CS-C4100 - Digital Health and Human Behavior project, Sleep analysis

I) Introduction

Northcube, Swedish company, created the mobile application *Sleep Cycle* in 2009. It acts as an automated alarm clock and tracker that analyzes sleep patterns and wakes you up during your lightest sleep period. Moreover it has a sound collection that includes music, meditations, and stories. They can help you relax your body, slow your breathing, or simply give some comfort.

The program listens to your noises and analyzes your sleep using its exclusive sound analysis technology. It correlates its findings with your habits, external circumstances, and everyday activities, assisting you in determining the positive and negative influences on your sleep. It awakens you up when you're in your lightest sleep phase and helps you wake up feeling refreshed and healed. The sleep study results are given as simple insights, graphs, and trends, with suggestions on how to adjust your habit to enhance your sleep.

The application uses a system of motion detection with the microphone to analyze movements and the accelerometer of the smartphone. Using theses and some sound filters, the application manages to categorize the different type of sleep that the subject is going through. It does tell a lot about the sleep quality in function of the time spend in each stage. Of course, the more time you spend in the deep sleep, the better your sleep quality will be.

Every person sleeps four to six cycles every night. There are many sleep phases in each sleep cycle. They all have a function in providing you with quality slumber, and the length of each sleep cycle fluctuates throughout the night. Sleep cycles vary from person to person. Sadly, we don’t have access on the time spent of each of theses stages and their types, so deeper analysis won’t be possible.

However, the data given by the application of the subject covers a relevant amount of data, which will greatly help us to make some complex models with machine learning. Missing information in some of the features is going to be an issue, but the strength of this database is its variety, so we will have the possibility to create predictors to complete the data.

Then, later we will be able to predict roughly the sleep quality of any day of our subject using the others features that we are given. By using the models we built with techniques such as clustering, dimension reduction, kernel methods, etc.

II) Problematic

Using this various amount of data, my goal will be to define any patterns and behaviors about the sleep of our subject compared to the average person. Then focus on which features are affecting the most its sleep quality, and then build a machine learning model around them.

III) Management of the data

Our data is a raw csv file from the Sleep Cycle application, it provides a lot of features about the studied person's health. We have information on each day from the 2014-12-29 to the 2018-02-16 (886 days in total) on the following features : **Start** and **End** (of sleep), **Sleep quality** (measured from the application), **Wake up** (provided by the user), **Sleep notes** (provided by the user), **Heart rate** (average during the night) and **Activity** (number of steps during the day).

```
df = pd.read_csv('sleepdata.csv',delimiter=";")
df
```

	Start	End	Sleep quality	Time in bed	Wake up	Sleep Notes	Heart rate	Activity (steps)
0	2014-12-29 22:57:49	2014-12-30 07:30:13	100%	8:32	:)	NaN	59.0	0
1	2014-12-30 21:17:50	2014-12-30 21:33:54	3%	0:16	:|	Stressful day	72.0	0

After a quick look at our data, we see that our data is incomplete, and that they are a lot of 0 values for activities steps, which seems impossible. Because theses values seem to be totally wrong, we have to replace them by a "NaN" value to keep consistency.

```
cols = ["Start","End","Sleep quality","Time in bed","Wake up","Sleep Notes"
df[cols] = df[cols].replace({'0':np.nan, 0:np.nan})

# Check the number of missing values in each column
df.isnull().sum()
```

```
Start                0
End                  0
Sleep quality        0
Time in bed          0
Wake up            641
Sleep Notes        235
Heart rate         725
Activity (steps)   418
dtype: int64
```

After this step, we notice that we're missing 76% of the data in Wake up, 27% in Sleep Notes, 83% in Heart rate and 48% in Activity. One of the most common strategy in this situation is simply dropping all the records with NaNs. But if we do that we our current data, not a single row will be remaining, so the thing we can do is taking this information for later. And, because theses features account for more than 5% the size of our dataset, so we should still keep them.

But we have to take this into account, since having that many missing values will greatly decrease the precision of our machine learning models in the next parts of the study. An article[1], (Ayilara, O.F., Zhang, L., Sajobi, T.T. *et al.,2019,* 106), estimated that when 50% of the data is missing but not at random, the bias reduction excelled 50 % and the RMSE (Root Mean Squared Error) was reduced by up to 45% with multiples features. Our conclusions on the accuracy should then not be aiming for a really good model, but just a decent one.

[1] Ayilara, O.F., Zhang, L., Sajobi, T.T. *et al.* (2019) Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry. *Health Qual Life Outcomes* 17, 106

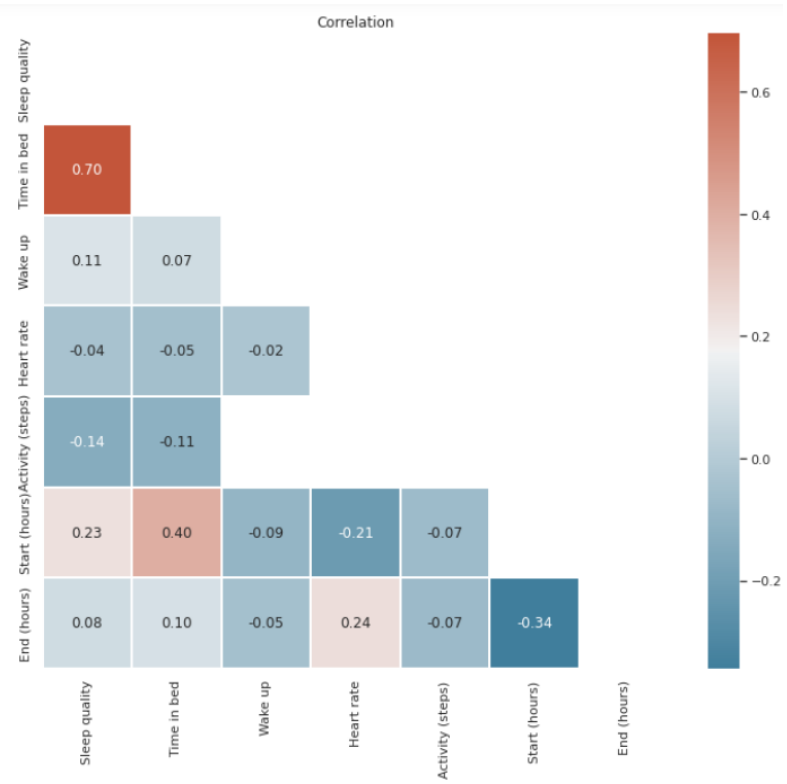
Now, we have to change types of some of our values to make them usable for models. First, **Wake up** values (by interpretation) to 1 (happy = ":)"), 0.5 (neutral =":|) and 0 (unhappy =":("), so that they're easier to handle with mean and standard deviation. Moreover, **start** and **End** time are timestamps so we cannot manipulate them, we should replace them and our current **Time in bed** which is a string by actual numbers which might be more consistent and can be used for the predictors. Moreover, **Sleep quality** is given as a string, so we'll have to convert it to int.

```
df['Wake up'] = df['Wake up'].replace({':)':1, ':|':0.5,':(':0})
df['Sleep quality'] = pd.Series([ val.replace("%","") for val in df['Sleep quality']]).astype(int)
```

Then we use the "Start" and "End" columns, to compute the total sleeping time of each row of our dataframe.

```
df['Start'] = pd.to_datetime(df['Start'])
df['End'] = pd.to_datetime(df['End'])
df['Time in bed'] = pd.Series(df['End'] - df['Start']).astype('timedelta64[s]')
df['Start (hours)'] = pd.Series([val.hour *3600 + val.minute *60 + val.second
                                 for val in df['Start']])/3600
df['End (hours)'] = pd.Series([val.hour *3600 + val.minute *60 + val.second
                               for val in df['End']])/3600
```

Now, we can compute our correlation matrix, using the features that we manage to extrapolate from our initial data :



