Project Report: Sentiment Analysis Bot

-By Ayush Singh

# 1. Introduction

The project is centered around building a Sentiment Analysis Bot that classifies text data as either positive or negative sentiment. The model is developed using Natural Language Processing (NLP) techniques and is deployed via a user-friendly interface built using Streamlit. The bot processes user input, performs sentiment analysis, and returns the sentiment of the given text.

# 2. Project Objectives

The primary objectives of this project are:  
- To develop a machine learning model capable of predicting the sentiment (positive or negative) of a given text.  
- To implement NLP techniques for text preprocessing, such as tokenization, stopword removal, and stemming.  
- To create an interactive web application using Streamlit that allows users to input text and receive sentiment predictions.

# 3. Data Preprocessing

The data preprocessing steps are crucial for the performance of the sentiment analysis model. The following steps were performed:  
- \*\*Text Cleaning\*\*: Removal of unnecessary characters and punctuation from the text.  
- \*\*Lowercasing\*\*: Converting all characters in the text to lowercase to maintain uniformity.  
- \*\*Tokenization\*\*: Splitting the text into individual words or tokens using NLTK's `word\_tokenize` function.  
- \*\*Stopword Removal\*\*: Eliminating common English stopwords using NLTK’s stopword list to reduce noise in the data.  
- \*\*Stemming\*\*: Reducing words to their base or root form using Porter Stemmer to simplify the input data.

# 4. Model Training

The machine learning model used in this project is a pre-trained model (`nlp\_model.pkl`) loaded using Joblib. The model was trained to classify text into two sentiment classes:  
- \*\*Positive\*\*: Indicates a favorable or happy sentiment.  
- \*\*Negative\*\*: Indicates an unfavorable or unhappy sentiment.

A TF-IDF vectorizer (`tfidf.pkl`) was used to transform the preprocessed text into numerical form suitable for model input. The TF-IDF vectorizer converts the text into a matrix of TF-IDF features.

# 5. Model Deployment Using Streamlit

The application was deployed using Streamlit, a framework that allows quick and easy development of web applications for machine learning models. The Streamlit app provides a simple user interface where users can input text, and by clicking a button, they can receive the sentiment prediction. The following steps were performed in the app:  
- \*\*User Input\*\*: The user enters the text they wish to analyze.  
- \*\*Text Processing\*\*: The input text undergoes preprocessing similar to what was done during model training.  
- \*\*Prediction\*\*: The processed text is passed to the model, and a sentiment prediction is made.  
- \*\*Output\*\*: The sentiment (positive or negative) is displayed to the user.

# 6. Challenges and Resolutions

- \*\*Input Shape Errors\*\*: Errors related to mismatched input shapes were encountered when passing data to the model. These were resolved by ensuring that the input data was correctly preprocessed and shaped before being passed to the model.  
- \*\*Accuracy and Thresholding\*\*: Determining the appropriate threshold for classifying a text as positive or negative was essential. The threshold was set such that if the model's predicted class index exceeded 0.6, the text was classified as positive.

# 7. Conclusion

The Sentiment Analysis Bot successfully predicts the sentiment of a given text using a combination of NLP techniques and machine learning. The integration with Streamlit provides a seamless user experience, allowing users to interact with the model in real-time.

# 8. Future Work

Potential future improvements include:  
- \*\*Model Optimization\*\*: Further tuning the model and experimenting with different algorithms to improve accuracy.  
- \*\*Expanding Sentiment Classes\*\*: Extending the sentiment analysis to include more nuanced sentiment classes, such as neutral or mixed.  
- \*\*Real-Time Data Analysis\*\*: Enhancing the bot to handle real-time sentiment analysis on data streams from social media or other sources.

# 9. References

- \*\*Streamlit Documentation\*\*: [Streamlit](https://docs.streamlit.io/)  
- \*\*NLTK Documentation\*\*: [NLTK](https://www.nltk.org/)  
- \*\*Scikit-learn Documentation\*\*: [Scikit-learn](https://scikit-learn.org/stable/)