In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
import csv
from sklearn.neighbors import KNeighborsClassifier
import xgboost as xgb
from sklearn.ensemble import RandomForestClassifier
import scipy.stats as stats
from sklearn import metrics
from sklearn.metrics import classification_report
import warnings
```

In [2]:
```
df = pd.read_csv('C:\\Users\\saswa\\OneDrive\\Desktop\\Pinaki-Iris-flower-classi
```

In [3]:
```
df.head(10)
```

Out[3]:

	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
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

	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

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [7]:
```
df["Species"].value_counts()
```

Out[7]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [8]:
```
df.columns
```

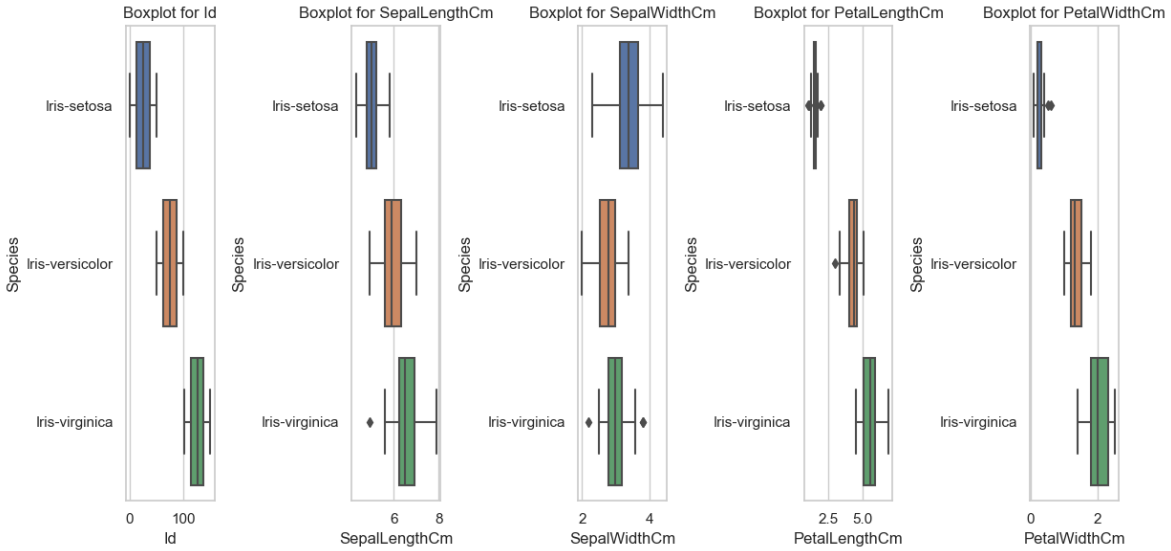
Out[8]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [9]:
```
df["SepalLengthCm"]=df["SepalLengthCm"].round(3)
df["SepalWidthCm"]=df["SepalWidthCm"].round(3)
df["PetalLengthCm"]=df["PetalLengthCm"].round(3)
df["PetalWidthCm"]=df["PetalWidthCm"].round(3)
```

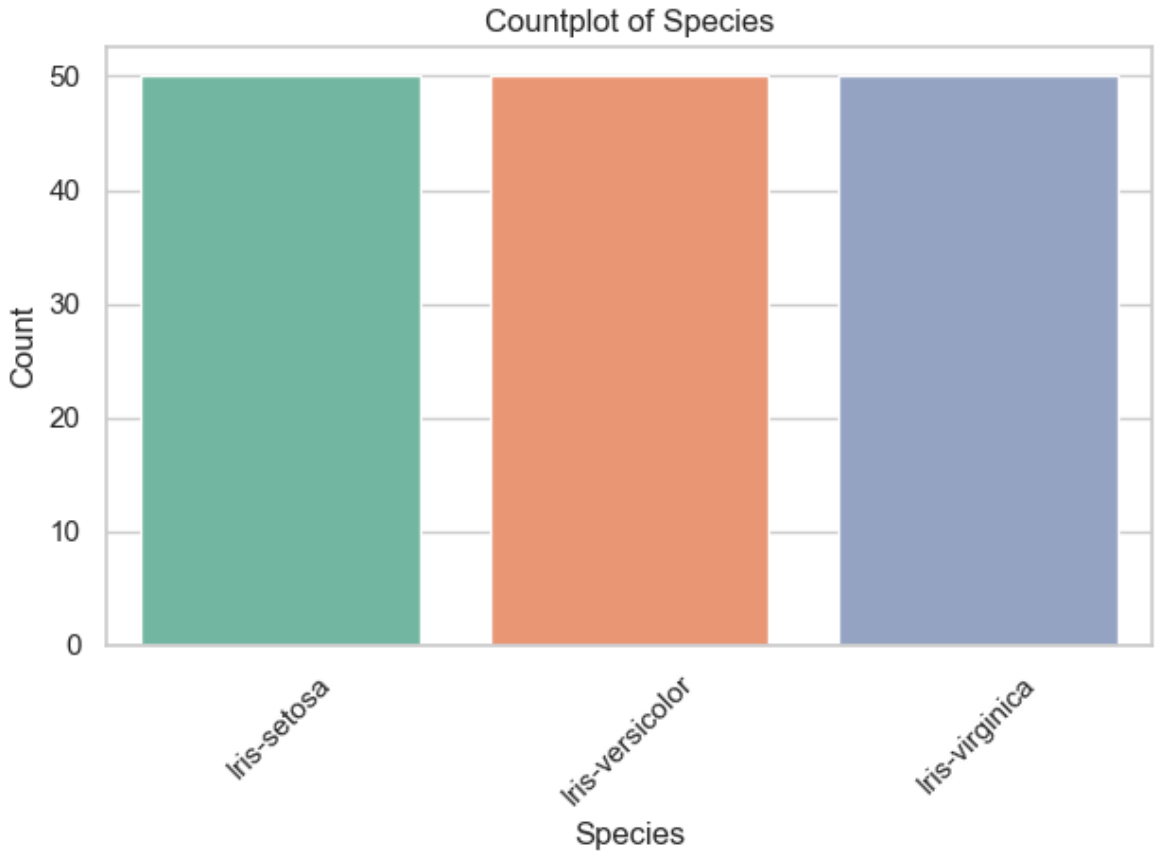
In [10]:
```
numerical_data = []
object_data = []

for column in df.columns:
    if df.dtypes[column] != 'object':
        numerical_data.append(column)
    else:
        object_data.append(column)
```

In [11]:
```
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")

