VINCENT GAGNON – GAGV25059800
Baccalauréat en Informatique

Rapport TP2

Travail d'apprentissage machine
présenté à

Julien Maitre
dans le cadre du cours
8INF436 Forage de données

Module d'informatique et de mathématique
Université du Québec à Chicoutimi
Le 12 Mars 2021

Table des matières

Description de l'ensemble de données

L'ensemble de données contient environ 10 ans d'observations quotidiennes sur la météo dans différents endroits en Australie. Il y a 23 attributs et un total de 145 460 entrées. On y retrouve deux classes, soit il pleut le lendemain ou soit-il ne pleut pas le lendemain. Le but est de prédire s'il va pleuvoir le lendemain.

Date

La date de l'observation, du 31 octobre 2007 au 24 juin 2017. On retrouve la même date dans plusieurs entrées car elles regroupent les données de plusieurs heures de la journée et de plusieurs stations météo.

Location

Le nom de l'endroit où se situe la station météo. On retrouve 50 stations dans l'ensemble de données.

MinTemp

La température minimum enregistrée par la station météo dans la journée, en degrés Celsius.

MaxTemp

La température maximum enregistrée par la station météo dans la journée, en degrés Celsius.

Rainfall

La quantité de pluie tombé dans la journée, en millimètres.

Evaporation

Mesure d'évaporation de l'eau dans la journée, en millimètres.

Sunshine

Nombre d'heures d'ensoleillement dans la journée.

WindGustDir

La direction de la rafale de vent la plus forte de la journée, prend les valeurs des points cardinaux.

WindGustSpeed

La vitesse de la rafale de vent la plus forte de la journée, en km/h.

WindDir9am

La direction du vent à 9h, prend les valeurs des points cardinaux.

WindDir3pm

La direction du vent à 15h, prend les valeurs des points cardinaux.

WindSpeed9am

La vitesse du vent à 9h, moyenne pris pendant 10 minutes avant 9h, en km/h.

WindSpeed3pmà

La vitesse du vent à 15h, moyenne pris pendant 10 minutes avant 15h, en km/h.

Humidity9am

Pourcentage d'humidité à 9h.

Humidity3pm

Pourcentage d'humidité à 15h.

Pressure9am

Pression atmosphérique réduit à la moyenne du niveau de la mer à 9h, en hectopascals(hpa).

Pressure3pm

Pression atmosphérique réduit à la moyenne du niveau de la mer à 15h, en hectopascals(hpa).

Cloud9am

Fraction du ciel couvert par les nuages à 9h, mesuré en oktas, qui est une unité de mesure qui enregistre combien de huitièmes du ciel sont obscurés par les nuages. Un 0 signifie un ciel clair et un 8 un ciel complètement obstrué.

Cloud3pm

Fraction du ciel couvert par les nuages à 15h, mesuré en oktas, qui est une unité de mesure qui enregistre combien de huitièmes du ciel sont obscurés par les nuages. Un 0 signifie un ciel clair et un 8 un ciel complètement obstrué.

Temp9am

Température en degrés Celsius à 9h.

Temp3pm

Température en degrés Celsius à 9h.

RainToday

Si les précipitations des 24 dernières heures précédent 9h sont plus élevé que 1mm, cet attribut est égal à 'Yes', sinon 'No'.

RainTomorrow

Est égal à 'Yes' si plus d'un millimètre de pluie est tombé le lendemain, sinon 'No'. Il y a 31 877 Yes et 110 316 No, donc 75,84% de journées où il n'a pas plu.

Méthodologie

La métrique de performance que nous utiliserons pour comparer les algorithmes sera la mesure de l'homogénéité des clusters. La précision d'un cluster sera égale au ratio des instances d'un classe que l'on retrouve le plus fréquemment par rapport au nombre total d'instances dans le cluster, on fera ensuite une moyenne des précisions de tous les clusters pour avoir la précision globale d'un algorithme. Un bon algorithme va donc placer les observations similaires dans un même cluster et nous pourrons conclure que ce genre d'instances mène à de la pluie le lendemain.

Hypothèses

Je crois que la précision moyenne des algorithmes de clustering sera au minimum le ratio de données qui sont des instances de journées où il n'a pas plu, c'est-à-dire, 75,84%. Je pense cela car ce serait logique car un algorithme qui classerait les données au hasard arriverait à cette précision parce que 75% des instances sont des journées où il ne pleut pas, et donc 75% des instances d'un cluster formé aléatoirement seront des journées où il ne pleut pas.

Vérification et pré-traitement des données

L'ensemble de données contenait plusieurs valeurs manquantes dans plusieurs attributs. Certaines d'entre eux était normales et d'autres non, il a donc fallu les remplacer afin de pouvoir appliquer les algorithmes de clustering.

