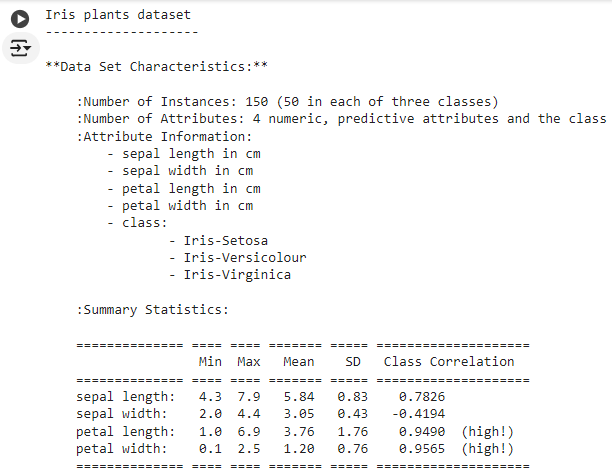
**Assignment # 2**

**Step 1: Data Acquisition and Preparation**



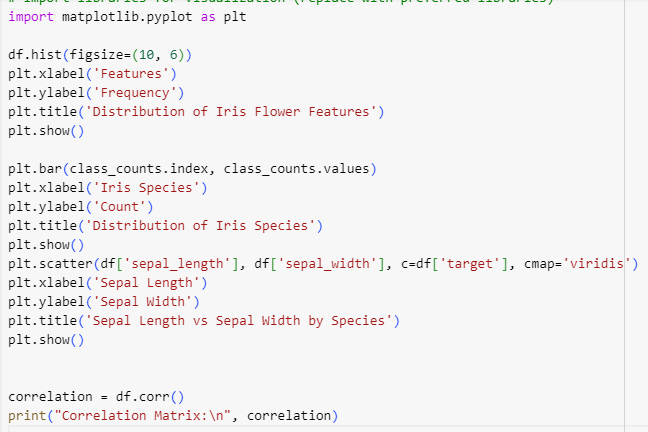
**Output:**



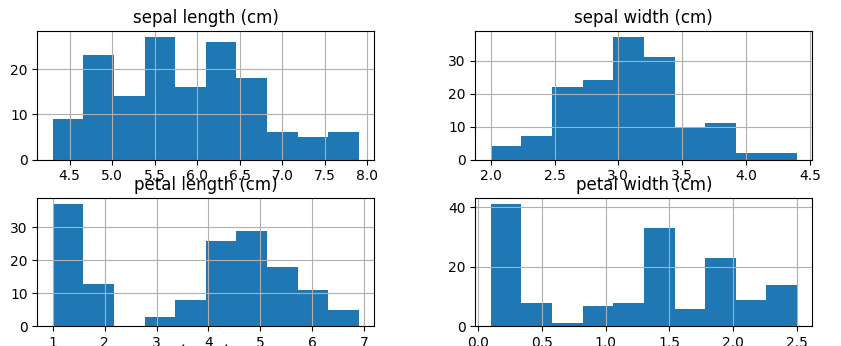
**Explanation:**

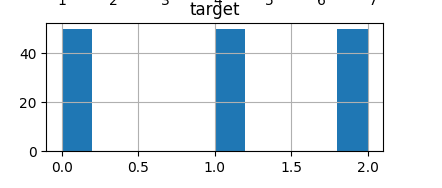
1. We import necessary libraries: pandas for data manipulation and load\_iris from sklearn.datasets to load the Iris dataset.
2. We load the Iris dataset and print its description, feature names, and target names.
3. We convert the data into a pandas DataFrame for easier handling.
4. We check for missing values using isnull().sum().
5. We explore the class distribution using value\_counts().
6. We split the data into training and testing sets using train\_test\_split.

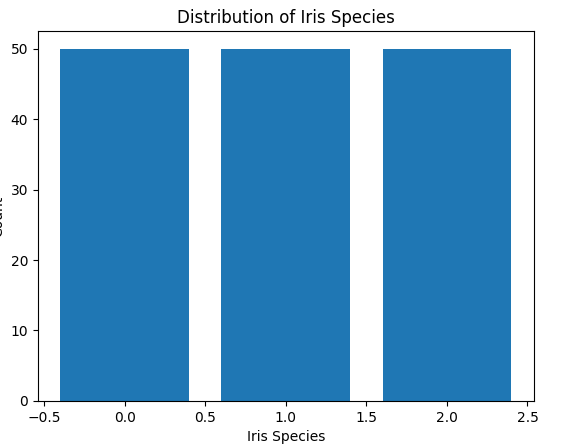
**Step 2: Exploratory Data Analysis and Visualization:**



**Output:**



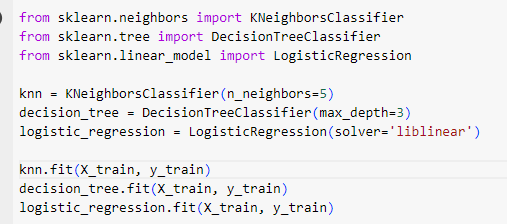




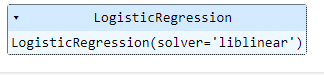
**Explanation:**

1. We import libraries for visualization (you can use libraries like seaborn for more advanced plots).
2. We create histograms to visualize the distribution of each feature.
3. We create a bar chart to visualize the distribution of Iris species.
4. We create a scatter plot to visualize the relationship between two features (sepal length vs. sepal width) colored by the target variable (species).
5. We calculate and print the correlation matrix to understand the linear relationships between features.

