Assignment 3

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

2.Load Dataset

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

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
#import dataset

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

```
data.head()
```

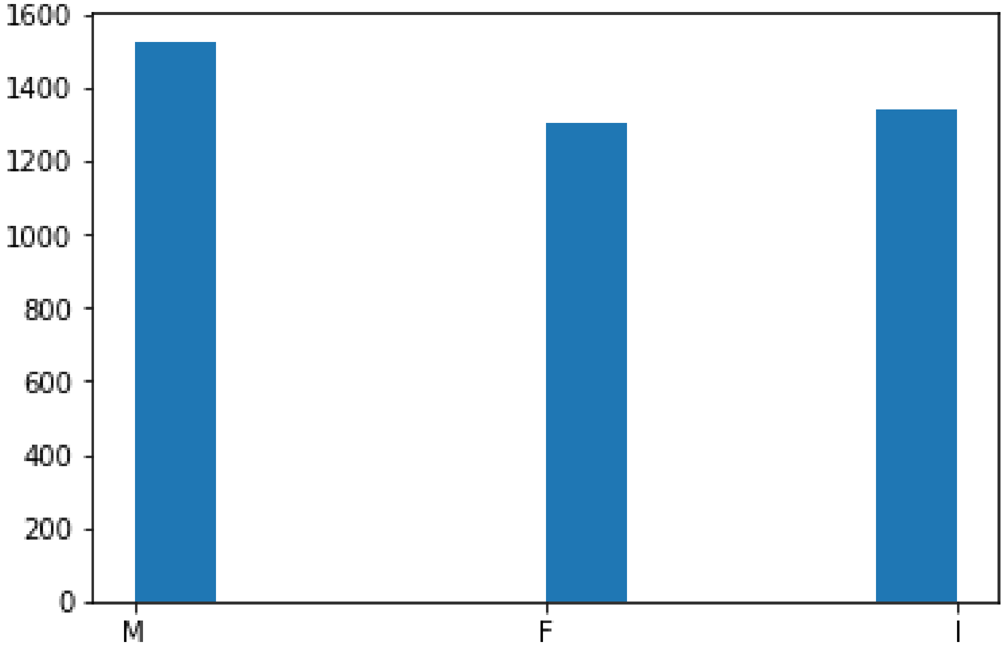
```
  Sex  Length  Diameter  Height  Whole weight  Shucked weight  Viscera
weight  \
0   M   0.455     0.365   0.095        0.5140          0.2245
0.1010
1   M   0.350     0.265   0.090        0.2255          0.0995
0.0485
2   F   0.530     0.420   0.135        0.6770          0.2565
0.1415
3   M   0.440     0.365   0.125        0.5160          0.2155
0.1140
4   I   0.330     0.255   0.080        0.2050          0.0895
0.0395

   Shell weight  Rings
0         0.150     15
1         0.070      7
2         0.210      9
3         0.155     10
4         0.055      7
```

1. Visualization

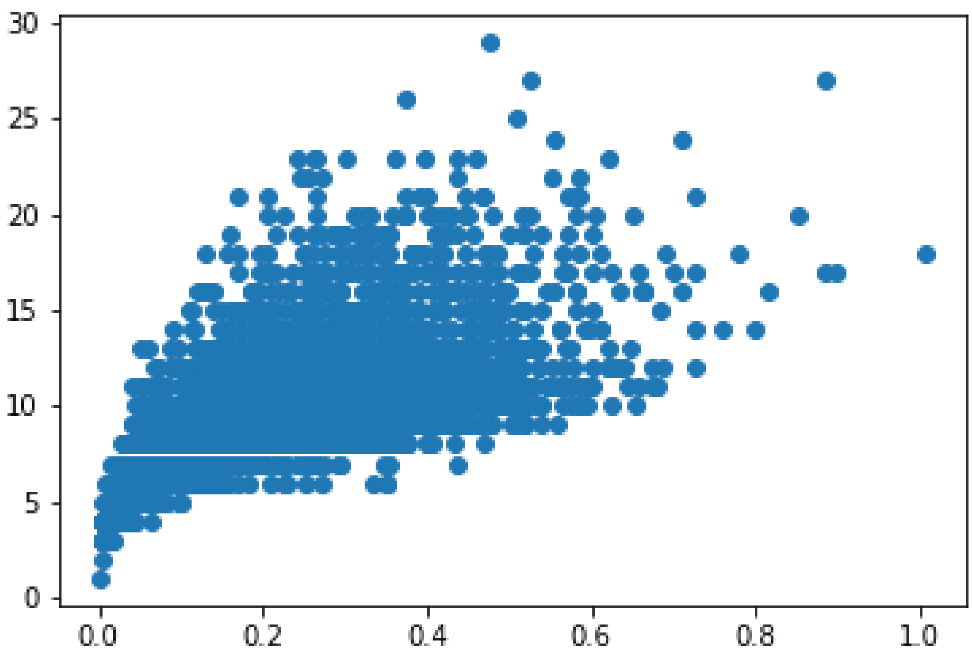
```
plt.hist(data['Sex'])
```

```
(array([1528.,    0.,    0.,    0.,    0., 1307.,    0.,    0.,    0.,
        1342.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



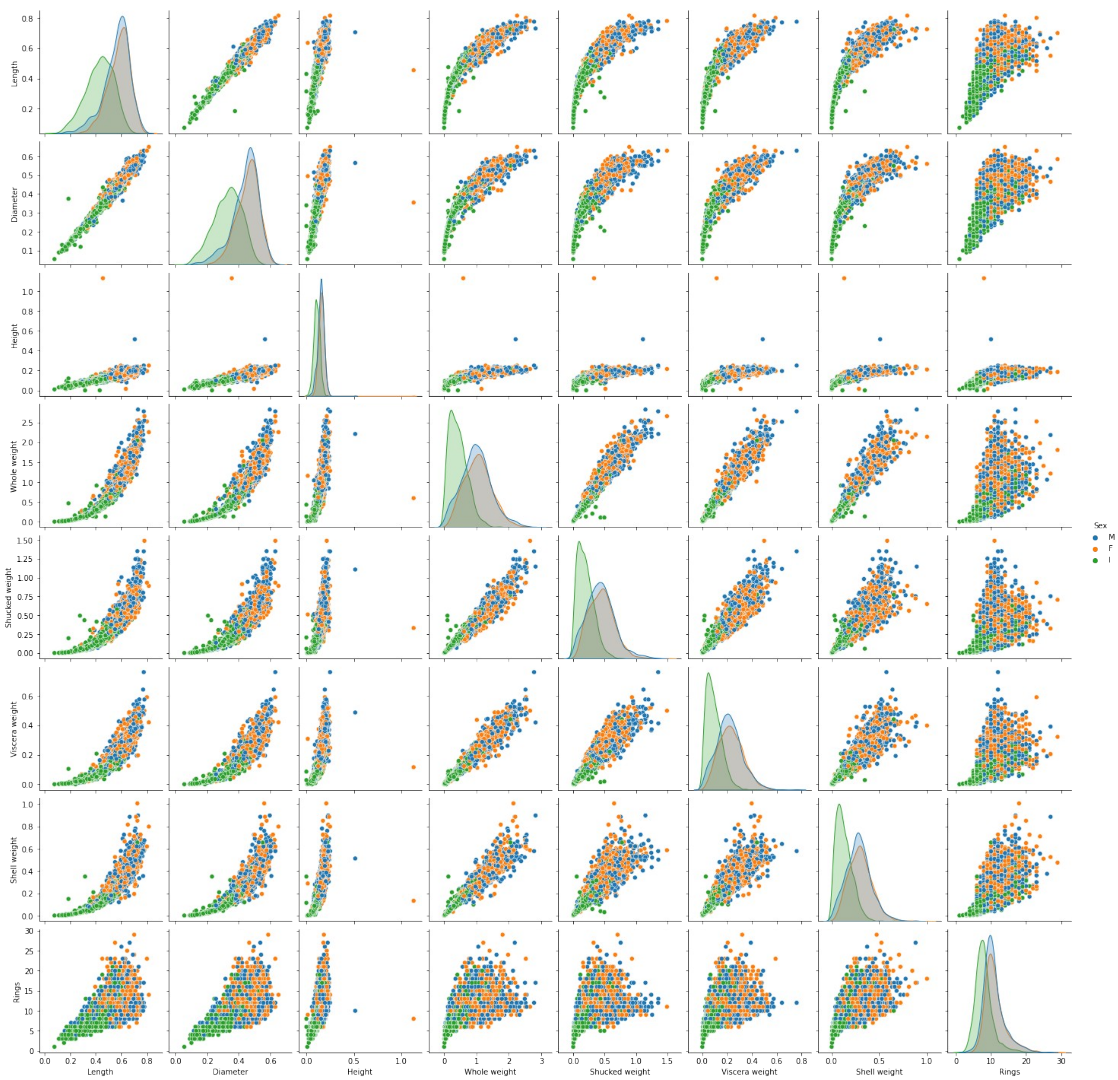

