Project Report: Data Analysis Using Random Forest Model

Introduction

In this project, the 'Random Forest' model was used to analyze the 'Spambase' dataset and classify messages as 'spam' or 'not spam'. Several steps were followed, starting from loading the data and selecting important features, to training and evaluating the model using multiple metrics.

Explain Code

1. Loading Data, Feature Selection, and Splitting the Data

- Importing Libraries: Essential libraries such as pandas, sklearn, and fetch_ucirepo were imported for data loading and analysis.

- Loading Data: The 'Spambase' dataset was loaded using the ucimlrepo library, which contains features representing the frequency of certain words in messages.

- Defining Features and Target: X was defined as the independent variable containing features, and y as the target variable representing the class (spam or not spam).

- Feature Selection Using Random Forest: A RandomForestClassifier model was trained on all features, followed by SelectFromModel to select the most important features, reducing the number of features from 57 to 24.

- Splitting the Data: Data split into 80% training and 20% testing sets using train_test_split.

- Training the Model: The 'Random Forest' model was trained on the reduced feature set.

- Model Evaluation: Predictions were made on the test set, and accuracy, precision, recall, and F1 score were evaluated.

2: Displaying the Confusion Matrix

- **- Description: This code displays the confusion matrix to analyze the model's performance in detail.**
- **- Results:**

- Correct Classifications: 559 correct 'not spam' and 325 correct 'spam' classifications.

- Misclassifications: 12 'not spam' as 'spam' and 25 'spam' as 'not spam'.

- **- Analysis: The model shows strong performance with minimal errors.**

3: Displaying a Single Decision Tree from the Random Forest Model

- **- Description: Displays one decision tree from the model to illustrate decision-making.**
- **- Results:**

- Root Node: Splits based on feature (e.g., 'capital_run_length_longest').

- Tree Depth: Limited to 2 for simplified visualization.

- Tree Splitting: Uses Gini Index to reduce impurity at each split.

- **- Analysis: Enhances understanding of feature importance in classification.**

Code 4: Additional Evaluation Using MSE and R-squared

- **- Description: Additional metrics for model evaluation.**
- **- Results:**

- MSE: Low value of 0.04, indicating accurate predictions.

- R-squared: 0.82, indicating the model explains 82% of data variance.

- **- Analysis: Confirms model's predictive efficiency.**

Conclusion

The 'Random Forest' model demonstrated excellent performance in classifying messages as 'spam' or 'not spam'. The results were supported by the confusion matrix and a decision tree illustration. The Gini Index improved accuracy, while MSE and R-squared metrics confirmed the model's strength in data interpretation.