Sleep quality is correlated the most with **Time in bed**, then **Start (hours)** and finally **Wake up** and **End (hours)**. Moreover, we can observe that "Activity (steps)" and "Sleep quality" are not correlated at all, despite the general belief that some exercice help for a

better sleep quality. Study on Sleep health[2] made the conjecture that some physical exercice every helps for increasing sleep quality but not to predict the duration of sleep. Each adult should be aiming for at at least 10 000 steps per day to have impact on their sleep, but our current individual walked more than 10 000 steps only 38 days out of the 450 we have data on. Which is 8% of the time, so too rarely to have a positive relation out of it. Moreover, strange observation we can make is that **Activity** is currently inversely correlated to **Sleep quality**, meaning that this actually has a negative impact on it.

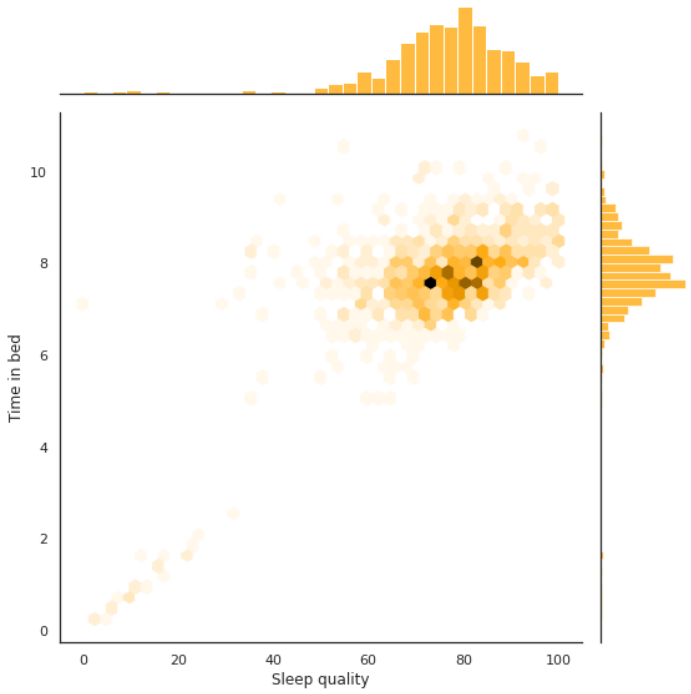
Now, we can start building our predictors, before trying to analyze some incomplete data :

```
from sklearn.impute import SimpleImputer

# Define the imputer. We use 'mean' as the imputing strategy.
imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
# We will continue our work on a copy of the original data and delete 'Sleep Notes'
data_preprocessed = df.copy()
# predict column wake up
data_preprocessed['Wake up'] = imp_mean.fit_transform(data_preprocessed[['Wake up']])
# predict column activity steps
data_preprocessed['Activity (steps)'] = imp_mean.fit_transform(data_preprocessed[['Activity (steps)']])
# predict column heart rate
data_preprocessed['Heart rate'] = imp_mean.fit_transform(data_preprocessed[['Heart rate']])
data_preprocessed.head()
```

IV) Analysis of the data

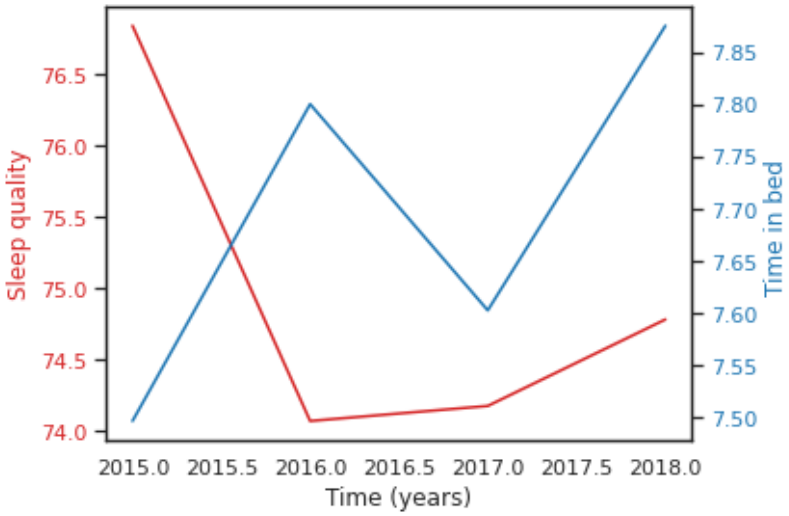
Now, we will try to plot and play with our data to extract some observations on the subject. First, we build a join plot of our features **Sleep quality** and **Time in bed** to observe how both evolve regarding each other.



