1. Raison du choix des algorithmes

La réalisation du projet a débuté par l’importation de plusieurs modules très important à la réalisation du projet et au tracé des différents graphiques demandés. En effet, panda permet entre autre d’ouvrir le fichier csv contenant les données, numpy facilite l’utilisation des différentes fonctions et matplotlip le tracé des graphiques.

On a choisi d’utiliser panda pour l’ouverture du fichier CSV puisque ce module permet une ouverture facile. Afin de pouvoir utiliser les données fournies on à chercher à transformer chaque colonne du tableau en liste. On fait cela en deux étapes, la première est de créer une liste de listes contenant chacune une des données. La deuxième de séparer la liste de listes en listes distinctes, en implémentant une fonction qui remplit les différentes listes (vides au départ) par les éléments des listes qui se trouvait dans la liste des listes.

On a remarqué que les valeurs des différentes données ont été réalisé par 6 différents capteurs tous en même temps. On a donc décidé de déparer chaque liste de données en 6 différentes listes chacune relative à chaque capteur, on s’est donc retrouvé avec 6 listes pour chaque donnée. On choisit de suivre cette méthode parce qu’elle nous paraissait logique et facile à comprendre.

Le format des dates donné dans le fichier CSV (‘2020-09-25 07:32:09 +0200’ par exemple) ne permettait pas d’utiliser les dates dans les différentes fonctions puisqu’elles n’étaient pas sous la forme String. Afin de transformer les dates sous la forme string et les utiliser dans les différentes fonctions, on a utilisé ‘datetime.strptime’. Notre choix a été motivé par l’utilisation facile et efficace de cette fonction.

A ce stade toutes les listes créées étaient utilisable pour le tracé des fonctions pour lesquelles il faut utiliser obligatoirement les listes de temps en format string. Pour tracer les différentes fonctions on a utilisé matplotlib qui permet d’ajouter les différentes composantes d’un graphique (titre, légende, noms des axes…). Pour le calcul des différentes données statistiques (min, max, écart-type, moyenne, variance, médiane et indice de corrélation…) on a utilisé des fonctions déjà présentes dans le module numpy ou math (mean, var, max, min …), car celles-là n’ont pas été prescrit par les professeurs et s’inscrivent dans une logique professionnelle qui demande beaucoup d’efficacité. En ce qui concerne le calcul de la moyenne acoustique il nous paraissait plus judicieux d’utiliser la moyenne logarithmique

L’indice humidex passe par un calcul de la température de rosée. On a choisi de créer 6 listes humidex et 6 listes de température de raison chacune relative à chaque identifiant. On a procédé de cette manière parque l’utilisation de listes permet le tracé de la courbe de l’indice humidex en fonction du temps.

Pour créer un graphique mettant en évidence les anomalies dans les données, il fallait tout d’abord repérer ces dites anomalies. Dans ce but nous avons adopté une méthode consistant à calculer la moyenne entre les deux termes entourant directement la donnée à vérifier (i-1 et i+1) et à vérifier si la donnée i appartient à un intervalle autour de cette moyenne, la longueur de cet intervalle pouvant être modulée par l’utilisateur. Ainsi si la valeur n’appartient pas au dit intervalle, elle est repérée comme anormale et est ajoutée à une liste. Pour faire apparaitre ces valeurs sur un graphique nous avons utilisés la fonction scatter de matplolib. Cette méthode nous a paru adaptée au problème car les données sont constituées de valeurs physique variant lentement en fonction du temps, elle a cependant ses limites et un examen critique est nécessaire car des données tels que des maximum ou des minimum locaux peuvent être interprétés à tort comme des anomalies

Enfin, pour déterminer les horaires, nous avons repéré sur le tableau des données des seuils brusques sur certaines grandeurs (volume sonore, luminosité et taux de CO2) entre leur niveau de jour et leur niveau de nuit. Pour tirer parti de cela nous avons alors de manière arbitraire choisit des valeurs de seuil (respectivement 28, 5 et 425) pour repérer la transition jour/nuit et ainsi noter l’heure où les données dépassent ces seuils. Il ne restait alors plus qu’à faire la médiane des valeurs obtenues en regardant le volume sonore, la luminosité et le CO2 pour avoir une approximation de l’horaire d’ouverture et de fermeture du bureau. Le point faible de cette méthode étant qu’elle ne peut s’adapter à d’autres bureaux.

1. Les essais infructueux

Lors de la réalisation des différentes fonctions on a rencontré plusieurs difficultés. En effet, on a eu du mal à transformer les colonnes du tableau CSV en listes pour pouvoir les utiliser dans les différentes fonctions. De plus, l’une des plus grandes difficultés rencontrées étaient de traiter les données tout en prenant en compte la différence de capteurs. Au début on a essayé d’implémenter une fonction qui divise les colonnes du tableau en listes et celles-ci en 6 listes relatives aux différents identifiants, mais celle-ci n’a pas marché.

Il était difficile de déterminer quel niveau de précision nous voulions pour repérer les données anormales, un niveau trop bas ne repérerait aucune anomalie alors qu’à l’inverse un niveau trop haut noterait comme anomalies des données qui ne les sont pas, nous avons finalement choisi de laisser le niveau de précision modulable par l’utilisateur du programme, plusieurs niveaux de précisions permettant de mieuxrendre compte de la réalité

L’élément que nous n’avons également par réussi à résoudreest de permettre la possibilité de spécifier un intervalle de temps dans la ligne de commande. On a implémenté une fonction qui permet de parcourir les listes des temps pour chaque identifiant et ne garder que la partie ‘année : mois : jours’, cependant ce nouveau format ne permet pas de tracer les différentes fonctions demandées. Cette partie étant optionnelle on a choisi de ne pas s’y attarder.

Nous avions dans un premier temps voulu déterminer les horaires d’ouvertures et de fermeture à partir de la dérivée des données fonctions du temps mais cela n’a pas porté ses fruits car il était impossible de trouver un seuil maximal qui correspondrait à toutes les données.

1. Méthode de travail

En ce qui concerne le travail de groupe, on réservait le créneau hebdomadaire d’informatique et de programmation pour travailler ensemble. En effet, via un appel vocal on échangeait sur le sujet tout en modifiant le code ensemble sur Github. Le reste de la semaine on travaillait en autonomie tout en partageant toutes les modifications sur Github. Le travail sur cette plateforme a été très compliqué au départ, à cause de sa nouveauté mais par la suite elle nous a permis d’avoir un échange optimal.