**Importing Necessary Modules:**

import pandas as pd

import numpy as np

import os

import matplotlib.pyplot as plt

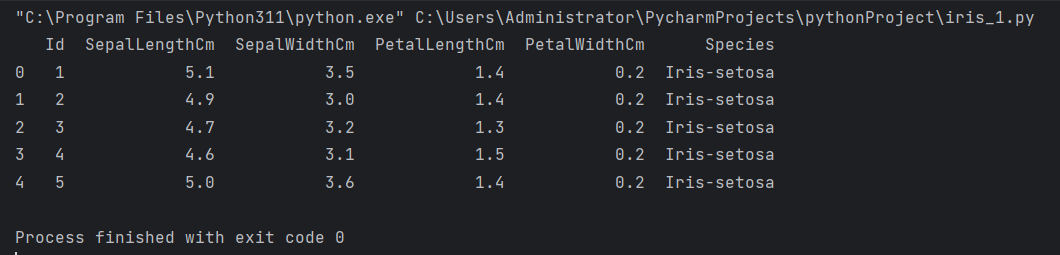
import seaborn as sns

Loading the Dataset:

df = pd.read\_csv("Iris.csv")

print(df.head())

The code reads the Iris dataset from a CSV file and displays the first few rows of the dataset.

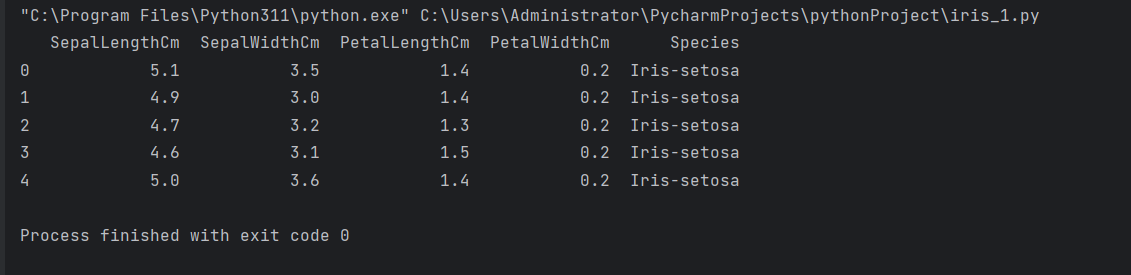


**Deleting a Column:**

df = df.drop(columns=['Id'])

print(df.head())

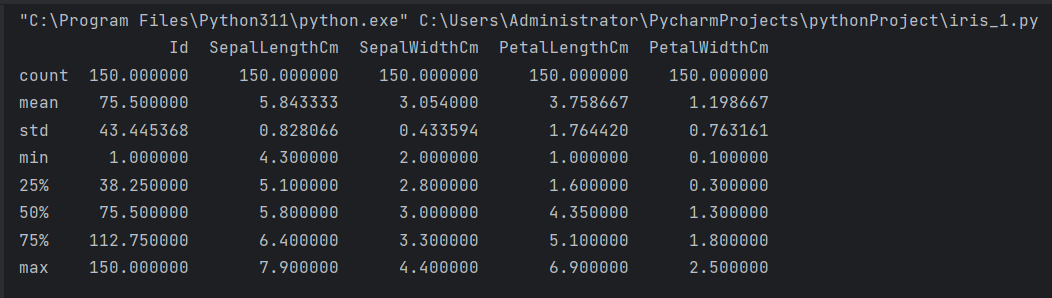
The 'Id' column is dropped from the dataset.



**Displaying Descriptive Statistics:**

print(df.describe())

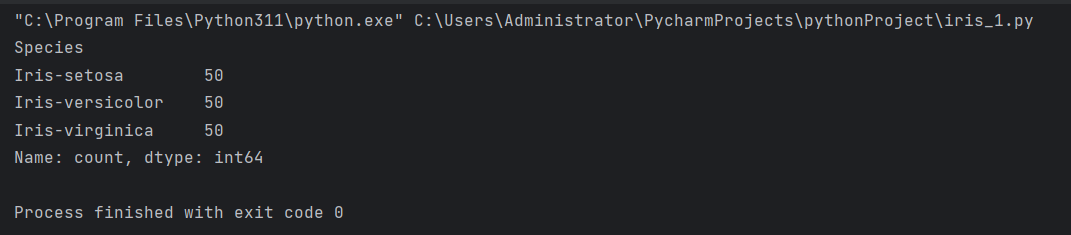
Descriptive statistics (count, mean, std, min, 25%, 50%, 75%, max) for numerical columns are displayed.



**Displaying Number of Samples on Each Class:**

print(df['Species'].value\_counts())

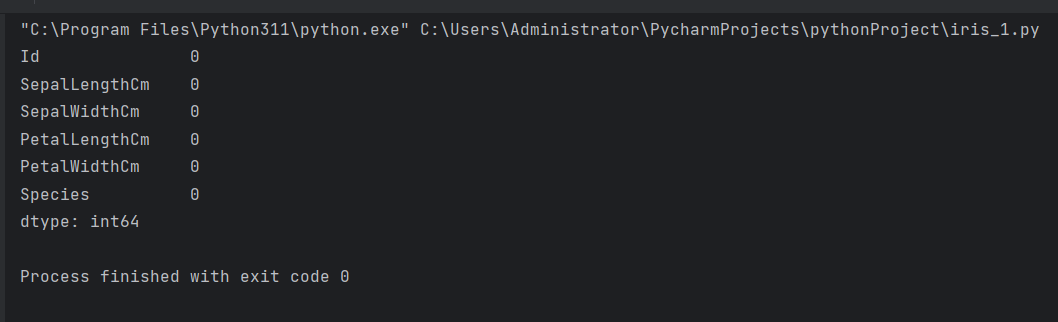
**Shows the distribution of samples across different classes in the 'Species' column.**



**Checking for Null Values:**

print(df.isnull().sum())

Displays the count of null values in each column.

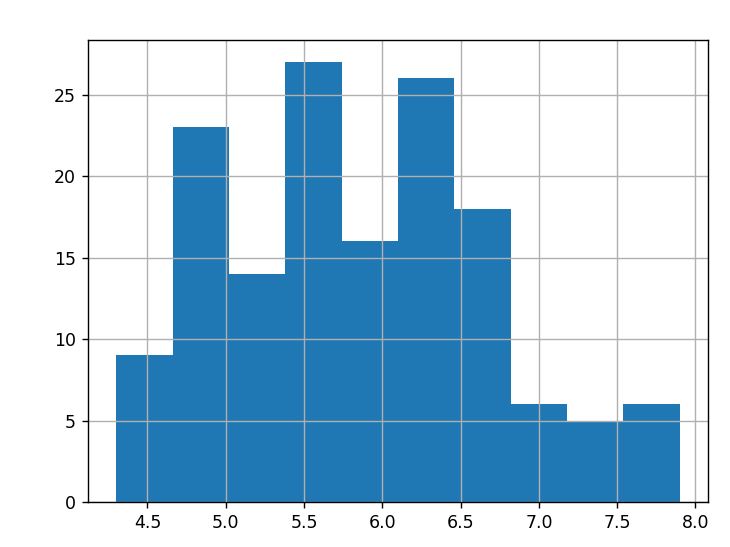


**Exploratory Data Analysis (EDA):**

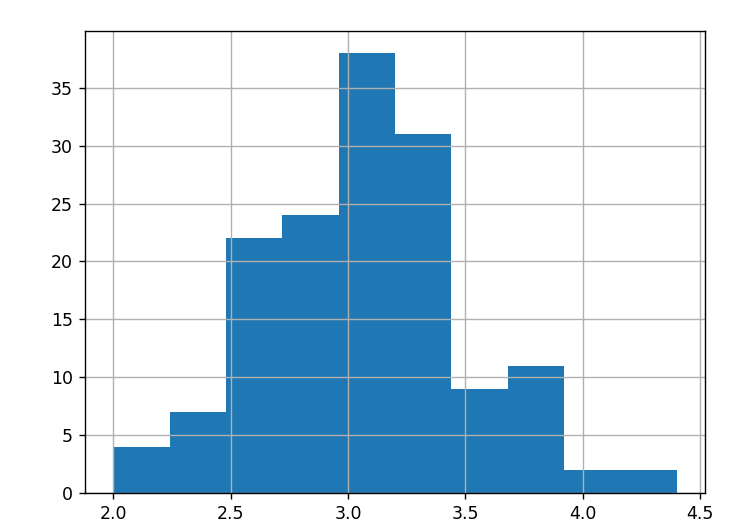
Several EDA plots are commented out, such as histograms and scatterplots, exploring the relationships between different features in the dataset.

**#Histograms**

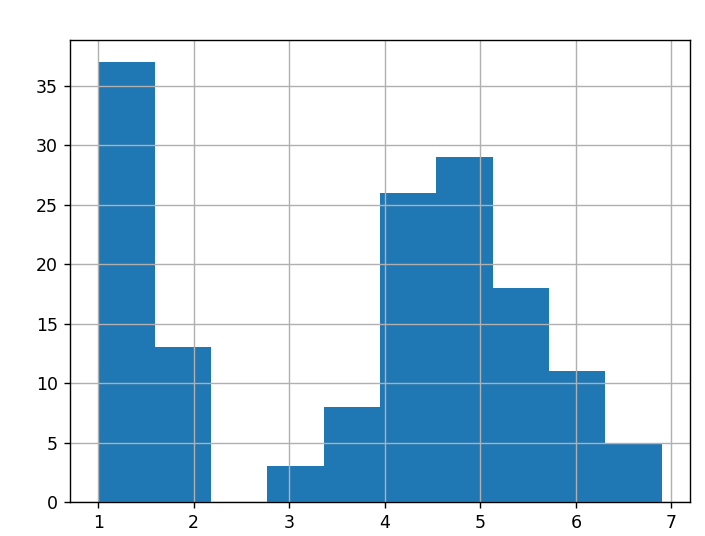
# df['SepalLengthCm'].hist()  
# plt.show()  
#



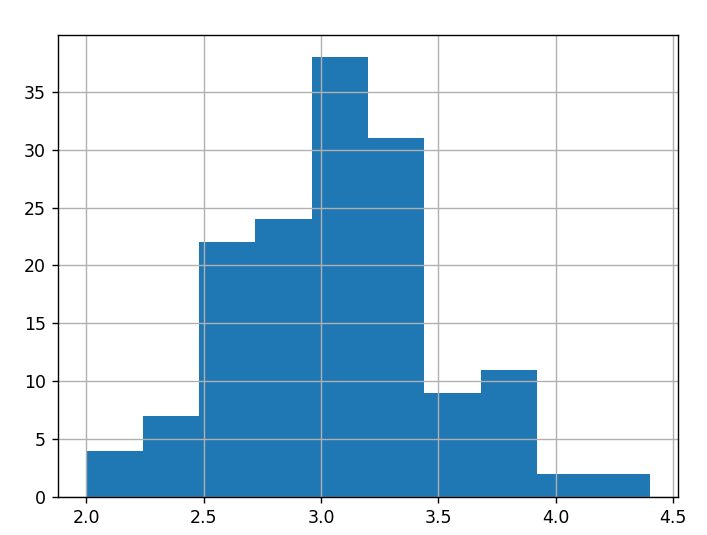
df['SepalWidthCm'].hist()  
plt.show()



df['PetalLengthCm'].hist()  
plt.show()

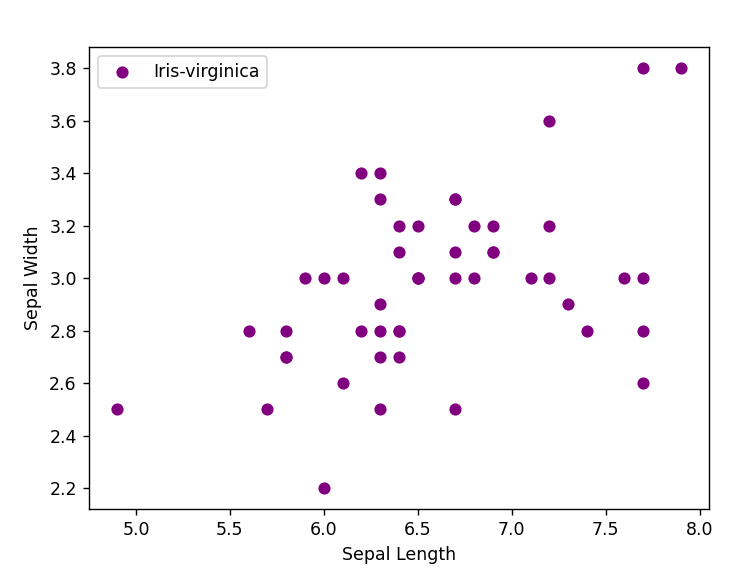


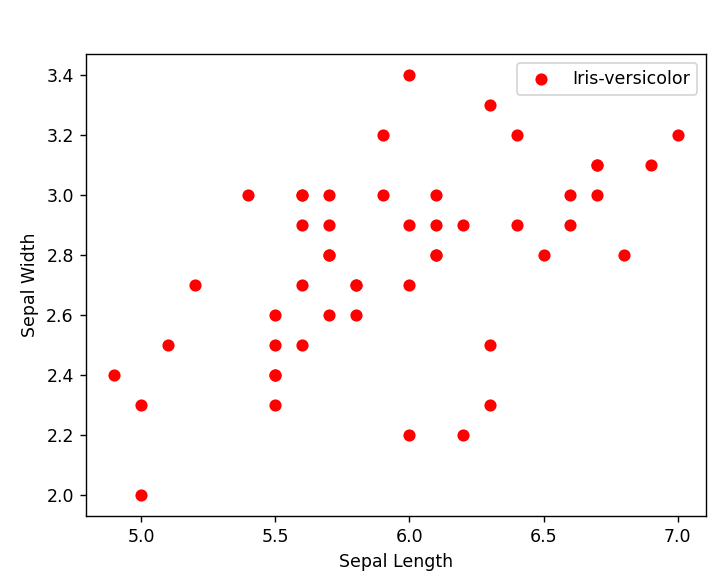
df['SepalWidthCm'].hist()  
plt.show()

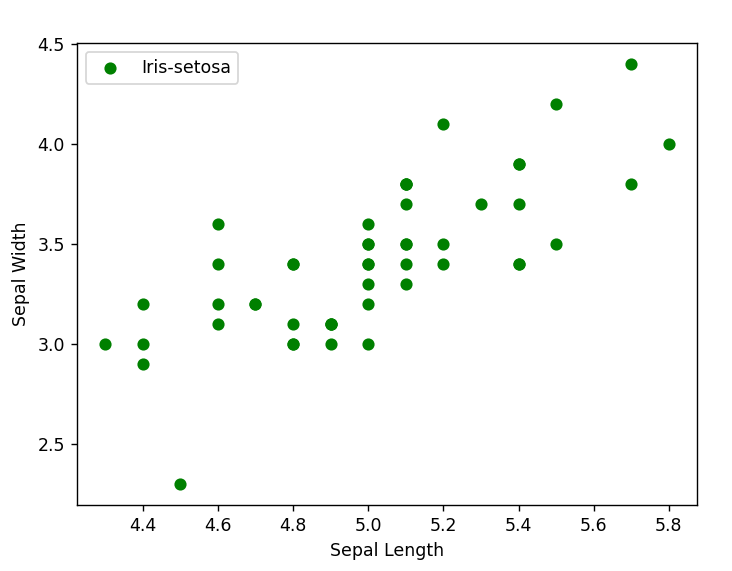


# #scatterplots  
# colors = ['purple', 'red', 'green']  
# species = ['Iris-virginica', 'Iris-versicolor', 'Iris-setosa']

