

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

RAPPORT DE STAGE

Année universitaire 2021/2022

Nom et Prénom de l'étudiant :

BOUSTIQUE SOUFIANE

Année d'études :

☐ M1 MIAGE

Organisme d'accueil : Apneal

Titre du rapport : Rapport de Stage M1 MIAGE

Tuteur de stage : Guillaume Cathelin

Dates du stage : 23 / 05 / 2022 au 09 / 09 / 2022

Je tiens à remercier vivement mon maître de stage, Mr **Guillaume Cathelin, CTO de la startup Apneal**, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats.

Je remercie également toute l'équipe d'**Apneal** pour leur accueil, leur esprit d'équipe et leur soutien pour mon intégration.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

SOMMAIRE :

<u>INTRODUCTION</u>	p3
<u>PARTIE I : ORGANISATION ET COMMUNICATION</u>	p4
1. Présentation de la Start-up	p4
1.1 Histoire de la Start-up	p4
1.2 Activité et caractéristiques globales	p5
1.3 Présentation de l'équipe	p6
2. Analyse de certaines dimensions culturelles et communicationnelles	p7
3. Analyse approfondie de la question du changement	p9
<u>PARTIE II : PARTIE INFORMATIQUE</u>	p10
1. Présentation des Missions Réalisées	p10
1.1 Labellisation de signaux	p10
1.2 Implémentations d'algorithmes pour la manipulation de fichiers de base de données du logiciel Noxturnal	p11
1.3 Implémentations d'Algorithmes pour la manipulation de fichiers de type EDF	p11
1.4 Entraînement et exploitation de signaux dans un réseau de Deep Learning	p12
1.5 Diagramme de Gantt des missions	p13
2. Description des Missions	p14
2.1 Labellisation de signaux	p14
2.2 Implémentations d'algorithmes pour la manipulation de fichiers de base de données du logiciel Noxturnal	p16
2.3 Implémentations d'Algorithmes pour la manipulation de fichiers de type EDF	p23
3. Conclusion de la Partie Informatique	p27
<u>CONCLUSION GENERALE</u>	p27
<u>BIBLIOGRAPHIE</u>	p28

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

INTRODUCTION :

Dans le cadre de mon année d'étude au M1 MIAGE à l'Université Paris-Dauphine, j'ai effectué du 23 mai 2022 au 09 septembre 2022, mon stage dans la start-up Apneal qui se situe dans les locaux d'Agoranov à l'adresse suivante : 96bis Bd Raspail, 75006, Paris.

Apneal est une start-up de deeptech dont le but est de développer une application mobile à base d'intelligence artificielle pour aider à diagnostiquer l'apnée du sommeil. J'ai choisi cette start-up, car elle traite de sujets en relation avec l'intelligence artificielle pour lesquelles j'éprouve une appétence et aussi, car c'est le domaine dans lequel je voudrais travailler à la fin de mon master. J'ai occupé la fonction de Data Analyst/Engineer.

Ma mission au sein d'Apneal a été d'une part la labellisation de signaux, le développement de fonctionnalité sur les bases de données, le traitement de fichiers de type Edf et enfin l'entraînement et exploitation de signaux dans un réseau de Deep Learning. J'ai été encadré par Guillaume Cathelin, le CTO de l'entreprise.

Dans une première partie, je vais présenter la start-up ensuite, je vais décrire et analyser l'environnement de travail et communicationnelles au sein de la start-up ainsi qu'une analyse de la question du changement au sein d'Apneal et enfin, je vais présenter en détail les missions informatiques réalisées.



Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

PARTIE I : ORGANISATION ET COMMUNICATION

1. Présentation de la Start-up :

1.1. Histoire de la Start-up :

Apneal est une start-up française dans le domaine de la biotechnologie créée en février 2021 par Guillaume Cathelain et Sévrin Benizri. Guillaume Cathelain, doctorant de l'École Pratique des Hautes Études-Université-PSL, est le directeur général de la technologie de la start-up et Sévrin Benizri est le directeur général de l'entreprise qui était auparavant entrepreneur du numérique, il a notamment développé plusieurs services pour les médecins et l'industrie pharmaceutique. Les fondateurs de la start-up se sont rencontrés lors d'une conférence sur la bio-tech, ils ont ainsi depuis noué des liens et partagé leur volonté commune de créer une start-up pour traiter la pathologie de l'apnée du sommeil. En effet, cette pathologie atteint environ 3 à 4 millions de personnes en France. Entre 70 % et 80 % d'entre eux l'ignorent et ne consultent pas pour traiter cette pathologie. Face à ce constat alarmant en termes de santé publique, ils ont décidé de fonder leur start-up pour faciliter le diagnostic l'apnée du sommeil grâce à l'intelligence artificielle. Ils ont décidé d'implanter leur start-up, au sein des locaux d'un incubateur de start-up : Agoranov.

Sévrin Benizri, CEO



Guillaume Cathelain, CTO



Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

1.2. Activité et caractéristiques globales :

L'apnée du sommeil provoque des épisodes anormalement fréquents d'interruptions (apnées) ou de réductions (hypopnées) de la respiration durant le sommeil. Ces interruptions provoquent des réveils de courte durée qui dégradent la qualité du sommeil. Les conséquences sur la vie quotidienne sont importantes par la fatigue consécutive à ces nuits peu réparatrices. Malheureusement, un diagnostic fiable et rapide reste souvent difficile à établir, tandis que les conséquences sont lourdes sur certaines pathologies comme la dépression. Dans cette perspective de faciliter le processus de diagnostic du patient, la stat-up Apneal a été créée pour la mise en place d'un diagnostic de l'apnée du sommeil à l'aide d'une application mobile. En effet, le but est d'utiliser les capteurs d'un smartphone fixé par une bande adhésive au thorax pour collecter les données pendant le sommeil ce qui permettra de livrer au médecin une série d'indicateurs sur la réalité du syndrome et sa gravité.

Le dispositif d'Apneal repose donc sur trois principaux outils disponibles dans la technologie du smartphone. Le premier capteur est tout simplement le micro, positionné sur le thorax en direction de la bouche, il permet d'enregistrer le son de la respiration et les ronflements. Puis l'accéléromètre vient enregistrer les petits mouvements du thorax. Il mesure les vibrations en trois dimensions. Le smartphone en est équipé pour compter les pas ou savoir si le téléphone est penché. Enfin le dernier type de capteur est le gyroscope qui mesure les rotations.



Avec une population estimée à 1 milliard de malades sur la planète, dont 80 % qui l'ignorent, les possibilités de développement d'Apneal sont considérables. L'apnée du sommeil peut générer des complications sévères sur la qualité de vie avec des implications en termes de fatigues, de maux de tête, de difficultés de concentration, et un impact sur l'espérance de vie quand elle est liée à des comorbidités cardio-vasculaires, à l'hypertension, au diabète voire aux troubles de la santé mentale.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

D'autre part, la start-up compte s'appuyer sur le modèle économique B to B, ce qui lui permettra de démarcher des clients tels que les centres de sommeil ou les prestataires de santé à domicile dans l'idée de pré-diagnostiquer et prioriser les patients. Puis la commercialisation pourra évoluer en B to C via des professionnels de santé qui prescriront l'examen qui sera facturé au patient. Cela lui donnerait alors une vocation distincte des autres objets marqués « santé » et destinés au grand public, comme une montre, une bague, un bracelet ou un matelas connecté. C'est bien sa certification en tant que dispositif médical que la startup compte acquérir, qui peut alors espérer voir sa solution remboursée par la Sécurité sociale.

Apneal ne dégage pas encore de revenus, mais a reçu l'appui de Bpifrance via le fonds French Tech Seed. La commercialisation doit intervenir courant 2023. Une première levée de fonds a déjà permis de sécuriser un million d'euros publics et privés au service du projet.

Comme évoqué précédemment, la start-up se situe dans les locaux d'Agoranov. Agoranov, regroupe une communauté d'entrepreneurs dans un environnement sécurisé et convivial. Elle permet d'intégrer une communauté de plus de 1000 entrepreneurs Anciens d'Agoranov et de profiter de partages d'expérience et d'échanges de bonnes pratiques. Elle permet également l'accès à un programme d'accélération composé de formations et d'événements : workshops et office hours animés par des experts externes, des industriels, des alumni et des événements de networking.

NB : cette partie a été grandement inspirée d'articles du web qui présentent la start-up sous forme d'interview.

1.3. Présentation de l'équipe :

L'équipe d'apneal est constitué de dix membres. L'équipe est composée de quatre Data scientists en CDI, deux développeurs en CDI et de deux stagiaires (en m'incluant) qui occupent la fonction de data Analyst/Engineer et enfin les deux fondateurs de la start-up. Le CTO Guillaume Cathelin travaille en grande partie en collaboration avec les Data scientists et le CEO Sévrin Benizri s'occupe de la partie médicale en collaboration avec des médecins. En effet, la start-up Apneal a besoin d'effectuer des tests de polysomnographie sur des groupes de patients pour constituer une base de données pour faire fonctionner son Intelligence Artificielle. Pour cela, l'équipe travaille en étroite collaboration avec des médecins (Psychiatre, Gériatre et Chirurgien) de l'Hôpital Henri-Mondor pour réaliser une étude comparative de l'Application Apneal et les tests de polysomnographie usuelle. De plus, l'équipe travaille aussi en collaboration avec Emmanuel Bacry (qui est actuellement chercheur senior au CNRS à l'Université Paris-Dauphine, directeur scientifique de l'INDS et directeur des projets Data / Santé à l'Ecole Polytechnique) en tant que consultant.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

2. Analyse de certaines dimensions culturelles et communicationnelles

La start-up Apneal est à distinguer des grandes entreprises qui portent une plus grande attention à l'image véhiculée dans les médias et au sein de leurs collaborateurs. Une start-up comme Apneal qui est très jeune, véhicule bien évidemment des valeurs de partage, d'entraide mais cette image n'est pas à leur stade leur principale préoccupation. En effet, il n'existe pas de document mettant en avant ces aspects.

D'autre part, comme évoqué précédemment la start-up se situe dans l'incubateur Agoranov, qui se décrit comme lieu de regroupement entre différentes start-up et de partage d'expérience et d'échanges de bonnes pratiques. Je vais donc par la suite décrire et analyser l'environnement de travail et communicationnelles au sein de la start-up et de l'incubateur Agoranov.

Au sein d'Apneal, j'ai réalisé plusieurs missions, pour lesquelles j'ai été accompagné par mon tuteur de stage, mais aussi par l'ensemble des membres de l'équipe qui m'ont fortement aidé à l'adaptation à leurs outils et méthodes de travail. Au sein d'Apneal, les ressources informatiques ont une place prépondérante pour la communication et l'échange entre les membres de l'équipe. En effet, le télétravail chez Apneal occupe une place importante, car la moitié des membres de l'équipe est en télétravail, trois jours sur cinq de la semaine, et le reste de l'équipe travaille en présentiel ou si besoin en télétravail. Pour communiquer, nous utilisons souvent des outils comme slack, Discord et google meetup lors de nos échanges. Le télétravail peut constituer dans certains cas un frein à la cohésion de l'équipe et à sa motivation. Mais chez Apneal, plusieurs dispositions ont été prises pour veiller à renforcer les liens entre les collaborateurs. Premièrement, à chaque début de semaine, le lundi à 11 h 30, une réunion en distanciel est organisée pour discuter en première partie du week-end de chaque membre et de raconter des anecdotes ou histoires éventuelles à partager, ce qui permet de détendre l'atmosphère et de captiver l'attention de l'équipe pour la suite de la réunion. Ensuite, en deuxième partie, chaque membre présente le sujet sur lequel il a travaillé durant la semaine ainsi que ses missions pour la semaine à venir. Ce qui permet d'assurer un suivi sur le déroulement des missions et de leur niveau aboutissement. Tous les membres de l'équipe sont connectés constamment sur Discord pour signifier un acte de présence et simuler une sorte de présence physique, ce qui permet aussi par la même occasion d'échanger si besoin entre les différents membres. Il est à noter qu'un calendrier partagé sur le web permet de renseigner le jour de travail en présentiel ou en télétravail des différents membres de l'équipe. De plus, un jour de la semaine (généralement le jeudi) tous les membres de l'équipe se regroupent en présentiel, et tous les membres déjeunent ensemble et discutent pour renforcer les liens de l'équipe. Personnellement, au début, j'étais réticent concernant cette méthode de travail, car je pensais que mon intégration à l'équipe serait plus difficile, mais je me suis rendu compte que cette organisation n'a constitué en aucun un frein à mon intégration. En effet, durant le jour de regroupement, les membres de l'équipe sont encore plus enthousiastes à l'idée de se rencontrer ce qui permet d'échanger sur plus de sujets et de briser la glace rapidement.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

Et enfin, deux jours par mois sont consacrés à des « team-building » pour renforcer la cohésion de l'équipe. J'ai eu l'occasion durant mon stage de participer à plusieurs activités organisées comme l'acrobranche, des Escape Game, à des repas ou encore à jouer à des jeux de sociétés. Ces activités ont permis grandement de souder l'équipe et surtout de découvrir mes collaborateurs sous un angle différent. Ces activités ont grandement renforcé notre esprit d'équipe, par exemple l'activité escape game nécessite le travail et la réflexion en équipe pour résoudre des énigmes. Un autre exemple concerne l'activité de l'acrobranche, certains membres d'équipes sont moins à l'aide avec cette activité, et d'autres sont plus à l'aide, et ces derniers n'ont pas hésité à aider et à motiver ceux en difficulté. L'esprit d'entraide et d'entente a été donc renforcé.

Il est important d'évoquer que bien que l'équipe soit petite, elle reste néanmoins très diversifiée et multiculturelle avec plusieurs nationalités. De plus, la moitié de l'équipe est constituée de femmes. Ces points m'ont beaucoup marqué, car il révèle que la start-up véhicule réellement des valeurs de partage et d'inclusions. Car il ne suffit pas d'affirmer et de vouloir véhiculer une image d'inclusion et de diversité pour l'être. Ces valeurs ne sont pas présentes uniquement dans la start-up Apneal. Mais également chez les autres start-ups du groupe Agoranov où j'ai pu constater une grande diversité culturelle. Ce point est très important, car il permet de constituer un environnement de travail plus convivial et inclusif. Par exemple dans notre équipe, une des data scientists est originaire du Liban, à son retour des vacances du Liban, elle a partagé avec l'équipe quelques gâteaux traditionnels du Liban. Et nous a raconté par la même occasion la recette du gâteau. Cet échange a permis aux autres membres d'en apprendre plus sur sa culture et ses traditions. Ces échanges sont donc très importants pour la cohésion de l'équipe.

Au sein d'Agoranov, plusieurs ateliers collaboratifs sont organisés pour permettre le partage des connaissances, d'idées et des savoirs. J'ai notamment pu participer à ces ateliers où j'ai découvert plusieurs start-ups ainsi que leur projet, cela a été très instructif et intéressant, car j'ai découvert des idées très innovantes et ambitieuses dans le monde de l'intelligence artificielle, qui est un domaine dans lequel je veux travailler après la fin de mon master. De plus, j'ai eu une discussion avec mon tuteur de stage pour lui demander l'apport réel de ses ateliers pour sa start-up. Il m'a expliqué que ce genre d'événement est très important pour lui, car il lui permet de nouer des contacts, et même dans certains cas retrouver des similitudes entre projets de start-up, ce qui permet d'échanger des idées et des solutions à la résolution de certains problèmes. Par exemple, il m'a raconté que grâce à ses événements, il a pu rencontrer Emmanuel Bacry, qui a été très intéressé par son projet et lui a proposé de lui apporter son expertise dans le domaine de l'intelligence artificielle et notamment celui du traitement des signaux. En effet, un jour par mois est organisé une réunion avec Emmanuel Bacry pour lui présenter l'avancement des data scientists dans leur mission, ce qui permet aussi d'évoquer des solutions alternatives aux problèmes et de donner conseils. J'ai assisté à toutes ses réunions et elles sont très intéressantes. Par exemple, pour développer une fonctionnalité sur l'application Apneal, j'ai pu assister au processus de réflexion et de mise en place d'un plan de travail et d'organisation pour réaliser cette fonctionnalité.

Enfin, pour conclure, l'environnement de travail au sein d'une start-up est très important pour sa productivité et son bien-être, c'est un point essentiel pour assurer la

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

durabilité et le succès de la start-up dans sa mission. L'échange et le partage grâce à l'organisation d'événements comme les ateliers collaboratifs constituent également un point primordial pour la réussite et le développement de la start-up.

3. Analyse approfondie de la question du changement

Cette partie ne sera pas détaillée par manque de ressource. En effet, comme expliqué la start-up est très jeune (créée en 2021), et donc la question du changement ne peut être traitée correctement.

Cependant, je peux évoquer en quelques lignes le processus d'innovation au sein de la start-up Apneal. En effet, Apneal réalise une application qui comporte plusieurs fonctionnalités. Pour réaliser ces fonctionnalités, il faut définir le cahier des charges ainsi que la planification de la mise en développement. Chez Apneal, j'ai assisté à une des réunions avec le professeur Bacry, pour parler du développement d'une fonctionnalité. La réunion a débuté par la présentation de l'objectif de la fonctionnalité ainsi que son intégration dans l'application, cette partie a été discutée par les data-scientist et les développeurs qui devaient présenter les attentes de chaque partie pour contribuer à la réalisation de la mission, il y a donc eu des échanges et des explications à l'aide de schémas pour définir correctement le fonctionnement et les contours de la fonctionnalité. Le professeur Bacry avec son expérience essayait d'aiguiller l'équipe sur la mise en production de la fonctionnalité dans l'application, car il a une grande expérience dans la planification et la réalisation de projets. Son expertise technique a aussi été d'une grande aide. À la fin de réunion, chacune des parties définissait un planning de production qu'ils validaient avec le CTO et le CEO.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

PARTIE II : PARTIE INFORMATIQUE

1. Présentation des missions réalisées :

Au cours de mon stage chez Apneal en tant que Data Engineer/Scientist, plusieurs missions m'ont été confiées. Dans cette partie, je vais donc énumérer toutes les missions réalisées en présentant une brève description sur plusieurs aspects puis un diagramme de Gantt sera présenté pour les différentes missions réalisées.

1.1. Labellisation de signaux :

- ➔ **Objectif** : Cette mission consiste à labelliser des signaux d'audios, de débit de respiration et mouvement du thorax de patients sur le logiciel médical Noxturnal. Cette labélisation constituera une base de données qui permettra d'entraîner notre modèle de Deep-Learning pour détecter les événements respiratoires.
- ➔ **Architecture / Outils** : La labélisation a nécessité l'utilisation de plusieurs outils qui sont les suivants :
 - Logiciel Médical Noxturnal.
 - Machine Virtuelle Windows.
 - Amazon Web Services (AWS) et plus précisément leur outil qui permet de stocker des données dans le cloud « Amazon s3 Bucket ».
 - Google Sheets.
- ➔ **Portée** : Cette mission constitue une phase importante et primordiale dans le but d'entraîner notre modèle de Deep-Learning, elle est donc importante pour nos data scientists qui grâce à cette labélisation pourront affiner et calibrer notre algorithme.
- ➔ **Taille** : La labellisation a été effectuée sur plusieurs groupes de patients. Durant mon stage, nous avons travaillé sur deux groupes de patients. Le premier groupe est constitué de 29 patients et le deuxième de 48 patients. Cette mission est réalisée par toute l'équipe. Un google Sheets a été mis en place pour suivre l'état d'avancement de la labellisation et la répartition du travail.
- ➔ **Durée** : La mission pour la labélisation de ces deux groupes de patients a duré trois mois.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

1.2. Implémentations d'algorithmes pour la manipulation de fichiers de base de données du logiciel Noxturnal :

- ➔ *Objectif* : L'objectif de cette mission a été d'implémenter des Algorithmes qui permettent de manipuler des fichiers de base de données du logiciel Noxturnal. Ce fichier est une base de données qui contient plusieurs données (données du patient, historique de modifications du fichier et le plus important les événements labellisés...). Il a donc été très important d'implémenter des fonctions permettant de fusionner et de générer ces fichiers, mais aussi de supprimer certaines données.
- ➔ *Architecture / Outils* : L'implémentation de ses algorithmes a nécessité l'utilisation de plusieurs outils qui sont les suivants :
 - AWS Sage Maker Notebook.
 - Machine Virtuelle Windows.
 - Amazon s3 Bucket.
 - Amazon SageMaker (Notebook Python).
 - DB Browser for SQLite.
 - GitLab et GitHub.
- ➔ *Taille* : Cette mission m'a été confiée et je l'ai réalisée tout seul sous la supervision de mon tuteur de stage.
- ➔ *Portée* : L'implémentation de ces algorithmes est très importante, car ils vont automatiser des tâches qui étaient auparavant effectuées manuellement. Elle a beaucoup aidé à corriger les erreurs potentielles effectuées lors de la labellisation et aussi aidé les data scientists pour la manipulation de ces fichiers.
- ➔ *Durée* : Cette mission a été réalisée en trois semaines en parallèles d'autres tâches. La durée allouée pour l'avancement sur cette mission a été de trois heures par jour. Il est important de noter que plusieurs versions de cet algorithme ont été mises à jour.

1.3. Implémentations d'Algorithmes pour la manipulation de fichiers de type EDF :

- ➔ *Objectif* : Un fichier de type EDF ou European Data Format contient plusieurs types de signaux. Ce fichier regroupe donc ces signaux avec des informations sur ces signaux. Le but de cette mission a été d'une part de rééchantillonner

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

et/ou d'augmenter le volume sonore du signal audio pour le rendre audible sur le logiciel noxturnal, et d'autre part de pouvoir fusionner des fichiers de type Edf et de rajouter des signaux à ces fichiers.

- ➔ **Architecture / Outils** : L'implémentation de ces algorithmes a nécessité l'utilisation de plusieurs outils qui sont les suivants :
 - Machine Virtuelle Windows.
 - Amazon s3 Bucket.
 - Amazon SageMaker (Notebook Python).
 - Edf Browser.
 - GitLab et GitHub.
- ➔ **Portée** : Cette mission a permis d'intégrer plusieurs signaux dans le fichier pour pouvoir ainsi les visualiser dans le logiciel noxturnal, elle aussi permis aux data scientists une meilleure visualisation de certaines courbes de probabilités (signal contenue dans le fichier Edf).
- ➔ **Taille** : Cette mission m'a été confiée et je l'ai réalisée tout seul sous la supervision de mon tuteur de stage.
- ➔ **Durée** : Cette mission a été réalisée en trois semaines en parallèles d'autres tâches. La durée allouée pour l'avancement sur cette mission a été de deux heures par jour.

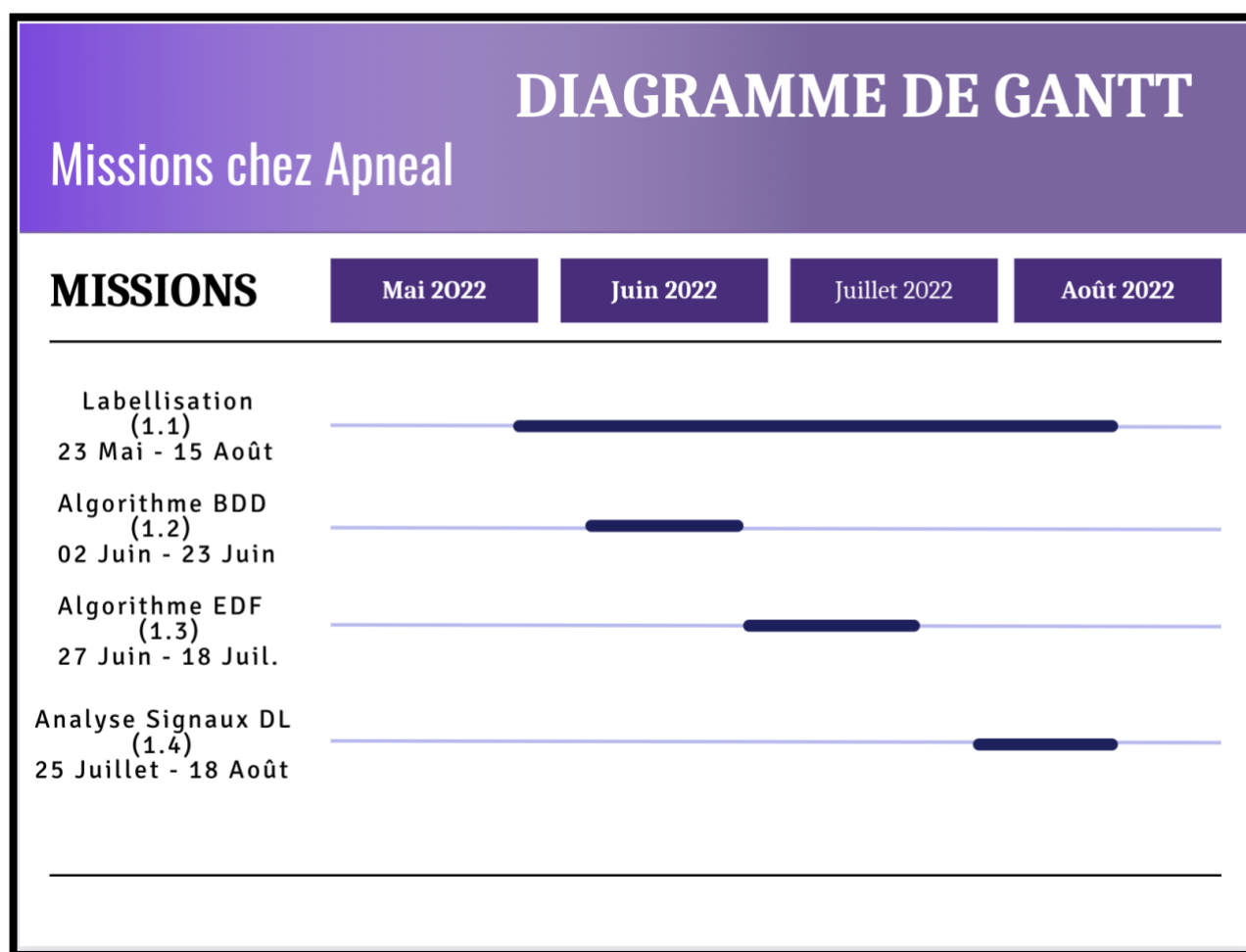
1.4. **Entraînement et exploitation de signaux dans un réseau de Deep Learning :**

- ➔ **Objectif** : L'objectif de cette mission est d'entraîner sur réseau de deep learning nommée « U-sleep » sur des signaux de types EEG (signaux qui mesurent l'activité électrique cérébrale) puis d'analyser et de comparer la segmentation de ces événements fournis par l'Algorithme.
- ➔ **Architecture / Outils** :
 - Amazon SageMaker (Notebook Python).
 - Noxturnal.
 - GitLab et GitHub.
 - U-sleep site web.
- ➔ **Taille** : Cette mission m'a été confiée et je l'ai réalisée tout seul sous la supervision de mon tuteur de stage.
- ➔ **Durée** : Cette mission a été réalisée en trois semaines en parallèle d'autres tâches.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

Cette mission ne sera pas décrite par la suite, car elle comporte beaucoup de termes techniques et elle est difficile à expliquer en termes simples. J'ai donc choisi parmi toutes les missions réalisées de ne pas la développer par la suite.

1.5. Diagramme de Gantt des missions :

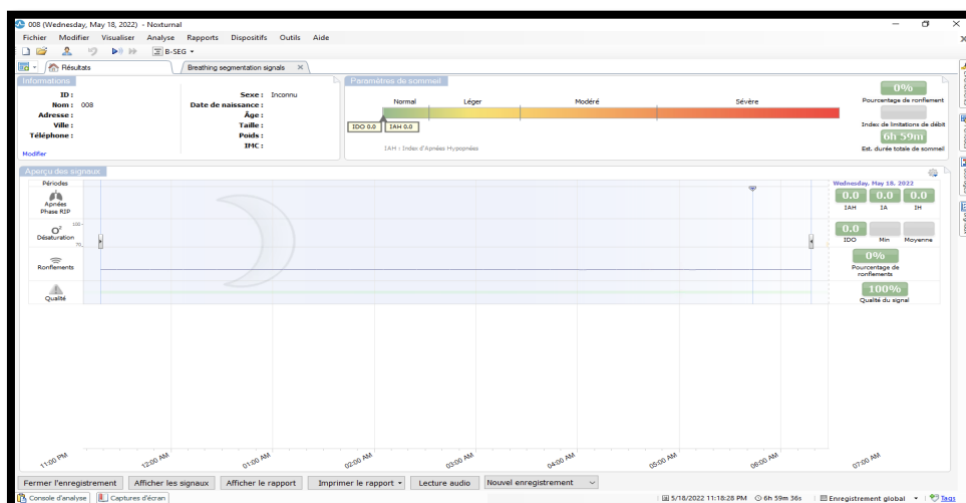


Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

2. Description des Missions

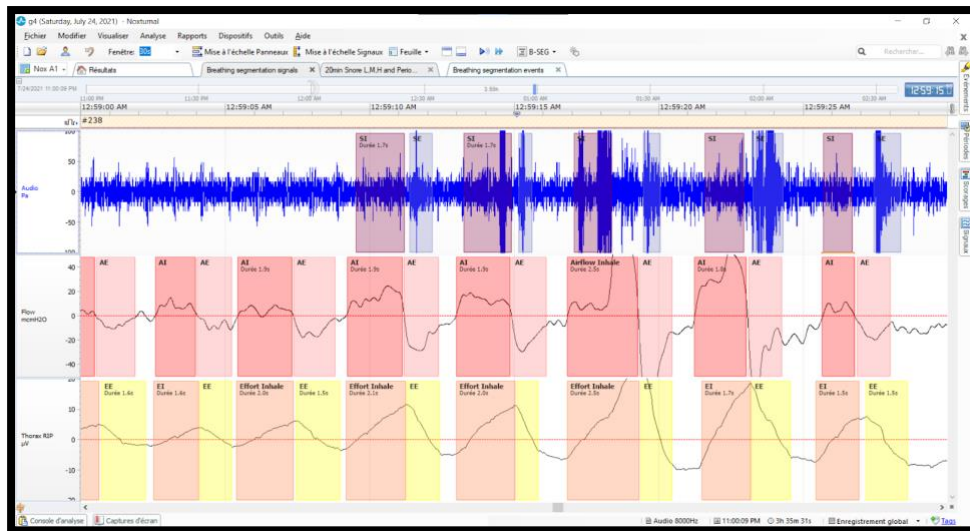
2.1. Labellisation de signaux :

Un des objectifs de la start-up Apneal est de mettre en place un diagnostic de l'apnée du sommeil grâce aux capteurs intégrés dans nos smartphones grâce à une application mobile. Pour cela, l'équipe de data scientists a besoin d'une base de données pour pouvoir entraîner son algorithme de deep learning. Pour constituer cette base de données, la start-up Apneal a réalisé un test de polysomnographie (examen complet qui permet d'évaluer les troubles respiratoires du sommeil) sur deux groupes de patients (un groupe de 28 patients et un deuxième de 41 patients). Pour chaque patient, est fourni un diagnostic de son sommeil sur le logiciel Noxturnal. Ci-dessous une photo du logiciel :



Le logiciel permet de visualiser les données du patient, ainsi que plusieurs signaux captés durant son sommeil. Le logiciel fournit un diagnostic sur l'état de sommeil du patient en permettant d'annoter les différents signaux effectués durant le test de polysomnographie. Parmi ses signaux, se distinguent ; le signal audio qui contient l'enregistrement audio du patient durant toute la phase de son sommeil, le signal du mouvement du thorax ainsi que le signal du flux de respirations nasal. Ces signaux sont importants, car la labellisation évoquée précédemment s'effectue sur ces signaux. La labellisation est constituée d'un ensemble de règles pour chaque signal qui permettent d'annoter ce signal. Ci-dessous une photo des signaux accompagnés de leurs annotations :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu



Comme on peut le voir sur la photo, l'enregistrement a commencé à 11 :00 PM et se finit vers 03 :00 AM. L'enregistrement est découpé en epoch (une epoch est une tranche de l'enregistrement qui dure 30 secondes), pour ainsi pouvoir le délimiter.

Ci-dessous les règles de labellisation décrites en anglais :

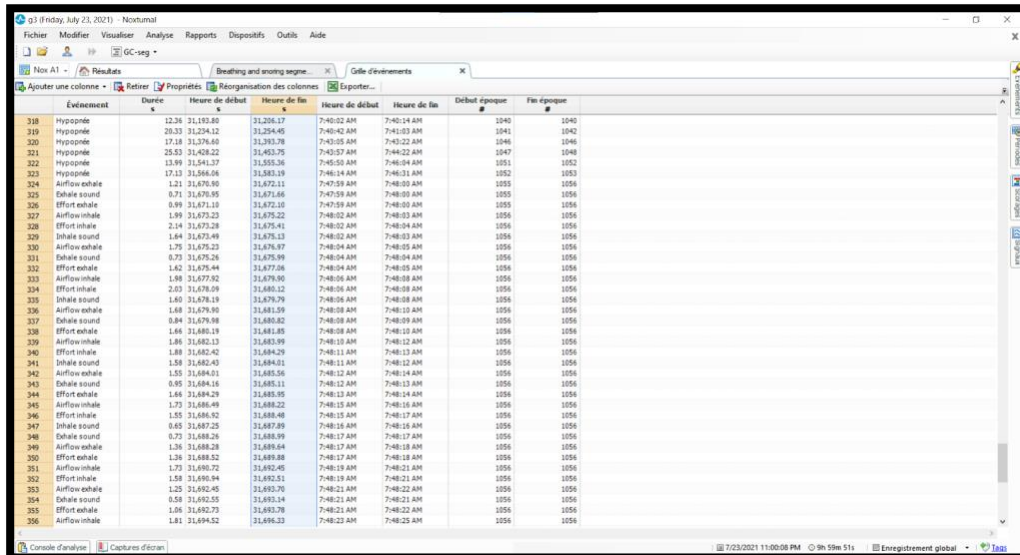
Scoring rules

1. Fully label events that overlap with adjacent epochs
2. 2 previous minutes (baseline) history taken into account
3. Effort events (on thorax only) :
 - a. Effort inhale is a pattern like $f(t) = -A + B \cdot (1 - \exp(-\alpha \cdot (t - t_0)))$ with A, B, alpha and t_0 positive constants, for $t > t_0$. It has a positive and decreasing slope.
 - b. Effort exhale is a pattern like $f(t) = A - B \cdot (1 - \exp(-\alpha \cdot (t - t_0)))$ with A, B, alpha and t_0 positive constants, for $t > t_0$. It has a negative slope whose absolute value decrease.
 - c. They are separated by pauses where thorax signal slope is zero.
4. Airflow events (on airflow only):
 - a. Airflow inhale is stable above baseline
 - b. Airflow exhale is stable below baseline
 - c. Baseline for airflow is not always zero
5. Sound events (on audio only):
 - a. Sound may be heard and seen on the audio periodogram. Be careful with the delay between speakers and the waveform cursor.
 - b. Snore have a characteristic low frequency vibration.
6. No minimum of amplitude is required
7. Inhales and exhales should not overlap
8. The events are scored on each signal : if you don't see or don't hear, you don't score. However, watching other signals and events can help focus on other physiological events, taking into account that sound events \subset airflow events \subset effort events.

Ces règles permettent d'identifier la phase d'inspiration et d'expiration sur les trois signaux.

Les annotations sont ensuite regroupées sous un tableau généré automatiquement par le logiciel comme sur la photo ci-dessous :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu



Eventement	Durée	Heure de début	Heure de fin	Début époque	Fin époque
318 Hypopnée	12.36	21,193.80	21,206.17	7:40:52 AM	7:40:54 AM
319 Hypopnée	26.33	21,254.12	21,280.45	7:40:42 AM	7:41:03 AM
320 Hypopnée	17.18	21,378.60	21,395.78	7:43:05 AM	7:43:22 AM
321 Hypopnée	25.53	21,428.22	21,453.75	7:43:57 AM	7:44:22 AM
322 Hypopnée	13.99	21,541.37	21,555.36	7:45:50 AM	7:46:04 AM
323 Hypopnée	17.13	21,568.06	21,585.19	7:46:14 AM	7:46:31 AM
324 Airflow exhalé	1.21	21,678.90	21,679.11	7:47:59 AM	7:48:00 AM
325 Exhalé sound	0.71	21,678.95	21,679.66	7:47:59 AM	7:48:00 AM
326 Effort exhalé	0.99	21,679.10	21,679.10	7:47:59 AM	7:48:00 AM
327 Airflow inhalé	1.99	21,679.23	21,679.22	7:48:02 AM	7:48:03 AM
328 Effort inhalé	2.14	21,679.28	21,679.41	7:48:02 AM	7:48:04 AM
329 Inhalé sound	1.64	21,679.49	21,679.13	7:48:02 AM	7:48:03 AM
330 Airflow exhalé	1.75	21,679.23	21,679.97	7:48:04 AM	7:48:05 AM
331 Exhalé sound	0.73	21,679.26	21,679.99	7:48:04 AM	7:48:04 AM
332 Effort exhalé	1.62	21,679.44	21,677.08	7:48:04 AM	7:48:05 AM
333 Airflow inhalé	1.88	21,679.02	21,679.90	7:48:06 AM	7:48:08 AM
334 Effort inhalé	2.03	21,679.09	21,680.12	7:48:06 AM	7:48:08 AM
335 Inhalé sound	1.60	21,679.19	21,679.79	7:48:06 AM	7:48:08 AM
336 Airflow exhalé	1.68	21,679.90	21,681.59	7:48:08 AM	7:48:10 AM
337 Exhalé sound	0.84	21,679.98	21,680.82	7:48:08 AM	7:48:09 AM
338 Effort exhalé	1.68	21,680.19	21,681.85	7:48:08 AM	7:48:10 AM
339 Airflow inhalé	1.68	21,682.13	21,683.99	7:48:10 AM	7:48:12 AM
340 Effort inhalé	1.88	21,682.42	21,684.29	7:48:11 AM	7:48:13 AM
341 Inhalé sound	1.58	21,682.40	21,684.01	7:48:11 AM	7:48:12 AM
342 Airflow exhalé	1.53	21,684.01	21,685.56	7:48:12 AM	7:48:14 AM
343 Exhalé sound	0.85	21,684.16	21,685.11	7:48:12 AM	7:48:13 AM
344 Effort exhalé	1.68	21,684.29	21,685.05	7:48:13 AM	7:48:14 AM
345 Airflow inhalé	1.73	21,686.48	21,688.22	7:48:15 AM	7:48:16 AM
346 Effort inhalé	1.85	21,688.52	21,689.48	7:48:15 AM	7:48:17 AM
347 Inhalé sound	0.85	21,687.25	21,687.99	7:48:16 AM	7:48:16 AM
348 Exhalé sound	0.73	21,688.26	21,688.99	7:48:17 AM	7:48:17 AM
349 Airflow exhalé	1.38	21,688.28	21,689.64	7:48:17 AM	7:48:18 AM
350 Effort exhalé	1.28	21,688.52	21,689.88	7:48:17 AM	7:48:18 AM
351 Inhalé sound	1.73	21,690.72	21,692.45	7:48:19 AM	7:48:21 AM
352 Effort inhalé	1.58	21,690.94	21,692.51	7:48:19 AM	7:48:21 AM
353 Airflow exhalé	1.25	21,692.45	21,693.76	7:48:21 AM	7:48:22 AM
354 Exhalé sound	0.58	21,692.55	21,693.14	7:48:21 AM	7:48:21 AM
355 Effort exhalé	1.06	21,692.73	21,693.79	7:48:21 AM	7:48:22 AM
356 Airflow inhalé	1.81	21,694.52	21,696.33	7:48:23 AM	7:48:25 AM

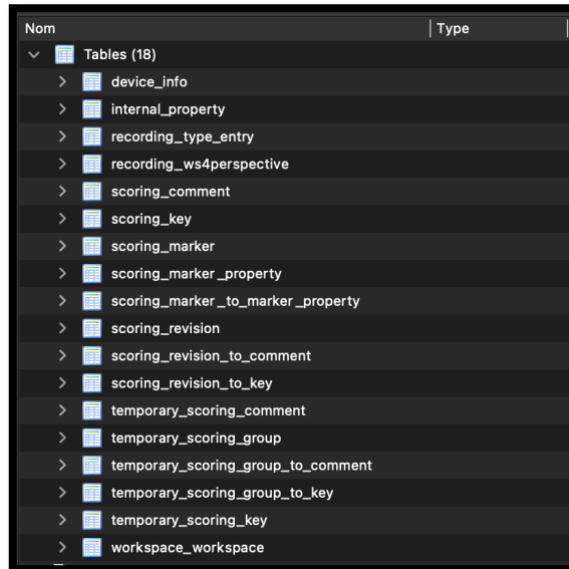
A la fin de la labellisation, deux fichiers sont créés, le premier est un fichier de type SQLite (Base de données) qui est mis à jour à la suite de la labellisation et un fichier excel qui contient le tableau de l'image ci-dessus. Ces deux fichiers sont ensuite téléversés dans le Amazon s3 Bucket (le cloud).

Enfin, le tableau ci-dessus est récupéré par le data-scientist, ce qui lui permet ensuite d'entraîner son algorithme. Pour résumer la labellisation constitue une étape importante dans le processus de la mise en place d'un outil pour la prédiction de l'apnée du sommeil. Cette mission m'a permis de me familiariser avec ce logiciel Noxturnal et d'en comprendre le fonctionnement, ce qui m'a permis en partie de mieux comprendre les objectifs et les attentes de certaines missions que je vais présenter par la suite. Enfin, cette mission s'insère dans le cadre du développement de l'application.

2.2. Implémentations d'algorithmes pour la manipulation de fichiers de base de données du logiciel Noxturnal :

Comme présenté dans la partie précédente, à la fin de la labellisation, un fichier de base de données de type SQLite est mis à jour. Il arrive dans certains cas que ce fichier soit mal enregistré et que la prochaine labellisation sur le même patient s'effectue sur la mauvaise version du fichier. Et par conséquent, il fallait réeffectuer la labellisation depuis la bonne version du fichier. Cette méthode, résout le problème, mais elle est très longue et fastidieuse. Il a été donc été nécessaire d'implémenter des algorithmes qui permettent de fusionner des fichiers de différentes versions pour contenir les scorages des deux versions pour ainsi avoir une solution rapide et efficace. J'ai donc travaillé sur cette mission. Tout d'abord voici ci-dessous une image qui présente la base de données :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu



Nom	Type
Tables (18)	
> device_info	
> internal_property	
> recording_type_entry	
> recording_ws4perspective	
> scoring_comment	
> scoring_key	
> scoring_marker	
> scoring_marker_property	
> scoring_marker_to_marker_property	
> scoring_revision	
> scoring_revision_to_comment	
> scoring_revision_to_key	
> temporary_scoring_comment	
> temporary_scoring_group	
> temporary_scoring_group_to_comment	
> temporary_scoring_group_to_key	
> temporary_scoring_key	
> workspace_workspace	

Comme nous pouvons, le voir la base de données contient plusieurs tables. Ici, je présente une description des tables les plus importantes sur lesquelles vont s'articuler la résolution du problème :

- **Scoring_key** : cette table contient tous les différents enregistrements effectués sur le fichier noxturnal.
- **Scoring_marker** : cette table est la plus importante, car elle contient tout l'historique des annotations. Comme on peut le voir sur la photo ci-dessous de la table, elle contient deux colonnes très importantes qui sont *starts_at* et *ends_at*. En effet, ces deux colonnes, renseignent sur la date et l'heure de début et la fin de l'annotation effectué sur l'enregistrement, cependant cette date et heure a été codé sous un format spécifique. En effet, il ne s'agit pas du format timestamp usuelle (format de mesure de date qui représente le nombre de secondes écoulées depuis le 1er janvier 1970), ce format est propre au logiciel noxturnal. Il a donc été nécessaire de pouvoir en-décoder et décoder cette date.
- **Scoring_revision** : cette table contient l'historique des feuilles sur lesquelles s'effectue l'enregistrement. En effet, une feuille d'enregistrement contient les annotations effectuées par un labelliseur précis. Par exemple, sur la photo ci-dessous dans la colonne « *name* », on a un enregistrement nommé RH qui correspond à une labellisation effectuée par un docteur pour annoter certains types de signaux.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

Table : **scoring_marker**

	id	starts_at	ends_at	notes	type	location	is_deleted	key_id
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
17822	17...	637639911099420000	637639911400270000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24
17823	17...	637639911527980000	637639911665610000	NULL	apnea-central	Resp.Flow-Cannula.Nasal	0	24
17824	17...	637639911726460000	637639911895480000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24
17825	17...	637639912762990000	637639912920090000	NULL	apnea-obstructive	Resp.Flow-Cannula.Nasal	0	24
17826	17...	637639913054910000	637639913382800000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24
17827	17...	637639913460550000	637639913737730000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24
17828	17...	637639918304480000	637639918520820000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24
17829	17...	637639924733780000	637639924855460000	NULL	respi-eosd_type1	Resp.Flow-Cannula.Nasal	0	24

Table : **scoring_key**

	id	date_created	name	owner	type
	Fi...	Filtre	Filtre	Filtre	Filtre
1	1	637640108531631689	Position	Respiratoire Flux Nasal	Automat
2	2	637640108531928744	Activité	Respiratoire Flux Nasal	Automat
3	3	637640108531968747	Apnée/Hypopnée	Respiratoire Flux Nasal	Automat
4	4	637640108587263198	Respiration paradoxale	Respiratoire Flux Nasal	Automat
5	5	637640108587312926	Ronflements	Respiratoire Flux Nasal	Automat

Table : **scoring_revision**

	id	name	is_deleted	tags	version	owner	date_created
	Fi...	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	RH	0		0		637879696372589637
2	2	B-SEG	0		0		637895265284921157
3	3	B-SEG	0		1		637908217064407640
4	4	B-SEG	0		2		637908246244187820
5	5	B-SEG	0		3		637908252700533057
6	6	BSEG-2	0		0		637919481719483513

Maintenant, je vais présenter la partie algorithmique qui m'a permis de résoudre ce problème. Tout d'abord, le langage utilisé est python, c'est le langage de programmation qui contient le plus de bibliothèque (Pandas et Numpy) et qui le plus optimisé pour traiter et manipuler les bases de données. L'interface utilisée est un notebook python sur la plateforme AWS sage-maker. Rappelons que le but de notre fonction est de pouvoir fusionner deux versions de fichiers de base de données de type SQLite. Voici tout d'abord ci-dessous la fonction principale :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

```
def MergeDataFiles1(name_file,name_file_to_be_modified,name_scorage,name_scorage_to_be_modified):
    """
    Usefull to merge Annotations , it take for example the scorage in name_scorage of file name_file and add it all in the scorage name_scorage_to_be_modified
    of name_file_to_be_modified , usefull for BSEG - BSEG2

    Des doublons peuvent exister après merge

    ATTENTION : toujours bien enregistrer les scorages des data avant de merge !
    """
    conn = sq.connect(name_file_to_be_modified)
    conn1 = sq.connect(name_file)

    request = " AND (location='Audio-Respiratory' OR location='Resp.Movement-Inductive.Thorax' OR location='Resp.Flow-Cannula.Nasal') "
    request = request + " AND "
    request = request + " ( type='cycles respiratoires-airflowinhale'"
    request = request + " OR type='cycles respiratoires-airflowinhale'"
    request = request + " OR type='cycles respiratoires-airflowexhale'"
    request = request + " OR type='cycles respiratoires-soundinhale'"
    request = request + " OR type='cycles respiratoires-soundexhale'"
    request = request + " OR type='cycles respiratoires-effortinhale'"
    request = request + " OR type='cycles respiratoires-effortexhale'"
    request = request + " OR type='snorebreath' ) "
    request1 = " (SELECT max(id) FROM scoring_revision WHERE name='" + name_scorage + "') )"
    request1 = " SELECT key_id FROM scoring_revision_to_key WHERE revision_id = " + request1
    request1 = " SELECT * FROM scoring_marker WHERE key_id IN ( " + request1
    request1 = request1 + request

    df1 = pd.read_sql_query(request1,conn1)
    df_r=real_annot_id(df1)

    key_id=addNewIdToScoringKey(conn)
    revisionId=addNewScoringRevision(conn,name_scorage_to_be_modified)
    addNewScoringRevisionToKey(revisionId,key_id,conn,name_scorage_to_be_modified)
    addNewDataToScoringMarker(df_r,key_id,conn)

    conn.commit()
    conn.close()
    conn1.close()
    conn1.close()

    ## ON SAUVEGARDE
    conn = sq.connect(name_file_to_be_modified)
    cur = conn.cursor()
    request1=" DELETE FROM temporary_scoring_group WHERE id IN (SELECT id FROM temporary_scoring_group)"
    cur.execute(request1)
    conn.commit()
    cur.close()
```

Cette fonction prend en entrée les deux fichiers SQLite et les noms des feuilles d'enregistrement qu'on souhaite fusionner et en sortie elle modifie le fichier (celui dans le deuxième argument de la fonction) pour y rajouter toutes les annotations de la feuille d'enregistrement du premier fichier (troisième argument) dans la feuille d'enregistrement souhaité du second fichier (quatrième argument). Il est important de noter que la fonction principale a connu plusieurs versions et a été optimisée en fonction des besoins, plusieurs fois, j'ai présenté ici la version finale, mais d'autres versions ont été implémentées en fonctions de besoins différents.

La fonction principale est composée de sous-fonctions. Voici une description des plus importantes :

- **addNewIdToScoringKey :**

```
def addNewIdToScoringKey(conn):
    """
    Add new Id to the table ScoringKey located in db file of conn

    """
    request = " SELECT * FROM scoring_key "
    df = pd.read_sql_query(request,conn)
    key_id = (df[['id']].iloc[-1][0])+1
    Date = encodeDateNow()
    row = pd.DataFrame({'id': [key_id], 'date_created': [Date], 'name': [''], 'owner': [''], 'type': ['Manual'] })
    addRowsToData(row, 'scoring_key', conn)

    return key_id
```

Le rôle de cette fonction est de rajouter une nouvelle ligne à la table **Scoring_key** du fichier qui sera modifié. En effet, cela permettra principalement de savoir la date à laquelle a été modifié notre fichier en lui rajoutant les nouveaux enregistrements. Ces nouveaux

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

enregistrements seront identifiés grâce à une clé primaire sur cette table, cette clé est très importante par la suite pour les autres tables où elle y sera identifiée comme une clé étrangère.

○ ***addNewScoringRevision :***

```
def addNewScoringRevision(conn,name_scorage):
    requetMaxId = " SELECT max(id) FROM scoring_revision "
    requetMaxVersion = " SELECT max(version) FROM scoring_revision WHERE name = '"+name_scorage+"'"
    df = pd.read_sql_query(requetMaxId,conn)
    maxRevisionId = int(df.loc[0]) + 1
    df = pd.read_sql_query(requetMaxVersion,conn)
    if (list(df.loc[0])[0] == None ) :
        # Scorage n'exite pas déjà , donc on le crée
        MaxVersion = 0
    else :
        MaxVersion = int(df.loc[0]) + 1
    Date = encodeDateNow()
    row = pd.DataFrame({ 'id' : [maxRevisionId], 'name': [name_scorage], 'is_deleted': [0], 'tags': [], 'version': [MaxVersion], 'owner': [], 'date_created': [Date] })
    addRowsToData(row , 'scoring_revision' , conn)

    return maxRevisionId
```

L'objectif de cette fonction est de rajouter dans la table ***Scoring_revision*** le nom de la feuille du scorage, soit en la créant si elle n'existe pas déjà dans la table sinon il suffit de rajouter une ligne dans la table en incrémentant la colonne version et aussi ne pas oublier de préciser la date da rajout. Ainsi le format sera bien respecté.

○ ***addNewDataToScoringMarker :***

```
def addNewDataToScoringMarker(data,key_id,conn):
    """
        Add a DataFrame to the table scoring_marker located in db file of conn
    """
    requetMaxId = " SELECT max(id) FROM scoring_marker "
    df = pd.read_sql_query(requetMaxId,conn)
    maxId = int(df.loc[0]) + 1
    L=[]
    for i in range(data['id'].size):
        L.append(maxId+i)
    Di=pd.DataFrame({ 'id' : L})
    data['id']=Di['id'].values

    data['key_id']=data[['key_id']].apply(lambda x: key_id, axis = 1)
    addRowsToData(data , 'scoring_marker' , conn)
```

Cette fonction est très importante car elle permet de rajouter dans la table ***Scoring_marker***, toutes les données d'annotations qu'on lui souhaite rajouter. Il suffit de veiller à créer un nouvel identifiant (id), pour chaque ligne de donnée et préciser la clé étrangère de la table ***Scoring_key*** évoquée précédemment.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

○ **real_annot_id :**

```
def real_annot_id(dataF):
    """
    dataF is a DataFrame

    take a data Frame of scoring_marker and return the a dataFram with annotations that still exist in the file
    """

    L_id_not_deleted = list(dataF[dataF['is_deleted']==0]['id'])
    out = L_id_not_deleted.copy()
    for id in L_id_not_deleted:
        tmp = dataF[dataF['id']==id]
        if(len(tmp) != 1): # id est unique
            return None
        starts_at = int(tmp['starts_at'])
        ends_at = int(tmp['ends_at'])
        typee = list(tmp['type'])[0]
        location = list(tmp['location'])[0]
        is_deleted = 1
        # We check if this sample has been deleted ( ie if the : id whith is_deleted=1 > id whith is_deleted=0)
        condition = (dataF['starts_at'] == starts_at) & (dataF['ends_at'] == ends_at) & ( dataF['type']==typee ) & ( dataF['location']==location )
        max_id = max(list(dataF[condition]['id']))
        deleted = int ( dataF[dataF['id']==max_id]['is_deleted'] )
        if( deleted == 1):
            out.remove(id)

    col=['starts_at', 'type', 'location', 'is_deleted']
    data_out = dataF[dataF['id'].isin(out)]

    return data_out.drop_duplicates(subset=col, keep='last')
```

Cette fonction est très importante, car elle permet d'optimiser le temps de fusion des deux fichiers (qui peuvent être très lourd dans certains cas) ainsi que la taille (quantité de données contenues dans un fichier) du fichier créé. En effet, la table **Scoring_marker**, contient tout l'historique des annotations ; la colonne **is_deleted** renseigne si la ligne a été supprimée depuis le logiciel, case à 1, sinon 0 si elle a été créée. Le but de cette fonction est de détecter si la ligne existe encore réellement dans le fichier quand on l'ouvre sur le logiciel Noxturnal (Nb : cela ne correspond pas au cas où **is_deleted** est égal à 1).

○ **coder_date & decoder_date :**

```
def coder_date(anne, mois, j, h, mn, s, ms):
    dt = datetime(anne, mois, j, h, mn, s, ms)
    seconds = dt.timestamp()
    S=abs(datetime.fromisocalendar(2, 1, 1).timestamp())+seconds+(365*24*3600)-(24*3600)
    return int(S*1000*10000)

def decoder_date(nb):
    nox_datetime = nb
    seconds_since_JC = nox_datetime//10000000
    milliseconds_since_JC = (nox_datetime - seconds_since_JC*10000000)//10000
    micro_seconds_since_JC = (nox_datetime - seconds_since_JC*10000000 - milliseconds_since_JC*10000)//10
    timedelta_since_JC = timedelta(seconds=seconds_since_JC, milliseconds=milliseconds_since_JC, microseconds=micro_seconds_since_JC)
    JC_datetime = datetime.fromisocalendar(1, 1, 1)
    converted_datetime = JC_datetime + timedelta_since_JC
    return converted_datetime
```

Ces deux fonctions sont très importantes, car elles permettent d'une part de coder une date et heure usuelle selon le format de la base de données (nécessaire pour coder les fonctions décrites ci-dessus). La fonction **coder_date** transforme la date et l'heure en milliseconde depuis le premier janvier de l'an 1, ce nombre est ensuite multiplié par 10 000 pour convenir au format du logiciel.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

La fonction ***decoder_date*** est toute simplement la fonction inverse, elle a été nécessaire car elle m'a permis de comprendre comment fonctionnent la table ***Scoring_marker***.

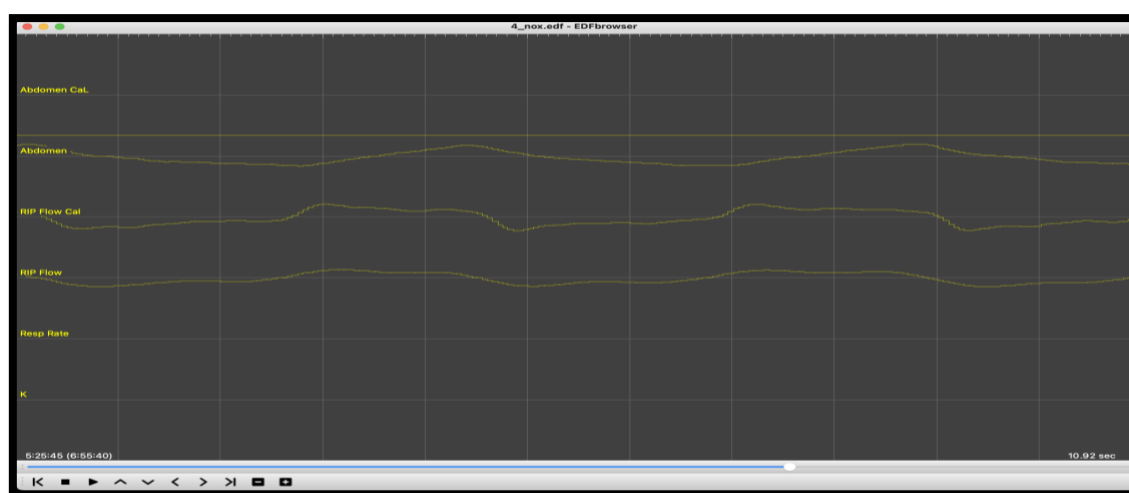
Enfin pour clore la description de cette mission, il est important de préciser que :

- Une partie du travail avant de réaliser ses algorithmes, a été de comprendre les fichiers manipulés et utilisés ainsi que la structure de la base, car il n'existe pas de ressources et de documents sur Internet expliquant le fonctionnement de ses bases de données et le code source du logiciels Noxturnal, un travail autodidacte a été donc nécessaire pour développer ces algorithmes.
- Le versionnage de cet algorithme a été réalisé grâce au GitHub de la start-up dans une branche spécifique.
- Cette mission s'insère dans la phase de développement de l'application mobile.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

2.3. Implémentations d'Algorithmes pour la manipulation de fichiers de type EDF :

Pour rappel, un fichier de type EDF contient plusieurs types de signaux. Ce fichier regroupe donc ces signaux avec des informations sur ces signaux. Le type de fichier Edf que j'ai manipulé contient tous les signaux captés durant l'examen de polysomnographie d'un patient ainsi que des informations personnelles sur ce patient. Ce type de fichier peut être ouvert soit sur le logiciel Edf Browser ou Noxturnal. Ci-dessous une image d'un fichier Edf ouvert sur Edf Browser :



Un des objectifs de cette mission a été de rééchantillonner et d'augmenter le volume sonore du signal audio pour le rendre audible et compatible avec le logiciel Noxturnal. En effet, celui-ci ne peut lire correctement un signal audio que s'il est échantillonné à la bonne fréquence (8 000 Hz) et aussi sous certaines autres conditions. Cette mission est très importante car sans elle la labellisation ne peut pas être effectuée sur le signal audio, ce qui par conséquent restreint les data scientists à effectuer correctement leur mission de prédiction. Contrairement à la mission précédente, il existe la bibliothèque « pyedflib » sur python qui permet de manipuler les fichiers Edf. Il a donc fallu avant de réaliser cette mission de découvrir et de comprendre le fonctionnement de cette bibliothèque en lisant la documentation.

Voici donc la fonction qui a permis résoudre ce problème :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

```
def resample_singal_audio_edf(name_file_source, new_name_file, reduce_volume=20):
    """
    pour les fichiers de psg -> resampling à 8000 hz
    """
    fichier_r = name_file_source
    freal = pyedflib.EdfReader(fichier_r)
    if (list(psl.loc[[10]].label)[0]) != "Mic":
        print("error in acces of singal Mic Number ")
        return False
    signal_audio1 = freal.readSignal(10)
    resampled_signal = scipy.signal.decimate(signal_audio1, 6)
    digital_min = -2 * (16 - 1)
    digital_max = 2 * (16 - 1) - 1
    nb = 360978 / reduce_volume
    resampled_signal1 = (resampled_signal - np.mean(resampled_signal))
    resampled_signal2 = resampled_signal1 * (nb / max(abs(resampled_signal1)))
    physical_min = -1.2 * max(np.max(np.abs(resampled_signal2)), 0.000001) # -1.2 pour pas atteindre le max ou min
    physical_max = physical_min + digital_max / digital_min
    signals = resampled_signal2
    freal.close()

    T = pyedflib.highlevel.read_edf(fichier_r)
    headers = {'label': 'Audio',
               'dimension': 'uV',
               'sample_rate': 8000.0,
               'sample_frequency': 8000.0,
               'physical_max': physical_max,
               'physical_min': physical_min,
               'digital_max': digital_max,
               'digital_min': digital_min,
               'prefilter': 'T11[10]{"prefilter"}',
               'transducer': 'T11[10]{"transducer"}'}

    # freal.close()
    T = pyedflib.highlevel.read_edf(fichier_r)
    T[0][10] = signals
    T[1][10] = headers

    for i in range(len(T[1])):
        if (T[1][i]['physical_min'] > min(T[0][i])):
            T[1][i]['physical_min'] = min(T[0][i]) - 1
        if (T[1][i]['physical_max'] < max(T[0][i])):
            T[1][i]['physical_max'] = max(T[0][i]) + 1

    # On corrige physical_min et physical_max
    digital_min = -2 * (16 - 1)
    digital_max = 2 * (16 - 1) - 1
    for i in range(1, len(T[0])):
        signal = T[0][i]
        physical_min = -1.2 * max(np.max(np.abs(signal)), 0.000001)
        physical_max = physical_min + digital_max / digital_min
        dict = T[1][i]
        dict['physical_max'] = round(physical_max, 1)
        dict['physical_min'] = round(physical_min, 1)
        dict['digital_max'] = digital_max
        dict['digital_min'] = digital_min

    pyedflib.highlevel.write_edf(new_name_file, T[0], T[1], header=T[2], file_type=-1)
    return T[2]
```

Cette fonction prend en entrée le fichier Edf qui contient les signaux ainsi qu'un nombre pour pouvoir réduire le volume sonore du signal audio et en sortie elle crée un nouveau fichier Edf avec le signal audio modifié. Une partie importante qui m'a permis de coder cette fonction provient de la bibliothèque « pyedflib » où se trouve la fonction « EdfReader » qui permet de parcourir le fichier Edf sous forme de tableau à trois entrées. La première contient un sous tableau avec tous les signaux en format de chiffres et nombres. La deuxième entrée contient un sous tableau pour chaque signal avec des informations sur ce signal comme le nom, la dimension, la fréquence, le max et le min, le nom du capteur etc. Et enfin la troisième entrée du tableau contient des informations personnelles sur le patient. Voici une photo qui montre ce tableau :

```
[array([9.81185626e-04, 9.81185626e-04, 9.81185626e-04, ...,
        7.62601371e+01, 5.41033735e+01, 4.34089811e+01]),
 array([9.81185626e-04, 9.81185626e-04, 9.81185626e-04, ...,
        7.94448446e+00, -1.63328978e+01, -2.00123190e+01]),
 array([9.81185626e-04, 9.81185626e-04, 9.81185626e-04, ...,
        3.10296202e+01, 1.25293530e+01, 8.86139416e+00])],
 [{'label': 'EOG LOC-A2',
   'dimension': 'uV',
   'sample_rate': 200.0,
   'sample_frequency': 200.0,
   'physical_max': 375.5885,
   'physical_min': -375.598,
   'digital_max': 32767,
   'digital_min': -32768,
   'prefilter': '',
   'transducer': ''},
 {'label': 'EOG ROC-A2',
   'dimension': 'uV',
   'sample_rate': 200.0,
   'sample_frequency': 200.0,
   'physical_max': 375.5885,
   'physical_min': -375.598,
   'digital_max': 32767,
   'digital_min': -32768,
   'prefilter': '',
   'transducer': ''},
 {'label': 'EEG C3-A2',
   'dimension': 'uV',
   'sample_rate': 200.0,
   'sample_frequency': 200.0,
   'physical_max': 375.5885,
   'physical_min': -375.598,
   'digital_max': 32767,
   'digital_min': -32768,
   'prefilter': '',
   'transducer': ''}],
 {'technician': '',
  'recording_additional': '',
  'patientname': 'X',
  'patient_additional': '',
  'patientcode': 'B-001',
  'equipment': '',
  'admincode': '',
  'gender': '',
  'startdate': datetime.datetime(2022, 4, 22, 22, 44, 54),
  'birthdate': '',
  'annotations': []}]
```

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

Une autre partie de la mission a été de pouvoir fusionner deux fichiers Edf, pour contenir tous les signaux souhaités dans un même fichier pour pouvoir ainsi les visualiser dans Noxturnal, ce qui permet par la suite aux labelliseurs et aux data scientists de naviguer et de manipuler plus facilement l'ensemble des signaux.

Voici la fonction :

```
def merge(fichier_nox, fichier_apneal, fichier_sortie, patient_name='X'):
    T_nox = pyedflib.highlevel.read_edf(fichier_nox)
    T_apneal = pyedflib.highlevel.read_edf(fichier_apneal)
    if ( T_apneal[2]['startdate']!=T_nox[2]['startdate'] ) == False ):
        print('Le startdate du fichier_nox est différent du fichier_apneal ')
        return None

    # Avant de merge on va corriger le physical/digital min et max du fichier nox
    for i in range(len(T_nox[1])):
        if( T_nox[1][i]['physical_min'] > min(T_nox[0][i]) ):
            T_nox[1][i]['physical_min'] = T_nox[1][i]['physical_min'] * 1.2
        if( T_nox[1][i]['physical_max'] < max(T_nox[0][i]) ):
            T_nox[1][i]['physical_max'] = T_nox[1][i]['physical_max'] * 1.2

    # Avant de merge on va corriger le physical/digital min et max du fichier apneal
    digital_min = -2 ** (16 - 1)
    digital_max = 2 ** (16 - 1) - 1
    for i in range(len(T_apneal[0])):
        signal = T_apneal[0][i]
        physical_min = - 1.2 * max(np.max(np.abs(signal)), 0.000001)
        physical_max = physical_min * digital_max / digital_min
        dicT = T_apneal[1][i]
        dicT['physical_max']=round(physical_max,1)
        dicT['physical_min']=round(physical_min,1)
        dicT['digital_max']=digital_max
        dicT['digital_min']=digital_min

    # Fin

    T_apneal[1][0]['label']='Audio'
    T_merge = [[],[],[]]
    T_merge[0] = T_nox[0] + T_apneal[0]
    T_merge[1] = T_nox[1] + T_apneal[1]
    T_merge[2] = T_nox[2]
    T_merge[2]['patientname']=patient_name

    pyedflib.highlevel.write_edf(fichier_sortie, signals = T_merge[0], signal_headers = T_merge[1], header=T_merge[2], file_type=-1)
```

Cette fonction a été simple à réaliser notamment grâce à la fonction « EdfReader » présenté plus haut. Elle prend en entrée deux fichiers Edf et produit en sortie un nouveau fichier Edf avec tous les signaux.

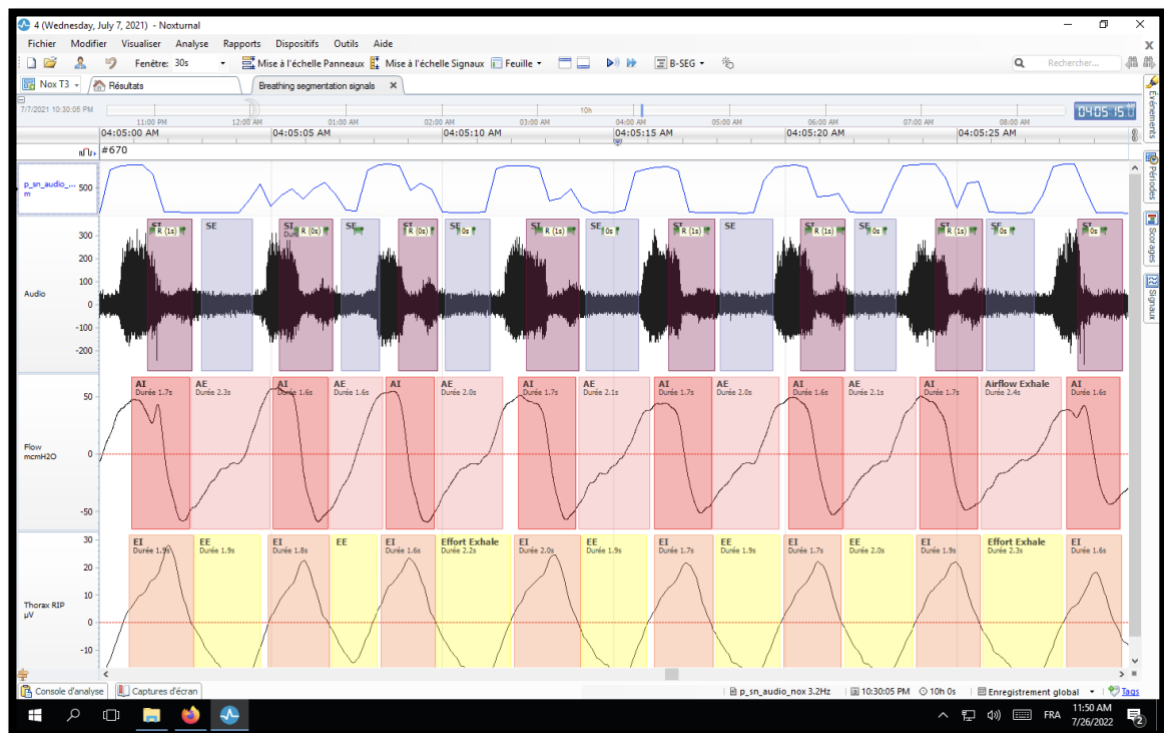
La dernière partie de la mission a été réalisée suite la demande d'un data scientist. En effet, le data scientist produit des tableaux de probabilité qui correspondent à la probabilité qu'un ronflement se situe sur une portion du signal audio. Le but étant de comparer cette probabilité avec les annotations réalisées par les labelliseurs. L'objectif de la mission est étant donné un tableau de probabilité contenu dans un fichier au format h5 de le transformer en fichier Edf et de fusionner celui-ci avec le fichier qui contient tous les signaux audios.

Voici la fonction qui prend en entrée un fichier h5 et le transforme en Edf :

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

```
def df_to_edf(file_name, fichier_sortie, stratDate):
    df = pd.read_hdf(file_name, 'df')
    digital_min = -2 * (16 - 1)
    digital_max = 2 * (16 - 1) - 1
    T = [[], [], []]
    for c in list(df.columns):
        tmp = np.asarray(list(df[c]))
        T[0].append(tmp)
        physical_min = -1.2 * max(np.max(np.abs(list(df[c]))), 0.000001)
        physical_max = physical_min * digital_max / digital_min
        header = {'label': c,
                  'dimension': '',
                  'sample_rate': 3.2,
                  'sample_frequency': 3.2,
                  'physical_max': round(physical_max, 1),
                  'physical_min': round(physical_min, 1),
                  'digital_max': digital_max,
                  'digital_min': digital_min,
                  'prefilter': '',
                  'transducer': ''}
        T[1].append(header)
    T[2] = pyedflib.highlevel.make_header(startdate=stratDate)
    return pyedflib.highlevel.write_edf(fichier_sortie, signals = T[0], signal_headers = T[1], header=T[2], file_type=-1)
```

Et voici le rendu final, ou on peut apercevoir la courbe de probabilité en haut en bleu qu'on compare avec les annotations sur le signal Audio.



Pour finir, cette mission s'insère dans la phase de développement de notre application mobile. Elle a été relativement plus simple à réaliser grâce aux documentations déjà existantes sur Internet. Les fonctions réalisées permettent de réaliser des tâches spécifiques et se sont avérées très utiles pour la visualisation et l'analyse des données surtout pour l'équipe de data scientists.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

3. Conclusion de la Partie Informatique

Pour récapituler, mon stage s'est articulé principalement sur la réalisation de quatre missions, une partie qui concerne la labellisation des données et la familiarisation avec le logiciel Noxturnal, une autre partie qui concerne la manipulation et le développement de fonctionnalité sur des bases de données et des fichiers Edf. Et enfin, la dernière partie concerne l'entraînement de données sur des algorithmes de deep-learning ainsi que l'analyse du résultat obtenu. Concernant la partie de développement des fonctionnalités, les résultats sont très satisfaisants, car ils permettent de résoudre concrètement le problème et ont apporté un réel gain de temps pour toute l'équipe. Il n'y a pas eu une hésitation sur le choix des outils pour le traitement de ses missions, car ce sont des outils fiables et très efficaces pour la réalisation de ses tâches. Personnellement, je suis très satisfait d'avoir pu mener à terme ces missions et surtout de découvrir de nouveaux outils, qui m'ont permis d'élargir mes connaissances en bio-informatique. D'autre part, si la durée du stage était plus grande, j'aurais souhaité réaliser des missions plus en relation avec l'analyse de données. C'est une partie que j'ai touché dans ma dernière mission et elle m'a beaucoup intéressé.

CONCLUSION GENERALE DU RAPPORT

Pour conclure, mon expérience au sein de la start-up Apneal a été très enrichissante. Elle m'a permis de découvrir le monde de la bio-technologie et leur importance dans le domaine médical. Les missions réalisées ont été très intéressantes, car elles m'ont permis de développer mes compétences dans les bases de données et de découvrir tous les outils nécessaires pour faire du deep-learning et surtout tout le processus menant à l'exploitation d'un algorithme d'intelligence artificielle, en commençant par la collecte des données au sein des patients grâce au téléphone puis à leur exportation dans le cloud, leur manipulation et pré-traitement (labellisation, traitement des signaux) pour les rendre exploitables pour l'algorithme. Je suis très satisfait des missions réalisées et que j'ai pu mener à bout, car elles ont réellement apporté un gain de temps pour les labelliseurs ainsi que pour les data scientists. J'ai notamment utilisé beaucoup l'outil Amazon Web Service, qui est un outil très pratique pour stocker, manipuler des données et pour la création de notebook. Avoir une expérience sur cet outil est une vraie valeur ajoutée pour mon CV. Si j'avais l'opportunité de réaliser une alternance au sein de cette start-up, j'aurais souhaité avoir des missions concernant l'optimisation et l'exploitation des algorithmes de deep-learning. Cette expérience a confirmé mon appétence pour les sujets concernant l'intelligence artificielle (IA) et aussi intrigué ma curiosité sur l'application de l'IA dans le domaine médicale. Mon parcours à Dauphine et les enseignements de mon Master m'ont été d'une aide fondamentale pour la compréhension et la réalisation des missions, surtout les cours de machine Learning et de python pour la manipulation des données ainsi que les cours sur les bases de données dispensés en L3. Après cette expérience, mon envie de devenir data scientist s'est confirmé et je souhaiterais continuer mon parcours dans ce domaine.

Place du Maréchal de Lattre de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

BIBLIOGRAPHIE

- <https://www.neftys.fr/actu-innov/apneal-l-application-medicale-qui-detecte-l-apnee-du-sommeil-grace-a-un-smartphone>
- **ARTHUR LE DENN**, <https://www.maddyness.com/?p=1329255>
- https://www.la Tribune.fr/supplement/la-tribune-now/apneal-veut-faciliter-la-detection-de-l-apnee-du-sommeil-891805.html?utm_medium=Social&utm_source=LinkedIn#Echobox=1638870506-1
- <https://www.horizon-ia.com/saison3-apneedusommeil/>