
Essay on the contribution of machine learning techniques to the detection of sleep disorders

M. Noiret, C. Yahiaoui

Under the direction of S. Chareyron

Table of contents

Introduction	3
Review of the literature	4
Health and economy	4
Sleep and health.....	5
A.I. applied to sleep disorders treatments	6
Empirical analysis	7
Data used in the study	7
Summary statistics of the dataset	8
The model.....	13
Metrics and scoring methods.....	13
Model selection	14
Application	18
Discussion	19
Conclusion.....	20
Bibliography.....	21
Appendix	24
Code	25

Introduction

If you're having trouble sleeping, you're not alone. For almost 70% of French people, sleep is a recurring source of problem. Since 1974, French adults have gone from an average of 8.5 hours sleep per night to less than 7 hours in 2024 (Odoxa, 2024). People in France are sleeping less and less. This observation is the result of a major change in our society. Firstly, more and more French people are working during nights. On average, this change in working patterns means that workers lose 1 hour of sleep compared to working during the day. Moreover, the average French person lives further away from their workplace. This distance is the cause of the development and expansion of our metropolises, where most jobs are concentrated. In the private sphere, the omnipresence of screens is the main cause of the reduction in sleep time. There is, of course, the time spent on smartphones, but also the recent emergence of numerous entertainment platforms, like Netflix or YouTube. Finally, another major recognized source of sleep loss is environmental pollution. This mainly includes noise and light pollution, but also air pollution and global warming, all of which have a negative impact on sleep quality.

We decided to choose this subject because there are a lot of sleep-related problems. Sleep is a common denominator for all human beings, and it is essential for us to take an interest in it, as it represents on average a third of our entire existence. In this essay, we will be looking more specifically at sleep disorders. These can be of various types and are defined by the Sleep Foundation as *“conditions that affect sleep quality, timing, or duration and impact a person’s ability to properly function while they are awake. These disorders can contribute to other medical problems, and some may also be symptoms for underlying mental health issues.”*

There are three main classifications of sleep disorders. The first category is dyssomnia. Dyssomnia is characterized by problems affecting sleep time and/or quality. This category includes insomnia, narcolepsy, hypersomnia, and disorders linked to jet lag. The second class of sleep disorders is called parasomnia. This class includes all disorders that affect the sleeper's behaviour during the night, without having any impact on the waking state. The most recurrent parasomnias in the population are bruxism (impact on teeth), sleepwalking, sleep apnea, sleep paralysis and Rapid Eye Movement (REM) sleep behaviour disorders. Finally, the last recognized sleep disorder is sleeping problems linked to psychiatric problems. This category includes all sleep disorders caused by mental health problems such as depression.

Most of these disorders are recurrent in the people affected, although they are not always conscious of them. That is why it is so important to detect these disorders in order to know what the patient's problem is, so that solutions can be found. According to the French National Institute of Health, one person out of three suffers from sleep disorders. The most common sleep disorder in France is insomnia. Even though this disorder is often not diagnosed by specialists, it is estimated that one out of five people in France suffers from insomnia. In most cases, the disorder is caused by anxiety or stress, but it also occurs regularly with Alzheimer's or Parkinson's disease. Insomnia is characterized by difficulties in falling asleep and/or waking up, as well as night-time awakenings. The consequences of this disorder are mainly daytime tiredness and concentration problems. In addition, various studies have shown that recurrent

insomnia also increases the risk of developing more serious health problems such as depression or cognitive impairment. Another sleep disorder affecting 4% of the population is sleep apnea. The main symptoms of this sleep disorder are snoring, pauses in breathing and waking up during the night. Its consequences can be dramatic since severe sleep apnea has serious consequences for the cardiovascular system. There are a wide variety of sleep disorders and, whatever their type, they all impair sleep quality and have a lasting effect on the patient's health.

The aim of our thesis is to provide an answer to this problem. How can the use of AI help healthcare professionals detect sleep disorders? By using data containing information on sleep disorders, we aim to improve their detection. We hope to help both doctors and people who are worried about their sleep to make an initial diagnosis of the presence or absence of a sleep disorder. A 1999 study by V. Kapur *et al.* on the cost of undiagnosed sleep apnea in the US found that the estimated cost to the medical system was an additional \$3.4 billion. Another 2006 study in Australia, by D R Hillman *et al.*, calculated that the total direct and indirect costs of sleep disorders were between 2 and 3% of Australia's GDP per year. Finally, a recent paper (2022, L Borsoi *et al.*) identified 26 clinical and non-clinical conditions (such as diabetes or car accidents) influenced by OSA in Italy. These conditions were responsible for an estimated economic loss of between 10.7 and 32 billion euros per year in Italy. The cost of reduced quality of life due to undertreatment of OSA was estimated by the authors to be between 2.8 and 9.0 billion euros per year, which is much higher than the current cost of diagnosing and treating OSA (234 million euros per year). In addition, our study will also enable us to identify the physical, behavioural, and social factors linked to sleep disorders. Our study will be organized as follows: a review of the literature, a description of the data and the model, the results and finally our conclusions.

Review of the literature

Health and economy

Health is a part of human life that economists did not initially consider. The first modern articles on the contribution of human capital to the economy were written in the 1960s. Gary Becker, a famous economist, wrote a theory of human capital in 1964, based on a theory written by Schultz 3 years earlier. In his theory, Becker defined this factor of production as all the physical and intellectual abilities of the labour force that are conducive to economic activity. From now on, investment in labour, and not just in equipment, became a must. Public policy was therefore influenced to invest, for example, in education to enable the workforce to take on more difficult tasks. We had to wait until 1972 and Grossman's article "*On the Concept of Health Capital and the Demand for Health*" to consider health as an important part of the workforce. In his theory, the author explains that each person is born with a certain capital of health that depreciates over the years of his life. According to Grossman, this depreciation can be slowed down by limiting certain risky behaviours. For example, the economist points out that a person's

diet, physical activity or access to medical services have a major impact on their productivity. Based on his assumptions, a large number of economists began to analyse the relationship between health and the economy from all angles. As a result, health has become a pillar of economic theory.

Sleep and health

Now that we have seen how health is linked to the economy, let us focus a little more on our subject: sleep. As human beings, we all know that sleep is essential for good functioning and that it is important to give ourselves enough time to rest. However, the vast majority of French people ignore these recommendations and sleep less than they should. We can imagine that sleep deprivation has only a minor impact on our daily lives and does not affect our health that much. We will try to show you why it is a very bad idea not to listen to your body. First, we will look at the results of some studies that have looked at the effects of sleep duration on health.

According to Santé Publique France in 2019, sleeping less than 6 hours a night increases the risk of developing diabetes, hypertension, heart disease, accidents, disrupts family relationships and quality of life and work. These significant results on the French population give us a first idea of the importance of sleep for health. The consequences of sleep deprivation are not just a little tiredness during the day, but an increased risk of developing diseases. Furthermore, a 2024 Odoxa study showed that chronic sleep deprivation can have dangerous consequences, such as a 28% increased risk of developing type 2 diabetes, an increased risk of cardiovascular disease, an increased risk of dementia and an increased risk of premature mortality. This study confirmed the findings of the Santé Publique France study but went further and found that not getting enough sleep can be a factor in serious illnesses. Now the risk appears to be very serious. If you do not give your body enough time to rest in a systematic way, you will damage your pancreas, your heart and your brain. The results are basically that you damage 3 vital organs out of 7, leaving your lungs, liver, kidneys and skin for the time being.

Cappuccio et al found similar results in 2010 and 2011. In their 2010 study they showed that sleep duration influences mortality from various causes, while in their 2011 paper they found a link with our cardiovascular system. Donald L Bliwise, a specialist in sleep medicine, analysed the effects of both short and long sleep duration in 2007. This specialist highlighted that not sleeping enough is a problem, we already know that, but sleeping too much is also a problem. He demonstrated the existence of a U-shaped relationship between health problems and sleep duration. We can summarize this paragraph by saying that a lack of sleep, or too much sleep, in the long term causes cardiovascular diseases, cerebral diseases and damages your pancreas and your general health.

A.I. applied to sleep disorders treatments

We have seen in the last paragraph that sleep is highly correlated with health; we will now focus on the spectrum of our sleep that we will study more specifically today: sleep disorders. Before focusing on previous studies similar to ours in their use of A.I. methods to treat sleep disorders, we'll look at some of the causes and consequences of these disorders.

Several previous studies, such as Bianchi 2013, have highlighted that sleep disorders are associated with other neurological disorders. More specifically, over the years, authors have found that sleep disorders are associated with neuromuscular diseases (*Culebras, 2008*), neurodegenerative diseases (*Chokroverty, 2009*) or neurocognitive diseases (*Lal et al., 2012*). In most cases, the various authors have focused on sleep apnea, as it is the most common disorder that is recognised to have a significant negative impact on human health. Firstly, what are the factors that favour the development of sleep apnea? Bianchi argued in his 2013 paper that obesity, older age, male sex, obstructed airways and cerebrovascular disease are some of the main factors that cause the appearance of sleep apnea. What are the different health effects of sleep apnea? Two studies from 2010, one by *Budhiraja et al.* and the other by Johnson et al, found that sleep apnea can cause cardiovascular problems, cerebrovascular morbidity and mortality, and in Johnson's case, strokes. The latter finding was confirmed by a 2012 study by Munoz et al in older people: *Central sleep apnea is associated with increased risk of ischemic stroke in the elderly*. We are now certain, based on all the literature we have studied, that sleep disorders are health problems that need to be addressed because they can be dangerous or even deadly for the people who suffer from them.

Now, let's focus on some studies who used AI in the service of health, and more specifically for sleep disorders. The first essay that we will analyse the most is *Brief digital sleep questionnaire powered by machine learning prediction models identifies common sleep disorders* by Schwartz et al in 2020 in Sleep Medicine. For their study, they first asked for participants on a social media site and then filtered them all. In the end, they had 3799 people in the first step and ended up with a sample of only 247 participants. Schwartz and his colleagues used ElasticNet models with lasso and ridge regularization, but also cross-validation and area under the ROC curve (AUC) to verify their results. If we focus on the variables they had, we can see that they first used common variables such as: gender, marital status, employment, age, BMI, but also variables on ethnicity, race, neck, waist and hip circumference. Regarding the variables related to sleep, their questionnaire asked participants: the amount of sleep needed, sleep duration during weekdays and weekends, sleep debt, insomnia, pauses in breathing, snoring, preference for morning or evening, and scores on the Functional Outcomes of Sleep Questionnaire (FOSQ) and the Epworth Sleepiness Scale (ESS), which are two questionnaires that evaluate the effects of your sleep on your life. In addition, they had questions that were common with other studies like variables on mental health or awakenings for example. Using their ElasticNet algorithm, the authors found the most important variables for predicting the two main sleep disorders, insomnia and sleep apnea. For insomnia, they included early morning and nocturnal awakenings, as well as snoring, as variables not to be missed. The model also included a variable for anxiety as an important variable, which shows us that mental health is very important in this type of study. For sleep apnea, the most important variables were snoring, ease

of waking up on weekday mornings, early morning awakenings and likelihood of falling asleep in a passive state. So, from these results we can see which variables we want to look at because they give a lot of information about sleep disorders.

To get a better idea of the machine learning techniques used to predict sleep disorders, we looked at the paper by *Gonzalo César Gutiérrez-Tobal et al.* in 2021, which summarized and analysed the publications on sleep apnea. Of the 19 studies they selected between 2004 and 2021, only 3 used models based only on non-biomedical variables. The vast majority of studies used medical data, such as blood oxygen saturation variables or physiological data, to perform their machine learning studies. The main machine learning methods used were logistic regression, multilayer perceptron, support vector machine, AdaBoost and CNN, with good results. For the validation part, the majority of the 19 studies used 2 or 3 subgroup tests, cross validation and bootstrapping.

