Vision and Language Group IITR

Project

AutoML

Prepared by:

Vishal Rao

1. INTRODUCTION

1. Hyperparameter optimization is crucial for improving machine learning models.
2. Hyperparameters are settings that control how models learn from data, set before training begins.
3. They differ from model parameters, which are adjusted during training based on data.
4. Effective tuning of hyperparameters can significantly enhance a model's performance.
5. This document focuses on Bayesian Optimization and Tree-structured Parzen Estimator (TPE) methods for optimizing hyperparameters, comparing their learning efficiency.

2. Hyperparameters

1. Hyperparameters are settings that adjust how machine learning models are trained.
2. They're fixed before training and aren't learned from the data like model parameters.
3. Hyperparameters steer how the training process behaves.
4. They impact how well the model performs, how quickly it learns, and how well it applies what it learns to new data.

Dataset and Preprocessing

Dataset: The analysis is based on the 'diabetes.csv' dataset.

Features and Target:

- **Features:** All columns except 'Outcome'.
- **Target:** 'Outcome' column.

Preprocessing:

1. **Numeric Features:** Standard Scaler
2. **Categorical Features:** One-Hot Encoder

Model and Pipeline

Model: GradientBoostingClassifier (GBC) with `loss='log_loss'`.

Pipeline: Combines preprocessing steps and the classifier.

Hyperparameter Optimization Methods

1. Randomized Search CV

Param Distribution:

- classifier__n_estimators: 50 to 200
- classifier__learning_rate: 0.01 to 0.3
- classifier__min_samples_leaf: 1 to 20
- classifier__max_features: ['sqrt', 'log2', None]

Results:

Evaluates 75 parameter combinations with 5-fold cross-validation using ROC AUC as the scoring metric.

2. Hyperopt

Search Space:

- learning_rate: Log-uniform distribution (0.01, 0.3)
- n_estimators: 50 to 200
- min_samples_leaf: 1 to 10
- max_features: Choice among 'sqrt', 'log2', None

Objective Function:

Minimize negative mean ROC AUC using 5-fold cross-validation.

Results:

75 iterations of evaluations.

3. Bayesian Optimization

Search Space:

- learning_rate: 0.01 to 0.3
- n_estimators: 50 to 200
- min_samples_leaf: 1 to 10

Objective Function:

Maximize mean ROC AUC using 5-fold cross-validation.

Results:

75 iterations, 20 initial random points.

Evaluation Metrics

Metric: ROC AUC Score

Evaluation Method: Predict probabilities on the test set and compute the ROC AUC score.

Results

Optimization Method	ROC AUC Score on Test Set
Randomized Search	0.812
Hyperopt	0.748
Bayesian Optimization	0.835

```
100%|██████████| 75/75 [00:48<00:00,  1.53trial/s, best loss: -0.8479562466674869]
```

iter	target	learni...	min_sa...	n_esti...
1	0.8392	0.1309	7.483	50.02
2	0.8378	0.09768	2.321	63.85
3	0.8455	0.06402	4.11	109.5
4	0.8373	0.1663	4.773	152.8
5	0.8395	0.06929	8.903	54.11
6	0.8338	0.2044	4.756	133.8
7	0.8425	0.05071	2.783	170.1
8	0.8296	0.2908	3.821	153.8
9	0.8335	0.2642	9.051	62.76
10	0.8394	0.02133	2.528	181.7
11	0.8466	0.03852	4.79	193.7
12	0.841	0.1646	7.227	97.33
13	0.8421	0.2091	8.512	52.74
14	0.8285	0.2275	9.9	162.2
15	0.8435	0.09133	8.104	65.48
16	0.8384	0.1399	9.177	94.04
17	0.8387	0.09345	2.17	52.91
18	0.8367	0.2069	2.905	89.83
19	0.8385	0.1526	1.48	136.1
20	0.8477	0.05255	6.304	155.0
21	0.8301	0.2642	6.731	154.5
22	0.8384	0.1644	8.086	76.23
23	0.8396	0.1828	3.349	162.0
24	0.8403	0.07598	2.4	192.1
25	0.8415	0.2206	7.558	79.2

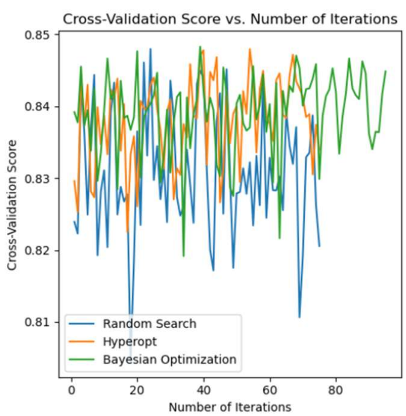
O

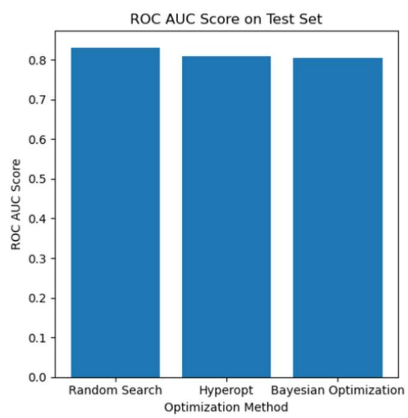
Observations

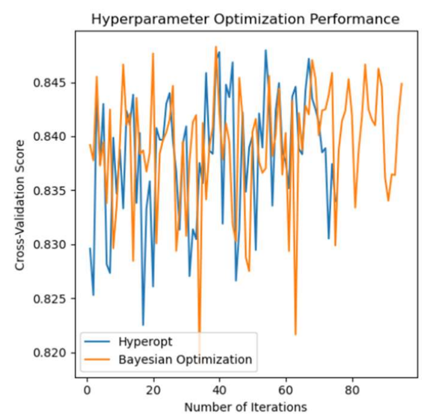
1. **Random Search CV:** Achieved a ROC AUC score of 0.812 on the test set.
2. **Hyperopt:** Achieved a ROC AUC score of 0.748 on the test set.
3. **Bayesian Optimization:** Achieved a ROC AUC score of 0.835 on the test set, showing the best performance among the three methods.

Plots and Analysis

1. **Cross-Validation vs. Number of Iterations:**
   - Plot showing the cross-validation score against the number of iterations for all three methods.
2. **ROC AUC on Test Set:**
   - Bar plot comparing the ROC AUC scores on the test set for Random Search, Hyperopt, and Bayesian Optimization.
3. **Hyperparameter Optimization Performance:**
   - Detailed performance plots for Hyperopt and Bayesian Optimization.







Conclusion

- **Bayesian Optimization** is a robust method for hyperparameter tuning, effectively balancing exploration and exploitation to achieve superior results with fewer iterations.
- **Hyperopt** and **Tree-structured Parzen Estimator (TPE)** are effective but generally show slightly lower performance compared to Bayesian Optimization.
- Systematic hyperparameter tuning significantly enhances model performance.