```
plt.scatter(data['Shell weight'],data['Rings'])
```

<matplotlib.collections.PathCollection at 0x7f57a1bbefd0>




```
sns.pairplot(data,hue='Sex')
```

<seaborn.axisgrid.PairGrid at 0x7f57a1b8ab10>



4.Descriptive statistics

```
data.describe()

             Length     Diameter       Height  Whole weight  Shucked
weight  \
count  4177.000000  4177.000000  4177.000000   4177.000000
4177.000000
mean      0.523992     0.407881     0.139516      0.828742
0.359367
std       0.120093     0.099240     0.041827      0.490389
0.221963
min       0.075000     0.055000     0.000000      0.002000
0.001000
25%       0.450000     0.350000     0.115000      0.441500
0.186000
50%       0.545000     0.425000     0.140000      0.799500
0.336000
```

```
75%        0.615000     0.480000     0.165000      1.153000
0.502000
max        0.815000     0.650000     1.130000      2.825500
1.488000

       Viscera weight  Shell weight        Rings
count     4177.000000   4177.000000  4177.000000
mean         0.180594      0.238831     9.933684
std          0.109614      0.139203     3.224169
min          0.000500      0.001500     1.000000
25%          0.093500      0.130000     8.000000
50%          0.171000      0.234000     9.000000
75%          0.253000      0.329000    11.000000
max          0.760000      1.005000    29.000000
```

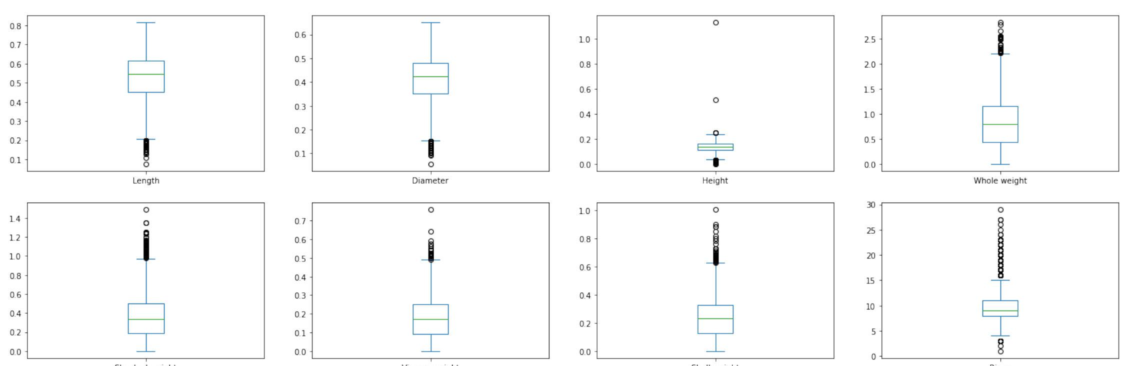
1. Missing Values

```
data.isna().sum() #No missing values
```

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

6.Outliers

```
data.plot(kind="box",subplots=True,layout=(7,4),figsize=(25,30));
```




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

```
      Length  Diameter  Height  Whole weight  Shucked weight  Viscera
weight  \
0.25   0.450      0.35   0.115        0.4415           0.186
0.0935
```

```
0.75   0.615      0.48   0.165       1.1530          0.502
0.2530

      Shell weight  Rings
0.25         0.130    8.0
0.75         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
```

```
lower=qnt.loc[0.25]-(1.5*iqr)
lower
```

```
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
```

```
upper=qnt.loc[0.75]+(1.5*iqr)
upper
```

```
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
```

```
data.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
```

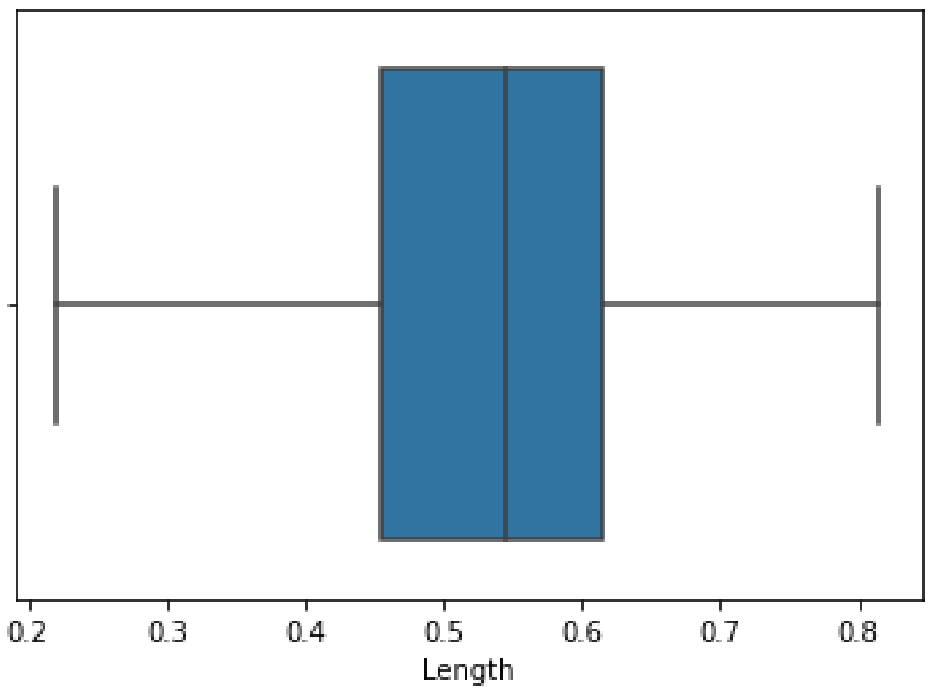
```
reduction.
  """Entry point for launching an IPython kernel.
```

```
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['Length']=np.where(data['Length']< 0.22,0.53,data['Length'])
```

```
sns.boxplot(x=data['Length'])
```

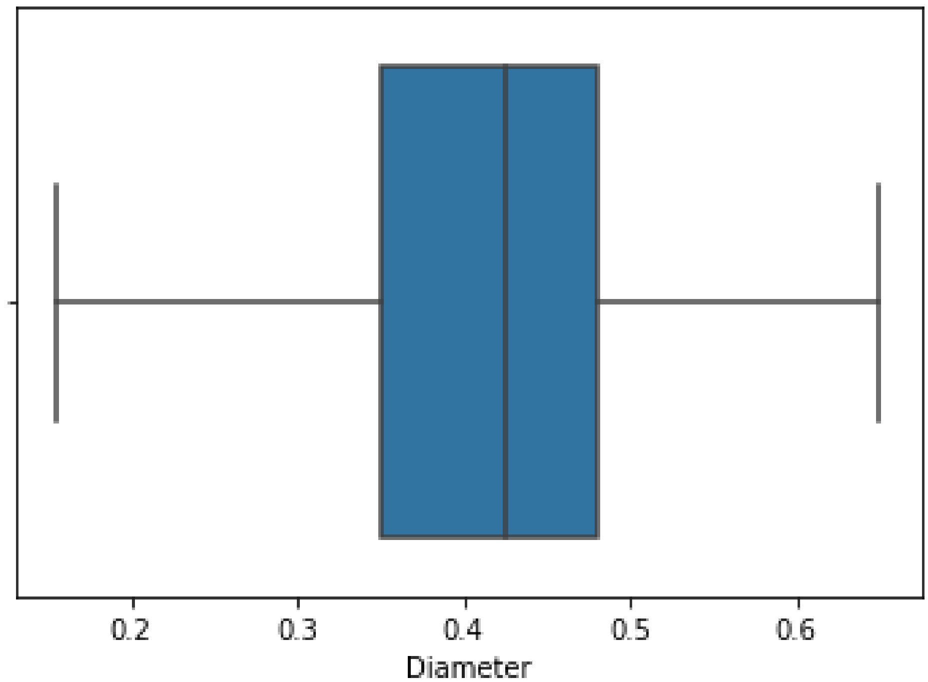
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b444310>
```




```
data['Diameter']=np.where(data['Diameter']<
0.155,0.407,data['Diameter'])
```

```
sns.boxplot(x=data['Diameter'])
```

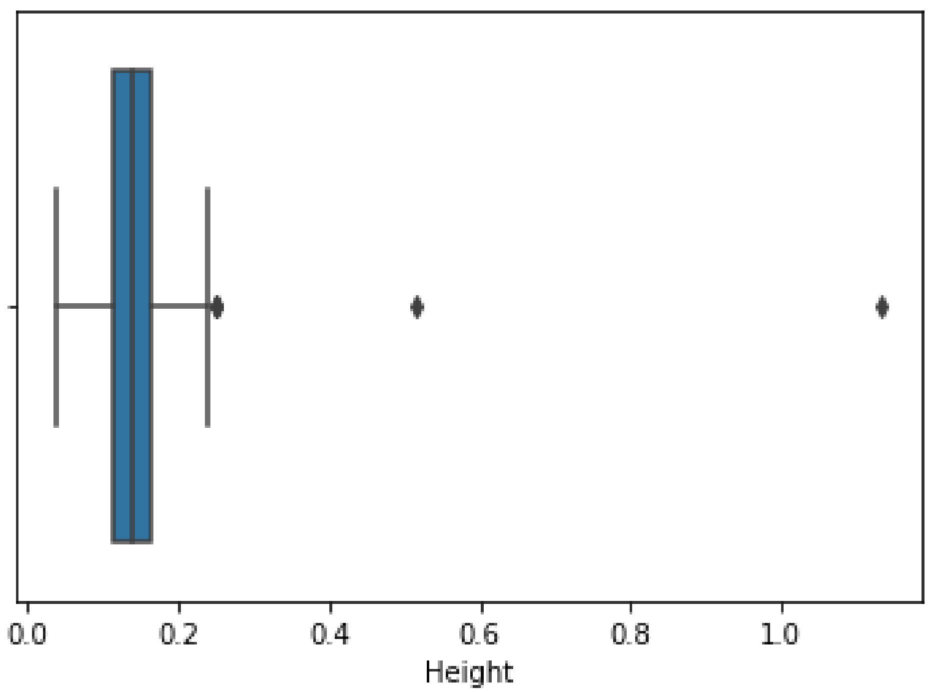
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b4305d0>
```




```
data['Height']=np.where(data['Height']< 0.04,0.14,data['Height'])
```

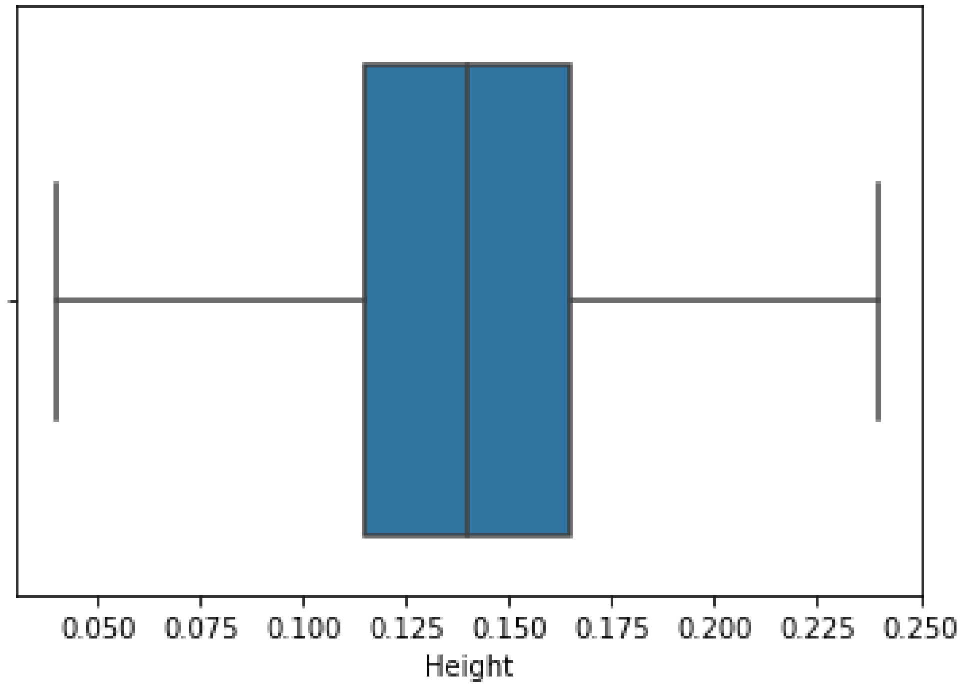
```
sns.boxplot(x=data['Height'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d96a210>
```



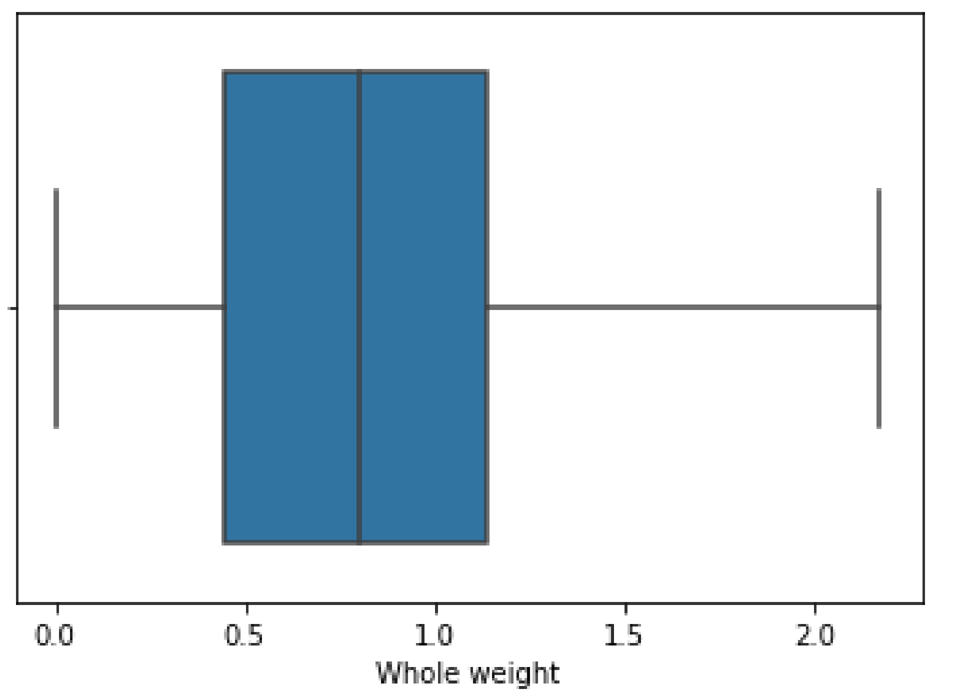

```
data['Height']=np.where(data['Height']>0.24,0.14,data['Height'])
sns.boxplot(x=data['Height'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b6e25d0>
```



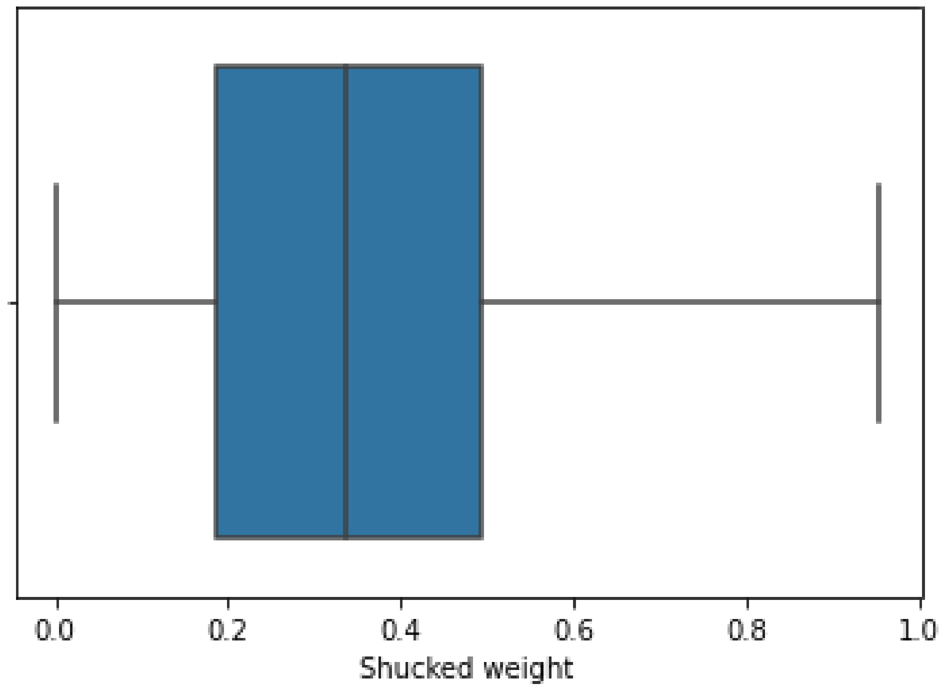

