In [1]:
```
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
a=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\calories.csv")
a
```

Out[2]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190	94	29	105	40.8	231
1	14861698	female	20	166	60	14	94	40.3	66
2	11179863	male	69	179	79	5	88	38.7	26
3	16180408	female	34	179	71	13	100	40.5	71
4	17771927	female	27	154	58	10	81	39.8	35
...	...	...	...	...	...	...	...	...	...
14995	15644082	female	20	193	86	11	92	40.4	45
14996	17212577	female	27	165	65	6	85	39.2	23
14997	17271188	female	43	159	58	16	90	40.1	75
14998	18643037	male	78	193	97	2	84	38.3	11
14999	11751526	male	63	173	79	18	92	40.5	98

15000 rows × 9 columns

In [3]:
```
a.head(5)
```

Out[3]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190	94	29	105	40.8	231
1	14861698	female	20	166	60	14	94	40.3	66
2	11179863	male	69	179	79	5	88	38.7	26
3	16180408	female	34	179	71	13	100	40.5	71
4	17771927	female	27	154	58	10	81	39.8	35

In [4]:
```
a.tail(5)
```

Out[4]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
14995	15644082	female	20	193	86	11	92	40.4	45
14996	17212577	female	27	165	65	6	85	39.2	23
14997	17271188	female	43	159	58	16	90	40.1	75
14998	18643037	male	78	193	97	2	84	38.3	11
14999	11751526	male	63	173	79	18	92	40.5	98

In [5]:
```
a.describe()
```

Out[5]:

	User_ID	Age	Height	Weight	Duration	Heart_Rate	Body_T
count	1.500000e+04	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.00
mean	1.497736e+07	42.789800	174.465133	74.966867	15.530600	95.518533	40.02
std	2.872851e+06	16.980264	14.258114	15.035657	8.319203	9.583328	0.77
min	1.000116e+07	20.000000	123.000000	36.000000	1.000000	67.000000	37.10
25%	1.247419e+07	28.000000	164.000000	63.000000	8.000000	88.000000	39.60
50%	1.499728e+07	39.000000	175.000000	74.000000	16.000000	96.000000	40.20
75%	1.744928e+07	56.000000	185.000000	87.000000	23.000000	103.000000	40.60
max	1.999965e+07	79.000000	222.000000	132.000000	30.000000	128.000000	41.50

In [6]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   User_ID     15000 non-null  int64  
 1   Gender      15000 non-null  object 
 2   Age         15000 non-null  int64  
 3   Height      15000 non-null  int64  
 4   Weight      15000 non-null  int64  
 5   Duration    15000 non-null  int64  
 6   Heart_Rate  15000 non-null  int64  
 7   Body_Temp   15000 non-null  float64
 8   Calories    15000 non-null  int64  
dtypes: float64(1), int64(7), object(1)
memory usage: 1.0+ MB
```

In [8]:
```
a['Age'].unique()
```

Out[8]:
```
array([68, 20, 69, 34, 27, 36, 33, 41, 60, 26, 21, 66, 32, 53, 39, 46, 50,
       67, 31, 48, 29, 42, 62, 38, 25, 24, 22, 74, 70, 44, 61, 63, 54, 47,
       35, 28, 77, 64, 45, 57, 49, 40, 55, 23, 43, 52, 79, 51, 59, 30, 37,
       56, 58, 78, 73, 76, 72, 65, 71, 75], dtype=int64)
```

In [9]:
```
a.groupby(['Gender']).count()
```

Out[9]:

Gender	User_ID	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
female	7553	7553	7553	7553	7553	7553	7553	7553
male	7447	7447	7447	7447	7447	7447	7447	7447

In [13]:
```
b=a.drop(['User_ID','Height','Weight','Duration'],axis=1)
b
```

Out[13]:

	Gender	Age	Heart_Rate	Body_Temp	Calories
0	male	68	105	40.8	231
1	female	20	94	40.3	66
2	male	69	88	38.7	26
3	female	34	100	40.5	71
4	female	27	81	39.8	35
...	...	...	...	...	...
14995	female	20	92	40.4	45
14996	female	27	85	39.2	23
14997	female	43	90	40.1	75
14998	male	78	84	38.3	11
14999	male	63	92	40.5	98

15000 rows × 5 columns

In [14]:

```
c=pd.get_dummies(b,dtype=int)
c
```

Out[14]:

	Age	Heart_Rate	Body_Temp	Calories	Gender_female	Gender_male
0	68	105	40.8	231	0	1
1	20	94	40.3	66	1	0
2	69	88	38.7	26	0	1
3	34	100	40.5	71	1	0
4	27	81	39.8	35	1	0
...	...	...	...	...	...	...
14995	20	92	40.4	45	1	0
14996	27	85	39.2	23	1	0
14997	43	90	40.1	75	1	0
14998	78	84	38.3	11	0	1
14999	63	92	40.5	98	0	1

15000 rows × 6 columns

In [17]:

```
y=c['Calories']
y
```

Out[17]:

```
0        231
1         66
2         26
3         71
4         35
        ...
14995     45
```

```
14996     23
14997     75
14998     11
14999     98
Name: Calories, Length: 15000, dtype: int64
```

In [19]:

```
x=c.drop(['Calories'],axis=1)
x
```

Out[19]:

	Age	Heart_Rate	Body_Temp	Gender_female	Gender_male
0	68	105	40.8	0	1
1	20	94	40.3	1	0
2	69	88	38.7	0	1
3	34	100	40.5	1	0
4	27	81	39.8	1	0
...	...	...	...	...	...
14995	20	92	40.4	1	0
14996	27	85	39.2	1	0
14997	43	90	40.1	1	0
14998	78	84	38.3	0	1
14999	63	92	40.5	0	1

15000 rows × 5 columns

In [20]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=55)
```

In [21]:

```
x_train
```

Out[21]:

	Age	Heart_Rate	Body_Temp	Gender_female	Gender_male
12943	43	83	38.2	1	0
659	48	87	39.5	0	1
7964	27	90	39.6	0	1
13569	29	96	40.6	0	1
993	24	101	41.0	1	0
...	...	...	...	...	...
14877	67	96	40.0	0	1
968	66	99	40.8	1	0
12583	64	83	39.2	0	1
4762	24	89	40.1	1	0
12749	66	91	40.0	1	0

10050 rows × 5 columns

In [22]:
```
y_train
```

Out[22]:
```
12943      8
659       34
7964      40
13569     88
993      145
        ...
14877     84
968      135
12583     32
4762      63
12749     61
Name: Calories, Length: 10050, dtype: int64
```

In [23]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[23]:
```
LinearRegression()
```

In [24]:
```
ypred=reg.predict(x_test)
ypred
```

Out[24]:
```
array([ -4.65157304,  62.73575692,  58.65968775, ..., 168.48952062,
       170.46822995,  55.54179259])
```

In [25]:
```
from sklearn.metrics import r2_score
r2_score(ypred,y_test)
```

Out[25]:
```
0.8462802527979185
```

In [26]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,ypred)
```

Out[26]:
```
510.9104768590488
```

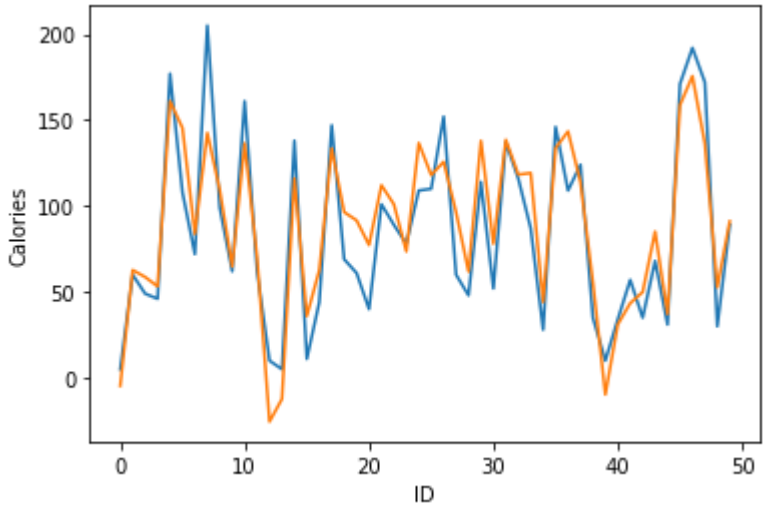
In [28]:
```
results=pd.DataFrame(columns=['Calories','Predicted'])
results['Calories']=y_test
results["Predicted"]=ypred
results=results.reset_index()
results['ID']=results.index
results.head(5)
```

Out[28]:

	index	Calories	Predicted	ID
0	10827	5	-4.651573	0
1	2990	60	62.735757	1
2	6780	49	58.659688	2
3	4828	46	53.224977	3
4	7602	177	160.989545	4

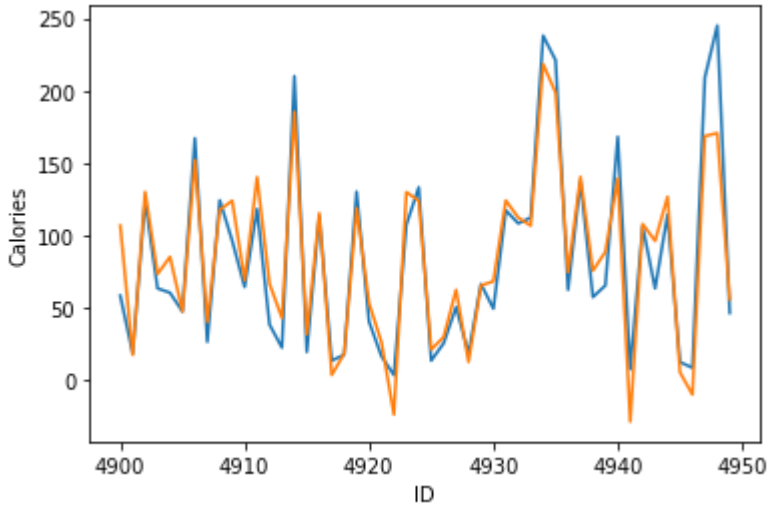
In [30]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='Calories',data=results.head(50))
sns.lineplot(x='ID',y='Predicted',data=results.head(50))
plt.plot()
```

