Name – Krishna Sharma

Branch-CSEAI

Section – B

University Roll No. – 202401100300139

Project Name – Iris Flow Representation

Methodology:

This code implements a machine learning pipeline using Python.

1. **Reading the Data:**
    - Loads data from a CSV file named iris.csv and previews its contents.
    - Displays the first few rows of the dataset for initial inspection.
2. **Data Preparation:**
    - Splits the dataset into features (X) and labels (y). Features include all columns except the last one, while the labels correspond to the last column.
3. **Training and Testing Split:**
    - Splits the dataset into training (80%) and testing sets (20%) using train_test_split to evaluate the model's performance.
4. **Model Training:**

    - Initializes and trains a k-Nearest Neighbors (k-NN) classifier with 3 neighbors on the training data.

5. **Evaluation:**
    - Uses the trained model to predict labels for the test data.
    - Evaluates the model's accuracy using accuracy_score.
6. **Visualization:**
    - **Confusion Matrix:** Visualizes model predictions and actual outcomes using a heatmap.
    - **Scatter Plot:** Plots sepal features to explore patterns among the classes.
    - **Pairplot:** Creates a pairwise scatter plot of all features for better visualization of class separability.
    - **Boxplots:** Displays the distribution of each feature across different classes.
    - **Histograms:** Shows the frequency distribution of each feature in the dataset.

7. **Error Handling:**

   - Provides error messages for missing or empty CSV files and handles unexpected issues gracefully.

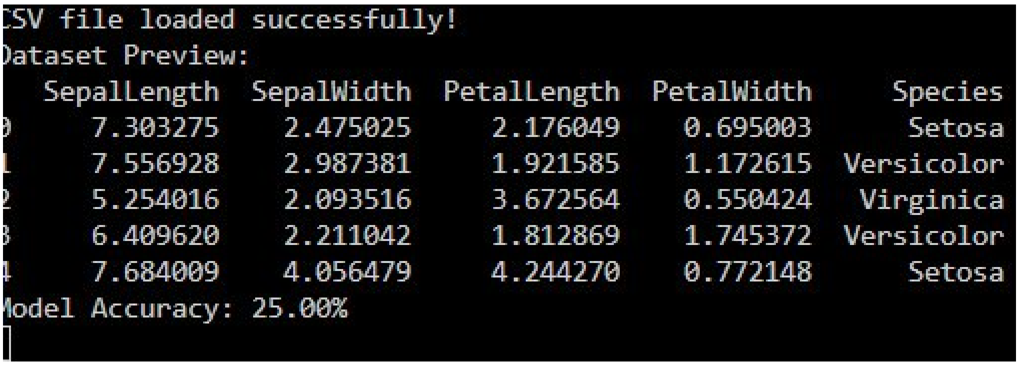
The aim is to classify the Iris dataset using a k-NN algorithm while evaluating and visualizing the model's performance and data distributions.

Code:

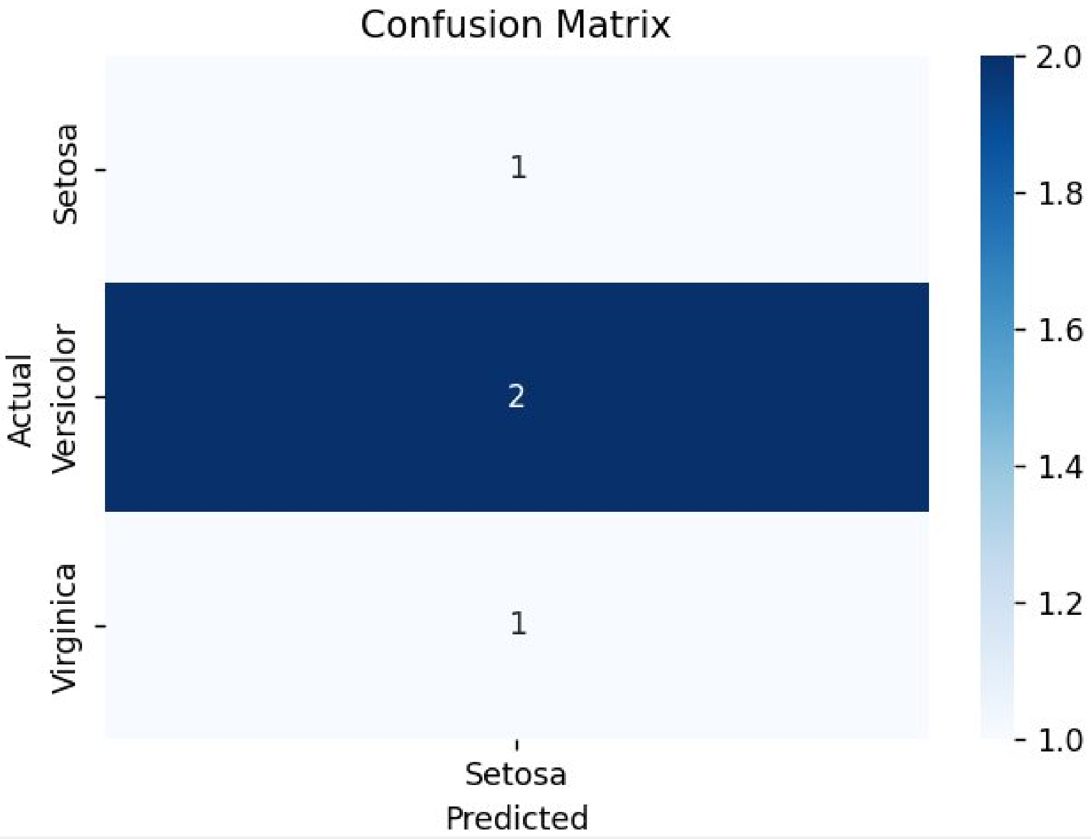
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Debugging: Print the current working directory
print(f"Current Working Directory: {os.getcwd()}")

try:
    # Read the CSV file
    data = pd.read_csv("iris.csv")  # Replace "iris.csv" with the correct file path if needed
    print("CSV file loaded successfully!")

    # Display the first few rows of the dataset
    print("Dataset Preview:")
    print(data.head())

    # Splitting features and labels
    X = data.iloc[:, :-1].values  # All columns except the last (features)
    y = data.iloc[:, -1].values  # The last column (labels)

    # Splitting the dataset 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)

    # Training the k-NN model
    knn = KNeighborsClassifier(n_neighbors=3)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)

    # Evaluating the model
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Model Accuracy: {accuracy * 100:.2f}%")

    # Plotting the Confusion Matrix
    cm = pd.crosstab(y_test, y_pred,
rownames=["Actual"], colnames=["Predicted"])
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, cmap="Blues",
fmt="d")
    plt.title("Confusion Matrix")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

    # Scatter Plot of Sepal Features (Example)
    plt.figure(figsize=(6, 4))
    for i, label in enumerate(data.iloc[:, -1].unique()):
        plt.scatter(data[data.iloc[:, -1] == label].iloc[:, 0],
                    data[data.iloc[:, -1] == label].iloc[:, 1],
                    label=label)
    plt.title("Sepal Feature Scatter Plot")
    plt.xlabel("Sepal Length")
    plt.ylabel("Sepal Width")
    plt.legend()
    plt.show()