for i in range(len(numerical_data)):
    plt.subplot(1, len(numerical_data), i + 1)
```

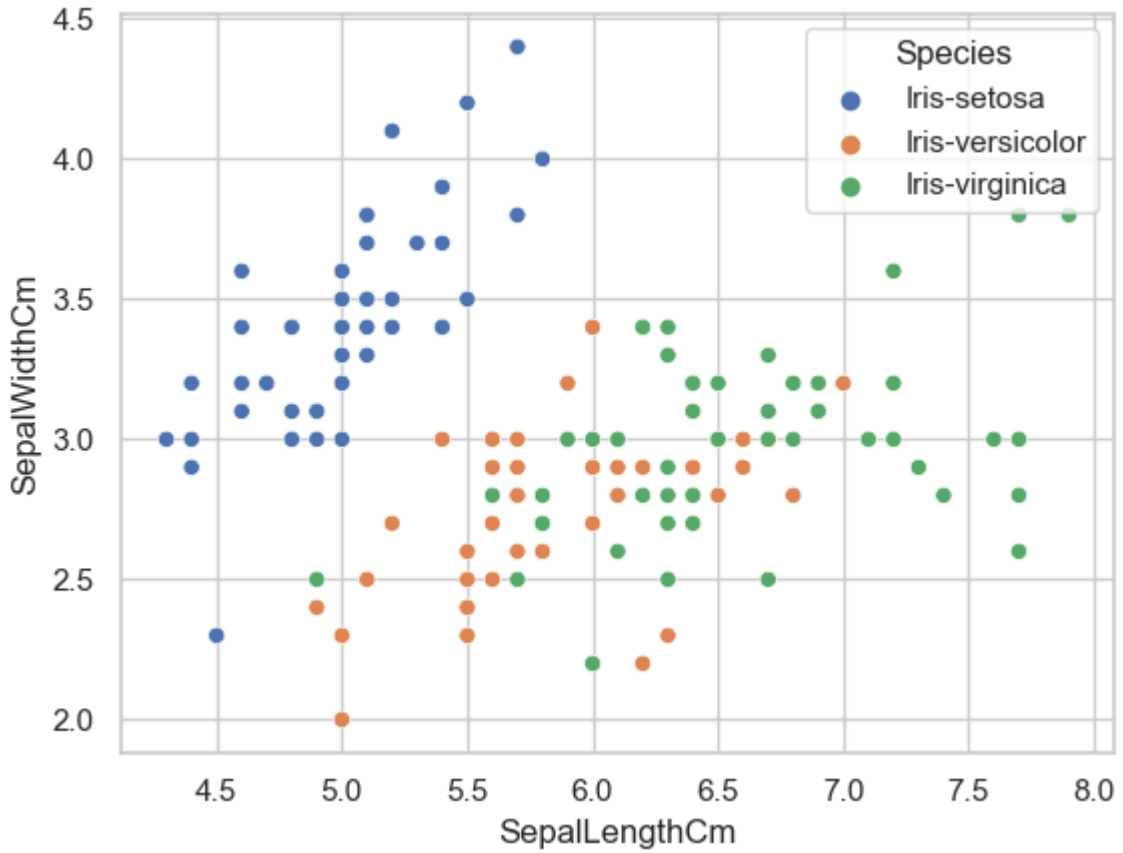
```
    sns.boxplot(x=numerical_data[i], y='Species', data=df, orient="h")
    plt.title(f'Boxplot for {numerical_data[i]}')
plt.tight_layout()
plt.show()
```



In [12]:
```
sns.countplot(data=df, x="Species", palette="Set2")

plt.title("Countplot of Species")
plt.xlabel("Species")
plt.ylabel("Count")

plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```

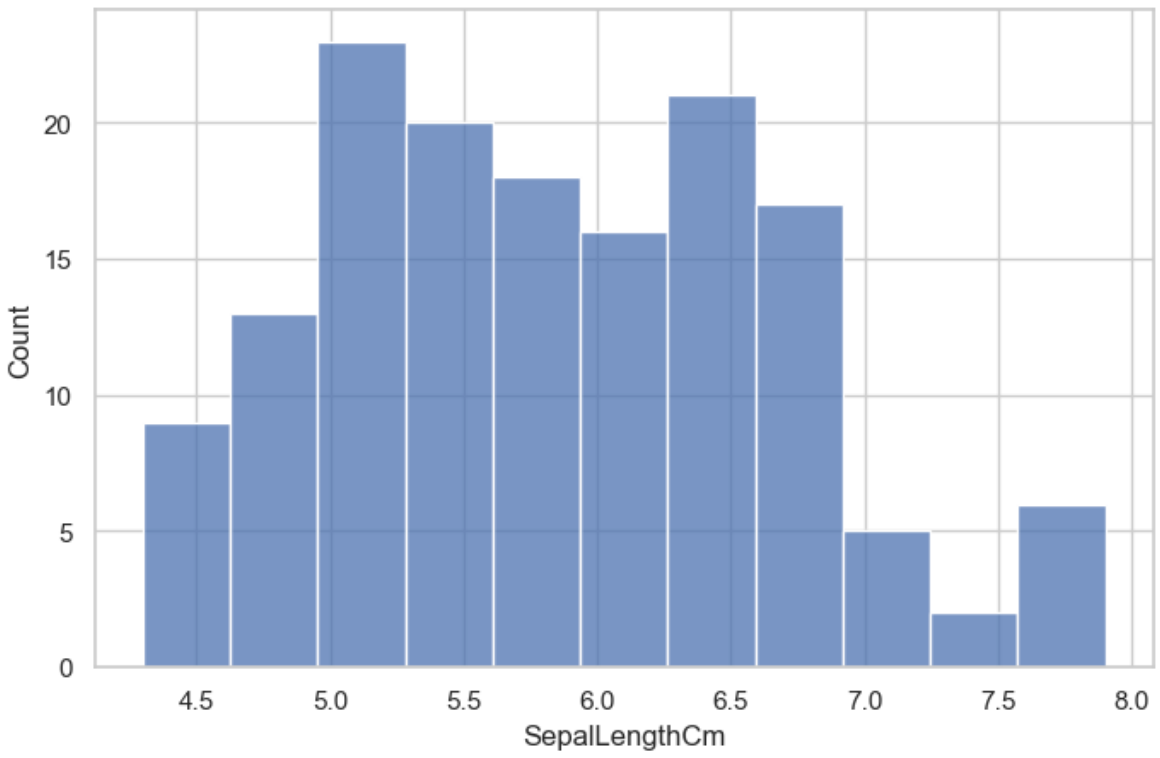


In [13]:
```
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', hue='Species', data=df)
plt.show()
```



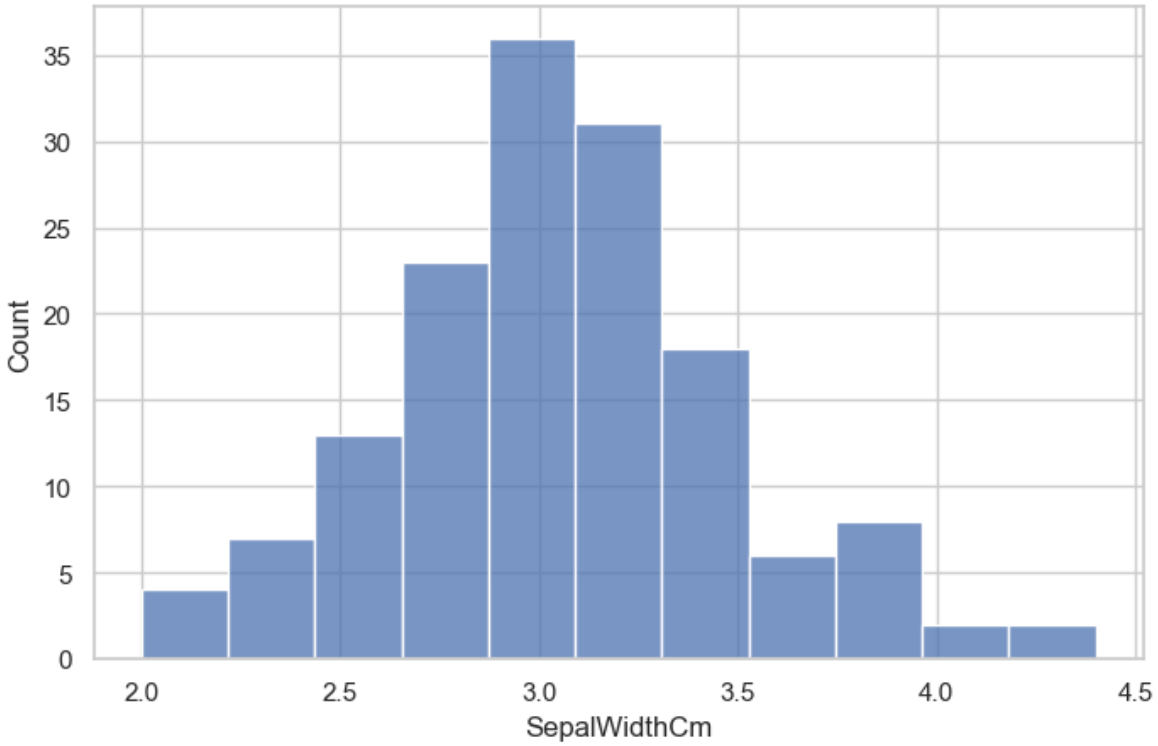
In [14]:
```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='SepalLengthCm', bins=11)
```

Out[14]: <Axes: xlabel='SepalLengthCm', ylabel='Count'>



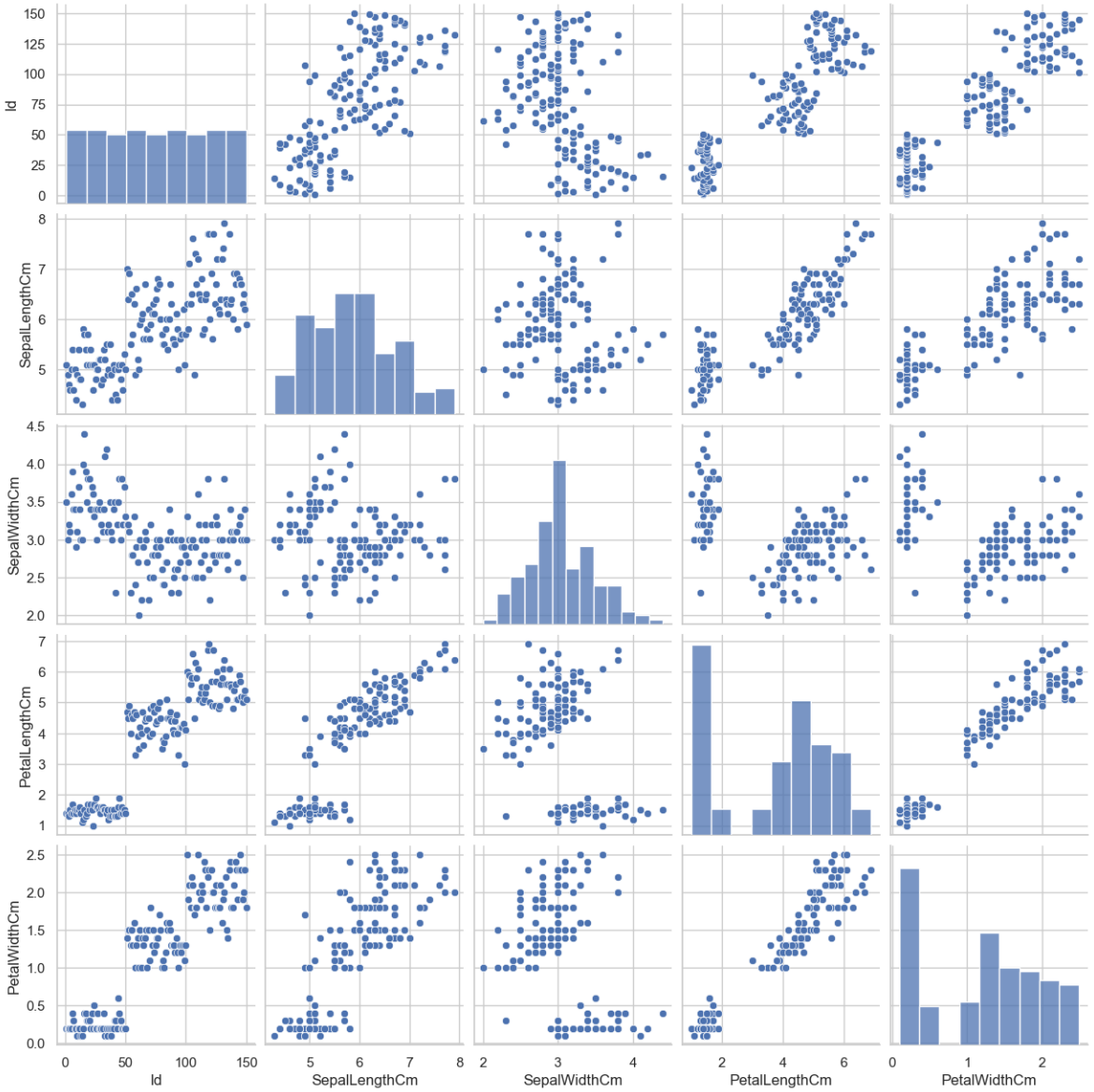
In [15]:
```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='SepalWidthCm', bins=11)
```

