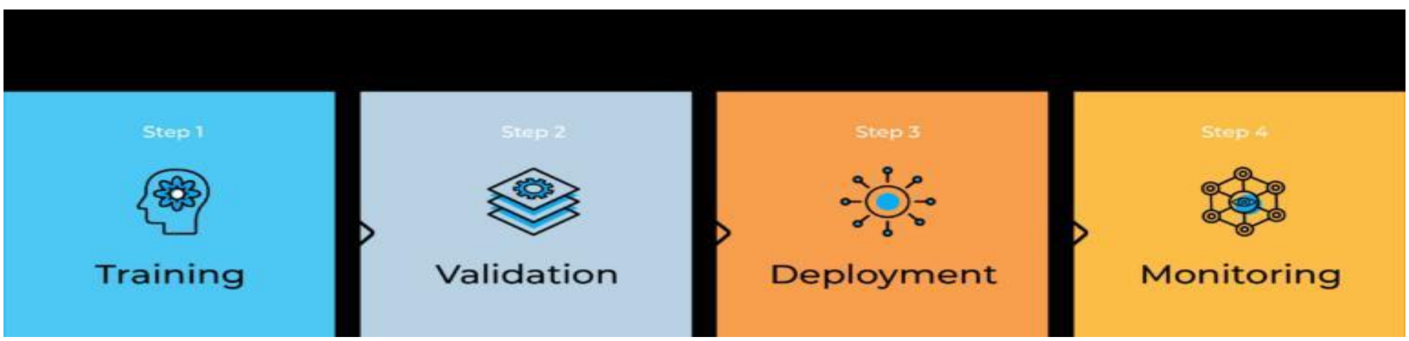
Machine Learning Model Deployment with IBM Cloud Watson Studio

Ensemble methods or hyperparameter tuning to optimize the model's performance.

Deploying a machine learning model, known as model deployment, simply means to integrate a machine learning model and integrate it into an existing production environment (1) where it can take in an input and return an output.

This process typically involves several steps, including selecting an appropriate web framework, setting up the necessary infrastructure, and integrating the model with the web application. Once the model is deployed, it can be used to make predictions or provide other intelligent services to web app users.



Monitoring the Model

Finally, monitoring and maintaining the machine learning model is the last step of the cycle. This process entails constantly auditing model artifacts and reiterating to check if the model is making reasonable predictions.

The key idea here is to identify, assess, and manage any issues post-model deployment.

Hyperparameter optimization:

Before I define hyperparameter optimization, you need to understand what a hyperparameter is.

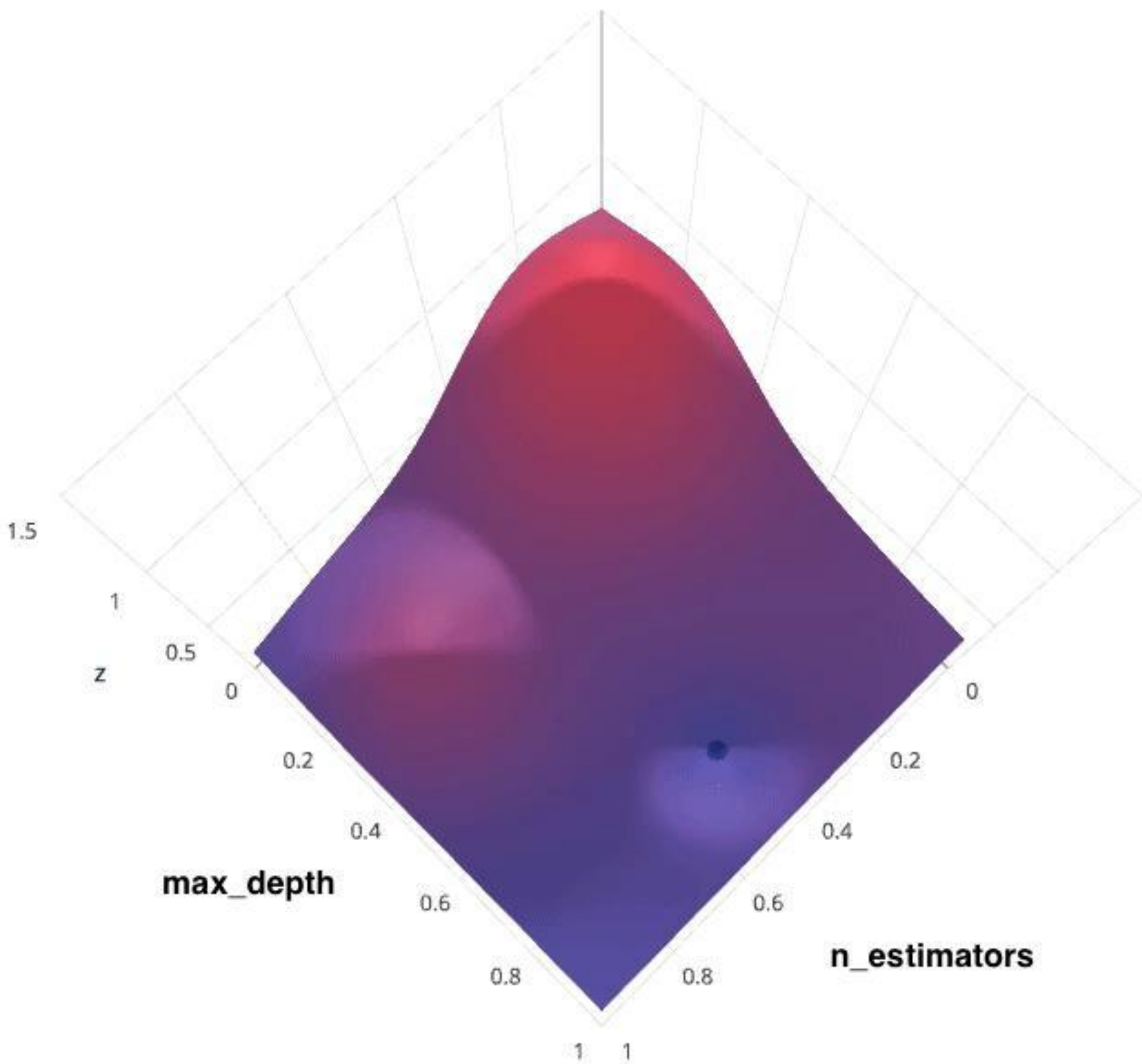
In short, hyperparameters are different parameter values that are used to control the learning process and have a significant effect on the performance of machine learning models.

