Data Analysis and Classifier Report: Wine Dataset

Group 3
Members: Matěj Mašata, Pavel Pravec

Introduction

This report provides an analysis of the wine dataset, the process of model training, and the testing outcomes for a classifier designed to predict wine types based on their chemical properties.

Data Description

The dataset comprises 178 instances across three files, detailing 13 features related to wine's chemical composition. The goal is to classify wines into one of three Cultivars from which the individual entries were derived. The specific names of these wine Cultivars are not provided.

Features:

1. Alcohol
2. Malic Acid
3. Ash
4. Alkalinity of Ash
5. Magnesium
6. Total Phenols
7. Flavonoids
8. Non-flavanoid Phenols
9. Proanthocyanins
10. Color Intensity
11. Hue
12. OD280/OD315 of Diluted Wines
13. Proline

The features are quantitative and measure various aspects of wine chemistry that influence flavor, color, and overall quality.

Data Files

- **ALL_wine.csv:** Contains the complete dataset for analysis.
- **TEST_wine.csv:** Reserved for testing the classifier, with 30 instances by default. Can be changed according to input to DataSplit() function.
- **TRAIN_wine.csv:** Used for training the classifier, containing 148 instances by default. Can be changed according to input to DataSplit() function.

Model Selection

A Random Forest Classifier is chosen for this task due to its effectiveness in handling datasets with multiple features and its robustness against overfitting. The classifier is known for its high accuracy in classification tasks and its ability to run efficiently on large datasets.

Code Description

Contained in classifier.py, the code includes functions for data handling and classifier operations:

1. **DataSplit()**: Splits the dataset.
2. **TrainClassifier()**: Trains the classifier using the training dataset, saving parameters for future use.
3. **TestClassifier()**: Tests the classifier's performance and outputs the accuracy.
4. **Classify()**: Classifies the wine based on the features provided.

Model Testing and Metrics Justification

The model's performance was evaluated using the accuracy metric as outputted by the TestClassifier() function, where the model achieved a 95% average accuracy rate. This high accuracy is indicative of the model's effectiveness in classifying the wine types correctly based on the test data.

Conclusion

The dataset is well-suited for the Random Forest classifier, given the variety and type of features. The training process is straightforward, leveraging the robustness of Random Forest, and the testing results are highly satisfactory, reflecting the model's capability to accurately classify the wine types.