In [2]:
```
# Name----Ankit
# Email Id----mrankit1950@gmail.com
```

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import seaborn as sns
```

In [18]:
```
iris_datasets=pd.read_csv(r"C:\Users\ANKIT MALL-PC\Desktop\internship ques
iris_datasets
```

Out[18]:

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [19]:
```
iris_datasets=iris_datasets.drop(columns=['Id'])
iris_datasets
```

Out[19]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [20]:
```
iris_datasets.describe()
```

Out[20]:

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

In [21]:
```
1 iris_datasets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

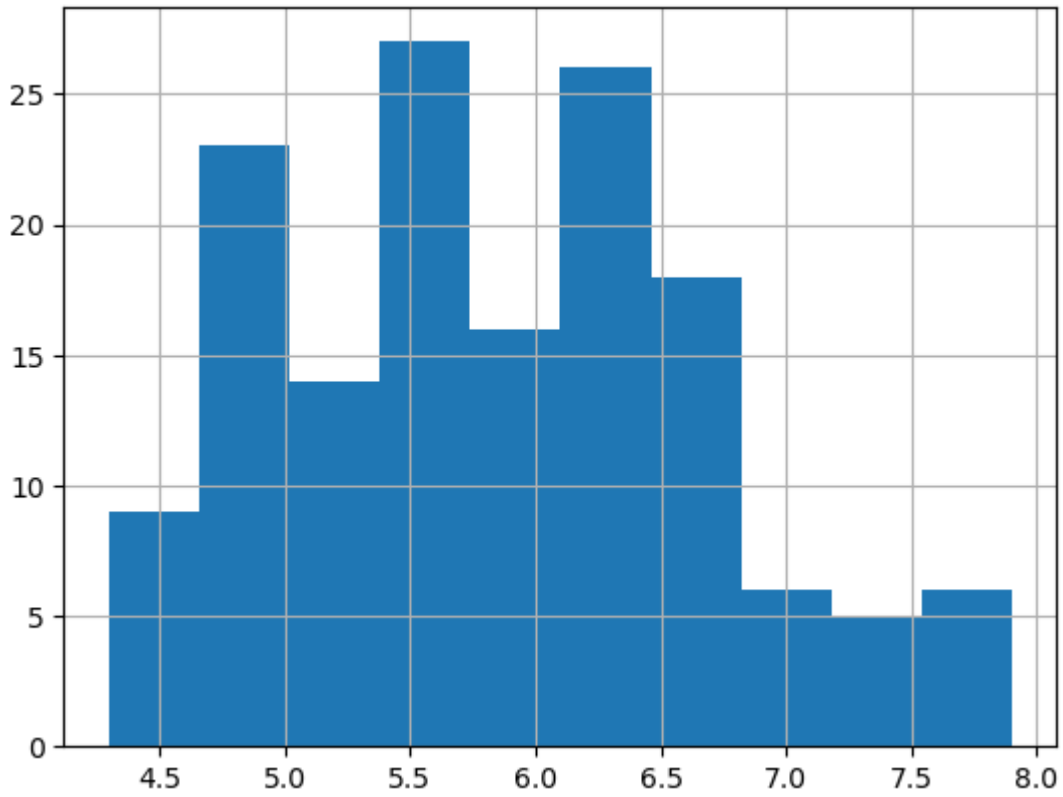
In [22]:
```
1 iris_datasets['Species'].value_counts()
```