With the help of these articles, we now have a better idea of what it might be interesting to try in our study in terms of the variables we can include, the machine learning method we can use, and the validation techniques that are commonly used.

Empirical analysis

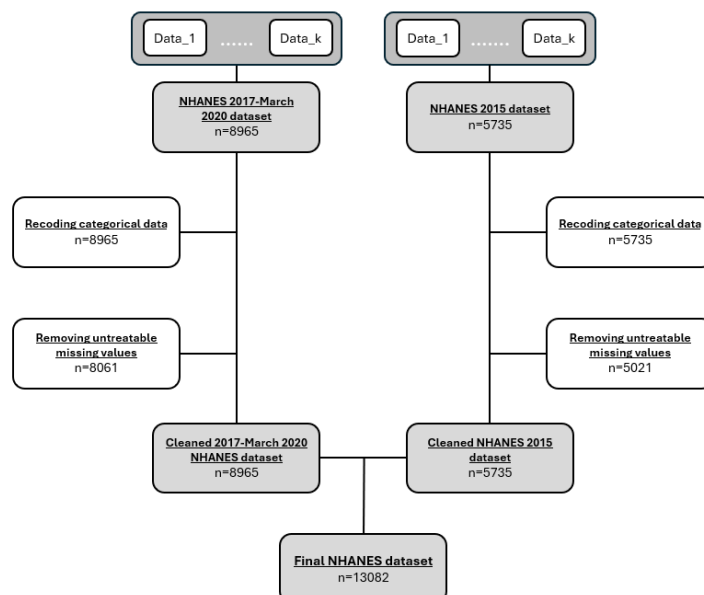
Data used in the study

The data used in this study comes from the National Health And Nutrition Examination Survey (NHANES) published by the Center of Disease Control and Prevention (CSC), a national health institute in the United States, which goal is to maintain a good level of public health and safety by using prevention and control of diseases, injuries and disabilities among the population. In this essay, two versions of this study are used: the pre pandemic examination data covering the 2017-March 2020 period and another covering the 2015-2016 period. The datasets are split into several tables, one for a specific topic. Both datasets are made to be representative of the United States population.

The Pre-pandemic version of the NHANES was sent to 27 066 people throughout the United States and 15 560 people accepted to respond. Among the respondents, only 14 300 people accepted to be examined. On the other side, the 2015-2016 version of the NHANES was sent to 15 327 people and only 9 971 people accepted to respond. From here, only 9 544 of them accepted to be examined. After merging the tables containing the needed variables, all the categorical variables were encoded. Once all the variables were cleaned (categorical variables, missing values, etc.), the final dataset contained 8608 observations and 113 variables for the pre-pandemic version the NHANES dataset, and 5735 observations and 113 variables for the 2015-2016 version. The variables cover different aspects of life of the respondents, such as personal description (age, gender, education, BMI, etc.), socio-economic situation (income, marital status, etc.), health conditions (physical health evaluation, mental health, physical activity level, sleep quality, etc.) and more.

This thesis aims for predicting whether or not a person suffers from sleep disorders. In the dataset, the variable « SLQ050 » seems to be a good target variable for prediction, since it indicates if a person went to see a doctor for sleeping disorders.

Figure 1: Data preprocessing diagram



Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 13082 observations.

Summary statistics of the dataset

The following table contains the summary statistics of the dataset used in the analysis:

Table 1 : Summary Statistics

Variable	Mean	Std
Gender		
- Female	0.509	0.500
- Male	0.491	0.500
Education		
- College degree	0.305	0.460
- College graduate	0.239	0.427
- No diploma	0.187	0.390
- Highschool	0.222	0.416
- Missing	0.047	0.212
Poverty		
- Ratio <1	0.180	0.384
- Ratio <2	0.235	0.424
- Ratio <3	0.147	0.354
- Ratio <4	0.092	0.289
- Ratio <5	0.072	0.258
- Ratio >=5	0.156	0.363
- Missing	0.118	0.322

<u>High cholesterol</u>		
- No	0.654	0.476
- Yes	0.340	0.474
- Missing	0.006	0.078
<u>Nutrition quality</u>		
- Excellent	0.076	0.264
- Very good	0.192	0.394
- Good	0.397	0.489
- Fair	0.264	0.441
- Poor	0.071	0.258
<u>Health insurance</u>		
- No	0.166	0.372
- Yes	0.831	0.374
- Missing	0.002	0.048
<u>Health condition</u>		
- Excellent	0.116	0.321
- Very good	0.271	0.445
- Good	0.372	0.483
- Fair	0.200	0.400
- Poor	0.040	0.197
<u>Mental Health problems</u>		
- No	0.901	0.298
- Yes	0.098	0.298
- Missing	0.001	0.017
<u>Intense Physical Activity</u>		
- No	0.759	0.428
- Yes	0.241	0.428
- Missing	0.000	0.020
<u>Moderate Physical Intensity</u>		
- No	0.574	0.495
- Yes	0.425	0.494
- Missing	0.001	0.023
<u>Sleep disorder</u>		
- No	0.721	0.448
- Yes	0.278	0.448
- Missing	0.000	0.020
<u>Smoking</u>		
- No	0.821	0.383
- Yes	0.179	0.383
<u>Diabetes</u>		
- No	0.855	0.352
- Yes	0.145	0.352
- Missing	0.001	0.023
<u>Fast Food Frequency</u>		
- Extreme	0.239	0.426
- Often	0.353	0.478
- Rare	0.214	0.410
- Very often	0.195	0.396
<u>Pees during night</u>		
- Never	0.261	0.439
- 1 times	0.359	0.480
- 2 times	0.184	0.387
- 3 times or more	0.196	0.397
<u>Asthma</u>		
- No	0.840	0.367
- Yes	0.160	0.366
- Missing	0.001	0.025

<u>Heart disease</u>		
- No	0.918	0.275
- Yes	0.034	0.181
- Missing	0.048	0.214
<u>Corona disease</u>		
- No	0.909	0.287
- Yes	0.041	0.198
- Missing	0.050	0.218
<u>Ever had heart attack</u>		
- No	0.91	0.286
- Yes	0.042	0.201
- Missing	0.048	0.213
<u>Ever had CVA</u>		
- No	0.912	0.284
- Yes	0.040	0.197
- Missing	0.048	0.213
<u>Ever had thyroid</u>		
- No	0.843	0.364
- Yes	0.109	0.311
- Missing	0.048	0.214
<u>Ever had cancer</u>		
- No	0.854	0.353
- Yes	0.099	0.299
- Missing	0.047	0.212
<u>Depression</u>		
- Missing	0.001	0.025
- Never	0.753	0.431
- Sometimes	0.173	0.378
- Very often	0.074	0.261
<u>Attention trouble</u>		
- Missing	0.001	0.026
- Never	0.832	0.374
- Sometimes	0.105	0.306
- Very often	0.062	0.242
<u>Dangerous thoughts</u>		
- Missing	0.001	0.032
- Never	0.962	0.191
- Sometimes	0.026	0.159
- Very often	0.011	0.105
<u>Stop breathing during sleep</u>		
- Often	0.060	0.238
- Rarely	0.763	0.425
- Sometimes	0.126	0.331
- Very often	0.051	0.220
<u>Age</u>	48.784	18.356
<u>Sedentarity</u>	5.817	3.355
<u>Doctor visits</u>	2.379	1.996
<u>BMI</u>	29.778	7.434
<u>Average sleep hours</u>	7.634	1.613

Source: NHANES (2017-March 2020 and 2015-2016), CDC.

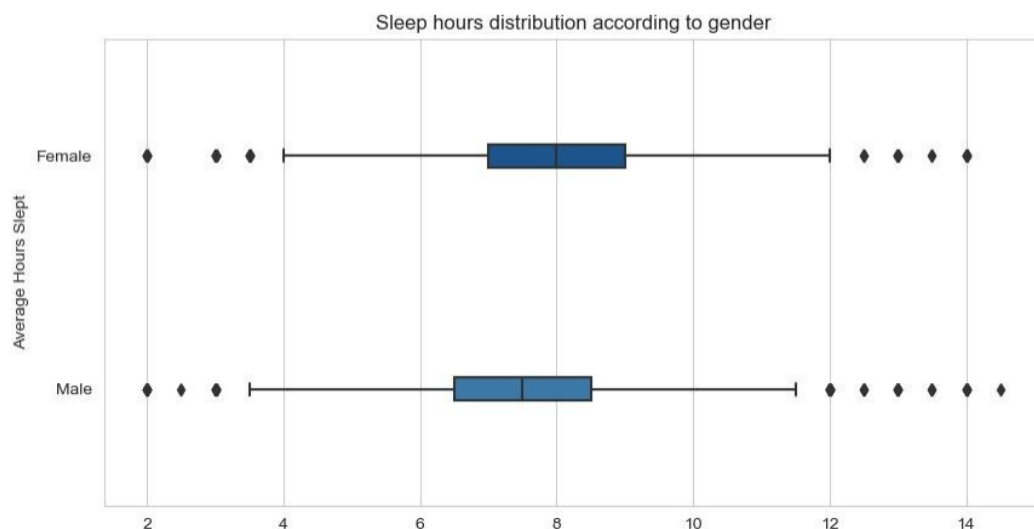
Field: People between 18 and 80 years old, 13082 observations.

The sample contains the information of 13082 respondents (table 1). Among this sample, 49,1% are male, and the average age is between 48 and 49 years old. The average Body Mass Index is 29.778, which shows that the people of this sample are rather heavy weighted/obese (a

BMI above 30 is a sign of obesity according to the World Health Organization). 54,4% of the respondents seem to have, at least, a bachelor's degree and only 18,7% have no diploma.

When we try to determine whether someone is subject to sleep disorders or not, it is inevitable to look at the sleep duration among the population. The sample shows an average of 7,63 hours of sleep with slight differences for men and women, with 7,50 and 7,76 hours of sleep respectively (appendix 1). When we try to plot the whole distribution of these two cohorts, we can quickly see that women in general get more sleep than men (figure 2). The median of women is also above men's (8 and 7,5 respectively). These results are statically significant and confirmed by the ANOVA test (appendix 2). These estimations seem to be reliable because they are very close to what other studies say about sleep (C. Rauch, 2021).

Figure 2: Distribution of sleep duration in hours by gender

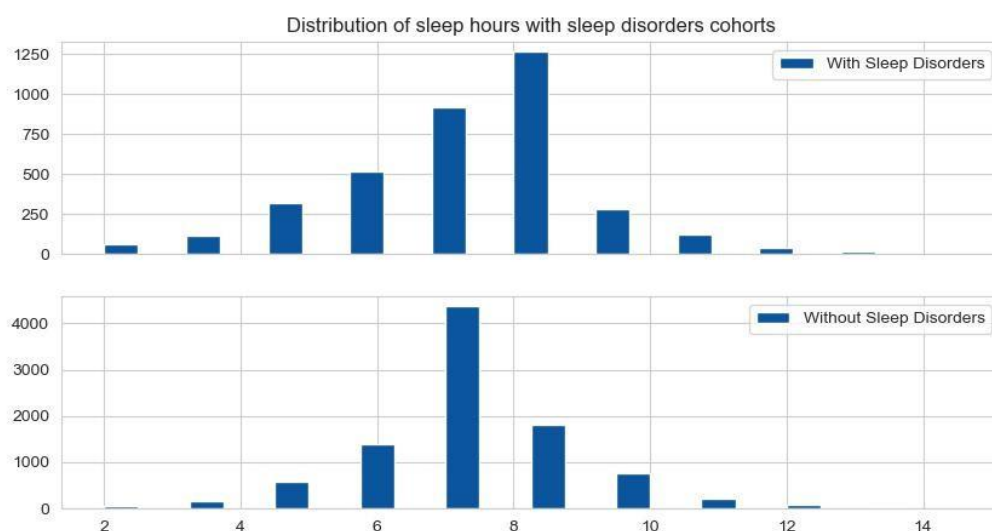


Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 13082 observations.

When we deal about sleep disorders, it is very likely to cause disorders in sleep durations. The sample shows a difference of sleep duration with and without sleep disorders. Majority of people that aren't subject to sleep disorders seem to sleep around 7 hours a day where people suffering from sleep disorders sleep mostly between 5 to 8 hours a day and very few above 8 hours a day (figure 3).

Figure 3: Distribution of sleep duration by sleep disorder cohorts

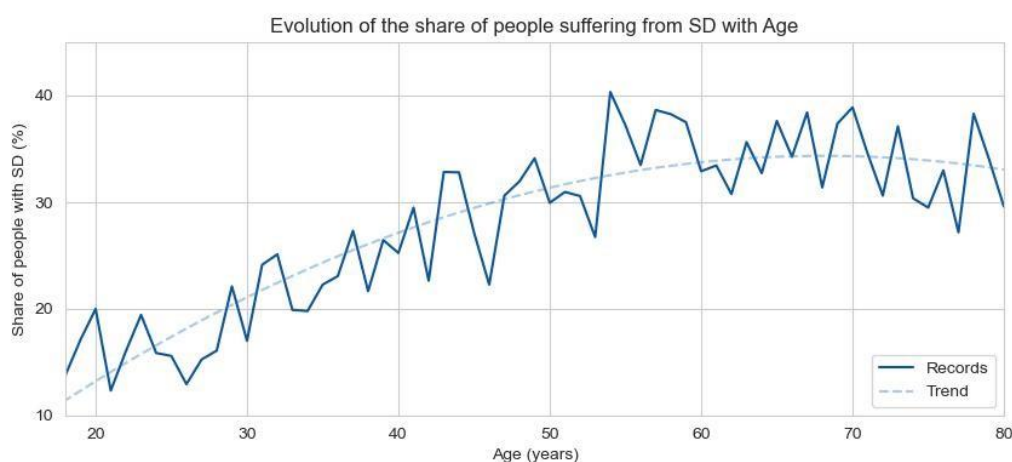


Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 13082 observations.

As the time passes and we grow old, our body weakens and multiple troubles can appear such as thyroid problems, cancers and sleep disorders. In our sample, the share of people suffering from sleep disorders tends to increase with age (figure 4). We can see that at 20 years old, the share of people suffering from sleep disorders is around 20% and grows to 35% between 50 and 60 years old. Past this age, the share doesn't seem to increase anymore and is stable. This link between age and sleep disorders have been researched in multiple studies, like K. Gulia's study in 2018 about the challenge of good sleep among the elderly.

Figure 4: Share of people subject to sleep disorders by age.

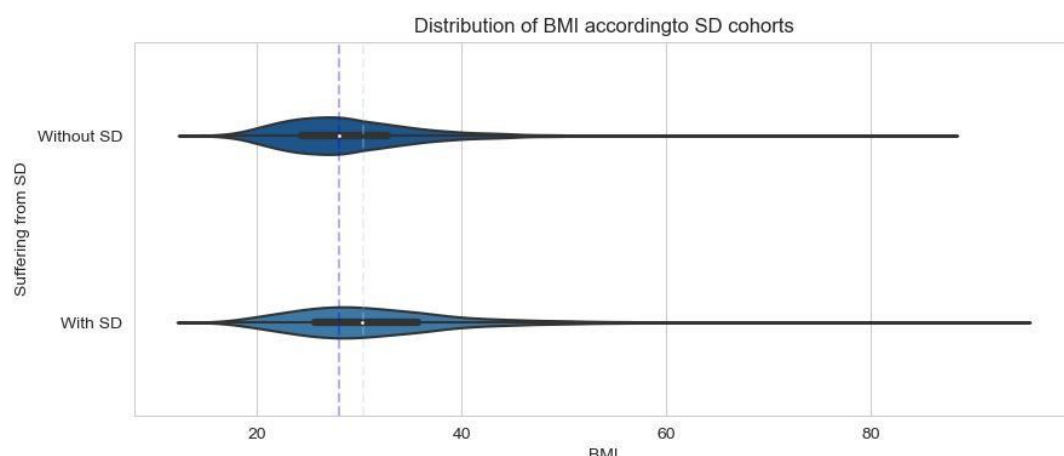


Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 13082 observations.

The potency to develop a sleep disorder is often increased if a person suffers from other health problems such as heart disease, diabetes, and obesity. Obesity has a direct impact on sleep disorders because sleep apnea may be caused by weight problems. This correlation can also be observed in our dataset (figure 4): we observe that in general, people suffering from sleep disorders present a higher BMI than those who don't: the median of the former group is equal to 30 while the latter's is equal to 26.

Figure 5: BMI distribution of the sleep disorders cohorts.



Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 13082 observations.

The model

Metrics and scoring methods

The main metrics used to compare the different models are the recall score, the accuracy score and the Area Under the Receiver Operator Characteristic Curve (ROC AUC) score. The recall score reflects the ability of a model to predict all the true positive in a dataset. For instance, in sample of 100 individuals where 60 of them are positives, if the recall of a model is equal to 80%, then out of these 60 positive individuals, the model is capable to retrieve 48 (80%) of them.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

However, only using the recall is not a judicious choice because we do not take account of the negative class. In this case, choosing the accuracy score, which measures the global effectiveness of the model. The accuracy score indicates the share of individuals well predicted regardless of their class. If a models' accuracy is equal to 90%, then out of 100 individuals, the model is able to predict 90 of them correctly whether the individual is positive or negative.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Positive\ (TP + FP) + Negative\ (FN + TN)}$$

And finally, the last metric we used to measure the models' performances is the Area Under the Receiver Operator Characteristic Curve (ROC AUC) score. The AUC score represents the probability to make a good prediction on a positive individual. If the AUC score is equal to 0.75, it means that a positive individual has a 75% chance to be well labelled.

$$AUC\ Score = \int_{x=0}^1 TPR(FPR^{-1}(x))dx$$

Model selection

After having cleaned the data and proceeded to its exploration, different models have been trained on the dataset, fine-tuned and cross validated (table 2). To fit the models, 80% of the data (10 466 observations) have been used in the training phase, while 20% of the data (2616 observations) have been used in the test phase.

Table 2: Trained models' benchmark

Model	Precision (0)	Recall (0)	Precision (1)	Recall (1)	Accuracy	Global score
Voting Classifier	0.85	0.76	0.51	0.66	0.73	0.70
Sequential	0.84	0.76	0.51	0.63	0.73	0.69
SGD Classifier*	0.84	0.77	0.51	0.61	0.73	0.68
XGB Classifier*	0.80	0.91	0.63	0.39	0.77	0.66
GB Classifier*	0.79	0.91	0.63	0.39	0.77	0.66
Logit Classifier*	0.79	0.93	0.65	0.34	0.77	0.65
RF Classifier*	0.78	0.94	0.68	0.32	0.77	0.65
BSGD Classifier*	0.79	0.92	0.62	0.36	0.76	0.65
DT Classifier*	0.78	0.77	0.41	0.43	0.67	0.59
KNN Classifier*	0.76	0.81	0.40	0.33	0.68	0.57

Source: NHANES (2017-March 2020 and 2015-2016), CDC.

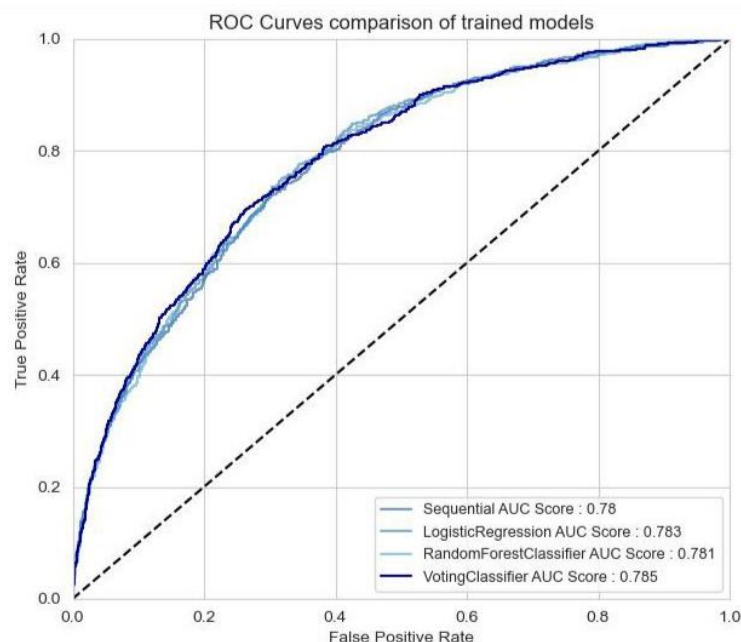
Field: People between 18 and 80 years old, 10465 observations.

*See appendix 3 for model description.

The model that performed the best was the Voting Classifier model that combined three different models: the Logistic Regression model, the Stochastic Gradient Descent Classifier model and the Gradient Boosting Classifier model. This model had the best recall score (66%), one of the best accuracy scores (73%) and the best global score (which is a weighted average of all the scores (0,13 for Precision (0), Recall (0) and Precision (1), 0,3 for Recall (1) and global accuracy). It is slightly better than the Sequential model which gets a global score of 0,69. The

models that performed the worst are the Decision Tree and K-Nearest Neighbors models that got 0,59 and 0,57 as global scores respectively. Note that only three models (Voting Classifier, Sequential and Stochastic Gradient Descent Classifier have a “good” recall). The area under the ROC curve score is also a good criterion to determine the performance of a model. The Area under the Roc curve score represents the probability of a model to correctly predict a positive observation. If a model had a score of 80%, this would mean that the model has an 80% chance to predict correctly a positive observation. Here again, the model with the best score is the Voting Classifier with a score of 78,5%, followed by the Logistic Regression with a score of 78,3% (figure 6). Notice that the Logistic Regression model has a good score here while the previous scores classed it 6th position.

Figure 6: ROC Curves of the 4 best models

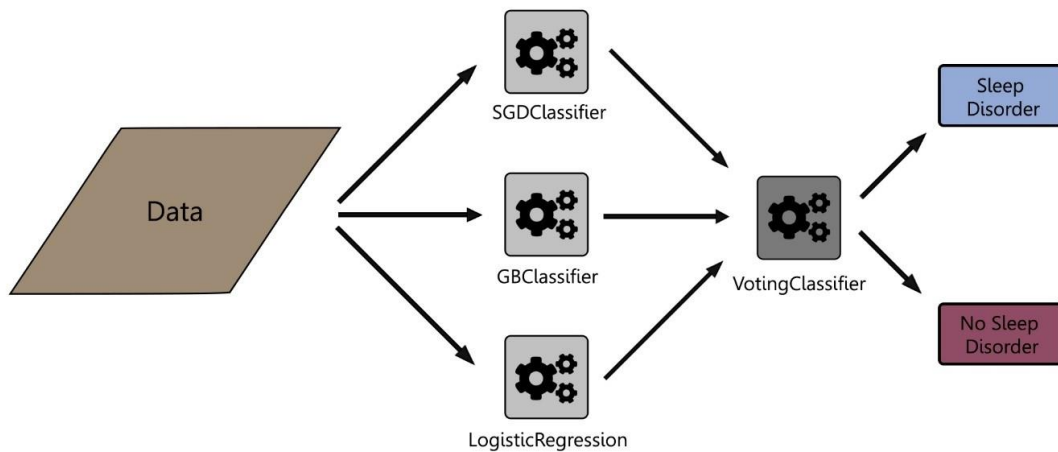


Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 10465 observations.

Because of its characteristics, we decided to choose the Voting Classifier model. This model is an ensemble model that combines multiples models in order to vote for the final output. This model is in fact made from three other models: the Stochastic Gradient Descent Classifier, the Gradient Boosting Classifier and the Logistic Regression (figure 7) that got trained and tuned before being integrated to the Voting Classifier model (appendix 3). The voting method chosen is the hard voting method, which means the models chooses the class according to a majority vote among the different models inside it.

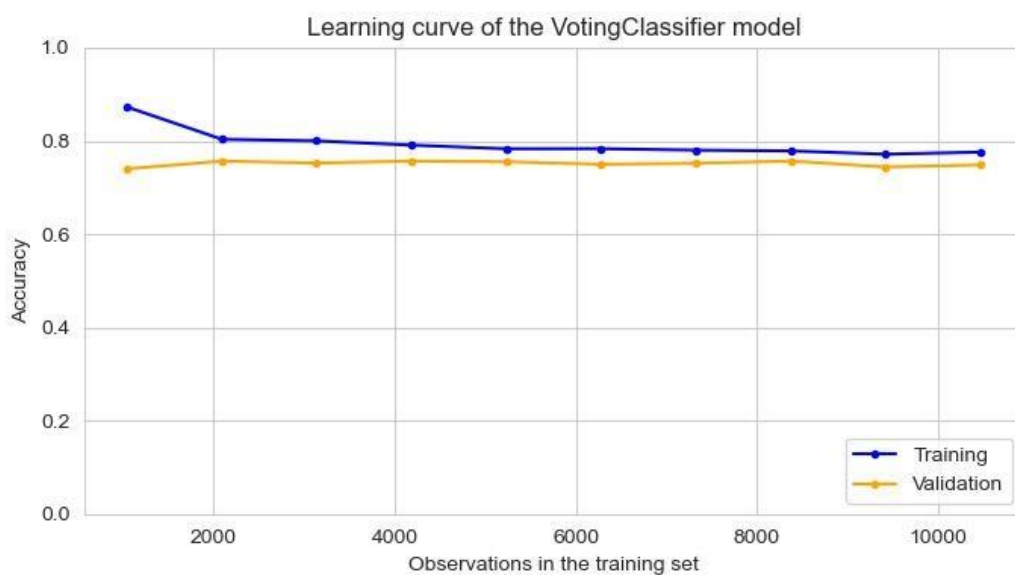
Figure 7: Structure of the Voting Classifier model.



Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 10465 observations

To ensure that our model is well fitted for the predictions, it is important to look at the learning curves. The learning curves of a model represents its performances on a dataset by regarding the size of the training and the validation sets. The Voting Classifier learning curves shows that with very few observations, the model's performances in the training and validation sets are quite low (figure 8). However, as the model trains over more observations, the models' performance tends to increase to reach its limit which is between 70% and 80% accuracy. Both training and test curves are close to each other and have a good performance. We can thus say that the Voting Classifier model is well fitted on the data.

Figure 8: Learning curve of the Voting Classifier model



Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 10465 observations.

If we inspect a little further the model's performance through the confusion matrix and the classification report (table 3 and table 4), multiple things are to notice. First of all, the global metrics show signs of a rather good model: precision, recall and F1 score are all above 70%, which is a good first step for a model. However, even though the models' performances might seem good, we can see that out of 930 real negative individuals, the model was able to correctly classify only 477 of them, which leaves 453 misclassified (figure 3). That is one big problem of this model. We can say the same of the positive individuals: out of 1687, 251 were misclassified, which represents 14,9% of the total positives. Some improvements might be required here too.

Table 3: Confusion matrix of the Voting Classifier model

	Real Positives	Real Negatives	Total
Positive predictions	1436	453	1889
Negative predictions	251	477	728
Total	1687	930	2617

Source: NHANES (2017-March 2020 and 2015-2016), CDC.

Field: People between 18 and 80 years old, 10465 observations.

Overall, the main improvements required for the model are on the positive part (figure 4). The model's performances for positive individuals are the following: 51%, 66% and 58% for the precision, recall and F1 scores, while 85%, 76% and 80% for the negative individuals. This represents a remarkable gap of performances between the two groups.

Table 4: Classification report of the Voting Classifier model

	Precision	Recall	F1 score	Support
Negatives	0.85	0.76	0.80	1889
Positives	0.51	0.66	0.58	728
Accuracy			0.73	2617
Macro Average	0.68	0.71	0.69	2617
Weighted Average	0.76	0.73	0.74	2617

Source: NHANES (2017-March 2020 and 2015-2016), CDC.

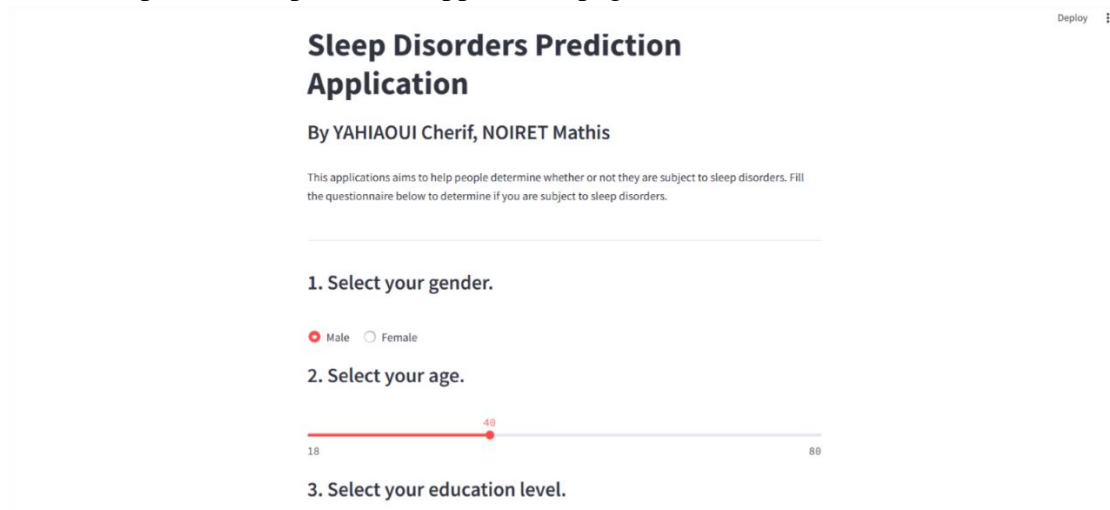
Field: People between 18 and 80 years old, 10465 observations.

To put it all together, the models' metrics show good overall performances of the model. Weighted average of precision, recall and F1 scores says that the model is able to predict a good number of individuals correctly. Despite that, if we investigate in detail, different problems appear, as the gap between positive and negative predictions for instance. These problems might be solved by adding more observations to the dataset through another implementation of the NHANES data or by the simulation of synthetic data from the original data, reducing the bias and preprocessing the data even further than already made.

Application

Based the model we had built previously, we created an application that takes the form of a questionnaire where a person can enter his information and know, thanks to the algorithm, whether his suffering from a sleep disorder or not (figure 9). The application is based on the Streamlit framework.

Figure 9: Sleep disorders prediction application page



Sleep Disorders Prediction Application

By YAHIAOUI Cherif, NOIRET Mathis

This applications aims to help people determine whether or not they are subject to sleep disorders. Fill the questionnaire below to determine if you are subject to sleep disorders.

1. Select your gender.

☒ Male ☐ Female

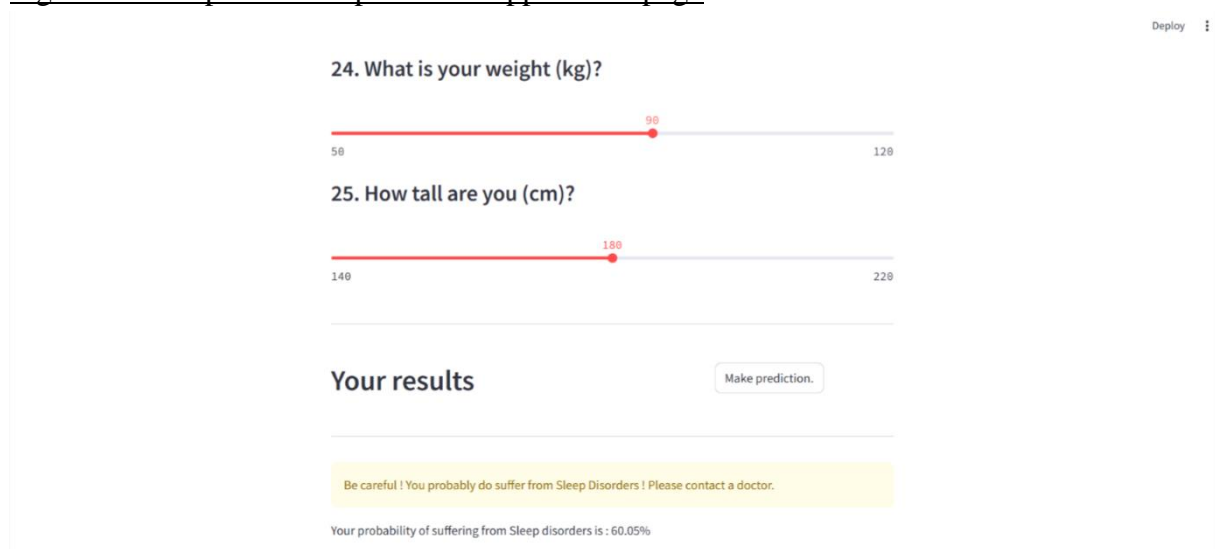
2. Select your age.

18 40 80

3. Select your education level.

Once the user opens the application, he is directly redirected to the questionnaire where he can enter his information through 25 questions that covers multiple aspects of his life (personal, economics, social, health, physical information, and habits). Once the user has finished to enter his information, he can press the “Make prediction.” button to discover if he’s subject to sleep disorders or not, according to the model. The application also shows the percentage of chance one has suffer from sleep disorders (figure 10).

Figure 10: Sleep disorders prediction application page



24. What is your weight (kg)?

50 90 120

25. How tall are you (cm)?

140 180 220

Your results Make prediction.

Be careful ! You probably do suffer from Sleep Disorders ! Please contact a doctor.

Your probability of suffering from Sleep disorders is : 60.05%

Discussion

To be complete on this subject, we are now going to discuss some of the biases that could have led to errors in our model. The most significant bias is the bias of missing variables. This bias is common in all studies but is particularly problematic when studying health in general. Health is a field that is very difficult to predict, given the infinite number of factors that influence the health of a person. Even with a large variety of variables in our model, it is impossible to estimate accurately, for example, risky behaviour, diet or a person's predisposition to develop certain disorders. If we had conducted our own study on the subject, we would have included more variables on the mental health of a person, his working environment, his drinking habits or his sleeping environment. For that reason, AI is beneficial to healthcare when it is acknowledged that outcomes are not always perfect and that a professional healthcare professional is required to make informed decisions based on all available information.

Another potential source of bias in our study is measurement error. As the questions asked of the sample were sometimes open to interpretation (e.g. how your health on a scale from is excellent to poor) or not very precise (e.g. average hours of sleep), measurement errors are numerous. In addition, it is possible that our target variable does not include all people with sleep disorders, as they are difficult to diagnose. Conversely, it is possible that some people report having a sleep disorder but do not actually have one.

