Importing the Dependencies

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

Data Collection and Processing

In [3]:

```
calories_data=pd.read_csv('calories.csv')
```

In [4]:

```
exercise_data=pd.read_csv('exercise.csv')
```

In [5]:

```
calories=pd.concat([exercise_data,calories_data['Calories']],axis=1)
```

In [6]:

```
calories.head()
```

Out[6]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8	35.0

In [7]:

```
calories.shape
```

Out[7]:

```
(15000, 9)
```

In [8]:

```
calories.isnull().sum()
```

Out[8]:

```
User_ID       0
Gender        0
Age           0
Height        0
Weight        0
Duration      0
Heart_Rate    0
Body_Temp     0
Calories      0
dtype: int64
```

Data Visualization

In [9]:

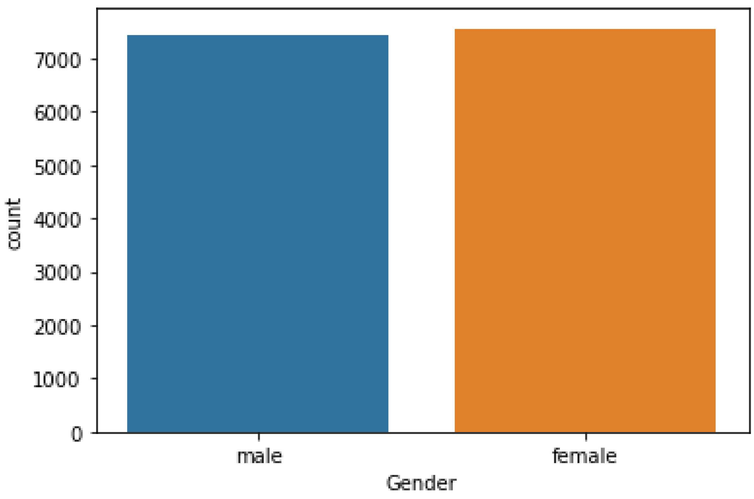
```
sns.countplot(calories['Gender'])
```

Loading [MathJax]/extensions/Safe.js

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the
only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[9]:

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



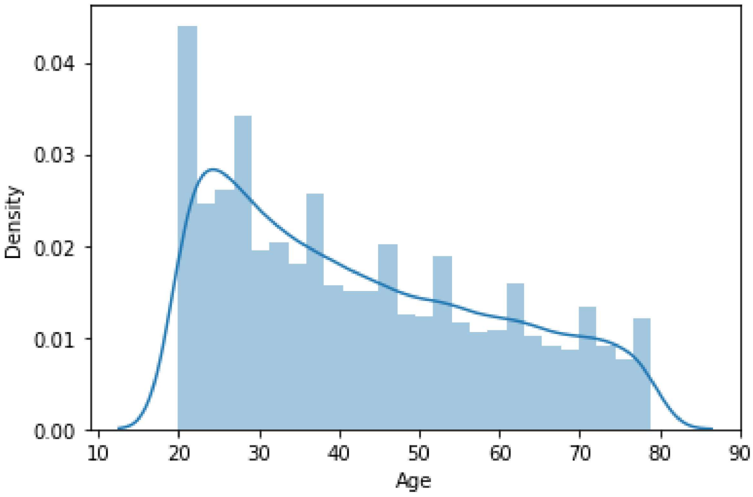
In [10]:

```python
sns.distplot(calories['Age'])
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```

Out[10]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



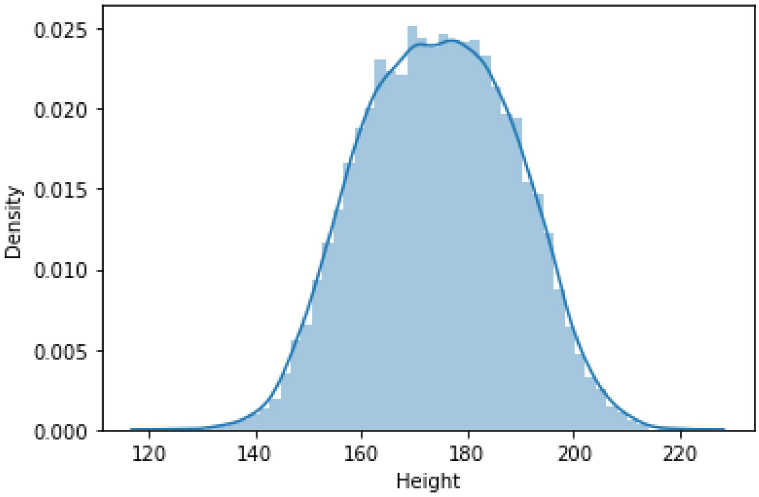
In [11]:

```python
sns.distplot(calories['Height'])
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```

Loading [MathJax]/extensions/Safe.js

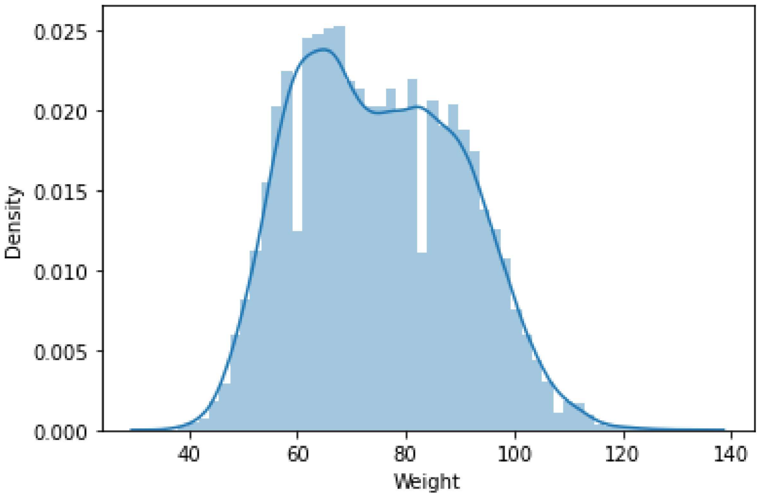
```
Out[11]: <AxesSubplot:xlabel='Height', ylabel='Density'>
```



```
In [12]: sns.distplot(calories['Weight'])
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histogr
ams).
  warnings.warn(msg, FutureWarning)
```

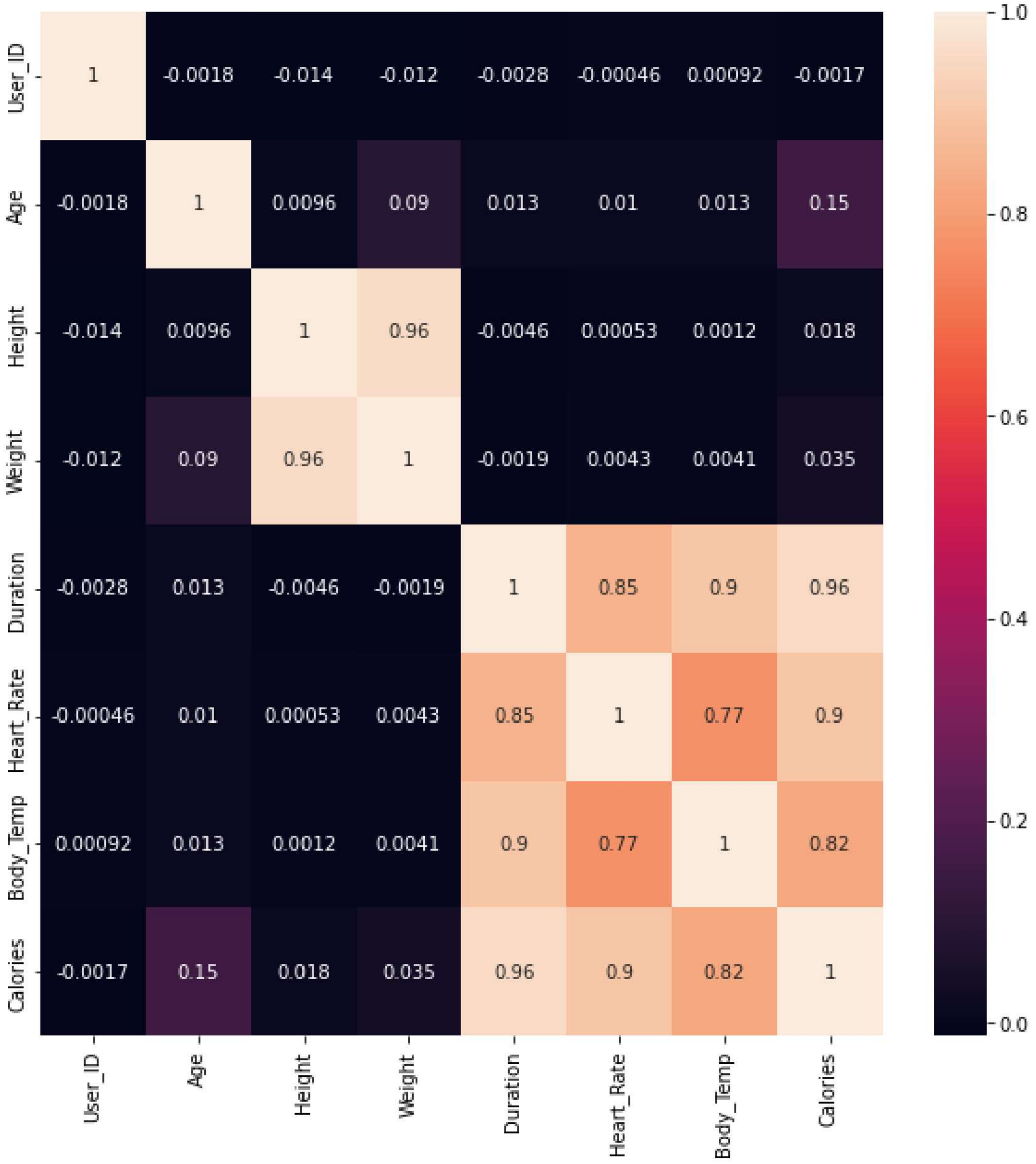
```
Out[12]: <AxesSubplot:xlabel='Weight', ylabel='Density'>
```



```
In [13]: correlation=calories.corr()
         plt.figure(figsize=(10,10))
         sns.heatmap(correlation,annot=True)
```

```
Out[13]: <AxesSubplot:>
```

Loading [MathJax]/extensions/Safe.js



In [14]:
```
calories.replace({"Gender":{'male':0,'female':1}},inplace=True)
```

In [15]:
```
calories.head()
```

Out[15]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	0	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	1	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	0	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	1	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	1	27	154.0	58.0	10.0	81.0	39.8	35.0

Separating features and target

In [16]:
```
X=calories.drop(columns=['User_ID','Calories'],axis=1)
Calories']
```

Loading [MathJax]/extensions/Safe.js

```
In [17]: print(X)
```

```
       Gender  Age  Height  Weight  Duration  Heart_Rate  Body_Temp
0           0   68   190.0    94.0      29.0       105.0       40.8
1           1   20   166.0    60.0      14.0        94.0       40.3
2           0   69   179.0    79.0       5.0        88.0       38.7
3           1   34   179.0    71.0      13.0       100.0       40.5
4           1   27   154.0    58.0      10.0        81.0       39.8
...       ...  ...     ...     ...       ...         ...        ...
14995       1   20   193.0    86.0      11.0        92.0       40.4
14996       1   27   165.0    65.0       6.0        85.0       39.2
14997       1   43   159.0    58.0      16.0        90.0       40.1
14998       0   78   193.0    97.0       2.0        84.0       38.3
14999       0   63   173.0    79.0      18.0        92.0       40.5

[15000 rows x 7 columns]
```

```
In [18]: print(Y)
```

```
0        231.0
1         66.0
2         26.0
3         71.0
4         35.0
         ...
14995     45.0
14996     23.0
14997     75.0
14998     11.0
14999     98.0
Name: Calories, Length: 15000, dtype: float64
```

Splitting the data into training and testing data

```
In [19]: X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
```

Model Training

Linear Regressor

```
In [20]: model1=LinearRegression()
         model1.fit(X_train,Y_train)
         prediction1=model1.predict(X_test)
         print(prediction1)
```

```
[ 18.90073679 112.37963061   2.12558084 ...  30.60482603 176.6454258
 182.98259929]
```

Mean Absolute Error for Linear Regressor

```
In [21]: mae=metrics.mean_absolute_error(Y_test,prediction1)
         print("Mean Absolute Error = ",mae)
```

```
Mean Absolute Error =  8.379646102595844
```

Random Forest Regressor

Loading [MathJax]/extensions/Safe.js

```
In [22]: model2=RandomForestRegressor()
         model2.fit(X_train,Y_train)
         prediction2=model2.predict(X_test)
         print(prediction2)

         [ 33.31 101.05  11.47 ...  37.42 184.96 172.61]
```

Mean Absolute Error for Random Forest Regressor

```
In [23]: mae=metrics.mean_absolute_error(Y_test,prediction2)
         print("Mean Absolute Error= ",mae)

         Mean Absolute Error =  1.84171
```

```
In [28]: pred = model2.predict([[1, 68, 50.0, 94.0, 90.0, 105.0, 10.8]])

         pred

         C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X
         does not have valid feature names, but RandomForestRegressor was fitted with f
         eature names
           warnings.warn(

Out[28]: array([204.28])
```

```
In [ ]:
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js