except FileNotFoundError:
    print("Error: The specified CSV file was not found. Make sure it exists in the correct directory.")
except pd.errors.EmptyDataError:
    print("Error: The CSV file is empty. Please check the file content.")
```

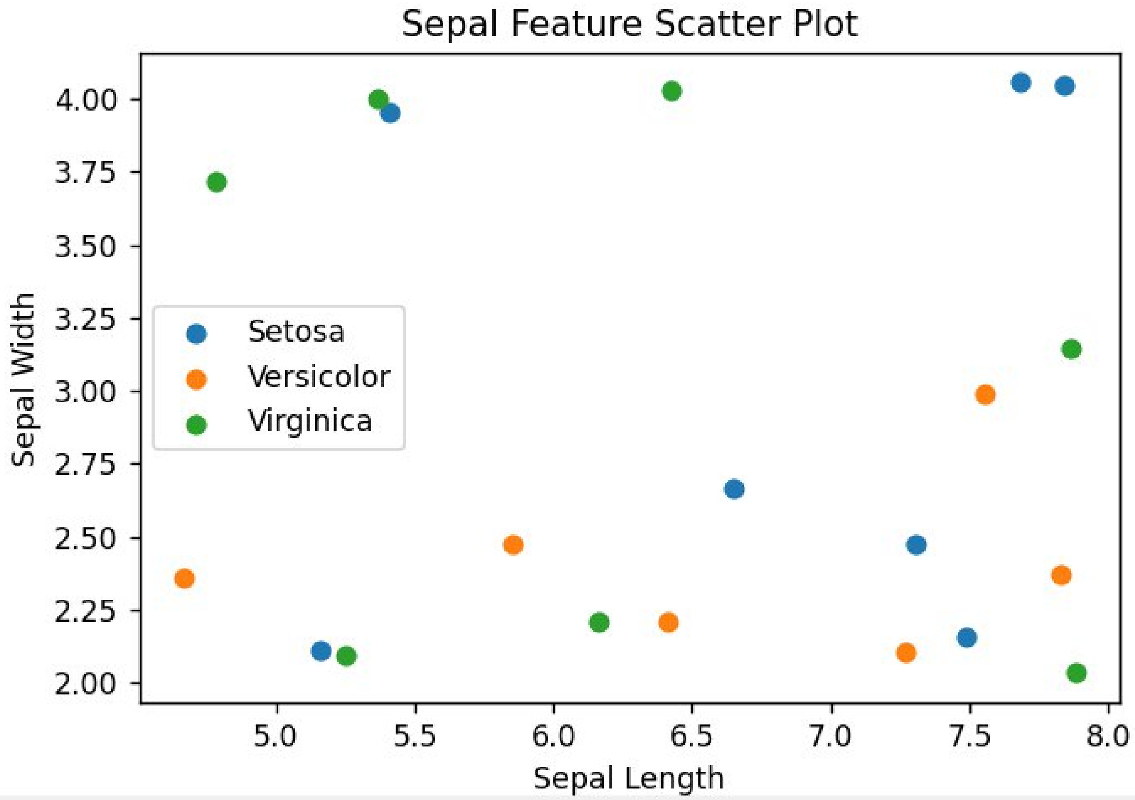
```
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

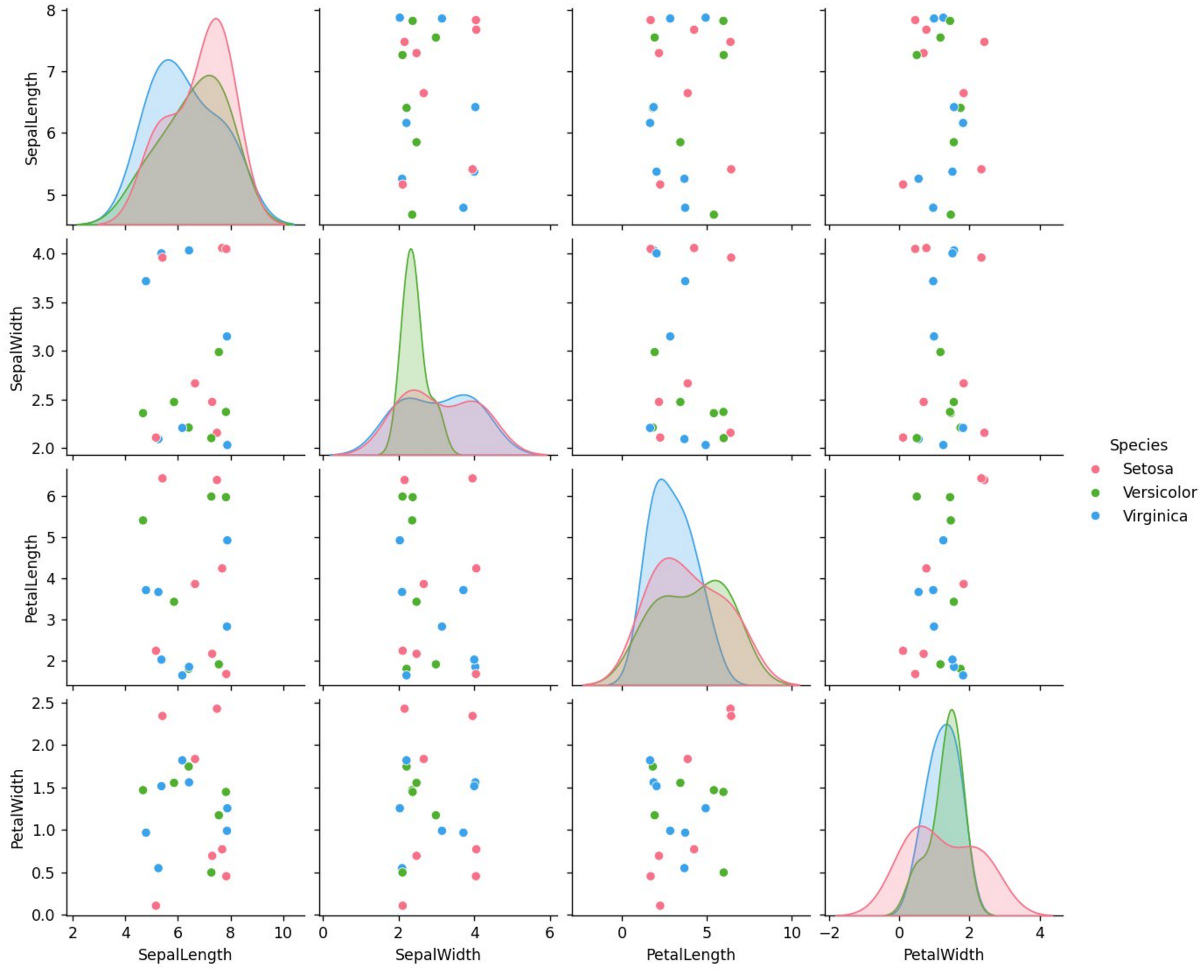
Code Output:



```
CSV file loaded successfully!
Dataset Preview:
   SepalLength  SepalWidth  PetalLength  PetalWidth     Species
0     7.303275    2.475025     2.176049    0.695003      Setosa
1     7.556928    2.987381     1.921585    1.172615  Versicolor
2     5.254016    2.093516     3.672564    0.550424   Virginica
3     6.409620    2.211042     1.812869    1.745372  Versicolor
4     7.684009    4.056479     4.244270    0.772148      Setosa
Model Accuracy: 25.00%
```




Sepal Feature Scatter Plot
Sepal Width
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal Length
Setosa
Versicolor
Virginica


SepalLength
8
7
6
5
SepalWidth
4.0
3.5
3.0
2.5
2.0
PetalLength
6
5
4
3
2
PetalWidth
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
8
10
SepalLength
0
2
4
6
SepalWidth
0
5
10
PetalLength
−2
0
2
4
PetalWidth
Species
Setosa
Versicolor
Virginica

How the Code Works Overall

1. It reads the Iris dataset from a CSV file.
2. Splits the data into training and testing sets.
3. Trains a k-NN classifier and evaluates its performance.
4. Visualizes the results using a confusion matrix and scatter plots.
5. Incorporates error handling to make the program robust and user-friendly.