Out[22]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

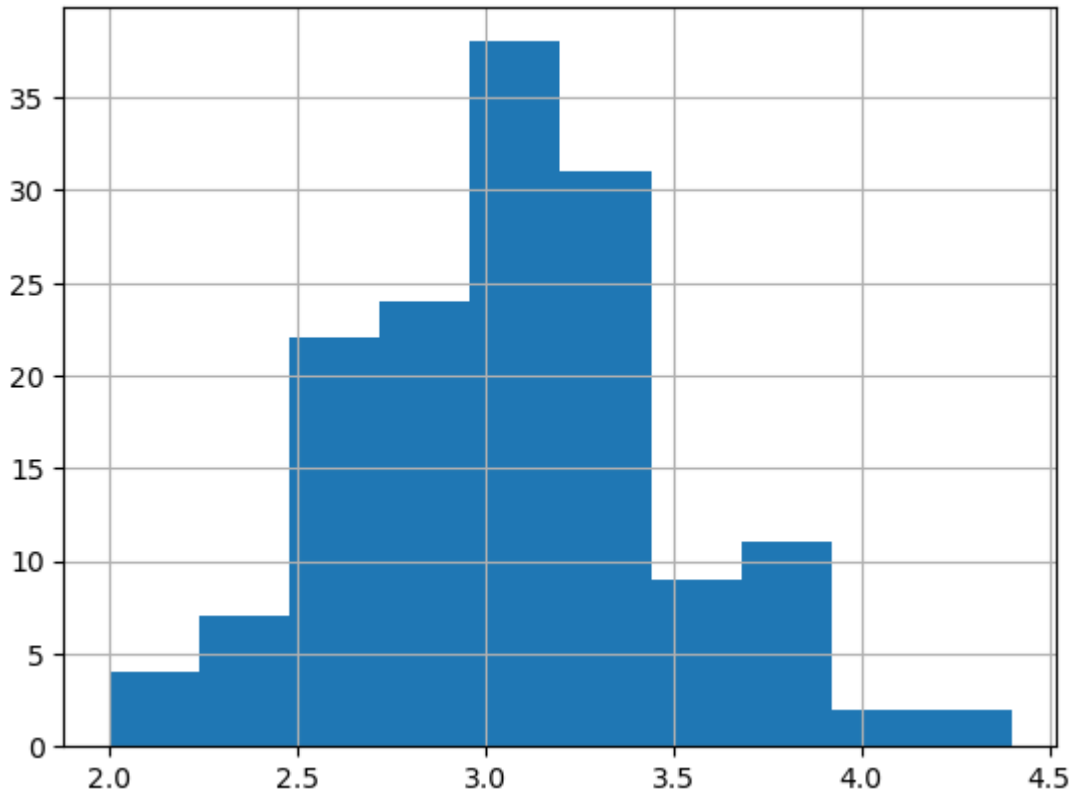
In [23]:
```
1 iris_datasets.isnull().sum()
```

Out[23]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

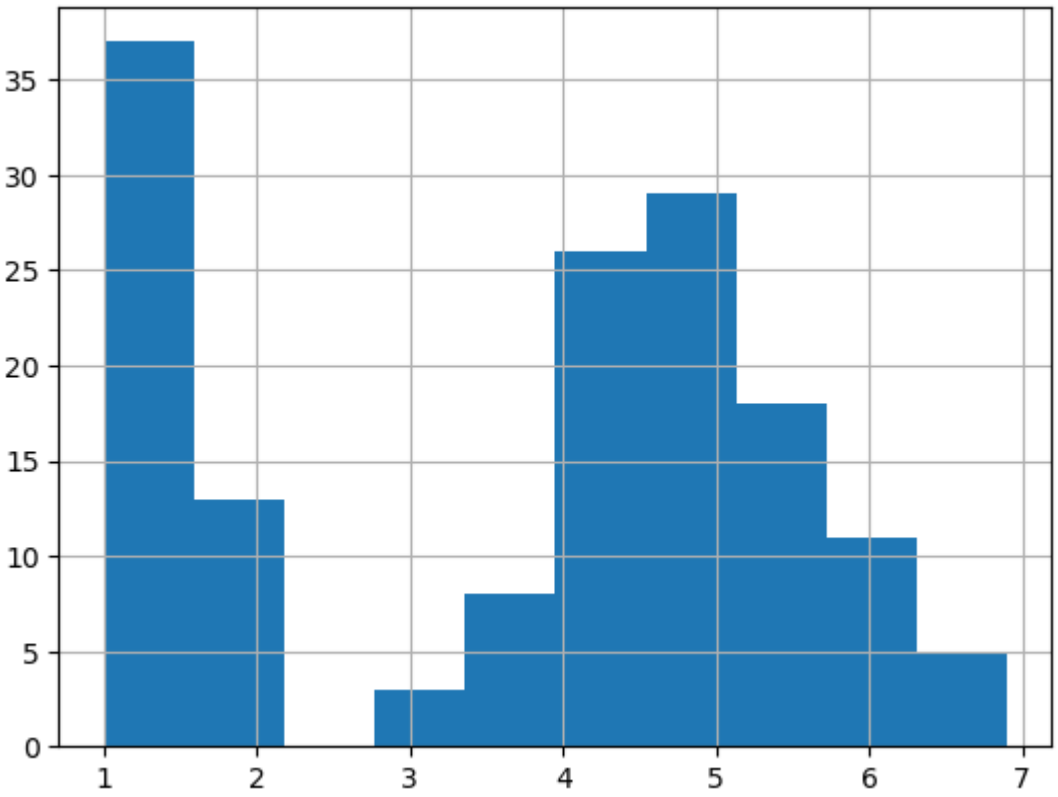
In [24]:
```
1 iris_datasets['SepalLengthCm'].hist();
```



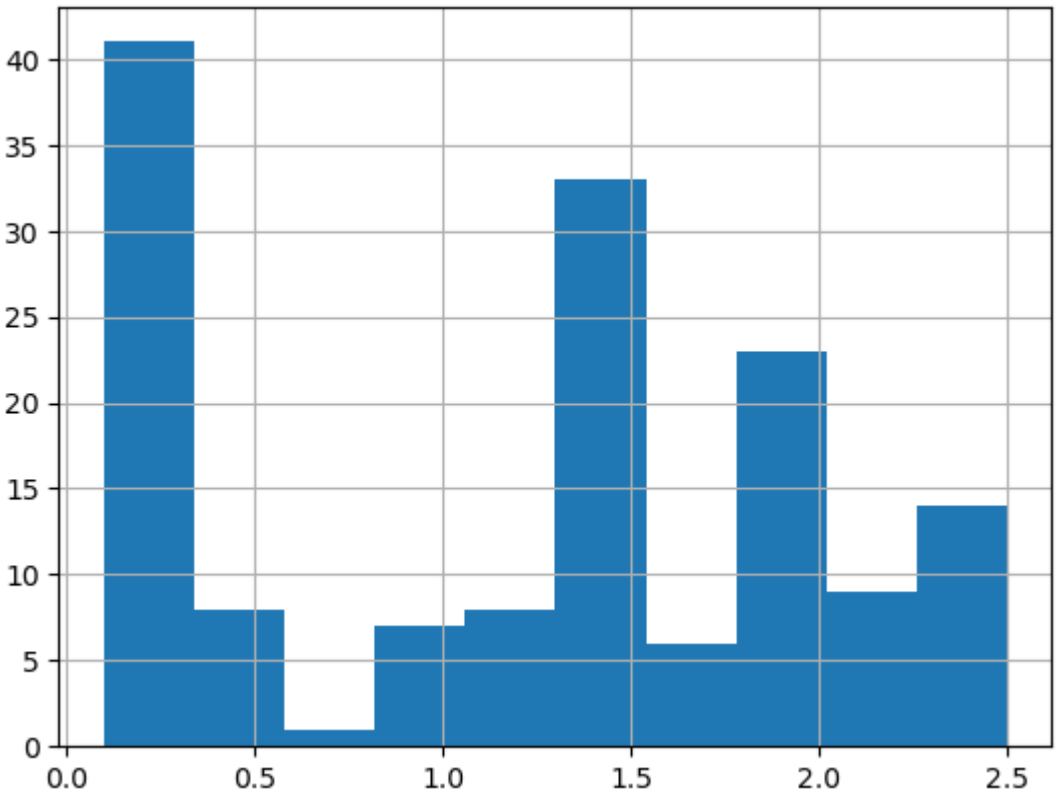
In [25]:
```
1 iris_datasets['SepalWidthCm'].hist();
```



In [26]:
```
1 iris_datasets['PetalLengthCm'].hist();
```



In [27]:
```
1 iris_datasets['PetalWidthCm'].hist();
```

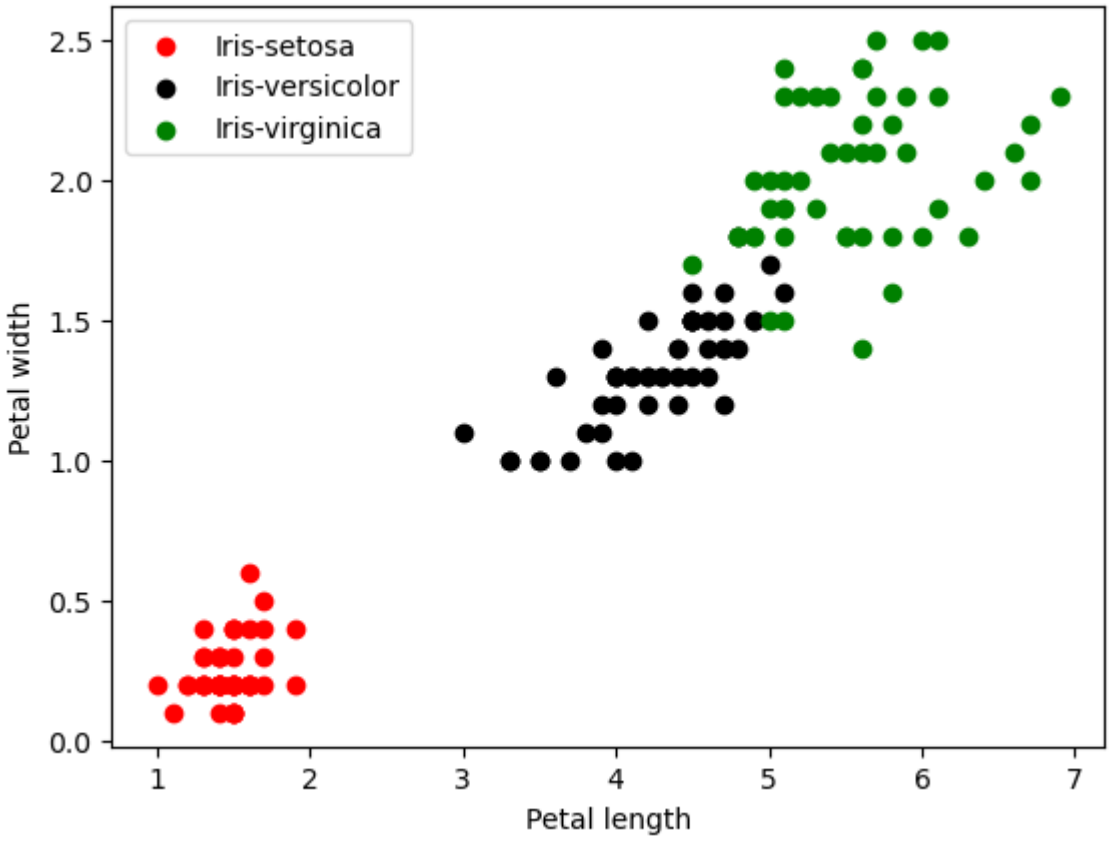


In [28]:
```
colors=['red','black','green']
Species=['Iris-setosa','Iris-versicolor' ,'Iris-virginica']
```

In [ ]:
```

```

In [29]:
```
for i in range(3):
    x=iris_datasets[iris_datasets['Species']==Species[i]]
    plt.scatter(x['PetalLengthCm'],x['PetalWidthCm'], c=colors[i], label=
plt.xlabel('Petal length')
plt.ylabel('Petal width')
plt.legend();
```



In [ ]:
```

```

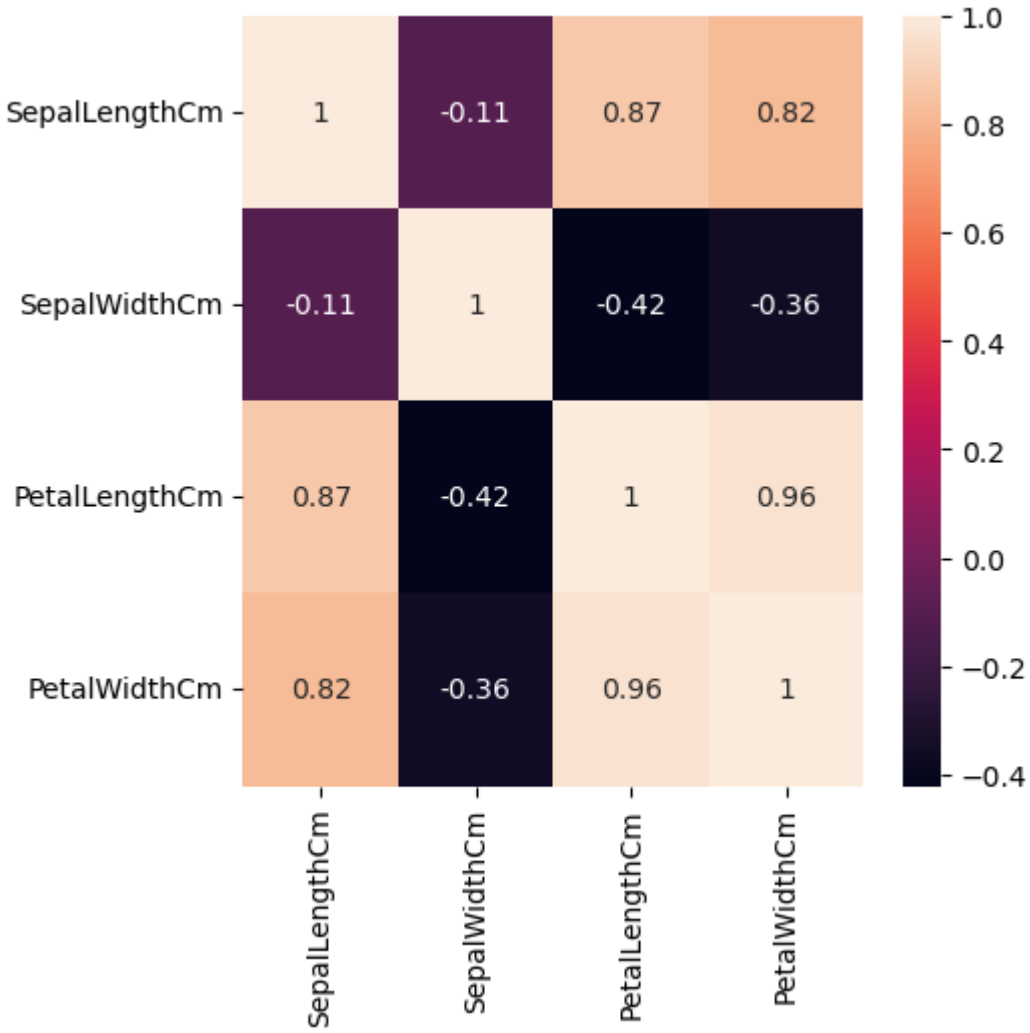
In [30]:
```
iris_datasets.corr()
```

Out[30]:

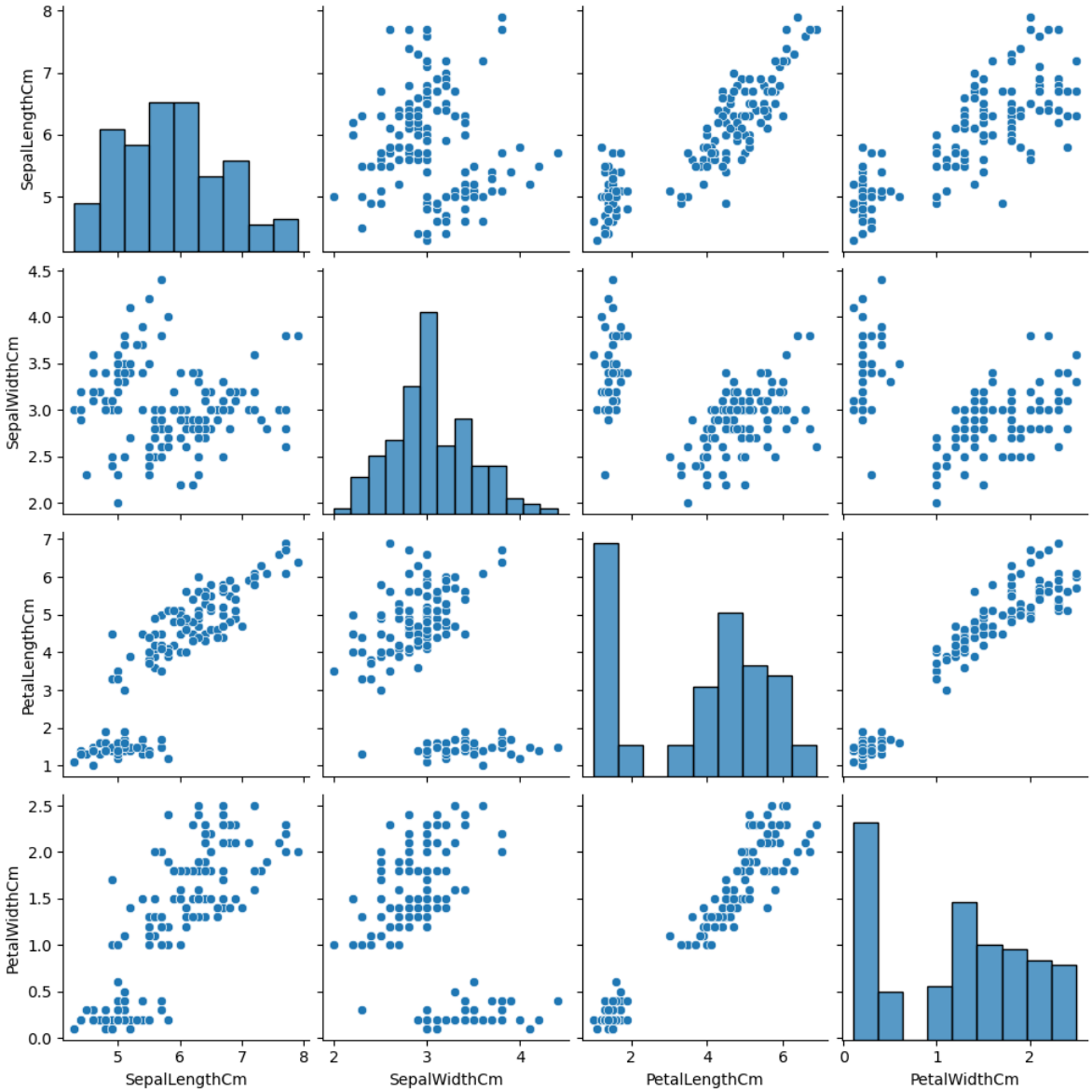
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

In [31]:
```
1 corr=iris_datasets.corr()
2 fig, ax=plt.subplots(figsize=(5,5))
3 sns.heatmap(corr,annot=True)
```

Out[31]: <AxesSubplot:>



In [32]:
```
1 sns.pairplot(iris_datasets);
```



In [33]:
```
1 from sklearn.preprocessing import LabelEncoder
2 le=LabelEncoder()
```

In [34]:
```
1 iris_datasets['Species']=le.fit_transform(iris_datasets['Species'])
```

In [35]:
```
1 iris_datasets.head()
```

Out[35]:

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

In [36]:
```
1 iris_datasets.tail()
```

Out[36]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

In [37]:
```
1 A=iris_datasets.drop(columns=['Species'])
2 B=iris_datasets['Species']
```

In [46]:
```
1 from sklearn.model_selection import train_test_split
2 x_train,x_test,y_train,y_test=train_test_split(A,B,train_size=0.85)
```

In [47]:
```
1 from sklearn.linear_model import LogisticRegression
2 model=LogisticRegression()
3 model
```

Out[47]:
```
LogisticRegression()
```

In [48]:
```
1 model.fit(x_train,y_train)
```

Out[48]:
```
LogisticRegression()
```

In [49]:
```
1 print('accuracy:',model.score(x_test,y_test)*100)
```

```
accuracy: 95.65217391304348
```

In [50]:
```
1 model.predict([[6.0,3.6,4.4,0.2]])
```

```
C:\Users\ANKIT MALL-PC\anaconda3\lib\site-packages\sklearn\base.py:450: UserW
arning: X does not have valid feature names, but LogisticRegression was fitte
d with feature names
  warnings.warn(
```

Out[50]:
```
array([1])
```

In [ ]:
```
1
```

In [ ]:
```
1
```