```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, classification_report

file_path ='/content/sample_data/project (2).csv'
df = pd.read_csv(file_path)

# Explore the dataset (optional)
print(df.head())

# Separate features (X) and target variable (y)
X = df.drop('Species', axis=1)  # Assuming 'species' is the target variable
y = df['Species']

# Split the data 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)

# Choose a machine learning algorithm (Decision Trees in this example)
model = DecisionTreeClassifier(random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
precision = precision_score(y_test, predictions, average='weighted')
recall = recall_score(y_test, predictions, average='weighted')

# Additional metrics and detailed report
classification_report_result = classification_report(y_test, predictions)

# Print results
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print("Classification Report:\n", classification_report_result)
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
Classification Report:
                  precision    recall  f1-score   support

      Iris-setosa       1.00      1.00      1.00        10
  Iris-versicolor       1.00      1.00      1.00         9
   Iris-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
```

New Section

```
import os

if os.path.isfile(r'C:\Users\Rehan Shariff  S M\Downloads\project (2).csv'):
    print("The file exists in the same directory as your notebook.")
else:
    print("The file does not exist in the same directory as your notebook.")
```

```
 The file does not exist in the same directory as your notebook.
```

Double-click (or enter) to edit

```
import os
os.listdir()
```

```
['.config', 'sample_data']
```

```
data=pd.read_csv("/content/sample_data/project (2).csv")
```

```
data.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
data.shape
```

```
(150, 6)
```

```
data.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
data.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

Exploratory Data Analysis

```
data['PetalLengthCm'].unique()
```

```
array([1.4, 1.3, 1.5, 1.7, 1.6, 1.1, 1.2, 1. , 1.9, 4.7, 4.5, 4.9, 4. ,
       4.6, 3.3, 3.9, 3.5, 4.2, 3.6, 4.4, 4.1, 4.8, 4.3, 5. , 3.8, 3.7,
       5.1, 3. , 6. , 5.9, 5.6, 5.8, 6.6, 6.3, 6.1, 5.3, 5.5, 6.7, 6.9,
       5.7, 6.4, 5.4, 5.2])
```

```
data['SepalLengthCm'].unique()
```

```
array([5.1, 4.9, 4.7, 4.6, 5. , 5.4, 4.4, 4.8, 4.3, 5.8, 5.7, 5.2, 5.5,
       4.5, 5.3, 7. , 6.4, 6.9, 6.5, 6.3, 6.6, 5.9, 6. , 6.1, 5.6, 6.7,
       6.2, 6.8, 7.1, 7.6, 7.3, 7.2, 7.7, 7.4, 7.9])
```

Cleaning data

```
import seaborn as sns
```

```
data.isnull().sum()
```

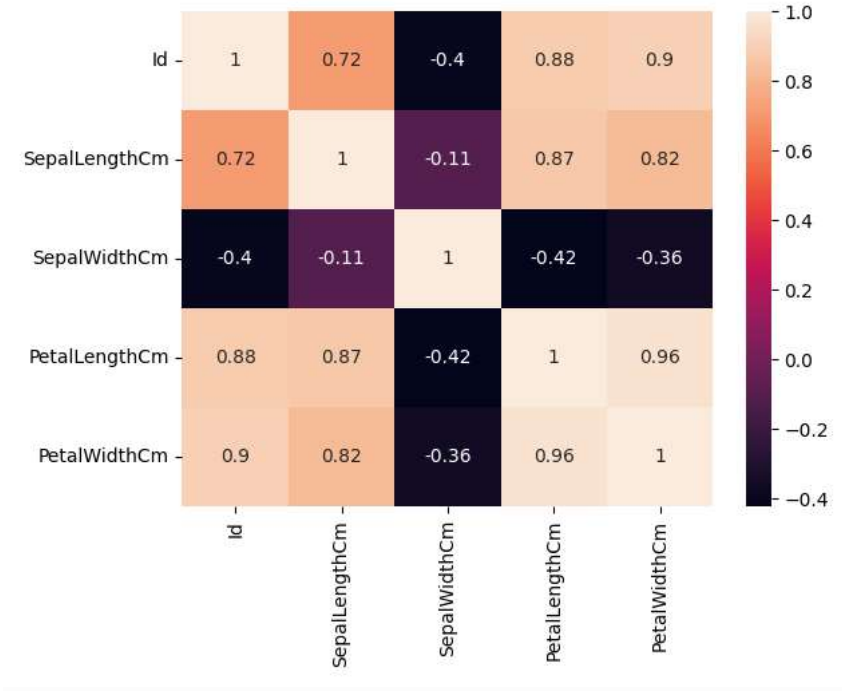
```
Id              0
SepalLengthCm   0
SepalWidthCm    0
PetalLengthCm   0
PetalWidthCm    0
```

```
Species          0
dtype: int64
```

Relationship Analysis

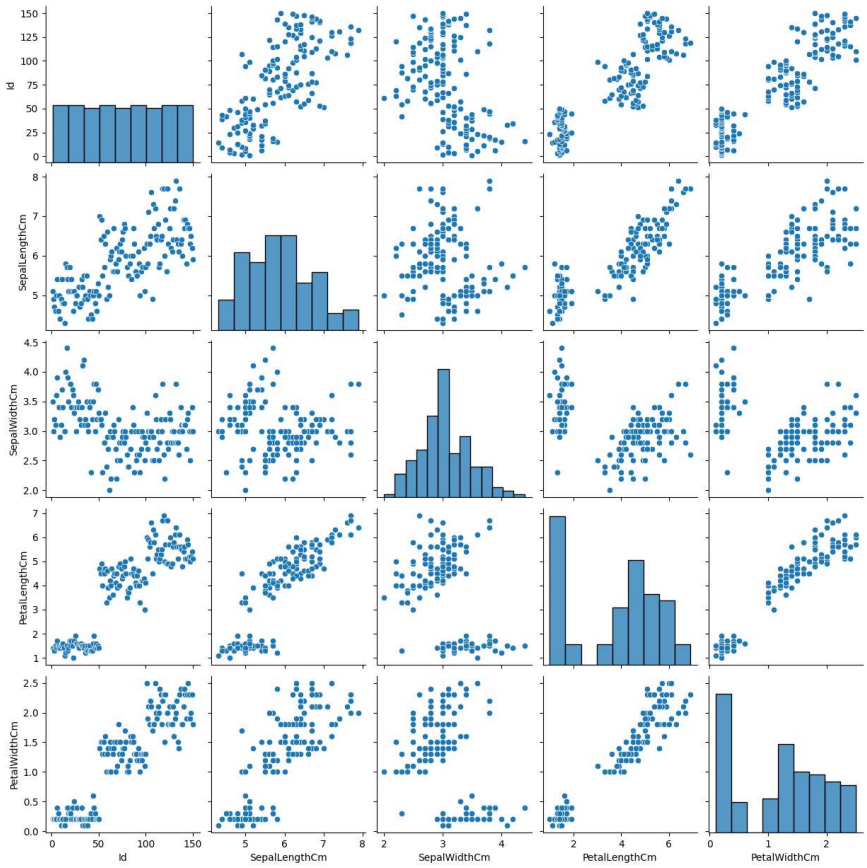
```
corelation=data.corr()
sns.heatmap(corelation,xticklabels=corelation.columns,yticklabels=corelation.columns,annot=True)
```

```
<ipython-input-65-8cd259c7fbe3>:1: FutureWarning: The default value of numeric_only i
  corelation=data.corr()
<Axes: >
```



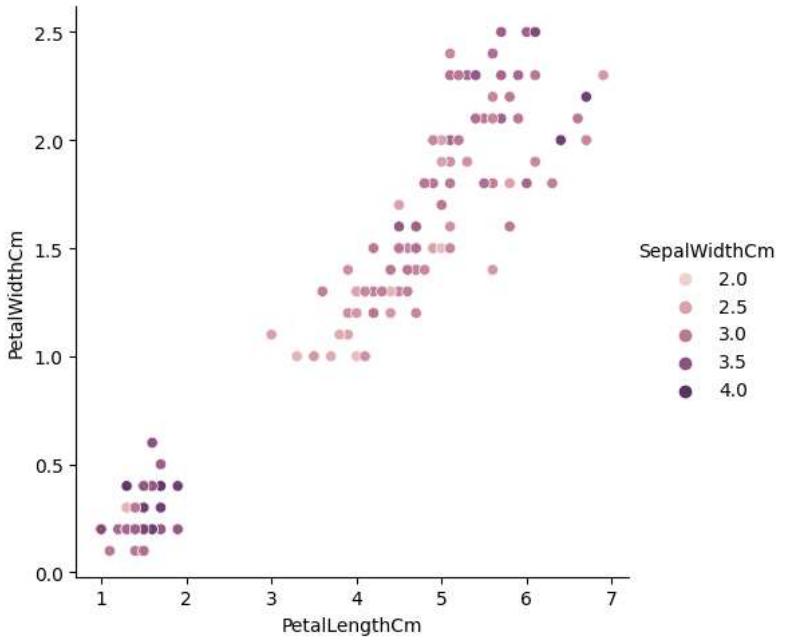

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f6a74e6ffa0>
```



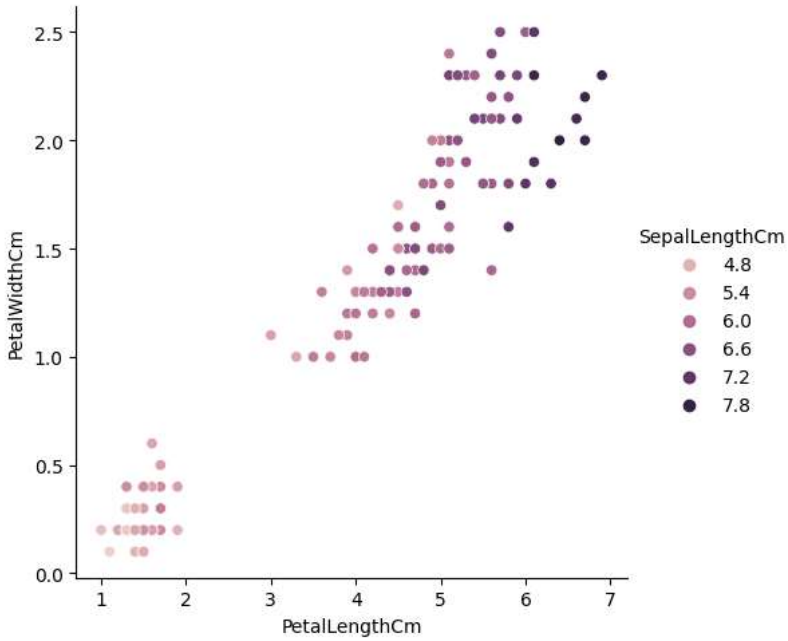

```
sns.relplot(x='PetalLengthCm',y='PetalWidthCm',hue='SepalWidthCm',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc1284039a0>
```



