Hyperparameter Optimization

Mohammad Irfan Uddin

Introduction:

The objective of this report is to explore Hyperparameter Optimization (HPO) with the goal of improving the predictive performance of machine learning models. To accomplish this, we apply different machine learning algorithms and conduct a meticulous tuning of their hyperparameters to identify the most accurate predictive models.

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 practical implementation of the HPO approach begins with the selection of programming languages and libraries. In this case, Python is the chosen language, and we utilize essential libraries such as Pandas for data manipulation, Scikit-learn for machine learning, SciPy for statistical analysis, GridSearchCV for finding the optimal parameter values from a given set of parameters in a grid and Warnings for managing alerts. The dataset is loaded and subsequently divided into training and testing sets through an 80-20 split.

Four machine learning algorithms are considered for evaluation: Random Forest, Support Vector Machine (SVM), Logistic Regression, and Decision Tree. To assess their performance, we employ a 5-fold cross-validation strategy, with the primary evaluation metric being accuracy. Furthermore, statistical tests are used to compare the performance of these models to select the most suitable algorithms for further analysis.

Hyperparaeter ranges for Grid Search:

Logistic Regression:

C: [0.1, 0.5, 1, 5, 10]

penalty: ['11', '12']

solver : ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],

Decision Tree:

max_depth: [None, 10, 20, 30, 50, 100]

min_samples_split: [2, 5, 10, 15]

min_samples_leaf': [1, 2, 4, 7]

Random Forest:

n_estimators: [10, 50, 100, 200]

max_depth: [None, 10, 20, 30]

min_samples_split: [2, 5, 10],

min_samples_leaf': [1, 2, 4]},

SVM:

C: [0.1, 0.8, 2, 10],

kernel: [‘linear', 'rbf', 'poly', 'sigmoid']

Hyperparameter ranges for BO:

'SVM':

C: Real(1e-5, 1e+5)

kernel: ['linear', 'rbf', 'poly', 'sigmoid']

Random Forest:

n_estimators': (10, 200),

max_depth': (10, 30),

min_samples_split': (1, 10),

min_samples_leaf': (1, 10)

Logistic Regression:

C: 91e-6, 1e+6)

penalty:['11', '12']

solver:['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']

Decision Tree:

max_depth: (1, 100)

min_samples_split: (2, 15)

min_samples_leaf: Integer(1, 7)

Results:

The results of the model evaluation are summarized as follows:

Model	Accuracy		
	Before_HPO	Bayesian_HPO	GridSearch_HPO
Random Forest	0.68	0.69	0.73
SVM	0.5	0.6	0.64
Logistic Regression	0.62	0.62	0.63
Decision Tree	0.64	0.64	0.66

Tuned Hyperparameter:

Decision Tree		
Hyperparameter	Bayesian	Grid Search
max_depth	24	None
min_samples_leaf	2	1
min_samples_split	11	2

SVM		
Hyperparameter	Bayesian	Grid Search
C	50603.55	0.5
Kernel	kbf	linear

Random Forest		
Hyperparameter	Bayesian	Grid Search
max_depth	21	30
min_samples_leaf	2	1
min_samples_split	4	5
n_estimators	46	100

Logistic Regression		
Hyperparameter	Bayesian	Grid Search
C	2.35	0.5
Penalty	None	12
Solver	newton-cg	newton-cg

Bonferroni-Dunn test results:

group1	group2	meandiff	p-adj	lower	upper
Decision Tree	Random Forest	0.0888	0.0007	0.0382	0.1394
Decision Tree	SVM	-0.0069	0.9373	-0.0575	0.0437
Random Forest	SVM	-0.0957	0.0003	-0.1463	-0.0451

Budget:

For Bayesian HPO, I have run the optimization with 10 iterations and it took about 27 minutes to run. I have tried with more iterations but Google Collab got disconnected, it didn't even run for 12 iterations.

For GridSerach, it took 22 minutes to run the code.

For three out of the four models, the accuracy has increased whereas the accuracy for logistic regression has remained almost same. The accuracy for SVM has increased by more than 26% after HPO.

Grid Search Evaluations Number:

Model	Evaluations
Random Forest	144
Logisitic Regression	50
SVM	16
Decision Tree	96
Total	306