République Algérienne Démocratique et Populaire
الجمهـــورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

École nationale
Supérieure
d'Informatique

المدرسة الوطنية العليا للإعلام الآلي

Master 2019-2020

Machine learning

DBSCAN spatial

Réalisé par :

- Haddad Oussama
- Mahamdi Mohammed

Groupe : 2

5 Novembre 2019

Table des matières

Introduction

l'objectif de ce TP est l'application de l'Algorithme BDSCAN(Density-Based Spatial Clustering of Applications with Noise) pour la classification non supervisée ,ce travail est fait en langage Python avec plusieurs packages :

1. sklearn
2. numpy
3. matplotlib
4. basemap
5. seaborn
6. pandas

le code source est accessible via ce lien :
https://github.com/MahamdiAmine/DBSCAN-clustering

il existe plusieurs algorithme de classification, et le choix de ces algorithms ce fait par rapport au domaine d'application , dans certain cas il nécessite l'intervention d'un expert pour fixer certains paramètres.

Les point fort du DBSCAN:

- La connaissance minimale du domaine est suffisante.
- L'algorithme peut découvrir des clusters de forme arbitraire.
- Très Efficace pour les grandes dataset.
- L'algorithme est déterministe

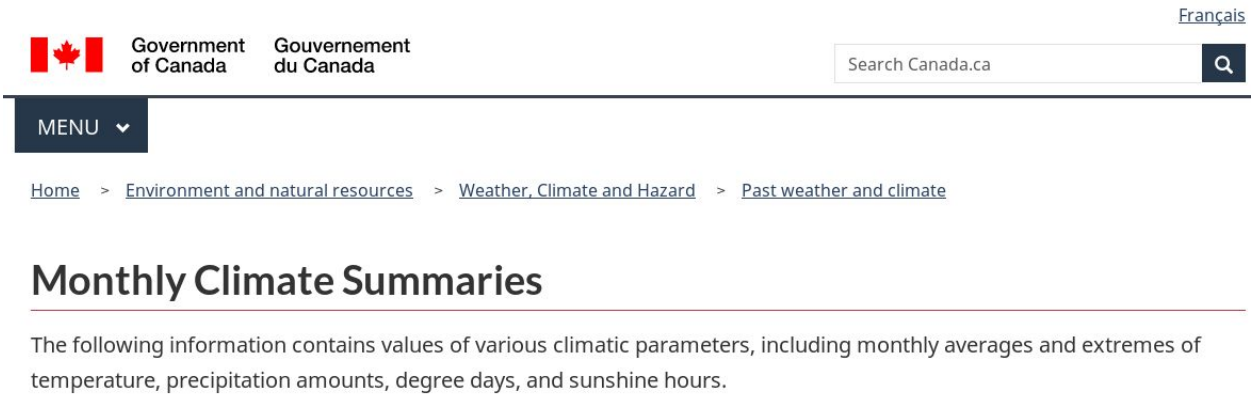
Le concept principal de l'algorithme DBSCAN consiste à localiser des régions de haute densité séparées les unes des autres par des régions de faible densité.

Tout les étapes de l'algorithme sont bien expliqué en cours.

Data set

Le site officiel du Canada

Le site officiel du gouvernement du Canada : Canada.ca



La section de la météo

Site web : Climate.weather.gc.ca

“Accédez aux conditions météorologiques historiques, à des données climatiques et à de l’information connexe pour plusieurs endroits à travers le Canada. Sur ce site, vous trouverez entre autres de l’information sur la température, les précipitations, les degrés-jours, l’humidité relative, la vitesse du vent et sa direction, les sommaires mensuels, les moyennes, les extrêmes et les normales climatiques.”

C’est un site officiel qui fournit des données fiables.

Le lien de la dataset

https://climate.weather.gc.ca/prods_servs/cdn_climate_summary_report_e.html?intYear=2019&intMonth=10&prov=&dataFormat=csv&btnSubmit=Download+data

Information de la dataset

Contenu

Données météorologiques du Canada pour le mois **Octobre 2019**

Date des donées

Octobre 2019

Taille (nombre des lignes)

Cette dataset contient 1240 données (lignes)

La structure (les colonnes)

Cette dataset contient 25 colonnes :

* Long === Longitude

* Lat === Latitude

* Stn_Name === Station Name

* Prov === Province

* Tm === Mean Temperature (°C)

* Tn === Lowest Monthly Minimum Temperature

* Tx === Highest Monthly Maximum Temperature

* DwTm === Days Without Valid Mean Temperature

* DwTx === Days Without Valid Maximum Temperature

* DwTn === Days Without Valid Minimum Temperature

* D === Mean Temperature Difference from Normal

* S === Snowfall (cm)

* DwS === Days Without Snowfall

* S%N === Percent of Normal Snowfall

* P === Total Precipitation (mm)

* DwP === Days Without Valid Precipitation

* P%N === Percent of Normal Precipitation

* Pd === No. of days with precipitation 1mm or More

* BS === Bright Sunshine days

* DwBS === Days Without valid Bright Sunshine

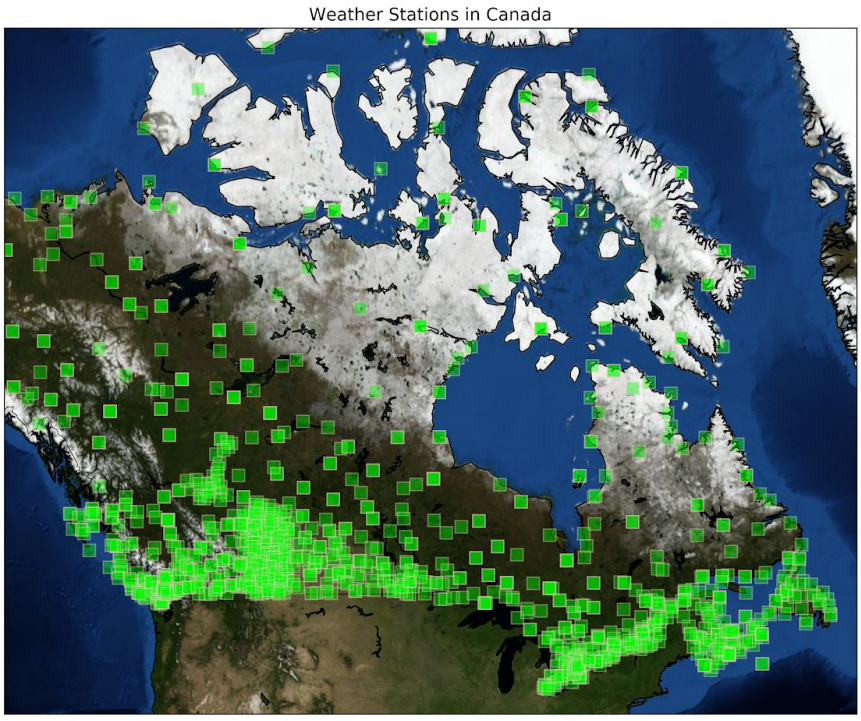
* BS% === Percent of Normal Bright Sunshine

* HDD === Degree Days Below 18° C

* CDD === Degree Days Above 18° C

* Stn_No === Station Number; Climate Station Identifier (1st 3 Digits==Indicate drainage basin, Last 4 Digits Sorting Alphabetically)

Affichage de la dataset



Voici un aperçu (dataset) sur la distribution des stations météorologiques sur la cartographie de Canada

Implémentation du code

Lors de l'implémentation du code ,on a suivi les concept des “Clean Code” qui sont :

- le respect du nom du variable significatives.
- Lecode explique bien l’intente.
- “**single responsibility**” function.
- l’utilisation du tests.
- un code incrémental et itérative.
- l’indépendance de architecture.

La première étape consiste à importer tous les packages nécessaires.

Ensuite on a définis quelque variables booléens:

show: pour monter les plots à chaque exécution.

Save : pour enregistrer et écraser les résultats à chaque exécution.

Ensuite on doit charger la dataset , cela se fait par la fonction **Pandas.read_csv()**

ensuite les donner nécessite un filtrage(par exemple il existe des enregistrements ont des valeurs nulles).

Avec la librairie **Basemap** on a créer une cartographie pour présenter les région du dataset avant la classification , et la visualisation se fait avec **matplotlib .**

Pour le Clustering on a essayé de combiner plusieurs variables , mais les variable essentiels sont:

- **longitude.**
- **Latitude.**
- **La température minimale.**
- **La température maximale.**
- **La température moyenne.**

On a fixer Epsilon à **0.34** et Le MinPts à **9**.

Enfin on affiche le résultat du clustering sur la cartography.

Remarque:

Pour exécuter le code il faut utiliser les arguments suivantes:

Show: booléen.

Save :booléen.

Data_path : le chemin vers le dataset.

Eps :Epsilon(valeur réelle).

MinPts : le nombre de points minimaux(entier).

Exemple:

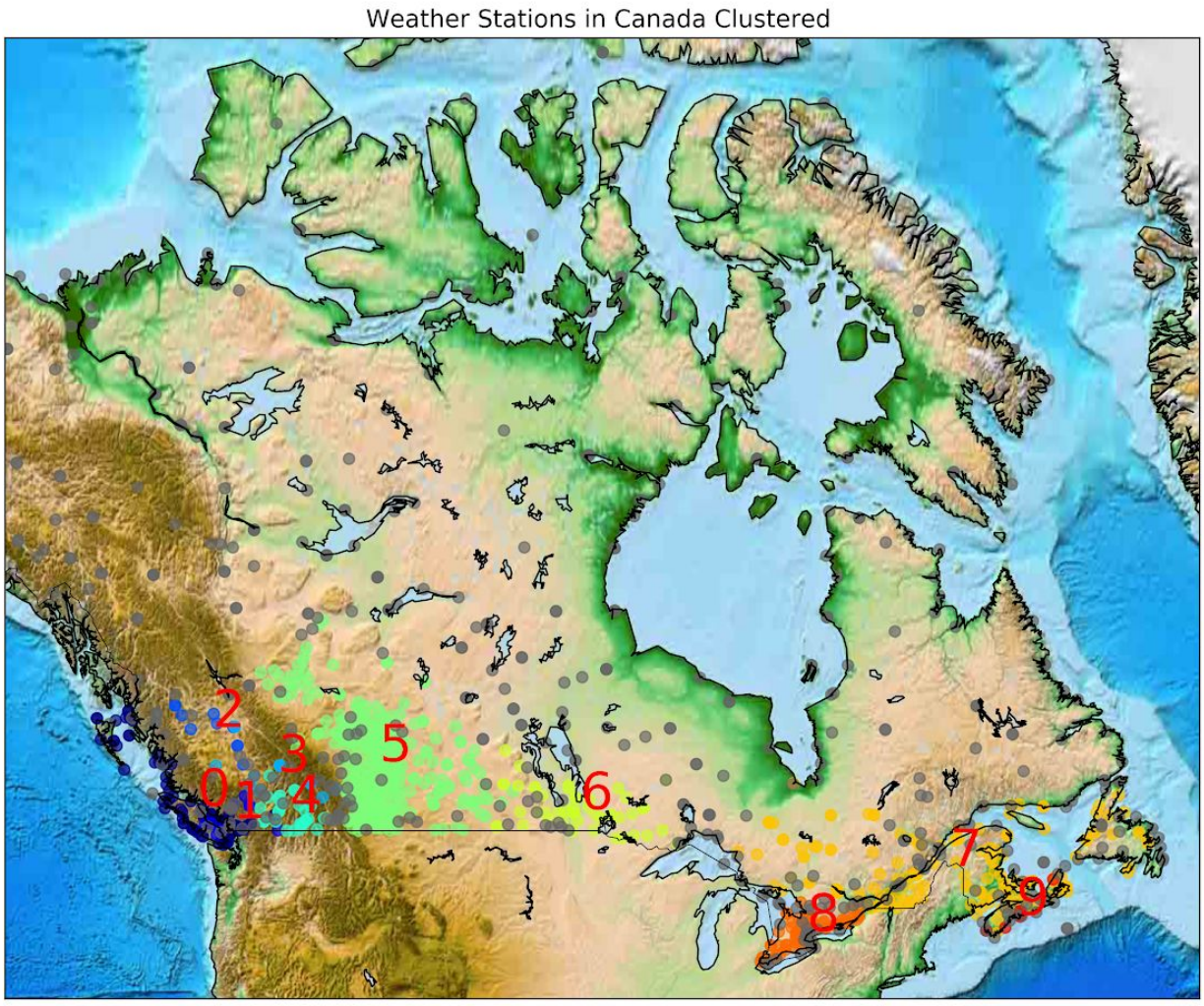
Python3 main.py True True en_climate_summaries_All_10-2019.csv 0.34 9

Résultats

Pour la configuration suivante:

Nombre des variables utilisées	5
Les variables	Lon, Lat, Tem_Max,Temp_Min,Temp_moy
La taille des instances	1194
Epsilon	0.34
MinPts	9

Nous avons obtenus les résultats suivants:



On a obtenu 10 clusters , la température moyenne dans chaque cluster est :

- Cluster 0, Average Mean Temp: 9.227835051546393
- Cluster 1, Average Mean Temp: 8.1
- Cluster 2, Average Mean Temp: 3.963636363636364
- Cluster 3, Average Mean Temp: 2.707142857142857
- Cluster 4, Average Mean Temp: 5.475
- Cluster 5, Average Mean Temp: 1.758545454545456

- Cluster 6, Average Mean Temp: 3.3204081632653057
- Cluster 7, Average Mean Temp: 7.2706249999999955
- Cluster 8, Average Mean Temp: 9.903225806451614
- Cluster 9, Average Mean Temp: 10.376923076923077

Interprétation des résultats

On a obtenu un clustering qui contient 10 cluster , et il sont les stations météo qui montre les mêmes conditions météorologiques , c'est t'a dire chaque cluster contient les endroits qu'il ont les même conditions météorologiques.

Pour l'interprétation , elle nécessite l'intervention d'un expert , mais en générale "les zones géographiques voisins ont les mêmes propriétés météo".

A partir de la représentation des clusters dans la cartographie ,on peut dire que nos résultats ont une très bonne qualité.

Accélération de l'exécution

L'exécution de ce programme dans une machine standard prend beaucoup de temps , pour cela on a utilisé une machine puissante de google , avec les performances :

- RAM 13 Go
- GPU acceleration de ram 12 Go
- Processeur Intel Xeon CPU

Grâce à Google qui offre ces machines performantes gratuitement sur Colab

Références

1. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”; Martin Ester et.al.

2. Les cours de Madame HAMDAD Leila.
3. Fast.ai

4. L’article: https://charleshsliao.wordpress.com/2017/05/30/clustering-algorithms-evaluation-in-python/ consulter le:05/11/2019