In [1]:
```
import pandas as pd
```

In [2]:
```
sr=pd.read_csv("calories.csv")
sr
```

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]:
```
sr.head()
```

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]:
```
sr.tail()
```

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]:
```
sr.shape
```

Out[5]:
```
(15000, 9)
```

In [6]:
```
sr.describe()
```

Out[6]:

	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 [7]:
```
sr.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]:
```
sr.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
```

In [9]:
```
br=sr.drop('User_ID',axis=1)
br
```

Out[9]:

	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 [10]:

```
br=pd.get_dummies(br,dtype=int)
br
```

Out[10]:

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

15000 rows × 9 columns

In [13]:

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

In [14]:

```
x
```

Out[14]:

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

15000 rows × 8 columns

In [15]:
```
y
```

Out[15]:
```
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 [62]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=42)
```

In [63]:
```
x_test.head()
```

Out[63]:

	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Gender_female	Gender_male
11499	45	154	52	26	107	40.6	1	0
6475	21	187	90	29	111	40.5	0	1
13167	58	176	77	11	90	40.0	0	1
862	35	182	89	24	108	40.8	0	1
5970	67	171	67	29	116	41.1	1	0

In [64]:
```
y_test.head()
```

Out[64]:
```
11499    173
6475     189
```

```
13167     53
862      161
5970     226
Name: Calories, dtype: int64
```

In [65]:
```
y_train.head()
```

Out[65]:
```
11810    191
970       75
1598      13
11805     52
2582     111
Name: Calories, dtype: int64
```

In [66]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[66]:
```
LinearRegression()
```

In [67]:
```
ypred=reg.predict(x_test)
```

In [68]:
```
ypred
```

Out[68]:
```
array([170.64343964, 192.26900724,  56.11908173, ...,  62.63627664,
       128.67424487, 135.03856171])
```

In [69]:
```
from sklearn.metrics import r2_score
r2_score (y_test,ypred)
```

Out[69]:
```
0.9670097908652697
```

In [70]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[70]:
```
129.59429728799708
```

In [26]:
```
Results=pd.DataFrame(columns=['Calories','predicted'])
Results['Calories']=y_test
Results['Predicted']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

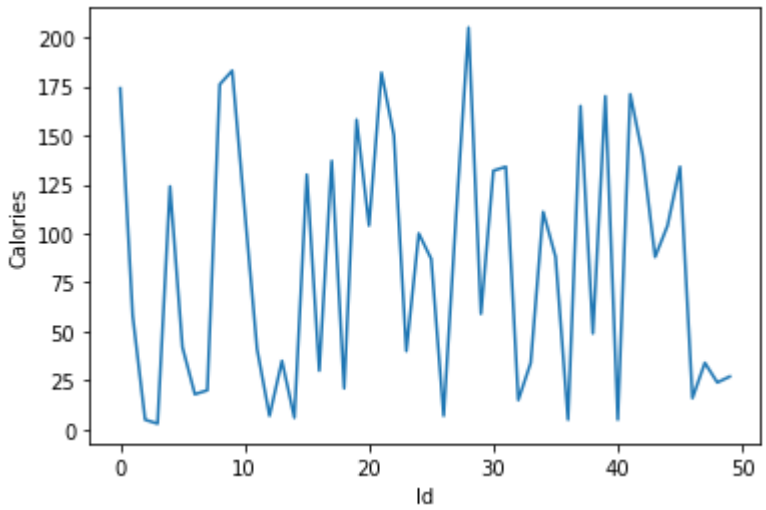
Out[26]:

	index	Calories	predicted	Predicted	Id
0	9944	174	NaN	161.632107	0
1	12	58	NaN	61.696479	1
2	5052	5	NaN	-2.894439	2
3	14845	3	NaN	-24.142551	3
4	12637	124	NaN	143.400608	4
5	9721	42	NaN	45.922950	5

	index	Calories	predicted	Predicted	Id
6	14141	18	NaN	5.094143	6
7	2372	20	NaN	28.782475	7
8	431	176	NaN	176.447455	8
9	6357	183	NaN	164.150580	9
10	12592	112	NaN	115.185402	10
11	7341	41	NaN	48.861047	11
12	76	7	NaN	8.311358	12
13	2715	35	NaN	36.008464	13
14	11145	6	NaN	-8.466337	14

In [27]:
```
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[27]:
```
[]
```



In [29]:
```
cor_mat=br.corr()
cor_mat
```

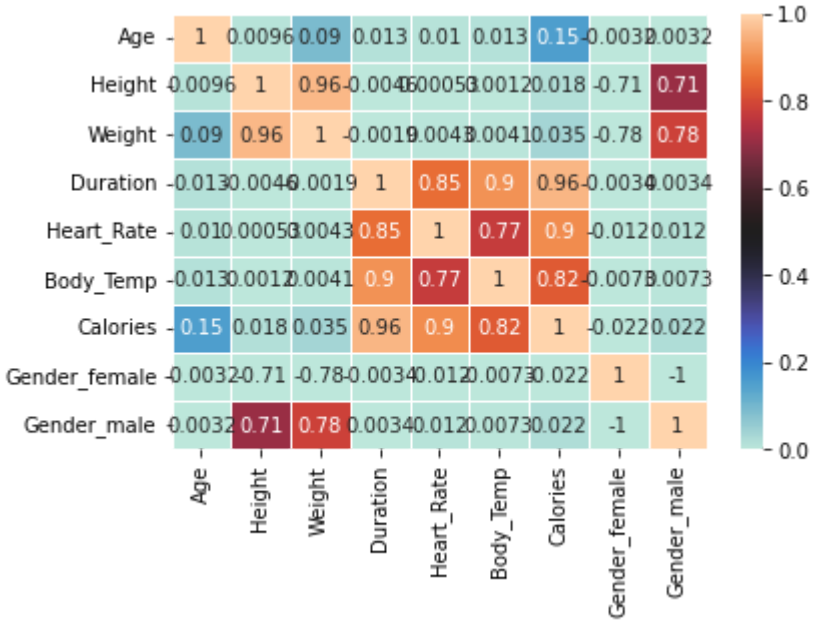
Out[29]:

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

	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories	Gende
Gender_male	0.003222	0.710534	0.783186	0.003440	0.011555	0.007264	0.022357	-

In [36]:
```
sns.heatmap(cor_mat,vmax=1,vmin=0,annot=True,linewidths=1,cmap='icefire')
```

Out[36]: <AxesSubplot:>



In [ ]: