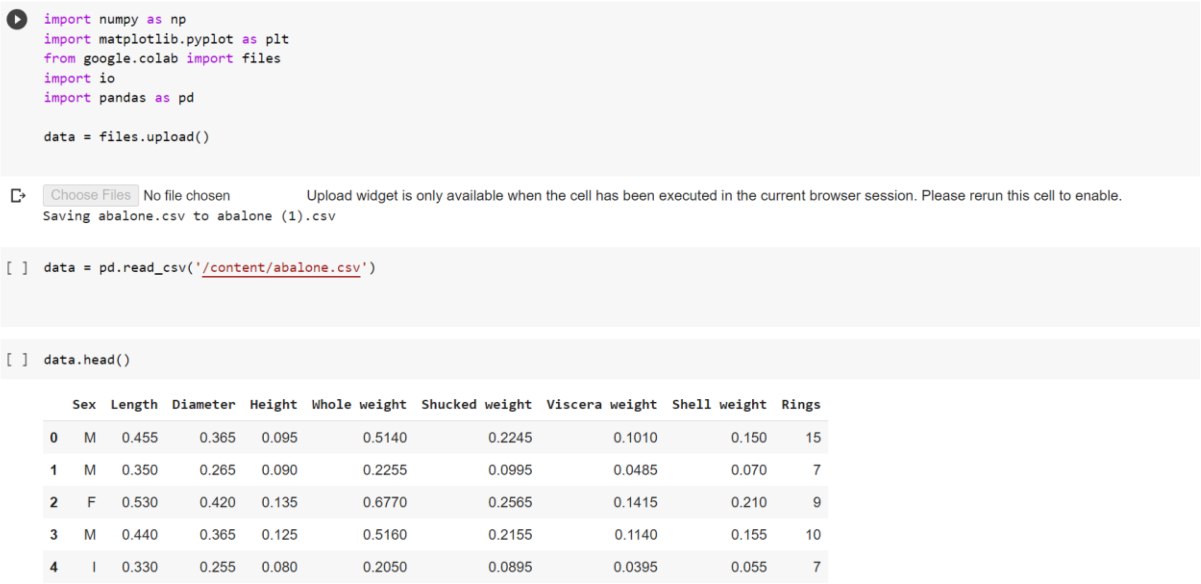
ASSIGNMENT 3

Student Name	R Abhirami
Student Roll Number	211519104004
Maximum Marks	10 Marks

Question 1&2:



```
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files
import io
import pandas as pd

data = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving abalone.csv to abalone (1).csv
```

```
[ ] data = pd.read_csv('/content/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
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Question 3:

Perform Below Visualizations.

· Univariate Analysis

· Bi-Variate Analysis

· Multivariate Analysis

Univariate Analysis

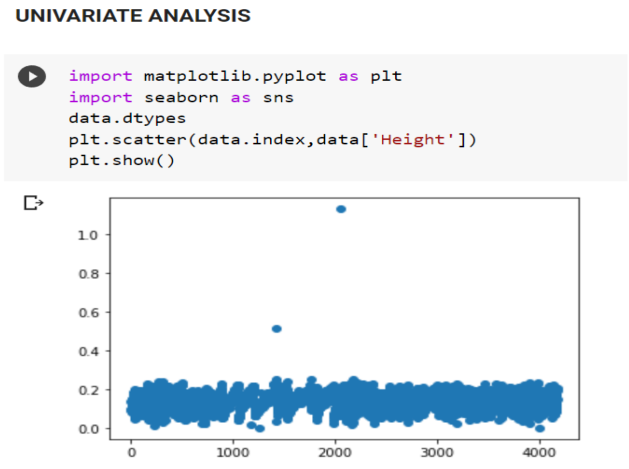
```
import matplotlib.pyplot as plt
import seaborn as sns
data.dtypes
plt.scatter(data.index,data['Height'])
plt.show()
```

Bi-Variate Analysis

