In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv(r"S:\DATA SCIENCE TCA ML\HeightWeightAge.csv")
data
```

Out[2]:

	Age	Height	Weight	Bmi	BmiClass
0	61	1.85	109.30	31.935720	Obese Class 1
1	60	1.71	79.02	27.023700	Overweight
2	60	1.55	74.70	31.092612	Obese Class 1
3	60	1.46	35.90	16.841809	Underweight
4	60	1.58	97.10	38.896010	Obese Class 2
...	...	...	...	...	...
736	34	1.86	95.70	27.662157	Overweight
737	44	1.91	106.90	29.302925	Overweight
738	25	1.82	88.40	26.687598	Overweight
739	35	1.88	98.50	27.868945	Overweight
740	45	1.93	109.90	29.504148	Overweight

741 rows × 5 columns

In [3]:
```
data.shape
```

Out[3]:
```
(741, 5)
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
Age          0
Height       0
Weight       0
Bmi          0
BmiClass     0
dtype: int64
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 741 entries, 0 to 740
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Age        741 non-null    int64  
 1   Height     741 non-null    float64
 2   Weight     741 non-null    float64
 3   Bmi        741 non-null    float64
 4   BmiClass   741 non-null    object 
dtypes: float64(3), int64(1), object(1)
memory usage: 29.1+ KB
```

In [6]:
```
data['BmiClass'].unique()
```

Out[6]:
```
array(['Obese Class 1', 'Overweight', 'Underweight', 'Obese Class 2',
       'Obese Class 3', 'Normal Weight'], dtype=object)
```

In [7]:
```
data['BmiClass'].value_counts()
```

Out[7]:
```
BmiClass
Normal Weight    342
Overweight       166
Underweight       96
Obese Class 3     62
Obese Class 2     55
Obese Class 1     20
Name: count, dtype: int64
```

In [9]:
```
data['BmiClass'].replace('Normal Weight',0,inplace=True)
data['BmiClass'].replace('Overweight',1,inplace=True)
data['BmiClass'].replace('Underweight',2,inplace=True)
```

```
data['BmiClass'].replace('Obese Class 3',3,inplace=True)
data['BmiClass'].replace('Obese Class 2',4,inplace=True)
data['BmiClass'].replace('Obese Class 1',5,inplace=True)
```

In [18]:
```
data
```

Out[18]:

	Age	Height	Weight	Bmi	BmiClass
0	61	1.85	109.30	31.935720	5
1	60	1.71	79.02	27.023700	1
2	60	1.55	74.70	31.092612	5
3	60	1.46	35.90	16.841809	2
4	60	1.58	97.10	38.896010	4
...	...	...	...	...	...
736	34	1.86	95.70	27.662157	1
737	44	1.91	106.90	29.302925	1
738	25	1.82	88.40	26.687598	1
739	35	1.88	98.50	27.868945	1
740	45	1.93	109.90	29.504148	1

741 rows × 5 columns

In [21]:
```
q1 = data['Weight'].quantile(0.25)
q3 = data['Weight'].quantile(0.75)
IQR = q3 - q1

min_range = q1 - 1.5 * IQR
max_range = q3 + 1.5 * IQR

data = data[data['Weight'] <= max_range]
```

In [30]:
```
z_score = (data['Bmi'] - data['Bmi'].mean()) / data['Bmi'].std()
data['z_score'] = z_score
data = data[data['z_score'] < 3]
```

In [31]:
```
data.describe()
```

Out[31]:

	Age	Height	Weight	Bmi	BmiClass	z_score
count	617.000000	617.000000	617.000000	617.000000	617.000000	6.170000e+02
mean	31.593193	1.703121	67.227326	22.871957	0.685575	8.752228e-16
std	11.647898	0.082761	15.590546	3.952428	0.974514	1.000000e+00
min	15.000000	1.460000	25.900000	12.150497	0.000000	-2.712626e+00
25%	22.000000	1.670000	60.300000	20.569330	0.000000	-5.825854e-01
50%	29.000000	1.720000	70.100000	23.323418	0.000000	1.142238e-01
75%	40.000000	1.750000	76.000000	25.419356	1.000000	6.445149e-01
max	61.000000	1.930000	109.900000	32.718367	5.000000	2.491231e+00

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [10]:
```
x=data[['Age','Height','Weight']]
y=data['Bmi']
```

In [33]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data= scaler.fit_transform(x)
```

In [34]:
```
x= pd.DataFrame(scaled_data, columns=x.columns)

x
```

Out[34]:

	Age	Height	Weight
0	2.522573	1.636157	0.958264
1	2.436718	0.006666	0.018847
2	2.436718	-1.855609	-0.115178
3	2.436718	-2.903139	-1.318921
4	2.436718	-1.506432	0.579767
...	...	...	...
736	0.204498	1.752549	0.536333
737	1.063045	2.334510	0.883805
738	-0.568193	1.286980	0.309856
739	0.290353	1.985334	0.623201
740	1.148899	2.567295	0.976878

741 rows × 3 columns

In [ ]:

In [ ]:

In [35]:
```
from sklearn.model_selection import train_test_split
```

In [36]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=43)
```

In [37]:
```
print('Complete Dataset Shape:',data.shape)
print('Shape of x1_train:',xtrain.shape)
print('Shape of x1_test:',xtest.shape)
print('Shape of y1_train:',ytrain.shape)
print('Shape of y1_test',ytest.shape)
```

```
Complete Dataset Shape: (617, 6)
Shape of x1_train: (518, 3)
Shape of x1_test: (223, 3)
Shape of y1_train: (518,)
Shape of y1_test (223,)
```

In [38]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(xtrain,ytrain)
```

Out[38]:

LinearRegression

```
LinearRegression()
```

In [ ]:

In [39]:
```
y_pred=model.predict(xtest)
y_pred[:5]
```

Out[39]:
```
array([25.59415329, 36.0285382 , 34.18289623, 26.91459167, 22.76391314])
```

In [40]:
```
ytest.head()
```

Out[40]:
```
11     25.401384
462    38.567493
461    35.492158
120    27.069388
359    22.862369
Name: Bmi, dtype: float64
```

In [41]:
```
from sklearn.metrics import r2_score

ypred = model.predict(xtest)
r2 = r2_score(ytest, ypred) * 100
print(r2)
```

```
97.92088351650433
```

In [42]:
```
model.score(xtrain,ytrain)
```

Out[42]:
```
0.9711024120564841
```

In [43]:
```
model.score(xtest,ytest)
```

Out[43]:
```
0.9792088351650433
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
import joblib
```

In [ ]:
```
joblib.dump(model, "BMI_CLASS_PREDICTION.pkl")
```

In [ ]:
```
model=joblib.load("BMI_CLASS_PREDICTION.pkl")
```

In [ ]:
```
newdata=np.array([34,1.86,95.70,27.662157]).reshape(1,-1)
newdata
```

In [ ]:
```
prediction=model.predict(newdata)
prediction
```

In [ ]:
```
if prediction[0]==0:
    print('Normal Weight')
elif prediction[0]==1:
    print('Overweight')
elif prediction[0]==2:
    print('Underweight')
elif prediction[0]==3:
    print('Obese Class 3')
elif prediction[0]==4:
    print('Obese Class 2')
else:
    print('Obese Class 1')
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: