DA Assignment 4

Problem Statement : Abalone Age Prediction

Building a Regression Model

Student Name : PRIYADHARSHINI K

Roll Number : 510119205012

```
# Load the Dataset
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

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

```
data=pd.read_csv("/content/drive/MyDrive/abalone.csv")
```

```
data.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15

```
data.describe()
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	8.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	9.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	11.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000

```
data.info()
```

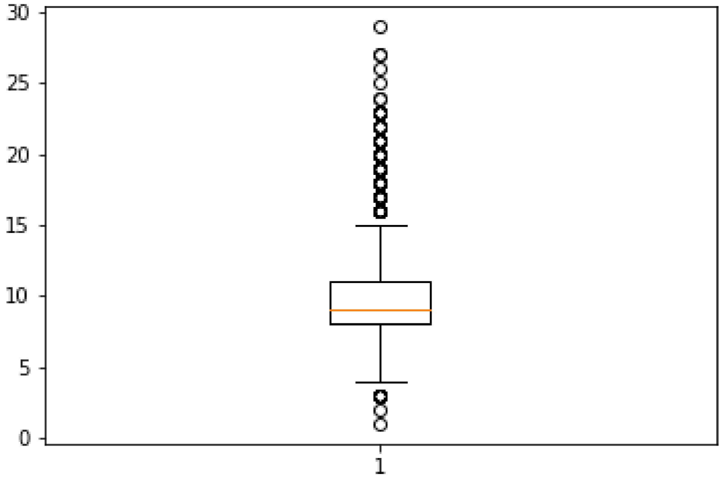
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4177 entries, 0 to 4176
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Sex             4177 non-null   object 
 1   Length          4177 non-null   float64
 2   Diameter        4177 non-null   float64
 3   Height          4177 non-null   float64
 4   Whole weight    4177 non-null   float64
 5   Shucked weight  4177 non-null   float64
 6   Viscera weight  4177 non-null   float64
 7   Shell weight    4177 non-null   float64
 8   Rings           4177 non-null   int64  
```

```
dtypes: float64(7), int64(1), object(1)
memory usage: 293.8+ KB
```

```
#Univariate Analysis
```

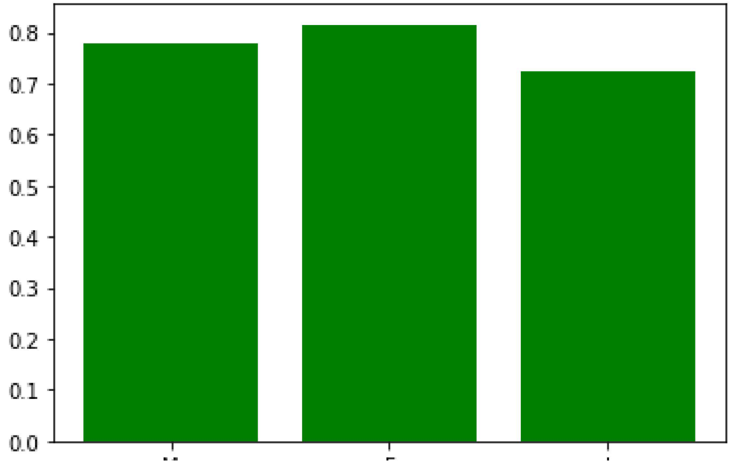
```
plt.boxplot(data['Rings'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f453e0bac90>,
  <matplotlib.lines.Line2D at 0x7f453e0c4210>],
 'caps': [<matplotlib.lines.Line2D at 0x7f453e0c4750>,
  <matplotlib.lines.Line2D at 0x7f453e0c4c90>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f453e0ba6d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f453e0cc250>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f453e120c10>],
 'means': []}
```



```
df=pd.DataFrame(data)
X=list(df.iloc[:,0])
Y=list(df.iloc[:,1])
plt.bar(X,Y,color='g')
```

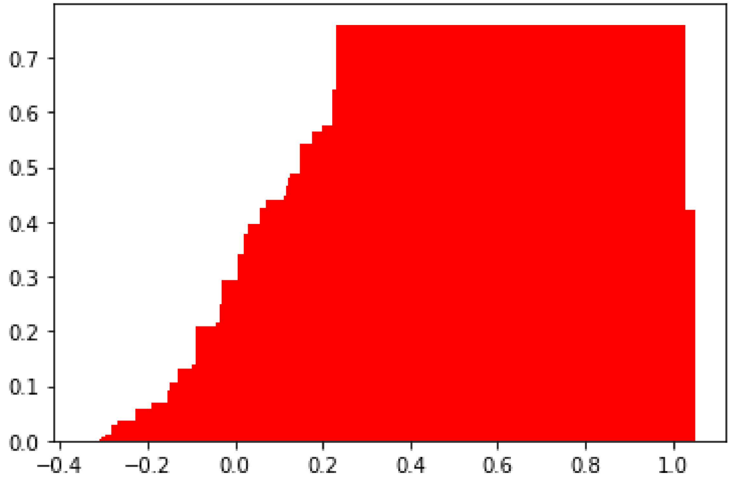
```
<BarContainer object of 4177 artists>
```



```
#Bivariate Analysis
```

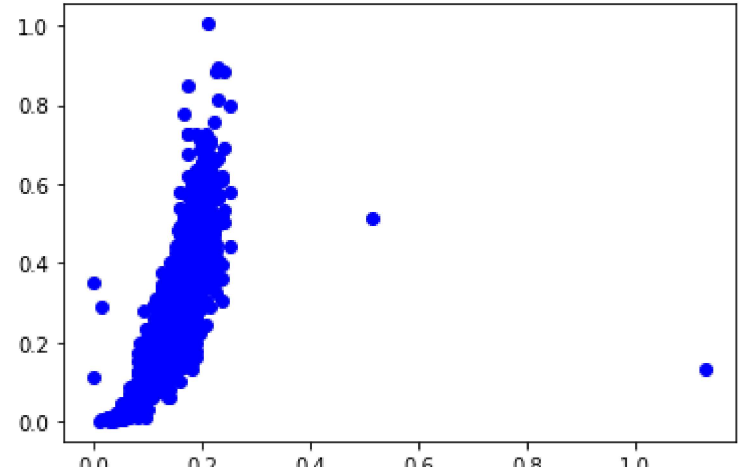
```
plt.bar(data['Diameter'],data['Viscera weight'],color='r')
```

```
<BarContainer object of 4177 artists>
```



```
plt.scatter(data['Height'],data['Shell weight'],color='b')
```

```
<matplotlib.collections.PathCollection at 0x7f4537dc8e50>
```



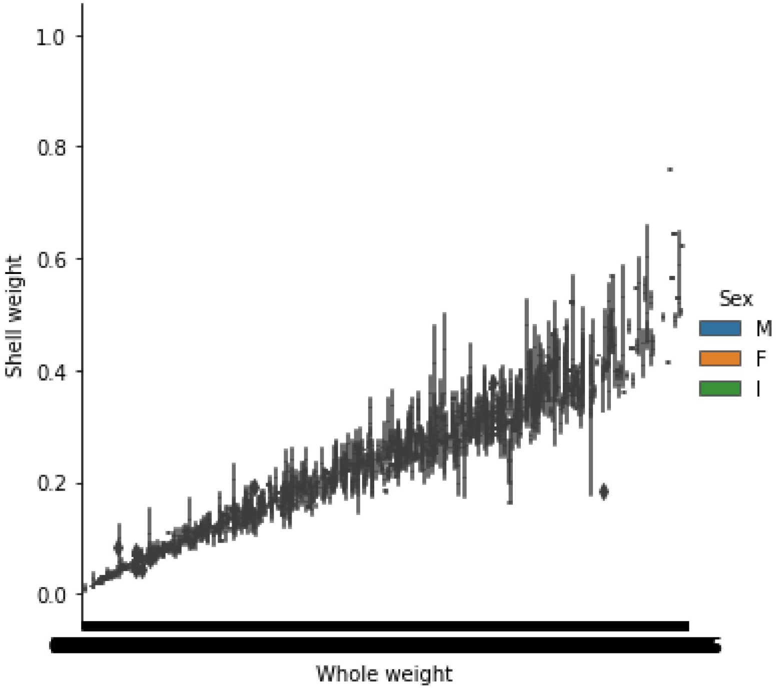
```
# Multivariate Analysis
```

```
sns.catplot(data=data,x="Height",y="Shucked weight",hue="Sex")
```

```
      <seaborn.axisgrid.FacetGrid at 0x7f4537bbe2d0>
```

```
sns.catplot(data=data,x="Whole weight",y="Shell weight",hue="Sex",kind="box")
```

```
      <seaborn.axisgrid.FacetGrid at 0x7f4538a19ed0>
```



```
# Perform Descriptive Statistics on the Dataset
```

```
data.mean()
```

```
      /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reduct
        """Entry point for launching an IPython kernel.
      Length            0.523992
      Diameter          0.407881
      Height            0.139516
      Whole weight      0.828742
      Shucked weight    0.359367
      Viscera weight    0.180594
```

```
Shell weight      0.238831
Rings             9.933684
dtype: float64
```

```
data.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reduct
  """Entry point for launching an IPython kernel.
Length            0.5450
Diameter          0.4250
Height            0.1400
Whole weight      0.7995
Shucked weight    0.3360
Viscera weight    0.1710
Shell weight      0.2340
Rings             9.0000
dtype: float64
```

```
data.describe()
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	8.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	9.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	11.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000

```
data.shape

     (4177, 9)
```

```
# Handling the Missing Values
```

```
data.isna().sum()

     Sex               0
     Length            0
     Diameter          0
     Height            0
     Whole weight      0
     Shucked weight    0
     Viscera weight    0
     Shell weight      0
     Rings             0
     dtype: int64
```

```
# Find the Outliers and Replace the Outliers
```

```
sns.boxplot(data['Height'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f4539b0ddd0>
```

```
qnt=data.quantile(q=[0.25,0.75])
qnt
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0.25	0.450	0.35	0.115	0.4415	0.186	0.0935	0.130	8.0
0.75	0.615	0.48	0.165	1.1530	0.502	0.2530	0.329	11.0

```
IQR=qnt.loc[0.75] - qnt.loc[0.25]
IQR
```

```
Length            0.1650
Diameter          0.1300
Height            0.0500
Whole weight      0.7115
Shucked weight    0.3160
Viscera weight    0.1595
Shell weight      0.1990
Rings             3.0000
dtype: float64
```

```
upper_extreme=qnt.loc[0.75]+1.5*IQR
upper_extreme
```

```
Length             0.86250
Diameter           0.67500
Height             0.24000
Whole weight       2.22025
Shucked weight     0.97600
```

```
     Viscera weight     0.49225
     Shell weight       0.62750
     Rings             15.50000
     dtype: float64
```

```
lower_extreme=qnt.loc[0.25]-1.5*IQR
lower_extreme
```

```
     Length            0.20250
     Diameter          0.15500
     Height            0.04000
     Whole weight     -0.62575
     Shucked weight   -0.28800
     Viscera weight   -0.14575
     Shell weight     -0.16850
     Rings             3.50000
     dtype: float64
```

```
df=data[(data['Height']<upper_extreme['Height'])&(data['Height']>lower_extreme['Height'])]
```

```
data.shape
```

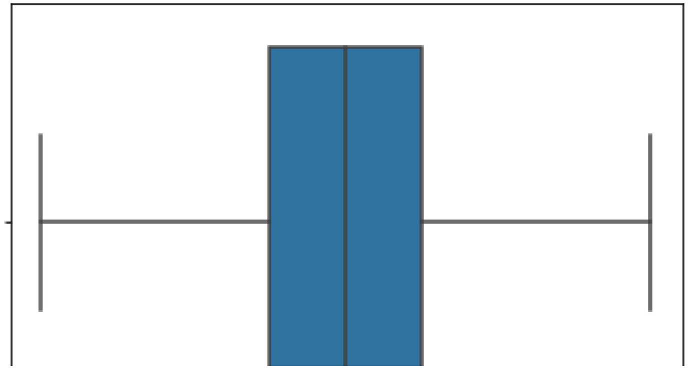
```
     (4177, 9)
```

```
df.shape
```

```
     (4148, 9)
```

```
sns.boxplot(df['Height'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f453932c750>
```



```
# Check for Categorical Columns and Perform Encoding
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
df['Sex']=le.fit_transform(df['Sex'])
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
```

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15

```
# Split the Data into Dependent and Independent Variables
```

2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9

```
y=df['Rings']
x=df.drop(columns=['Rings'],axis=1)
```