Out[30]:
```
[]
```



In [32]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='Calories',data=results.tail(50))
sns.lineplot(x='ID',y='Predicted',data=results.tail(50))
plt.plot()
```

Out[32]:
```
[]
```



In [33]:
```
cor=c.corr()
cor
```

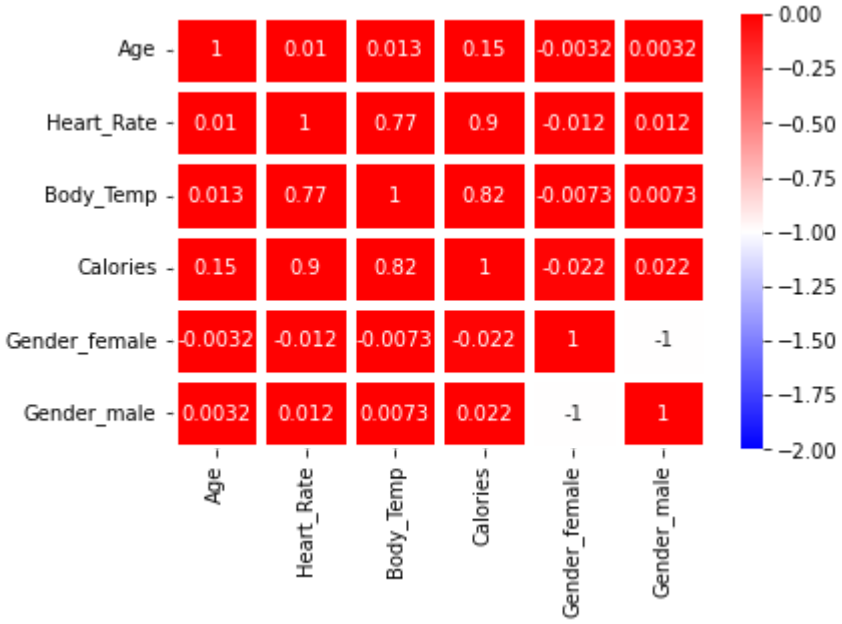
Out[33]:

	Age	Heart_Rate	Body_Temp	Calories	Gender_female	Gender_male
Age	1.000000	0.010482	0.013175	0.154395	-0.003222	0.003222
Heart_Rate	0.010482	1.000000	0.771529	0.897882	-0.011555	0.011555
Body_Temp	0.013175	0.771529	1.000000	0.824558	-0.007264	0.007264

	Age	Heart_Rate	Body_Temp	Calories	Gender_female	Gender_male
Calories	0.154395	0.897882	0.824558	1.000000	-0.022357	0.022357
Gender_female	-0.003222	-0.011555	-0.007264	-0.022357	1.000000	-1.000000
Gender_male	0.003222	0.011555	0.007264	0.022357	-1.000000	1.000000

In [34]:
```
import seaborn as sb
sb.heatmap(cor,vmax=0,vmin=-2,annot=True,linewidth=-5,cmap="bwr")
```

Out[34]: <AxesSubplot:>



In [ ]: