Wine Quality Prediction Report

Project Overview

This report details the implementation and evaluation of a machine learning model for predicting wine quality using the "WineQuality_Red.csv" dataset. The dataset contains various physicochemical properties of red wine, with the target variable being the wine quality score (a continuous value). The goal was to build a regression model to predict wine quality and assess its performance using appropriate metrics.

Methodology

The dataset was loaded and preprocessed using Python with libraries such as numpy, sklearn, and csv. The script implemented a **Decision Tree Regressor** from scikit-learn to predict wine quality. Key steps included:

- **Data Loading**: The dataset was read from "WineQuality_Red.csv", with the first column as the target (wine quality) and the remaining columns as features (e.g., acidity, alcohol content).
- **Preprocessing**: No feature selection or low variance filtering was applied in this run (feat_select=0, lv_filter=0), as indicated by the script's global parameters.
- **Model Training**:
    - The data was split into training (65%) and testing (35%) sets using train_test_split.
    - A Decision Tree Regressor was trained with the following parameters: criterion='squared_error' (previously 'mse'), splitter='best', min_samples_split=3, min_samples_leaf=1, and random_state=1.

- **Evaluation Metrics**:
    - **RMSE (Root Mean Squared Error)**: Measures the average magnitude of prediction errors.
    - **Explained Variance**: Indicates the proportion of variance in the target variable explained by the model.

Results

The Decision Tree Regressor was evaluated on the test set, yielding the following results:

- **RMSE**: 0.7179
- **Explained Variance**: 0.1230

Analysis

- **RMSE (0.7179)**: This value indicates the average error in the predicted wine quality scores. Given that wine quality in this dataset typically ranges from 3 to 8 (on a 10-point scale), an RMSE of 0.7179 suggests that the model's predictions are, on average, off by less than one quality point, which is reasonable for a regression task. However, there is room for improvement, as errors of this magnitude can still impact the reliability of predictions in a practical setting.

- **Explained Variance (0.1230)**: This low value indicates that the model explains only 12.3% of the variance in the wine quality scores. This suggests that the Decision Tree Regressor struggles to capture the underlying patterns in the data effectively. A higher explained variance (closer to 1) would indicate better predictive power.
- **Stability**: The script notes that the model's performance is stable, as indicated by consistent RMSE and Explained Variance scores across runs. This stability suggests that the model is not overly sensitive to the random splits in the data.

Discussion

The results highlight both strengths and limitations of the current approach:

- **Strengths**:
    - The model achieves a relatively low RMSE, indicating that predictions are within a reasonable error margin for a regression task.
    - The stability of the performance metrics suggests that the model is reliable across different data splits.
- **Limitations**:
    - The low Explained Variance score (0.1230) indicates that the model fails to capture a significant portion of the variance in wine quality. This could be due to the complexity of the relationship between the features and the target variable, which a single Decision Tree may not fully model.
    - The script notes that the criterion was updated from 'mse' to 'squared_error'. While this change aligns with scikit-learn's updated terminology (as of version 0.24), it does not affect the model's behavior, as both terms refer to the same loss function.

Recommendations

To improve the model's performance, the following steps are recommended:

1. **Feature Selection**: Enable feature selection (feat_select=1) to identify and retain the most relevant features, potentially reducing noise and improving model performance.
2. **Hyperparameter Tuning**: Experiment with the Decision Tree's parameters, such as max_depth, min_samples_split, and min_samples_leaf, to prevent overfitting and improve generalization.

3. **Alternative Models**: Test more robust models like Random Forest or Gradient Boosting Regressors, which can better capture complex relationships in the data.
4. **Cross-Validation**: Enable cross-validation (cross_val=1) to obtain a more reliable estimate of the model's performance across different data splits.
5. **Feature Engineering**: Explore additional preprocessing steps, such as normalizing features (norm_features=1) or creating interaction terms, to better capture relationships between variables.

Conclusion

The Decision Tree Regressor provides a stable but limited solution for predicting wine quality, with an RMSE of 0.7179 and an Explained Variance of 0.1230. While the model achieves reasonable prediction errors, its low Explained Variance indicates that it fails to capture most of the variance in the target variable. Future work should focus on feature selection, hyperparameter tuning, and exploring ensemble methods to enhance predictive performance. This project demonstrates a foundational approach to regression tasks and provides a starting point for more advanced modeling techniques.