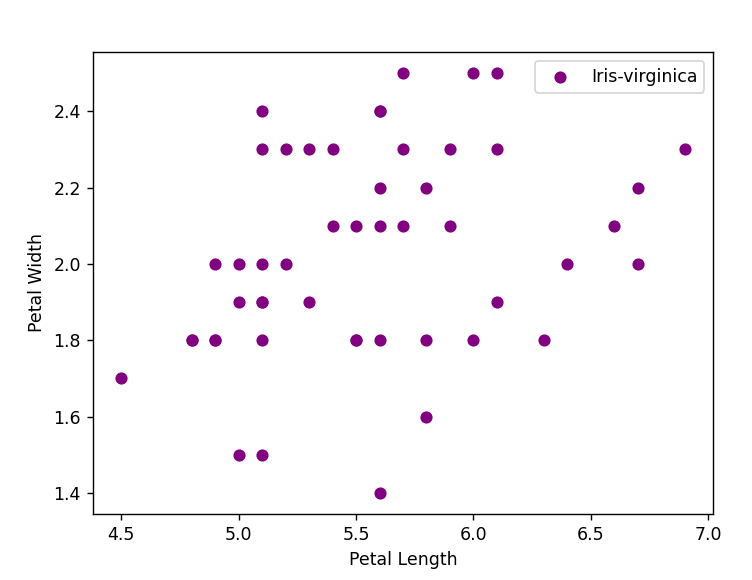
colors = ['purple', 'red', 'green']  
species = ['Iris-virginica', 'Iris-versicolor', 'Iris-setosa']  
  
for i in range(3):  
 x = df[df['Species'] == species[i]]  
 plt.scatter(x['SepalLengthCm'], x['SepalWidthCm'], c = colors[i], label=species[i])  
 plt.xlabel("Sepal Length")  
 plt.ylabel("Sepal Width")  
 plt.legend()  
 plt.show()

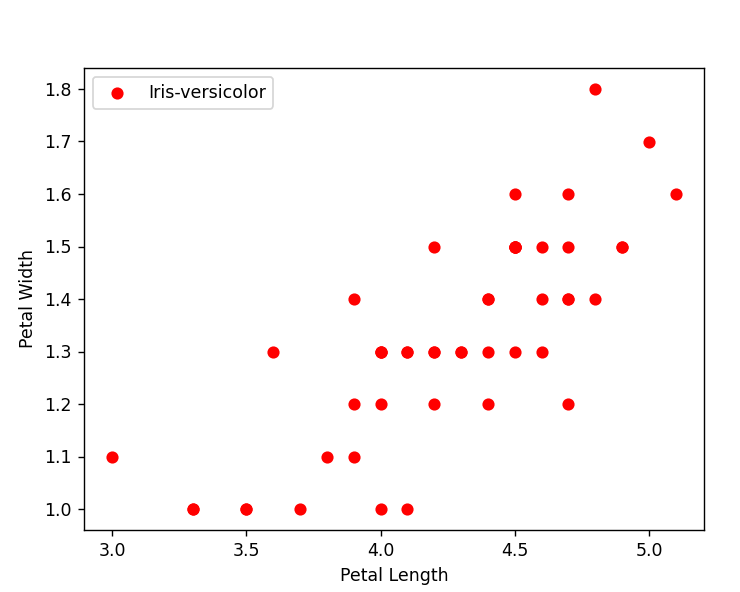


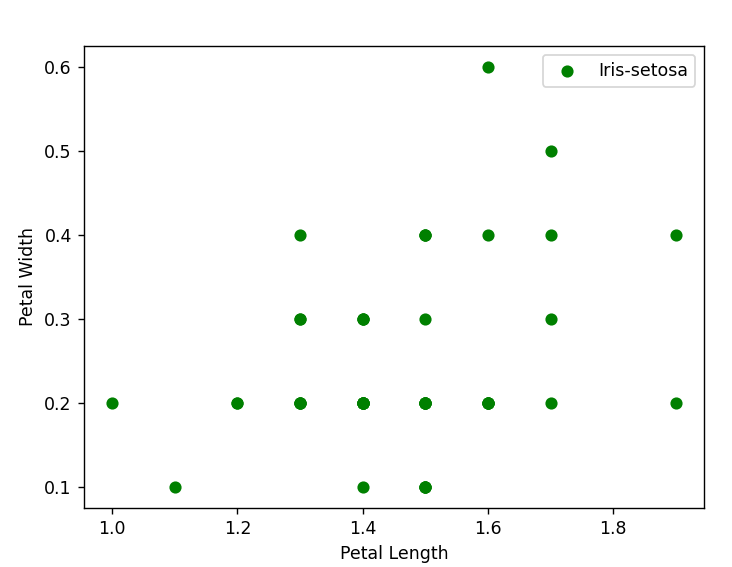




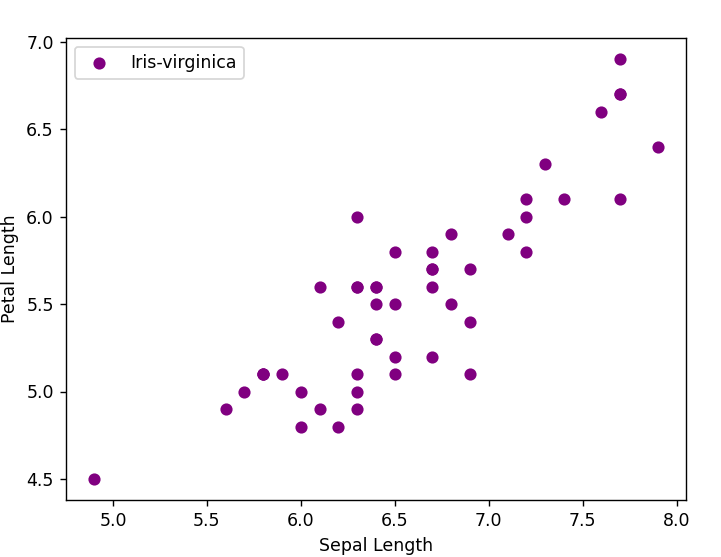
for i in range(3):  
 x = df[df['Species'] == species[i]]  
 plt.scatter(x['PetalLengthCm'], x['PetalWidthCm'], c=colors[i], label=species[i])  
 plt.xlabel("Petal Length")  
 plt.ylabel("Petal Width")  
 plt.legend()  
 plt.show()

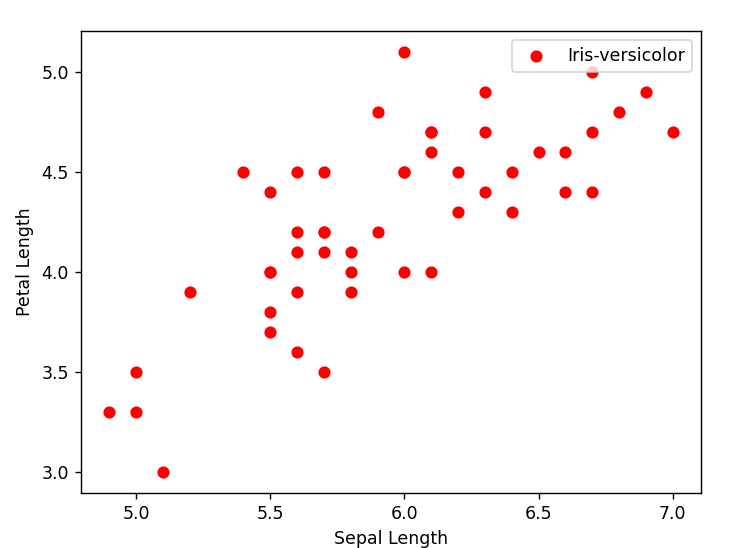


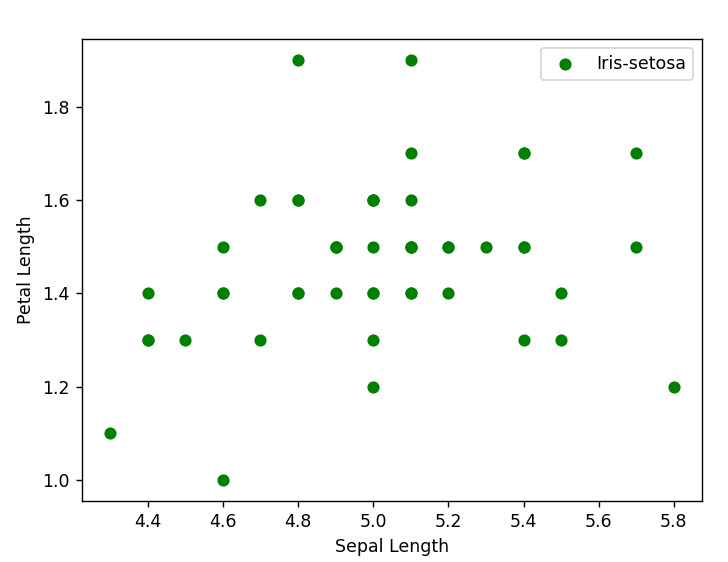




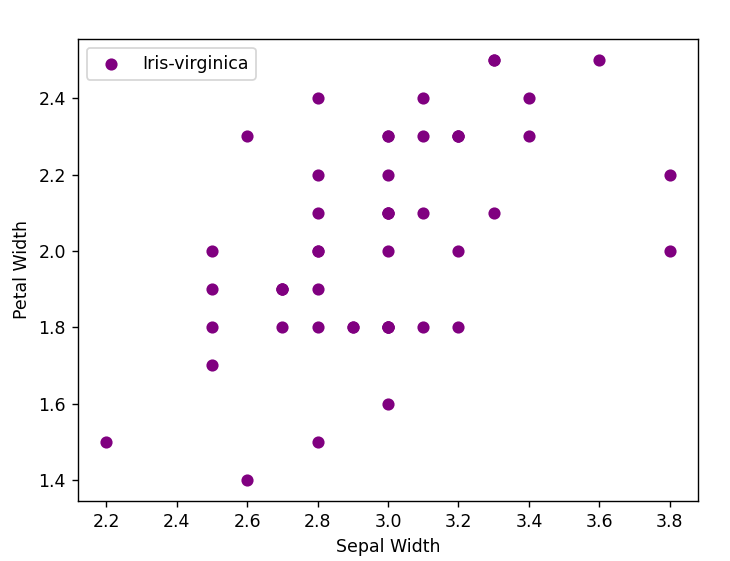
for i in range(3):  
 x = df[df['Species'] == species[i]]  
 plt.scatter(x['SepalLengthCm'], x['PetalLengthCm'], c = colors[i], label=species[i])  
 plt.xlabel("Sepal Length")  
 plt.ylabel("Petal Length")  
 plt.legend()  
 plt.show()

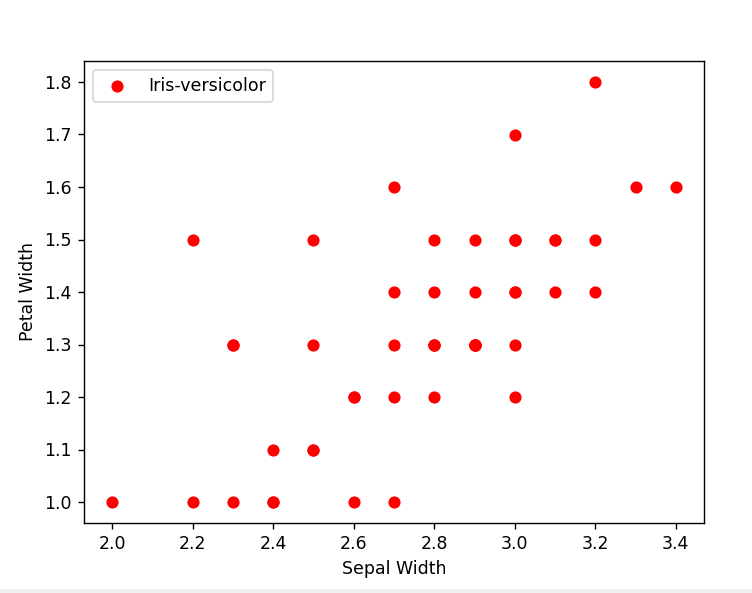


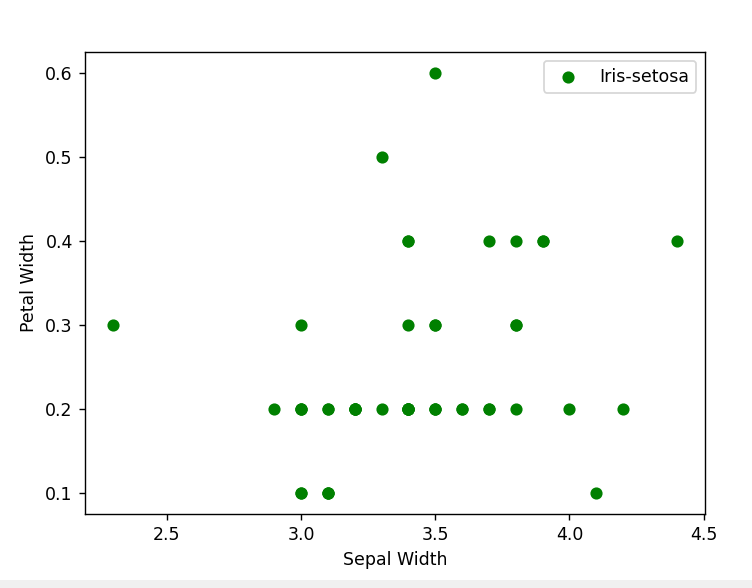




for i in range(3):  
 x = df[df['Species'] == species[i]]  
 plt.scatter(x['SepalWidthCm'], x['PetalWidthCm'], c = colors[i], label=species[i])  
 plt.xlabel("Sepal Width")  
 plt.ylabel("Petal Width")  
 plt.legend()  
 plt.show()







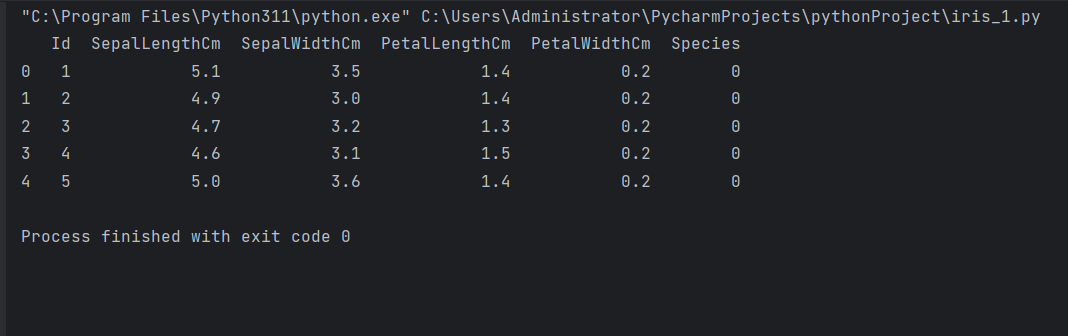
**Label Encoding:**

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['Species'] = le.fit\_transform(df['Species'])

print(df.head())

The 'Species' column is encoded using LabelEncoder, converting categorical labels into numeric form.

**Model Training**

from sklearn.model\_selection import train\_test\_split

x = df.drop(columns=['Species'])

y = df['Species']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.30)

**Insights Derived and Advanced Analysis:**

The initial part of the code focuses on loading the dataset, performing basic data cleaning by dropping unnecessary columns, and checking for missing values.

Descriptive statistics provide an overview of the central tendency, dispersion, and shape of the dataset.

Exploratory Data Analysis (EDA) involves visualizations like histograms and scatterplots to understand the distribution and relationships between different features.

Label Encoding is applied to convert categorical labels into a format suitable for machine learning algorithms.

The model training section is partially commented out, leaving room for further development and model training using techniques like logistic regression.