Dikshant Buwa-04 Mayank Jadhav-20 Yash Sankhe-53 Arpit Sutariya-59

XGBoost-based Spam Classification System: Improving Email Security

Abstract:

Spam messages, or unsolicited and unwanted texts, pose significant challenges in maintaining the integrity of digital communication channels. To combat this issue, automated spam message detection systems have become crucial for protecting users from malicious content and preserving the efficiency of communication platforms. In this study, we propose a robust text classification approach using XGBoost, a powerful gradient boosting algorithm, to identify and classify spam messages effectively. Our method leverages the XGBoost algorithm's ability to handle high-dimensional data and nonlinear relationships, making it suitable for text classification tasks. As a feature extraction technique, we employ CountVectorizer to convert raw text messages into numerical vectors, representing the frequency of words in the text. This step ensures the compatibility of the data with the XGBoost algorithm.

Methodology:

1. **Data Collection:** Obtain a dataset containing a collection of text messages labeled as spam or non-spam (ham). Dataset collected from Kaggle.
2. **Data Preprocessing:** Remove any irrelevant information, such as special characters, symbols, or HTML tags, from the messages.Handle any missing or null values in the dataset.
3. **Data Splitting** : Split the dataset into training and testing sets to evaluate the model's performance accurately. Typically, use a 70-30 or 80-20 split for training and testing, respectively.
4. **XGBoost Model Training:** Initialize the XGBoost classifier with suitable hyperparameters. Train the XGBoost model on the training data, using the feature vectors and corresponding labels (spam or non-spam). During training, the model will build an ensemble of decision trees to learn the relationship between the feature vectors and the target labels.
5. **Model Evaluation:** Use the trained XGBoost model to predict the labels of the test data (messages).Evaluate the model's performance using various metrics, such as accuracy.

Process (Boosting algorithm):

1. **Pandas:** A Python library used for data manipulation and analysis. It provides data structures and functions needed to work with structured data, like CSV files, in a way that is both efficient and easy to use.

2. **Seaborn:** A data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
3. **Matplotlib:** A data visualization library in Python that provides a flexible way to create various types of plots and charts.
4. **NumPy:** A fundamental package for numerical computations in Python. It provides support for large, multi-dimensional arrays and matrices.
5. **Train-Test Split:** The process of dividing the dataset into training and testing subsets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance.
6. **Accuracy Score:** A metric used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of the total instances.

XGBoost Algorithm:

XGBoost (Extreme Gradient Boosting) is a popular and powerful gradient boosting algorithm used for both regression and classification tasks. It is an ensemble learning method that combines multiple weak learners (decision trees) to create a strong learner.