## 4.5 Plotting plotly Pie Chart

```
random_x = volume
names = ee

fig = px.pie(values=random_x, names=names,title=heading)

st.plotly_chart(fig)
```

## 4.6 Converting Array to Pandas Dataframe, Saving It, and Printing it

```
# Creating a dataframe from the tweets list above
tweets_df1 = pd.DataFrame(tweets_list1, columns=['Datetime', 'Tweet Id', 'Text', 'Username',
'Sentiment', 'Month', 'Positive', 'Negative', 'Neutral'])

tweets_df1.to_pickle("ayush.pkl")
output = pd.read_pickle("ayush.pkl")
st.write(output)
```

## 4.7 Generating Line Chart

```
def custom_legend_name(new_names):
    for i, new_name in enumerate(new_names):
        fig.data[i].name = new_name

month = 1

yaxis = []
posaxis = []
nueaxis = []

while(month < 13):

    New_df = output.loc[output["Month"] == month]
    New_df2 = New_df.loc[New_df["Sentiment"] == "Negative"]
    New_df3 = New_df.loc[New_df["Sentiment"] == "Positive"]
```

```

New_df4 = New_df.loc[New_df["Sentiment"] == "Neutral"]

if(len(New_df2) > 0):

    percentage2 = len(New_df2)/len(output)
    percentage3 = percentage2 * 100
    yaxis.append(percentage3)
    pos = len(New_df3)/len(output)
    pos2 = pos * 100
    posaxis.append(pos2)

    nue = len(New_df4)/len(output)
    nue2 = nue * 100
    nueaxis.append(nue2)
else:
    yaxis.append(0)
    posaxis.append(0)
    nueaxis.append(0)

month = month + 1

xaxis = ["January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"]
st.balloons()
st.write(len(xaxis),len(yaxis),len(posaxis),len(nueaxis))

fig = px.line( x=xaxis, y=[yaxis,posaxis,nueaxis], title=heading, markers= True)
fig.update_layout(
    title="Sentimental Analysis By Month",
    xaxis_title="Month",
    yaxis_title="Sentiment",

    legend_title="Sentiment By Month",
    font=dict(
        family="Courier New, monospace",
        size=18,
        color="RebeccaPurple"
    )
)
custom_legend_name(["Negative Sentiment", "Positive Sentiment", "Neutral Sentiment"])
st.plotly_chart(fig)
st.write(xaxis)
st.write("Negative",yaxis)

st.write("Positive",posaxis)
st.write("Neutral",posaxis)

```

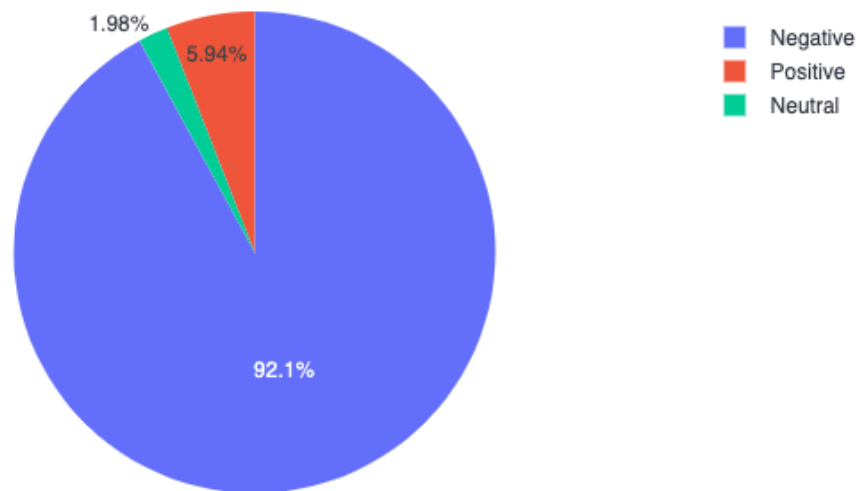
## 4.7 Most Negative, Positive, and Neutral Tweets

```
st.write("Most Negative Tweet")
New_df = output.loc[output["Negative"] == output["Negative"].max()]
st.write(New_df)
st.write("Most Positive Tweet")
New_df = output.loc[output["Positive"] == output["Positive"].max()]
st.write(New_df)
st.write("Most Neutral Tweet")
New_df = output.loc[output["Neutral"] == output["Neutral"].max()]
st.write(New_df)
```

## 5. Result

### 5.1 Pie chart

Sentimental Analysis Of Ukraine War Tweets

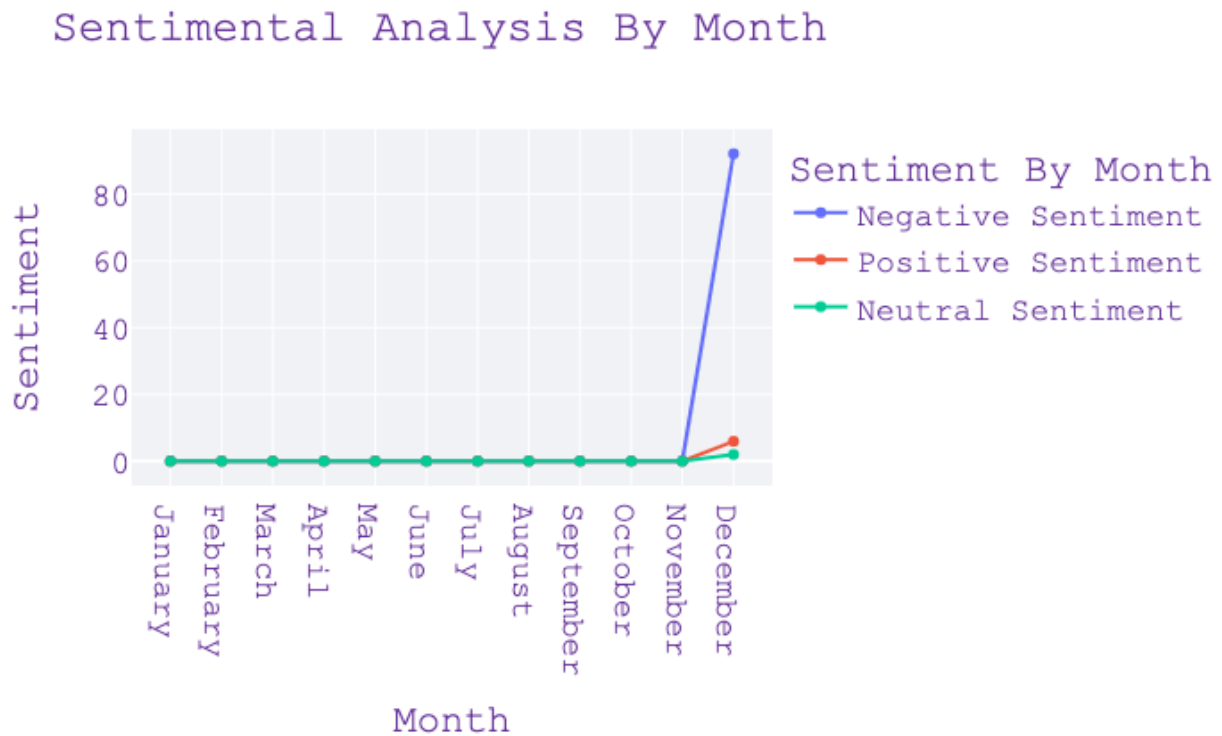


### 5.2 Pandas Dataframe

	Datetime	Tweet Id	Text	Username	Sentiment	Month	Positive	Negative	Neutral
0	2021-12-30T23:31:21+00:00	1476697447493689345	Bloomberg Global Survey p...	vtchakarova	Negative	12	7.2000	8.1000	84.7000
1	2021-12-30T18:31:14+00:00	1476621920414191616	Ukraine War VHS https://t.c...	Bochatov	Negative	12	0.0000	43.8000	56.2000
2	2021-12-30T14:54:49+00:00	1476567457565401088	@rec777777 @CalmPolite ...	nigethugrug	Negative	12	0.0000	14.0000	86.0000
3	2021-12-30T13:58:30+00:00	1476553285091971072	Update of the Russian milit...	PierreCrom	Negative	12	0.0000	10.6000	89.4000
4	2021-12-30T09:53:30+00:00	1476491626881105920	Four ceasefire violations by ...	zhuk2886	Negative	12	0.0000	25.8000	74.2000
5	2021-12-30T04:09:31+00:00	1476405059865001984	@Andre997F @QuakDr @re...	GuellichChr	Negative	12	0.0000	14.5000	85.5000
6	2021-12-30T01:19:52+00:00	1476362366652764162	@ulrichspeck Die Ukraine w...	WOLKEFN	Negative	12	0.0000	21.0000	79.0000
7	2021-12-30T00:51:10+00:00	1476355143822102528	#Ukraine war Hauptthema ...	aniho91	Negative	12	0.0000	9.1000	90.9000
8	2021-12-30T00:37:25+00:00	1476351683718356994	@caitoz 1) What is really goi...	Putin_Potatoes	Positive	12	5.0000	3.5000	91.5000
