

```
In [76]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [337]: x=pd.read_csv(r"C:\Users\user\Downloads\14_Iris - 14_Iris.csv")
```

Out[337]:

	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 [338]: x=x.head(10)
```

Out[338]:

	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 [339]:

```
<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 [340]:

```
Out[340]: Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
                'Species'],
              dtype='object')
```

In [341]: `d=x[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]`

Out[341]:

	<b>Id</b>	<b>SepalLengthCm</b>	<b>SepalWidthCm</b>	<b>PetalLengthCm</b>	<b>PetalWidthCm</b>
<b>0</b>	1	5.1	3.5	1.4	0.2
<b>1</b>	2	4.9	3.0	1.4	0.2
<b>2</b>	3	4.7	3.2	1.3	0.2
<b>3</b>	4	4.6	3.1	1.5	0.2
<b>4</b>	5	5.0	3.6	1.4	0.2
<b>5</b>	6	5.4	3.9	1.7	0.4
<b>6</b>	7	4.6	3.4	1.4	0.3
<b>7</b>	8	5.0	3.4	1.5	0.2
<b>8</b>	9	4.4	2.9	1.4	0.2
<b>9</b>	10	4.9	3.1	1.5	0.1

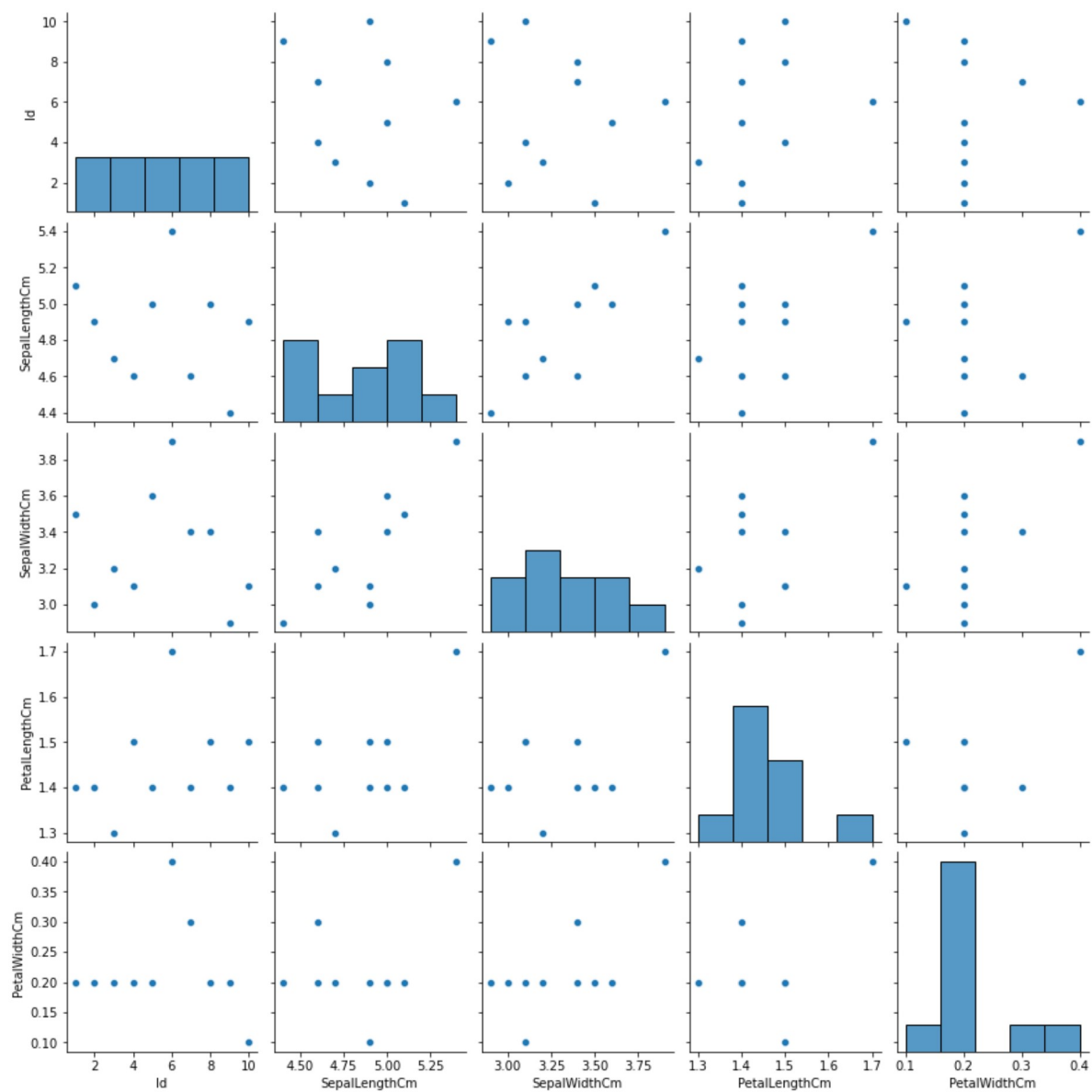
In [342]:

Out[342]:

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

In [343]:

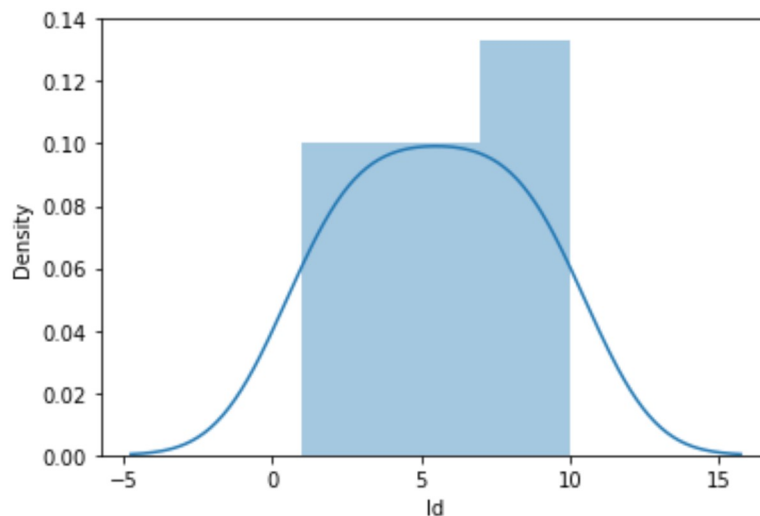
Out[343]: <seaborn.axisgrid.PairGrid at 0x190cfc3cd90>



In [344]:

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[344]: &lt;AxesSubplot:xlabel='Id', ylabel='Density'&gt;



In [346]:

In [347]:

Out[347]: &lt;AxesSubplot:&gt;

In [349]: `x=x1[['Id']]`

In [350]: *# to split my dataset into training and test data*

```
from sklearn.model_selection import train_test_split
```

In [351]: **from** sklearn.linear\_model **import** LinearRegression

```
lr=LinearRegression()
```

Out[351]: LinearRegression()

In [352]:

```
8.881784197001252e-16
```

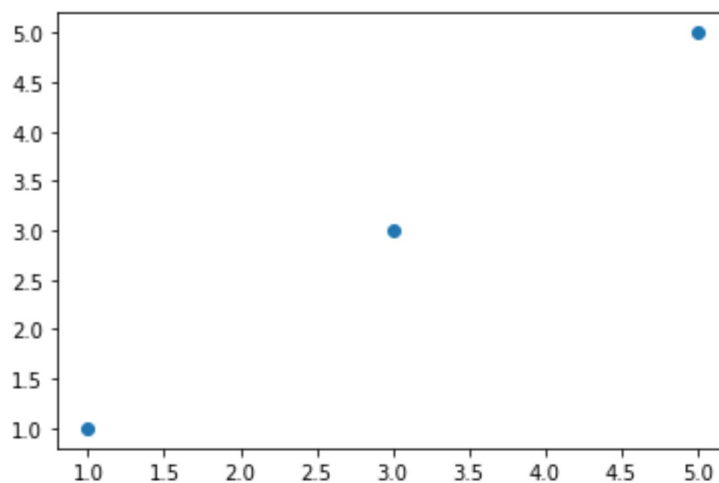
In [353]: `coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])`

Out[353]:

Co-efficient	
Id	1.0

In [354]: `prediction=lr.predict(x_test)`

Out[354]: <matplotlib.collections.PathCollection at 0x190d12da940>



In [355]:

Out[355]: 1.0

In [356]:

Out[356]: 1.0

In [357]:

```
In [358]: rr=Ridge(alpha=10)
          rr.fit(x_train,y_train)
```

```
Out[358]: 0.8263803793745714
```

```
In [359]: la=Lasso(alpha=10)
```

```
Out[359]: Lasso(alpha=10)
```

```
In [360]:
```

```
Out[360]: -4.783163265306122
```

```
In [361]: from sklearn.linear_model import ElasticNet
          en=ElasticNet()
```

```
Out[361]: ElasticNet()
```

```
In [362]:
```

```
Out[362]: array([0.86331939])
```

```
In [363]:
```

```
Out[363]: array([3.48814505, 1.76150628, 5.21478382])
```

```
In [364]:
```

```
Out[364]: 0.8981868898186889
```

```
In [365]:
```

```
Out[365]: 0.8919613140915289
```

```
In [366]:
```

```
In [367]:
```

```
Mean Absolute Error 4.440892098500626e-16
```

```
In [368]:
```

```
Mean Squared Error 3.2869204384208827e-31
```

```
In [369]:
```

```
Root Mean Squared Error 5.733167046599011e-16
```

```
In [ ]:
```

