ae55fgxev

July 28, 2023

[12]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[13]:
```
df=pd.read_csv("/content/16_Sleep_health_and_lifestyle_dataset.csv")
df
```

[13]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	None
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
2	3	Male	28	Doctor	6.2	6	60	8	Normal				
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese				
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese				
..	…	…	…	…	…	…	…	…	…				
369	370	Female	59	Nurse	8.1	9	75	3	Overweight				
370	371	Female	59	Nurse	8.0	9	75	3	Overweight				
371	372	Female	59	Nurse	8.1	9	75	3	Overweight				
372	373	Female	59	Nurse	8.1	9	75	3	Overweight				
373	374	Female	59	Nurse	8.1	9	75	3	Overweight				

```
2           125/80          75        10000          None
3           140/90          85         3000    Sleep Apnea
4           140/90          85         3000    Sleep Apnea
..             …          …           …             …
369         140/95          68         7000    Sleep Apnea
370         140/95          68         7000    Sleep Apnea
371         140/95          68         7000    Sleep Apnea
372         140/95          68         7000    Sleep Apnea
373         140/95          68         7000    Sleep Apnea

[374 rows x 13 columns]
```

[14]:
```
df.head()
```

[14]:
```
   Person ID Gender  Age            Occupation  Sleep Duration  \
0          1   Male   27     Software Engineer             6.1
1          2   Male   28                Doctor             6.2
2          3   Male   28                Doctor             6.2
3          4   Male   28  Sales Representative             5.9
4          5   Male   28  Sales Representative             5.9

   Quality of Sleep  Physical Activity Level  Stress Level BMI Category  \
0                 6                       42             6   Overweight
1                 6                       60             8       Normal
2                 6                       60             8       Normal
3                 4                       30             8        Obese
4                 4                       30             8        Obese

  Blood Pressure  Heart Rate  Daily Steps Sleep Disorder
0         126/83          77         4200          None
1         125/80          75        10000          None
2         125/80          75        10000          None
3         140/90          85         3000    Sleep Apnea
4         140/90          85         3000    Sleep Apnea
```

1 DATA CLEANING AND DATA PREPROCESSING

[15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Person ID                374 non-null    int64
 1   Gender                   374 non-null    object
 2   Age                      374 non-null    int64
```

```
 3   Occupation               374 non-null    object
 4   Sleep Duration           374 non-null    float64
 5   Quality of Sleep         374 non-null    int64
 6   Physical Activity Level  374 non-null    int64
 7   Stress Level             374 non-null    int64
 8   BMI Category             374 non-null    object
 9   Blood Pressure           374 non-null    object
 10  Heart Rate               374 non-null    int64
 11  Daily Steps              374 non-null    int64
 12  Sleep Disorder           374 non-null    object
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

[16]:
```
df.describe()
```

[16]:

	Person ID	Age	Sleep Duration	Quality of Sleep	\
count	374.000000	374.000000	374.000000	374.000000	
mean	187.500000	42.184492	7.132086	7.312834	
std	108.108742	8.673133	0.795657	1.196956	
min	1.000000	27.000000	5.800000	4.000000	
25%	94.250000	35.250000	6.400000	6.000000	
50%	187.500000	43.000000	7.200000	7.000000	
75%	280.750000	50.000000	7.800000	8.000000	
max	374.000000	59.000000	8.500000	9.000000	

	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	374.000000	374.000000	374.000000	374.000000
mean	59.171123	5.385027	70.165775	6816.844920
std	20.830804	1.774526	4.135676	1617.915679
min	30.000000	3.000000	65.000000	3000.000000
25%	45.000000	4.000000	68.000000	5600.000000
50%	60.000000	5.000000	70.000000	7000.000000
75%	75.000000	7.000000	72.000000	8000.000000
max	90.000000	8.000000	86.000000	10000.000000

[17]:
```
df.columns
```

[17]:
```
Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder'],
      dtype='object')
```

[18]:
```
df1=df.dropna(axis=1)
df1
```

[18]:
```
     Person ID  Gender  Age            Occupation  Sleep Duration  \
0            1    Male   27     Software Engineer             6.1
1            2    Male   28                Doctor             6.2
2            3    Male   28                Doctor             6.2
3            4    Male   28  Sales Representative             5.9
4            5    Male   28  Sales Representative             5.9
..         …     …  …                   …              …
369        370  Female   59                Nurse             8.1
370        371  Female   59                Nurse             8.0
371        372  Female   59                Nurse             8.1
372        373  Female   59                Nurse             8.1
373        374  Female   59                Nurse             8.1

     Quality of Sleep  Physical Activity Level  Stress Level BMI Category  \
0                   6                       42             6   Overweight
1                   6                       60             8       Normal
2                   6                       60             8       Normal
3                   4                       30             8        Obese
4                   4                       30             8        Obese
..                …                        …           …          …
369                 9                       75             3   Overweight
370                 9                       75             3   Overweight
371                 9                       75             3   Overweight
372                 9                       75             3   Overweight
373                 9                       75             3   Overweight

    Blood Pressure  Heart Rate  Daily Steps Sleep Disorder
0           126/83          77         4200           None
1           125/80          75        10000           None
2           125/80          75        10000           None
3           140/90          85         3000    Sleep Apnea
4           140/90          85         3000    Sleep Apnea
..             …         …          …             …
369         140/95          68         7000    Sleep Apnea
370         140/95          68         7000    Sleep Apnea
371         140/95          68         7000    Sleep Apnea
372         140/95          68         7000    Sleep Apnea
373         140/95          68         7000    Sleep Apnea

[374 rows x 13 columns]
```

[19]:
```
df1.columns
```

[19]:
```
Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder'],
```

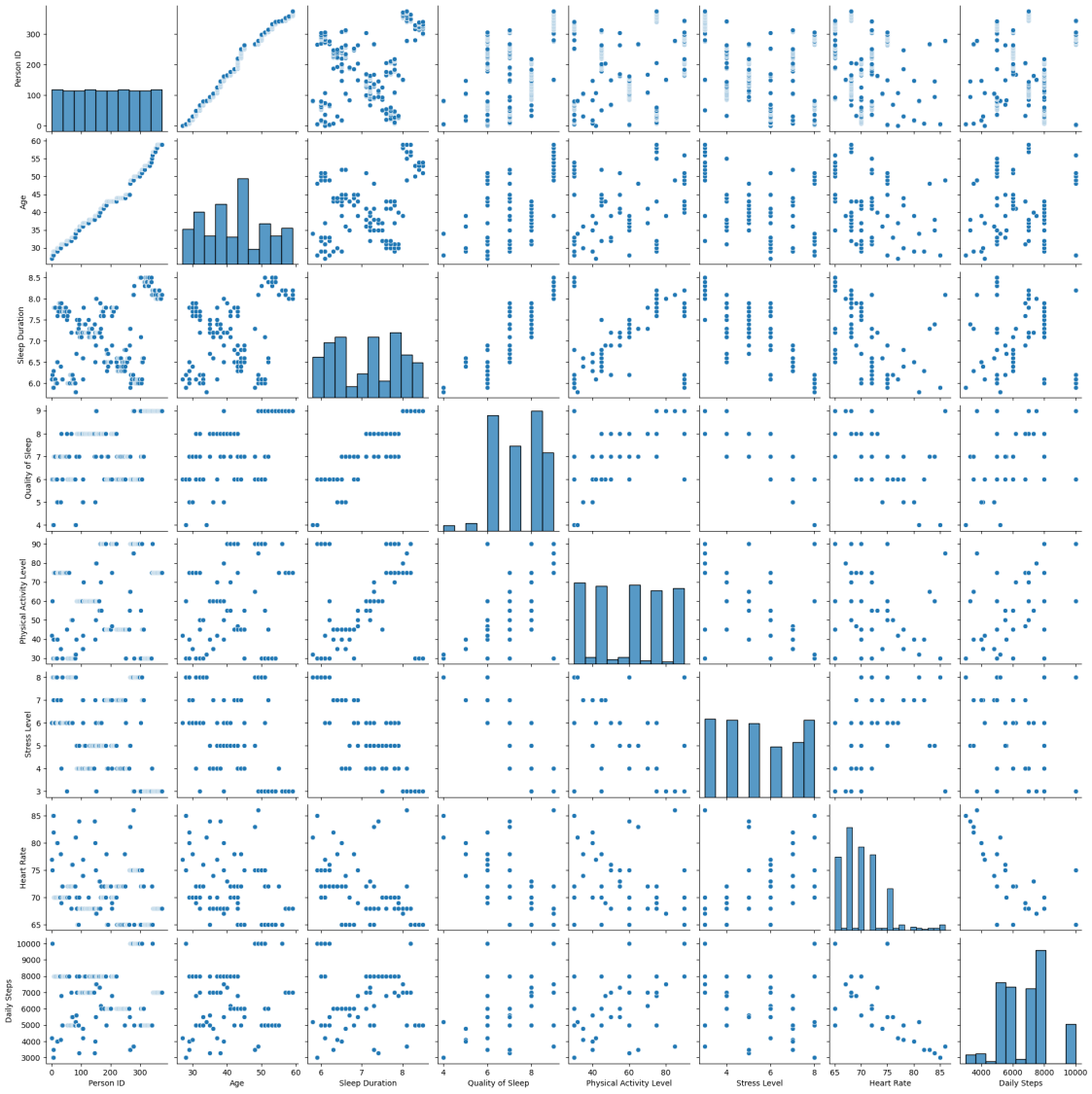
```
dtype='object')
```

[20]:
```
df1=df1[['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder']]
```

2 EDA AND VISUALIZATION

[21]:
```
sns.pairplot(df1)
```

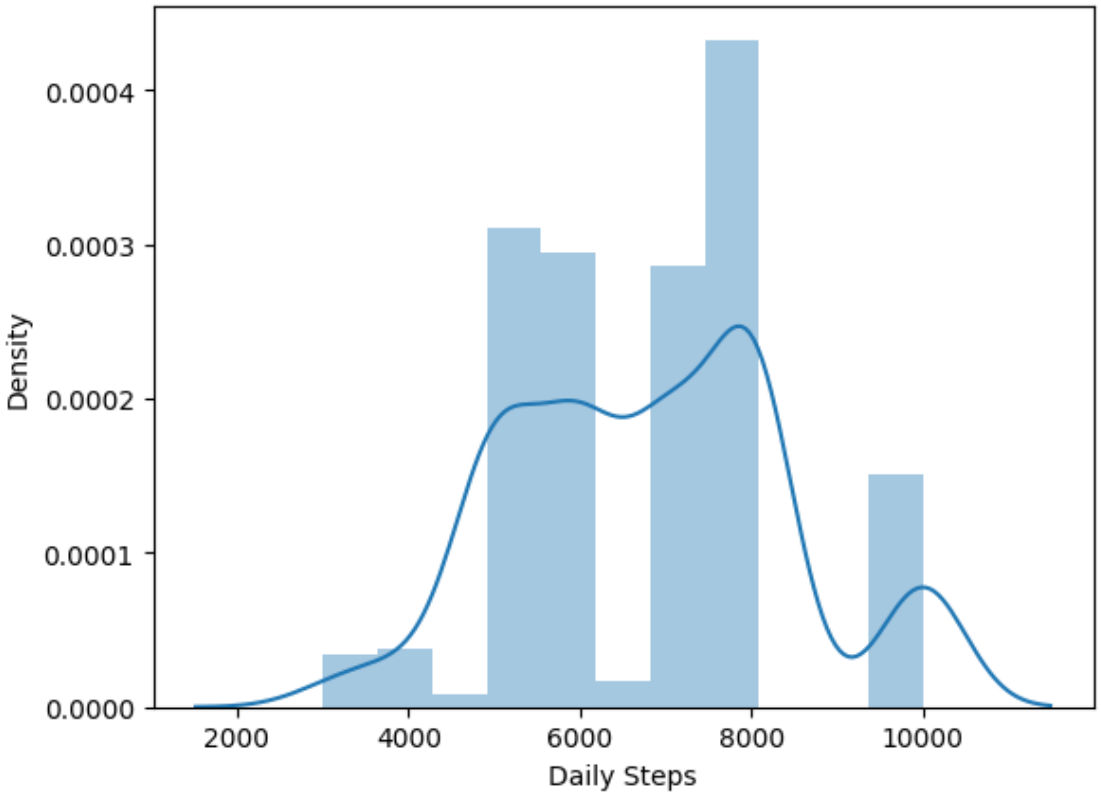
[21]: <seaborn.axisgrid.PairGrid at 0x7a76b3771ea0>