9	2021-12-29T18:16:35+00:00	1476255844031418372	@ArmyStanding @ReadeAl...	SolidarityFairy	Negative	12	5.2000	30.2000	64.6000
10	2021-12-29T17:55:49+00:00	1476250618742427649	war with China gotta protec...	Villgecrazylady	Negative	12	16.3000	28.1000	55.6000
11	2021-12-29T16:51:18+00:00	1476234381865873409	A Ukraine War and the End ...	glen_darcy	Negative	12	0.0000	28.1000	71.9000
12	2021-12-29T10:54:31+00:00	1476144594618400770	Strategic Ambiguity and the...	tkmeyer2020	Negative	12	0.0000	41.4000	58.6000
13	2021-12-29T06:44:48+00:00	1476081749520052226	@ShayKhatiri Will ukraine ...	prashGana	Negative	12	13.1000	18.9000	68.0000
14	2021-12-29T02:16:51+00:00	1476014320668401665	https://t.co/6BPHwf8nDv #...	kodiak_seattle	Negative	12	0.0000	48.3000	51.7000
15	2021-12-29T01:02:42+00:00	1475995658758807555	#Russia - #Ukraine war dru...	Shaya_7	Negative	12	3.9000	27.1000	69.0000
16	2021-12-29T00:27:34+00:00	1475986815744188419	Referencing infamous Sovie...	The_IntelHub	Negative	12	0.0000	29.0000	71.0000



## 5.3 Line Chart (Sentimental Analysis By Month)



## 5.4 Filtered Dataframe with most positive, negative, and neutral tweets

Most Negative Tweet

	Datetime	Tweet Id	Text	Username
47	2021-12-27T17:45:01+00:00	1475523124959465480	#Russia #Ukraine #war ht...	mark_melbi
55	2021-12-27T15:18:06+00:00	1475486153285550090	#Ukraine #war #ww3 http...	DylanRBLXC

