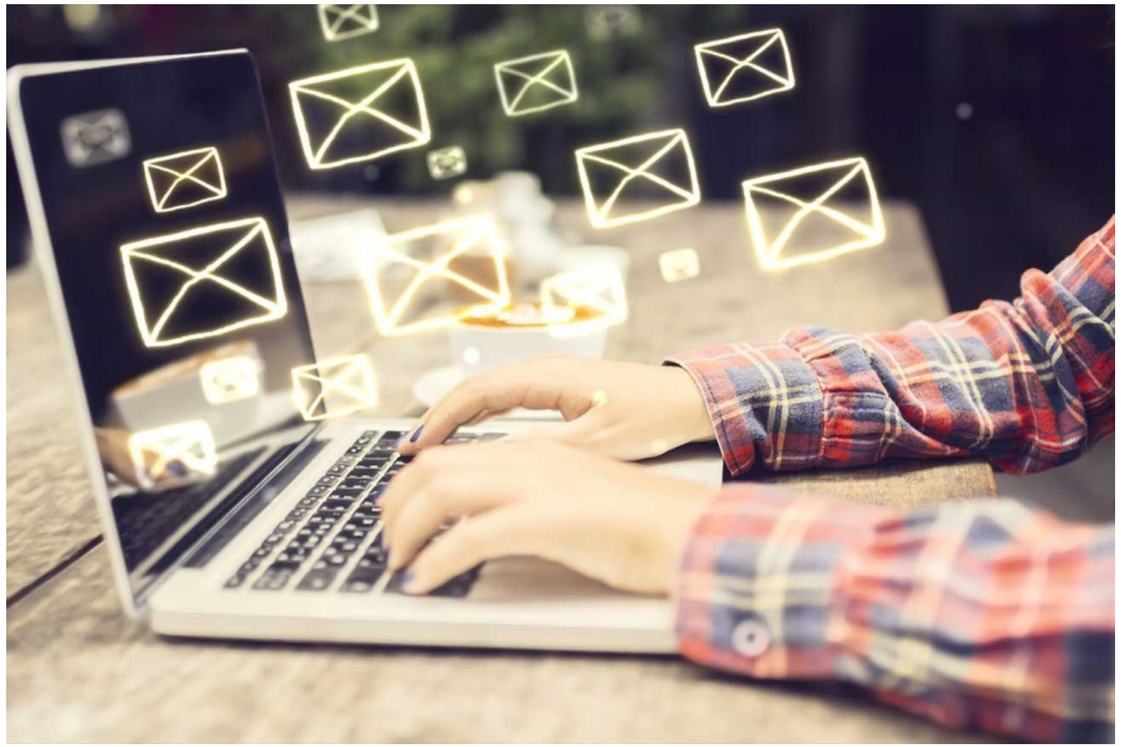
Spam Detector

for Short Message Service (SMS)



Group Members :-

Siddhesh Parab

Aamir Indi

Affan Boral

What?

A spam detector using NLP (Natural Language Processing) is a machine learning model that analyzes text messages to identify and filter out unwanted or unsolicited content, commonly found in emails or short messages. It leverages NLP techniques to detect patterns and features in text data, enabling it to distinguish between legitimate and spam messages based on factors such as keywords, context, and linguistic analysis. This technology is essential for maintaining communication channels free from unwanted or harmful content, enhancing user experience, and protecting against phishing and fraud.

Why?

Protecting User Privacy: It filters out unwanted and potentially harmful messages, ensuring user data remains secure.

Enhancing User Experience: By removing spam, it ensures users receive relevant and meaningful content, improving their experience.

Reducing Fraud and Scams: Identifies and blocks fraudulent messages, helping prevent financial and identity theft.

Improving Productivity: Minimizes distractions caused by spam, allowing users to focus on important communication.

Maintaining Reputation: Ensures that legitimate messages reach users, maintaining the sender's credibility and trustworthiness.

TechStack

- **Frontend** :- Html, CSS
- **Backend** :- Python, Flask

Steps :-

- ***Exploratory Data Analysis (EDA):***

Exploring NaN values in dataset: Analyzing the dataset for missing values to ensure data completeness.
Plotted countplot for SMS labels Spam vs. Ham: Visualizing the distribution of spam and non-spam (ham) messages to understand the class balance.

- **Data Cleaning:**
  Removing special characters and numbers using regular expressions: Preprocessing the text data by eliminating unwanted characters.
  Converting the entire SMS into lowercase: Ensuring consistent text case for analysis.
  Tokenizing the SMS by words: Breaking down SMS messages into individual words for analysis.
  Removing the stop words: Eliminating common words that do not carry significant meaning.
  Lemmatizing the words: Reducing words to their base form to simplify analysis.
  Joining the lemmatized words: Creating cleaned and preprocessed text data.
  Building a corpus of messages: Organizing the preprocessed text data for modeling.

Model Building and Evaluation:
Metric: F1-Score: Using the F1-score as a performance metric to evaluate model accuracy, precision, and recall.
Multinomial Naive Bayes: Implementing a Multinomial Naive Bayes classifier with an F1-score of 0.943.
Decision Tree: Using a Decision Tree classifier with an F1-score of 0.98.
Random Forest: Employing a Random Forest classifier with an F1-score of 0.994.
Voting (Decision Tree + Multinomial Naive Bayes): Combining models through voting to enhance accuracy, resulting in an F1-score of 0.98

Model Prediction:
The trained model is used for making predictions on incoming SMS messages, classifying them as spam or non-spam based on the learned patterns.

Thank You!