```
import seaborn as sns
sns.boxplot(x='Sex',y='Height',data=data)
plt.show()
```

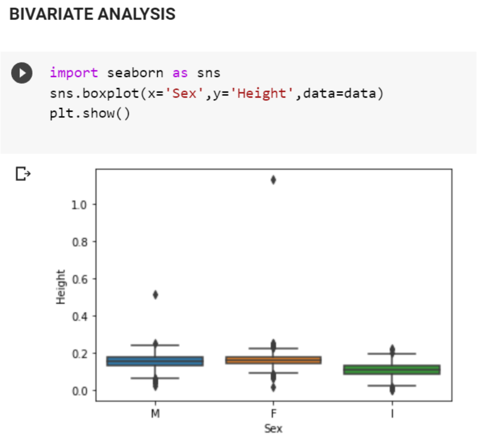
Multi-Variate Analysis

```
import seaborn as sns
sns.pairplot(data,hue="Rings",height=3)
plt.show()
```



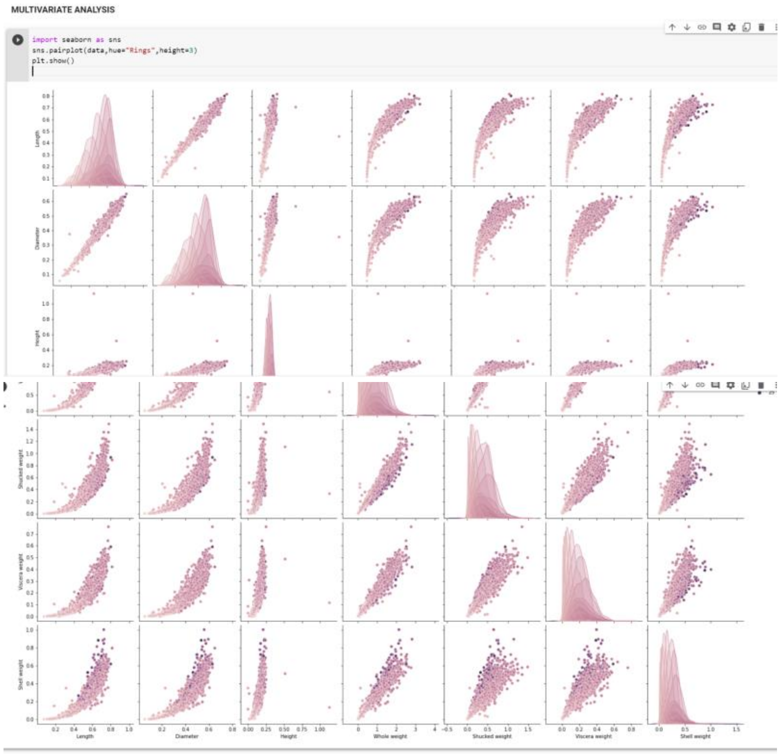
UNIVARIATE ANALYSIS

```
import matplotlib.pyplot as plt
import seaborn as sns
data.dtypes
plt.scatter(data.index,data['Height'])
plt.show()
```



BIVARIATE ANALYSIS

