K.I.E.T. Group of Institutions

Ghaziabad

Name: Arnav Singh

Branch: CSE(AI) - A

Roll No: 64

Date: 11/03/2025

Project Report On -. Iris Flower Classification

Introduction

The **Iris Flower Classification** project is a well-known machine learning task that involves classifying iris flowers into different species based on their physical characteristics. This classification is achieved using a dataset that consists of **sepal length, sepal width, petal length, and petal width** as input features. The dataset, commonly referred to as the **Iris dataset**, was introduced by the British statistician **Ronald Fisher** in 1936 and remains one of the most widely used datasets in machine learning and data science.

Methodology

1. **Data Collection:** The dataset consists of Sepal length, Sepal width, Petal length, Petal width and species.
2. **Data Processing:** The dataset is cleaned by handling missing values.
3. **Visualization Techniques:**
    - Co relation matrix for all four quantities.
    - Scatter plot for species.
    - Violin plot and box plot for sepal width and length.
    - A 3D plot to represent data in 3D.
4. **Tools Used:** Python, Pandas, Matplotlib, and Seaborn.

Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
# Check for duplicate rows
duplicate_rows = df[df.duplicated()]
print("Duplicate Rows:")
print(duplicate_rows)

# Drop duplicate rows (inplace=True modifies the
original DataFrame)
df.drop_duplicates(inplace=True)

# Verify that duplicates have been removed
print("\nDataFrame after removing duplicates:")
print(df.info())

# Assuming df is already defined and loaded as in the
previous code

# Now you can work with the cleaned DataFrame 'df'
print(df.head()) # Display first few rows
print(df.describe()) # Summary statistics
# ... further analysis ...
# Create a pairplot to visualize relationships
between features
# Replace 'species' with the actual name of the
column containing species information
# If the column name is 'Species', use hue='Species'
sns.pairplot(df, hue='Species')  # Changed 'species'
to 'Species'
plt.show()

# Create a heatmap to visualize correlation between
features
# Select only numerical features for correlation
calculation
numerical_features =
df.select_dtypes(include=['number']) # Select
numerical columns only
correlation_matrix = numerical_features.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True,
cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

```
# Create a boxplot to visualize the distribution of each feature for each species
for column in df.columns[:-1]:  # Exclude the last column (species)
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Species', y=column, data=df)
    plt.title(f'Boxplot of {column} by Species')
    plt.show()

# Create a violin plot to visualize the distribution of each feature for each species
for column in df.columns[:-1]:
    plt.figure(figsize=(8, 6))
    sns.violinplot(x='Species', y=column, data=df)
    plt.title(f'Violin Plot of {column} by Species')
    plt.show()
```

Output/Result

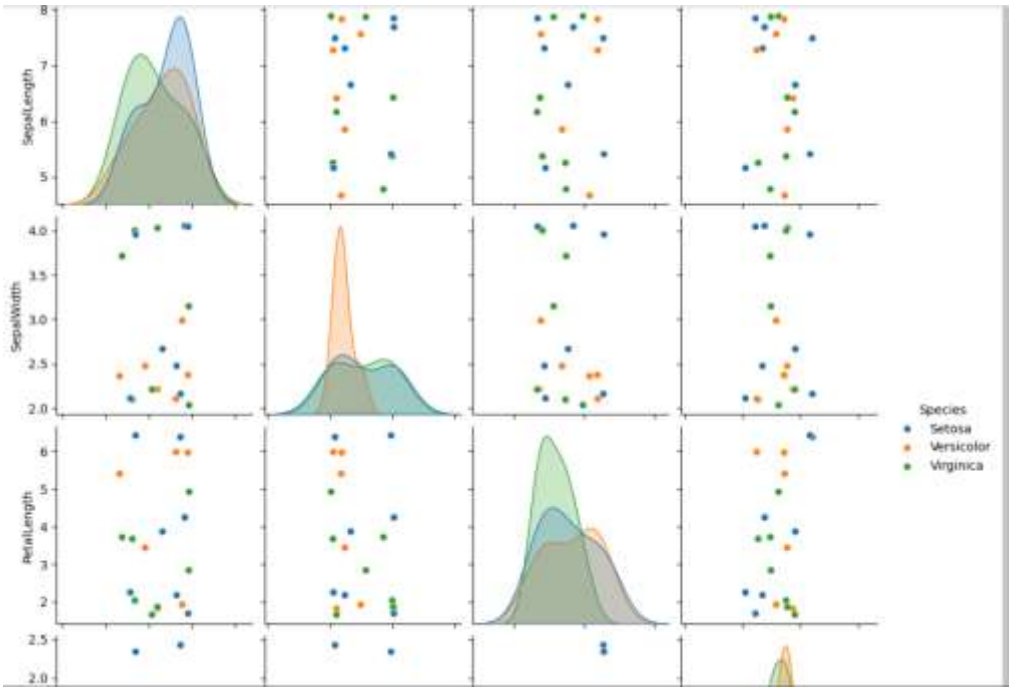
Observation from the analysis is given below:.

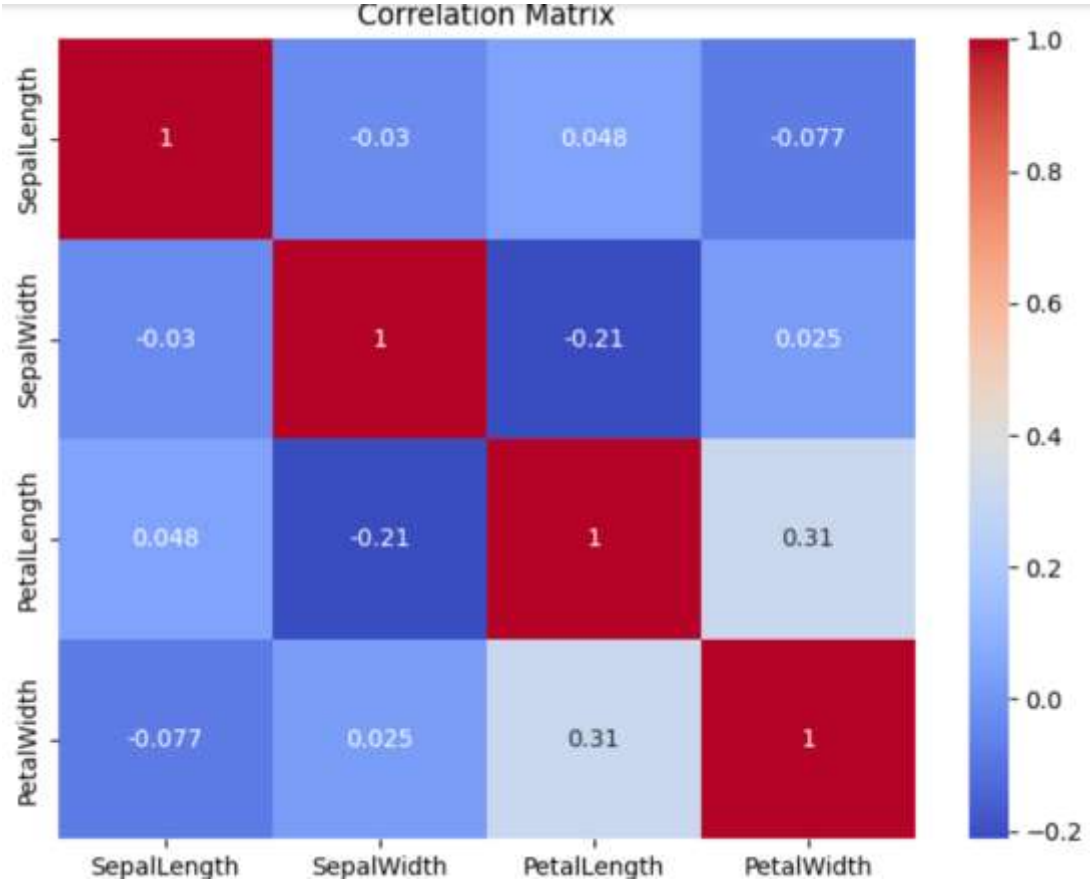
```
print(df.head())
```

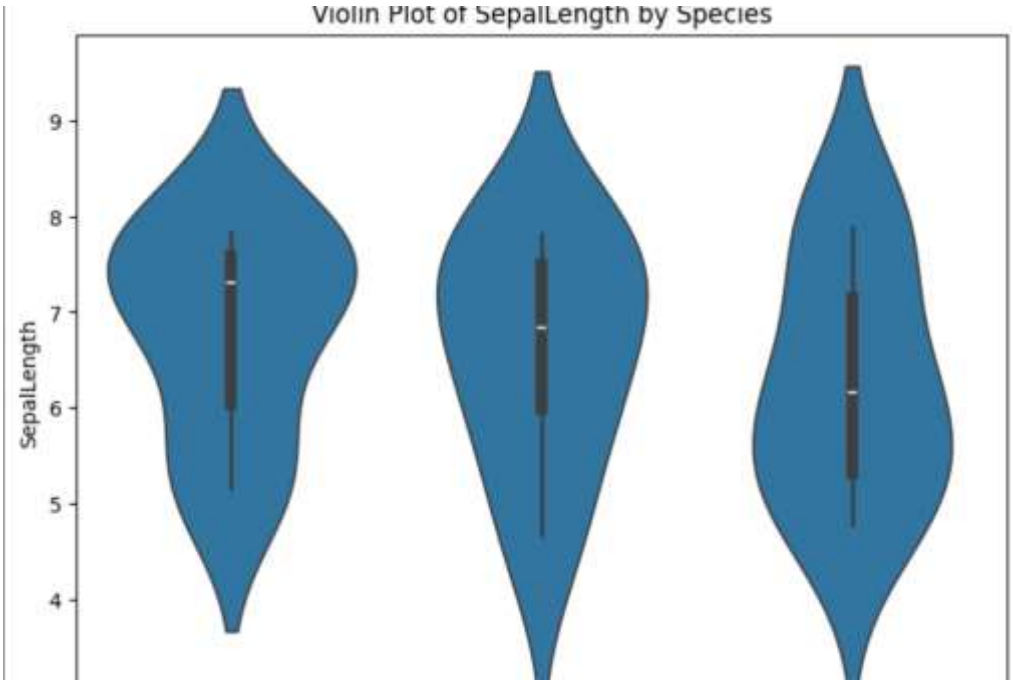
```
   SepalLength  SepalWidth  PetalLength  PetalWidth     Species
0     7.303275    2.475025     2.176049    0.695003      Setosa
1     7.556928    2.987381     1.921585    1.172615  Versicolor
2     5.254016    2.093516     3.672564    0.550424   Virginica
3     6.409620    2.211042     1.812869    1.745372  Versicolor
4     7.684009    4.056479     4.244270    0.772148      Setosa
```

