kaviyadevi 20106064

```
In [6]: #to import libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [4]: #to import dataset
        data=pd.read_csv(r"C:\Users\user\Downloads\14_Iris - 14_Iris.csv")
        data
```

```
Out[4]:
          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 [7]: #to display top 5 rows
        data.head()
```

```
Out[7]:
     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
```

DATA CLEANING AND PREPROCESSING

In [8]:

```
#
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [9]:

```
#to display summary of statistics(here to know min max value)
data.describe()
```

Out[9]:

	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 [10]:

```
#to display the column heading
data.columns
```

Out[10]:

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

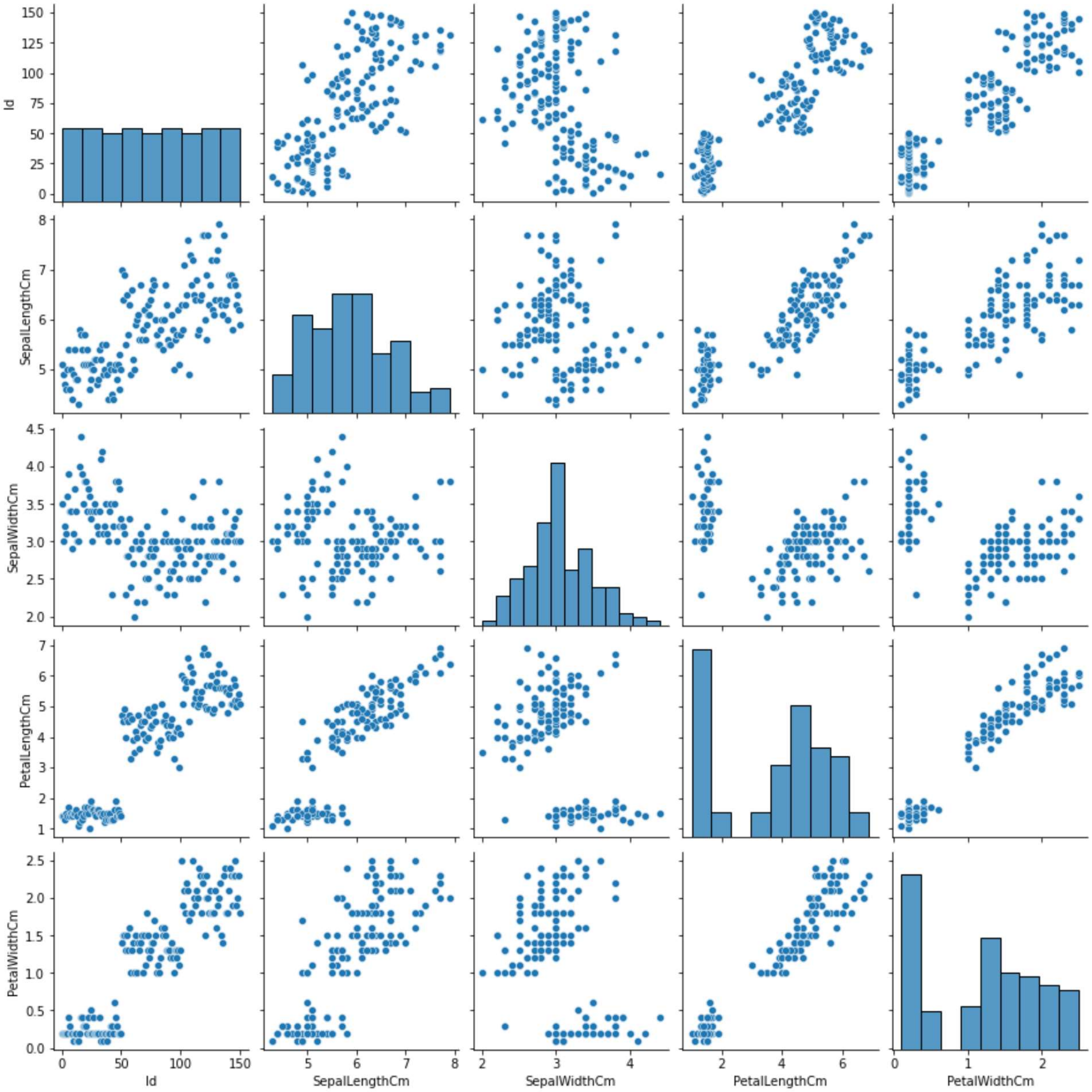
In [11]:

```
#here there is no missing values (identified through info() 5000 data are describ
```

EDA and DATA VISUALIZATION

In [12]:
```
sns.pairplot(data)
```

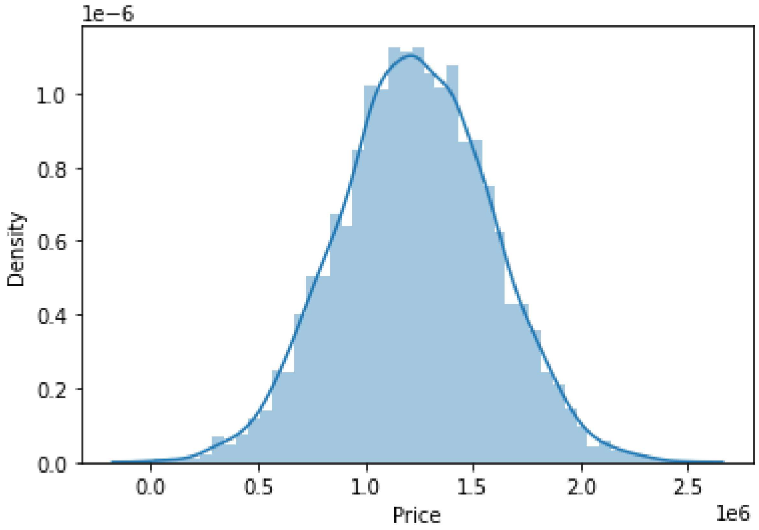
Out[12]:
```
<seaborn.axisgrid.PairGrid at 0x2fbc4f3f2b0>
```



In [41]:
```
sns.distplot(data['PetalWidthCm'])
```

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

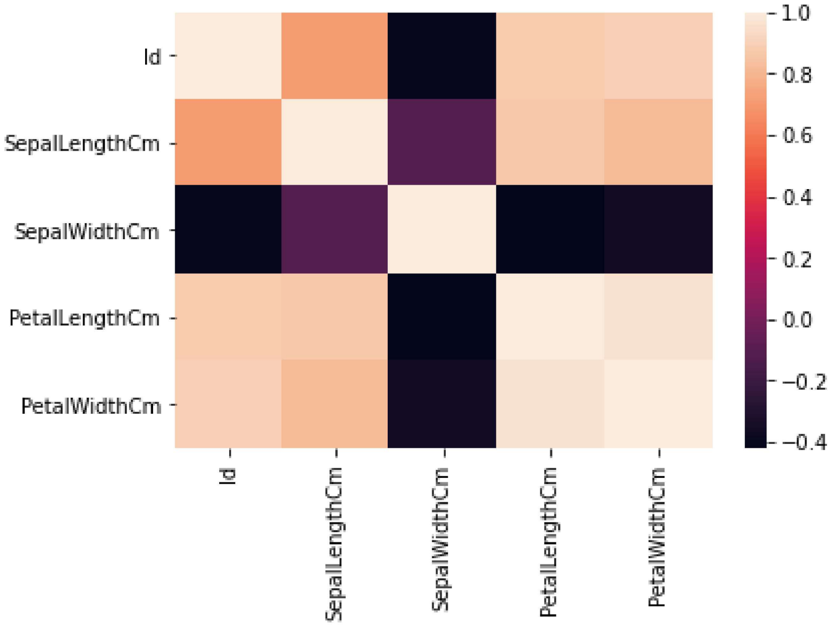
Out[41]:
```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



In [13]:
```
df=data[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species']]
```

In [14]:
```
sns.heatmap(df.corr())
```

Out[14]:
```
<AxesSubplot:>
```



TRAINING MODEL

In [18]:
```
x=df[['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm']]
y=df['PetalWidthCm']
```

In [19]:
```
#to split my dataset into trainning and test

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 [20]:

```
from sklearn.linear_model import LinearRegression

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

Out[20]:

```
LinearRegression()
```

In [21]:

```
#to find intercept
print(lr.intercept_)
```

```
-0.8001697979277445
```

In [22]:

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

Out[22]:

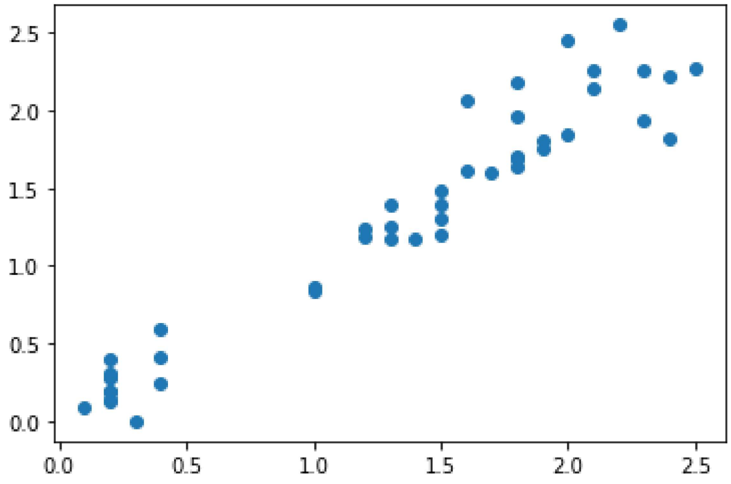
	Co-efficient
Id	0.003633
SepalLengthCm	-0.113643
SepalWidthCm	0.268687
PetalLengthCm	0.414671

In [23]:

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

Out[23]:

```
<matplotlib.collections.PathCollection at 0x2fbc79cc190>
```



In [24]:

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

```
0.9270500245152584
```

RIDGE AND LASSO REGRESSION

In [25]:

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

```
In [26]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)

Out[26]: Ridge(alpha=10)
```

```
In [27]: rr.score(x_test,y_test)

Out[27]: 0.919423436518253
```

```
In [28]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[28]: Lasso(alpha=10)
```

```
In [29]: la.score(x_test,y_test)

Out[29]: 0.6607429271533307
```