NAME – SAARA ANAND

REG NO – 21BCE8156

ARTIFICIAL INTELLIGENCE PROJECT:

TOPIC – SENTIMENT ANALYSIS AND EMOJIFY WEB APP USING NATURAL LANGUAGE PROCESSING IN PYTHON

What is Sentiment Analysis?

Sentiment analysis is contextual text mining that recognises and extracts subjective information from source material. It assists businesses in understanding the social sentiment of their brands, products, and services while keeping an eye on online discussions.

The process of identifying positive or negative sentiment in text is known as sentiment analysis. Businesses frequently employ it to analyse social media data for sentiment, evaluate brand repute, and comprehend clientele.

What is Natural Language Processing?

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written. It is an integral component of Artificial Intelligence(AI).

Objective of this project:

This project analyses the hidden emotion behind a given text using Natural Language Processing. It would then classify the input text as positive, negative or neutral. It would also be analysing the input text and would give the output as the most suitable emoji such as a smiling, sad or an expressionless face.

Compiler used: VS CODE

Language used: Python

Modules used:

Streamlit: Streamlit is an open-source app framework in Python language. It helps us create web apps for data science and machine learning in a short time.

Python Libraries used:

Textblob: TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

NLTK: NLTK is a library which gives an easy access to a lot of lexical resources and allows users to work with categorization, classification and many other tasks.

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.

Beautiful Soup: Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work. This is referred to as Web Scraping.

Emoji: Emoji Module is a Python package that allows us to use and print emojis through a Python program, and we can even use this module to use emojis inside an application we are creating using Python. Emoji library in Python is the most common through which we can print Emojis using a Python program, and we should note that Emoji Module is not an in-built module in Python. Using Emoji Module is very simple, and we just have to remember the name of the emoji we want to print in the output.

SOURCE CODE:

import streamlit as st

# NLP Pkgs

from textblob import TextBlob

import pandas as pd

# Emoji

import emoji

# Web Scraping Pkg

from bs4 import BeautifulSoup

from urllib.request import urlopen

# Fetch Text From Url

@st.cache

def get\_text(raw\_url):

    page = urlopen(raw\_url)

    soup = BeautifulSoup(page)

    fetched\_text = ' '.join(map(lambda p:p.text,soup.find\_all('p')))

    return fetched\_text

def main():

    """Sentiment Analysis Emoji App """

    st.title("Sentiment Analysis Emoji App")

    activities = ["Sentiment","Text Analysis on URL","About"]

    choice = st.sidebar.selectbox("Choice",activities)

    if choice == 'Sentiment':

        st.subheader("Sentiment Analysis")

        st.write(emoji.emojize('Everyone :red\_heart: Streamlit '))

        raw\_text = st.text\_area("Enter Your Text","Type Here")

        if st.button("Analyze"):

            blob = TextBlob(raw\_text)

            result = blob.sentiment.polarity

            if result > 0.0:

                custom\_emoji = ':smile:'

                st.write(emoji.emojize(custom\_emoji))

            elif result < 0.0:

                custom\_emoji = ':disappointed:'

                st.write(emoji.emojize(custom\_emoji))

            else:

                st.write(emoji.emojize(':expressionless:'))

            st.info("Polarity Score is:: {}".format(result))

    if choice == 'Text Analysis on URL':

        st.subheader("Analysis on Text From URL")

        raw\_url = st.text\_input("Enter URL Here","Type here")

        text\_preview\_length = st.slider("Length to Preview",50,100)

        if st.button("Analyze"):

            if raw\_url != "Type here":

                result = get\_text(raw\_url)

                blob = TextBlob(result)

                len\_of\_full\_text = len(result)

                len\_of\_short\_text = round(len(result)/text\_preview\_length)

                st.success("Length of Full Text::{}".format(len\_of\_full\_text))

                st.success("Length of Short Text::{}".format(len\_of\_short\_text))

                st.info(result[:len\_of\_short\_text])

                c\_sentences = [ sent for sent in blob.sentences ]

                c\_sentiment = [sent.sentiment.polarity for sent in blob.sentences]

                new\_df = pd.DataFrame(zip(c\_sentences,c\_sentiment),columns=['Sentence','Sentiment'])

                st.dataframe(new\_df)

    if choice == 'About':

        st.subheader("About:Sentiment Analysis Emoji Web App leveraging NLP")

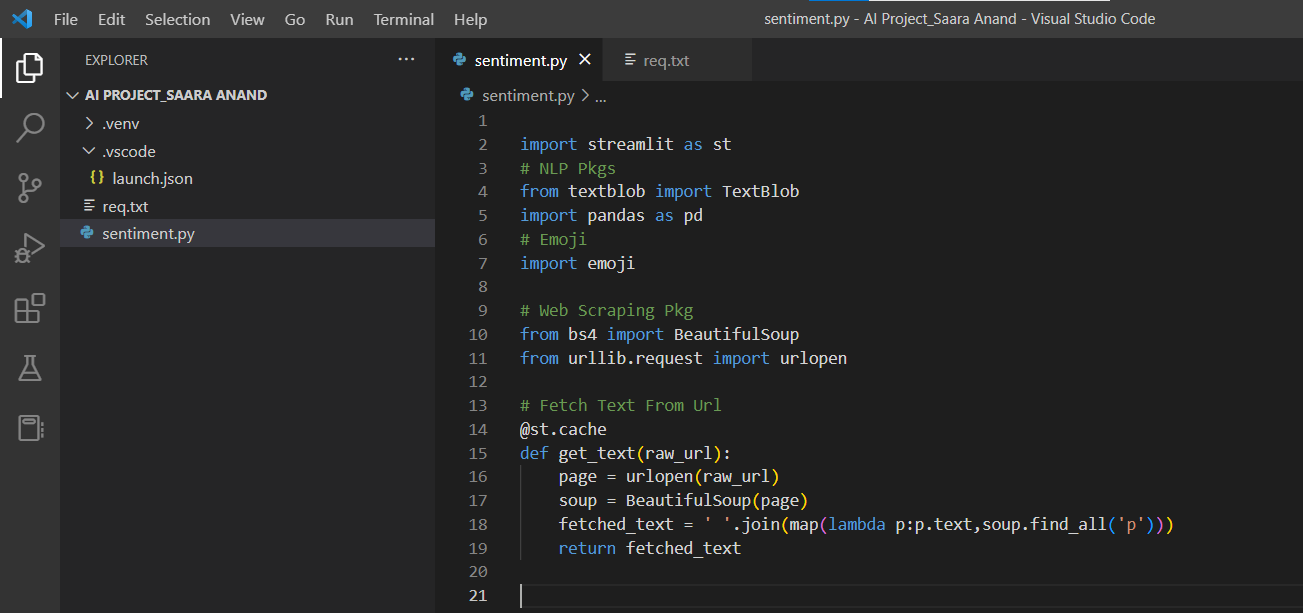
        st.info("Built with Streamlit,Textblob and Emoji in Python")

        st.text("Done by: Saara Anand (21BCE8156)")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

CODE SCREENSHOTS-



Text

Description automatically generated

Text

Description automatically generated

Req.txt-

Graphical user interface, application

Description automatically generated

Terminal-

Text

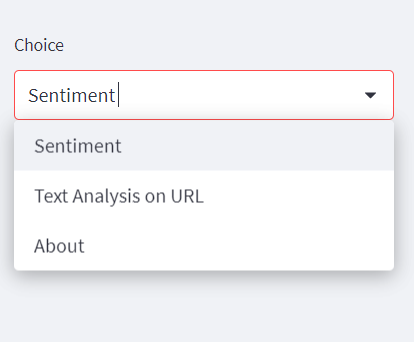
Description automatically generated

SENTIMENT ANALYSIS EMOJI APP:

Graphical user interface, text, application

Description automatically generated

Features provided:



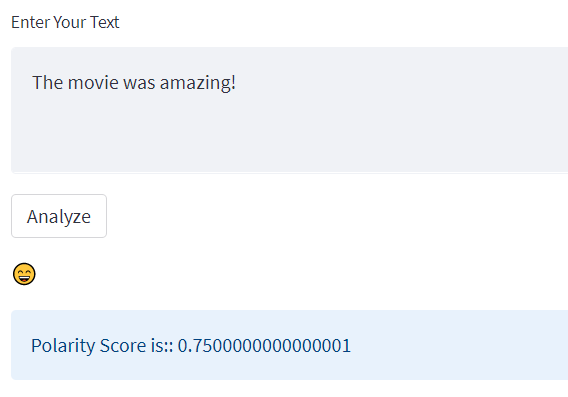
Sentiment: This returns whether the given input text is positive, negative or neutral.

Text Analysis on URL: This is used to scrape the information from a given URL and analyse it.

About: This returns the polarity score of the text along with the most suitable emoji.

Examples:

Positive text-



Neutral text-

Graphical user interface, text, application

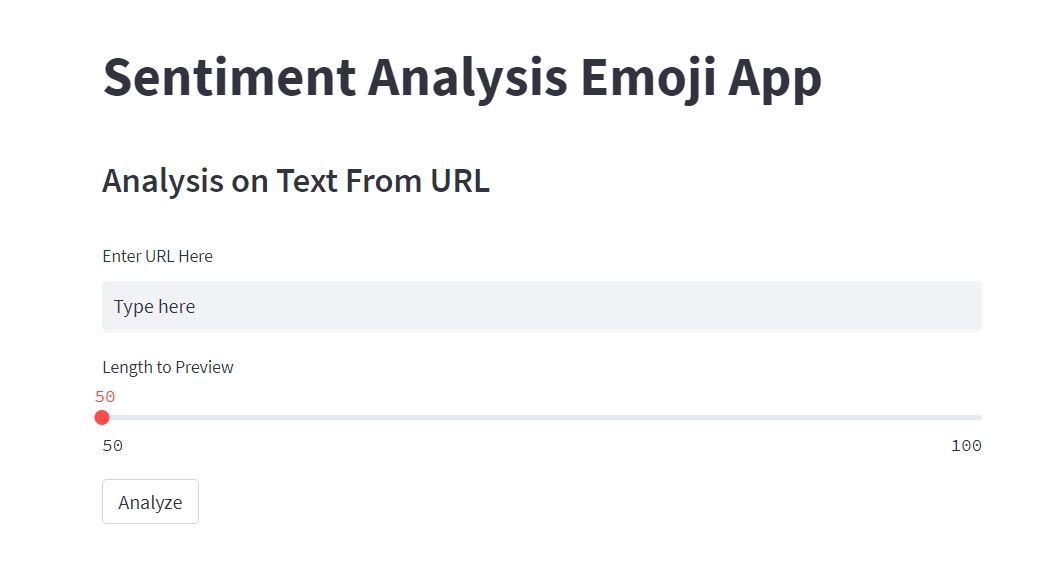
Description automatically generated

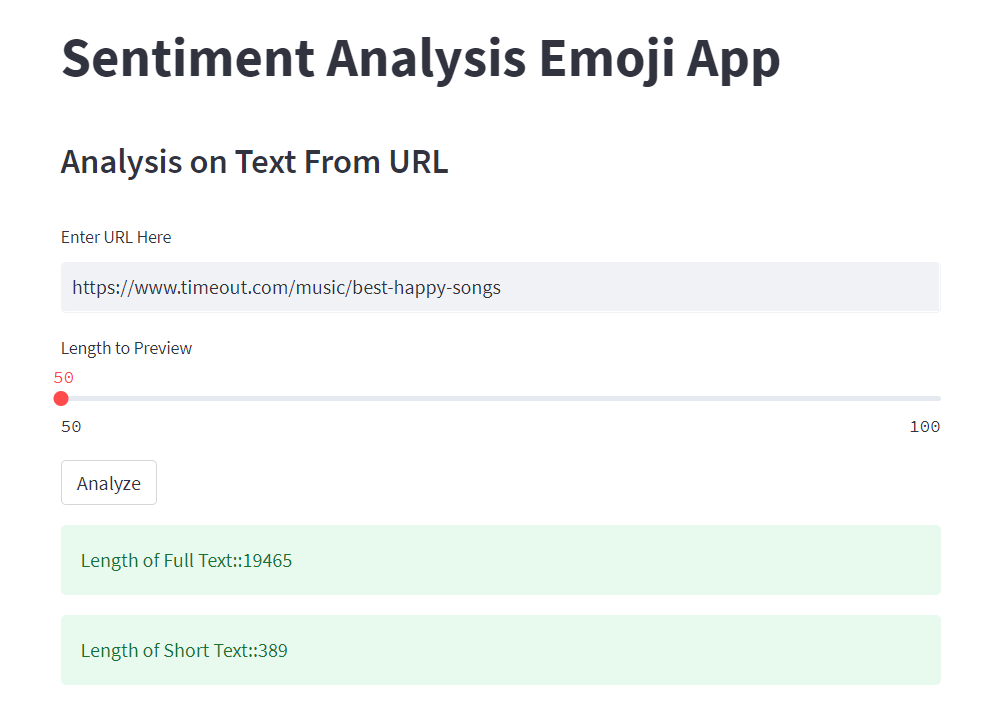
Negative text-

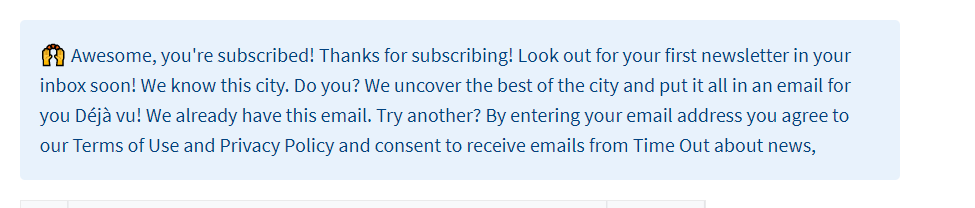
Graphical user interface, text, application