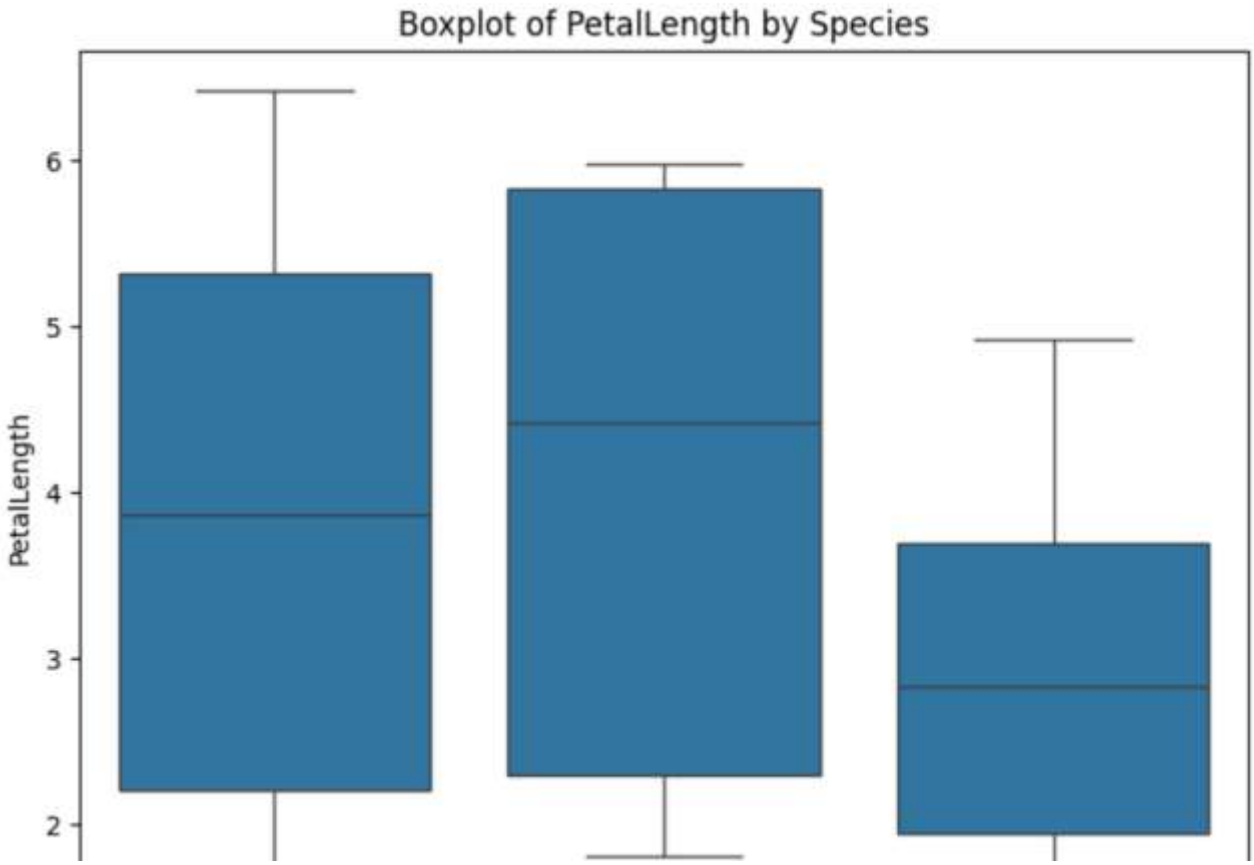
```
[5] print(df.info())
```

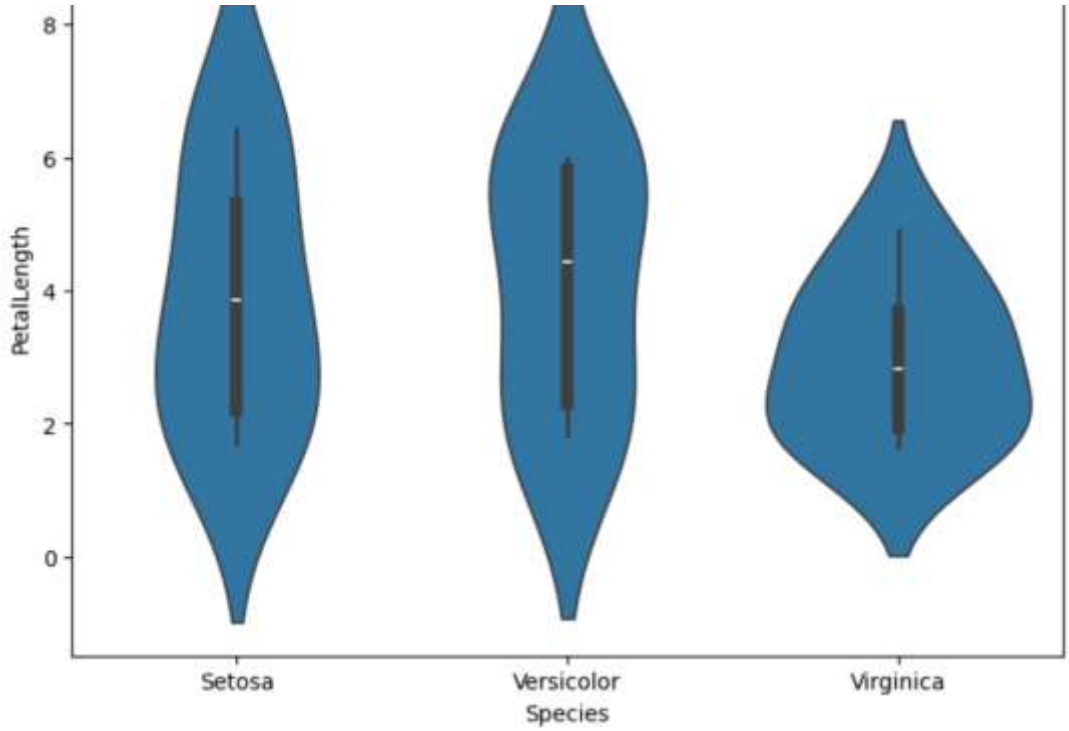
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   SepalLength  20 non-null     float64
 1   SepalWidth   20 non-null     float64
 2   PetalLength  20 non-null     float64
 3   PetalWidth   20 non-null     float64
 4   Species      20 non-null     object
dtypes: float64(4), object(1)
memory usage: 932.0+ bytes
None
```

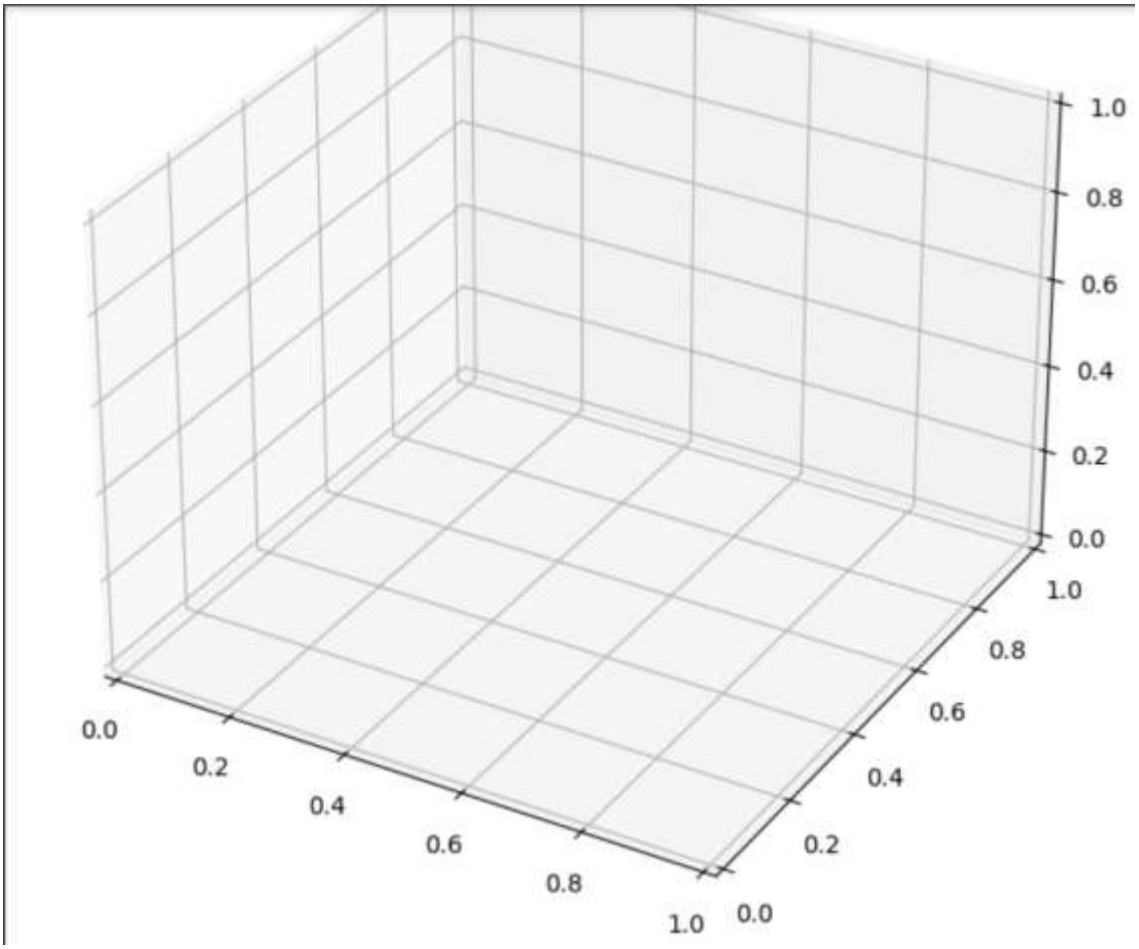



Correlation Matrix
SepalLength
SepalWidth
PetalLength
PetalWidth
1
-0.03
0.048
-0.077
-0.03
1
-0.21
0.025
0.048
-0.21
1
0.31
-0.077
0.025
0.31
1
SepalLength
SepalWidth
PetalLength
PetalWidth
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Violin Plot of SepalLength by Species
SepalLength
9
8
7
6
5
4


Boxplot of PetalLength by Species
PetalLength
6
5
4
3
2


PetalLength
8
6
4
2
0
Setosa
Versicolor
Virginica
Species



References/Credits:

- Dataset: [source: Kaggle]
- Libraries: Pandas, Matplotlib, Seaborn