```
data['Whole weight']=np.where(data['Whole
weight']>2.18,0.83,data['Whole weight'])
sns.boxplot(x=data['Whole weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d84d550>
```



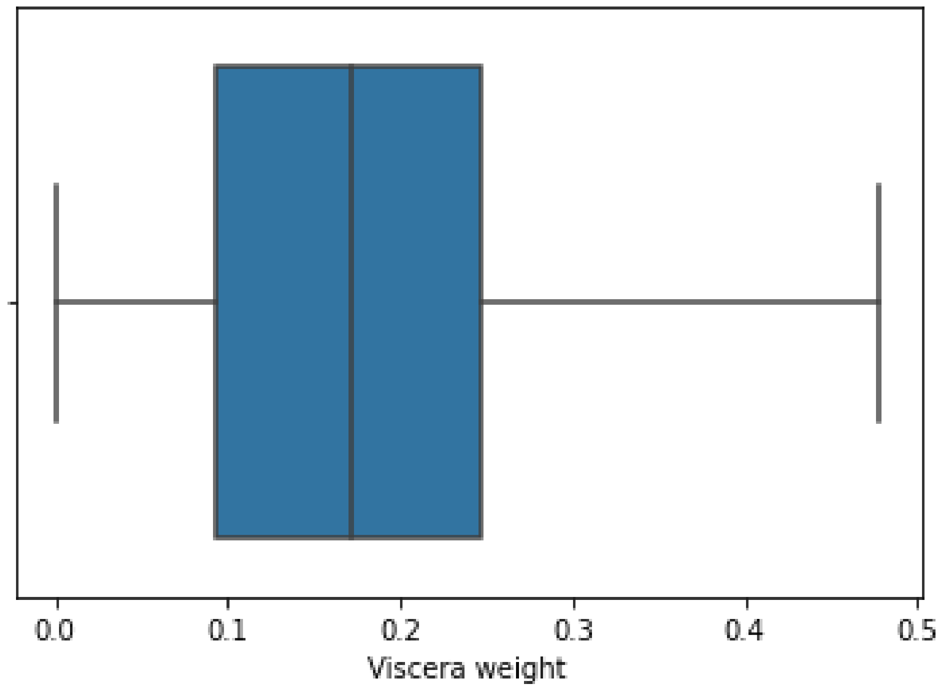
```
data['Shucked weight']=np.where(data['Shucked weight']>0.958,0.359367,data['Shucked weight'])
sns.boxplot(x=data['Shucked weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b560ad0>
```



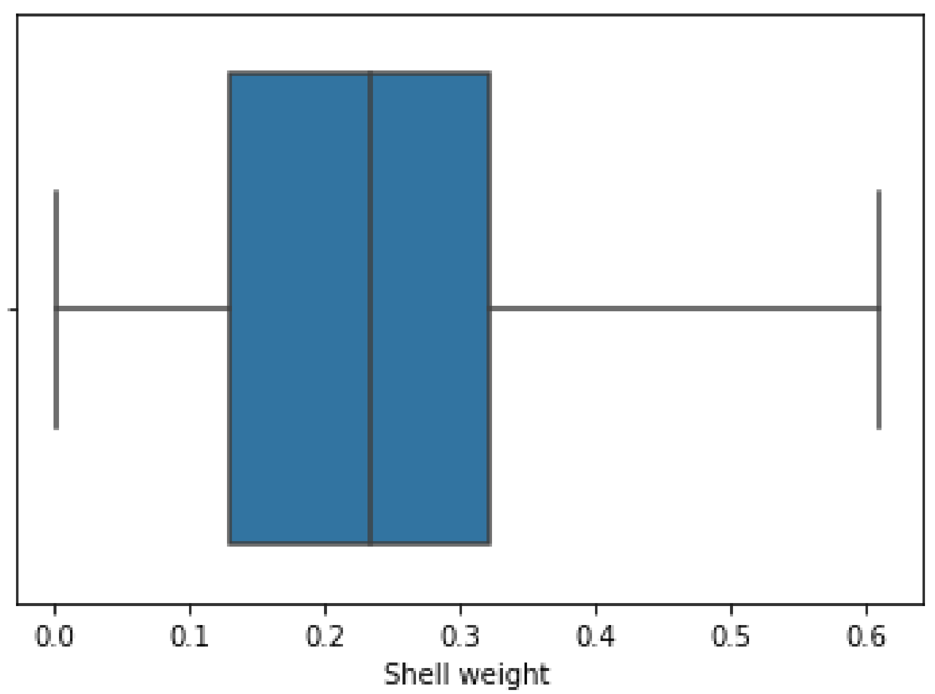

```
data['Viscera weight']=np.where(data['Viscera weight']>0.478,0.18,data['Viscera weight'])
sns.boxplot(x=data['Viscera weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d76fa10>
```



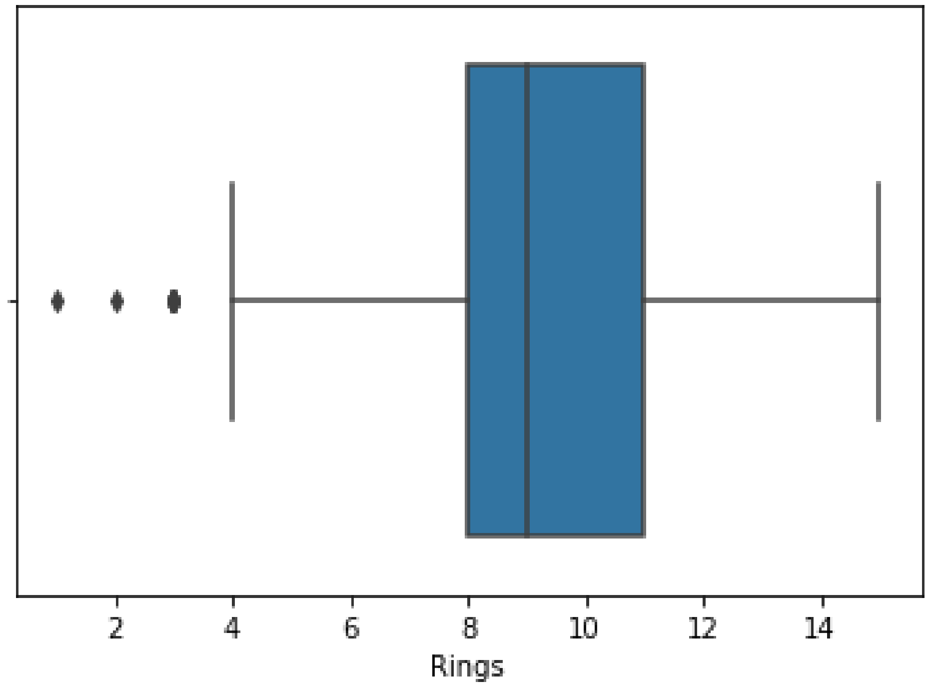

```
data['Shell weight']=np.where(data['Shell
weight']>0.61,0.238831,data['Shell weight'])
sns.boxplot(x=data['Shell weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b75fc90>
```



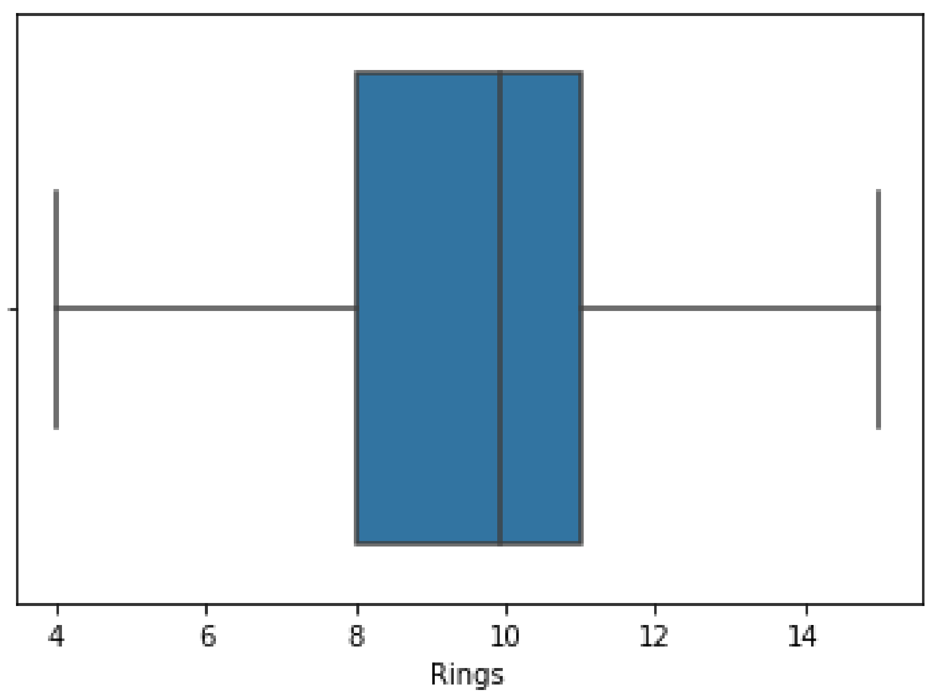

```
data['Rings']=np.where(data['Rings']>15.5,9.933684,data['Rings'])
sns.boxplot(x=data['Rings'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b6a6b10>
```

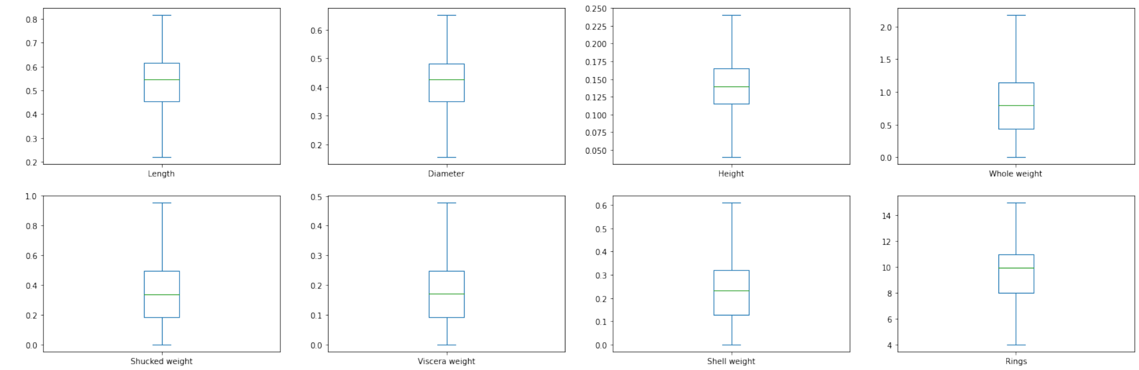



```
data['Rings']=np.where(data['Rings']<3.5,9.933684,data['Rings'])
sns.boxplot(x=data['Rings'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579b6e1dd0>
```




```
data.plot(kind="box",subplots=True,layout=(7,4),figsize=(25,30));
```



1. Categorical value and encoding

```
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   float64
dtypes: float64(8), object(1)
memory usage: 293.8+ KB
```

```
data['Sex'].unique()
```

```
array(['M', 'F', 'I'], dtype=object)
```

```
#sex is categorical
df=pd.get_dummies(data,columns=['Sex'])
# data['Sex'].replace({'I':2,'M':1,'F':0})
df.head()
```

```
   Length  Diameter  Height  Whole weight  Shucked weight  Viscera 
weight  \
0   0.455     0.365   0.095        0.5140          0.2245          
0.1010   
1   0.350     0.265   0.090        0.2255          0.0995          
0.0485   
2   0.530     0.420   0.135        0.6770          0.2565          
0.1415   
3   0.440     0.365   0.125        0.5160          0.2155          
0.1140   
```

```
4   0.330     0.255   0.080        0.2050          0.0895
0.0395

   Shell weight  Rings  Sex_F  Sex_I  Sex_M
0         0.150   15.0      0      0      1
1         0.070    7.0      0      0      1
2         0.210    9.0      1      0      0
3         0.155   10.0      0      0      1
4         0.055    7.0      0      1      0
```

1. Split data into dependent and independent

```
# dependent variable
y=df['Rings'].values
y

array([15.,  7.,  9., ...,  9., 10., 12.])

#independent variable

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

array([[0.455, 0.365, 0.095, ..., 0.   , 0.   , 1.   ],
       [0.35 , 0.265, 0.09 , ..., 0.   , 0.   , 1.   ],
       [0.53 , 0.42 , 0.135, ..., 1.   , 0.   , 0.   ],
       ...,
       [0.6  , 0.475, 0.205, ..., 0.   , 0.   , 1.   ],
       [0.625, 0.485, 0.15 , ..., 1.   , 0.   , 0.   ],
       [0.71 , 0.555, 0.195, ..., 0.   , 0.   , 1.   ]])
```

9.Scale the independent

```
from sklearn.preprocessing import scale

x=scale(x)

x

array([[-0.66489959, -0.50167301, -1.19856285, ..., -0.67483383,
        -0.68801788,  1.31667716],
       [-1.60274931, -1.57291477, -1.332413  , ..., -0.67483383,
        -0.68801788,  1.31667716],
       [ 0.00499306,  0.08750996, -0.12776168, ...,  1.48184628,
        -0.68801788, -0.75948762],
       ...,
       [ 0.6302262 ,  0.67669293,  1.74614038, ..., -0.67483383,
        -0.68801788,  1.31667716],
       [ 0.85352375,  0.78381711,  0.27378876, ...,  1.48184628,
        -0.68801788, -0.75948762],
       [ 1.61273542,  1.53368634,  1.47844008, ..., -0.67483383,
        -0.68801788,  1.31667716]])
```

10 split the data into training and testing

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

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,rando
m_state=0)
```