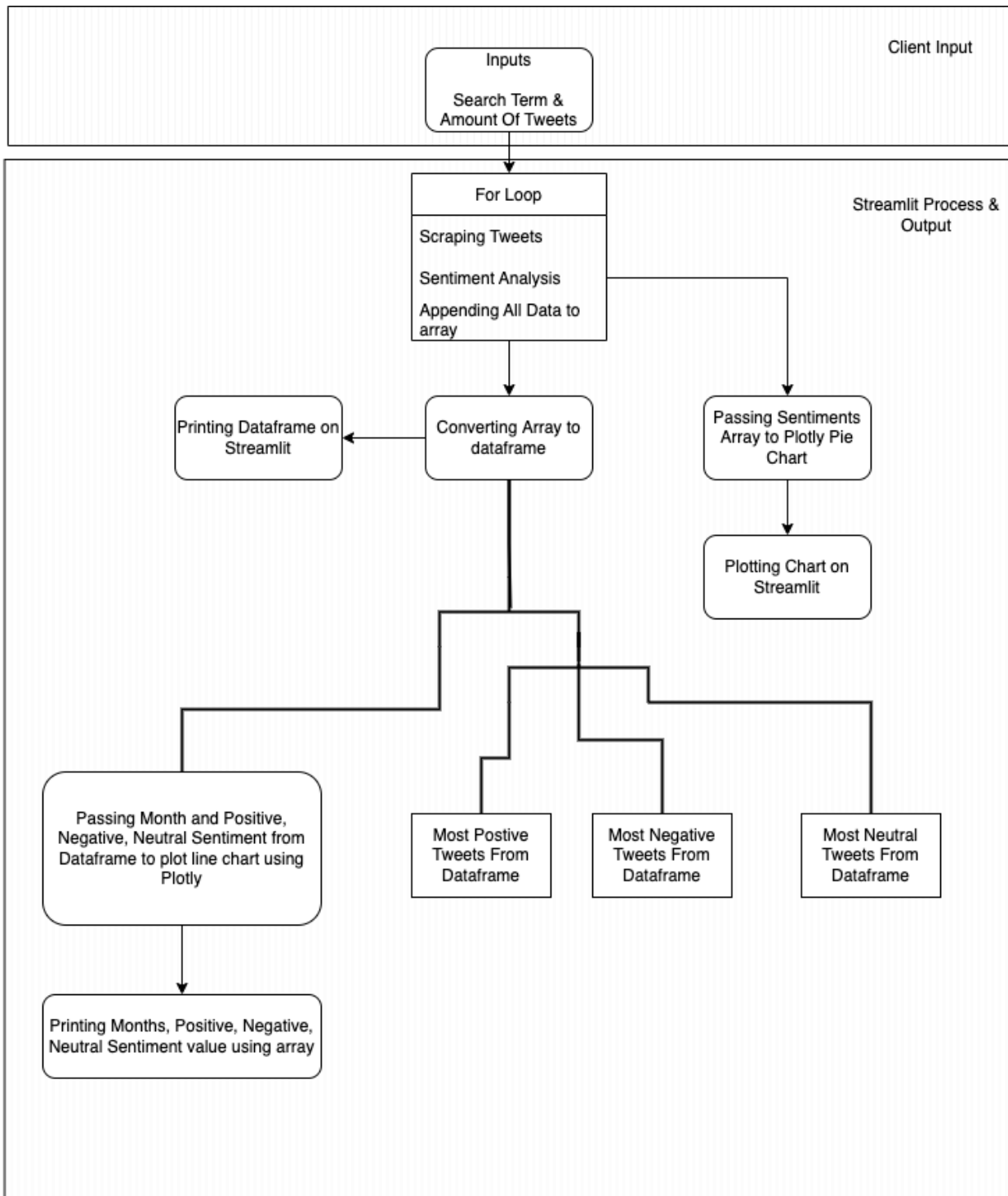
Most Positive Tweet

	Datetime	Tweet Id	Text	Username
88	2021-12-25T19:17:54+00:00	1474821722960449538	Please write or call your e...	tshtatland

Most Neutral Tweet

	Datetime	Tweet Id	Text	Username
21	2021-12-28T15:25:08+00:00	1475850309775900678	#UKRAINEWAR #USAID2U...	yourbrother
75	2021-12-26T14:31:31+00:00	1475112042545745921	US Senators Sound Ukrai...	KolozegInfo

## 6. Architecture Diagram



## 7. Conclusion

In the project tweet sentiment analyser was able to show the emotional change and the historical emotional change of an entire trend. This change can show the entire mood of an trend and can pinpoint the answer to various problems.

Another noticeable thing was that this project is able to efficeitly scrape tweets in very less time without any limitations which was never provided before in other sentiment analysers.

## 8. References

1. **Rashi Desai - Data Analyst**  
<https://medium.com/dataseries/how-to-scrape-millions-of-tweets-using-snsrape-195ee3594721>
2. **Aditya Beri - Blogger at towardsdatascience**  
<https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>