Prédiction de faibles oscillations du cerveau en sommeil profond

Master 2 - Ingéniere Statistique

Data challenge

Auteurs :
M. Aymeric LECHEVRANTON
M. Benjamin MARTINEAU
M. Antoine GAUTHIER

Encadrant :
M. Bertrand MICHEL

Version du
2 décembre 2019

Table des matières

1 Introduction

Ce rapport présente notre participation au Challenge Data 2019 à travers le projet Dreem qui à pour objectif de prédire l'oscillation lente du sommeil profond du cerveau [1].

Selon l'étude de l'INSV et la MGEN en 2015, parmi les Français interrogés un tiers déclare avoir un trouble du sommeil. Dans cette population interrogée, c'est 73% qui prétendent se réveiller au moins une fois dans la nuit, environ 30 minutes. Cette étude a également montré l'impact des nouvelles technologies sur la qualité du sommeil. En effet, l'exposition tardive et prolongée à des appareils électroniques (télévision, smartphone, tablette, ordinateur) détériore la qualité du sommeil par son effet stimulant et addictif. Le sommeil est caractérisé par différents cycles : sommeil léger, sommeil profond, sommeil paradoxal et le réveil. Parmi ces différentes phases, le sommeil profond joue un rôle important car il participe à la consolidation de la mémoire, la restauration d'énergie et la libération d'hormones. De plus, outre les nouvelles technologies, le sommeil profond est directement impacté par l'âge, le stress ou encore la maladie. Pour toutes ces raisons, la qualité du sommeil est un enjeu de santé publique et comprendre les cycles de sommeil pourrait être d'une grande aide.

Plusieurs méthodes existent pour analyser le sommeil : l'activité électrique dans le cerveau est mesurée par électroencéphalographie (EEG), l'activité musculaire par électromyographie (EMG), les mouvements oculaires par électro oculographie, ... Mais toutes ses méthodes sont souvent difficiles à mettre en place et inconfortables pour le patient. Pour cela, l'entreprise Dreem, fondée en 2015 par deux ingénieurs polytechnicien, propose un service visant à améliorer la qualité du sommeil grâce la transmission de signaux sonores par conduction osseuse. Pour ce faire, elle a créé un bandeau capable de mesurer, par le biais de capteurs (EEG, accéléromètre, pulsomètre) les phases de sommeil de façon pratique et confortable. Le bandeau Dreem analyse tout au long de la nuit les signaux afin de prédire les oscillations et plus particulièrement les ondes de grandes magnitudes et de basses fréquences qui sont caractéristiques du sommeil profond.

L'objectif principal de ce Data Challenge - Dreem consiste à prévoir la présence ou non de faible oscillation dans la seconde qui suit un enregistrement EEG. Chaque enregistrement représente 10 secondes d'enregistrement commençant 10 secondes avant la fin d'une oscillation lente. Dans ce rapport, nous vous présenterons les données et effectuerons une analyse de celles-ci. Nous parcourrons ensuite les premières méthodes algorithmiques utilisées et leurs résultats. Puis, nous concentrerons nos explications sur le modèle retenu en justifiant sa genèse et en présentant ses performances.

Vous trouverez, joint à ce rapport, un notebook Jupiter retraçant les étapes de notre projet programmé en Python. Ce rapport a été rédigé sur Overlief et le notebook écrit en collaboration à l'aide de l'environnement Colaboratory.

1. `https://challengedata.ens.fr/participants/challenges/10/ranking/public`

2 Présentation du jeu de données

Pour ce défi, nous disposons de 500 000 enregistrements EEG récoltés sur 1699 utilisateurs et acquit à partir du bandeau Dreem. Ces données ont été découpés en deux échantillons (train-test). L'échantillon train contient de 261 634 enregistrements récupérés sur 850 utilisateurs. L'échantillon test contient 238 366 enregistrements récupérés sur 879 utilisateurs. Les enregistrements que nous avons à disposition sont d'une durée de 10 secondes et ont une fréquence d'échantillonnage de 125Hz (ce qui équivaut à 1250 variables). Nous disposons également de 11 variables ayant pour but de synthétiser le sommeil avant le début de l'enregistrement. Les voici :

- Nombre d'oscillations lentes précédent l'enregistrement
- Amplitude moyenne des oscillations lentes précédent l'enregistrement
- Durée moyenne des oscillations lentes précédent l'enregistrement
- Amplitude de l'oscillation lente actuelle
- Durée de l'oscillation lente actuelle
- Étape de sommeil actuelle [2]
- Temps écoulé depuis que la personne s'est endormie
- Temps passé en sommeil profond jusqu'à présent
- Temps passé en sommeil léger jusqu'à présent
- Temps passé en sommeil paradoxal (REM) jusqu'à présent
- Temps passé en phase de réveil jusqu'à présent

L'objectif est d'effectuer une classification en trois classes (0,1,2) :

- Aucune oscillation lente ne commence dans la seconde qui suit
- Une oscillation lente de faible amplitude commence dans la seconde qui a suivi
- Une oscillation lente de forte amplitude commence dans la seconde suivante

Pour résumer nous disposons d'un échantillon d'apprentissage de taille $261\,634 \times 1261$ grâce auquel nous devons effectuer une classification en trois classes d'un jeu de données de taille $238\,366 \times 1261$.

3 Analyse des données

Dans cette partie, nous avons choisi de découper nos données en deux : les 11 premières variables et les variables de l'EEG.

3.1 Analyse en composantes principales (ACP)

Compte tenu du grand nombre d'informations, nous avons souhaité synthétiser l'information contenue dans ces variables. Pour cela, nous avons choisi d'effectuer une ACP. De plus, nous pensions que cela nous aurait permis de détecter d'éventuels individus extrêmes. Cependant comme le montre la figure 1 très peu d'individus se détachent du groupe. De plus, en coloriant les individus en fonction de leur label (valeurs de y), il n'est pas possible de distinguer des regroupements d'individus. Nous constatons également que beaucoup d'informations est détenu par le premier axe (80,48% de l'inertie). Ce qui veut dire que le graphique est très représentatif. En ce qui concerne les données issues de l'EEG, le résultat est similaire : pas de regroupement

2. variable qualitative

en fonction des classes et pas d'individu qui se distingue significativement (cf. figure 2). Mais à l'inverse des 11 premières variables, le pourcentage de variance explique est faible pour les axes 1, 2 (13,36%). Cela est du au grand nombre de variable. Afin de sélectionner d'éventuels individus extrêmes, nous regarderons les axes suivants. Mais l'information transmise reste identique.

FIGURE 1 – Analyse en composante principale des 11 premières variables.

FIGURE 2 – Analyse en composante principale des variables de l'EEG.

3.2 Analyse de la corrélation

Toujours avec l'objectif de mieux connaître notre jeu de données, nous avons choisi de regarder la corrélation entre les différentes variables. En ce qui concerne les 11 premières variables, les corrélations sont relativement faibles mis à part pour les variables 0, 8, 9 et 10 qui ont des indices de corrélation supérieurs à 0.6. Le temps passé en sommeil profond, léger, paradoxal ainsi que le nombre d'oscillations lentes précédant l'enregistrement, évolueraient donc d'une façon similaire (cf. figure 3). Concernant les variables de l'EEG, la corrélation entre les variables est uniquement existante entre variables proches les unes de autres et ceci s'explique simplement par le fait qu'elles sont liées par le tracé continue de l'EEG. En Annexe, la Figure 14 montre bien qu'il n'existe pas de lien avec des variables éloignées.

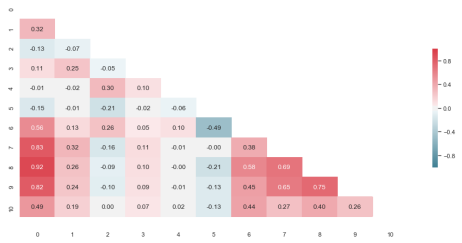


FIGURE 3 – Corrélation entre les 11 premières variables

Pour compléter ces deux approches, nous avons regardé plus en détail le comportement des différentes variables en fonction des labels. Pour cela nous avons mis en oeuvre différentes techniques descriptives : boxplot, courbe de moyenne et de variance, étude des minima et maxima,... A travers cette analyse, les résultats les plus satisfaisants concernent les courbes de moyennes. En effet, lorsque l'on regarde les moyennes des EEG par classe, nous constatons que les moyennes par groupes sont identiques et égales à 0 uniquement jusqu'à environ la variable

1000. De plus, nous constatons que les courbes se formant pour chaque label prennent des valeurs différentes surtout aux niveaux des pics (cf. figure 4). Un second point a attiré notre attention : certaines valeurs maximales et minimales des EEG ne sont pas du même ordre de grandeur que les autres (cf. tableau 1).

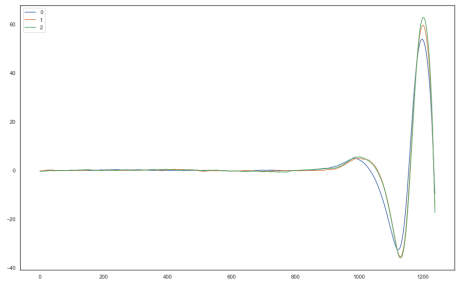


FIGURE 4 – Plot des valeurs moyennes de l'EEG par groupe

	Maxima	Minima
1	2074.31	-5312.12
2	1589.10	-1988.31
3	1251.41	-1850.55
4	1167.81	-1519.64
5	1133.55	-1173.36
7	...	...

TABLE 1 – Valeurs extrêmes des enregistrements EEG

Dans l'idée que cela pourrait apporter du bruit dans le jeu de données, nous avons conservé la localisation de tous les individus et variables qui nous paraissaient anormaux. Cela nous permettra de tester nos modèles avec et sans ces valeurs. Si en effet ces individus/variables apportent du bruit, nous nous attendons à une amélioration des résultats.

4 Ajout de données complémentaires

N'obtenant pas de résultats satisfaisants en travaillant seulement sur le jeu de données initiale, nous avons choisi d'étudier plus en détail les signaux EEG. Les signaux enregistrés sont des électroencéphalogramme, ou EEG, et représente l'activité du cerveau pendant 10 secondes au cours d'une période de repos, sans stimulation sonore favorisant le sommeil profond.

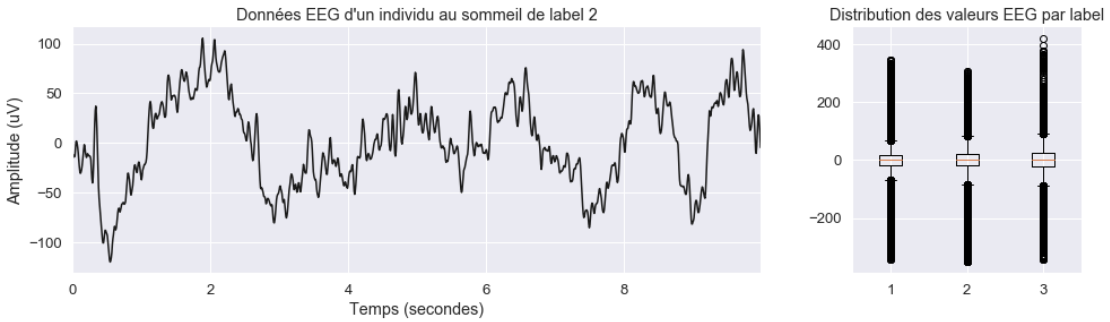


FIGURE 5 – Électroencéphalogramme d'un individu ayant eu une faible oscillation de haute amplitude dans la seconde suivante.

A partir de ces EEG, nous avons réalisé différentes manipulations de traitement du signal notamment par la décomposition des signaux en différentes bandes de fréquences distinctes sur le plan fonctionnel. En effet, il existe différents rythmes cérébraux dont 4 qui jouent un rôle

prépondérant dans l'analyse cérébrale du sommeil. Ces bandes de fréquences, présentées dans la Table 2, sont les suivantes : Delta, Thêta, Alpha et Bêta. La bande de fréquences Delta est celle qui nous intéresse le plus car elle permet de se focaliser sur les faibles oscillations (qui prédominent en cas de sommeil profond). Pour effectuer une décomposition du signal pour une bande de fréquences, la méthode la plus utilisée est la transformation de Fourier qui permet de de calculer une estimation de la densité spectrale de puissance (ou périodogramme). Pour faire de la décomposition du signal en bande de fréquences, la méthode la plus utilisée est la transformation de Fourier qui permet de de calculer une estimation de la densité spectrale de puissance (ou périodogramme). En ce qui nous concerne, nous souhaitons calculer la puissance de bande moyenne propre à chaque bande de fréquence et pour cela, la méthode la plus communément utilisé est le périodogramme de Welch. Cette méthode consiste à faire la moyenne de transformation de Fourier consécutive par fenêtres. Ce procédé permet de réduire la variance des puissances individuelles aux dépens de la résolution en fréquence. Une seconde méthode plus récente est également utilisée pour le calcul des bandes moyennes : la méthode Multitaper. Cette méthode a l'avantage de réduire la variance tout en conservant une bonne résolution. Seul inconvénient, les temps de calcul pour cette méthode sont beaucoup plus longs que pour la méthode de Welch (1 minute Welch $\approx$ 5 heures Multitaper). De plus les résultats des deux méthodes sont relativement similaires pour peu qu'il n'y ai pas ou peu de bruit. La fonction utilisé pour ces calculs est BANDPOWER [3] du package YASA qui permet de choisir la méthode que l'on souhaite utiliser. Dans l'objectif d'obtenir le meilleur score, nous avons choisit d'utiliser la méthode Multitaper.

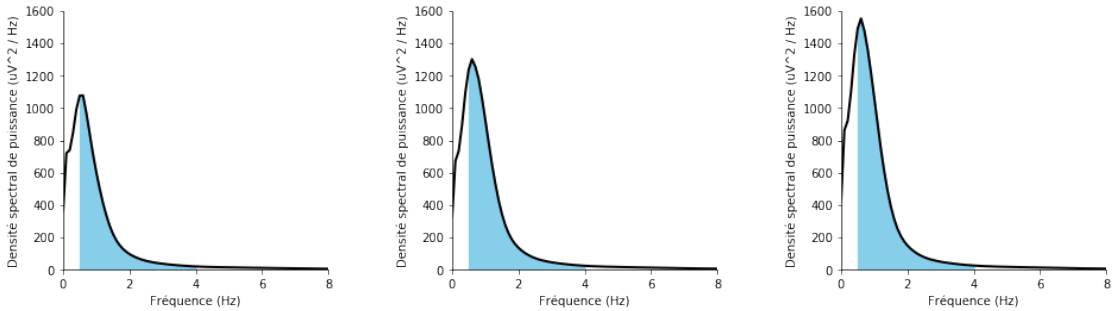
TABLE 2 – Bandes de fréquences issues de la décomposition du signal EEG.

Band	Frequency (Hz)	Caractéristiques
Delta	0.5-4	Sommeil profond ou lésion cérébrale
Thêta	4-8	État de somnolence, hypnose ou méditation
Alpha	8-12	État de conscience apaisé
Bêta	12-30	État d'éveil normal conscient

En complément des bandes spectrales, nous avons approché la densité spectrale de puissance moyenne avec la méthode Multitaper qui nous permet d'avoir un périodogramme plus précis que par la méthode de Welch et moins variables que par la méthode spectrale. Pour cela nous avons utilisé PSD.ARRAY.MULTITAPER du package MNE.TIME.FREQUENCY. Nous remarquons sur ces périodogrammes que la densité spectrale de puissance est particulièrement élevée en basse fréquence (0 à 2 Hz) dans les cas de faible oscillation (Figures 6-b et 6-c). Par ailleurs, l'augmentation de la densité en basse fréquence augmente la densité des faibles oscillations visibles en bleu. Nous pouvons alors supposer que plus il y a de faibles oscillations (entre 0.5 et 4 Hz) dans l'enregistrement de 10 secondes, plus la probabilité d'avoir une faible oscillation dans la seconde qui suit est élevée.

Suite à cette supposition, nous avons trouvé une fonction du package YASA permettant de faire un résumé (grâce à 10 variables (cf. table 3)) des faibles oscillations présentes dans un EEG. Nous avons donc fait tourner cette fonction, appelée SW_DETECT sur la totalité de notre échantillon afin de connaître pour chaque individu le nombre de faibles oscillations présente dans son signal. Dans le but d'obtenir un maximum d'informations, nous avons pour chaque

3. Fonction implémentée à partir de la fonction PSD.ARRAY.MULTITAPER du package MNE.TIME.FREQUENCY.



(a) (b) (c)

FIGURE 6 – Périodogramme par la méthode multitapper pour les cas sans évènement (a), les cas de faibles oscillations à faibles amplitudes (b) et fortes amplitudes (c).

individu conservé la dernière faible oscillation, la durée totale de faibles oscillations détectées dans l'enregistrement et la fréquence moyenne de celles-ci.

Start	End	Duration	Amplitude	RMS	AbsPower	RelPower	Frequency	Oscillations	Symmetry
3.32	4.06	0.74	81.80	19.65	2.72	0.49	12.85	10	0.67

TABLE 3 – Résumé d'un EEG grâce à la fonction SW_DETECT

5 Méthode sélectionnée, variables retenues et résultats obtenus

5.1 Variables sélectionnés pour le modèle

En proposant le challenge, la société Dreem avait proposé une méthode utilisant Random Forest comme Benchmark. Pour établir leur modèle, ils avaient utilisé les 11 premières variables puis avaient choisi de résumer l'information compte tenue dans l'EEG grâce à des moyenne, maximum et minimum globaux.

En ce qui nous concerne, nous avons également conservé les 11 premières variables, mais nous les avons accompagné des 348 dernières variables de l'EEG. Nous avons ensuite ajouté des valeurs de moyenne, médiane, variance, minimun et maximum locaux. Le calcul de ses valeurs s'effectue sur les valeurs des EEG et à l'intérieur des intervalles suivants : [0,500] ; [500,850] ; [850, 1250]. Nous avons ensuite complété le jeu de données par les valeurs différentes de bandes spectrales, du périodogramme Multitaper ainsi que la dernière faible oscillation et le nombre de faibles oscillations présentent dans l'enregistrement. Nous obtenons ainsi un jeu de données de taille : 261 634 individus par 1 029 variables.

5.2 Algorithme de référence

Les méthodes dont nous allons faire la présentation ici reposent sur la sagesse des foules. Cette assertion suppose que la synthèse des réponses de milliers de personnes prises au hasard est souvent meilleure que celle d'un expert. De la même façon, l'agrégation de prédicteurs (classificateurs ou régresseurs) permettent souvent d'obtenir une meilleure prédiction que si nous prenions les meilleurs prédicteurs individuellement. C'est sur ce principe que des méthodes, dites ensemblistes, sont apparues. En d'autres termes, un groupe de prédicteurs constitue un ensemble et sont ainsi à la base des algorithmes d'apprentissage d'ensemble, ou méthode ensembliste.

L'algorithme le plus connu dans cette classe d'algorithmes est la Forêt Aléatoire, ou Random Forest, et c'est cet algorithme qui est utilisé dans le benchmark. Le Random Forest repose sur l'entraînement d'un ensemble d'arbre de décision, ou decision tree, chacun sur un sous-ensemble aléatoire différent du jeu d'entraînement : c'est le bagging [8]. La prédiction finale est la classe ayant obtenu le plus de votes à partir des prédictions de chacun des arbres de décision. Le vote majoritaire, ou vote rigide, n'est pas le seul type de vote, il existe également le vote souple utilisant les probabilités des classes (si les classifieurs utilisés la fournissent) et sélectionnant la classe avec la plus grande moyenne des probabilités sur l'ensemble des classificateurs. En outre, le moyen utilisé pour obtenir un ensemble de classificateurs diversifiés dans la méthode Random Forest est l'utilisation du bagging par l'entraînement d'un même algorithme pour chaque prédicteur (ici l'arbre de décision) mais en l'entraînant sur des sous-ensembles différents extraits aléatoirement du jeu d'entraînement et avec remise. Dans le cas sans remise, on parle de pasting.

Le bagging provoque le fait qu'une partie des observations ne sera pas tirée pour l'entraînement. Ces observations sont appelées hors sélection, ou out-of-bag instances. Sachant que les prédicteurs n'utilisent pas les oob qui leur sont propres durant l'entraînement, ils peuvent être évalués sur leur oob. Pour le benchmark proposé par Dream, le score out-of-bag est de 0,5048. Ce score est proche de celui obtenu au Benchmark de 0,5049.

5.3 Algorithme sélectionné - Light GBM

Tout d'abord, nous avons testé la majorité des algorithmes composant l'état de l'art en Machine Learning : Random Forest, Réseaux de neuronnes artificiels (ANN), Machine à vecteur support (SVM), Régression Logistique, k-plus-proche voisins (kNN), Adaptative Boosting (AdaBoost) et des méthodes ensemblistes regroupant certains algorithmes cités précédement (Stacking). Nous avons également tenté d'utiliser les méthodes basées sur les séries temporelles, mais nous n'avons pas décelé de tendances dans les EEG qui nous auraient permis de continuer dans cette voie. Néanmoins, l'algorithme qui a montré les meilleurs performances sur notre jeu de données s'est avéré être Light GBM. Cette algorithme est une version améliorée de l'algorithme XGBoost [1, 7].

L'algorithme XGBoost repose sur les méthodes de boosting. Le principe du boosting est un ajout séquentiel de prédicteurs parmi un ensemble de prédicteurs qui vont chacun tenter de corriger les prédictions faites par leurs prédécesseurs sur les observations d'entraînement. Pour ce faire, et contrairement à l'algorithme AdaBoost qui modifie le poids des observations à chaque itération, XGBoost va ajuster un nouveau modèle sur les erreurs résiduelles du prédicteur

précédent (les prédicteurs sont des arbres de décisions). Cette manière de former les arbres, qui caractérise les algorithmes Gradient Boosting Decision Tree, est néanmoins coûteuse en temps de calcul. Par ailleurs, XGBoost et Light GBM utilisent tous les deux le Histogram based splitting pour trouver le découpage optimal à chaque noeud et de manière moins coûteuse que dans les méthodes classiques de gradient boosting utilisant le Pre-sorted algorithm [4]. XGBoost met également en place un échantillonnage des variables dans la construction des arbres en plus de l'échantillonnage sur le nombre de noeuds terminaux dans les arbres. Ces ajouts réduisent la corrélation entre les arbres, augmentent le biais mais réduisent la variance. De plus, la méthode utilisée pour arriver au minimum est l'utilisation du Newton Boosting qui se base sur la méthode d'approximation de Newton-Raphson [5]. Cependant, des améliorations ont été apporté ces dernières années à l'algorithme XGBoost, le rendant plus performant et moins coûteux en temps de calcul : c'est le cas de l'algorithme Light GBM.

La particularité de Light GBM réside principalement dans sa capacité à réduire la complexité de la construction des histogrammes utilisés dans le splitting des arbres. Dans cet objectif, Light GBM utilise deux techniques de sous-échantillonage des données et des variables pour la création de l'histogramme, appelés Gradient Based One Side Sampling (GOSS) et Exclusive Feature Bundling (EFB). Ces techniques permettent de réduire drastiquement la complexité des calculs. Le GOSS est un échantillonnage unilatéral basé sur le gradient et part du principe que tous les points ne contribuent pas de manière égale à l'entraînement. En effet, les données à faibles gradients ont tendance à être mieux entraîné car elles sont proches d'un minimum local. Le GOSS se concentre donc sur les données avec de forts gradients, car c'est là où la marge de progression du modèle est la plus élevée. Ainsi, lors du splitting, la technique choisie est de créer un échantillonnage d'importance comportant des données à fort gradient et quelques données à faible gradient, pour ne pas modifier trop fortement la distribution des données en pratiquant un échantillonnage biaisé (le cas si on supprimait simplement les données à fort gradient). D'autre part, Light GBM utilise le EFB qui consiste à construire des variables exclusives. Le EFB cherche à détecter des variables qui sont fragmentées, c'est-à-dire qu'elles ne sont jamais égales à zéros ensemble. De ce fait, le EFB va les regrouper pour ne former plus qu'une seule variable. Cette opération permet de diminuer les temps de calcul sans perdre aucune information [6].

Après avoir obtenu les meilleurs scores avec light GBM, nous avons tenté d'optimiser les paramètres. Pour cela, nous avons configuré le taux d'apprentissage à 0.1, le nombre d'estimateurs à 162, l'état aléatoire de départ à 33 et le maximum de profondeur des arbres à 10. Ainsi nous avons pu gagner quelques places dans le classement et augmenter notre score.

5.4 Résultat et analyse du modèle

L'algorithme Light GBM a été sélectionné comme le plus performant. C'est ce que nous prouvons dans la Table 4 où nous remarquons effectivement que Light GBM se distingue clairement des autres algorithmes en termes de temps de calcul et en terme de performance. En effet, quand XGBoost met 40 minutes pour obtenir un résultat, Light GBM met moins de 3

4. Histogram based splitting transforme les données en données binaires ce qui facilite grandement les calculs par rapport au Pre-sorted algorithm qui énumère tous les points de partage possibles sur les données pré-triées.

5. Cette méthode prend un chemin plus direct que la descente de gradient car elle utilise une approximation de la dérivée seconde.

6. Le problème pour trouver les regroupement les plus efficace étant NP-hard, il est résolu par un algorithme d'approximation.

minutes et obtient un meilleur résultat. Ces résultats sont toutefois à nuancer car obtenus dans une démarche train-test classique pour limiter le temps de calcul [7] inhérent à une validation croisée, mais reste cohérent dans le cadre d'une validation croisée. En effet, nous avons effectuer des validations croisées à 8-folds sur Light GBM uniquement et les scores obtenues était en moyenne plus élevées pour la version ajustée par rapport à la version non-ajustée. En outre, notre meilleur modèle est Light GBM ajusté avec 182 arbres, une profondeur maximale de 22 étages pour chaque arbre et un alea fixé à 33. Cet algorithme nous a permis d'obtenir un score de 0.5338 et de nous placer en 3ème position de ce Challenge sur 96 participants recensés comme le montre la Figure 11 en Annexe.

Algorithmes	Random Forest non-ajusté	Random Forest ajusté	XGBoost non-ajusté	XGBoost ajusté	Light GBM non-ajusté	Light GBM ajusté
Temps de calcul (en secondes)	34.94	357.74	1425.52	2439.15	118.00	160.16
Score (train-test)	0.4710	0.5133	0.5284	0.5299	0.5322 (DC - 0.5330)	**0.5309 (DC - 0.5338)**

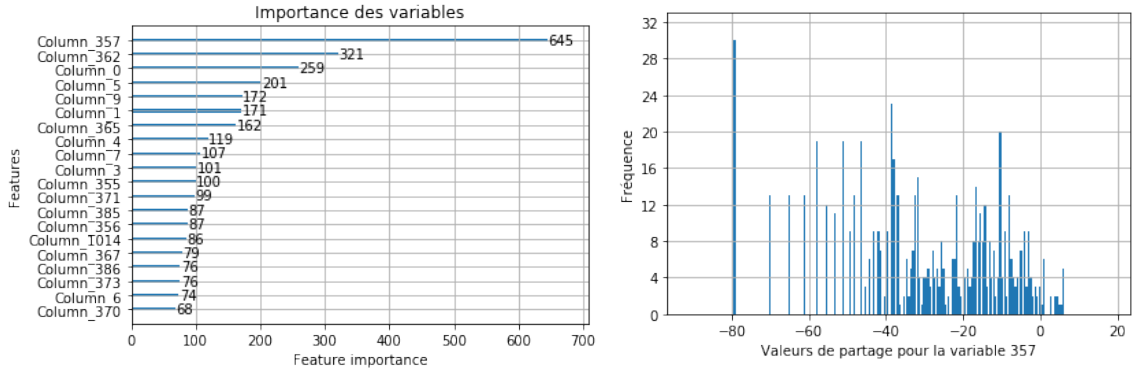
TABLE 4 – Temps de calcul et scores des principaux algorithmes sélectionnés

Dans un but d'ajustement performant et à partir de l'algorithme Light GBM, nous avons pu obtenir plus d'informations sur les variables importantes dans la construction du modèle et donc les plus discriminantes. La Figure 7-a montre que la variable 357 (la dernière valeur de l'EEG) est la plus importante dans la construction des arbres. Par ailleurs, nous observons que 4 variables sont présentes plus de 200 fois et 11 variables sont présentes plus de 100 fois dans les constructions d'arbres. En se penchant sur la Figure 7-b, nous remarquons que les valeurs de la variables 357 utilisées pour la division des branches des arbres apparaissent sans distribution particulière. Un exemple d'arbre, l'arbre d'indice 0, est disponible en Annexe (Figure 10). Malgré des tentatives multiples pour ajuster le modèle en supprimant les variables les moins importantes, nous avons toujours eu un meilleur modèle de prévision en ajoutant des données qu'en en supprimant. L'objectif de cet ajustement étant d'obtenir le meilleur score au Data Challenge, nous avons par conséquent choisi de prendre le plus de données dans notre modèle tant que celles-ci augmentaient les résultats.

La matrice de confusion de la Table 5, nous montre que notre algorithme n'est toutefois pas efficace dans ses prévisions comme nous avions pu nous en rendre compte lors de l'ajustement complexe et peu efficace en terme de résultats. Nous remarquons qu'une grande majorité des prévisions sont faites sur le label 0. C'est en effet le label le plus présent dans les jeux de données train et test fourni par Dreem. En d'autre terme, l'algorithme n'arrive pas à discriminer efficacement les différents labels, ni même les différents états que sont l'absence de faible oscillation (label 0) et la présence de faibles oscillations (labels 1 et 2).

La Figure 8 représentant les courbes ROC des 3 labels renforce nos observations précédemment effectuées sur la matrice de confusion. En effet, nous pouvons voir que les prévisions sont mauvaises et très proche du hasard pour les labels 1 et 2 où le score serait de 33% dans le cas du hasard et de 44% si l'on ne prédisait que le label 0. Les courbes ROC des Figures 8-b-c renforce ces observations : l'AUC [8] moyen pour les labels 1 et 2 sont respectivement de 0.65 et 0.7. A travers la Figure 8-a, nous remarquons également que le label 0 est le mieux prédit avec

7. Le découpage train-test est le même pour tous les calculs.
8. Area Under the Curve



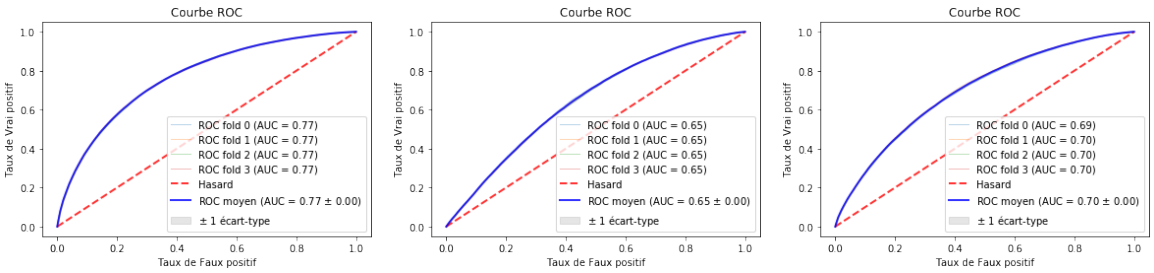
(a) (b)

FIGURE 7 – Variables les plus fréquentes pour diviser les arbres (a) et l'histogramme des valeurs de la variables 357 qui divisent les arbres (b).

		Prévision			Effectif Total
		0	1	2	
Valeurs	0	0.784 (17780)	0.147 (3338)	0.068 (1552)	22670
	1	0.437 (6891)	0.351 (5538)	0.212 (3339)	15768
	2	0.356 (4950)	0.322 (4475)	0.321 (4464)	13889
Effectif Total		29621	13351	9355	52327

TABLE 5 – Matrice de confusion en effectif et en proportion de prévision sur le nombre réel.

un AUC moyen de 0.77, mais cela est due principalement au fait qu'il soit majoritaire dans les jeux de données fournis. Par ailleurs, le jeu de données étant très grands et les EEG sensiblement dans le même ordre de grandeur, la variabilité des courbes ROC par validation croisée à 4-folds est très proche de 0. Cependant, nous avons également effectué cette démarche pour les 1000 premiers individus du jeu de données et une faible variabilité est toutefois bien présente (Figure 13 en Annexe).



(a) (b) (b)

FIGURE 8 – Courbes ROC du label 0 (a), label 1 (b) et label 2 (c).

6 Conclusion

La société Dreem a obtenu un score de 0.5049 en utilisant la méthode RandomForest. Ce un score intéressant car si le modèle était mauvais, le score serait autour de 0.3333. Après avoir testé de nombreuses méthodes et remarquer que les meilleurs résultats provenaient de Light GBM, nous avons tenté d'optimiser les paramètres. Ainsi nous avons obtenu un résultat de 0.5338. Cela ne nous paraissait pas extraordinaire comparer au Benchmark mais relativement correct du point de vue du classement. En effet nous sommes 3ème et le 1er nous devance de 0.0016.

Nous avons trouver ce sujet très enrichissant, il nous a permis de tester des modèles que nous ne connaissions pas sur un jeu de données volumineux. Nous avons souhaité effectuer une démarche en partie tournée vers l'analyse des données mais cela n'a pas spécialement porté ses fruits. En effet, les meilleurs résultats ont été obtenues sans sélection particulière d'individus ou de variables. Seul la sélection d'une partie des variables de l'EEG est issue de l'analyse des données.

Nous pensons avoir du mal à améliorer significativement les résultats en procédant seulement les résultats des EEG. Nous pensons que pour s'améliorer le modèle aurait besoin de nouvelles informations. Cela pourrait être des données enregistrées par le bandeau Dreem : la respiration, le rythme cardiaque, ... Pour aller encore plus loin, nous pourrions imaginer ajouter des données issues d'électro-oculographie, d'électromyographie, ...

Enfin, nous tenons à remercier l'ENS Ulm pour l'organisation de ce Data Challenge et la prospection qu'elle effectue en amont de celui-ci. Nous remercions également Mr Michel pour son enseignement en Apprentissage Statistique qui nous a ainsi permis d'appliquer certaines techniques de Machine Learning.

Références

[1] Tianqi CHEN et Carlos GUESTRIN : Xgboost : A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[2] Peter D. WELCH : The use of fast fourier transform for the estimation of power spectra : A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, pages 70–73, 1967.

[3] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE : *Deep Learning*. Mit press édition, 2016.

[4] Aurélien GÉRON : *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. Dunod édition, 2017.

[5] Trevor HASTIE, Robert TIBSHIRANI et Jerome FRIEDMAN : *The Element of Statistical Learning*. Springer édition, 2013.

[6] INSTITUT NATIONAL DU SOMMEIL ET DE LA VIGILANCE : Sommeil et nouvelles technologies. `https://institut-sommeil-vigilance.org/wp-content/uploads/2019/02/DP_Journee_Sommeil2016.pdf`. Online ; accessed 27 November 2019.

[7] Guolin KE, Qi MENGLIAW, Thomas FINLEY, Taifeng WANG, Wei CHEN, Weidong MA, Qiwei YE et Tie-Yan LIU : Lightgbm : A highly efficient gradient boostingdecision tree. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 3149–3157, 2017.

[8] Vladimir SVETNIK, Andy LIAW, Christopher TONG, J. Christopher CULBERSON, Robert P. SHERIDAN et Bradley P. FEUSTON : Random forest : A classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information and Computer Sciences*, page 43, 2003.

[9] VALLAT, RAPHAËL : Bandpower of a EEG signal. `https://raphaelvallat.com/bandpower.html`. Online ; accessed 01 December 2019.

Annexe

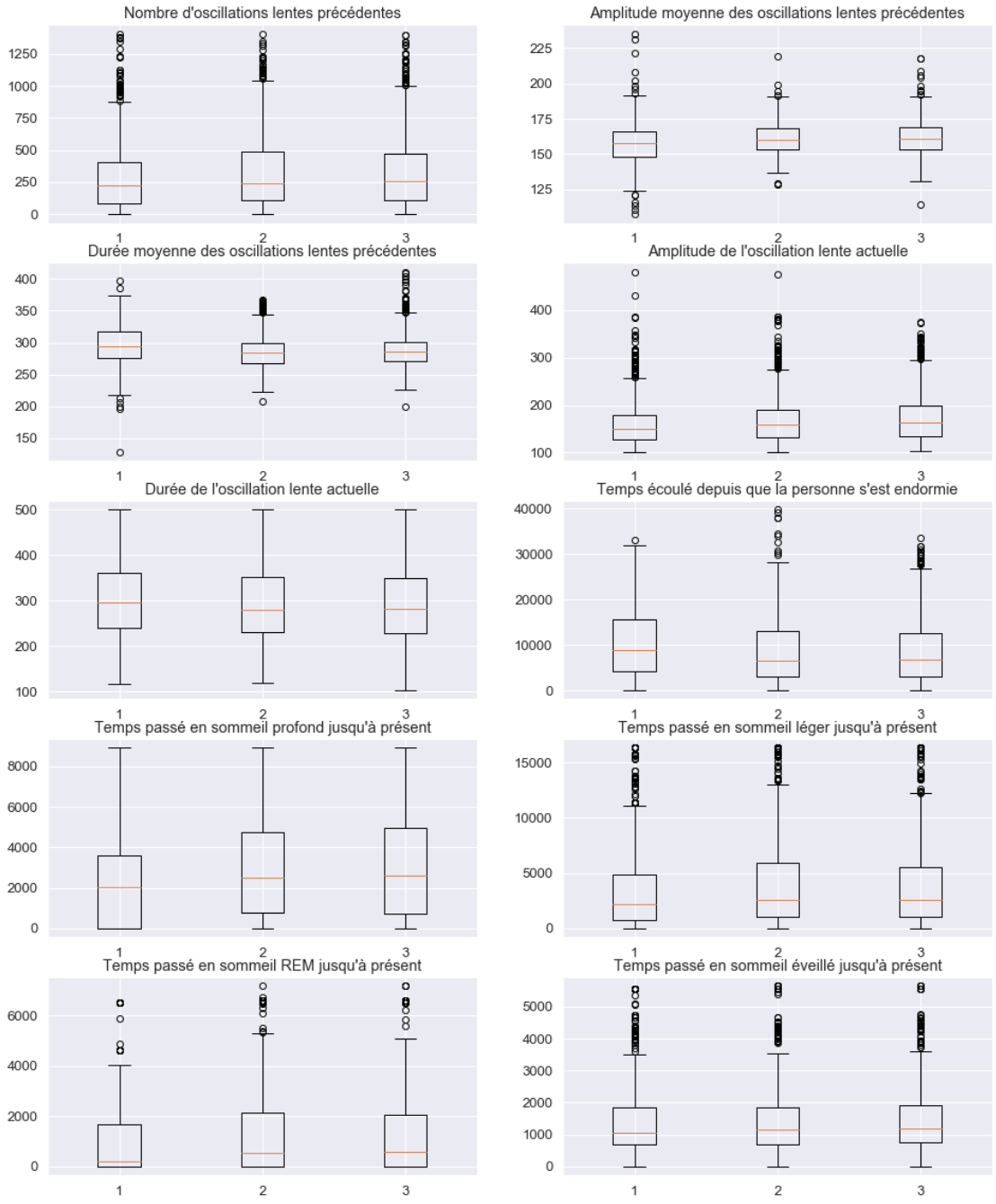


FIGURE 9 – Diagrammes en boîtes des premières variables du jeu de données Dreem

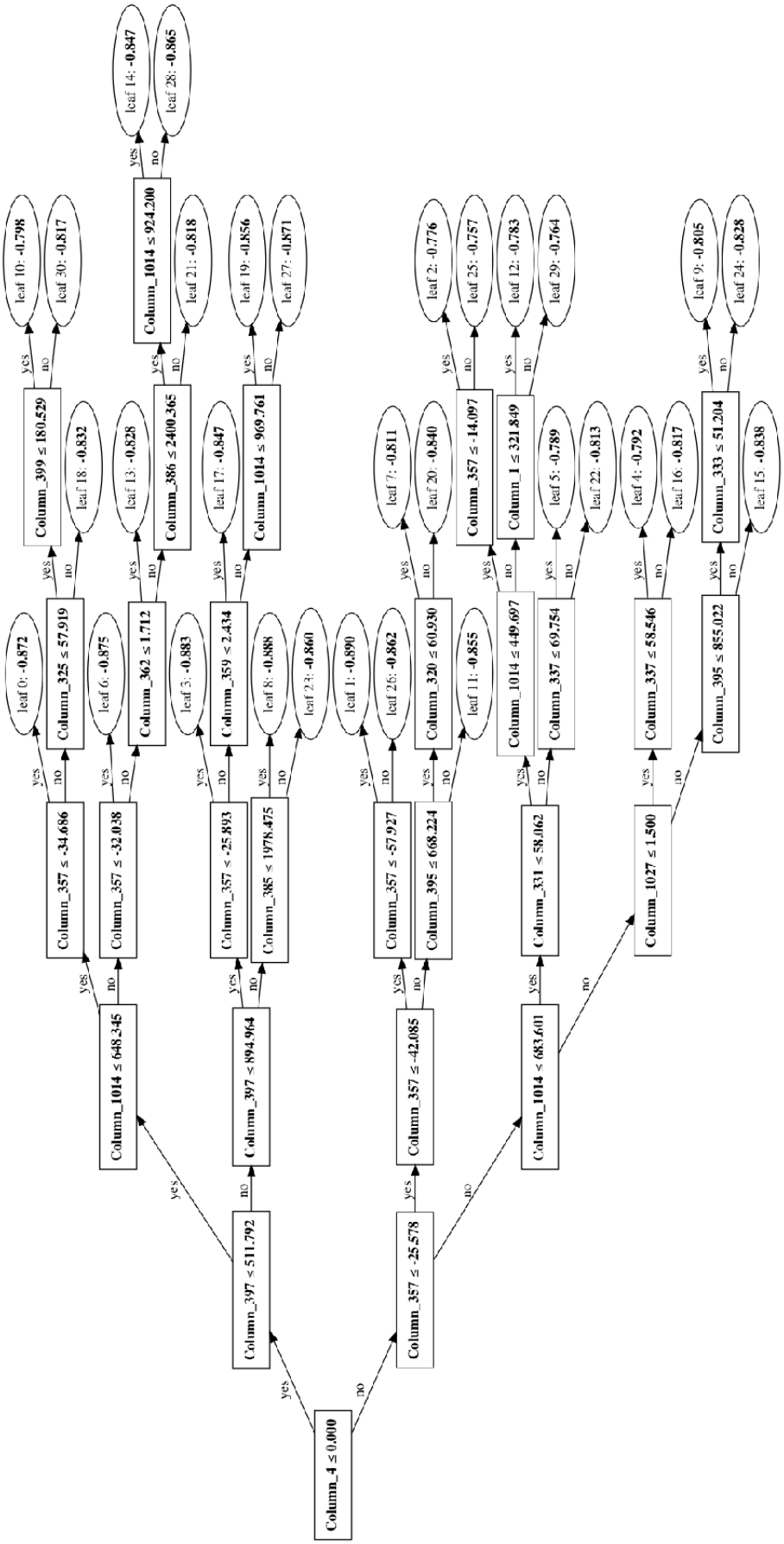


FIGURE 10 – Premier arbre de Light GBM parmi les 182 autres arbres

Predict brain deep sleep slow oscillation
by Dreem
currently running

Introduction | Public ranking | Academic ranking | Intermediate academic ranking | Final ranking | Submissions

This is the public ranking which evolve during the entire season. We use the public score (the score made by participant on the public data set) to set the ranks.

Ranking	Date	User(s)	Public score
1	April 2, 2019, 9:26 p.m.	Armand	0.5354
2	March 28, 2019, 12:16 a.m.	TPoiree	0.5340
3	Dec. 1, 2019, 1:18 p.m.	AymericLechevranton	0.5338
4	April 1, 2019, 8:03 p.m.	gizacard	0.5334
5	Nov. 27, 2019, 11:26 p.m.	Benjamin_Martineau	0.5333
6	Nov. 28, 2019, 2:47 p.m.	hereweare	0.5333
7	Nov. 29, 2019, 12:42 p.m.	rete	0.5333
8	Nov. 30, 2019, 6:52 p.m.	Camben & NRoger	0.5320
9	Nov. 26, 2019, 5:25 p.m.	AntoineGauthier	0.5310
10	Nov. 29, 2019, 3:32 p.m.	lumaluma	0.5304
11	March 26, 2019, 9:07 a.m.	cbensoussan & Mounir	0.5303
12	April 3, 2019, 7:01 a.m.	remysun	0.5293

FIGURE 11 – Classement au dimanche 1 décembre à 21h34

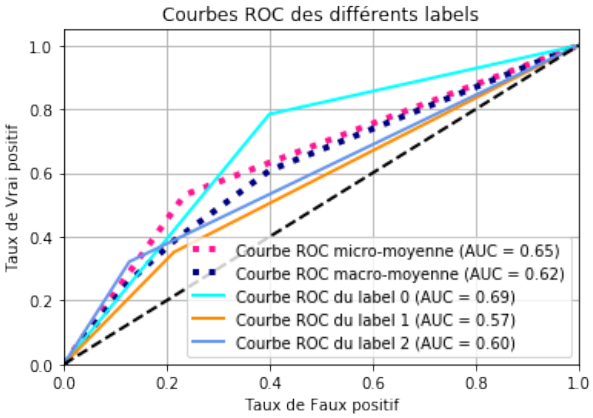


FIGURE 12 – Courbes ROC des 3 labels

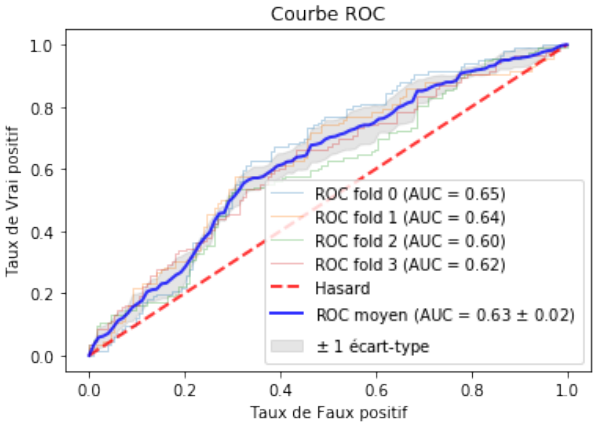


FIGURE 13 – Courbe ROC du label 0 pour les 1000 premiers individus

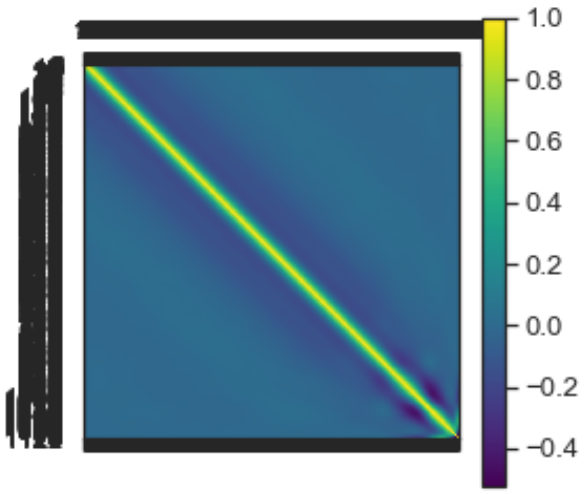


FIGURE 14 – Corrélation entre les variables des dizaines