

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
	2	False	False	False	False	False
	3	False	False	False	False	False
	4	False	False	False	False	False

	145	False	False	False	False	False
	146	False	False	False	False	False
	147	False	False	False	False	False
	148	False	False	False	False	False
	149	False	False	False	False	False

150 rows × 6 columns

```
In [5]: d.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]: d.columns
```

```
Out[6]: Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
              'Species'],
              dtype='object')
```

```
In [7]: d.index
```

```
Out[7]: RangeIndex(start=0, stop=150, step=1)
```

```
In [8]: d=d.head(10)
d
```

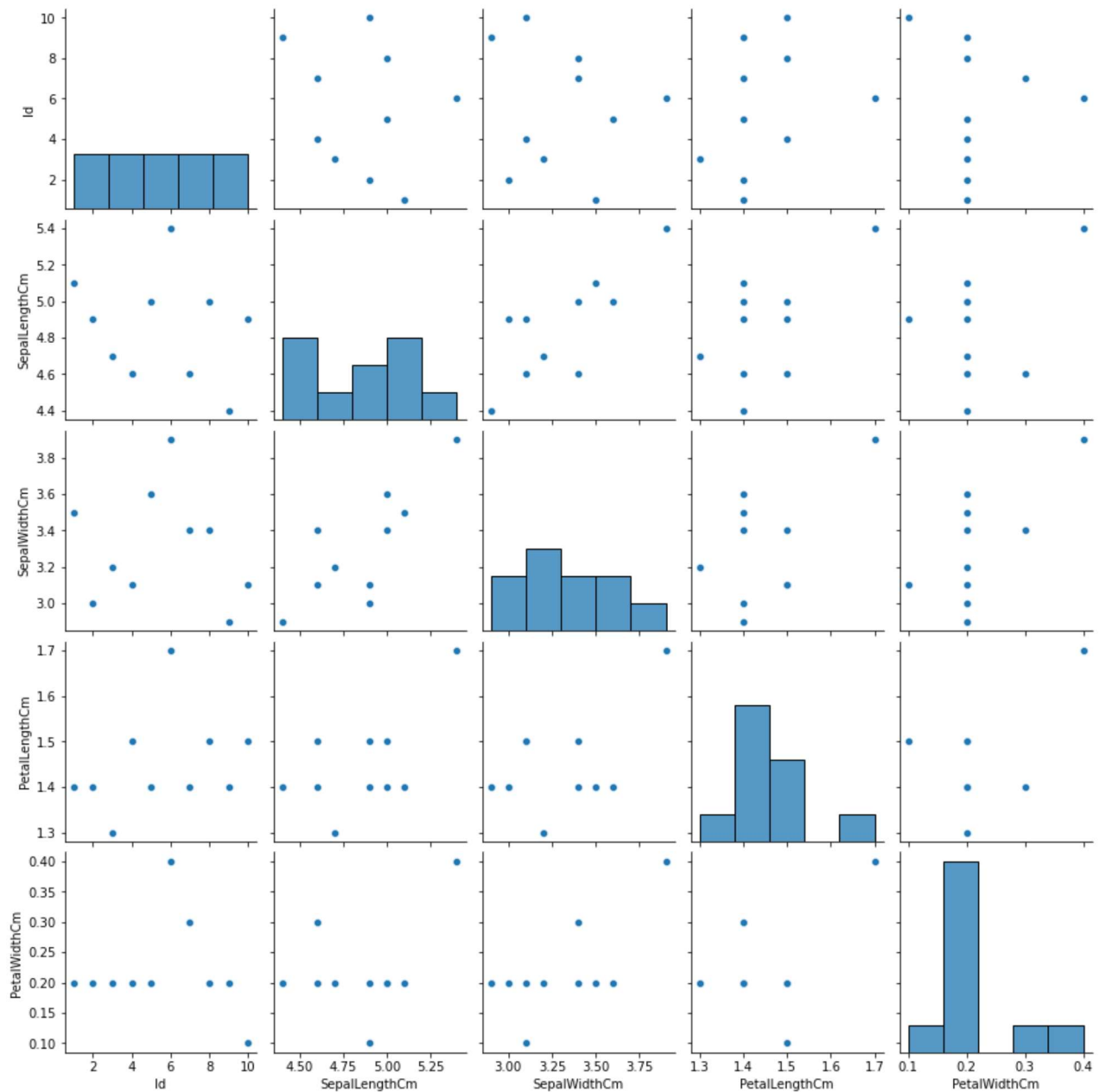
```
Out[8]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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 [9]: sns.pairplot(d)
```

