

IRIS FLOWER CLASSIFIER

1. Importing Necessary Libraries

- Libraries like `pandas`, `seaborn`, `matplotlib`, and `sklearn` are used for data manipulation, visualization, and machine learning.

2. Loading the Dataset

- The Iris dataset is loaded from the specified file path (`TASK-1/Iris.csv`).

3. Exploring the Dataset

- Displays the first 5 rows of the dataset (`head()`), dataset information (`info()`), and summary statistics (`describe()`).

4. Dropping Unnecessary Columns

- Removes the `Id` column if it is present in the dataset.

5. Handling Missing Values

- Checks for missing values. If found, they are filled using the column mean (`fillna(data.mean())`).

6. Data Visualization

- Visualizes relationships between features using `seaborn.pairplot()` with species-based coloring.

7. Preparing Data for Modeling

- Splits the data into features (`x`) and labels (`y`) by separating the `Species` column.

8. Splitting Data into Training and Testing Sets

- Divides the dataset into training (80%) and testing (20%) subsets using `train_test_split()`.

9. Hyperparameter Tuning with GridSearchCV

- Optimizes the Random Forest Classifier using `GridSearchCV` with hyperparameters like `n_estimators`, `max_depth`, and `min_samples_split`.

10. Training the Random Forest Classifier

- The best model from grid search is trained on the training data.

11. Making Predictions

- Predictions are made on the test set using the trained model.

12. Evaluating the Model

- Evaluation metrics include:
 - **Confusion Matrix**
 - **Classification Report** (precision, recall, F1-score)
 - **Accuracy Score**

13. Visualizing Feature Importance

- Displays the importance of each feature in predicting the target using a horizontal bar plot.

Key Outputs:

- Performance metrics and feature importance visualization are the main results of this program.