Project Report on Data Analysis Using Naive Bayes Model

Introduction

The aim of this project is to classify messages as "spam" or "not spam" using the Naive Bayes model.
The project includes several steps, starting with data import and exploratory analysis, followed by
selecting important features, training the model, and evaluating its performance.

explain the Code:

1. Installing and Importing Libraries

- Objective:

Install the `ucimlrepo` library to download data from the UC Irvine repository and import the necessary libraries for data analysis and model building.

- Details:

This step involves installing the `ucimlrepo` library, essential for downloading the dataset, and importing data analysis libraries (`pandas` and `numpy`), data visualization (`matplotlib`),
and tools from `sklearn` for splitting, classification, and evaluation.

2. Loading the Data

- Objective:

Load the "Spambase" dataset, which contains various features representing the frequency of
specific words in messages and their classifications as "spam" or "not spam".

- Details:

The data is loaded through the `ucimlrepo` library, setting features (`X`) and target variable (`y`) to prepare the data for processing and analysis.

3. Exploratory Data Analysis and Feature Relationships

- Objective:

Understand relationships between features and identify the main patterns in the data to guide
feature selection.

- Details:

We used exploratory data analysis techniques, such as a heatmap, to examine the correlations
between features, helping to identify features that are correlated with each other or the target variable.
This analysis helped in selecting important features later on.

4. Selecting Important Features Using Random Forest

- Objective:

Reduce the number of features and identify the most influential features for classification using
the Random Forest model.

- Details:

The Random Forest model was used to determine which features had the most significant impact
on the target variable, reducing the number of features from 57 to 35 important ones.

5. Splitting the Data into Training and Testing Sets

- Objective:

Split the data into two sets (80% for training and 20% for testing) to evaluate the model's performance on unseen data.

- Details:

The data was randomly split into training and testing sets to ensure that the model's performance could be evaluated on new, unseen data, aiding in assessing its generalization ability.

6. Training the Naive Bayes Model

- Objective:

Train a Gaussian Naive Bayes model using the training data.

- Details:

The Naive Bayes algorithm was applied to learn the patterns and relationships between the features and the target variable, enabling the model to predict the correct class for each message.

7. Predicting Using the Model

- Objective:

Apply the trained model to the test data to predict the target classes.

- Details:

After training, the model was used to make predictions on the test set, allowing us to compare
the predictions with the actual labels to evaluate the performance.

8. Evaluating the Model’s Performance

- Objective:

Measure the model’s efficiency using performance metrics such as accuracy, F1 Score, and
confusion matrix.

Results:
- Accuracy: The model achieved an accuracy of 0.9044 or 90.44%, reflecting a good performance in
classifying messages.
- F1 Score: The F1 Score was 0.9045, indicating a good balance between precision and recall, and
thus an effective model for classification.
- Confusion Matrix:
  - Correctly classified not spam messages: 528

- Incorrectly classified not spam messages as spam: 43
- Correctly classified spam messages: 305
- Incorrectly classified spam messages as not spam: 45

Analysis of Results

These results show that the model performs well in classifying messages. The model is capable of distinguishing between "spam" and "not spam" messages with high accuracy, achieving a good balance between precision and recall, as indicated by the F1 Score. However, there are some misclassifications, suggesting potential areas for further improvement.

Conclusion

This project demonstrates the steps for building a Naive Bayes model to classify messages as "spam" or "not spam". The process included exploratory data analysis, selecting important features, training the model, and evaluating its performance. The final results indicate that the model performs well in predicting the classes.