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

In [2]:
```
data = pd.read_csv('calories.csv')
data
```

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]:
```
data.head(10)
```

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
5	15130815	female	36	151	50	23	96	40.7	123
6	19602372	female	33	158	56	22	95	40.5	112
7	11117088	male	41	175	85	25	100	40.7	143
8	12132339	male	60	186	94	21	97	40.4	134
9	17964668	female	26	146	51	16	90	40.2	72

In [4]:
```
data.tail(10)
```

Out[4]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
14990	19715870	female	22	190	79	19	96	40.3	89
14991	10050978	male	51	181	87	9	91	39.6	44
14992	14722670	male	27	170	70	13	92	40.1	46
14993	13584585	male	45	179	78	11	98	39.9	60
14994	18209611	female	48	159	57	10	94	39.8	52
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]:
```
data.describe()
```

Out[5]:

	User_ID	Age	Height	Weight	Duration	Heart_Rate	Bod
count	1.500000e+04	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000
mean	1.497736e+07	42.789800	174.465133	74.966867	15.530600	95.518533	40
std	2.872851e+06	16.980264	14.258114	15.035657	8.319203	9.583328	0
min	1.000116e+07	20.000000	123.000000	36.000000	1.000000	67.000000	37
25%	1.247419e+07	28.000000	164.000000	63.000000	8.000000	88.000000	39
50%	1.499728e+07	39.000000	175.000000	74.000000	16.000000	96.000000	40
75%	1.744928e+07	56.000000	185.000000	87.000000	23.000000	103.000000	40
max	1.999965e+07	79.000000	222.000000	132.000000	30.000000	128.000000	41

In [6]:
```
data.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 [7]:
```
data.isnull().sum()
```

Out[7]:
```
User_ID       0
Gender        0
Age           0
Height        0
Weight        0
Duration      0
Heart_Rate    0
Body_Temp     0
Calories      0
dtype: int64
```

In [8]:
```
data[('Gender')].unique()
```

Out[8]:
```
array(['male', 'female'], dtype=object)
```

In [9]:
```
data.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 [10]:
```
data1=data.drop(['User_ID'],axis=1)
```

In [11]:
```
data1
```

Out[11]:

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

15000 rows × 8 columns

In [12]:
```
data1.replace({'Gender':{'male':1,'female':0}},inplace=True)
data1
```

Out[12]:

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

15000 rows × 8 columns

In [13]:
```
y=data1['Calories']
```

In [14]:
```
y
```

Out[14]:
```
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 [15]:
```
x=data1.drop(['Calories'],axis=1)
```

In [16]:
```
x
```

Out[16]:

	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp
0	1	68	190	94	29	105	40.8
1	0	20	166	60	14	94	40.3
2	1	69	179	79	5	88	38.7
3	0	34	179	71	13	100	40.5
4	0	27	154	58	10	81	39.8
...	...	...	...	...	...	...	...
14995	0	20	193	86	11	92	40.4
14996	0	27	165	65	6	85	39.2
14997	0	43	159	58	16	90	40.1
14998	1	78	193	97	2	84	38.3
14999	1	63	173	79	18	92	40.5

15000 rows × 7 columns

In [17]:
```
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=42)
```

In [18]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(x_train,y_train)
```

Out[18]:

▾ LinearRegression

LinearRegression()

In [19]:
```
ypred=reg.predict(x_test)
```

In [20]:
```
ypred
```

Out[20]:
```
array([170.57288049, 192.11306598,  56.15350808, ...,   9.16714006,
       171.24764866,  76.50120096])
```

In [21]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[21]:
```
0.9663701348612177
```

In [22]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,ypred)
```

Out[22]:
```
132.91211125710086
```

In [23]:
```
res=pd.DataFrame(columns=['Calories','predicted'])
res['Calories']=y_test
res['predicted']=ypred
res=res.reset_index()
res['ID']=res.index
```

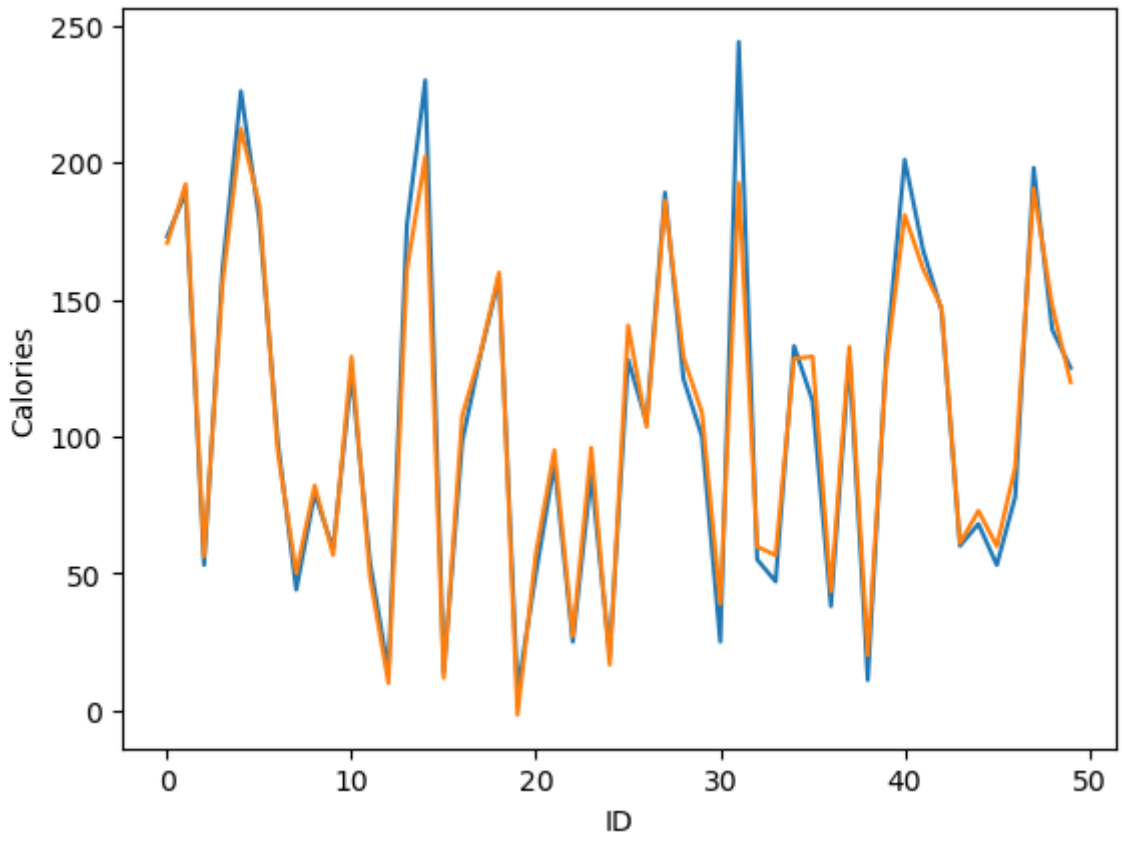
In [24]:
```
res.head(10)
```

Out[24]:

	index	Calories	predicted	ID
0	11499	173	170.572880	0
1	6475	189	192.113066	1
2	13167	53	56.153508	2
3	862	161	155.543653	3
4	5970	226	212.300083	4
5	6706	179	184.340136	5
6	3017	98	94.337860	6
7	3781	44	50.047039	7
8	3898	79	82.063578	8
9	2250	59	56.640077	9

In [25]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='Calories',data=res.head(50))
sns.lineplot(x='ID',y='predicted',data=res.head(50))
plt.plot()
```

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



In [26]:
```
new=[[1,22,175,80,25,75,43.5]]
```

In [27]:
```
real=reg.predict(new)
real
```

Out[27]:
```
array([42.69675717])
```

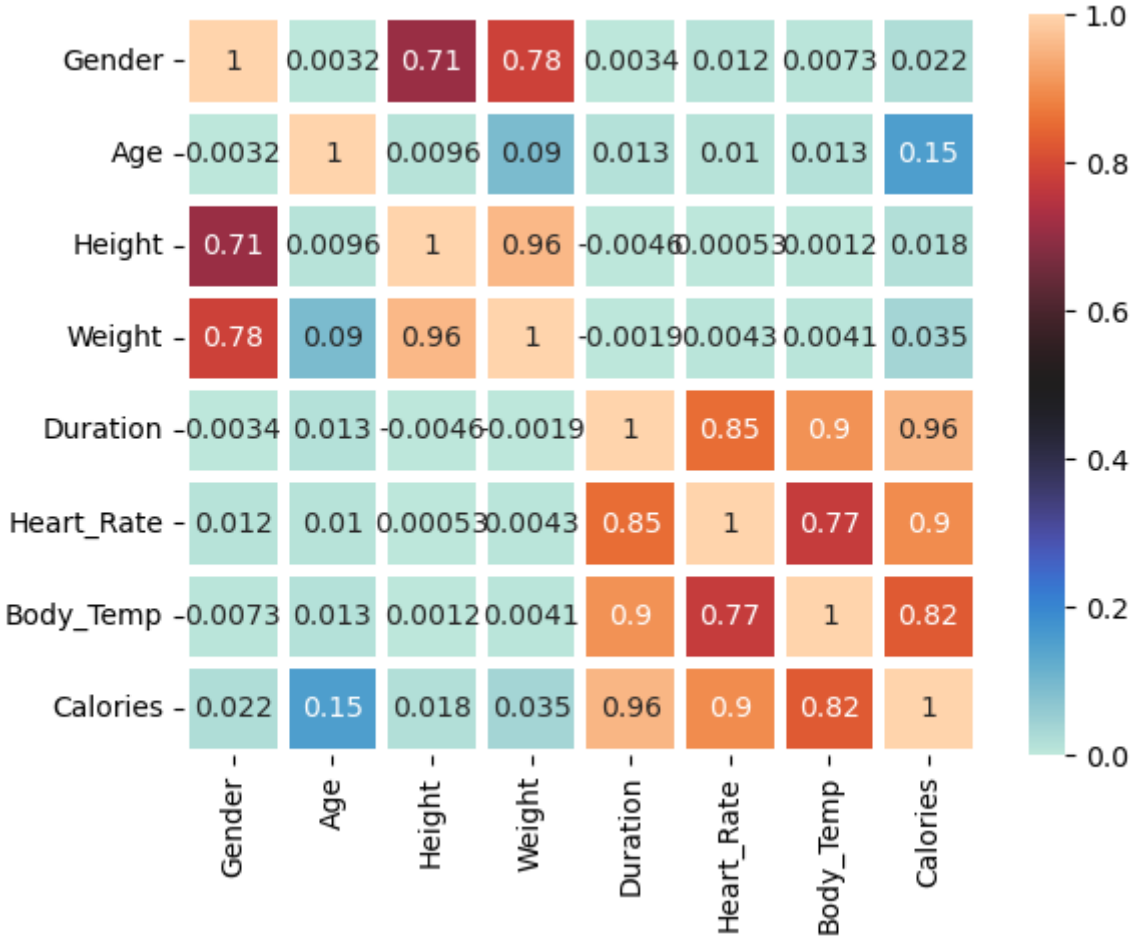
In [28]:
```
cor_mat=data1.corr()
cor_mat
```

Out[28]:

	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
Gender	1.000000	0.003222	0.710534	0.783186	0.003440	0.011555	0.007264	0.022357
Age	0.003222	1.000000	0.009554	0.090094	0.013247	0.010482	0.013175	0.154395
Height	0.710534	0.009554	1.000000	0.958451	-0.004625	0.000528	0.001200	0.017537
Weight	0.783186	0.090094	0.958451	1.000000	-0.001884	0.004311	0.004095	0.035481
Duration	0.003440	0.013247	-0.004625	-0.001884	1.000000	0.852869	0.903167	0.955421
Heart_Rate	0.011555	0.010482	0.000528	0.004311	0.852869	1.000000	0.771529	0.897882
Body_Temp	0.007264	0.013175	0.001200	0.004095	0.903167	0.771529	1.000000	0.824558
Calories	0.022357	0.154395	0.017537	0.035481	0.955421	0.897882	0.824558	1.000000

In [29]:
```
sns.heatmap(cor_mat,vmax=1,vmin=0,annot=True,linewidth=3,cmap='icefire')
```

Out[29]:
```
<Axes: >
```



In [ ]:
```
```