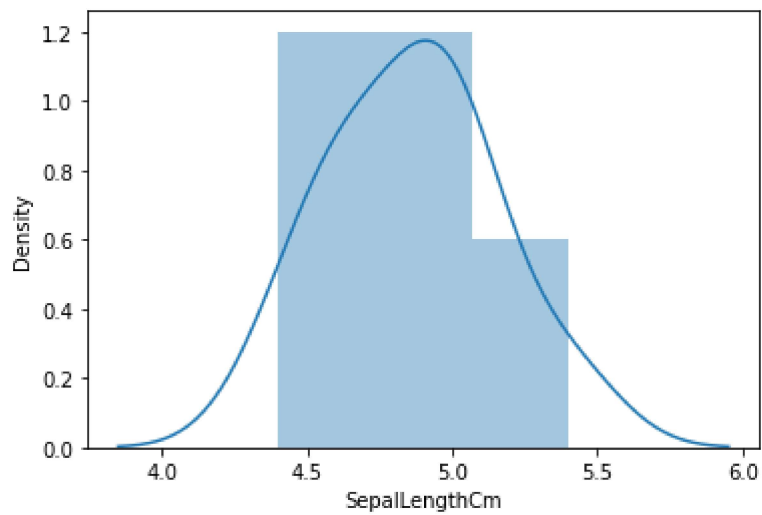
```
Out[9]: <seaborn.axisgrid.PairGrid at 0x1e0df7f3640>
```



```
In [10]: sns.distplot(d['SepalLengthCm'])
```

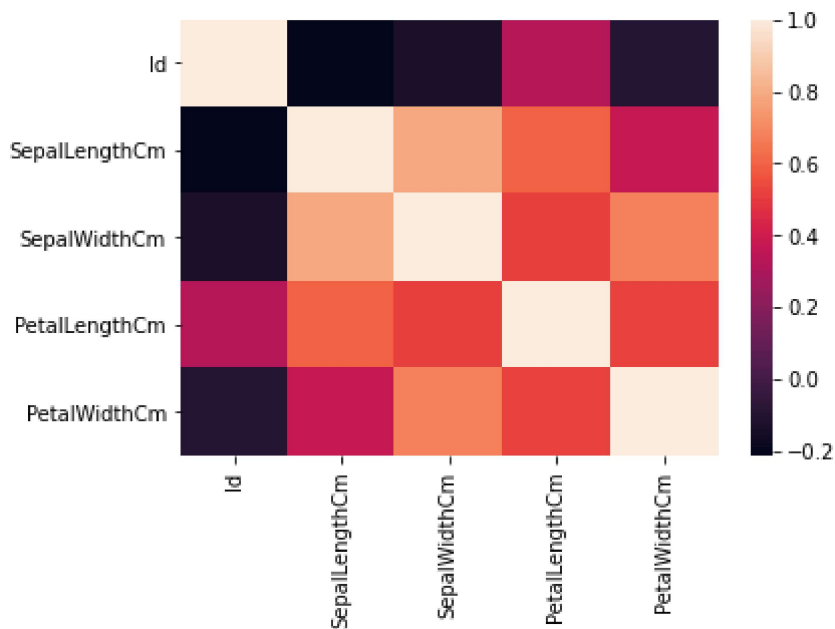
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function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

```
Out[10]: <AxesSubplot:xlabel='SepalLengthCm', ylabel='Density'>
```



```
In [11]: d1=d[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
              'Species']]
          sns.heatmap(d1.corr())
```

Out[11]: <AxesSubplot:>



```
In [25]: x=d1[['Id', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
          y=d1[ 'SepalLengthCm']
```

```
In [26]: 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 [27]: from sklearn.linear_model import LinearRegression
```

```
In [28]: lr=LinearRegression()
          lr.fit(x_train,y_train)
```

Out[28]: LinearRegression()

```
In [29]: print(lr.intercept_)
```

0.1466220572853194

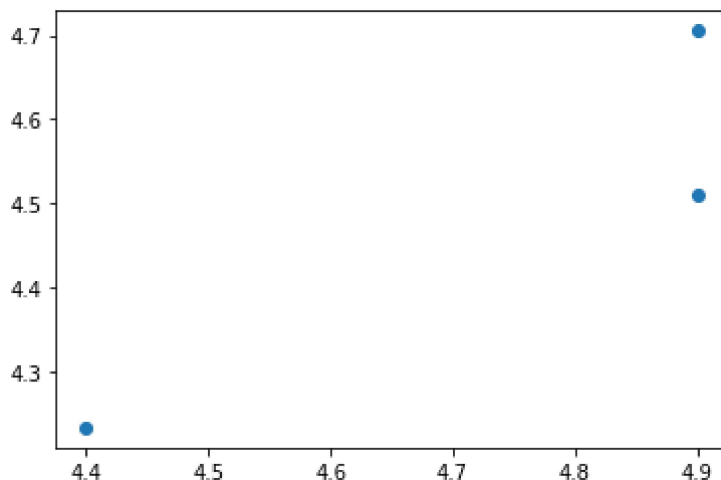
```
In [30]: coeff =pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[30]:

	Co-efficient
Id	-0.024053
SepalWidthCm	1.097749
PetalLengthCm	1.046166
PetalWidthCm	-1.726584

```
In [31]: prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[31]: <matplotlib.collections.PathCollection at 0x1e0e6848b20>



```
In [32]: print(lr.score(x_test,y_test))
```

-0.30153872660560554

```
In [33]: print(lr.score(x_train,y_train))
```

0.9290828562654438

```
In [34]: from sklearn.linear_model import Ridge,Lasso
```

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

Out[35]: Ridge(alpha=10)

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

Out[36]: -0.4890661720429623

In [37]: `la=Lasso(alpha=10)`
`la.fit(x_train,y_train)`

Out[37]: Lasso(alpha=10)

In [38]: `la.score(x_test,y_test)`

Out[38]: -0.5893877551020346

In []: