Spam-Ham Analysis Project Report

Name: Vikash Pandey

Code link: https://github.com/vikash282/upskillcampus/blob/main/spam_ham_analysis.py
Report link: https://github.com/vikash282/upskillcampus/blob/main/spam_ham_analysis_Vikash_USC_UCT.pdf

1. Introduction
The Spam-Ham Analysis project aims to build a machine learning model to classify email messages as either spam or ham (non-spam). This is a crucial task in natural language processing (NLP) to filter out unwanted and potentially harmful spam emails from users' inboxes, improving email communication efficiency and security.

2. Project Objectives
- Preprocess and clean the email dataset.
- Implement feature extraction techniques to transform text data into numerical form.
- Develop and train a classification model to accurately distinguish between spam and ham emails.
- Evaluate the model's performance using appropriate metrics.

3. Dataset
3.1 Source
The dataset used for this project is publicly available and contains labeled email messages. Each email is labeled as either "spam" or "ham".

3.2 Description
- **Number of Samples**: 10000
- **Features**: primary feature is "messages" and target is "spam or ham"

4. Data Preprocessing
4.1 Data Cleaning
- Removal of HTML tags and special characters.
- Conversion of text to lowercase to maintain uniformity.
- Removal of stop words (common words that do not contribute to the meaning of the text).

4.2 Tokenization
- Splitting the text into individual words (tokens).

4.3 Stemming and Lemmatization
- Reducing words to their base or root form.

5. Feature Extraction
5.1 Bag of Words (BoW)
- Creating a matrix of token counts for each email.

5.2 Term Frequency-Inverse Document Frequency (TF-IDF)
- Converting the BoW matrix into a TF-IDF matrix to reflect the importance of words.

6. Model Development
6.1 Model Selection
- Naive Bayes classifier was chosen due to its effectiveness in text classification tasks.

6.2 Training the Model
- The dataset was split into training and testing sets.
- The Naive Bayes classifier was trained on the training set.

7. Model Evaluation

7.1 Evaluation Metrics

- **Accuracy**: The ratio of correctly predicted instances to the total instances.
- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.
- **Recall**: The ratio of correctly predicted positive observations to the all observations in actual class.
- **F1 Score**: The weighted average of Precision and Recall.

7.2 Results

Accuracy: 92%

Precision: 89%

Recall: 88%

F1 Score: 88%

8. Conclusion

The Spam-Ham Analysis project successfully developed a model to classify emails as spam or ham. The Naive Bayes classifier achieved satisfactory performance, demonstrating its capability in handling text classification tasks.

9. Future Work

- Experimenting with different feature extraction techniques like Word2Vec or GloVe.
- Implementing more advanced classification algorithms like Support Vector Machines (SVM) or deep learning models.
- Enhancing the model by incorporating additional features such as email metadata.

10. References

- Dataset source- kaggle
- resources- medium and GeekForGeek
- libraries- Nltk, scikit-learn, textblob, genism, pandas, numpy, tensorflow

---

This report provides a comprehensive overview of the Spam-Ham Analysis project, detailing each step from data preprocessing to model evaluation. Feel free to add or modify sections as needed to better suit your specific project requirements.