***Report***

**By: Kashmala Azeen Chattha**

**231370094**

**Spam Email Detection using Streamlit**

**1. Introduction**

Spam emails are unwanted and potentially harmful messages that clutter inboxes. This project aims to build a **Spam Email Detection** system using **Streamlit**, a Python framework for interactive web applications. The application allows users to input an email message and classify it as **Spam** or **Ham** (not spam) using a **Naïve Bayes Classifier** trained on a dataset of spam and non-spam emails.

**2. Implementation Steps**

**Task 1: Install and Set Up Streamlit**

1. Install Streamlit using:
2. pip install streamlit
3. Verify installation:
4. streamlit hello
5. Create a basic Streamlit script:
6. import streamlit as st
7. st.title("My First Streamlit App")

**Task 2: Creating a Basic Streamlit App**

1. Created a new Python file: **app.py**.
2. Added a title and header using st.title() and st.header().
3. Included a text input where users enter their name.
4. Added a button to display a welcome message.
5. Implemented a **selectbox** to choose a favorite programming language.

**Task 3: Adding Widgets for Interactivity**

1. Implemented a **slider** for selecting numbers between 1-100.
2. Added a **checkbox** that displays a message when checked.
3. Used **radio buttons** to select user expertise level.
4. Displayed selected options dynamically using st.write().
5. Added a **file uploader** for uploading email datasets (CSV format).

**Task 4: Data Preprocessing & Visualization**

1. Loaded the **Spam Email Dataset** using pandas.
2. Implemented text preprocessing:
   * Converted text to **lowercase**.
   * Removed **punctuation and special characters**.
   * Removed **stopwords** using NLTK.
   * Applied **stemming** to normalize words.
3. Displayed the cleaned dataset in st.dataframe().
4. Created **visualizations**:
   * **Bar Chart** showing spam vs. ham distribution using matplotlib.
   * **Word Cloud** displaying common words in spam emails.
   * **Interactive filtering** where users select email categories.

**Task 5: Running & Sharing the App**

1. Ran the app using:
2. streamlit run app.py
3. Deployed the app on **Streamlit Cloud**.

**Task 6: Creating the Spam Detection Dashboard**

1. **Merged all previous tasks** into a single dashboard.
2. Used **Streamlit sidebar** for navigation.
3. Arranged different sections using **containers & columns**.
4. Trained a **Naïve Bayes classifier** on the preprocessed dataset.
5. Provided a **text input box** for users to check if an email is spam.

**3. Code Implementation**

The final application consists of:

* **Data Loading & Preprocessing**
* **Machine Learning Model Training (Naïve Bayes Classifier)**
* **Interactive User Interface with Streamlit**
* **Data Visualization (Bar Charts & Word Cloud)**
* **Spam Detection Feature**

**4. Screenshots of the Running App**

(Attach screenshots of the different sections of your Streamlit app here.)

**5. Conclusion**

This project successfully implements a **Spam Email Detection System** using **Streamlit and Machine Learning**. The interactive dashboard allows users to analyze the dataset, visualize spam email patterns, and classify new emails effectively. The use of **Natural Language Processing (NLP) techniques** such as **stemming, stopword removal, and TF-IDF vectorization** improved the model’s accuracy. Future enhancements could include **deploying the app online, using deep learning models, or integrating real-time email filtering**.

**Screen shots**

