In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\14_Iris - 14_Iris.csv")
data
```

	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 [3]:
```
df=data.head(100)
df
```

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
...	...	...	...	...	...	...
95	96	5.7	3.0	4.2	1.2	Iris-versicolor
96	97	5.7	2.9	4.2	1.3	Iris-versicolor
97	98	6.2	2.9	4.3	1.3	Iris-versicolor
98	99	5.1	2.5	3.0	1.1	Iris-versicolor
99	100	5.7	2.8	4.1	1.3	Iris-versicolor

100 rows × 6 columns

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             100 non-null    int64  
 1   SepalLengthCm  100 non-null    float64
 2   SepalWidthCm   100 non-null    float64
 3   PetalLengthCm  100 non-null    float64
 4   PetalWidthCm   100 non-null    float64
 5   Species        100 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 4.8+ KB
```

In [5]:

```
df.describe()
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	100.000000	100.000000	100.000000	100.000000	100.000000
mean	50.500000	5.471000	3.094000	2.862000	0.785000
std	29.011492	0.641698	0.476057	1.448565	0.566288
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	25.750000	5.000000	2.800000	1.500000	0.200000
50%	50.500000	5.400000	3.050000	2.450000	0.800000
75%	75.250000	5.900000	3.400000	4.325000	1.300000
max	100.000000	7.000000	4.400000	5.100000	1.800000

In [6]:

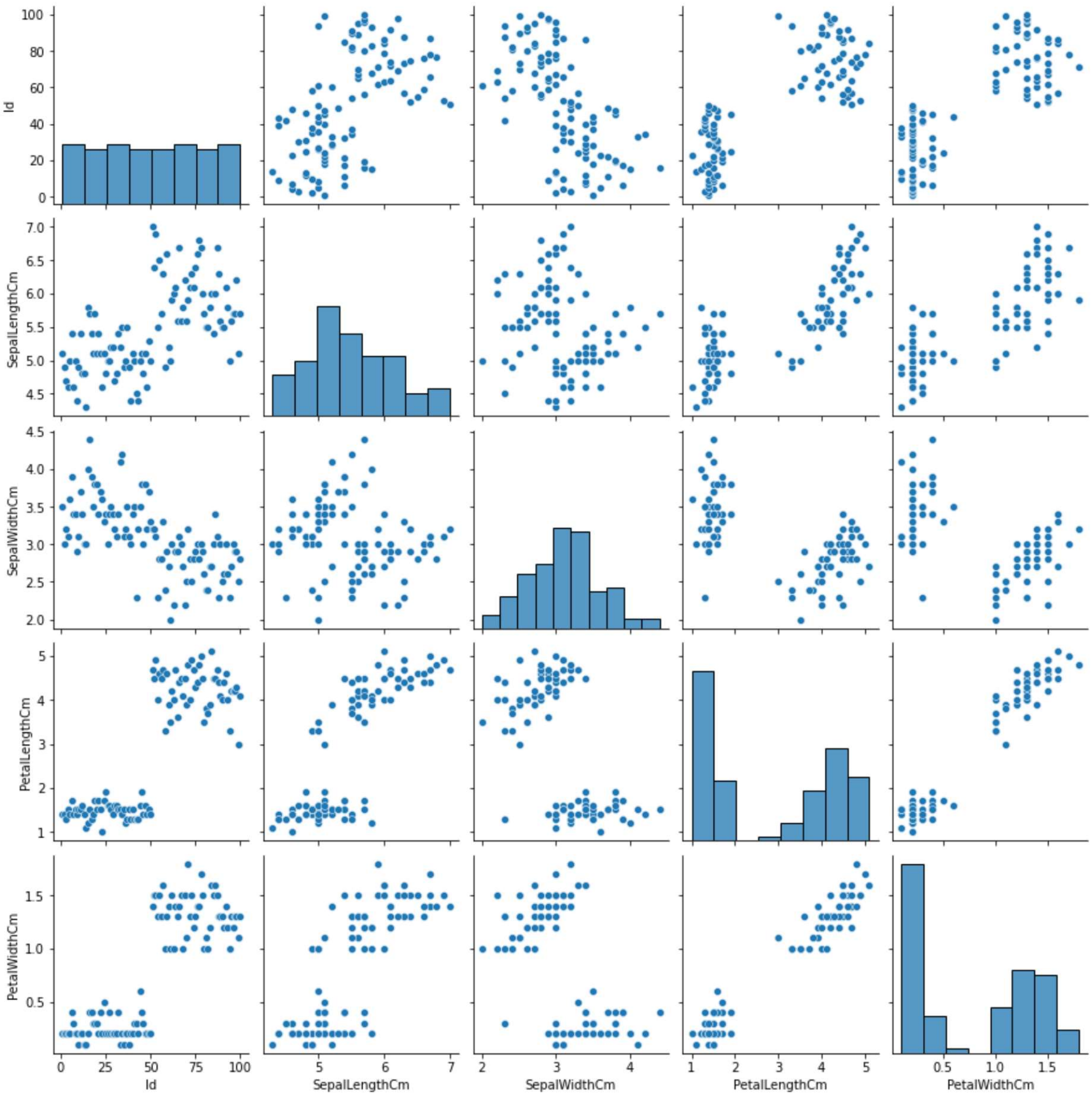
```
df.columns
```

Out[6]:

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

```
In [7]: sns.pairplot(df)
```

```
Out[7]: <seaborn.axisgrid.PairGrid at 0x1a207f28bb0>
```



In [8]:
```
da=df[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species']]
da
```

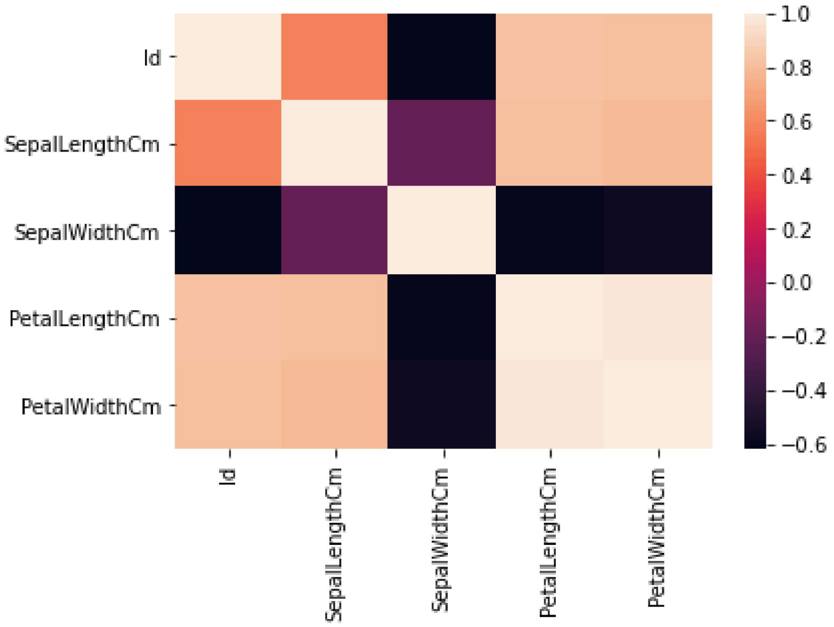
Out[8]:

	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
...	...	...	...	...	...	...
95	96	5.7	3.0	4.2	1.2	Iris-versicolor
96	97	5.7	2.9	4.2	1.3	Iris-versicolor
97	98	6.2	2.9	4.3	1.3	Iris-versicolor
98	99	5.1	2.5	3.0	1.1	Iris-versicolor
99	100	5.7	2.8	4.1	1.3	Iris-versicolor

100 rows × 6 columns

In [9]:
```
sns.heatmap(da.corr())
```

Out[9]:
```
<AxesSubplot:>
```



In [20]:
```
x=da[['Id','SepalWidthCm', 'PetalLengthCm', 'SepalLengthCm']]
y=da[ 'PetalWidthCm']
```

In [21]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [22]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[22]:

```
LinearRegression()
```

In [23]:

```
print(lr.intercept_)
```

```
-0.5740330435115729
```

In [24]:

```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[24]:

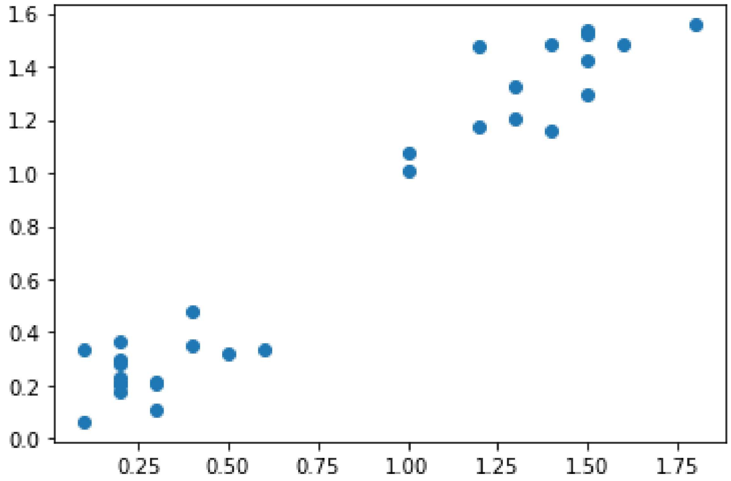
	Co-efficient
Id	0.001411
SepalWidthCm	0.101129
PetalLengthCm	0.387199
SepalLengthCm	-0.025663

In [25]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[25]:

```
<matplotlib.collections.PathCollection at 0x1a20b23d5b0>
```



In [30]:

```
print(lr.score(x_test,y_test))
```

```
0.9444278480231519
```

```
In [31]: print(lr.score(x_train,y_train))

         0.9669842680744002
```

```
In [32]: from sklearn.linear_model import Ridge,Lasso
```

```
In [33]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)

Out[33]: Ridge(alpha=10)
```

```
In [34]: rr.score(x_test,y_test)

Out[34]: 0.9267122206100494
```

```
In [35]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[35]: Lasso(alpha=10)
```

```
In [36]: la.score(x_test,y_test)

Out[36]: 0.281076798925889
```

```
In [ ]:
```