

## *Rapport d'évaluation*



# ***RainsBerryPy- Projet Météo***

**Promotion** : DataScientist – Octobre 2021.

**Participants** :

- Lionel BOTTAN
- Samuel GUERIN
- Julien COQUARD

## Table des matières

---

I.	Preprocessing - Notebook NB1.....	4
1.1	Observation du jeu de données initial .....	4
1.2	Ajout de variables supplémentaires.....	4
1.3	Gestion des valeurs manquantes .....	6
II.	DataViz – Notebook NB2 .....	7
2.1	Corrélations de RainTomorrow et Sunshine .....	7
2.2	Représentation cartographique .....	8
2.3	Influence pour la prévision de pluie .....	9
2.3.1	- Influence de certains critères (indépendamment du climat).....	9
2.3.2	- Influence de la pluie des jours précédents sur la pluie du lendemain.....	9
2.3.3	- Influence des vents sur la pluie .....	10
2.4	Distribution des températures au cours de l'année suivant le climat .....	11
III.	Modélisation de RainTomorrow – Notebooks NB3.1 et NB3.2.....	12
3.1	Outils pour améliorer les performances – Notebook NB3.1.....	12
3.1.1	- Équilibrage des classes .....	13
3.1.2	- Validation croisée pour détecter un problème de surapprentissage .....	14
3.1.3	- Comparaison des performances en modifiant les seuils de probabilités pour la détection de la classe 1.....	15
3.1.3	- Comparaison des performances pour différents traitements des valeurs manquantes ...	16
3.1.4	- Sélection de variables.....	17
3.1.5	- Conclusion .....	19
3.2	Algorithmes testés - Notebook NB3.2.....	19
3.2.1	- Performances des modèles – Courbe de ROC.....	20
3.2.2	- Performances des modèles – Selon le seuil de détection.....	20
3.3	Conclusion .....	22
IV.	Interprétabilité de notre modèle final (XGBOOST) - Notebook NB4.....	23
4.1	Interprétabilité Globale avec Shap.....	23
4.2	Interprétabilité Locale avec Lime .....	24
4.3	Shapash .....	26
4.4	Intérêt graphique du modèle XGBoost .....	28
4.5	Intérêt graphique des arbres de décision .....	29
V.	Clustering des villes - Notebook NB5 .....	31
5.1	Introduction à la classification des climats de Koppen .....	31
5.2	1 <sup>ère</sup> lettre : type de climat – Notebook NB5.1 .....	32
5.3	2 <sup>ème</sup> lettre : régime pluviométrique – Notebook 5.2.....	33
5.4	3 <sup>ème</sup> lettre : variations de températures – Notebook 5.3 .....	35

5.5	Combinaison des classifications – Notebook 5.4 .....	36
VI.	Séries Temporelles – Notebook NB6.....	39
6.1	Résultats mensuels pour les sept villes .....	40
6.1.1	– Analyse de Canberra sur les trois indicateurs .....	41
6.1.2	- Analyse de Cairns sur les 3 indicateurs .....	42
6.1.3	- Conclusion des analyses de séries temporelles sur les villes .....	43
6.2	Étude de la saisonnalité de <i>Rainfall</i> sur deux climats .....	43
VII.	Deep Learning – Notebook NB7 .....	45
7.1	Introduction.....	45
7.2	Modèles denses classiques.....	45
7.3	Modèles FASTAI.....	47
7.4	Conclusion Deep Learning .....	47
VIII.	Conclusion générale .....	48
IX.	Annexes - autres modélisations de RainTomorrow .....	49
9.1	Modélisation de la pluie à J+3 et J+7 – Arbre de décision .....	49
9.2	Modélisation de la pluie à J+1 et J+7 – LightGBM et Shapash .....	50
X.	Répartition de la charge – Diagramme de Gantt du projet RainsBerryPy .....	53
XI.	Bibliographie & Sitographie.....	55

# I. Preprocessing - Notebook NB1

## 1.1 Observation du jeu de données initial

### Doc. 1 : Informations du jeu de données initial

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  145460 non-null object
1   Location              145460 non-null object
2   MinTemp               143975 non-null float64
3   MaxTemp               144199 non-null float64
4   Rainfall              142199 non-null float64
5   Evaporation           82670 non-null float64
6   Sunshine              75625 non-null float64
7   WindGustDir           135134 non-null object
8   WindGustSpeed         135197 non-null float64
9   WindDir9am            134894 non-null object
10  WindDir3pm            141232 non-null object
11  WindSpeed9am          143693 non-null float64
12  WindSpeed3pm          142398 non-null float64
13  Humidity9am           142806 non-null float64
14  Humidity3pm           140953 non-null float64
15  Pressure9am           130395 non-null float64
16  Pressure3pm           130432 non-null float64
17  Cloud9am              89572 non-null float64
18  Cloud3pm              86102 non-null float64
19  Temp9am               143693 non-null float64
20  Temp3pm               141851 non-null float64
21  RainToday             142199 non-null object
22  RainTomorrow          142193 non-null object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Le jeu de données possède **145 460** entrées et **23** colonnes dont :

- La date de l'observation (Date).
- La ville dans laquelle se situe la station météo (Location).
- La variable cible RainTomorrow dont la valeur (Yes ou No) indique s'il a plu ou non le lendemain de l'observation.
- 20 variables décrivant les conditions atmosphériques du jour de l'observation :

Le jeu de données contient un mélange de variables explicatives **catégorielles** (type object) et de variables explicatives **numériques** (type float64) :

- 14 variables continues : MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Temp9am, Temp3pm.
- 2 variables discrètes (Nombre d'octas, de 0 à 9) : Cloud9am, Cloud3pm
- 4 variables catégorielles non-numériques : WindGustDir, WindDir3am, WindDir3pm, RainToday.

#### Remarques :

- Les valeurs de la variable RainToday (Yes, No) sont définies par la variable Rainfall (Yes si précipitations > 1mm).
- Plusieurs variables possèdent de nombreuses valeurs manquantes qu'il faudra gérer.

### Doc. 2 : Tableau descriptif des variables catégorielles

	Date	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday	RainTomorrow
count	145460	145460	135134	134894	141232	142199	142193
unique	3436	49	16	16	16	2	2
top	2013-11-12	Canberra	W	N	SE	No	No
freq	49	3436	9915	11758	10838	110319	110316

Le jeu de données comporte **3436** journées d'observations météorologiques (entre décembre 2008 et juin 2017) réalisées par **49** stations météorologiques (Location).

## 1.2 Ajout de variables supplémentaires

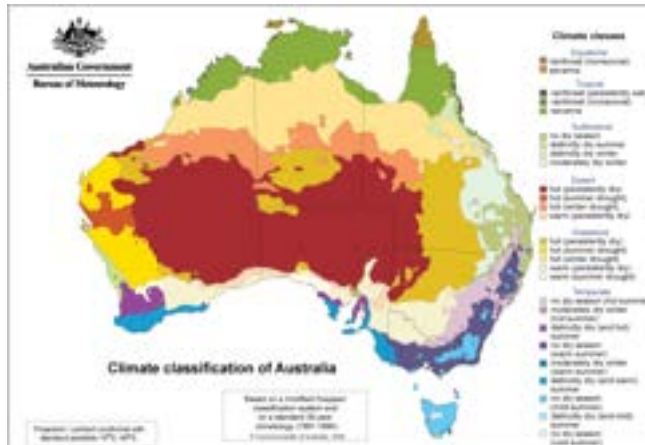
- Numérisation des deux variables booléennes RainToday et RainTomorrow.
- Décomposition de la date en trois variables : Année, Mois, Jour.
- Ajout de deux variables de coordonnées géographiques : Latitude, Longitude.
- Ajout de deux variables indiquant le climat :
  - Climat\_Koppen : classe climatique dans la classification de Köppen.
  - Clim\_type : type de climat regroupant plusieurs classes de Köppen, définie à partir de Climat\_Koppen
- Ajout de 3x3 variables précisant la direction des vents (définies à partir de WindGustDir, WindDir9am et WindDir3pm):
  - WindGust\_Ang, Wind9am\_Ang, Wind3pm\_Ang : angle correspondant (en degrés) sur le cercle trigonométrique (ie. E=0° et rotation dans le sens direct).
  - WindGust\_cos, Wind9am\_cos, Wind3pm\_cos : cosinus de l'angle (abscisse des coordonnées trigo).
  - WindGust\_sin, Wind9am\_sin, Wind3pm\_sin : sinus de l'angle (ordonnée des coordonnées trigo).
- Pluie à J-1, J-2, J+1, J+2
- Circularisation de la variable Mois (<https://datascientest.com/numeriser-des-variables>). De cette façon, les mois de décembre et janvier ont des valeurs proches.

## Justification de l'ajout d'une variable indiquant le climat

**Constat :** L'Australie est une grande île (7 700 000 km<sup>2</sup>) située à la frontière entre l'océan Indien à l'ouest et l'océan Pacifique à l'est, à des latitudes intermédiaires (15°- 40° Sud). Les climats y sont variés. Le nord-est est sous l'influence de climats chauds et humides (tropicaux et subtropicaux) à mousson estivale. Le climat devient plus froid et tempéré au sud-est. On observe un méditerranéen au sud-ouest (mousson hivernale) et des climats plus secs au centre du pays lorsqu'on s'éloigne du littoral.

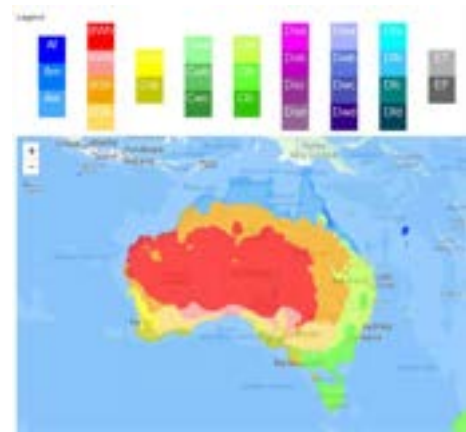
### Hypothèses :

- L'importance des précipitations et leur répartition annuelle sont différentes selon le type de climats, avec des périodes de moussons plus ou moins marquées sur des saisons différentes.
- Les variables atmosphériques n'ont pas toutes la même influence sur la pluie selon le type de climat. La direction des vents apportant la pluie dépend de la distance et de la position de la station météorologique par rapport à l'océan.



**Doc. 3a :** Carte de répartition des climats en Australie

Source: Bureau of Meteorology - <http://www.bom.gov.au>



**Doc. 3b :** Carte des classes climatiques de Köppen

Sources: plantmaps.com

## Justification de l'ajout de variables précisant la direction des vents

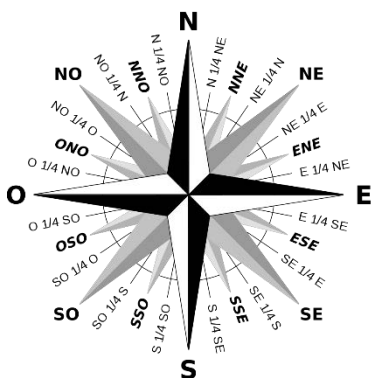
Les variables WindGustDir, WindDir9am et WindDir3pm indiquent la direction des vents sous forme de points cardinaux (32 modalités : voir Doc. 4).

L'objectif est de numériser ces variables afin d'indiquer la direction W-E et S-N des vents.

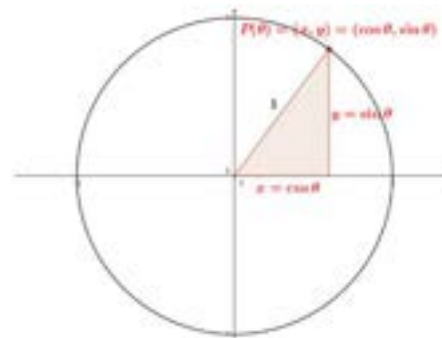
Une première série de trois variables est créée dans la but d'indiquer l'angle du vent en degré selon la position du point d'origine du vent sur le cercle trigonométrique (WinGust\_Ang,...).

Deux autres séries de trois variables sont ensuite calculées à partir de l'angle (WinGust\_cos,... et WindGust\_sin...). Elles indiquent le cosinus et le sinus de l'angle, autrement dit la direction W-E et S-N du vent (voir Doc. 5) :

- Un cosinus négatif correspond à un vent d'ouest, alors qu'un cosinus positif correspond à un vent d'est.
- Un sinus négatif correspond à un vent de sud, alors qu'un sinus positif correspond à un vent de nord



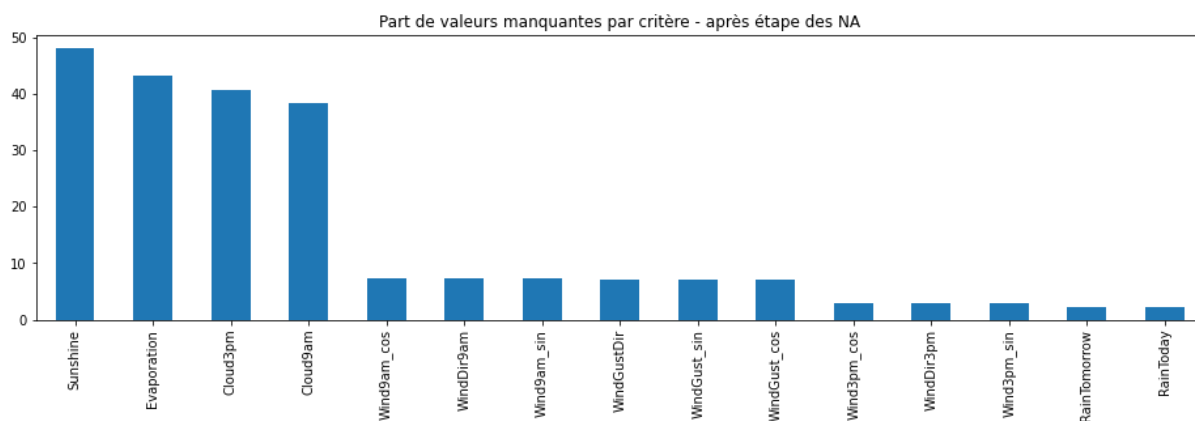
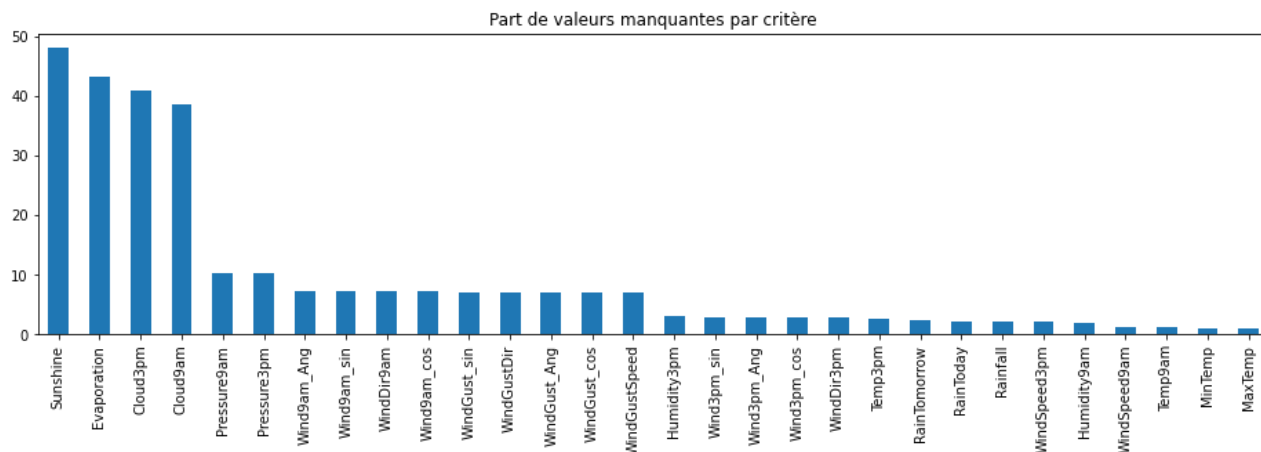
**Doc. 4 :** Rose des vents à 32 branches



**Doc. 5 :** Cercle trigonométrique

### 1.3 Gestion des valeurs manquantes

- Le Doc. 6 liste les variables avec des valeurs manquantes, triées par ordre décroissant de valeurs manquantes : 4 variables avec une forte proportion de valeurs manquantes (Sunshine / Evaporation / Cloud3pm / Cloud9am). On utilisera l'algorithme KNN-Imputer (<https://www.youtube.com/watch?v=QVEJJNsz-eM>) qui permet d'imputer des NA en fonction des voisins. Le nombre de voisins est à paramétrer, nous avons pris cinq.
- Les autres variables avec moins de 10% de valeurs manquantes peuvent être traitées avec la méthode interpolate.



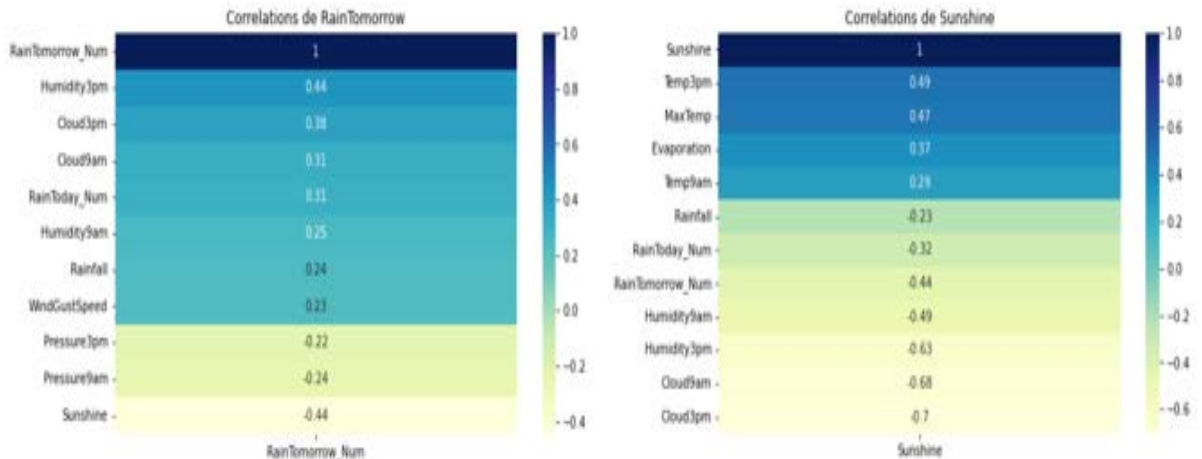
**Doc. 6 : Part de valeurs manquantes avant et après traitement**

## II. DataViz – Notebook NB2

### 2.1 Corrélations de RainTomorrow et Sunshine

#### Hypothèse :

- Intuitivement, on peut penser que la pluie du lendemain est liée à la pluie du jour ou aux températures. Même chose pour l'ensoleillement qui doit être lié aux températures.

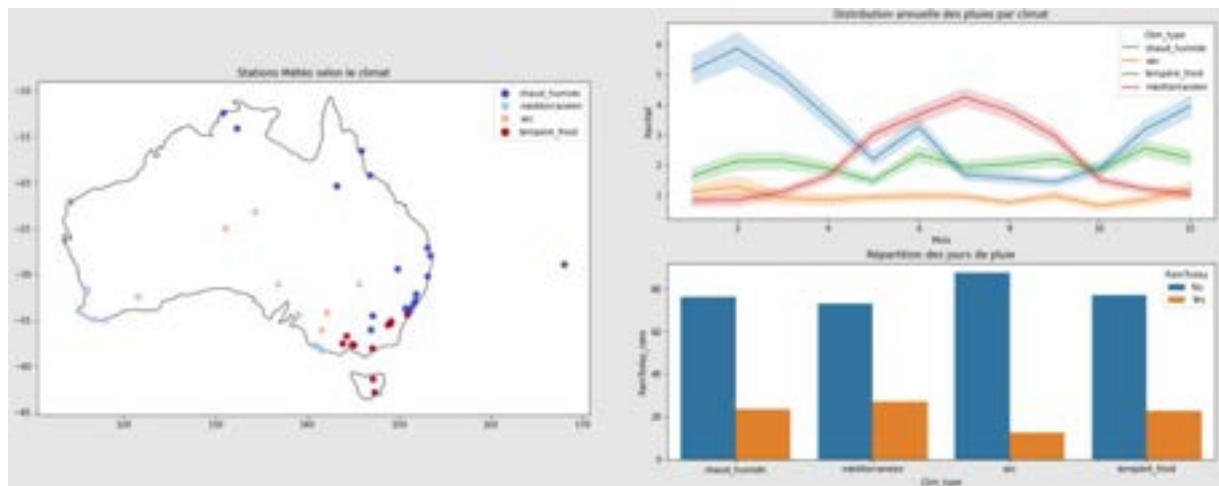


**Doc. 7 :** Corrélations entre RainTomorrow /Sunshine et les autres variables

#### Observations :

- L'analyse des corrélations nous montre que les liaisons entre les différents critères sont nombreuses.
- Quelles sont les variables les plus corrélées à RainTomorrow ?
  - Ensoleillement : Sunshine
  - Humidité : 3pm et 9am
  - Couverture nuageuse : 3pm et 9am
  - Pluie du jour : RainToday
  - Pression atmosphérique : Pressure3pm et Pressure9am
- L'ensoleillement (Sunshine) est corrélé à RainTomorrow\_num malgré presque 50% de valeurs manquantes pour cette variable. Quand on regarde les corrélations, on peut imaginer de traiter ces valeurs manquantes en régressant Sunshine sur les critères les plus corrélés, à savoir :
  - Couverture nuageuse : 3pm et 9am
  - Humidité : 3pm et 9am
  - Température : Temp3pm, MaxTemp, Temp9am

## 2.2 Représentation cartographique



**Doc. 8 : Répartition géographique des climats et distributions des pluies par climat**

### Hypothèse :

- Les différents climats australiens vont avoir un impact important dans le volume mensuel de précipitations ou dans la prévision de la pluie du lendemain.

### Observations :

Les stations météo d'Australie sont regroupées en 4 climats différents :

- méditerranéen : stations du sud-ouest et du sud-centre
- chaud\_humide (tropical et subtropical humide) => côte est du pays
- tempéré\_froid (tempéré océanique + montagnard) => plutôt sud-est
- sec (chaud et semi-aride, voire aride) => intérieur du pays

La distribution mensuelle des précipitations (Doc. 8) illustre bien les différences de climat (mousson estivale pour le climat tropical, hivernale pour le climat méditerranéen).

Pour les stations au climat sec, on observe 9% de jours de pluie alors que pour les autres on est aux alentours de 22, 23%.

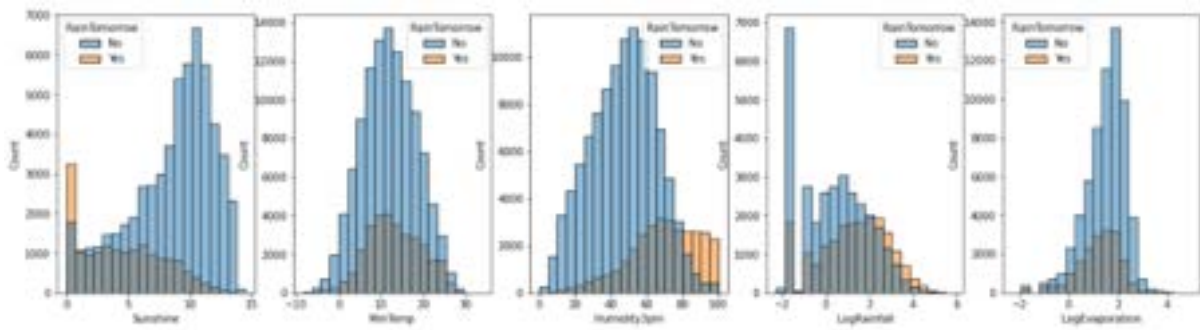


**Notre hypothèse est vérifiée.**



## 2.3 Influence pour la prévision de pluie

### 2.3.1 - Influence de certains critères (indépendamment du climat)



Doc. 9 : Distribution de cinq variables explicatives en fonction de RainTomorrow

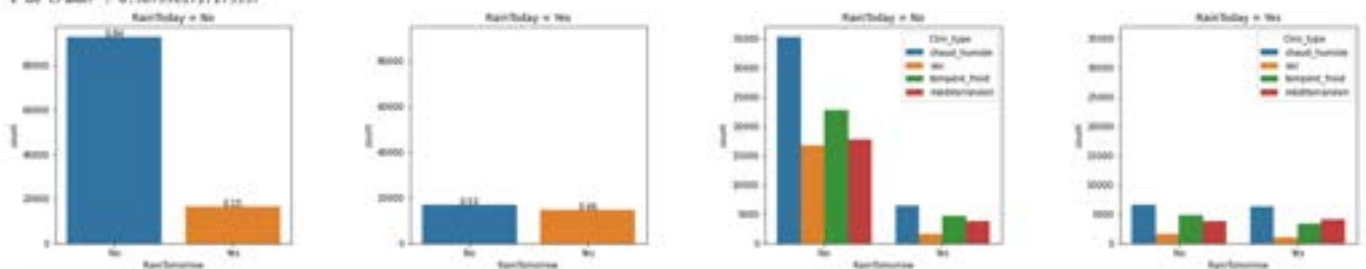
#### Constats :

- La distribution des variables Sunshine et Humidity3pm est bien différente selon RainTomorrow.
- Pour MinTemp, la distribution est relativement similaire.
- Pour Rainfall et Evaporation, il faut appliquer la fonction log pour neutraliser l'influence des valeurs extrêmes. On voit aussi l'influence plus importante de Rainfall sur RainTomorrow (distribution différente).

