Related Articles

Hyperparameters of Random Forest Classifier

Last Updated : 22 Jan, 2021

In this article, we are going to learn about different hyperparameters that exist in a Random Forest Classifier. We have already learnt about the implementation of Random Forest Classifier using scikit-learn library in the article https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/.

Hyperparameters are configurations that cannot be learnt from the regular data that we provide to the algorithm, these are inbuilt to the algorithm and each algorithm has its own predefined set of hyperparameters. Hyperparameters are often tuned for increasing model accuracy, and we can use various methods such as GridSearchCV, RandomizedSearchCV as explained in the article https://www.geeksforgeeks.org/hyperparameter-tuning/.

A deep understanding of hyperparameters is required because they are responsible for deciding how quickly a model can fit onto the data to produce accurate results. On the other hand, not finding the optimal values of hyperparameters can also result in less accuracy because of overfitting issue. Therefore, we will be having a closer look at the hyperparameters of random forest classifier to have a better understanding of the inbuilt hyperparameters:

- **n_estimators:** We know that a random forest is nothing but a group of many decision trees, the n_estimator parameter controls the number of trees inside the classifier. We may think that using many trees to fit a model will help us to get a more generalized result, but this is not always the case. However, it will not cause any overfitting but can certainly increase the time complexity of the model. **The default number of estimators is 100** in scikit-learn.
- **max_depth:** It governs the maximum height upto which the trees inside the forest can grow. It is one of the most important hyperparameters when it comes to increasing the

accuracy of the model, as we increase the depth of the tree the model accuracy increases upto a certain limit but then it will start to decrease gradually because of overfitting in the model. It is important to set its value appropriately to avoid overfitting. **The default value is set to None,** None specifies that the nodes inside the tree will continue to grow until all leaves become pure or all leaves contain less than **min_samples_split** (another hyperparameter).

- **min_samples_split:** It specifies the minimum amount of samples an internal node must hold in order to split into further nodes. If we have a very low value of min_samples_splits then, in this case, our tree will continue to grow and start overfitting. By increasing the value of min_samples_splits we can decrease the total number of splits thus limiting the number of parameters in the model and thus can aid in reducing the overfitting in the model. However, the value should not be kept very large that a number of parameters drop extremely causing the model to underfit. We generally keep min_samples_split value between 2 and 6. **However, the default value is set to 2.**
- **min_samples_leaf:** It specifies the minimum amount of samples that a node must hold after getting split. It also helps to reduce overfitting when we have ample amount of parameters. Less number of parameters can lead to overfitting also, we should keep in mind that increasing the value to a large number can lead to less number of parameters and in this case model can underfit also. **The default value is set to 1.**
- **max_features:** Random forest takes random subsets of features and tries to find the best split. max_features helps to find the number of features to take into account in order to make the best split. It can take four values "**auto**", "**sqrt**", "**log2**" and **None**.
  - In case of auto: considers max_features = sqrt(n_features)
  - In case of sqrt: considers max_features = sqrt(n_features), it is same as auto
  - In case of log2: considers max_features = log2(n_features)
  - In case of None: considers max_features = n_features
- **max_leaf_nodes:** It sets a limit on the splitting of the node and thus helps to reduce the depth of the tree, and effectively helps in reducing overfitting. If the value is set to None, the tree continues to grow infinitely.
- **max_samples:** This hyperparameter helps to choose maximum number of samples from the training dataset to train each individual tree.

These are the major hyperparameters that are present implicitly in the random forest classifier which is required to be tuned in order to increase the accuracy of our training model.

Attention reader! Don’t stop learning now. Get hold of all the important Machine Learning Concepts with the **Machine Learning Foundation Course** at a student-friendly price and become industry ready.

Like 0

Previous Next

ADVERTISEMENT BY ADRECOVER

RECOMMENDED ARTICLES

Page : 1 2 3

01 IBM HR Analytics on Employee Attrition & Performance using Random Forest Classifier
15, Aug 20

02 Random Forest Classifier using Scikit-learn
04, Sep 20

05 Random Forest Regression in Python
14, Jun 19

06 NLP | Classifier-based Chunking | Set 2
22, Feb 19

04, Sep 20

03 Difference Between Model Parameters VS HyperParameters
11, May 20

04 Hyperparameters Optimization methods - ML
20, Jun 20

07 Calculate Efficiency Of Binary Classifier
15, Nov 17

08 Image Classifier using CNN
23, Dec 17

Article Contributed By :

saurabh48782
@saurabh48782

Vote for difficulty

Easy | Normal | Medium | Hard | Expert

Article Tags : Machine Learning

Practice Tags : Machine Learning

Improve Article Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

ADVERTISEMENT BY ADRECOVER

National Team Jerseys

Explore the signature red jerseys sported by the USA Basketball Women's National Team.

Nike

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

@geeksforgeeks , Some rights reserved