PROJECT DOCUMENTATION

PROJECT TITLE:

TEXT ANALYSIS USING PYTHON

GROUP MEMBERS:

PRANAV JOSHI

SHREYAS DHARASHIVKAR

SWAROOP BANKAR

CHAPTER1 - INTRODUCTION

PROJECT SCOPE:

The project scope is to build a web application for text analysis. The application provides various functionalities to analyze text, including spam or ham detection, sentiment analysis, stress detection, hate and offensive content detection, and sarcasm detection.

The scope of the project includes the following:

Data Preprocessing: There are certain functions which includes functions to preprocess the input text, such as converting it to lowercase, tokenizing, removing stopwords and punctuation, and applying stemming. These preprocessing steps help in transforming the raw text into a suitable format for analysis.

Model Training: It loads pre-existing datasets for each analysis task and trains machine learning models using these datasets. It utilizes different models for different tasks, such as logistic regression, decision tree regressor, and random forest classifier. The models are trained using the transformed text data obtained after preprocessing.

Text Analysis Functionality: This also implements each text analysis functionality, allowing users to input text and receive predictions or analysis results based on the trained models. The application provides a user-friendly interface using Streamlit, where users can interact with the different functionalities and view the results.

User Interface: This project includes a Streamlit-based user interface that allows users to navigate between different text analysis options, input their text for analysis, and view the corresponding predictions or analysis results. It also provides additional information and images to explain the concepts and applications of each analysis task.

Deployment: The code is designed to be deployed as a web application, allowing users to access and utilize the text analysis functionalities through a web browser.

The project scope focuses on text analysis tasks and aims to provide users with a convenient way to perform various text analysis operations through a user-friendly web interface.

This project may require additional modifications and adaptations based on specific requirements and resources.

Technology used:

The technology used in this project is Python programming language. The backend of the project is developed using various machine learning models while the frontend of the project is built using streamlit.

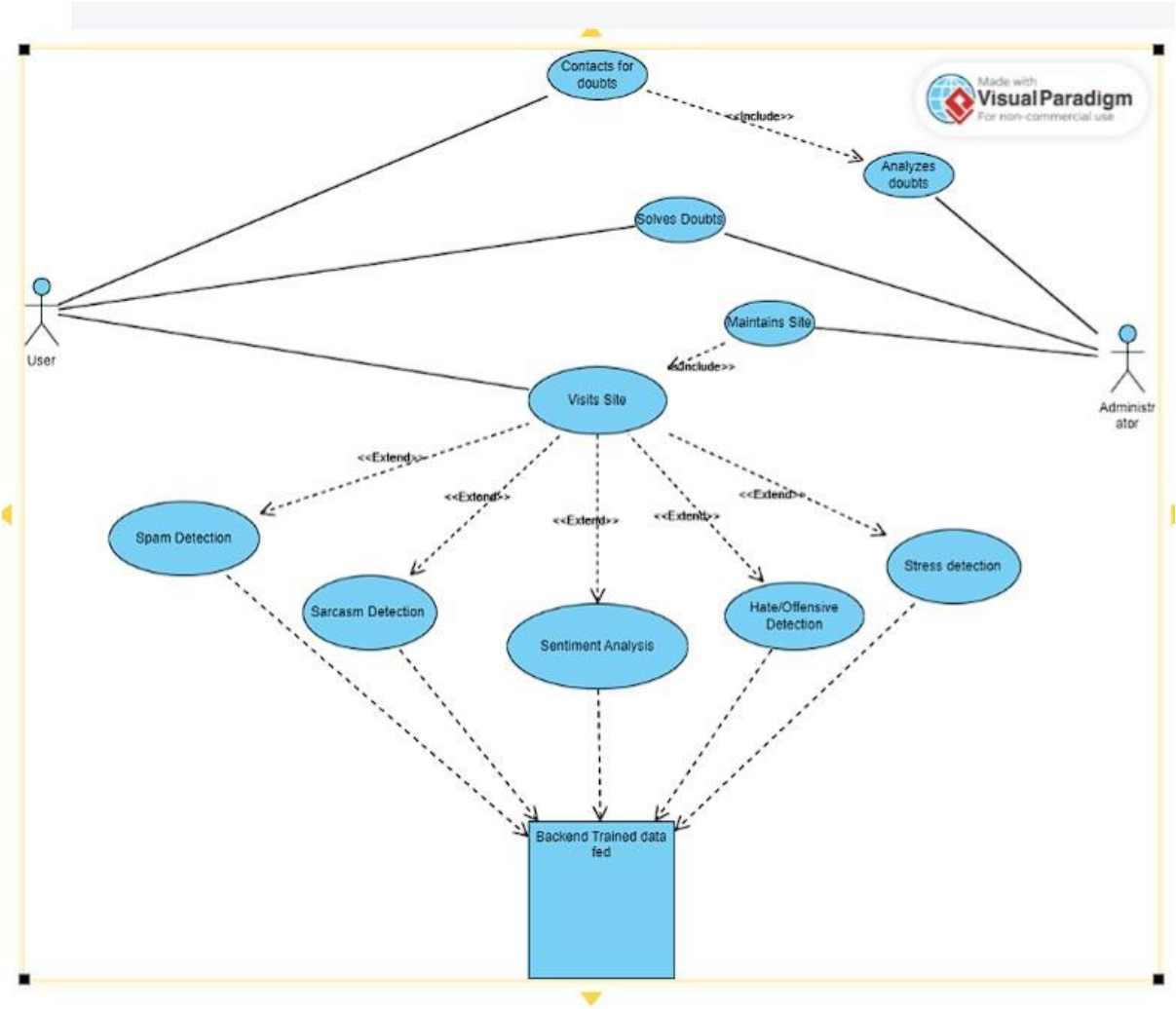
Streamlit – A python based framework which is used to build efficient web applications related data scraping and machine learning.

Uses of this project:

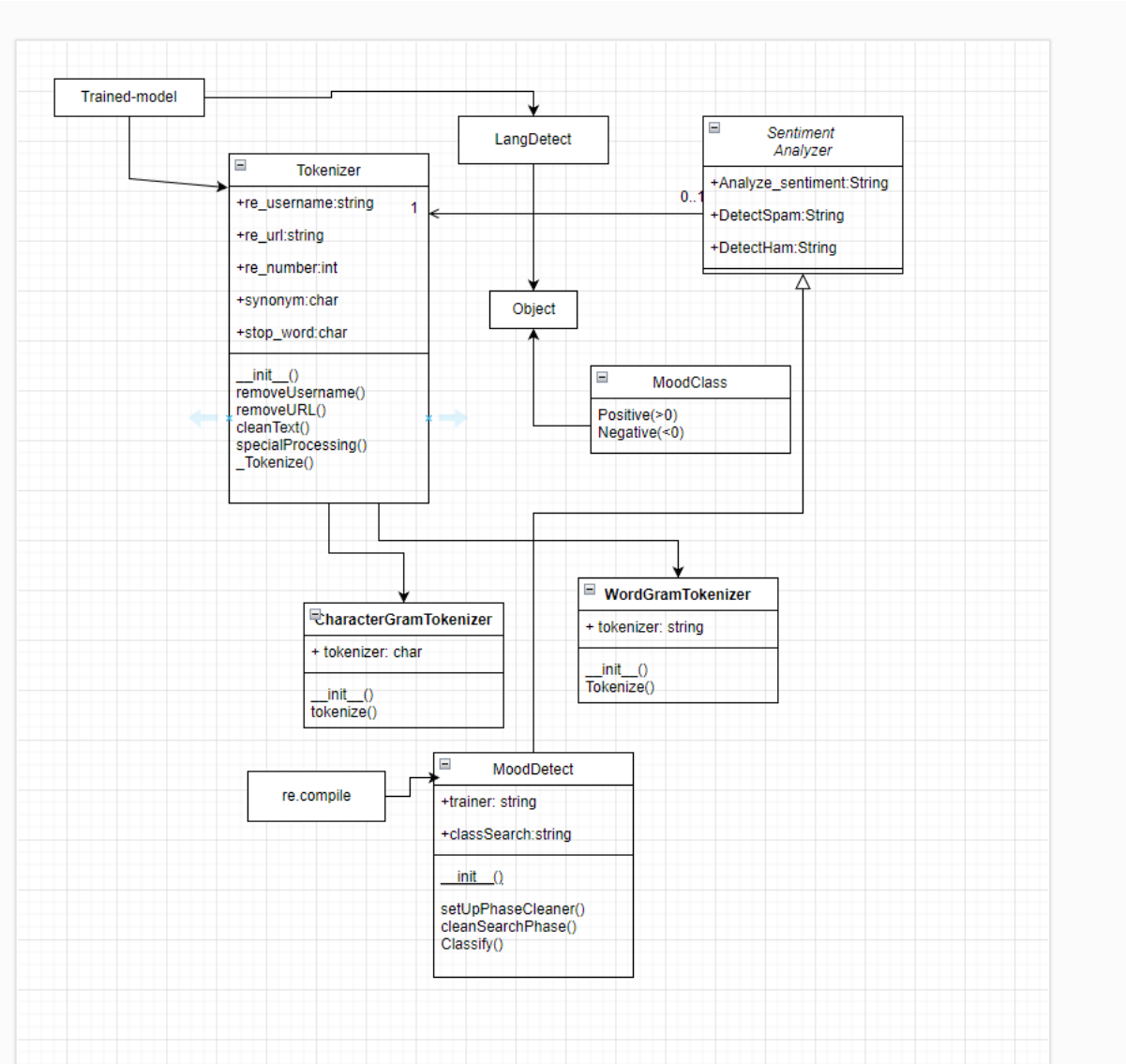
This project provides tools and functionalities to extract meaningful insights from textual data, enabling automated analysis and decision-making in various domains, including marketing, customer service, mental health, and content moderation. It can save time and effort by automating the process of analyzing large volumes of text and providing valuable information for businesses and individuals which can be used for taking correct decisions in the industry.

CHAPTER -2 DESIGN

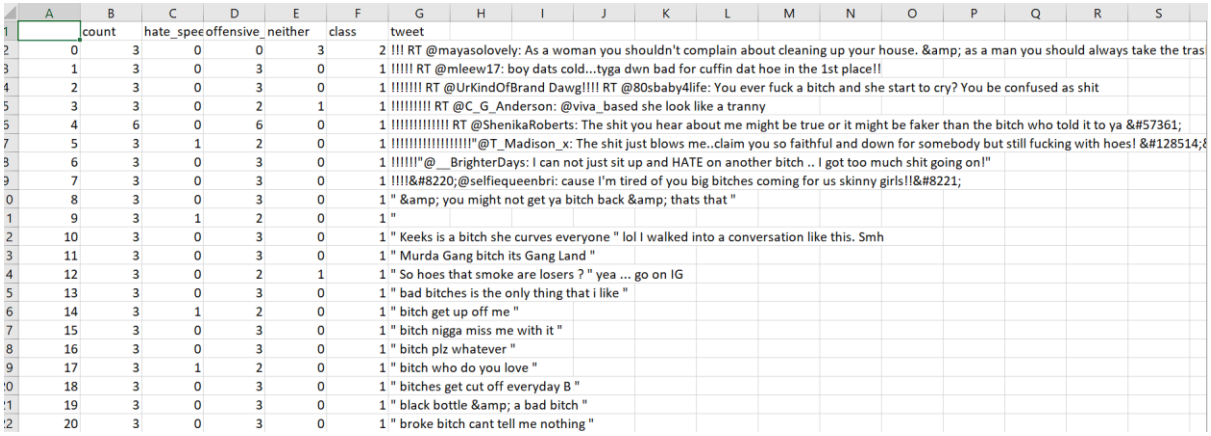
USE CASE DIAGRAM:



CLASS DIAGRAM:



DATABASE DESIGN:

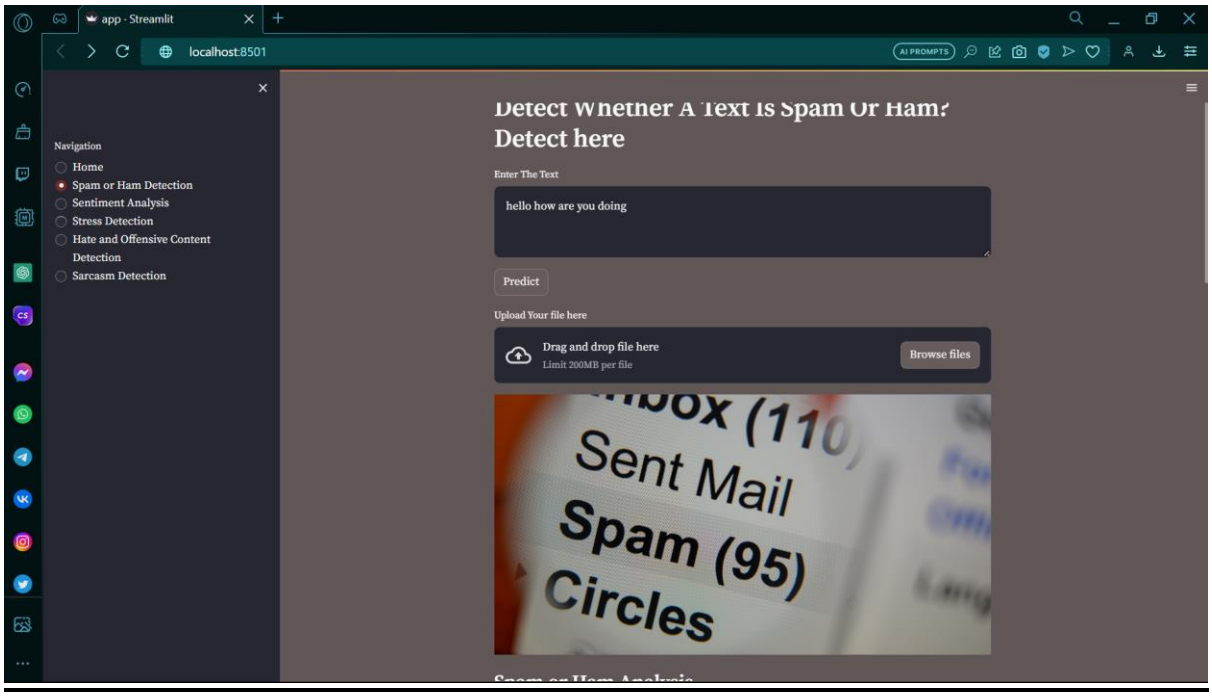


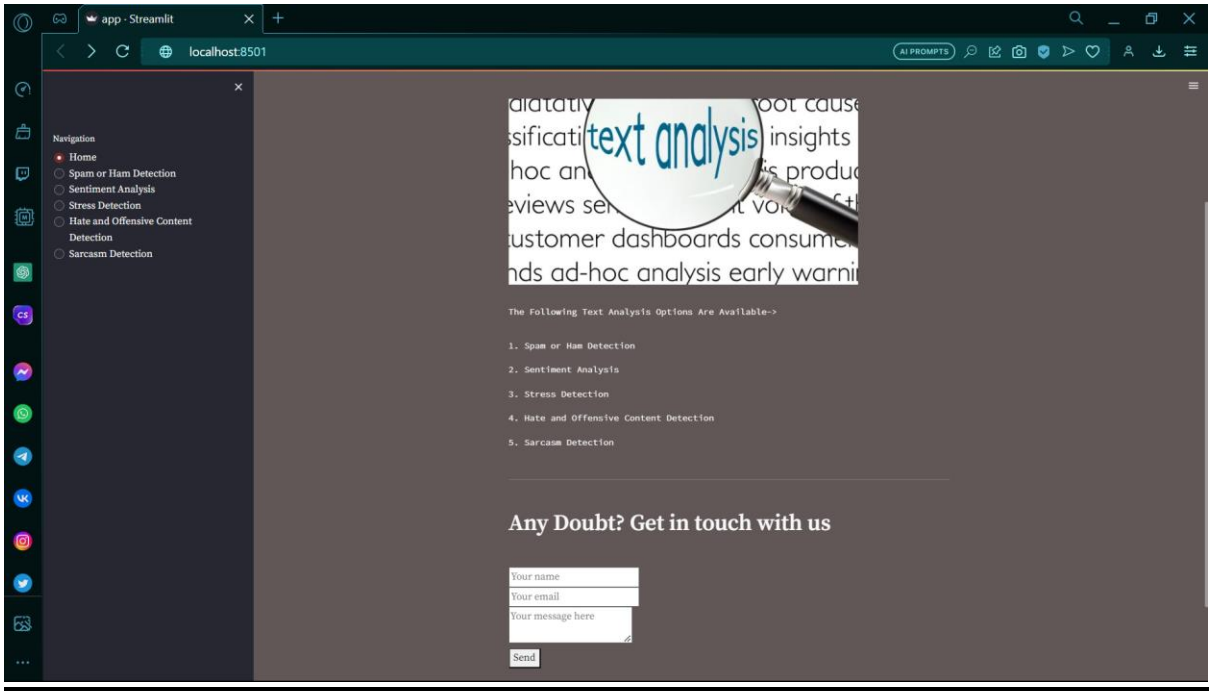
		count	hate_spee	offensive_	neither	class	tweet
2	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the tras
3	1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
4	2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
5	3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
6	4	6	0	6	0	1	!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 
7	5	3	1	2	0	1	!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂8
8	6	3	0	3	0	1	!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"
9	7	3	0	3	0	1	!!!!“@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!”
10	8	3	0	3	0	1	" & you might not get ya bitch back & thats that "
11	9	3	1	2	0	1	"
12	10	3	0	3	0	1	" Keeks is a bitch she curves everyone " lol I walked into a conversation like this. Smh
13	11	3	0	3	0	1	" Murda Gang bitch its Gang Land "
14	12	3	0	2	1	1	" So hoes that smoke are losers ? " yea ... go on IG
15	13	3	0	3	0	1	" bad bitches is the only thing that i like "
16	14	3	1	2	0	1	" bitch get up off me "
17	15	3	0	3	0	1	" bitch nigga miss me with it "
18	16	3	0	3	0	1	" bitch plz whatever "
19	17	3	1	2	0	1	" bitch who do you love "
20	18	3	0	3	0	1	" bitches get cut off everyday B "
21	19	3	0	3	0	1	" black bottle & a bad bitch "
22	20	3	0	3	0	1	" broke bitch cant tell me nothing "

	A	B
1	Tweet	Label
2	I love findi	0
3	USER :/ by	0
4	USER I abs	0
5	We're hiri	0
6	Every time	0
7	USER it's n	0
8	First dose	0
9	USER reply	0
10	God will su	0
11	If you app	0
12	USER why	0
13	Traveling t	0
14	I keep wak	0
15	USER than	0
16	The girl ne	0
17	USER had	0
18	USER Only	0
19	Time to st	0
20	Slapen #tv	0
21	Swag is jus	0
22	In a bit of	0

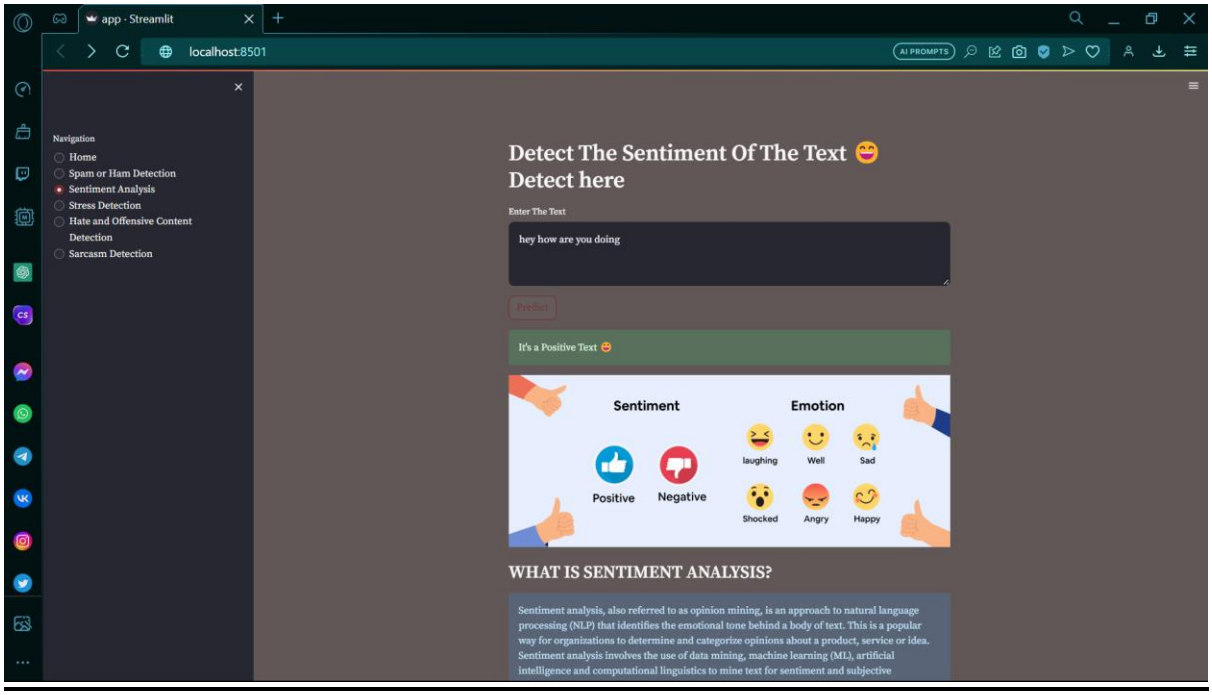
CHAPTER -3 -USER INTERFACE

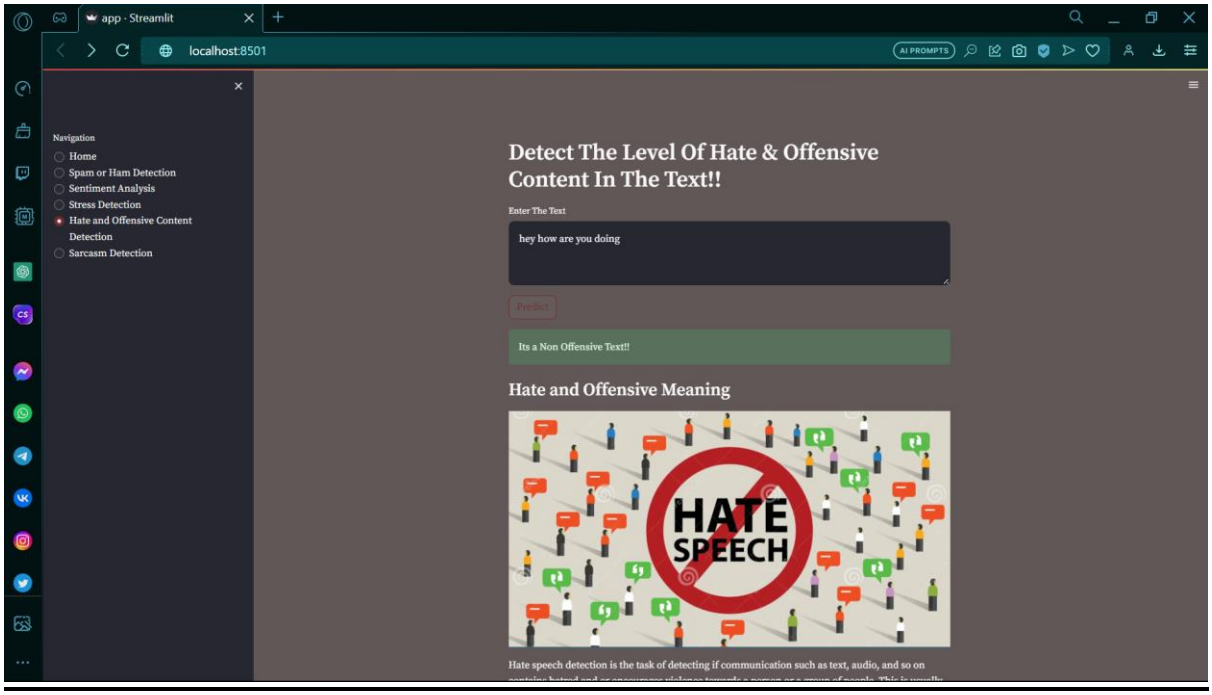
INPUT DESIGN:

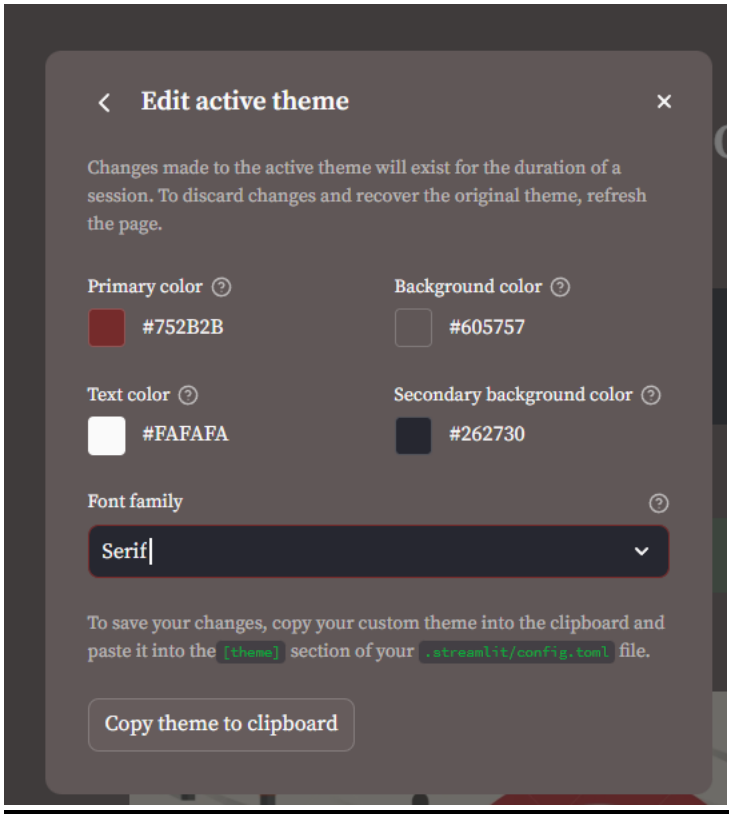


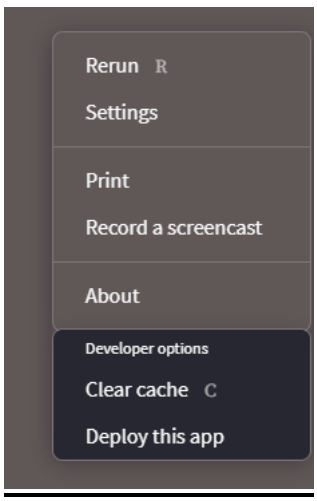


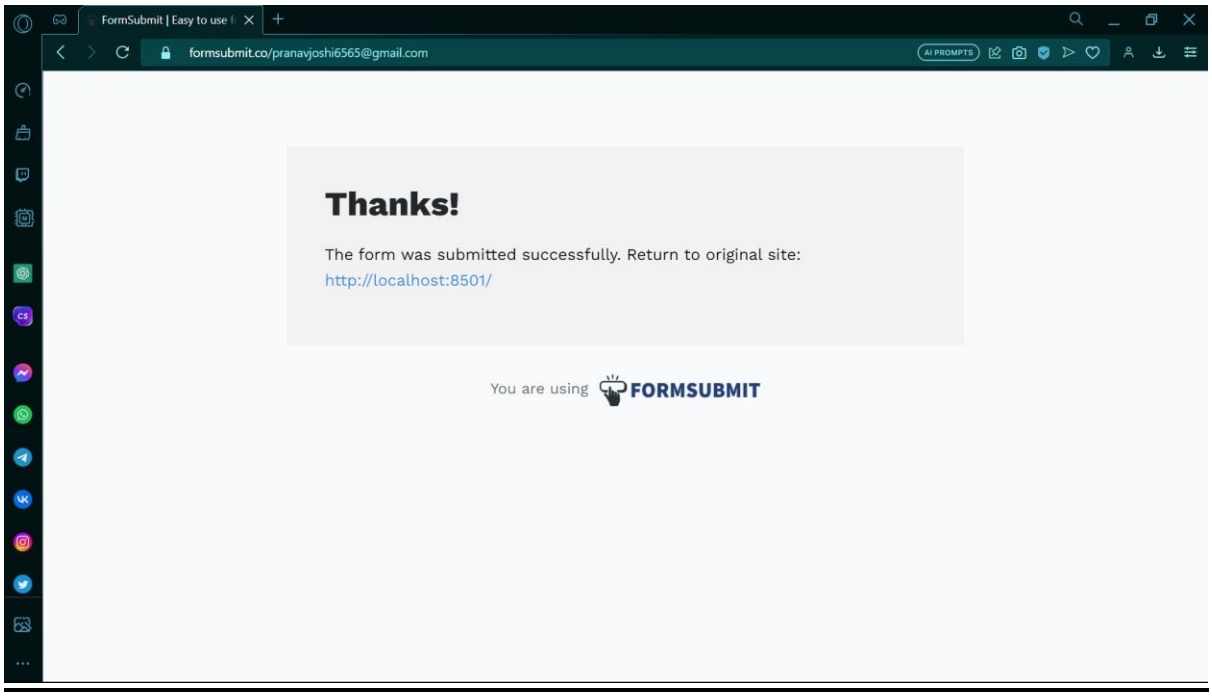
Output Design:

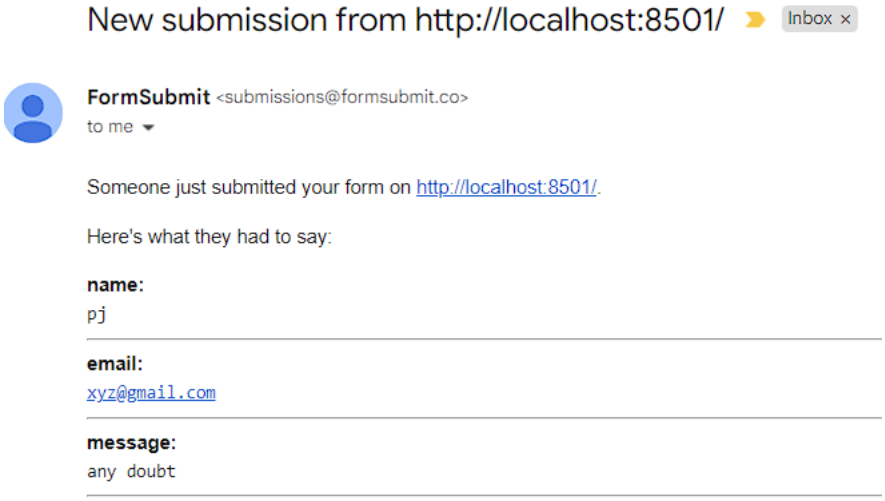





Edit active theme
Changes made to the active theme will exist for the duration of a session. To discard changes and recover the original theme, refresh the page.
Primary color
#752B2B
Background color
#605757
Text color
#FAFAFA
Secondary background color
#262730
Font family
Serif
To save your changes, copy your custom theme into the clipboard and paste it into the [theme] section of your .streamlit/config.toml file.
Copy theme to clipboard

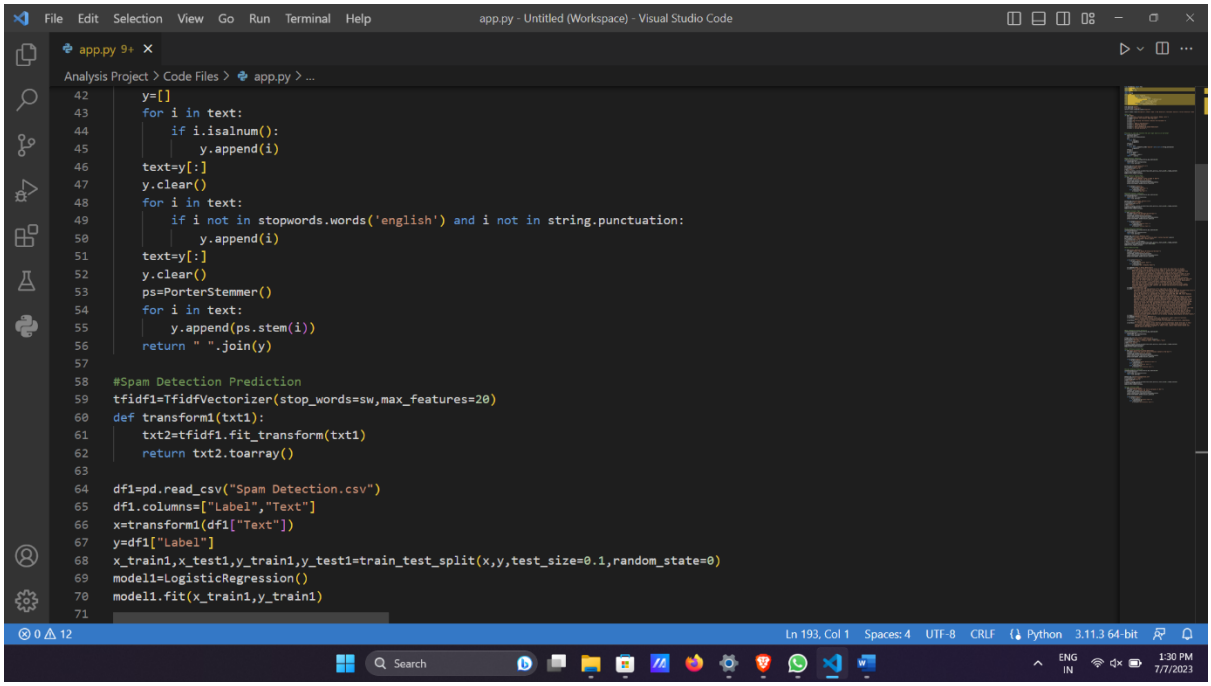

Rerun R
Settings
Print
Record a screencast
About
Developer options
Clear cache C
Deploy this app



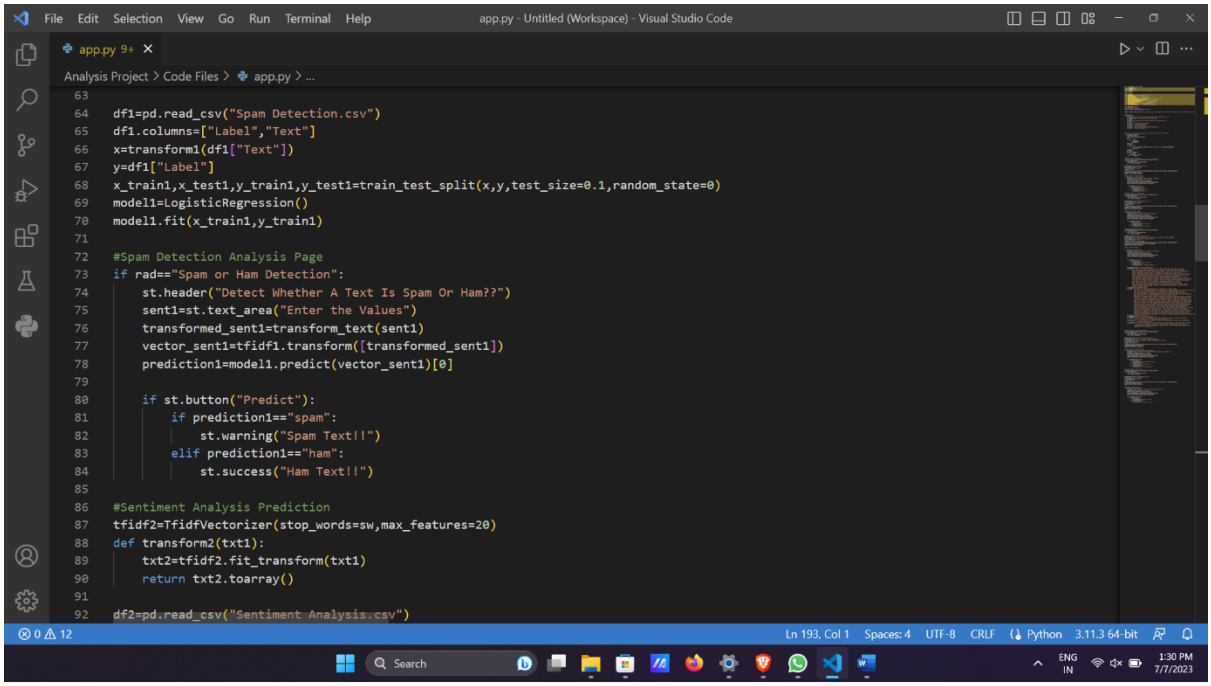


Users queries will be solved after getting their doubts in our email.

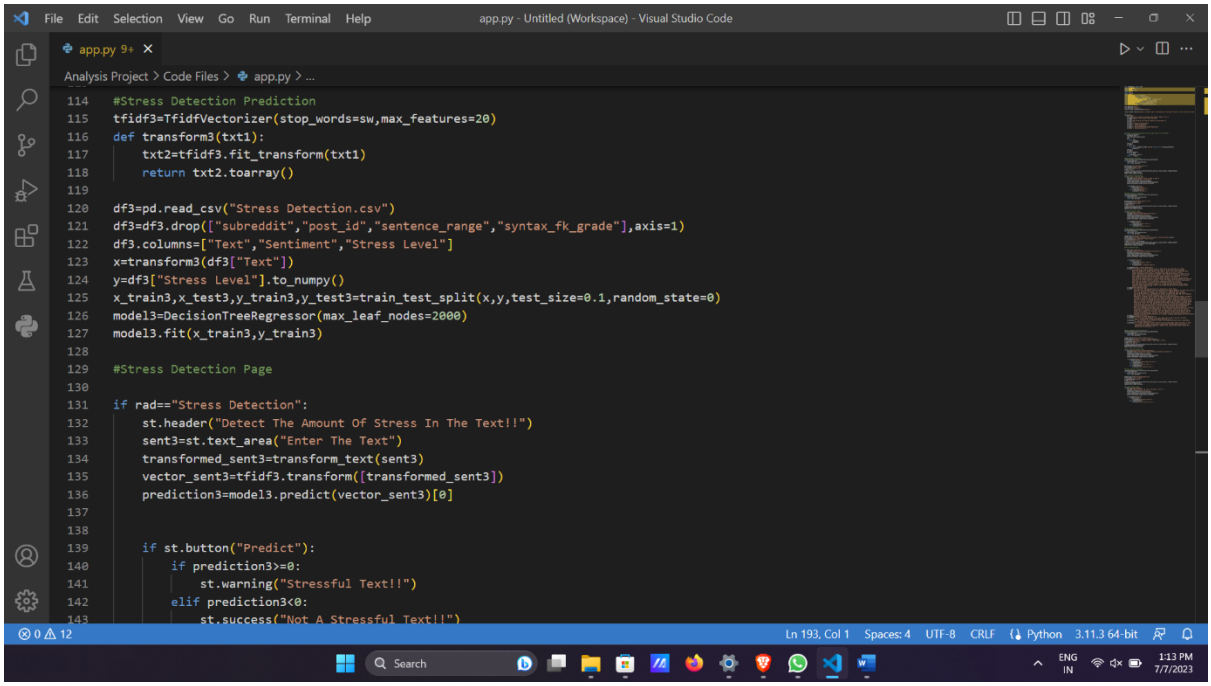
CHAPTER-4-CODE

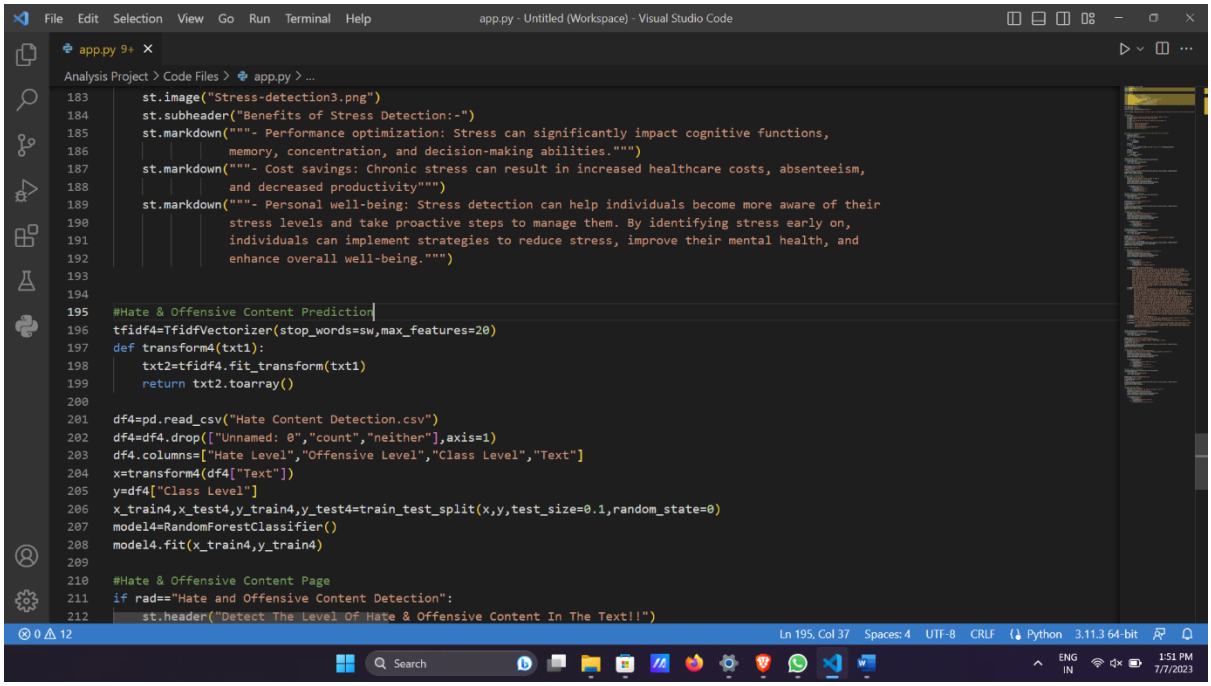


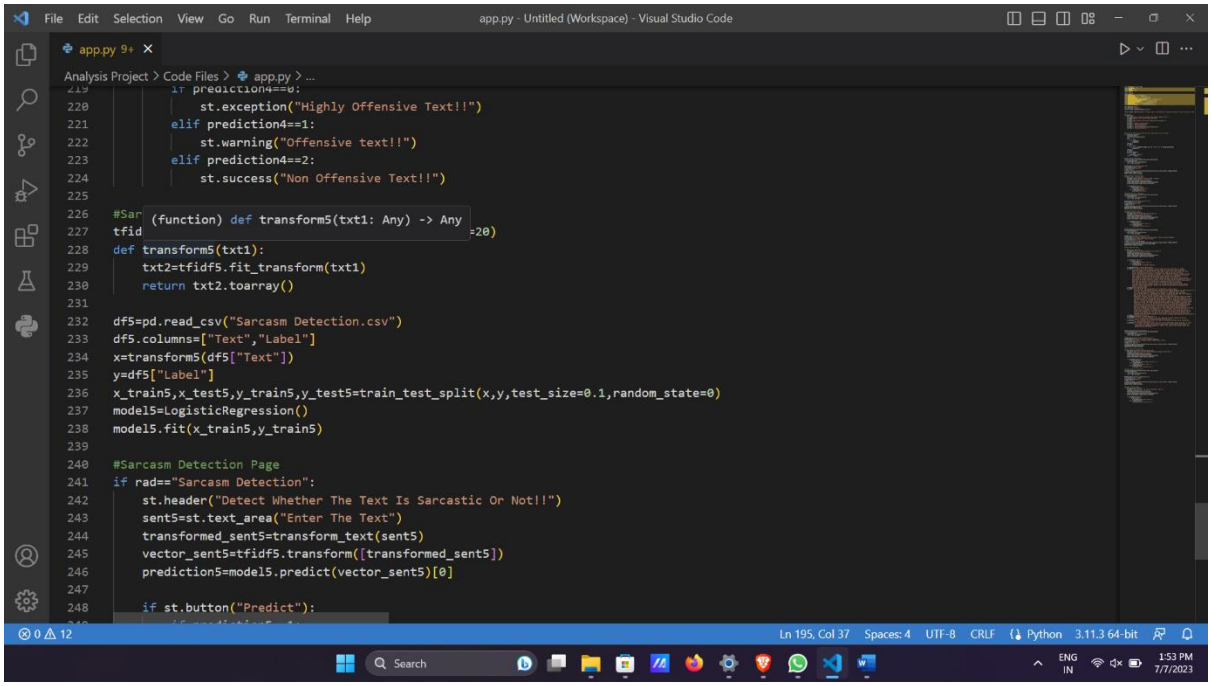
```
    y=[]
    for i in text:
        if i.isalnum():
            y.append(i)
    text=y[:]
    y.clear()
    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)
    text=y[:]
    y.clear()
    ps=PorterStemmer()
    for i in text:
        y.append(ps.stem(i))
    return " ".join(y)

#Spam Detection Prediction
tfidf1=TfidfVectorizer(stop_words=sw,max_features=20)
def transform1(txt1):
    txt2=tfidf1.fit_transform(txt1)
    return txt2.toarray()

df1=pd.read_csv("Spam Detection.csv")
df1.columns=["Label","Text"]
x=transform1(df1["Text"])
y=df1["Label"]
x_train1,x_test1,y_train1,y_test1=train_test_split(x,y,test_size=0.1,random_state=0)
model1=LogisticRegression()
model1.fit(x_train1,y_train1)
```

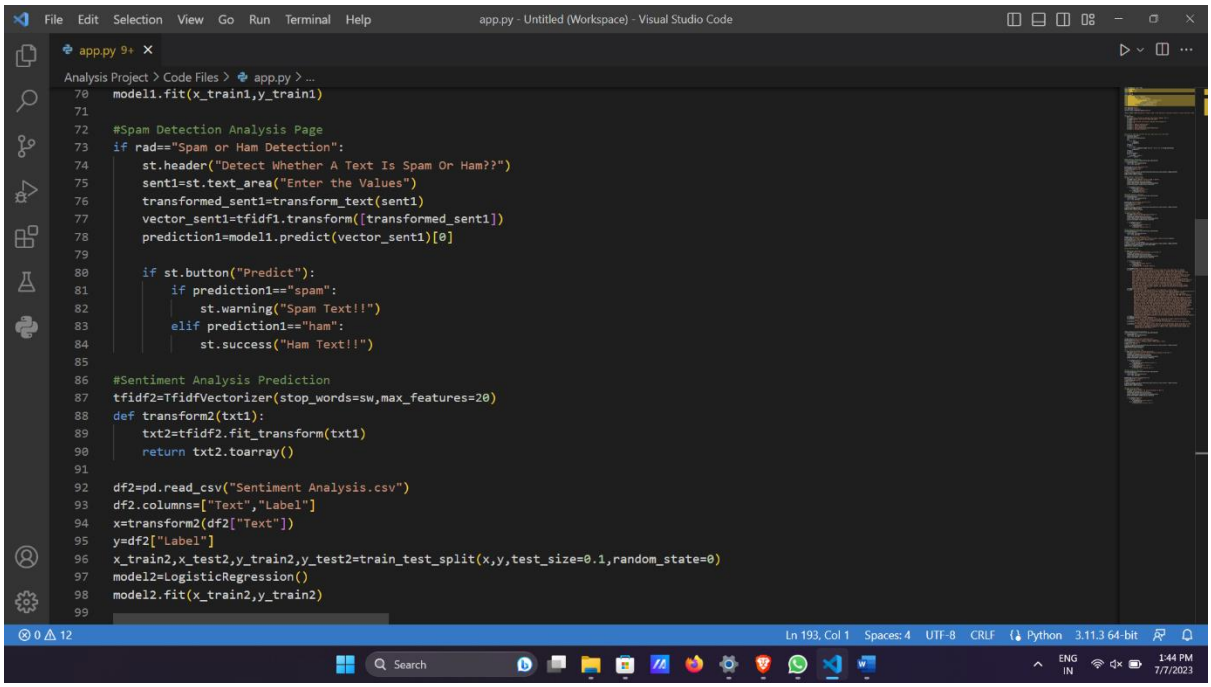


```
df1=pd.read_csv("Spam Detection.csv")
df1.columns=["Label","Text"]
x=transform1(df1["Text"])
y=df1["Label"]
x_train1,x_test1,y_train1,y_test1=train_test_split(x,y,test_size=0.1,random_state=0)
model1=LogisticRegression()
model1.fit(x_train1,y_train1)

#Spam Detection Analysis Page
if rad=="Spam or Ham Detection":
    st.header("Detect Whether A Text Is Spam Or Ham??")
    sent1=st.text_area("Enter the Values")
    transformed_sent1=transform_text(sent1)
    vector_sent1=tfidf1.transform([transformed_sent1])
    prediction1=model1.predict(vector_sent1)[0]

    if st.button("Predict"):
        if prediction1=="spam":
            st.warning("Spam Text!!")
        elif prediction1=="ham":
            st.success("Ham Text!!")

#Sentiment Analysis Prediction
tfidf2=TfidfVectorizer(stop_words=sw,max_features=20)
def transform2(txt1):
    txt2=tfidf2.fit_transform(txt1)
    return txt2.toarray()

df2=pd.read_csv("Sentiment Analysis.csv")
```


File Edit Selection View Go Run Terminal Help
app.py - Untitled (Workspace) - Visual Studio Code
app.py 9+
Analysis Project > Code Files > app.py > ...
#Stress Detection Prediction
tfidf3=TfidfVectorizer(stop_words=sw,max_features=20)
def transform3(txt1):
txt2=tfidf3.fit_transform(txt1)
return txt2.toarray()
df3=pd.read_csv("Stress Detection.csv")
df3=df3.drop(["subreddit","post_id","sentence_range","syntax_fk_grade"],axis=1)
df3.columns=["Text","Sentiment","Stress Level"]
x=transform3(df3["Text"])
y=df3["Stress Level"].to_numpy()
x_train3,x_test3,y_train3,y_test3=train_test_split(x,y,test_size=0.1,random_state=0)
model3=DecisionTreeRegressor(max_leaf_nodes=2000)
model3.fit(x_train3,y_train3)
#Stress Detection Page
if rad=="Stress Detection":
st.header("Detect The Amount Of Stress In The Text!!")
sent3=st.text_area("Enter The Text")
transformed_sent3=transform_text(sent3)
vector_sent3=tfidf3.transform([transformed_sent3])
prediction3=model3.predict(vector_sent3)[0]
if st.button("Predict"):
if prediction3>=0:
st.warning("Stressful Text!!")
elif prediction3<0:
st.success("Not A Stressful Text!!")
Ln 193, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
1:13 PM 7/7/2023


File Edit Selection View Go Run Terminal Help
app.py - Untitled (Workspace) - Visual Studio Code
app.py 9+
Analysis Project > Code Files > app.py > ...
st.image("Stress-detection3.png")
st.subheader("Benefits of Stress Detection:-")
st.markdown("""- Performance optimization: Stress can significantly impact cognitive functions,
memory, concentration, and decision-making abilities.""")
st.markdown("""- Cost savings: Chronic stress can result in increased healthcare costs, absenteeism,
and decreased productivity""")
st.markdown("""- Personal well-being: Stress detection can help individuals become more aware of their
stress levels and take proactive steps to manage them. By identifying stress early on,
individuals can implement strategies to reduce stress, improve their mental health, and
enhance overall well-being.""")
#Hate & Offensive Content Prediction
tfidf4=TfidfVectorizer(stop_words=sw,max_features=20)
def transform4(txt1):
txt2=tfidf4.fit_transform(txt1)
return txt2.toarray()
df4=pd.read_csv("Hate Content Detection.csv")
df4=df4.drop(["Unnamed: 0","count","neither"],axis=1)
df4.columns=["Hate Level","Offensive Level","Class Level","Text"]
x=transform4(df4["Text"])
y=df4["Class Level"]
x_train4,x_test4,y_train4,y_test4=train_test_split(x,y,test_size=0.1,random_state=0)
model4=RandomForestClassifier()
model4.fit(x_train4,y_train4)
#Hate & Offensive Content Page
if rad=="Hate and Offensive Content Detection":
st.header("Detect The Level Of Hate & Offensive Content In The Text!!")
Ln 195, Col 37 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
1:51 PM 7/7/2023


File Edit Selection View Go Run Terminal Help
app.py - Untitled (Workspace) - Visual Studio Code
app.py 9+
Analysis Project > Code Files > app.py > ...
(function) def transform5(txt1: Any) -> Any
st.exception("Highly Offensive Text!!")
elif prediction4==1:
st.warning("Offensive text!!")
elif prediction4==2:
st.success("Non Offensive Text!!")
#Sar
tfid
=20)
def transform5(txt1):
txt2=tfidf5.fit_transform(txt1)
return txt2.toarray()
df5=pd.read_csv("Sarcasm Detection.csv")
df5.columns=["Text","Label"]
x=transform5(df5["Text"])
y=df5["Label"]
x_train5,x_test5,y_train5,y_test5=train_test_split(x,y,test_size=0.1,random_state=0)
model5=LogisticRegression()
model5.fit(x_train5,y_train5)
#Sarcasm Detection Page
if rad=="Sarcasm Detection":
st.header("Detect Whether The Text Is Sarcastic Or Not!!")
sent5=st.text_area("Enter The Text")
transformed_sent5=transform_text(sent5)
vector_sent5=tfidf5.transform([transformed_sent5])
prediction5=model5.predict(vector_sent5)[0]
if st.button("Predict"):
Ln 195, Col 37 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
1:53 PM 7/7/2023


File Edit Selection View Go Run Terminal Help
app.py - Untitled (Workspace) - Visual Studio Code
app.py 9+
Analysis Project > Code Files > app.py > ...
model1.fit(x_train1,y_train1)
#Spam Detection Analysis Page
if rad=="Spam or Ham Detection":
st.header("Detect Whether A Text Is Spam Or Ham??")
sent1=st.text_area("Enter the Values")
transformed_sent1=transform_text(sent1)
vector_sent1=tfidf1.transform([transformed_sent1])
prediction1=model1.predict(vector_sent1)[0]
if st.button("Predict"):
if prediction1=="spam":
st.warning("Spam Text!!")
elif prediction1=="ham":
st.success("Ham Text!!")
#Sentiment Analysis Prediction
tfidf2=TfidfVectorizer(stop_words=sw,max_features=20)
def transform2(txt1):
txt2=tfidf2.fit_transform(txt1)
return txt2.toarray()
df2=pd.read_csv("Sentiment Analysis.csv")
df2.columns=["Text","Label"]
x=transform2(df2["Text"])
y=df2["Label"]
x_train2,x_test2,y_train2,y_test2=train_test_split(x,y,test_size=0.1,random_state=0)
model2=LogisticRegression()
model2.fit(x_train2,y_train2)
Ln 193, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
Search
ENG IN
1:44 PM 7/7/2023

```
from tracemalloc import stop
import streamlit as st

import numpy as np
import pandas as pd
import re
import string
import nltk
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier

nltk.download('punkt')

nltk.download('stopwords')

sw=nltk.corpus.stopwords.words("english")
```

```
#Sentiment Analysis Page

if rad=="Sentiment Analysis":
    st.header("Detect The Sentiment Of The Text!!")
    sent2=st.text_area("Enter The Text")
    transformed_sent2=transform_text(sent2)
    vector_sent2=tfidf2.transform([transformed_sent2])
    prediction2=model2.predict(vector_sent2)[0]

    if st.button("Predict"):
        if prediction2==0:
            st.warning("It's a Negetive Text :(")
        elif prediction2==1:
            st.success("It's a Positive Text :) ")

    st.image("SentimentAnalysisImage.png")


    st.subheader("WHAT IS SENTIMENT ANALYSIS?")

    st.info("""

            Sentiment analysis, also referred to as opinion mining, is an approach to natu
            that identifies the emotional tone behind a body of text.
```

CHAPTER -5 BIBLIOGRAPHY

https://www.kaggle.com/datasets

https://docs.streamlit.io/

https://formsubmit.co/

SPECIAL THANKS TO ASHWINI MAM FOR GUIDING THROUGHOUT THE PROJECT.