1. Build the model

```
from sklearn.linear_model import LinearRegression

MLR = LinearRegression()
```

1. Train the model

```
MLR.fit(x_train,y_train)

LinearRegression()
```

1. Test the model

```
y_pred = MLR.predict(x_test)
y_pred

array([11.02365129,  8.53128844, 10.31763804,  6.01778772,
10.51663617,
       10.89512079,  8.53196211,  9.54586451,  7.95152418,
10.72178073,
        8.19238901,  6.65294931,  8.75326108,  9.27460362,
6.18153918,
        9.70376591,  8.18452763, 11.71357511, 10.74267823,
7.72965036,
        7.43397439,  6.82788969,  8.22939127,  8.2348197 ,
8.82476593,
       10.51345867,  9.59497196, 11.19988831,  9.37302799,
10.08604571,
        8.45324439,  8.66800526, 10.88798522, 11.84005218,
7.5373072 ,
        9.27974088,  8.34202688, 10.38254851,  8.08254611,
10.65089215,
       10.88811422,  9.45949769, 10.41139418, 10.48701828,
11.13070906,
        9.36162378,  9.80512449, 11.2951419 , 10.12506148,
7.62164999,
       10.35728491,  7.26624166,  9.08538073, 11.25316736,
9.474184  ,
        8.52830245,  7.27348366,  8.04099155,  7.26550445,
7.0395887 ,
        9.89253159,  9.66439031, 10.09193427,  7.97386002,
7.64935223,
       10.96582784, 11.41070407, 11.35895017,  8.58799886,
11.94575588,
        8.74620319, 12.50735682, 10.74835336,  9.64759271,
9.83530772,
```

```
        8.41026344,  9.40644191,  9.52895378, 10.56541533,
8.7013532 ,
        9.7670024 ,  6.40487946,  7.45598827, 11.39853523,
9.15789339,
        7.83546821,  9.96684249, 11.41990891,  7.11742083,
7.24436966,
        9.16285502, 10.47904748,  8.79231556,  5.9082895 ,
10.90752403,
        6.47414787,  9.06052315,  8.30465427, 11.86601677,
10.33232021,
       10.2744247 ,  9.66334047, 10.22898588, 10.40454048,
5.39520268,
       10.55472615,  7.68430725,  6.89129645,  7.5155344 ,
10.83233917,
       13.45300333, 10.70567597, 10.40656464,  9.00164862,
11.15303693,
       10.34722953, 11.12933409, 10.68415957,  8.12100617,
9.02072389,
        7.11136852, 10.95197738,  6.94468787,  8.46111397,
10.22042457,
       10.75440176, 10.57852592, 10.93541514,  7.61916515,
10.40572274,
       10.12603322,  8.70338674,  9.60530608, 10.64631183,
10.84270895,
       10.4761154 ,  9.83340183,  8.4058751 ,  7.39202714,
11.86734229,
        9.95434057, 10.14417203,  7.23838717,  9.15980009,
10.90326059,
       10.1232996 ,  8.33240338,  7.71050908,  7.71583601,
6.94324823,
        8.80850702, 11.94417741,  7.12856184, 10.48419937,
8.14500812,
        7.08252022,  9.96347253,  7.1814626 , 11.22338772,
8.21143832,
       10.05274103,  7.80970804,  9.12979101, 10.54498862,
5.84913363,
       10.15530829,  7.80237104,  6.86143402, 10.72872808,
8.90643711,
        8.65106054,  8.02190199,  8.84069972, 10.49156114,
11.06367457,
       10.15616176,  6.25121954,  8.24247075,  8.29980727,
10.19620653,
        8.88402839, 10.41495808,  8.93468817,  8.57335993,
10.18536169,
        9.81124297,  9.75113181,  8.26582913,  6.84376733,
9.10167901,
       10.06842574,  9.7056066 ,  8.50473395, 10.04221647,
10.21483885,
        9.34117335,  7.78326258,  9.017291  ,  7.2227604 ,
11.44924959,
```

```
         9.43605705,  9.68196466, 12.64732542,  9.74052333,
9.83311448,
        10.95300262,  7.34164072, 11.25166692,  8.92675143,
10.74264592,
         6.24157975,  9.05196494, 10.41558489,  9.79361244,
10.64192093,
         8.98627249,  8.8530722 , 10.36484886,  6.18337987,
11.39209837,
        10.35167091, 11.28579772,  7.99744479,  9.79469752,
11.06964569,
        11.5038125 , 11.03731587,  9.4371465 , 11.78481308,
8.5962956 ,
         9.55800658,  8.97520047,  6.33582782, 10.73275444,
8.49003424,
        10.9768094 ,  7.79014651,  6.07252956,  8.30392393,
9.79660595,
         8.98549151,  9.81318203, 10.27121809, 10.21460751,
9.06363434,
         9.13332595, 10.56526963,  7.84827343, 10.27044904,
11.8371726 ,
        11.24843427,  8.56546181, 11.28604499, 13.66644576,
9.31242148,
        10.43484148,  7.33753379,  8.67965132,  9.11132755,
11.90739446,
        10.21549923,  9.10800624, 10.57926192,  7.8924746 ,
9.70098758,
         8.17759292,  8.98154753,  7.78026489,  7.51876621,
8.04839563,
         9.82187447,  8.87420721,  8.60991464,  8.60260248,
6.66932669,
        10.71413731, 11.21991745,  9.33000812, 10.12575586,
9.84505524,
         7.3608023 , 11.19687773,  7.56371831,  9.70991817,
9.90636044,
        10.13989273,  7.69655559, 10.17666427, 10.279817  ,
10.98951057,
         7.44950848,  8.43525714,  6.17616945,  8.72001621,
7.331762  ,
         8.70683904,  6.94464974,  6.58569497,  8.92298619,
12.16954851,
         9.25512937,  8.60560346, 11.28932034,  8.85188983,
9.25388696,
         8.21442742,  9.80449809,  9.99583257, 10.45040867,
9.60706078,
        10.17462466, 10.51429966,  7.82529387,  9.05475689,
12.13310757,
        10.9884888 ,  8.78452478,  6.88948856,  7.879859  ,
10.05785367,
        10.46400889,  9.71902958,  9.17043486, 10.49160165,
11.09421192,
```

```
        9.57692216,  8.52173342,  9.40491815, 12.82371082,
10.08395123,
        8.56473319, 10.82990608,  8.68423384,  9.95981653,
9.03931389,
       11.07868643,  8.31829561,  8.64065327, 10.63805058,
10.47297313,
        8.61394017,  9.59791303, 10.35100503, 10.59256415,
11.11582595,
        7.38744794,  6.31940745, 10.59914591, 11.30354105,
7.44562712,
        6.77334464,  8.44018591, 11.41705426, 10.37329033,
10.53151791,
        9.68722133, 10.49898487,  9.79845754,  7.88256664,
9.79014746,
        9.04614691,  9.89083524,  7.90206254,  9.71464629,
9.13770368,
        7.32393506,  6.76416334,  9.47587999,  8.83817917,
9.78786456,
       10.5607793 ,  7.21366218, 10.84493143,  7.98345232,
11.5385508 ,
       11.00182679, 11.69220755,  7.86370362, 10.22128841,
7.20173378,
        9.1045296 ,  9.48814039,  9.90449746, 11.53754504,
7.42195675,
       10.64492281, 12.68882666,  9.43218963,  7.66667689,
6.78838922,
       12.4270438 ,  9.84926179,  9.62905675,  9.6599187 ,
10.10605277,
        7.39275033, 10.82070701,  8.21580076, 10.44431111,
10.00660562,
        7.86385714,  9.45216302,  7.15169274, 11.78612085,
8.64994417,
       10.25387361,  7.53565603, 11.30042491, 12.60623583,
9.4451939 ,
        9.80049785,  7.65701351, 10.43651856,  9.21559213,
12.0292347 ,
        9.43674673,  9.89122282, 10.48829024, 12.10872768,
10.57342177,
        9.26207825,  8.7070252 , 10.4847813 , 10.55758463,
6.438221  ,
        8.60663443, 10.02136949, 10.58786163, 10.61035235,
11.74547772,
        8.67526264, 10.9107008 ,  9.23390493, 11.01118909,
8.26863737,
        9.13369183, 11.20960106,  9.83087258,  9.7255109 ,
10.56641406,
        7.85824889,  6.91504224, 11.3794128 ,  8.1110583 ,
6.97298541,
        7.20301438, 11.34435301,  6.20766205,  9.9584154 ,
10.14670555,
```

```
        7.84878626,  6.92609399,  9.80502191,  7.03652758,
4.42284616,
        8.62588186, 10.48376689,  9.83515223,  9.24651961,
11.09021142,
        7.42028521, 12.07297248,  9.04888704,  9.98006909,
9.55841692,
       11.13792986,  7.73562252, 13.14163828,  9.76977286,
8.80393599,
        8.53386297, 10.24823378,  7.40114539, 12.19405951,
8.18516302,
        9.29258099,  7.74168514,  7.73549892,  8.2528931 ,
9.99541923,
        9.79730046, 11.34944567,  8.99871751, 10.60184659,
12.07459996,
       11.11191066, 10.20169343,  9.5613815 ,  7.9860126 ,
10.71335546,
        7.15193829,  9.52008642, 10.50385894, 10.01161046,
7.10639028,
        7.30774585,  8.76689354,  6.52565374,  9.98325369,
9.69514382,
        8.57582212, 10.83810379,  7.14794768, 10.95891474,
10.02089771,
       10.31402367, 10.55622121, 11.64856169,  7.66284484,
6.19957781,
        9.9314708 ,  9.53077317, 12.03847583, 12.17318536,
9.40968753,
        7.97639201, 10.36443176, 10.45410955, 11.8195622 ,
10.05107372,
        9.50373076,  9.83150774, 10.15623073,  9.78037501,
7.23501753,
       10.73161815,  9.26914037,  9.65031094,  8.38114022,
8.20401876,
        8.41577489,  7.84033609,  8.83278262, 10.29171599,
5.75625783,
       10.03618685, 10.93648841, 11.35470672,  9.13204143,
11.59375183,
        7.64548049, 10.07580442,  8.60989182, 10.98874428,
8.0428049 ,
       10.03487882,  8.32082381,  8.95430156, 10.44696989,
9.78485152,
       10.15130267, 14.4138759 ,  8.00295431, 10.69683743,
10.13667149,
        7.75602942,  8.93551456,  9.62058152, 10.23768553,
10.13280524,
       10.63022151,  9.88283702, 10.7994296 ,  9.4203121 ,
13.76350741,
        7.36211081,  8.89477138,  8.01298095,  9.71964431,
7.76962104,
        9.87954256,  9.23000838,  8.24480177,  9.75707829,
8.3401242 ,
```

```
        7.69248093, 10.45660064,  9.99540552, 10.97136942,
7.02292937,
       11.12665446,  6.65269111,  6.06923912,  6.79437197,
9.0245297 ,
        9.55960732, 10.89058021,  9.95356288,  8.31876242,
11.18195259,
        6.348208  ,  8.76685449,  9.78380457,  7.72182384,
10.00876199,
       10.4942607 ,  8.6771521 ,  8.0227051 ,  6.6482521 ,
10.24825341,
        8.48019074, 11.40512333,  9.85931689,  9.84956847,
11.07700143,
        9.7694319 ,  9.79781417,  7.9148079 ,  8.9365744 ,
11.02888406,
        7.4890541 , 11.34427268,  9.42680753,  8.5032721 ,
10.06622936,
       11.27639765,  8.30154971,  7.52765032, 11.98422714,
9.89235363,
       10.04140114, 11.29788061,  7.28869778,  9.95390067,
10.47080466,
       10.44322463,  8.1687981 , 10.10847573,  9.15318928,
9.87791837,
        9.67622494,  7.13990858,  8.68253386,  7.80925338,
8.66997323,
        8.88796902,  5.31941982, 12.35698794,  8.91224238,
10.09293777,
        9.7776893 , 10.26613217, 10.70777213,  9.6365558 ,
9.65280679,
        9.50186889,  7.33833511,  8.98778627, 10.40376118,
8.96763422,
        8.84534374,  7.69171042,  8.65738684,  9.48233168,
7.09880127,
       11.13223266,  7.61593531,  8.14033072, 10.48392036,
11.16476491,
        9.96841797,  7.73865916, 11.17650829, 10.40883863,
10.46197262,
        8.11909864,  8.62403066,  6.45601655,  6.93413359,
10.41912181,
        9.81487216, 13.82913236, 12.14365228, 10.8934645 ,
6.22946403,
        7.6694437 ,  9.00581268,  6.79076954,  8.69901346,
12.76974484,
       11.7950662 ,  9.67397262, 10.57059534,  9.68484004,
9.41309062,
        7.73252651, 10.49428517, 10.54304916,  9.25752383,
12.87262306,
        9.82741707, 10.96874555, 10.70970551, 11.50597744,
8.17973589,
       10.48714192, 11.67314078, 11.08209236,  8.81031906,
8.75431974,
```

```
        7.11968611, 10.28861782,  9.65858199, 11.20486501,
9.32735489,
       10.47293087,  9.46466381,  8.1308579 , 11.93023936,
10.15093859,
        6.30849186,  8.20050422,  6.3399775 , 10.19329142,
7.24330273,
        9.45478237, 10.42536125,  8.03663773,  7.44782382,
7.80111459,
       11.11513908,  7.36363297,  9.38436282,  7.09671399,
11.39848098,
        6.28933055,  9.54982001,  9.06025363, 10.57576027,
10.65398892,
        9.1615044 , 10.79538712, 10.83425502, 11.80786647,
8.35091735,
       10.33582146,  8.69937146,  9.40049662,  9.40571276,
6.38668674,
       10.15947267,  9.24185953,  7.05134292,  7.18077445,
10.22435515,
        7.38795289,  9.72940018,  9.29340394,  8.2925775 ,
9.16583171,
        8.82154456,  8.64008073,  8.19654932, 10.66415937,
8.73075719,
       11.0064091 ,  9.85387389,  7.21458338,  9.74824301,
11.2120979 ,
        7.04560644,  6.85749941,  7.44315695,  6.56619483,
10.54307414,
        8.7146793 , 10.40592815,  8.66643278, 10.47332785,
8.84477047,
        7.05274494,  9.08233825, 11.25829949,  9.58599069,
9.59220694,
        8.95440262,  7.13375312, 11.14042852,  7.36599452,
9.98897125,
       11.88140739, 11.00817347,  7.62134802,  8.1482848 ,
8.39504862,
        9.08400894,  9.33882675,  7.96726906,  9.91602019,
10.34536517,
        9.79298111, 10.24712188, 11.44898727,  9.71833965,
11.19933015,
        9.64108247,  9.46465453,  9.75535007,  8.67190929,
9.99091481,
        9.09869232,  7.88893657,  8.90956615,  6.43993534,
6.87152784,
        8.89845174,  9.1585959 , 10.9247976 ,  7.7092857 ,
8.88064446,
        8.28877327,  8.56419113,  9.92026262, 10.84805718,
10.89178546,
        9.3761516 ,  9.61810349, 10.88244183, 10.4188466 ,
9.22835284,
        6.37493317,  8.8979983 , 10.51568586,  6.7706697 ,
7.94360523,
```

```
        9.46775408, 10.48271815,  9.78198745,  6.86493145,
7.08634063,
        8.47572268, 10.05366792,  9.01848706,  7.86438998,
11.03746355,
        6.08987441])
```

1. Measure the performance using metrics

```
from sklearn.metrics import r2_score
acc = r2_score(y_test,y_pred)

acc

0.48342896299561555

df.head()

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

   Shell weight  Rings  Sex_F  Sex_I  Sex_M
0         0.150   15.0      0      0      1
1         0.070    7.0      0      0      1
2         0.210    9.0      1      0      0
3         0.155   10.0      0      0      1
4         0.055    7.0      0      1      0

MLR.predict([[0.455,0.365,0.095,0.5140,0.2245,0.101,0.150,0,0,1]])

array([9.79146941])
```