[2]Blagrove M, Owens DS, MacDonald I, Sytnik N, Tucker P, Folkard S. (1998 Dec) Time of day effects in, and the relationship between, sleep quality and movement. J Sleep Res.;7(4):233-9. doi: 10.1046/j.1365-2869.1998.00119.x. PMID: 9844849.

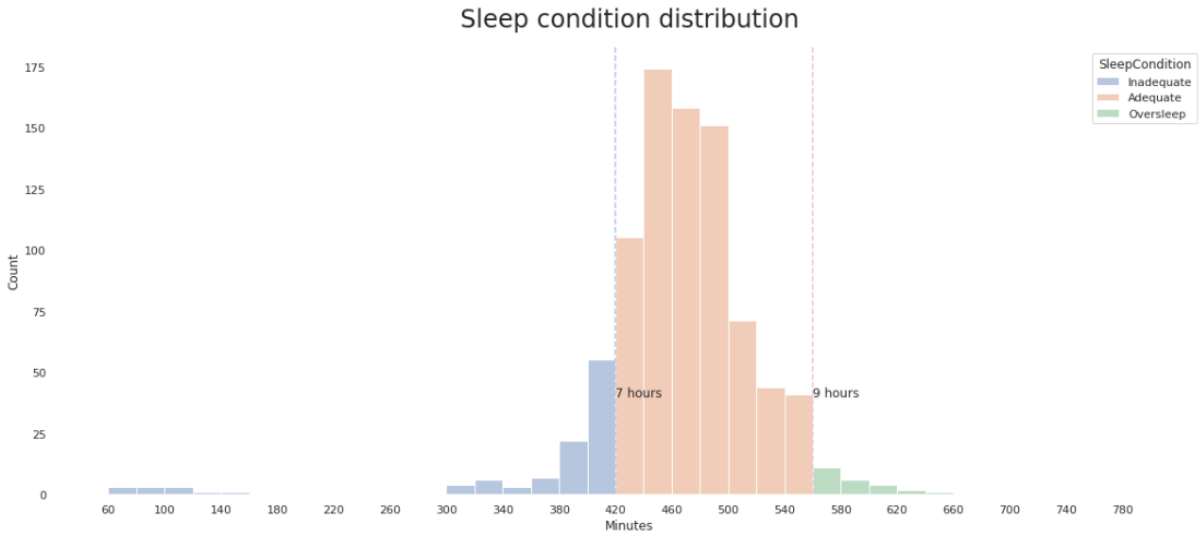
As we can see, their repartition is similar, as we expected from the correlation. Moreover, we can also see two peaks in proportion with one which is near the average of both, at quality of 74.9% and 7.65 hour of sleep. And the other which is more off center at 82.3% of quality and 8.05 hour of sleep.

With data over more than 4 years, we can ask ourselves if theses data changed over the years. Did the sleep quality and hours of sleep took a turn over time ? So I made different bins according the years and plotted the results on a twin graph.



Sadly, the changes are not tremendous; over 4 years, the sleep quality only dropped by roughly 2% and the time in bed increased of around 21 minutes. So, because the evolutions that we can observe throughout the years are not that relevant, so we cannot make any assumptions.