4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

```
x.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210
3	2	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055

```
# Scale the Independent Variables
```

```
from sklearn.preprocessing import scale
```

```
names = x.columns
names
```

```
Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight'],
      dtype='object')
```

```
x=scale(x)
x
```

```
     array([[ 1.1486969 , -0.60262947, -0.45591118, ..., -0.61827733,
             -0.73839808, -0.64915159],
            [ 1.1486969 , -1.49739064, -1.48593272, ..., -1.18589191,
             -1.22152247, -1.22814023],
            [-1.27860557,  0.03648565,  0.11060067, ..., -0.472968  ,
             -0.36570212, -0.2149101 ],
            ...,
            [ 1.1486969 ,  0.63299309,  0.67711252, ...,  0.74853858,
              0.97784382,  0.49435099],
            [-1.27860557,  0.84603146,  0.78011468, ...,  0.77351362,
              0.73398103,  0.40750269],
            [ 1.1486969 ,  1.57036193,  1.50112976, ...,  2.65572357,
              1.7968547 ,  1.84773695]])
```

```
x=pd.DataFrame(x,columns=[names])
```

```
x.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
0	1.148697	-0.602629	-0.455911	-1.194353	-0.653289	-0.618277	-0.738398	-0.649152
1	1.148697	-1.497391	-1.485933	-1.327737	-1.246310	-1.185892	-1.221522	-1.228140
2	-1.278606	0.036486	0.110601	-0.127275	-0.318237	-0.472968	-0.365702	-0.214910
3	1.148697	-0.730452	-0.455911	-0.394044	-0.649178	-0.659146	-0.618767	-0.612965
4	-0.064954	-1.667821	-1.588935	-1.594507	-1.288449	-1.231301	-1.304344	-1.336701

```
# Split the data into training and testing
```

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

```
df.iloc[:,:-1]
df.iloc[:,-1]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.05,random_state=0)
```

```
x_train
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
531	-0.064954	-0.048730	-0.043903	0.006110	-0.522763	-0.729530	-0.333494	-0.439268
2973	-1.278606	0.846031	0.831616	-0.794199	0.625252	0.753079	0.094416	0.382172
528	-1.278606	-0.773060	-0.764918	-0.794199	-0.928730	-0.868028	-0.876434	-0.938646
2062	-1.278606	0.803424	0.831616	0.539649	0.460809	0.600959	0.577541	0.237425
991	-1.278606	0.590385	0.419607	0.272879	0.580030	1.025534	0.361285	0.074584
...	...	...	...	...	...	...	...	...
1033	-1.278606	1.272108	1.604132	1.473342	2.067208	1.833818	1.658819	2.267504
3264	1.148697	0.079093	0.110601	0.539649	-0.229849	-0.388961	-0.158649	-0.106350
1653	-1.278606	0.675601	0.986119	1.073187	1.237801	1.670345	0.761588	1.384546
2607	-0.064954	-1.454783	-1.279928	-1.194353	-1.242199	-1.204056	-1.230725	-1.217284
2732	-0.064954	-0.006122	-0.095404	-0.260660	-0.385042	-0.071097	-0.485333	-0.287284

3940 rows × 8 columns

```
x_test
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
825	-0.064954	-0.687845	-0.867920	-1.060968	-0.932842	-0.822619	-0.747600	-1.090630
4077	-1.278606	1.655577	1.861637	1.740111	1.886321	1.883768	2.220164	1.739177
971	-1.278606	0.206916	0.368106	0.406264	0.326172	0.832546	-0.029816	-0.157011
1499	1.148697	1.059070	0.780115	0.006110	0.705418	0.519222	0.572940	-0.178723
3176	-1.278606	-1.199137	-1.125425	-1.060968	-1.149700	-1.131401	-1.143302	-1.155767
...	...	...	...	...	...	...	...	...
1927	1.148697	0.888639	0.831616	0.406264	0.864722	1.223064	0.595945	0.356841
2052	1.148697	0.206916	0.007599	-0.260660	0.079508	0.385265	0.278464	-0.193198
3169	-1.278606	0.718208	0.625611	0.539649	0.380644	0.296717	0.407297	0.581199

```
y_train
```

```
536     11
2997     9
533      9
2079     9
1000    11
        ..
1042    12
3289    15
1667    11
2630     6
2756     8
Name: Rings, Length: 3940, dtype: int64
```

```
y_test
```

```
834      7
4106    11
980      9
```

```
1513     8
3201     6
        ..
1942     9
2069     9
3194    12
998      8
604     12
Name: Rings, Length: 208, dtype: int64
```

```
# Build the Model
```

```
from sklearn.linear_model import LinearRegression

lin_reg=LinearRegression()
```

```
# Train the Model
```

```
lin_reg.fit(x,y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that a
  FutureWarning,
LinearRegression()
```

```
# Test the Model
```

