# Données météorologique françaises de 2001 à 2010 - NF26

TD1 3-4 | BOURGEON Théodore | Haoxuan Dong

Printemps 2019

**Résumé**

Dans le cadre de l'UV NF26, nous devons réaliser un stockage de données en haute volumétrie relative aux résultats météorologiques METAR de station française de 2001 à 2010. Nous devrons être en mesure de répondre aux objectifs détaillés dans l'introduction et présenter ces résultats de manière interprétable. Nous utiliserons Python, Cassandra et Spark. Le code se situe dans le répertoire dans le lien en annexe A.

## 1 Introduction

Nous devrons construire un outil répondant aux questions suivantes :

1. Pour un point donné de l'espace, je veux pouvoir avoir un historique du passé, avec des courbes adaptées. Je veux pouvoir mettre en évidence la saisonnalité et les écarts à la saisonnalité.
2. À un instant donné je veux pouvoir obtenir une carte me représentant n'importe quel indicateur.
3. Pour une période de temps donnée, je veux pouvoir clusteriser l'espace, et représenter cette clusterisation.

## Table des matières



## 2 Présentation des données

Les données présentent un certain nombre de caractéristiques :

1. Station : Three or four characters site identifier
2. Valid : Timestamp of the observation
3. lon : Longitude
4. lat : Latitude
5. Tmpf : Air Temperature in Fahrenheit, typically at 2 meters
6. Dwpf :Dew Point Temperature in Fahrenheit, typically at 2 meters
7. Relh : Relative Humidity in %
8. Drct : Wind Direction in degrees from north
9. Sknt : Wind Speed in knots
10. P01i : One hour precipitation for the period from the observation time to the time of the previous hourly precipitation reset. Values are in inches. This value may or may not contain frozen precipitation melted by some device on the sensor or estimated by some other means.
11. Alti : Pressure altimeter in inches
12. Mslp : Sea Level Pressure in millibar
13. Vsby : Visibility in miles
14. Gust : Wind Gust in knots
15. Skyc1 : Sky Level 1 Coverage
16. Skyc2 : Sky Level 2 Coverage
17. Skyc3 : Sky Level 3 Coverage
18. Skyc4 : Sky Level 4 Coverage
19. Skyl1 : Sky Level 1 Altitude in feet

20. Skyl2 : Sky Level 2 Altitude in feet
21. Skyl3 : Sky Level 3 Altitude in feet
22. Skyl4 : Sky Level 4 Altitude in feet
23. Wxcodes : Present Weather Codes
24. Feel : Apparent Temperature in Fahrenheit
25. Ice_accretion_1hr : Ice Accretion over 1 Hour
26. Ice_accretion_3hr : Ice Accretion over 3 Hours
27. Ice_accretion_6hr : Ice Accretion over 6 Hours
28. Peak_wind_gust : Peak Wind Gust (knots)
29. Peak_wind_drct : Peak Wind Gust Direction (deg)
30. Peak_wind_time : Peak Wind Gust Time
31. Metar : unprocessed reported observation in METAR format

On ajoutera la longitude et la latitude afin de pouvoir effectuer une indexation géographique. Les données vides seront caractérisées par le symbole $M$. [1]

Exemple d'une ligne de données : *LFRN, 2001-12-31 15 :00, -1.7339, 48.0689, 41.00, 30.20, 65.16, 50.00, 10.00, M, 30.45, M, 6.21, M, FEW, M, M, M, 3000.00, M, M, M, M, M, M, M, M, M, M, 34.25, LFRN 311500Z 05010KT 9999 FEW030 05/M01 Q1031 NOSIG*

Dans chaque famille de colonnes nous n'insèrerons pas toutes les caractéristiques. Seulement certaines d'entre elles qui nous seront utiles pour les affichages. En effet la quantité de données est colossale et pour accélérer l'insertion, on maximisera la taille des batchs (qui ont une taille maximale).

# 3 Objectif 1

*Pour un point donné de l'espace, je veux pouvoir avoir un historique du passé, avec des courbes adaptées. Je veux pouvoir mettre en évidence la saisonnalité et les écarts à la saisonnalité.*

## 3.1 Description du stockage

### 3.1.1 Première interprétation

Ici, pour répondre au mieux à la question nous avons choisi comme clé :

1. Partitionnement : Station
2. Tri : Année, Mois, Jours, Heure, minutes

Dans ce cas, la clé de partitionnement n'est bornée que parce qu'on travaille sur un jeu de données fixé. On peut ainsi se permettre de l'utiliser telle quelle. Sinon on aurait du y ajouter une donnée temporelle pour limiter la taille des partitions.

1. Plus d'informations sur les caractéristiques ici : https ://www.weather.gov/media/asos/aum-toc.pdf

### 3.1.2 Deuxième interprétation

Nous avons par la suite, interprété "Un point donné de l'espace" non plus comme une simple station mais comme des coordonnées géographiques quelconques. Pour cela nous implémenterons une indexation géographique. Nous utiliserons une notion de « pavés de localisation » que l'on appellera « area ». Ces areas quadrillerons la France. Chaque relevé GPS pourra alors correspondre à l'une de ces areas afin d'être utilisé comme clé de partition. Ici nous avons choisi d'arrondir au degré (noté respectivement $lon_t$ et $lat_t$). Cela formera des pavés couvrant environ 12321 km2.

1. Partitionnement : $lon_d$, $lat_d$
2. Tri : Station,longitude, latitude, Année, Mois, Jours, Heure, minutes

Ainsi nous aurons plusieurs méthodes d'étude :

1. Pour un point géographique donné, nous chercherons dans le pavé correspondant, s'il y a une station ou plusieurs et nous afficherons le résultats de la station la plus proche.
2. Pour un point géographique donné, nous chercherons dans les 9 pavés (de $lon_t - 1$ à $lon_t + 1$ et $lat_t - 1$ à $lat_t + 1$ et nous afficherons le résultat de la station la plus proche. En effet nous pouvons nous trouver à la bordure d'un pavé et être plus proche d'une station d'un pavé adjacent que d'une station à l'intérieur du pavé actuel.

## 3.2 Résultats

### 3.2.1 Première interprétation

On souhaite obtenir des graphiques de température contenant la température réelle, la moyenne et le ressenti. Dans un premier temps, à la demande des informations d'une station et d'une année précise, on génère les graphiques suivant :

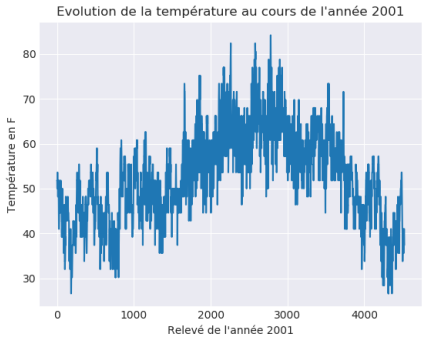


FIGURE 1 – Évolution de la température au cours de l'année 2001

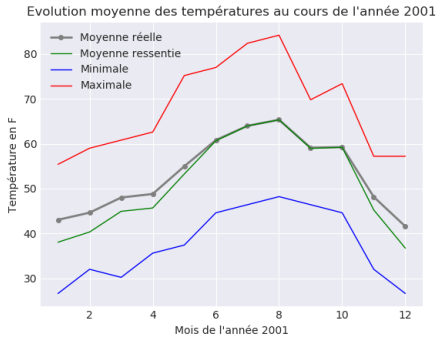


FIGURE 2 – Évolution moyenne des températures au cours de l'année 2001

Nous pouvons également obtenir des informations uniquement depuis la station :

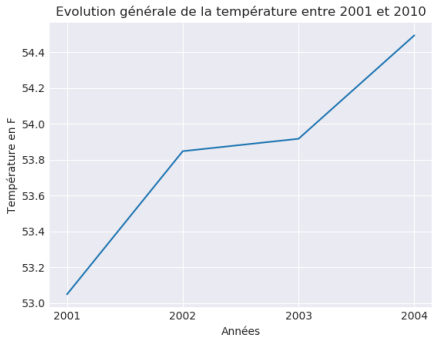


FIGURE 3 – Évolution générale des températures entre 2001 et 2010

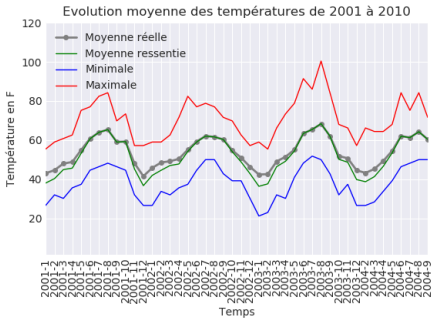


FIGURE 4 – Évolution moyenne des températures entre 2001 et 2010

### 3.2.2 Deuxième interprétation

Pour un point GPS nous obtenons des graphiques similaires qui représentent les résultats de la station la plus proche du pavé ou de la fenêtre correspondant.

# 4 Objectif 2

*À un instant donné je veux pouvoir obtenir une carte me représentant n'importe quel indicateur.*

## 4.1 Description du stockage

Pour répondre au mieux à la question nous avons choisi comme clés :

1. Partitionnement : Année
2. Tri : Mois, Jours, Heure, minutes, Station

Ici la clé de partitionnement est bornée.

## 4.2 Résultats

Nous avons défini un instant donné comme une date complète (sous le format $YYYY-MM-DDHH:MM$). Il est important de noter que nous possédons des informations uniquement pour les heures et demi-heures.


