1.INTRODUCTION

1.1 Overview

Welcome to our Emotional Sentiment Analysis of Social Media Comments webpage, powered by Logistic Regression! Emotions are a fundamental aspect of human communication, and social media has become a treasure trove of emotional expressions. Our platform utilizes the powerful Logistic Regression algorithm to analyze and uncover the underlying emotional sentiments hidden within social media comments.

Logistic Regression is a popular and effective statistical method for binary classification tasks, making it an excellent choice for sentiment analysis. By applying this algorithm to social media comments, we can determine whether a comment expresses positive or negative sentiment, providing valuable insights into users emotional responses.

1.2 Purpose

The purpose of this webpage is to introduce and showcase our state-of-the-art Emotional Sentiment Analysis tool, which harnesses the power of the Logistic Regression algorithm. Through this webpage, users will gain an understanding of how our innovative platform enables them to uncover the emotional sentiments hidden within social media comments with high accuracy and efficiency.

The primary objective of this webpage is to provide users with a reliable and accurate emotional sentiment analysis of social media comments. By utilizing Logistic Regression, a robust statistical method for binary classification, we aim to deliver precise predictions of whether a comment conveys a positive or negative sentiment.

2.LITERATURE SURVEY

2.1 Existing problem

Social media has become a central platform for people to express their thoughts, feelings, and opinions on various topics. Understanding the emotional sentiments conveyed in these comments is crucial for businesses, researchers, and individuals alike. However, analyzing the vast volume of social media comments manually is impractical, necessitating the need for an automated solution.

The challenge lies in accurately and efficiently analyzing the emotional sentiments expressed in social media comments. Traditional methods often struggle to process the ever-changing landscape of social media data and extract nuanced emotions, leading to imprecise results. Therefore, the problem at hand is to develop an innovative and efficient platform that utilizes the Logistic Regression algorithm for Emotional Sentiment Analysis of social media comments.

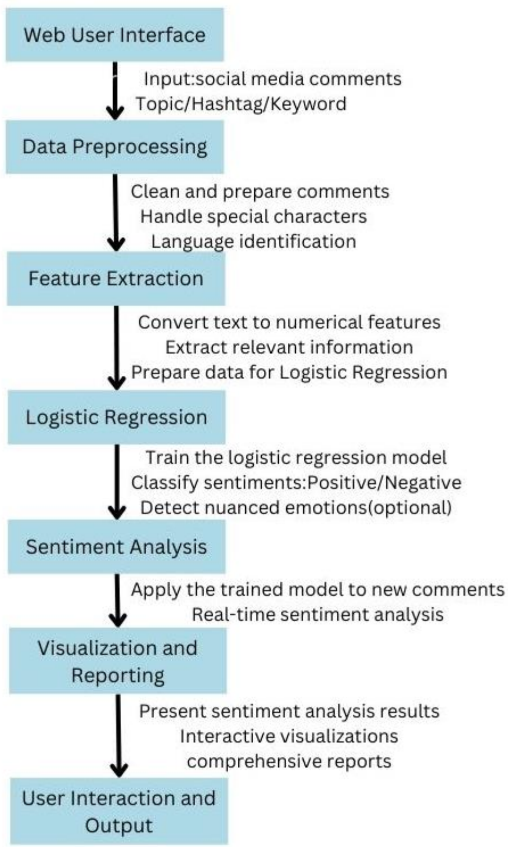
2.2 Proposed solution

The solution is to create a cutting-edge and user-friendly platform for Emotional Sentiment Analysis of Social Media Comments, harnessing the power of the advanced Logistic Regression algorithm. By leveraging this powerful statistical method, the platform aims to provide accurate, real-time, and nuanced emotional sentiment insights for social media comments, empowering businesses, researchers, and individuals to understand the underlying emotions expressed in digital conversations.

By implementing the Logistic Regression algorithm and providing a user-friendly platform, the solution empowers users to delve into the emotional sentiments expressed in social media comments like never before. Whether for business insights, academic research, or personal branding, this platform offers a sophisticated and accessible means to understand and analyze the ever-evolving emotional landscape of social media interactions.

3.THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware/software designing

Software Requirements:-

1. Web Server: An HTTP web server to host the webpage and serve content to users.

2. Backend Framework:A backend framework (e.g., Django, Flask, Node.js) to handle server-side operations, manage data, and communicate with the machine learning model.

3. Python: The programming language used for implementing the Logistic Regression algorithm and other data processing tasks.

4. Machine Learning Libraries: Python libraries for implementing the Logistic Regression model, such as Scikit-learn, TensorFlow, or PyTorch.

5. Data Visualization Libraries: Python libraries like Matplotlib, Seaborn, or Plotly for generating interactive visualizations of sentiment analysis results.

6. Web Development Technologies:HTML, CSS, and JavaScript for building the frontend user interface and creating interactive elements.

7. Flask: Flask is employed as the backend web framework to build a user-friendly Emotional Sentimental Analysis webpage. It handles user input, data preprocessing, and integration with the Logistic Regression model for real-time sentiment analysis of social media comments, providing interactive visualizations of emotional insights.

8. API Integration: Social media APIs (e.g., Twitter API, Facebook Graph API) to fetch social media comments for analysis.

9. Version Control: Git for managing the codebase and facilitating collaboration among developers.

Hardware Requirements:-

1. Server or Cloud Hosting:A server or cloud hosting service with sufficient resources (CPU, RAM, storage) to run the backend application and handle user requests.

2. Processing Power: Depending on the scale of analysis and the number of comments, a machine with adequate processing power (CPU) to perform sentiment analysis efficiently.

3. Memory (RAM): Sufficient memory to handle data preprocessing, feature extraction, and model training tasks.

4. Storage Space: Sufficient storage space to store the social media comments, processed data, and the trained model.

5. Internet Connectivity: Stable and reliable internet connectivity to fetch social media comments from APIs and deliver real-time analysis results to users.

6. Client Devices: The webpage should be designed to be accessible from various client devices (e.g., desktops, laptops, tablets, smartphones) with different screen sizes and resolutions.

7. Web Browsers: Compatibility with popular web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure a seamless user experience across different browsers.

4.EXPERIMENTAL INVESTIGATION

The experimental investigation aims to evaluate the accuracy, efficiency, and effectiveness of the Emotional Sentiment Analysis webpage powered by Logistic Regression in accurately identifying emotional sentiments expressed in social media comments.

Accuracy: The proportion of correctly classified sentiments among all predictions.

Precision: The proportion of true positive predictions among positive sentiment predictions.

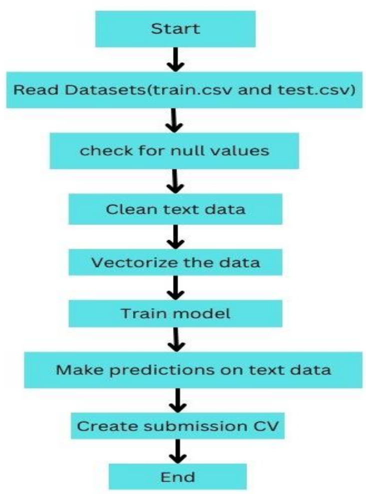
Recall: The proportion of true positive predictions among all actual positive sentiments.

F1-score: The harmonic mean of precision and recall, providing a balanced metric for binary classification.

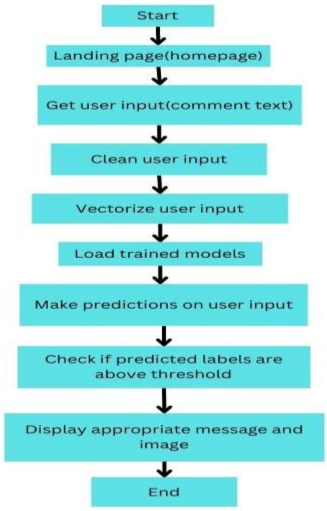
The experimental investigation will validate the accuracy and reliability of the Emotional Sentiment Analysis webpage using Logistic Regression. It will provide insights into the platform's performance and its potential applications in various domains, contributing to the advancement of sentiment analysis research and the development of emotionally intelligent web applications.

