COULANGE Renan

HELIES Albane

Rapport Projet Programmation et Algorithmique

Dans ce rapport se trouve le script du projet de programmation réalisé sur Spyder (Anaconda) en langage python.

* Tout d’abord, nous avons décidé de séparer le fichier CSV initial en 6 fichiers CSV contenant respectivement les valeurs associées aux 6 capteurs différents. Nous avons ouvert ces 6 documents dans un fichier Spyder. Cependant, chaque information extraite des documents était une chaîne de caractères, rendant l’exploitation des données impossible. Il a fallu convertir toutes les données numériques (chaînes de caractères str) en flottant (float). Pour cela, nous avons tout d’abord dû supprimer les guillemets du document afin de n’avoir que des valeurs numériques séparées de virgules.
* Nous avons par la suite écrit les programmes prenant une liste en argument permettant d’obtenir les valeurs statistiques (**minimum**, **maximum**, **médiane**, **écart-type** et **moyenne**) de ladite liste.
* Nous avions en tête de tracer les courbes représentatives des valeurs d’une donnée pour chaque capteur en fonction de la date à laquelle avaient été fait les relevés. Il fallait donc agir sur les dates afin de les mettre en abscisse de ces courbes. Pour cela, nous avons fixé la première date/premier horaire comme origine des temps et avons converti toutes les dates/horaires en secondes grâce au programme **dateconv** : nous avons ainsi posé temps\_c1, temps\_c2, temps\_c3, temps\_c4, temps\_c5 et temps\_c6, 6 listes dont le premier terme était zéro et recensant tous les horaires (en secondes) auxquels avaient été faits les relevés.
* Ensuite, pour comparer les capteurs entre eux, nous avons rédigé le programme **difference** qui nous a permis de calculer la distance entre les valeurs d’un capteur et d’un autre par rapport à une même donnée. Nous avons ainsi posé 15 listes par donnée (bruit, température, humidité, lumière et taux de CO2), comparant ainsi tous les capteurs entre eux.
* Nous avons alors décidé de normaliser ces listes obtenues. Prenons l’exemple du bruit : il fallait déterminer le maximum et le minimum parmi toutes les valeurs de bruit relevées au premier horaire, puis diviser toutes ces valeurs par l’indice de normalisation , et répéter l’opération pour le deuxième horaire, etc… et cela pour chaque donnée.

Pour cela, nous avons regroupé toutes les distances entre les différents capteurs issues du programme **difference** en sous-listes dans une même liste :

L= [ [noise\_c1c2, noise\_c1c3,…, noise\_c5c6], [temp\_c1c2, …, temp\_c5c6], …, [carb\_c1c2, …, carb\_c5c6] ]

Nous avons fait attention à poser n comme étant la taille de la plus petite sous-liste avant de normaliser toutes les listes en utilisant une boucle *for j in range(n)* pour qu’il n’y ait pas d’erreur « index out of range ». Cependant, lorsque nous avons calculé les moyennes de ces listes normalisées, nous avons obtenu des valeurs supérieures à 1 ! Nous avons compris que nous avions oublié, dans le programme **normalise,** de supprimer tous les éléments des sous-listes dont l’indice était supérieur à n afin que toutes les sous-listes soient de taille n.

Nous avons nommé les listes normalisées pour chaque donnée : *donnée\_norm.*

* Nous avons donc par la suite calculé les moyennes de ces listes normalisées (moyennes des différences entre les capteurs normalisées) pour chaque donnée (nommées *InitialeDonnéeN°1erCapteurN°2èmeCapteur\_moy* (par exemple n12\_moy appliqué au bruit entre les capteurs 1 et 2)) et avons regroupé ces moyennes dans des tableaux pour favoriser une meilleure clarté grâce au module *numpy* *(np.array) : InitialeDonnée\_tab*
* Nous avons tracé les courbes représentatives des valeurs relevées par les 6 capteurs pour chacune des 5 données en superposant les courbes représentant les mêmes données (exemple : toutes les courbes relatives au bruit sur un même graphique). Cette méthode nous a permis d’observer grossièrement les similarités et différences entre les différents capteurs.

Nous avons également écrit une autre manière de faire apparaître les courbes en faisant apparaître les 6 courbes (une pour chaque capteur) séparément afin de favoriser la clarté des résultats.

* Enfin, nous avons rédigé une fonction finale nommée **verdict** prenant un flottant en argument (supposément la moyenne calculée tout juste précédemment). Cette fonction devait déterminer si finalement, les capteurs étaient similaires ou non par rapport à une donnée. Pour cela, nous avons analysé les courbes et les avons comparées aux valeurs des moyennes obtenues. Nous avons décidé de poser approximativement un seuil *s=0,22* au-dessus duquel les capteurs seraient considérés comme « non similaires » par le programme. Il a fallu faire attention à convertir ce seuil défini au début du programme en flottant afin de pouvoir le comparer à la moyenne rentrée en argument.
* On obtient finalement le résultat ci-dessous, qui pourrait un peu différer en fonction du seuil choisi :

**S** : similaires

**NS** : non similaires

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bruit | Température | Humidité | Lumière | Taux de CO2 |
| Capteur 1 et 2 | S | NS | NS | S | S |
| Capteur 1 et 3 | S | NS | NS | S | S |
| Capteur 1 et 4 | S | NS | NS | S | S |
| Capteur 1 et 5 | NS | NS | NS | S | S |
| Capteur 1 et 6 | S | NS | NS | S | S |
| Capteur 2 et 3 | S | S | S | S | S |
| Capteur 2 et 4 | S | S | S | S | S |
| Capteur 2 et 5 | S | S | S | S | NS |
| Capteur 2 et 6 | S | S | S | S | S |
| Capteur 3 et 4 | S | S | S | S | S |
| Capteur 3 et 5 | S | S | S | S | S |
| Capteur 3 et 6 | S | S | S | S | S |
| Capteur 4 et 5 | S | S | S | S | NS |
| Capteur 4 et 6 | S | S | S | S | NS |
| Capteur 5 et 6 | S | S | S | S | S |

* Nous pouvons avancer quelques constats en se rappelant que le seuil a été choisi grossièrement.
* Pour le bruit, on peut supposer que le capteur 5 est différent des 5 autres, qui eux, semblent assez similaires entre eux.
* Pour la température et l’humidité, le capteur isolé est clairement le capteur 1.
* Pour la lumière, aucun capteur ne semble différent des autres. On ne peut pas tirer de conclusion pour autant. On peut supposer que les résultats ne sont pas mis en avant à cause d’un manque de précision du seuil fixé.
* Pour le taux de CO2, les résultats tendent à montrer que les capteurs 4 et 5 ne sont pas similaires aux 4 autres.
* Il était également demandé d’écrire un programme capable de donner le coefficient de **corrélation** entre deux capteurs. Nous l’avons créé à partir de deux autres programmes, l’un déterminant l’**écart-type**, l’autre la **covariance**. Le programme prend ainsi en argument toutes les listes de valeurs relatives à une même donnée. Nous avons décidé d’y intégrer la fonction *random.choice* afin qu’elle choisisse au hasard deux capteurs pour calculer leur coefficient de corrélation.