FIGURE 5 – Détail des informations météorologiques du 2001-01-01 00 :00

# 5 Objectif 3

*Pour une période de temps donnée, je veux pouvoir clusteriser l'espace, et représenter cette clusterisation.*

## 5.1 Description du stockage

On pourrait ici reprendre le modèle du stockage de l'objectif 2. Cependant nous en avons implémenté un nouveau afin de nous familiariser encore avec cet environnement.

1. Partitionnement : Année, Mois, Jours
2. Tri : Station, Heures, Minutes

### 5.2 Résultats

Afin de clusteriser cet espace nous effectuerons, pour une période donnée, un k-means sur les caractéristiques suivantes : Température moyenne, % d'humidité le vent et sa direction. Nous représentons, en fonction du nombre de clusters choisis, une cartographie des stations regroupées par cluster. Il est important de noter que l'algorithme des K-means dépend des points d'initialisation et nous n'atteignons pas forcément des optimums globaux mais locaux. Ici nous n'utilisons les k-means que dans sa partie modèle : nous nous servirons de sa clusterisation pour l'afficher mais pas sa fonction prédictive.

Voici l'algorithme générique des K-means sur lequel nous nous sommes basés pour la réalisation de cette partie.

```
dataSet : jeu de données considéré
centroïdes ← [x_i in R^d]
while [condition d'arrêt du k-means]:
 |  next_sum_pts ← 0_R^(K\times d)
 |  next_sum_nb ← 0_R^K
 |  for x_i in dataSet:
 |   |  k ← index du centroïde le plus proche
 |   |  next_sum_pts[K] ← next_sum_pts[K] + x_i
 |   |  next_sum_nb[K] ← next_sum_nb[K] + x_i
 |
 |  for k in 0:K-1:
 |   |  centroïdes[k] ←
 |          next_sum_pts[k] / next_sum_nb[k]
 |
```

FIGURE 6 – Algorithme des K-means

FIGURE 7 – 5 clusters des stations du 01-01-2001 au 31-10-2001

FIGURE 8 – 2 clusters des stations du 01-01-2001 au 10-03-2008

Nous pouvons vérifier ce résultat de manière intuitive en fonction des climats réels de ces zones.

Dans un premier temps, l'algorithme des K-means du package scikit-learn (cf. fichier processingQ3.py) a été utilisé afin de construire la carte et les fonctions annexes (sur un petit jeu de données). Par la suite nous avons repris cette partie pour effectuer l'algorithme des k-means via Spark sur le jeu de données total (cf. fichier processingQ3_kmeans et kmeans.py).

## 6 Interface CLI

Pour des raisons de simplicité, nous avons mis en place une interface en ligne de commande qui permet d'accéder aux différents objectifs.

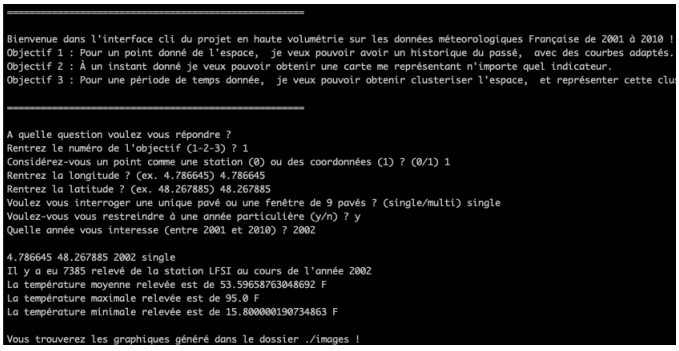


FIGURE 9 – Interface cli

## 7 Conclusion

Nous avons dû faire face aux différentes problématiques générées par ces objectifs, dans la conception du stockage, de l'insertion, de l'utilisation des indicateurs et de la présentation des informations. Nous avons, pour chaque ques-

tion détaillé les indicateurs qui nous semblaient les plus pertinents. Une étude similaire pourrait être faite sur tous les indicateurs. De plus en fonction des besoins (de nouveaux objectifs) nous pourrions approfondir certains domaines. Nous avons décidé de travailler sur un jeu de données aussi grand que possible mais la quantité de données massives générait des interruptions dans l'insertion. Nous avons optimisé la taille des batchs afin d'accélérer le processus et nous avons travaillé dans un premier temps sur une partie des données avant d'insérer, jusqu'aux limites techniques, un maximum de données dans nos familles de colonnes. Le travail avec des données en haute volumétrie engendre de nombreux challenges et une approche différente à l'analyse des données.

## Table des figures



## Références


[1] ASOS-AWOS-METAR Data Download Lien.
[2] Geohash Lien.
[3] Script Lien.
[4] Geohash Lien.
[5] Python Cassandra Driver Lien.
[6] Folium package Lien.
[7] Matplotlib package Lien.


## A Dépôt de code

Vous pourrez trouver le détail du code réalisé pour le projet au sein ici : https://gitlab.utc.fr/thbourge/nf26_p19