```
import seaborn as sns
sns.boxplot(x='Sex',y='Height',data=data)
plt.show()
```

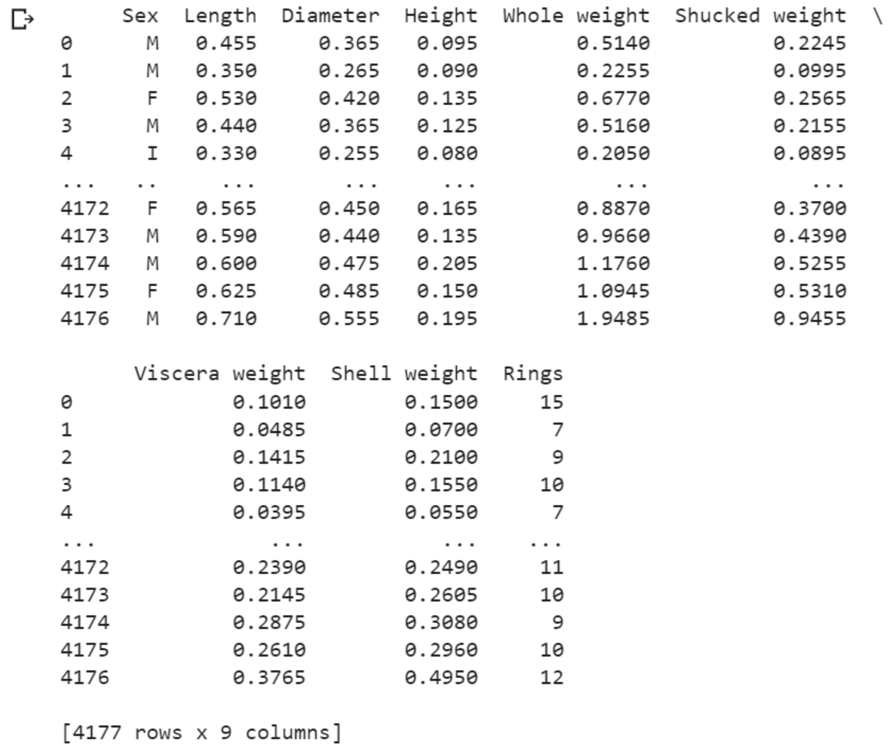


MULTIVARIATE ANALYSIS

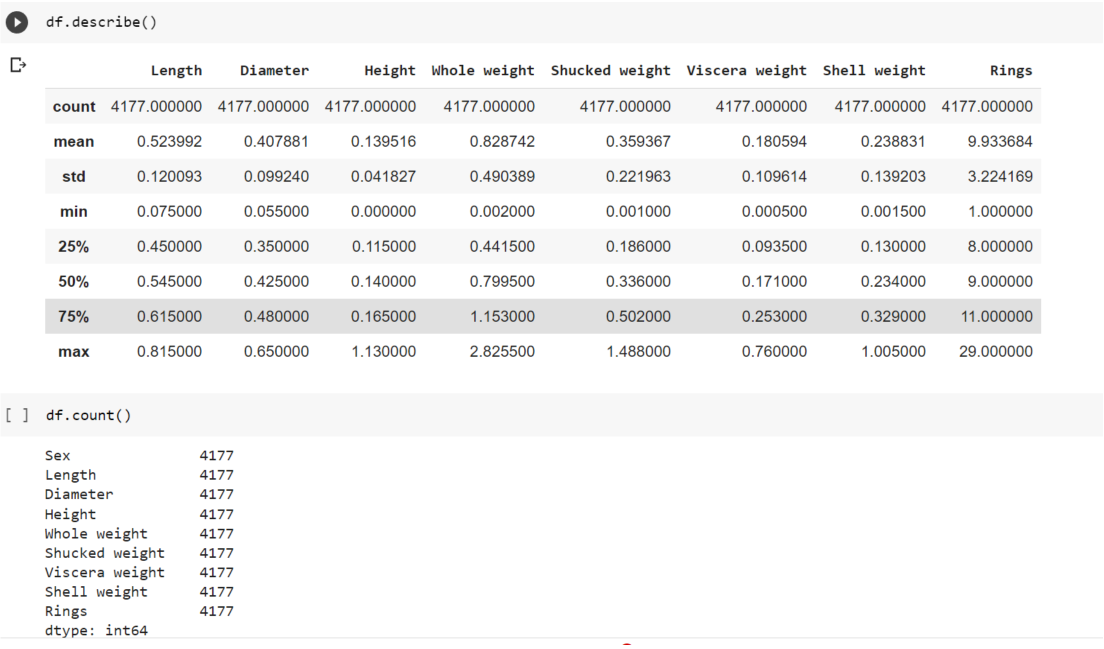
```
import seaborn as sns
sns.pairplot(data,hue="Rings",height=3)
plt.show()
```

Question 4. Perform descriptive statistics on the dataset.

```
import pandas as pd
import numpy as np
df = pd.DataFrame(data)
print (df)
df.describe()
df.count()
```



```
       Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0        M   0.455     0.365   0.095        0.5140          0.2245
1        M   0.350     0.265   0.090        0.2255          0.0995
2        F   0.530     0.420   0.135        0.6770          0.2565
3        M   0.440     0.365   0.125        0.5160          0.2155
4        I   0.330     0.255   0.080        0.2050          0.0895
...     ..     ...       ...     ...           ...             ...
4172     F   0.565     0.450   0.165        0.8870          0.3700
4173     M   0.590     0.440   0.135        0.9660          0.4390
4174     M   0.600     0.475   0.205        1.1760          0.5255
4175     F   0.625     0.485   0.150        1.0945          0.5310
4176     M   0.710     0.555   0.195        1.9485          0.9455

      Viscera weight  Shell weight  Rings
0             0.1010        0.1500     15
1             0.0485        0.0700      7
2             0.1415        0.2100      9
3             0.1140        0.1550     10
4             0.0395        0.0550      7
...              ...           ...    ...
4172          0.2390        0.2490     11
4173          0.2145        0.2605     10
4174          0.2875        0.3080      9
4175          0.2610        0.2960     10
4176          0.3765        0.4950     12

[4177 rows x 9 columns]
```



```
df.describe()
```

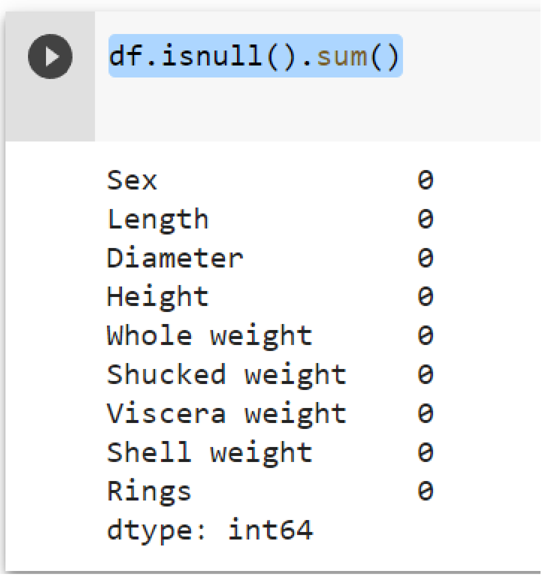
	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	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

```
[ ] df.count()
```

```
Sex               4177
Length            4177
Diameter          4177
Height            4177
Whole weight      4177
Shucked weight    4177
Viscera weight    4177
Shell weight      4177
Rings             4177
dtype: int64
```

Question 5. Check for Missing values and deal with them.

```
df.isnull().sum()
```

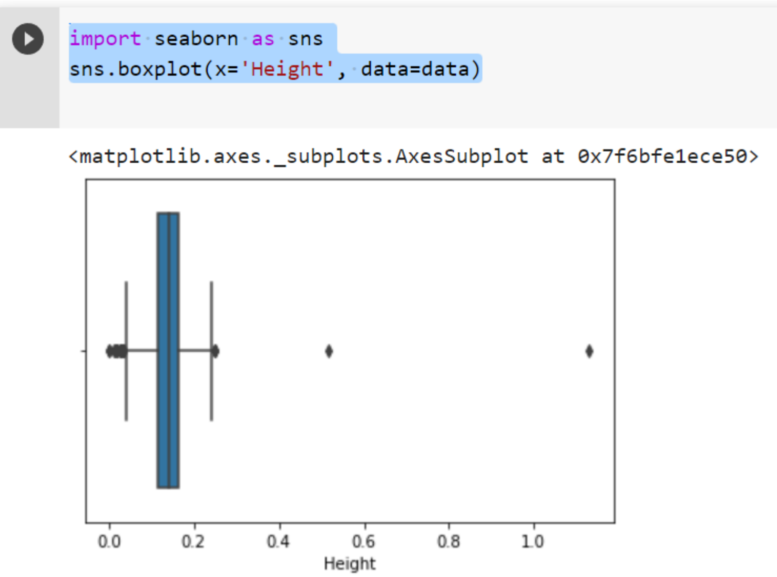


```
df.isnull().sum()
```

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

Question 6 Find the outliers and replace them outliers

```
import seaborn as sns
sns.boxplot(x='Height', data=data)
```

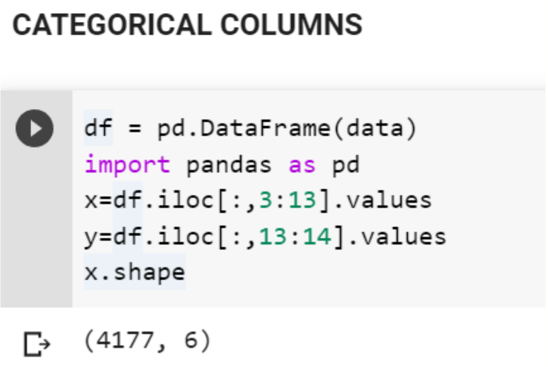


```
import seaborn as sns
sns.boxplot(x='Height', data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6bfe1ece50>
```

Question 7. Check for Categorical columns and perform encoding.

```
df = pd.DataFrame(data)
import pandas as pd
x=df.iloc[:,3:13].values
y=df.iloc[:,13:14].values
x.shape
```



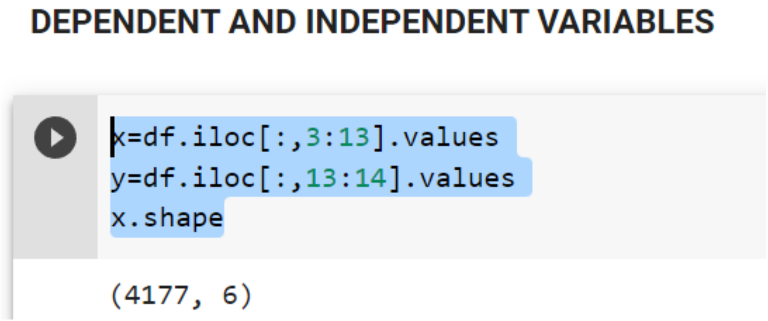
CATEGORICAL COLUMNS

```
df = pd.DataFrame(data)
import pandas as pd
x=df.iloc[:,3:13].values
y=df.iloc[:,13:14].values
x.shape
```

```
(4177, 6)
```

Question 8 Split the data into dependent and independent variables.

```
x=df.iloc[:,3:13].values
y=df.iloc[:,13:14].values
x.shape
```



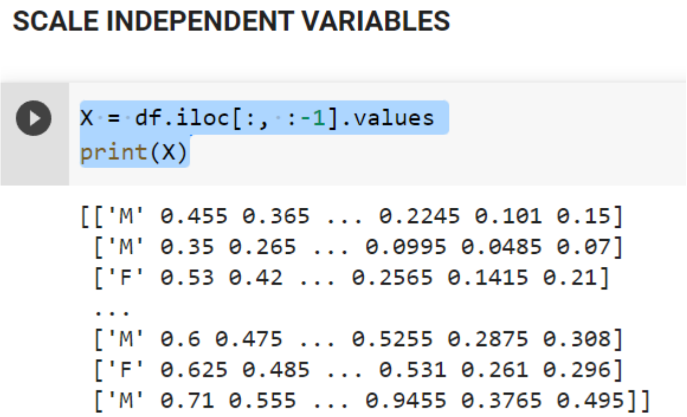
DEPENDENT AND INDEPENDENT VARIABLES

