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

In [2]:
```
data=pd.read_csv(r"C:\Users\DELL\Downloads\SmartWatch Data.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	X1	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart
0	1	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.
1	2	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.
2	3	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.
3	4	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.
4	5	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.

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

Out[4]:
```
X1                    0
age                   0
gender                0
height                0
weight                0
steps                 0
hear_rate             0
calories              0
distance              0
entropy_heart         0
entropy_setps         0
resting_heart         0
corr_heart_steps      0
norm_heart            0
intensity_karvonen    0
sd_norm_heart         0
steps_times_distance  0
device                0
activity              0
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6264 entries, 0 to 6263
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   X1                    6264 non-null   int64  
 1   age                   6264 non-null   int64  
 2   gender                6264 non-null   int64  
 3   height                6264 non-null   float64
 4   weight                6264 non-null   float64
 5   steps                 6264 non-null   float64
 6   hear_rate             6264 non-null   float64
 7   calories              6264 non-null   float64
 8   distance              6264 non-null   float64
 9   entropy_heart         6264 non-null   float64
 10  entropy_setps         6264 non-null   float64
 11  resting_heart         6264 non-null   float64
 12  corr_heart_steps      6264 non-null   float64
 13  norm_heart            6264 non-null   float64
 14  intensity_karvonen    6264 non-null   float64
 15  sd_norm_heart         6264 non-null   float64
 16  steps_times_distance  6264 non-null   float64
 17  device                6264 non-null   object 
 18  activity              6264 non-null   object 
dtypes: float64(14), int64(3), object(2)
memory usage: 929.9+ KB
```

In [7]:
```
data.shape
```

Out[7]:
```
(6264, 19)
```

In [8]:
```
data.describe()
```

Out[8]:

	X1	age	gender	height	weight	steps	hear_rate	calories	distance	entropy
count	6264.000000	6264.000000	6264.000000	6264.000000	6264.000000	6264.000000	6264.000000	6264.000000	6264.000000	6264.0
mean	1771.144317	29.158525	0.476533	169.709052	69.614464	109.562268	86.142331	19.471823	13.832555	6.0
std	1097.988748	8.908978	0.499489	10.324698	13.451878	222.797908	28.648385	27.309765	45.941437	0.7
min	1.000000	18.000000	0.000000	143.000000	43.000000	1.000000	2.222222	0.056269	0.000440	0.0
25%	789.750000	23.000000	0.000000	160.000000	60.000000	5.159534	75.598079	0.735875	0.019135	6.1
50%	1720.000000	28.000000	0.000000	168.000000	68.000000	10.092029	77.267680	4.000000	0.181719	6.1
75%	2759.250000	33.000000	1.000000	180.000000	77.300000	105.847222	95.669118	20.500000	15.697188	6.2
max	3670.000000	56.000000	1.000000	191.000000	115.000000	1714.000000	194.333333	97.500000	335.000000	6.4

In [9]:
```
data.drop("X1",axis=1,inplace=True)
```

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

Out[10]:

	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart_ste
0	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.0000
1	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.0000
2	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.0000
3	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.0000
4	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.9828

In [12]:
```
data.columns
```

Out[12]:
```
Index(['age', 'gender', 'height', 'weight', 'steps', 'hear_rate', 'calories',
       'distance', 'entropy_heart', 'entropy_setps', 'resting_heart',
       'corr_heart_steps', 'norm_heart', 'intensity_karvonen', 'sd_norm_heart',
       'steps_times_distance', 'device', 'activity'],
      dtype='object')
```

In [13]:
```
data["device"].unique
```

Out[13]:
```
<bound method Series.unique of 0       apple watch
1       apple watch
2       apple watch
3       apple watch
4       apple watch
           ...
6259        fitbit
6260        fitbit
6261        fitbit
6262        fitbit
6263        fitbit
Name: device, Length: 6264, dtype: object>
```

In [15]:
```
data["activity"].unique()
```

Out[15]:
```
array(['Lying', 'Sitting', 'Self Pace walk', 'Running 3 METs',
       'Running 5 METs', 'Running 7 METs'], dtype=object)
```

In [16]:
```
dummy=pd.get_dummies(data["device"])
```

In [17]:
```
df=pd.concat([data,dummy],axis=1)
```

In [18]:
```
df.head()
```

Out[18]:

	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart_ste
0	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.0000
1	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.0000
2	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.0000
3	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.0000
4	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.9828

In [19]:
```
df.drop("device",inplace=True,axis=1)
```

In [20]:
```
df.head()
```

Out[20]:

	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart_ste
0	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.0000
1	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.0000
2	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.0000
3	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.0000
4	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.9828

In [21]:
```
import seaborn as sns
```

In [22]:
```
from sklearn.preprocessing import LabelEncoder
```

In [23]:
```
le=LabelEncoder()
```

In [24]:
```
df["activity"]=le.fit_transform(df["activity"])
```

In [25]:
```
df.head()
```

Out[25]:

	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart_ste
0	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.0000
1	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.0000
2	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.0000
3	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.0000
4	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.9828

In [26]:
```
df["activity"].unique()
```

Out[26]:
```
array([0, 5, 4, 1, 2, 3])
```

In [28]:
```
x=df.drop("activity",axis=1)
y=df["activity"]
```

In [29]:
```
from sklearn.model_selection import train_test_split
```

In [30]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.8,random_state=0)
```

