Hyperparameter Tuning for Cross-Validation

Group Week 04
-
Advanced Topic 5

Agenda

- Objectives
- Theory
- Results
- Conclusions

Theory

Cross-validation

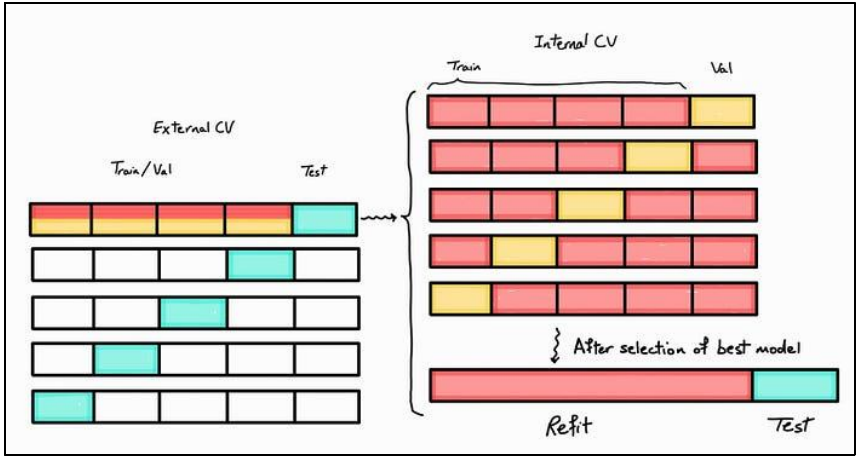

Fig.1 Cross validation methodology.

- Evaluation performance technique.
- Main idea:
  - Split the data set into <u>different subsets of</u> **training** and **validation** sets.
  - Train and evaluate the model **repeatedly** with different subsets of training and validation sets respectively.
  - Average the performance from all the condutcted train-validation process.

Theory

Hyperparameters

- Parameters of a machine learning model that **are not learned during training**, <u>but are set before training begins.</u>
  - Examples:
    - Learning rate and number of layers in neural network
    - Depth and width of a decision tree
    - Regularization strength of a linear regression model
    - Initial population, crossover rate, and mutation rate of the genetic algorithm.

- **Parameters is not the same as Hyperparameters.** Parameters are learned from the data during the training process.
  - Examples:
    - Weights in a neural network
    - Coefficient in linear regression

Theory

Hyperparameters v. Parameters

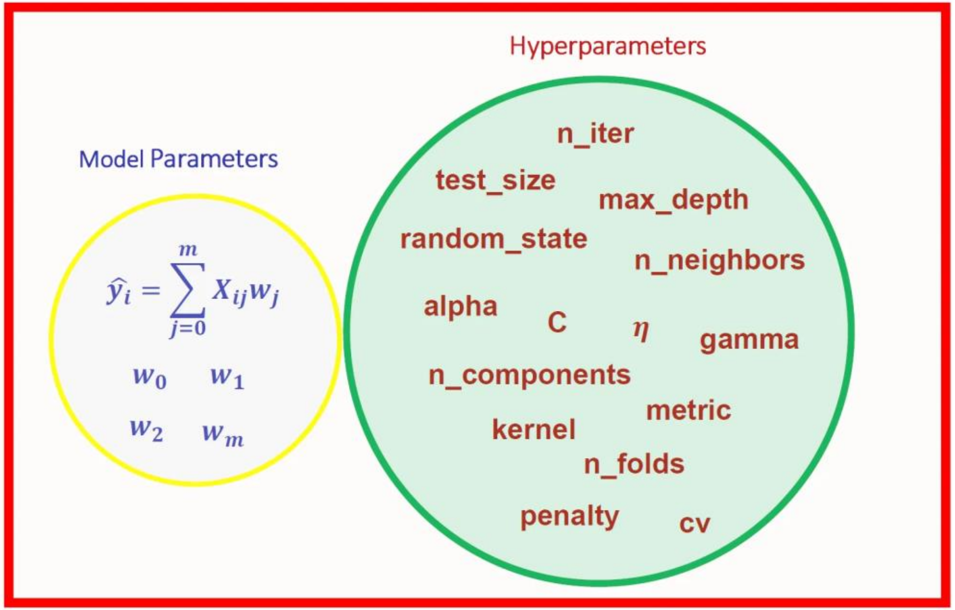

Fig.2 Comparison of model parameters and hyperparameters
(**Source:** https://towardsdatascience.com/model-parameters-and-hyperparameters-in-machine-learning-what-is-the-difference-702d30970f6).

Theory

Hyperparameters tuning

- Hyperparameter tuning can profoundly influence a model's performance.
  - Learning rate helps increase the speed of model learning.
  - Regularization parameters (L1/L2) help preventing overfitting/underfitting.
  - Batch size for training helps for resource management.
  - etc.
- Poor hyperparameter tuning could also lead to instability, inaccuracy, and overall affect the effectiveness of a machine learning solution.

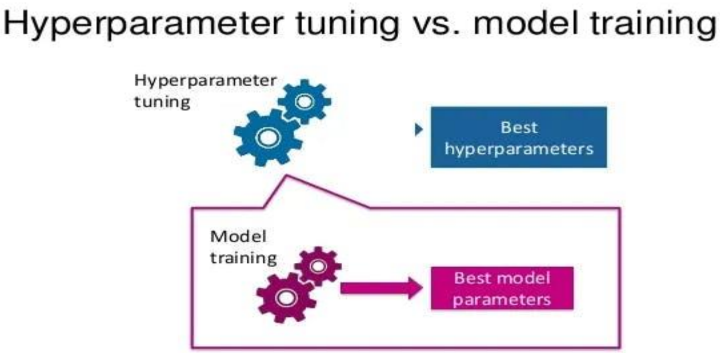

Fig.3 How hyperparameter tuning affect the model training.

Theory

Hyperparameters tuning (cross-validation)

- **Main goal:**
  Find the combination of hyperparameters that produces the **best performance on the validation data.**
- Main methods used for hyperparameter tuning:
  - Grid Search
  - Random Search
  - Bayesian Optimization

Theory

Hyperparameters tuning
(Grid search)

- Grid of hyperparameter values is defined first.
- The model is trained and evaluated for each combination of hyperparameters in the grid.
- The combination of hyperparameters that results in the best performance is then selected as the **optimal parameters.**

Theory

Hyperparameters tuning (Random search)

- Hyperparameters are randomly sampled from a specified distribution.
- The model is trained and evaluated for each set of hyperparameters.
- The hyperparameters set that results in the best performance is then selected as the **optimal parameters.**

Theory

Hyperparameters tuning (Bayesian optimization)

- A probabilistic model is used to model the relationship between the hyperparameters and the model's performance.
- The model is updated as new evaluations are made.
- The next set of hyperparameters to be evaluated is selected based on the current model.

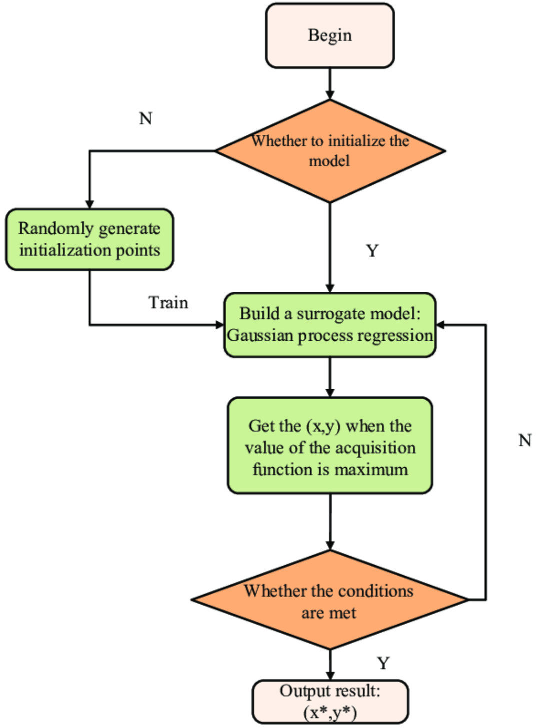

Fig.4 Scheme of bayesian optimization.

Results

Hyperparameters tuning

- Dataset:
  Wine data set from *scikit-learn.*

- Problem type:
  Classification

- Method:
  Random Forest Classifier

- Hyperparameter:
  - `max_depth`
  - `max_features`
  - `min_samples_leaf`
  - `min_samples_split`
  - `n_estimators`

Fig.1……………………

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0

Fig.5 Model's dataset.

Check the code here!

Results

Hyperparameters tuning (Grid search)

- Best fitting score: 0.97857

- Best hyperparameters:
  - max_depth : None
  - max_features : sqrt
  - min_samples_leaf : 1
  - min_samples_split: 2
  - n_estimators : 100

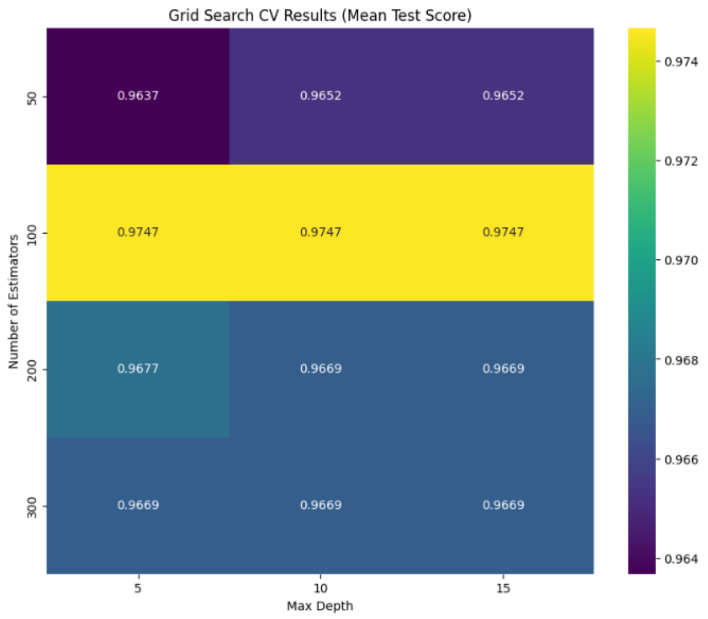

Fig.6 Grid search results.

Results

Hyperparameters tuning (Random search)

- Best fitting score: 0.97857

- Best hyperparameters:
  - ```max_depth   : 5```
  - ```max_features   : log2```
  - ```min_samples_leaf : 1```
  - ```min_samples_split: 10```
  - ```n_estimators   : 98```

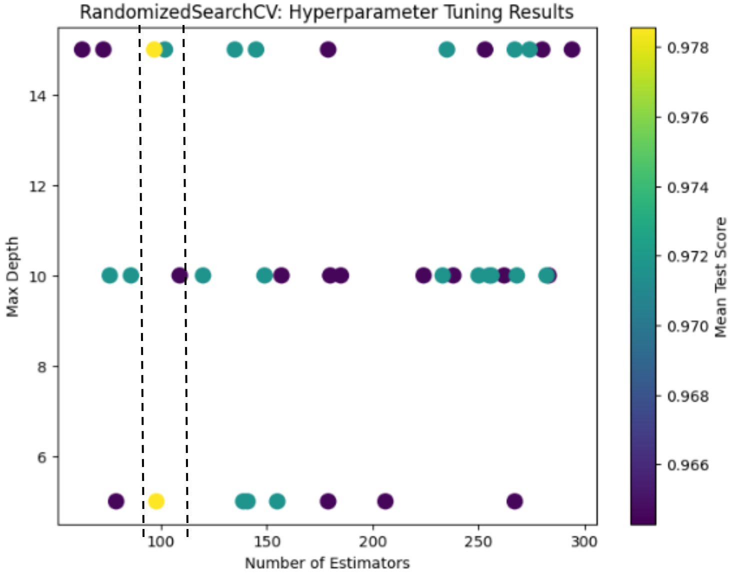

Fig.7 Randomized search results.

Results

Hyperparameters tuning (Bayesian optimization)

- Best fitting score: 0.97857

- Best hyperparameters:
  - `max_depth        : 15`
  - `max_features     : log2`
  - `min_samples_leaf : 1`
  - `min_samples_split: 2`
  - `n_estimators     : 111`

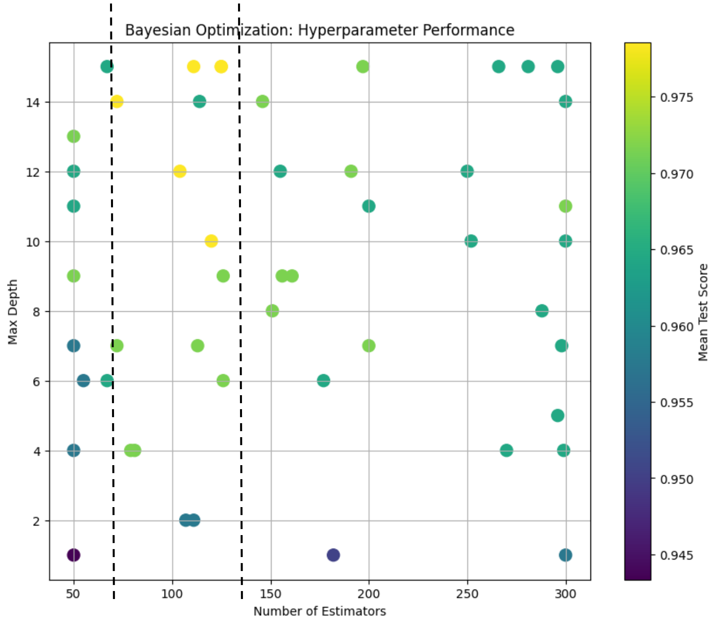

Fig.8 Bayesian optimization results.

Conclusions

- Cross validation and hyperparameter tuning are important techniques for improving the performance of machine learning models.
- Hyperparameter tuning helps to find the optimal set of hyperparameters for a model.
- **Grid Search:**
  - Pros: Easy to implement.
  - Cons: Exhaustive, can be computationally expensive.
- **Random Search:**
  - Pros: Faster than grid search, scalable.
  - Cons: No guarantees of optimal solution.
- **Bayesian Search:**
  - Pros: Using probabilistic models or heuristic, making search more coefficient.
  - Cons: Computational overhead → Not justifed for simple tuning tasks.

Thank you!