[22]:
```
sns.distplot(df1[ 'Daily Steps'])
```

```
<ipython-input-22-5d148120d475>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1[ 'Daily Steps'])
```

[22]: <Axes: xlabel='Daily Steps', ylabel='Density'>

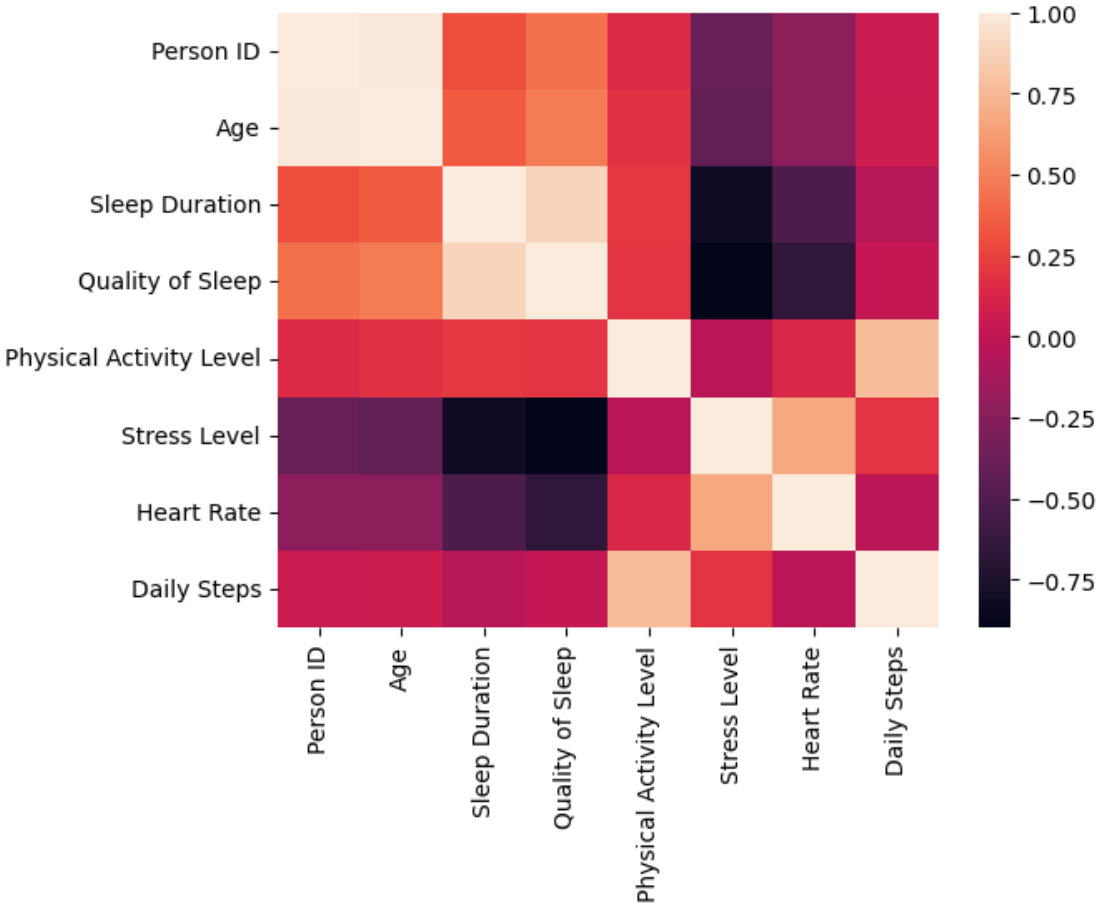


[23]:
```
sns.heatmap(df1.corr())
```

```
<ipython-input-23-3ed1a1a51dc0>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
```

```
sns.heatmap(df1.corr())
```