In [31]:
```
x_train
```

Out[31]:

	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_he
2369	29	0	163.0	61.0	117.000000	130.115979	0.753500	0.051710	6.107633	5.857432	78.531302	
4778	25	1	160.0	75.0	1.000000	129.750000	1.000000	1.000000	6.162890	4.232097	6.125000	
4132	20	1	180.0	79.3	4.568421	76.235019	68.000000	15.751579	6.247928	6.247928	75.668850	
3101	49	0	152.0	48.6	5.826087	90.750000	0.250667	0.004776	6.142147	5.787845	60.846154	
3082	49	0	152.0	48.6	5.600000	102.500000	0.252000	0.003566	6.142147	5.787845	60.846154	
...	...	...	...	...	...	...	...	...	...	...	...	
4931	36	0	157.5	53.6	4.226835	76.171225	89.000000	15.755374	6.169925	6.169925	75.672837	
3264	31	0	158.0	59.1	6.833333	150.625000	0.246000	0.006145	6.195296	6.001153	84.200000	
1653	36	1	173.0	85.7	520.545455	87.000000	20.033067	0.352624	5.945808	5.928982	71.000000	
2607	30	0	168.0	56.0	10.771429	78.531302	0.344533	0.008327	6.135030	6.080236	72.000000	
2732	22	0	168.0	62.0	126.000000	101.000000	0.756000	0.056500	6.075165	6.153087	56.200000	

5011 rows × 18 columns

In [32]:
```
y_train
```

Out[32]:
```
2369    2
4778    0
4132    2
3101    2
3082    1
       ..
4931    2
3264    3
1653    5
2607    0
2732    2
Name: activity, Length: 5011, dtype: int32
```

In [34]:
```
x_train.shape
```

Out[34]:
```
(5011, 18)
```

In [35]:
```
y_train.shape
```

Out[35]:
```
(5011,)
```

In [36]:
```
x_test.shape
```

Out[36]:
```
(1253, 18)
```

In [37]:
```
y_test.shape
```

Out[37]:
```
(1253,)
```

In [39]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [40]:
```
model=RandomForestClassifier()
```