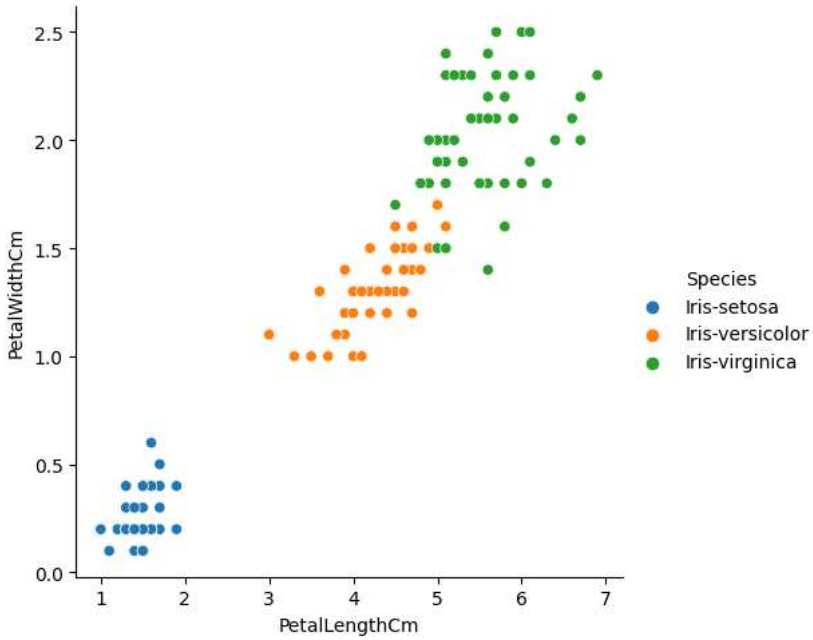

```
sns.relplot(x='PetalLengthCm',y='PetalWidthCm',hue='SepalLengthCm',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc12830c4c0>
```



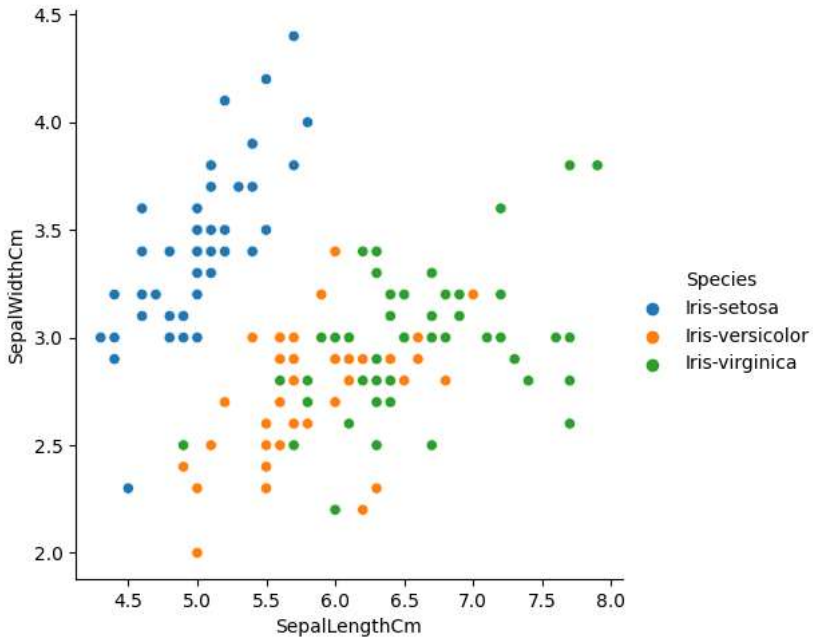

```
sns.relplot(x='PetalLengthCm',y='PetalWidthCm',hue='Species',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc12829cf10>
```



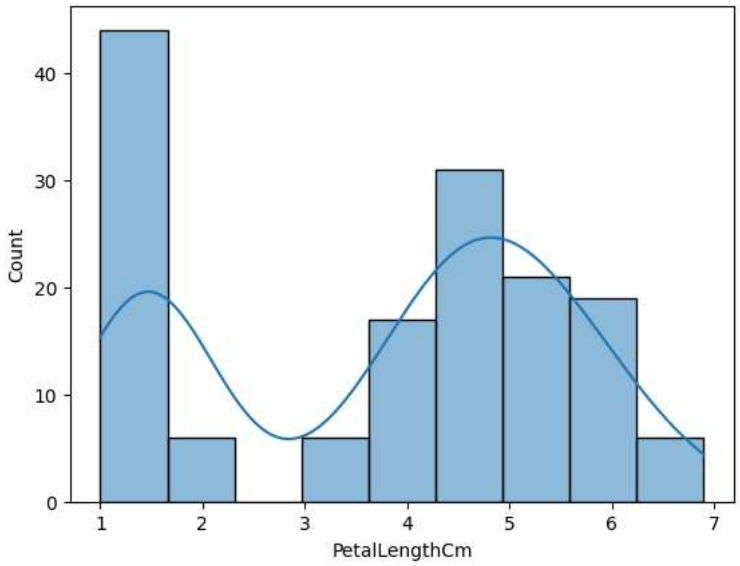

```
sns.relplot(x='SepalLengthCm',y='SepalWidthCm',hue='Species',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc12475d390>
```



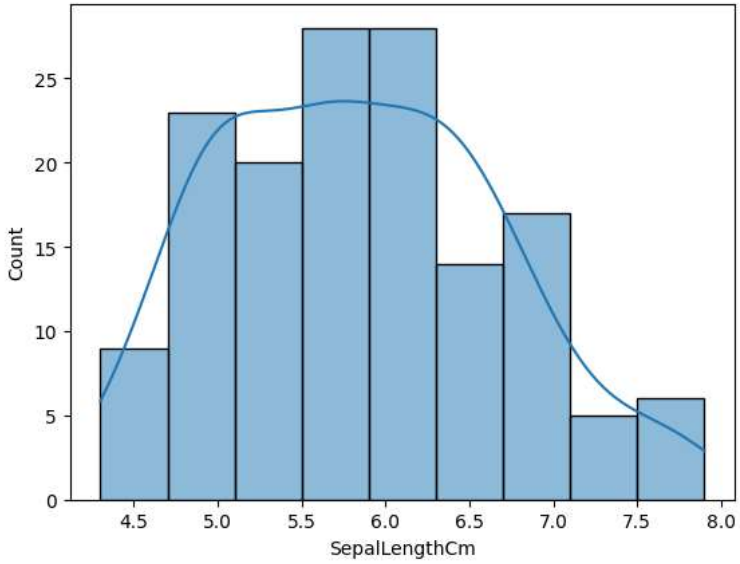

```
sns.histplot(data['PetalLengthCm'], kde=True)
```

```
<Axes: xlabel='PetalLengthCm', ylabel='Count'>
```



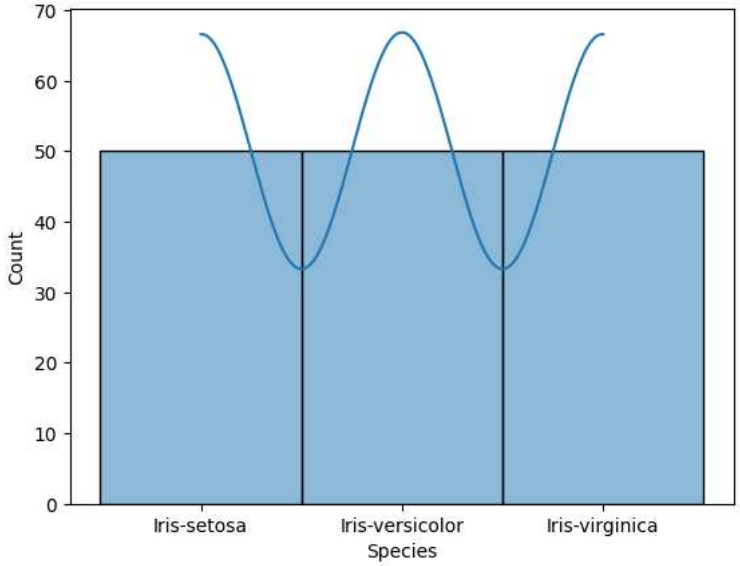

```
sns.histplot(data['SepalLengthCm'], kde=True)
```

```
<Axes: xlabel='SepalLengthCm', ylabel='Count'>
```



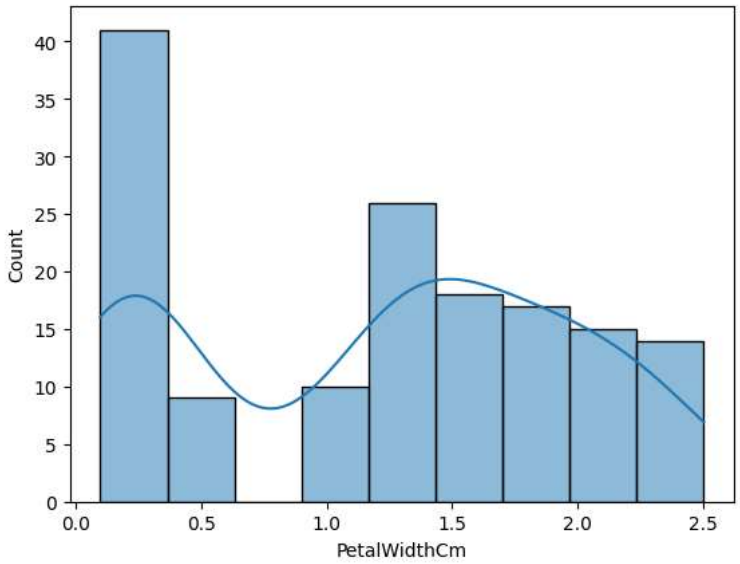

```
sns.histplot(data['Species'], kde=True)
```

```
<Axes: xlabel='Species', ylabel='Count'>
```



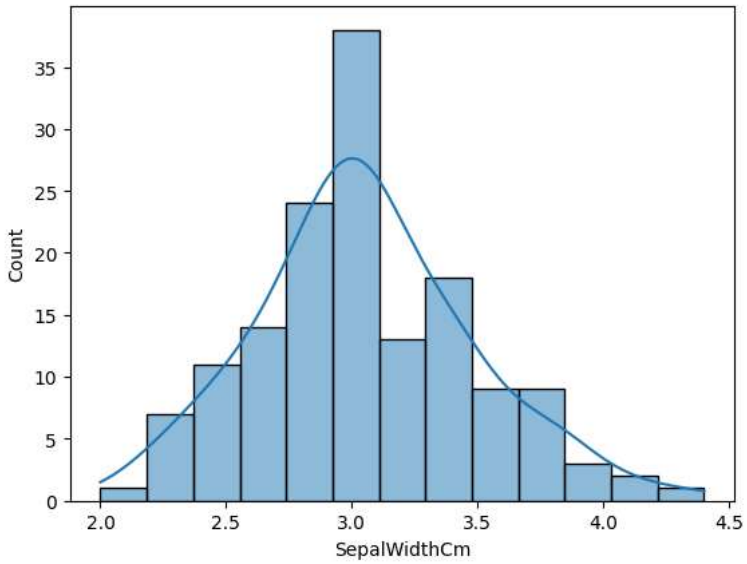
```
sns.histplot(data['PetalWidthCm'], kde=True)
```

```
<Axes: xlabel='PetalWidthCm', ylabel='Count'>
```




```
sns.histplot(data['SepalWidthCm'], kde=True)
```

```
<Axes: xlabel='SepalWidthCm', ylabel='Count'>
```




```
sns.distplot(data['PetalWidthCm'])
```

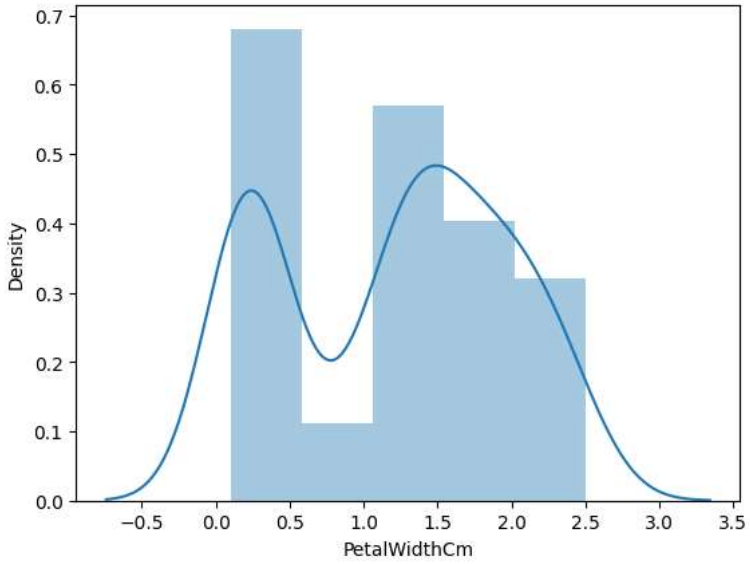
```
<ipython-input-27-5f29ad1397c0>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['PetalWidthCm'])
<Axes: xlabel='PetalWidthCm', ylabel='Density'>
```




```
sns.distplot(data['PetalLengthCm'])
```

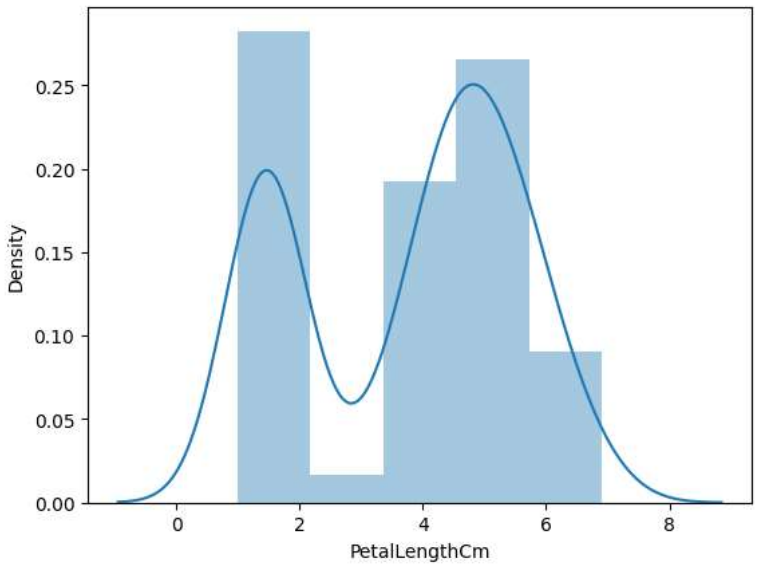
```
<ipython-input-29-888e02e458eb>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['PetalLengthCm'])
<Axes: xlabel='PetalLengthCm', ylabel='Density'>
```




```
sns.distplot(data['SepalLengthCm'])
```

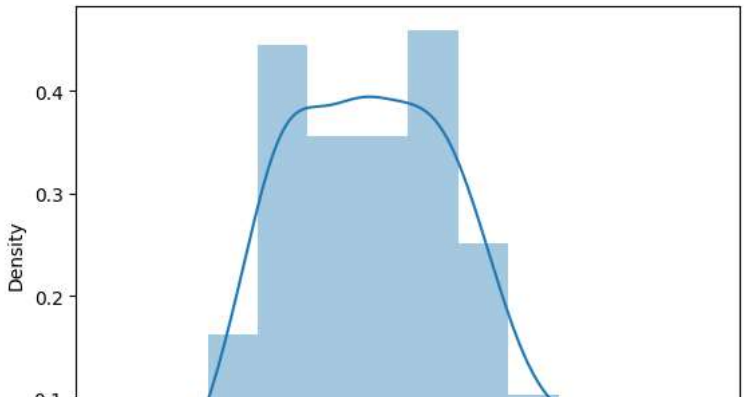
```
<ipython-input-30-cc4dca1bb6c5>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['SepalLengthCm'])
<Axes: xlabel='SepalLengthCm', ylabel='Density'>
```




```
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Load data from CSV file
data = pd.read_csv('')

# Extract data columns
Species = df['Species']
PetalLengthCm = df['PetalLengthCm']
SepalLengthCm = df['SepalLengthCm']
PetalWidthCm = df['PetalWidthCm']
SepalWidthCm = df['SepalWidthCm']

# Create a 3D plot
fig = plt.figure(figsize=(17,6))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot
```