Email/SMS Spam Classifier

Introduction

This project involves the implementation of a machine learning-based system to classify text data. The objective is to preprocess the dataset, evaluate different models, and finalize the most effective model based on accuracy and precision for deployment.

Objectives

1. Develop a text classification system.
2. Compare multiple machine learning models for performance.
3. Finalize and implement the best-performing model.
4. Deploy the system using an interactive user interface via Streamlit.

Theory

1. Machine Learning Models

Several machine learning models were evaluated in this project:

- **Logistic Regression (LR):** A regression-based classification model leveraging the sigmoid function to predict probabilities.
- **Random Forest Classifier (RFC):** An ensemble learning method using multiple decision trees for classification.
- **Support Vector Machine (SVM):** A supervised learning algorithm that finds the hyperplane maximizing class separation.
- **Multinomial Naive Bayes (MNB):** A probabilistic learning model particularly effective for text classification tasks.
- **Stacking Classifier:** Combines predictions from multiple models using a meta-classifier.
- **Voting Classifier:** Aggregates predictions from various models using majority or weighted voting.

2. Preprocessing Techniques

- **Text Tokenization:** Splitting text into meaningful units (tokens).
- **TF-IDF Vectorization:** Converts text data into numerical format by calculating Term Frequency-Inverse Document Frequency scores.
- **Stopwords Removal:** Eliminating commonly used words that do not contribute to the classification task.

3. Evaluation Metrics

- **Accuracy:** Ratio of correctly predicted instances to the total instances.
- **Precision:** Ratio of true positive results to the total predicted positives, providing insight into the model's specificity.

4. Final Model Selection

After comparing accuracy and precision across models, the **Multinomial Naive Bayes (MNB)** model was selected as the final model due to its superior performance.

Tools Used

- **Python Libraries:**
    - pandas: For data manipulation.
    - sklearn: For model implementation and evaluation.
    - nltk: For natural language processing tasks.
    - numpy: For numerical operations.
- **Streamlit:** Framework for building interactive web applications.

Code Workflow

Step 1: Import Libraries and Dataset

- Imported essential libraries for data preprocessing, model building, and evaluation.
- Loaded the dataset containing text data and labels.

Step 2: Data Preprocessing

- Removed stopwords using nltk.

- Applied tokenization and TF-IDF vectorization to convert textual data into numerical format.

Step 3: Model Implementation

- Tested and evaluated multiple models, including Logistic Regression, Random Forest, SVM, Multinomial Naive Bayes, Stacking, and Voting classifiers.
- Calculated accuracy and precision for each model.

Step 4: Model Comparison

- Compared models based on their accuracy and precision metrics.
- Observed that the Multinomial Naive Bayes (MNB) model achieved the highest performance.

Step 5: Final Model Selection

- Selected MNB as the final model.
- Implemented the model for prediction tasks.

Step 6: Deployment

- Developed an interactive web application using Streamlit for users to input text and receive classification results.

Conclusion

This project successfully developed and deployed a text classification system. By testing multiple models, including Stacking and Voting classifiers, the Multinomial Naive Bayes (MNB) model was finalized due to its superior accuracy and precision. The system is accessible through a user-friendly interface built using Streamlit, enabling seamless text classification.