In [41]:
```
model.fit(x_train,y_train)
```

Out[41]:
```
RandomForestClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [42]:
```
y_pred=model.predict(x_test)
```

In [43]:
```
y_pred
```

Out[43]:
```
array([5, 0, 5, ..., 5, 5, 4])
```

In [44]:
```
from sklearn.metrics import confusion_matrix
```

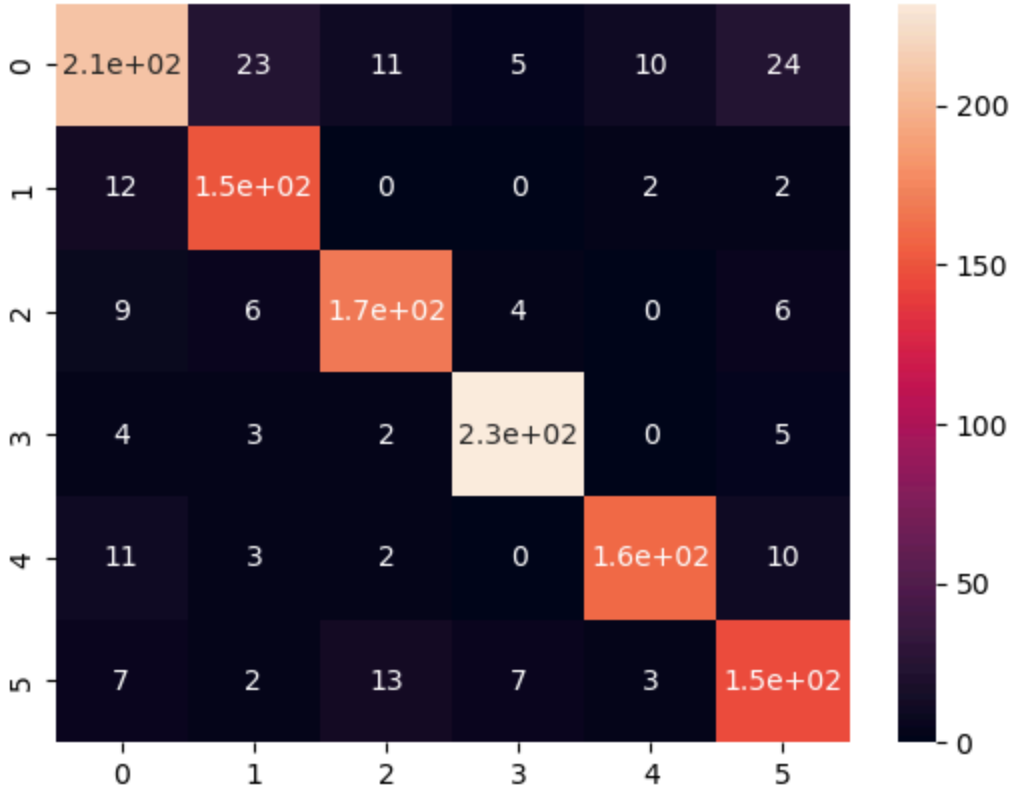
In [45]:
```
confusion_matrix=confusion_matrix(y_test,y_pred)
```

In [46]:
```
print(confusion_matrix)
```

```
[[210  23  11   5  10  24]
 [ 12 151   0   0   2   2]
 [  9   6 168   4   0   6]
 [  4   3   2 232   0   5]
 [ 11   3   2   0 160  10]
 [  7   2  13   7   3 146]]
```

In [47]:
```
sns.heatmap(confusion_matrix,annot=True)
```

Out[47]: <Axes: >



In [49]:
```
from sklearn.metrics import classification_report,precision_score,recall_score,f1_score
```

In [52]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.74      0.78       283
           1       0.80      0.90      0.85       167
           2       0.86      0.87      0.86       193
           3       0.94      0.94      0.94       246
           4       0.91      0.86      0.89       186
           5       0.76      0.82      0.79       178

    accuracy                           0.85      1253
   macro avg       0.85      0.86      0.85      1253
weighted avg       0.85      0.85      0.85      1253
```

In [ ]:
```

```