Pour les valeurs manquantes qui représentent des températures, un taux d'humidité ou la pression atmosphérique, on ne fait que mettre la moyenne de toutes les instances comme remplacement. Cela semble approprié et n'impactera pas les algorithmes de clustering.

Pour les valeurs manquantes qui représentent l'évaporation de la pluie, le temps d'ensoleillement, la vitesse du vent, la couverture du ciel et la pluie tombée, on ne fait que les remplacer par 0. Car dans cet ensemble de données les valeurs manquantes de ces attributs représentent une absence de la caractéristique dans la journée. Par exemple, une valeur manquante dans la colonne Rainfall signifie qu'il n'a pas plu dans cette journée, on peut donc la remplacer par 0 sans perdre la signification.

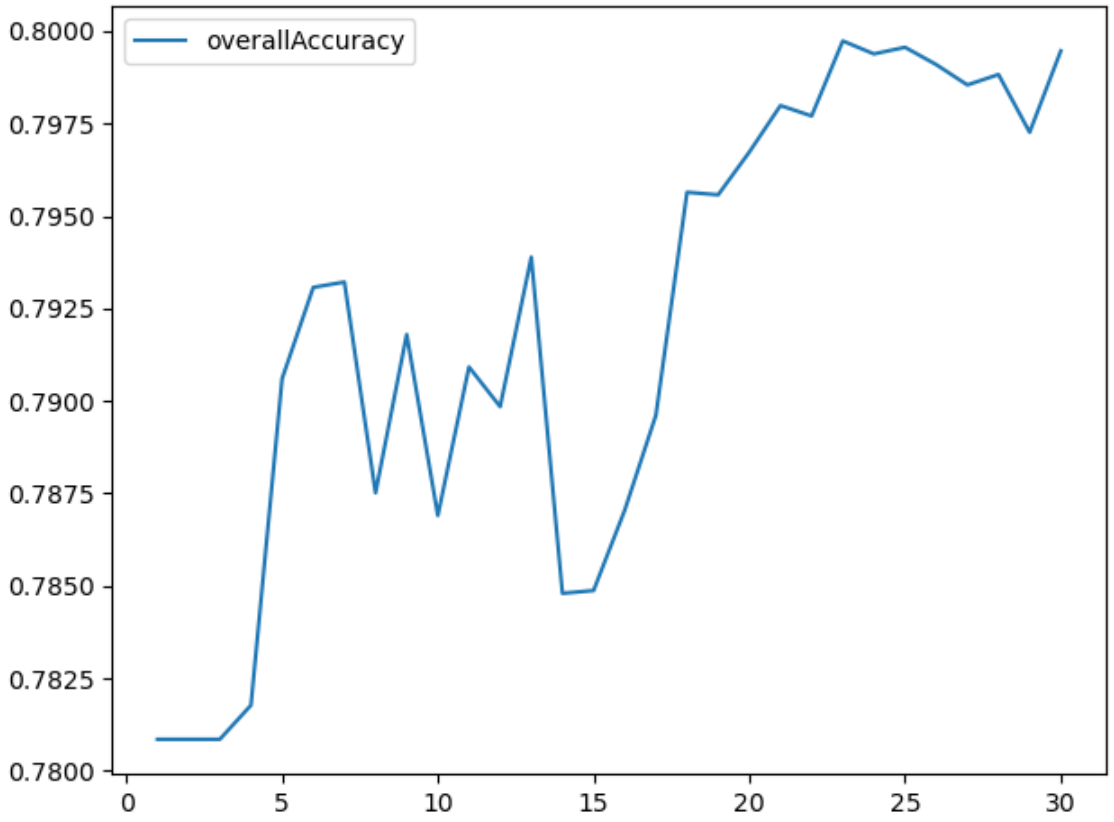
Pour les valeurs booléennes qui indiquent s'il a plu dans la journée ou le lendemain, on ne fait que remplacer la valeur manquante par la valeur la plus fréquente de l'ensemble des instances, qui est non.

Finalement, pour les valeurs de la direction du vent qui manque, on ne fait que les remplacer par une nouvelle catégorie, « No Wind ». Les valeurs manquantes de ces colonnes sont des instances ou il n'y avait pas de vent, et donc pas de direction pour le vent, on créer donc la nouvelle catégorie pour remplacer la valeur manquante.

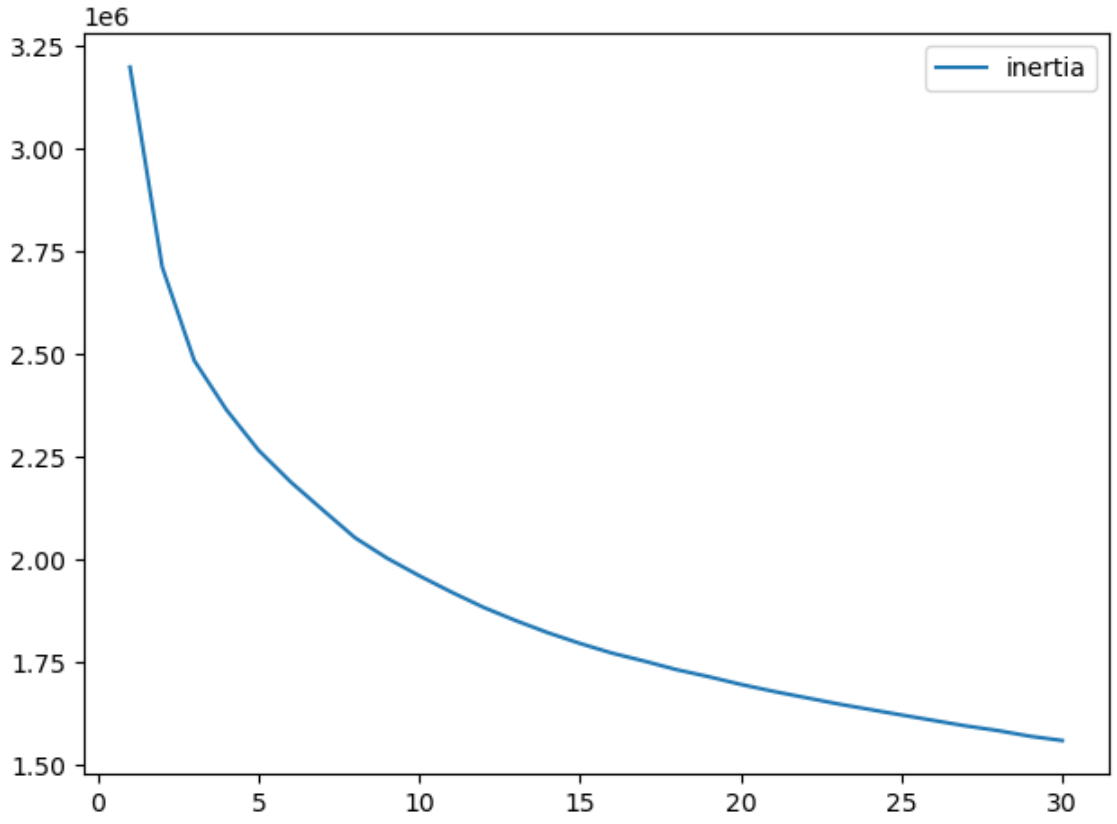
Après avoir remplacé les valeurs manquantes, on transforme les attributs qui sont des catégories en nombre avec la classe LabelEncoder de sklearn afin de pouvoir utiliser nos algorithmes de clustering. Ensuite, on met à l'échelle les caractéristiques pour ne pas biaiser les modèles.

Application de l'algorithme de clustering K-Means

On commence avec l'algorithme de clustering k-means, cet algorithme va classer les instances de tel sorte que l'inertie entre les instances d'un même cluster soit le plus petit possible. C'est nous qui devons spécifier le nombre de clusters que l'algorithme utiliseras et donc nous commencerons par trouver le nombre de clusters qui mène à la meilleure performance. Voici le graphique de la performance global de l'algorithme selon le nombre de clusters :



Voici le graphique de l'inertie intra-clusters selon le nombre de clusters :



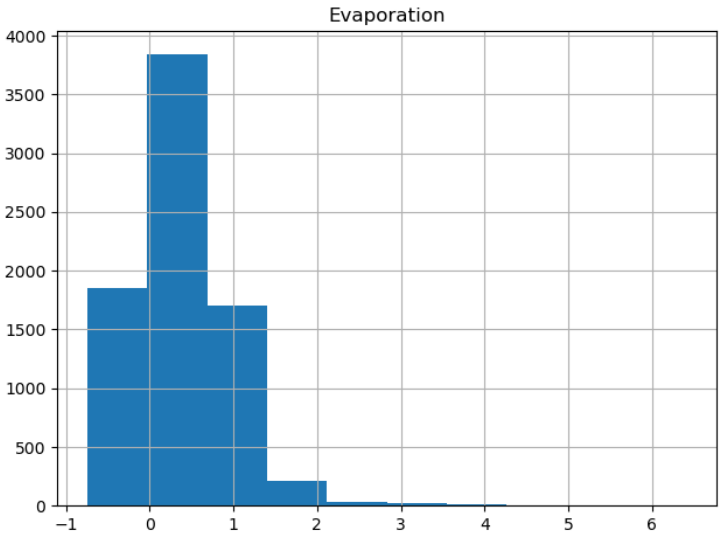
On voit que généralement un nombre de clusters plus élevé augmente la performance, dans notre cas la meilleure précision est obtenue lorsque nous utilisons 23 clusters, avec une précision de 79,97%. On remarque aussi que l'inertie intra-cluster diminue plus il y a de clusters, mais à un certain point la diminution ne vaut plus la peine comparativement au nombre de clusters. Nous allons donc conserver un nombre de cluster de 23 pour le reste des expérimentations.

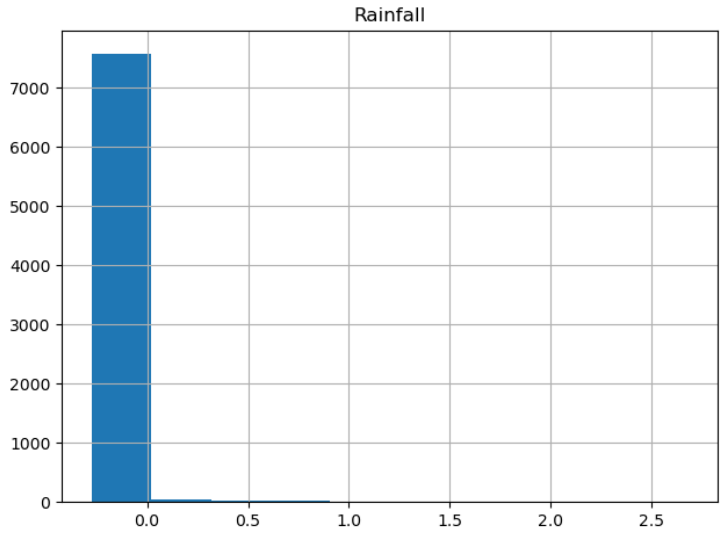
Voici le tableau de précision de chaque cluster individuellement :

Le cluster numéro 8 a une précision de 96% alors que le cluster 14 en a une de 51%. Avec un écart-type de 13,59%, La variation de la précision entre les clusters n'est pas si élevée.

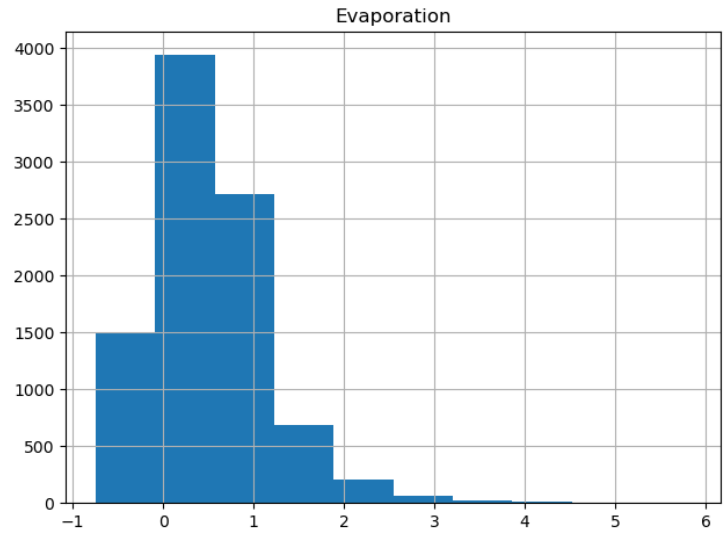
0	0.719901
1	0.892772
2	0.769463
3	0.673619
4	0.685696
5	0.940547
6	0.825929
7	0.919198
8	0.961513
9	0.748227
10	0.611252
11	0.584418
12	0.869394
13	0.950278
14	0.516598
15	0.911030
16	0.667365
17	0.903851
18	0.944339
19	0.708576
20	0.586987
21	0.792908
22	0.802580

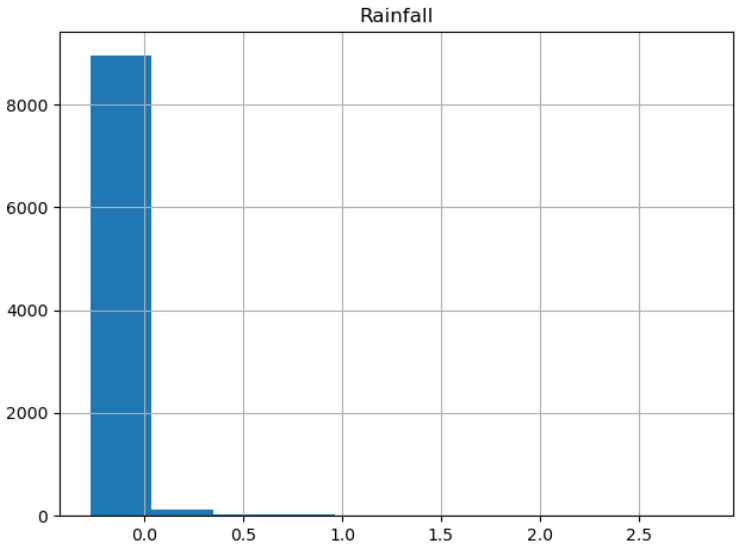
Nous allons ensuite jeter un coup d'œil aux histogrammes des colonnes Evaporation et Rainfall des cluster 8 et 13. J'ai choisi ces colonnes car logiquement cela serait ces colonnes qui aurait plus d'impact sur s'il va pleuvoir le lendemain. Je crois cela car normalement une plus grande évaporation de la pluie tombée formera plus de nuage de pluie qui mèneront à de la pluie dans le futur, on devrait effectuer une analyse des composants principaux pour en être certain. J'ai choisi les clusters 8 et 13 car ce sont ceux qui ont la plus grande précision, respectivement 96,15% et 95,02%. Voici pour le cluster 8 :





Et pour le cluster 13 :

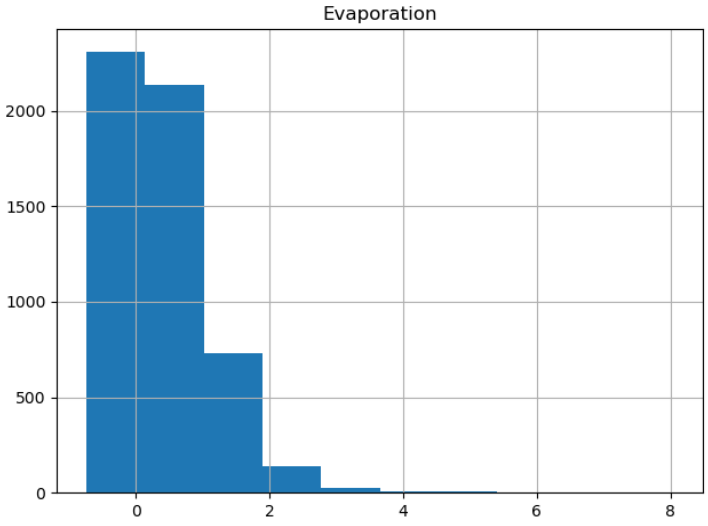


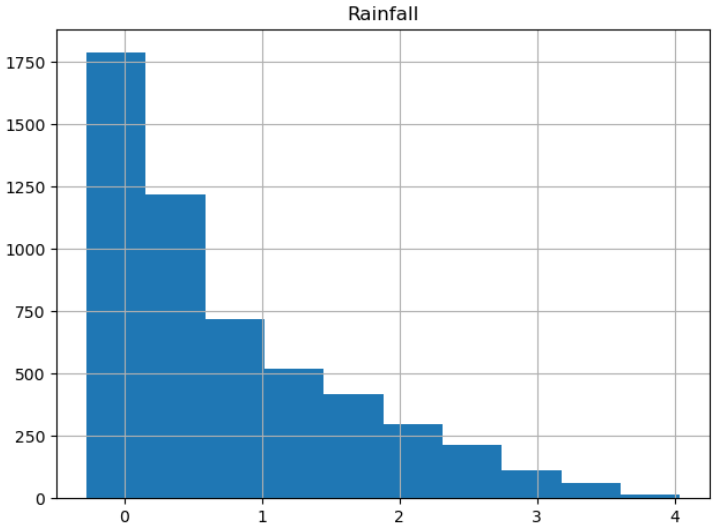


On peut remarquer que ces deux clusters sont pratiquement pareils si l'on regarde seulement ces attributs. Ce sont tous les deux des clusters où l'évaporation de l'eau est faible et où il n'y a pas ou presque pas tombé de pluie.

On peut donc voir que ces deux clusters sont très bons pour rassembler les instances qui se ressemblent d'où la précision supérieure à 95%.

J'ai répété la même expérience mais pour le cluster 14, qui a une précision de seulement 51,66% :





On voit tout de suite que la distribution des valeurs de ces attributs est beaucoup plus dispersée que les autres clusters, surtout l'attribut Rainfall. Cela a sans doute un impact négatif sur la précision et explique probablement la précision médiocre de ce cluster.

Maintenant que nous avons explorer K-Means, nous allons passer à un deuxième algorithme de clustering.

Application de l'algorithme de clustering hiérarchique

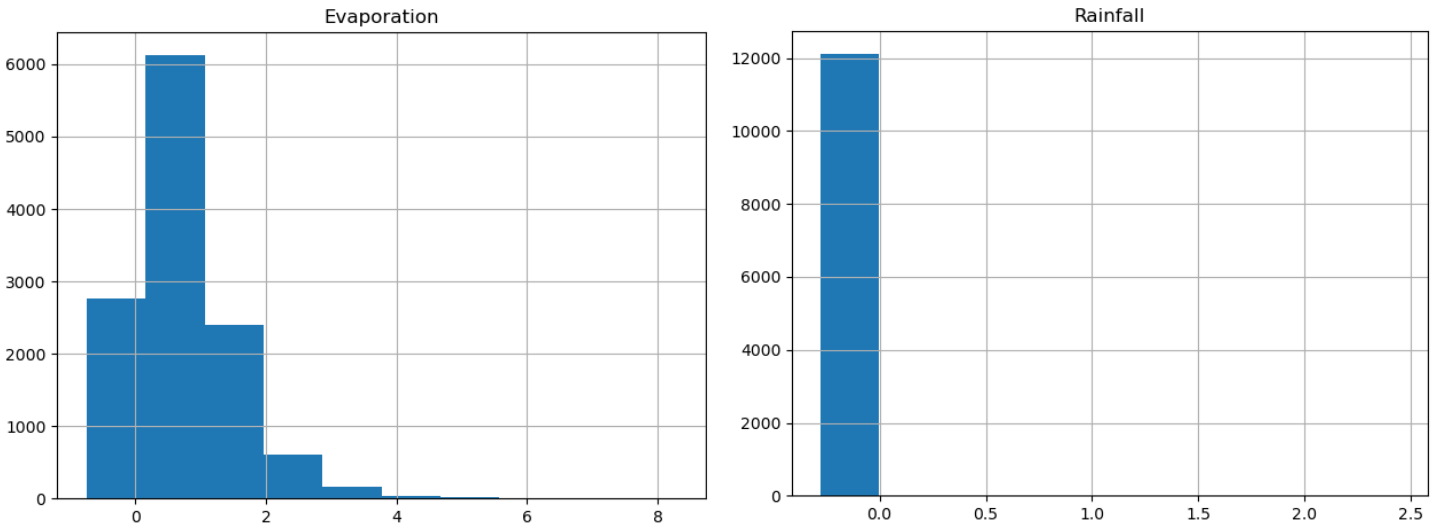
On continue l'exploration des algorithmes avec le clustering hiérarchique, celui-ci construira un dendrogramme avec les instances et regroupera les instances similaires ensemble jusqu'à ce qu'il n'y ait qu'un seul cluster. Ensuite on doit trouver à quel endroit on veut « couper » le dendrogramme afin d'obtenir le nombre de clusters voulu. Nous utiliserons la méthode de la variance minimale de Ward et la métrique Euclidienne afin de classer les instances de l'ensemble de données.

En coupant le dendrogramme pour obtenir 23 clusters tel que convenu lors des tests sur l'algorithme k-means, on obtient une précision moyenne de 79,35%. Voici la liste plus détaillée des précisions par clusters :

Le cluster 6 avec la précision la plus élevé de 95,91% et le cluster 0 avec la précision la plus faible de 50,26%, l'écart-type est de 13,67%, ces résultats sont pratiquement les même que l'algorithme de clustering k-means.

```
Accuracy by cluster from hierarchical clustering:
 0     0.502563
1     0.821244
2     0.879127
3     0.622307
4     0.718261
5     0.918603
6     0.959056
7     0.904859
8     0.620486
9     0.766467
10    0.518171
11    0.698529
12    0.604480
13    0.664147
14    0.656165
15    0.789185
16    0.956554
17    0.901092
18    0.893300
19    0.770751
20    0.818035
21    0.840364
22    0.856577
```

J’ai aussi tracé la distribution des valeurs d’un attributs des instances pour le meilleur cluster, le 6, la voici :



Je dirais que c’est quasiment un copier-coller de la distribution des meilleurs clusters de l’algorithme K-Means et on peut donc conclure que l’algorithme de clustering hiérarchique sépare les instances très similairement à K-Means.

Nous allons maintenant passer au dernier algorithme que nous allons explorer, DBSCAN.

Application de l'algorithme de clustering DBSCAN

On finit nos tests avec l'algorithme DBSCAN. Celui-ci va regrouper les instances selon la densité de ceux-ci dans un espace déterminé. Nous déterminerons une distance minimale pour laquelle deux instances sont des voisins et aussi une valeur qui déterminera combien de voisin il faut pour former un cluster. Nous avons plusieurs hyperparamètres que nous pouvons modifier mais nous nous concentrerons sur celui qui dicte la distance minimale que deux instances doivent avoir entre eux pour être considéré des voisins.