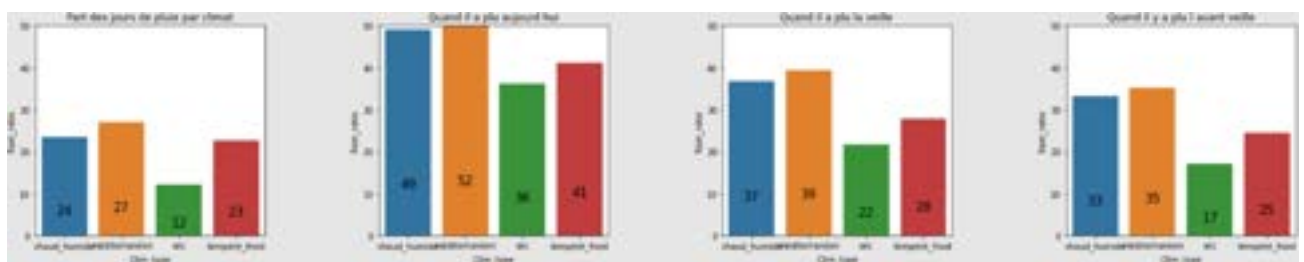
### 2.3.2 - Influence de la pluie des jours précédents sur la pluie du lendemain

Influence de RainToday sur RainTomorrow

Test du chi2 entre RainToday et RainTomorrow :  
statistique du test chi2 = 13799.479649324368  
p-value du test chi2 = 0.0  
degré de liberté du test chi2 = 1  
V de Cramer : 0.9879665727173337



Doc. 10a : Distribution de RainToday en fonction de RainTomorrow, sur l'ensemble du jeu de données, puis par climat



Doc. 10b : Part des jours de pluie par climat

#### Interprétation :

S'il ne pleut pas aujourd'hui, on n'observera pas de pluie le lendemain dans 84% des cas. La réciproque n'est pas vraie : s'il pleut aujourd'hui, la pluie ne persistera que dans la moitié des cas environ.

Cette tendance s'observe pour tous les climats et se vérifie aussi sur la pluie des jours précédents (dans une moindre mesure tout de même, surtout pour le climat tempéré froid).

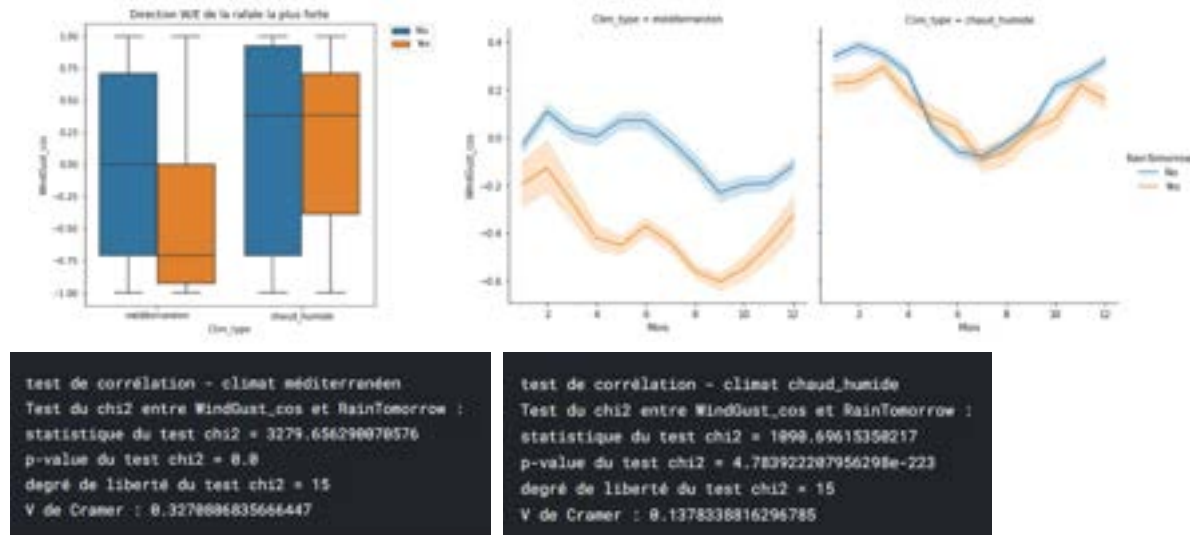
### Hypothèses :

- La direction des vents dominants diffère d'un climat à l'autre.
- La direction des vents n'exerce pas la même influence sur la pluie selon le climat.

Pour tester ces hypothèses, on se concentre sur deux climats différents :

- Le climat chaud\_humide (côte est, nord-est, océan Pacifique)
- Le climat méditerranéen (côte sud, sud-ouest, océan Indien)

On étudiera uniquement la direction W-E de la rafale la plus forte (WindGust\_cos).



**Doc. 11 :** Distribution des vents W-E en climat méditerranéen et (sub)tropical humide – test de corrélation du Chi2

### Observations :

La direction W-E de la rafale la plus forte varie au cours de l'année. En climat méditerranéen, le cosinus est plus faible en cas de pluie le lendemain, ce qui traduit une direction du vent provenant plutôt de l'ouest.

### Conclusion des tests statistiques du Chi2 :

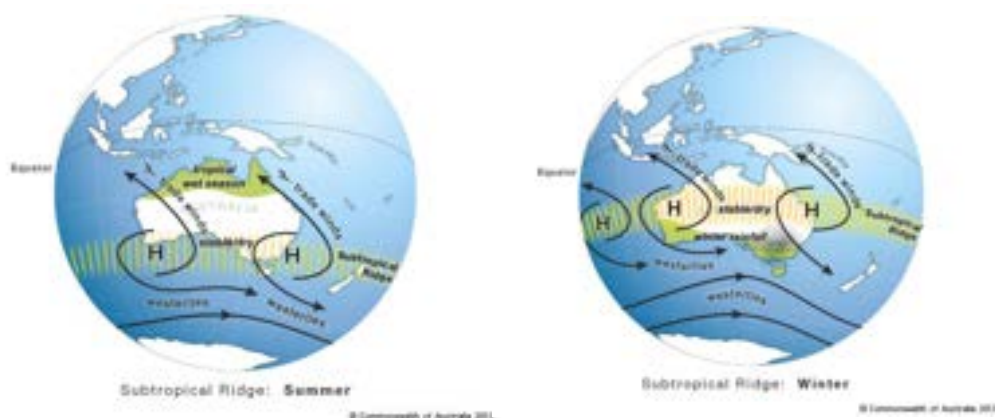
La direction ouest-est de la rafale la plus forte est corrélée à la pluie en climat méditerranéen (p-value = 0) et en climat chaud\_humide (p\_value = 0) mais la corrélation est plus importante en climat méditerranéen (V de Cramer plus élevé).

**Nos hypothèses sont validées.**

### Interprétation :

Les régions méditerranéennes sont situées sur la côte sud ou sud-ouest de l'île (océan Indien). Les pluies sont apportées par des fronts nuageux venant de l'ouest.

Le phénomène est d'autant plus marqué en hiver, en raison de la migration vers le nord de la crête subtropicale. Il s'agit d'une ceinture de hautes-pressions (anticyclones) qui permet le maintien d'un temps sec sur le sud du pays en été. Pendant l'hiver austral, cette ceinture remonte vers le nord, donc la côte sud de l'Australie se retrouve sous cette crête, ce qui permet aux fronts froids dépressionnaires situés sous la ceinture de parcourir le sud du pays, apportant fréquemment des pluies.



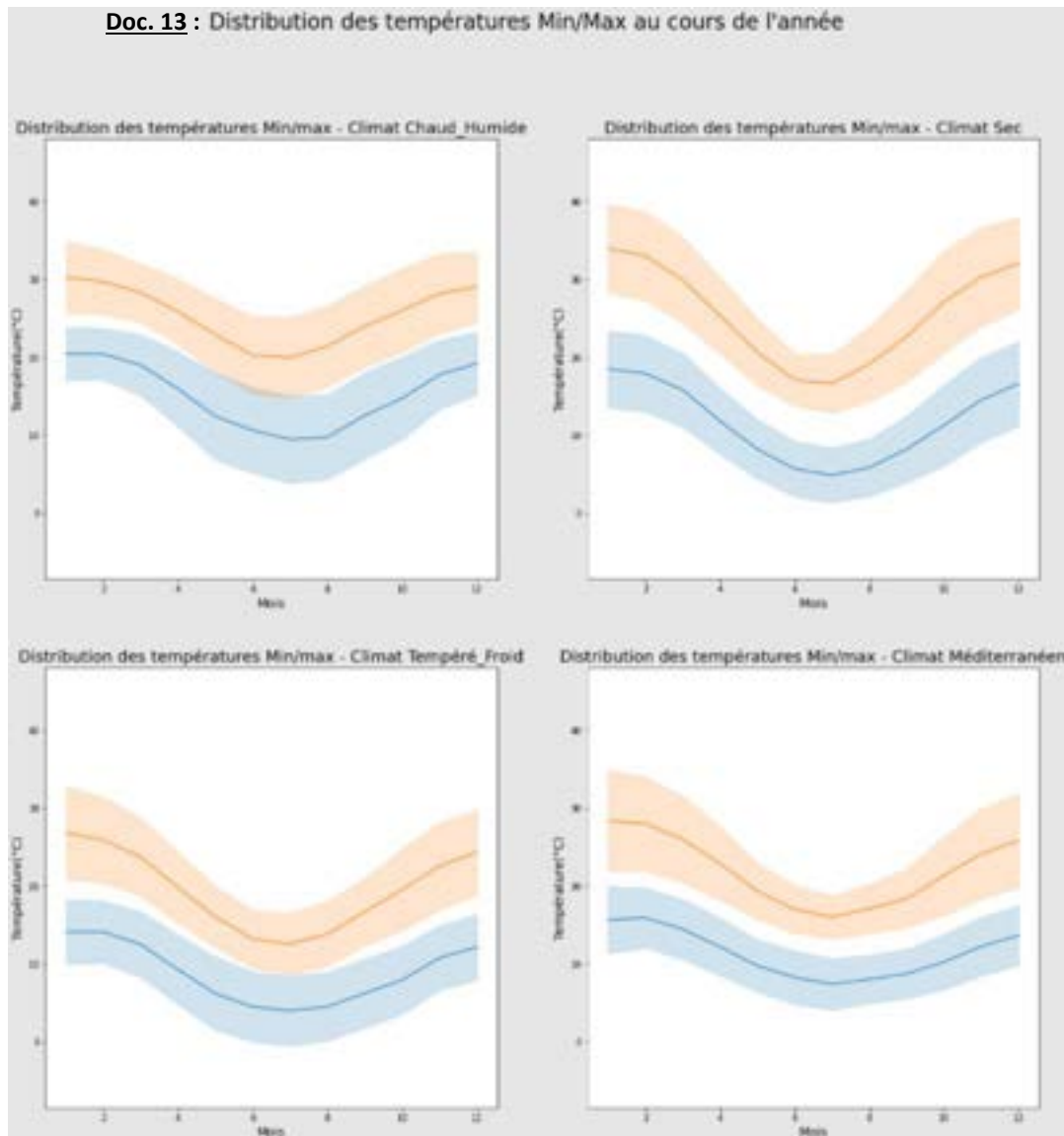
**Doc. 12 :** Déplacement de la crête subtropicale entre l'été et l'hiver austral - Source : Bureau of Meteorology

## 2.4 Distribution des températures au cours de l'année suivant le climat

### Hypothèses :

- Climat méditerranéen : température clémente en hiver et chaude en été avec variations min/max restreintes,
- Climat sec : température plus chaude que pour les autres climats, voire très chaude en été. Les variations de température sont très importantes sur une journée (zone désertique),
- Climat chaud\_humide : température clémente en hiver et chaude en été avec variations min/max restreintes,
- Climat tempéré\_froid : température plus basse du continent en hiver et moins chaude en été.

**Doc. 13 :** Distribution des températures Min/Max au cours de l'année



### Conclusions :

Les hypothèses énoncées plus tôt sont vérifiées. En visualisant l'écart type sur les courbes, nous constatons aussi :

- Climats méditerranéen et sec : très faible dispersion des températures au cours de l'année.
- Climat chaud\_humide : l'écart-type est ici plus important que pour les autres climats.

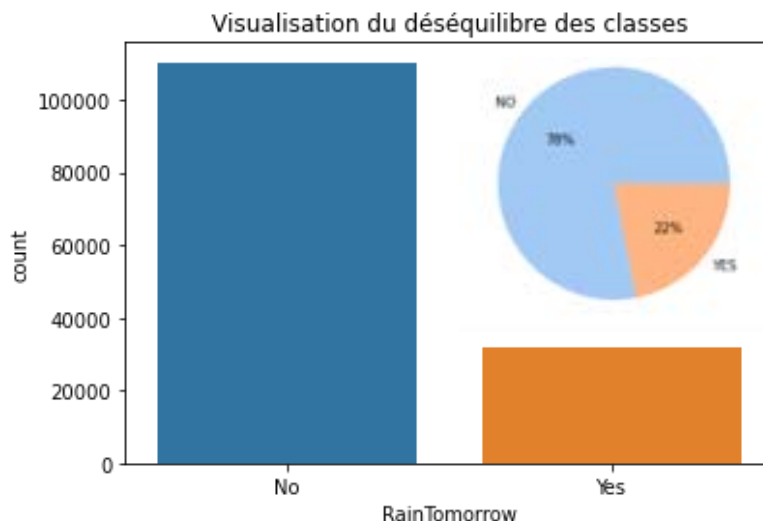
### III. Modélisation de RainTomorrow – Notebooks NB3.1 et NB3.2

**Objectif** : Prédire la valeur de la variable cible : *RainTomorrow*.

Elle signifie : « A-t-il plu le jour suivant, oui ou non ? »

La valeur est « Yes » si les précipitations mesurées pour ce jour dépassent 1mm. Cette valeur correspond à la classe 1.

Nous avons affaire à un problème de **classification binaire**. Le phénomène est vérifié dans 22% des cas. Le jeu de données est donc déséquilibré.



Ce déséquilibre doit être pris en compte dans notre stratégie car il peut affecter grandement les performances de nos modèles.

La section 3.1 se focalisera sur ce problème et plus généralement sur les outils que nous pouvons mettre en œuvre pour améliorer les performances de nos modèles en appliquant différents traitements aux données d'entraînement.

La section 3.2 portera sur une comparaison des performances des différents algorithmes de *machine learning* classique et leur optimisation.

#### 3.1 Outils pour améliorer les performances – Notebook NB3.1

**Objectif** : Étudier les performances d'un modèle de *machine learning* en fonction des différents traitements que l'on peut appliquer au jeu de données : traitement des valeurs manquantes, sélection de variables, rééchantillonnage.

Les résultats présentés dans cette section sont issus d'un modèle de forêt aléatoire (*Random Forest*) non optimisé (hyperparamètres par défaut). Les résultats sont similaires pour d'autres modèles testés (non présentés).

**Les différents traitements du jeu de données et méthodes testés :**

- 3.1.1 - Équilibrage des classes
- 3.1.2 - Validation croisée pour détecter un problème de surapprentissage
- 3.1.3 - Comparaison des performances en modifiant le seuil de probabilité de détection de la classe 1
- 3.1.4 - Comparaison des performances pour différents traitements des valeurs manquantes
- 3.1.5 - Sélection de variables

### 3.1.1 - Équilibrage des classes

Algorithme : Random Forest

Jeu de données : weatherAUS\_imputer (valeurs manquantes interpolées par KNN Imputer)

⇒ **Avant rééquilibrage**

Performances sur train :

Classe	Précision	Rappel	Score F1	Accuracy
0	1	1	1	1
1	1	1	1	

Performances sur test :

Classe	Précision	Rappel	Score F1	Accuracy
0	0.87	0.96	0.91	0.85
1	0.76	0.49	0.59	

Ces résultats illustrent le déséquilibre entre les deux classes. Les métriques sont en effet bien meilleures sur la classe majoritaire 0 que sur la classe minoritaire 1, en particulier le rappel.

Nous allons donc équilibrer notre échantillon dans le but d'améliorer la détection de la classe 1.

Deux possibilités :

- Sous échantillonnage de la classe 0 (*RandomUnderSampler*)
- Sur échantillonnage de la classe 1 (*RandomOverSampler*)

Nous présentons ici les résultats obtenus par sous-échantillonnage.

⇒ **Après rééquilibrage**

Performances sur train :

Classe	Précision	Rappel	Score F1	Moyenne géométrique
0	1	1	1	1
1	1	1	1	

Performances sur test :

Classe	Précision	Rappel sensibilité	Score F1	Moyenne géométrique
0	0.93	0.80	0.86	0.79
1	0.53	0.79	0.63	

Pour la classe 1, on constate que le score F1 est légèrement meilleur après équilibrage qu'avant, en raison d'un score de rappel beaucoup plus élevé, proche de celui de la classe 0.

Toutefois, ce gain significatif de rappel se fait au détriment de la précision qui baisse de façon importante entre train-set et le test-set. Notons que celle-ci était significativement plus élevée pour le jeu de données déséquilibré.

Dans les quatre sections suivantes, nous présenterons les techniques mises en œuvre pour rechercher les causes de ce faible score de précision et tenter de l'améliorer.

### 3.1.2 - Validation croisée pour détecter un problème de surapprentissage

Algorithme : Random Forest

Jeu de données : weatherAUS\_imputer (valeurs manquantes interpolées par KNN Imputer)

**Hypothèse : La chute de précision entre le train et le test est due à un problème de surapprentissage.**

**Méthode utilisée :** `cross_val_score` avec un `StratifiedKFold` comprenant 10 sous-échantillons mélangés du jeu de données rééquilibré.

À chaque itération, le modèle est entraîné et évalué sur des portions différentes du jeu de données.

**Résultats :**

```
Scores de précision: [0.8 0.79 0.8 0.79 0.79 0.8 0.79 0.8 0.79 0.79]
Score moyen : 0.79
Ecart-type : 0.01
```

**Observations :** Les scores obtenus sur les différents échantillons aléatoires sont très proches et élevés. La validation croisée ne permet donc pas de mettre en évidence un problème de surapprentissage. Comment expliquer alors la chute de précision entre train et test après rééquilibrage ?

**Interprétation de la différence entre précision et rappel après rééquilibrage :**

La différence de performances sur le train-set et le test-set est en partie inhérente au rééquilibrage des classes qui ne se fait que sur le jeu de données train. Or, sur le jeu de données test, la classe 1 est toujours sous-représentée.

Ainsi, le test-set contient plus d'objets de la classe 0 que le train-set, dont une partie sera nécessairement mal classée par le modèle (faux positifs). Le taux de faux-positif sera donc plus élevé sur le test-set que sur train-set, ce qui entraînera une chute de précision.

En revanche, le test-set contient moins d'objets de la classe 1 que le train-set. Ainsi, même si certains d'entre eux sont mal détectés (faux négatifs), ils sont peu nombreux et le taux de faux négatif restera donc faible.

Rappelons les formules de calcul de la précision et du rappel :

$\text{Précision} = \text{Nombre de vrais positifs} / \text{Nombre de prédictions positives} = VP / (VP + FP)$

$\text{Rappel} = \text{Nombre de vrais positifs} / \text{Nombre d'observations réellement positives} = VP / (VP + FN)$

Avec VP : Vrais positifs, FP : Faux positifs, FN : Faux négatifs

On voit bien que le nombre de faux positifs (FP) n'entre pas dans le calcul du rappel. Celui-ci est donc moins sensible au déséquilibre des classes que la précision.

Nous allons vérifier ces interprétations en évaluant le modèle sur un jeu de données test préalablement rééquilibré.

**Performances du modèle Random Forest sur un test-set rééquilibré (préalablement entraîné sur un train-set rééquilibré):**

	pre	rec	spe	f1	geo
0	0.79	0.80	0.79	0.80	0.80
1	0.80	0.79	0.80	0.79	0.80

**Conclusion :**

On obtient cette fois de bonnes performances sur le test-set, proches de celles du train-set.

Les prédictions du modèle sont donc bonnes sur un jeu de données test rééchantillonné, ce qui confirme que nous n'avons pas affaire à un problème de surapprentissage.

Toutefois, gardons en tête que cette technique de rééchantillonnage n'est pas applicable dans la réalité pour les futures prédictions de notre modèle, ces données étant par définition non étiquetées donc non équilibrables. **La technique de rééchantillonnage ne peut servir qu'à entraîner le modèle.**

Or, nous aurons probablement affaire à des données déséquilibrées, le déséquilibre de classe étant général (pour tous les climats et toutes les années). Notre problème de précision persiste donc.

Dans la section suivante, nous allons comparer de façon plus détaillée les scores avec et sans rééquilibrage en modifiant le seuil de probabilité pour la détection de la classe 1.

### 3.1.3 - Comparaison des performances en modifiant les seuils de probabilités pour la détection de la classe 1

Algorithme : Random Forest

Jeu de données : weatherAUS\_imputer (valeurs manquantes interpolées par KNN Imputer)

**Hypothèse : Les scores dépendent fortement du seuil de probabilité choisi.**

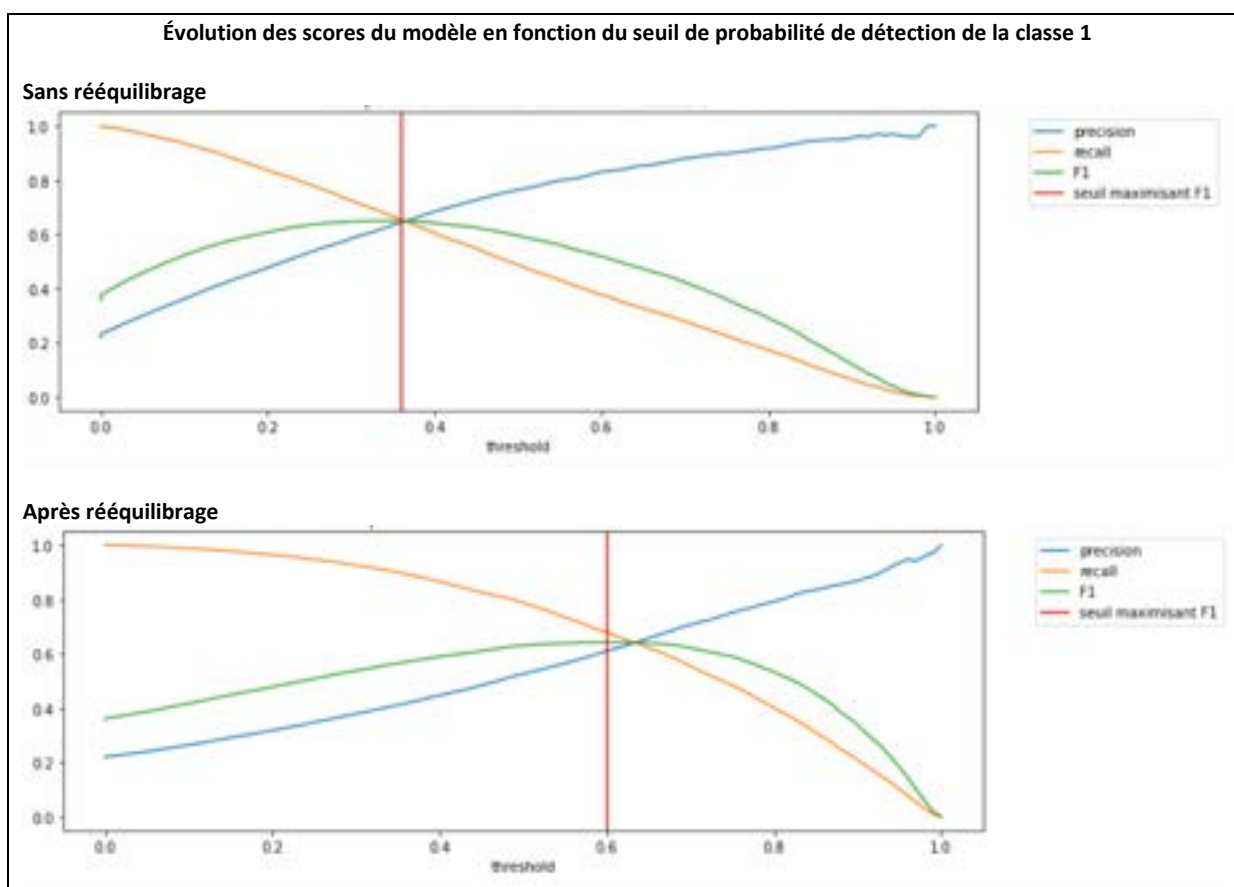
La principale métrique utilisée pour comparer les performances sera le score F1. Il a l'avantage de combiner les scores de précision et de rappel. **Les scores présentés sont ceux de la classe 1.**

⇒ **Avec un seuil de probabilité non modifié (50%) :**

Score	Précision - seuil 50%	Rappel - seuil 50%	F1 - seuil 50%
Sans rééquilibrage	0.76	0.49	0.59
Après rééquilibrage	0.53	0.79	0.63

Si l'on se limite à un seuil de probabilité de 50%, les performances sont légèrement meilleures après rééchantillonnage. Ce gain est dû au rappel, mais la précision est quant à elle beaucoup plus faible.

⇒ **Avec un seuil de probabilité maximisant le score F1 :**



Score	Précision pour F1 max	Rappel pour F1 max	F1 max	Seuil maximisant F1
Sans rééquilibrage	0.64	0.66	0.65	0.36
Après rééquilibrage	0.61	0.67	0.64	0.59

**Observations :**

En choisissant un seuil de probabilité maximisant F1, on constate que les deux modèles ont des performances voisines pour les trois métriques (précision, rappel et score F1).

**La différence majeure entre les deux modèles est le seuil de probabilité permettant d'obtenir le meilleur score F1 (0,59 après rééquilibrage contre 0,36 sans rééquilibrage).**

Sans rééchantillonnage, il est donc nécessaire de baisser fortement le seuil de probabilité pour détecter correctement la classe 1, alors qu'avec rééchantillonnage, on obtient un score F1 plus proche du maximum et un bon score de rappel avec la méthode *predict* simple (seuil de 50%).



### Conclusion :

Cette étude montre que le rééchantillonnage ne permet pas d'améliorer réellement les performances du modèle.

La technique de rééchantillonnage permet toutefois d'améliorer le F1-score pour un seuil de probabilité standard fixé à 0,5. Pour ce seuil, la détection de la classe 1 sera meilleure. Ce gain se fait cependant au détriment d'une perte de précision.

Cette technique est donc pertinente si l'on cherche à détecter un maximum de jours pluvieux, au risque d'avoir des faux positifs.

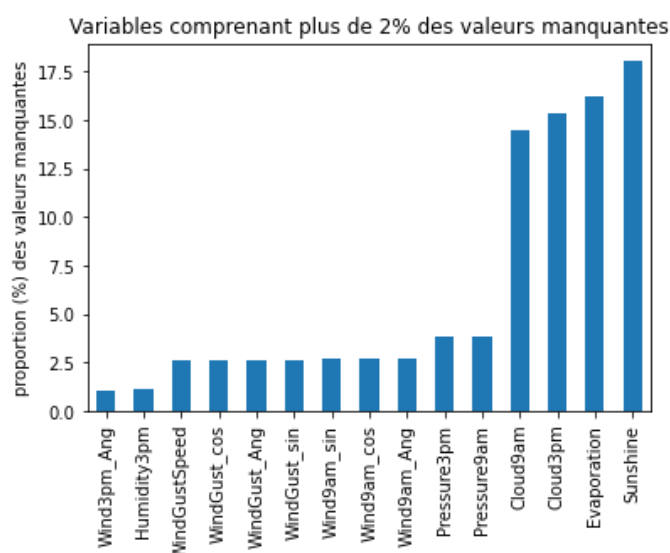
Elle permet également d'entraîner plusieurs modèles sans trop se soucier du seuil de probabilité qui peut rester fixé à 0,5.

**Pour ces raisons, nous appliquerons un rééchantillonnage du jeu de données train pour tous les modèles que nous testerons (section 3.2). Nous rechercherons également à chaque fois le seuil de probabilité de détection qui maximise le score F1.**

### 3.1.3 - Comparaison des performances pour différents traitements des valeurs manquantes

Algorithme : Random Forest

**Hypothèse : Les performances dépendent de la méthode de traitement des valeurs manquantes.**



### Observations :

Les variables *Cloud9am*, *Cloud3pm*, *Evaporation* et *Sunshine* regroupent à elles seules plus de 63 % des valeurs manquantes.

Trois techniques ont été utilisées pour traiter les valeurs manquantes et créer trois jeux de données :

1. *dropna* : suppression des observations possédant des valeurs manquantes par la méthode *dropna*.
2. *dropvar* : suppression des quatre variables possédant le plus de valeurs manquantes (*Cloud9am*, *Cloud3pm*, *Evaporation*, *Sunshine*), puis suppression des observations restantes possédant des NaN.
3. *imputer* : remplacement des valeurs manquantes par la méthode *KNN-Imputer* (voir section 1).

Les scores obtenus sur train sont tous égaux à 1 ou presque. Nous présentons ici les scores obtenus sur test :

Méthode utilisée	Taille du jeu de données	Précision		Rappel		Score F1		Accuracy
		Classe 0	Classe 1	Classe 0	Classe 1	Classe 0	Classe 1	
1. <i>dropna</i>	56 564	0.94	0.55	0.81	0.83	0.87	0.66	0.81
2. <i>dropvar</i>	113 675	0.93	0.53	0.81	0.79	0.87	0.64	0.80
3. <i>imputer</i>	145 455	0.93	0.53	0.80	0.79	0.86	0.63	0.79

### Conclusion :

Le jeu de données *dropna* présentent les meilleures performances, en plus d'être le plus rapide.

L'interpolation des valeurs manquantes a donc un impact négatif sur les performances du modèle. Le modèle est plus performant s'il est entraîné exclusivement sur des données réelles.



La différence de performance entre *dropna* et *dropvar* suggère qu'au moins une partie des variables supprimées de *dropvar* (*Cloud9am*, *Cloud3pm*, *Evaporation*, *Sunshine*) ont de l'importance pour l'entraînement du modèle.

**Nous utiliserons donc préférentiellement le jeu de données *dropna* pour entrainer nos algorithmes.** En plus d'un gain probable de performances, les temps de calcul seront significativement réduits, notamment pour la recherche des meilleurs hyperparamètres.

Dans la section suivante, nous chercherons à sélectionner les meilleures variables pour l'entraînement. Les performances obtenues sur le jeu de données *dropna* serviront de référence.

### 3.1.4 - Sélection de variables

Algorithme : *Random Forest*

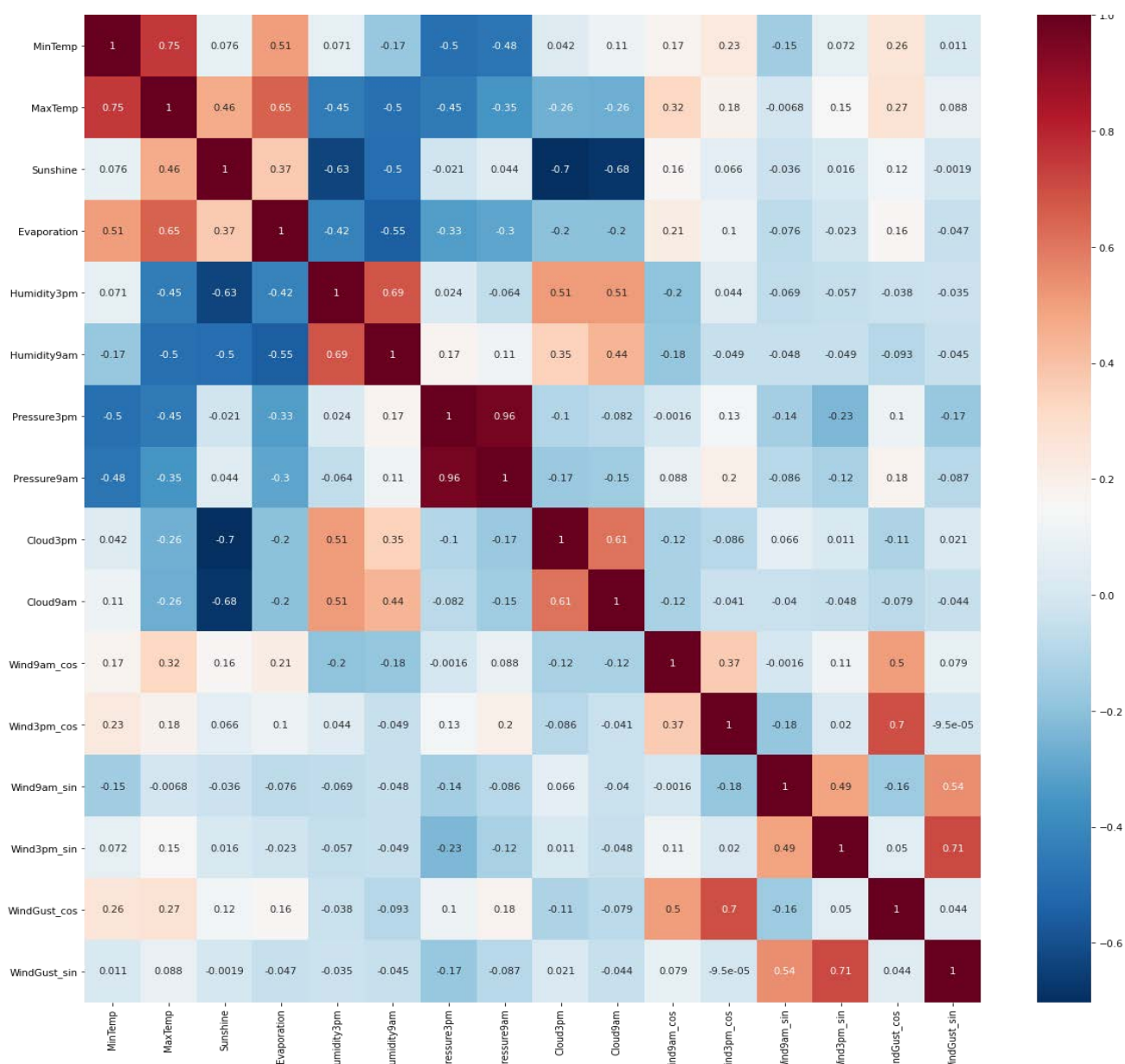
Jeu de données : *dropna*

**Hypothèse : Des variables peu pertinentes perturbent le modèle, ce qui affecte ses performances.**



#### 1<sup>re</sup> technique : suppression de variables corrélées entre elles

Carte thermique des variables explicatives



### Observations :

Plusieurs variables du jeu de données sont des mesures de la même grandeur physique, à deux moments de la journée : 9h et 15h. Ces deux variables présentent un coefficient de corrélation de Pearson supérieur à 0,6. Cette redondance de variables mesurant la même chose pourrait donner plus de poids à certaines grandeurs physiques dans l'entraînement du modèle et nuire aux performances.

L'objectif est de tester les performances du modèle après avoir supprimé l'une des deux variables.

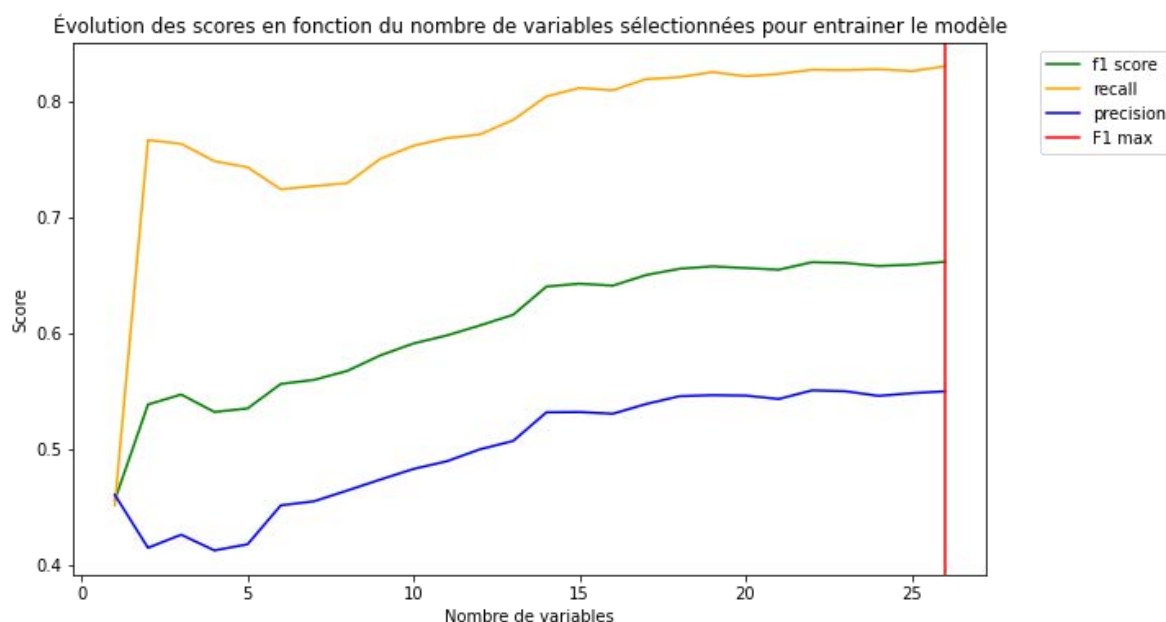
### Comparaison des performances obtenues sur le test-set :

Score	Précision	Rappel	F1	Accuracy
Toutes les variables	0.55	0.83	0.66	0.81
Variables sélectionnées	0.53	0.85	0.65	0.80

**Conclusion** : Les scores des trois métriques sont proches de celles obtenues à partir du jeu de données contenant toutes les variables. La suppression de variables corrélées entre elles ne permet donc pas d'améliorer les performances.

### ⇒ 2<sup>e</sup> technique : sélection de variables avec SelectKBest

L'objectif est de sélectionner les variables **les plus corrélées à la cible** en se basant sur un test du chi2 et de les intégrer une à une au modèle.



### Conclusion :

À partir de six variables, on observe une croissance de toutes les métriques au fur et à mesure qu'on intègre des variables au modèle. Il n'est donc pas nécessaire de supprimer des variables pour améliorer les scores. Toutefois, on remarque que les scores évoluent peu après 15 variables. La diminution du nombre de variables nous permettrait donc d'entraîner des modèles plus gourmands en ressource.

*Remarque : même constat avec l'algorithme RFE (voir Notebook).*

**Nous conserverons toutes les variables explicatives pour entraîner nos modèles.**

- Le rééchantillonnage permet d'obtenir des scores légèrement meilleurs, mais c'est surtout le choix du seuil de décision qui a le plus d'impact sur les performances.
- L'interpolation des valeurs manquantes par KNN Imputer réduit les performances au lieu de les améliorer. Il est préférable d'utiliser un jeu de données où les valeurs manquantes ont simplement été supprimées.
- Le retrait de certaines variables n'améliore pas les performances. Toutes les variables peuvent être utilisées pour entraîner nos modèles.

## 3.2 Algorithmes testés - Notebook NB3.2

### Les algorithmes suivants ont été testés en prenant en compte les résultats des analyses précédentes :

- Rééquilibrage du jeu de données avec *RandomUnderSampler*.
- Conservation de toutes les variables prédictives.
- Choix de l'algorithme sur le dataset sans les NA (données réelles)
- En revanche, application possible sur les données interpolées ce qui aurait l'intérêt de pouvoir avoir des prédictions sur les observations qui ont des valeurs manquantes (par exemple, les stations qui ne mesurent pas certains indicateurs).

### Liste des algorithmes testés :

- Arbre de décision
- Boosting sur arbre de décision (Adaboost classifieur)
- Isolation Forest (détection d'anomalies) => non présenté car vraiment trop dégradé.
- Régression logistique
- SVM
- KNN
- Random Forest
- Light GBM
- Bagging Classifieur
- Stacking Classifieur (avec les modèles préentraînés RandomForest, SVM et LogisticRegression)
- 

### Optimisation des modèles :

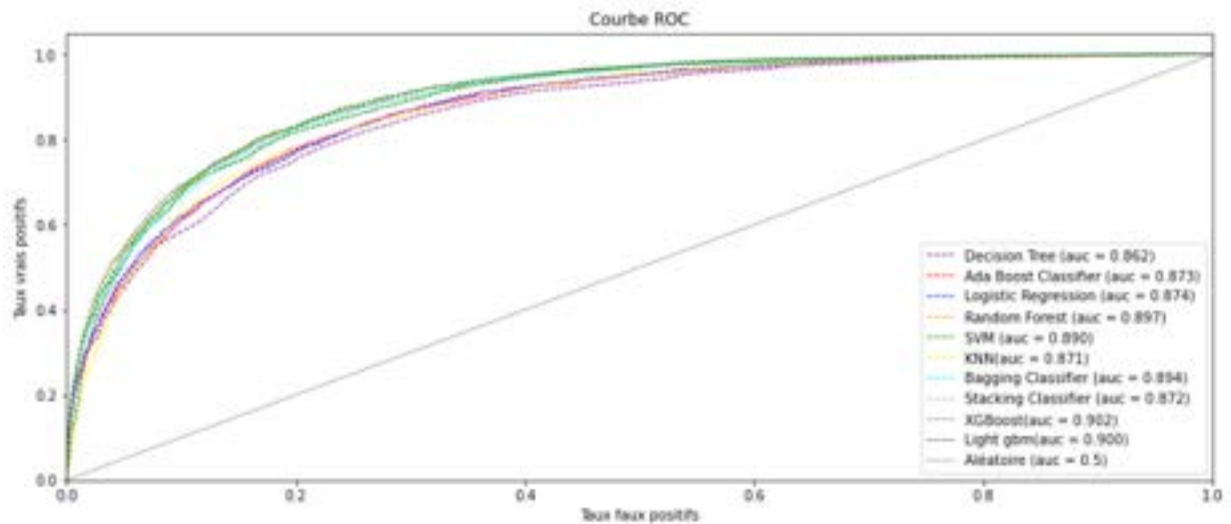
- Une grille de recherche sur les hyperparamètres a été construite pour les modèles avec le choix de maximiser le f1 comme métrique de performance et 3 folds pour limiter le surapprentissage.

### Choix du modèle :

- Le modèle final sera choisi au regard de la courbe de ROC, de l'AUC globale et surtout des métriques f1\_score, precision, rappel sur la classe à modéliser.

### Définitions :

- La *precision* correspond au taux de prédictions correctes parmi les prédictions positives. Elle mesure la capacité du modèle à ne pas faire d'erreur lors d'une prédiction positive.
- Le *recall* correspond au taux d'individus positifs détectés par le modèle. Il mesure la capacité du modèle à détecter l'ensemble des individus positifs.
- Le *F1-score* évalue la capacité d'un modèle de classification à prédire efficacement les individus positifs, en faisant un compromis entre la *precision* et le *recall* (moyenne harmonique).

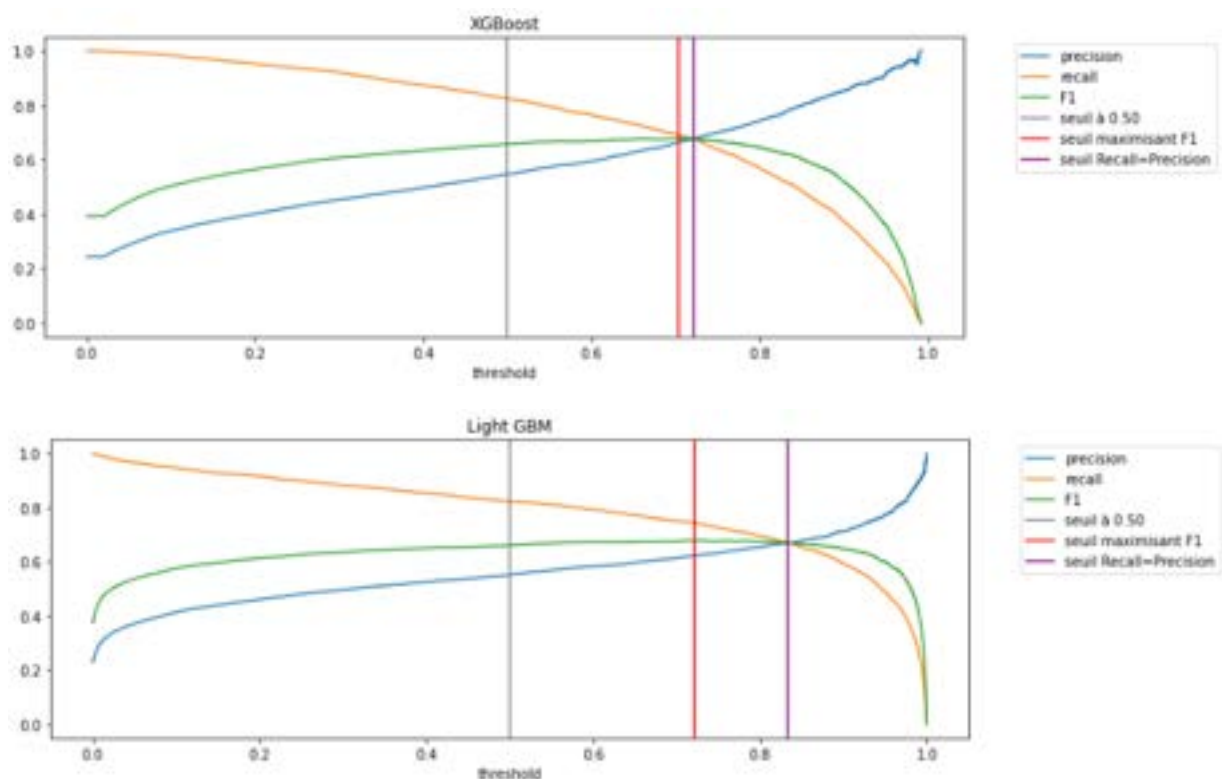


### 3.2.2 - Performances des modèles – Selon le seuil de détection

#### 3 seuils (thresholds) choisis :

- Seuil classique à 0.50
- Seuil pour maximiser le f1-score
- Seuil pour precision = recall

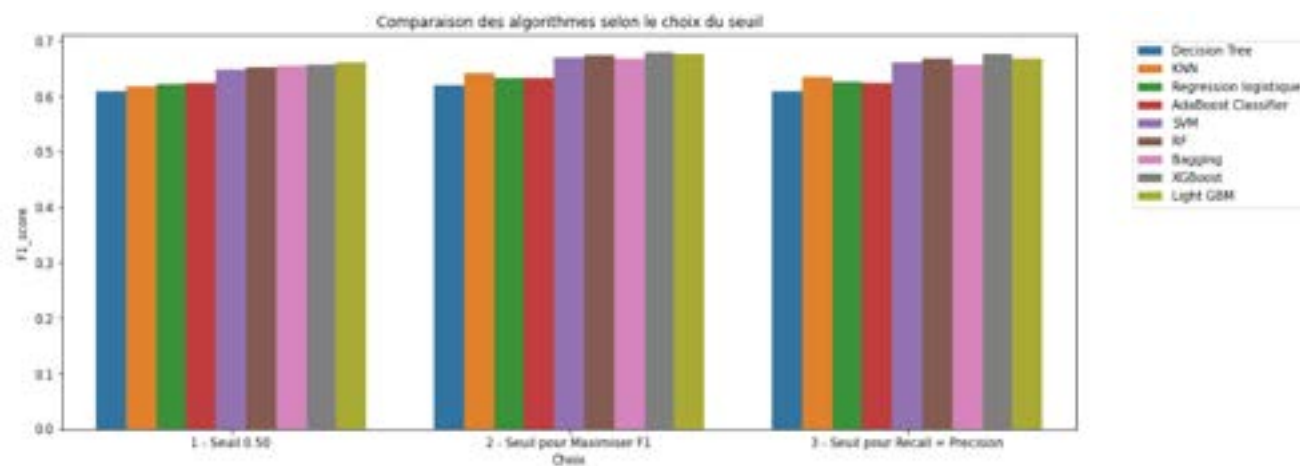
#### 2 exemples graphiques illustrant l'impact du seuil dans le calcul des métriques



## Résultats complets avec les 3 métriques

Choix	threshold	precision	recall	F1_score	Modele
1 - Seuil 0.50	0.500000	0.497462	0.7840	0.608696	Decision Tree
1 - Seuil 0.50	0.500000	0.499753	0.8100	0.618132	KNN
1 - Seuil 0.50	0.500000	0.520559	0.7748	0.622729	Regression logistique
1 - Seuil 0.50	0.500000	0.518165	0.7816	0.623186	AdaBoost Classifier
1 - Seuil 0.50	0.500000	0.534902	0.8184	0.646957	SVM
1 - Seuil 0.50	0.500000	0.539405	0.8268	0.652874	RF
1 - Seuil 0.50	0.500000	0.543646	0.8196	0.653693	Bagging
1 - Seuil 0.50	0.500000	0.546780	0.8252	0.657740	XGBoost
1 - Seuil 0.50	0.500000	0.551854	0.8216	0.660238	Light GBM
2 - Seuil pour Maximiser F1	0.606667	0.544019	0.7168	0.618571	Decision Tree
2 - Seuil pour Maximiser F1	0.659632	0.619523	0.6448	0.631909	Regression logistique
2 - Seuil pour Maximiser F1	0.503342	0.573797	0.7060	0.633070	AdaBoost Classifier
2 - Seuil pour Maximiser F1	0.624320	0.593421	0.7000	0.642320	KNN
2 - Seuil pour Maximiser F1	0.598000	0.612709	0.7328	0.667395	Bagging
2 - Seuil pour Maximiser F1	0.712590	0.647125	0.6932	0.669370	SVM
2 - Seuil pour Maximiser F1	0.607980	0.624362	0.7340	0.674756	RF
2 - Seuil pour Maximiser F1	0.721262	0.621658	0.7440	0.677349	Light GBM
2 - Seuil pour Maximiser F1	0.703967	0.664873	0.6920	0.678165	XGBoost
3 - Seuil pour Recall = Precision	0.654832	0.581480	0.6380	0.608430	Decision Tree
3 - Seuil pour Recall = Precision	0.506383	0.623750	0.6240	0.623875	AdaBoost Classifier
3 - Seuil pour Recall = Precision	0.673146	0.625750	0.6260	0.625875	Regression logistique
3 - Seuil pour Recall = Precision	0.700509	0.634146	0.6344	0.634273	KNN
3 - Seuil pour Recall = Precision	0.663000	0.655777	0.6584	0.657086	Bagging
3 - Seuil pour Recall = Precision	0.742709	0.661335	0.6616	0.661468	SVM
3 - Seuil pour Recall = Precision	0.662000	0.666933	0.6672	0.667067	RF
3 - Seuil pour Recall = Precision	0.833885	0.668533	0.6688	0.668666	Light GBM
3 - Seuil pour Recall = Precision	0.720859	0.675330	0.6756	0.675465	XGBoost

## Graphiquement avec uniquement le f1-score



### 3.3 Conclusion

- La comparaison des algorithmes sur la courbe de ROC nous donne une liste de quatre algorithmes sensiblement plus performants que les autres :

- la Random Forest
- le Bagging
- la XGBoost
- la Light GBM

- Les comparaisons sur le F1\_score en choisissant différents seuils de probabilités (0.50, F1\_max, recall=precision) vont nous conduire à préférer la XGBOOST qui est légèrement plus performante que la lightGBM sur le seuil "recall=precision".



## IV. Interprétabilité de notre modèle final (XGBOOST) - Notebook NB4

L'interprétabilité est importante dès que les résultats d'un modèle influent grandement sur des décisions importantes. En entreprise par exemple, expliquer à des équipes non-initiées le fonctionnement d'un modèle pose toujours son lot de défis.

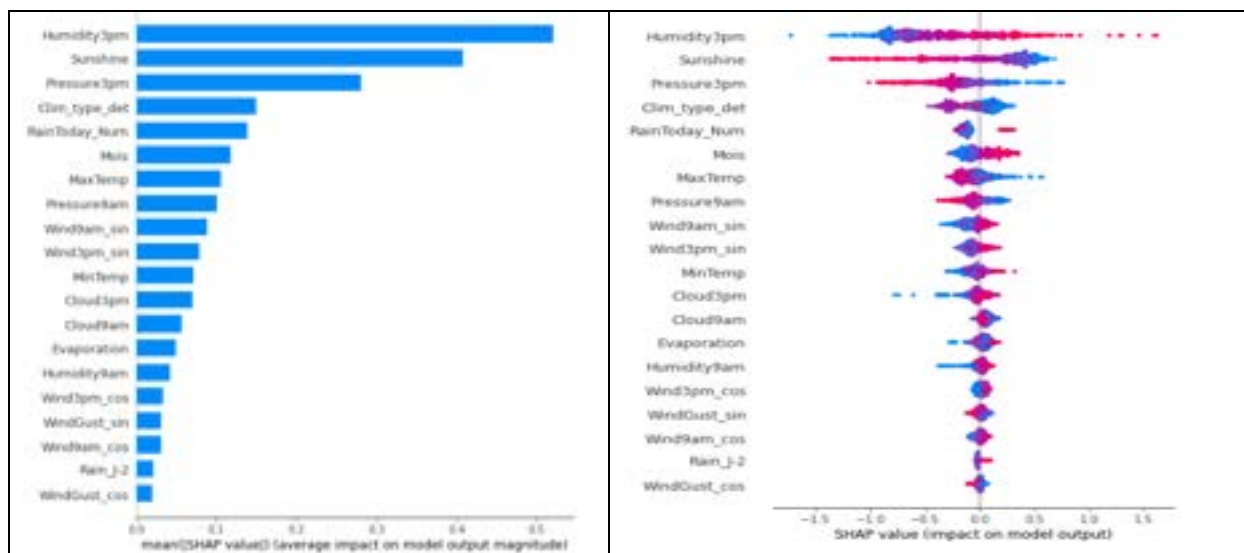
Nous allons explorer plusieurs types d'interprétation dans cette section :

- globale avec le package Shap
- locale avec le package Lime
- globale et locale avec Shapash
- avec des arbres de décision
- avec les outils disponibles de XGBoost.

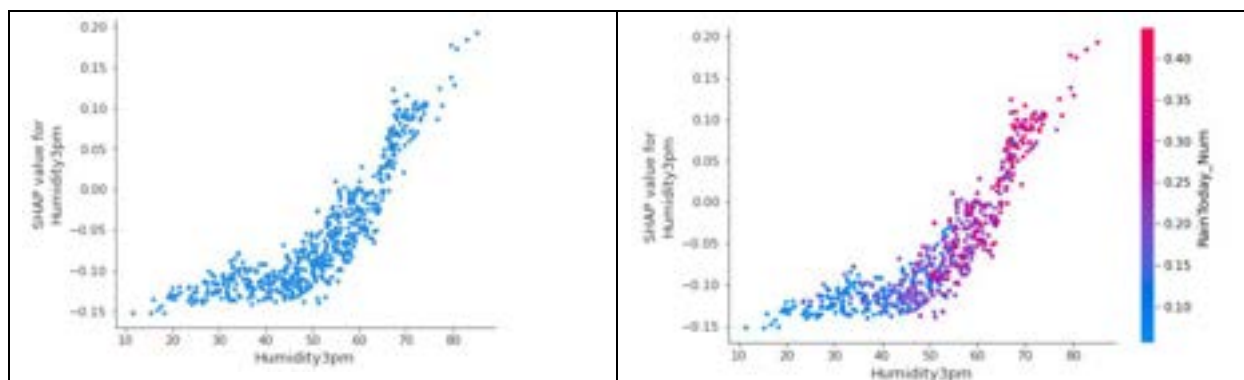
### 4.1 Interprétabilité Globale avec Shap

Dans cette section, nous allons appliquer Shap sur les moyennes mensuelles par station. L'histogramme de gauche nous permet de voir l'importance des facteurs les plus explicatifs de notre modèle. Le graphique de droite nous permet de voir comment chaque facteur influence notre variable cible.

Ainsi, si l'humidité à 3PM est importante, il y a une probabilité plus importante d'avoir de la pluie le lendemain. Le raisonnement est le même pour l'ensoleillement mais quand celui-ci est faible.



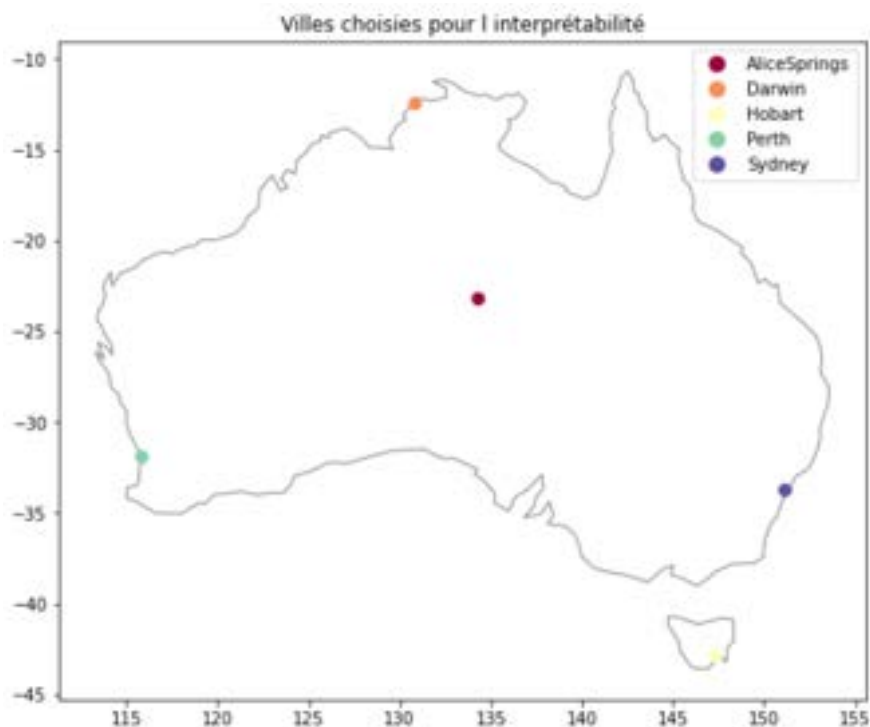
Les graphiques ci-dessous de *dependance plot* montrent l'effet d'une variable sur l'ensemble des données avec la possibilité de colorer les points en fonction d'une variable (exemple ici RainToday\_Num).





## 4.2 Interprétabilité Locale avec Lime

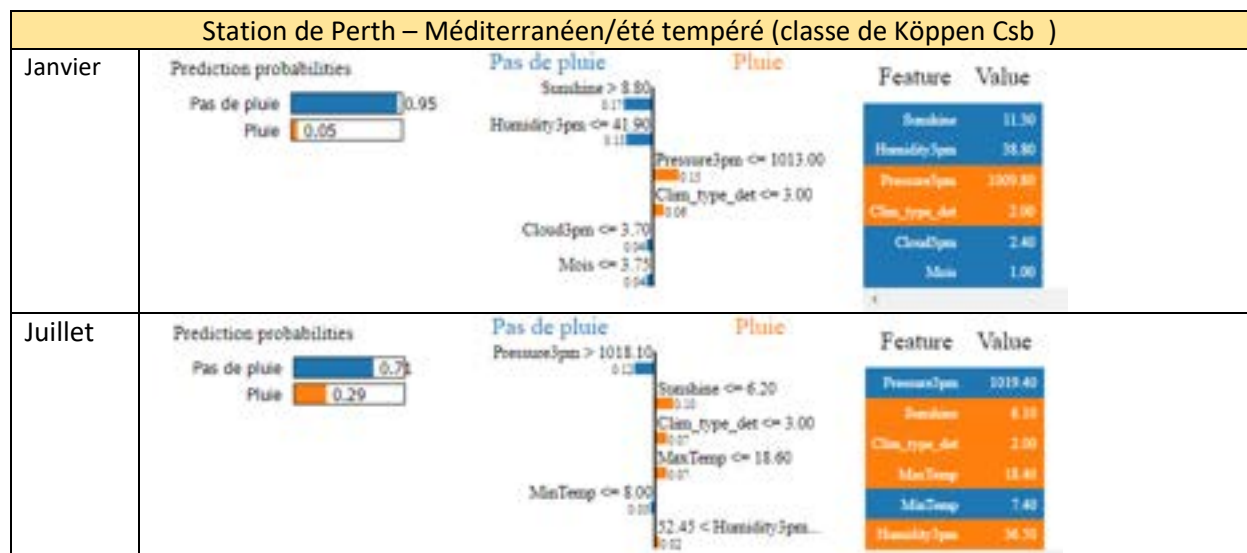
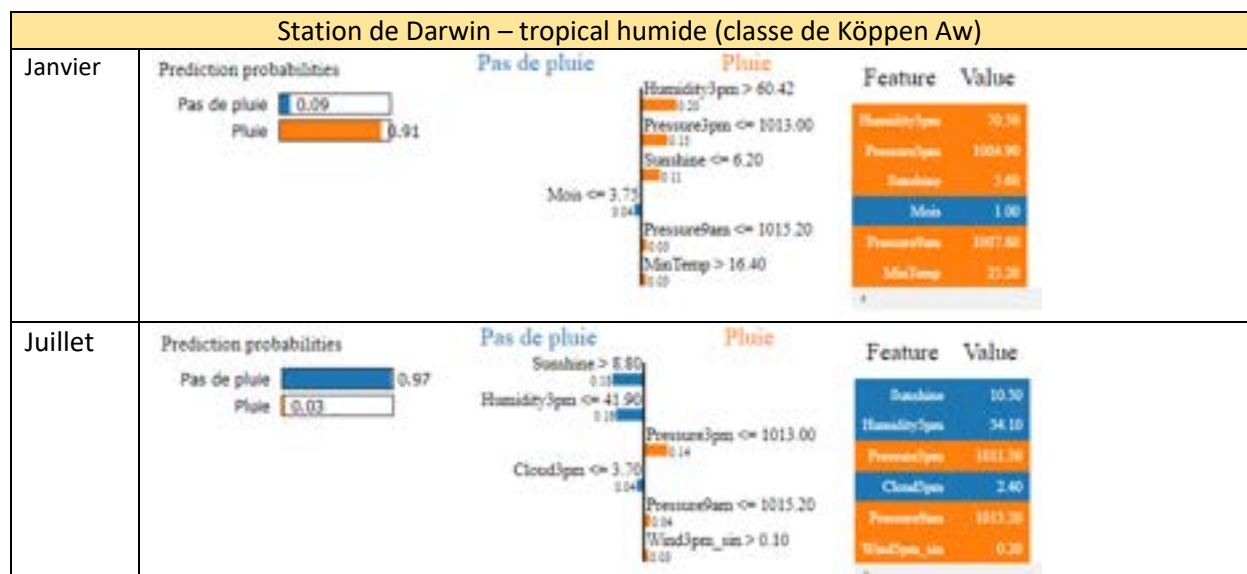
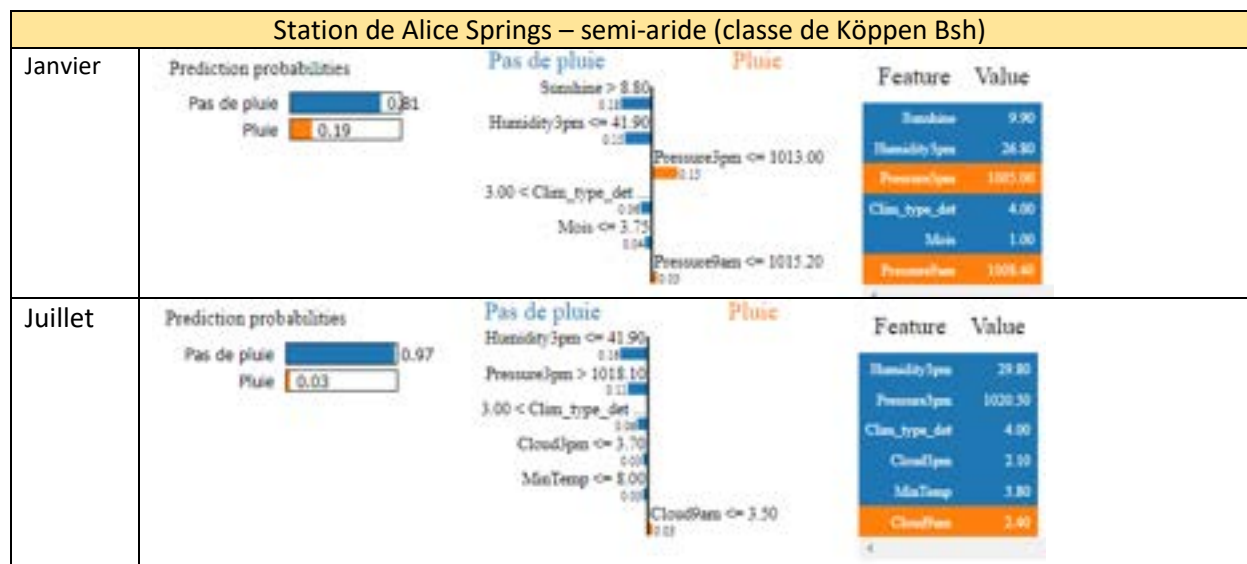
Dans cette section, nous allons appliquer lime sur les moyennes mensuelles par ville. On voit ainsi localement quels sont les facteurs explicatifs de *RainTomorrow*, ces facteurs pouvant être différents pour une même ville mais sur des mois différents.



On voit ainsi que pour Alice Springs, on est quasiment certain qu'il n'y aura pas de pluie le lendemain en raison d'une faible humidité, d'un fort ensoleillement et de hautes pressions. A *contrario*, on est presque certain d'avoir de la pluie en janvier à Darwin.

Station de Sydney – subtropical humide (classe de Köppen Cfa)			
Janvier	Prediction probabilities Pas de pluie 0.57 Pluie 0.43	Pas de pluie Pluie	Feature Value Pressure3pm 1012.19 MaxTemp 27.50 Clim_type_det 3.00 Mois 1.00 Pressure9am 1014.00 Wind9am_sin -0.10
Juillet	Prediction probabilities Pas de pluie 0.61 Pluie 0.39	Pas de pluie Pluie	Feature Value Pressure3pm 1018.90 Humidity3pm 49.50 Sunshine 6.70 MaxTemp 18.00 Clim_type_det 3.00 WindGust_sin -0.20





Station de Hobart (Tasmanie)– Océanique/été tempéré (classe de Köppen Cfb )				
Janvier	Prediction probabilities Pas de pluie 0.76 Pluie 0.24	Pas de pluie 41.90 < Humidity3pm 3.00 < Clim_type_det Mois <= 3.75	Pluie Pressure3pm <= 1013.00 10.03 Pressure9am <= 1015.20 10.03 Wind3pm_uin > 0.20 10.03	Feature Value Pressure3pm 1010.18 Humidity3pm 48.40 Clim_type_det 1.00 Mois 1.00 Pressure9am 1011.56 Wind3pm_uin 0.40
Juillet	Prediction probabilities Pas de pluie 0.42 Pluie 0.58	Pas de pluie 3.00 < Clim_type_det 0.07 MinTemp <= 8.00 0.04 Pressure9am <= 1015.20 10.04 Wind3pm_uin > 0.10 10.05	Pluie Sunshine <= 6.20 0.00 MaxTemp <= 18.60 10.07 Pressure9am <= 1015.20 10.04 Wind3pm_uin > 0.10 10.05	Feature Value Sunshine 0.00 Clim_type_det 1.00 MaxTemp 12.83 MinTemp 1.10 Pressure9am 1015.20 Wind3pm_uin 0.30

