Data set number:44

Data set description:

The dataset consists of 4601 instances. The dataset has 58 numeric features and two binary Class (target). All 58 features have 1000 unique values, and there were no missing values in the data. The target classifies the instances into two groups of positive and negative. The task can be to predict the target value for queries. In this project, we want to, adjust the min_samples_leaf parameter to at least 5 different values, and evaluate the training and test roc_auc scores using 10-fold cross-validation for the dataset. Then, we point out the regions of overfitting and underfitting on the graph between min_samples_leaf parameter on the x-axis and mean roc_auc score (of the 10 folds) on the y-axis.Finally, we use GridSearchCV to search for the best parameter and generate the results of 10-fold cross-validation.

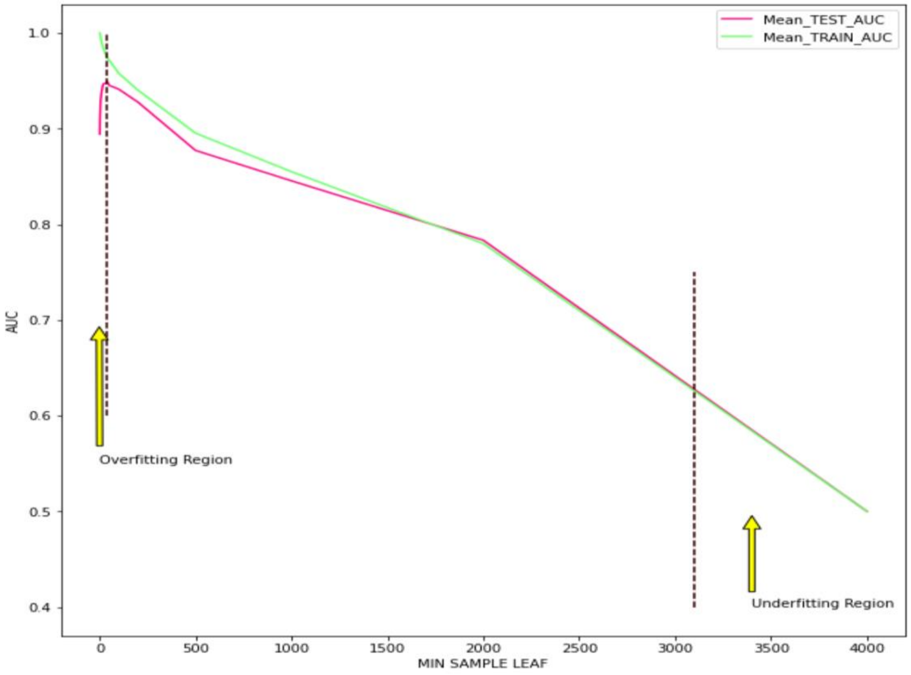
More information about the data can be found here: https://www.openml.org/d/44

Task (1):

These values were considered as min_samples_leaf parameter:

[1,2,5,10,15,20,30,40,50,100,200,500,1000,2000,4000]

Firstly, for each value, training, and test roc_auc scores on 10-fold cross-validation were measured. Next, a plot was generated with the min_samples_leaf parameter values on the x-axis and mean roc_auc score (of the 10 folds) on the y-axis.



- Overfitting occurs when the model has too much complexity and is able to fit the noise in the training data, resulting in high accuracy on the training data but poor accuracy on the test data. This could happen if the value of min_samples_leaf is too small, meaning the model is allowed to have too many leaf nodes and is therefore too complex. In this case, as the value of min_samples_leaf decreases (meaning the model becomes more complex), the training accuracy may continue to increase while the test accuracy may plateau or even decrease. As illustrated in the graph above, when min_samples_leaf is small, we see high accuracy on the training data and less accuracy on the test data and resulting in overfitting.

- Underfitting occurs when the model is too simple and is unable to capture the underlying patterns in the data, resulting in low accuracy on both the training and test data. This could happen if the value of min_samples_leaf is too large, meaning the model is not allowed to have enough leaf nodes and is therefore too simple. In this case, as the value of min_samples_leaf increases (meaning the model becomes less complex), both the training and test accuracy may improve but will likely plateau at some point. As illustrated in the graph above, when min_samples_leaf is not small , we see accuracy on both the training and test data was decreased and resulting in underfitting.

To avoid overfitting and underfitting, you would want to choose a value of min_samples_leaf that results in the best performance on the test data. This may involve trying several different values of min_samples_leaf and comparing the accuracy on the test data for each. The optimal value of min_samples_leaf is the one that results in the best accuracy on the test data, while also being simple enough to avoid overfitting. In task (2), GridSearchCV was used to find the best value of the min_samples_leaf parameter.

Task (2):

In this task, by using GridSearchCV, we find the best value of the min_samples_leaf that leads to better model performance.

The benefit of using GridSearchCV to find the best minimum sample leaf parameter is that it automates the process of trying out different values for this parameter and selects the one that produces the best performance. Using GridSearchCV to find the best minimum sample leaf parameter can help to avoid overfitting or underfitting, as it finds the optimal balance between complexity and generalization performance.