The third possible major bias we found is related to the sample we had. Even if our data come from a reliable institution, a sample is never perfectly representative of the population. It is therefore normal to discuss our results, as our sample may contain extreme observations or a part of the population that is over- or under-represented. Finally, it is difficult to generalise our results, especially to countries with very different populations, as the population studied is entirely American. In conclusion, the data used in this essay were the most recent and complete for all the variables under consideration. However, it is important to acknowledge that uncertainty and biases remain, which are impossible to fully eliminate.

Conclusion

In conclusion, we have established the importance of addressing sleep disorders. These disorders affect a significant portion of the global population, directly impacting the health of those who suffer from them and indirectly impacting the economy through the costs associated with them. This is the reason we chose this subject; we believed that AI could be a solution to detect sleep disorders as fast as possible in order to treat the patients earlier. To conduct our study, we identified a consistent and reliable American database that evaluated a significant number of the variables we were seeking. Furthermore, we were aware that the information about American people is not significantly different from that of European people. Additionally, this database contains variables that we had not previously considered, such as the variable on mental health, which has a significant impact on sleep disorders. We then selected variables and encoded them to facilitate the use of machine learning techniques that we wanted to try. Our initial results were not as we had hoped because of missing variables, biases and AI models that were not fully adapted. For that reason, we opted for a hybrid model that combined the strengths of our different models of SGB, GB and Logistic Regression, which produced more satisfactory results. The accuracy for patients who did not suffer from sleep disorders was better than anticipated, with approximately 80% of those diagnosed being correctly identified. However, our results on people with sleep disorders were less performant, especially in terms of precision, with a score of 51%. These contrasting results were to be expected, given that, in the context of disease diagnosis, it is often easier to rule out a disorder than to diagnose it. We believe that by using an application, we can share our model with as many individuals as possible. The app has the potential to facilitate a preliminary diagnostic for individuals experiencing sleep issues, thereby guiding them to the appropriate specialist. The objective of our app in the future is twofold: firstly, to facilitate an earlier diagnosis, and secondly, to predict the type of sleep disorder that is most likely to affect the patient. To further elaborate on the subject, we believe that a proper study should be conducted with the specific aim of predicting the presence of a sleep disorder using AI. A dedicated study into sleep disorders would provide more comprehensive and consistent data, enabling a better understanding of the nuances that exist in our world.

Bibliography

Autret, A., Lucas, B., Mondon, K., Hommet, C., Corcia, P., Saudeau, D., & De Toffol, B. (2001). Sleep and brain lesions: a critical review of the literature and additional new cases. *Neurophysiologie Clinique/Clinical Neurophysiology*, 31(6), 356-375.

Barro, R. (1996). Health and economic growth. *World Health Organization*, 1-47.

Barro, R. J. (2013). Health and economic growth. *Annals of economics and finance*, 14(2), 329-366.

Bertisch, S. M., Pollock, B. D., Mittleman, M. A., Buysse, D. J., Bazzano, L. A., Gottlieb, D. J., & Redline, S. (2018). Insomnia with objective short sleep duration and risk of incident cardiovascular disease and all-cause mortality: Sleep Heart Health Study. *Sleep*, 41(6), zsy047.

Bianchi, M. T. (2013). Sleep Deprivation and Neurological Diseases. In *Sleep Deprivation and Disease: Effects on the Body, Brain and Behavior* (pp. 47-63). New York, NY: Springer New York.

Bliwise, D. L., & Young, T. B. (2007). The parable of parabola: what the U-shaped curve can and cannot tell us about sleep. *Sleep*, 30(12), 1614-1615.

Bloom, D. E., Canning, D., & Sevilla, J. P. (2001). The effect of health on economic growth: theory and evidence.

Borsoi, L., Armeni, P., Donin, G., Costa, F., & Ferini-Strambi, L. (2022). The invisible costs of obstructive sleep apnea (OSA): Systematic review and cost-of-illness analysis. *PloS one*, 17(5), e0268677.

Brass, S. D., Duquette, P., Proulx-Therrien, J., & Auerbach, S. (2010). Sleep disorders in patients with multiple sclerosis. *Sleep medicine reviews*, 14(2), 121-129.

Budhiraja, R., Budhiraja, P., & Quan, S. F. (2010). Sleep-disordered breathing and cardiovascular disorders. *Respiratory Care*, 55(10), 1322-1332.

Cappuccio, F. P., Cooper, D., D'Elia, L., Strazzullo, P., & Miller, M. A. (2011). Sleep duration predicts cardiovascular outcomes: a systematic review and meta-analysis of prospective studies. *European heart journal*, 32(12), 1484-1492.

Cappuccio, F. P., D'Elia, L., Strazzullo, P., & Miller, M. A. (2010). Sleep duration and all-cause mortality: a systematic review and meta-analysis of prospective studies. *Sleep*, 33(5), 585-592.

Chokroverty, S. (2009, September). Sleep and neurodegenerative diseases. In *Seminars in neurology* (Vol. 29, No. 04, pp. 446-467). © Thieme Medical Publishers.

Cordina-Duverger, E., Houot, E., Tvardik, N., El Yamani, M., Pilorget, C., & Guénel, P. (2019). Prévalence du travail de nuit en France: caractérisation à partir d'une matrice emplois-expositions. *Bull Epidemiol Hebd*, (8–9), 168-174.

Culebras, A., & Kelly, J. J. (2008). Sleep disorders and neuromuscular diseases. *Reviews in Neurological Diseases*, 5(3), 153-158.

Fang, H., Tu, S., Sheng, J., & Shao, A. (2019). Depression in sleep disturbance: a review on a bidirectional relationship, mechanisms and treatment. *Journal of cellular and molecular medicine*, 23(4), 2324-2332.

Fernandez-Mendoza, J., & Vgontzas, A. N. (2013). Insomnia and its Impact on Physical and Mental Health. *Current Psychiatry Reports/Current Psychiatry Reports*, 15(12).

Ford, D. E., & Kamerow, D. B. (1989). Epidemiologic study of sleep disturbances and psychiatric disorders: an opportunity for prevention?. *Jama*, 262(11), 1479-1484.

F, Z. (2023, 6 octobre). *Trouble du sommeil : causes possibles, conseils pour mieux dormir*.

Gutiérrez-Tobal, G. C., Álvarez, D., Kheirandish-Gozal, L., Del Campo, F., Gozal, D., & Hornero, R. (2022). Reliability of machine learning to diagnose pediatric obstructive sleep apnea: Systematic review and meta-analysis. *Pediatric Pulmonology*, 57(8), 1931-1943.

Ha, S., Choi, S. J., Lee, S., Wijaya, R. H., Kim, J. H., Joo, E. Y., & Kim, J. K. (2023). Predicting the Risk of Sleep Disorders Using a Machine Learning–Based Simple Questionnaire: Development and Validation Study. *Journal of medical Internet research*, 25, e46520.

Hermann, D. M., & Bassetti, C. L. (2009). Sleep-related breathing and sleep-wake disturbances in ischemic stroke. *Neurology*, 73(16), 1313-1322.

Hillman, D. R., Murphy, A. S., Antic, R., & Pezzullo, L. (2006). The economic cost of sleep disorders. *Sleep*, 29(3), 299-305.

Hossain, J. L., & Shapiro, C. M. (2002). The prevalence, cost implications, and management of sleep disorders: an overview. *Sleep and Breathing*, 6(02), 085-102.

Johnson, K. G., & Johnson, D. C. (2010). Frequency of sleep apnea in stroke and TIA patients: a meta-analysis. *Journal of clinical sleep medicine*, 6(2), 131-137.

Kapur, V., Blough, D. K., Sandblom, R. E., Hert, R., de Maine, J. B., Sullivan, S. D., & Psaty, B. M. (1999). The medical cost of undiagnosed sleep apnea. *Sleep*, 22(6), 749-755.

Lal, C., Strange, C., & Bachman, D. (2012). Neurocognitive impairment in obstructive sleep apnea. *Chest*, 141(6), 1601-1610.

Leger, D. (1994). The cost of sleep-related accidents: a report for the National Commission on Sleep Disorders Research. *Sleep*, 17(1), 84-93.

Lévy, P., Kohler, M., McNicholas, W. T., Barbé, F., McEvoy, R. D., Somers, V. K., ... & Pépin, J. L. (2015). Obstructive sleep apnea syndrome. *Nature reviews Disease primers*, 1(1), 1-21.

Malhotra, A., Orr, J. E., & Owens, R. L. (2015). On the cutting edge of obstructive sleep apnea: where next?. *The Lancet Respiratory Medicine*, 3(5), 397-403.

Mayer, G., Jennum, P., Riemann, D., & Dauvilliers, Y. (2011). Insomnia in central neurologic diseases—occurrence and management. *Sleep medicine reviews*, 15(6), 369-378.

Odoxa. (2024, 28 février). *Les Français ont un mauvais sommeil. . . c'est encore pire pour Les professionnels de santé - Odoxa. Overview of sleep & sleep disorders : Indian Journal of Medical Research.* (s. d.). LWW.

Palma, J. A., Urrestarazu, E., & Iriarte, J. (2013). Sleep loss as risk factor for neurologic disorders: a review. *Sleep medicine*, 14(3), 229-236.

Professional, C. C. M. (s. d.). *Sleep disorders*. Cleveland Clinic.

Rauch, C. (2021). Social Inequalities and the Desynchronisation of Sleep within Couples *Economie et Statistique / Economics and Statistics*, 522-523, 81-104.

Riemann, D., Nissen, C., Palagini, L., Otte, A., Perlis, M. L., & Spiegelhalder, K. (2015). The neurobiology, investigation, and treatment of chronic insomnia. *The Lancet Neurology*, 14(5), 547-558.

Schwartz, A. R., Cohen-Zion, M., Pham, L. V., Gal, A., Sowho, M., Sgambati, F. P., ... & Pillar, G. (2020). Brief digital sleep questionnaire powered by machine learning prediction models identifies common sleep disorders. *Sleep Medicine*, 71, 66-76.

Sleep Disorder Treatments | NHLBI, NIH. (2022, 24 mars). NHLBI, NIH.

Sommeil · Inserm, La science pour la santé. (s. d.). Inserm.

Tregear, S., Reston, J., Schoelles, K., & Phillips, B. (2009). Obstructive sleep apnea and risk of motor vehicle crash: systematic review and meta-analysis. *Journal of clinical sleep medicine*, 5(6), 573-581.

Tsai, J. C. (2010). Neurological and neurobehavioral sequelae of obstructive sleep apnea. *NeuroRehabilitation*, 26(1), 85-94.

World Health Organization : WHO. (2024, 1 mars). *Obésité et surpoids.*

Appendix

Appendix 1: Average hours slept by gender

	Number of observations	Hours slept
Gender		7,63
Male	6429	7,50
Female	6653	7,76

Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 13082 observations.

Appendix 2: Statistical tests

	Test used	Test Statistic	p-Value
Figure 2			
Normality test (Male)	Shapiro	0,977	<0,001
Normality test (Female)	Shapiro	0,977	<0,001
Mean comparison test	ANOVA	86,554	<0,001
Figure 5			
Normality test (SD*)	Shapiro	0,933	<0,001
Normality test (without SD*)	Shapiro	0,936	<0,001
Mean comparison test	ANOVA	270,389	<0,001

Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 10465 observations.

*SD: Sleep disorders.

Appendix 3: Models description

Model Name	Full Model Name
SGD	Stochastic Gradient Descent
XGB	eXtreme Gradient Boosting
GB	Gradient Boosting
RF	Random Forest
BSGD	Bagging Stochastic Gradient Descent
DT	Decision Tree
KNN	K-Nearest Neighbors

Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 10465 observations.

Appendix 4: Voting Classifier models' details.