Nous commençons avec une distance eps=0.5 :

Cluster Results for DBSCAN

	cluster	clusterCount
0	-1	145024
1	9	235
2	12	53
3	13	29
4	1	23
5	14	21
6	0	16
7	5	11
8	4	9
9	11	8
10	7	6
11	10	5
12	8	5
13	6	5
14	3	5
15	2	5

On voit immédiatement qu'une distance de 0.5 est trop peu élevé car presque toutes nos instances n'ont pas été classer dans un cluster. On essaie donc ensuite avec eps=3 :

Cluster Results for DBSCAN

	cluster	clusterCount
0	0	143995
1	-1	1422
2	5	18
3	3	6
4	6	5
5	4	5
6	1	5
7	2	4

Une distance de 3 est trop élevée, presque toutes nos instances sont dans le même cluster, on réduit donc la distance à 1.5 :

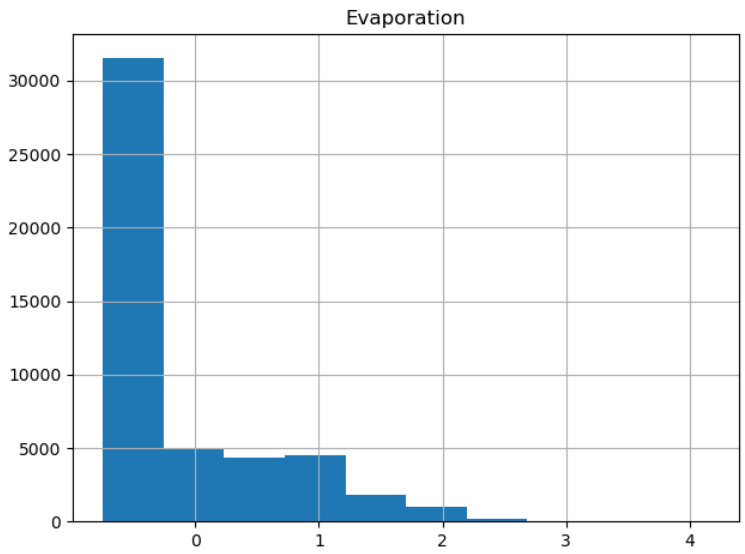
Cluster Results for DBSCAN

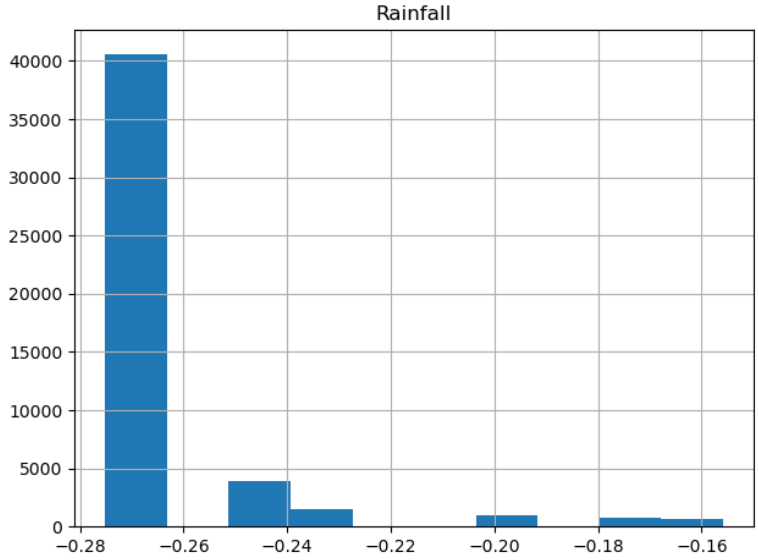
	cluster	clusterCount
0	-1	87733
1	0	48494
2	17	1728
3	6	689
4	59	506
..	...	...
621	54	2
622	328	2
623	159	2
624	414	2
625	511	2

On arrive finalement à un résultat médiocre, avec plusieurs essais c'est la distance qui donne le meilleur résultat. On retrouve tout de même plus de la moitié des instances qui ne sont pas classé dans un cluster.

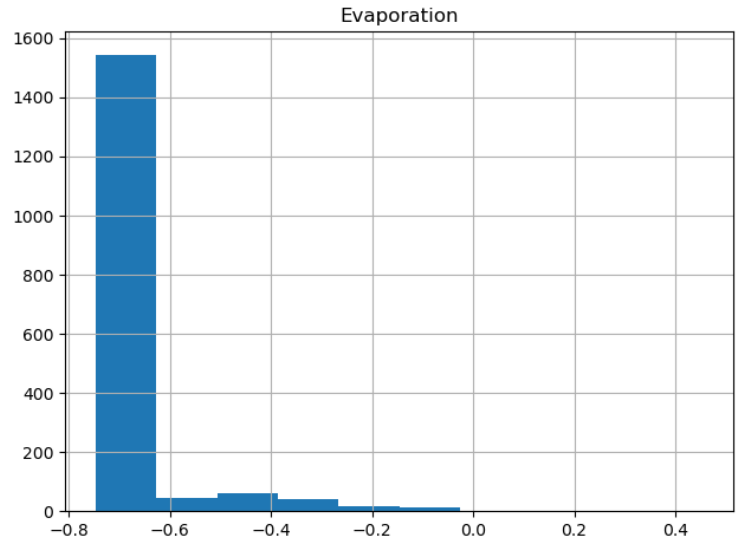
Le deuxième paramètre que l'on peut modifier est min_samples, qui détermine le nombre d'instances voisines nécessaire pour qu'un cluster soit formé. Après plusieurs essais, le meilleur nombre est celui par défaut qui est 5. Avec les paramètres eps=1.5 et min_samples=5, Avec 625 clusters il serait assez inutile de montrer la précision par cluster, on peut tout de même calculer un écart-type de 16,08%. Cela est plus grand que les autres algorithmes mais tout de même faible. On obtient une précision globale de l'algorithme de 78,42%.

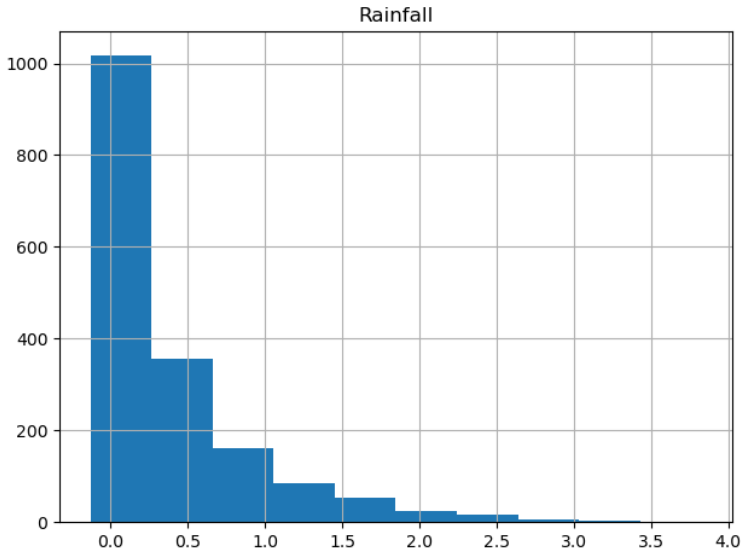
J’ai encore une fois fait un histogramme de la distribution des attributs Evaporation et Rainfall sur les deux clusters qui contenaient le plus de valeurs, les clusters 0 et 17. Voici pour le cluster 0 :





Voici le cluster 17 :





On voit que la séparation des instances selon ces attributs n’est pas aussi bonne que K-Means mais on observe tout de fois des différences entre les deux clusters. Le cluster 0 semble regrouper des instances de jour où il n’y a pas plu alors que le cluster 17 contient plus de journées où il y a eu de la pluie. Il y donc une certaine séparation des instances mais on voit avec les histogrammes qu’elle est loin d’être parfaite.

Conclusion de l'étude

Finalement, les trois algorithmes de clustering ont présenté des résultats très similaire si on ne regarde que la précision moyenne des clusters. Cependant, dès que l'on creuse un peu plus et que l'on connait un peu plus de détails on se rend compte que ces algorithmes sont loin d'être parfaits.

K-Means a la meilleure précision globale avec une précision de 79,97%. Il possède aussi la plus petite variation de précision inter-cluster.

On retrouve environ la même histoire avec le clustering hiérarchique avec une précision très similaire à l'algorithme K-Means.

En revanche, l'histoire est différente pour DBSCAN. Malgré le fait que la précision du modèle soit de 78%, plus de la moitié de instances ne sont pas dans des clusters. On ne peut donc pas se fier du tout à ce modèle et il est de loin le pire des trois.

Je crois que mon hypothèse de départ était bonne, les trois algorithmes sont tombés en haut de 76% de précision. Je ne crois pas que cela soit très impressionnant si l'on sait qu'il n'y a que 25% des journées en Australie ou il pleut. Cependant, pour un algorithme d'apprentissage non supervisé qui ne connait pas les labels des instances, c'est impressionnant. Le fait de pouvoir prédire s'il va pleuvoir en Australie correctement 80% du temps est très respectable.

Conclusion Générale

Pour conclure, j'ai bien apprécié ce travail et aussi de pouvoir mettre en pratique des connaissances acquises sur des données du monde réel. J'ai appris quelques choses nouvelles sur la météorologie, surtout en rapport avec des façons de mesurer l'évaporation de l'eau ou bien la couverture du ciel.

C'est surtout intéressant et surprenant de voir à quel points un modèle non supervisé qui ne connait pas du tout les labels des instances est capable de les classer de façon cohérente et d'arriver à prédire avec une précision acceptable les instances non classées.