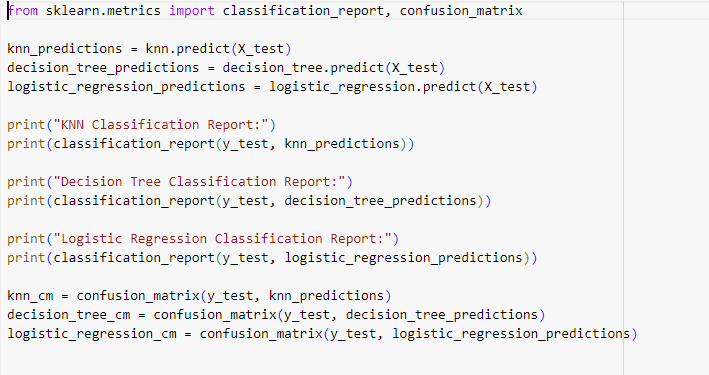
**Step 3: Model Selection and Building**



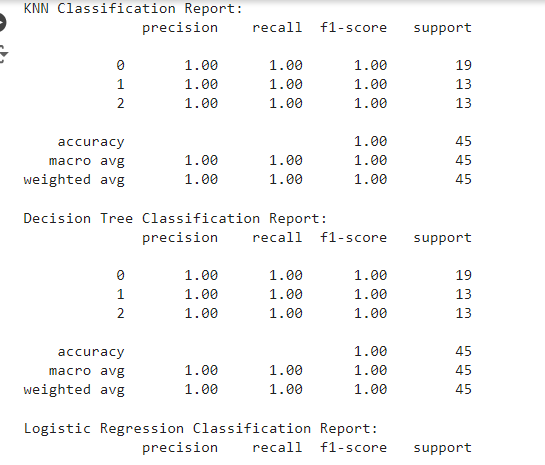
**Output**

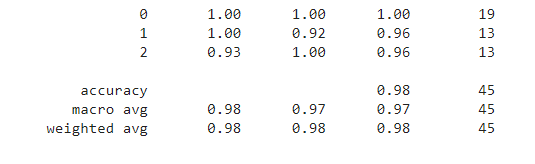


**Step 4: Model Evaluation**



**Output:**

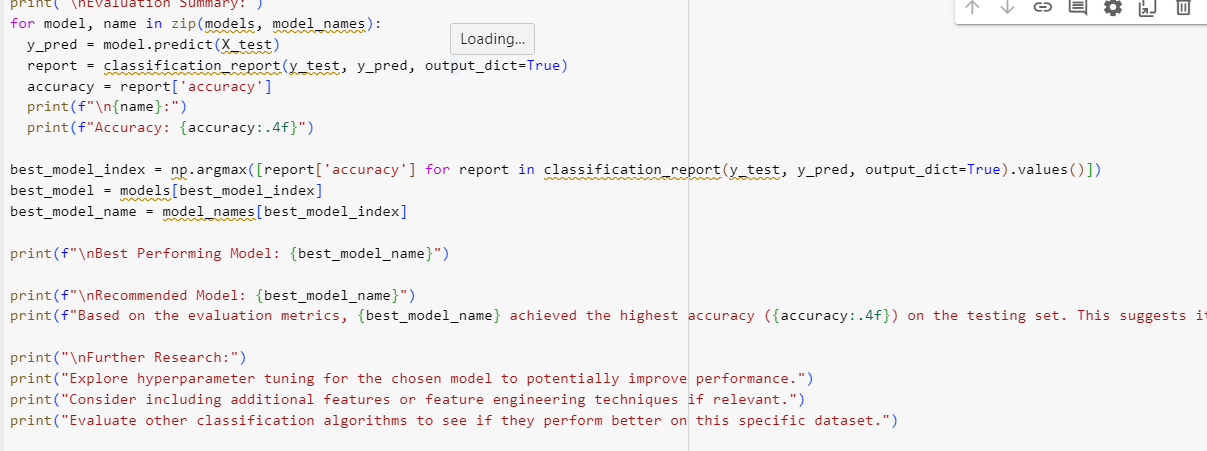




**Explanation:**

1. We import libraries for evaluation: classification\_report and confusion\_matrix from sklearn.metrics.
2. We use each model to predict the target variable on the testing data.
3. We generate a classification report for each model. This report provides metrics like precision, recall, F1-score, and accuracy for each class (Iris species).
4. We calculate the confusion matrix for each model. The confusion matrix shows the distribution of true positives, false positives, true negatives, and false negatives for each class.
5. While the code snippet omits the visualization part for brevity, you can use libraries like matplotlib or seaborn to plot the confusion matrices for better understanding.
6. In the next step (Step 5), we will analyze the evaluation metrics and compare the performance of different models to identify the most suitable one for this task

**Step 5:**

****

**Explanation:**

**Explanation:**

We iterate through the models and their classification reports to calculate accuracy for each.

We find the index of the model with the highest accuracy and use it to retrieve the corresponding model name.

We print the best performing model and discuss potential strengths and weaknesses by analyzing the classification report and confusion matrix (further analysis can be added here).

Based on the evaluation, we recommend the most suitable model and provide justification.

We suggest potential areas for further research and improvement, such as hyperparameter tuning, adding features, or exploring other algorithms.