An example of hyperparameters in the Random Forest algorithm is the number of estimators (n_estimators), maximum depth (max_depth), and criterion. These parameters are tunable and can directly affect how well a model trains.

So then hyperparameter optimization is the process of finding the right combination of hyperparameter values to achieve maximum performance on the data in a reasonable amount of time.

This process plays a vital role in the prediction accuracy of a machine learning algorithm. Therefore Hyperparameter optimization is considered the trickiest part of building machine learning models.

Most of these machine learning algorithms come with the default values of their hyperparameters. But the default values do not always perform well on different types of Machine Learning projects. This is why you need to optimize them in order to get the right combination that will give you the best performance.



Alternative Hyperparameter Optimization techniques

Now I will introduce you to a few alternative and advanced hyperparameter optimization techniques/methods. These can help you to obtain the best parameters for a given model.

We will look at the following techniques:

1.Hyperopt

2.Scikit Optimize

3.Optuna

1. Ray Tune

Ray provides a simple, universal API for building distributed applications. Tune is a Python library for experiment execution and hyperparameter tuning at any scale. Tune is one of the many packages of Ray. Ray Tune is a Python library that speeds up hyperparameter tuning by leveraging cutting-edge optimization algorithms at scale.

2. Optuna

Optuna is designed specially for machine learning. It’s a black-box optimizer, so it needs an objective function. This objective function decides where to sample in upcoming trials, and returns numerical values (the performance of the hyperparameters). It uses different algorithms, such as GridSearch, Random Search, Bayesian and Evolutionary algorithms to find the optimal hyperparameter values.

3. HyperOpt

From the official documentation, Hyperopt is a Python library for serial and parallel optimization over awkward search spaces, which may include real-valued, discrete, and conditional dimensions.

Hyperopt uses Bayesian optimization algorithms for hyperparameter tuning, to choose the best parameters for a given model. It can optimize a large-scale model with hundreds of hyperparameters.

4. Scikit-Optimize

Scikit-Optimize is an open-source library for hyperparameter optimization in Python. It was developed by the team behind Scikit-learn. It’s relatively easy to use compared to other hyperparameter optimization libraries.

It has sequential model-based optimization libraries known as Bayesian Hyperparameter Optimization (BHO). The advantage of BHO is that they find better model settings than random search in fewer iterations.

Hyperparameter optimization libraries (free and open source):

Ray.tune: Hyperparameter Optimization Framework

Optuna

Hyperopt

Polyaxon

Talos

BayesianOptimization

Metric Optimization Engine

Spearmint

GPyOpt

Scikit-Optimize

Hyperparameter optimization libraries (everybody’s favorite commerial library):

SigOpt

Implementation examples:

Bayesian optimisation for smart hyperparameter search

Bayesian optimization with scikit-learn

Bayesian optimization with hyperopt

Jeremy Jordan

Conclusion

Hyperparameter tuning is an essential aspect of training and optimizing machine learning models. This iterative process involves selecting and refining the settings of the learning algorithm that will impact the model's performance, such as learning rate, epoch number, and regularization parameters. Tuning can be performed manually, using a grid search, a random search, or through automated methods like Bayesian optimization.