ROYAUME DU MAROC
--*-*-*
HAUT COMMISSARIAT AU PLAN
--*-*-*-*-*-*-*
INSTITUT
NATIONAL
DE STATISTIQUE ET D'ECONOMIE APPLIQUEE
INSEA

Projet de Fin d'Etudes

*****

Clustering des saisons agricoles en se basant sur les données météorologiques

Préparé par : ***Mme Oumayma CHATIBI***

Sous la direction de : ***M. Mohammed EL HAJ TIRARI*** **(INSEA)**
M. Hamza LAHKIM **(AgriEdge - UM6P)**

Soutenu publiquement comme exigence partielle en vue de l'obtention du

Diplôme d'Ingénieur d'Etat

Filière : Statistique et économie appliquée

Devant le jury composé de :

- ***M. Mohammed EL HAJ TIRARI (INSEA)***
- ***M. Fouad EL ABDI (INSEA)***
- ***M. Hamza LAHKIM (AgriEdge - UM6P)***

Juin 2022 / PFE N° 165

Résumé

Depuis sa création en 2018, AgriEdge a pris une place de plus en plus importante dans la digitalisation, le soutien et l'amélioration de la production agricole. En effet, l'exploitation de l'agriculture de précision apporte aux agriculteurs des solutions pouvant affecter les performances économiques et environnementales de la production.

Dans ce contexte, AgriEdge adopte une approche basée sur la recherche et le développement pour promouvoir une agriculture durable. En transformant les données en propositions exploitables pour les agriculteurs afin d'améliorer la rentabilité.

L'objectif de notre étude est de pouvoir détecter des similitudes entre les saisons agricoles en précisant la culture et la zone géographique, et en exploitant les données météorologiques. Notre étude porte sur le regroupement des saisons agricoles pour deux types de cultures : les agrumes et le blé.

La première étape est la spécification de la période des saisons agricoles et des stations géographiques ainsi que des variables météorologiques ayant une influence directe sur les cultures agricoles. Ensuite, la collecte de ses données quotidiennes à travers les API, puis la préparation, l'organisation et la visualisation des saisons agricoles pour les deux cultures choisies. La deuxième étape est le regroupement des saisons agricoles pour chaque variable météorologique en utilisant l'algorithme k-means adopté pour les séries chronologiques et la mesure de similarité « Dynamic Time Wrapping » (DTW). Pour la dernière étape, il s'agit d'exploiter tous les résultats valides des clusterings précédents de chaque variable et d’admettre une approche statistique afin de retrouver les similarités entre les saisons en exploitant toutes les variables météorologiques en même temps.

Mots clés : Agriculture de précision, saison agricole, culture agricole, API, données météorologiques, clustering, Time series K-means, déformation dynamique temporelle.

Abstract

Since its creation in 2018, AgriEdge has taken an increasingly importante place in supporting and improving agricultural production. Indeed, the exploitation of precision agriculture provides farmers solutions that can affect the economic and environmental performance of production.

In this context, AgriEdge adopts an approach based on research and development to promote sustainable agriculture. By turning data into actionable propositions for farmers to improve profitability.

The objective of our study is to be able to detect similarities between agricultural seasons by specifying the crop and the geographical area, and by exploiting meteorological data. Our study focuses on the clustering of agricultural seasons for two types of crops: citrus fruits and wheat.

The first step is the specification of the period of the agricultural seasons and the geographical stations as well as the meteorological variables with a direct influence on the agricultural crops. Then, the collection of its daily data through the APIs, then the preparation, organization and visualization of the agricultural seasons for the two crops chosen. The second step is the clustering of agricultural seasons for each meteorological variable using the k-means algorithm adopted for time series and the DTW similarity measure. For the last step, it is a question of exploiting all the results of the previous clustering of each variable and adopting a statistical approach in order to find the similarities between the seasons while exploiting all the meteorological variables at the same time.

Keywords: Precision agriculture, agricultural season, agricultural cultivation, API, meteorological data, clustering, Time series Kmeans, temporal dynamic deformation.

Dédicace

A mes chers parents : Je dédie ce travail à mes chers parents qui m'ont toujours aidé et m'ont motivé à renforcer mes efforts, et qui m'ont préparé un ensemble d'amour et de conscience que je l'apprécierai toujours.

Aux personnes les plus proches de mon cœur : Myawo, Karima, Soukaina.

A mes amis : Pour leurs encouragements, leurs partages et leurs conseils.

Oumayma

Remerciements

Au terme de ce projet, j'exprime mes respects et ma gratitude à l'égard de Monsieur **LAHKIM Hamza**, mon encadrant de stage pour son accueil chaleureux et sa confiance. Je lui exprime mes remerciements de m'avoir aidé à mener à bien mon travail grâce à son suivi et son dévouement tout en me donnant de ses conseils et ses remarques pertinentes.

Ma reconnaissance et mes remerciements les plus profonds vont également à toute l'équipe **AgriEdge** qui m'a accompagné pédagogiquement et techniquement en termes de statistique et d'agronomie.

Je tiens particulièrement à témoigner ma gratitude à Monsieur **EL HAJ TIRARI Mohammed**, Professeur de l'INSEA de m'avoir honoré par son encadrement, la qualité de ses orientations et ses précieux critiques et conseils tout au long de la période de stage de fin d'étude.

J'adresse également l'expression de ma vive gratitude et remerciement à tout le personnel le corps professoral et administratif de l'INSEA et tous ceux qui ont aidé de près ou de loin à la réalisation de ce travail.

Je ne saurais oublier de remercier tous mes collègues et toute personne qui a contribué de près ou de loin dans la réussite de ce stage.

Je tiens également à remercier M. Fouad EL ABDI, pour le grand honneur qu'il nous fait en acceptant de juger ce travail. Veuillez trouver ici, professeur, l'expression de nos sincères remerciements.

Enfin, merci à tous ceux et celles qui feuilletteront ces pages.

Table des matières

Liste des abréviations

AGDD: Aggregated grow of degree day (Temperature du degree jour cumulée)

API : Application Programming Interface (Interface de programmation applicative)

APRE : Aggregated precipitation (Précipitation cumulée)

DTW : Dynamic time warping (La déformation dynamique temporelle)

PRECIPITATIONCAL : Précipitations calibrées

QV2M : Humidité spécifique à 2 mètres

T2M_MAX : Température à 2 Mètres Maximum.

T2M_MIN : Température à 2 mètres minimum

WS10M : Vitesse du vent à 10 mètres

Liste des figures

Liste des tableaux

Introduction

La gestion de l'écosystème, le contrôle des cycles biologiques des espèces domestiquées, produire de la nourriture et des ressources utiles aux sociétés sont parmi les objectifs essentiels dans l'agriculture.

Au Maroc, l'agriculture est un facteur indispensable de la croissance, puisqu'elle représente 20% du PIB, et par conséquent 40% de l'emploi total et procure des revenus directs ou indirects à 15 millions de personnes. Elle produit tous les travaux liés au milieu naturel et permet la culture et la récolte d'organismes utiles à savoir : Cultures végétales et animales.

L'amélioration de ce secteur aura la puissance de redynamiser le marché du travail et puis amender et progresser le revenu national brut.

Pour ce fait, le mélange entre l'innovation et la digitalisation est capable de transformer l'agriculture en un secteur plus inclusif et rentable, discerné optiquement au niveau du stockage, accès, manipulations, et analyse des informations.

Dans le but de maximiser les rendements, Agri-Edge, un business unit marocain qui fait partie de l'écosystème de l'Université Mohammed VI Polytechnique, est une entité spécialisée dans la transformation des données en propositions décisionnelles pour les agriculteurs en se basant sur la recherche et le développement pour promouvoir une agriculture durable.

Parmi les facteurs qui influencent de plus le rendement agricole nous citons les états du sol, les engrais, les techniques des semis, et les conditions climatiques et leurs variations au cours de l'année. Pour notre étude, nous nous intéressons à l'étude de la variation climatique au Maroc selon chaque région et chaque cycle de culture.

Sous la problématique suivante : Comment grouper, classifier et détecter les ressemblances entre les années agricoles en prenant en considération les données météorologiques et les cultures cultivées ainsi que le coté spatial ?

L'étude sera répartie en 4-quatre parties :

- La première partie repose sur le contexte d'étude, donnant un aperçu sur les facteurs météorologiques qui influencent la production Agricole.
- La deuxième concerne la spécification et la collecte des données météorologiques dans les régions sélectionnées au Maroc en utilisant les API, l'organisation de ces données et la description des données.
- La troisième est celle du Clustering des saisons agricoles selon chaque variable météorologique.
- La quatrième partie est l'exploitation de toutes les variables météorologiques pour la réalisation du clustering des saisons agricoles.

CHAPITRE I

Contexte de l’étude

CHAPITRE I : Contexte de l'étude

Ce chapitre est consacré à la présentation de l'organisme d'accueil et ses diverses activités, ainsi que la bibliographie du projet, la problématique et les objectifs.

1. Présentation de l'organisme d'accueil

Agri Edge a été créé en 2018 par le groupe OCP en collaboration avec l'Université Mohammed VI Polytechnique.

Le business unit utilise une approche basée sur la recherche et le développement pour promouvoir une agriculture durable. En transformant les données en propositions décisionnelles pour les agriculteurs en vue d'améliorer la rentabilité. Il est basé sur l'utilisation des données collectées via des capteurs, des images satellites et des images de drone, ensuite les exploitées avec des modèles algorithmiques et un suivi riche en conseil agronomiques.

Agri Edge a développé plusieurs services parmi lesquels :

N-IndeX : C'est un indice capable à estimer la quantité d'azote optimale à appliquer au le blé à ses différents stades agronomiques, afin de garantir une fertilisation raisonnée du Blé et améliorer l'utilisation d'engrais. Cette estimation se fera grâce au traitement des images satellites. L'indice permet de faire des économies très importantes en termes de quantité d'Azote (surtout avec les prix en hausse exponentielle ces dernières années), avec des rendements plus importants.

AquaEdge : C'est une application du conseil en irrigation, elle exploite la puissance de l'intelligence artificielle et le savoir agronomique pour enfin fournir la stratégie d'irrigation optimale de la ferme et permettre le suivi en temps réel de l'état hydrique des terres agricoles.

Avec trois offres :

- Efficience d'irrigation plus élevée.
- Réduction des coûts de pompage.
- Economie de l'eau.

Yield Edge: Vu que la compréhension du système alimentaire est essentielle, AgriEdge a lancé ce service qui sert à prédire les valeurs du rendement future, grâce à la combinaison des sources de données météorologiques, environnementaux, des cultures et du sol en matière d'apprentissage automatique.

Cattle Edge : un nouveau service garantissant le suivi des bovins 24h sur 7 jours. Ce dernier offre plusieurs solutions digitales tel que : suivre la chaleur des bovins, monitorer le vêlage via des alertes de détresse pré vêlage et post vêlage, recevoir des alertes sur les problèmes et maladies, protection contre le vol des bovins lorsqu'ils dépassent les limites de la firme, ainsi qu'économiser les couts de gestation.

Également, le business unit maintien maintes collaborations avec les communautés

scientifiques internationales composées d'experts reconnus en agriculture de précision, ce qui lui permet d'organiser des évènements internationaux comme les journées *Agri Analytics Days*. C'est une plateforme qui vise à explorer le Big Data afin de fournir de nouveaux outils de prise de décision efficaces et aider au développement durable de l'agriculture.

Cet évènement met à terre les meilleures conditions de rassemblement de toutes les communautés intéressées par le développement d'une agriculture durable dans le monde entier, avec un objectif d'explorer les perspectives de big data et les défis de leur mise en œuvre dans les domaines agricoles.

2. Contexte du projet

2.1 Le secteur agricole

Au Maroc, l'agriculture a toujours été un pilier important de l'économie et de la société, et sa performance affecte l'ensemble de l'économie. En fait, le dynamisme du secteur agricole exportateur a fait du Maroc l'un des premiers exportateurs agricoles mondiaux. Ainsi que, d'après le ministère de l'économie et des finances dans le rapport intitulé « Le secteur agricole marocain : Tendances structurelles, enjeux et perspectives de développement » [1], ce secteur est considéré le principal contributeur du PIB environ de 14%. Cependant, il existe une variation importante de 11% à 18% selon les conditions climatiques des territoires. Et comme le taux de croissance du pays est étroitement lié au taux de croissance de la production agricole. Donc l'amélioration de ce secteur est indispensable.

2.1.1 L'évolution des politiques et stratégies agricoles au Maroc depuis l'indépendance

Le Maroc a mis en place divers politiques et stratégues agricoles dès 1956 jusqu'à maintenant, comme présenter dans le graph ci-dessous :

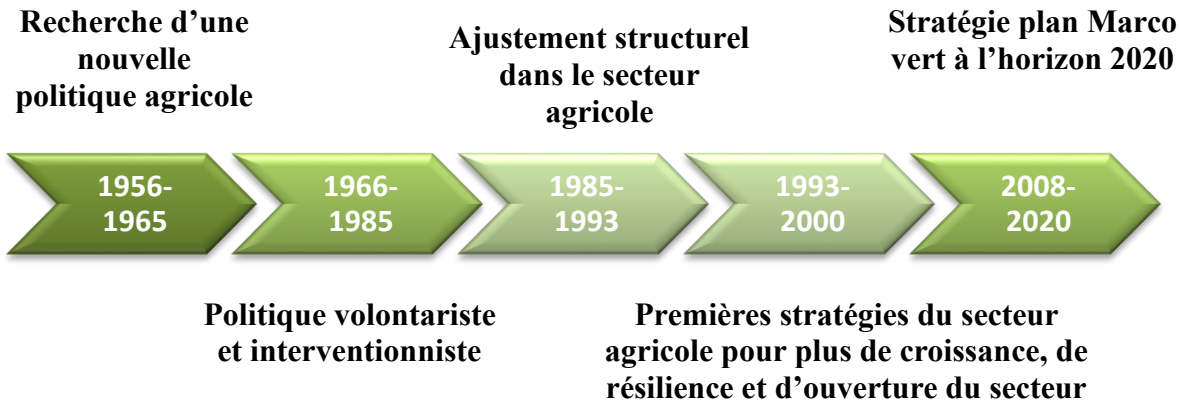


Figure 1: Chronologie des politiques et stratégies agricoles au Maroc

Source : DEPF Etudes, Le secteur agricole marocain : Tendances structurelles, enjeux et perspectives de développement, 2019.

2.1.1 L'agriculture de précision [2]

Au milieu des années 1980, l'agriculture de précision a marqué sa première apparition aux États-Unis, en combinant diverses technologies : géomatique, informatique, électronique et agronomie, afin d'apporter aux agriculteurs des solutions qui améliorent les performances économiques et environnementales de la production.

2.1.1.1 Les objectifs d'agriculture de précision

L'agriculture de précision facilite le travail grâce à une gestion de la consommation des ressources : énergie, eau, intrants, finance…

Dans le but de :

- Améliorer le confort de travail.
- Amender les compétences agronomiques.
- Etablir une compréhension efficace de la relation entre les paires sol-culture et les conditions météorologiques.
- Elaborer une meilleure gestion des cultures.
- Répartition plus intelligente des économies sur des intrants plus précieux.
- Garantir une meilleure utilisation des intrants, réduisant ainsi l'impact environnemental.
- Etude de la variabilité des saisons agricoles.
- Maintenir des stratégies et développer des prévisions pour les rendements agricoles prochains.

2.1.1.2 Les outils d'agriculture de précision

Dans le but de concrétiser l'agriculture de précision, il faut suivre le cheminement des interventions sur le végétal qui dit « J'observe puis je décide et enfin j'agis ». Le schéma ci-dessous représente les outils d'observation et d'analyse chez l'agriculture de précision :

Les outils d'observation

- La météo connectée : Les stations météorologiques intelligents envoient des informations sur des paramètres de pluviométrie, de vitesse du vent, de température de l'air…
- Les objets connectés : caméras, capteurs et machines de terrain aux réglages proactifs.
- Les images satellitaires et celles prises par des drones.

Outils d'analyse et d'aide à la décision

- Les données de la météo- afin d'alimenter des modèles informatiques qui qui préviennent des risques de maladies ou les risques météo.
- Les algorithmes de reconnaissance d'images et vision par ordinateur permettent d'identifier l'état et les besoins des champs en temps réel.

2.2 Littératures sur le sujet

2.2.1 Les saisons agricoles

La saison agricole est une période de l'année délimitée par deux phases celles des semis et des récoltes, au long de cette période les cultures agricoles confrontent une succession des évènements et des conditions climatiques. Chaque culture est caractérisée par sa propre saison agricole qui débutera et finira dans des dates spécifiques ; cependant pour plusieurs cultures les dates de début et de fin des saisons agricoles commenceront à partir du mois Octobre jusqu'à Juillet.

Généralement les caractéristiques des saisons agricoles dépendent des variations climatiques [2], tel que le pourcentage des précipitations, les degrés de température, l'humidité ainsi que la vitesse de vent… ainsi que les coordonnées géographiques de la zone agricole et la culture étudiée.

2.2.2 Problématique

Les données agricoles sont très diverses et non structurées. Afin de pouvoir expliquer le rendement par exemple, la distribution n'est souvent pas normale vu que l'échantillon dépend fortement de la météo. En fonction des spécificités des conditions météorologiques, l'échantillonnage peut générer deux à trois différentes distributions différentes du rendement, avec une moyenne, écart type et parfois type de distribution différentes (voir figure 2). Ainsi, la modélisation devient de plus en plus difficile, surtout pour faire des modèles linéaires pour lesquels la normalité est une condition nécessaire.

Sachant que la variabilité de la distribution du rendement est souvent associée aux conditions météorologiques (et bien sûr la position géographique), pouvoir clustériser les saisons agricoles en fonction des paramètres météo serait un atout pour améliorer la modélisation. L'approche de clustering qu'on vise développer sera intégrée dans le pipeline de préparation des données pour plusieurs modèles dans l'entité d'accueil.

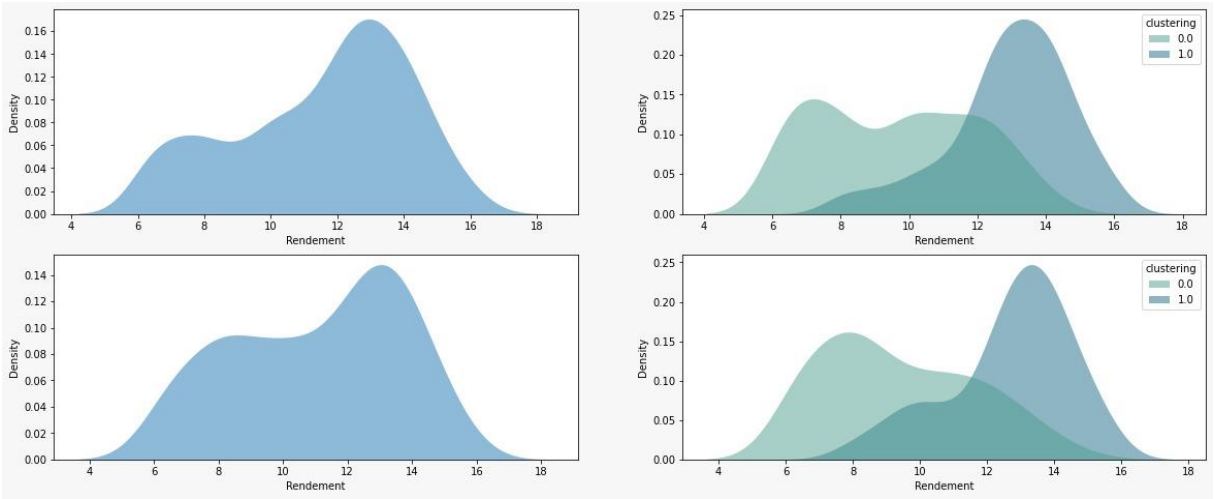


Figure 2: Transformation de la distribution à l'aide du clustering

Sortie python

La question qui se pose donc est comment classifier et détecter les ressemblances entre les années agricoles en prenant en considération les données météorologiques et les cultures cultivées ?

CHAPITRE II

Collecte, prétraitement et description des données

CHAPITRE II : Collecte, prétraitement et description des données

Ce chapitre est dédié au choix des variables bénéfiques à l'étude, ensuite à la collecte de donnée à partir des API, et enfin la préparation et la description de ces données.

1. Spécification des variables météorologiques

Afin d'étudier les variations des climats dans le secteur agricole, il faut sélectionner les paramètres météorologiques qui ont une influence directe sur notre culture agricole. Ainsi, les variables choisies sont les suivantes :

Les précipitations : sont les météores qui tombent dans une atmosphère, et qui peuvent tomber sous trois formes en fonction de la température de l'air liquide, pluie verglaçante, bruine verglaçante, et solide.

La température : est le degré de la chaleur ou de froid de l'atmosphère en un lieu.

L'humidité : est la présence d'eau ou de vapeur d'eau dans l'air

La vitesse des vents.

2. Spécification de la démarche à suivre

Toute culture agricole est caractérisée par sa propre saison agricole, donc il faut spécifier les cultures que nous allons étudier et puis sélectionner les villes pour lesquelles les données de météo seront collectées. Pour ce faire, nous avons choisis les cultures et déterminer les villes et régions dans lesquelles ces cultures sont produites le plus.

Dans le tableau ci-dessous, nous avons présenté quelques exemples de cultures agricoles avec les stations (villes) où on les produit le plus.

Culture choisie	Villes - Stations de météo	Début- Fin de la saison agricole
Blé	Casablanca	Octobre - Juin
Maïs	Tadla	Mai - Novembre
Les betteraves à sucre	Kénitra	Fin juillet - fin mars
Agrumes (Cas : Oranges)	Berkane	Toute l'année
Abricots	Marrakech	Février - Mai
Pomme de terre	Moulouya	Février - Juin
Olives	Meknès	Avril - Mi-Novembre

Tableau 1 : Culture agricoles et villes de production (dates début-fin de la saison agricole)

3. Collecte de données à partir des API

3.1 Les API

L'interface de programmation d'applications (API) qui est un ensemble de définitions et de protocoles, sert à faciliter la création ainsi que l'intégration de logiciels d'applications.

Les API permettent de communiquer avec d'autres produits et services sans savoir les détails de leur mise en œuvre.

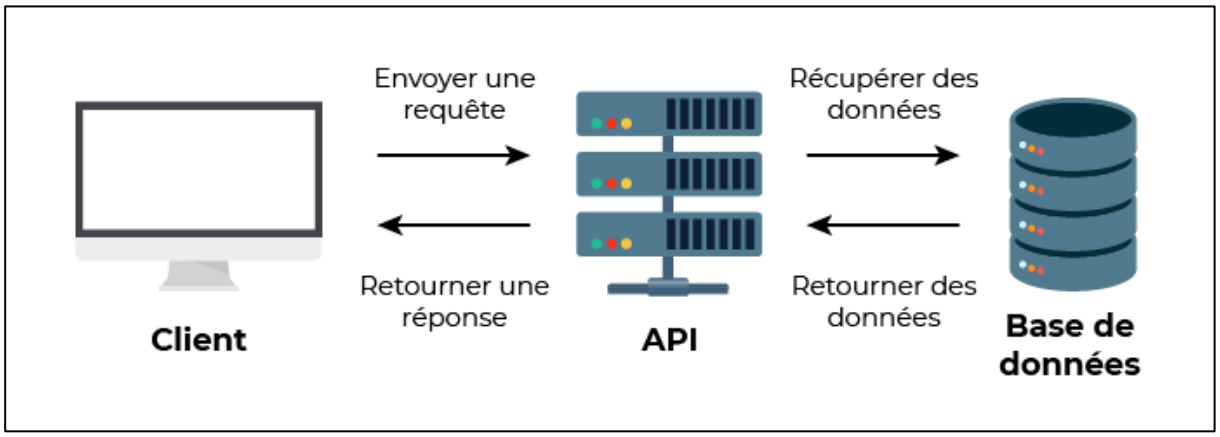


Figure 3 : Processus des API

Source : Adoptez les API REST pour vos projets web - OpenClassrooms

Une API est capable de renforcer les relations entre les entreprises et les clients. D'une part, pour les entreprises, c'est un moyen pratique de promouvoir leur propre entreprise auprès de leurs clients tout en assurant la sécurité de leurs systèmes backend. Et d'autre part, pour les clients, les API fournissent les moyens d'accéder aux données qui peuvent être utilisées pour alimenter leur recherche ou le développement de produits.

3.2 Le fonctionnement des API

Les API fonctionnent à travers quelques opérations et interactions [2], appelées « Méthodes ». Celles-ci peuvent être de plusieurs natures :

Lecture seule : sert à la récupération d'informations ; afin de les traiter côté utilisateur.

Lecture et écriture : permet la récupération de données afin de les traiter, ainsi que la modification, l'ajout ou la suppression d'informations.

Ecriture seule : Ce type de méthode permet l'ajout d'informations, sans possibilité d'obtenir celles déjà présentes.

Le travail avec les API nécessite une requête pour l'accès à l'information, nous présentons deux types de requêtes :

La requête REST : Sert à appeler une URL, avec différents paramètres pour affiner la demande. Elle est la plus utilisée, avec près de 75% des APIs utilisant ce mode de requête.

La requête SOAP : Cette requête demande un format bien précis, et nécessite un envoi plus conséquent que la méthode REST. Ce système est moins utilisable.

Apres la spécification de la requête il faut sélectionner le type de réponse. Pour cela nous avons plusieurs types parmi lesquels, on peut citer :

XML : Le format XML permet de véhiculer un nombre d'informations énorme.

JSON : Le format JSON est un format en pleine expansion, il permet le transfert d'informations dans un format facilement utilisable dans la majorité des langages de programmation.

CSV : Le format avec la plus simple forme avec une organisation facile.

3.3 La collecte des données journalières

Pour la collecte de nos données, nous utilisons les API du site Nasa Power[1], ce dernier permet la collecte des données météorologiques sous plusieurs mesures temporelles tel que : des données journaliers, mensuelles et annuelles ; et sur toute zone géographique spécifiée.

Pour notre projet, nous avons spécifié la collecte des données à partir de 1981 jusqu'à aujourd'hui, et ceci pour les variables suivantes :

T2M_MAX : Temperature at 2 Meters Maximum (C).
T2M_MIN : Temperature at 2 Meters Minimum (C).
PRECIPITATIONCAL : precipitation Calibrated (mm/day).
WS10M : Wind Speed at 10 Meters
QV2M : Specific Humidity at 2 Meters

Nous allons télécharger les données journalières pour deux stations géographiques. Nous avons donc spécifié les coordonnées de longitude et latitude des villes sélectionnées. Et ensuite, nous avons collecté ces données pour toutes les variables citées ci-dessus.

4. Organisation et présentation des données

4.1 Pré-traitement et organisation des données

4 .1.1 Cas de la culture des agrumes

4.1.1.1 Station sélectionnée

Pour la culture des agrumes, nous avons choisis la région Oriental vu qu'elle est considérée la zone principale de production des agrumes. Pour la station de mesure nous avons exploité les données de la ville Berkane.

4.1.1.2 La saison agricole des agrumes

Vu que les arbres d'agrumes existent toute l'année tout en interagissant avec les conditions météorologiques durant toute l'année, nous avons traité les ressemblances entre les

[1] Le projet power fournit des ensembles de données solaires et météorologiques issues de la recherche de la NASA pour soutenir les énergies renouvelables, l'efficacité énergétique des bâtiments et les besoins agricoles.

saisons agricoles pour chaque année de janvier jusqu'à décembre.

4.1.1.3 Organisation des saisons agricoles

Suite à la collecte de données, comme observé sur le tableau 2 ci-dessous, nous avons obtenu une suite d'année dès 1981 jusqu'à 2021 :

Date	Année	T2M_MAX	T2M_MIN	PRECTOTCORR	QV2M	WS10M
1981-01-01	1981	15.80	4.40	0.02	4.88	3.84
1981-01-02	1981	15.63	4.37	0.00	4.70	2.78
1981-01-03	1981	16.55	4.08	0.00	4.88	2.70
1981-01-04	1981	16.73	5.01	0.00	5.37	4.91
1981-01-05	1981	16.09	4.49	0.47	6.04	4.12
...	...	...	...	...	...	...
2021-12-27	2021	21.62	12.52	0.02	9.28	5.18
2021-12-28	2021	20.51	11.30	0.02	9.22	4.40
2021-12-29	2021	23.68	11.62	0.00	7.39	3.04
2021-12-30	2021	25.40	11.49	0.02	5.49	2.83
2021-12-31	2021	22.26	10.96	0.02	5.68	2.38

Tableau 1: Données collectées pour la station Berkane

Sortie python

Ainsi, l'étape suivante consiste à créer une nouvelle colonne pour la désignation du jour et mois qui est nécessaire pour l'étape du clustering.

Date	Année	T2M_MAX	T2M_MIN	PRECTOTCORR	QV2M	WS10M	Jour-Mois
1981-01-01	1981	15.80	4.40	0.02	4.88	3.84	01-01
1981-01-02	1981	15.63	4.37	0.00	4.70	2.78	02-01
1981-01-03	1981	16.55	4.08	0.00	4.88	2.70	03-01
1981-01-04	1981	16.73	5.01	0.00	5.37	4.91	04-01
1981-01-05	1981	16.09	4.49	0.47	6.04	4.12	05-01
...	...	...	...	...	...	...	...
2021-12-27	2021	21.62	12.52	0.02	9.28	5.18	27-12
2021-12-28	2021	20.51	11.30	0.02	9.22	4.40	28-12
2021-12-29	2021	23.68	11.62	0.00	7.39	3.04	29-12
2021-12-30	2021	25.40	11.49	0.02	5.49	2.83	30-12
2021-12-31	2021	22.26	10.96	0.02	5.68	2.38	31-12

Tableau 2: la désignation du jour et mois

Sortie python

4.1.1.4 Transformation des variables : température et précipitation

Les cultures agricoles demandent un cumul de température, et généralement cela se réalise grâce au degré jour (Growing degree day GDD) qui est une mesure empirique utilisée pour calculer l'accumulation de chaleur, et estimer la durée du développement biologique telle que la croissance des plantes.

Le degré jour se calcule par la formule suivante :

$$\mathbf{GDD} = \frac{\mathbf{T_{max}} + \mathbf{T_{min}}}{\mathbf{2}} - \mathbf{T_{base}}$$

Avec :

$\mathbf{T}_{base}$ est la température de base : La température en dessous de laquelle signifie que la croissance des plantes est de zéro. Notons que chaque culture se caractérise par sa propre Tbase (par exemple Blé Tbase=6))

Pour notre travail, l'étude sera réalisée avec le cumul du degré jour (AGDD), et aussi le cumul de la précipitation pour améliorer la comparaison statistique entre les saisons agricoles.

Date	Année	AGDD	APRE	QV2M	WS10M	Jour-Mois
1981-01-01	1981	0.000	0.02	4.88	3.84	01-01
1981-01-02	1981	0.000	0.02	4.70	2.78	02-01
1981-01-03	1981	0.000	0.02	4.88	2.70	03-01
1981-01-04	1981	0.000	0.02	5.37	4.91	04-01
1981-01-05	1981	0.000	0.49	6.04	4.12	05-01
...	...	...	...	...	...	...
2021-12-27	2021	2426.125	442.34	9.28	5.18	27-12
2021-12-28	2021	2429.030	442.36	9.22	4.40	28-12
2021-12-29	2021	2433.680	442.36	7.39	3.04	29-12
2021-12-30	2021	2439.125	442.38	5.49	2.83	30-12
2021-12-31	2021	2442.735	442.40	5.68	2.38	31-12

Tableau 3: Création des variables AGDD et APRE

Sortie python

4.1.1.5 Pivoter le tableau en mettant les jours en colonnes et les années en lignes pour la variable température moyenne

Dans le but de suivre les années agricoles et pouvoir présenter les mesures des variables météorologiques pour chaque année spécifique, nous avons organisé des tableaux pour chaque variable de météo. A l'aide du pivotement, et en précisant dans l'index l'année, et en colonnes les valeurs de la colonne « Jour-Mois », et en remplissant nos tableaux par les mesures de chaque variable de météo spécifique.

Nous avons réalisé le même format de pivotement pour toutes les autres variables.

Jour-Mois	01-01	01-02	01-03	01-04	01-05	01-06	01-07	01-08	01-09	01-10	...	30-10	30-11	30-12	31-01	31-03	31-05	31-07	31-08	31-10	31-12
Année																					
1981	0.000	1.215	13.545	110.020	189.350	365.325	658.470	995.875	1381.790	1702.870	...	1947.700	2089.060	2136.790	1.215	108.825	357.640	984.170	1370.045	1953.550	2139.560
1982	0.000	22.360	36.500	95.265	188.430	373.405	691.255	1100.730	1497.880	1809.580	...	1959.295	2028.660	2029.900	22.360	93.870	366.940	1090.650	1487.395	1962.250	2029.900
1983	0.000	8.150	27.470	95.945	202.005	379.735	706.600	1124.300	1500.780	1872.475	...	2134.060	2288.145	2312.260	5.960	94.750	373.285	1110.335	1490.490	2140.370	2312.260
1984	0.145	5.715	12.990	35.765	159.485	272.655	522.070	952.680	1336.365	1679.965	...	1844.525	1950.615	1977.370	5.210	34.625	265.605	938.595	1324.270	1849.810	1977.370
1985	0.000	6.625	75.235	101.115	225.205	379.525	694.085	1119.225	1549.825	1913.615	...	2156.950	2284.645	2309.655	6.460	95.000	372.210	1107.475	1535.655	2164.625	2309.655

Tableau 4 : La base de données contenant l'AGDD en format lignes pour chaque saison

Sortie python

Nous avons remarqué que le 29/02 est vide pour toutes les années bissextiles. Pour cette raison nous avons supprimé ce jour de la base de données. Et ceci, après réalisation du cumul pour nos variables (AGDD et APRE) dans le but de ne pas affecter nos cumules finaux.

4.1.2 Cas de la culture du blé

4.1.2.1 Station sélectionnée (cas blé)

Pour la culture du blé nous avons choisi la région Casablanca-Settat vu qu'elle est considérée parmi les zones principales de production des céréales (18%), et pour la station nous allons exploiter les données de la ville Casablanca.

4.1.2.2 La saison agricole du blé

Le cycle de vie du blé commence à partir du mois d'octobre et termine vers la fin du mois juin. [3]

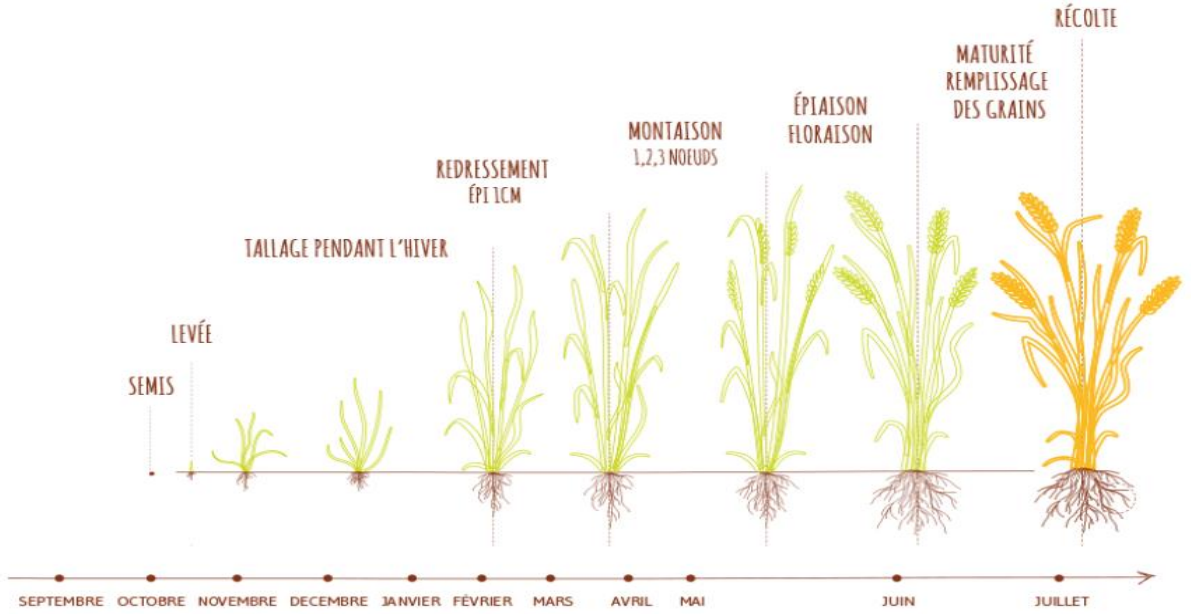


Figure 4: Le cycle de croissance du blé

Source : Le blé : quand est-il semé, cycle, blé tendre et blé dur – VIVESCIA

4.1.1.3 Organisation des saisons agricoles (cas blé)

La collecte des données se présentent comme suit :

Date	Année	DOY	T2M_MAX	T2M_MIN	PRECTOTCORR	QV2M	WS10M
1981-01-01	1981	1	17.35	6.94	0.00	5.37	4.63
1981-01-02	1981	2	16.41	7.51	0.00	5.55	2.73
1981-01-03	1981	3	15.64	7.35	0.00	6.23	2.33
1981-01-04	1981	4	15.99	8.46	0.00	6.16	1.66
1981-01-05	1981	5	15.60	6.87	0.12	6.10	2.78
...	...	...	...	...	...	...	...
2021-12-27	2021	361	21.40	13.94	0.08	10.74	2.56
2021-12-28	2021	362	21.58	13.41	0.06	10.44	3.92
2021-12-29	2021	363	22.42	12.16	0.02	9.22	4.04
2021-12-30	2021	364	26.50	13.14	0.00	7.26	3.20
2021-12-31	2021	365	25.55	14.19	0.00	7.08	3.55

Tableau 5: Données collectées pour la station Casablanca (cas blé)

Sortie Python

L'étape suivante consiste à filtrer uniquement la phase de la saison agricole blé, et à créer une colonne qui désigne la saison agricole de chaque compagnie, qui se débute à partir du premier octobre et finira jusqu'à la fin du mois juillet de l'année suivante, et une autre colonne

pour la désignation, du jour et mois.

Date	T2M_MAX	T2M_MIN	PRECTOTCORR	QV2M	WS10M	Saison agricole	Jour-Mois
1981-10-01	26.27	17.76	0.22	12.08	3.08	1981-1982	01-10
1981-10-02	25.27	17.73	1.04	10.99	3.68	1981-1982	02-10
1981-10-03	26.15	15.08	0.01	9.77	3.16	1981-1982	03-10
1981-10-04	26.07	15.93	0.07	10.93	2.96	1981-1982	04-10
1981-10-05	27.90	17.87	0.13	11.29	5.68	1981-1982	05-10
...	...	...	...	...	...	...	...
2021-12-27	21.40	13.94	0.08	10.74	2.56	2021-2022	27-12
2021-12-28	21.58	13.41	0.06	10.44	3.92	2021-2022	28-12
2021-12-29	22.42	12.16	0.02	9.22	4.04	2021-2022	29-12
2021-12-30	26.50	13.14	0.00	7.26	3.20	2021-2022	30-12
2021-12-31	25.55	14.19	0.00	7.08	3.55	2021-2022	31-12

Tableau 6: Création de la colonne saison agricole et la désignation du jour et mois (cas blé)

Sortie Python

4.1.1.4 Transformation des variables : température et précipitation (cas blé)

En suivant la même démarche utilisée pour les agrumes, nous avons calculé nos variables AGDD et APRE.

Saison agricole	Date	AGDD	APRE	QV2M	WS10M	Jour-Mois
1981-1982	1981-10-01	16.015	0.22	12.08	3.08	01-10
1981-1982	1981-10-02	31.515	1.26	10.99	3.68	02-10
1981-1982	1981-10-03	46.130	1.27	9.77	3.16	03-10
1981-1982	1981-10-04	61.130	1.34	10.93	2.96	04-10
1981-1982	1981-10-05	78.015	1.47	11.29	5.68	05-10
...	...	...	...	...	...	...
2021-2022	2021-12-27	1089.335	85.74	10.74	2.56	27-12
2021-2022	2021-12-28	1100.830	85.80	10.44	3.92	28-12
2021-2022	2021-12-29	1112.120	85.82	9.22	4.04	29-12
2021-2022	2021-12-30	1125.940	85.82	7.26	3.20	30-12
2021-2022	2021-12-31	1139.810	85.82	7.08	3.55	31-12

Tableau 7: Création des variables AGDD et APRE (cas blé)

Sortie Python

4.1.1.5 Pivoter le tableau en mettant les jours en colonnes et les saisons agricole en lignes pour la variable AGDD

Dans le but de suivre les saisons agricoles au niveau de la culture blé et pouvoir présenter les mesures des variables météorologiques pour chaque saison agricole spécifique, nous avons organisé des tableaux pour chaque variable de météo comme établit dans le cas des agrumes. A l'aide du pivotement, et en précisant dans l'index la saison agricole, et en colonnes les valeurs de la colonne « Jour-Mois », débutant dans le cas du blé à partir du mois Octobre et finira en fin juin de l'année suivante ; et en remplissant nos tableaux par les mesures de chaque variable de météo spécifique.

Jour-Mois	01-10	02-10	03-10	04-10	05-10	...	26-06	27-06	28-06	29-06	30-06
Saison agricole											
1981-1982	16.015	31.515	46.130	61.130	78.015	...	3193.775	3209.370	3226.625	3246.275	3268.565
1982-1983	16.745	34.125	51.390	68.375	84.710	...	2956.540	2971.775	2987.275	3003.265	3018.815
1983-1984	16.585	34.335	53.185	71.845	92.545	...	3072.515	3088.825	3104.495	3120.070	3135.080
1984-1985	15.810	31.210	49.060	65.900	79.365	...	2971.225	2989.575	3007.200	3024.980	3043.010
1985-1986	21.910	40.550	58.510	75.395	91.890	...	2982.320	2998.750	3015.225	3030.595	3046.280

Tableau 8 : La base de données contenant l'AGDD en format lignes pour chaque saison (cas blé)

Sortie python

4.2 Visualisation des données

Le but de notre projet est de comparer les ressemblances entre les saisons agricoles. Pour cela, nous allons présenter les séries de données pour chaque variable météorologique.

4.2.1 Cas des Agrumes

4.2.1.1 Pour la variable AGDD (Cas agrumes)

Dans notre visualisation, nous allons afficher la tendance des séries selon chaque année :

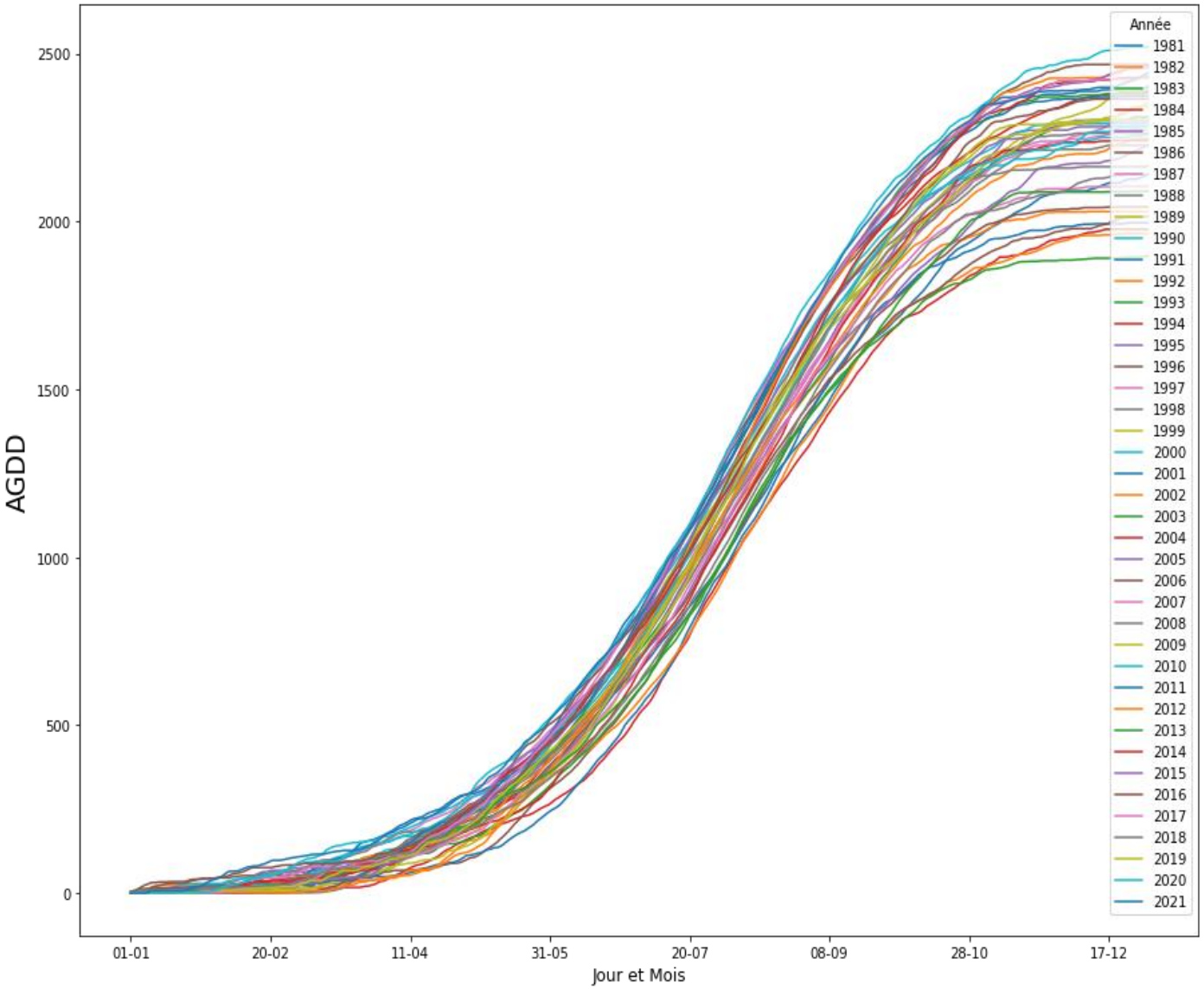


Figure 5: Présentation de toutes les saisons agricoles selon AGDD (cas agrumes)

Sortie python

La variabilité de l'AGDD a une tendance d'être minimale en mois janvier et février dans la plupart des années, mais dès le mois avril les séries augmentent positivement.

Durant toute l'année, les séries d'AGDD diffèrent d'une saison à l'autre, par exemple :

- Des séries avec un retard dans l'accroissement au début de l'année, ceci reflète la quantité de l'AGDD faible. (Année 2018)
- Des séries avec des cumuls de température GDD différentes.

Avec une première vision des courbes de ces années, nous avons remarqué qu'il y a des similarités et ressemblances au niveau de plusieurs années en se basant sur l'allure des courbes (voir figure ci-dessous)

Par exemple :

Le groupe d'année : 2006, 2013, 2021

Le groupe d'année : 1981, 2018

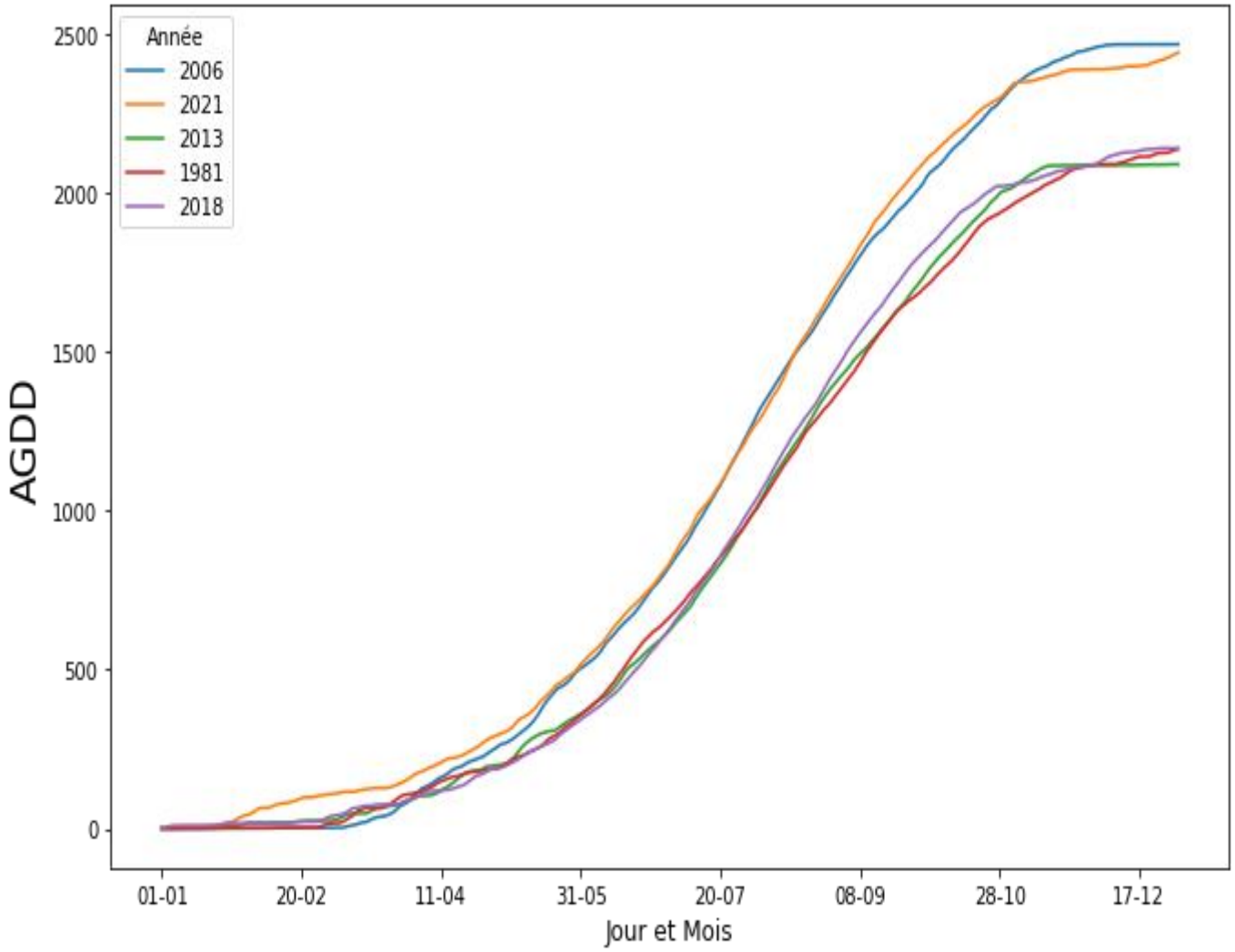


Figure 6: Ressemblances remarquables entre quelques années agricoles selon AGDD (cas agrumes)

Sortie Python

Malgré que nous ayons pu sélectionner les années qui se ressemblent par rapport à la température AGDD visuellement, il faut prouver ces résultats statistiquement.

4.2.1.2 Pour la variable APRE (cas agrumes)

Dans notre visualisation nous avons affiché la tendance de APRE selon chaque année, comme montré dans la figure ci-dessous :

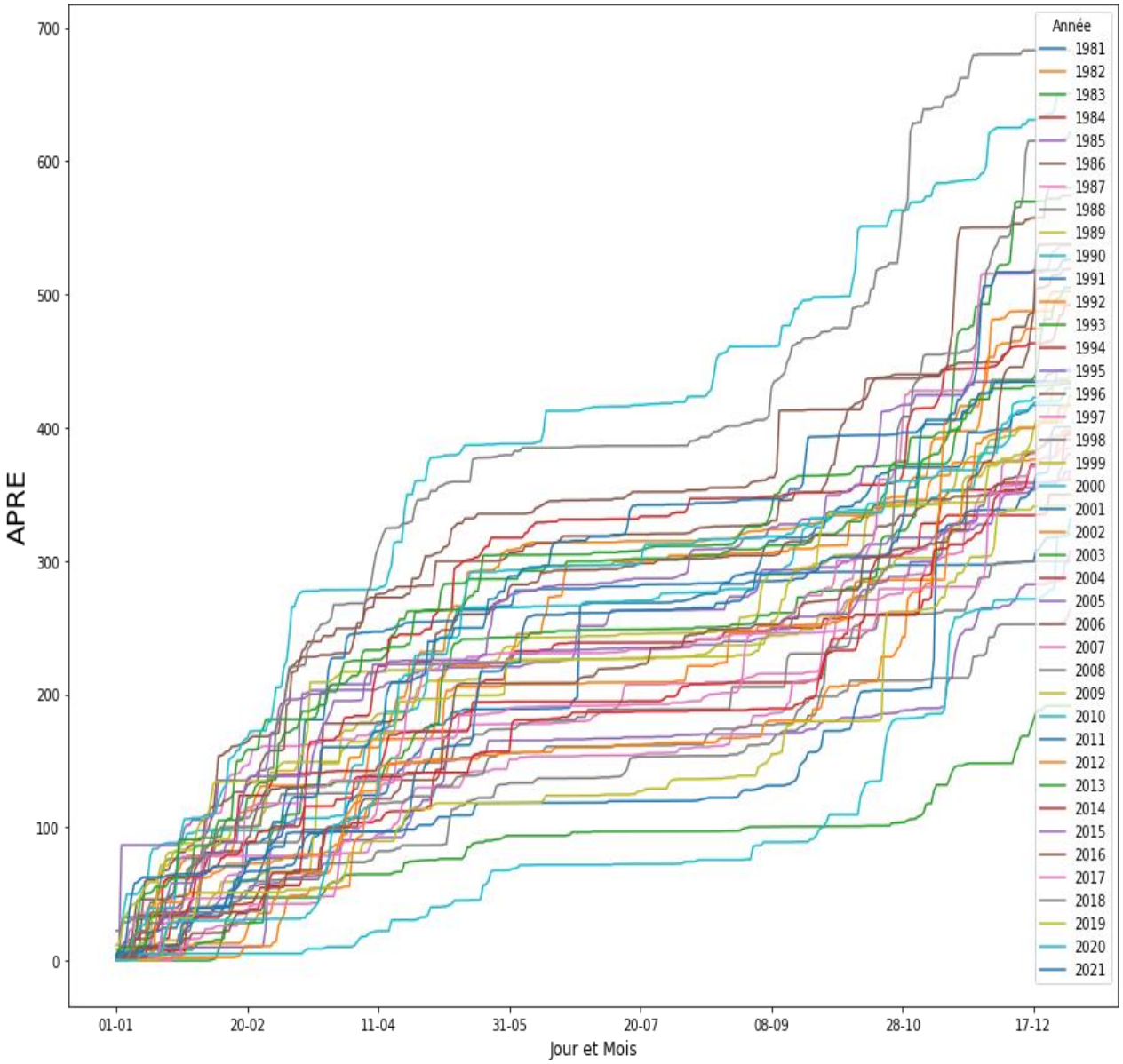


Figure 7: Présentation de toutes les saisons agricoles selon APRE (cas agrumes)

Sortie Python

La APRE est une précipitation cumulée, à partir du premier janvier les séries croient jusqu'à 400mm puis elle stagne à partir du mois 5 jusqu'à le mois 9 pour récupérer avec une évolution mesurée par 300mm.

Durant toute l'année, les séries d'APRE diffèrent d'une saison à l'autre, par exemple :

- Des séries avec un retard dans l'accroissement au début de l'année, ceci reflète la quantité des précipitations faible. (Année 2021)

- Des séries avec des périodes de stagnation très longue, qui s'étalent jusqu'à le début du mois 12. (Année 2013)

- Des séries commencent et continuent avec un bon cumul de précipitation toute

l'année.

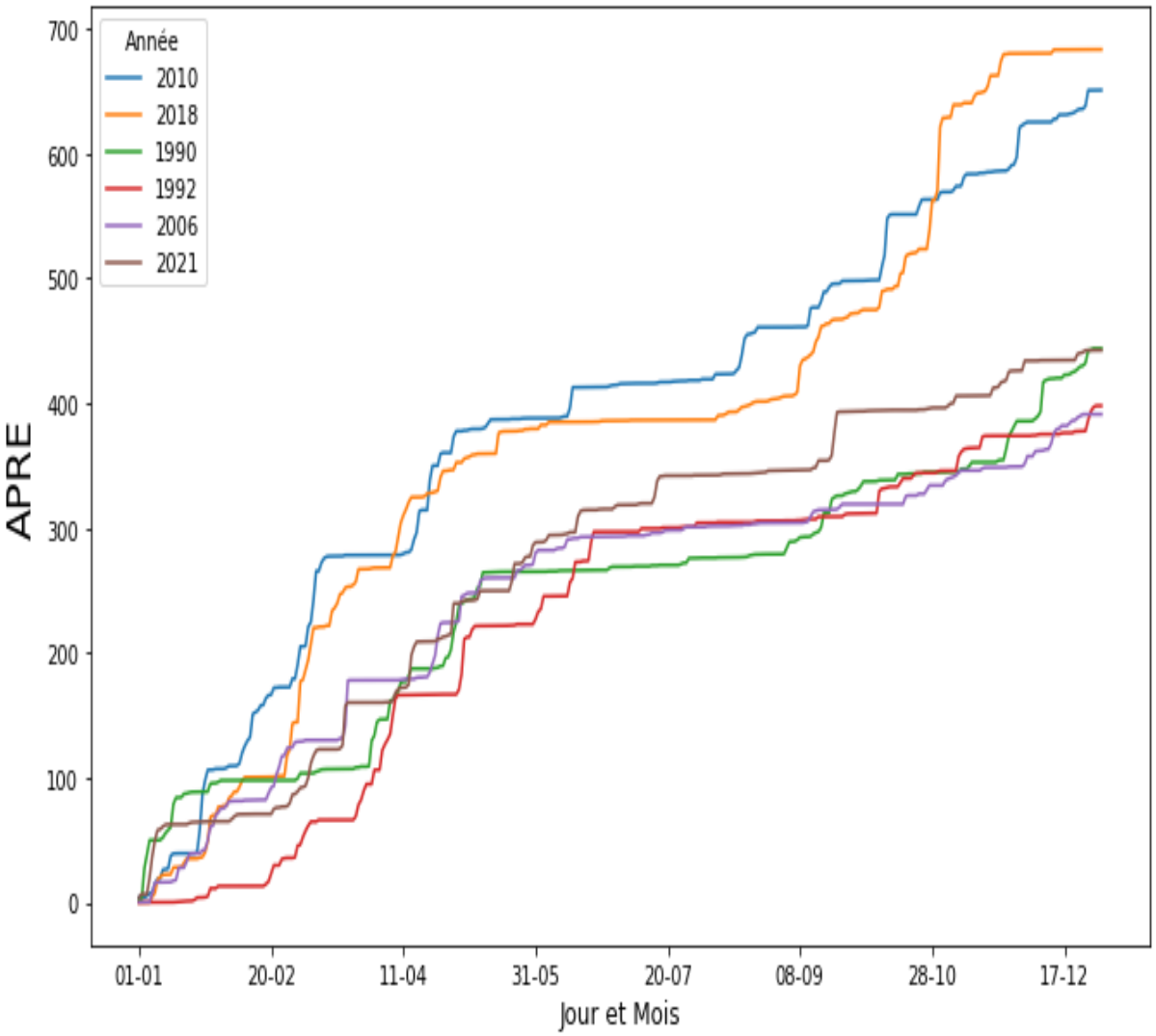


Figure 8: Ressemblances remarquables entre quelques années agricoles selon APRE (cas agrumes)

Sortie Python

Selon une première vision de ces distributions nous remarquons qu'il y a une similarité relative dans l'évolution au niveau de années :

Le groupe d'années : 2010, 2018

Le groupe d'années : 1990, 1992, 2006, 2021

4.2.1.3 Pour la variable QV2M (cas agrumes)

Dans notre visualisation, nous allons afficher la tendance selon chaque année :

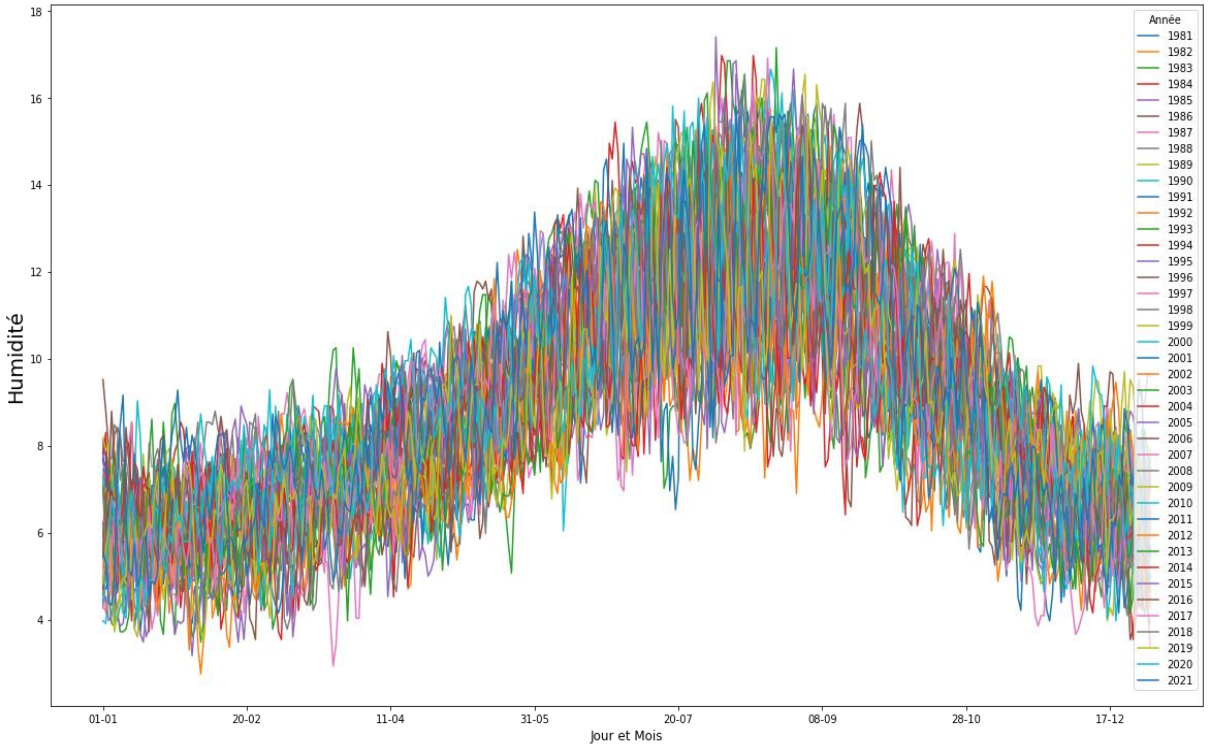


Figure 9: Présentation de toutes les saisons agricoles selon Humidité (cas agrumes)

Sortie Python

La variabilité de l'humidité a une tendance d'être maximale entre le mois 5 et le mois 10 dans la plupart des années, il est difficile de détecter les ressemblances à cause du grand nombre de fluctuations existants, et même si nous diminuons les années observées, ne nous pouvons pas décider les groupes de similitudes, comme il est montré dans la figure suivante :

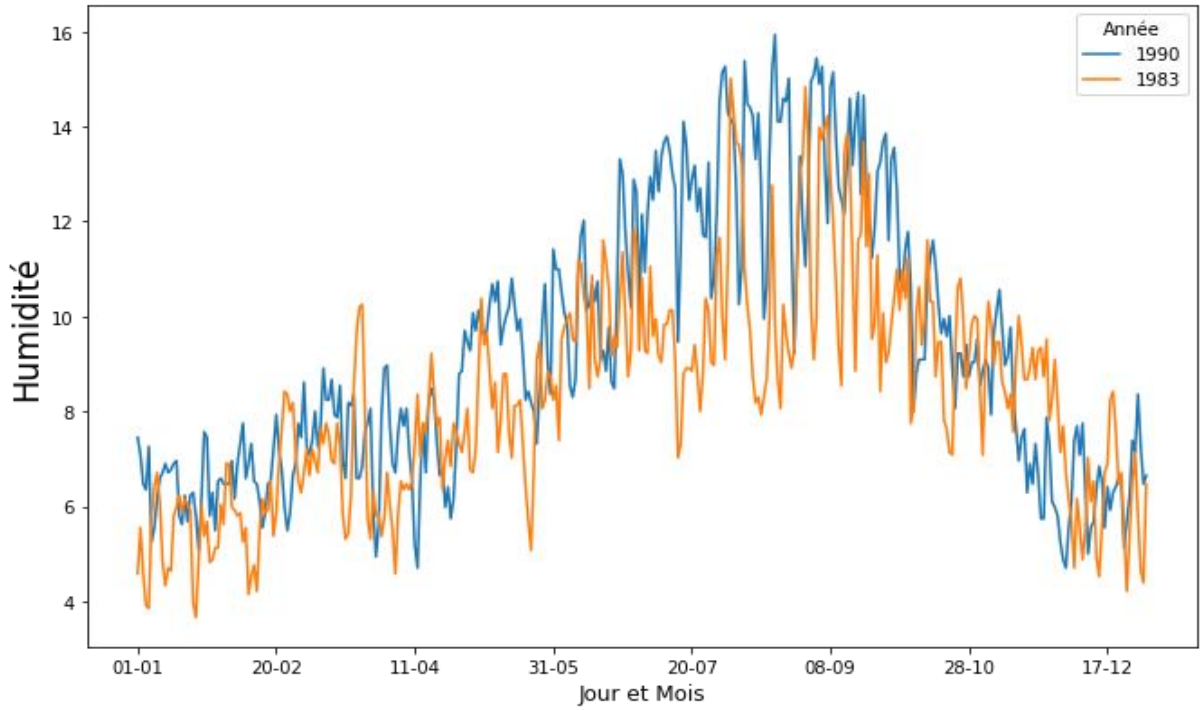


Figure 10: Ressemblances entre deux années agricoles selon Humidité (cas agrumes)

Sortie Python

4.2.1.4 Pour la variable vitesse de vents

Dans notre visualisation, nous avons affiché la tendance selon chaque année :

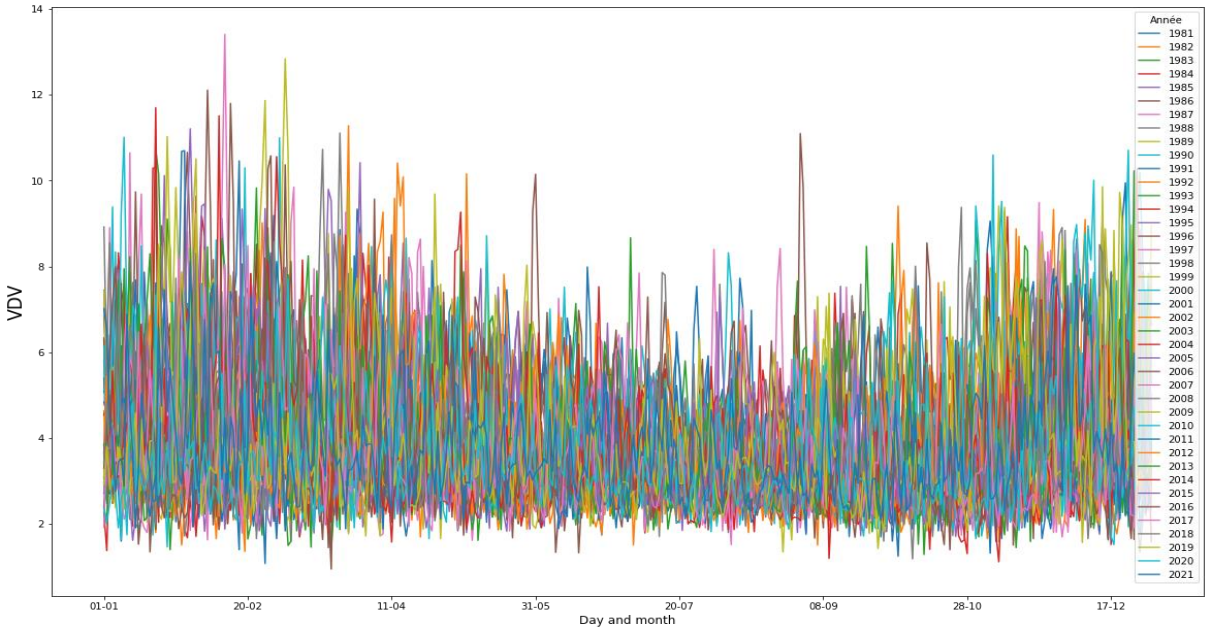


Figure 11: Présentation de toutes les saisons agricoles selon vitesse de vents (Cas agrumes)

Sortie Python

La variabilité des séries de VDV est caractérisée par beaucoup de fluctuation et même si on diminue les années observées, ne nous pouvons pas décider les groupes, comme montrer dans la figure suivante :

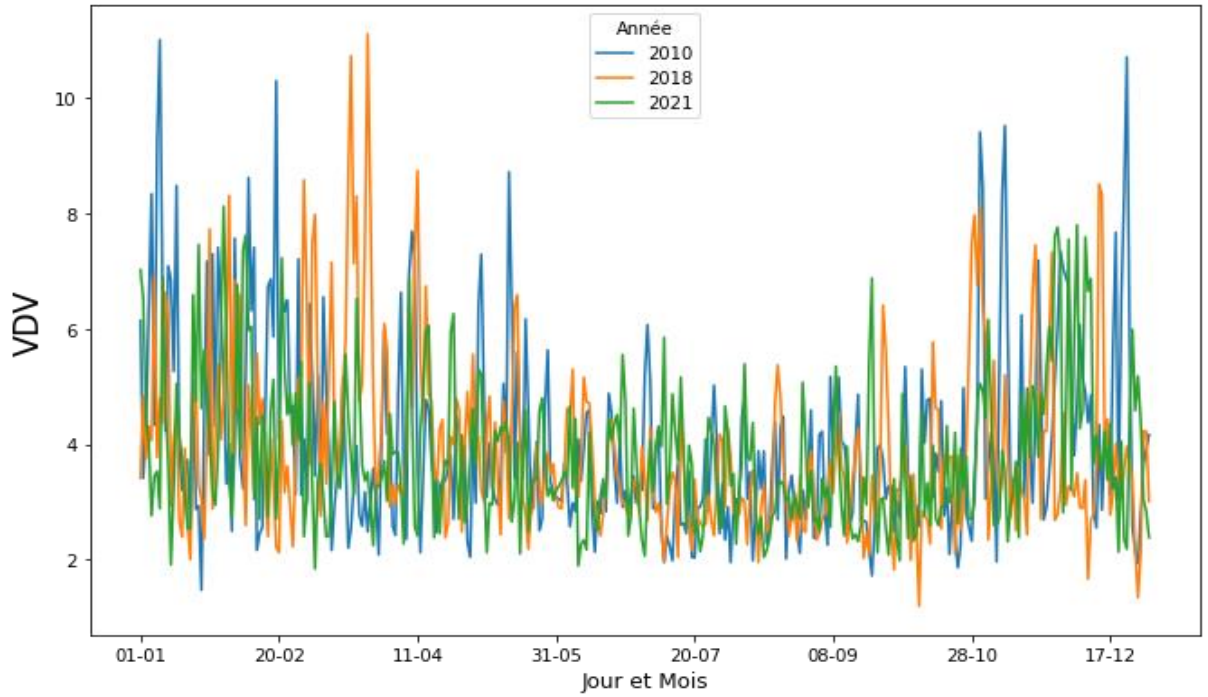


Figure 12: Ressemblances entre deux années agricoles selon la vitesse de vent (cas agrumes)

Sortie: Python

4.2.2 Cas du blé

4.2.2.1 Pour la variable AGDD

Dans notre visualisation nous allons afficher la tendance selon chaque saison agricole :

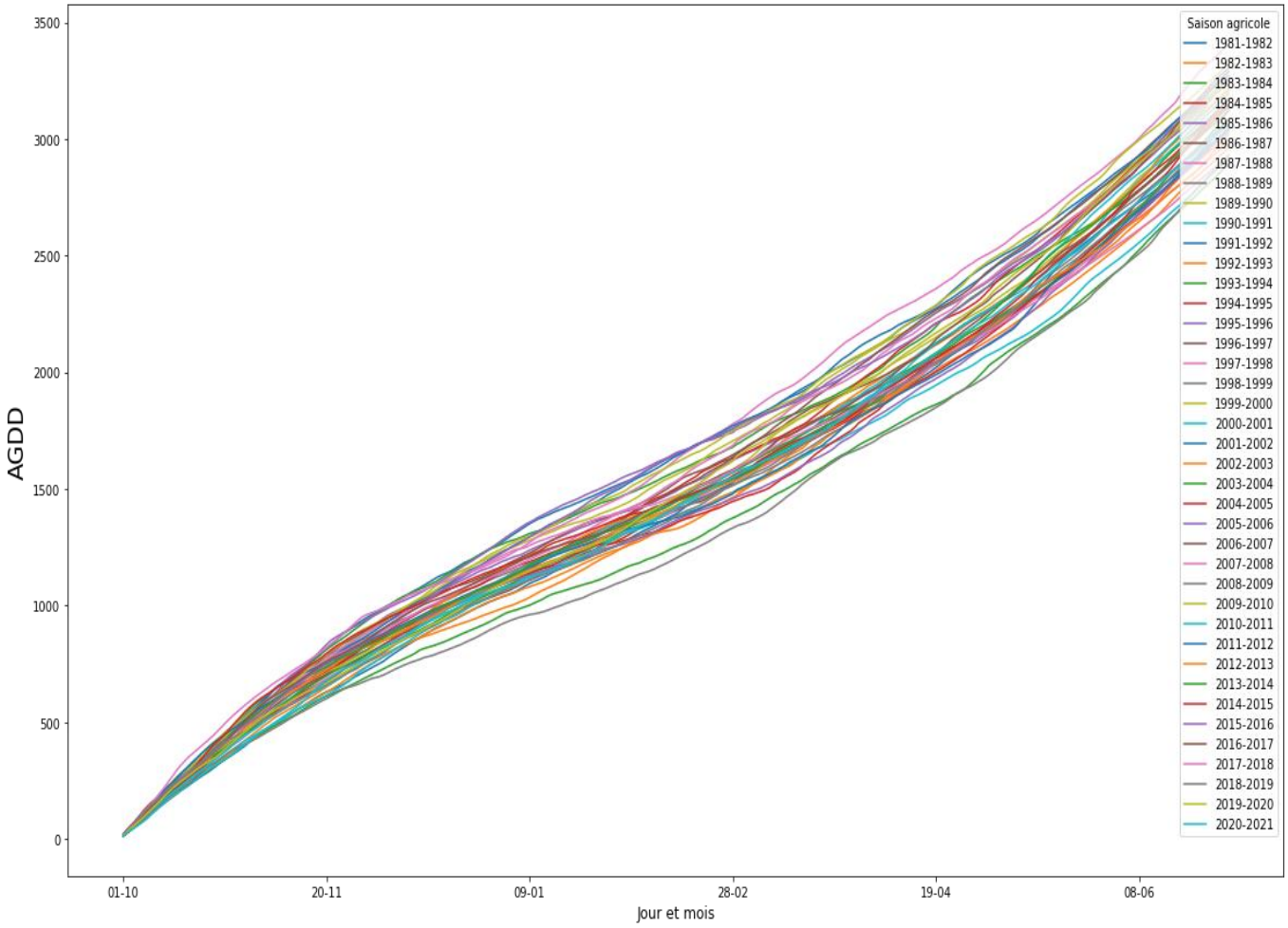


Figure 13: Présentation de toutes les saisons agricoles selon AGDD (cas Blé)

Sortie Python

La variabilité de l'AGDD du blé a une tendance d'être minimale au début de la saison agricole dans la plupart des années, mais après quelque mois les séries croient positivement.

Avec une première vision des courbes de ces années nous avons remarqué qu'il y a des similarités et ressemblances au niveau de plusieurs saisons en se basant sur l'allure des courbes par exemple :

Le groupe des saisons : '2001-2002', '2007-2008'

Le groupe des saisons : '1990-1991', '2008-2009'

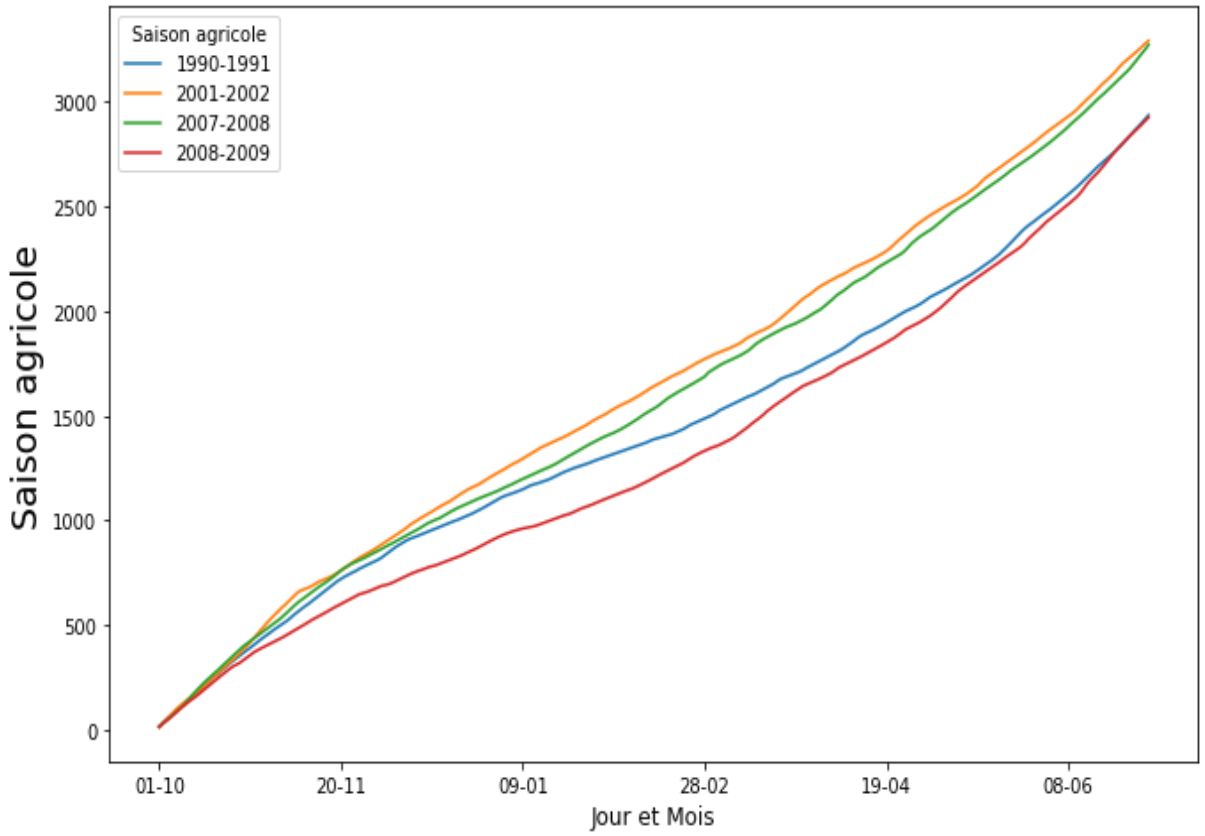


Figure 14: Ressemblances remarquables entre quelques années agricoles selon AGDD (cas blé)

Sortie Python

Malgré que nous puissions sélectionner les années qui se ressemblances par rapport à la température AGDD visuellement, il faut prouver cela statistiquement.

4.2.1.2 Pour la variable APRE

Dans notre visualisation nous allons afficher la tendance des saisons agricoles au niveau de l'APRE selon chaque année :

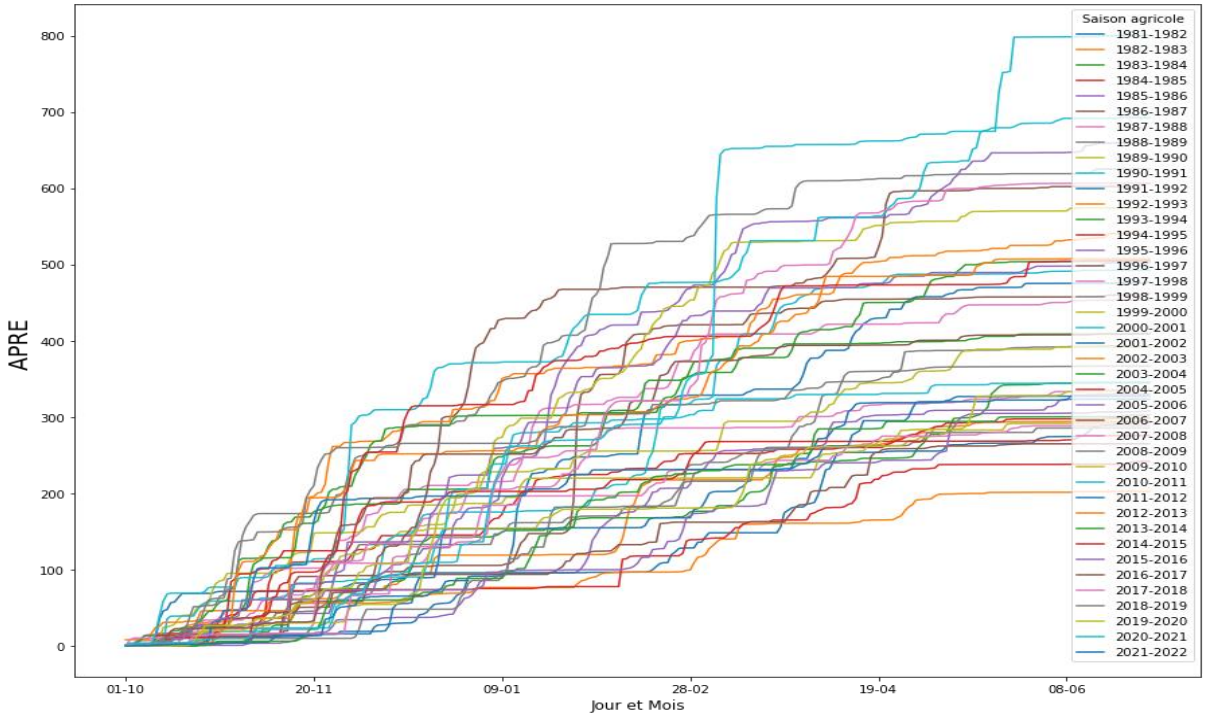


Figure 15: Présentation de toutes les saisons agricoles selon APRE (Cas blé)

Sortie Python

La APRE est une précipitation cumulée, et à partir du premier octobre les séries augmentent puis il stagne dès le mois 5. Nous avons remarqué aussi que dans des saisons la série croient avec un rythme, mais dès le mois janvier ces dernières croient d'une manière très remarquable, ceci est justifié par la construction des saisons agricoles du blé qui débute vers la fin d'une année et continue pour l'année suivante.

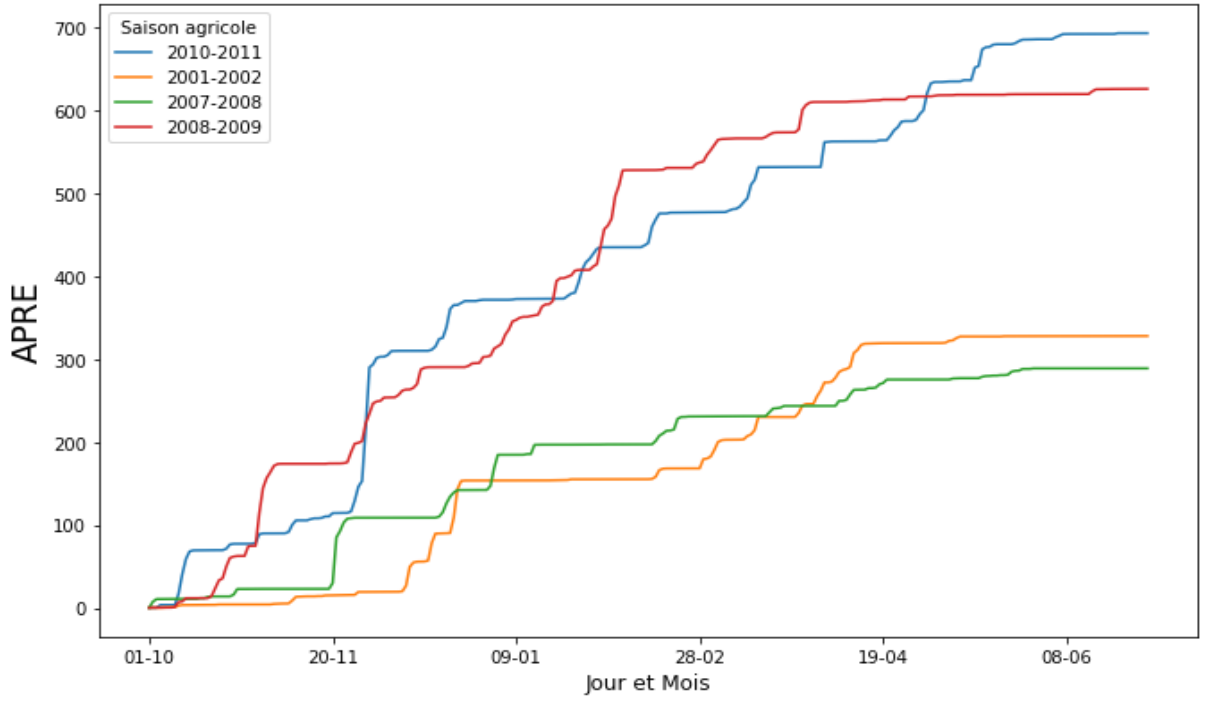


Figure 16: Ressemblances remarquables entre quelques années agricoles selon APRE (Cas blé)

Sortie Python

Selon une première vision de ces distributions nous remarquons qu'il y a une similarité relative dans l'évolution au niveau de années :

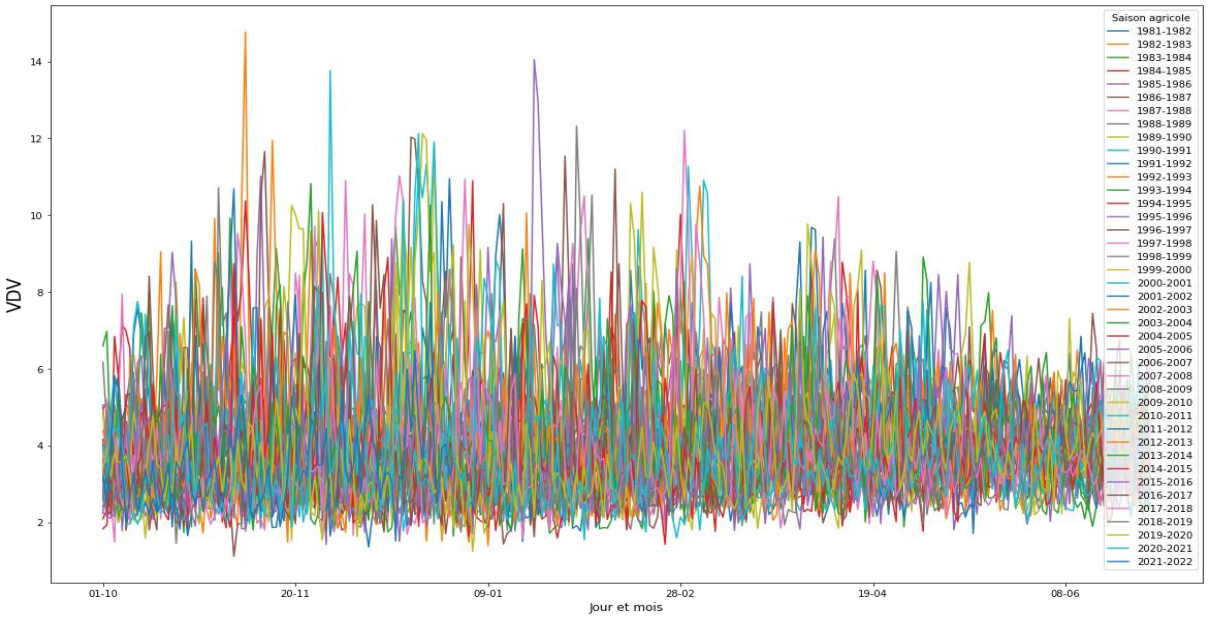
Le groupe de saisons : '2010-2011','2001-2002',

Le groupe de saisons : '2007-2008', '2008-2009'

4.2.1.3 Pour la variable QV2M

Dans notre visualisation, nous allons afficher la tendance selon chaque année :

Sortie Python



Sortie Python

Figure 17: Présentation de toutes les saisons agricoles selon Humidité (cas blé)

Il est difficile de détecter les ressemblances à cause du grand nombre de fluctuations existants, nous essayons de visualiser deux années :

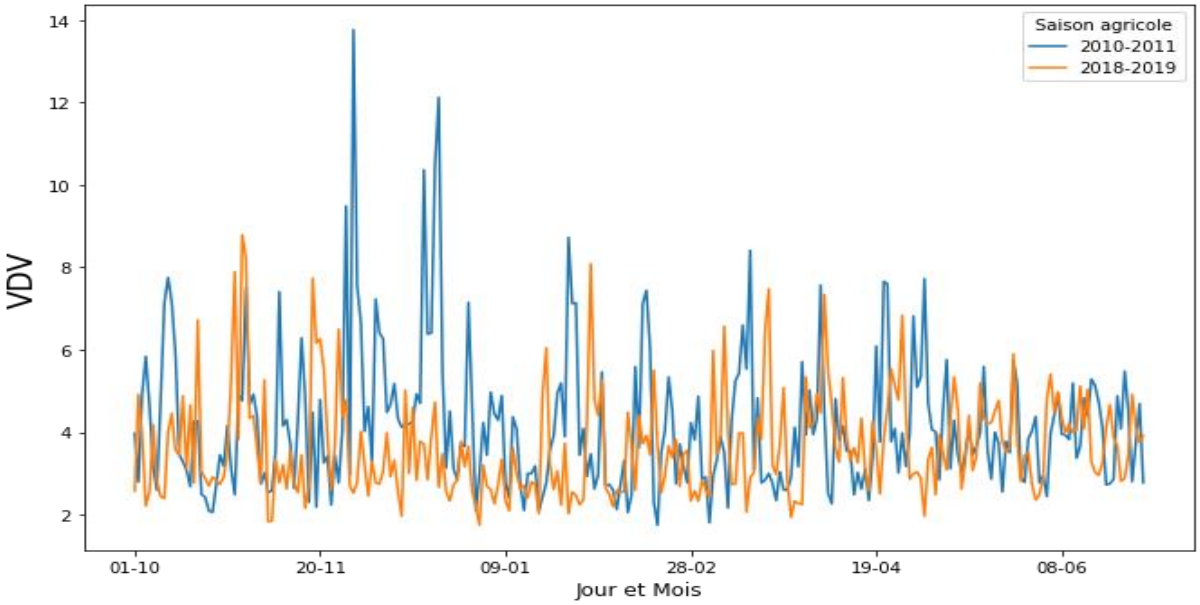


Figure 18: Ressemblances entre deux années agricoles selon Humidité (cas blé)

Nous ne pouvons pas remarquer des différences d'humidité entre ces deux saisons agricoles mais il faut tester cela statistiquement.

4.2.1.3 Pour la variable vitesse de vents

Dans notre visualisation nous allons afficher la tendance selon chaque année :

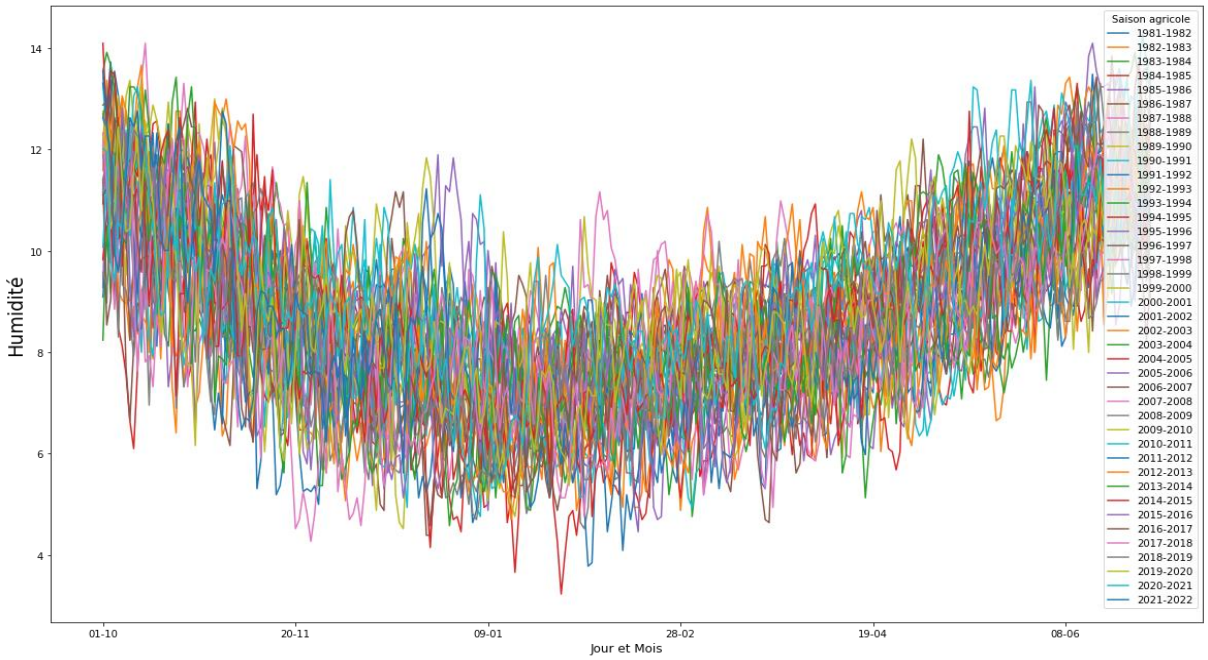


Figure 19: Présentation de toutes les saisons agricoles selon vitesse de vents (Cas blé)

Sortie Python

La variabilité des séries de VDV est caractérisée par beaucoup de fluctuation et même si on diminue les années observées ne nous pouvons pas décider les groupes
Visuellement

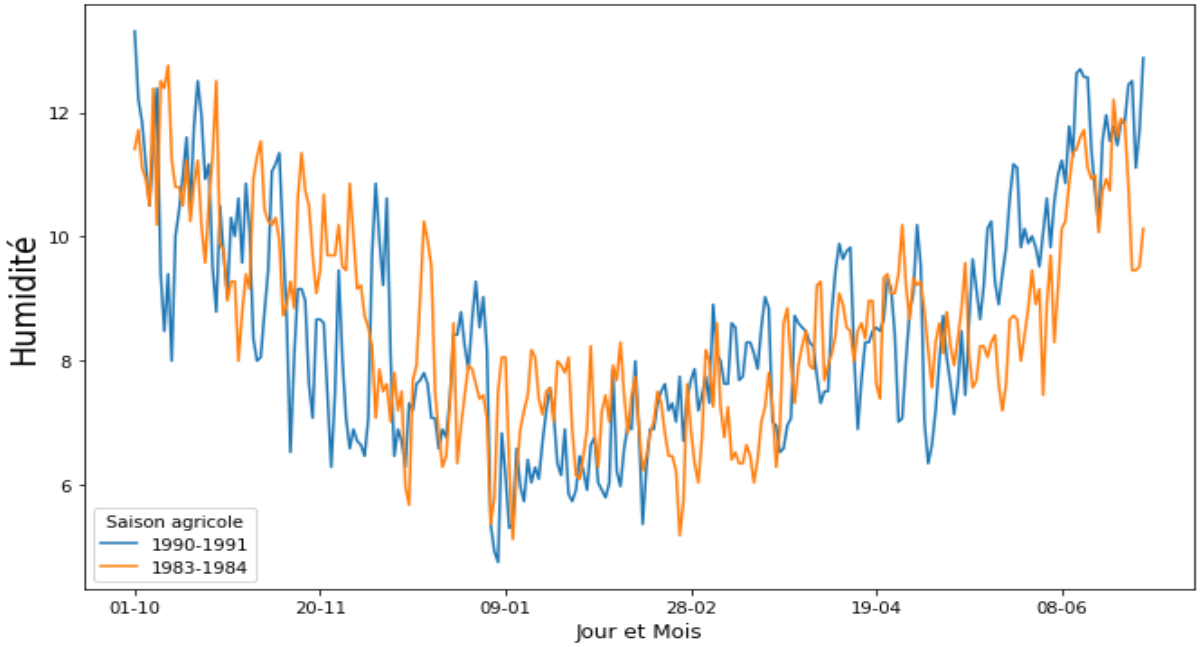


Figure 20: Ressemblances entre deux années agricoles selon vitesse de vents (Cas blé)

CHAPITRE III

Clustering des saisons agricoles selon chaque variable météorologiques

CHAPITRE III : Clustering des saisons agricoles selon chaque variable météorologique

Dans le but de justifier les résultats visualisés dans le chapitre précèdent, nous avons adopté plusieurs méthodes pour la quête des similarités entre les saisons agricoles, notre travail a été décomposé en deux partis, la première consiste à réaliser un clustering mais sur l'année en sa globalité pour la culture agrumes, et la deuxième sera basée sur le clustering des saisons agricoles du blé.

1. Revue sur les clusterings [4]

1.1 Généralités

Nommé aussi une classification non supervisée, le clustering a pour objectif de créer des groupes d'observations homogènes sur la base de l'observation de p descripteurs, à condition que les observations au sein du même groupe soient les plus similaires, et les groupes soient les plus différents possibles les uns des autres.

- Les observations sont décrites par un ensemble de p variables explicatives.
- L'ensemble Xi = (Xi1, . . ., Xip) sont les variables explicatives pour l'individu i ($1 \leq i \leq n$)

Le but est de mettre chaque individu dans un des K groupes tel que :

- $Zi \in \{1, . . ., K\}$ est le numéro du groupe de l'individu i
- (Z1, . . ., Zn) est la partition des n individus en K groupes

La figure suivante montre un groupe d'observation avant et après le clustering :

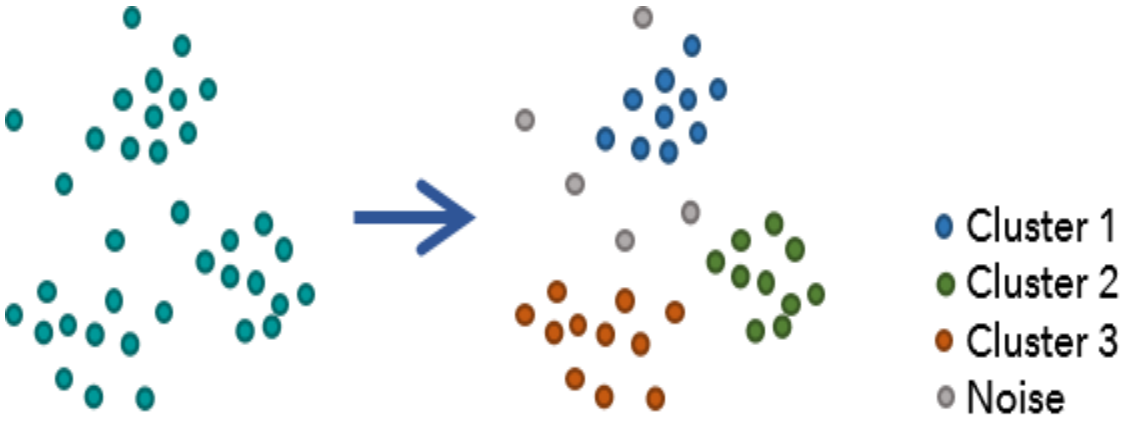


Figure 21: Nuage de points avant et après le clustering

Un bon clustering a deux conditions :

Homogènes : les éléments d'un même cluster sont similaires.
Séparés : les éléments de différents clusters sont différents.

1.2 La distance entre les individus (observations)

Généralement, le clustering est fondé sur la méthode du calcul des distances entre les observations.

Soit x_i et x_j deux données (observation / individus) différentes dont nous voulons calculer la distance.

Parmi les distances les plus utilisés nous citons [5]:

- **La distance euclidienne :**

$$D_n(x_i, x_j) = \sqrt{\sum_{k=1}^{n_n} (x_{ik} - x_{jk})^2}$$

- **La distance Minkowsky :**

$$D_{np}(x_i, x_j) = \left(\sum_{k=1}^{n_n} (x_{ik} - x_{jk})^2 \right)^2$$

- **La distance de Manhattan :**

$$\boldsymbol{D_n(x_i, x_j) = \sum_{k=1}^{n_n} |x_{ik} - x_{jk}|}$$

Lors de l'utilisation de ces distances, il faut normaliser les variables avec échelles différents, ce traitement permet de diminuer le biais dommageable lors du calcul de distances.

Pour le cas des séries temporelles nous avons un autre outil de comparaisons et calcul de distances :

- **DTW : La déformation temporelle dynamique** [6]

La déformation temporelle dynamique est un outil d'évaluation des similitudes entre les séries chronologiques ou les suites qui peuvent varier au cours du temps. Elle a été introduite indépendamment dans la littérature Sakoe and Chiba.[2]

La déformation temporelle dynamique cherche l'alignement temporel [3] qui minimise la distance euclidienne entre les séries alignées.

[2] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust. 1978;26:43–49.

[3] Un alignement temporel est une correspondance entre les index temporels des deux séries chronologiques.

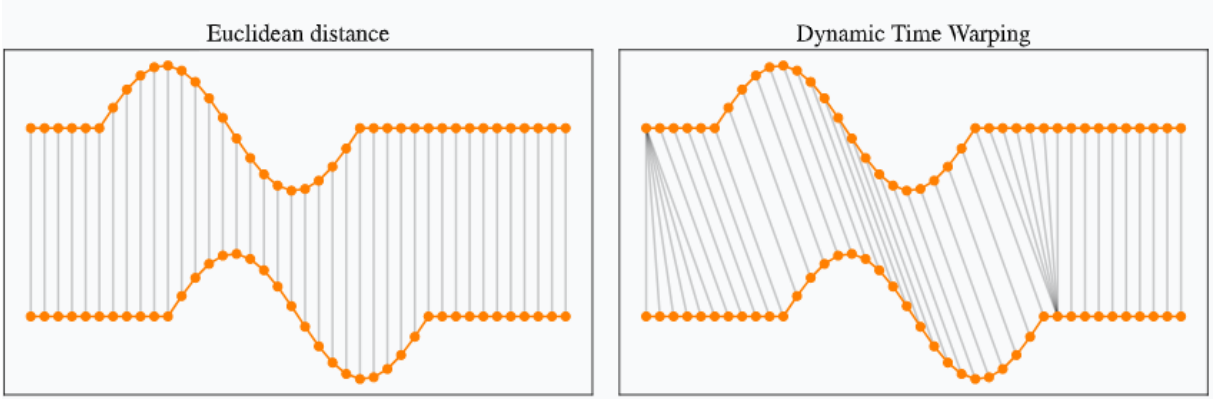


Figure 22: Comparaison entre la distance euclidienne et la déformation temporelle dynamique

Source: An introduction to Dynamic Time Warping (rtavenar.github.io)

La déformation temporelle dynamique sert à minimiser la distance euclidienne entre les séries chronologiques alignées sous tous les alignements temporels admissibles. Dans la figure ci-dessous les points en bleu correspondent à des répétitions d'éléments de séries chronologiques induites par l'alignement temporel optimal récupéré par DTW.

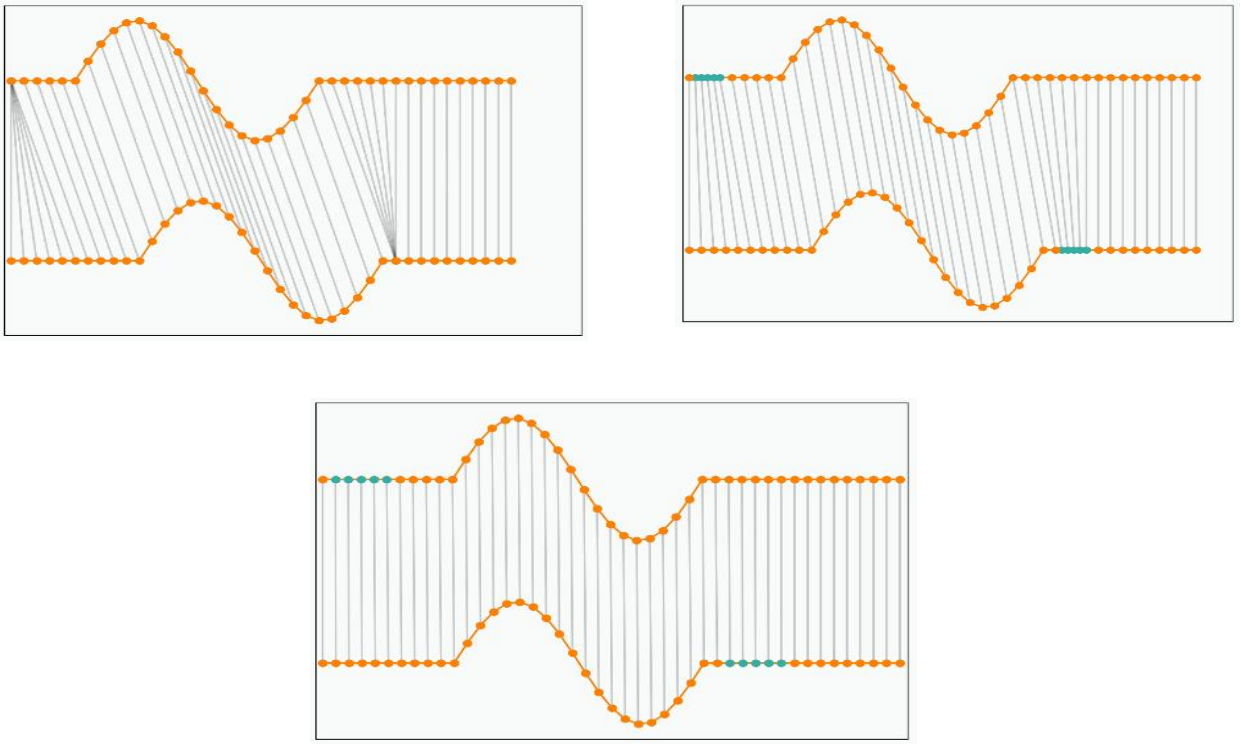


Figure 23: Fonctionnement de DTW

Source : An introduction to Dynamic Time Warping (rtavenar.github.io)

En effet, une fois insérés ces éléments répétés, toutes les correspondances deviennent verticales, ce qui reflète le comportement typique de la distance euclidienne.

Le problème d'optimisation s'écrit comme suit [7] :

$$\mathbf{DTW_q(x,x')} = \min_{\boldsymbol{\pi}\in\mathbf{A(x,x')}} \left(\sum_{\mathbf{(i,j)}\in\boldsymbol{\pi}} \mathbf{d}\left(\mathbf{x_i}, \mathbf{x'_j}\right)^{\mathbf{q}} \right)^{\frac{\mathbf{1}}{\mathbf{q}}}$$

Un chemin d'alignement π de longueur K est une séquence de K paires d'index $\left((i_0, j_0), \dots, (i_{K-1}, j_{K-1})\right)$ et $\mathcal{A}(x', x^r)$ est considéré comme l'ensemble de tous les chemins admissibles.

Un chemin est dit admissible lorsque le parcours remplit les conditions suivantes :

- Le début (resp. la fin) des séries chronologiques sont appariés ensemble

$$\pi_0 = (0,0)$$
$$\pi_{K-1} = (n-1, m-1)$$

- La séquence augmente de manière monotone dans les deux i et j et tous les index de séries chronologiques doivent apparaître au moins une fois

$$i_{k-1} \leq i_k \leq i_k - 1 + 1$$
$$j_{k-1} \leq j_k \leq j_k - 1 + 1$$

Contraintes supplémentaires :

La déformation temporelle dynamique est invariante des décalages temporels, quelle que soit leur durée temporelle. Et dans notre cas de traitement des cultures agricoles, il faut imposer des contraintes supplémentaires sur l'ensemble des chemins admissibles. Afin de permettre uniquement les invariances aux déformations locales.

De telles contraintes se traduisent généralement par l'application d'entrées non nulles dans Api pour rester proche de la diagonale. La bande de Sakoe-Chiba est une bande de largeur constante paramétrée par un rayon r.

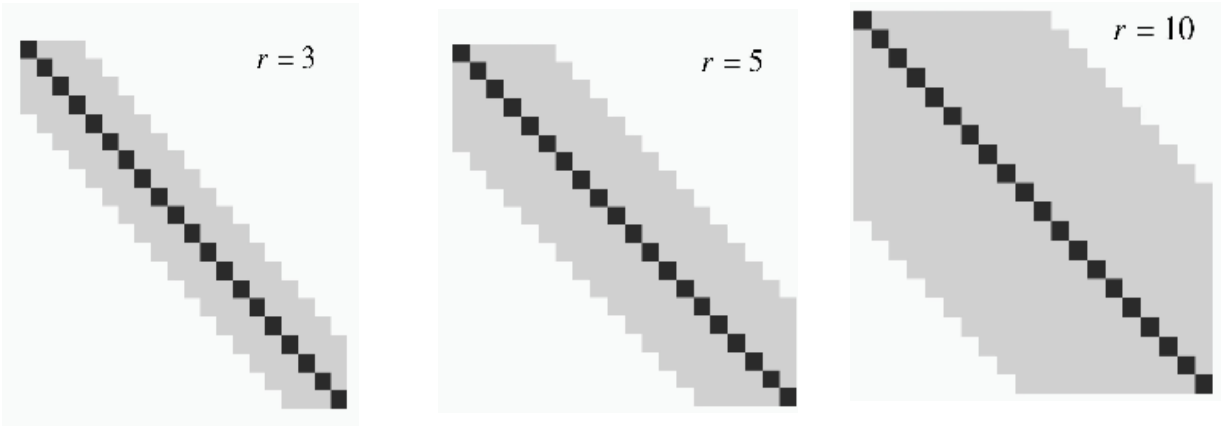


Figure 24:Visualisation des contraintes globales DTW (bande de Sakoe-Chiba)

Source: An introduction to Dynamic Time Warping (rtavenar.github.io)

En pratique, les contraintes globales sur les chemins DTW admissibles limitent l'ensemble des correspondances possibles pour chaque élément d'une série chronologique. Le nombre de correspondances possibles pour un élément est toujours 2r+1 pour les contraintes

de Sakoe-Chiba.

Et grace à ces contraintes, nous pouvons limiter l'invariance de décalage seulement aux changements locaux. Alors, DTW avec une contrainte de rayon de bande sakoe-Chiba r est invariant aux décalages temporels de magnitude allant jusqu'à r, mais n'est plus invariant à des décalages temporels plus longs.

Comparaison entre la distance euclidienne avec l'utilisation de DTW en cas d'utilisation du K-means :

Dans la figure ci-dessous chaque sous-figure représente les séries d'un amas donné ainsi que leur centroïde (en orange).

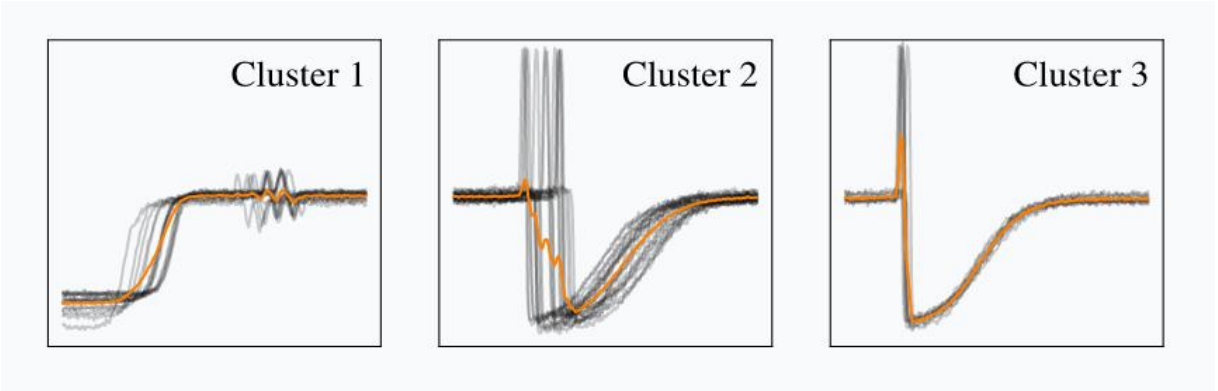


Figure 25: Utilisation de la distance euclidienne dans k-means clusterin

Source: An introduction to Dynamic Time Warping (rtavenar.github.io)

Comme on peut le constater, premièrement, les barycentres de chaque cluster ne sont pas particulièrement représentatifs des séries chronologiques rassemblées dans les clusters, et deuxièmement le cluster 2 mélange deux formes de séries chronologiques distinctes. Ce qu'est n'est pas très satisfaisant.

Par contre, dans le cas de la figure ci-dessous, les séries chronologiques dans chaque groupe sont très similaires jusqu'à un décalage temporel qui va être pris en considération, et que nous pouvons gérer avec la contrainte de sakoe chiba.

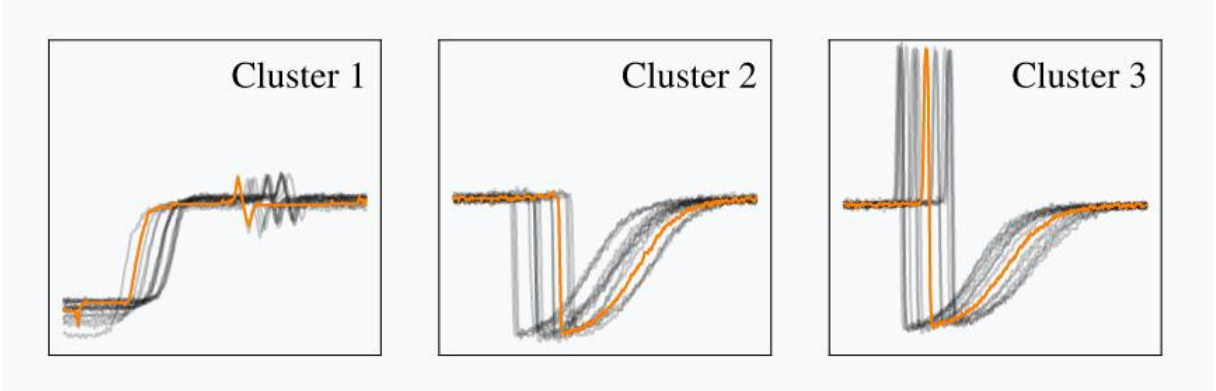


Figure 26: Utilisation de la déformation temporelle dynamique dans k-means clustering

Source: An introduction to Dynamic Time Warping (rtavenar.github.io)

1.3 Les méthodes de clustering

1.3.1 Méthodes hiérarchiques : La CAH (classification ascendante hiérarchique)

Cette méthode consiste à fournir un ensemble de partitions passant du moins en moins fines obtenus par un regroupement successif de parties, lorsque l'algorithme termine, nous allons récupérer une hiérarchie de partitions en n classes, avec diminution de l'inertie interclasses dans chaque agrégation.

Algorithme

- Calculer la différence entre les N objets (observations) deux à deux.
- Regrouper deux objets qui minimisent les critères d'agrégation donnés (Création d'une classe contenant ces deux objets)
- Calculer la dissimilarité entre la classe déjà obtenue et les N-2 autres objets.
- Regrouper les deux objets ou classes d'objets dont le regroupement minimisera le critère d'agrégation donné.

Réaliser les mêmes étapes jusqu'à tous les objets soient groupés.

Stratégies d'agrégation :

- **La méthode de Ward :** Cette méthode repose sur le critère de Ward, qui se repose sur le regroupement des classes de sorte que la variance inter groupe reste la plus grande et la variance intra groupe la plus petite.

$$\mathbf{D(C_1, C_2) = \frac{w_A w_B}{w_B + w_B} d(g_1, g_2)}$$

Avec W_A représente le poids de la classe A et g_A le centre de gravité de de la classe A.

Le centre de gravité de la classe union obtenue s'écrit comme suit :

$$\mathbf{g = \frac{w_A\, g_1 \; + w_B\, g_2}{w_{C_1} + w_{C_2}}}$$

- **Stratégie du saut minimum** : nommée aussi single linkage, est une méthode qui considère la plus petite distance entre éléments comme la distance de calcul :

$$\mathbf{\Delta(A, B) = \min_{i \in A, j \in B} d(i, j)}$$

- **Stratégie du saut maximum** : nommée aussi méthode du diamètre ou complète linkage, cette dernière considère la distance entre parties est la plus grande distance entre éléments des deux parties.

$$\boldsymbol{\Delta(A,B) = \max_{i \in A, j \in B} d(i,j)}$$

- **Stratégie suivant la distance moyenne entre les classes :** Nommé aussi Between-groups Average Linkage **est** une méthode qui prend en considération la moyenne de distances entre les classes pour chaque individu**.**

$$\boldsymbol{\Delta(A,B) = \frac{1}{card(A)card(B)} \sum_{i \in A, j \in B} d(i,j)}$$

- **La méthode des barycentres :** Le point de référence correspond à la moyenne des points des classes. Ensuite, les distances sont calculées à partir de ces moyennes qui vont représenter les classes.

$$\boldsymbol{D(A,B) = d(g_1, g_2)}$$

Avec g_1, g_2 sont respectivement les barycentres de A et B

<u>Le choix du nombre de classes</u>

Pour ce choix, les graphiques sont très utiles. Parmi ces derniers les plus utilisés sont :

Le graphique Semi-partial R-squared (SPRSQ) : Il mesure la perte d'inertie interclasse causée en regroupant 2 classes. L'objectif est d'avoir une inertie interclasse maximum, il faut chercher un faible SPRSQ suivi d'un fort SPRSQ à l'agrégation suivante c'est à dire un pic pour k classes et un creux pour k+1 classes indiquera une bonne classification en k+1 classes.

Le dendrogramme : est un graphique sous forme d'arbre binaire, constitué de plusieurs agrégations consécutives jusqu'à ce que tous les individus se réunissent dans une classe. La hauteur de la branche indique proportionnellement la distance entre les deux objets groupés.

Afin savoir le nombre de classes, on coupe le graphique avant une forte perte d'inertie.

1.3.2 Méthodes de partitionnement

Dans ce type de clustering, on commence par une division arbitraire en K classes (K sélectionné) et avec une amélioration itérative nous essayons d'amener une convergence des critères sélectionnés. Basé sur les centres mobiles :

Algorithme :

- Choix aléatoire de k points de l'espace considérés des centres des classes.
- Tous les individus sont affectés à la classe dont le centre est le plus proche au sens de la distance choisie.
- Construire ainsi k classes d'individus
- Calculer les barycentres des classes créées qui deviennent les k nouveaux centre.
- Répéter et itérer les deux étapes précédentes jusqu'à ce que le critère à Minimiser (inertie intra-classes) ne décroisse plus de manière significative (minimum local), ou jusqu'à atteindre un nombre d'itérations fixées.

La méthode de k-means : les barycentres des classes ne sont pas recalculés à la fin des affectations, mais à la fin de chaque allocation d'un individu à une classe. L'algorithme est ainsi plus rapide.

La méthode des nuées dynamiques : ce n'est plus un seul point qui représente une classe mais un noyau de points constitués d'éléments représentatifs de la classe. Cela permet de corriger l'influence d'éventuelles valeurs extrêmes sur le calcul du barycentre.

Cas des séries temporelles

Le clustering avec les séries temporelles se réalise par le même mécanisme sauf que dans ce cas la spécificité est de prendre la série toute entière et la comparer avec les autres séries existantes, en utilisant les métriques adaptées.

Grace au package Tslearn [8], nous pouvons réaliser la manipulation et le clustering des séries temporelles, dans ce package la série chronologique n'est rien de plus qu'un tableau bidimensionnel dont la première dimension correspond à l'axe temporel et la seconde étant la dimensionnalité de l'entité, et l'importation des séries temporelles demandent que chaque ligne représente une seule série chronologique (et les séries chronologiques d'un jeu de données ne sont pas obligées d'avoir une longueur similaire).

Au temps du travail avec les bases de données des séries chronologiques, il sera utile de redimensionner ces dernières par la méthode Min Max scaler donnée par la formule suivante :

$$\mathbf{X_{new}} = \frac{\mathbf{X} - \mathbf{X_{min}}}{\mathbf{X_{max}} - \mathbf{X_{min}}}$$

- **Choix de nombre de classe compatible :**

Nous avons utilisé la méthode KElbowVisualiser [9], cette méthode sert à sélectionner le nombre optimal de clusters en ajustant le modèle avec une plage de valeur et en présentant cela sur un graphique, si le graphique linéaire ressemble à un bras, alors le « coude » c'est-à-dire le point d'inflexion sur la courbe est une bonne indication que le modèle sous-jacent convient le mieux à ce point.

Le KElbowVisualizer implémente la méthode « elbow » de sélection du nombre optimal de clusters, par exemple pour le clustering K-means qui est un algorithme d'apprentissage automatique simple et non supervisé qui regroupe les données en un nombre spécifié k de clusters. Et comme l'utilisateur est obliger de spécifier à l'avance quel k choisir, il faut choisir le bon k pour le jeu de données.

La méthode elbow exécute le clustering k-means sur le jeu de données pour une plage de valeurs pour k (par exemple k de 1 à 15). Ensuite pour chaque valeur de k, elle calcule un score moyen pour tous les clusters. Par défaut, le score est calculé, avec la somme des distances carrées de chaque point à son centre assigné. D'autres mesures peuvent également être utilisées telles que le coefficient de silhouette moyen pour tous les échantillons.

Lorsque ces mesures globales pour chaque modèle sont tracées, il est possible de déterminer visuellement la meilleure valeur pour k. Si le graphique linéaire ressemble à un bras, alors le « coude » (le point d'inflexion sur la courbe) est la meilleure valeur de k.

1.3.3 Validation du clustering

Les statistiques de validation des regroupements peuvent être classées en 4 méthodes [10] :

Validation du clustering relatif : Consiste à évaluer la structure du clustering en variant différentes valeurs de paramètres pour le même algorithme. Tel que varier le nombre de classes k. cette méthode est généralement utilisé pour déterminer le nombre optimal de classes.

Validation de clustering interne : Exploite les informations internes du processus de clustering afin d'évaluer la qualité d'une structure de clustering sans référence à des informations externes.

Validation de clustering externe : Sert à comparer les résultats d'une analyse de cluster à un résultat connu de l'extérieur, comme des étiquettes de classe fournies en externe.

Validation de la stabilité du clustering : C'est une version spéciale de la validation interne. Son rôle est d'évaluer la cohérence d'un résultat de clustering en le comparant avec les clusters obtenus après la suppression de chaque colonne.

- **Mesures utilisées pour évaluer et valide les clusters**

Silhouette Score

Le score de silhouette et le tracé de silhouette sont utilisés afin de mesurer la distance de séparation entre les clusters, le score affiche une mesure de la proximité de chaque point d'un cluster avec les autres points des clusters voisins. Cette mesure est caractérisée par une plage de [-1, 1] pour inspecter visuellement les similitudes au sein des grappes et les différences entre les grappes. Lorsque les coefficients de silhouette sont élevés c'est-à-dire plus ils sont proches de 1, plus les échantillons de la grappe sont éloignés des échantillons des grappes voisines. [11]

- Une valeur de 0 indique que l'échantillon est sur ou très près de la limite de décision entre deux clusters voisins.
- Les valeurs négatives indiquent que ces échantillons ont peut-être été affectés dans un mauvais cluster.
- La taille/épaisseur des silhouettes est également proportionnelle au nombre d'échantillons à l'intérieur de ce cluster.

Le score silhouette est calculé à l'aide de la distance intra-grappe moyenne et la distance moyenne la plus proche de la grappe pour chaque échantillon. [12]
Son coefficient est donné par la relation suivante :

$$\mathbf{i\ n(n - i) / max(i, n)}$$

Avec:

n : est la distance entre chaque échantillon et le groupe le plus proche dont l'échantillon ne fait pas partie

i : est la distance moyenne à l'intérieur de chaque groupe.

En calculant la moyenne des coefficients de silhouette, nous pouvons obtenir un score de silhouette global qui est utilisé pour décrire la performance de l'ensemble de la population avec une seule valeur.

Calinski-Harabasz Index

L'indice de Calinski-Harabasz est également connu sous le nom de critère de rapport de variance. Il est parmi les mesures d'évaluation des algorithmes de clustering souvent utilisé pour évaluer la qualité de la division par un algorithme de clustering pour un nombre donné de clusters. [13]

Le score est calculé comme un rapport entre la somme de la dispersion inter-grappes et la somme de la dispersion intra-grappes pour tous les amas (où la dispersion est la somme des distances au carré).

Plus l'indice est élevé, meilleures sont les performances, car un CH élevé signifie que les clusters sont bien éloignés et que dans chaque cluster les amas (pour notre cas : les séries temporelles) sont plus proches les unes des autres. [14]

L'indice de Calinski-Harabasz (CH) est calculé suivant trois étapes :

- **Calcul de la dispersion entre les clusters :**

La dispersion inter-clusters ou la somme des carrés entre les groupes (BGSS) mesure la somme pondérée des distances au carré entre les centroïdes d'un amas et le centroïde de l'ensemble de données (barycentre).

$$\mathbf{BGSS} = \sum_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{n_k} \times \|\mathbf{C_k} - \mathbf{C}\|^{\mathbf{2}}$$

Avec :

K : le nombre de clusters
n_k : le nombre d'observations en cluster k
C_k : le centroïde de cluster k
C : le centroïde du jeu de données (barycentre)

- **Calcul de la dispersion intra-cluster :**

La dispersion intra-cluster ou la somme intra-groupe des carrés (WGSS) mesure la somme des distances au carré entre chaque observation et le centroïde d'un même amas.

Pour chaque cluster k nous calculerons le $WGSS_k$ est calculé comme suit :

$$\mathbf{WGSS_k} = \sum_{\mathbf{i=1}}^{\mathbf{n_k}} ||\mathbf{X_{ik}} - \mathbf{C_k}\,\|^{\mathbf{2}}$$

Avec :

n_k : le nombre d'observations dans le cluster k
X_{ik} : la i -ème observation dans le cluster k
C_k : le centroïde du cluster k

Ensuite additionner toutes les sommes intra déjà calculées pour chaque groupe :

$$\mathbf{WGSS} = \sum_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{WGSS_k}$$

WGSSk : la somme intra-groupe des carrés du cluster k
K : le nombre de clusters

- **Calculer l'indice de Calinski-Harabasz :**

L'indice de Calinski-Harabasz est calculé comme suit :

$$\mathbf{CH} = \frac{\dfrac{\mathbf{BGSS}}{\mathbf{K-1}}}{\dfrac{\mathbf{WGSS}}{\mathbf{N-K}}} = \frac{\mathbf{BGSS}}{\mathbf{WGSS}} \times \frac{\mathbf{N-K}}{\mathbf{K-1}}$$

Avec :

BGSS : somme des carrés entre les clusters
WGSS : somme des carrés à l'intérieur du cluster
K : nombre total de clusters
N : nombre total d'observations

Selon cet indice, les meilleurs clustering sont ceux avec les grandes valeurs de CH.

1. Clustering des saisons agricoles

Notre objectif est de trouver des classes d'années qui se ressemblent suivant des variables de météo, alors nous allons considérer le clustering de chaque variable météorologique pour les deux cultures.

1.3Clustering des saisons agricoles pour la culture agrumes

1.3.2 Selon l'AGDD (cas agrumes)

Nous disposons dans chaque ligne de la base de données, d'une série temporelle (saison agricole) observée sur 365 jours, comme marqué dans le tableau suivant :

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	0.000	0.000	0.000	0.000	0.000	...	2126.900	2129.905	2132.995	2136.790	2139.560
1982	0.000	0.870	2.685	3.440	5.265	...	2029.900	2029.900	2029.900	2029.900	2029.900
1983	0.000	0.000	0.000	0.000	0.000	...	2312.260	2312.260	2312.260	2312.260	2312.260
1984	0.145	0.295	0.295	0.295	0.295	...	1977.370	1977.370	1977.370	1977.370	1977.370
1985	0.000	0.000	0.000	0.110	0.245	...	2306.025	2309.655	2309.655	2309.655	2309.655
1986	1.235	2.430	2.430	2.430	2.430	...	2043.845	2043.845	2043.845	2043.845	2043.845
1987	0.000	0.000	0.000	0.000	0.000	...	2300.220	2300.220	2300.220	2300.220	2300.220
1988	0.000	0.000	0.550	0.870	3.835	...	2263.380	2263.380	2263.380	2263.380	2263.380
1989	0.000	0.000	0.000	0.000	0.000	...	2387.525	2389.235	2391.690	2391.725	2391.725
1990	0.360	0.955	0.955	0.955	0.955	...	2292.650	2293.505	2294.365	2295.460	2295.785

Tableau 9: Base de données de l'AGDD

Afin de réaliser une bonne comparaison entre nos séries temporelles, il faut normaliser la base de données entre 0 et 1, et pour ce faire nous avons opté à utiliser l'outil de la normalisation Min-Max dont nous avons soustrait en toute observation à l'instant i le minimum de toute la base de données et diviser par la différence entre le maximum et le minimum. (Voir tableau ci-dessous)

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.844132	0.845325	0.846551	0.848057	0.849156
1982	0.000000	0.000345	0.001066	0.001365	0.002090	...	0.805634	0.805634	0.805634	0.805634	0.805634
1983	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.917698	0.917698	0.917698	0.917698	0.917698
1984	0.000058	0.000117	0.000117	0.000117	0.000117	...	0.784786	0.784786	0.784786	0.784786	0.784786
1985	0.000000	0.000000	0.000000	0.000044	0.000097	...	0.915224	0.916664	0.916664	0.916664	0.916664

Tableau 10: Base de données après normalisation Min-Max cas AGDD (cas agrumes)

Sortie Python

L'étape suivante c'est de choisir le nombre K de cluster, alors nous avons opté à utiliser la méthode d'elbow avec l'algorithme Timeseries Kmeans et sa métrique DTW ainsi la contrainte de Sakoe Chiba avec un rayon égal 10 (voir la figure 27), vu que les agrumes passent d'un stade à autre en environ un mois donc nous avons fixé que le rayon sera la durée de stade divisée par 3.

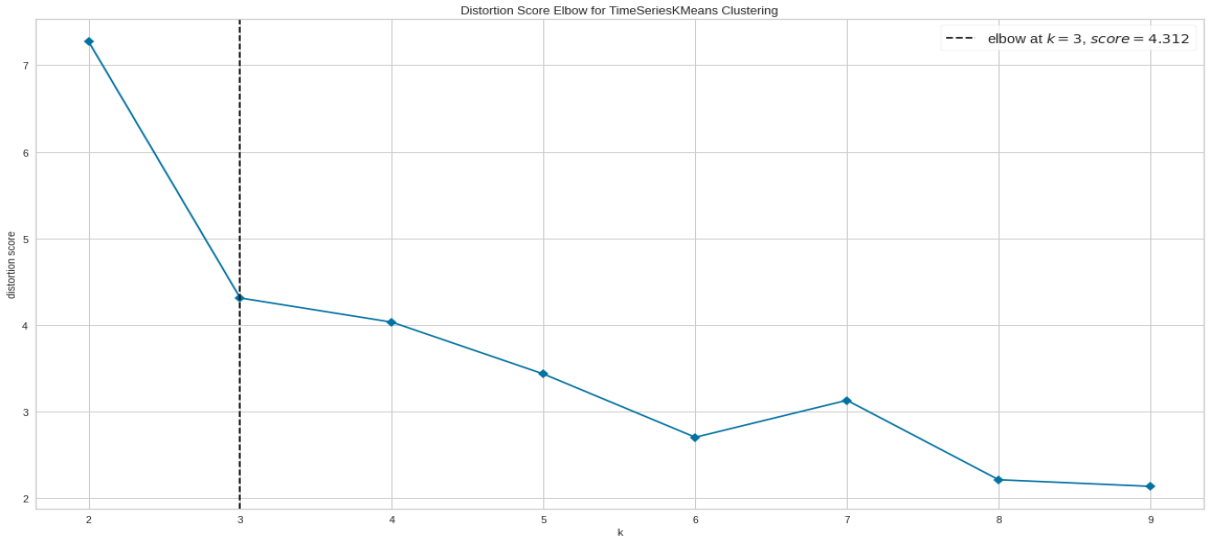


Figure 27: Le nombre de classe pour l'AGDD (cas agrumes)

Sortie Python

Donc selon le graphique précèdent, nous conclurons qu'on a trois classes pour le cas d'AGDD. Nous avons réalisé ensuite à travers l'algorithme Timeseries k-means le clustering pour trouver les résultats suivants :

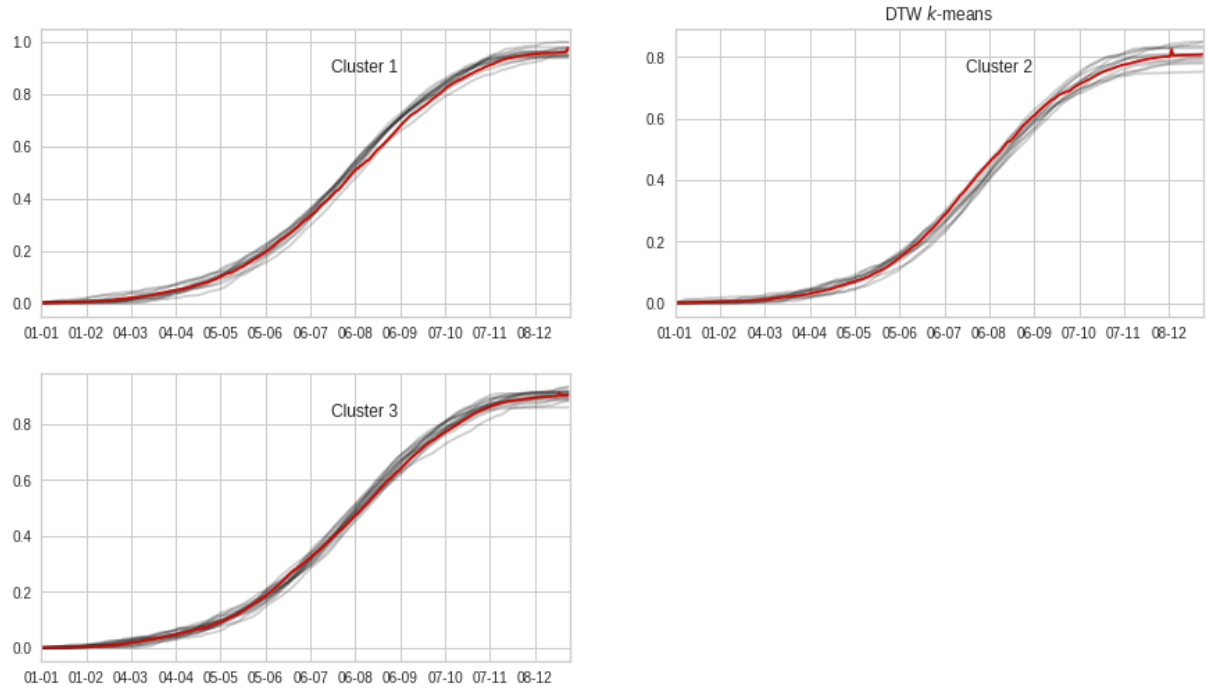


Figure 28: Groupement des années agricole par cluster selon l'AGDD (cas agrumes)

Sortie Python

Cluster 1 : [1989, 1994, 2001, 2003, 2006, 2011, 2012, 2014, 2015, 2016, 2017, 2020, 2021]
Cluster 2 : [1981, 1982, 1984, 1986, 1991, 1992, 1993, 1996, 2007, 2013, 2018]
Cluster 3 : [1983, 1985, 1987, 1988, 1990, 1995, 1997, 1998, 1999, 2000, 2002, 2004, 2005, 2008, 2009, 2010, 2019]

Le graph ci-dessous montre le groupement des années par cluster :

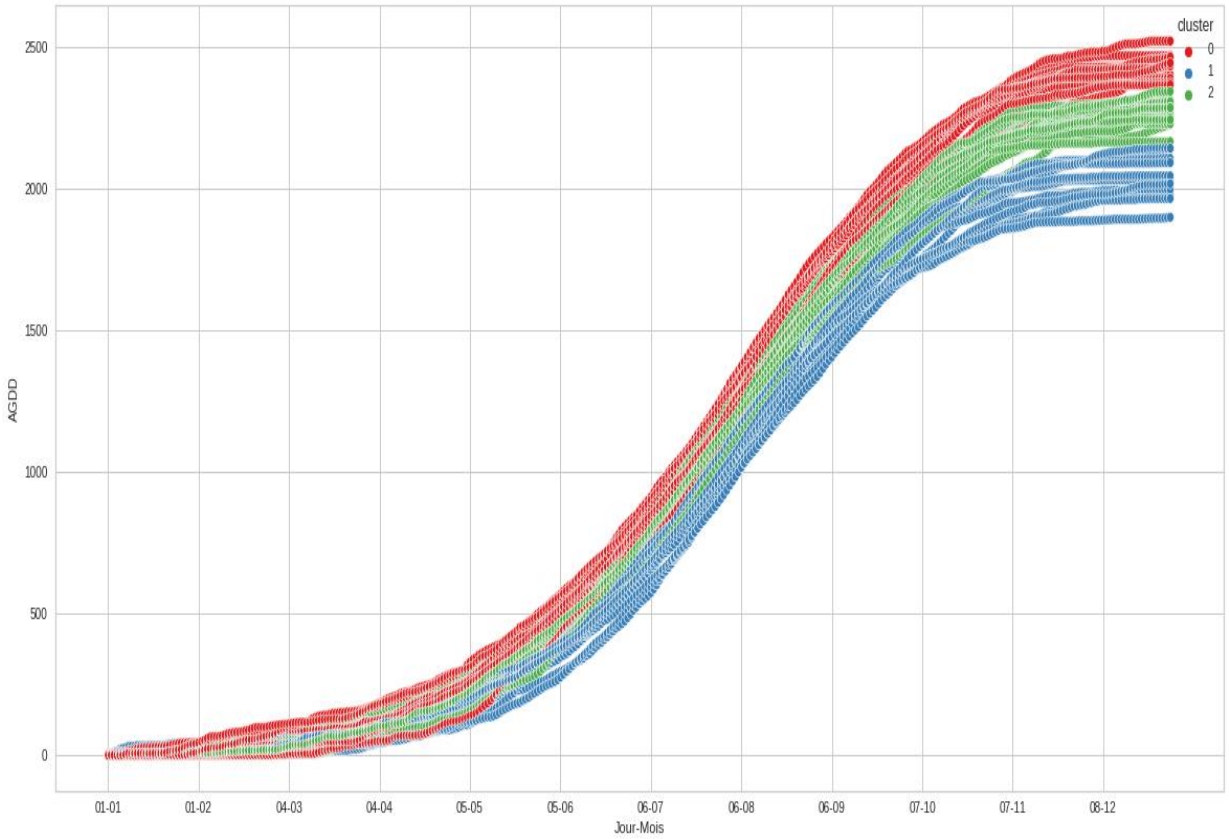


Figure 29: Résultat du clustering pour AGDD (cas agrumes)

Sortie Python

- **La validation et qualité du clustering :**

Silhouette score :

Pour ce faire nous utilisons deux méthodes celle du score silhouette, cette dernière montre que tant que l'indice est proche de 1 nos années en clusters sont bien positionnées.

Nous avons trouvé que le score de silhouette égale 0.6, alors il est moyennement bon et signifie que nos clusters sont moyennement séparés.

Indice Calinski Harabasz :

Cet indice aussi de son côté montre que le nombre optimal de classes est 3, puisqu'il contient le plus grand score de Calinski Harabasz (un score égal à 58.702) parmi toutes les cas k possible, comme indiqué dans la figure ci-dessous.

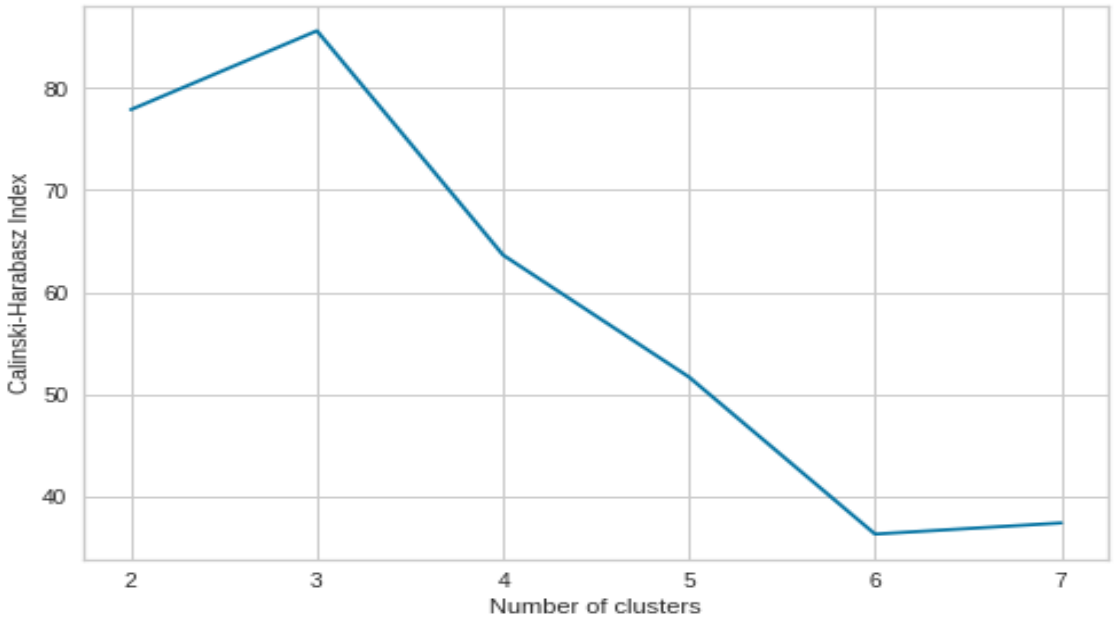


Figure 30: Score de Calinski Harabasz selon les différents k possible de cluster

Sortie Python

Donc nous adoptons 3 classes :

- Le premier groupe (voir figure 29 : couleur rouge) se caractérise par une AGDD grande par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est maximale et s'augmente jusqu'à 2500C.

- Le deuxième groupe (voir figure 29 : couleur bleu) est marqué par une AGDD minimale par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est minimale et diminue jusqu'à moins de 2000C.

- Le troisième groupe (voir figure 29 : couleur vert) se caractérise par une AGDD moyenne par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est moyenne et s'augmente jusqu'à 2300C.

1.3.3 Selon l'APRE

Nous disposons de chaque ligne une série temporelle (Saison agricole) observé sur 365 jours marqué dans le tableau suivant :

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	0.02	0.02	0.02	0.02	0.49	...	318.33	318.56	318.72	318.72	331.45
1982	0.67	0.67	0.67	0.67	0.67	...	502.05	502.05	502.05	502.05	502.05
1983	0.00	0.00	0.00	0.00	0.00	...	191.25	191.25	191.25	191.25	191.25
1984	0.01	0.10	0.17	7.59	7.81	...	360.46	360.55	360.61	360.71	360.76
1985	0.00	0.00	0.00	0.00	0.68	...	285.36	285.50	297.95	299.43	307.34

Figure 31: Base de données de l'APRE (cas agrumes)

Sortie Python

Par la même démarche de l'AGDD nous avons étudié le clustering de APRE, débutant par normalisation de la base de données

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	0.000029	0.000029	0.000029	0.000029	0.000717	...	0.465831	0.466167	0.466401	0.466401	0.485030
1982	0.000980	0.000980	0.000980	0.000980	0.000980	...	0.734679	0.734679	0.734679	0.734679	0.734679
1983	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.279867	0.279867	0.279867	0.279867	0.279867
1984	0.000015	0.000146	0.000249	0.011107	0.011429	...	0.527482	0.527614	0.527701	0.527848	0.527921
1985	0.000000	0.000000	0.000000	0.000000	0.000995	...	0.417584	0.417789	0.436007	0.438173	0.449748

Figure 32: Base de données après normalisation Min-Max cas APRE (cas agrumes)

Sortie Python

L'étape suivante c'est de choisir le nombre k de clusters, alors nous avons utilisé la méthode d'elbow (voir figure 33) avec l'algorithme TimeseriesKmeans et sa métrique DTW ainsi la contrainte de Sakoe Chiba avec un rayon égal 10.

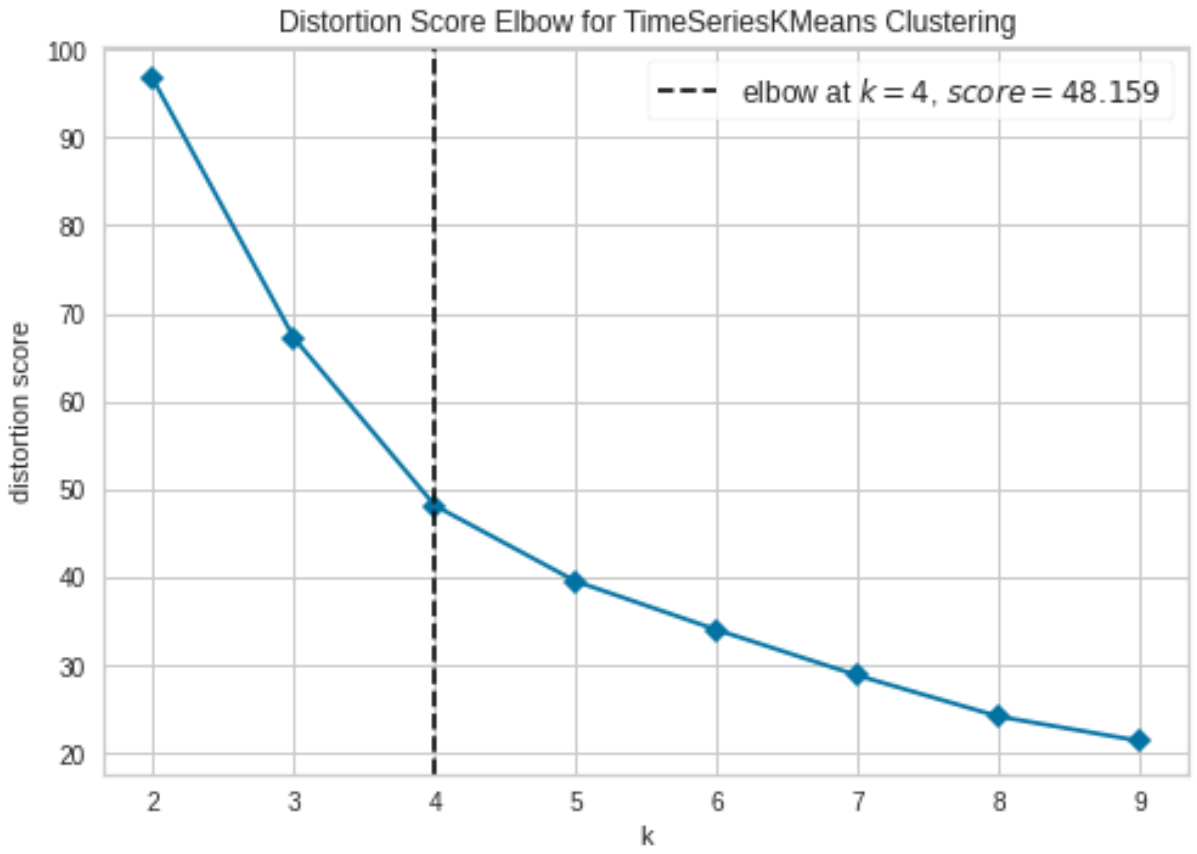


Figure 33: Le nombre de classes pour l'APRE (cas agrumes)

Sortie Python

Nous avons retenu quatre classes pour le cas d'APRE, alors nous avons réalisé le clustering en k=4 à travers l'algorithme Timeseies kmeans.

La figure ci-dessous représente la distribution des saisons par cluster (en gris les saisons agricoles qui se ressemblent et en rouge la moyenne en chaque cluster).

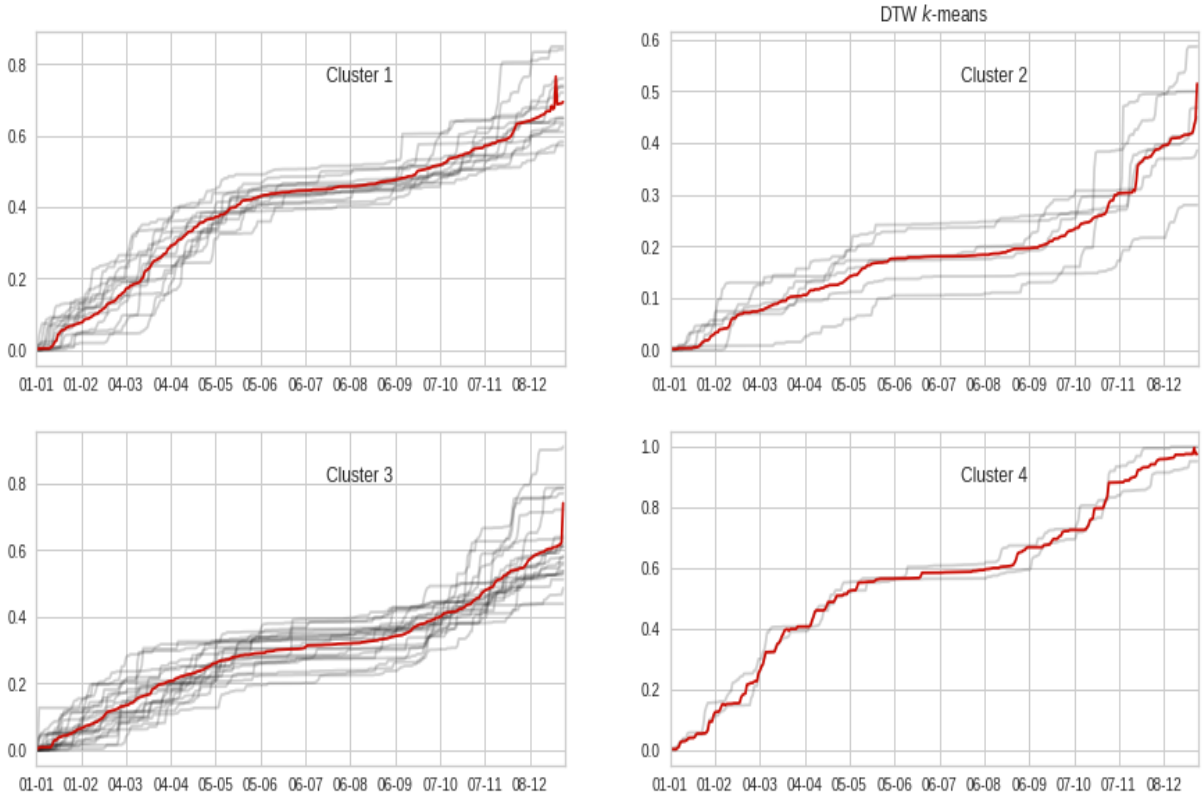


Figure 34: Groupement des années agricole par cluster selon l'APRE (cas agrumes)

Sortie Python

A partir de cette figure, l'algorithme à bien réussis à catégoriser les saisons agricole similaires en termes de silhouette en quatre groupes.

Cluster 1 : [1982, 1986, 1990, 1991, 1992, 1996, 2003, 2004, 2006, 2013, 2015, 2020, 2021]

Cluster 2 : [1983, 1985, 1998, 2000, 2001, 2019]

Cluster 3 : [1981, 1984, 1987, 1988, 1989, 1993, 1994, 1995, 1997, 1999, 2002, 2005, 2007, 2008, 2009, 2011, 2012, 2014, 2016, 2017]

Cluster 4 : [2010, 2018]

Le graphique ci-dessous montre le groupement des années par cluster :

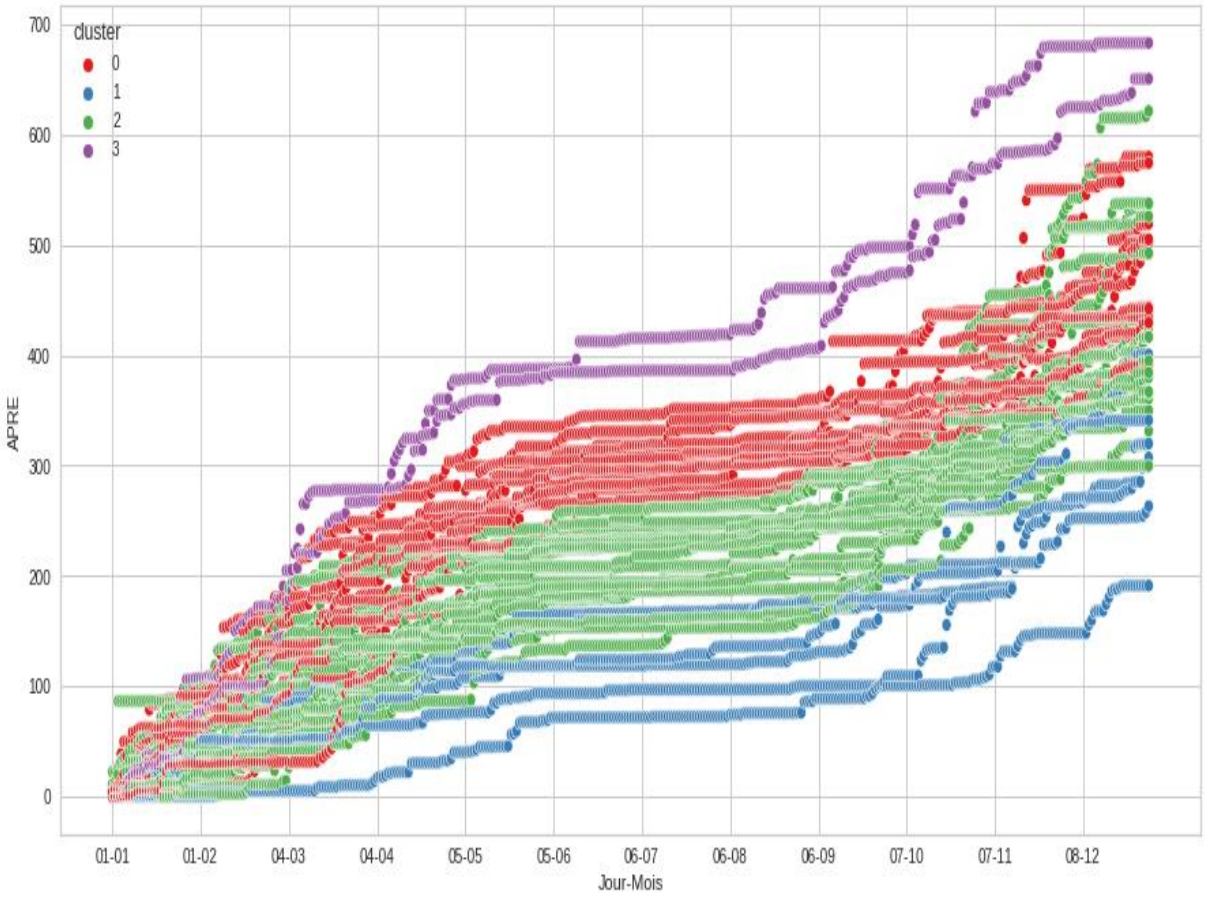


Figure 35: Résultat du clustering pour APRE (cas agrumes)

Sortie Python

- **La qualité et la validation du clustering :**

Afin d'étudier la qualité nous avons utilisé deux méthodes :

Silhouette score

Cette dernière, indique que si l'indice est proche de 1, et donc nos années en clusters sont bien positionnées.
Pour l'AGDD, Nous avons trouvé que le score de silhouette égale à 0.547, il est moyennement bon et signifie que nos clusters sont moyennement séparés.

Indice Calinski Harabasz

Cet indice aussi de son côté, montre que le nombre optimal de classes est 4 (voir figure ci-dessous), puisqu'il contient le plus grand score de Calinski Harabasz (un score égal à 58.702) parmi toutes les k possible.

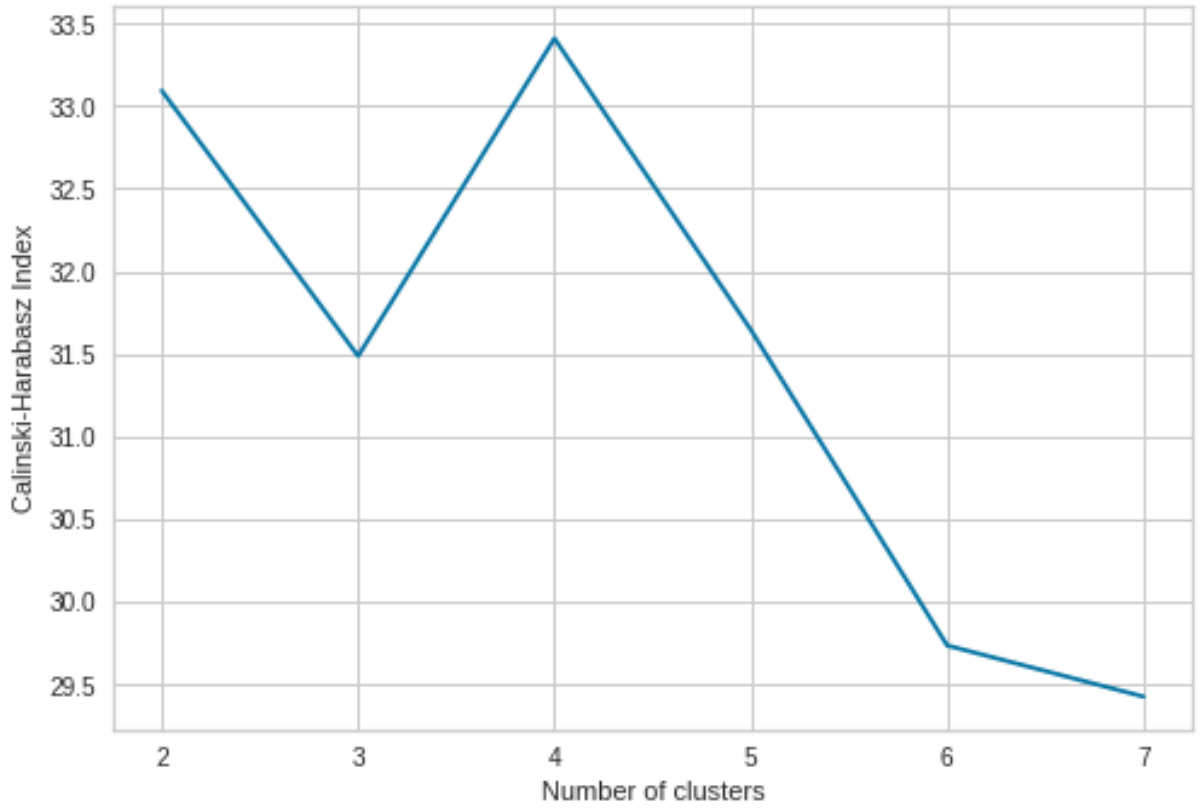


Figure 36: Score de Calinski Harabasz selon les différents k possible de cluster pour APRE (cas agrumes)

Sortie Python

Donc, après les tests de qualité nous avons validé quatre groupes au niveau de l'APRE (voir figure 35) :

- Le premier groupe (voir figure 35 couleur rouge) se caractérise par une APRE moyenne par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est moyenne et s'augmente jusqu'à 500mm.
- Le deuxième groupe (voir figure 35 couleur bleu) se particule par une APRE minimale par rapport aux autres groupes durant toute l'année, ainsi que sa valeur APRE vers de la fin d'année est minimale et diminue jusqu'à moins de 100mm.
- Le troisième groupe (voir figure 35 couleur verte) se caractérise par une APRE moyenne mais faible que celle du groupe 1 durant toute la saison, sauf qu'elle commence à être du même niveau avec les saisons du groupe 1 vers la fin de la saison agricole.
- Le quatrième groupe (voir figure 35 couleur mouve) se caractérise par des APRE grandes par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est maximale et s'augmente jusqu'à 700mm.