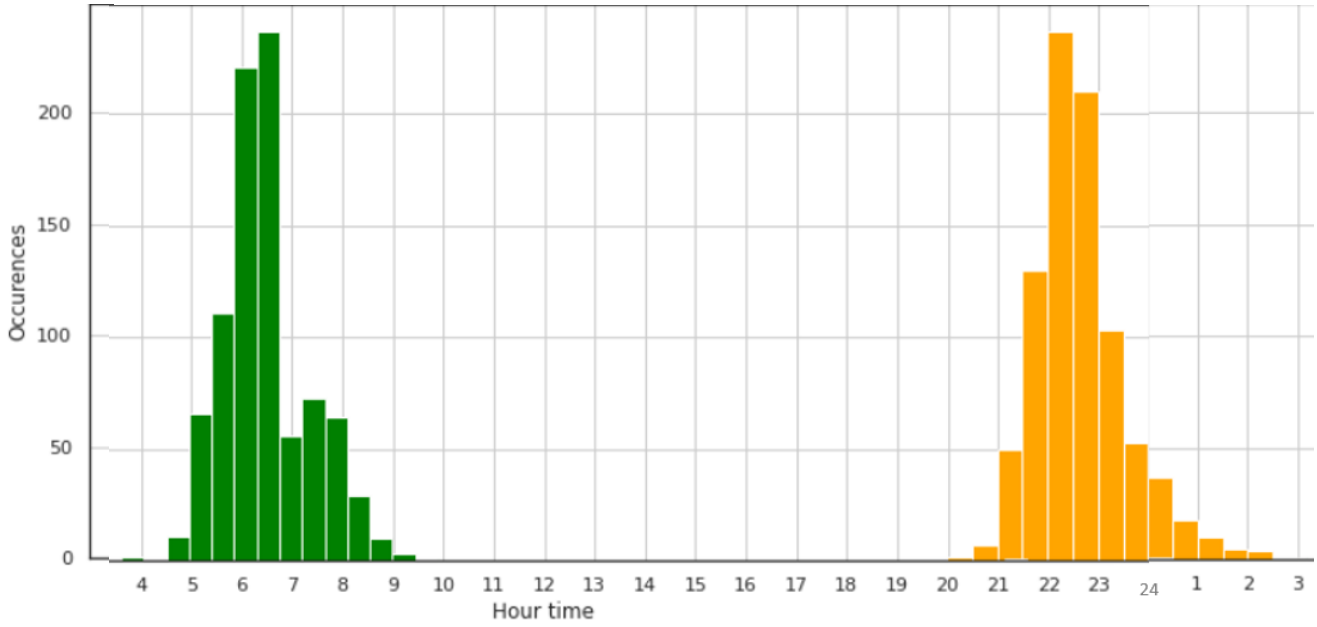
The following plot is composed with 3 bins, that are based on the number of hours our subject is sleeping during theses years. They're categorized in 3 bins : **Oversleep** if the sleep duration is > 9 hours, **Adequate** if the sleep duration lies between 7 hours (excluded) and 9



hours (included) and **Inadequate** if the sleep duration is ≤ 7 hours . Theses bins are based on a medicine study[3], (Watson NF, Badr MS, Belenky G, Bliwise DL, Buxton OM, Buysse D, Dinges DF, Gangwisch J, Grandner MA, Kushida C, Malhotra RK, Martin JL, Patel SR, Quan SF, Tasali E.(2015 Jun 1;38(6))), leading to the consensus that an adult needs between 7 and 9 hours an appropriate sleeping time. Opposing the conventional wisdom that it should be between 6 and 8 hours.

Looking at this data, we can see that the person studied has most of his time spent an adequate time of sleep (80% of the time from calculations) and the average is closer to the 8 hours boundary than the 7 hour or 9 hour boundary. So, we can make the assumption that this person is actually a good model of time in bed according to the article.

Next, we will plot about the proportion of occurrences at which our subject wakes up and goes to sleep, to see if the subject neglects his sleep schedule on the behalf of his sleeping time.



By looking at the graph, we can see that the bins repartition of waking up and going to sleep are similar. So we can make the assumption that the hours of sleep stay almost consistent. But we can see as well that there is an inconsistency at the 7 hour bin, which should be greater than 8 and 9 by the repartition of start sleep time. It is possible that this is due to an alarm that wakes up our subject around 6 for work for any time he went sleep, and he rather stay in bed until 8 or 9 when he's not forced of waking up (for a better rest).

```
dfbefore = df[df['End (hours)'] < 7]
dfafter = df[df['End (hours)'] > 7]

print(dfbefore['Sleep quality'].mean())
print(dfafter['Sleep quality'].mean())
print(dfafter['Sleep quality'].mean() - dfbefore['Sleep quality'].mean())
```

```
73.67899408284023
78.7914691943128
5.112475111472563
```

[3] Watson NF, Badr MS, Belenky G, Bliwise DL, Buxton OM, Buysse D, Dinges DF, Gangwisch J, Grandner MA, Kushida C, Malhotra RK, Martin JL, Patel SR, Quan SF, Tasali E. (2015 Jun 1)Recommended Amount of Sleep for a Healthy Adult: A Joint Consensus Statement of the American Academy of Sleep Medicine and Sleep Research Society. Sleep;38(6):843-4. doi: 10.5665/sleep.4716. PMID: 26039963; PMCID: PMC4434546.

From the calculations we can see that our theory on the studied person might be true, he actually has a better sleep quality when he wakes up after 7 am (by 5.1 points on average).

V) Machine learning to predict sleep quality

Now, we will try to train our data with machine learning to create some models to predict the "Sleep quality" given other features. For this task, we will use the most relevant features of our data to train the models, so the most correlated ones with the "Sleep quality". Theses are the "Wake up", "Start (hours)", "End (hours)", "Time in bed" (and Activity (steps), which is inversly correlated to "Sleep quality", so can be quite useful).

First we normalize our data before creating any model, it is relevant because all of our data do not share the same range. Then we will try to process our data with the first model we choose, we will start with logistic regression :

```
clf = LogisticRegression()
clf.fit(train_x, train_y)
print('Accuracy for logistic regression model is :', clf.score(test_x,test_y))
```

```
Accuracy for logistic regression model is : 0.04504504504504504
```

The score is pretty low, so this model is not really useful for predictions.

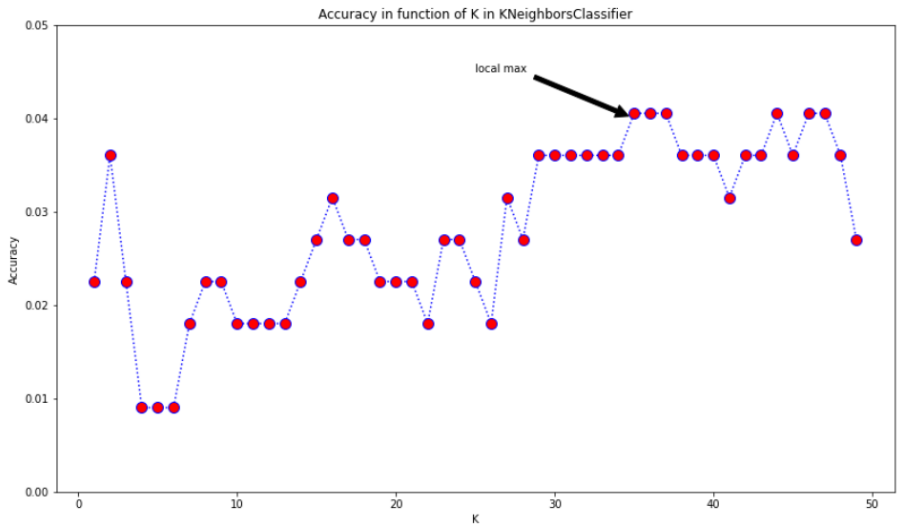
Next, we do the linear regression :

```
clf2 = LinearRegression()
clf2.fit(train_x, train_y)
print('Accuracy for linear regression model is :', clf2.score(test_x,test_y))
```

```
Accuracy for linear regression model is : 0.445295700473054
```

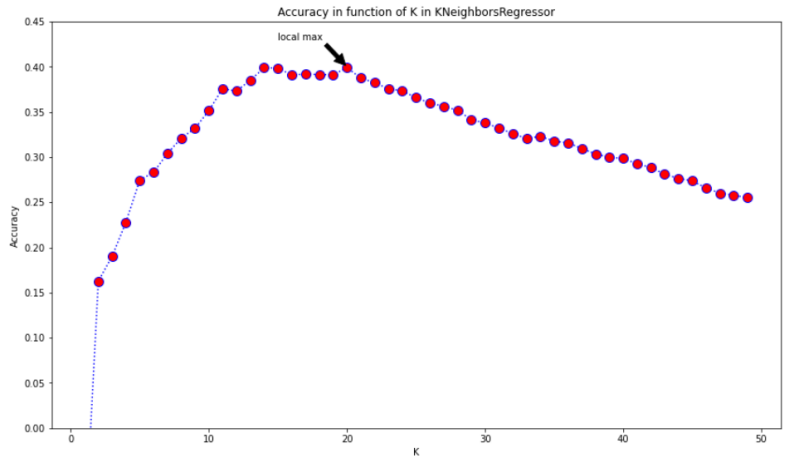
The score is good, but can still be improved, so we keep trying others models. This is already very satisfactory knowing the amount of missing data that we're struggling with. For the next model, we will use some clustering, with the KNN (K-nearest neighbors), we will try with different 'K'(neighbors) to see with which one will give us the best accuracy and choose it.

```
The maximum accuracy for KNN is obtained for 35 neighbors at : 0.04054054054054054
```



In a same manner as with the logistic regression, the score is bad so the model poorly predicts the sleep quality. But there are others methods using clustering, we will try to use as well KNeighborsRegressor and RadiusNeighborsRegressor in similar ways than KNN, theses might be promising after looking at the results from the logistic regresion model. Let's try the KNeighborsRegressor model :

```
The maximum accuracy for KNR is obtained for 20 neighbors at : 0.39893419173659916
```



And now the Radius Neighbors Regressor model :

```
The maximum accuracy for RNR is obtained for radius 0.2564102564102564 at : 0.4210477280834568
```

This new model is working really well, the results are good. Sadly, plotting is complicated because of the linspace range.

Now, we will try to have best results using the Baies Naive Methods, determines the probability of each feature set and uses that to determine the probability of the classification itself:

```
gauss = GaussianNB()
multi = MultinomialNB()
ypred = model.predict(test_x)
multiNaive = multi.fit(train_x,train_y)
print("The accuracy of Naive Bayes gaussian is",accuracy_score(test_y, ypred),
      "and for multinomial :",multiNaive.score(test_x,test_y))
```

```
The accuracy of Naive Bayes gaussian is 0.02252252252252252 and for multinomial : 0.03153153153153153
```

Theses methods are specialized in large datasets, so our is too small to have high quality precision there.

Next, we're using Stochastic Gradient Descent next, which is popular in the neural network world, where it’s used to optimize the cost function :

```
clfSGDC = SGDClassifier()

clfSGDC.fit(train_x, train_y)

print(" Accuracy for StochasticGradient Descent Classifier :", clfSGDC.score(test_x, test_y), "%")
```

```
Accuracy for StochasticGradient Descent Classifier : 0.018018018018018018 %
```

And finally for the last one we will use a Random Forest Classifier, which is averaging the predictions from all of the trees, resulting in better performance than any single tree in the model :

```
clfRDC = RandomForestClassifier()

clfRDC.fit(train_x, train_y)

print(" Accuracy for StochasticGradient Descent Classifier :", clfRDC.score(test_x, test_y), "%")
```

```
Accuracy for StochasticGradient Descent Classifier : 0.036036036036036036 %
```

We now have some results we can compare, having multiples models and accuracy :

```
Accuracy for logistic regression model is : 0.04504504504504504
Accuracy for linear regression model is : 0.44529570047305356
The maximum accuracy for KNN is obtained for 35 neighbors at : 0.04054054054054054
The maximum accuracy for KNR is obtained for 20 neighbors at : 0.39893419173659916
The maximum accuracy for RNR is obtained for radius 0.2564102564102564 at : 0.4210477280834568
The accuracy of Naive Bayes gaussian is 0.02252252252252252 and for multinomial : 0.03153153153153153
Accuracy for StochasticGradient Descent Classifier : 0.018018018018018018 %
Accuracy for Random Forest Classifier : 0.036036036036036036 %
```

So, we can see that the best machine learning models for our data are regression models, namely : the Linear Regression model first, then the Radius Neighbors Regressor model and finally the K Neighbors Regressor model. An accuracy of 45% is very satisfactory for our model, by taking into account the data from the first article we looked into (RMSE was reduced by up to 45% with multiples features). But a big weakness in our dataset is the fact that it's actually really small and missing a way too important proportion of information, leaving our models in bad states for predictions.

VI) Conclusions and discussion

In conclusion, we managed to notice some patterns in our subject's behavior. For example, the lack of activity steps and the fact that it doesn't affect the sleep quality that much. Moreover, we could see that his sleep quality and time in bed wouldn't really shift during the years, he was consistent at a pretty good rate.

In fact, he was getting a good amount of sleep 4 out of 5 nights and rarely overslept. However, because of that consistency, his sleep schedule was slightly affected; his bed time was spread on a range of 2 hours, whereas the start and end time of sleep would be spread on roughly a 5 hours range.

Finally, we could make the assumption that our subject was either a student or already working by looking at the proportion of his sleep schedule. Often waking up around 6 am at an abnormal higher rate in comparison to the next hours, and getting a worst sleep when doing so.

The "Wake up" feature can play a huge role in the sleep quality on that term, because the person would be in a worst mood by waking up for work. Then it would be less objective on the actual energy that the person feels at the end of their sleep, feeling weaker even if they were in the same physical state later on the morning.

Talking about limitations, this study posses a lot of them in different stages. First, one major problem is the lack of information on the subject: age, gender, occupation. With

theses, it would be possible to have better understanding of the “Time in bed” needed on average for a person like that, or if the “Heart rate” is actually normal during the sleep.

Moreover, the lack of data is way too important, at a point that creating a model feels meaningless, to aim for an accuracy of 50%. The person should have given more efforts to provide data, so we can have a better analysis of its sleeping behaviors.

References :

- Blagrove M, Owens DS, MacDonald I, Sytnik N, Tucker P, Folkard S. (1998 Dec) Time of day effects in, and the relationship between, sleep quality and movement. J Sleep Res.;7(4):233-9. doi: 10.1046/j.1365-2869.1998.00119.x. PMID: 9844849.

- Ayilara, O.F., Zhang, L., Sajobi, T.T. et al. (2019) Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry. Health Qual Life Outcomes 17, 106

- Watson NF, Badr MS, Belenky G, Bliwise DL, Buxton OM, Buysse D, Dinges DF, Gangwisch J, Grandner MA, Kushida C, Malhotra RK, Martin JL, Patel SR, Quan SF, Tasali E. (2015 Jun 1) Recommended Amount of Sleep for a Healthy Adult: A Joint Consensus Statement of the American Academy of Sleep Medicine and Sleep Research Society. Sleep;38(6):843-4. doi: 10.5665/sleep.4716. PMID: 26039963; PMCID: PMC4434546.

- Alycia N. Sullivan Bisson, Stephanie A. Robinson, Margie E. Lachman (2019) Walk to a better night of sleep: testing the relationship between physical activity and sleep,Sleep Health,Volume 5, Issue 5, Pages 487-494,ISSN 2352-7218.

- Hespanhol Junior, L.C., Pillay, J.D., van Mechelen, W. et al. (2015) Meta-Analyses of the Effects of Habitual Running on Indices of Health in Physically Inactive Adults. Sports Med 45, 1455–1468

- Dr. Michael Breus (December 13, 2022), The Benefits of Exercise For Sleep