In [1]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [2]:
```
data=pd.read_csv("abalone.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	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

In [4]:
```
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
```

In [5]:
```
data.isnull().sum()/data.shape[0]
```

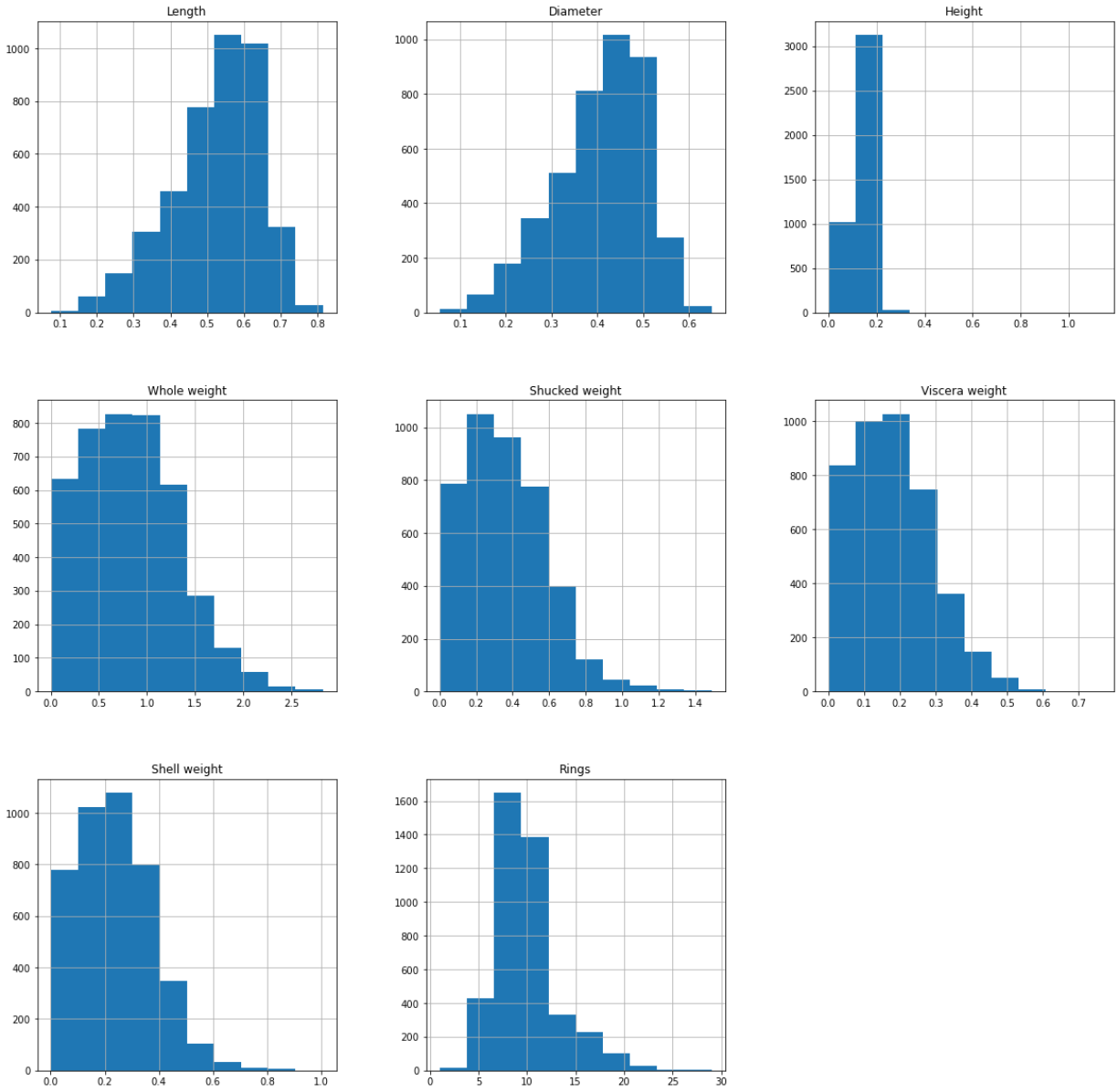
Out[5]:
```
Sex               0.0
Length            0.0
Diameter          0.0
Height            0.0
Whole weight      0.0
Shucked weight    0.0
Viscera weight    0.0
Shell weight      0.0
Rings             0.0
dtype: float64
```

In [6]:
```
data.shape
```

Out[6]:
```
(4177, 9)
```

In [7]:
```
data.hist(figsize=(20,20))
```

Out[7]:
```
array([[<AxesSubplot:title={'center':'Length'}>,
        <AxesSubplot:title={'center':'Diameter'}>,
        <AxesSubplot:title={'center':'Height'}>],
       [<AxesSubplot:title={'center':'Whole weight'}>,
        <AxesSubplot:title={'center':'Shucked weight'}>,
        <AxesSubplot:title={'center':'Viscera weight'}>],
       [<AxesSubplot:title={'center':'Shell weight'}>,
        <AxesSubplot:title={'center':'Rings'}>, <AxesSubplot:>]],
      dtype=object)
```



In [8]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [9]:
```
data['Sex']=le.fit_transform(data['Sex'])
```

In [10]:
```
data
```

Out[10]:

	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
4173	2	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	2	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	0	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	2	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

4177 rows × 9 columns

In [11]:
```
data.describe()
```

Out[11]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	1.052909	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594
std	0.822240	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614
min	0.000000	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000
max	2.000000	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000

In [12]:
```
X = data.drop('Rings',1)
y = data['Rings']
X_train,X_test,y_train,y_test = train_test_split(X,y,random_state=2)
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_7904/2454726055.py:1: FutureWarning: I
n a future version of pandas all arguments of DataFrame.drop except for the arg
ument 'labels' will be keyword-only
  X = data.drop('Rings',1)
```

In [13]:
```
lm=LinearRegression()
```

In [14]:
```
lm.fit(data,y)
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
lm.coef_
```

Out[15]:
```
array([ 3.86667275e-16, -5.21804822e-15, -1.71737624e-14, -2.68882139e-16,
        2.98372438e-15,  1.34575510e-14, -6.80358547e-15, -2.94902991e-14,
        1.00000000e+00])
```

In [16]:
```
lm.intercept_
```

Out[16]:
```
0.0
```

In [17]:
```
y.mean()
```

Out[17]:
```
9.933684462532918
```

In [18]:
```
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
```

In [19]:
```
ybar=lm.predict(data)
```

In [20]:
```
r2_score(y,ybar)
```

Out[20]:
```
1.0
```

In [21]:
```
print(f'The Mean Squared Error of our model is: {mean_squared_error(y,ybar)} \nTh
```

```
The Mean Squared Error of our model is: 8.384751475437836e-30
The Mean Abs Error of the model is : 2.0296588605799904e-15
```

In [ ]:
```

```