	SGD Classifier	GB Classifier	Logistic Regression
Hyperparameters			
HP 1	random_state : 301	random_state : 32	-
HP 2	loss="log"	n_estimators=200	-
HP 3	-	min_samples_split=16	-

Source: NHANES (2017-March 2020 and 2015-2016), CDC.
Field: People between 18 and 80 years old, 10465 observations.

*SD: Sleep disorders

Code

```
##### Model Training #####
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import os
import warnings
from math import modf
import seaborn as sns
from scipy.stats import kruskal, chi2_contingency
import datetime as dt
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import DBSCAN
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier,
AdaBoostClassifier, VotingClassifier, BaggingClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.decomposition import PCA
from sklearn.metrics import recall_score, accuracy_score, make_scorer,
roc_auc_score
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.layers import Dense
from tensorflow.keras import Sequential
from tensorflow.keras.metrics import Recall
from tensorflow.random import set_seed

warnings.filterwarnings('ignore')

os.chdir(r'C:\Users\yahia\Python\M2 Mémoire')

temp=None

tables=['SQL', 'DBQ', 'DEMO', 'BMX']

for i in os.listdir('data'):
```

```

if ('SLQ' in i) | ('DBQ' in i) | ('DEMO' in i) | ('BMX' in i) | ('ALQ' in i) | ('HUQ' in i) |
('PAQ' in i) | ('SMQ' in i) | ('DPQ' in i) | ('KIQ' in i):
    dataset = pd.read_sas(r"data/" + i)
    dataset['SEQN'] = dataset['SEQN'].astype(str).apply(lambda x: x.split('.')[0])
    print("Importation de ", i, f" ({dataset.shape[0]} x {dataset.shape[1]})")

```

```

if type(temp)==type(None):
    temp = dataset.copy()
else:
    temp = pd.merge(temp, dataset, how='left', on='SEQN')
os.system("clear")
os.system("cls")

```

```

var_list=['RIAGENDR',
'RIDAGEYR',
'DMDMARTZ',
'ALQ121',
'DBQ700',
'HUQ010',
'HUQ090',
'PAQ605',
'PAD680',
'SLD012',
'SLD013',
'SLQ050',
'SMQ040',
'KIQ480',
'DPQ020',
'DPQ030',
'DPQ070',
'DPQ090',
'SLQ040',
'BMXBMI']

```

```

labels=['Sexe',
'Age',
'Statut',
'Alcool',
'Q_nutri',
'Sante',
'Sante_ment',
'Aphys_fort',

```

```
'Sedentarite',
'Sommeil_sem',
'Sommeil_we',
'Troub_sommeil',
'Fumeur',
'Pee_par_night',
'depressed',
'sleep_pb',
'attention_trouble',
'dangerous_bvr',
'stop_breathing',
'BMI']
```

```
df = temp[var_list]
df.columns = labels
```

```
miss=[7, 9, 77, 99, 7777, 9999]
df['Sexe'] = np.where(df['Sexe']==1, 1, 0)
df['Alcool'] = np.select([df['Alcool']==0, ~df['Alcool'].isin([miss])], [0, 1], 0)
df['depressed'] = np.select([df['depressed']==0, ~df['depressed'].isin([miss])], [0, 1], 0)
df['Sommeil_h']=(df['Sommeil_sem']+df['Sommeil_we'])/2
df['sleep_pb'] = np.select([df['sleep_pb']==0, ~df['sleep_pb'].isin([miss])], [0, 1], 0)
```

```
cate=[col for col in df.drop(['BMI', 'Sommeil_h', 'Age', 'Sedentarite', 'Sommeil_sem',
'Sommeil_we'], axis=1).columns]
dummies=pd.get_dummies(df[cate], dummy_na=True)
dft=pd.DataFrame()
dft[dummies.columns] = dummies
```

```
# Preprocessing
```

```
def import_table(path, year):
```

```
    os.chdir(path)
```

```
    temp=None
```

```
    for i in os.listdir(f'data/{year}'):
        dataset = pd.read_sas(f'data/{year}/{i}')
        dataset['SEQN'] = dataset['SEQN'].astype(str).apply(lambda x: x.split('.')[0])
```

```
    if type(temp)==type(None):
```

```

        temp = dataset.copy()
    else:
        temp = pd.merge(temp, dataset, how='left', on='SEQN')
    #print("Importation de ", i, f" ({dataset.shape[0]} x {dataset.shape[1]})")
    os.system("clear")
    os.system("cls")

var_list=['RIAGENDR',
          'RIDAGEYR',
          'DMDEDUC2',
          'INDFMPIR',
          'BPQ020',
          'BPQ080',
          'DBQ700',
          'HIQ011',
          'HUQ010',
          'HUQ090',
          'PAQ605',
          'PAQ620',
          'PAD680',
          'SLD012',
          'SLQ050',
          'SMQ040',
          "DIQ010",
          "DBD900",
          "HUQ051",
          "KIQ480",
          "MCQ010",
          "MCQ160B",
          "MCQ160C",
          "MCQ160E",
          "MCQ160F",
          "MCQ160M",
          "MCQ220",
          "DPQ020",
          "DPQ030",
          "DPQ070",
          "DPQ090",
          "SLQ040",
          "BMXBMI"]

labels=['Sexe',
        'Age',
        'Educ',

```

```

'Pauverty',
'PA_forte',
'Chol_fort',
'Q_nutri',
'Ass_maladie',
'Sante',
'Sante_ment',
'Aphys_fort',
'Aphys_mod',
'Sedentarite',
'Sommeil_sem',
'Troub_sommeil',
'Fumeur',
"Diabete",
"FastFood",
"Nb_medecin",
"Pee_par_night",
"Asthma",
"Heart_disease",
"corona_disease",
"had_heart_atk",
"had_CVA",
"had_tyroid",
"had_cancer",
"depressed",
"sleep_pb",
"attention_trouble",
"dangerous_bvr",
"stop_breathing",
"BMI"]

```

```

df = temp[var_list]
df.columns = labels
print("longueur df:",df.shape)
return df

```

```

def cleaning(df):
    print('Rows before cleaning',df.shape)
    # Subject : Smoking variable
    df['Fumeur'] = np.select([df['Fumeur'].isin([1, 2])], ["Smoke_Yes"], "Smoke_No")

    # Subject : Pauverty
    df['Pauverty'] = df['Pauverty'].apply(lambda x: modf(x)[1])
    df['Pauverty'] = np.select([~df["Pauverty"].isin([np.NaN, 'nan'])],

```



```

        "stop_breathing_often",
        "stop_breathing_sometimes"], "stop_breathing_rarely")

# Subject : 'Dangerous_thoughts', 'attention_trouble', 'sleep_problems' variables
for col in ["dangerous_bvr", "attention_trouble", "depressed"]:
    df[col] = np.select([round(df[col]) <= 5], [round(df[col])], np.NaN)
    df[col] = np.select([df[col]==0,
                        df[col]==1,
                        df[col].isin([2,3])],
                        [f"{col}_never",
                        f"{col}_sometimes",
                        f"{col}_very_often"], f"{col}_ms")

    cols=["had_cancer", "had_tyroid", "had_heart_atk", "had_CVA", 'Heart_disease',
'corona_disease',
        "PA_forte", 'Chol_fort', "Ass_maladie", 'PA_forte', 'Sante_ment', 'Aphys_fort',
'Aphys_mod', 'Troub_sommeil',
        "Asthma"]
    for col in cols:
        df[col] = np.select([df[col]==1,
                            df[col]==2],
                            [f"{col}_Yes",
                            f"{col}_No"],
                            f"{col}_ms")

# Subject : Diabete variable
df["Diabete"] = np.select([df["Diabete"]==1,
                            df["Diabete"].isin([2,3])],
                            ["Diabete_Yes",
                            "Diabete_No"],
                            "Diabete_ms")

# Subject : Nutrition & Health
df = df.dropna()
missing=[7, 9, 77, 99]

for col in ['Q_nutri', 'Sante']:
    df = df[(~df[col].isin(missing))]
    df[col]=np.select([df[col]==i for i in range(1, 6)],
                      [f"{col}_{i}" for i in range(1, 6)])

# Subject : Doctor visits number

```

```

df["Nb_medecin"] = round(df["Nb_medecin"])
df = df[df["Nb_medecin"]!=99]

# Subject : Average hours slept
df["Sleep_h"] = df[["Sommeil_sem"]].mean(axis=1)
df = df.drop(["Sommeil_sem"], axis=1)

# Subject : Sedentariness
df=df[df["Sedentarite"]<1500]
df["Sedentarite"]=df["Sedentarite"]/60

df['sleep_pb']=np.select([df['sleep_pb'].isin([1, 2, 3]), df['sleep_pb']==0],
['sleep_pb_Yes', 'sleep_pb_No'], "sleep_pb_ms")
print('Rows after cleaning',df.shape)
return df

def binarize(df, output='data'):
    types=pd.DataFrame(df.dtypes, columns=["Type"]).reset_index()
    numeric_cols=types[types["Type"]=="float64"]["index"]
    cate_cols=types[types["Type"]=="object"]["index"]

    if output=="data":
        data=pd.get_dummies(df[cate_cols], drop_first=True,prefix="", prefix_sep="")
        data[numeric_cols] = df[numeric_cols]
        return data
    elif output=="datax":
        datax=pd.get_dummies(df[cate_cols],prefix="", prefix_sep="")
        datax[numeric_cols] = df[numeric_cols]
        return datax

def relationships(data, target):
    cols=data.drop(target, axis=1).columns
    pval=[]

    for col in cols:
        chi2=chi2_contingency(pd.crosstab(data[col], data[target]))
        pval.append(chi2[1])
    temp=pd.DataFrame({'Column':cols,
                       'PValue':pval})
    #temp=temp[temp['PValue']<0.1]
    return temp

def strip(x, strings=' '):

```



```

    for i in str(x):
        x=x.replace(strings, "")
    return x

df_2017=import_table(r'C:\Users\yahia\Python\M2 Mémoire', 2017)
df_2015=import_table(r'C:\Users\yahia\Python\M2 Mémoire', 2015)

data = binarize(pd.concat([cleaning(df_2015),cleaning(df_2017)]).drop('sleep_pb',
axis=1))
print(f'Format de la table 'data': {dataz.shape}")
data.head()

# Statistics

fig, ax = plt.subplots(2, sharex=True, figsize=(10,5))

data[data["Troub_sommeil_Yes"]==1]['Sleep_h'].hist(width=0.5, ax=ax[0])
data[data["Troub_sommeil_Yes"]==0]['Sleep_h'].hist(width=0.5, ax=ax[1]);

#data.groupby("Troub_sommeil_Yes")['Sleep_h'].value_counts().plot() #.hist()

ax[0].title.set_text('Distribution of sleep hours with sleep disorders cohorts')
ax[0].legend(["With Sleep Disorders"])
ax[1].legend(["Without Sleep Disorders"]);

plt.savefig('Sleep_by_SD.jpeg')

sns.set_style('whitegrid')
plt.figure(figsize=(10,5))
sns.boxplot(data=data, y='Male', x='Sleep_h', width=0.1, orient="h");

plt.xlabel("")
plt.ylabel('Average Hours Slept')
plt.yticks(ticks=[0,1],labels=['Female', "Male"])
plt.title('Sleep hours distribution according to gender');

plt.savefig('Sleep_by_gender.jpeg')

# Plotting the average share of people suffering from SD
data.groupby('Age')['Troub_sommeil_Yes'].mean().plot(figsize=(10,4),
label="Records")

# Modeling and plotting a non-linear regression

```

```

modeling = np.polyfit(data.groupby('Age')['Troub_sommeil_Yes'].mean().index,
data.groupby('Age')['Troub_sommeil_Yes'].mean(), 2)

