"""

IRIS DATA ANALYSIS AND LOGISTIC REGRESSION

This script demonstrates a basic analysis of the Iris dataset using exploratory data analysis (EDA) techniques,

including statistical summaries and visualizations. Additionally, it trains a Logistic Regression model on the

Iris dataset and evaluates its performance using accuracy, precision, and recall metrics.

LIBRARIES:

- pandas: Data manipulation library

- numpy: Numerical computing library

- sklearn: Scikit-learn, a machine learning library

- matplotlib.pyplot: Data visualization library for creating plots and charts

- seaborn: Statistical data visualization library based on Matplotlib

FUNCTIONS/STEPS:

1. Import necessary libraries:

- pandas: data manipulation

- numpy: numerical computing

- train\_test\_split: split the dataset into training and testing sets

- LogisticRegression: logistic regression model

- accuracy\_score, precision\_score, recall\_score: evaluation metrics

- load\_iris: load the Iris dataset

- matplotlib.pyplot: create plots

- seaborn: create statistical visualizations

2. LOAD THE IRIS DATASET:

- Use the load\_iris function from sklearn.datasets to load the Iris dataset.

- Create a pandas DataFrame to store the dataset with feature names and target labels.

3. PERFORM EXPLORATORY DATA ANALYSIS (EDA):

- Display basic statistics of the dataset using the describe() method.

- Visualize the distribution of each feature using histograms.

- Visualize relationships between features using pairplots.

4. SPLIT THE DATA INTO TRAINING AND TESTING SETS:

- Use the train\_test\_split function from sklearn.model\_selection to split the dataset.

5. CHOOSE AND TRAIN A LOGISTIC REGRESSION MODEL:

- Use LogisticRegression from sklearn.linear\_model.

- Train the model on the training set.

6. PREDICT ON DATA SET:

- Use the trained model to make predictions on the testing set.

7. EVALUATE MODEL PERFORMANCE:

- Calculate and print accuracy, precision, and recall using appropriate metrics.

Note: Ensure that the necessary libraries are installed in the Python environment before running the script.

"""

# The actual code follows...

# (Include the entire original code here)