Breast Cancer Classification using Random Forest and Cross-Validation

Overview

This script implements a Random Forest Classifier to classify breast cancer cases using the Breast Cancer Dataset from sklearn. It evaluates the model using k-fold cross-validation, optimizes hyperparameters using GridSearchCV, and reports final performance metrics on a test set.

Steps in the Code

1. Load Dataset and check the features - The dataset is loaded using load_breast_cancer(), which provides real-world breast cancer data. We also display the features and labels.
2. Data Splitting and plotting - The dataset features are plotted then split into features and labels then into 80% training and 20% testing using train_test_split(), ensuring stratification to maintain class balance.
3. Cross-Validation - A 5-fold cross-validation is performed using cross_val_score() to assess model performance before hyperparameter tuning.
4. Hyperparameter Tuning - GridSearchCV is used to search for the best combination of hyperparameters:
    ○ Number of trees (n_estimators)
    ○ Maximum tree depth (max_depth)
    ○ Minimum samples required for a split (min_samples_split)
    ○ Minimum samples required per leaf (min_samples_leaf)

5. Model Evaluation - The best model from GridSearchCV is selected and evaluated on the test set. Performance metrics include:

    ○ **Accuracy**: Measures overall correctness in prediction.
    ○ **Precision**: Indicates the proportion of true positives among predicted positives.
    ○ **Recall / sensitivity**: Measures the ability to detect all positive cases in the data.
    ○ **F1-Score**: A balance between precision and recall.
    ○ **Specificity**: proportion of actual negative cases correctly identified as negative.

Final Output

The script prints:

- Best hyperparameters
- Cross-validation accuracy
- Test set performance metrics (Accuracy, Precision, Recall, F1-score)
- Plots of target and feature distributions, feature importance and learning curves.

Conclusion

This approach ensures robust model selection by preventing overfitting through cross-validation, fine-tuning hyperparameters and monitoring curves for optimal performance. Github link here