Spam Message Classifier

Project Report

Malak Mohamed Ragab - 2305303
Menna Walid Mahmoud - 2305304
Logina Mahmoud Elhag - 2305532

Spam Message Classifier Web App

Spam Message Classifier

1. Introduction

This report documents the development of a Spam Message Classifier using Streamlit, Natural Language Processing (NLP), and a Machine Learning model. The app takes in a message, processes the text, predicts whether it's spam or not (called ham), and displays the result with confidence scores. It makes machine learning accessible through a clean web interface.

2. Project Goals

- Build a machine learning model to detect spam messages.
- Design a user-friendly web interface with Streamlit.
- Apply natural language preprocessing techniques.
- Visualize classification results with confidence levels.

3. Libraries and Tools

```
import streamlit as st import
pickle import string from
nltk.corpus import stopwords
import nltk
```

These libraries are used for UI, model management, and NLP.

4. Text Preprocessing

```
def preprocess_text(text):
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.lower()
    stop_words = set(stopwords.words('english'))
    words = text.split()
    words = [word for word in words if word not in stop_words]
    return ' '.join(words)
```

This function cleans the input text by removing punctuation, converting to lowercase, and removing common English stopwords (like 'the', 'is'). This helps the model focus on important words.

Spam Message Classifier

5. Loading the Model

```
@st.cache_resource
def load_model():

    with open('spam_classifier.pkl', 'rb') as f:
        vectorizer, classifier = pickle.load(f)
    return vectorizer, classifier
```

The model is loaded only once and cached, making the app faster. It includes both the vectorizer (which turns text into numbers) and the classifier.

Spam Message Classifier

6. Streamlit Interface

```
st.title("Spam Message Classifier")
st.write("This app predicts whether a text message is spam or not (ham).")
user_input = st.text_area("Enter a message to classify:", "")
```

The app uses Streamlit to create a simple web UI where users can enter a message and get predictions.

7. Prediction Logic and Output

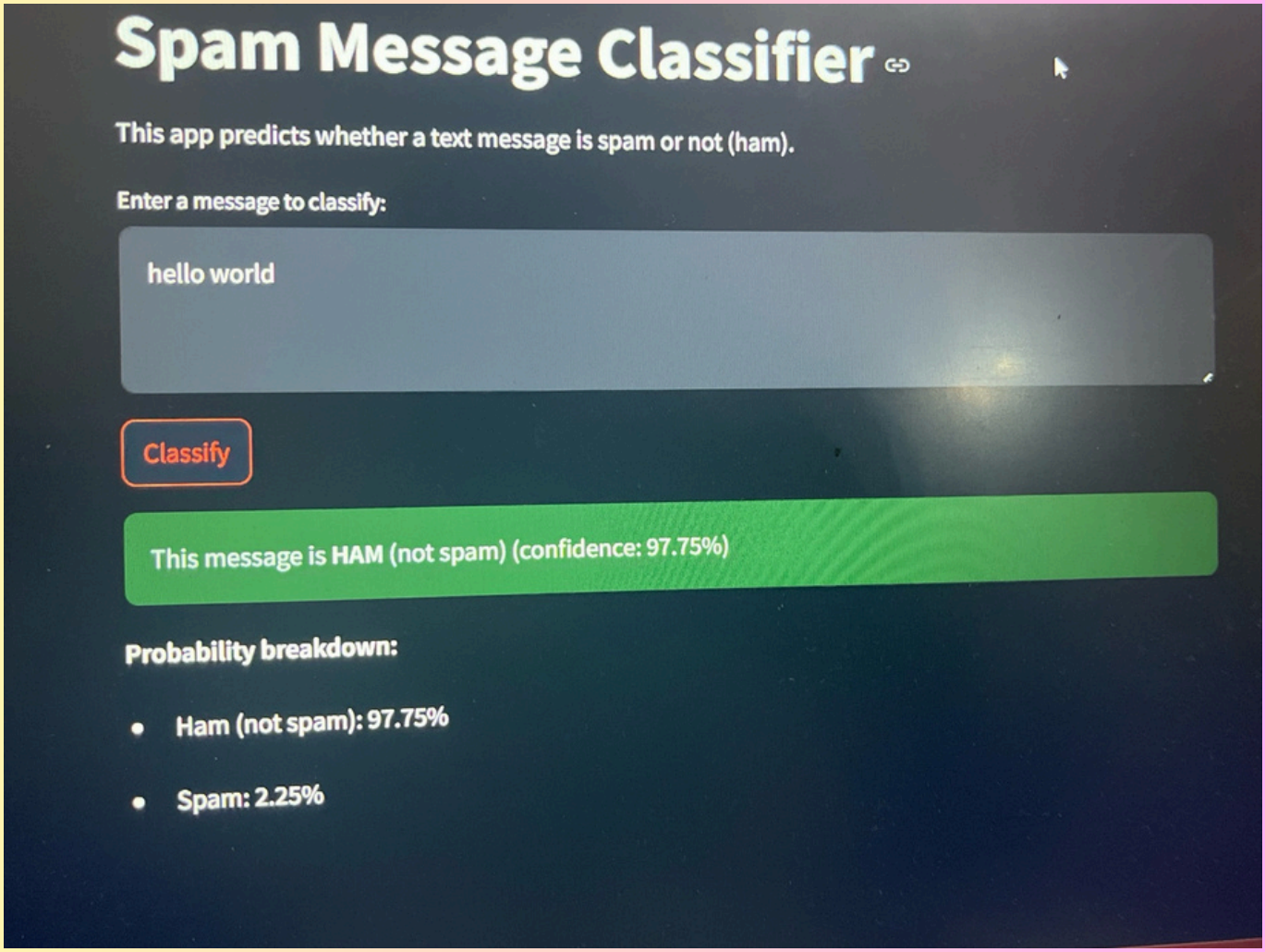
```
if st.button("Classify"):
    if user_input:
        processed_text = preprocess_text(user_input) text_vector
        = vectorizer.transform([processed_text]) prediction =
        classifier.predict(text_vector) prediction_proba =
        classifier.predict_proba(text_vector)

        if prediction[0] == 1:
            st.error("This message is **SPAM**...")
        else:
            st.success("This message is **HAM**...")
```

When the classify button is clicked, the app processes the text, transforms it, runs the model, and shows the result. It displays both label and prediction probability.

8. Example Output Screenshot

Spam Message Classifier



9. How to Run the App

```
streamlit run spam_detector_app.py
```

To launch the web application, use the above command from your terminal.

10. Conclusion

This project integrates natural language processing, machine learning, and web development to provide a real-time spam detection tool. With a purple-themed UI and instant feedback, it's practical for educational and real-world use. Future improvements may include multilingual support, batch message processing, and model explainability (e.g., highlighting important words).