Hyperparameter Tuning

M.SAXENA

Hyperparameter Tuning

Hyperparameter tuning is an important step in machine learning model development. Hyperparameters are the parameters that are set before the learning process begins and cannot be learned from the data. They control the behavior of the learning algorithm and can significantly impact the performance of the model.

M.SAXENA

Here are some common techniques for hyperparameter tuning:

Manual Search

You can manually specify different values for hyperparameters and evaluate the model's performance for each combination. This approach can be time-consuming and may not cover all possible combinations, but it provides a good starting point.

Grid Search

Grid search involves defining a grid of hyperparameter values and exhaustively searching through all possible combinations. It evaluates the model's performance for each combination and selects the one with the best performance. Grid search is a simple and systematic approach but can be computationally expensive for large hyperparameter spaces.

M.SAXENA

Random Search

Random search randomly samples hyperparameter values from predefined ranges. It performs multiple evaluations and selects the best-performing combination. Random search is less computationally intensive compared to grid search and can be more efficient when the hyperparameter space is large.

M.SAXENA

Bayesian Optimization

Bayesian optimization uses probabilistic models to model the performance of the machine learning model as a function of the hyperparameters. It iteratively explores the hyperparameter space by selecting new samples based on the previous evaluations. Bayesian optimization is more efficient than grid and random search and can often find good solutions with fewer evaluations.

M.SAXENA

Automated Hyperparameter Tuning

There are libraries and frameworks available that provide automated hyperparameter tuning algorithms, such as scikit-learn's GridSearchCV or RandomizedSearchCV, Optuna, or Keras Tuner. These tools streamline the process and handle the optimization internally, allowing you to define the hyperparameter search space and evaluation criteria.

M.SAXENA

Cross-Validation

When performing hyperparameter tuning, it's important to use cross-validation to get a robust estimate of the model's performance. Cross-validation involves splitting the data into multiple subsets, training the model on a portion of the data, and evaluating it on the remaining subset. By repeating this process with different subsets, you can get a more reliable estimate of the model's performance for a given set of hyperparameters.

M.SAXENA

Some examples of model hyperparameters

- The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
- The learning rate for training a neural network.
- The C and sigma hyperparameters for support vector machines.
- The k in k-nearest neighbors.

Follow me

M.Saxena