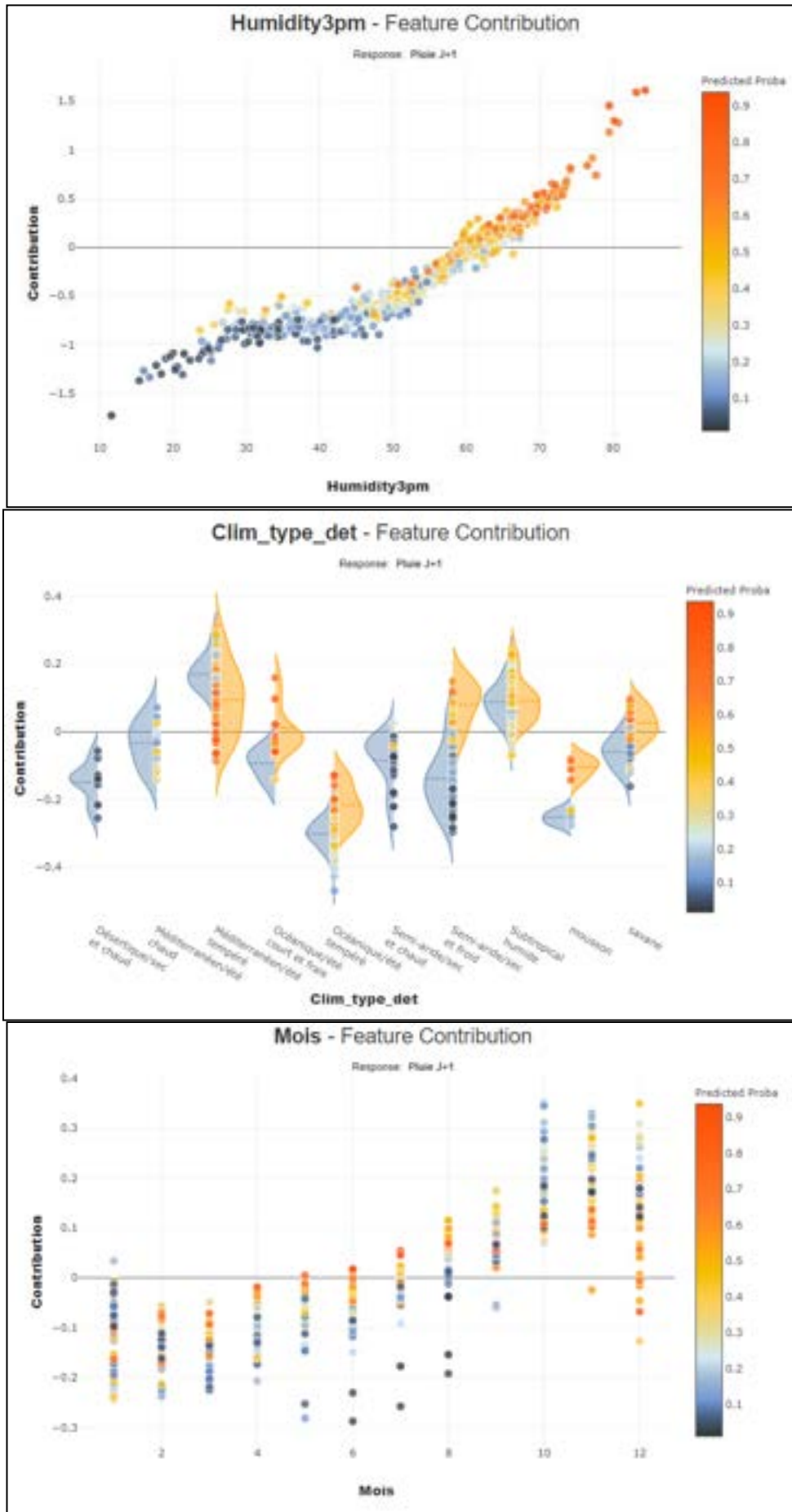
### 4.3 Shapash

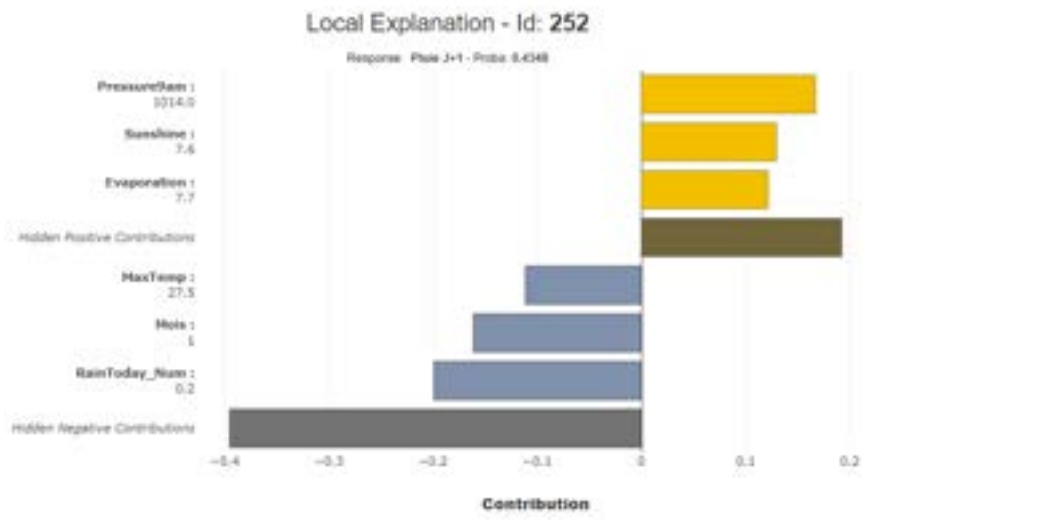
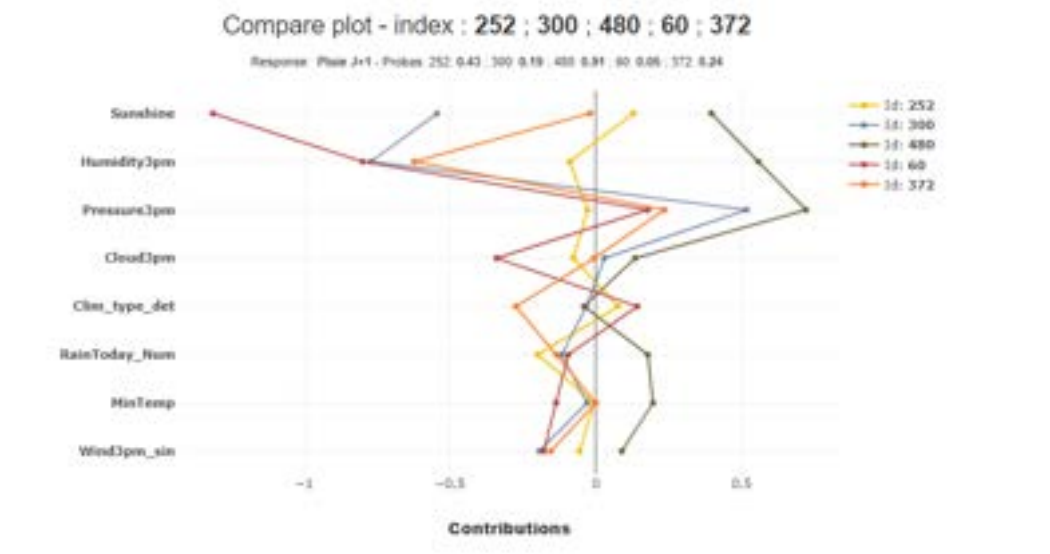
<https://medium.com/oss-by-maif/shapash-une-nouvelle-solution-ossbymaif-pour-une-intelligence-artificielle-plus-transparente-c216f9ddb2e9>

Shapash est une librairie Python qui vise à rendre le Machine Learning intelligible par le plus grand nombre. Concrètement, il s'agit d'une surcouche à d'autres librairies d'intelligibilité (Shap, Lime). Il dispose d'une webapp permettant de visualiser les résultats d'un modèle d'une manière interactive mais aussi d'outils à exécuter en lignes de commande.



→ Globalement - Contribution des *features* principales



<p>Exemple de sydney au mois de janvier</p>	
<p>Comparaison de plusieurs prédictions sur le même graphique</p>	<pre>idx1 = Select([Select("Location" == "Sydney") &amp; (Select("Mois" == 1)).index.tolist()[0] idx2 = Select([Select("Location" == "AliceSprings") &amp; (Select("Mois" == 1)).index.tolist()[0] idx3 = Select([Select("Location" == "Darwin") &amp; (Select("Mois" == 1)).index.tolist()[0] idx4 = Select([Select("Location" == "Perth") &amp; (Select("Mois" == 1)).index.tolist()[0] idx5 = Select([Select("Location" == "Hobart") &amp; (Select("Mois" == 1)).index.tolist()[0]  spl_31.plot.compare_plot(index=[idx1, idx2, idx3, idx4, idx5], max_features=8)</pre> 

#### 4.4 Intérêt graphique du modèle XGBoost

Le modèle XG Boost permet d'afficher des graphiques d'importances des différentes variables utilisées pour entraîner le modèle.

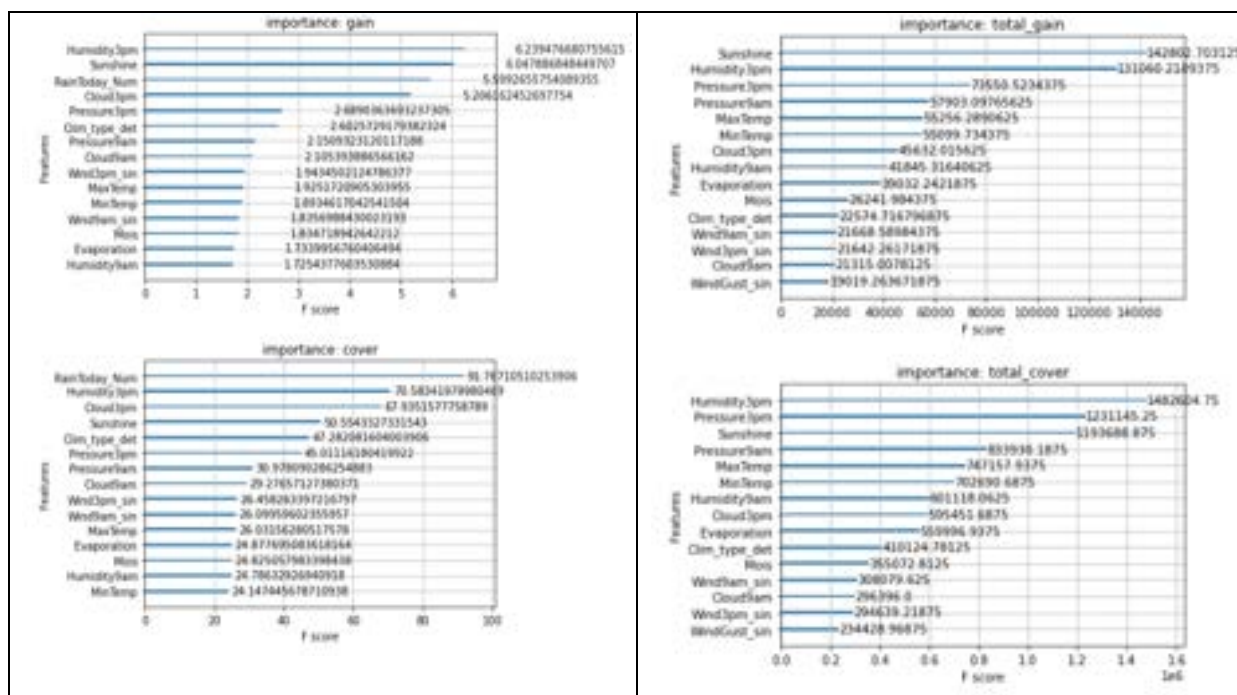
Il existe trois options pour mesurer l'importance des variables dans XG Boost :

'Weight' : le pourcentage représentant le nombre relatif de fois qu'une variable apparaît dans les arbres du modèle.

'Cover' : Le nombre de fois qu'une variable est utilisée pour séparer les données dans l'ensemble des arbres, pondérés par le nombre de données d'entraînement qui passent par ces séparations.

'Gain' : La réduction moyenne de la fonction de perte obtenue lors de l'utilisation d'une variable pour séparer une branche. Cette dernière option est généralement considérée comme la métrique la plus importante pour interpréter l'importance relative des variables dans l'entraînement du modèle.

## Graphique d'importance des variables du modèle XG Boost



On constate une cohérence avec le graphique d'importance fourni par Shap, où les premières variables discriminantes sont les mêmes (Humidity3pm, Pressure3pm et Sunshine).

## 4.5 Intérêt graphique des arbres de décision

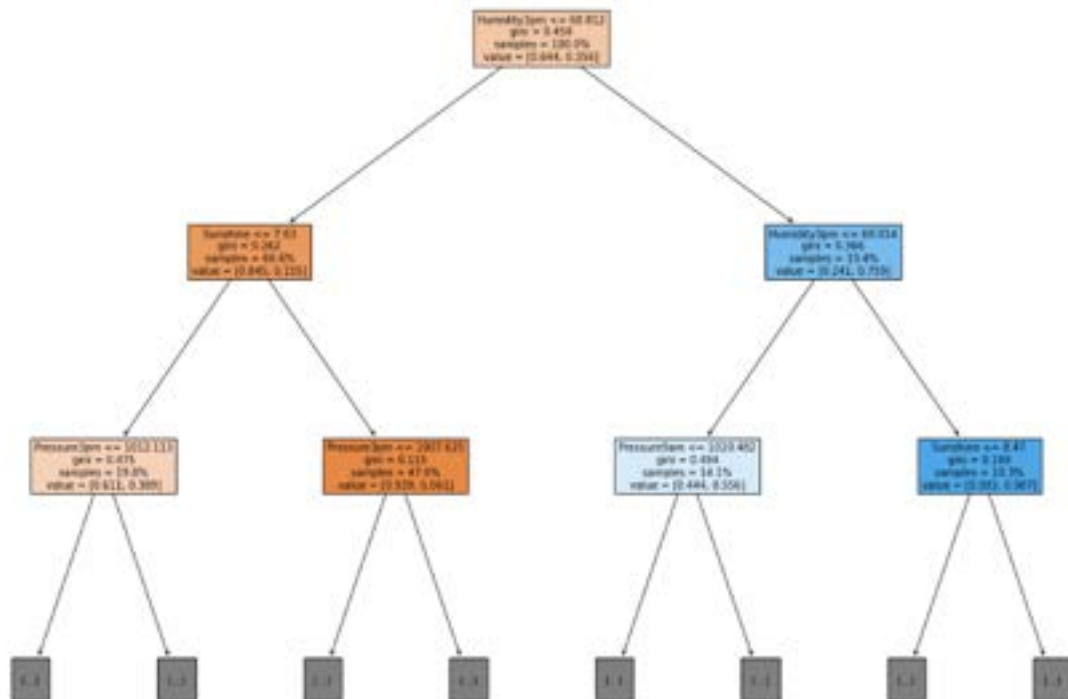
Les arbres de décision ne sont pas les modèles plus performants, mais ils apportent des informations intéressantes sur les variables les plus significatives.

On peut donc faire un arbre de décision sur nos prédictions issues de la XGBoost pour voir le découpage du premier arbre.

On voit que le premier critère est l'humidité à 3pm suivi de Sunshine et de la pression atmosphérique. Pour modéliser RainTomorrow, on constate que le climat n'intervient pas dans les premiers critères.

Importance	
Humidity3pm	0.598348
Sunshine	0.161411
Pressure3pm	0.105370
Pressure9am	0.049233
RainToday_Num	0.020159
MinTemp	0.013777
Wind3pm_sin	0.010341
Cloud3pm	0.009027

## Aperçu de l'arbre de décision





## V. Clustering des villes - Notebook NB5

**Objectif :** Dans cette section, nous allons classer les villes en fonction de leurs caractéristiques météorologiques. Nous avons appliqué une démarche permettant de se rapprocher de la classification des climats de Köppen.

### 5.1 Introduction à la classification des climats de Köppen

La classification de Köppen est une classification des climats fondée sur les précipitations et les températures. Un climat, selon cette classification, est repéré par un code de deux ou trois lettres :

- 1<sup>ère</sup> lettre : type de climat
- 2<sup>ème</sup> lettre : régime pluviométrique
- 3<sup>ème</sup> lettre : variations de températures.

La combinaison de ces sous-classifications donne la classification de climat de Köppen suivante :

Classe	Types de climats
A	<ul style="list-style-type: none"><li>• <a href="#">Équatorial</a> : Af</li><li>• <a href="#">Mousson</a> : Am</li><li>• <a href="#">Savane</a> : Aw, As</li></ul>
B	<ul style="list-style-type: none"><li>• <a href="#">Désertique</a> : BWh, BWk, BWn</li><li>• <a href="#">Semi-aride</a> : BSh, BSk, BSn</li></ul>
C	<ul style="list-style-type: none"><li>• <a href="#">Subtropical humide</a> : Cfa, Cwa</li><li>• <a href="#">Océanique</a> : Cfb, Cwb, Cfc, Cwc</li><li>• <a href="#">Méditerranéen</a> : Csa, Csb, Csc</li></ul>
D	<ul style="list-style-type: none"><li>• <a href="#">Continental humide</a> : Dfa, Dwa, Dfb, Dwb</li><li>• <a href="#">Subarctique</a> : Dfc, Dwc, Dfd, Dwd</li><li>• <a href="#">Continental méditerranéen</a> : Dsa, Dsb, Dsc, Dsd</li></ul>
E	<ul style="list-style-type: none"><li>• <a href="#">Toundra</a> : ET</li><li>• <a href="#">Inlandsis</a> ou <a href="#">calotte glaciaire</a> : EF</li></ul>

Nous allons donc par l'intermédiaire des méthodes de clustering essayer de regrouper les villes d'Australie par type de climat et les comparer au climat de Köppen auxquelles elles sont usuellement affectées.

La 1<sup>ère</sup> lettre est la définition générale du climat. Nous allons donc faire une classification (kmeans) sur les données mensuelles des stations avec l'ensemble des données météo disponibles.

Les deux lettres suivantes concernent d'une part le régime pluviométrique et d'autre part les variations de température. Dans cette étape, nous allons tester une classification adaptée aux séries temporelles qui s'applique indicateur par indicateur.

## 5.2 1<sup>ère</sup> lettre : type de climat – Notebook NB5.1

Algorithme : kmeans

Jeu de données : weatherAUS\_imputer (valeurs manquantes interpolées par KNN Imputer)

**Objectif :** Classifier les villes selon leurs caractéristiques météorologiques globales correspondant à la 1<sup>ère</sup> lettre de la classification de Köppen.

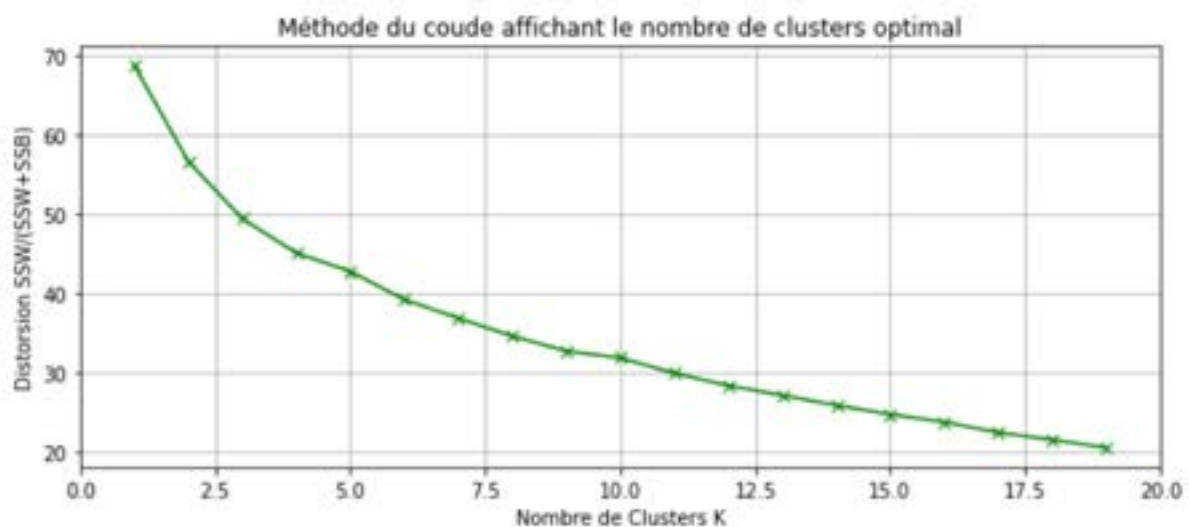
Cette section traite de la classification des villes à partir des données météo disponibles. On va traiter notre dataframe avant d'appliquer la kmeans pour avoir une ligne par villes avec les indicateurs disponibles en colonne.

La saisonnalité des climats étant importante, on va calculer des moyennes mensuelles par mois et mettre le tout en colonne. Ainsi, pour chaque indicateur, on aura 12 colonnes correspondants aux moyennes mensuelles.

Liste des indicateurs utilisés :

'MinTemp','MaxTemp','Rainfall','Evaporation','Sunshine','WindGustSpeed','WindSpeed9am','WindSpeed3pm','Humidity9am','Humidity3pm','Pressure9am','Pressure3pm','Cloud9am','Cloud3pm','Temp9am','Temp3pm','RainToday\_Num'.

Ensuite, on va utiliser l'algorithme de clustering (kmeans) en itérant sur le nombre de classes à tester et affichant les distorsions en fonction du nombre de classes.



Dans notre cas, le coude n'est pas très visible mais on voit un aplatissement pour 10 classes. On va choisir 10 classes dans notre classification, ce qui correspond aussi aux nombres de classes issues de la classification de Köppen.

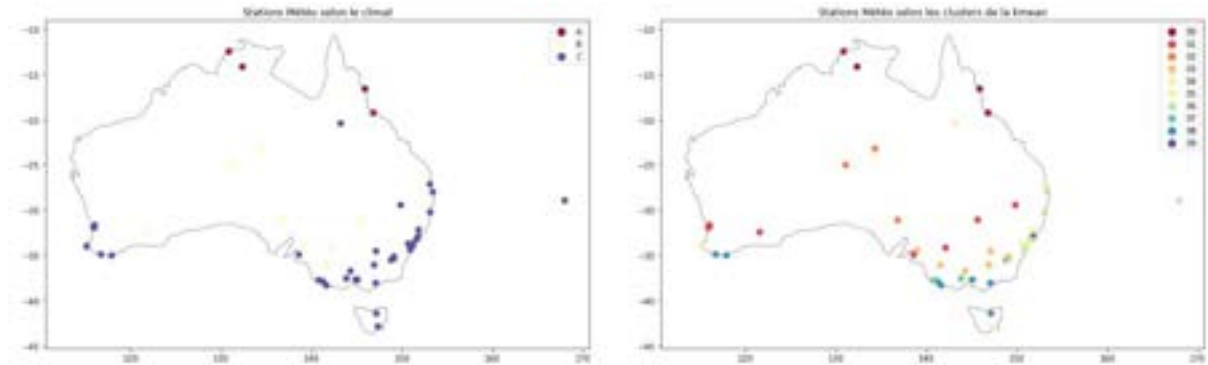
Quand on compare nos classes avec les climats, on constate que les climats « extrêmes » sont bien identifiés (désertique/savane/mousson) et que c'est un peu plus disparate sur des climats plus tempérés (« Méditerranéen/été tempéré », « Océanique/été tempéré », « Subtropical humide »).

En se focalisant uniquement sur la 1<sup>ère</sup> lettre du climat Köppen, on remarque que l'on arrive facilement à isoler les climats de type A. Les climats de type B et C ont les classes 2 et 8 en commun, ce qui ne permet de conclure sur la performance de notre classification à ce stade.



classes	0	1	2	3	4	5	6	7	8	9
Climat_Koppen_1L										
A	0	0	0	0	0	0	4	0	0	0
B	0	0	3	0	0	0	0	3	2	0
C	4	7	5	5	3	7	0	0	5	1

La carte ci-dessous illustre nos propos sur les modalités communes entre classification de Köppen – 1<sup>ère</sup> lettre – et les classes issues de la kmeans.



### 5.3 2<sup>ème</sup> lettre : régime pluviométrique – Notebook 5.2

Algorithme : *TimeSeriesKmeans / Metric dtw*

Jeu de données : *weatherAUS\_imputer* (valeurs manquantes interpolées par KNN Imputer)

**Objectif :** Classifier les villes selon leurs régimes pluviométriques correspondant à la 2<sup>ème</sup> lettre de la classification de Köppen.

Cette section traite de la classification des régimes pluviométriques des villes à partir de séries temporelles.

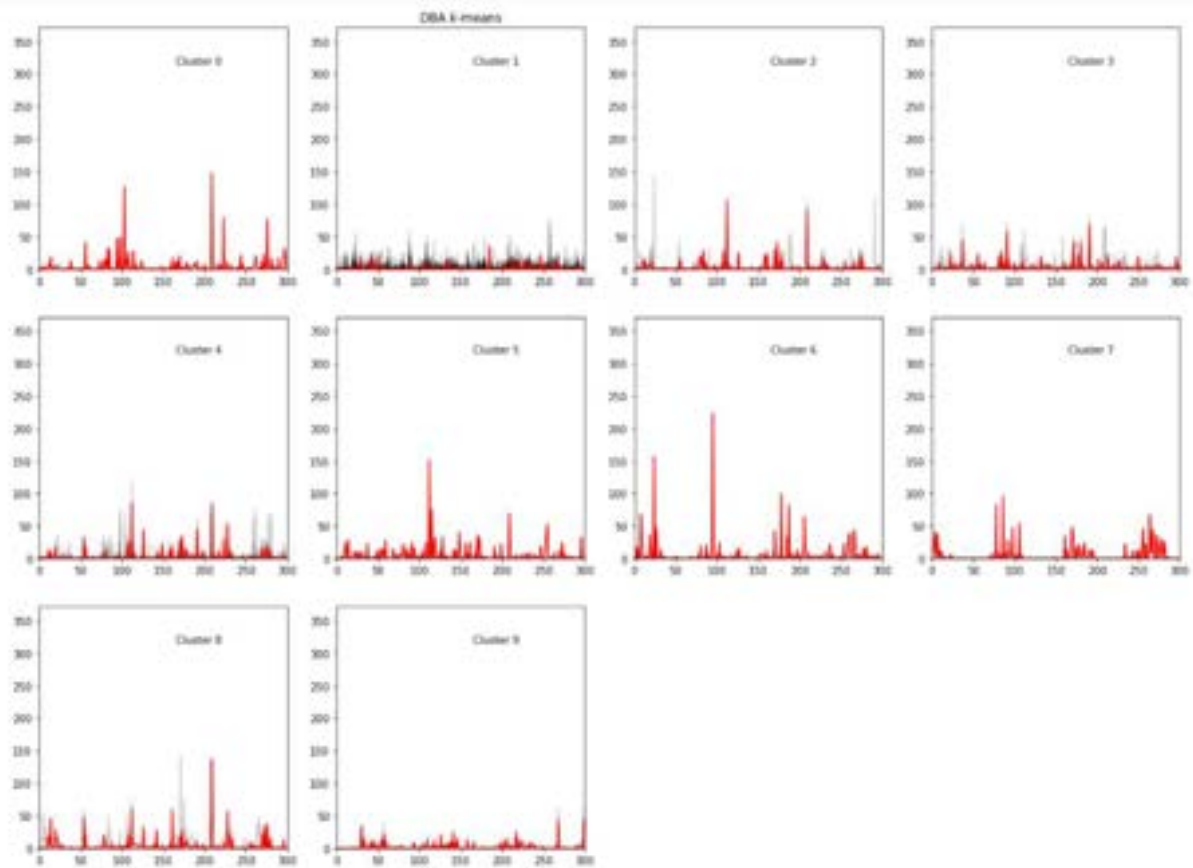
Ici on va utiliser l'algorithme Kmeans applicable aux séries temporelles disponible dans le package tslearn. Le lien suivant explique les principes de fonctionnement de cet algorithme et sa mise en œuvre :

<https://towardsdatascience.com/how-to-apply-k-means-clustering-to-time-series-data-28d04a8f7da3>

On va tout d'abord créer un jeu de données avec des séries temporelles de précipitations pour chacune des villes. Pour que les séries temporelles soient de même dimension et porte sur la même période, nous avons considéré 3 ans et demi de précipitations à partir de janvier 2014. Sur cette période, nous n'avons aucun relevé manquant (un relevé par jour, pas de saut de jour).

Les graphiques suivants présentent les différents clusters identifiés avec pour chacun :

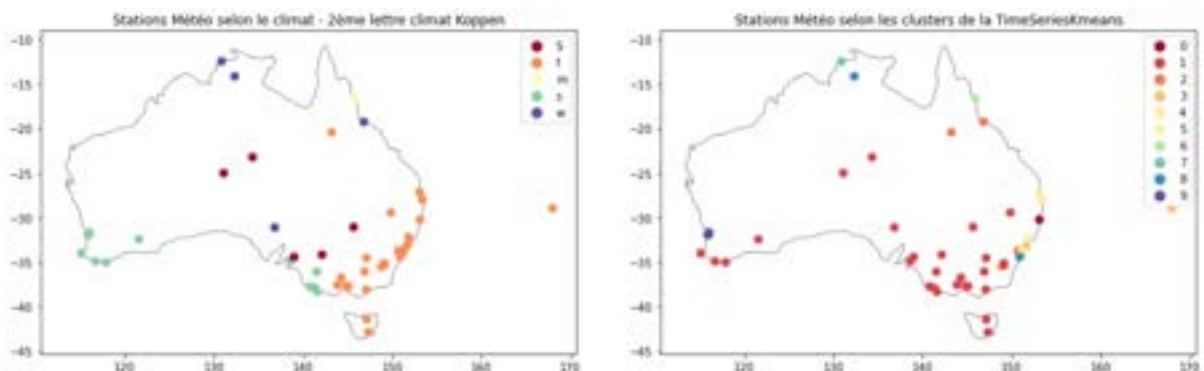
- Les séries temporelles en gris clair,
- et la série temporelle « centroïd » en rouge.



La comparaison de nos clusters avec les classifications des régimes pluviométriques est reportée dans le tableau suivant :

	Clusters	0	1	2	3	4	5	6	7	8	9
Climat_Koppen_2L											
S	0	5	0	0	0	0	0	0	0	0	0
f	1	13	3	3	5	1	0	0	1	0	0
m	0	0	0	0	0	0	1	0	0	0	0
s	0	9	0	0	0	0	0	0	0	0	3
w	0	1	1	0	0	0	0	1	1	0	0

Comme on pouvait le pressentir sur la représentation du cluster 1, il ne dissocie pas suffisamment les régimes pluviométriques de Köppen puisque 4 des 5 régimes y sont présents. Le régime « m : mousson » est lui bien identifié dans le cluster 6. Le régime f – régime associé au climat humide, si on exclut le cluster 1, se retrouve seul dans de nombreux clusters. La carte ci-dessous permet d'illustrer notre propos sur les modalités communes entre climat de Köppen et classes issues de la TimeSeriesKmeans.



## 5.4 3ème lettre : variations de températures – Notebook 5.3

Algorithme : *TimeSeriesKmeans / Metric dtw*

Jeu de données : *weatherAUS\_imputer* (valeurs manquantes interpolées par KNN Imputer)

**Objectif :** Classifier les villes selon la variation des températures correspondant à la 3<sup>ème</sup> lettre de la classification de Köppen.

Cette section traite de la classification des variations de températures des villes à partir de séries temporelles.

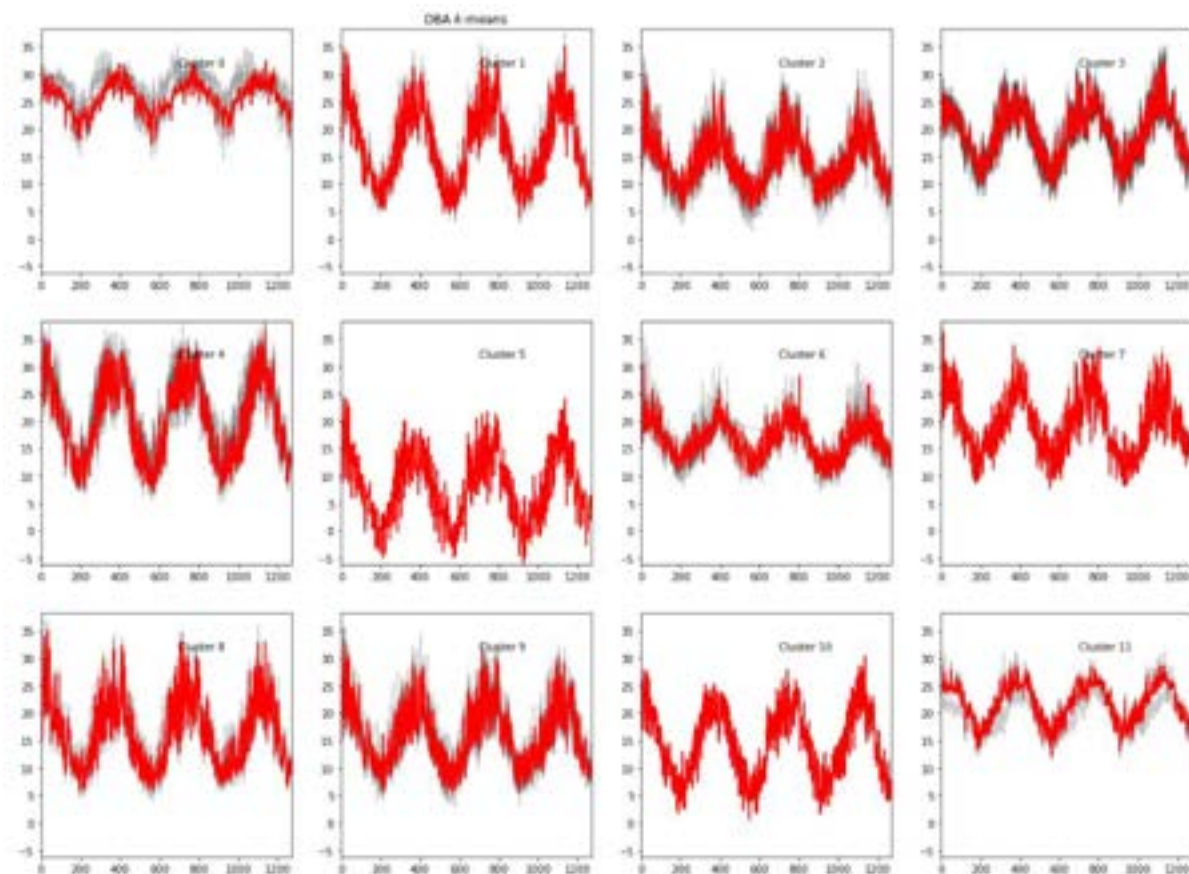
Tout comme pour l'étude sur le régime pluviométrique, on va utiliser l'algorithme Kmeans applicable aux séries temporelles disponible dans le package *tslearn*.

On va tout d'abord créer un jeu de données avec des séries temporelles de températures moyennes construits à partir des relevés de températures minimales et maximales pour chacune des villes.

On étudiera la même période que pour l'étude du régime pluviométrique.

Les graphiques suivants présentent les différents clusters identifiés avec pour chacun :

- Les séries temporelles en gris clair,
- et la série temporelle « centroïd » en rouge.

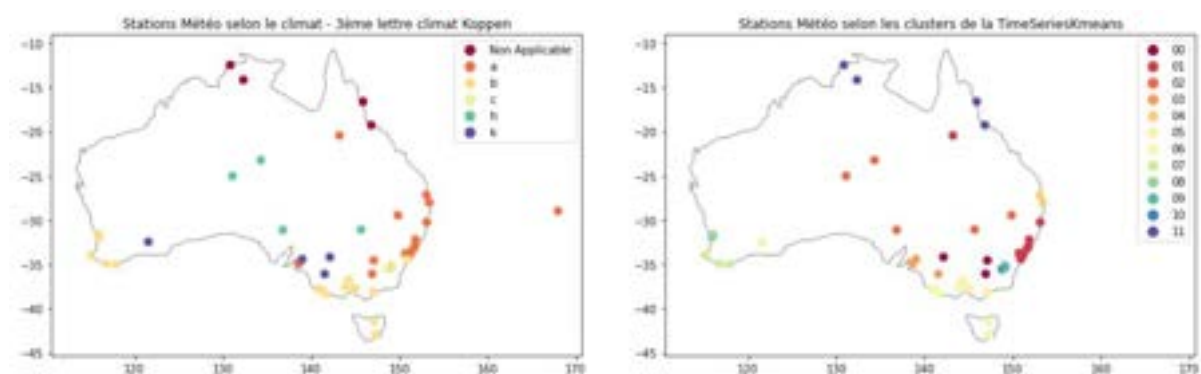


La comparaison de nos clusters avec les classifications des variations de température est reportée dans le tableau suivant :

Clusters_labels	00	01	02	03	04	05	06	07	08	09	10	11
Climat_Koppen_T												
Non Applicable	0	0	0	0	0	0	0	0	0	0	0	4
a	2	9	1	1	3	0	0	0	0	0	0	0
b	0	1	0	0	0	4	6	4	3	2	0	0
c	0	0	0	0	0	0	0	0	0	0	1	0
h	0	0	4	0	0	0	0	0	0	0	0	0
k	1	0	0	2	0	1	0	0	0	0	0	0

L'ensemble des classifications des variations de température est dans l'ensemble bien exécuté.

La carte ci-dessous permet d'illustrer notre propos sur les modalités communes entre climat de Köppen et classes issues de la TimeSeriesKmeans.



## 5.5 Combinaison des classifications – Notebook 5.4

**Objectifs :** À partir des classifications obtenues précédemment, nous allons comparer les clusters identifiés aux climats de Köppen un à un.

En regroupant dans un tableau l'ensemble des classifications obtenues pour chacune des localisations, nous obtenons les résultats sous la forme suivante :

	Location	Koppen	1L	2L	3L	Global_Clusters
0	Adelaide	Csa	1	1	3	1-1-3
1	Albany	Csb	8	1	7	8-1-7
2	Albury	Cfa	3	1	0	3-1-0
3	AliceSprings	BSh	2	1	2	2-1-2
4	BadgerysCreek	Cfa	4	2	1	4-2-1
5	Ballarat	Cfb	7	1	6	7-1-6
6	Bendigo	Cfb	3	1	5	3-1-5
7	Brisbane	Cfa	4	4	4	4-4-4

On dénombre ainsi 32 clusters globaux différents pour 11 classification de Köppen

En encodant la variable Global\_Clusters, de 0 à 31, on obtient pour chaque climat de Köppen les correspondances suivantes :

Global_Clusters_enc 1	
Koppen	
Am	1

Il existe une seule localisation classée Am et on ne la retrouve que sous un seul cluster. Nous avons réussi à l'isoler.

Global_Clusters_enc 0 2 3				
Koppen				
Aw	1	1	1	1

Les localisations classées Aw sont les seuls à se retrouver sous les clusters 0, 2 et 3.

Global_Clusters_enc 5 9		
Koppen		
BSh	1	2

Global_Clusters_enc 9	
Koppen	
Bwh	1

Isolé en cluster 5, le climat de Köppen BSh fait aussi partie du cluster 9 similaire à Bwh, climat très proche mais aussi avec un climat de type Cfa dont une occurrence se retrouve classée sous le cluster 5.

Global_Clusters_enc 4 11		
Koppen		
BSk	1	1

Global_Clusters_enc 7 11		
Koppen		
Bsk	1	1

Les climats de Köppen BSk et Bsk sont assez proches ce qui explique qu'ils soient tous deux regroupés sous le cluster global 11.

Global_Clusters_enc 5 10 14 15 19 20 21 22 23 24 31												
Koppen												
Cfa	1	2	3	1	2	1	1	1	1	1	1	1

Hormis le cluster 5, tous les autres clusters sont propres au climat Cfa.

Global_Clusters_enc 12 13 16 17 18 25 26 28 29									
Koppen									
Cfb	1	2	1	1	1	1	1	2	1

Global_Clusters_enc	8	18	26	29	30
Koppen					
Csb	3	1	1	2	2

Les clusters 18,26,29 sont communs à Cfb et Csb, climat assez proches. Tous les autres clusters sont propres à Cfb et Csb.

Global_Clusters_enc 27	
Koppen	
Cfc	1

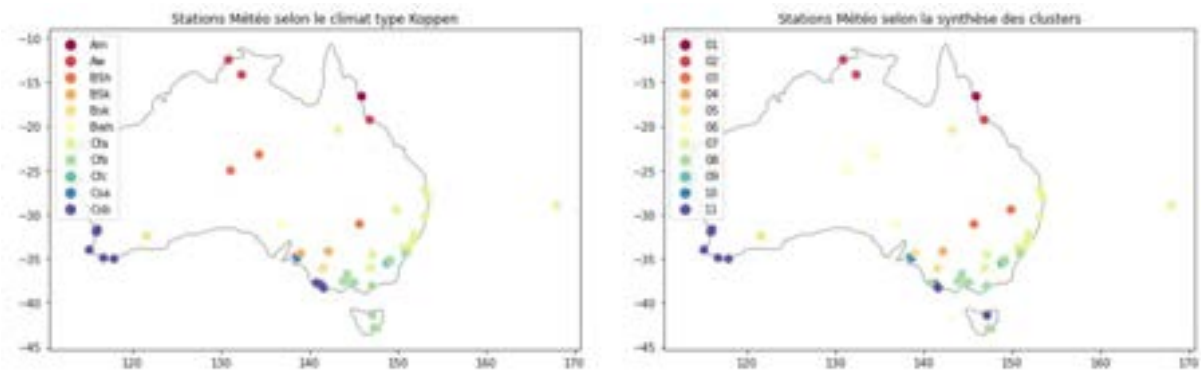
Il existe une seule localisation classée Cfc et on ne la retrouve que sous un seul cluster. Nous avons réussi à l'isoler.

## Köppen

Csa 1

Il existe une seule localisation classée Csa et on ne la retrouve que sous un seul cluster. Nous avons réussi à l'isoler.

Pour synthèse, nous avons regroupé les clusters assignés aux climats de Köppen à la suite de la comparaison ci-dessus. Les résultats sont représentés sur la carte ci-dessous.



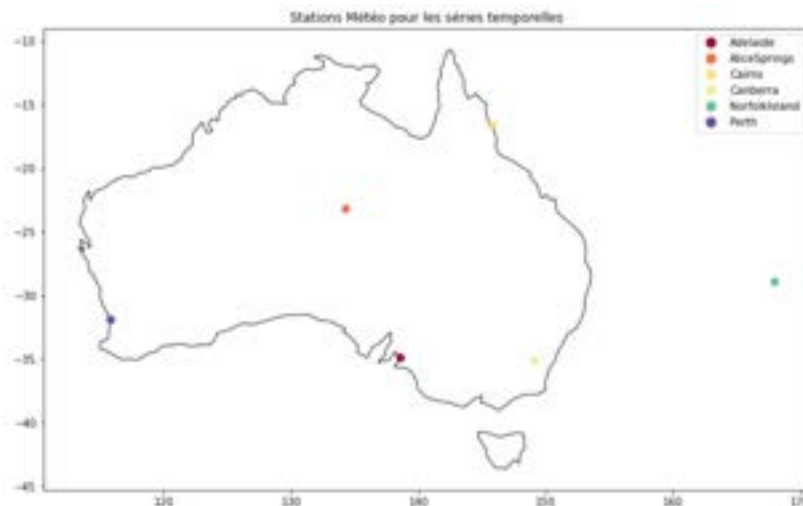
**Conclusion :** La méthodologie de clustering mise au point lors de cette itération s'est basée sur une approche en 3 étapes orientée par une approche métier : la classification de Köppen. Les résultats obtenus sont très satisfaisants avec pas ou peu de classes communes d'un climat de type Köppen à un autre.

## VI. Séries Temporelles – Notebook NB6

Cette section traite des séries temporelles sur différents indicateurs. Notre choix s'est porté sur les indicateurs suivants : *RainFall*, *Humidity3pm*, *MaxTemp*. Deux études ont été menées :

### → 6.1 : Étude sur sept villes représentatives des climats australiens.

Villes	Climat	Classe de Köppen
Canberra	océanique	Cfb
NorfolkIsland	subtropical / océanique	Cfa
Darwin	tropical humide	Aw
Cairns	tropical humide	Am
Perth	méditerranéen	Csb
Adelaide	méditerranéen	Csa
AliceSprings	sec	BSh



### → 6.2 : Étude sur deux climats aux saisons des pluies opposées, en regroupant l'ensemble des stations. Cette étude se limitera à *Rainfall*.

Climat	Classes de Köppen
méditerranéen	Csa + Csb
tropical humide	Aw + Am

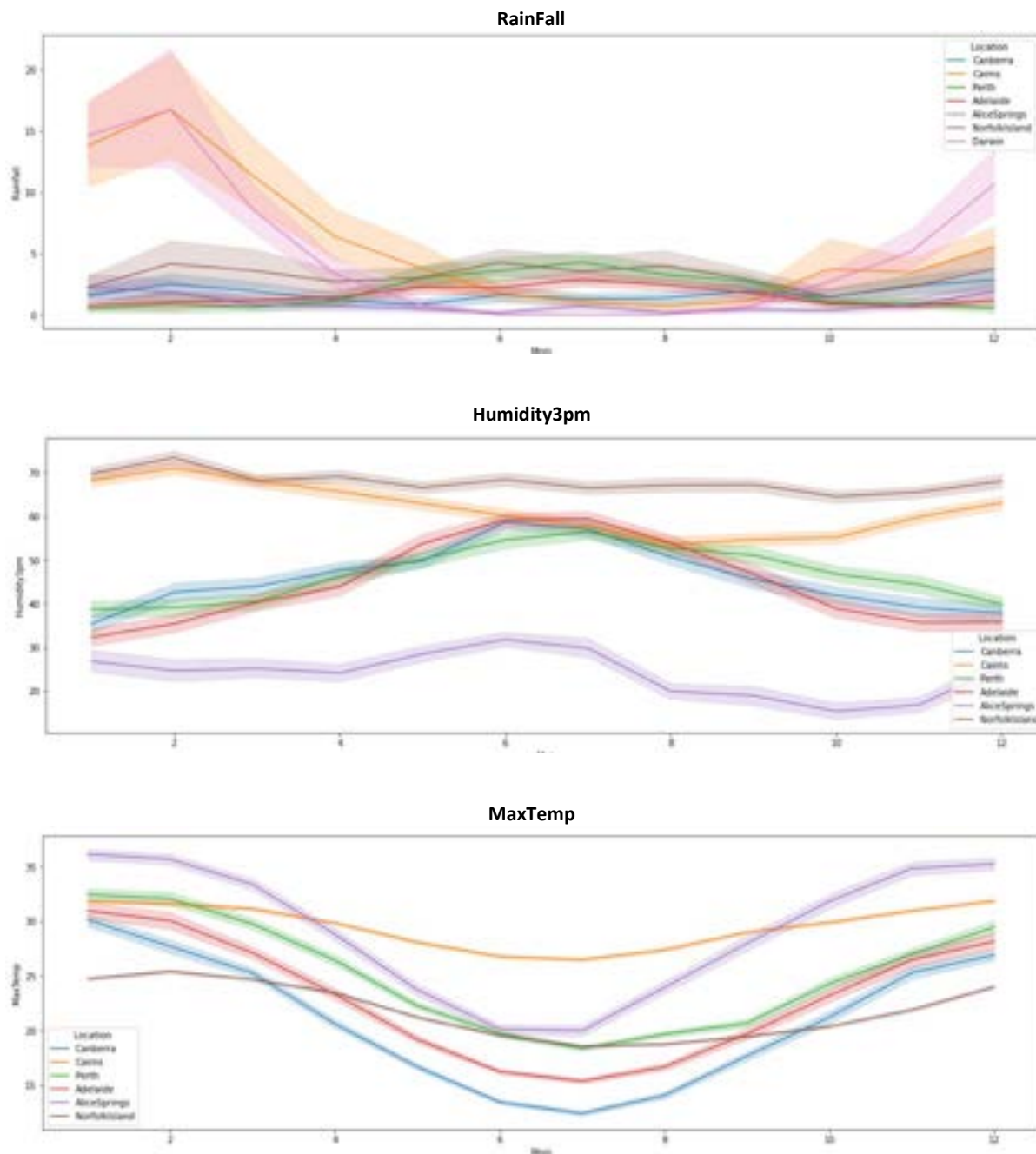
#### Méthodologie :

- Interpolation des valeurs manquantes sur les données quotidiennes.
  - Prévisions faites sur les données mensuelles.
  - Conservation des 24 derniers mois comme base de validation des modèles.
  - Algorithmes testés :
    - Autoarima : pour trouver les meilleurs paramètres des SARIMA
    - SARIMAX : pour appliquer notre modèle final (qui peut être ajusté par rapport à l'Autoarima)
    - Prophet (algorithme de Facebook) en complément de SARIMAX.
  - Comparaison des performances des modèles :
    - Deux métriques de mesure de l'erreur :
      - RMSE (erreur moyenne quadratique) : MaxTemp et Humidity3pm
      - WMAPE (Weighted Mean Absolute Percentage Error) pour RainFall
- => Métrique intéressante pour évaluer les erreurs lorsque les valeurs réelles sont nulles ou proches de zéro. (<https://resdntalien.github.io/blog/wmape/>)
- Remarque : D'autres métriques, telle que la MAE, ont été calculées. Elles présentent toutes des résultats concordants pour l'ensemble des modèles testés et ne seront pas présentées dans le rapport.
- Pourcentage de corrélation de Pearson entre les valeurs réelles et prédites.



## 6.1 Résultats mensuels pour les sept villes

### Visualisation de l'évolution des moyennes mensuelles pour les trois indicateurs



#### **Observations et interprétations :**

La saisonnalité de *Rainfall* est particulièrement marquée pour Cairns et Darwin avec un pic de précipitations important en février. Ces deux villes étant situées en climat tropical, elles possèdent une période de mousson importante en été.

Pour *Humidity3pm*, la saisonnalité n'est pas très marquée mais les niveaux sont bien différents entre Alice Springs (climat sec) et Norfolk Island (climat humide).

*MaxTemp* possède une saisonnalité importante pour les villes situées au sud (climats méditerranéen et océanique), tandis que les villes situées plus proche de l'équateur (Cairns et Darwin – climat tropical) présentent un hiver beaucoup plus doux et donc une saisonnalité moins marquée.

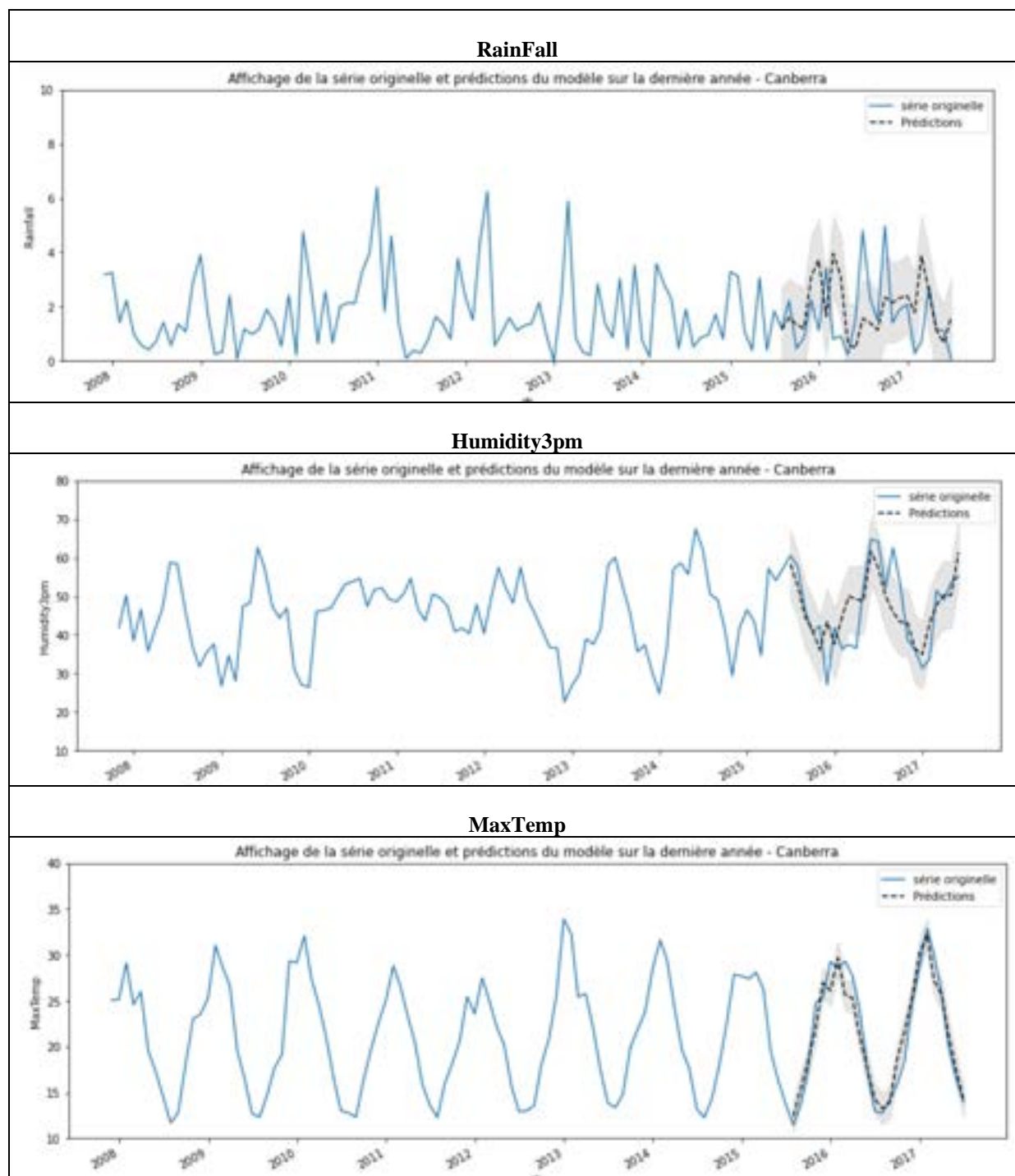
Les deux sous-sections suivantes détaillent les résultats obtenues pour deux villes : Canberra et Cairns.



En vert, les meilleurs modèles au sens de la métrique choisie.\*

Indicateur	Métrique	SARIMA	Prophet
RainFall	WMAPE	Additif - 72,7 %	73,22 %
	Corrélation (Pearson)	19,5 %	3,7 %
Humidity 3pm	RMSE	9,18	7,56
	Corrélation (Pearson)	70,0 %	71,2 %
MaxTemp	RMSE	1,925	1,922
	Corrélation (Pearson)	97,0 %	96 %

\* sauf mention contraire, les modèles sont multiplicatifs



En vert, les meilleurs modèles au sens de la métrique choisie.\*

Indicateur	Métrique	SARIMA	Prophet
RainFall	WMAPE	Additif – 57,8 %	53,7 %
	Corrélation (Pearson)	54,1 %	61,0 %
Humidity 3pm	RMSE	6,089	6,093
	Corrélation (Pearson)	66,15 %	61,2 %
MaxTemp	RMSE	0,956	1,163
	Corrélation (Pearson)	93,4 %	93,4 %

\* sauf mention contraire, les modèles sont multiplicatifs

### RainFall



### Humidity3pm



### MaxTemp



Comme on pouvait s'y attendre, les variations aléatoires quotidiennes rendent les prédictions plus difficiles sur *Rainfall* que sur *MaxTemp*, comme le montre la superposition des courbes des prédictions et de la série originelle. Pour *MaxTemp*, le coefficient de corrélation dépasse en effet 90 % pour tous les modèles. *Humidity3pm* présente sur ce point, un profil intermédiaire.

Pour les trois indicateurs météorologiques, les performances sont meilleures sur Cairns que sur Canberra. La différence entre les deux villes est particulièrement marquée pour *Rainfall*, avec un coefficient de corrélation de 61 % pour Cairns (comparable à celui d'*Humidity*), alors qu'il n'est que de 20 % pour Canberra. Cette différence peut s'expliquer si l'on prend en compte le climat des deux villes. Cairns présente en effet un climat tropical, avec des saisons plus marquées en termes de précipitations que Canberra, dont le climat est océanique.

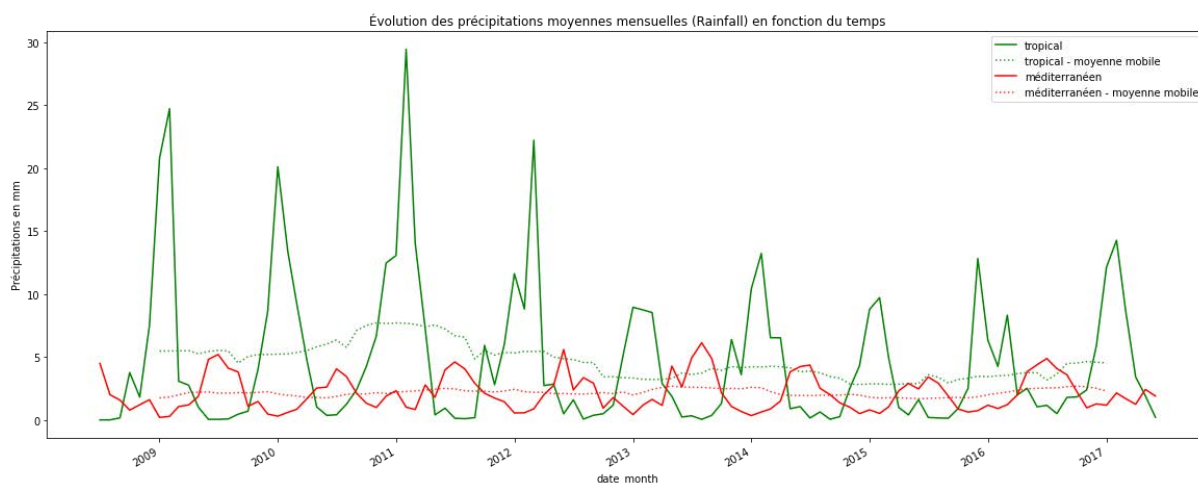
## 6.2 Étude de la saisonnalité de *Rainfall* sur deux climats

Ici on s'intéresse non plus à une ville mais à la moyenne mensuelle de l'ensemble des villes d'un climat donné.

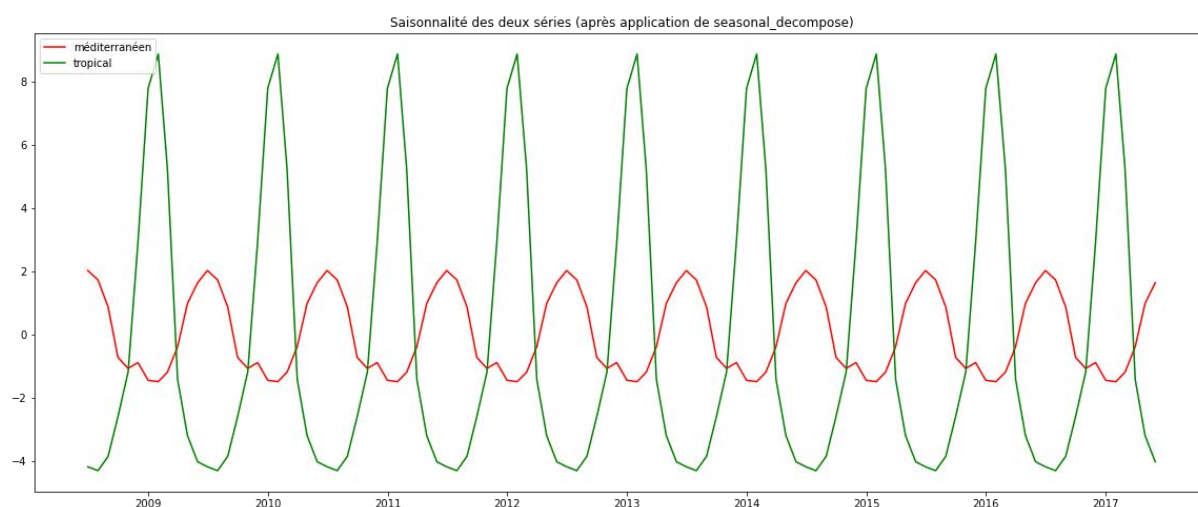
**Hypothèse de travail :** La variable *Rainfall* présente une forte périodicité pour les climats caractérisés par une période de mousson : climat tropical (Aw + Am) et climat méditerranéen (Csa + Csb).

La période de mousson est différente en climat méditerranéen (mousson hivernale) et en climat tropical (mousson estivale). Il est donc nécessaire d'étudier ces deux climats séparément.

La méthodologie est la même que celle utilisée pour les analyse par ville.



### Visualisation et décomposition des séries :



### **Observations :**

Les graphiques confirment notre hypothèse : les deux séries possèdent une forte saisonnalité mais avec un décalage d'une demi-période environ.

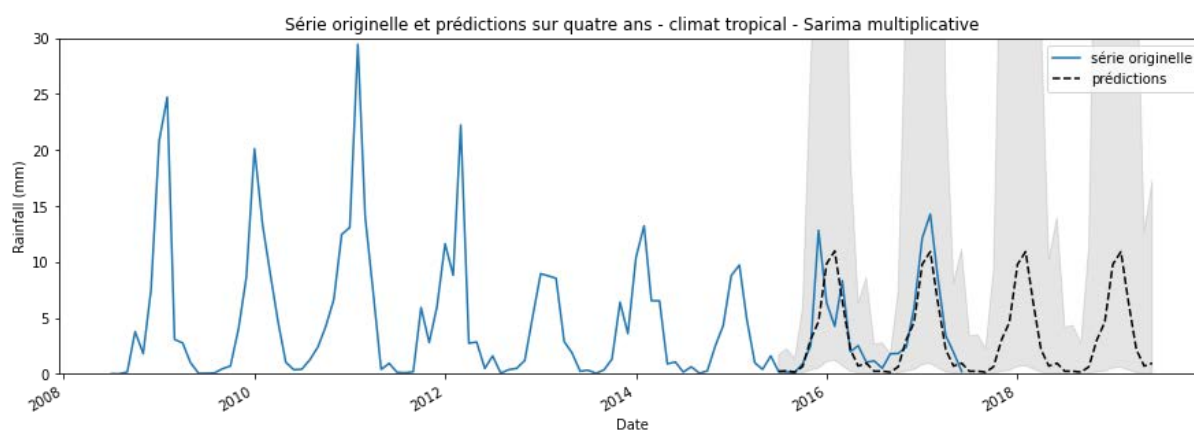
La moyenne mobile, calculée sur 12 mois, évolue peu, mais les séries ne sont pas complètement stationnaires.

Le climat tropical présente notamment une diminution des pics de précipitations après 2012.

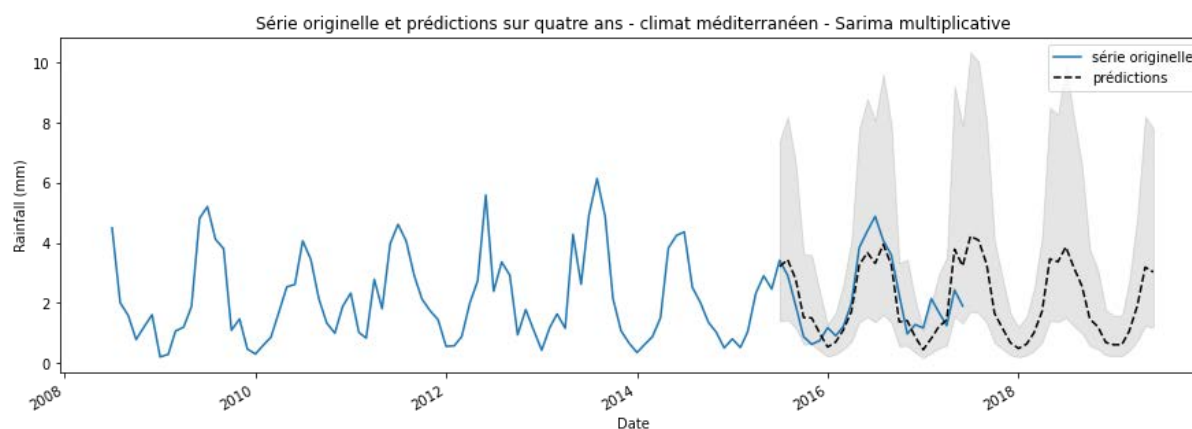
### **Évaluation des modèles prédictifs des séries temporelles :**

Climat	Métrique	SARIMA (multiplicative)	Prophet
Tropical (Aw+Am)	WMAPE	43,8 %	60,6 %
	Corrélation (Pearson)	77,5 %	73,0 %
Méditerranéen (Csa+Csb)	WMAPE	30,0 %	36,0 %
	Corrélation (Pearson)	81,0 %	75,7 %

### **Climat tropical :**



### **Climat méditerranéen :**



### **Conclusion :**

Les performances sont meilleures sur ces deux climats que sur les villes prises indépendamment, avec des erreurs plus faibles et un coefficient de corrélation dépassant les 75 %.

On remarque aussi de performances légèrement meilleures pour le climat méditerranéen que pour le climat tropical, si l'on considère l'erreur WMAPE. Cette différence peut s'interpréter par une meilleure stationnarité de la série méditerranéenne, visible en observant la courbe de la moyenne mobile.

## VII. Deep Learning – Notebook NB7

### 7.1 Introduction

L'objectif de cette section est de tester des modèles de Deep Learning pour prédire *RainTomorrow* et de comparer les performances obtenues aux modèles de Machine Learning classique présentés dans la section 3.

Pour rappel, les scores obtenus par les deux meilleurs algorithmes sur nos données de test (seuil  $\text{recall}=\text{precision}$ ) :

Modèle	Précision	Rappel	Score F1
XGBoost	0.675	0.675	0.675
Light GBM	0.668	0.669	0.669

### 7.2 Modèles denses classiques

Plusieurs modèles ont été construits en faisant varier les caractéristiques suivantes :

- Augmentation du nombre de couches : de 4 à 5 couches de neurones
- Augmentation du nombre de neurones des couches
- Changement de la fonction d'activation : *tanh*, *ReLU*
- Changement de l'initialisateur : *normal*, *Xavier*, *HeNormal*
- Diminution de la taille du batch : 32, 16

L'ensemble des résultats obtenus par les premiers modèles sont assez similaires, les performances ne sont pas significativement différentes, en particulier si l'on considère les variations d'un entraînement à l'autre. Le meilleur modèle donne les résultats suivants :

#### Performance sur train

Classe	Précision	Rappel	Score F1	Accuracy
0	0.89	0.95	0.92	0.87
1	0.77	0.60	0.67	

#### Performance sur test

Classe	Précision	Rappel	Score F1	Accuracy
0	0.88	0.94	0.91	0.86
1	0.72	0.55	0.63	

Par la suite, suivant la même logique que pour les modèles de Machine Learning classique, le meilleur modèle a été testé après opérations de sur-échantillonnage ou sous-échantillonnage.

Les résultats sont présentés dans les tableaux ci-dessous :

#### Après Sous-échantillonnage :

##### Performance sur train

Classe	Précision	Rappel	Score F1	Accuracy
0	0.78	0.89	0.83	0.82
1	0.87	0.76	0.81	

##### Performance sur test

Classe	Précision	Rappel	Score F1	Accuracy
0	0.91	0.86	0.89	0.83
1	0.59	0.71	0.65	

#### Après Sur-échantillonnage :

### Performance sur train

Classe	Précision	Rappel	Score F1	Accuracy
0	0.84	0.88	0.86	0.83
1	0.87	0.83	0.85	

### Performance sur test

Classe	Précision	Rappel	Score F1	Accuracy
0	0.92	0.86	0.89	0.83
1	0.59	0.72	0.65	

### Conclusion :

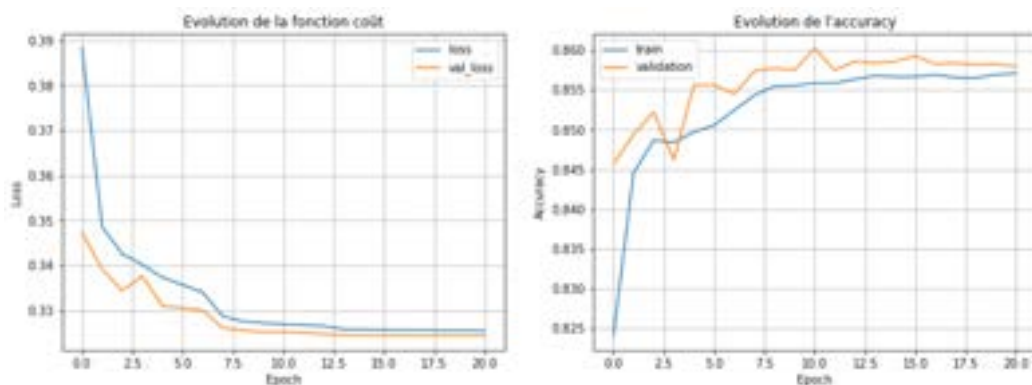
Sur un jeu de données rééchantillonné, les modèles Dense obtiennent des performances similaires à celles des meilleurs modèles de Machine Learning classique, sans toutefois les dépasser.

L'ajout de couches et de neurones supplémentaires améliorent sensiblement le score f1, mais les écarts de performance sont cependant aléatoires (légères variations d'un entraînement à l'autre). Les différentes fonctions d'activation et initialiseurs couramment utilisés donnent des performances similaires.

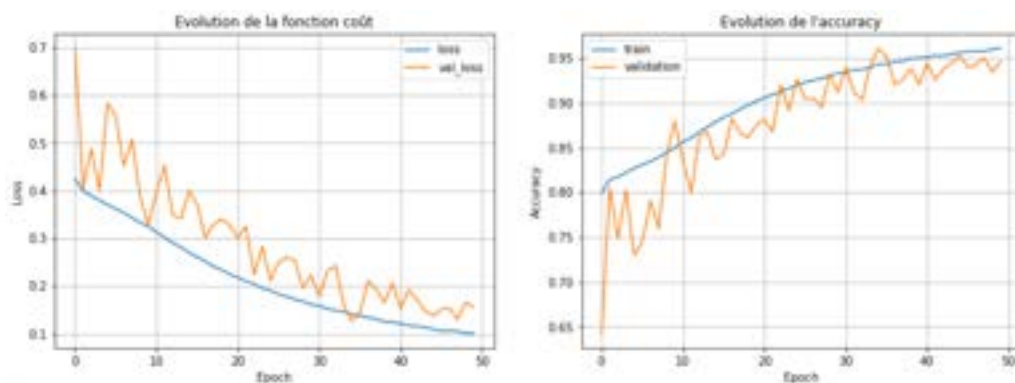
Le rééchantillonnage des classes n'améliore pas les performances des modèles qui sont similaires. On observe les mêmes différences entre précision et rappel que pour les modèles de Machine Learning classique. L'évolution des métriques au cours des époques est plus erratique après rééchantillonnage, comme en témoignent les graphiques ci-dessous, y compris en augmentant fortement le nombre d'époques (voir Notebook NB7 pour plus de détails).

### Évolution des métriques au cours de l'entraînement

#### *Jeu de données déséquilibré*



#### *Jeu de données équilibré*



### 7.3 Modèles FASTAI

Pour compléter l'étude ci-dessus, un modèle de Deep Learning utilisant la bibliothèque FastAI a été développé en s'inspirant de la littérature disponible sur le web :

<https://confusedcoders.com/data-science/deep-learning/how-to-apply-deep-learning-on-tabular-data-with-fastai>

Les performances obtenues par le modèle sont reportées dans les tableaux ci-dessous :

#### *Performance sur train*

Classe	Précision	Rappel	Score F1	Accuracy
0	0.87	0.95	0.91	0.85
1	0.73	0.51	0.60	

#### *Performance sur test*

Classe	Précision	Rappel	Score F1	Accuracy
0	0.87	0.94	0.91	0.85
1	0.72	0.51	0.59	

Les performances obtenues avec sur ou sous-échantillonnage conduisent aux mêmes conclusions que pour les réseaux de neurones denses classiques.

### 7.4 Conclusion Deep Learning

Les modèles de Deep Learning développés n'ont pas démontré de meilleurs résultats que les modèles de Machine Learning classique étudiés en début de projet.

Par ailleurs, au-delà des performances peu convaincantes sur notre jeu de données, le manque d'interprétabilité des modèles de Deep Learning par rapport au Machine Learning classique ne pousse pas à les développer davantage lors de ce projet.



## VIII. Conclusion générale

Notre projet RainsBerryPy nous a permis de mettre en application les différents apprentissages de la formation de Datascientist commencée en octobre 2021 : preprocessing, manipulation de dataframe, DataViz, Machine Learning, interprétabilité, clustering, séries temporelles et même Deep Learning.

Un projet complet qui nous a permis de mettre en avant notre esprit d'initiative en recherchant :

1. Des éléments nécessaires à notre modélisation : climat de Köppen, circularisation de la variable mois, ...
2. De nouvelles bibliothèques/algorithmes : KNN imputer, Light gbm, Shapash, tslearn, Prophet, FastAI...

Aussi, la collaboration au sein de notre groupe s'est très bien déroulée et a démontré que le travail en distanciel (devenue une norme depuis la crise sanitaire) n'entache en rien sa performance.

Nous tenions aussi à remercier notre mentor Laurène qui a su questionner notre travail et en assurer sa cohérence à chaque itération.

Pour finir, les possibles évolutions que nous pourrions apporter à notre travail seraient les suivantes :

- Ajout de variables de géolocalisation de la pluie : pour chaque ligne indiquer la distance de la ville la plus proche où il pleut,
- Injection de notre résultat de clustering dans notre modèle même si les résultats pourraient être sensiblement similaires à la classification de Koppen,
- Ajout d'images satellites au jeu de données avec utilisation d'algorithmes de deep learning CNN voir RNN,
- Utilisation d'algorithme de deep learning RNN sur les séries temporelles.

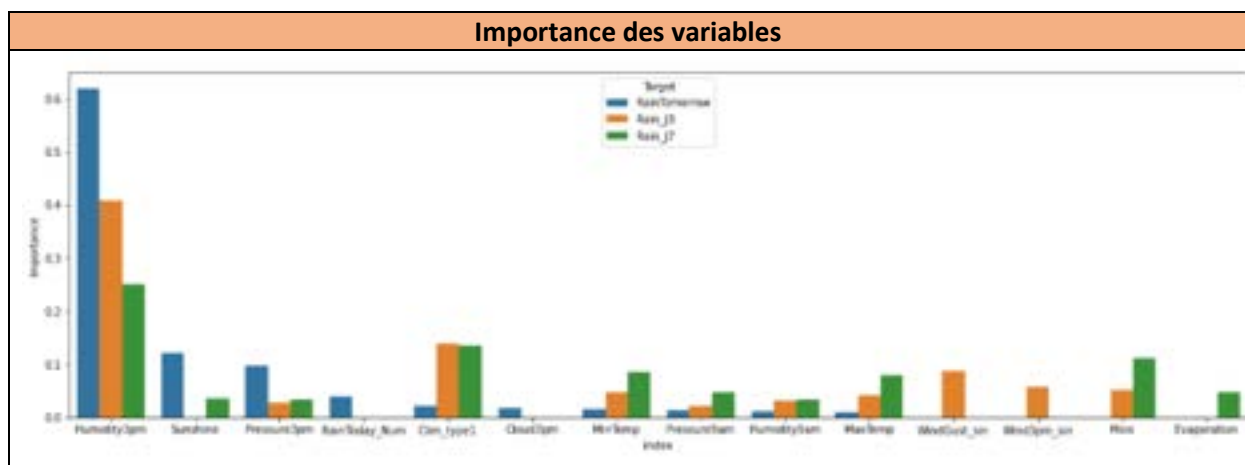
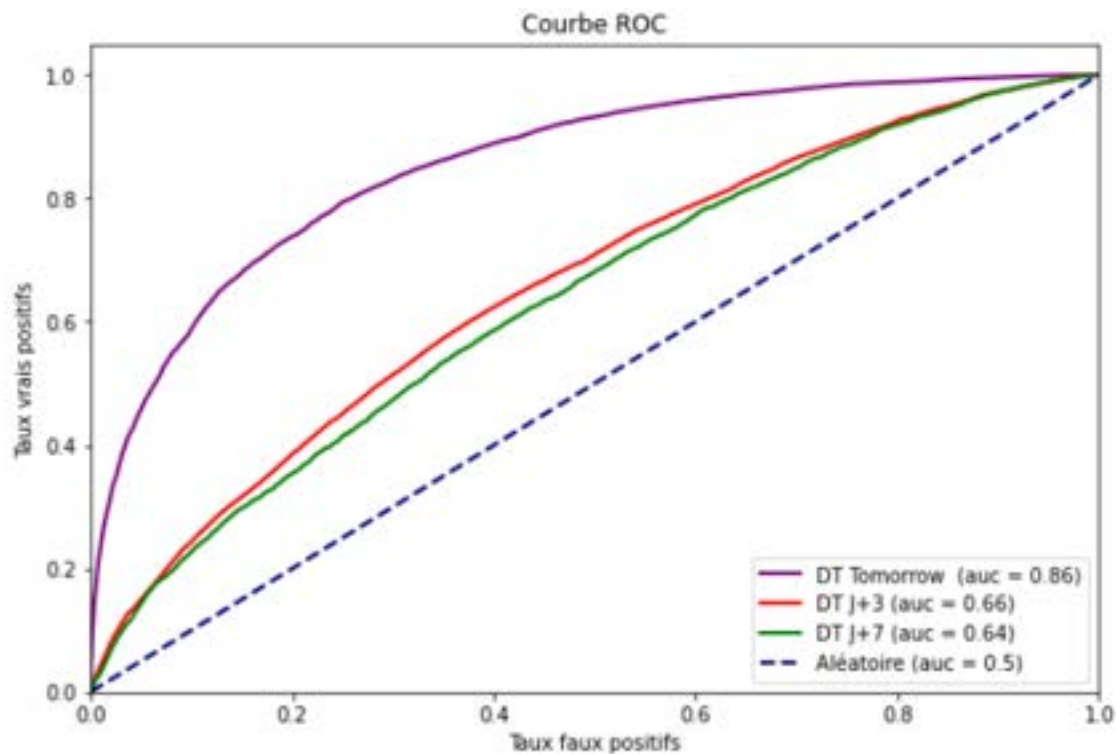
## IX. Annexes - autres modélisations de RainTomorrow

### 9.1 Modélisation de la pluie à J+3 et J+7 – Arbre de décision

Dans la section VI, nous avons modélisé si le jour suivant, il a plu (au moins 1 mm). Nous pourrions complexifier la tâche en tentant de modéliser s'il a plu à J+3, J+7 ou autres.

Nous avons initié l'exercice, à partir d'arbres de décisions qui ont le mérite d'être faciles à décrire et rapides à tourner. Ils nous permettent de se faire une opinion assez rapidement sur le sujet.

Les courbes de ROC ci-dessous illustrent la plus grande difficulté à prédire la pluie à J+3 et J+7. Il y a une nette différence par rapport à la prédiction de la pluie du lendemain.



Même si elle garde un impact important, on voit que Humidity3pm perd de son pouvoir explicatif alors que le climat, le mois ou la température ont un effet plus important à J+3, J+7.

## 9.2 Modélisation de la pluie à J+1 et J+7 – LightGBM et Shapash

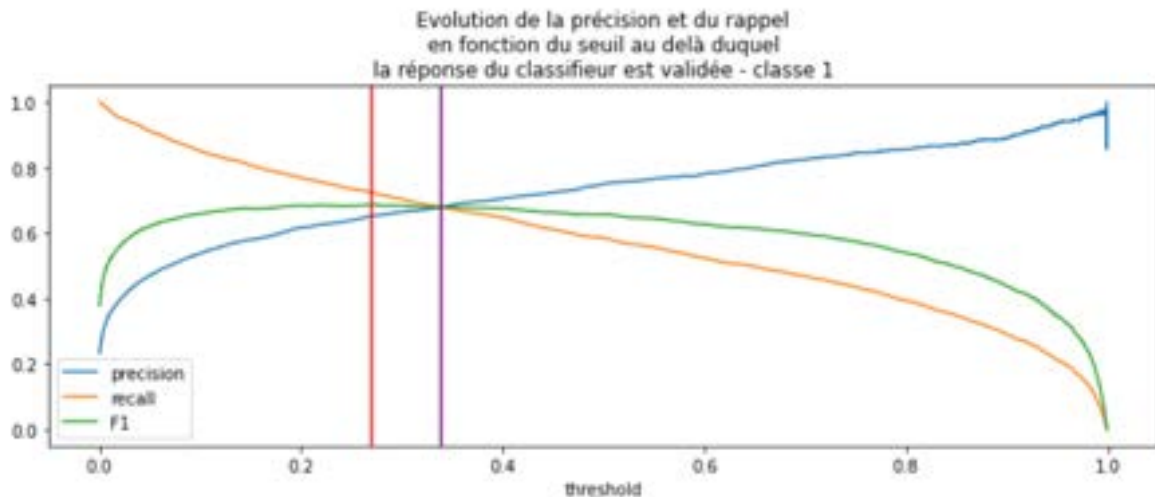
Cette modélisation a été faite sur la base non rééchantillonnée et nous avons ensuite modifié les seuils de prédiction pour maximiser le F1 ou rappel = précision sur la classe minoritaire (classe 1 à modéliser).

- **Pluie à J+1 :**

- Courbe rouge : seuil pour maximiser le F1 sur la classe à modéliser (1)
- Courbe violette : seuil pour que précision = rappel sur la classe à modéliser (1)

⇒ le seuil où rappel = précision (en violet) ne fait pas perdre beaucoup sur le F1 et peut être un arbitrage intéressant.

LIGHT GBM J+1



Score sur test pour Maximiser le F1 : 0.27056423042289074

	precision	recall	f1-score	support
0	0.92	0.89	0.90	8813
1	0.65	0.72	0.69	2500
accuracy			0.85	11313
macro avg	0.79	0.81	0.80	11313
weighted avg	0.86	0.85	0.86	11313

Score sur test pour Rappel = Precision : 0.3388693602709504

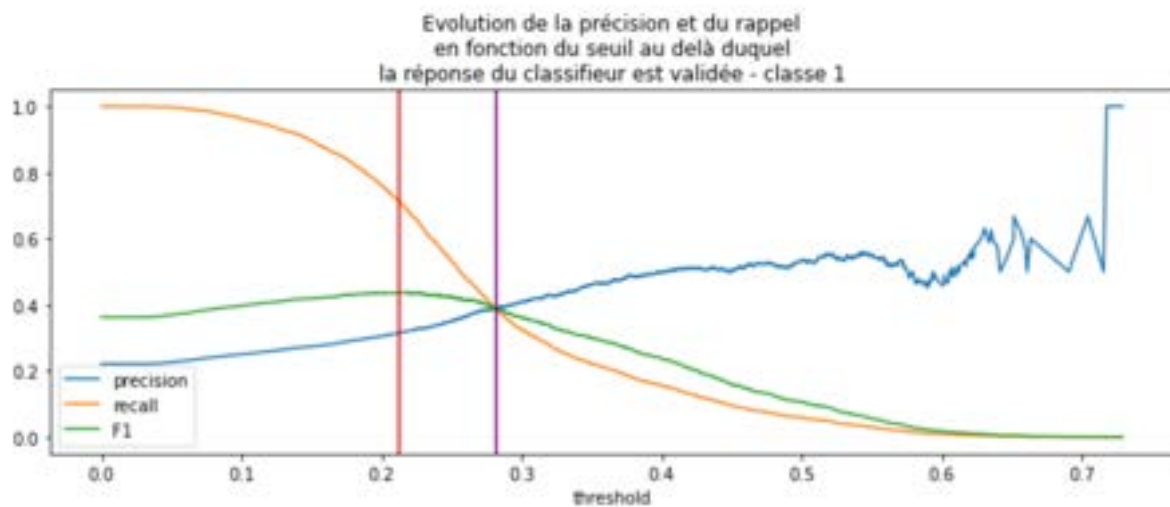
	precision	recall	f1-score	support
0	0.91	0.91	0.91	8813
1	0.68	0.68	0.68	2500
accuracy			0.86	11313
macro avg	0.79	0.79	0.79	11313
weighted avg	0.86	0.86	0.86	11313

- **Pluie à J+7 :**

- Courbe rouge : seuil pour maximiser le F1 sur la classe à modéliser (1)
- Courbe violette : seuil pour que précision = rappel sur la classe à modéliser (1)

⇒ Sur J+7, le graphique nous montre bien que le choix du seuil entre F1 et recall=précision va conduire à des résultats sur la matrice de confusion bien différents.

LIGHT GBM J+7



Score sur test pour Maximiser le F1 : 0.2119489616920855

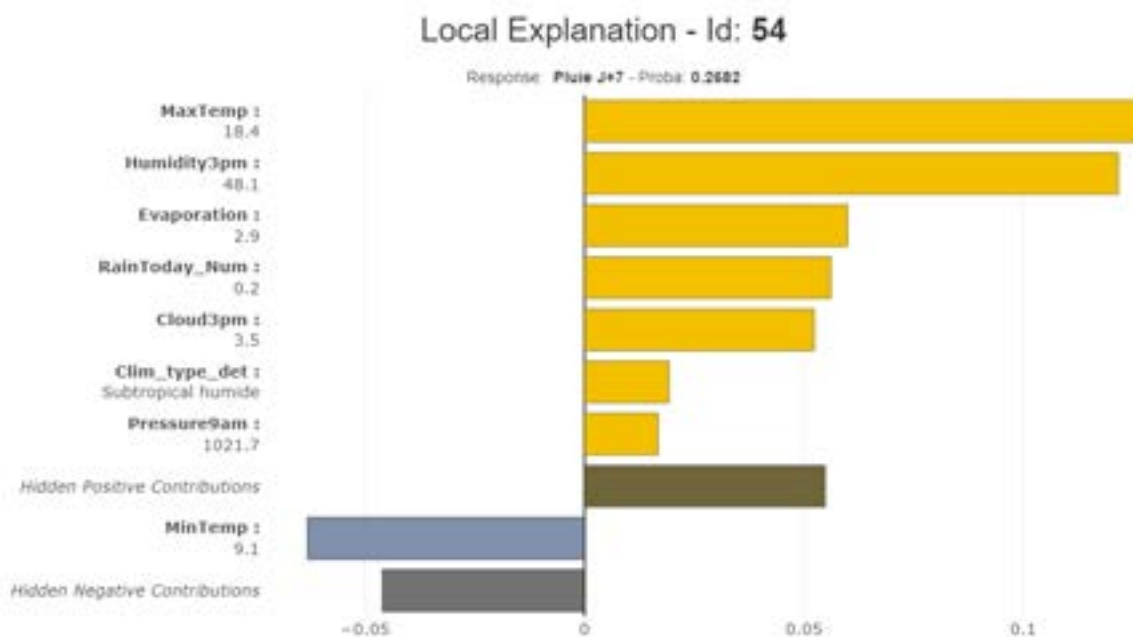
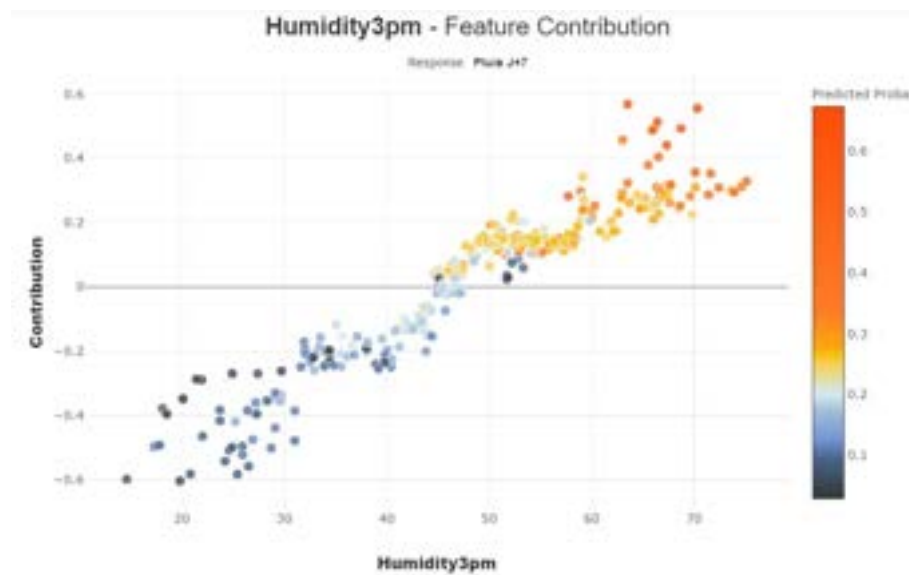
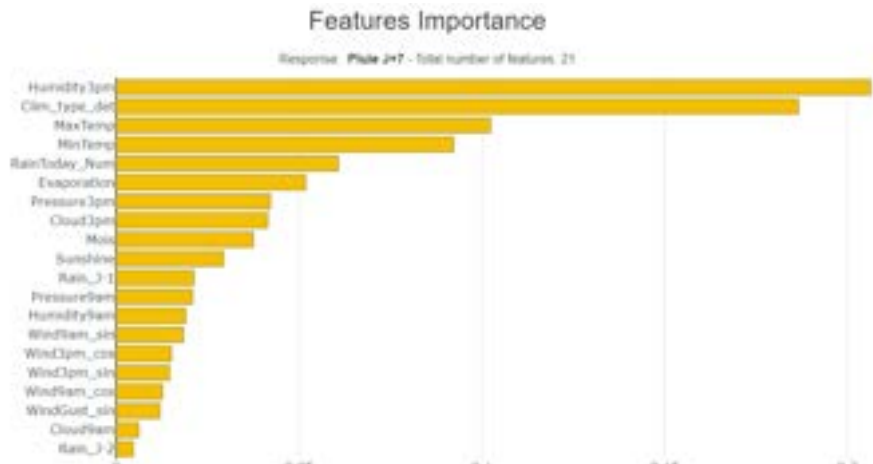
	precision	recall	f1-score	support
0	0.89	0.57	0.69	8813
1	0.33	0.75	0.46	2500
accuracy			0.61	11313
macro avg	0.61	0.66	0.57	11313
weighted avg	0.76	0.61	0.64	11313

Score sur test pour Rappel = Precision : 0.28150298538154445

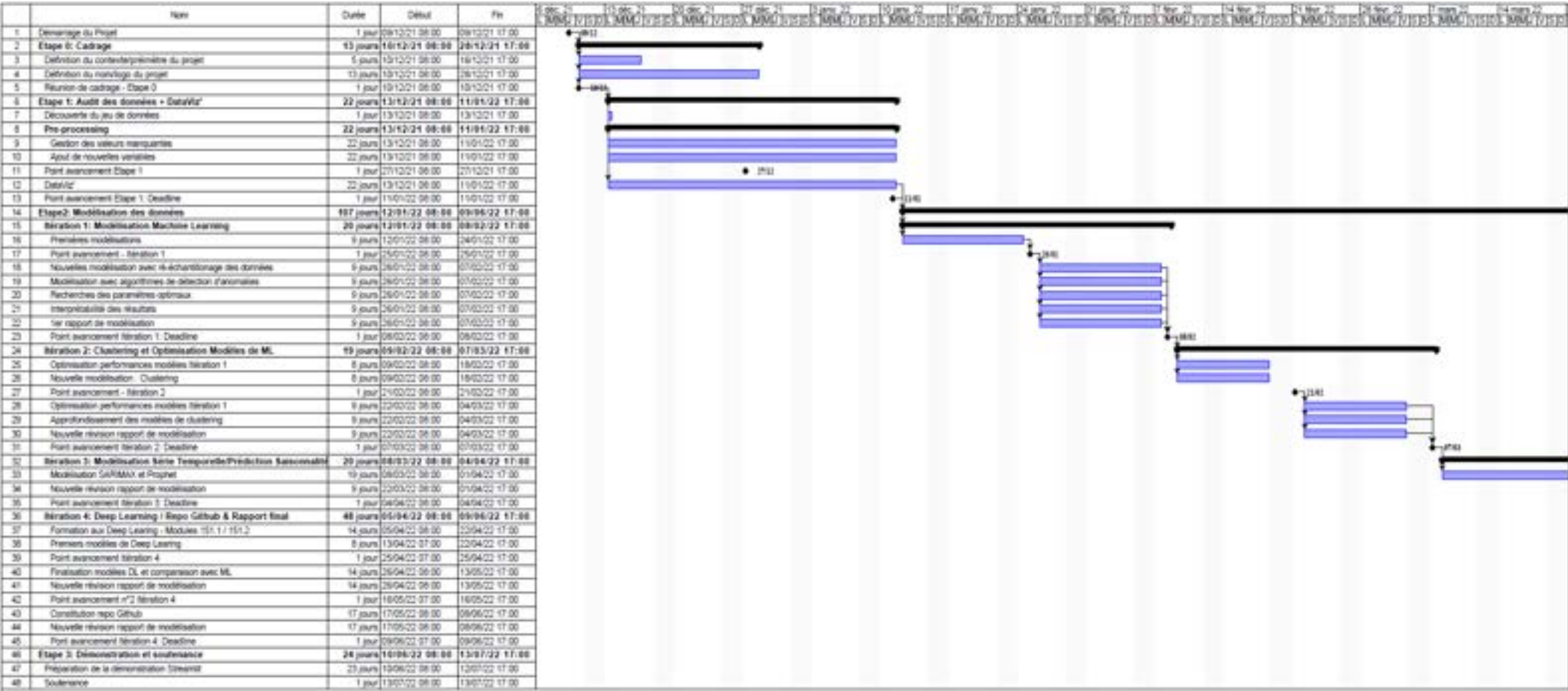
	precision	recall	f1-score	support
0	0.83	0.83	0.83	8813
1	0.39	0.39	0.39	2500
accuracy			0.73	11313
macro avg	0.61	0.61	0.61	11313
weighted avg	0.73	0.73	0.73	11313

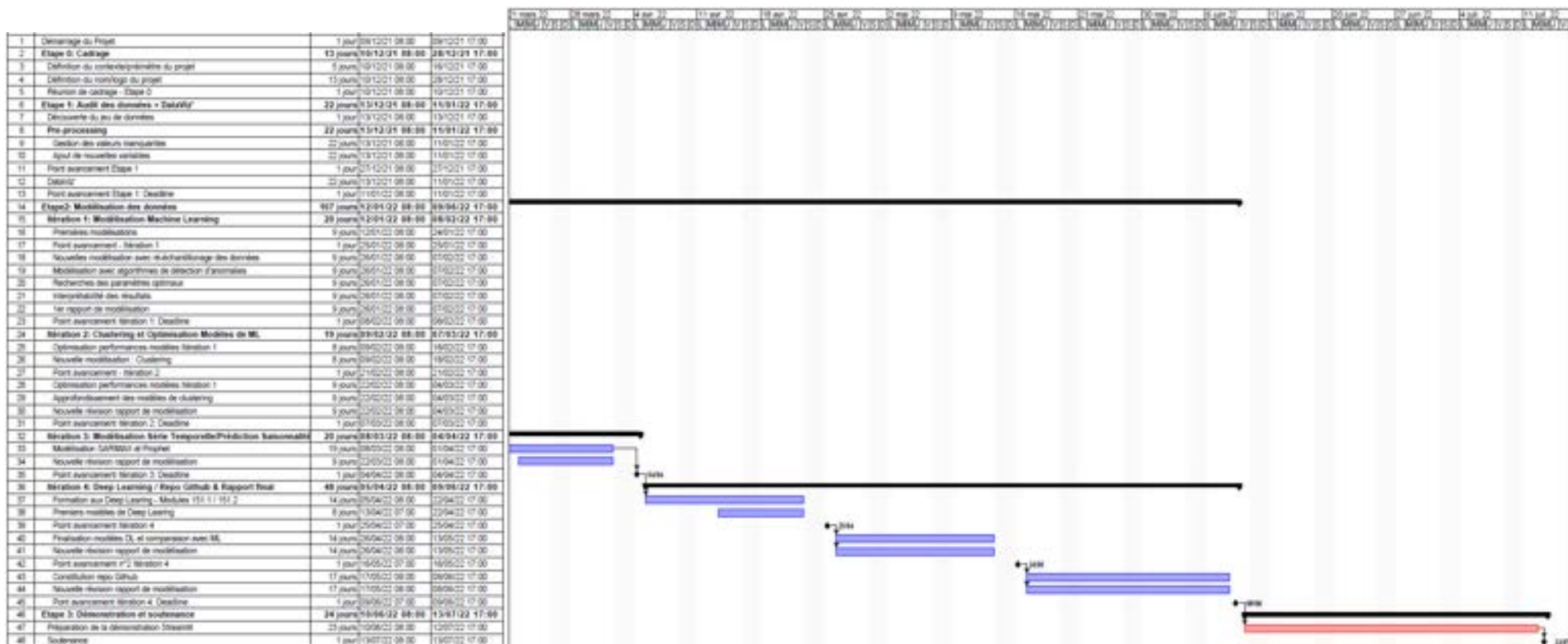
- **Interprétabilité via shapash :**

Pour la modélisation à J+1 ou J+7, Shapash a été utilisé pour la partie interprétabilité et cet outil apporte une restitution très intéressante.



X. Répartition de la charge – Diagramme de Gantt du projet RainsBerryPy







## XI. Bibliographie & Sitographie

### SITOGRAPHIE

Source du dataset sur Kaggle :

<https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package>

#### Dataviz et ajout des variables climatiques

- Bureau of Meteorology (gouvernement australien) :  
<http://www.bom.gov.au/>
- Autre sites pour la définition des climats :  
<https://fr.climate-data.org>  
<https://plantmaps.com/>
- DataViz cartographique : Bibliotheque geopandas (<https://geopandas.org/en/stable/>)
- 

#### Preprocessing

- Circularisation de la variable Mois :  
<https://datascientest.com/numeriser-des-variables>
- KNN-Imputer :  
<https://medium.com/@kyawsawhtoon/a-guide-to-knn-imputation-95e2dc496e>

#### Algorithmes de Machine Learning classique et optimisation

- Light GBM : algorithme dont la réputation est de converger rapidement  
<https://datascience.eu/fr/apprentissage-automatique/quest-ce-que-la-gbm-legere/>
- Randomized GridSearchCV (réduction du temps de calcul pour la recherche d'hyperparamètres)  
<https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10>
- Les métriques de la matrice de confusion : <https://kobia.fr/classification-metrics-f1-score/>
- Shapash : outil d'interprétabilité globale et locale : <https://shapash.readthedocs.io/en/latest/>

#### Clustering :

- Classification de séries temporelles :  
<https://towardsdatascience.com/how-to-apply-k-means-clustering-to-time-series-data-28d04a8f7da3>

#### Séries temporelles

- Intérêt de la métrique WMAPE :  
<https://resdntalien.github.io/blog/wmape/>

#### Deep Learning

- Initialiseur et fonction d'activation :  
<https://github.com/christianversloot/machine-learning-articles/blob/main/random-initialization-vanishing-and-exploding-gradients.md>  
<https://github.com/christianversloot/machine-learning-articles/blob/main/he-xavier-initialization-activation-functions-choose-wisely.md>
- Fast AI :  
<https://confusedcoders.com/data-science/deep-learning/how-to-apply-deep-learning-on-tabular-data-with-fastai>

---

### BIBLIOGRAPHIE

Hastie *et al.*, 2009, The Elements of Statistical Learning – Second edition, Springer  
VanderPlas, 2018, Python Data Science Handbook, O'Reilly