HYPERPARAMETER TUNNING

Understanding Parameters and Hyperparameters in Machine Learning :

Parameters :

Parameters are the internal variables of a model that are learned from the data during the training process. They define the model's representation of the underlying patterns in the data.

For example:

- **In a linear regression model, the parameters are the coefficients of the predictors.**
- **In a neural network, the parameters are the weights and biases of the nodes.**
- **In a decision tree, the parameters are the split points and split criteria at each node.**

The goal of the training process is to find the optimal values for these parameters, which minimize the discrepancy between the model's predictions and the actual outcomes.

Hyperparameters :

In machine learning, hyperparameters are parameters whose values are set before the learning process begins. These parameters are not learned from the data and must be predefined. They help in controlling the learning process and can significantly influence the performance of the model. In a neural network, hyperparameters might include the learning rate, the number of layers in the network, or the number of nodes in each layer.

- **In a support vector machine, the regularization parameter C or the kernel type can be considered as hyperparameters.**
- **In a decision tree, the maximum depth of the tree is a hyperparameter. The best values for hyperparameters often cannot be determined in advance, and must be found through trial and error.**

Why the word 'hyper'?

The choice of the word is primarily a naming convention to differentiate between the two types of values (internal parameters and guiding parameters) that influence the behaviour of a machine learning model.

HYPERPARAMETER TUNNING

What is Hyperparameter Tuning?

Hyperparameter tuning refers to the process of optimizing the hyperparameters of a machine learning algorithm to achieve the best possible performance. Since hyperparameters are predefined values that are not learned during training, we adjust and fine-tune them to find the optimal settings for a given problem. The goal is to maximize the algorithm's accuracy, minimize error, or optimize another performance metric based on the task.

Techniques for Hyperparameter Tuning :

1.GridSearchCV :

GridSearchCV is an exhaustive search method where you specify a grid of hyperparameters and the algorithm tests all possible combinations to find the best one.

How it works:

- **You define a range of values for each hyperparameter you want to tune.**
- **The model is trained and evaluated for each combination of these hyperparameters using cross-validation.**
- **The combination that performs best is selected.**

Pros:

- **Simple to implement.**
- **Guarantees finding the best combination within the defined grid.**

Cons:

- **Computationally expensive, especially with large grids or high-dimensional data.**
- **Time-consuming as it evaluates all combinations, even unlikely ones.**

2.RandomSearchCV

RandomSearchCV is a more efficient approach where hyperparameter values are randomly sampled from the specified ranges for a fixed number of iterations.

How it works:

- **You specify the ranges for hyperparameters and the number of iterations to test.**
- **The algorithm selects random combinations within those ranges.**
- **It evaluates and returns the best combination.**

Pros:

- **Faster than GridSearchCV since it doesn't evaluate all combinations.**
- **Suitable for large hyperparameter spaces.**

Cons:

- **May miss the best combination if it isn't randomly selected during the search.**

3.Advanced Techniques :

Advanced techniques use more sophisticated algorithms to efficiently search for the best hyperparameters. These include:

a. Bayesian Optimization

- **Uses probability models to predict the performance of hyperparameter combinations.**
- **Focuses on promising areas of the hyperparameter space, reducing the number of evaluations.**
- **Libraries: *Hyperopt*, *Optuna*.**

b. Tree-structured Parzen Estimators (TPE)

- **A type of Bayesian Optimization that models the objective function and focuses on the most promising hyperparameter combinations.**
- **Often used in conjunction with frameworks like *Optuna* or *Hyperopt*.**

c. Genetic Algorithms

- **Inspired by the process of natural selection, where "populations" of hyperparameter combinations evolve to improve performance over generations.**
- **Libraries: *TPOT*, *DEAP*.**

d. Population-Based Training (PBT)

- **Combines exploration and exploitation by continuously updating hyperparameters during training.**
- **Often used in deep learning frameworks like TensorFlow.**

e. Automated Machine Learning (AutoML)

- **Tools like *Auto-sklearn*, *H2O AutoML*, or *Google AutoML* automate hyperparameter tuning along with other aspects of model selection.**

Key Considerations

- **Start with RandomSearchCV if you have limited computational resources.**
- **Use GridSearchCV for small hyperparameter spaces or when you want exhaustive evaluation.**
- **Consider Bayesian Optimization or AutoML for complex models or larger datasets.**