1.3.4 Selon l'humidité

Nous disposons dans chaque ligne de la base de données, d'une série temporelle (Saison agricole) observée sur observée sur 365 jours, comme marqué dans le tableau suivant :

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	4.88	4.70	4.88	5.37	6.04	...	6.77	7.26	7.69	6.84	7.51
1982	7.08	6.16	5.31	4.94	6.10	...	6.16	5.55	5.98	5.86	4.88
1983	4.58	5.55	4.58	3.91	3.85	...	7.14	5.80	4.64	4.39	6.47
1984	7.02	7.51	6.84	6.53	4.52	...	5.80	5.43	5.31	4.58	4.39
1985	4.76	5.25	5.00	4.76	6.47	...	6.29	6.23	6.35	6.10	5.98

Tableau 11: Base de données de l'humidité (cas agrumes)

Sortie python

Par la même démarche de l'AGDD nous étudions le clustering de l'humidité

Jour-Mois Année	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
1981	0.145392	0.133106	0.145392	0.178840	0.224573	...	0.274403	0.307850	0.337201	0.279181	0.324915
1982	0.295563	0.232765	0.174744	0.149488	0.228669	...	0.232765	0.191126	0.220478	0.212287	0.145392
1983	0.124915	0.191126	0.124915	0.079181	0.075085	...	0.299659	0.208191	0.129010	0.111945	0.253925
1984	0.291468	0.324915	0.279181	0.258020	0.120819	...	0.208191	0.182935	0.174744	0.124915	0.111945
1985	0.137201	0.170648	0.153584	0.137201	0.253925	...	0.241638	0.237543	0.245734	0.228669	0.220478

Sortie python

Tableau 12: Base de données après normalisation Min-Max de l'humidité (cas agrumes)

L'étape suivante c'est de choisir le nombre K de cluster, alors nous avons opté à utiliser la méthode d'elbow avec l'algorithme Timeseries Kmeans et sa métrique DTW ainsi la contrainte de Sakoe Chiba avec un rayon égal 10 (voir la figure 37), vu que les agrumes passent d'un stade à autre en environ un mois donc nous avons fixé que le rayon sera la durée de stade divisée par 3

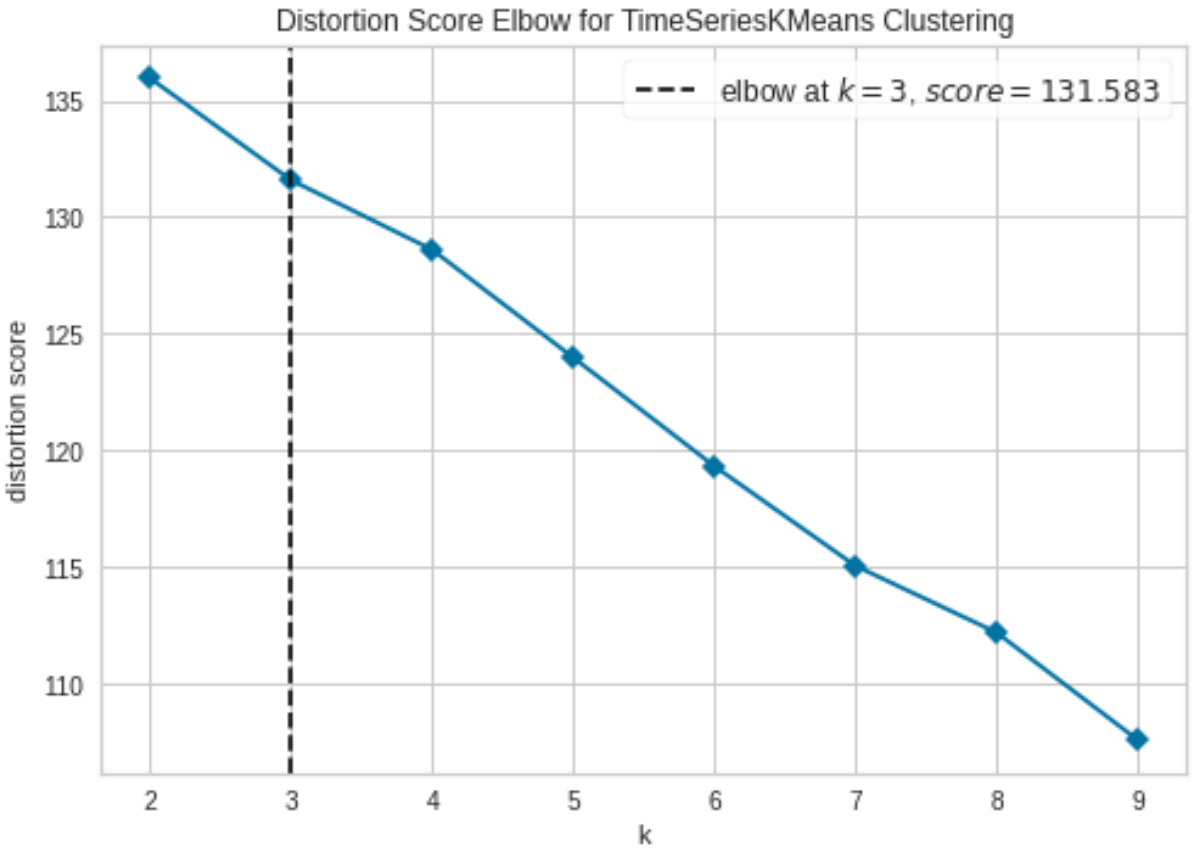


Figure 37: Le nombre de classe pour l'humidité cas agrumes

Sortie Python

Donc selon le graphique précèdent, nous avons conclu qu'il existe trois classes pour le cas de l'humidité. Nous avons réalisé ensuite à travers l'algorithme Timeseries k-means le clustering pour trouver les résultats suivants.

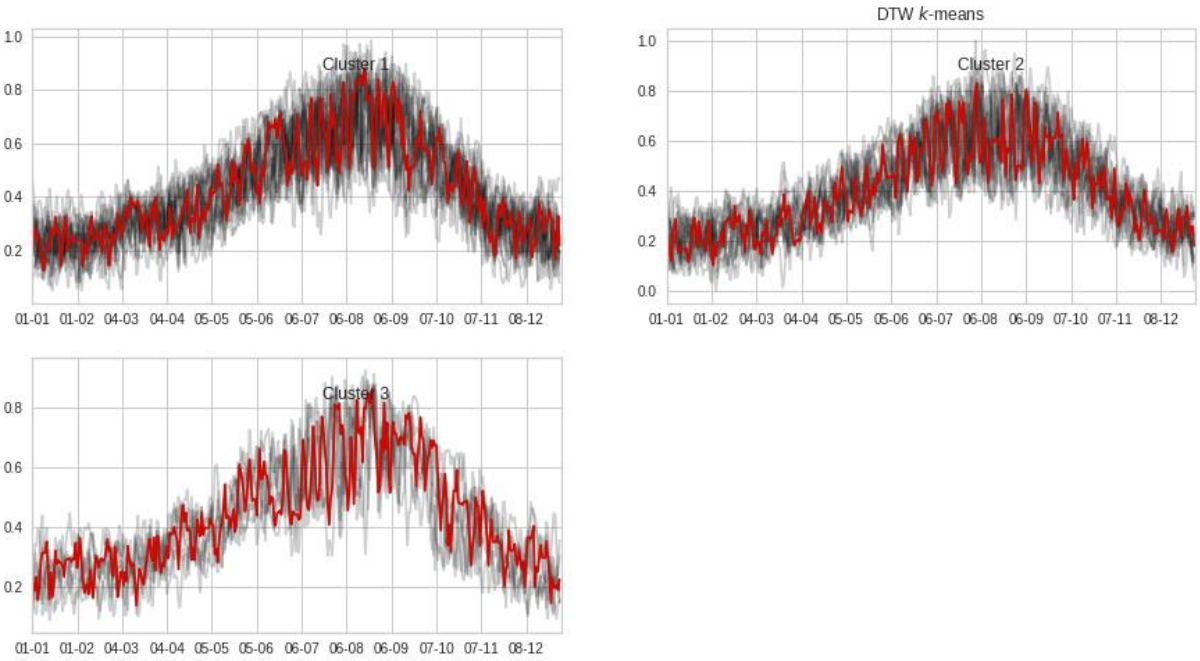


Figure 38: Groupement des années agricole par cluster selon l'humidité

Sortie Python

Les groupes trouvés sont les suivants :

Cluster 1 : [1981, 1987, 1990, 1991, 1994, 1995, 1996, 1999, 2001, 2002, 2003, 2004, 2006, 2009, 2010, 2011, 2013, 2017, 2018]
Cluster 2 : [1983, 1984, 1985, 1986, 1992, 1993, 2000, 2005, 2007, 2008, 2012, 2014, 2015, 2016, 2019]
Cluster 3 : [1982, 1988, 1989, 1997, 1998, 2020, 2021]

Le graph ci-dessous montre le groupement des années par cluster :

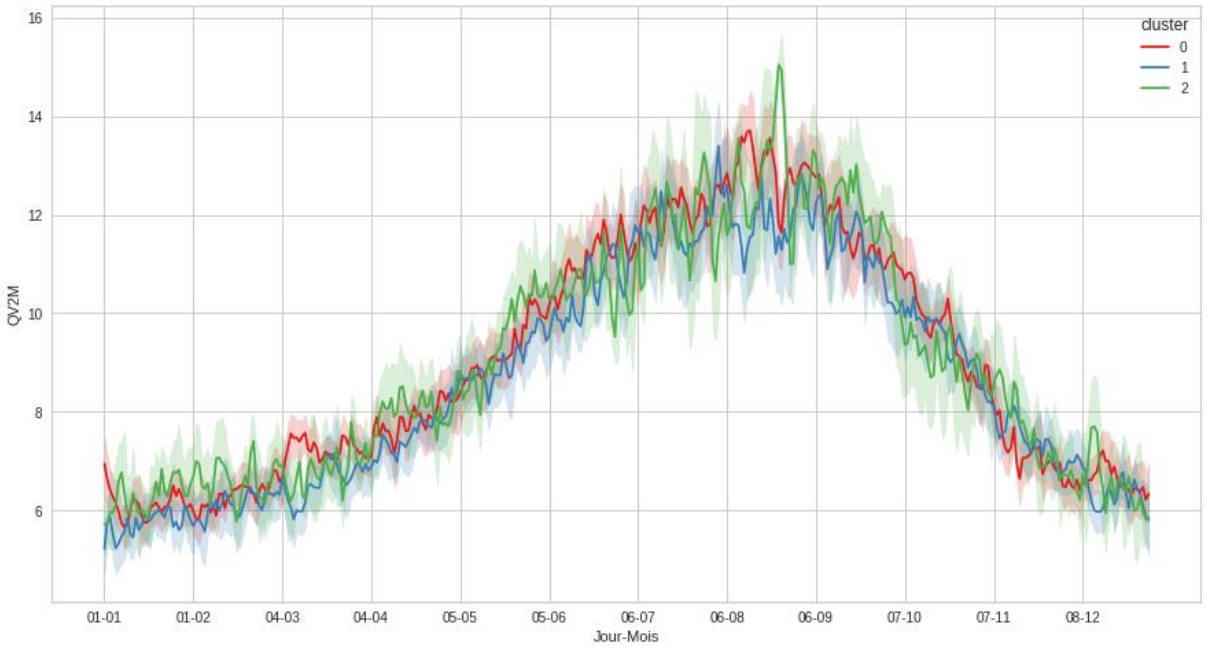


Figure 39: Résultat du clustering pour l'humidité (cas agrumes)

Sortie Python

Après le clustering, les résultats retenus ne sont pas vraiment significatifs, ils ne reflètent pas des différences entre les groupes, donc il y a une absence de spécificité.

- **La validation et qualité du clustering :**

Silhouette score :

Nous avons trouvé un score de silhouette de l'ordre de 0.01. Donc, nous pouvons conclure que les clusters ne sont pas bien séparés et donc ils sont de mauvaise qualité.

En effet, nous n'allons pas exploiter les résultats des clusters trouvées avec mauvaise qualité.

1.3.5 Selon la vitesse de vent

Nous disposons dans chaque ligne de la base de données, d'une série temporelle (Saison agricole) observée sur 365 jours :

Jour-Mois	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
Année											
1981	3.84	2.78	2.70	4.91	4.12	...	5.77	7.80	5.88	7.36	5.87
1982	3.83	3.58	3.64	2.70	2.74	...	3.23	3.16	2.20	3.24	3.48
1983	2.97	2.55	3.43	3.77	5.03	...	4.29	5.10	4.03	2.81	2.93
1984	2.27	1.38	3.91	6.05	3.88	...	7.48	4.32	4.25	3.70	1.59
1985	2.41	3.91	3.71	3.21	7.70	...	2.99	5.49	6.27	5.44	7.93

Tableau 13: Base de données de vitesse de vent

Sortie Python

Par la même démarche de l'AGDD, nous avons étudié le clustering de la vitesse de vent

Jour-Mois	01-01	02-01	03-01	04-01	05-01	...	27-12	28-12	29-12	30-12	31-12
Année											
1981	0.231942	0.146870	0.140449	0.317817	0.254414	...	0.386838	0.549759	0.395666	0.514446	0.394864
1982	0.231140	0.211075	0.215891	0.140449	0.143660	...	0.182986	0.177368	0.100321	0.183788	0.203050
1983	0.162119	0.128411	0.199037	0.226324	0.327448	...	0.268058	0.333066	0.247191	0.149278	0.158909
1984	0.105939	0.034510	0.237560	0.409310	0.235152	...	0.524077	0.270465	0.264848	0.220706	0.051364
1985	0.117175	0.237560	0.221509	0.181380	0.541734	...	0.163724	0.364366	0.426966	0.360353	0.560193

Tableau 14: Base de données après normalisation Min-Max cas vitesse de vent

Sortie Python

L'étape suivante c'est de choisir le nombre K de cluster, alors nous avons opté à utiliser la méthode d'elbow avec l'algorithme Timeseries Kmeans et sa métrique DTW ainsi la contrainte de Sakoe Chiba avec un rayon égal 10 (voir la figure 40), vu que les agrumes passent d'un stade à autre en environ un mois donc nous avons fixé que le rayon sera la durée de stade divisée par 3.

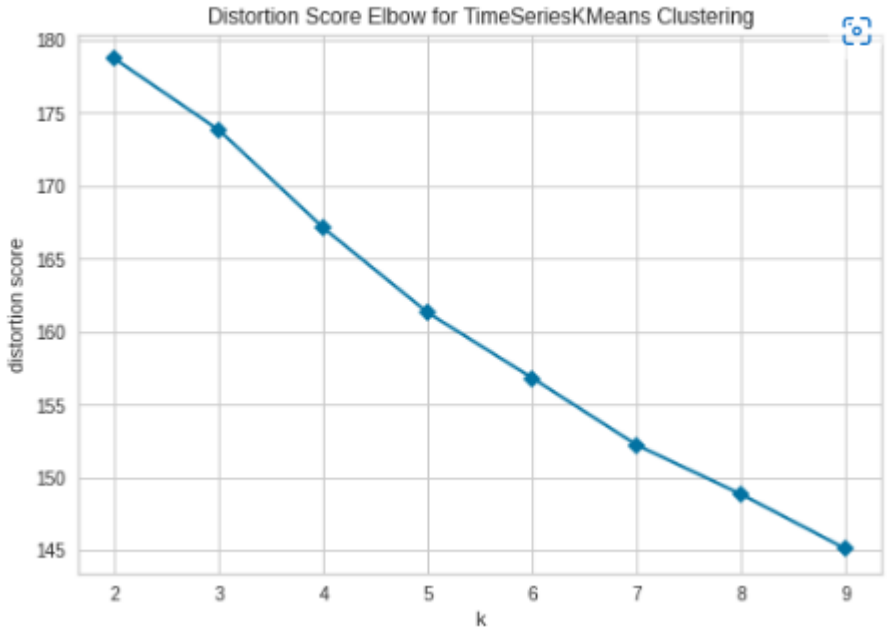


Figure 40: Le nombre de classe pour l'humidité cas agrumes

Sortie Python

Le graphique suivant montre l'absence d'un coude, et cela indique que toutes les années se groupent dans le meme cluster à cause du grand nombre de fluctuations. Nous utlisons aussi le indice de silhoutte et nous avons trouvé que cet indice est près de zero.

En effet, nous n'allons pas réaliser ce clustering pour éviter les résultats des clusters avec mauvaise qualité.

1.4Clustering des saisons agricoles pour la culture : Blé

1.4.2 Selon l'AGDD

Nous disposons dans chaque ligne de la base de données, d'une série temporelle (saison agricole) observée sur 365 jours :

Jour-Mois	01-10	02-10	03-10	04-10	05-10	...	26-06	27-06	28-06	29-06	30-06
Saison agricole											
1981-1982	16.015	31.515	46.130	61.130	78.015	...	3193.775	3209.370	3226.625	3246.275	3268.565
1982-1983	16.745	34.125	51.390	68.375	84.710	...	2956.540	2971.775	2987.275	3003.265	3018.815
1983-1984	16.585	34.335	53.185	71.845	92.545	...	3072.515	3088.825	3104.495	3120.070	3135.080
1984-1985	15.810	31.210	49.060	65.900	79.365	...	2971.225	2989.575	3007.200	3024.980	3043.010
1985-1986	21.910	40.550	58.510	75.395	91.890	...	2982.320	2998.750	3015.225	3030.595	3046.280

Tableau 15: Base de données de l'AGDD (cas Blé)

Sortie Python

Afin de réaliser une bonne comparaison entre nos séries temporelles, il faut normaliser la base de données entre 0 et 1, pour ce faire nous avons utilisé l'outil de la normalisation Min-Max dont on soustrait en toute observation à l'instant i le minimum de toute la base de données et on divise par la différence entre le maximum et le minimum.

Jour-Mois	01-10	02-10	03-10	04-10	05-10	...	26-06	27-06	28-06	29-06	30-06
Saison agricole											
1981-1982	0.001182	0.005749	0.010056	0.014476	0.019452	...	0.937599	0.942194	0.947279	0.953070	0.959638
1982-1983	0.001397	0.006518	0.011606	0.016611	0.021425	...	0.867691	0.872180	0.876748	0.881460	0.886042
1983-1984	0.001350	0.006580	0.012135	0.017634	0.023733	...	0.901866	0.906672	0.911290	0.915880	0.920303
1984-1985	0.001121	0.005659	0.010919	0.015882	0.019850	...	0.872018	0.877426	0.882619	0.887859	0.893172
1985-1986	0.002919	0.008412	0.013704	0.018680	0.023540	...	0.875288	0.880129	0.884984	0.889513	0.894135

Tableau 16: Base de données après normalisation Min-Max (cas Blé)

Sortie Python

L'étape suivante c'est de choisir le nombre K de cluster, asuivant méthode elbow avec l'algorithme Timeseries Kmeans et sa métrique DTW ainsi la contrainte de sakoe chiba avec un rayon égal 10, vu que les agrumes passent d'un stade à autre environ un mois dans nous avons fixe que le rayon sera la durée de stade divisée par 3.

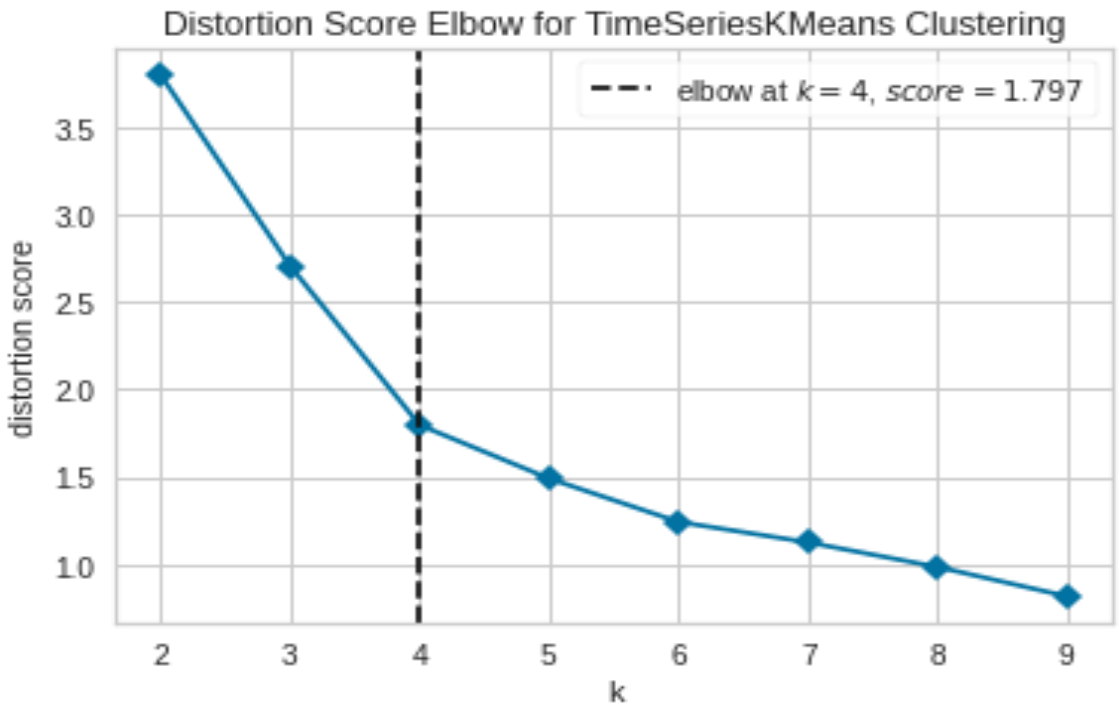


Figure 41: Le nombre de classe pour l'AGDD (cas Blé)

Sortie Python

Nous avons constaté que pour le cas de Blé, nous avons quatre classes, nous réalisons ensuite à travers Timeseies kmeans

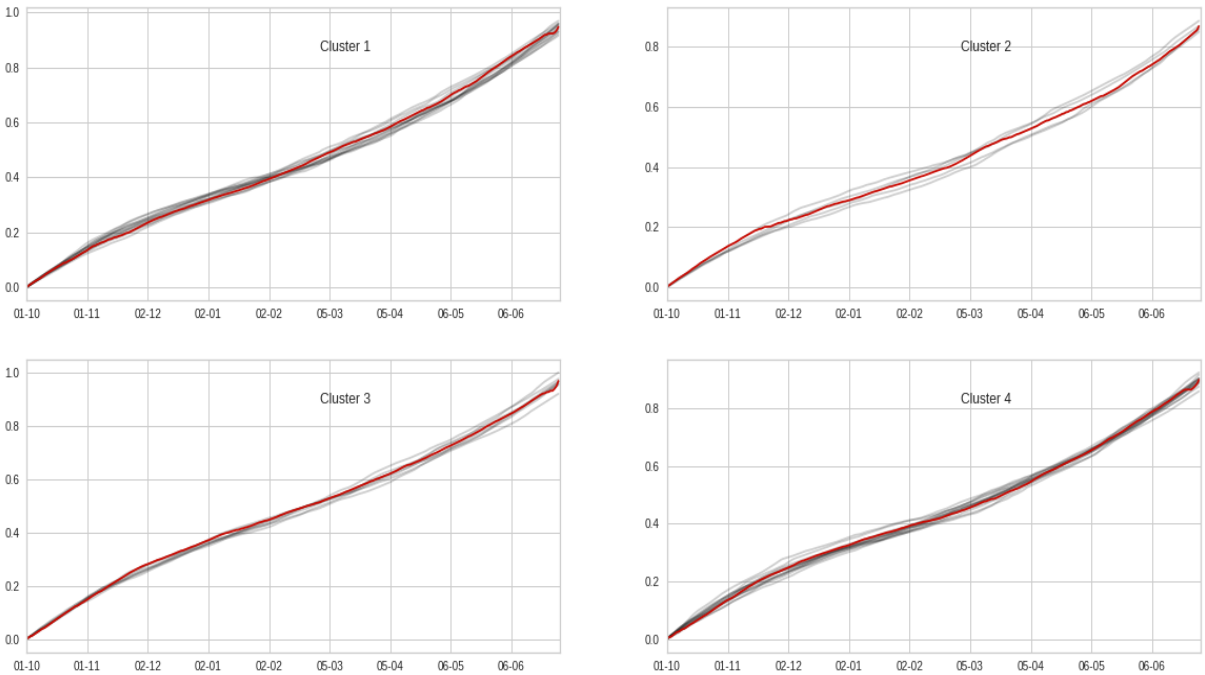


Figure 42: Groupement des années agricole par cluster selon l'AGDD (cas Blé)

Sortie Python

Cluster 1 : ['1986-1987', '1994-1995', '1996-1997', '1999-2000', '2000-2001', '2002-2003', '2006-2007', '2007-2008', '2010-2011', '2013-2014', '2014-2015', '2016-2017', '2019-2020']
Cluster 2 : ['1982-1983', '1990-1991', '1992-1993', '1993-1994', '2008-2009']
Cluster 3 : ['1981-1982', '1983-1984', '1989-1990', '1995-1996', '1997-1998', '2001-2002', '2009-2010', '2015-2016']
Cluster 4 : ['1984-1985', '1985-1986', '1987-1988', '1988-1989', '1991-1992', '1998-1999', '2003-2004', '2004-2005', '2005-2006', '2011-2012', '2012-2013', '2017-2018', '2018-2019', '2020-2021']

Le graphique ci-dessous montre le groupement des années par cluster :

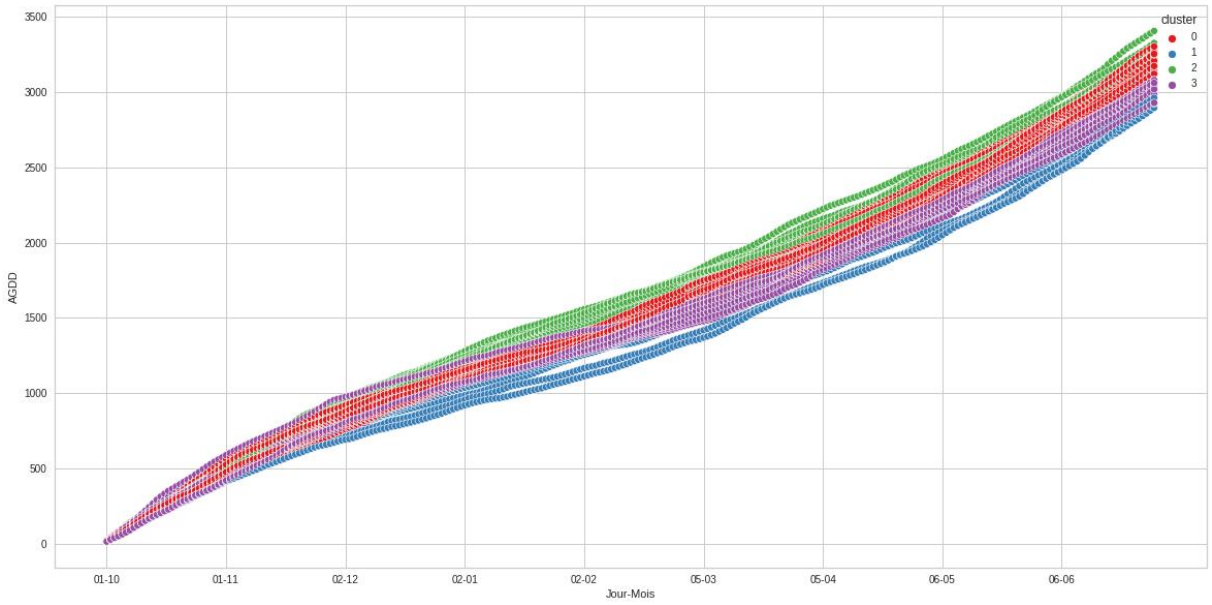


Figure 43: Résultat du clustering pour AGDD (cas Blé)

Sortie Python

- **La validation et qualité du clustering :**

Silhouette score :

Pour ce faire nous utilisons deux méthodes celle de silhouette, cette dernière montre que tant que l'indice est proche de 1 nos années en clusters sont bien positionnées.
Nous avons trouvé que silhouette score = 0.52, il est moyennement bon et signifie que nos clusters sont moyennement séparés.

Indice Calinski Harabasz :

Cet indice aussi de son côté montre que le nombre optimal de classes est 4, puisqu'il contient le plus grand score de Calinski Harabasz (un score égale à 42.102) parmi toutes les cas k possible comme indiquée dans la figure ci-dessous.

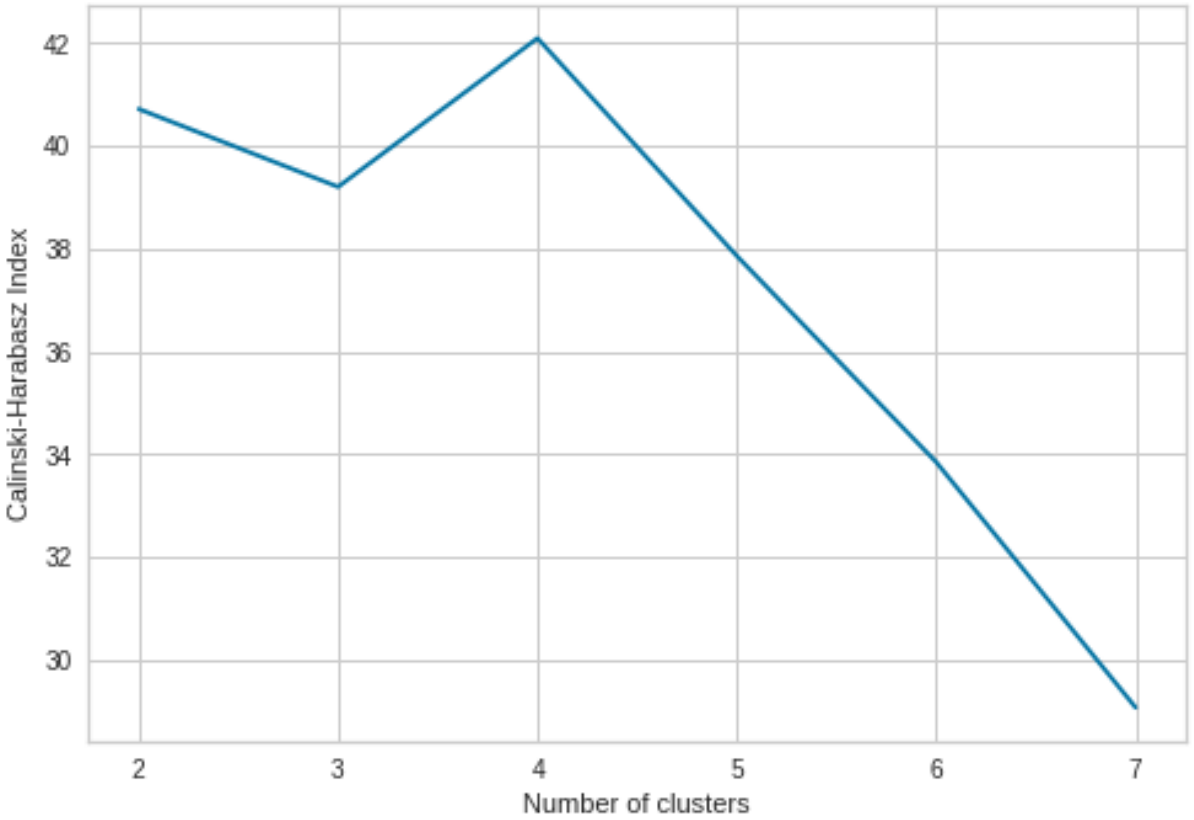


Figure 44: Score de Calinski Harabasz selon les différents K possible de cluster

Sortie Python

Donc nous adoptons 4 classes :

Chaque groupe d'année se caractérise par une température :

- Le premier groupe se caractérise par des AGDD grandes par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est maximale et s'augmente jusqu'à 3500.

- Le deuxième groupe se caractérise par une AGDD minimale par rapport aux autres groupes durant toute l'année, ainsi que sa valeur au niveau de la fin d'année est minimale et diminue jusqu'à moins de 2800.

- Le troisième groupe se caractérise par une AGDD moyenne par rapport aux autres groupes durant toute la saison agricole ainsi que sa valeur au niveau de la fin d'année est moyenne et s'augmente jusqu'à 3200.

- Le quatrième groupe se caractérise par une AGDD moyenne mais faible que celle du groupe 3 durant toute la saison agricole, ainsi que sa valeur au niveau de la fin d'année est moyenne et s'augmente jusqu'à 3200.

1.4.3 Selon l'APRE

Nous disposons dans chaque ligne de la base de données, d'une série temporelle (Saison agricole) observé sur 365 jours :

Jour-Mois	01-10	02-10	03-10	04-10	05-10	...	26-06	27-06	28-06	29-06	30-06
Saison agricole											
1981-1982	0.22	1.26	1.27	1.34	1.47	...	475.55	475.55	475.58	475.58	477.08
1982-1983	0.00	0.00	0.00	0.00	0.00	...	294.00	294.00	294.00	294.00	294.00
1983-1984	0.00	0.00	0.00	0.00	0.00	...	345.01	345.01	345.01	345.01	345.01
1984-1985	0.25	0.29	0.43	1.81	2.28	...	298.62	298.62	298.62	298.62	298.62
1985-1986	0.00	0.07	0.07	0.07	0.07	...	329.91	329.91	329.91	329.91	329.91

Tableau 17: Base de données de l'APRE (cas blé)

Sortie Python

Par la même démarche de l'AGDD nous étudions le clustering de APRE :

Jour-Mois	01-10	02-10	03-10	04-10	05-10	...	26-06	27-06	28-06	29-06	30-06
Saison agricole											
1981-1982	0.000275	0.001574	0.001586	0.001674	0.001836	...	0.594059	0.594059	0.594096	0.594096	0.595970
1982-1983	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.367266	0.367266	0.367266	0.367266	0.367266
1983-1984	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.430988	0.430988	0.430988	0.430988	0.430988
1984-1985	0.000312	0.000362	0.000537	0.002261	0.002848	...	0.373037	0.373037	0.373037	0.373037	0.373037
1985-1986	0.000000	0.000087	0.000087	0.000087	0.000087	...	0.412125	0.412125	0.412125	0.412125	0.412125

Tableau 18: Base de données après normalisation Min-Max cas APRE (cas blé)

Sortie Python

L'étape suivante c'est de choisir le nombre K de cluster, alors nous opté à utiliser la méthode elbow avec l'algorithme Time series k-means et sa métrique DTW ainsi la contrainte de Sakoe Chiba avec un rayon égal à 10.

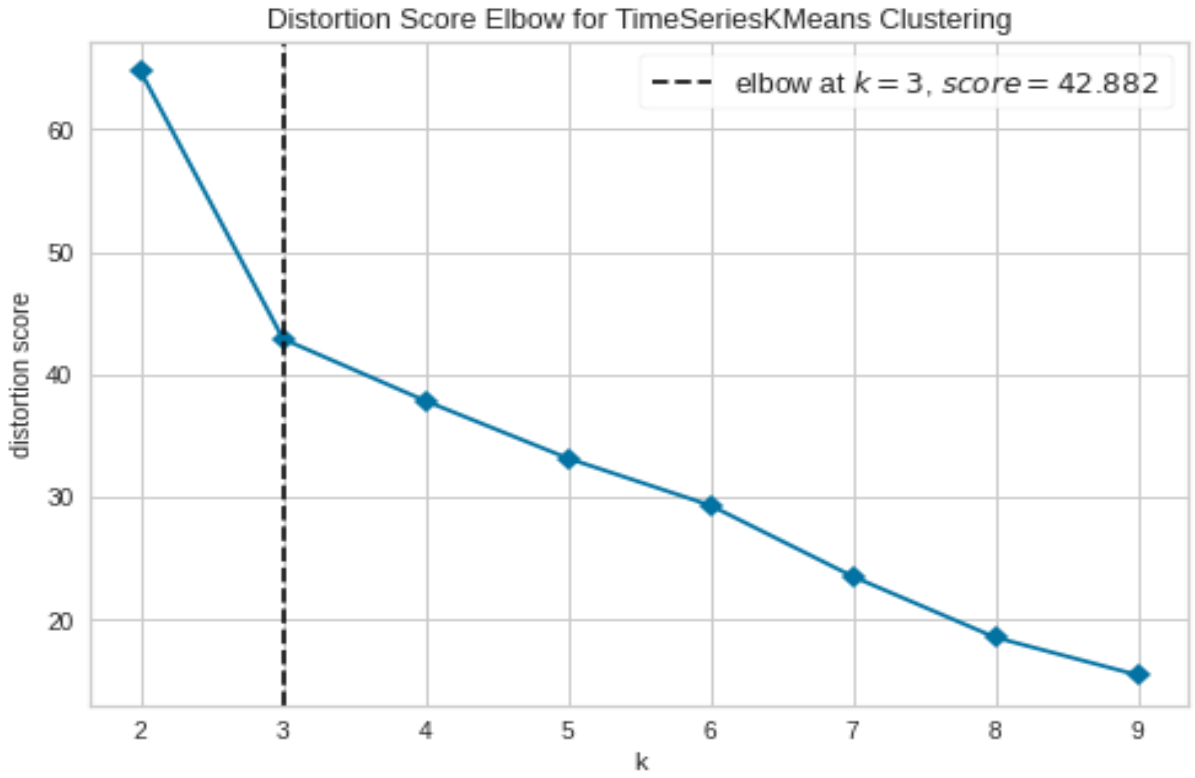


Figure 45: Le nombre de classe pour l'APRE (cas blé)

Sortie Python

D'après la figure ci-dessus nous constatons qu'on a 3 classes pour le cas d'APRE, nous réalisons ensuite à travers Timeseies kmeans

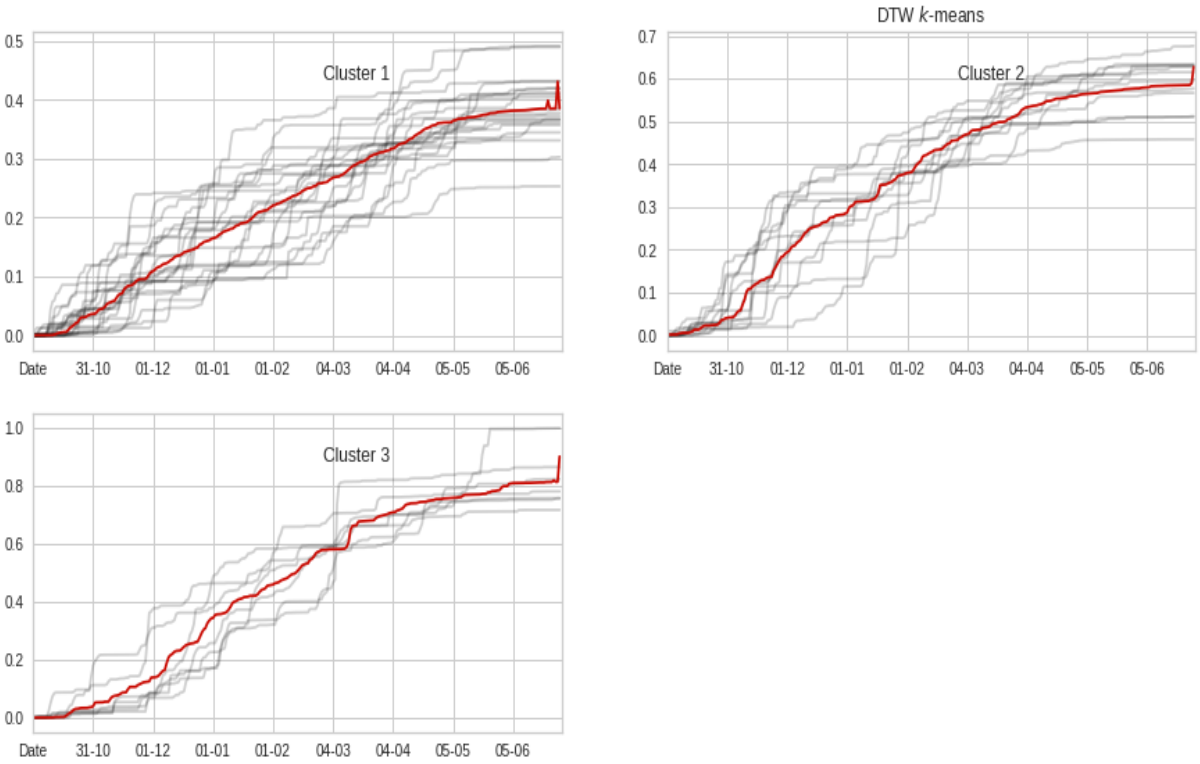


Figure 46: Groupement des années agricole par cluster selon l'APRE (cas blé)

Sortie Python

Cluster 1 : ['1982-1983', '1983-1984', '1984-1985', '1985-1986', '1988-1989', '1989-1990', '1991-1992', '1992-1993', '1994-1995', '1997-1998', '1998-1999', '1999-2000', '2000-2001', '2001-2002', '2004-2005', '2006-2007', '2007-2008', '2011-2012', '2013-2014', '2015-2016', '2019-2020']
Cluster 2: ['1981-1982', '1986-1987', '1987-1988', '1990-1991', '1993-1994', '2002-2003', '2003-2004', '2005-2006', '2012-2013', '2014-2015', '2016-2017', '2018-2019']
Cluster 3: ['1995-1996', '1996-1997', '2008-2009', '2009-2010', '2010-2011', '2017-2018', '2020-2021']

Le graphique ci-dessous montre le groupement des années par cluster :

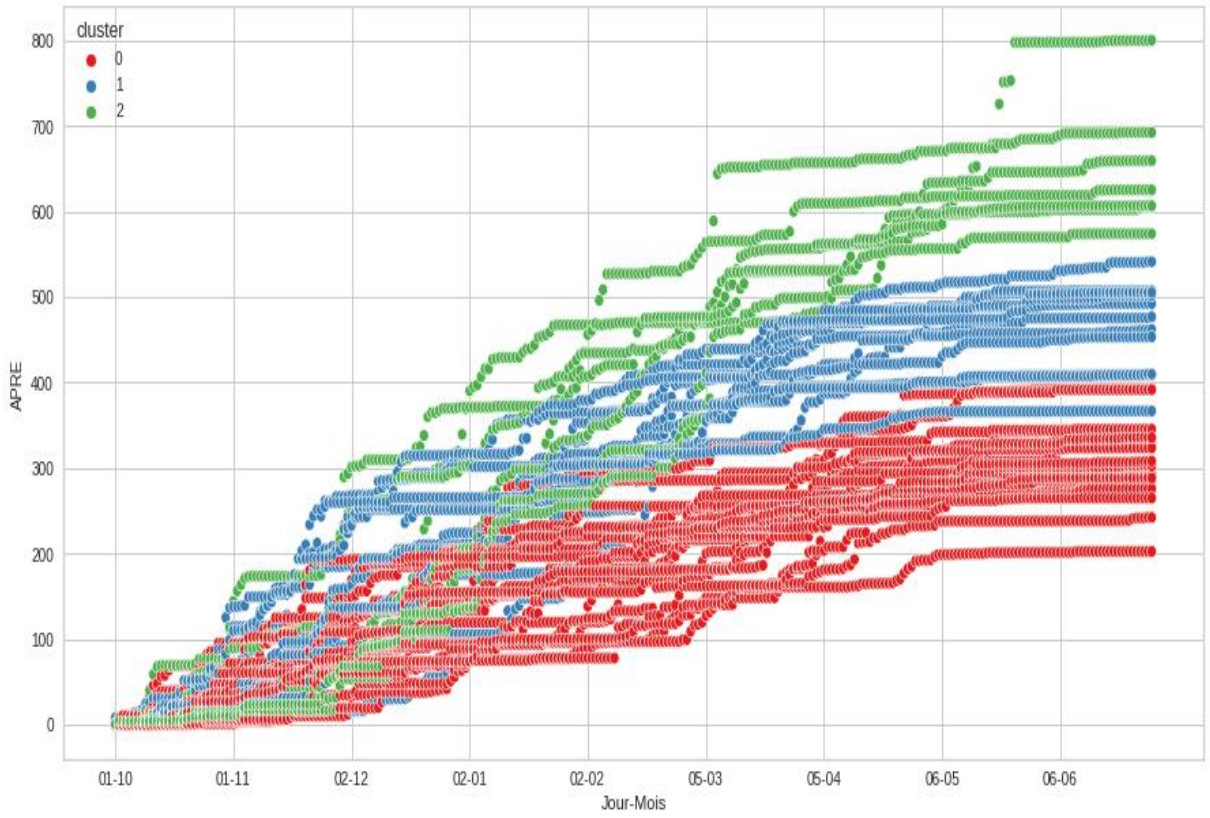


Figure 47: Résultat du clustering pour APRE (cas blé)

Sortie Python

- **La qualité du clustering :**

Silhouette score

Pour ce faire nous utilisons deux méthodes celle de silhouette, cette dernière montre que tant que l'indice est proche de 1 nos années en clusters sont bien positionnées.
Nous avons trouvé que silhouette score = 0.5, il est moyennement bon et signifie que nos clusters sont moyennement séparés.

Indice Calinski Harabasz

Cet indice aussi de son côté montre que le nombre optimal de classes est 3, puisqu'il contient le plus grand score de Calinski Harabasz (un score égal à 60,33) parmi toutes les cas k possible comme indiquée dans la figure ci-dessous.

Pour les deux autres variables (humidité et vitesse de vents) nous avons tester des clustériser mais comme le cas des agrumes, ces derniers ne s'adaptent pas à un clustering de qualité pour les saisons agricoles.

CHAPITRE IV

Exploitation de toutes les variables météorologiques à la fois dans la réalisation du clustering des saisons agricoles

CHAPITRE IV : Exploitation de toutes les variables météorologiques à la fois dans la réalisation du clustering des saisons agricoles.

Dans ce chapitre, nous allons utiliser les résultats validés des clusterings précédents, et composer une nouvelle base de données décrivant les caractéristiques des groupes déjà trouvées pour chaque variable, puis faire un nouveau clustering à partir de cette base de données, et enfin présenter les résultats.

1. Clustering combinant toutes les variables à la fois pour les agrumes

Afin de réaliser un clustering en exploitant toutes les variables météorologiques précédentes, nous avons adopté une approche dont nous étudions les classes trouvez dans chaque variable et nous donnons des caractéristiques spécifiques à chaque groupe dans chaque variable.

1.1 Pour l'AGDD (cas agrumes)

L'approche que nous avons suivie est, tout d'abord, regrouper les années qui se ressemblent dans chaque groupe et ensuite sélectionner le maximum de toutes les colonnes, et puisque dans l'AGDD nous utilisons le cumul alors le maximum sera la valeur de la dernière colonne pour chaque année en chaque cluster.

Cas du premier cluster :

Jour-Mois Année	01-01	02-01	...	cluster	Max
1989	0.000	0.000	...	0	2391.725
1994	2.395	4.590	...	0	2384.675
2001	1.745	3.175	...	0	2401.500
2003	1.650	5.355	...	0	2388.170
2006	0.000	0.000	...	0	2468.325
2011	0.680	0.680	...	0	2374.210
2012	0.000	0.000	...	0	2457.215
2014	0.000	0.000	...	0	2428.940
2015	0.000	0.000	...	0	2463.305
2016	2.165	4.365	...	0	2365.420
2017	0.000	0.000	...	0	2433.070
2020	0.000	0.000	...	0	2519.630
2021	0.000	0.000	...	0	2442.735

Cas du deuxième cluster

Jour-Mois Année	01-01	02-01	...	cluster	Max
1981	0.000	0.000	...	1	2139.560
1982	0.000	0.870	...	1	2029.900
1984	0.145	0.295	...	1	1977.370
1986	1.235	2.430	...	1	2043.845
1991	1.080	1.720	...	1	1996.710
1992	0.000	0.000	...	1	1964.455
1993	0.000	0.000	...	1	1898.195
1996	4.680	6.930	...	1	2016.070
2007	0.465	0.465	...	1	2105.595
2013	0.000	0.000	...	1	2090.635

Cas du dernier cluster

Jour-Mois Année	01-01	02-01	...	cluster	Max
2010	0.455	1.180	...	3	2284.415
2018	1.385	2.915	...	3	2141.580

Après ce traitement, nous avons calculé, pour chaque cluster, pour chaque cluster, nous calculons la moyenne du maximum de l'AGDD pour enfin désigner à toutes ces années une AGDD moyenne qui caractérisent toutes les années agricole au sein du cluster à la fois.

Cas du premier groupe de l'AGDD : nous avons trouvé que la moyenne des AGDD pour le cluster 1 égale 2036.719.

Ensuite avec la même façon nous appliquons cela à tous les trois clusters trouver en AGDD comme le tableau suivant montre :

Année	La moyenne de l'AGDD pour chaque cluster	cluster
1981	2036.719545	1
1982	2036.719545	1
1983	2276.135000	2
1984	2036.719545	1
1985	2276.135000	2
1986	2036.719545	1
1987	2276.135000	2
1988	2276.135000	2
1989	2424.532308	0
1990	2276.135000	2
1991	2036.719545	1

Tableau 19: Caractérisation de chaque année appartenant à un cluster par une AGDD moyenne (cas agrumes)

Sortie python

1.2 Pour l'APRE (cas agrumes)

Pour l'APRE nous avons appliqué la même méthode puisqu'elle désigne aussi un cumul, passant d'abord par détecter les maximums et puis calculer la moyenne dans chaque groupe.

Année	La moyenne de l'APRE pour chaque cluster	cluster
1981	423.772500	2
1982	471.521538	0
1983	303.960000	1
1984	423.772500	2
1985	303.960000	1
1986	471.521538	0
1987	423.772500	2
1988	423.772500	2
1989	423.772500	2
1990	471.521538	0
1991	471.521538	0

Tableau 20 : Caractérisation de chaque année appartenant à un cluster par une APRE moyenne (cas agrumes)

Ensuite, nous rassemblons ces résultats dans un même tableau :

Année	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster
1981	2036.719545	423.772500
1982	2036.719545	471.521538
1983	2276.135000	303.960000
1984	2036.719545	423.772500
1985	2276.135000	303.960000
1986	2036.719545	471.521538
1987	2276.135000	423.772500
1988	2276.135000	423.772500
1989	2424.532308	423.772500

Tableau 21: Tableau contenant les données des moyennes de l'AGDD et l'APRE pour chaque cluster (cas agrumes)

A partir du tableau ci-dessus, nous pouvons observer des années qui étaient dans les mêmes groupes soit dans l'AGDD ou dans L'APRE, donc forcément, elles ont des caractéristiques similaires, pour maintenir les résultats de ressemblances pour toutes les années, il faut passer par un clustering.

L'étape suivante est la normalisation de la base de données vu que nos variables ont des mesures différentes, et pour ce faire nous allons utiliser l'outil de normalisation MinMax.

Les résultats obtenus sont représentés dans le tableau suivant :

Année	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster
1981	0.000000	0.329944
1982	0.000000	0.461437
1983	0.662408	0.000000
1984	0.000000	0.329944
1985	0.662408	0.000000
1986	0.000000	0.461437
1987	0.609629	0.329944
1988	0.609629	0.329944
1989	1.000000	0.329944
1990	0.609629	0.461437
1991	0.000000	0.461437

Tableau 22: Base de données après normalisation Min-Max cas APRE (cas agrumes)

1.1.1 Clustering avec K-means (cas agrumes)

Après la préparation de la base de données, nous allons utiliser le clustering à travers l'algorithme standard de k-means avec la distance euclidienne.

Pour ce faire, on choisit tout d'abord le nombre K optimal de cluster :

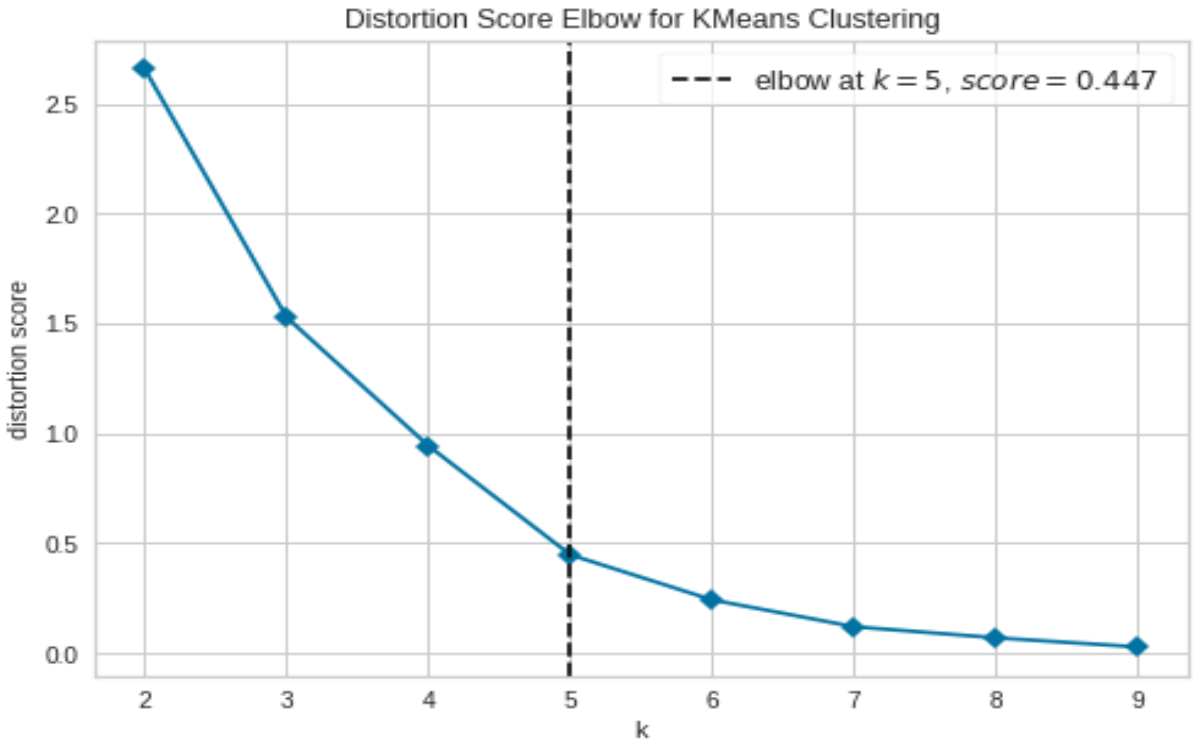


Figure 48: Le nombre de classe pour l'AGDD et l'APRE avec k-means (cas agrumes)

Sortie Python

D'après la figure ci-dessus, et après avoir appliqué la méthode d'elbow, nous constatons que nous avons 5 classes avec ressemblances :

Cluster 1 : [2010, 2018]
Cluster 2 : [1987, 1988, 1990, 1995, 1997, 1999, 2002, 2004, 2005, 2008, 2009]
Cluster 3 : [1981, 1982, 1984, 1986, 1991, 1992, 1993, 1996, 2007, 2013]
Cluster 4 : [1989, 1994, 2003, 2006, 2011, 2012, 2014, 2015, 2016, 2017, 2020, 2021]
Cluster 5 : [1983, 1985, 1998, 2000, 2001, 2019]

1.1.2 Clustering avec la méthode hiérarchiques CAH (cas agrumes)

Afin de tester les résultats obtenus par le clustering du partitionnement, nous avons utilisé le clustering qui se base sur la méthode hiérarchique.

Par la même méthode en k-means, nous avons appliqué la méthode d'elbow sur l'algorithme Agglomérative clustering, nous pouvons retourner que le nombre de classe égale 5.

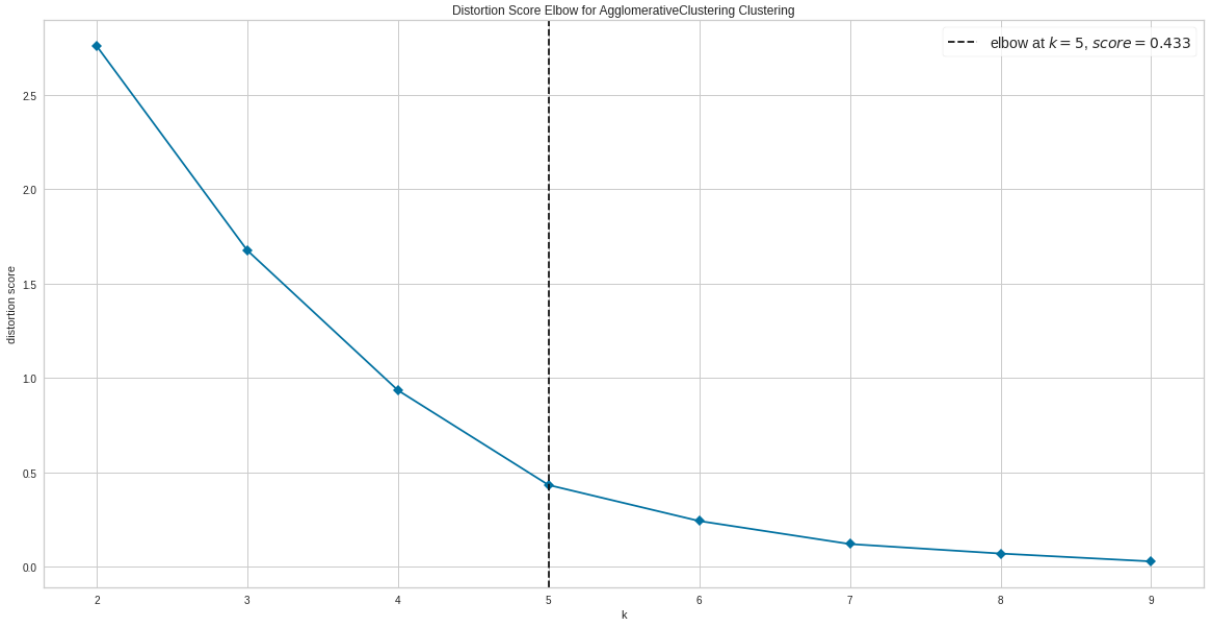


Figure 49: Le nombre de classe pour l'AGDD et l'APRE avec l'algorithme du clustering hiérarchique (cas agrumes)

Sortie Python

Le Dendrogramme suivant permet de visualiser les ressemblances du clustering final obtenu :

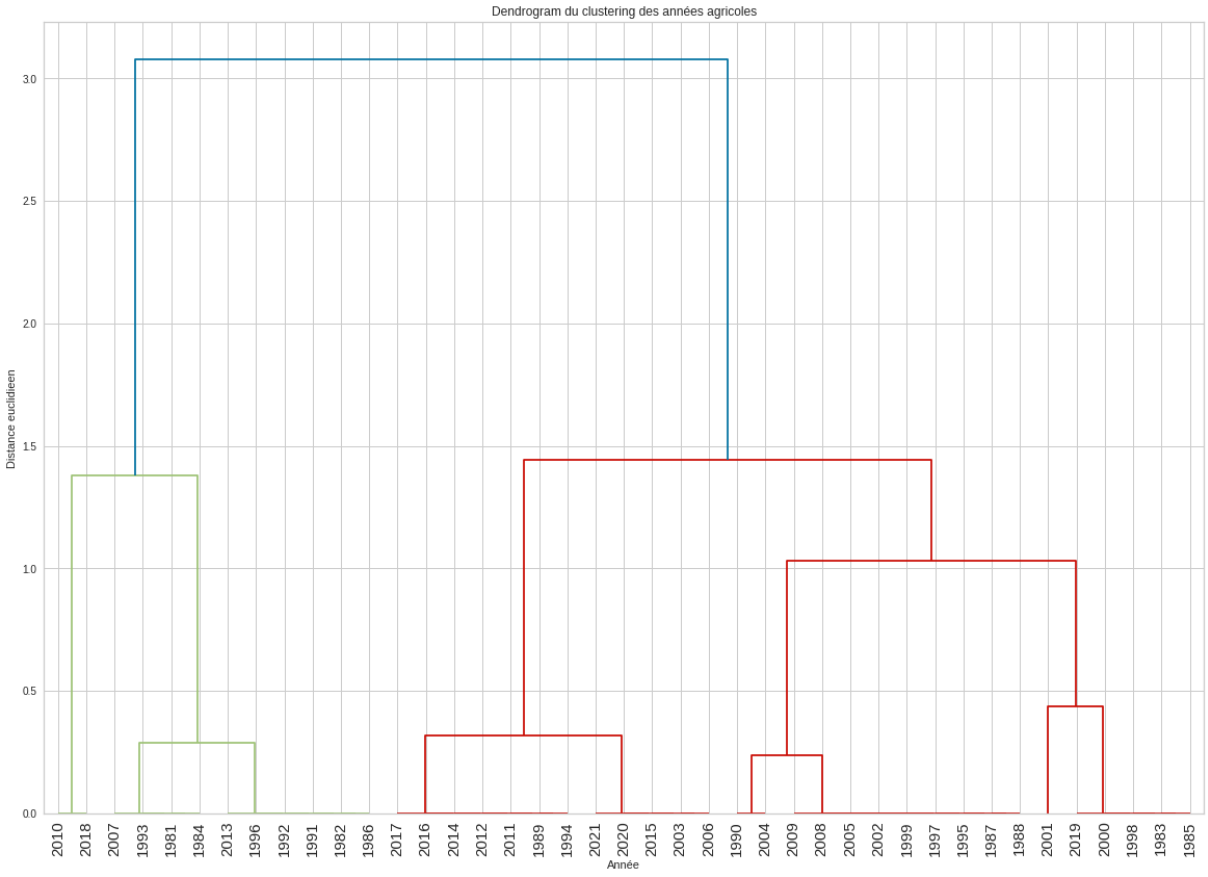


Figure 50: Dendrogramme du clustering final (cas agrumes)

Alors nos groupes de classes similaires sont les mêmes trouvées par k-means:

Cluster 1 : [1987, 1988, 1990, 1995, 1997, 1999, 2002, 2004, 2005, 2008, 2009]
Cluster 2 : [1981, 1982, 1984, 1986, 1991, 1992, 1993, 1996, 2007, 2013]
Cluster 3 : [1989, 1994, 2001, 2003, 2006, 2011, 2012, 2014, 2015, 2016, 2017, 2020, 2021]
Cluster 4 : [2010, 2018]
Cluster 5 : [1983, 1985, 1998, 2000, 2019]

Nous réalisons le tableau suivant tout en exprimant les caractéristiques de chaque groupe :

column	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster
metric	mean	mean
Cluster 1	2156.427273	667.09
Cluster 2	2276.135	432.454143
Cluster 3	2036.719545	452.421923
Cluster 4	2424.532308	443.667933
Cluster 5	2300.867885	303.96
Overall Dataset	2258.95439	433.248049

Tableau 23: Différences entre les clusters selon l'AGDD et l'APRE

- Le premier groupe est caractérisé par une moyenne d'AGDD égale 2156.42 C qui est ma plus faible par rapport aux autres groupes, mais une APRE maximale et égale 667.09 mm et donc des précipitations grandioses.

- Le deuxième groupe est caractérisé par une AGDD et une APRE relativement moyennes.

- Le troisième groupe est marqué avec des années agricoles de la plus faible valeur de l'AGDD, et donc les niveaux de la température au cours de ces années étaient faibles, et avec une valeur de l'APRE grandes.

- Le quatrième groupe est distingué par la plus grande valeur de l'AGDD, ce qui va forcément influencée les cultures d'agrumes, mais avec une APRE relativement moyenne.

- Le cinquième groupe est particularisé par une APRE minimale et donc les plus faibles précipitations, avec une AGDD tend vers 2300.86C ces années-là peuvent être dite comme des années de sécheresse.

2. Clustering combinant toutes les variables à la fois pour le blé

2.1 Pour AGDD (cas blé)

Afin de réaliser un clustering en exploitant toutes les variables précédentes de météo, nous avons adopté la même approche utilisée avec les agrumes.

Nous avons présenté chaque groupe de saisons avec ressemblances et le maximum de l'APRE.

Cas premier cluster :

Jour-Mois	01-10	02-10	...	cluster	Max
Saison agricole					
1986-1987	16.415	33.240	...	0	3145.290
1994-1995	13.285	27.200	...	0	3243.180
1996-1997	15.165	29.695	...	0	3272.675
1999-2000	16.690	32.605	...	0	3219.035
2000-2001	14.495	30.765	...	0	3194.110
2002-2003	16.350	33.310	...	0	3208.850
2006-2007	18.580	37.615	...	0	3121.695
2007-2008	16.755	32.585	...	0	3268.785
2010-2011	17.215	33.900	...	0	3263.535
2013-2014	17.910	36.175	...	0	3167.550
2014-2015	18.975	39.625	...	0	3172.725
2016-2017	16.730	33.170	...	0	3302.715
2019-2020	16.180	32.525	...	0	3253.270

Cas deuxième cluster

Jour-Mois	01-10	02-10	03-10	...	30-06	cluster	Max
Saison agricole							
1982-1983	16.745	34.125	51.390	...	3018.815	1	3018.815
1990-1991	18.280	35.570	51.320	...	2932.420	1	2932.420
1992-1993	16.315	32.530	48.850	...	2961.305	1	2961.305
1993-1994	12.005	25.995	39.465	...	2894.525	1	2894.525
2008-2009	15.730	31.455	45.370	...	2922.320	1	2922.320

Cas du troisième cluster

Jour-Mois	01-10	02-10	03-10	...	30-06	cluster	Max
Saison agricole							
1981-1982	16.015	31.515	46.130	...	3268.565	2	3268.565
1983-1984	16.585	34.335	53.185	...	3135.080	2	3135.080
1989-1990	18.900	36.640	53.025	...	3260.930	2	3260.930
1995-1996	16.535	34.700	51.335	...	3289.615	2	3289.615
1997-1998	16.640	33.860	50.485	...	3405.535	2	3405.535
2001-2002	17.730	35.295	52.270	...	3287.735	2	3287.735
2009-2010	15.845	32.115	48.745	...	3326.130	2	3326.130
2015-2016	17.180	34.640	52.600	...	3247.860	2	3247.860

Cas du dernier cluster

Jour-Mois	01-10	02-10	03-10	...	30-06	cluster	Max
Saison agricole							
1984-1985	15.810	31.210	49.060	...	3043.010	3	3043.010
1985-1986	21.910	40.550	58.510	...	3046.280	3	3046.280
1987-1988	17.890	34.660	50.820	...	2984.845	3	2984.845
1988-1989	15.220	33.885	50.825	...	3071.340	3	3071.340
1991-1992	12.845	27.755	42.910	...	3039.510	3	3039.510
1998-1999	14.750	28.770	42.685	...	3084.160	3	3084.160
2003-2004	17.550	34.655	52.635	...	3124.600	3	3124.600
2004-2005	19.860	39.915	59.175	...	3150.000	3	3150.000
2005-2006	17.570	34.440	52.700	...	3032.210	3	3032.210
2011-2012	19.120	38.505	57.185	...	3080.660	3	3080.660
2012-2013	15.875	32.315	49.345	...	3018.675	3	3018.675
2017-2018	17.410	36.030	56.020	...	2929.040	3	2929.040
2018-2019	18.050	36.870	55.715	...	3079.805	3	3079.805
2020-2021	14.380	27.760	39.485	...	3061.135	3	3061.135

Ensuite, nous avons calculons la moyenne de maximum de l'AGDD, afin de désigner à toutes ces années une l'APRE moyenne qui les caractérises toutes à la fois.

