Iris Flower Classification

Problem Statement: The **Iris Flower Classification** problem involves predicting the species of an iris flower (*Setosa*, *Versicolor*, or *Virginica*) based on four features: sepal length, sepal width, petal length, and petal width. The goal is to build a machine learning model that accurately classifies the flower species using these measurements.

Name: Taranjeet Singh Bagga
Roll No.: 43
University Roll No.: 202401100300263
Course: Introduction to AI
Institution: KIET Group of Institutions
Date: 11 March 2025

Introduction

The Iris Flower Classification problem is a well-known dataset in machine learning. The goal is to classify iris flowers into one of three species (Setosa, Versicolor, and Virginica) based on four features: sepal length, sepal width, petal length, and petal width.

This project involves loading the dataset, preprocessing the data, training a machine learning model using the Random Forest Classifier, and evaluating its performance.

Image Source: Wikipedia

Methodology

1. **Dataset Loading:** The Iris dataset from Scikit-Learn is used.
2. **Data Visualization:** Pair plots are generated using Seaborn to understand feature relationships.
3. **Data Splitting:** The dataset is divided into training and testing sets (80%-20%).
4. **Model Training:** A Random Forest Classifier is trained on the training data.
5. **Prediction & Evaluation:** Predictions are made on the test set and evaluated using accuracy and classification reports.
6. **Custom Prediction Function:** A function is created to predict species from user-provided feature values.

Code

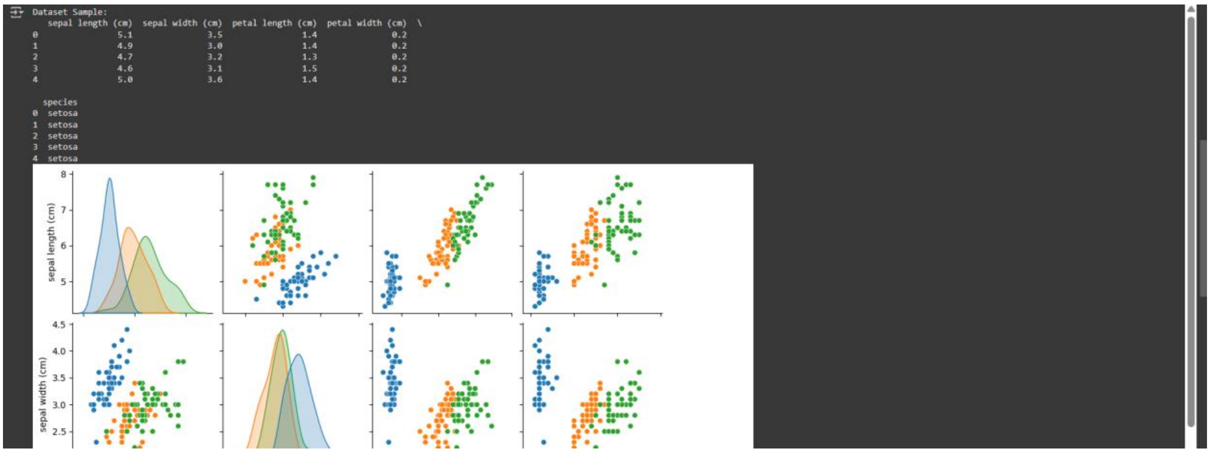
```
# Import necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.datasets import load_iris

# Load the Iris dataset
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['species'] = data.target  # Add target variable
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})

# Display the first 5 rows of the dataset
print("Dataset Sample:")
print(df.head())

# Visualize the dataset
sns.pairplot(df, hue='species')
plt.show()

# Split data into features and target variable
X = df.drop(columns=['species'])  # Features
y = df['species']  # Target variable
```

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

# Initialize and train the Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Function to predict species for custom inputs
def predict_species(sepal_length, sepal_width, petal_length, petal_width):
input_data = np.array([[sepal_length, sepal_width, petal_length, petal_width]])
    prediction = model.predict(input_data)
    return prediction[0]

