Pipeline Optimization

Mohammad Irfan Uddin

Introduction:

In this experiment, we aim to develop and optimize machine learning models to predict the quality of wines based on various features. The problem involves a classification task, where the quality of wine is the target variable, and we will explore the use of different machine learning algorithms and hyperparameter tuning techniques to achieve optimal predictive performance.

Dataset Description:

The dataset under consideration, the Wine Quality Dataset, is composed of 11 features and contains 1599 samples. The target variable is wine quality, which is a categorical attribute. One notable aspect of this dataset is the absence of missing values, simplifying the preprocessing phase. From dataset, we establish a clear understanding of the dataset's characteristics and set the stage for subsequent model selection.

Experimental Setup:

The pipeline begins with data preprocessing, where missing values are imputed, numeric features are scaled, and categorical features are encoded. This organized preprocessing is enclosed within a ColumnTransformer, harmonizing the treatment of different feature types. Subsequently, the pipeline includes either a RandomForestClassifier or an SVM classifier based on the optimization process. Hyperparameters of these classifiers, as well as preprocessing steps, are systematically tuned using Optuna. The pipeline automates the training of models, making it convenient to compare the accuracy of the base model with default parameters against the optimized model with tuned hyperparameters. For both the classifier and the preprocessing steps I have used Optuna for HPO, with a budget of 200 trials. Preprocessing steps include imputation and scaling strategies for both numeric and categorical features.

Hyperparameters Ranges:

For Random Forest:

n_estimators: (10, 200)

max_depth: (2,40)

For SVC:

C: (1e-5, 1e5)

Gamma: (1e-5, 1e5)

Hyperparameters for numeric transformer:

num_imputer_strategy : ['mean', 'median', 'most_frequent']

num_scaler: ['StandardScaler', 'MinMaxScaler']

Hyperparameters for categorical transformer:

cat_imputer_strategy: ['most_frequent', 'constant']

cat_encoder_handle_unknown: ['error', 'ignore']

Results:

Model	Accuracy	
	Before	ML_Pipe (Optuna)
Random Forest	0.68	0.75
SVM	0.5	0.63

Above Bayesian and GridSearch accuracies are from other HPO exercise, for which I have used same dataset and runtimes are close to one another.

Tuned Hyperparameter with Optuna:

Random Forest:

n_estimators: 103, max_depth: 11, num_imputer_strategy: most_frequent, num_scaler: StandardScaler, cat_imputer_strategy: most_frequent, cat_encoder_handle_unknown: ignore

SVM:

C: 12957, kernel: kbf