```

```

f=np.poly1d(modeling)

```

```

xs=[i for i in range(81)]
ys=[f(x) for x in xs]

```

```

plt.plot(xs, ys, alpha=0.4, linestyle='--', label='Trend')

```

```

# Graphic options

```

```

plt.title('Evolution of the share of people suffering from SD with Age')
plt.ylabel('Share of people with SD (%)')
plt.xlabel('Age (years)')
plt.xlim(18, 80)
plt.ylim(0.1, 0.45)
plt.xticks(ticks=[0.10, 0.20, 0.30, 0.40], labels=['10', '20', '30', '40']);
plt.legend(loc='lower right')

```

```

plt.savefig('Age_SD.jpeg')

```

```

plt.figure(figsize=(10,4))
sns.violinplot(data=data,x='BMI', y='Troub_sommeil_Yes', width=0.2, orient="h");

```

```

plt.axvline(x=30.4, ymin=0, ymax=1, color='lightblue', linestyle='--', alpha=0.3)
plt.axvline(x=28, ymin=0, ymax=1, color='blue', linestyle='--', alpha=0.3)
plt.title('Distribution of BMI accordingto SD cohorts')
plt.ylabel("Suffering from SD")
plt.yticks(ticks=[0,1], labels=['Without SD', "With SD"]);

```

```

plt.savefig('BMI_SD.jpeg')

```

```

# Modèles finaux

```

```

file_n=0

```

```

def save_model(model_name, model):
    global file_n
    if file_n ==0:
        with open('Model results.csv', 'w') as file:
            file.write('Model name, Precision 0, Recall 0, Precision 1, Recall 1, Accuracy\n')

```

```

y_pred=model.predict(X_test)

```

```

if model_name=="Sequential":
    y_pred=np.where(y_pred>0.5, 1, 0)
    acc=accuracy_score(y_test, y_pred)
    p0=classification_report(y_test, y_pred).split("\n")[2].split('    ')[2].strip()
    r0=classification_report(y_test, y_pred).split("\n")[2].split('    ')[3].strip()
    p1=classification_report(y_test, y_pred).split("\n")[3].split('    ')[2].strip()
    r1=classification_report(y_test, y_pred).split("\n")[3].split('    ')[3].strip()
    row=[model_name, p0, r0, p1, r1, round(acc,2)]
    text=', '.join([str(i) for i in row])
    characters="]["
    with open('Model results.csv', 'a') as file:

        file.write(f'{strip(str(text),characters)} \n')
    file_n+=1

save_model('SGDClassifier', cv)

data_temp=data.copy()

X = data_temp.drop(['Troub_sommeil_Yes'], axis=1)
y = data_temp['Troub_sommeil_Yes']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=338)

## SGD Classifier

#from sklearn.metrics import
from sklearn.model_selection import cross_val_score

sgd = SGDClassifier(random_state=301,loss="log",)

params={"l1_ratio":[i/100 for i in range(50)],
        'epsilon':[i/10 for i in range(10)]}

sgd_cv=GridSearchCV(sgd, params, scoring='f1')

sgd.fit(X_train, y_train)

preds = sgd.predict(X_test)
print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))

```

```

print(recall_score(y_test, preds))

save_model('SGDClassifier', sgd)

## Logistic Regression

lr = LogisticRegression()
lr.fit(X_train, y_train)
preds = lr.predict(X_test)

save_model('LogisticRegression', lr)

## XGBoost

scores=[]
n=0
for estim in range(100,1501,100):
    for rs in range(0,100,5):

        xgb = XGBClassifier(n_jobs=3, n_estimators=estim, random_state=rs)
        xgb.fit(X_train, y_train)
        preds = xgb.predict(X_test)
        p0=classification_report(y_test, preds).split('\n')[2].split(' ')[2].strip()
        r0=classification_report(y_test, preds).split('\n')[2].split(' ')[3].strip()
        p1=classification_report(y_test, preds).split('\n')[3].split(' ')[2].strip()
        r1=classification_report(y_test, preds).split('\n')[3].split(' ')[3].strip()
        scores.append([p0, r0, p1, r1, accuracy_score(y_test, preds), estim, rs])
    n=n+1
print(round(n/15*100,2))

scored=pd.DataFrame(scores, columns=['Precision_0', 'Recall_0', 'Precision_1',
'Recall_1', 'Accuracy', 'n_estimators', 'random_state'])
for col in scored.columns:
    scored[col]=scored[col].astype(float)
#scored[scored['Accuracy']>0.726][scored['Recall_1']>0.6]

#cv=GridSearchCV(xgb, params, scoring='f1')
#save_model('XGBClassifier', cv)

xgb = XGBClassifier(n_jobs=3, n_estimators=1500, random_state=0)
xgb.fit(X_train, y_train)
preds = xgb.predict(X_test)

```

```

#save_model('XGBClassifier', xgb)

## RandomForestClassifier

rfc = RandomForestClassifier()

params={"n_estimators":[i for i in range(100,1500, 100)],
        'min_samples_split':[i for i in range(10,21,2)],
        "random_state":[i for i in range(50)]}

cv=GridSearchCV(dtc, params, scoring='f1')

cv.fit(X_train, y_train)

preds = cv.predict(X_test)

print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

rfc      =      RandomForestClassifier(n_estimators=1000,      random_state=0,
min_samples_split=10)

params={"n_estimators":[1000],
        'min_samples_split':[i for i in range(10,21,2)],
        "random_state":[0]}

cv=RandomizedSearchCV(rfc, params, scoring='f1')

rfc.fit(X_train, y_train)

preds = rfc.predict(X_test)

print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

save_model('RandomForestClassifier', rfc)

```

```

## KNeighborsClassifier

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=1, p=2, weights="uniform")

params={"n_neighbors":[i for i in range(10)],
        'weights':['uniform', 'distance'],
        'p':[1, 2]}

knn_cv=GridSearchCV(knn, params, scoring='recall')

knn.fit(X_train, y_train)


preds = knn.predict(X_test)

print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

save_model('KNearestNeighbors', knn_cv)


## Decision Tree Classifier

dtc = DecisionTreeClassifier(random_state=42)

params={"random_state":[i for i in range(100)],
        'min_samples_split':[i for i in range(10,21,2)],
        "random_state":[i for i in range(50)]}

dt_cvc=GridSearchCV(dtc, params, scoring='recall')

dtc.fit(X_train, y_train)


preds = dtc.predict(X_test)
#print(cross_val_score(dtc, X, y, cv=5, scoring=make_scorer(recall_score,beta=2)))
print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

```

```

save_model('DecisionTreeClassifier', dtc)

## Gradient Boosting

GradientBoostingClassifier

gbc_cv = GradientBoostingClassifier(random_state=32, n_estimators=1200,
min_samples_split=16)

params={"n_estimators":[i for i in range(100, 1501, 100)],
        'min_samples_split':[i for i in range(10,21,2)],
        "random_state":[i for i in range(50)]}

#gbc_cv=RandomizedSearchCV(gbc, params, scoring='recall')

gbc_cv.fit(X_train, y_train)

preds = gbc_cv.predict(X_test)
#print(cross_val_score(dtc, X, y, cv=5, scoring=make_scorer(recall_score,beta=2)))
print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

gbc_cv.best_params_

save_model('GradientBoostingClassifier', gbc_cv)

## Sequential

set_seed(2)

model = Sequential(
    [Dense(80, input_shape=(X_train.shape[1],)),
     #Dense(40, activation="relu"),
     Dense(10, activation="relu"),# 10, epoch 5 : 79% accuracy
     Dense(1, activation='sigmoid')])

model.compile(optimizer='adam', metrics=[Recall(), 'accuracy'],
loss='binary_crossentropy')

```

```

model.fit(X_train, y_train, validation_split=0.2, epochs=3, use_multiprocessing=True)

# 80 10 1, val_acc=65%, val_recall=68%

preds=model.predict(X_test)
preds=np.where(preds>0.5,1,0)

print(preds)
print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

save_model('Sequential', model)

model.save('Sequential.h5')

## Ensemble SGD + GB + LR

vc=VotingClassifier([('SGD', SGDClassifier(random_state=301, loss="log")),
                    ('GradientBoosting', GradientBoostingClassifier(random_state=32,
n_estimators=1200, min_samples_split=16)),
                    ('LogisticRegression', LogisticRegression())], voting='soft')

vc.fit(X_train, y_train)
preds=vc.predict(X_test)

print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

import pickle

with open('VCmodel.pkl','wb') as f:
    pickle.dump(vc,f)

save_model('VotingSGDxGBxLR', vc)

## Bagging SGD

bc=BaggingClassifier(SGDClassifier(), n_estimators=1000)

bc.fit(X_train, y_train)
preds=bc.predict(X_test)

```



```

print(classification_report(y_test, preds))
print(accuracy_score(y_test, preds))
print(recall_score(y_test, preds))

save_model('BaggingSGD', bc)

# Best Model Results

## Model Comparison Table

results = pd.read_csv('Model Results.csv')
results['Global Score'] = np.dot(results.iloc[:,1:], [0.4/3,0.4/3,0.4/3,0.3,0.3])
results = results.sort_values('Global Score', ascending=False)
results.index = [i+1 for i in range(results.shape[0])]
results

## Roc Curve

from sklearn.metrics import roc_curve, plot_roc_curve, roc_auc_score
from keras.models import load_model

plt.figure(figsize=(7,6))

model=load_model('Sequential.h5')

models = [model, lr, rfc, vc]
names=      ['Sequential',      'LogisticRegression',      'RandomForestClassifier',
'VotingClassifier']

curves=pd.DataFrame()

for name,model in zip(names, models):
    if name=="Sequential":
        y_pred=model.predict(X_test)
    else:
        y_pred=model.predict_proba(X_test)[:,-1]
    fpr, tpr, thr = roc_curve(y_test, y_pred)
    if 'Voting' not in name:
        plt.plot(fpr,      tpr,      label=f"{ name}      AUC      Score      :
{round(roc_auc_score(y_test,y_pred),3)}", alpha=0.6)
    else:
        plt.plot(fpr,      tpr,      label=f"{ name}      AUC      Score      :
{round(roc_auc_score(y_test,y_pred),3)}", color='darkblue')

```

```

plt.legend(loc="lower right", fontsize=9)
plt.plot([0,1], [0,1], linestyle='--', color='black')
plt.xlim(0,1)
plt.ylim(0,1)
plt.title('ROC Curves comparison of trained models')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate');
plt.savefig('ROC_Curves.jpeg')

## Validation curve

from sklearn.model_selection import learning_curve

train_sizes, train_scores, valid_scores= learning_curve(vc, X, y, train_sizes=[i/10 for i
in range(1,11)])

plt.figure(figsize=(8,4))
plt.plot(train_sizes,train_scores.mean(axis=1),      label="Training",      marker='.',
color="blue")
plt.plot(train_sizes,valid_scores.mean(axis=1),      label="Validation",      marker='.',
color="orange")
plt.fill_between(train_sizes,train_scores[0]-train_scores[1],train_scores[0]-
train_scores[2])
plt.ylim(0,1)
plt.title('Learning curve of the VotingClassifier model')
plt.legend(loc='lower right')
plt.xlabel('Observations in the training set')
plt.ylabel('Accuracy');
plt.savefig('VC Learning Curve.jpeg')

# Annexes

## Normality test for sleep duration

from scipy.stats import shapiro, f_oneway

x_male=data[data["Troub_sommeil_Yes"]==1]['BMI']
x_female=data[data["Troub_sommeil_Yes"]==0]['BMI']
print(shapiro(x_male))
print(shapiro(x_female))

## Anova test for Sleep duration

```

```
f_oneway(x_male,x_female)
```

```
##### Application #####
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import os
import warnings
from math import modf
import seaborn as sns
from scipy.stats import kruskal, chi2_contingency
import datetime as dt
import streamlit as st
import base64
import pickle
```

```
warnings.filterwarnings('ignore')
```

```
def encode_yn(var):
    var = np.select([var=='Yes'], [['1', '0']], ['0', '0'])
    return var.tolist()
```

```
def encode_true(var):
    var = np.select([var==True], [['1', '0']], ['0', '0'])
    return var.tolist()
```

```
warnings.filterwarnings('ignore')
```

```
st.title("Sleep Disorders Prediction Application")
st.subheader('By YAHIAOUI Cherif, NOIRET Mathis')
st.write("")
st.write("This applications aims to help people determine whether or not they are subject to sleep disorders. Fill the questionnaire below to determine if you are subject to sleep disorders.")
st.write('---')
```

```
st.subheader('1. Select your gender.')
options=['Male', 'Female']
sex = st.radio(label="", options=['Male', 'Female'], horizontal=True)
sex=np.where(sex=='Male', 1, 0)+1-1
```

```

st.subheader('2. Select your age.')
age=st.slider(label="", min_value=18, max_value=80)

```

```

st.subheader('3. Select your education level.')
options=[" --- Select your Diploma --- ",
         "No Diploma",
         "Highschool Diploma",
         "College Degree",
         "College Graduate"]
educ=st.selectbox(label="", options=options)
educ=np.select([educ==i for i in options],
               [[0,0,0,1],
                [0,1,0,0],
                [0,0,1,0],
                [0,0,0,0],
                [1,0,0,0]],[0,0,0,1])

```

```

st.subheader('4. Select your Income/Pauverty ratio level.')
options=[" --- Select your ratio --- ",
         "I/P < 1",
         "I/P < 2",
         "I/P < 3",
         "I/P < 4",
         "I/P < 5",
         "I/P > 5"]
pauverty=st.selectbox(label="", options=options)
pauverty=np.select([pauverty==i for i in options],
                   [[0,0,0,0,0,0],
                    [0,0,0,0,0,0],
                    [1,0,0,0,0,0],
                    [0,1,0,0,0,0],

```

```
[0,0,1,0,0,0],  
[0,0,0,1,0,0],  
[0,0,0,0,1,0]], [0,0,0,0,0,0])
```

```
st.subheader('5. Do you have high cholesterol level?')  
high_chol=st.selectbox(label="", options=[" --- Selection --- ",  
                                           "Yes",  
                                           "No"],key="chol")  
high_chol=encode_yn(high_chol)
```

```
st.subheader('6. What is your diet quality?')  
options=[" --- Selection --- ",  
         "Very good",  
         "Good",  
         "Average",  
         "Bad",  
         "Very Bad"]  
dietq=st.selectbox(label="", options=options,key='diet')  
dietq=np.select([dietq==i for i in options],  
                [[0,0,0,0],  
                 [0,0,0,1],  
                 [0,0,1,0],  
                 [0,1,0,0],  
                 [1,0,0,0],  
                 [0,0,0,0]], [0,0,0,0])
```

```
st.subheader('7. Do you have an health insurance?')  
health_i=st.selectbox(label="", options=[" --- Selection --- ",  
                                           "Yes",  
                                           "No"], key="oezijaoi")  
health_i=np.where(health_i=='Yes', [1, 0], [0,0])
```

```

st.subheader('8. How would you rate your global health status?')
options=[" --- Selection --- ",
         "Very good",
         "Good",
         "Average",
         "Bad",
         "Very Bad"]
health=st.selectbox(label="", options=options,key="ghealth")
health=np.select([health==i for i in options],
                [[0,0,0,0],
                 [0,0,0,1],
                 [0,0,1,0],
                 [0,1,0,0],
                 [1,0,0,0],
                 [0,0,0,0]])

```

```

st.subheader('9. Do you suffer from mental health problems?')
options=[" --- Selection --- ",
         "Yes",
         "No"]
health_m=st.selectbox(label="", options=options,key="health_m")
health_m=encode_yn(health_m)

```

```

st.subheader('10. Do you practice intense physical activity?')
options=[" --- Selection --- ",
         "Yes",
         "No"]
aphys_fort=st.selectbox(label="", options=options,key="aphysfort")
aphys_fort=encode_yn(aphys_fort)

```

```

st.subheader('11. Do you practice moderate physical activity?')

```

```
options=[" --- Selection --- ",
        "Yes",
        "No"]
aphys_mod=st.selectbox(label="", options=options,key='aphysmod')

aphys_mod=encode_yn(aphys_mod)
```

```
st.subheader('12. How many hours a day do you stay in state of sedentarity? (sleep
excluded)')
sedentarity=st.slider(label="", min_value=0, max_value=24, key='sedent')
```

```
st.subheader('13. What is your average sleep duration (in hours)?')
sleep_h=st.slider(label="", min_value=0, max_value=24, key='sleep_sem')
```

```
st.subheader('14. Do you smoke?')
options=[" --- Selection --- ",
        "Yes",
        "No"]
smoke=st.selectbox(label="", options=options,key='smoke')
smoke=np.where(smoke=="Yes", 1, 0)+1-1
```

```
st.subheader('15. Are you subject to diabete?')
options=[" --- Selection --- ",
        "Yes",
        "No"]
diabete=st.selectbox(label="", options=options,key='diabete')
diabete=encode_yn(diabete)
```

```

st.subheader('16. How often do you eat fastfood?')
options=[" --- Selection --- ",
         "Never",
         "Rarely",
         "Often",
         "Very Often"]
fastfood=st.selectbox(label="", options=options,key="fastfood")

fastfood=np.select([fastfood==i for i in options],
                  [[0,0,0],
                   [0,0,0],
                   [0,1,0],
                   [1,0,0],
                   [0,0,1]])

```

```

st.subheader('17. How often did you consult a doctor in the past year?')
options=[" --- Selection --- ",
         "Never",
         "1 time",
         "2 times",
         "3 times",
         "4 times",
         "5 times",
         "6 times",
         "7 times",
         "8 times or more"]
nb_doctor=st.selectbox(label="", options=options)

nb_doctor=np.select([nb_doctor==i for i in options],
                  [0, 0, 1, 2, 3, 4, 5, 6, 7, 8])

```



```
st.subheader('18. How many times do you wake up to pee during night?')
```

```
options=[" --- Selection --- ",
```

```
        "Never",
```

```
        "1 time",
```

```
        "2 times",
```

```
        "3 times or more"]
```

```
pee_per_night=st.selectbox(label="", options=options, key="pee")
```

```
pee_per_night=np.select([pee_per_night==i for i in options],
```

```
                        [[0,0,0],
```

```
                        [0,0,0],
```

```
                        [1,0,0],
```

```
                        [0,1,0],
```

```
                        [0,0,1]])
```

```
st.subheader('19. Check the following cases if you have already been subject to :')
```

```
col1, col2, col3, col4 = st.columns(4)
```

```
with col1:
```

```
    asthma=st.checkbox("Asthma")
```

```
    heart_d=st.checkbox("Heart disease")
```

```
with col2:
```

```
    corona=st.checkbox("Corona disease")
```

```
    heart_a=st.checkbox("Heart attack")
```

```
with col3:
```

```
    cva=st.checkbox("CVA")
```

```
    thyroid=st.checkbox("Tyroid")
```

```
with col4:
```

```
    cancer=st.checkbox("Cancer")
```

```
disease=[*encode_true(asthma),    *encode_true(heart_d),    *encode_true(corona),  
*encode_true(heart_a), *encode_true(cva), *encode_true(thyroid), *encode_true(cancer)]
```

```
st.subheader('20. How often do you feel depressed?')
options=[" --- Selection --- ",
         "Never",
         "Sometimes",
         "Often"]
depressed=st.selectbox(label="", options=options,key="depressed")

depressed=np.select([depressed==i for i in options],
                   [[0,0,0],
                    [1,0,0],
                    [0,1,0],
                    [0,0,1]])
```

```
st.subheader('21. Do you suffer from attention trouble?')
options=[" --- Selection --- ",
         "Never",
         "Sometimes",
         "Often"]
att_tr=st.selectbox(label="", options=options,key='atttr')

att_tr=np.select([att_tr==i for i in options],
                 [[0,0,0],
                  [1,0,0],
                  [0,1,0],
                  [0,0,1]])
```

```
st.subheader('22. Do you have intense negative thoughts (suicide thoughts,
mutilation)?')
options=[" --- Selection --- ",
         "Never",
         "Sometimes",
         "Often"]
```

```
negative_th=st.selectbox(label="", options=options,key='negative_th')
negative_th=np.select([negative_th==i for i in options],
                      [[0,0,0],
                       [1,0,0],
                       [0,1,0],
                       [0,0,1]])
```

```
st.subheader('23. Do you stop breathing during sleep?')
options=[" --- Selection --- ",
         "Never",
         "Rarely",
         "Sometimes",
         "Often"]
stop_breathing=st.selectbox(label="", options=options,key='breath')
stop_breathing=np.select([negative_th==i for i in options],
                         [[0,0,0],
                          [0,0,0],
                          [1,0,0],
                          [0,1,0],
                          [0,0,1]])
```

```
st.subheader('24. What is your weight (kg)?')
weight=st.slider(label="", min_value=50, max_value=120)
```

```
st.subheader('25. How tall are you (cm)?')
height=st.slider(label="", min_value=140, max_value=220)
```

```
bmi = weight/((height/100)**2)
```

```
lists_of_vars=[*[sex],
                *educ,
                *pauverty,
                *high_chol,
                *dietq,
                *health_i.tolist(),
                *health,
                *health_m,
                *aphys_fort,
                *aphys_mod,
                *[0],
                *[smoke],
                *diabete,
                *fastfood,
                *pee_per_night,
                *disease,
                *depressed,
                *att_tr,
                *negative_th,
                *stop_breathing,
                *[age],
                *[sedentarity],
                nb_doctor.tolist(),
                *[bmi],
                *[sleep_h]]
```

```
st.divider()
```

```
coll1, coll2, coll3 = st.columns(3)
```

```
with coll1:
```

```
    st.header('Your results')
```

```
with coll3:
```

```
    st.write("")
```

```
    button=st.button('Make prediction.')
```

```
st.divider()
```

```

def sleep_graph():
    data=pd.read_csv('sleep_h_norm.csv')
    sns.kdeplot(data=data['Norm_Values'])

def results(positive=False):

    if positive==False:
        st.success('Congratulations ! You probably do not suffer from Sleep Disorders')
        #st.pyplot(sleep_graph())
        proba = model.predict_proba(np.array(lists_of_vars).reshape(1,-1))[:,1][0]
        st.write(f'Your probability of suffering from Sleep disorders is
{round(proba*100,2)}%')
        colll1, colll2 = st.columns(2)
        barplot=pd.DataFrame({'x':[0,1], 'y':[71.1, 28.9]})
        st.bar_chart(data=barplot['y'])
    else:
        st.warning('Be careful ! You probably do suffer from Sleep Disorders ! Please
contact a doctor.')
        #st.pyplot(sleep_graph())
        proba = model.predict_proba(np.array(lists_of_vars).reshape(1,-1))[:,1][0]
        st.write(f'Your probability of suffering from Sleep disorders is
: {round(proba*100,2)}%')

```

```

if button==True:

```

```

    with open('VCmodel.pkl', 'rb') as f:
        model = pickle.load(f)

```

```

    pred = model.predict(np.array(lists_of_vars).reshape(1,-1))

```

```

    if pred==0:
        results(positive=False)
    else:

```

```
results(positive=True)
```