Email Spam Classification Project

Introduction

This project aims to build a spam email classifier using text classification techniques in Python. The goal is to identify and classify emails as either spam or ham (non-spam) based on their content. The project utilizes popular machine learning algorithms such as Naive Bayes, Naive Bayes Multinomial, and J48 (C4.5 Decision Tree) for classification.

Project Steps

1. Data Collection

The project starts by collecting email data from a CSV file named 'mail_data.csv'. This dataset contains email messages along with their corresponding labels indicating whether they are spam or ham.

2. Pre-processing

Pre-processing involves data cleaning and handling missing values. Rows with missing values are dropped from the dataset, and missing values are replaced with null strings. The dataset is further processed to label spam emails as 0 and ham emails as 1.

3. Feature Extraction

Text data is converted into numerical features using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique. This step prepares the data for the classification algorithms.

4. Feature Selection

The Best First Feature Selection algorithm is used to select the most relevant features for classification. This step helps in reducing the dimensionality of the dataset and improving the performance of the algorithms.

5. Algorithm Selection and Training

Three classification algorithms, namely Naive Bayes, Naive Bayes Multinomial, and J48 (C4.5 Decision Tree), are applied to the pre-processed and feature-selected data. The models are trained on the training dataset and evaluated on the test dataset.

6. Model Evaluation

The accuracy of each algorithm is calculated using the accuracy score metric. The project aims to compare the accuracy, time, and error rate of the different algorithms to determine which one performs best for email spam classification.

Running the Code

1. Ensure you have the necessary dataset named 'mail_data.csv'.

2. Execute the provided code in a Python environment, such as Jupyter Notebook.

3. The code will load, preprocess, and transform the data, apply feature selection, train the classification algorithms, and evaluate their accuracy.

4. The results will be displayed in a tabular format, showing the accuracy of each algorithm.

Conclusion

The project demonstrates the use of machine learning techniques for email spam classification. By applying different classification algorithms and evaluating their performance, we can determine the most suitable algorithm for accurately identifying spam emails.

Feel free to contact me “Bilal Khan” Email: bilalkhan31c7@gmail.com