```
x=df.iloc[:,3:13].values
y=df.iloc[:,13:14].values
x.shape
```

```
(4177, 6)
```

Question 9. Scale the independent variables

```
X = df.iloc[:, :-1].values
print(X)
```



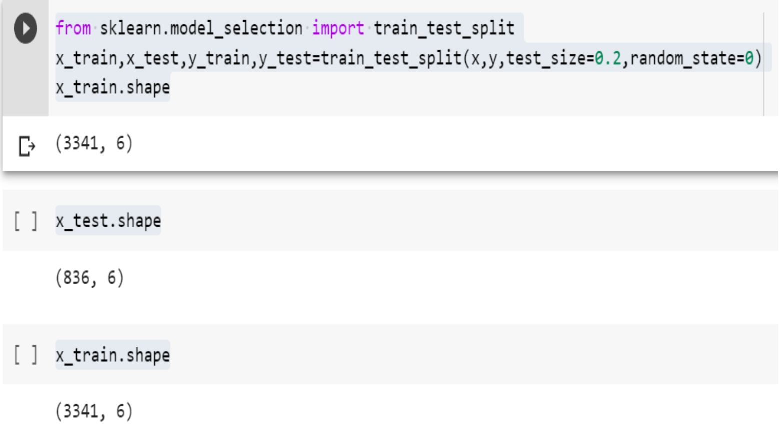
SCALE INDEPENDENT VARIABLES

```
X = df.iloc[:, :-1].values
print(X)
```

```
[['M' 0.455 0.365 ... 0.2245 0.101 0.15]
 ['M' 0.35 0.265 ... 0.0995 0.0485 0.07]
 ['F' 0.53 0.42 ... 0.2565 0.1415 0.21]
 ...
 ['M' 0.6 0.475 ... 0.5255 0.2875 0.308]
 ['F' 0.625 0.485 ... 0.531 0.261 0.296]
 ['M' 0.71 0.555 ... 0.9455 0.3765 0.495]]
```

Question 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,random_st
ate=0)
x_train.shape
x_test.shape
x_train.shape
```



```
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,random_state=0)
x_train.shape
```

```
(3341, 6)
```

```
[ ] x_test.shape
```

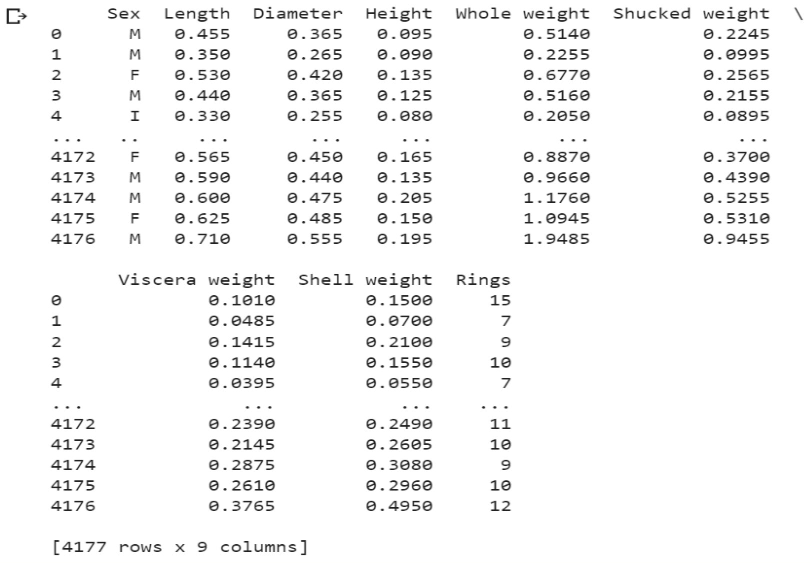
```
(836, 6)
```

```
[ ] x_train.shape
```

```
(3341, 6)
```

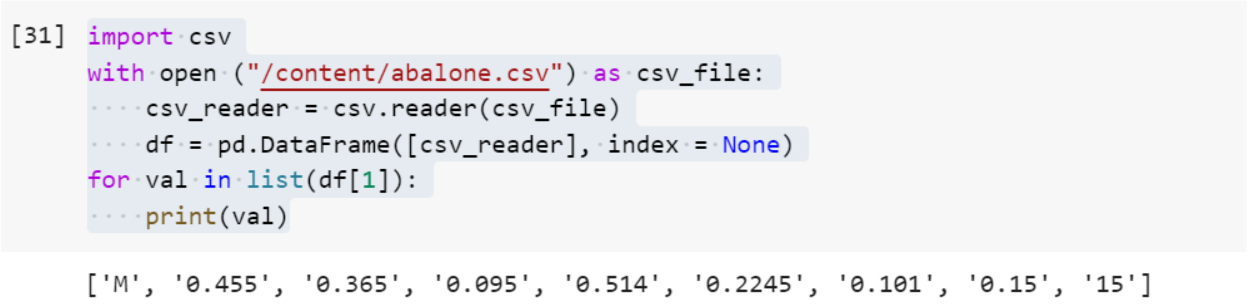
Question 11. Build the Model

```
my_dict=pd.read_csv("/content/abalone.csv")
df = pd.DataFrame(my_dict)
print(df)
```



```
       Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0        M   0.455     0.365   0.095        0.5140          0.2245   
1        M   0.350     0.265   0.090        0.2255          0.0995   
2        F   0.530     0.420   0.135        0.6770          0.2565   
3        M   0.440     0.365   0.125        0.5160          0.2155   
4        I   0.330     0.255   0.080        0.2050          0.0895   
...     ..     ...       ...     ...           ...             ...   
4172     F   0.565     0.450   0.165        0.8870          0.3700   
4173     M   0.590     0.440   0.135        0.9660          0.4390   
4174     M   0.600     0.475   0.205        1.1760          0.5255   
4175     F   0.625     0.485   0.150        1.0945          0.5310   
4176     M   0.710     0.555   0.195        1.9485          0.9455   

      Viscera weight  Shell weight  Rings  
0             0.1010        0.1500     15  
1             0.0485        0.0700      7  
2             0.1415        0.2100      9  
3             0.1140        0.1550     10  
4             0.0395        0.0550      7  
...              ...           ...    ...  
4172          0.2390        0.2490     11  
4173          0.2145        0.2605     10  
4174          0.2875        0.3080      9  
4175          0.2610        0.2960     10  
4176          0.3765        0.4950     12  

[4177 rows x 9 columns]
```

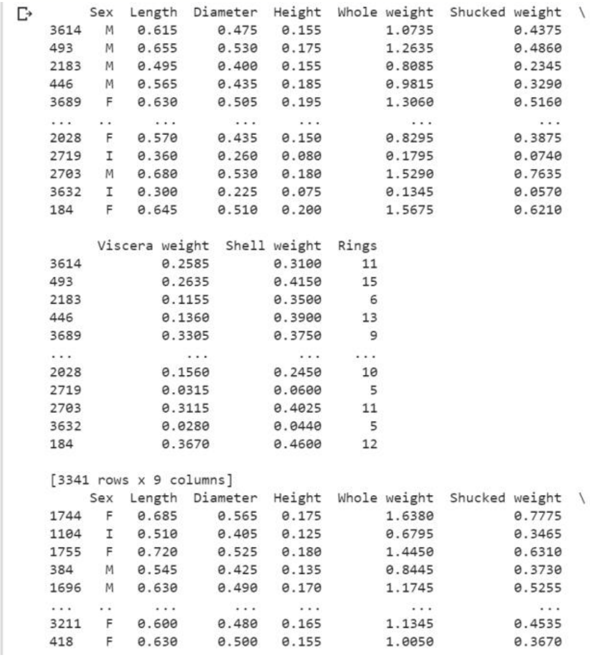
```
import csv
with open ("/content/abalone.csv") as csv_file:
    csv_reader = csv.reader(csv_file)
    df = pd.DataFrame([csv_reader], index = None)
for val in list(df[1]):
    print(val)
```



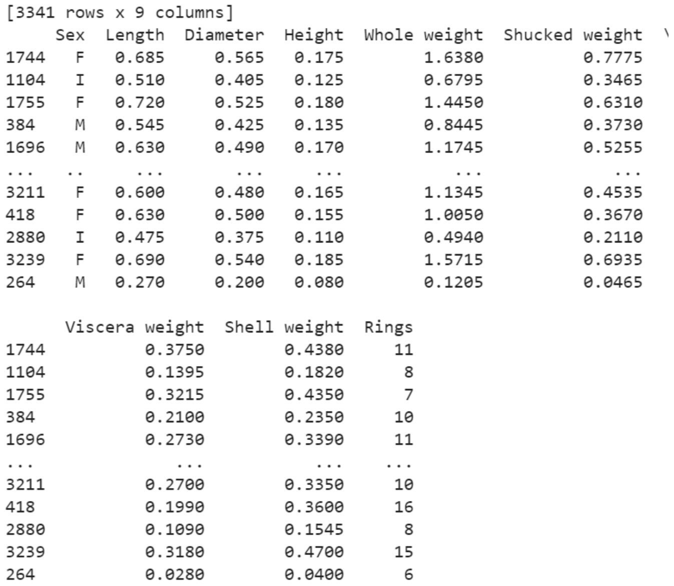
```
[31] import csv
     with open ("/content/abalone.csv") as csv_file:
         csv_reader = csv.reader(csv_file)
         df = pd.DataFrame([csv_reader], index = None)
     for val in list(df[1]):
         print(val)

     ['M', '0.455', '0.365', '0.095', '0.514', '0.2245', '0.101', '0.15', '15']
```

Question 12 & 13 Train and Test the Model

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, test_size=0.2)
print(train)
print(test)
```



```
      Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
3614    M   0.615     0.475   0.155        1.0735          0.4375
493     M   0.655     0.530   0.175        1.2635          0.4860
2183    M   0.495     0.400   0.155        0.8085          0.2345
446     M   0.565     0.435   0.185        0.9815          0.3290
3689    F   0.630     0.505   0.195        1.3060          0.5160
...    ..     ...       ...     ...           ...             ...
2028    F   0.570     0.435   0.150        0.8295          0.3875
2719    I   0.360     0.260   0.080        0.1795          0.0740
2703    M   0.680     0.530   0.180        1.5290          0.7635
3632    I   0.300     0.225   0.075        0.1345          0.0570
184     F   0.645     0.510   0.200        1.5675          0.6210

       Viscera weight  Shell weight  Rings
3614           0.2585        0.3100     11
493            0.2635        0.4150     15
2183           0.1155        0.3500      6
446            0.1360        0.3900     13
3689           0.3305        0.3750      9
...               ...           ...    ...
2028           0.1560        0.2450     10
2719           0.0315        0.0600      5
2703           0.3115        0.4025     11
3632           0.0280        0.0440      5
184            0.3670        0.4600     12

[3341 rows x 9 columns]
      Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
1744    F   0.685     0.565   0.175        1.6380          0.7775
1104    I   0.510     0.405   0.125        0.6795          0.3465
1755    F   0.720     0.525   0.180        1.4450          0.6310
384     M   0.545     0.425   0.135        0.8445          0.3730
1696    M   0.630     0.490   0.170        1.1745          0.5255
...    ..     ...       ...     ...           ...             ...
3211    F   0.600     0.480   0.165        1.1345          0.4535
418     F   0.630     0.500   0.155        1.0050          0.3670
```



```
[3341 rows x 9 columns]
     Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
1744   F   0.685     0.565   0.175        1.6380          0.7775
1104   I   0.510     0.405   0.125        0.6795          0.3465
1755   F   0.720     0.525   0.180        1.4450          0.6310
384    M   0.545     0.425   0.135        0.8445          0.3730
1696   M   0.630     0.490   0.170        1.1745          0.5255
...   ..     ...       ...     ...           ...             ...
3211   F   0.600     0.480   0.165        1.1345          0.4535
418    F   0.630     0.500   0.155        1.0050          0.3670
2880   I   0.475     0.375   0.110        0.4940          0.2110
3239   F   0.690     0.540   0.185        1.5715          0.6935
264    M   0.270     0.200   0.080        0.1205          0.0465

      Viscera weight  Shell weight  Rings
1744          0.3750        0.4380     11
1104          0.1395        0.1820      8
1755          0.3215        0.4350      7
384           0.2100        0.2350     10
1696          0.2730        0.3390     11
...              ...           ...    ...
3211          0.2700        0.3350     10
418           0.1990        0.3600     16
2880          0.1090        0.1545      8
3239          0.3180        0.4700     15
264           0.0280        0.0400      6
```

Question 14. Measure the performance using Metrics.

```
pd.crosstab(Y_test,y_predict)
```

```
print(classification_report(Y_test,y_predict))
```