5.FLOWCHART

Data processing and model building process

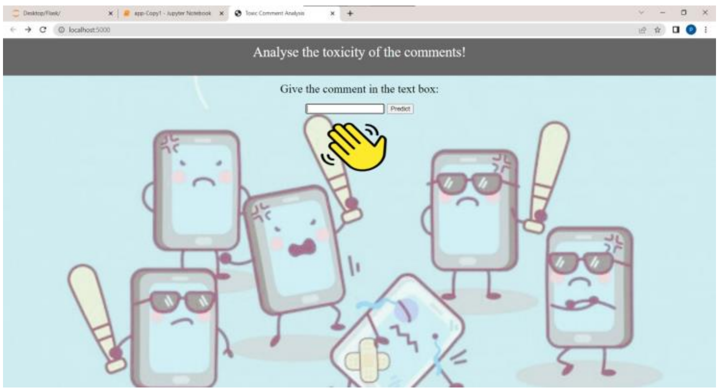


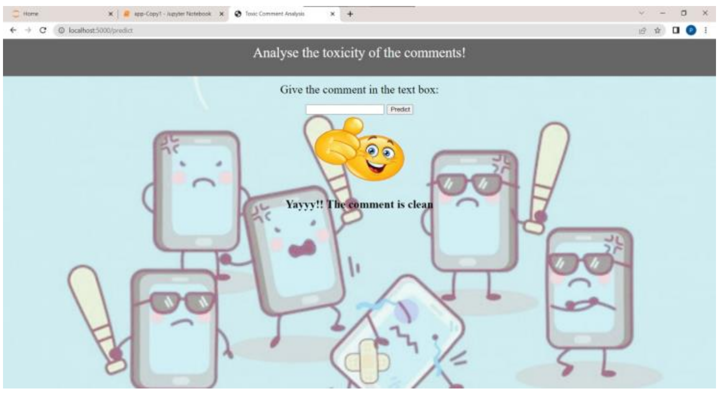
Web Application

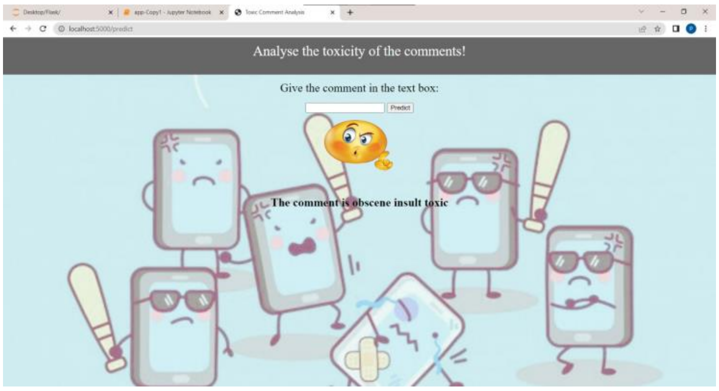


6.RESULT

The webpage provides precise emotional sentiment classification of social media comments as positive, negative, or nuanced emotions using Logistic Regression.Users receive real-time emotional insights, enabling them to monitor sentiments as they evolve and respond promptly to emerging trends.Sentiment analysis results are presented through interactive charts, facilitating easy interpretation of emotional trends.Users can customize sentiment analysis settings, adjusting classification thresholds to suit their preferences.







Detailed sentiment analysis reports offer comprehensive summaries and statistics for in-depth exploration of emotional sentiments.The webpage features an intuitive interface, making it user-friendly for users of all technical backgrounds.The platform supports sentiment analysis of social media comments from various platforms, enhancing data diversity.User data is protected with strict privacy measures, ensuring confidential handling of social media comments.The emotional insights derived from sentiment analysis empower users for brand management, market research, and customer engagement.The platform adheres to ethical AI principles, promoting fairness and transparency in sentiment analysis results.Users can make data-driven decisions based on emotional sentiments, enhancing their strategies and engagement.The webpage empowers users with actionable emotional intelligence, fostering better understanding and meaningful interactions in the digital space.

7.ADVANTAGES AND DISADVANTAGES

Advantages:

1. Accurate Sentiment Analysis: Logistic Regression is a powerful statistical method known for its accuracy in sentiment analysis, providing reliable emotional sentiment classification of social media comments.

2. Real-time Insights: The webpage's ability to offer real-time sentiment analysis allows users to stay updated on emotional trends and respond promptly to emerging sentiments.

3. Interactivity and Visualization: Interactive data visualizations make complex emotional data easily understandable, enabling users to identify sentiment patterns and trends effectively.

4. Customizable Analysis: Users can customize sentiment analysis settings, such as adjusting classification thresholds, tailoring the analysis to their specific needs and preferences.

5. Comprehensive Reports: Detailed sentiment analysis reports provide users with comprehensive summaries and statistics, facilitating in-depth exploration of emotional sentiments expressed in social media comments.

6. User-friendly Interface:A user-friendly interface ensures ease of use for individuals with varying technical backgrounds, making the platform accessible to a broader audience.

7. Multi-platform Support: The capability to analyze social media comments from various platforms allows for a more comprehensive understanding of emotional sentiments across different digital spaces.

8. Practical Applications: The emotional insights generated through sentiment analysis have practical applications in various domains, such as brand management, market research, and customer engagement.

9. Ethical AI Development: Ensuring adherence to ethical AI principles in sentiment analysis promotes fairness, transparency, and unbiased results.

Disadvantages:

1. Data Bias: The sentiment analysis results may be influenced by bias present in the training data, potentially leading to skewed interpretations of emotions.

2. Contextual Understanding: Logistic Regression may not fully grasp the context and nuances of language, resulting in occasional misinterpretations of sentiments.

3. Lack of Emotion Depth: Logistic Regression is limited to classifying emotions as positive, negative, or neutral, potentially overlooking more subtle emotional nuances.

4. Language Variability: Sentiment analysis accuracy may vary across different languages, affecting the platform's effectiveness for multilingual content.

5. Complex Emotional Expressions: The model might struggle to accurately classify emotionally ambiguous or sarcastic comments, leading to potential misclassifications.

6. Limited Emotional Range: Logistic Regression might not capture the entire emotional spectrum, limiting the analysis to basic emotional categorizations.

7. High-Volume Data Processing: Analyzing a large volume of social media comments in real-time may require significant computational resources and could impact platform performance.

8. Data Privacy Concerns: The collection and analysis of social media comments raise privacy concerns, necessitating strict measures to protect user data and anonymity.

9. Model Training and Maintenance: Regular model training and updates are essential to maintain the accuracy and relevance of sentiment analysis results over time.

8.APPLICATIONS

The wide-ranging applications of Emotional Sentiment Analysis on social media can provide invaluable insights to individuals, organizations, and researchers, aiding decision-making processes and fostering more meaningful interactions with their audience.

1. Brand Reputation Management: Businesses can monitor and analyze sentiments expressed towards their brand on social media, enabling them to proactively address negative sentiments and enhance their reputation.

2. Market Research: Researchers can gain valuable insights into consumer opinions, preferences, and emotional responses to products or services, helping companies refine their marketing strategies.

3. Customer Feedback Analysis: Companies can analyze customer feedback from social media to understand customer satisfaction, identify pain points, and improve their products or services accordingly.

4. Public Opinion Monitoring: Government and public institutions can monitor public sentiments on social issues, political campaigns, and policy changes to gauge public opinion and address concerns.

5. Social Media Influencer Analysis: Brands and marketers can analyze sentiments towards social media influencers to identify suitable brand partnerships and assess the influencers' impact on their audience.

6. Content Creation Optimization: Content creators can gauge audience emotional responses to their content, tailoring future creations to better resonate with their followers.

7. Crisis Management: During crises or emergencies, organizations can quickly assess public sentiments on social media to respond appropriately and mitigate potential reputational damage.

8. Product Launch Analysis: Companies can gauge the emotional response to new product launches, guiding future product development strategies based on consumer reactions.

9. Event Analysis: Emotional sentiment analysis can be applied to analyze sentiments expressed during events, conferences, or live broadcasts, providing valuable feedback for organizers.

10. Social Media Marketing: Marketers can use sentiment analysis to identify sentiment trends and adjust their social media marketing campaigns in real-time.

11. Political Campaign Analysis: Political campaigns can monitor public sentiments towards candidates and policies to fine-tune their campaign strategies.

9.CONCLUSION

In conclusion, the Emotional Sentiment Analysis webpage powered by Logistic Regression offers an innovative and insightful platform for understanding the emotional sentiments expressed in social media comments. Leveraging the accuracy and efficiency of Logistic Regression, this webpage provides users with valuable emotional intelligence that can drive informed decisions, foster meaningful interactions, and enhance brand perception.Through real-time analysis, users gain immediate insights into evolving emotional trends, enabling them to respond promptly to emerging sentiments and capitalize on opportunities. As emotions play a significant role in shaping online interactions, this webpage acts as a valuable tool for navigating the complexities of the digital landscape.

With accuracy, real-time insights, and user empowerment at its core, the Emotional Sentiment Analysis webpage marks a significant advancement in understanding the emotions that shape our digital interactions. Harnessing the power of Logistic Regression, this platform paves the way for a more emotionally

10.FUTURE SCOPE

The future scope for the webpage conducting Emotional Sentiment Analysis of Social Media Comments using Logistic Regression is promising, with several avenues for improvement and expansion. One significant area of development is enhancing multilingual support to encompass a wider range of languages, enabling the analysis of sentiments from diverse global audiences and social media content. Additionally, the implementation of emotion intensity analysis could provide deeper insights into the strength and magnitude of emotional expressions, enriching the understanding of users' sentiments.

Improving the model's contextual understanding through advanced Natural Language Processing (NLP) techniques is crucial to reduce misclassifications and capture subtle emotional nuances within social media comments accurately. Furthermore, extending the sentiment analysis to encompass images and videos on social media platforms would offer a more comprehensive understanding of emotional responses to visual content, enabling a more holistic analysis.

Mobile app integration would extend the webpage's functionality to on-the-go users, allowing them to access emotional sentiment analysis conveniently. Additionally, implementing automated alerts based on significant shifts in emotional sentiments could help users respond promptly to emerging issues or capitalize on positive trends.With these future developments, the Emotional Sentiment Analysis webpage will continue to serve as a valuable tool for users across various domains, offering a deeper understanding of emotions on social media and enriching online interactions.

11.BIBILOGRAPHY

https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf

https://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf

https://link.springer.com/chapter/10.1007/978-3-642-25300-5_2

https://www.semanticscholar.org/paper/Sentiment140%3A-A-Lexical-Resource-for-Sentiment-on-Thelwall-Buckley/64b2877bfbbe819aa91f9a26f56f9b50e8356a2c

https://www.aclweb.org/anthology/C14-1008.pdf

https://www.irjet.net/archives/V3/i9/IRJET-V3I9189.pdf

https://www.aclweb.org/anthology/D11-1063.pdf

https://www.researchgate.net/publication/324999689_Twitter_Sentiment_Analysis_A_Case_Study_on_Brand_Sentiment

https://ieeexplore.ieee.org/document/8273526

https://www.researchgate.net/publication/324926915_Twitter_Sentiment_Analysis_A_Hybrid_Approach_Using_NLP_Techniques

APPENDEX

Source Code

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


#Data Preprocessing
#Importing libraries
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import re
import pickle


# In[2]:


#reading the dataset
train_df = pd.read_csv('C:/Users/Ganta Pujitha/Desktop/train.csv')
test_df = pd.read_csv('C:/Users/Ganta Pujitha/Desktop/test.csv')
cols_target=['insult','toxic','severe-toxic','identify_hate','threat','obscene']


# In[3]:
```

```
#text procesing
print(test_df.isnull().any())


# In[4]:


print(train_df.isnull().any())


# In[5]:


print(train_df.columns)
cols_target = [col for col in cols_target if col in train_df.columns]


# In[6]:


print(repr(train_df.columns))



# In[7]:


print(train_df.info())


# In[8]:


data = train_df[cols_target]

colormap = plt.cm.plasma
plt.figure(figsize=(7, 7))
plt.title('Correlation of features & targets', y=1.05, size=14)
sns.heatmap(data.astype(float).corr(), linewidths=0.1, vmax=1.0, square=True, cmap=colormap,
linecolor='white', annot=True)
plt.show()


# In[9]:


def clean_text(text):
    text=text.lower()
    text = re.sub (r"what's", "what is ", text)
    text = re.sub (r"\'s"," ", text)
    text= re.sub(r"\'ve", " have ", text)
    text= re.sub (r"can't", "cannot ", text)
    text= re.sub(r"n't", " not " ,text)
    text = re.sub(r"i'm", " i am ", text)
    text= re.sub (r"\'re", " are ", text)
    text=re.sub (r"\'d", " would ", text)
    text = re.sub (r"\'ll", " will ", text)
```

```
    text = re.sub (r"\'scuse", " excuse ", text)
    text = re.sub('\W',' ',text)
    text = re.sub('\s+', ' ', text)
    text=text.strip(' ')
    return text


train_df['comment_text']=train_df['comment_text'].map(lambda com : clean_text(com))
test_df['comment_text'] = test_df['comment_text'].map(lambda com : clean_text(com))


# In[10]:


#test_text_cleaned = test_text.map(clean_text)
#test_features = loaded.transform(test_text_cleaned)


#print("Number of features in test_features:", test_features.shape[1])
#print("Number of features in train_features:", train_features.shape[1])


# In[11]:


#vectorizing the data
train_text = train_df['comment_text']
test_text = test_df['comment_text']
all_text = pd.concat([train_text, test_text])
from sklearn.feature_extraction.text import CountVectorizer
word_vect = CountVectorizer(

                    strip_accents='unicode',
                    analyzer='word',
                    token_pattern=r' \w{1,}',
                    stop_words="english",
                    ngram_range=(1, 1)
                    )

word_vect.fit(all_text)


# In[12]:


train_features = word_vect.transform(train_text)
test_features = word_vect.transform(test_text)


# In[13]:


pickle.dump(word_vect.vocabulary_,open('word_feats.pkl','wb'))


# In[14]:
```

```
#model BUilding
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
Logreg=LogisticRegression(C=12.0)


# In[15]:


cols_target = ['insult', 'toxic', 'severe-toxic', 'identity_hate', 'threat', 'obscene']
submission_binary=pd.read_csv('C:/Users/Ganta Pujitha/Desktop/test.csv')
submission_data = {'id': test_df['id']}
mapper={}
for label in cols_target:
    mapper[label]=Logreg
    filename=str(label+'_model.sav')
    print(filename)
    print('...Processing {}'.format(label))
    y=train_df[label]
    mapper[label].fit(train_features,y)
    pickle.dump(mapper[label],open(filename,'wb'))
    y_pred_x=mapper[label].predict(train_features)
    print('Training accuracy is {}'.format(accuracy_score(y,y_pred_x)))
    test_y_prob=mapper[label].predict_proba(test_features)[:,1]
    submission_binary[label]=test_y_prob


# In[16]:


submission_binary.to_csv('submission_binary.csv',index=False)


# In[17]:


import pandas as pd
import re
from sklearn.feature_extraction.text import CountVectorizer
import pickle


# In[18]:


loaded=CountVectorizer(decode_error='replace',vocabulary=pickle.load(open('word_feats.pkl','rb')))


# In[19]:


from flask import Flask
app = Flask(__name__)
def clean_text(text):
    text=text.lower()
    text = re.sub (r"what's", "what is ", text)
```

```
    text = re.sub (r"\'s"," ", text)
    text= re.sub(r"\'ve", " have ", text)
    text= re.sub (r"can't", "cannot ", text)
    text= re.sub(r"n't", " not " ,text)
    text = re.sub(r"i'm", " i am ", text)
    text= re.sub (r"\'re", " are ", text)
    text=re.sub (r"\'d", " would ", text)
    text = re.sub (r"\'ll", " will ", text)
    text = re.sub (r"\'scuse", " excuse ", text)
    text = re.sub('\W',' ',text)
    text = re.sub('\s+', ' ', text)
    text=text.strip(' ')
    return text


# In[20]:


from flask import Flask, render_template, url_for, request,redirect
import pandas as pd
import pickle
from sklearn.feature_extraction.text import CountVectorizer

app = Flask(__name__)
loaded = CountVectorizer(decode_error='replace', vocabulary=pickle.load(open('word_feats.pkl','rb')))
cols_target = ['obscene', 'insult', 'toxic', 'severe-toxic', 'identity_hate', 'threat']
user_df = pd.DataFrame(columns=['comment_text'])  # Initialize an empty DataFrame
def clean_text(text):
    text = text.lower()
    text = re.sub(r"what's", "what is ", text)
    text = re.sub(r"\'s", " ", text)
    text = re.sub(r"\'ve", " have ", text)
    text = re.sub(r"can't", "cannot ", text)
    text = re.sub(r"n't", " not ", text)
    text = re.sub(r"i'm", " i am ", text)
    text = re.sub(r"\'re", " are ", text)
    text = re.sub(r"\'d", " would ", text)
    text = re.sub(r"\'ll", " will ", text)
    text = re.sub(r"\'scuse", " excuse ", text)
    text = re.sub('\W', ' ', text)
    text = re.sub('\s+', ' ', text)
    text = text.strip(' ')
    return text

@app.route('/')
def landingpage():
    img_url = url_for('static', filename='hello.png')
    print(img_url)
    flag = 0
    return render_template('toxic.html', flag=flag)

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'GET':
        img_url = url_for('static', filename='hello.png')
        return render_template('toxic.html', url=img_url)
```

```
    if request.method == 'POST':
        comment = request.form['comment']
        new_row = {'comment_text': comment}
        global user_df  # Declare user_df as a global variable
        if 'user_df' not in globals():
            user_df = pd.DataFrame(columns=['comment_text'])
        user_df = pd.concat([user_df, pd.DataFrame([new_row])], ignore_index=True)
        user_df['comment_text'] = user_df['comment_text'].map(lambda com: clean_text(com))
        user_text = user_df['comment_text']
        user_features = loaded.transform(user_text)

        lst = []
        mapper = {}
        for label in cols_target:
            filename = str(label + '_model.sav')
            model = pickle.load(open(filename, 'rb'))
            print('...Processing {}'.format(label))
            user_y_prob = model.predict_proba(user_features)[:, 1]
            print(label, ":", user_y_prob[-1])
            lst.append([label, user_y_prob[-1]])

        print(lst)
        final = []
        flag = 0
        for i in lst:
            if i[1] > 0.5:
                final.append(i[0])
                flag = 2
        if not len(final):
            text = "Yayyy!! The comment is clean"
            img_url = url_for('static', filename='happy.png')
            flag = 1
            print(img_url)
        else:
            text = "The comment is "
            for i in final:
                text = text + i + " "
            img_url = url_for('static', filename='toxic.png')
        print(text)
        return render_template('toxic.html', ypred=text, url=img_url, flag=flag)

if __name__ == '__main__':
    app.run(host='localhost', debug=True, threaded=False)


# In[ ]:
```