

Assignment -3

Assignment Date	30 September 2022
Student Name	Anjana Deivasigamani
Student Roll Number	2116190701020
Maximum Marks	2 Marks

ntp_assignment3.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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

[ ] data=pd.read_csv("/content/abalone.csv")
```

Descriptive Statistics

```
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

```
[ ] 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

ntp_assignment3.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
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
```

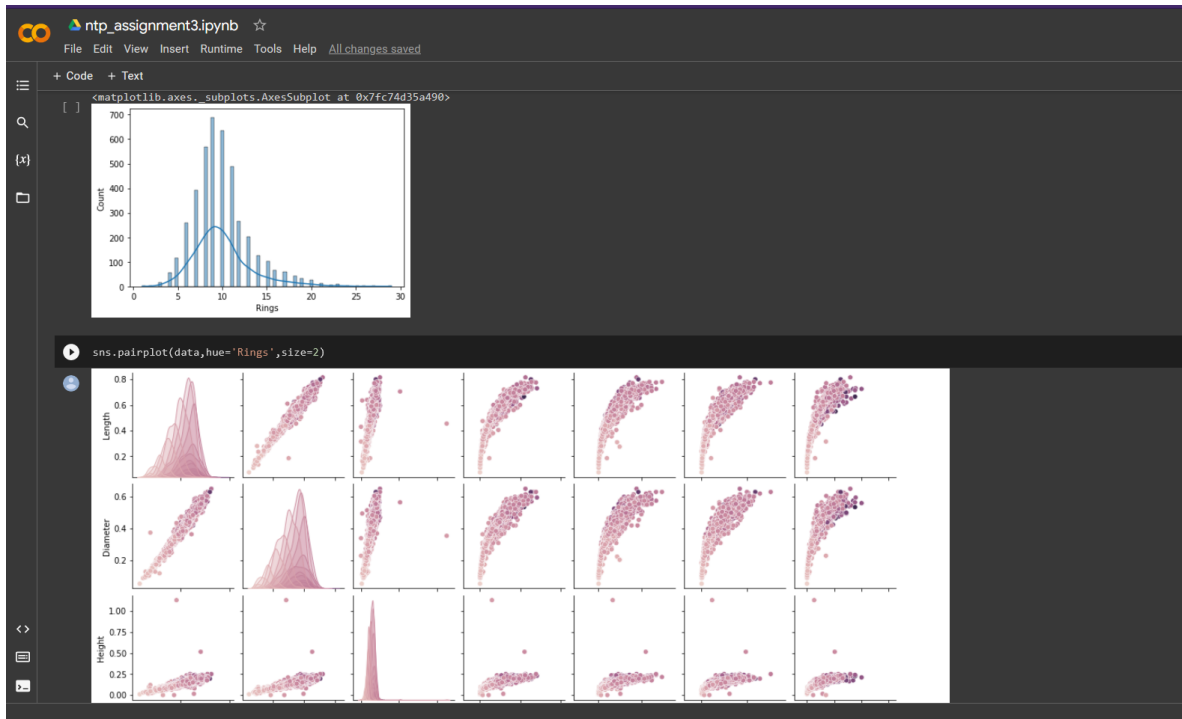
Checking Missing Values

```
[ ] data['Length'] = data['Length'].fillna(data['Length'].mean())
data['Diameter'] = data['Diameter'].fillna(data['Diameter'].mean())
data['Height'] = data['Height'].fillna(data['Height'].mean())
data['Whole weight'] = data['Whole weight'].fillna(data['Whole weight'].mean())
data['Shucked weight'] = data['Shucked weight'].fillna(data['Shucked weight'].mean())
data['Viscera weight'] = data['Viscera weight'].fillna(data['Viscera weight'].mean())

data['Shell weight'] = data['Shell weight'].fillna(data['Shell weight'].mean())
data['Rings'] = data['Rings'].fillna(data['Rings'].mean())

data
```

	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	0.1500	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7



ntp_assignment3.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Length Diameter Height Whole weight Shucked weight Viscera weight Shell weight

Splitting the data into independent and dependent variables

```
X=data.iloc[:, :-1].values
y=data.iloc[:, -1].values
```

Checking for categorical data and performing encoding

```
from sklearn.preprocessing import LabelEncoder
labelencoder_X_1 = LabelEncoder()
y = labelencoder_X_1.fit_transform(y)
data['Sex']=labelencoder_X_1.fit_transform(data['Sex'])
data['Sex'].unique()
```

```
array([2, 0, 1])
```

```
[ ] data
```

	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	0.1500	15
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	2	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7
...
4172	0	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11

ntp_assignment3.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

4177 rows × 9 columns

Splitting the data into training set and test set

```
[ ] 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,random_state=0)
```

Training the model

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X_train, y_train)
```

LinearRegression()

Testing the model

```
[ ] y_pred = reg.predict(X_test)
    np.set_printoptions(precision=2)
    print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[12.12 12. ]
 [ 8.66  7. ]
 [ 9.35 10. ]
 ...
 [ 8.01  8. ]
 [17.83 17. ]
 [10.77 14. ]]
```