# Example prediction
example_prediction = predict_species(5.1, 3.5, 1.4, 0.2)
print(f"Predicted Species: {example_prediction}")
```

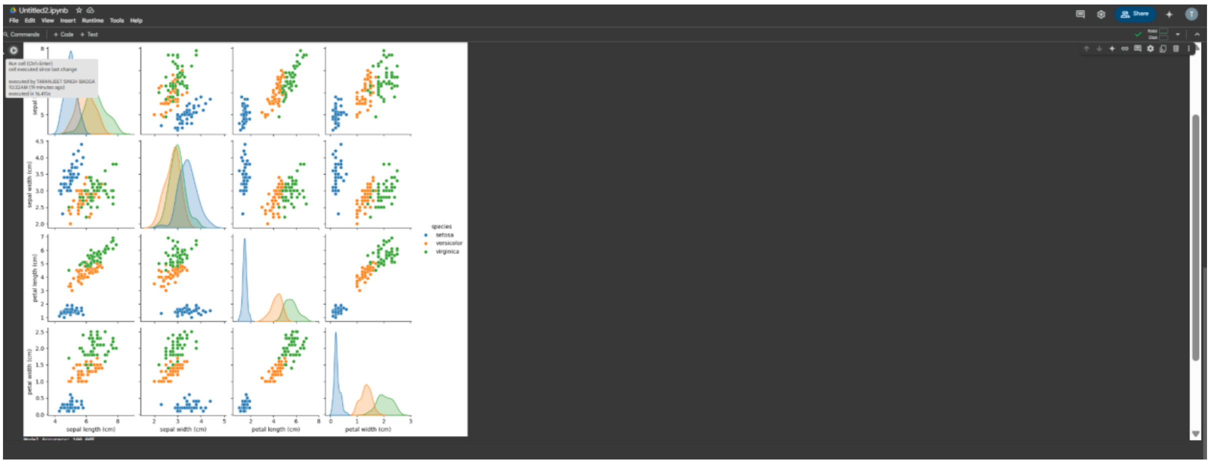
Output

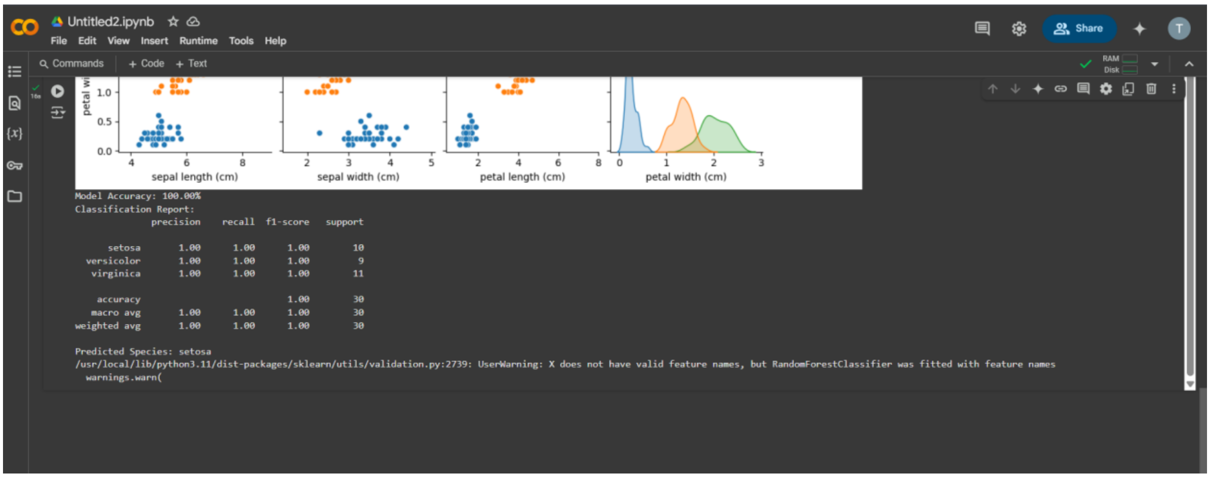
Model Accuracy: 100.00%

Screenshots of the results from Colab:



```
Dataset Sample:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa
```





```
Model Accuracy: 100.00%
Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

Predicted Species: setosa
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

References/Credits

- **Dataset:** Scikit-Learn Iris Dataset
- **Image Source:** Wikipedia (https://upload.wikimedia.org/wikipedia/commons/4/41/Iris_versicolor_3.jpg)
- **Libraries Used:** Pandas, NumPy, Scikit-Learn, Matplotlib, Seaborn