Ensuite avec la même façon nous appliquons cela à tous les quatre clusters trouvés en AGDD comme il est montré dans le tableau suivant :

Saison agricole	La moyenne de l'AGDD pour chaque cluster	cluster
1981-1982	3277.681250	2
1982-1983	2945.877000	1
1983-1984	3277.681250	2
1984-1985	3053.233571	3
1985-1986	3053.233571	3
1986-1987	3217.955000	0
1987-1988	3053.233571	3
1988-1989	3053.233571	3
1989-1990	3277.681250	2
1990-1991	2945.877000	1

Tableau 24: Caractérisation de chaque année appartenant à un cluster par une AGDD moyenne

2.1 Pour APRE (cas blé)

Pour l'APRE nous avons appliqué la même méthode puisqu'elle désigne aussi un cumul, passant d'abord par détecter les maximums et puis calculer la moyenne dans chaque groupe.

Saison agricole	La moyenne de l'APRE pour chaque cluster	cluster
1981-1982	469.379167	1
1982-1983	308.435238	0
1983-1984	308.435238	0
1984-1985	308.435238	0
1985-1986	308.435238	0
1986-1987	469.379167	1
1987-1988	469.379167	1
1988-1989	308.435238	0
1989-1990	308.435238	0
1990-1991	469.379167	1
1991-1992	308.435238	0

Tableau 25: Caractérisation de chaque année appartenant à un cluster par une APRE moyenne (cas blé)

Le tableau suivant montre le rassemblement les résultats trouvées auparavant

Saison agricole	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster
1981-1982	3277.681250	469.379167
1982-1983	2945.877000	308.435238
1983-1984	3277.681250	308.435238
1984-1985	3053.233571	308.435238
1985-1986	3053.233571	308.435238
1986-1987	3217.955000	469.379167
1987-1988	3053.233571	469.379167
1988-1989	3053.233571	308.435238
1989-1990	3277.681250	308.435238
1990-1991	2945.877000	469.379167
1991-1992	3053.233571	308.435238

Tableau 26: Tableau contenant les données des moyennes de l'AGDD et l'APRE pour chaque cluster (Cas blé)

Sortie Python

A partir du tableau nous pouvons observer des années qui étaient dans les même groupes soit dans l'AGDD ou dans L'APRE, donc forcément ils ont des caractéristiques identiques, pour maintenir les résultats de ressemblances pour toutes les saisons agricoles il faut passer par un clustering.

L'étape suivante est donc la normalisation de la base de données vu que nos variables ont des mesures différentes, et pour ce faire nous allons utiliser l'outil de normalisation MinMax, afin d'avoir une base de données sous la forme suivante :

Saison agricole	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster
1981-1982	1.000000	0.468167
1982-1983	0.000000	0.000000
1983-1984	1.000000	0.000000
1984-1985	0.323554	0.000000
1985-1986	0.323554	0.000000
1986-1987	0.819996	0.468167
1987-1988	0.323554	0.468167
1988-1989	0.323554	0.000000
1989-1990	1.000000	0.000000
1990-1991	0.000000	0.468167
1991-1992	0.323554	0.000000

Tableau 27: Base de données après normalisation Min-Max cas APRE (cas blé)

2.1.1 Clustering avec K-means (cas blé)

Nous allons par la suite utiliser le clustering à travers l'algorithme standard de k-means avec la distance euclidienne.

Pour ce faire, nous avons choisis le nombre k optimal de cluster par la méthode d'elbow :

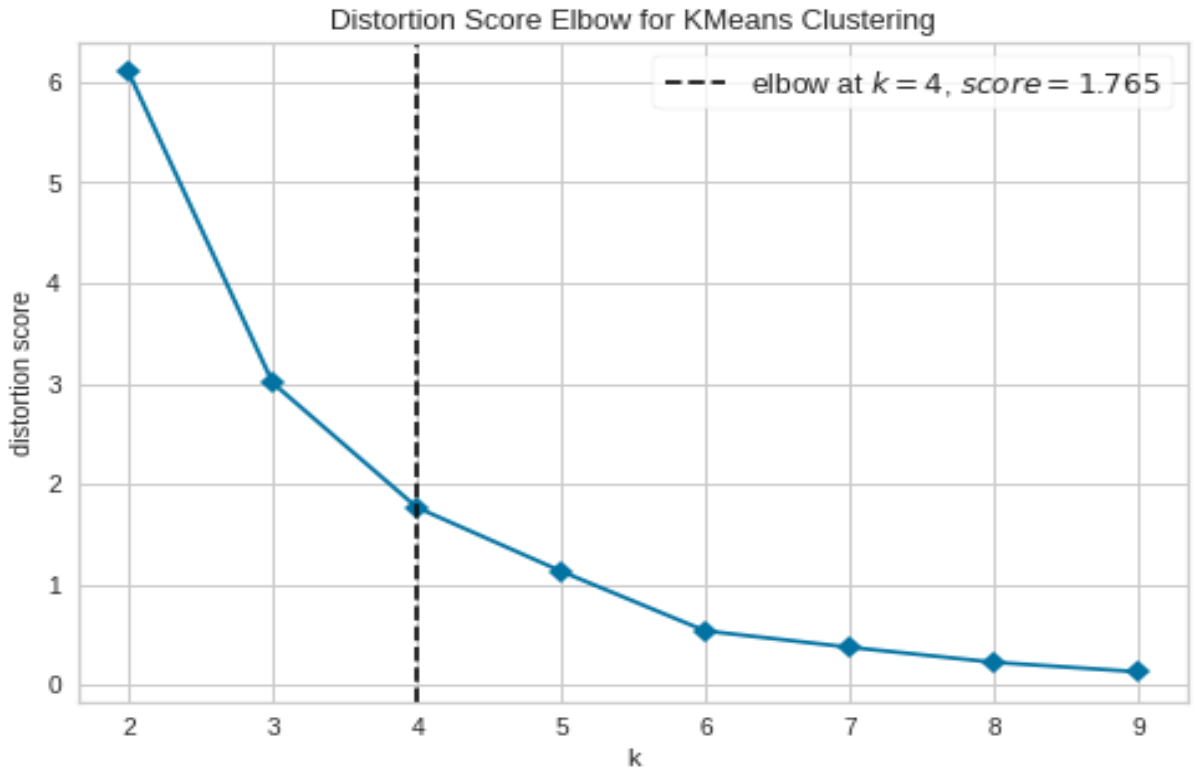


Figure 51 : Le nombre de classe pour l'AGDD et l'APRE avec l'algorithme du clustering hiérarchique (cas blé)

Sortie Python

Nous aurons enfin quatre groupes d'années avec ressemblances :

Cluster 1 : ['1982-1983', '1984-1985', '1985-1986', '1988-1989', '1991-1992', '1992-1993', '1998-1999', '2004-2005', '2011-2012']

Cluster 2 : ['1981-1982', '1986-1987', '1995-1996', '1996-1997', '2002-2003', '2009-2010', '2010-2011', '2014-2015', '2016-2017']

Cluster 3 : ['1987-1988', '1990-1991', '1993-1994', '2003-2004', '2005-2006', '2008-2009', '2012-2013', '2017-2018', '2018-2019', '2020-2021']

Cluster 4 : ['1983-1984', '1989-1990', '1994-1995', '1997-1998', '1999-2000', '2000-2001', '2001-2002', '2006-2007', '2007-2008', '2013-2014', '2015-2016', '2019-2020']

Afin de tester les résultats nous utilisons aussi le clustering qui se base sur la méthode hiérarchique.

2.1.2 Clustering avec la méthode hiérarchiques (cas blé)

Par la même méthode d'elbow appliquée sur l'algorithme Agglomérative clustering, nous pouvons retourner que le nombre de classe égale 4.

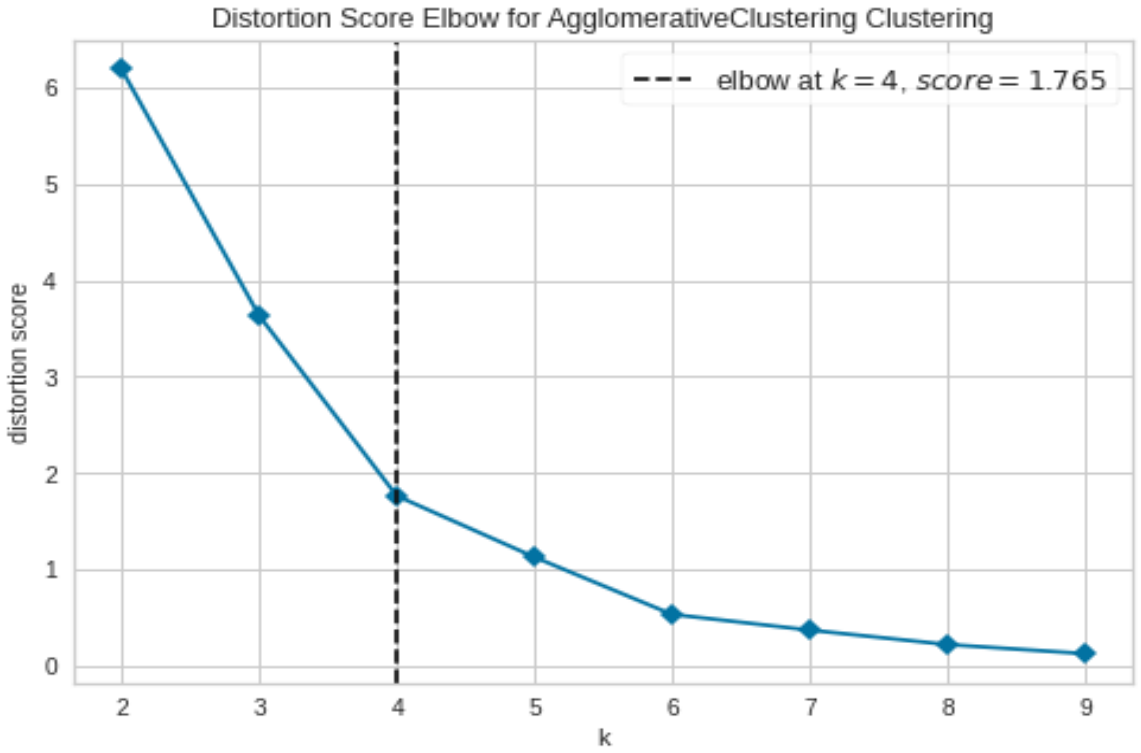


Figure 52: Le nombre de classe pour l'AGDD et l'APRE avec l'algorithme du clustering hiérarchique (cas blé)

Également à travers cette méthode de clustering, le nombre de classe optimal est quatre groupes.

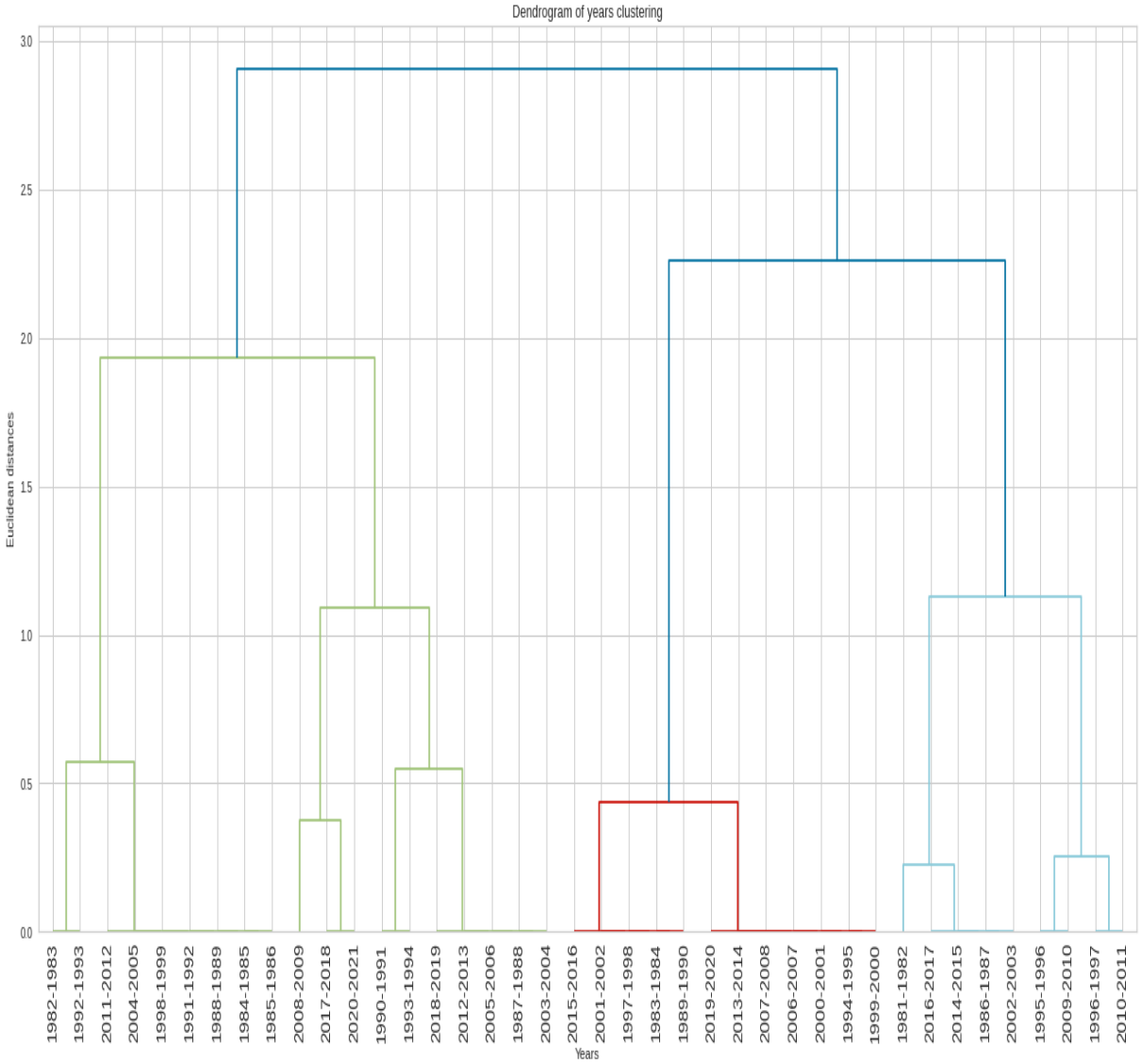


Figure 53: Dendrogramme du clustering final (cas blé)

Alors nos groupes de classes similaires sont les mêmes trouvées par k-means:

Cluster 1 : ['1982-1983', '1984-1985', '1985-1986', '1988-1989', '1991-1992', '1992-1993', '1998-1999', '2004-2005', '2011-2012']

Cluster 2 : ['1981-1982', '1986-1987', '1995-1996', '1996-1997', '2002-2003', '2009-2010', '2010-2011', '2014-2015', '2016-2017']

Cluster 3 : ['1987-1988', '1990-1991', '1993-1994', '2003-2004', '2005-2006', '2008-2009', '2012-2013', '2017-2018', '2018-2019', '2020-2021']

Cluster 4 : ['1983-1984', '1989-1990', '1994-1995', '1997-1998', '1999-2000', '2000-2001', '2001-2002', '2006-2007', '2007-2008', '2013-2014', '2015-2016', '2019-2020']Cluster 5 : [1983, 1985, 1998, 2000, 2019]

Le tableau ci-après montre les caractéristiques de chaque groupe :

column	La moyenne de l'AGDD pour chaque cluster	La moyenne de l'APRE pour chaque cluster	F
metric	mean	mean	
Cluster 1	3029.376556	308.435238	
Cluster 2	3237.86375	550.637315	
Cluster 3	3021.0266	524.228417	
Cluster 4	3242.840937	308.435238	
Overall Dataset	3138.238	416.879	

Tableau 28: Différences entre les clusters selon l'AGDD et l'APRE (cas blé)

- Le premier groupe est caractérisé par une moyenne d'AGDD égale 3029.36C qui est relativement faible par rapport aux autres groupes, et même une APRE minimale et égale 308.43mm.

- Le deuxième groupe est marqué avec des années agricoles de la plus grande valeur de l'APRE, et donc les niveaux de la précipitation au cours de ces années étaient importants, pourtant, l'AGDD était aussi grande avec une valeur égale 3237.02C.

- Le troisième groupe est distinguer par la plus faible valeur de l'AGDD, mais avec une APRE grandes ce qui va forcément influencer la culture du blé.

- Le quatrième groupe est particulariser par une AGDD maximale et une APRE minimale donc les plus faibles précipitations, et alors nous pouvons dire que les saisons agricoles noté dans le cluster 4 sont avec des caractéristiques de sécheresse.

Conclusion générale

L’objectif de notre projet était de clustériser les saisons agricoles en se basant sur les données météorologiques. Avant d’entamer notre étude, nous avons adopté une approche avec laquelle le clustering sera appliquée d’une part sur chaque variable toute seule, puis exploiter les résultats et conclure avec un clustering final. La première contrainte était le choix de la culture agricole, sa saison et la station géographique à considérer. La deuxième contrainte était le choix des variables météorologiques qui ont une influence directe sur notre culture agricole.

Dans la phase de la spécification des contraintes nous avons fixé de travailler sur deux types de cultures avec différentes saisons agricoles : Agrumes et blé, ensuite grâce aux API nous avons réussis à collecter les données météorologiques pour les deux villes sélectionnées, puis nous avons commencé notre prétraitement et l’organisation de la base de données pour les deux cultures, et visualiser nos saisons agricoles.

La tâche suivante était le clustering pour chaque variable, et puisque nous interagissons avec des séries temporelles, donc nous avons choisis un algorithme adapté au séries temporelles intitulé time series K-means.

Avec cet algorithme, notre mesure de similarité entre les séries était la déformation dynamique temporelle. Et comme résultats nous avons réussis de faire des clusters avec une qualité moyennement bonne pour des variables.

Ensuite, l’étape suivante était de prendre les résultats validés et les exploités dans un deuxième clustering, pour ce faire ; nous avons opté pour un deuxième avec deux algorithmes : hiérarchique CAH et K-means standard, et comme conclusion nous avons trouvé 5 groupes de cluster pour le cas des agrumes et quatre groupes pour celui du blé.

Enfin, nous avons essayé de comparer les caractéristiques de chaque cluster et donner quelques profiles à ces années qui se ressemblent.

Finalement, la similarité des saisons agricoles dépend principalement de la culture étudiée et la région géographique ou elle se produit, et comme nous avons trouvé les facteurs météorologiques qui alimente la différence en termes des saisons agricoles sont principalement les AGDD et APRE.

Bibliographie / Webographie

[1] DEPF Etudes, «Le secteur agricole marocain : Tendances structurelles, enjeux et perspectives de développement,» 2019.

[2] «Adoptez les API REST pour vos projets web,» [En ligne].

[3] «Comment le blé est-il semé, récolté, durant quelle saison,» 2022. [En ligne]. Available: https://www.vivescia.com/grand-angle/tous/cereale-quel-est-le-cycle-du-ble.

[4] M.-J. Huguet, Apprentissage non supervisé : Méthodes de Clustering, 2020-2021.

[5] E. H. T. Mohammed, Cours analyse de données.

[6] S. Chiba, «Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust».

[7] R. Tavenard, «An introduction to Dynamic Time Warping,» [En ligne]. Available: https://rtavenar.github.io/blog/dtw.html.

[8] R. Tavenard, «tslearn Documentation, Release 0.5.2,» [En ligne]. Available: https://readthedocs.org/projects/tslearn/downloads/pdf/latest/.

[9] The scikit-yb developers, «Clustering Visualizers : Elbow Method,» [En ligne]. Available: https://www.scikit-yb.org/en/latest/api/cluster/elbow.html.

[10] A. Kassambara, «Cluster Validation Statistics: Must Know Methods,» [En ligne]. Available: https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/.

[11] «Selecting the number of clusters with silhouette analysis on KMeans clustering,» [En ligne]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html.

[12] GeeksforGeeks, «Indice de silhouette,» [En ligne]. Available: https://selvaweddingphotography.com/fr/geeksforgeeks-fran%c3%a7ais/.

[13] GeeksforGeeks, «Calinski-Harabasz Index – Cluster Validity indices,» [En ligne]. Available: https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/.

[14] M. Sidyakov, «Calinski-Harabasz Index,» [En ligne]. Available: https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/.

[15] «Maroc Agriculture de Précision,» [En ligne]. Available: http://www.agriculturalsystem.net/agriculture-de-precision/.

[16] E. C. Emmanuel CHOISNEL, «AGROMÉTÉOROLOGIE : Influence du climat sur la production agricole,» [En ligne]. Available: https://www.universalis.fr/encyclopedie/agrometeorologie/1-influence-du-climat-sur-la-production-agricole/.

[17] GeeksforGeeks, «Calinski-Harabasz Index – Cluster Validity indices,» [En ligne]. Available: https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/.

Annexes

Annexe 1 : Outils utilisés

Google Colaboratory : C'est un produit de Google Research qui permet d'écrire et d'exécuter les codes python par le biais du navigateur. C'est un environnement adapté au machine learning et à l'analyse de donnée. Il est hébergé de notebooks Jupyter, ne nécessite aucune configuration et permet d'accéder gratuitement à des ressources informatiques.

Python : Python est le langage de programmation de choix pour les data scientists. C'est un langage facile à utiliser et d'une bonne disponibilité des bibliothèques. Nous allons utiliser dans ce travail plusieurs bibliothèques, pour évaluer et purifier les données et construire notre modèle.

Parmi les bibliothèques utilisées :

Pandas et Numpy pour la manipulation et l'organisation des matrices et tableaux multidimensionnels.

Matplotlib et Seaborn pour la visualisation des données.

ScikitLearn pour la modélisation et le clustering.

Tslearn pour la modélisation et le clustering mais spécialiser en séries temporelles.

Annexe 2 : Codes utilisés

- **Code des API pour la collecte de données des journalières à travers NASA POWER avec précisions de zone géographique désirée :**

```
import pandas as pd
import requests
import csv
locations = [(33.5731104, -7.5898434),(32.004262,-
6.578339),(34.15,6.40),(34.9280,-2.3281),(32.68, -
4.74), (33.57, 5.33)]
output = r""
url = r"https://power.larc.nasa.gov/api/temporal/daily/point?start
=1981&end=2021&longitude={longitude}&latitude={latitude}&community
=ag&parameters=T2M_MAX,T2M_MIN,PRECTOTCORR,QV2M,WS10M&format=csv&h
eader=False"

for latitude, longitude in locations:
     request = url.format(longitude=longitude, latitude=latitude)
    response = requests.get(url=request, verify=True,
timeout=30.00)
    open(f'Data_{latitude,longitude}.csv', 'wb').write(response.co
ntent)
```

- **Code d'organisation des saisons agricoles (cas blé)**

```
series=series[(series.index.month >= 10) | (series.index.month
<= 6)]
series=series[series.index >=dt(1981,9,30)]
series

years = series["Année"].unique()
series["Saison agricole"] = [0 for i in range(series.shape[0])]
for year in years:
  series.loc[(series.index >= dt(year, 10, 1)) & (series.index
<= dt(year+1, 6, 30)), "Saison agricole"] = f"{year}-{year+1}"
series=series[["T2M_MAX",'T2M_MIN','PRECTOTCORR','QV2M','WS10M'
,'Saison agricole']]
series
series=series.reset_index()
```

```
series['Jour-Mois']=series["Date"].dt.strftime('%d-%m')
series
```

- **Code la transformation de la base de données AGDD et cumul de la précipitation**

  - **Cas du des agrumes**

```
Tbase={'Agrumes': 13, 'Blé':6}

series[f"GDD des {list(Tbase.keys())[0]}"]=(series["T2M_MAX"]+s
eries["T2M_MIN"])/2-Tbase['Agrumes']

series['GDD des Agrumes'] = series['GDD des Agrumes'].clip(lowe
r = 0)
series['AGDD'] = series.groupby(['Année'])['GDD des Agrumes'].t
ransform(pd.Series.cumsum)
series['APRE'] = series.groupby(['Année'])['PRECTOTCORR'].trans
form(pd.Series.cumsum)
series.head(n=369)
#Affichage des données
df=series[['Année',"AGDD",'APRE','QV2M','WS10M',"Jour-Mois"]]
df.head(n=1523)
```

  - **Cas du blé**

```
Tbase={'Agrumes': 13, 'Blé':6}
series[f"GDD du {list(Tbase.keys())[1]}"]=(series["T2M_MAX"]+se
ries["T2M_MIN"])/2-Tbase['Blé']

series[f"GDD du {list(Tbase.keys())[1]}"] = series[f"GDD du {li
st(Tbase.keys())[1]}"].clip(lower = 0)
series['AGDD'] = series.groupby(['Saison agricole'])[f"GDD du {
list(Tbase.keys())[1]}"].transform(pd.Series.cumsum)
series['APRE'] = series.groupby(['Saison agricole'])['PRECTOTCO
RR'].transform(pd.Series.cumsum)
series.head(n=369)
#Affichage des données organisées
df=series[["Date","AGDD",'APRE','QV2M','WS10M','Jour-
Mois','Saison agricole']]
df=df.set_index('Saison agricole')
```

```
df
```

- **Code du pivotement de la base de données (AGDD)**

```
cols=df["Jour-Mois"].unique().tolist()
T= df.pivot_table(index="Année", columns="Jour-
Mois", values="AGDD")[cols[:-1]]
T.head(n=10)
```

- **Code de la méthode kelbow_visualizer (AGDD)**

```
from yellowbrick.cluster.elbow import kelbow_visualizer
metric_params = {"global_constraint":"sakoe_chiba", "sakoe_chib
a_radius":   10}
visualizer=kelbow_visualizer(TimeSeriesKMeans(random_state=5, m
etric='dtw', metric_params=metric_params), Tnorm, k=(2,10),loca
te_elbow=True,timings=False)
num_K= visualizer.elbow_value_
```

- **Code et sortie des prédiction avec l'algorithme TimeSeriesKMeans, avec la mesure 'DTW' et la contrainte de Sakoe chiba (AGDD)**

```
# On choisit la distance DTW et on fixe 'r' à 10
metric_params = {"global_constraint":"sakoe_chiba", "sakoe_chib
a_radius": 10}
# sakoe_chiba_radius=10
models = tslearn.clustering.TimeSeriesKMeans(n_clusters=3, metr
ic='dtw',random_state=5, metric_params=metric_params)
predictions = models.fit_predict(Tnorm)
print(predictions)
```

```
1 # On choisit la distance DTW et on fixe 'r' à 10
2 metric_params = {"global_constraint":"sakoe_chiba", "sakoe_chiba_radius": 10}
3 # sakoe_chiba_radius=10
4 models = tslearn.clustering.TimeSeriesKMeans(n_clusters=3, metric='dtw',random_state=5, metric_params=metric_params)
5 predictions = models.fit_predict(Tnorm)
6 print(predictions)
```

```
[1 1 2 1 2 1 2 2 0 2 1 1 1 0 2 1 2 2 2 2 0 2 0 2 2 0 1 2 2 2 0 0 1 0 0 0 0
 1 2 0 0]
```

- **Code des visualisations des clusters avec base données normalisé (AGDD)**

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15,8))
```

```
X_train = Tnorm.values
for yi in range(3):
    plt.subplot(2, 2, yi + 1)
    for xx in X_train[predictions == yi]:
        _index = T.columns.values
        n_indices = _index.shape[0]
        _index = [_index[i] for i in range(n_indices) if i%31==
0 ]
        plt.plot(xx.ravel(), "k-", alpha=.2)
        plt.xticks(ticks = [i for i in range(n_indices) if  i%3
1==0], labels = _index)
    plt.plot(models.cluster_centers_[yi].ravel(), "r-")
    plt.xlim(0, X_train.shape[1])
    # plt.ylim(-10, 10)
    plt.text(0.55, 0.85,'Cluster %d' % (yi + 1),
                transform=plt.gca().transAxes)
    if yi == 1:
        plt.title("DTW $k$-means")
```

- **Code des visualisations des clusters dans le même graphique avec seaborn (AGDD)**

```
T["cluster"] = predictions
T.head()
T.insert(0, "Date", T.index)
T_melted = pd.melt(T, id_vars =  ["Date", "cluster"], value_nam
e="AGDD")
T_melted.reset_index(inplace=True, drop = True)
import seaborn as sns
_index = Tnorm.columns.values
n_indices = _index.shape[0]
_index = [_index[i] for i in range(n_indices) if i%31==0 ]
plt.figure(figsize = (20, 10))
sns.scatterplot(x = "Jour-
Mois", y = "AGDD", hue = "cluster", data = T_melted, palette="S
et1")
flatui = ["#9b59b6", "#3498db", "orange"]
# sns.set_palette(flatui)
plt.xticks(ticks = [i for i in range(n_indices) if  i%31==0], l
abels = _index)
plt.show()
```

- **Code des métriques d'évaluations**

```
#The Silhouette Score
metric_params = {"global_constraint":"sakoe_chiba", "sakoe_chib
a_radius": 10}
tslearn.clustering.silhouette_score(Tnorm,predictions, metric="
dtw",random_state=5, metric_params=metric_params)
#L'indice de Calinski-Harabasz
#Grapique
results = {}
for i in range(2,8):
    metric_params = {"global_constraint":"sakoe_chiba", "sakoe_
chiba_radius": 10}
    TimeSeriesKMean =  tslearn.clustering.TimeSeriesKMeans(n_cl
usters=i, metric='dtw',random_state=10, metric_params=metric_pa
rams)
    labels = TimeSeriesKMean.fit_predict(Tnorm)
    db_index = calinski_harabasz_score(Tnorm, labels)
    results.update({i: db_index})
plt.plot(list(results.keys()), list(results.values()))
plt.xlabel("Number of clusters")
plt.ylabel("Calinski-Harabasz Index")
plt.show()
#Score
sklearn.metrics.calinski_harabasz_score(Tnorm,predictions )
```

- **Code de calcul des moyennes**

```
pd.set_option('max_columns', 6)
Tcarac = {i: T[T.cluster == i] for i in T.cluster}
for i in T.cluster :
    Tcarac[i]['Max']=Tcarac[i].iloc[:, 0:366].max(axis=1)
pd.set_option('max_columns', 6)
Tcarac

agdd_means=pd.DataFrame(agdd_means,columns = ['La moyenne de l\
'AGDD pour chaque cluster'],index=T.index)
agdd_means=agdd_means.join(T[["cluster"]])
agdd_means
agdd_means=pd.DataFrame(agdd_means,columns = ['La moyenne de l\
'AGDD pour chaque cluster'],index=T.index)
agdd_means=agdd_means.join(T[["cluster"]])
```

```
agdd_means
```

- **Code de k-means dans le clustering final**

```
models = KMeans(n_clusters=5,random_state=100)
predictions = models.fit_predict(dfscaled)
print(predictions)
```

- **Code de CAH dans le clustering final**

```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=5, affinity='euclide
an')
model.fit(dfscaled)
```

- **Code du dendrogramme dans le clustering final**

```
import scipy.cluster.hierarchy as sch
plt.rcParams["figure.figsize"] = [20, 14]
dendrogram = sch.dendrogram(sch.linkage(dfscaled, method  = "wa
rd"),leaf_rotation=90, leaf_font_size=15 ,show_contracted=True,
labels=dfscaled.index)
plt.title('Dendrogram du clustering des années agricoles')
plt.xlabel('Année')
plt.ylabel('Distance euclidieen')
plt.axhline(y=1000, c='black', lw=2, linestyle='dashed',linewid
th=10)

plt.show()
```