Email Spam and Non-Spam Classification

Project Domain / Category

Data Science/Machine Learning

Abstract / Introduction

Email becomes a powerful tool for communication as it saves a lot of time and cost. It is one of the most popular and secure medium for online transferring and communication messages or data through the web. But, due to the social networks, most of the emails contain unwanted information which is called spam. To identify such spam email is one of the important challenges.

In this project we will use PYTHON text classification technique to identify or classify email spam message. We will find accuracy, time and error rate by applying suitable algorithms (such as NaiveBayes, NaiveBayesMultinomial and J48 etc.) on Email Dataset and we will also compare which algorithm is best for text classification.

Functional Requirements:

Administrator will perform all these tasks.

1. **Collect Data Set**
   - Gathering the data for Email spam contains spam and non-spam messages
2. **Pre-processing**
   - As most of the data in the real world are incomplete containing noisy and missing values. Therefore we have to apply Pre-processing on your data.
3. **Feature Selection**
   - After the pre-processing step, we apply the feature selection algorithm, the algorithm which deploy here is Best First Feature Selection algorithm.
4. **Apply Spam Filter Algorithms.**
   - **Handle Data:** Load the dataset and split it into training and test datasets.
   - **Summarize Data:** summarize the properties in the training dataset so that we can calculate probabilities and make predictions.
   - **Make a Prediction:** Use the summaries of the dataset to generate a single prediction.
   - **Make Predictions:** Generate predictions given a test dataset and a summarized training dataset.
   - **Evaluate Accuracy:** Evaluate the accuracy of predictions made for a test dataset as the percentage correct out of all predictions made.
5. **Train & Test Data**
   - Split data into 70% training & 30% testing data sets.

6. **Confusion Matrix**
    - Create a confusion matrix table to describe the performance of a classification model.
7. **Accuracy**
    - Find Accuracy of all algorithm and compare.

Tools:

- Python
- Anaconda

Prerequisite:

Artificial intelligence Concepts, Machine learning.

Supervisor:

Name: Muhammad Tayyab Waqar
Email ID: tayyab.waqar@vu.edu.pk
Skype ID: maliktayyab786_1