Description automatically generated

ANALYSIS ON TEXT FROM URL:







It returns the sentiment score of all the sentences appearing in the URL, as follows:

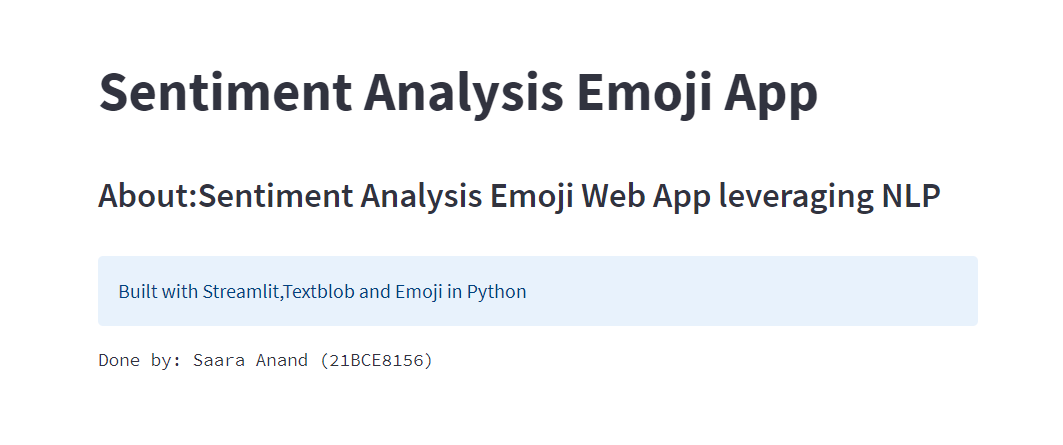
Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

ABOUT-



APPLICATIONS/REAL TIME USES OF THE APP-



### Social media monitoring

Social media posts often contain some of the most honest opinions about your products, services, and businesses because they’re unsolicited.

With the help of sentiment analysis software, you can wade through all that data in minutes, to analyse individual emotions and overall public sentiment on every social platform.

### Customer support

Customer support management presents many challenges due to the sheer number of requests, varied topics, and diverse branches within a company – not to mention the urgency of any given request.

Sentiment analysis with NLP reads regular human language for meaning, emotion, tone, and more, to understand customer requests, just as a person would.

### Product analysis

Find out what the public is saying about a new product right after launch, or analyse years of feedback you may have never seen.

Sentimental Analysis using Natural Language Processing has been proposed to be applied to the collected data **to predict user information such as location, mood, activity, mental status, depression, anxiety, stress and soon**

By using this rich source of data and information, can efficient model which provides report of person's **depression** symptoms will be designed.

 Emotional language and linguistic features used in **social media posts** have been proven to indicate feelings that characterize major **depression**.

When any user **uses** an **emoticon**(**emoji**) in electronic communication then they are effectively tagging their **text** with an emotional state.

Graphical user interface, application

Description automatically generated

Emojis are often used to convey emotions, so by looking at the emojis used in a piece of text, it can be easier to understand the sentiment behind the words. In addition, emojis can also help to capture the sentiment that traditional **sentiment analysis** techniques might miss.

For example, if a person uses an emoji that conveys sarcasm, this might not be picked up by conventional methods but would be captured by emoji analysis. As a result, using emojis in **sentiment analysis** can help to provide a more accurate and complete picture of the sentiment behind a piece of text.

IMPROVEMENTS THAT CAN BE MADE TO INCREASE ITS SCOPE IN THE FUTURE -

The app gives the appropriate polarity score and emoji for the text, however it tends to focus on the literal meaning of sentences and doesn’t properly detect sarcasm and idioms.

It focusses mainly on the key words that are being used and assigns a particular set of words to a particular emotion.

This model especially focusses on the popular human emotions and is trained in such a way to understand them.

The further upgradation can be done on the part of it analysing the exact emotion behind the given text without just considering the syntactical structure, but also taking into account of the semantics.

This can further be used with the Bert Model or GPT3 NLP Models to increase its scope in future.

CONCLUSION-

Thus this Sentiment Analysis and Emojify Model is deployed on web, using Streamlit and can be used for many such applications.

It uses the concept of inferring emotions from text by training the model/machine about syntax and semantics.

-------------------------------------X-------------------------------------------

THANK YOU!