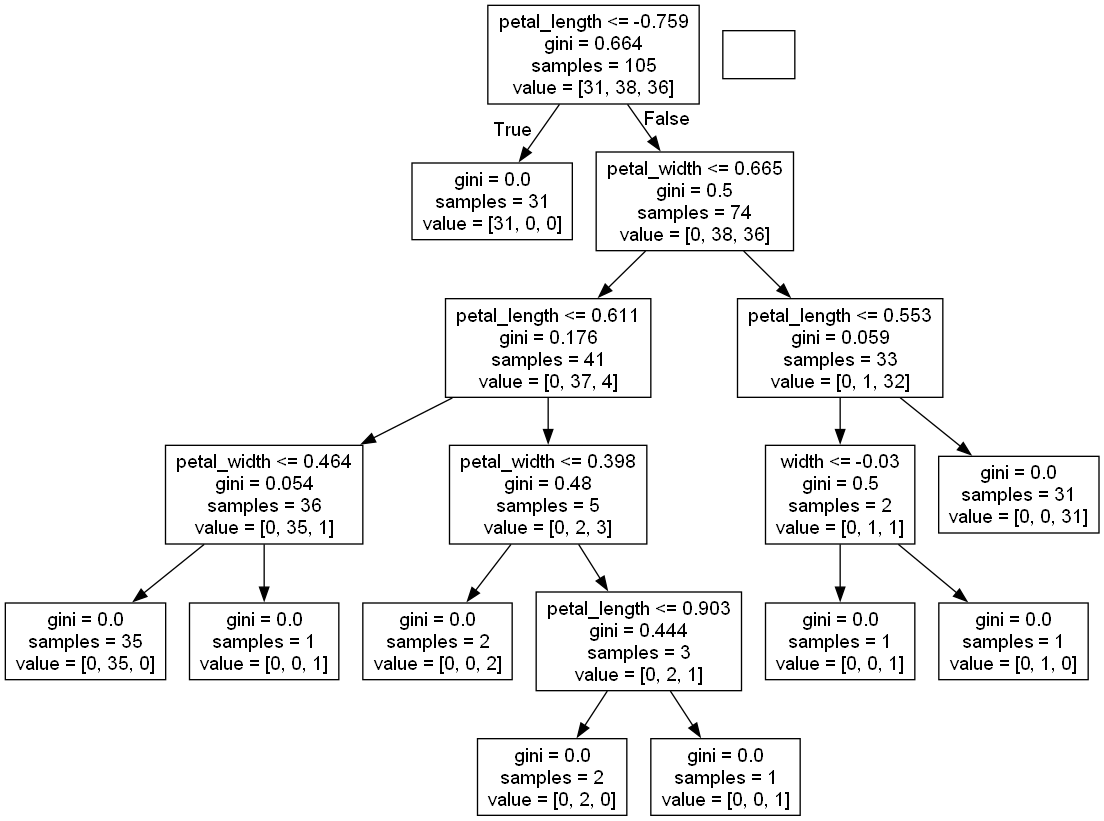
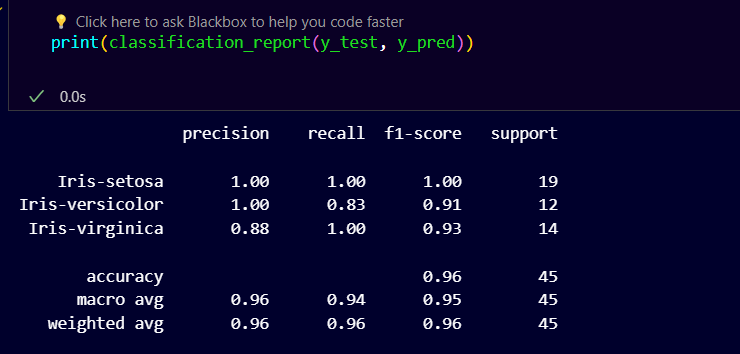
**Machine Learning Evaluation**

Steps:

1. Imported the necessary modules. (eg: pandas, sklearn, StandardScaler etc.)
2. Read the data from the "iris\_dataset.csv" file into a dataframe data.
3. Checked for null values in the dataframe using the isnull() method.
4. Created the feature matrix X by dropping the "class" column from the dataframe.
5. Created the target vector y with the values from the "class" column.
6. Checked the shape of the training and testing datasets.
7. Preprocessing of the dataset:
   1. Initialized a StandardScaler object scaler.
   2. Scaled the training and testing sets using fit\_transform() method of a scaler.
   3. Assigns the scaled training and testing sets to X\_train\_scaled and X\_test\_scaled respectively.
8. Initialized and fitted a DecisionTreeClassifier object clf with a random state of 32 to the training dataset.
9. Predicted the target values for the scaled testing set using the predict() method of clf. Assigned the predicted values to y\_pred.
10. Calculated the accuracy of the classifier by comparing the predicted values with the actual values in the testing set using the accuracy\_score() function. Also prints the classification\_report including precision, recall, F-1 score.
11. Generated a dot data representation of the decision tree classifier using the export\_graphviz() function.
12. Created a graph from the dot data using graph\_from\_dot\_data() method of pdp.
13. Wrote the graph to a PNG file named "decision\_tree.png".
14. Hyperparameter Tuning:
    1. Initialized a GridSearchCV object grid\_search with the classifier, hyperparameter grid, and 3-fold cross-validation.
    2. Fitted the grid search to the scaled training set and the target vector.
    3. Retrieved the best hyperparameters and best accuracy from the grid search results.
    4. Prints the classification report for the tuned model, which includes precision, recall, F1-score, and support for each class.

**Graph of the decision tree:**  


Classification Report (before Hyperparameter Tuning)



Classification Report (after Hyperparameter Tuning)