[23]: <Axes: >



3 TO TRAIN THE MODEL AND MODEL BULDING

[24]:
```
x=df[['Person ID', 'Age','Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level','Heart␣
 ↪Rate']]
y=df['Daily Steps']
```

[25]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

[26]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[26]:
```
LinearRegression()
```

[27]:
```
lr.intercept_
```

[27]:
```
12860.540704206087
```

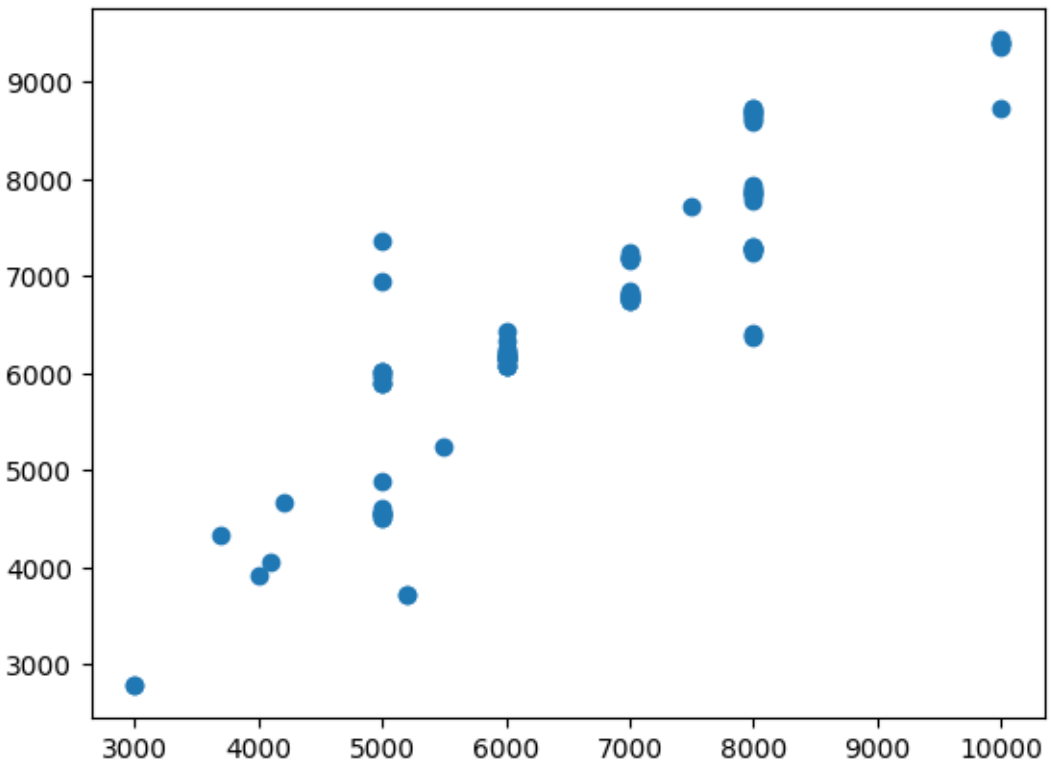
[28]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

[28]:

	Co-efficient
Person ID	-3.188181
Age	38.621466
Sleep Duration	-489.927731
Quality of Sleep	399.546768
Physical Activity Level	67.213478
Stress Level	561.840332
Heart Rate	-192.564330

[29]:
```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

[29]:
```
<matplotlib.collections.PathCollection at 0x7a76f6f6ffa0>
```



4 ACCURACY

[30]:
```
lr.score(x_test,y_test)
```

[30]: 0.8231836222067487

[31]:
```
lr.score(x_train,y_train)
```

[31]: 0.7961816944079384

[32]:
```
from sklearn.linear_model import Ridge,Lasso
```

[33]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

[33]: Ridge(alpha=10)

[34]:
```
rr.score(x_test,y_test)
```

[34]: 0.8252327501210341

[35]:
```
rr.score(x_train,y_train)
```

[35]: 0.7945713550041034

[36]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

[36]: Lasso(alpha=10)

[37]:
```
la.score(x_test,y_test)
```

[37]: 0.8246285070318662

[38]:
```
la.score(x_train,y_train)
```

[38]: 0.7950466633574356