Out[15]: <Axes: xlabel='SepalWidthCm', ylabel='Count'>

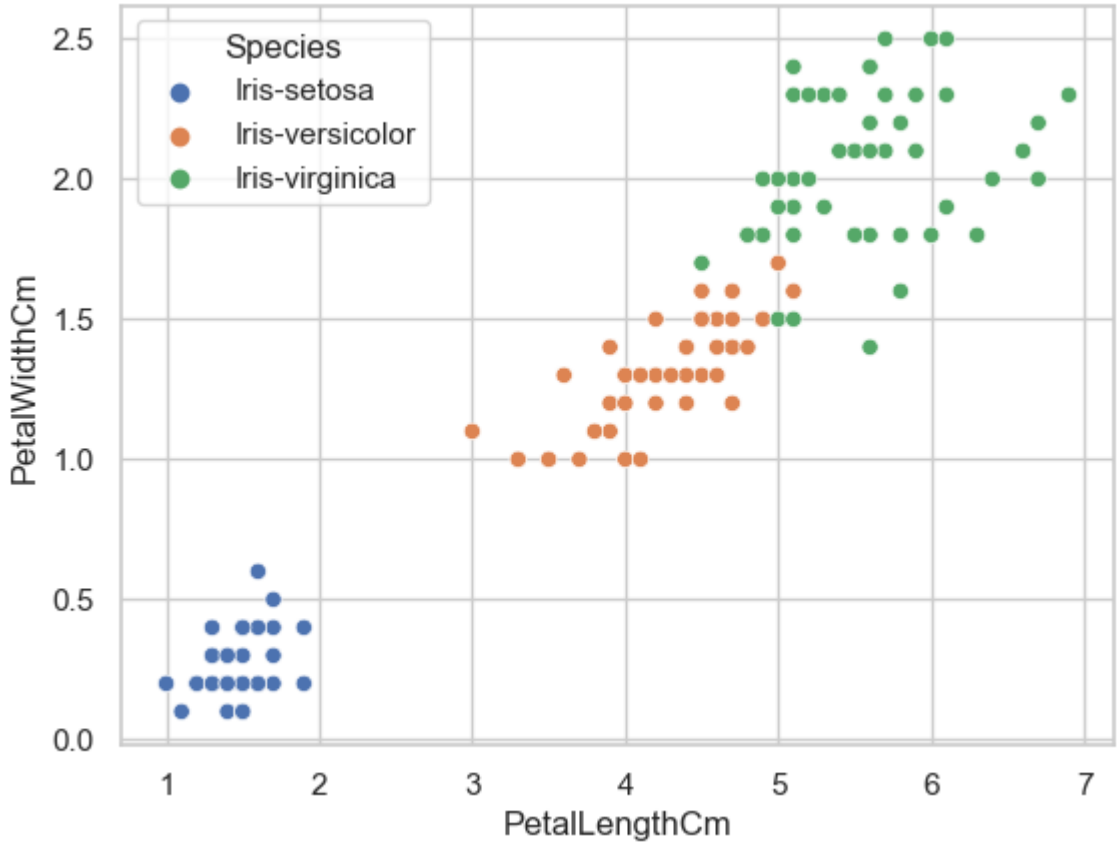


In [16]:
```
sns.pairplot(df)
```

Out[16]: <seaborn.axisgrid.PairGrid at 0x27e5d6e0b90>

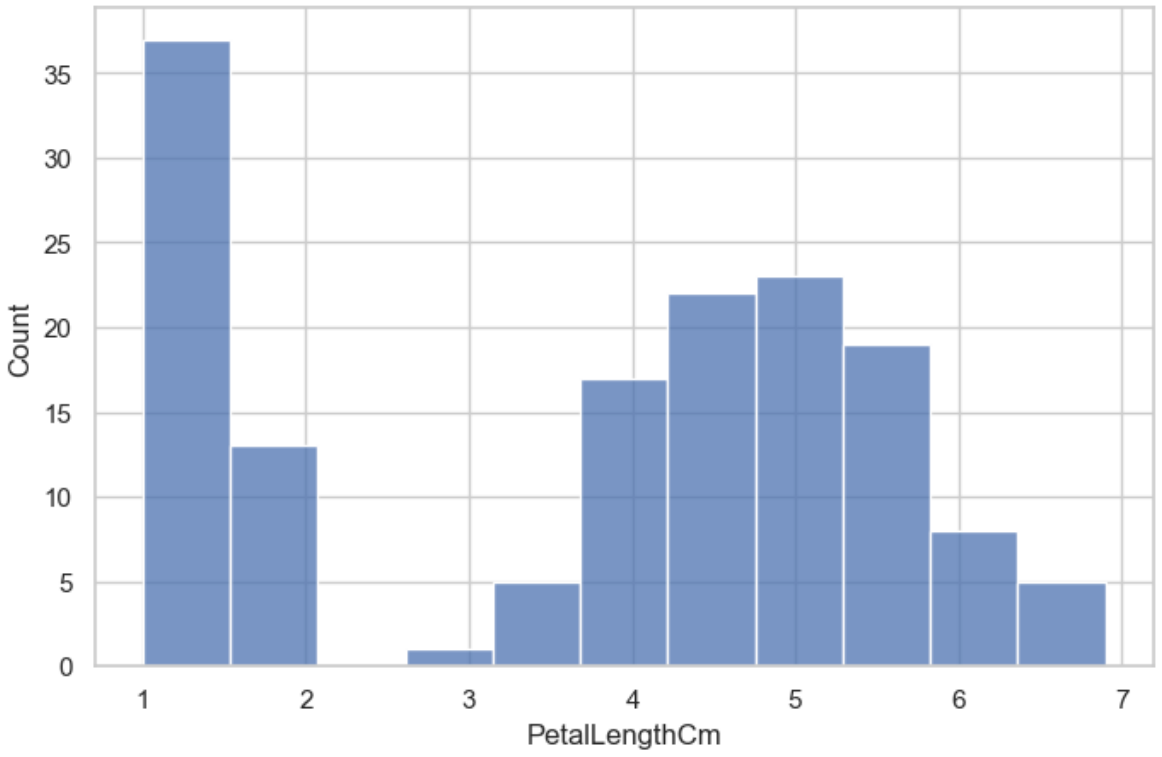


In [17]:
```
sns.scatterplot(x='PetalLengthCm', y='PetalWidthCm', hue='Species', data=df)
plt.show()
```



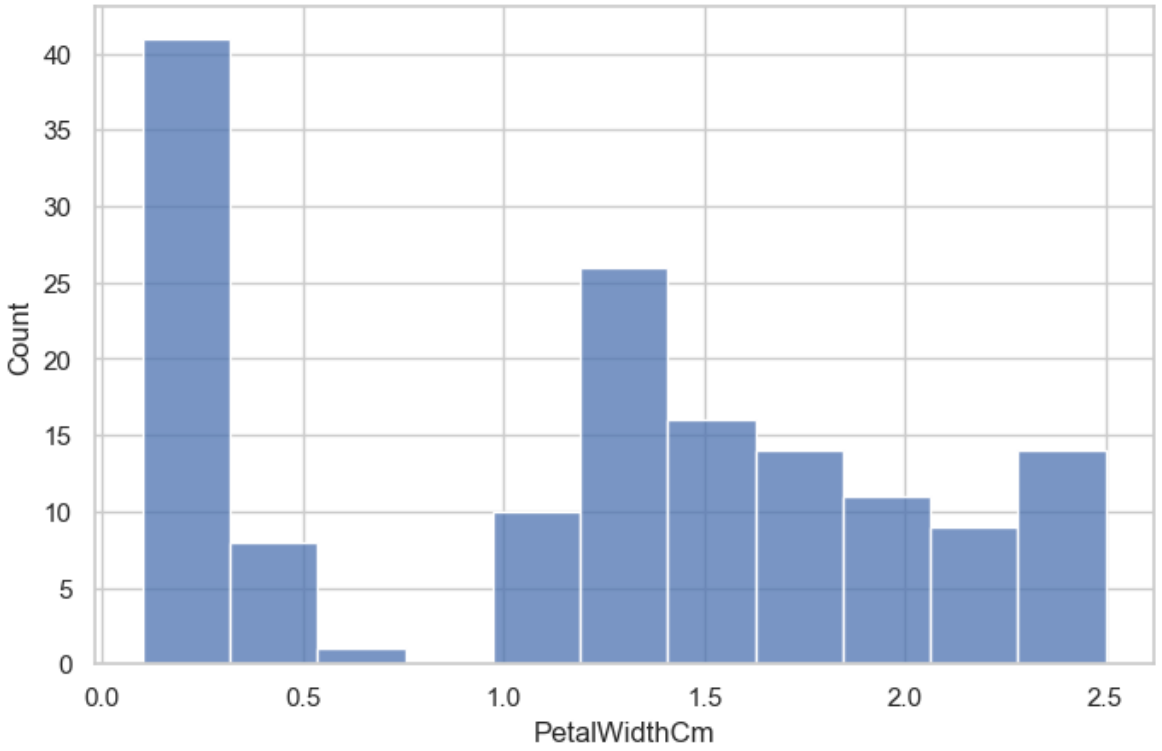
In [18]:
```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='PetalLengthCm', bins=11)
```

Out[18]: <Axes: xlabel='PetalLengthCm', ylabel='Count'>



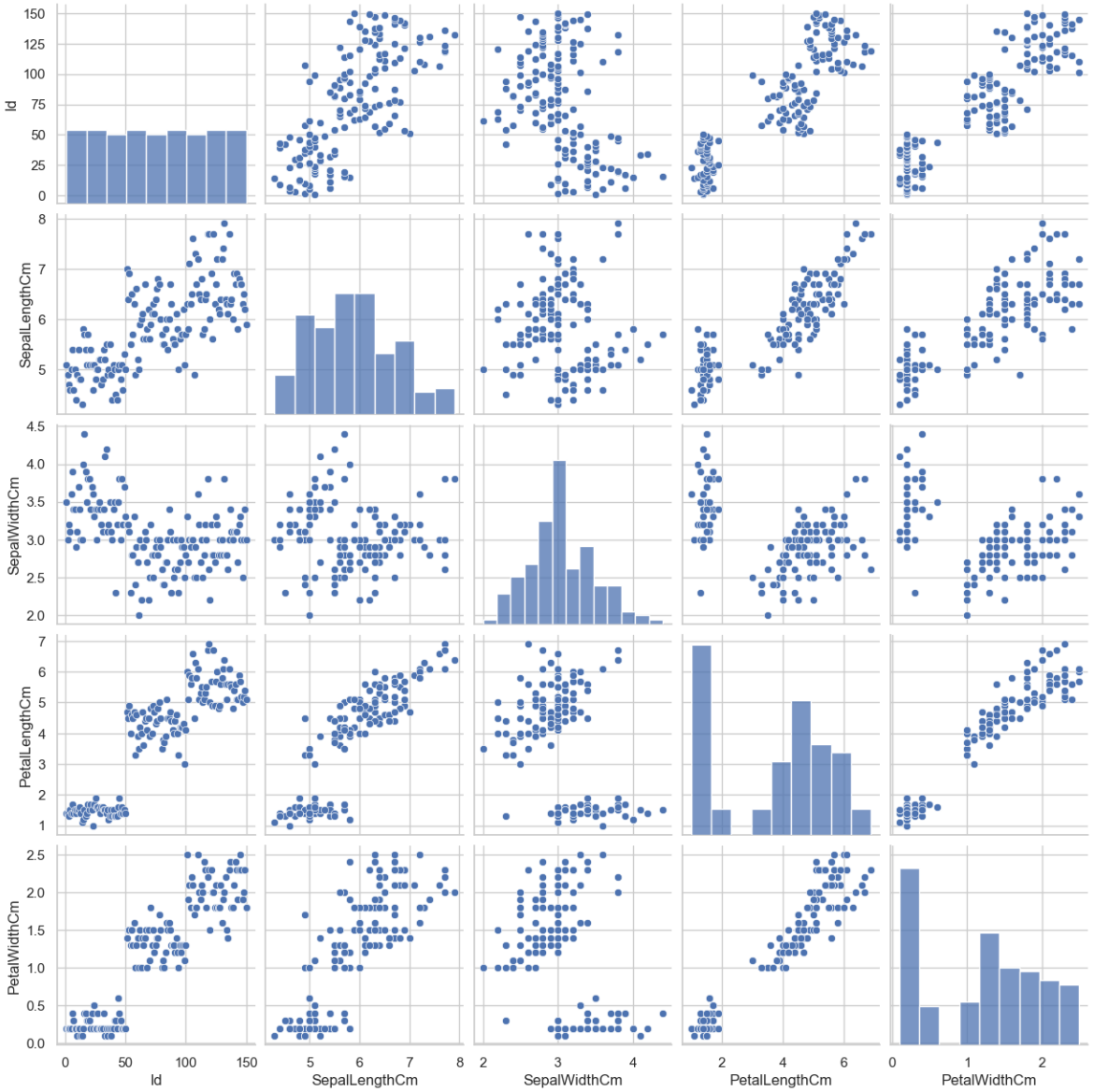
In [19]:
```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='PetalWidthCm', bins=11)
```

Out[19]: <Axes: xlabel='PetalWidthCm', ylabel='Count'>



In [20]:
```
sns.pairplot(df)
```

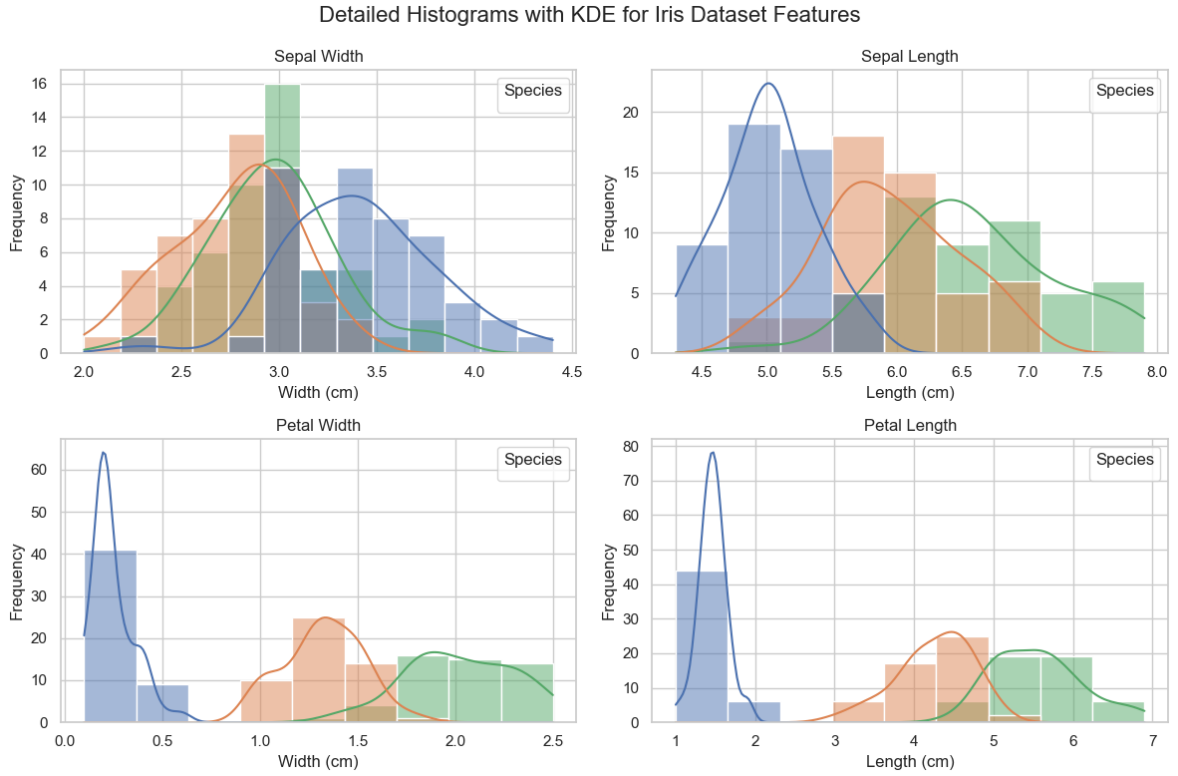
Out[20]:
```
<seaborn.axisgrid.PairGrid at 0x27e5fd20390>
```



In [21]:
```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Detailed Histograms with KDE for Iris Dataset Features", fontsize=

plot_settings = {
    "SepalWidthCm": {"title": "Sepal Width", "xlabel": "Width (cm)"},
    "SepalLengthCm": {"title": "Sepal Length", "xlabel": "Length (cm)"},
    "PetalWidthCm": {"title": "Petal Width", "xlabel": "Width (cm)"},
    "PetalLengthCm": {"title": "Petal Length", "xlabel": "Length (cm)"}
}

for i, feature in enumerate(plot_settings.keys()):
    row, col = divmod(i, 2)
    ax = axes[row, col]

    sns.histplot(data=df, x=feature, hue='Species', kde=True, ax=ax)

    ax.set_title(plot_settings[feature]["title"])
    ax.set_xlabel(plot_settings[feature]["xlabel"])
    ax.set_ylabel("Frequency")

    ax.legend(loc='upper right', title='Species')

plt.tight_layout()
plt.subplots_adjust(top=0.9)
```

```
plt.show()
```

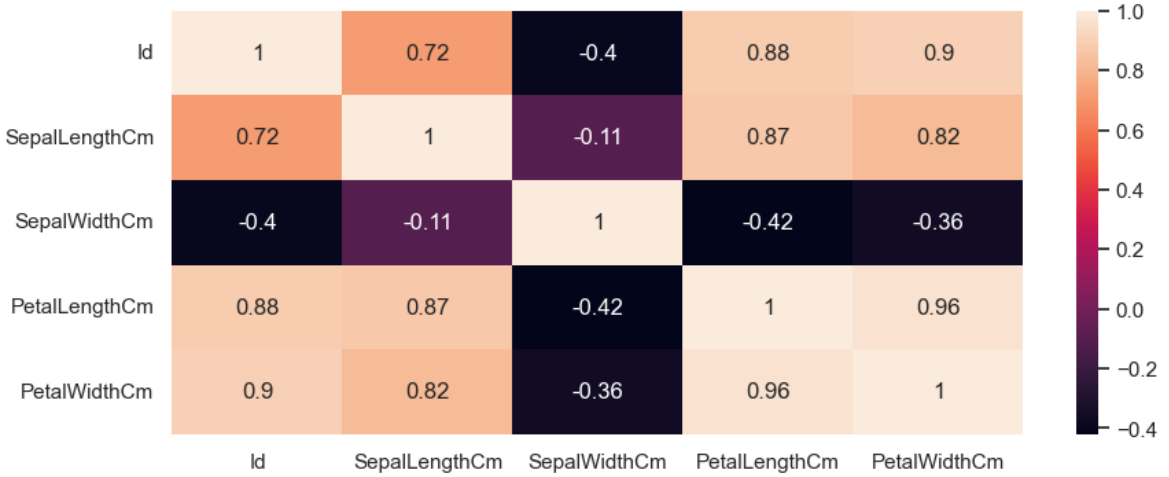
```
No artists with labels found to put in legend.  Note that artists whose label sta
rt with an underscore are ignored when legend() is called with no argument.
No artists with labels found to put in legend.  Note that artists whose label sta
rt with an underscore are ignored when legend() is called with no argument.
No artists with labels found to put in legend.  Note that artists whose label sta
rt with an underscore are ignored when legend() is called with no argument.
No artists with labels found to put in legend.  Note that artists whose label sta
rt with an underscore are ignored when legend() is called with no argument.
```



In [22]:
```
plt.figure(figsize=(10,4))
sns.heatmap(df.corr(),annot=True)
plt.show
```

```
C:\Users\saswa\AppData\Local\Temp\ipykernel_2844\660786042.py:2: FutureWarning: T
he default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

Out[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [23]:
```
X = df.drop(['Species',"SepalWidthCm"],axis=1)
y = df['Species']
```

In [24]:
```
sc = StandardScaler()
X = sc.fit_transform(X)
```

In [25]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random
RF = RandomForestClassifier(n_estimators=100, random_state=42)
RF.fit(X_train, y_train)
```

Out[25]:
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [26]:
```
print(RF.score(X_train,y_train))
print(RF.score(X_test,y_test))
```

```
1.0
1.0
```

In [27]:
```
y_pred_rf=RF.predict(X_test)
y_pred_rf
```

Out[27]:
```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa'], dtype=object)
```

In [28]:
```
df=pd.DataFrame({"y_pred_rf":y_pred_rf,"y_test":y_test})
df
```

Out[28]:

	y_pred_rf	y_test
73	Iris-versicolor	Iris-versicolor
18	Iris-setosa	Iris-setosa
118	Iris-virginica	Iris-virginica
78	Iris-versicolor	Iris-versicolor
76	Iris-versicolor	Iris-versicolor
31	Iris-setosa	Iris-setosa
64	Iris-versicolor	Iris-versicolor
141	Iris-virginica	Iris-virginica
68	Iris-versicolor	Iris-versicolor
82	Iris-versicolor	Iris-versicolor
110	Iris-virginica	Iris-virginica
12	Iris-setosa	Iris-setosa
36	Iris-setosa	Iris-setosa
9	Iris-setosa	Iris-setosa
19	Iris-setosa	Iris-setosa
56	Iris-versicolor	Iris-versicolor
104	Iris-virginica	Iris-virginica
69	Iris-versicolor	Iris-versicolor
55	Iris-versicolor	Iris-versicolor
132	Iris-virginica	Iris-virginica
29	Iris-setosa	Iris-setosa
127	Iris-virginica	Iris-virginica
26	Iris-setosa	Iris-setosa
128	Iris-virginica	Iris-virginica
131	Iris-virginica	Iris-virginica
145	Iris-virginica	Iris-virginica
108	Iris-virginica	Iris-virginica
143	Iris-virginica	Iris-virginica
45	Iris-setosa	Iris-setosa
30	Iris-setosa	Iris-setosa
22	Iris-setosa	Iris-setosa
15	Iris-setosa	Iris-setosa

	y_pred_rf	y_test
65	Iris-versicolor	Iris-versicolor
11	Iris-setosa	Iris-setosa
42	Iris-setosa	Iris-setosa
146	Iris-virginica	Iris-virginica
51	Iris-versicolor	Iris-versicolor
27	Iris-setosa	Iris-setosa

In [29]:
```
scaler=StandardScaler()
X_train=scaler.fit_transform(X_train)
X_train
```

Out[29]:
```
array([[-1.69110144, -1.01827123, -1.39489006, -1.35865217],
       [-1.03515906, -0.7730102 , -1.33696359, -1.49272181],
       [ 1.54175742, -0.03722712,  0.74838929,  0.92053173],
       [ 0.20644615,  0.20803391,  0.40083048,  0.51832281],
       [ 0.22987266,  1.06644751,  0.51668341,  0.38425317],
       [-1.40998328, -0.52774918, -1.45281653, -1.09051288],
       [-1.55054236, -0.52774918, -1.33696359, -1.35865217],
       [ 0.1127401 , -0.40511866, -0.06258127, -0.28609504],
       [ 1.3309188 ,  0.57592545,  0.74838929,  0.38425317],
       [ 1.42462485,  0.69855596,  0.98009517,  0.78646209],
       [-0.02781898,  0.94381699,  0.34290401,  0.25018353],
       [ 0.76868247,  1.67960008,  1.32765398,  1.72494958],
       [ 0.4641378 , -0.15985763,  0.22705107,  0.11611389],
       [ 0.67497642,  2.17012213,  1.61728632,  1.18867101],
       [-0.23865761, -0.28248815,  0.40083048,  0.38425317],
       [-1.78480749, -0.89564072, -1.39489006, -1.35865217],
       [ 1.07322715,  2.29275265,  1.67521279,  1.05460137],
       [-0.21523109, -0.03722712,  0.1691246 , -0.28609504],
       [-1.12886512, -0.7730102 , -1.39489006, -1.35865217],
       [-0.84774696, -1.01827123, -1.45281653, -1.22458253],
       [-0.7540409 , -0.89564072, -1.10525771, -1.09051288],
       [-0.37921669, -1.01827123, -0.17843421, -0.28609504],
       [ 1.09665366,  0.57592545,  0.63253635,  0.78646209],
       [-1.22257117, -1.26353226, -1.10525771, -1.35865217],
       [-1.19914466, -1.01827123, -1.27903712, -1.35865217],
       [-1.24599769, -0.89564072, -1.22111065, -0.95644324],
       [ 0.41728477, -0.28248815,  0.22705107,  0.11611389],
       [-0.87117347, -0.89564072, -1.33696359, -1.35865217],
       [ 0.44071128, -0.15985763,  0.22705107, -0.01795576],
       [ 0.95609458,  2.29275265,  1.67521279,  1.32274066],
       [-0.68376136, -1.50879329, -1.39489006, -1.35865217],
       [ 0.48756431,  0.45329494,  0.28497754,  0.11611389],
       [ 0.86238853, -0.15985763,  0.69046282,  1.05460137],
       [-1.01173255, -0.40511866, -1.39489006, -1.35865217],
       [ 1.44805136,  0.20803391,  0.57460988,  0.78646209],
       [ 0.58127037, -0.03722712,  0.74838929,  0.92053173],
       [-0.33236366,  0.20803391,  0.11119813, -0.28609504],
       [ 0.18301964, -0.52774918,  0.40083048,  0.38425317],
       [ 1.6823165 ,  0.45329494,  0.9221687 ,  1.4568103 ],
       [-0.54320228, -0.40511866,  0.11119813,  0.11611389],
       [-1.66767493, -0.52774918, -1.22111065, -1.09051288],
       [ 0.39385826, -1.01827123, -0.29428715, -0.28609504],
       [ 0.8155355 ,  0.69855596,  0.86424223,  0.92053173],
       [-0.63690833, -1.01827123, -1.39489006, -1.35865217],
       [-0.96487952, -1.01827123, -1.510743  , -1.35865217],
       [ 0.08931358, -0.40511866, -0.00465481, -0.1520254 ],
       [ 0.01903404,  1.06644751,  0.69046282,  0.65239245],
       [-0.98830604, -1.14090175, -1.33696359, -1.49272181],
       [ 0.88581504, -0.03722712,  0.74838929,  1.59087994],
       [-1.6208219 , -1.01827123, -1.33696359, -1.35865217],
       [-0.77746742, -1.01827123, -1.27903712, -0.8223736 ],
       [-0.14495155,  0.08540339,  0.57460988,  0.78646209],
       [ 0.51099082, -0.89564072, -0.46806656, -0.1520254 ],
       [ 1.02637412,  1.31170853,  1.0959481 ,  1.4568103 ],
       [ 0.15959312,  0.20803391,  0.74838929,  0.51832281],
       [ 1.35434531,  0.33066442,  1.03802163,  0.25018353],
       [ 1.37777182,  2.29275265,  1.32765398,  1.4568103 ],
       [ 0.3001522 , -0.40511866,  0.11119813,  0.11611389],
       [-1.59739539, -1.75405432, -1.39489006, -1.35865217],
       [-1.48026282, -1.87668483, -1.56866947, -1.49272181],
```

```
       [ 1.00294761,  0.20803391,  0.69046282,  0.38425317],
       [ 1.14350669,  1.67960008,  1.26972751,  0.78646209],
       [-1.71452795, -1.50879329, -1.33696359, -1.35865217],
       [-1.38655677, -0.89564072, -1.39489006, -1.22458253],
       [-0.89459998, -1.75405432, -1.45281653, -1.35865217],
       [-0.09809853,  0.57592545,  0.63253635,  0.38425317],
       [ 1.40119834,  0.57592545,  1.03802163,  1.59087994],
       [-1.64424841, -1.50879329, -1.39489006, -1.22458253],
       [ 0.83896201,  1.18907802,  0.98009517,  1.18867101],
       [ 0.55784385,  0.57592545,  1.26972751,  1.72494958],
       [-1.73795447, -1.38616278, -1.45281653, -1.35865217],
       [-0.30893715,  0.33066442,  0.51668341,  0.25018353],
       [-0.51977577,  0.82118648,  0.45875695,  0.38425317],
       [ 1.1669332 ,  0.45329494,  0.57460988,  0.78646209],
       [-0.61348182,  1.43433905,  0.51668341,  0.25018353],
       [ 0.90924155,  0.69855596,  0.86424223,  1.4568103 ],
       [-0.70718788, -0.89564072, -1.27903712, -1.35865217],
       [ 1.47147788,  1.31170853,  0.9221687 ,  1.18867101],
       [-0.35579017,  0.08540339,  0.22705107,  0.38425317],
       [ 1.65888998,  0.82118648,  0.80631576,  1.05460137],
       [ 0.06588707, -0.15985763, -0.17843421, -0.28609504],
       [-0.4026432 , -0.7730102 ,  0.05327166,  0.25018353],
       [ 0.34700523,  0.33066442,  0.45875695,  0.25018353],
       [-0.82432044, -1.6314238 , -1.45281653, -1.22458253],
       [-0.42606971,  0.94381699,  0.45875695,  0.11611389],
       [ 0.32357872, -0.40511866,  0.34290401, -0.01795576],
       [-0.66033485, -0.65037969, -1.33696359, -1.35865217],
       [ 0.27672569, -0.28248815,  0.1691246 ,  0.11611389],
       [ 0.72182945,  1.80223059,  1.44350692,  0.78646209],
       [ 1.12008018,  1.06644751,  1.0959481 ,  1.18867101],
       [-1.29285071, -0.89564072, -1.33696359, -1.09051288],
       [-0.44949623, -1.14090175, -0.29428715, -0.28609504],
       [ 1.58861044,  1.06644751,  1.0959481 ,  1.72494958],
       [ 1.23721274,  1.67960008,  1.15387457,  0.51832281],
       [-0.9180265 , -1.14090175, -1.33696359, -1.49272181],
       [ 1.49490439,  1.06644751,  1.03802163,  1.59087994],
       [-1.76138098, -1.14090175, -1.39489006, -1.35865217],
       [-0.56662879,  1.31170853,  0.63253635,  0.38425317],
       [ 1.26063926,  1.9248611 ,  1.32765398,  0.92053173],
       [ 0.62812339,  0.57592545,  1.03802163,  0.78646209],
       [ 0.53441734, -0.15985763,  0.1691246 ,  0.11611389],
       [ 0.93266807,  0.82118648,  0.98009517,  0.78646209],
       [ 0.25329918,  0.57592545,  0.34290401,  0.11611389],
       [-0.0512455 ,  0.69855596,  0.28497754,  0.11611389],
       [ 1.04980063, -0.28248815,  0.63253635,  1.05460137],
       [ 1.70574301,  0.08540339,  0.74838929,  0.78646209],
       [-1.31627723, -0.52774918, -1.22111065, -1.35865217],
       [-0.12152504,  0.33066442,  0.11119813,  0.11611389],
       [ 0.69840293, -1.14090175,  0.40083048,  0.65239245],
       [-1.45683631, -0.03722712, -1.510743  , -1.35865217],
       [ 0.37043174, -0.03722712,  0.11119813, -0.01795576],
       [ 0.60469688,  1.55696956,  1.21180104,  1.18867101]])
```

In [30]:
```
type(X_train)
X_test=scaler.fit_transform(X_test)
X_test
```

Out[30]:
```
array([[ 0.07772639,  0.25621067,  0.58987181,  0.05585913],
       [-1.15294139, -0.21299441, -1.03834577, -1.06789518],
       [ 1.08463638,  2.13303102,  1.78389803,  1.42933663],
       [ 0.18960527,  0.1389094 ,  0.48132397,  0.43044391],
       [ 0.14485372,  1.07731957,  0.64414573,  0.30558231],
       [-0.86205628, -0.56489823, -1.14689361, -0.94303359],
       [-0.12365561, -0.33029569, -0.00714131,  0.18072072],
       [ 1.59927927,  1.19462085,  0.80696748,  1.42933663],
       [-0.0341525 ,  0.37351194,  0.48132397,  0.43044391],
       [ 0.27910838, -0.09569314,  0.15568045,  0.05585913],
       [ 0.90563016,  0.72541576,  0.80696748,  1.05475186],
       [-1.28719605, -1.26870586, -1.20116753, -1.31761837],
       [-0.75017739, -0.44759696, -1.25544145, -1.19275678],
       [-1.35432339, -1.15140459, -1.14689361, -1.31761837],
       [-1.13056561, -0.91680205, -1.14689361, -1.06789518],
       [-0.30266183,  0.49081322,  0.58987181,  0.5553055 ],
       [ 0.77137549,  0.72541576,  1.18688492,  1.30447504],
       [-0.01177673, -0.33029569,  0.15568045, -0.06900246],
       [-0.32503761, -0.21299441,  0.48132397,  0.18072072],
       [ 1.39789727,  0.60811449,  1.07833708,  1.30447504],
       [-0.90680783, -1.38600713, -1.09261969, -1.19275678],
       [ 1.28601838,  0.25621067,  0.69841965,  0.80502868],
       [-0.97393516, -1.03410332, -1.09261969, -0.94303359],
       [ 1.30839416,  0.60811449,  1.07833708,  1.17961345],
       [ 1.37552149,  2.36763356,  1.51252844,  1.05475186],
       [ 1.68878238,  0.9600183 ,  0.8612414 ,  1.42933663],
       [ 0.8608786 ,  0.9600183 ,  1.18688492,  0.80502868],
       [ 1.64403082,  1.07731957,  1.24115884,  1.42933663],
       [-0.54879539, -1.26870586, -1.20116753, -1.06789518],
       [-0.88443205, -1.26870586, -1.09261969, -1.19275678],
       [-1.06343828, -1.5033084 , -1.41826321, -1.19275678],
       [-1.22006872, -0.21299441, -1.14689361, -0.94303359],
       [-0.10127984,  0.9600183 ,  0.42705005,  0.30558231],
       [-1.30957183, -1.26870586, -1.09261969, -1.19275678],
       [-0.61592272, -1.73791095, -1.25544145, -1.19275678],
       [ 1.71115816,  0.49081322,  0.75269356,  0.92989027],
       [-0.41454072,  0.60811449,  0.48132397,  0.43044391],
       [-0.95155939, -0.79950077, -1.14689361, -1.19275678]])
```

In [31]:
```
from warnings import filterwarnings
filterwarnings('ignore')
print("The Accuracy : ",RF.score(X_test,y_test) * 100)
```

```
The Accuracy :  100.0
```

In [32]:
```
pre2 = RF.predict(X_test)
from warnings import filterwarnings
filterwarnings('ignore')
```

In [33]:
```
for i in range(len(pre2)):
    print("The given Data is:  ",X_test[i],"The predicted Output is:   ","-->>",
```

```
The given Data is:   [0.07772639 0.25621067 0.58987181 0.05585913] The predicted
Output is:    -->> Iris-versicolor
The given Data is:   [-1.15294139 -0.21299441 -1.03834577 -1.06789518] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [1.08463638 2.13303102 1.78389803 1.42933663] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [0.18960527 0.1389094  0.48132397 0.43044391] The predicted
Output is:    -->> Iris-versicolor
The given Data is:   [0.14485372 1.07731957 0.64414573 0.30558231] The predicted
Output is:    -->> Iris-versicolor
The given Data is:   [-0.86205628 -0.56489823 -1.14689361 -0.94303359] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.12365561 -0.33029569 -0.00714131  0.18072072] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [1.59927927 1.19462085 0.80696748 1.42933663] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.0341525   0.37351194  0.48132397  0.43044391] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [ 0.27910838 -0.09569314  0.15568045  0.05585913] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [0.90563016 0.72541576 0.80696748 1.05475186] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-1.28719605 -1.26870586 -1.20116753 -1.31761837] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.75017739 -0.44759696 -1.25544145 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-1.35432339 -1.15140459 -1.14689361 -1.31761837] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-1.13056561 -0.91680205 -1.14689361 -1.06789518] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.30266183  0.49081322  0.58987181  0.5553055 ] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [0.77137549 0.72541576 1.18688492 1.30447504] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.01177673 -0.33029569  0.15568045 -0.06900246] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [-0.32503761 -0.21299441  0.48132397  0.18072072] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [1.39789727 0.60811449 1.07833708 1.30447504] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.90680783 -1.38600713 -1.09261969 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [1.28601838 0.25621067 0.69841965 0.80502868] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.97393516 -1.03410332 -1.09261969 -0.94303359] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [1.30839416 0.60811449 1.07833708 1.17961345] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [1.37552149 2.36763356 1.51252844 1.05475186] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [1.68878238 0.9600183  0.8612414  1.42933663] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [0.8608786  0.9600183  1.18688492 0.80502868] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [1.64403082 1.07731957 1.24115884 1.42933663] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.54879539 -1.26870586 -1.20116753 -1.06789518] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.88443205 -1.26870586 -1.09261969 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
```

```
The given Data is:   [-1.06343828 -1.5033084  -1.41826321 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-1.22006872 -0.21299441 -1.14689361 -0.94303359] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.10127984  0.9600183   0.42705005  0.30558231] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [-1.30957183 -1.26870586 -1.09261969 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [-0.61592272 -1.73791095 -1.25544145 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
The given Data is:   [1.71115816 0.49081322 0.75269356 0.92989027] The predicted
Output is:    -->> Iris-virginica
The given Data is:   [-0.41454072  0.60811449  0.48132397  0.43044391] The predic
ted Output is:    -->> Iris-versicolor
The given Data is:   [-0.95155939 -0.79950077 -1.14689361 -1.19275678] The predic
ted Output is:    -->> Iris-setosa
```

In [34]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,pre2))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        15
Iris-versicolor       1.00      1.00      1.00        11
 Iris-virginica       1.00      1.00      1.00        12

       accuracy                           1.00        38
      macro avg       1.00      1.00      1.00        38
   weighted avg       1.00      1.00      1.00        38
```

In [ ]:
```

```