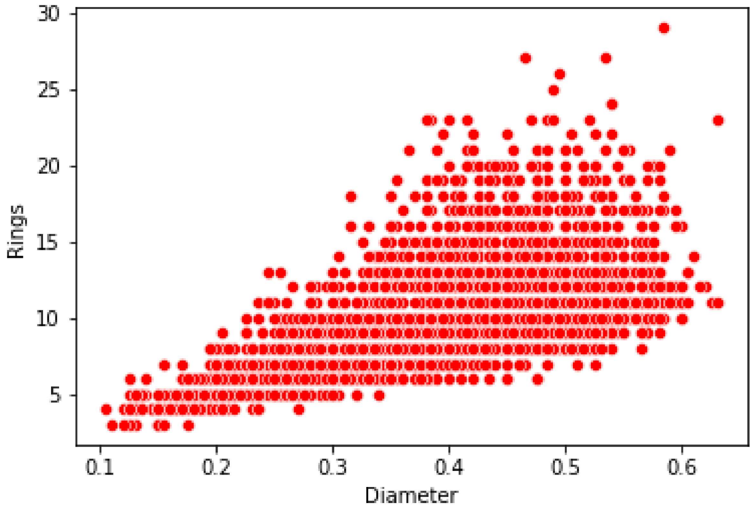
```
lin_reg.predict(x)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that a
  FutureWarning,
array([ 8.50595288,  7.31132061, 10.72922803, ..., 11.51826065,
        9.37712194, 11.10957517])
```

predict=lin_reg.predict(x) predict

```
sns.scatterplot(df['Diameter'],df['Rings'],color='red')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f4532dd79d0>
```



```
def viz_linear():
  sns.scatter(x,y,color='red')
  sns.plot(x,prediction,color='blue')

  viz_linear()
```

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=2)
x_poly=poly_reg.fit_transform(x)
x_poly
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that a
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that a
  FutureWarning,
array([[ 1.        ,  1.1486969 , -0.60262947, ...,  0.54523172,
         0.47933228,  0.42139778],
       [ 1.        ,  1.1486969 , -1.49739064, ...,  1.49211715,
         1.50020089,  1.50832843],
       [ 1.        , -1.27860557,  0.03648565, ...,  0.13373804,
         0.07859308,  0.04618635],
       ...,
       [ 1.        ,  1.1486969 ,  0.63299309, ...,  0.95617853,
         0.48339806,  0.2443829 ],
       [ 1.        , -1.27860557,  0.84603146, ...,  0.53872815,
         0.29909925,  0.16605844],
       [ 1.        ,  1.1486969 ,  1.57036193, ...,  3.2286868 ,
         3.32011482,  3.41413183]])
```

```
pol_reg=LinearRegression()
pol_reg.fit(x_poly,y)
prediction=pol_reg.predict(x_poly)
```

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	2	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

```
# Performance using Evaluation Metrics.
```

```
from sklearn import metrics
from sklearn.metrics import mean_squared_error
```

```
metrics.r2_score(y,prediction)
```

```
    0.5790843515069326
```

```
np.sqrt(mean_squared_error(y,prediction))
```

```
    2.077178028336633
```

```
metrics.r2_score(y,prediction)
```

```
    0.5790843515069326
```

```
np.sqrt(mean_squared_error(y,prediction))
```

```
    2.077178028336633
```

```
print('R_squared:',metrics.r2_score(y,prediction))
print('MSE:',mean_squared_error(y,prediction))
print('RMSE:',np.sqrt(mean_squared_error(y,prediction)))
```

```
    R_squared: 0.5790843515069326
    MSE: 4.314668561404462
    RMSE: 2.077178028336633
```

✓ 0s completed at 11:10 PM ● ✕