In [ ]:

```
```

In [146]:

```
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [147]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\14_Iris.csv")
a
```

Out[147]:

	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 [148]:

```
a=a.head(10)
a
```

Out[148]:

	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 [149]:

```
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             10 non-null     int64  
 1   SepalLengthCm  10 non-null     float64
 2   SepalWidthCm   10 non-null     float64
 3   PetalLengthCm  10 non-null     float64
 4   PetalWidthCm   10 non-null     float64
 5   Species        10 non-null     object 
dtypes: float64(4), int64(1), object(1)
memory usage: 608.0+ bytes
```

In [150]:

```
# to display summary of statastic
a.describe()
```

Out[150]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	10.00000	10.000000	10.000000	10.000000	10.000000
mean	5.50000	4.860000	3.310000	1.450000	0.220000
std	3.02765	0.291357	0.307137	0.108012	0.078881
min	1.00000	4.400000	2.900000	1.300000	0.100000
25%	3.25000	4.625000	3.100000	1.400000	0.200000
50%	5.50000	4.900000	3.300000	1.400000	0.200000
75%	7.75000	5.000000	3.475000	1.500000	0.200000
max	10.00000	5.400000	3.900000	1.700000	0.400000

In [151]:

```
# to display colum heading
a.columns
```

Out[151]:

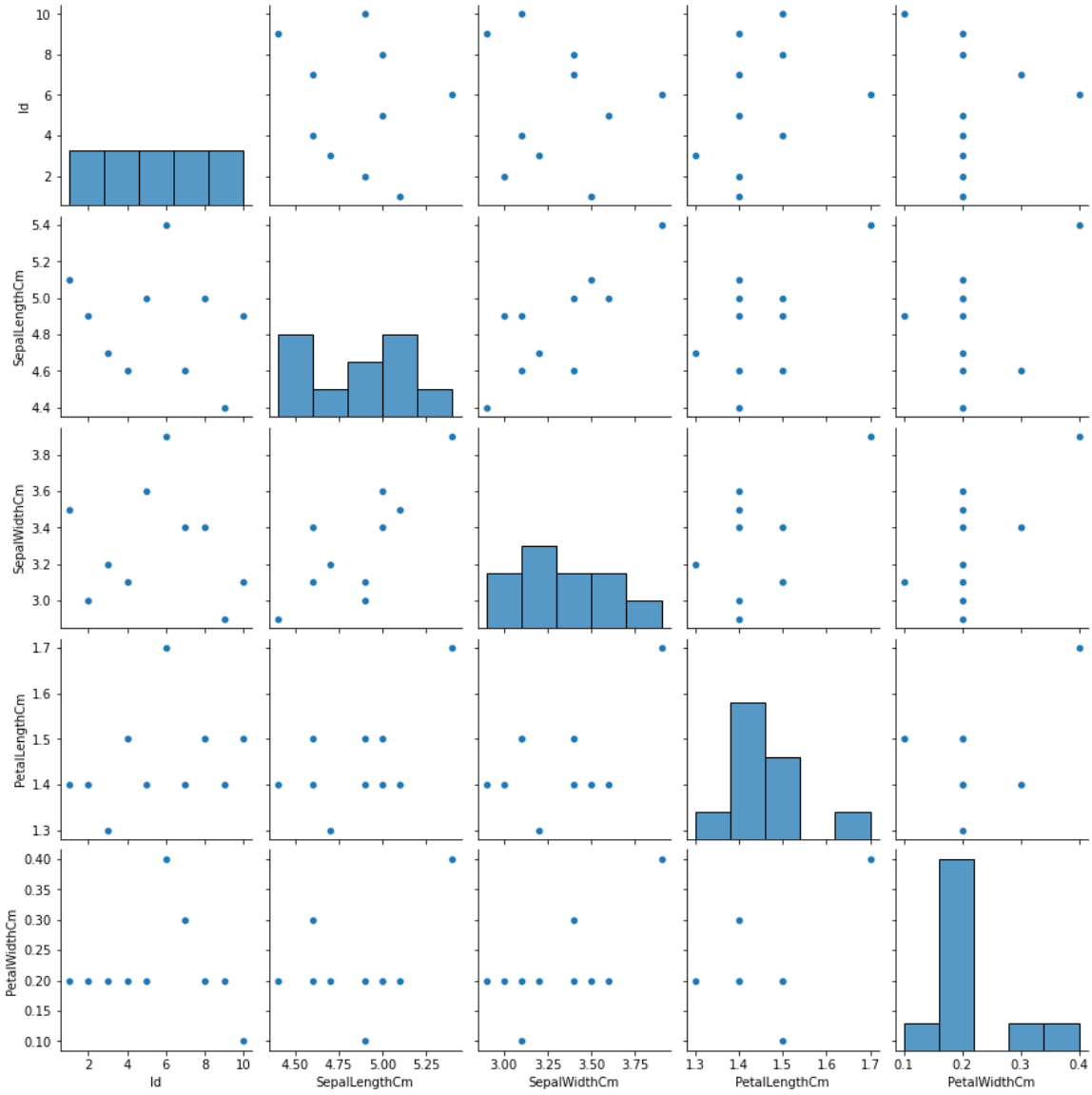
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidth
Cm',
       'Species'],
      dtype='object')
```

In [152]:

```
sns.pairplot(a)
```

Out[152]:

```
<seaborn.axisgrid.PairGrid at 0x203e154e430>
```

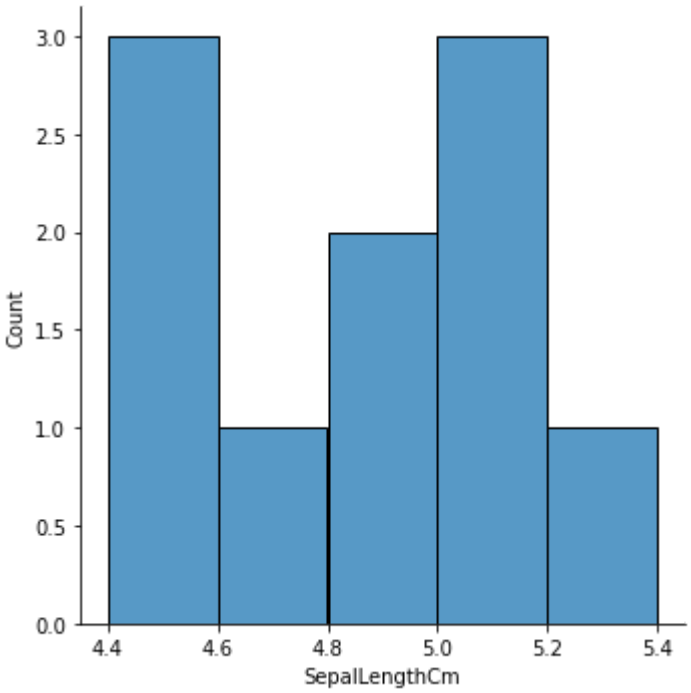


In [153]:

```
sns.displot(a["SepalLengthCm"])
```

Out[153]:

```
<seaborn.axisgrid.FacetGrid at 0x203e20de6a0>
```



In [154]:

```
b=a[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species']]
b
```

Out[154]:

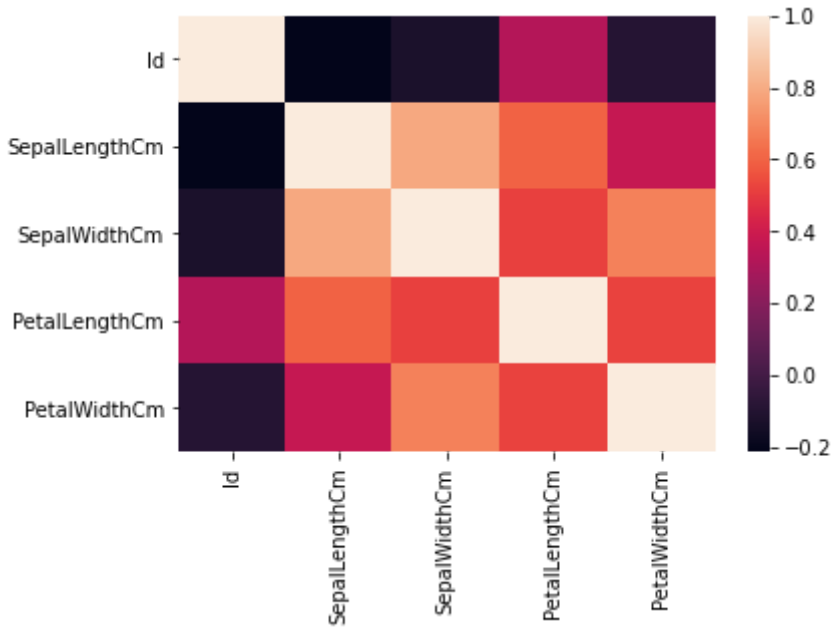
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	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 [155]:

```
sns.heatmap(b.corr())
```

Out[155]:

<AxesSubplot:>



In [160]:

```
x=a[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
y=a['SepalLengthCm']
```

In [161]:

```
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 [162]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[162]:

```
LinearRegression()
```

In [163]:

```
lr.intercept_
```

Out[163]:

```
4.440892098500626e-15
```

In [164]:

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

Out[164]:

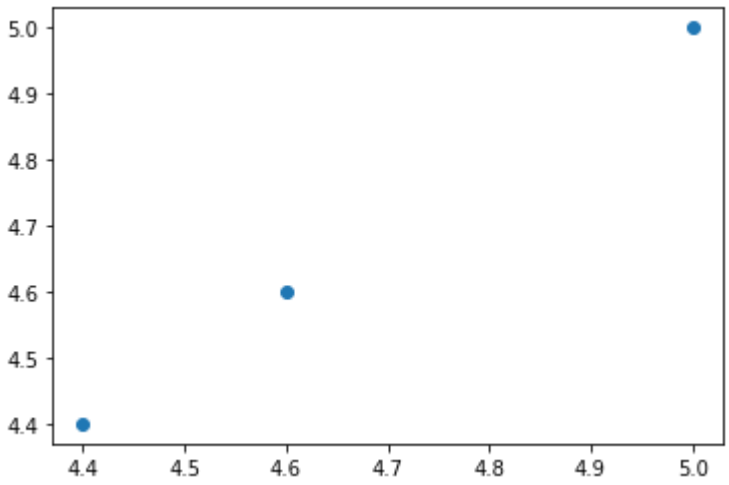
	Co-efficient
Id	2.240189e-17
SepalLengthCm	1.000000e+00
SepalWidthCm	-1.688848e-16
PetalLengthCm	-5.196410e-16
PetalWidthCm	3.155290e-16

In [165]:

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

Out[165]:

```
<matplotlib.collections.PathCollection at 0x203e256a850>
```



In [166]:

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

Out[166]:

```
1.0
```

In [167]:

```
lr.score(x_train,y_train)
```

Out[167]:

```
1.0
```

In [168]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [169]:

```
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[169]:

```
Ridge(alpha=10)
```

In [170]:

```
rr.score(x_test,y_test)
```

Out[170]:

```
0.679631923027884
```

In [171]:

```
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

Out[171]:

```
Lasso(alpha=10)
```

In [172]:

```
la.score(x_test,y_test)
```

Out[172]:

```
0.0
```

In [ ]:

```

```