```
# -*- coding: utf-8 -*-
"""Iris DT classification.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1c6eOfI6fKUjpbykxZOZ9JjlxgV2AFIBK
"""

# Import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the Iris dataset
iris = load_iris()
# Convert the dataset into a DataFrame
data = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data['species'] = iris.target

data.info()

# Display the first few rows of the dataset
print(data.head())

import seaborn as sns
correlation_matrix = data.corr()
# Print the correlation matrix
plt.figure(figsize=(14, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()

# Split the dataset into features (X) and target (y)
X = data.drop('species', axis=1)
y = data['species']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
test_size=0.2, random_state=42)

# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)
# Train the model
clf.fit(X_train, y_train)
# Predict on the test set
y_pred = clf.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
y_pred= clf.predict(X_test)
results=X_test.copy()
results['Actual']=y_test
results['Predicted']=y_pred
value=X.columns
results

# Print the classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred,
target_names=iris.target_names))

# Print the confusion matrix
print("\nConfusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
print(cm)

# Visualize the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```