

# Rapport de modélisation sur les données météorologiques Australiennes

Formation : Data Scientist (05/2023 – 04/2024)

Encadrant : Francesco MADRISOTTI

Réaliseurs : Sophie BERTHIER

Luciano LANGHI

Quyen THIEU MARCAUD

Le 20/02/2024 – Version finale

## Table des matières

1	Introduction.....	4
1.1	Objectifs.....	4
1.2	Contexte historique et enjeux .....	4
1.2.1	Contexte géographique et climatique .....	4
2	Exploration des données et visualisation .....	5
2.1	Sources de données.....	5
2.1.1	Kaggle.....	5
2.1.2	Bureau of Meteorology.....	6
2.2	Variables du dataset.....	6
2.3	Statistiques descriptives.....	8
2.3.1	Variables catégorielles.....	8
2.3.2	Variables numériques .....	12
2.4	Corrélation.....	15
2.5	Analyse détaillée des variables.....	17
2.5.1	RainTomorrow.....	17
	Figure 10 : Nombre de journées sèches et pluvieuses par <i>Location</i> .....	18
2.5.2	<i>Location</i> , <i>MaxTemp</i> et <i>RainTomorrow</i> .....	18
2.5.3	Analyse temporelle et géographique .....	20
2.6	Valeurs manquantes.....	23
2.6.1	Vue globale.....	23
2.6.2	Répartition géographique.....	27
2.6.3	Répartition temporelle .....	28
3	Pre-processing et feature engineering.....	30
3.1	Nettoyage des données .....	30
3.1.1	Doublons.....	30
3.1.2	Traitement des valeurs extrêmes.....	31
3.1.3	Suppression de variables.....	31
3.1.4	Suppression des observations .....	31
3.1.5	Complétion des données manquantes à l'aide d'autre source de données complémentaire .....	32
3.1.6	Imputation des données manquantes .....	34
3.2	Transformation des données .....	39
3.2.1	Booléens .....	39
3.2.2	Directions du vent.....	39
3.3	Ajout de variables.....	39
3.3.1	Coordonnées des villes .....	39
3.3.2	Amplitude thermique .....	39
3.3.3	Information climatique .....	40
3.3.4	Corrélations des nouvelles variables.....	43

3.3.5	Normalisation et standardisation .....	45
4	Conclusions sur le preprocessing.....	46
5	Introduction - Modélisation .....	46
5.1	Méthodologie.....	46
5.2	Approches.....	47
6	Prédiction de la variable <i>RainTomorrow</i> .....	47
6.1	Rappel sur déséquilibre .....	47
6.2	Métriques.....	48
6.3	Résultats de la classification par approches « classiques » via scikit-learn.....	48
6.3.1	Modèles étudiés .....	48
6.3.2	Optimisation de métriques.....	48
6.3.3	Impact du feature enginerring.....	56
6.3.4	Seuil de probabilité.....	57
6.3.5	Interprétabilité des modèles.....	61
6.4	Deep Learning avec Keras et TensorFlow.....	67
6.4.1	DNN.....	67
6.4.2	RNN.....	77
7	Prédiction de la pluie à un horizon de temps .....	80
7.1	Objectif et méthodologie .....	80
7.2	Limite théorique .....	80
7.3	Comportement détaillé .....	82
7.4	Analyse par zone climatique.....	82
7.5	Prédictions .....	85
7.6	Interprétabilité .....	86
7.7	Conclusions .....	88
8	Prédiction de la variable <i>MaxTemp</i> .....	88
8.1	Présentation .....	88
8.2	Résultats de la régression par approches « classiques » via scikit-learn .....	88
8.3	Séries Temporelles par SARIMAX .....	89
8.4	Deep Learning .....	91
8.4.1	RNN.....	91
9	Autres variables cibles .....	96
10	Conclusion .....	96
10.1	Constats .....	96
10.2	Limites et perspectives .....	96

# 1 Introduction

## 1.1 Objectifs

Ce projet consiste à prédire des variables météorologiques à partir d'un jeu de données contenant dix ans de relevés sur de nombreuses stations météo australienne.

Dans un premier temps, nous tenterons de prédire s'il pleuvra le lendemain (variable *RainTomorrow*). Nous étendrons ensuite nos prévisions à d'autres variables, telle la température et nous tenterons d'effectuer des prévisions portant sur plusieurs jours.

## 1.2 Contexte historique et enjeux

La prédiction des conditions météorologique est un domaine particulièrement ancien, qui a été un enjeu pour de nombreuses sociétés au fil des siècles, et ce dès l'invention de l'agriculture à la préhistoire. Initialement prédictive au travers de pratiques divinatoires, les méthodes de prédictions se sont enrichies au fil des siècles : l'importance des nuages a été établie par les babyloniens il y a 8.500 ans, celles des relevés météorologiques par la Chine il y a 5.000 ans, et les innombrables dictons populaires en France témoignent d'une part de la place que le domaine revêt auprès de chacun, d'autre part de la diversité des liens constatés (« Noël au balcon, Pâques au tison », par exemple, indiquant qu'une température élevée fin décembre en impliquerait une faible trois mois plus tard).

Aujourd'hui, il s'agit d'un enjeu économique crucial dans de nombreux secteurs, qu'il s'agisse bien entendu toujours de l'agriculture, mais aussi de l'aéronautique, du tourisme, du BTP, des assurances, etc.

Les prévisions météorologiques ont l'avantage d'être tout à la fois un domaine connu par le grand public depuis de nombreuses années et de mobiliser des techniques poussées pour créer des modèles de qualité.

Nous garderons à l'esprit que les modèles les plus puissants actuels ne permettent que difficilement de prédire de façon fiable au-delà de 7 jours.

### 1.2.1 Contexte géographique et climatique

L'Australie est une immense île située entre l'Océan Pacifique et Indien. Elle se trouve dans l'hémisphère sud, ce qui implique que les saisons sont décalées de 6 mois par rapport à celles de l'hémisphère nord.

Au centre du pays se trouvent d'immenses déserts, occupant 18% du territoire. Leur climat est aride.

La Cordillère australienne (Great Dividing Range) est une immense chaîne de montagne longeant toute la côte est. Sa partie méridionale se nomme les Alpes australiennes. Elle comprend son point culminant (2 228m) et est enneigée.

De vastes forêts tropicales longent également la côte est, en particulier sur la partie nord-est, entre l'océan Pacifique et la Cordillère australienne. Il s'agit de zones humides.

Le reste du pays est constitué de plaines de basse altitude, avec une végétation de savane tropicale au nord et de forêt de type méditerranéenne au sud.

De nombreuses îles entourent l'île principale, telles l'île de Tasmanie, au sud-est, et l'île de Norfolk, à plus de 1400km à l'est.

L'Australie possède des climats variés, tropical au nord avec des précipitations particulièrement importantes du fait de la mousson, jusqu'à un climat désertique au centre avec des températures élevées et peu de précipitations, en passant par un climat tempéré au sud-est. Plusieurs cartes de climats existent, avec des répartitions qui divergent sensiblement et des stratifications plus ou moins riches. En voici une :

# Australie

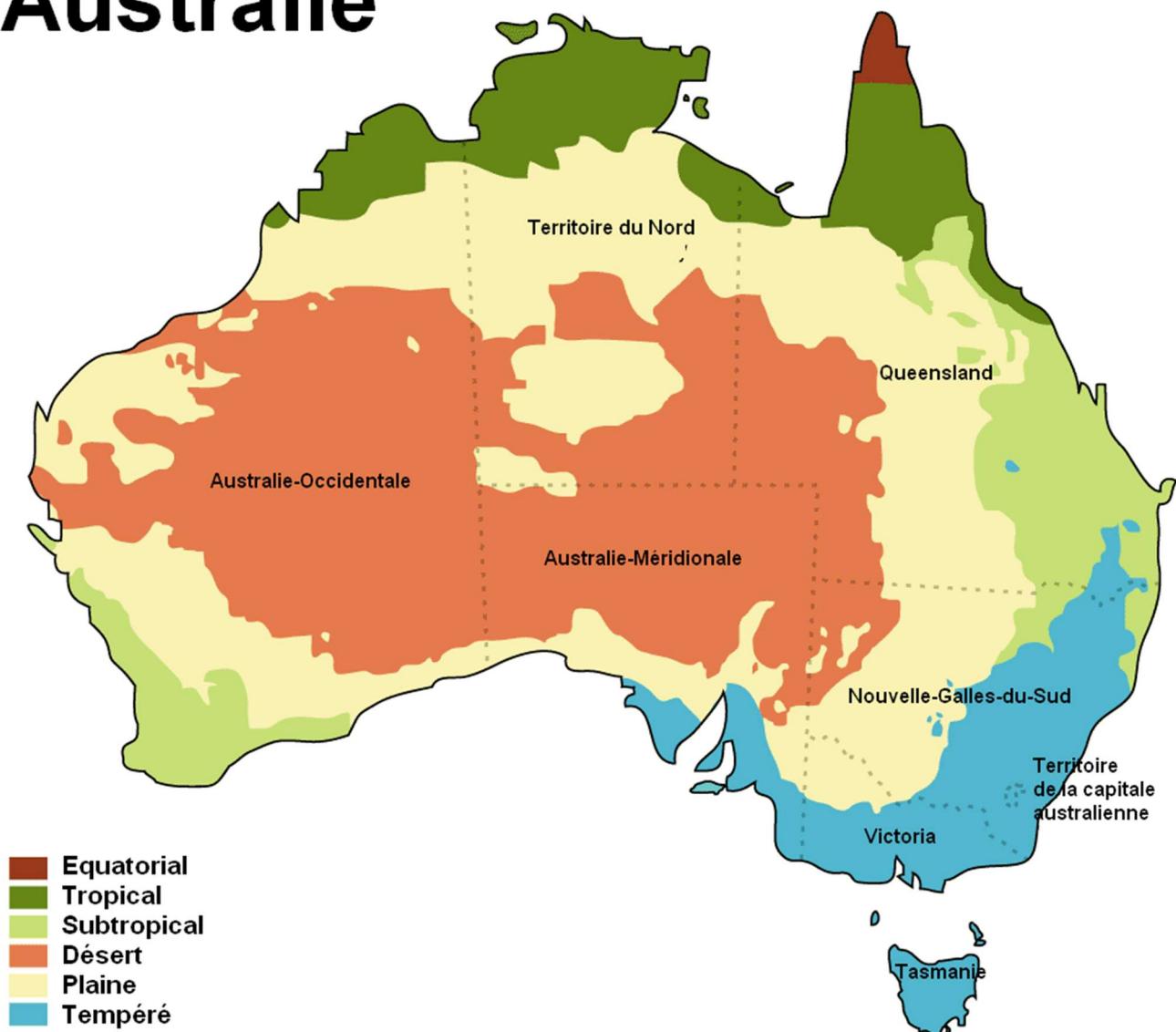


Figure 1 : Carte des climats australiens.

Src : Wikipédia (source : Wikipédia : [https://fr.wikipedia.org/wiki/Climat\\_de\\_l'Australie](https://fr.wikipedia.org/wiki/Climat_de_l'Australie) )

## 2 Exploration des données et visualisation

### 2.1 Sources de données

#### 2.1.1 Kaggle

Le data set est celui disponible sur Kaggle pour le projet « Rain in Australia » (<https://www.kaggle.com/datasets/jspphyg/weather-dataset-rattle-package> ).

Ce dataset contient presque 10 ans d'observations météorologiques quotidiennes provenant de plusieurs stations météorologiques australiennes. Ces observations sont des observations météorologiques quotidiennes réalisées à 9h et 15h sur une période de 10 ans, du 01/11/2007 au 25/06/2017.

### 2.1.2 Bureau of Meteorology

Le site du Bureau of Meteorology du gouvernement australien (<http://www.bom.gov.au/climate/data/>) propose de nombreuses données consultables en lignes. Malheureusement, le site ne permet pas d'obtenir directement un jeu comportant toutes les variables du dataset Kaggle. Il ne permet en effet d'obtenir les mêmes features que le dataset Kaggle que sur les 14 derniers mois. Pour la période couverte par le dataset Kaggle, il n'est possible que de télécharger quelques variables (précipitations et températures). Dans tous les cas, ce téléchargement doit s'effectuer pour chaque station météorologique, laquelle n'est pas directement indiquée dans le dataset original.

Au final, utiliser le site du Bureau of Meteorology pour enrichir notre dataset ou bien renseigner des données manquantes ne pourra malheureusement pas se faire de façon simple, ni même par une approche de Webscrapping. Elle ne pourra se faire que très ponctuellement en ciblant certaines variables pour des villes particulières.

## 2.2 Variables du dataset

Le dataset contient les 23 variables présentées dans le Tableau 1.

No	Nom de colonne	Unité	Explication
1	Date	Timestamp	Date d'observation
2	Location	Chaîne de caractères	Nom du lieu de la station météo
3	MinTemp	Degrés Celsius	Température minimum en 24 heures jusqu'à 9am
4	MaxTemp	Degrés Celsius	Température maximum en 24 heures jusqu'à 9am
5	Rainfall	Millimètres	Précipitation en 24 heures jusqu'à 9am
6	Evaporation	Millimètres	Évaporation en 24 heures jusqu'à 9am
7	Sunshine	Heure	Soleil radieux en 24 heures jusqu'à minuit
8	WindGustDir	16 points cardinaux	Direction de la rafale de vent la plus forte en 24 heures jusqu'à minuit
9	WindGustSpeed	Kilomètres par heure	Vitesse de la rafale de vent la plus forte en 24 heures jusqu'à minuit
10	WindDir9am	16 points cardinaux	Direction de vent à 9am
11	WindDir3pm	16 points cardinaux	Direction de vent à 3pm
12	WindSpeed9am	Kilomètres par heure	Vitesse de vent à 9am
13	WindSpeed3pm	Kilomètres par heure	Vitesse de vent à 3pm
14	Humidity9am	Pourcentage	Humidité relative à 9am
15	Humidity3pm	Pourcentage	Humidité relative à 3pm
16	Pressure9am	Hectopascals	Pression atmosphérique réduite au niveau moyen de la mer à 9am
17	Pressure3pm	Hectopascals	Pression atmosphérique réduite au niveau moyen de la mer à 3pm
18	Cloud9am	Huitièmes	Fraction de ciel obscurcie par les nuages à 9am
19	Cloud3pm	Huitièmes	Fraction de ciel obscurcie par les nuages à 3pm
20	Temp9am	Degrés Celsius	Température à 9am
21	Temp3pm	Degrés Celsius	Température à 3pm
22	RainToday	Binaire (Yes, No)	La journée en cours a-t-elle reçu des précipitations supérieures à 1 mm en 24 heures jusqu'à 9h ?
23	RainTomorrow	Binaire (Yes, No)	Le lendemain a-t-il reçu des précipitations dépassant 1 mm en 24 heures jusqu'à 9am ?

**Tableau 1 : Les variables de l'ensemble de données**

Le Tableau 2 représente un Overview (généré par la librairie *ydata\_profiling*) du dataframe de 22 colonnes. L'ensemble de données contient 145 460 d'observations dont il y a 21 observations qui sont redondantes.

# Overview

Overview	Alerts <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">26</span>	Reproduction
<b>Dataset statistics</b>		
<hr/>		
Number of variables		22
Number of observations		145460
Missing cells		343248
Missing cells (%)		10.7%
Duplicate rows		21
Duplicate rows (%)		< 0.1%
Total size in memory		25.5 MiB
Average record size in memory		184.0 B
<hr/>		
<b>Variable types</b>		
<hr/>		
Categorical		4
Numeric		16
Boolean		2

**Tableau 2 : Overview du dataset**

Les 22 variables se divisent en 3 types dont

- 4 variables catégorielles : *Location*, *WindGustDir*, *WindDir9am* et *WindDir3pm*
- 2 variables booléennes : *RainToday*, *RainTomorrow*
- 16 variables numériques.

A noter que deux variables sont directement déduites d'autres informations :

- *RainToday* est indiquée comme True si *Rainfall >1*
- *RainTomorrow* d'une date donnée est égale à *RainToday* de la date du lendemain, pour *Location* donnée.

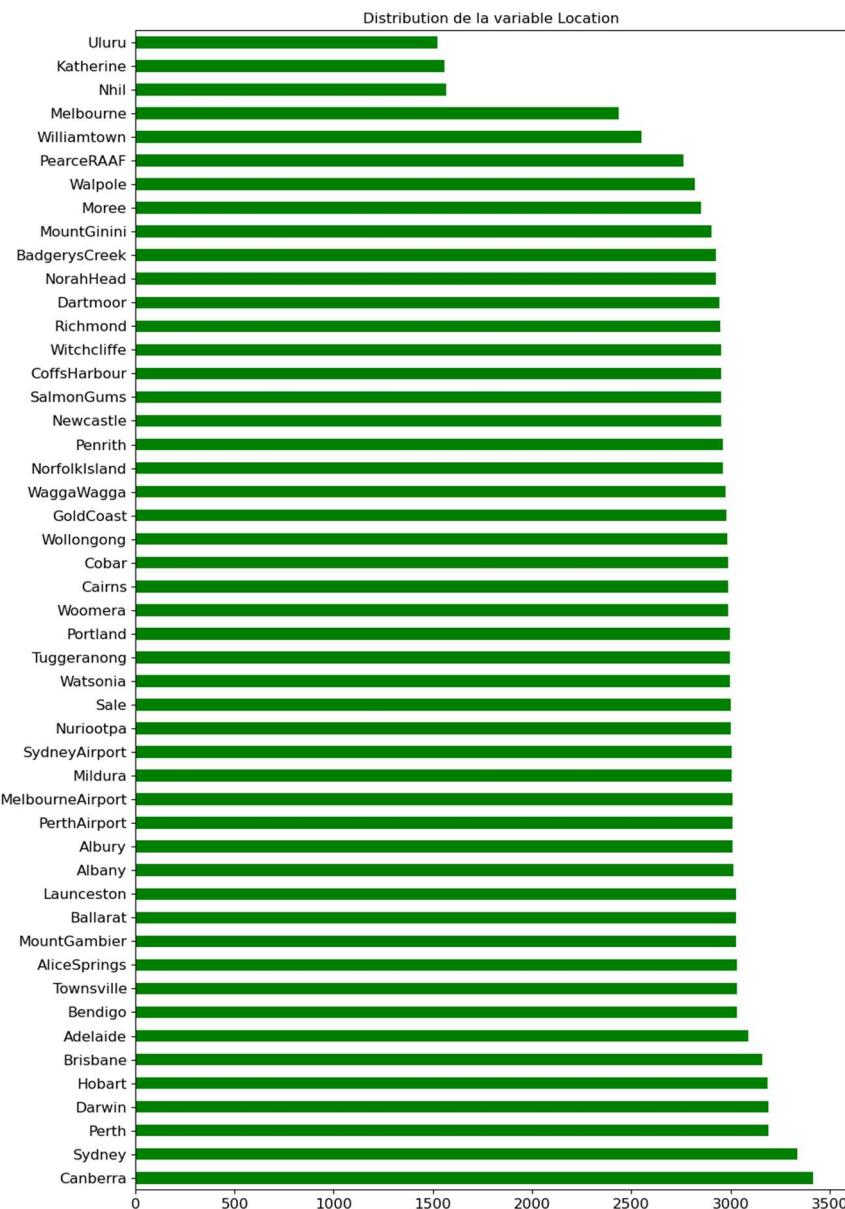
Nous avons vérifié et confirmé ces deux affirmations.

## 2.3 Statistiques descriptives

Dans cette section, nous présentons une vue globale sur les distributions des variables du dataframe.

### 2.3.1 Variables catégorielles

La Figure 2 représente la distribution de la variable *Location* qui contient 49 stations météorologiques. Les trois stations météorologiques *Uluru*, *Katherine* et *Nhil* contiennent environ deux fois moins d'observations que les autres. Les deux stations *Sydney* et *Canberra* contiennent plus d'observations que les autres. Le nombre d'observations des autres stations reste à peu près homogène.



**Figure 2 : Distribution de Location**

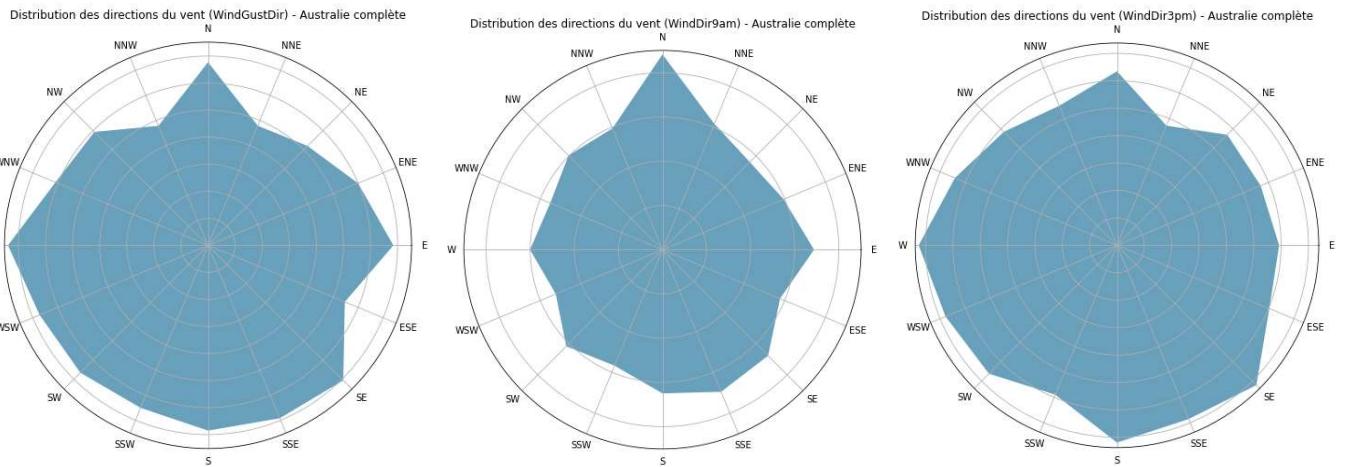
Remarquons également que le nombre maximal de journées est de 3418 pour Canberra. La plupart des villes ont un nombre de 3 000 journées environ. Or, nous avons vu que la plage de dates couvre la période du 1/11/07 au 25/06/17, soit 3525 journées. Il manque donc l'équivalent d'environ 1 an et demi de mesures pour la plupart des villes, aucune n'est exhaustive sur la plage de dates. Ce point est partiellement important à prendre en compte pour l'analyse par séries temporelles.

Plusieurs Location possèdent le suffixe « Airport », semblant indiquer que certaines stations météorologiques sont assez proches (« Perth » / « PerthAirport », « Melbourne » / « MelbourneAirport » et « Sydney » / « SydneyAirport »).

La Figure 3 représentent la distribution des trois autres variables catégorielles, *WinDir9am*, *WindDir3pm* et *WindGustDir* sont trois variables catégorielles indiquant la direction du vent, respectivement à 9h00, 15h00, ainsi que pour la rafale de vent la plus forte. Les valeurs possibles sont les 16 directions cardinales (N, NNE, NE, ...).

- *WindGustDir* : la direction des rafales de vent est plus fréquemment à l'ouest et moins fréquemment au nord-nord-est
- *WindDir9am* : à 9 heures du matin, la direction du vent est significativement plus fréquemment au nord, et moins fréquemment vers l'ouest-sud-ouest.
- *WindDir3pm* : à 15h, le vent souffle plus fréquemment vers le sud-est et moins fréquemment vers le nord-nord-est (à l'instar de *WindGustDir*).

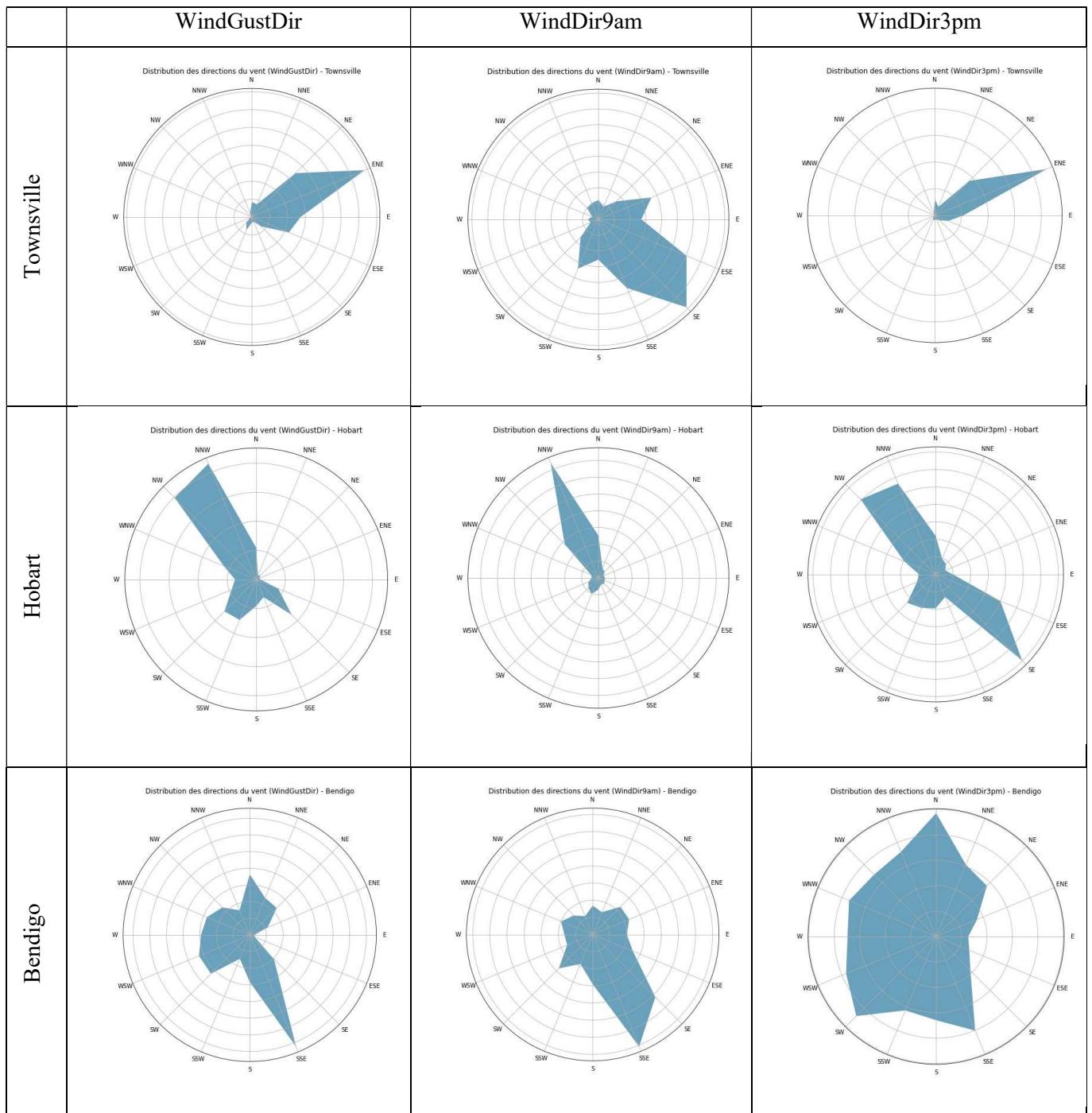
Toutefois, la distribution des directions du vent semble assez bien répartie sur l'ensemble du jeu de données.



**Figure 3: Distribution des variables concernant la direction du vent**

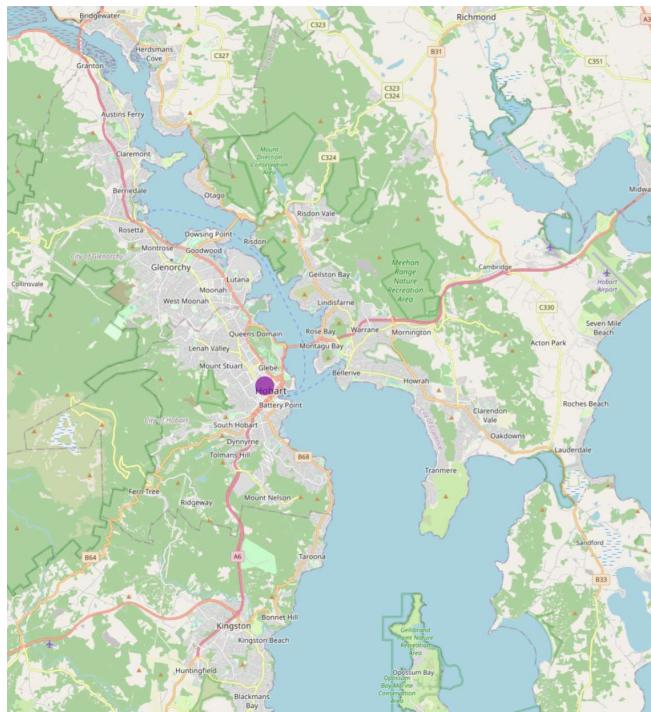
La distribution de la direction du vent est en revanche radicalement différente selon les *Location*, comme nous pouvons le voir dans la Figure 4 à Townsville, Hobart et Bendingo. Il est même étonnant de constater qu'il y a également des distributions très différentes pour une même ville selon la variable observée. Par exemple, on constate que le vent souffle quasiment toujours vers l'ENE à Townsville à 15h00 ainsi que pour les bourrasques, alors qu'il ne souffle que rarement dans cette direction à 9h00 ! La différence de direction prédominante du vent pour une ville côtière entre le matin et l'après-midi peut être expliquée par des effets météorologiques spécifiques, tels que la brise marine, résultant du réchauffement plus rapide de la terre par rapport à la mer, entraînant un mouvement d'air de la mer vers la terre durant la journée.

Ces trois villes ne sont pas particulières dans le jeu de données. Nous retrouvons le même type de différences dans les distributions du vent pour l'ensemble des Location.



**Figure 4 : Distribution des variables concernant la direction du vent à Townsville, Hobart et Bendigo**

Ces différences peuvent parfois se comprendre en observant simplement la situation géographique du lieu. On voit dans la Figure 5 par exemple que la ville de Hobart se trouve dans un estuaire confiné entre une montagne au nord-est et une autre au sud-ouest, expliquant assez logiquement que les vents ne puissent mécaniquement que circuler vers le nord-ouest ou le sud-est.



**Figure 5: Situation géographique de Hobart**

Il nous est possible de tester la corrélation de ces 3 variables qualitatives avec *RainTomorrow* avec un test de  $\chi^2$ , avec l'hypothèse nulle supposant qu'il n'existe pas de corrélation.

La méthode « correlation\_vent » calcule la p-value issu du  $\chi^2$  pour chacune de ces 3 variables afin d'en tester la corrélation avec *RainTomorrow*. La p-value est très inférieure à 0,05 en globalité, tout comme pour la plupart des villes, ce qui permet de rejeter l'hypothèse nulle et donc d'affirmer l'existence d'une corrélation.

Seules trois villes présentent une p-value pour l'une de ces 3 variable supérieure à 0,05. Cependant, même sur ces villes, il y a chaque fois au moins une variable qualitative avec une variable qualitative ayant une p-value <0,05

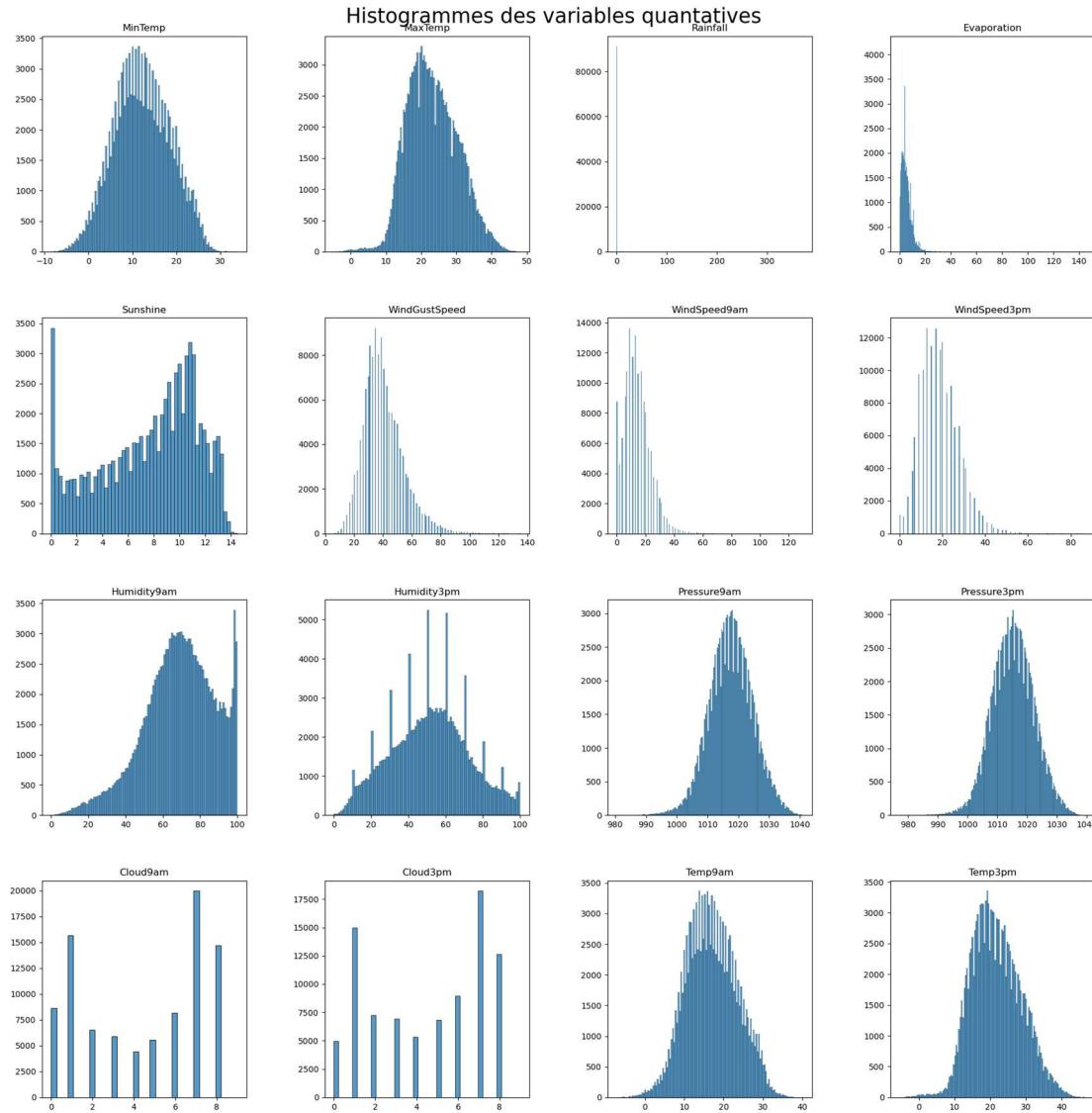
Villes ayant au moins une p-value >0,05 :

	WindGustDir	WindDir9am	WindDir3pm
Canberra	0,00	0,06	0,07
Tuggeranong	0,17	0,00	0,00
Townsville	0,00	0,07	0,00

Nous pouvons donc déduire que la direction du vent est corrélée à chaque ville par au moins une variable. La force de cette corrélation n'est toutefois pas connue, le  $\chi^2$  ne permettant pas de la déterminer.

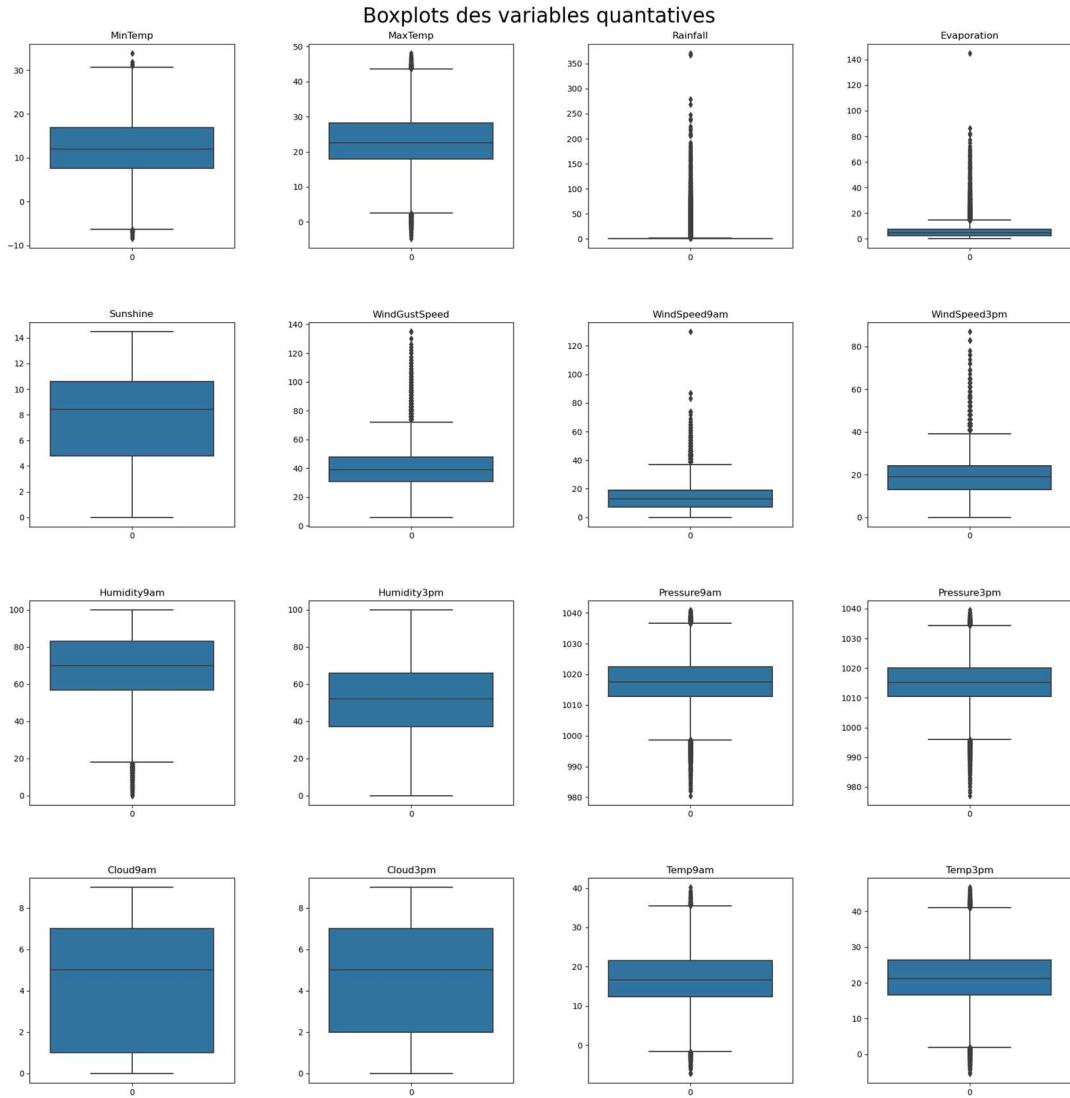
### 2.3.2 Variables numériques

La Figure 6 représente la distribution de chaque variable numérique. Nous pouvons remarquer que seuls certaines d'entre elles sont distribuées presque normalement, comme *MinTemp*, *Humidity3pm*, *Pressure9am*, *Pressure3pm*, *Temp9am*, *Temp3pm*) tandis que d'autres sont soit asymétriques à droite, soit à gauche.



**Figure 6: Histogrammes des variables numériques**

La Figure 7 représente les boxplots des variables numériques continues. On constate qu'il existe une grande variation dans la gamme de valeurs de chaque variable, de sorte qu'un processus de mise à l'échelle sera nécessaire avant la phase de modélisation. On voit en particulier que les deux variables de pression atmosphérique ont un ordre de grandeur de 1000 alors que la plupart des autres sont de l'ordre de quelques dizaines.



**Figure 7: Boxplots des variables quantitatives**

Les boxplots des 16 variables quantitatives nous montrent aussi que plusieurs variables possèdent un nombre important d'outliers. C'est notamment le cas de *Rainfall*, dont la boxplot semble montrer que toute valeur non nulle est aberrante. Cela s'explique assez simplement : *Rainfall* correspond au niveau de précipitations en millimètres. Lorsqu'elle vaut plus de 1, alors *RainToday* est égale à True. Or, nous avons vu précédemment que seulement 22,4% des lignes ont un *RainToday* (ou bien un *RainTomorrow*) à True. Cela implique que dans 77,6% des cas, *Rainfall* a une valeur inférieure à 1. On comprend alors assez aisément que dès que survit une averse, le résultat des précipitations enregistré se retrouvera nécessairement en outlier.

En réalité, bien qu'il existe ici de nombreuses valeurs aberrantes d'un point de vue mathématique, il s'agit bel et bien de données réelles, et non de données erronées dans le jeu de données. Nous trouvons par exemple pour les quatre variables concernées des températures comprises entre -7°C et +46°C, ce qui n'a rien d'absurde. Il en va de même pour les autres variables : les outliers de la pression atmosphérique, de la vitesse des vents et des taux d'humidité ont tous des valeurs compatibles avec des données météorologiques correctes.

Par conséquent, nous faisons à ce stade le choix de conserver l'intégralité des outliers du jeu de données. Cela impliquera d'être très vigilants sur l'usage de calculs basés sur des moyennes.

## 2.4 Corrélation

La Figure 8 représente la corrélation entre les variables numériques. On peut constater qu'aucune variable quantitative n'est fortement corrélée avec *RainTomorrow*. Il existe toutefois des corrélations intéressantes (comprises entre 0,25 et 0,5) avec *Sunshine*, *Humidity3pm*, *Humidity9am*, *Cloud9am*, *Cloud9pm*, *RainToday*.

Pour *Cloud9am* et *Cloud9pm*, comme nous le verrons après, il s'agit malheureusement de deux features ayant un taux élevé de données nulles.

A l'inverse, *RainTomorrow* semble très peu corrélée aux températures (de 0,03 à 0,19).

La vitesse du vent à 9h00 et 15h00 est elle aussi très peu corrélée avec *RainTomorrow* (0,09). La vitesse des rafales l'est en revanche davantage (0,23).

Nous constatons également de fortes corrélations entre d'autres features. Assez logiquement, c'est le cas de la température maximale (*MaxTemp*) avec celle relevée à 15h (*Temp3pm*), et de *MinTemp* avec *Temp9am*. Plus étonnant de prime abord, c'est également le cas de *MaxTemp* avec *Temp9am* (0,89), ainsi que *Temp9am* et *Temp3pm*. Les températures min et max présentent aussi une corrélation intéressante (0,74). Bref, nous constatons que l'ensemble des variables de températures présentent une très forte corrélation entre elles.

C'est aussi le cas de la pression : les variables *Pressure3am* et *Pressure9pm* sont très fortement corrélées (0,96).

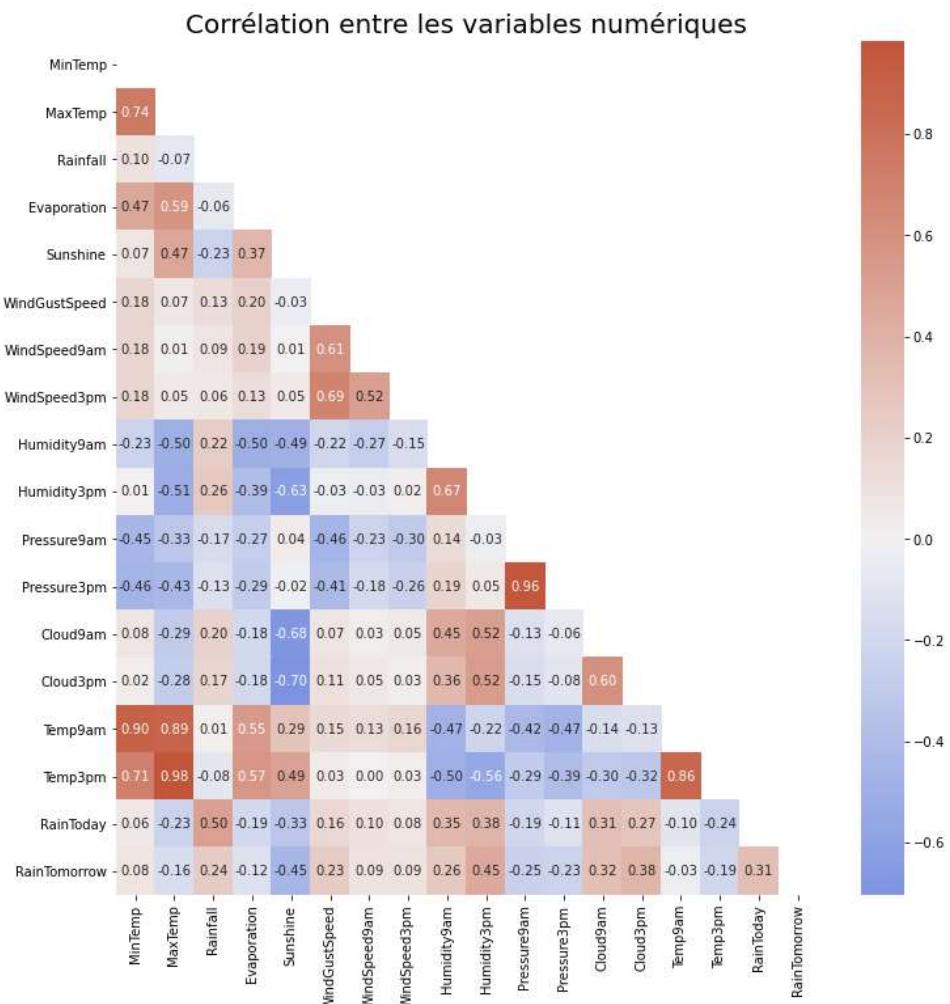
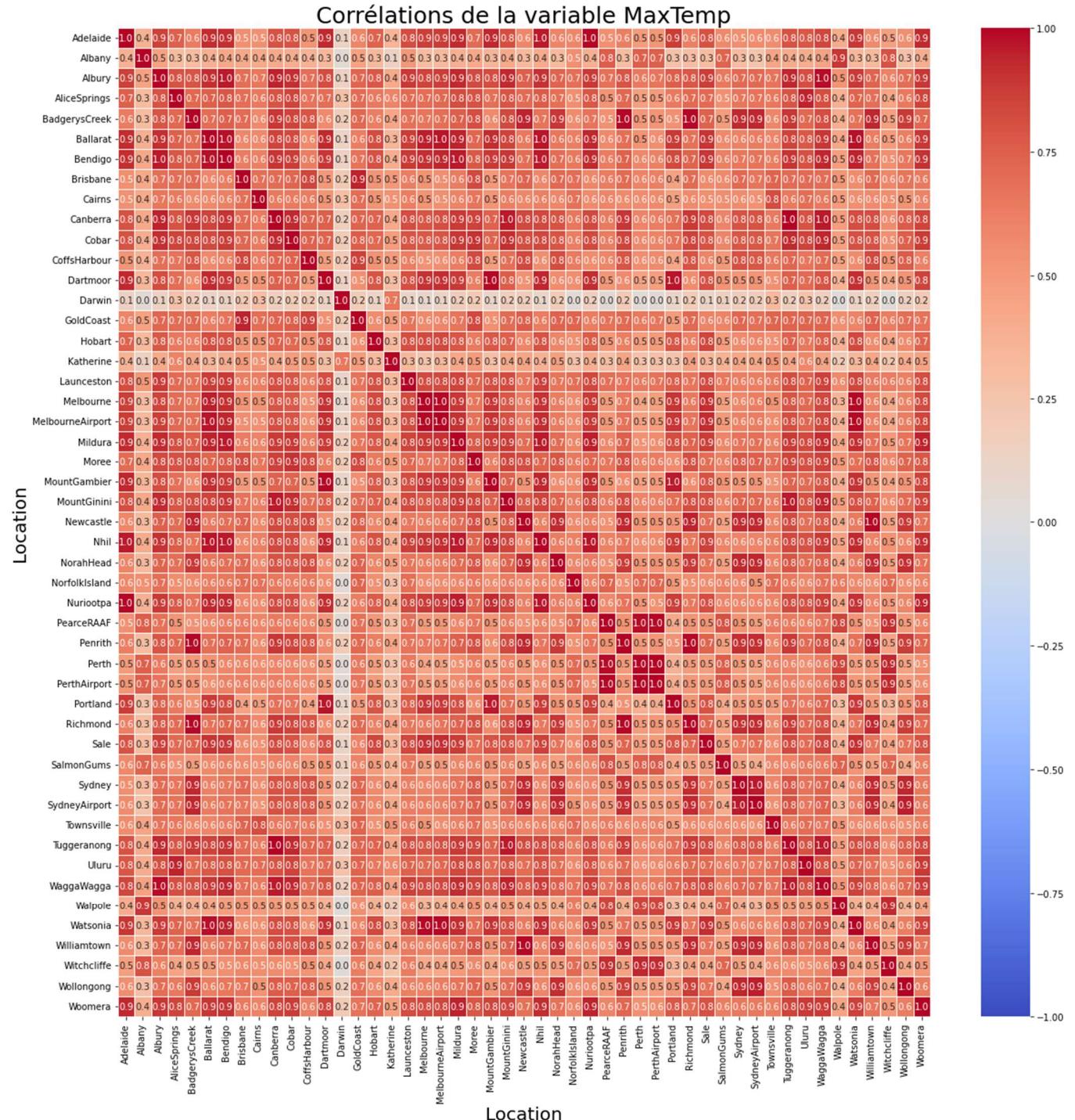


Figure 8: Corrélation des variables numériques

Il est également intéressant d'observer les corrélations entre différents lieux pour une même variable, que ce soit pour clusteriser les villes par climats ou pour trouver des villes corrélées pour une variable donnée.

Nous voyons dans la Figure 9 la corrélation des différentes villes pour la variable "MaxTemp". On constate que certaines villes ont une corrélation très proche de 1, comme c'est le cas de la ville Ballarat avec la ville Bendigo ou la ville de Melbourne. Ceci est logique puisque ces villes sont géographiquement proches les unes des autres. Par contre, pour les villes plus éloignées comme Darwin, la similitude avec le reste des villes est beaucoup plus faible.



## 2.5 Analyse détaillée des variables

Dans cette section, nous allons analyser en détail certaines variables ainsi que les relations entre certaines variables.

### 2.5.1 RainTomorrow

Regardons en premier lieu la variable *RainTomorrow*, indiquant s'il pleuvra le lendemain. C'est en effet cette variable que nous allons utiliser dans un premier temps comme variable cible pour notre modélisation. Elle revêt donc une importance particulière.

Notons que dans la mesure où *RainTomorrow*, comme nous l'avons vu plus haut, est égale à *RainToday* de la veille pour un même lieu, les observations ci-après sont également valables pour *RainToday*.

Sur l'ensemble du dataset, il y a 2.2% des observations n'ayant pas d'information sur cette variable. On observe 22,4% de journées pluvieuses, donc que le nombre de jours où il ne pleut pas est environ 4 fois plus grand que le nombre de jours où il pleut. Cela nous suggère que dans la phase de modélisation, nous pouvons peut-être mettre en œuvre des techniques d'équilibrage des données pour ne pas confondre notre modèle.

Nous constatons également derrière ce rapport de 1 à 4 sur les modalités de *RainTomorrow* se cache une disparité très importante selon les *Location*. Ainsi, il ne pleut que 6,7% des journées à Woomera, village de 300 habitant situé dans le désert, contre 36,5% à Portland, ville portuaire du sud.

Pas moins de 19 Location sur 49 présentent un taux de journées pluvieuses inférieur à 20%, et 3 ont moins un taux inférieur à 10% (Uluru, Woomera, AliceSprings). Ce déséquilibre très important pour certaines villes va représenter un défi pour notre modèle.

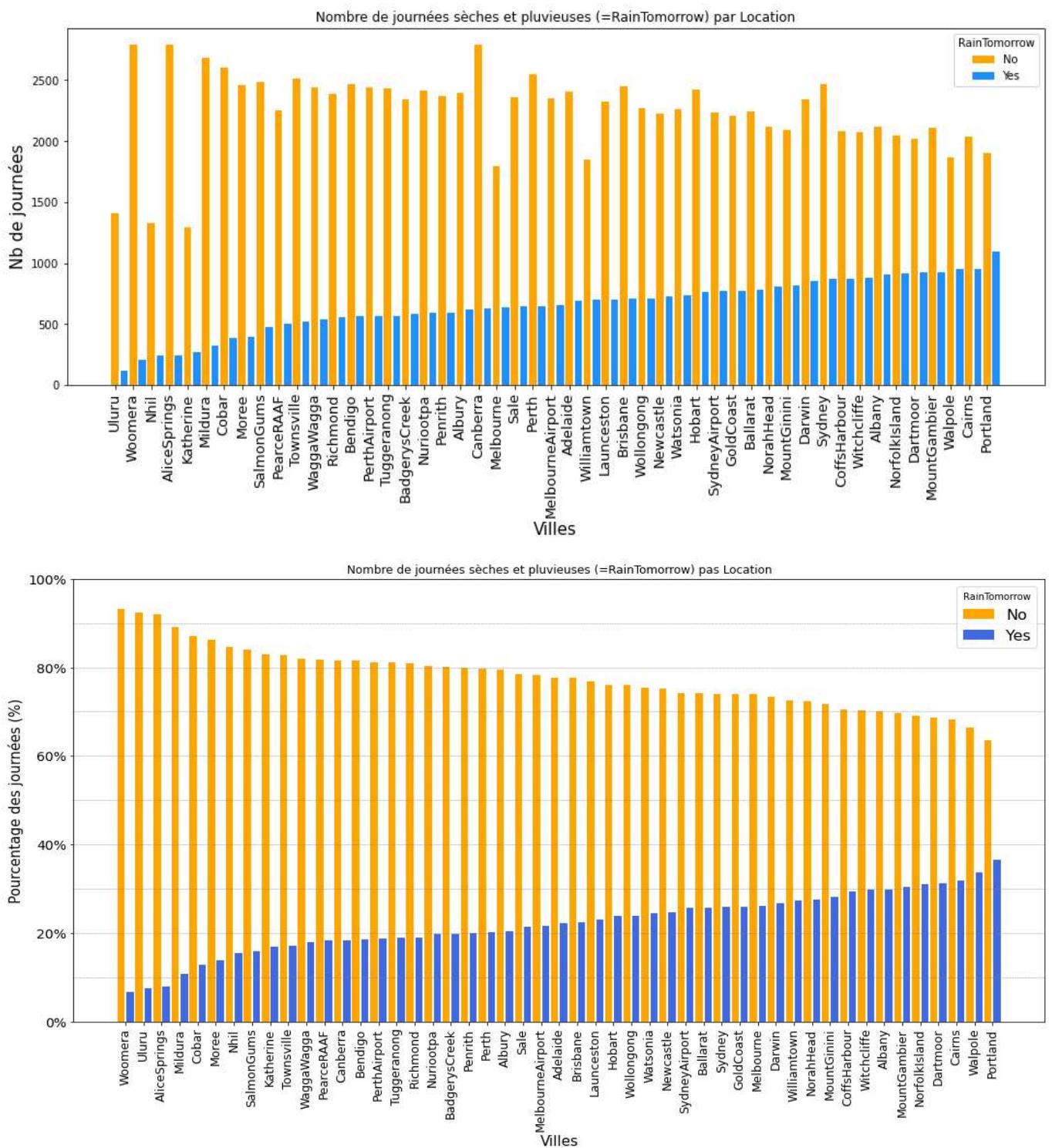
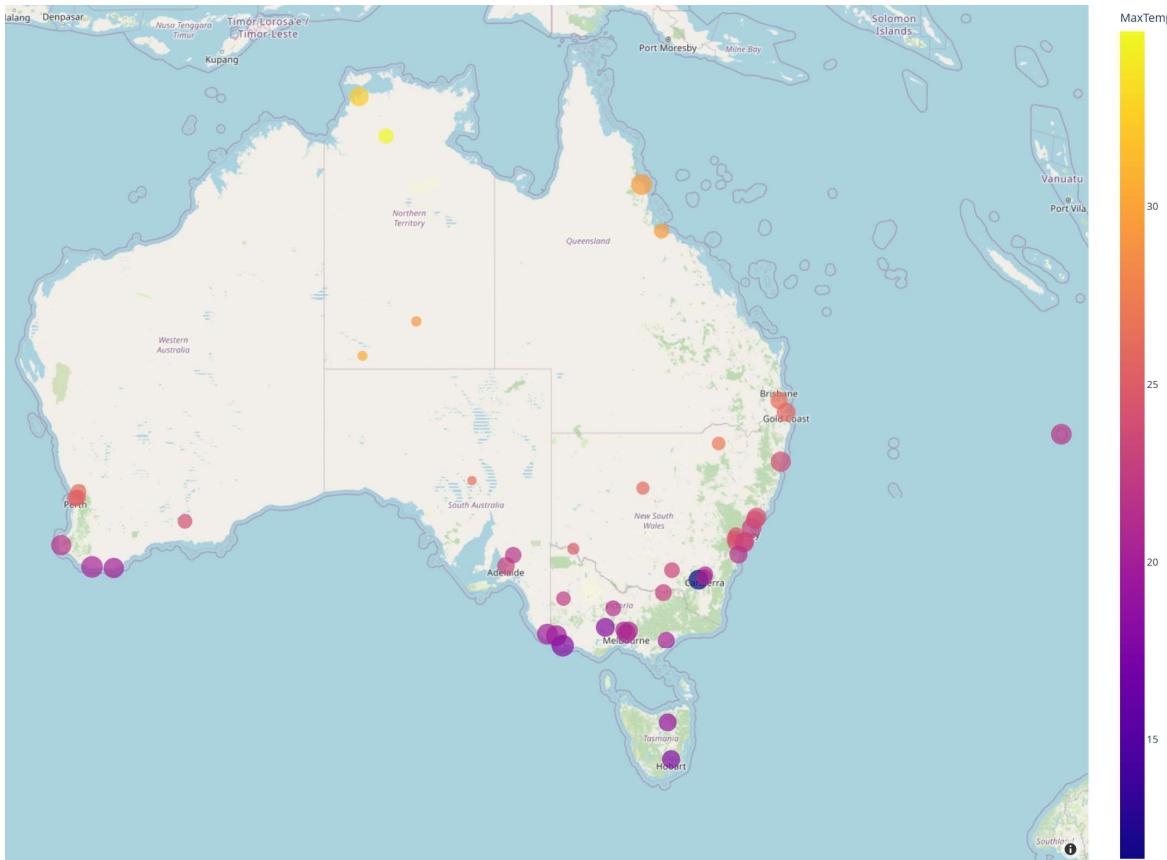


Figure 10 : Nombre de journées sèches et pluvieuses par *Location*

### 2.5.2 *Location, MaxTemp* et *RainTomorrow*

Nous avons, en suite, recherché les latitudes et longitudes des 49 stations météorologiques dans l'objectif de pouvoir les situer sur la carte de l'Australie et mieux comprendre certaines données. Pour cela, nous avons croisé le nom de chaque *Location* avec une liste de villes australiennes de la Australia Cities Database

de Kaggle (<https://www.kaggle.com/datasets/maryamalizadeh/worldcities-australia>), ainsi qu'avec la liste des stations météorologiques ([http://www.bom.gov.au/climate/data/lists\\_by\\_element/alphaAUS\\_139.txt](http://www.bom.gov.au/climate/data/lists_by_element/alphaAUS_139.txt)). Cette opération a nécessité du travail de vérifications manuelles, du fait d'homonymies (Woomera) ou d'orthographes différents (Nhil versus Nhill)



**Figure 11: Location – Couleur : température maximale moyenne – Diamètre : taux de journées pluvieuses**

La Figure 11 montre la répartition des données sur les 49 lieux renseignés. La couleur indique la température maximale moyenne, le diamètre indique la moyenne de la variable *RainTomorrow*. Ainsi, le petit cercle orange représentant Uluru tout au centre de la carte témoigne qu'il pleut très peu dans cette ville et que les températures maximales sont élevées en moyenne (30°).

A l'inverse, le gros point bleu de MountGinini au sud-est témoigne d'une température maximale très faible (11°) et de précipitations plus importantes.

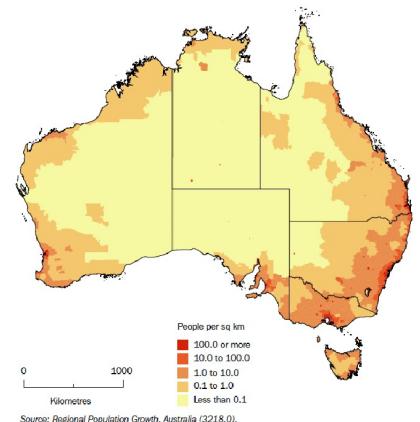
Notons que la ville d'Uluru est située au cœur du désert australien, alors que MountGini est une montagne culminant à 1762m.

On remarque un gradient nord-sud assez net pour les températures maximales.

On remarque également que la fréquence des *RainTomorrow* positif se réduit lorsqu'on s'éloigne des côtes.

Hormis quelques outliers tels MountGinini, on peut également constater une certaine homogénéité climatique entre des villes proches géographiquement, ce qui nous amène à envisager la création de clusters. Enfin, nous voyons que la station de Norfork Island est particulièrement isolée géographiquement, sur une petite île au large oriental.

Il est frappant de constater que le jeu de données comporte essentiellement des informations sur des villes proches des côtes. C'est un point assez logique du fait de la géographie australienne, et c'est également conforme à la répartition de la population sur la carte australienne, comme nous pouvons le voir sur la Figure 12 issue du site de l'Australien Bureau of Statistics :



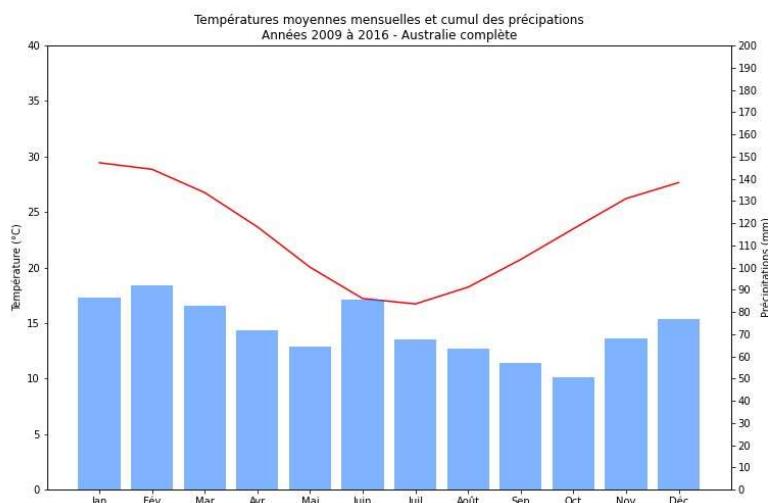
**Figure 12 : Répartition de la population Australienne**

### 2.5.3 Analyse temporelle et géographique

Les données sont trop riches pour pouvoir effectuer une visualisation complète de chaque variable selon chaque lieu au fil du temps, aussi nous nous concentrerons ici sur la température maximale et les précipitations. Ce choix n'est pas aléatoire : d'une part il s'agit des deux variables classiquement utilisées pour la représentation de données météorologiques dans le temps, d'autre part le niveau de précipitation permet de déduire logiquement la valeur de *RainToday* pour un jour donné : prédire *Rainfall* implique donc de pouvoir prédire *RainToday*, et possiblement *RainTomorrow*.

Dans les Figure 13 et Figure 14, nous indiquons la moyenne mensuelle de températures maximales ainsi que le total mensuel des précipitations. Ces données sont effectuées ici uniquement sur huit ans, de 2009 à 2016 inclus, afin de disposer d'années complètes et raisonnablement renseignées.

Il s'agit ici de graphe classique en météorologie. En revanche, l'échelle des précipitations est classiquement graduée avec 2mm pour 1° sur l'échelle des températures : cette proportion d'échelle est un usage habituel pour les pays à climat tempéré, mais n'est pas adapté au climat australien. Nous avons ici choisi plutôt de mettre 5mm pour 1° afin d'avoir des représentations visuelles qui exploitent l'ensemble du schéma pour la majorité des lieux.



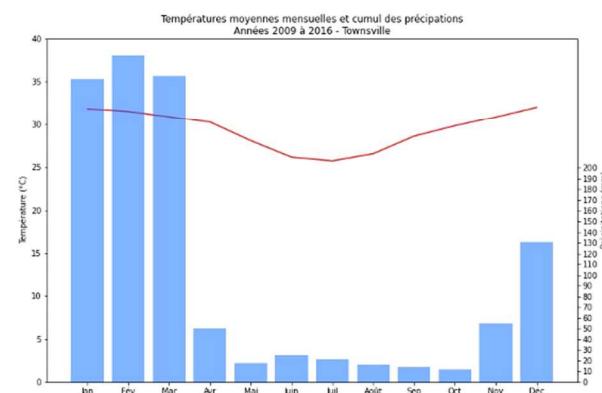
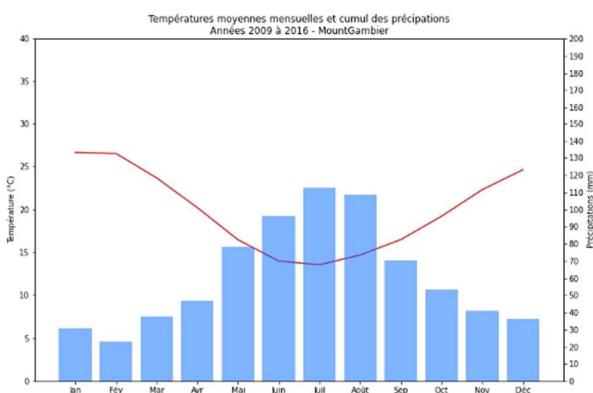
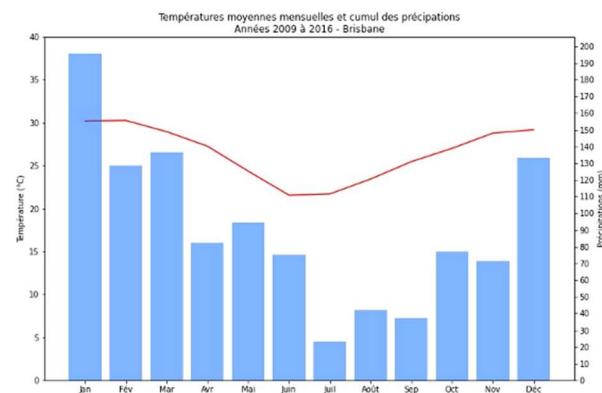
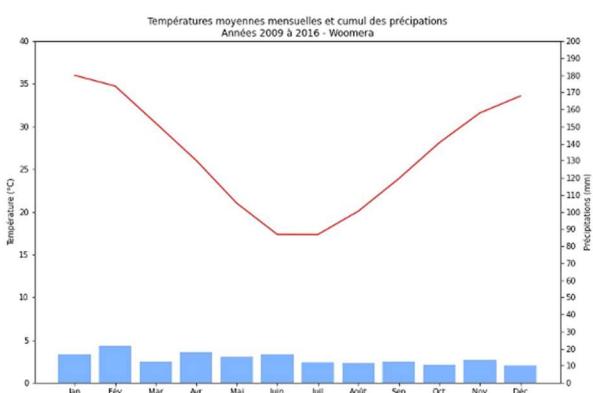
**Figure 13 : Températures moyennes mensuelles et cumul des précipitations de 2009 à 2016 – Australie complète**

Sur ce premier graphe figurent les données pour l'ensemble des 49 location. On y découvre une température élevée en janvier ( $30^{\circ}\text{C}$ ) et plus faible en juillet ( $16^{\circ}\text{C}$ ). Cela s'explique par le fait que l'Australie est située dans l'hémisphère sud. Les saisons sont donc symétriques à celles de la France.

Les précipitations ne semblent visuellement pas corrélées aux températures. Elles varient de 50mm à 90mm en total mensuel, en moyenne par lieu.

Observons maintenant ce même graphique pour 4 lieux très différents. Les échelles allant jusqu'à  $40^{\circ}$  pour les températures et 200mm pour les précipitations sont identiques pour une meilleure comparaison visuelle des schémas, à l'exception de Townsville qui présente des précipitations exceptionnelles et que nous avons choisie pour cette raison.

Woomera est située dans le désert, Brisbane sur la côte est, Mount Gambier sur le côté sud, Townsville est au nord-est.



**Figure 14: Températures moyennes mensuelles et cumul des précipitations de 2009 à 2016 – Woomera, Brisbane, Mount Gambier et Townsville**

Ces quatre graphiques dans la Figure 14 illustrent à quel point il existe plusieurs climats en Australie. Si l'allure de la courbe des températures reste similaire, elle ne présente pas les mêmes valeurs pour chaque ville. Les différences sont tout à fait frappantes concernant les précipitations, avec des niveaux particulièrement faibles à Woomera, une saison pluviale de décembre à mars (été australien) et sèche de juillet à septembre (hiver australien) à Brisbane, des saisons pluviales inversées à Mount Gambier par rapport à Brisbane, des pluies diluviennes à Townsville pendant l'été suivi d'une saison sèche sur le reste de l'année.

Nous pouvons conclure de cela qu'il existe des différences significatives concernant les précipitations et la température maximale entre l'Australie dans son ensemble d'une part et chaque ville d'autre part. En

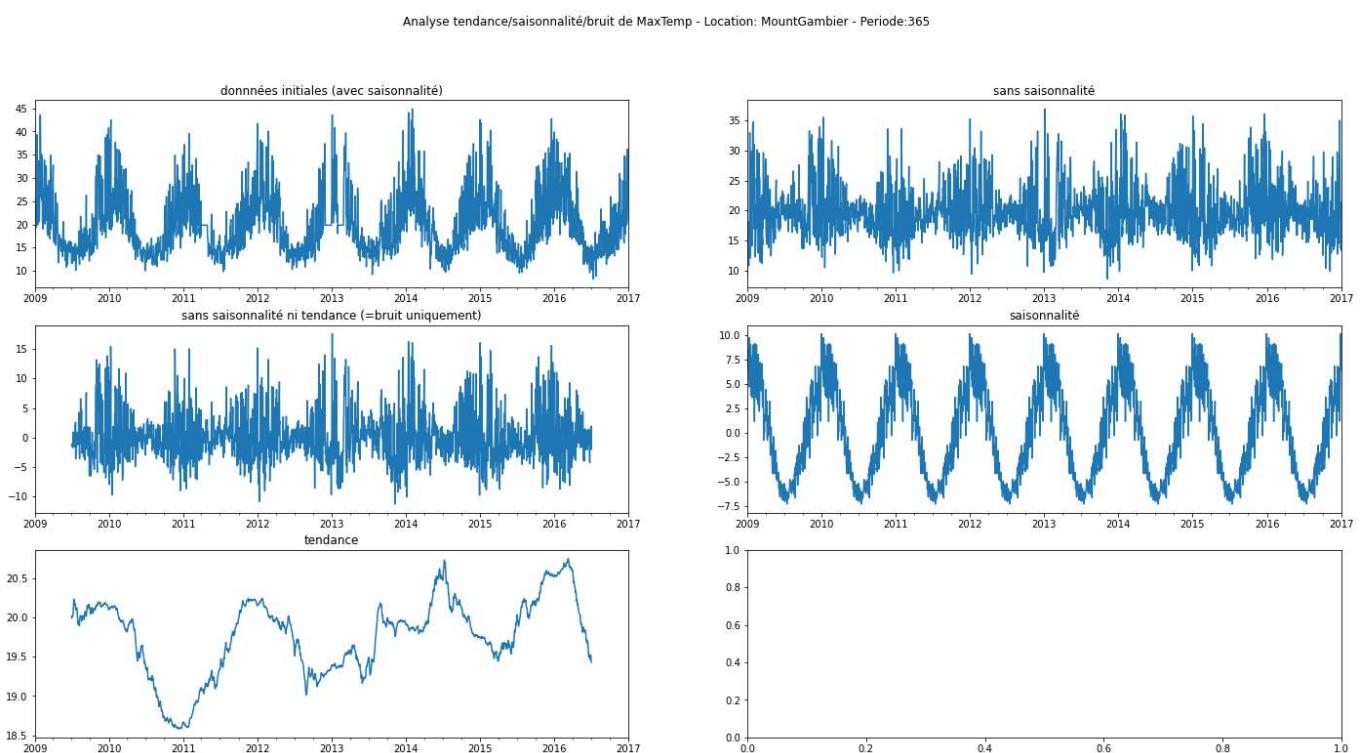
revanche, les villes situées proches géographiquement présentent des caractéristiques communes. Il semblerait donc pertinent d'effectuer des regroupements de villes en fonction de caractéristiques climatiques.

La nature même des données incite à rechercher des saisonnalités dans les différentes variables.

Pour effectuer cette analyse, il convient au préalable de remplir les plages de dates totalement absentes du jeu de données (fonction « reindexation temporelle() »)

Assez logiquement, la décomposition de la température maximale (*MaxTemp*) suit un schéma saisonnier de 365 jours. La fonction « *test\_max\_saisonalite()* » teste plusieurs nombre de journées de saisonnalité. Une durée de 365 jour correspond bien à la fois à une variance maximale de la saisonnalité et d'une variation minimale du bruit.

Nous regardons dans la Figure 15 les décompositions saisonnières pour les variables *MaxTemp* et *Rainfall* de Mount Gambier.



**Figure 15 : Analyse tendance/saisonnalité/bruit de *MaxTemp* - MountGambier**

Nous constatons ici que la saisonnalité explique des variations de températures de  $-7^\circ$  à  $+10^\circ$ , celles-ci variant entre  $12^\circ$  et  $40^\circ$ , soit une amplitude saisonnière de  $17^\circ$  par rapport à une amplitude sur les données initiales de  $28^\circ$ . Ces chiffres encourageants sont à modérer par le bruit restant significatif, puisque variant entre  $-7^\circ$  et  $+12^\circ$ , soit une amplitude de  $19^\circ$ . Nous verrons dans la partie modélisation si un modèle ARIMA arrive malgré ce constat à prédire de façon pertinente les températures.

Ce même schéma sur *Rainfall*, voir la Figure 16, montre des résultats assez similaires : une saisonnalité réelle, mais un résidu restant significatif, et même plus important en variation que le poids de la saisonnalité. Cela nous rend plutôt pessimiste sur la qualité de prévisions avec un modèle de série temporelle univariée tel que SARIMA pour notre variable cible « *RainTomorrow* ».

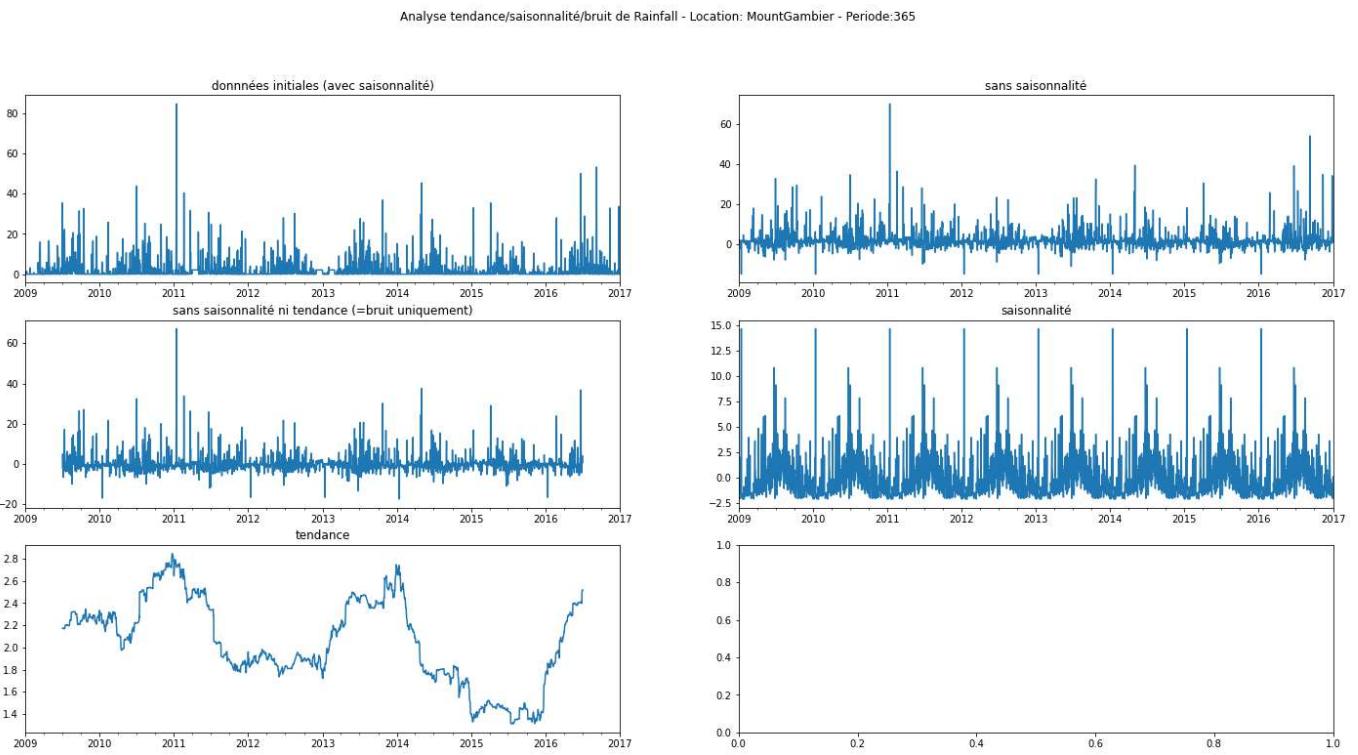


Figure 16 : Analyse tendance/saisonnalité/bruit de *MaxTemp* - MountGambier

## 2.6 Valeurs manquantes

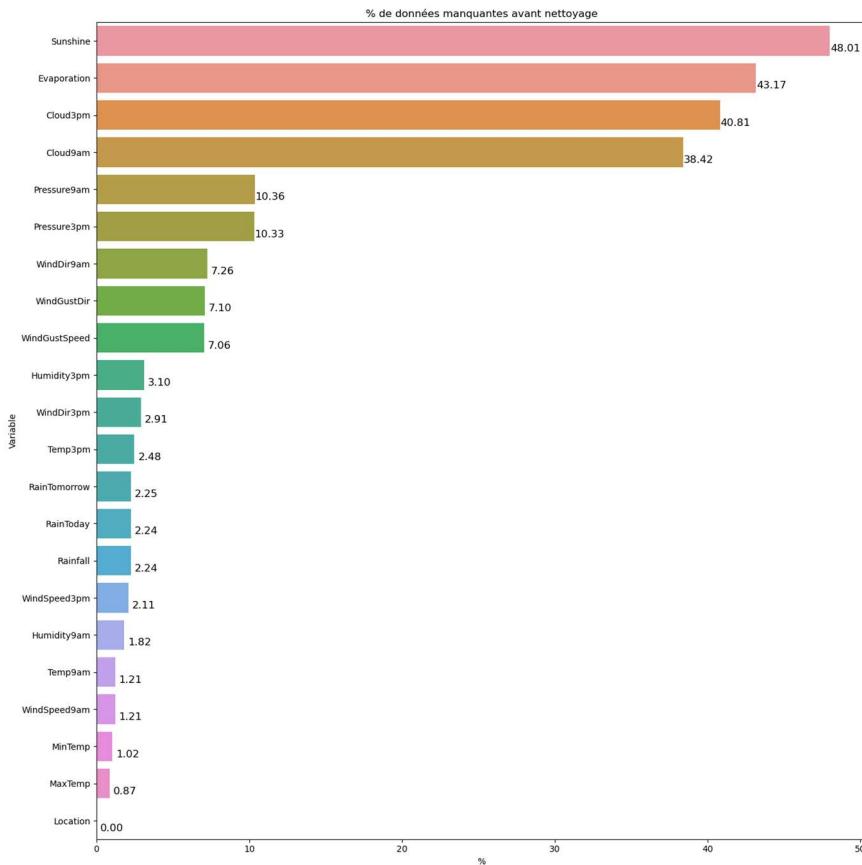
### 2.6.1 Vue globale

Après avoir compris l'ensemble de données, nous devons rechercher les valeurs manquantes. Les valeurs manquantes dans un ensemble de données jouent un rôle très important dans un projet. Si elles ne sont pas traitées, les résultats risquent de ne pas être pertinents.

Les données manquantes peuvent être divisées en 3 catégories :

- **Missing Completely at Random (MCAR)**: L'absence de données est complètement au hasard. C'est-à-dire que les valeurs manquantes n'ont aucune corrélation avec d'autres valeurs de l'ensemble de données observées ou manquantes.
- **Missing at Random (MAR)** : L'absence de données est liée à d'autres variables observées, mais pas à la variable manquante elle-même. En d'autres termes, la probabilité qu'une variable soit manquante peut dépendre d'autres informations déjà présentes dans les données.
- **Not Missing at Random (NMAR)**: c'est le scenario le plus complexe. Dans ce cas, la probabilité que les données soient manquantes dépend de la variable manquante elle-même, même après avoir pris en compte d'autres variables observées.

La Figure 17 représentant le pourcentage de valeurs manquantes pour chaque variable montre que toutes les variables contiennent des valeurs manquantes, sauf *Location* qui est complète. Nous pouvons voir que les quatre variables *Sunshine*, *Evaporation*, *Cloud9am* et *Cloud3pm* comportent un grand nombre de valeurs manquantes.

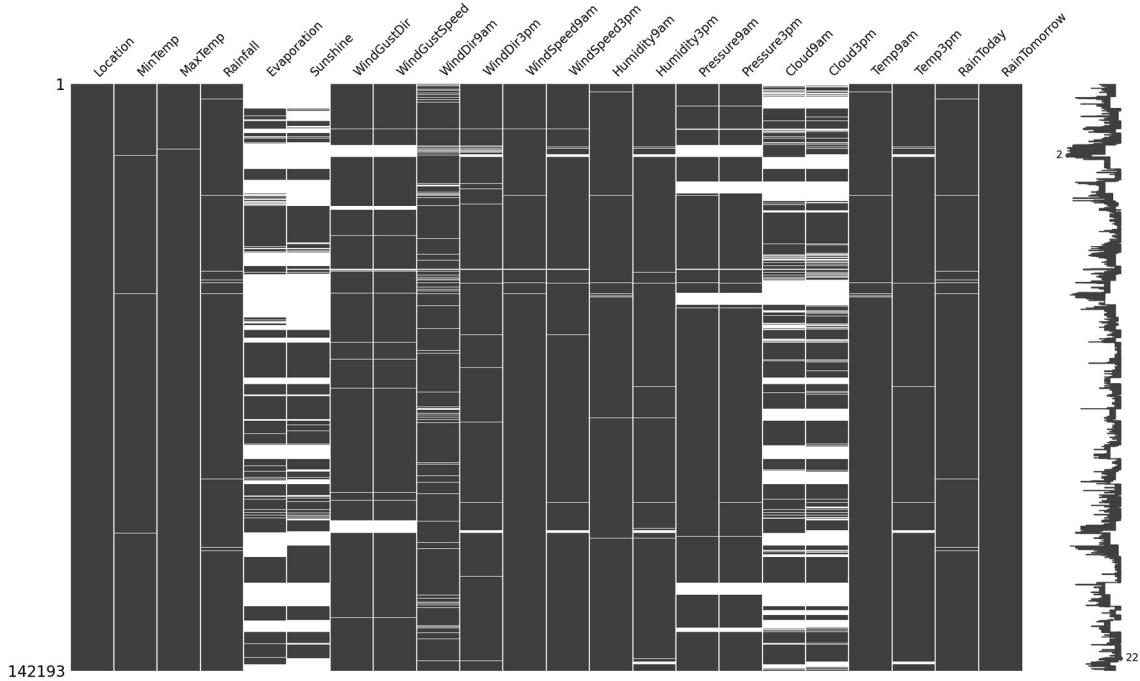


**Figure 17: Pourcentage de valeurs manquantes pour chaque variable**

Ci-après nous voyons une matrice des valeurs manquantes de chaque variable. La couleur de chaque cellule de la matrice est basée sur l'existence ou non des données. Si la couleur est noire, les données existent. Si la couleur est blanche, les données sont manquantes. A partir de ce graphique, nous avons une image de la proportion de données manquantes dans une ligne (observation) ou une colonne (variable).

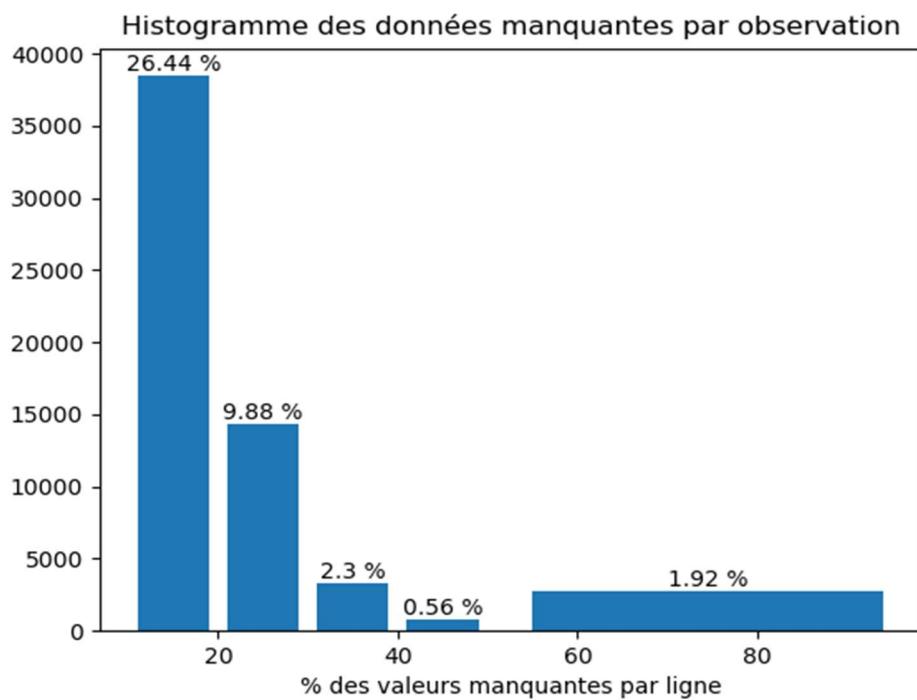
Comme le montre la Figure 18 résultant, les colonnes *Evaporation*, *Sunshine*, *Cloud9am* et *Cloud3pm* affichent de grandes parties de données manquantes. Cela a été identifié dans le graphique à barres ci-dessus, mais l'avantage supplémentaire est qu'on peut voir comment ces données manquantes sont distribuées dans le dataframe.

Sur le côté droit de la matrice se trouve une sparkline qui va de 0 à gauche au nombre total de colonnes dans le cadre de données à droite. Lorsqu'une ligne a une valeur dans chaque colonne, la ligne sera à la position maximale à droite. A mesure que les valeurs manquantes commencent à augmenter dans cette ligne, la ligne se déplacera vers la gauche. On peut observer qu'il y a des lignes (observations) contenant un grand nombre de valeurs manquantes.



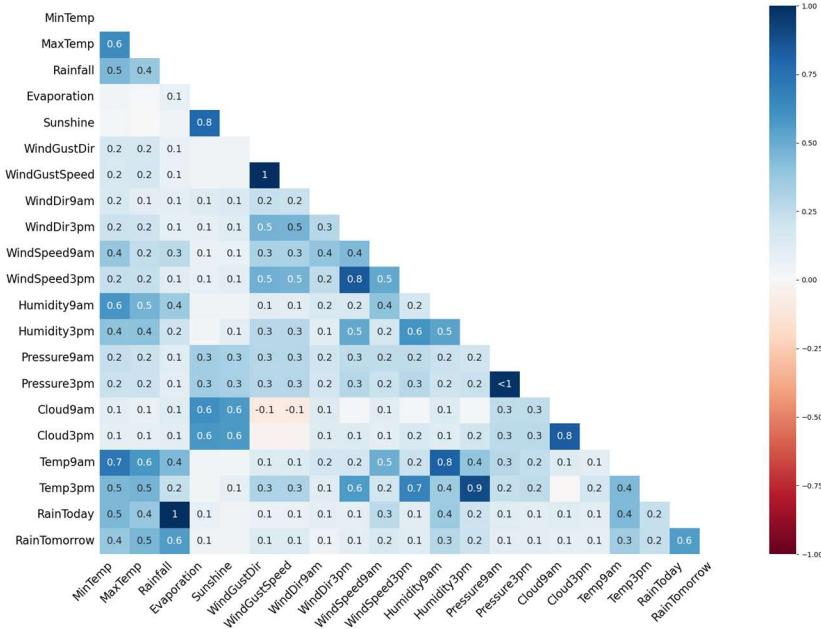
**Figure 18 : Matrice des valeurs manquantes**

Nous pouvons également nous intéresser au taux de variables manquantes non pas par colonnes, mais par ligne, c'est-à-dire par observation. La Figure 19 nous permet de constater que 1,92% des lignes possèdent plus de la moitié des informations manquantes, ce qui les rendra difficilement exploitables.



**Figure 19 : Pourcentage de valeurs manquantes pour chaque observation**

Au total, 41% des lignes possèdent au moins une variable nulle. Il ne semble donc pas envisageable de supprimer toutes ces lignes, et des solutions de remplacement de valeurs manquantes devront être déployées. Pour cela, avant d'envisager une solution basée sur l'exploitation d'autres features, il nous faut connaître la corrélation de nullité entre les différentes variables. C'est ce que montre la Figure 20.

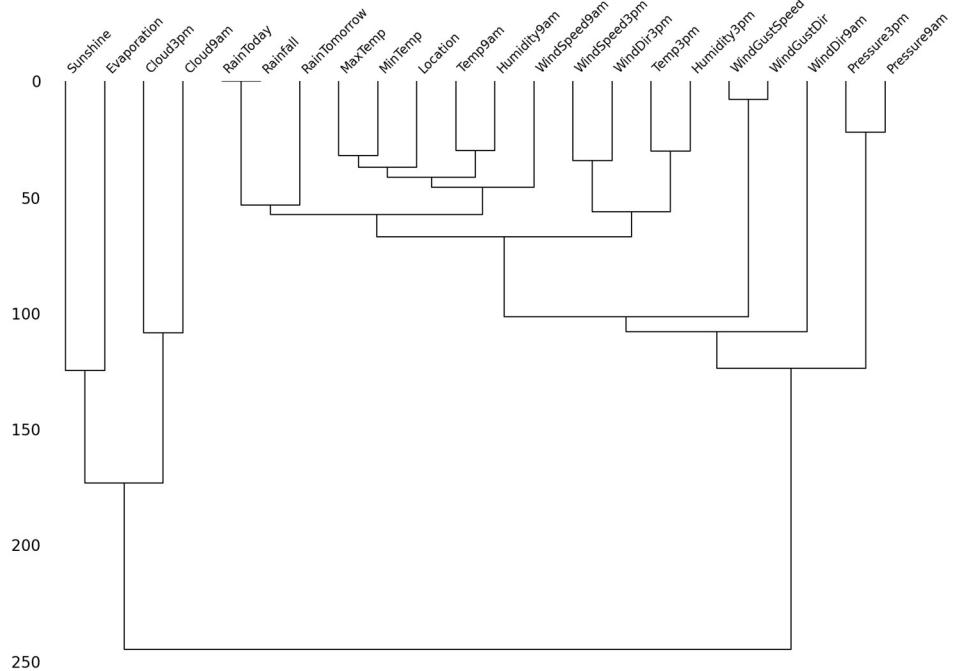


**Figure 20 : Corrélation de nullité entre les variables**

- Les valeurs proches de 1 indiquent que la présence de valeurs manquantes dans une variable est corrélée à la présence de valeurs manquantes dans une autre variable.
- Les valeurs proches de -1 indiquent que la présence de valeurs manquantes dans une variable est anti-corrélée à la présence de valeurs manquantes dans une autre variable. Autrement dit, lorsque des valeurs manquantes sont présentes dans une variable, des valeurs de données sont présentes dans l'autre variable et inversement.
- Les valeurs proches de 0 indiquent qu'il y a peu ou pas de relation entre la présence de valeurs manquantes dans une variable et dans une autre.

Nous pouvons voir dans l'ensemble de données que la variable *WindGustSpeed* et la *WindGustDir* ont une corrélation de 1, ce qui souligne que si la valeur de *WindGustSpeed* est manquante, la valeur de *WindGustDir* sera également manquante. On observe le même effet entre *RainToday* et *Rainfall*. Ce dernier point avait été précédemment vérifié lorsque nous avions regardé si *RainToday* était bien égal à True lorsque *Rainfall* est supérieure à 1 : il ne sera donc malheureusement pas possible de renseigner *RainToday* grâce à *Rainfall*, à moins d'enrichir le jeu de données initial par des données sur *Rainfall* complémentaires.

A partir de ces corrélations, nous pouvons représenter un regroupement hiérarchique des variables comme dans la Figure 21 qui ont de fortes corrélations de nullité. Si plusieurs variables sont regroupées au niveau zéro, la présence de valeurs manquantes dans l'une de ces variables est directement liée à la présence ou à l'absence de valeurs manquantes dans les autres colonnes. Plus les variables sont séparées dans l'arbre, moins les valeurs manquantes sont susceptibles d'être corrélées entre les variables. Dans le graphique de dendrogram, nous pouvons voir qu'il y a deux groupes distincts. Le premier se trouve sur le côté gauche (*Sunshine*, *Evaporation*, *Cloud9am* et *Cloud3pm*) qui ont tous un degré élevé de la valeur manquante. La seconde est à droite, avec le reste des variables qui sont plus complètes.



**Figure 21 : Dendrogramme de nullité entre les variables**

## 2.6.2 Répartition géographique

Regardons maintenant les données manquantes par lieux représentées dans le Tableau 3. Chaque ligne du prochain graphique indique, pour chaque *Location*, le nombre d'enregistrements nuls de chaque variable. La dernière colonne indique le nombre d'enregistrements total (nuls et non nuls).

Nous avions déjà identifié précédemment que les quatre variables *Evaporation*, *Sunshine*, *Cloud9am* et *Cloud3pm* ont un taux élevé de données manquantes. Ce graphe nous montre que cela dépend en réalité énormément des villes. Ainsi, ces variables sont totalement absentes pour certains lieux, et presque toujours renseignées pour d'autres !

Cette représentation nous permet de nous rendre compte que 15% des données *Rainfall* de Williamtown sont manquantes, ce qui est une proportion beaucoup plus élevée que pour les autres villes, alors même que Williamtown est globalement bien renseignée. Nous avons donc téléchargé les données pour *Rainfall* de Williamtown sur le site du Bureau of Meteorology.

La répartition des données nulles sur les autres features est très disparate. Melbourne concentre notamment une large proportion des données nulles sur nombre de variables. Dans une moindre mesure, c'est également le cas de quelques autres lieux (Albany, Canberra, Coffs Harbour, Mount Ginini, Newcastle, PearceRAAF, Sydney, Williamtown). La question du maintien de ces villes dans le jeu de données pourrait se poser pour la modélisation, mais, étant donné notre problématique, cela nous priverait de la possibilité de prédire la météo pour ces villes, ce qui constituerait une perte importante de qualité.

Enfin, nous voyons que certaines villes expliquent à elles seules un nombre important de valeurs manquantes sur une variable, comme pour *WindGustDir*, dont une grande partie s'explique par Albany, Newcastle et Sydney. La ville de Sydney étant par ailleurs plutôt bien renseignée, nous aurions souhaité retrouver cette information dans des relevés météo. Malheureusement, seuls les 14 derniers mois sont disponibles pour les variables relatives au vent sur le site du Bureau of Meteorology. Nous devrons donc tenter de reconstituer les données manquantes par des approches que nous aborderons un peu plus loin.

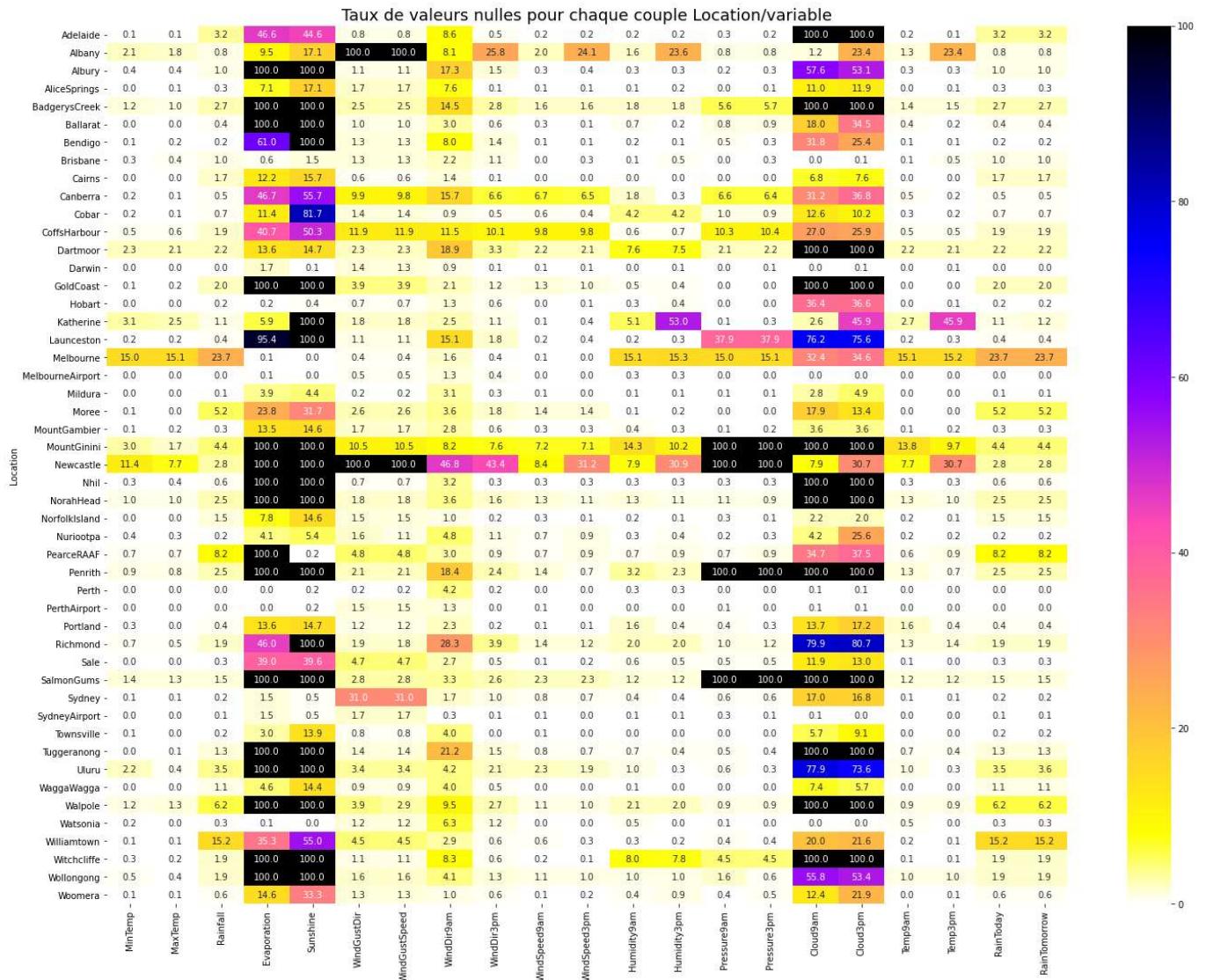


Tableau 3 : Pourcentage des valeurs manquantes pour chaque couple Location/variable

### 2.6.3 Répartition temporelle

Les données disponibles vont du 1<sup>er</sup> novembre 2007 jusqu'au 25 juin 2017, ce qui représente 3525 journées. Toutefois, les enregistrements météo ne recouvrent pas l'intégralité de cette plage. On voit sur le graphique précédent que pour la plupart des villes, seules 3000 journées environ sont disponibles.

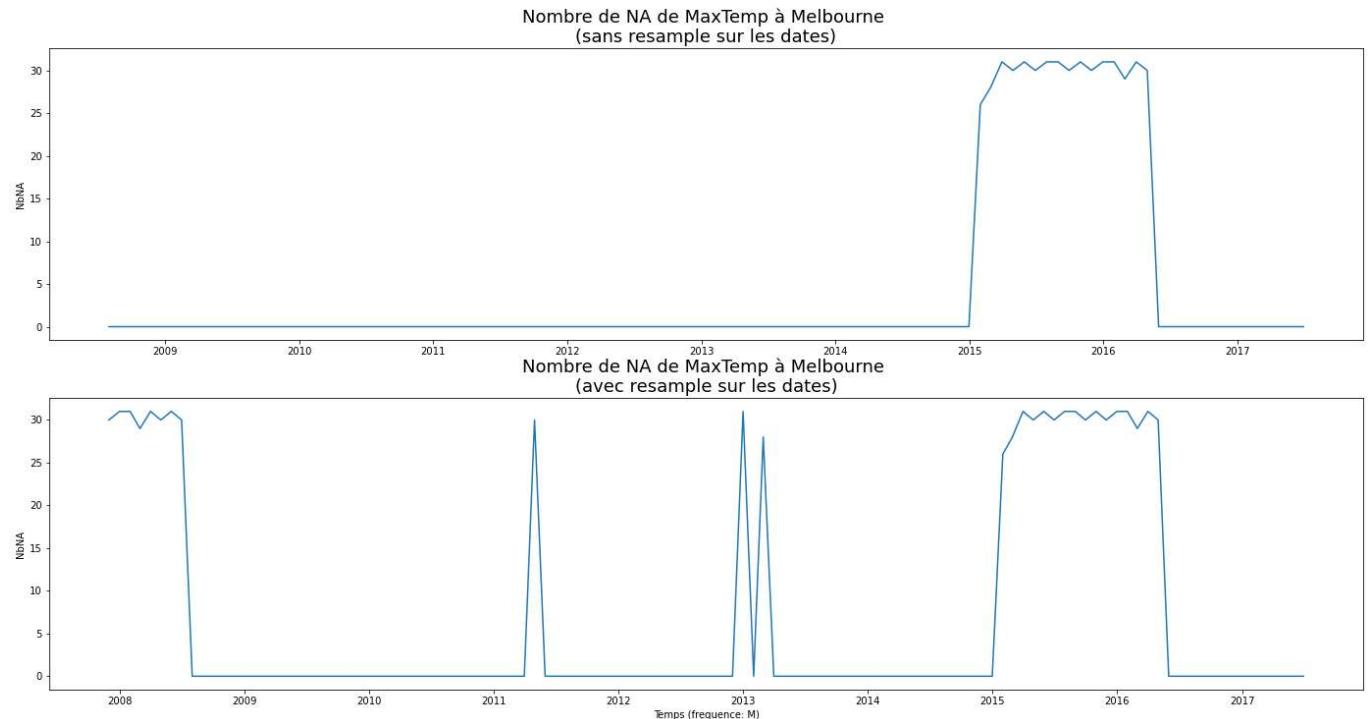
Afin de pouvoir analyser la répartition des données temporellement, il faut préalablement ajouter les journées absentes du jeu de données pour chaque Location. En effet, en l'absence de cette étape, seules les journées connues dans le dataset généreront un NA pour une variable. Les journées absentes quant à elle n'entraîneront pas de NA. Or, nous avons besoin de savoir s'il y a des journées totalement manquantes.

Exemple sur la variable *MaxTemp* pour Melbourne : les deux graphiques présentés dans la Figure 22 indiquent le nombre de journées par mois pour lesquelles il y a des NA vus sur la variable *MaxTemp* à Melbourne.

Le premier graphique, effectué sur le dataset non rééchantilloné, montre qu'il manque la quasi-totalité des données pour chaque mois pour *MaxTemp* de 2015 à mi 2016 mais ne témoigne pas d'autres données manquantes.

Le second graphique, réalisé sur des données rééchantillonée sur l'intervalle complet des dates met en évidence d'autres périodes pour lesquelles *MaxTemp* est inconnue. Il s'agit de dates qui étaient totalement absentes des données d'origines. Par conséquent, *MaxTemp* n'est ici pas seule concernée : si une journée est manquante pour un lieu donné, il va de soi que l'intégralité des variables est manquante pour cette période. Nous pouvons ainsi déduire de ces deux graphes que, pour la ville de Melbourne, il n'existe aucune variable avant mi 2008, ainsi qu'en avril 2011, décembre 2012 et février 2013.

(généré avec : *comparaison\_avec\_sans\_dates\_reindexees("Melbourne", "MaxTemp", "M")*)



**Figure 22 : Nombre de valeurs manquantes de MaxTemps à Melbourne**

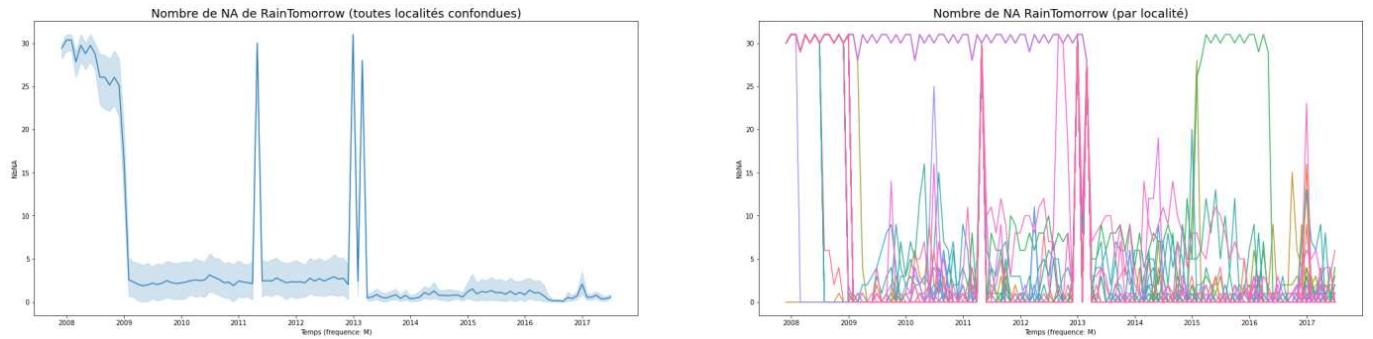
Tentons maintenant de représenter les valeurs manquantes simultanément pour tous les lieux et selon le temps. La Figure 23 illustre les données manquantes pour *RainTomorrow*, la Figure 24 pour *MaxTemp*.

En première colonne, les graphes représentent le nombre de journées pour chaque mois pour lesquels la variable est NA, toutes Location confondues. En seconde colonne, le graphe représente la même chose, mais avec cette fois une courbe par Location. Ces 49 courbes superposées sont évidemment difficilement lisibles en détail mais permettent de montrer quelques tendances intéressantes.

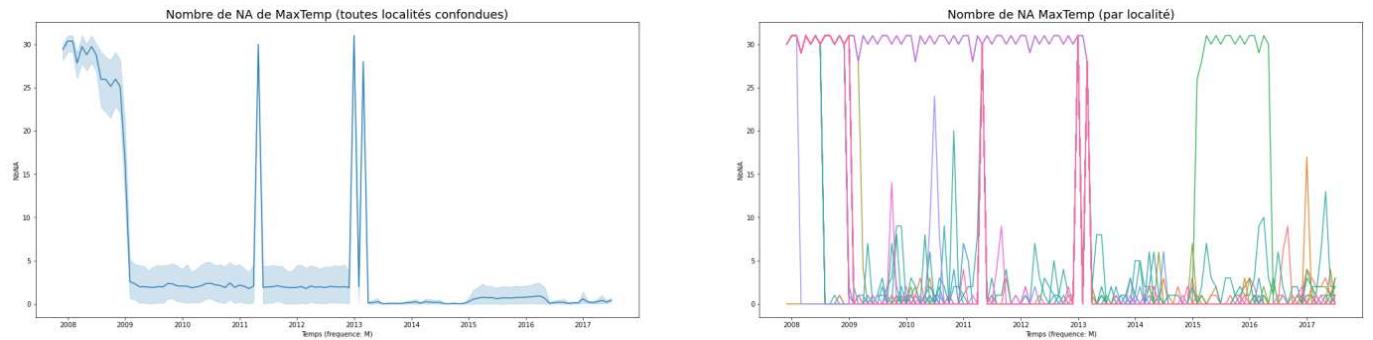
*MaxTemp* et *RainTomorrow* sont globalement rarement disponibles avant 2009. Nous voyons aussi qu'il n'existe aucune donnée pour aucune Location pour le mois d'avril 2011. Il en va de même des mois de décembre 2012 et février 2013. Ces remarques sont vraies pour l'intégralité des features.

Sur *MaxTemp*, il y a une variance de nullité importante sur le nombre de NA/mois/Location entre 2009 et fin 2012. Celle-ci s'affaiblit grandement ensuite, à l'exception de la période de début 2015 à mi 2016. La variance de nullité reste globalement importante sur toutes les périodes pour *RainTomorrow*.

Nous voyons également qu'il y a des Locations qui ont des mois entiers (voire des années !) sans *MaxTemp* ni *RainTomorrow* renseigné. C'est par exemple le cas pour notre tro de villes peu renseignées déjà vu précédemment, Nihl, Katherine et Uluru, qui ne disposent d'aucune donnée avant 2013. Enfin, nous voyons qu'il n'y a jamais de mois pendant lequel *MaxTemp* ou *RainTomorrow* serait disponible intégralement pour l'ensemble des *Location*.



**Figure 23 : Nombre de valeurs manquantes de *RainTomorrow***



**Figure 24 : Nombre de valeurs manquantes de *MaxTemp***

Nous n'allons pas reprendre ici ces graphes pour l'ensemble des variables, mais nous les avons observés (fonction « analyse\_variables\_temps() »). Le graphe de gauche est ainsi quasi identique pour toutes les variables, hormis pour celles dont l'absence est plus fréquente. Le graphe de ces dernières présente logiquement une moyenne mensuelle de NA plus élevée et une variance plus forte. Le graphe de droite, représentant le nombre de NA mensuel par localités, est en revanche très différent selon les variables témoignant d'une forte disparité de la disponibilité des variables suivant les localités.

La faible disponibilité des données avant 2019 pour les différentes variables tend à faire penser que les données antérieures au 1<sup>er</sup> janvier 2019 ne sont pas exploitables (hormis pour Canberra, et Sydney).

La question se pose de la façon de traiter les trois mois intégralement absents des données (avril 2011, décembre 2012, février 2013) : si certaines variables présentent un cycle annuel permettant d'envisager une reprise de la valeur à la même date sur d'autres années (*MaxTemp* par exemple), il n'en va pas de même pour toutes les variables, en particulier la variable cible *RainTomorrow*.

### 3 Pre-processing et feature engineering

#### 3.1 Nettoyage des données

##### 3.1.1 Doublons

Le nettoyage des données est un point essentiel à effectuer avant toute modélisation.

Parmi les premiers éléments constatés, nous avons vu qu'il y avait 22 lignes dupliquées. Chaque enregistrement correspondant à une date pour un lieu précis, les doublons sont donc dans notre jeu de données de véritables données redondantes, et non pas des informations complémentaires pouvant coïncider dont le doublonnage pourrait être une information pertinente. Un simple appel à `drop_duplicates()` permet d'évacuer ces lignes.

### 3.1.2 Traitement des valeurs extrêmes

Comme vu plus haut, les valeurs extrêmes de notre jeu de données sont certes des outliers d'un point de vue mathématiques, mais ne sont pas des données aberrantes au regard des échelles de valeurs et des types de données météorologiques. Nous faisons donc le choix de conserver l'intégralité des outliers après les avoir analysés.

### 3.1.3 Suppression de variables

L'analyse des corrélations à mis en évidence un lien fort entre les quatre variables de température d'une part, et les deux variables de pression d'autre part. La colinéarité de ces variables pouvant nuire à la bonne qualité des prédictions du modèle, nous pourrons supprimer plusieurs features : MinTemp, Temp9am, Temp3pm, Pressure9am. Nous conserverons cependant la possibilité de les conserver ou non selon les modèles utilisés.

### 3.1.4 Suppression des observations

Le traitement de données manquantes est une étape essentielle de la préparation des données pour l'analyse et la modélisation. Dans notre ensemble de données, plusieurs stratégies ont été adoptées pour traiter ces valeurs manquantes.

Il s'agit là d'un point particulièrement complexe à gérer dans notre jeu de données, car, comme nous l'avons vu plus haut, des données sont manquantes sur toutes les plages de dates, pour tous les lieux et pour toutes les variables, dans des proportions différentes.

Cependant, quatre variables ressortent particulièrement, avec environ 40% de données manquantes. Nous porterons donc une attention particulière à l'impact de ces variables sur la qualité des prédictions, et à la cohérence des imputations qui leur auront été faites.

Sur un plan temporel, très peu de données sont disponibles avant le 1<sup>er</sup> janvier 2009. Nous pouvons donc supprimer les données antérieures pour les modélisations qui prendront en compte les dates. Nous conserverons en revanche bien ces observations pour les autres types de modèles.

Trois villes (Katherine, Nhil, Uluru) disposent de moitié moins d'enregistrement que les autres, leurs relevés ne débutant qu'en 2013, soit plus de quatre ans après les premiers relevés des autres stations.

Nous avons adopté une approche progressive pour la suppression des lignes contenant des données manquantes.

- **Suppression des lignes avec des données manquantes pour la variable cible :** La variable cible *RainTomorrow* est essentielle pour notre modèle prédictif. Puisque c'est ce que nous cherchons à prédire, les lignes contenant des valeurs manquantes pour cette variable ont été supprimées, représentant 2.2% de l'ensemble de données. Cette suppression est justifiée car imputer la variable cible pourrait introduire un biais ou une inexactitude dans nos prédictions.
- **Suppression des lignes avec une forte proportion de données manquantes :** Avant de procéder à l'imputation des données manquantes, il est essentiel d'évaluer si certaines lignes ont une proportion exorbitante de valeurs manquantes, au point que leur utilité est mise en question. Dans notre jeu de données, toutes les lignes contenant plus de 50% de valeurs manquantes ont été jugées comme n'ayant pas suffisamment d'information pour être utiles et ont donc été supprimées.

Pour autant, ces suppressions vont dépendre du modèle utilisé : pour une modélisation par série temporelle, il nous sera indispensable de disposer des données sur toute la plage de dates. Les lignes avec des données manquantes seront alors conservées et les données seront déduites par l'une des approches listées plus bas.

### 3.1.5 Complémentation des données manquantes à l'aide d'autre source de données complémentaire

Le site du Bureau of Meteorology nous permet de consulter les relevés des stations. Malheureusement, ce téléchargement ne peut se faire que par station météo, et non en globalité, et seuls les 14 derniers mois permettent de disposer de toutes les variables. En pratique, seules les variables *Rainfall* (dont on peut déduire *RainToday* et *RainTomorrow*), *MaxTemp* et *Sunshine* peuvent être téléchargées. Ce téléchargement doit se faire par *variable* et par station météo, sachant que notre jeu de données n'indique qu'un nom de ville, et non le nom précis de la station. Pour Sydney, par exemple, il n'existe pas moins de 8 stations possibles dont il faudrait donc analyser les données pour déterminer laquelle est la plus proche de notre dataset. Ce travail ne peut donc pas simplement s'effectuer par webscrapping et représenterait un temps considérable d'analyse manuelle. Nous faisons donc le choix de ne télécharger que les données de *MaxTemp* et *Rainfall* pour les Locations qui ont taux de NA élevé. C'est le cas de Melbourne, PearceRAAF et Williamtown pour *Rainfall*, et de Melbourne et Newcastle pour *MaxTemp*. L'exploitation de *Sunshine* nécessiterait de la télécharger pour une quarantaine de Location, ce qui serait trop chronophage en analyse : nous restons sur notre choix d'abandonner cette donnée.

L'obtention de données complémentaires est particulièrement précieuse puisque cela nous permet de déduire logiquement la valeur de *RainToday* manquants, mais également de notre variable cible *RainTomorrow* !

La variable *Rainfall* dispose d'une particularité : contrairement aux autres variables, elle n'indique pas forcément uniquement la valeur pour le jour donné (à savoir le niveau de précipitations en mm), mais cumule parfois les valeurs des jours précédents lorsqu'ils ne sont pas renseignés. En d'autres termes, il est probable que le relevé des pluviomètres ne se faisait pas chaque jour sur la période analysée.

Show in table... ▾

Key: Units = mm 12.3 = Not quality controlled. ↓ = Part of accumulated total  
Move mouse over rainfall total to view the period of accumulation.

Graph 2014 ▾

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Graph	[Graph]											
1st	0	0	↓	0	6.2	↓	0	↓	0.2	0	↓	0
2nd	0.4	0	34.2	0	↓	2.2	0	↓	0	0	↓	3.2
3rd	0	0	4.6	0	↓	0	0	2.0	3.6	0	1.0	0
4th	0	0	0	↓	↓	0	0	0.2	0	0	0	8.6
5th	0	0	0	↓	1.8	1.8	0	0	↓	0	0	↓
6th	0	↓	↓	0	↓	0	0	0	↓	0	17.8	↓
7th	0	↓	30.2	14.4	↓	0	0	0	↓	0	0	25.2
8th	0	↓	0	1.2	36.2	0	↓	35.6	0.4	0	0	4.2
9th	0	↓	0	↓	1.0	0	↓	0.2	7.2	0	0	6.0
10th	0	1.0	0	↓	21.2	0	↓	6.0	0	0	0	0
11th	0	0	↓	21.0	2.0	↓	0.2	0	0	0	0	17.2
12th	0	0	↓	6.8	0	↓	↓	↓	0	0.4	↓	↓
13th	0	0	8.2	12.4	↓	↓	↓	↓	0	0	0	↓
14th	0	0	6.8	0.2	↓	8.6	↓	5.4	12.8	↓	↓	↓
15th	0	0	9.2	0.2	↓	2.4	↓	0	↓	↓	↓	9.4
16th	0	0	20.0	0	0.8	1.0	↓	0	↓	↓	↓	0
17th	0	0.8	0.8	0	0	2.6	33.4	0.2	↓	0.8	0	0
18th	0	27.0	0	0	0	↓	↓	0.4	0	↓	0	1.2
19th	0	0	0	0	0.2	1.0	↓	21.8	↓	↓	0	0
20th	0	17.0	0	0	0	↓	5.0	21.4	↓	18.0	0	0
21st	0.6	0	↓	0	0	↓	2.0	1.2	↓	1.0	0	0
22nd	1.0	0	↓	0	0	6.2	0.2	↓	0.4	0	0	0
23rd	0.6	0	↓	0	↓	0	0.2	↓	0	0	0	2.4
24th	0.6	1.6	2.6	0	↓	0	0	↓	0	↓	12.4	2.4
25th	↓	0	0.4	↓	0.4	0	↓	17.2	0	↓	16.8	0
26th	7.0	0	2.0	↓	0	0	↓	7.0	↓	1.2	0	↓
27th	0	4.4	5.4	23.0	0.2	↓	↓	21.4	↓	0	0	↓
28th	0	17.4	↓	7.2	0	↓	12.8	12.4	3.6	0	0	8.2
29th	0	↓	0	0	0	↓	0	↓	0	0	0	28.0
30th	0	32.4	1.0	↓	0.6	0	↓	0	0	0	0	0.4
31st	0	11.0	10.0	↓	0	6.8	↓	0	0	0	0	0
Highest Daily	1.0	17.4	11.0	20.0	14.4	21.2	2.6	21.8	6.0	12.8	17.8	17.2
Monthly Total	10.2	67.4	94.4	106.4	75.0	73.0	34.8	145.4	55.2	40.6	57.4	108.2

Tableau 4 : Rainfall de Williamtown, année 2014 – site du Bureau of Meteorology (conforme à notre dataset)

Cela a deux conséquences :

- à l'issue d'une plage de dates non renseignées, si *Rainfall* est inférieure à 1mm, alors elle est également inférieure à 1 pour chaque journée de la plage concernée. C'est le cas du 27 au 30 juin.
- Lorsque *Rainfall* est correctement renseignée pour une date donnée, sa valeur ne porte en réalité pas sur ce jour, mais est à répartir sur les jours qui précédent. Dans l'exemple ci-dessus, il n'a en réalité pas plus 27mm le 18 février : il a plu 27mm au total entre le 5 et le 27 février. Il est même possible qu'il n'ait pas plu du tout le 27 février.

Ce constat nous permet de savoir que lorsqu'il a plu moins d'un millimètre une certaine journée, alors il a également plu moins d'un millimètre les jours précédent ayant une valeur non renseignée pour *Rainfall*.

Cela nous permet donc de savoir que dans ce type de situation, les NA de *RainToday* peuvent être remplacés par False. De même, *RainTomorrow* de la veille pourra être affecté à False.

En revanche, lorsque la valeur de *Rainfall* qui suit une séquence de NA est supérieure à un millimètre, il est impossible de déterminer la répartition de la pluviométrie sur la plage de dates. Nous laisserons donc en NA les *Rainfall* dans cette seconde situation.

### 3.1.6 Imputation des données manquantes

Une fois les étapes initiales de suppression terminées, trois méthodes d'imputation distinctes ont été envisagées pour traiter les données manquantes restants dans les variables numériques.

**Imputation par la Moyenne** : cette méthode remplace les valeurs manquantes par la moyenne de la colonne correspondante.

**Imputation par la Médiane** : les valeurs manquantes sont remplacées par la médiane de la colonne.

Les graphiques Figure 25 et Figure 26 illustrent les distributions des variables avant et après l'imputation. Sur chaque graphique :

- La courbe noire dépeint la distribution de la variable en présence des données manquantes.
- Les courbes rouge et bleue illustrent respectivement les distributions après l'imputation par moyenne et par médiane.

Les méthodes d'imputation par la moyenne ou par la médiane sont certes simples à mettre en œuvre. Toutefois, comme le montrent les graphiques elles peuvent altérer considérablement la distribution des variables, en particulier lorsque celles-ci présentent un taux élevé de données manquantes. De plus, rappelons que nous avons fait le choix de conserver les outliers, ce qui implique une instabilité de la moyenne.

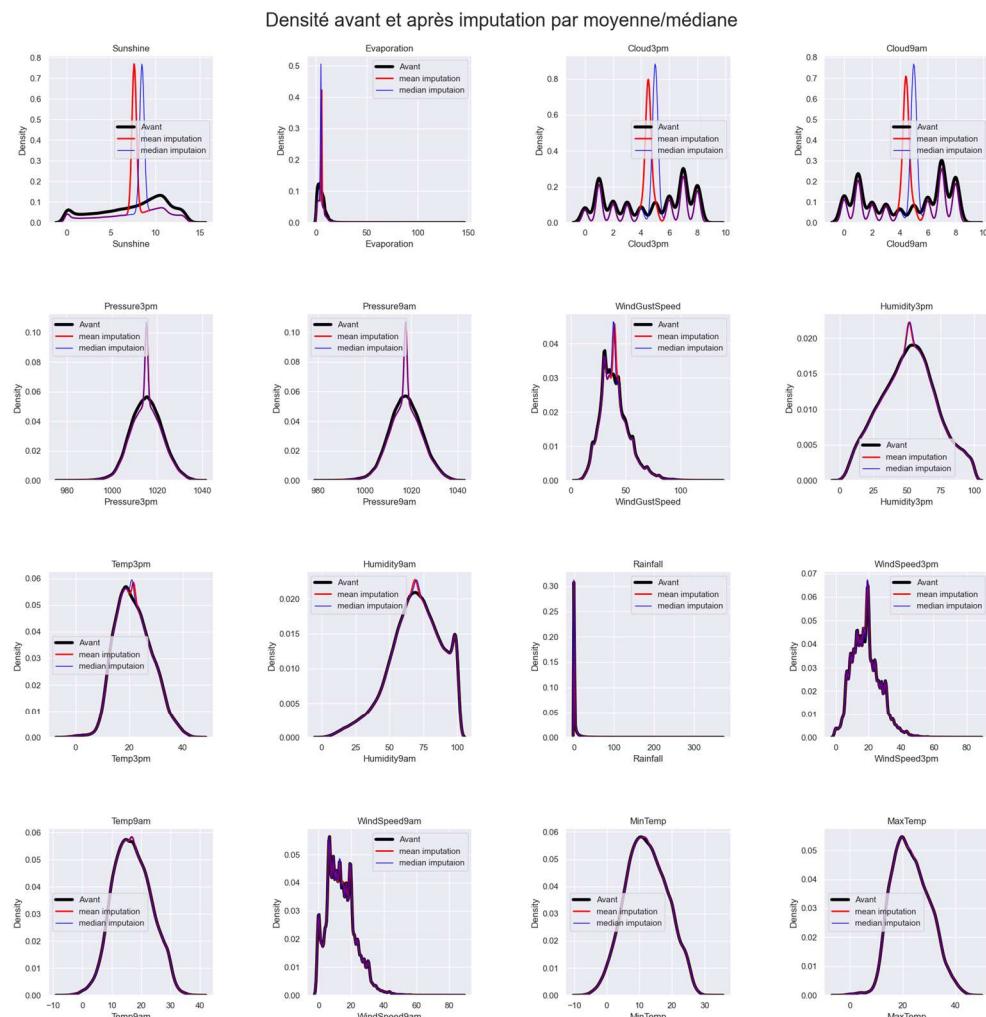
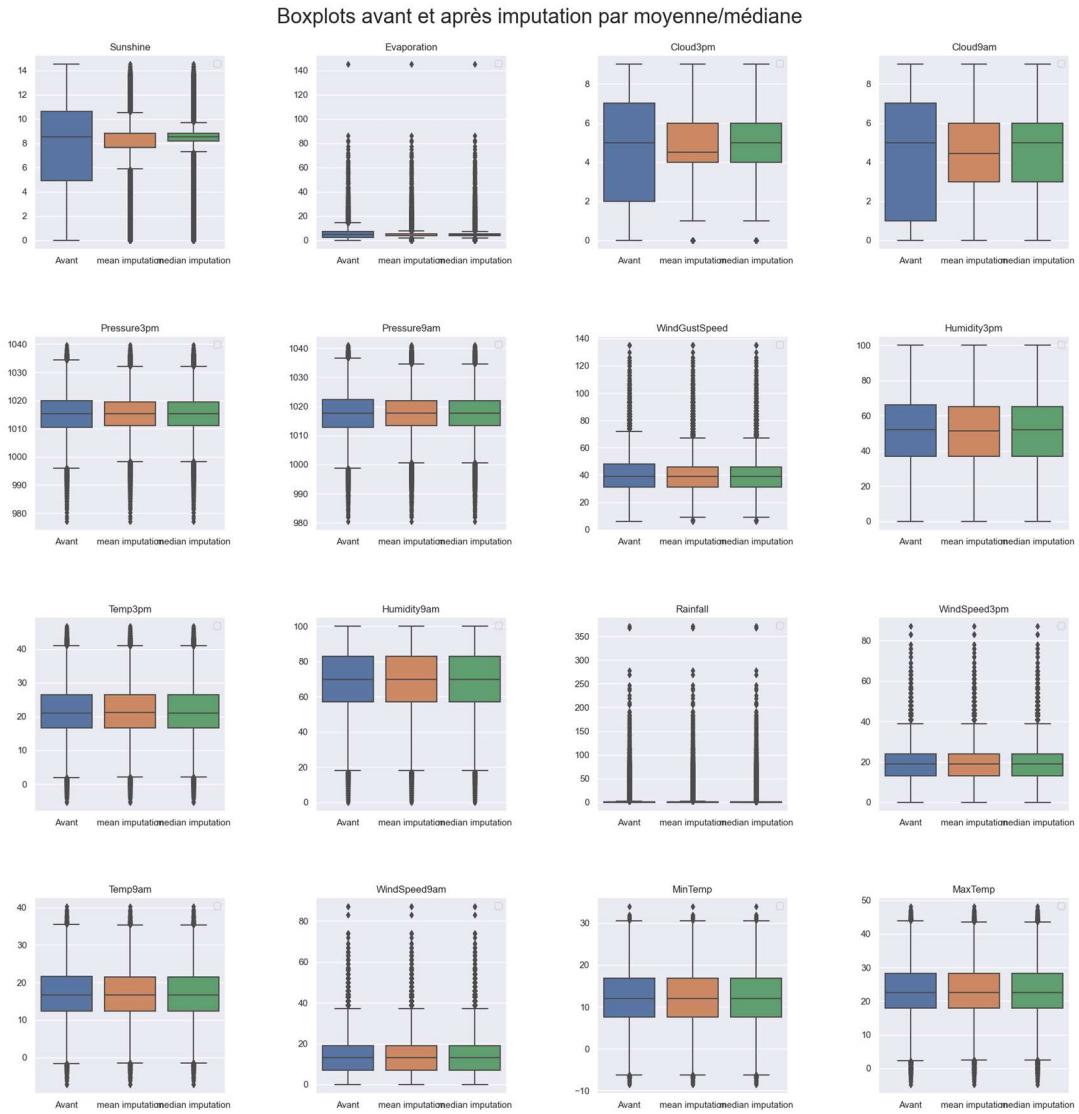


Figure 25: Distribution des variables avant et après l'imputation par moyenne/médiane



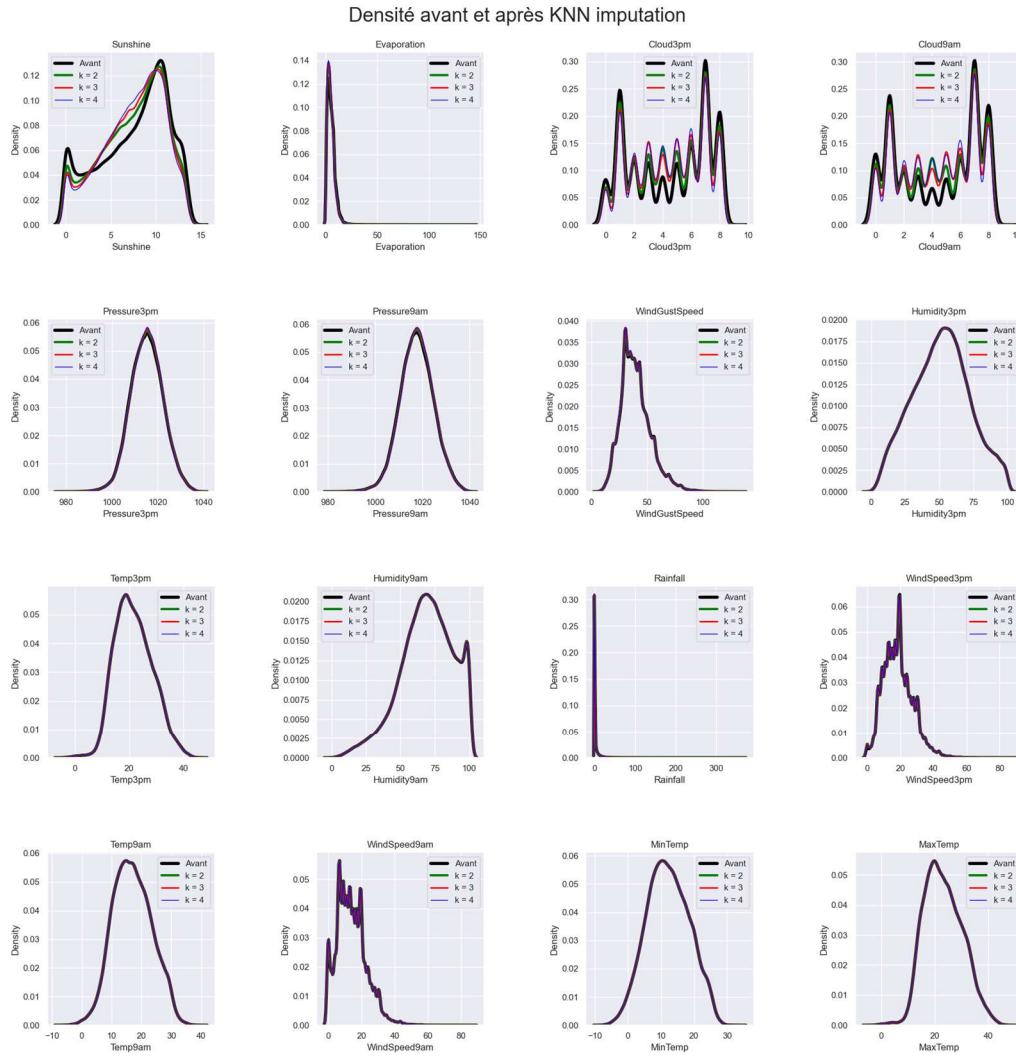
**Figure 26 : Boxplots des variables avant et après l'imputation par moyenne/médiane**

**Imputation KNN :** L'imputation basée sur les  $k$  plus proches voisins est une méthode plus sophistiquée qui prend en compte les similarités entre les observations pour imputer les données manquantes. Au lieu de remplir avec une valeur unique (comme la moyenne ou la médiane), elle utilise les  $k$  observations les plus similaires pour estimer la valeur manquante. Bien que cette méthode puisse être plus précise, elle peut être assez gourmande en temps et en ressources, en particulier pour de grands ensembles de données. La raison est en effet, pour chaque point avec une valeur manquante, l'algorithme KNN essaie de trouver les «  $k$  » voisins les plus proches en calculant la distance entre le point cible et tous les autres points. Ensuite, il utilise ces «  $k$  » voisins pour imputer la valeur manquante. Pour un ensemble de données de taille 145 460 lignes dont il y a environ 60% de lignes contenant au moins une valeur manquante, cela signifie potentiellement des milliards de calculs de distance.

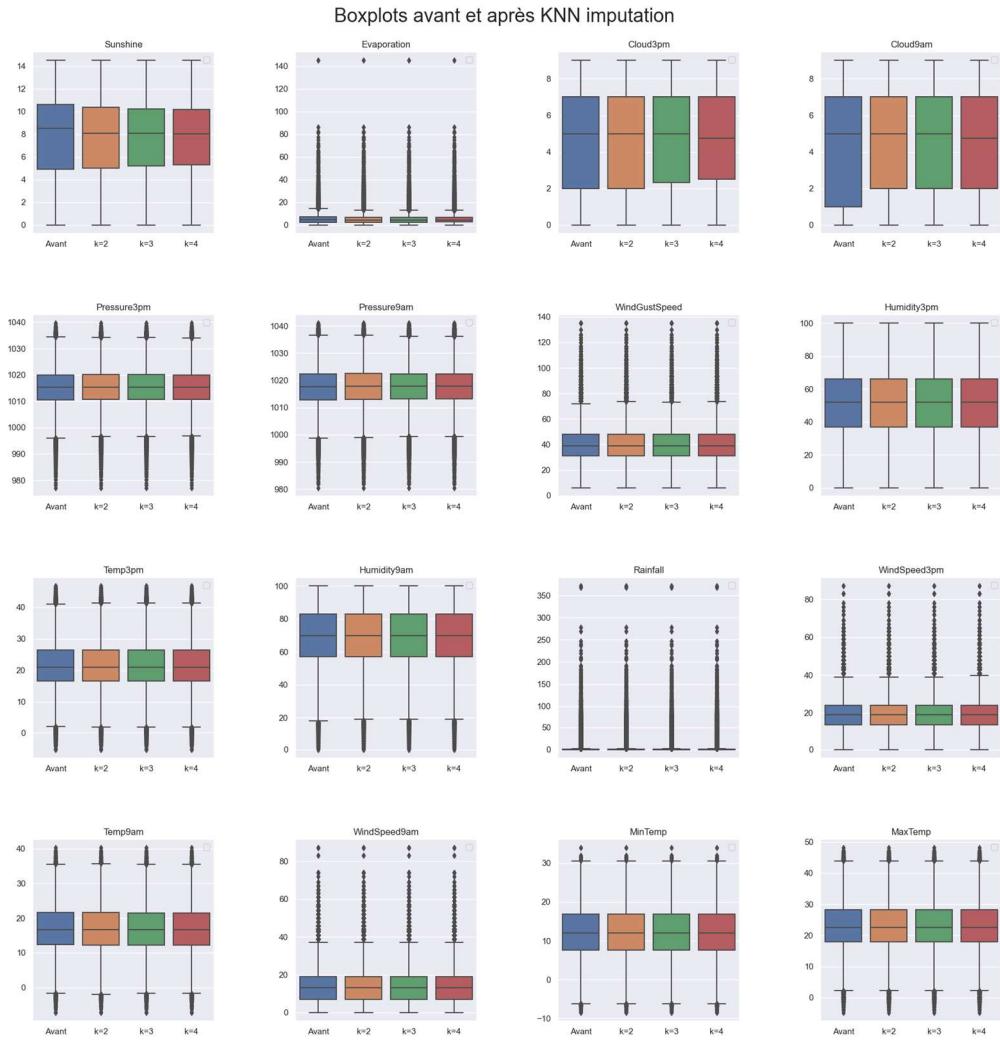
Pour déterminer la valeur optimale du paramètre  $k$  dans l'imputation KNN, nous avons testé différentes valeurs pour  $k$ , allant de 2 à 4. Les graphiques ci-dessous illustrent les distributions des variables avant et après l'imputation. Sur chaque graphique :

- La courbe noire dépeint la distribution de la variable en présence des données manquantes.
- Les courbes verte, rouge et bleue illustrent respectivement les distributions après l'imputation KNN pour  $k = 2, 3, 4$ .

Il est à noter que pour les variables, *Sunshine*, *Evaporation*, *Cloud9am* et *Cloud3pm*, qui présentent un pourcentage élevé de valeurs manquantes, l'imputation KNN impacte plus sensiblement leurs. En revanche, pour les autres variables avec moins de 10% de valeurs manquantes, l'imputation KNN ne perturbe pas de manière significative leur distribution initiale.



**Figure 27 : Densités des variables avant et après l'imputaion KNN**



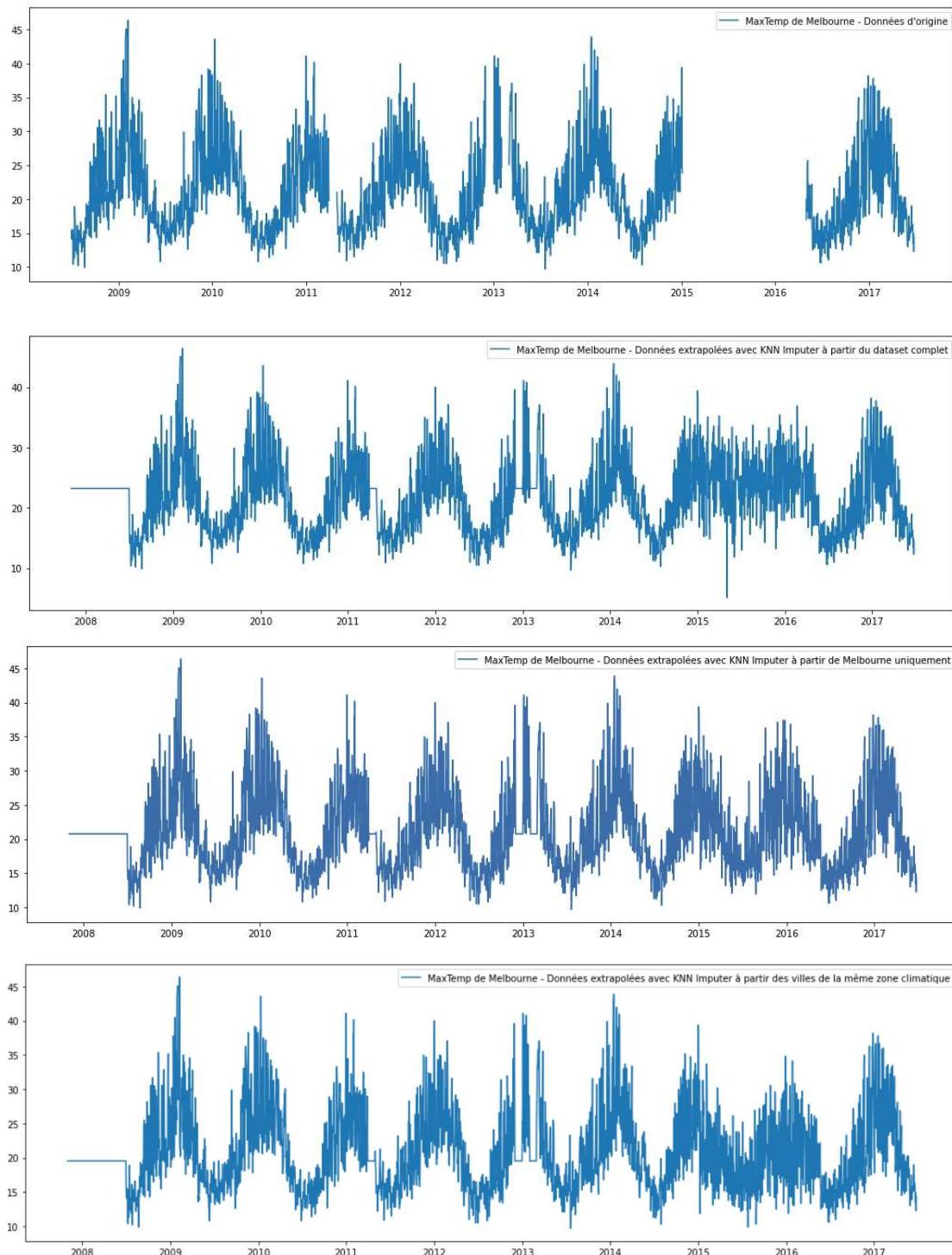
**Figure 28: Boxplot des variables avant et après l'imputation KNN,  $k = 2, 3, 4$**

Après avoir analysé les différentes méthodes d'imputation, il est clair que l'imputation KNN offre des résultats plus fidèles et cohérents comparativement aux méthodes d'imputation par la moyenne ou la médiane. Les distorsions introduites par ces deux dernières techniques, particulièrement visibles dans notre contexte, rendent l'imputation KNN nettement supérieure en matière de préservation de la structure initiale des données. Par conséquent, nous avons décidé d'adopter l'imputation KNN comme méthode privilégiée pour traiter les données manquantes dans cet ensemble de données.

Du fait des diversités climatiques et des spécificités locales vu dans la première partie, nous avons observé les résultats de la KNN imputation effectuée dans un premier temps sur l'ensemble du dataset, puis dans un second temps uniquement pour une ville donnée. Nous voyons ici le graphique de *MaxTemp* de Melbourne :

- Le premier graphe indique les données originales du dataset, débutant mi 2008, comportant un trou de début 2015 à mi 2016, ainsi que des trous pour les mois d'avril 2011, décembre 2012 et février 2013
- Le second graphe montre le résultat de la KNN imputation ( $k=3$ ) à partir de l'ensemble du dataset : les données de 2015 à mi 2016 sont renseignées d'une façon non satisfaisante, les autres plages manquantes sont remplacées par une valeur unique (temps d'exécution d'environ 30 minutes)

- Le troisième graphe a été réalisé sur un KNN imputer ( $k=3$ ) uniquement sur les données de Melbourne : la plage de 2015 à mi 2016 est renseignée de façon plus satisfaisante, mais les autres plages manquantes sont également remplies par une valeur unique (temps d'exécution instantané)
- Le quatrième graphe a réalisé l'imputation à partir des données des *Location* de la même zone climatique (temps d'exécution d'environ 30 secondes)



## 3.2 Transformation des données

### 3.2.1 Booléens

Nous avons deux variables booléennes : *RainToday* et *RainTomorrow*. De façon assez classique, nous allons remplacer les True par des 1 et les False par des 0.

### 3.2.2 Directions du vent

Nous avons trois variables (*WindGustDir*, *WindDir9am*, *WindDir3pm*) donnant la direction du vent selon 16 modalités. Une possibilité est d'effectuer un encodage OneHot, ce qui aboutira au remplacement de ces 3 variables par 45 nouvelles. C'est évidemment considérable.

Une autre approche consiste à considérer le vent selon à partir d'une approche trigonométrique. Un vent ENE pourra ainsi être vu comme un vent d'angle  $\pi/8$ , un vent sud (S) pourra être considéré comme un vent d'angle  $3/2\pi$ , etc. L'angle ne permettra cependant pas au modèle de percevoir qu'un angle de  $15/8\pi$  est très proche d'un angle de 0. Plutôt que l'angle, nous allons donc considérer les composantes directionnelles X et Y, grâce respectivement au cosinus et au sinus de l'angle du vent.

Ce faisant, nous substituerons seulement 6 nouvelles variables numériques aux trois variables qualitatives.

De plus, allons multiplier ces nouvelles variables par les trois variables de vitesse de vent. Nous disposerons ainsi dans nos trois nouvelles variables de la vitesse du vent pour chaque composante directionnelle, et nous pourrons éventuellement supprimer les trois variables numériques de vitesse initiales.

## 3.3 Ajout de variables

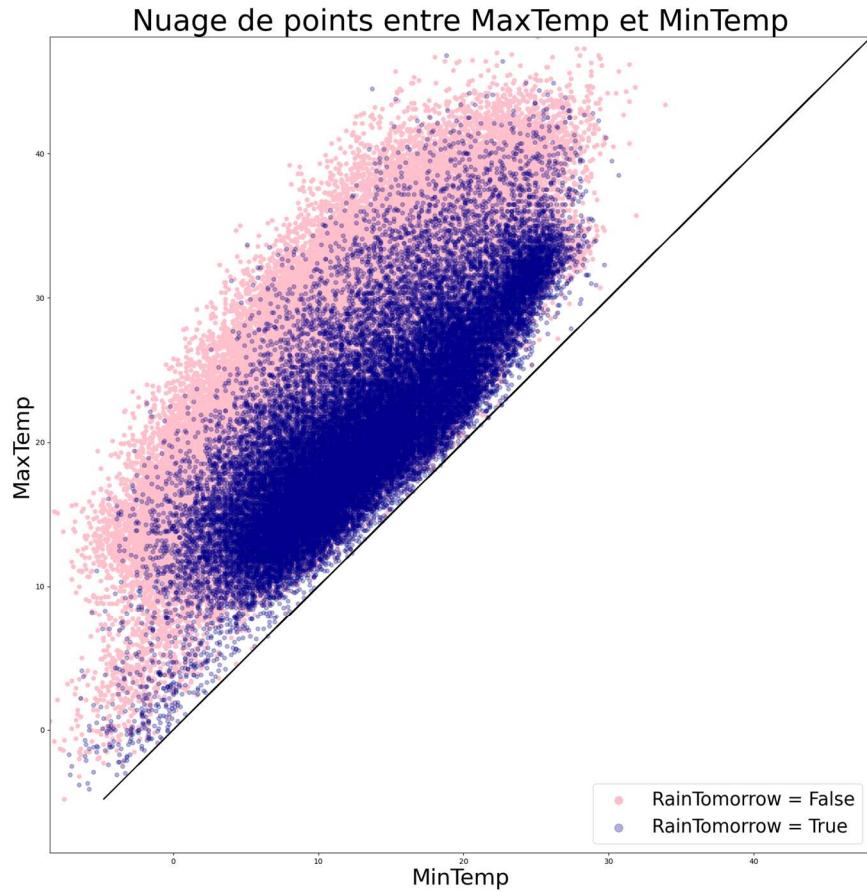
### 3.3.1 Coordonnées des villes

Comme vu en début de rapport, nous avons rapidement recherché les coordonnées de chaque ville afin de disposer d'une représentation graphique. Outre leur intérêt visuel, ces variables peuvent avoir un véritable intérêt pour le modèle. Nous avons en particulier identifié plus haut qu'il semblait y avoir un lien direct entre la latitude et la température maximale des villes.

Ces variables peuvent également permettre de supprimer la variable qualitative *Location*, qui dispose de 49 modalités, ce qui crée 49 variables une fois encodées en OneHot.

### 3.3.2 Amplitude thermique

En regardant le graphique des pairplot entre chaque variable numérique, nous pouvons constater un lien intéressant entre les températures Minimales et Maximales.



**Figure 29 : Nuage de points entre MaxTemp et MinTemp**

La Figure 29 trace en bleu les journées avec un *RainTomorrow* positif, et les positionne sur un graphe aux coordonnées (*MinTemp*, *MaxTemp*). Les points bleus semblent largement positionnés un peu au-dessus de la première bissectrice, ce qui signifie qu'une faible amplitude thermique pourrait être fortement associée au fait qu'il pleuve le lendemain. Ce n'est pour autant pas systématique, car il existe aussi de nombreux points bleus très au-dessus de cette droite. Il est donc peut-être pertinent d'ajouter une nouvelle variable correspondant à l'amplitude thermique, que nous nommerons *AmplitudeTemp*. Il est d'ailleurs intéressant de constater qu'alors que *RainTomorrow* n'était corrélé qu'à moins de 0,19 avec chaque variable de température individuellement, elle l'est à 0,33 avec cette nouvelle variable.

Notons aussi que cette nouvelle variable est corrélée à 0,75 avec *Humidity3pm*, 0,58 avec *Sunshine* et 0,53 avec *Cloud9am*. Là aussi, il s'agit de valeurs supérieures à ce que nous avions avec les variables initiales de température.

Il nous semble donc particulièrement intéressant d'ajouter cette feature.

### 3.3.3 Information climatique

Les différents climats australiens, qui se traduisent par des niveaux de températures et de précipitation très différents selon les lieux nous incitent à créer une variable catégorielle indiquant le type de climat pour chaque lieu.

Une approche simple serait de recherche sur Internet cette information, par exemple sur Wikipédia, ou bien par repérage géographique à partir de cartes de zones climatiques.

Nous allons plutôt opter par une approche par clusterisation, qui nous semble tout à la fois plus pertinente à mettre en œuvre afin de profiter des spécificités de notre jeu de données et intéressante pour automatiser ce processus.

La clusterisation climatique a été réalisée en reprenant la moyenne et l'écart-type de chaque feature, à l'exception des variables du vent. Les coordonnées géographiques ne sont pas utilisées. Il aurait bien sûr été possible de laisser les features du vent, mais nous avons choisi de les enlever, car la clusterisation obtenue nous amenait à pas moins de 10 clusters. De plus, elle distinguait alors des villes géographiquement très proches, telles Perth et PerthAirport, car le vent souffle beaucoup plus sur l'aéroport que sur la ville alors que les précipitations sont très similaires. Malgré tout, nous aurions bien entendu pu utiliser cette clusterisation utilisant le vent, mais nous avons choisi de les retirer pour une meilleure interprétabilité des résultats.

Regardons plus en détail sur la carte de la Figure 31 comment sont répartis les 7 clusters. Outre les deux exemples cités plus haut, nous retrouvons un groupe sur la côte est, un sur des villes côtières du sud, un autre qui est intermédiaire entre les villes côtières et le désert.

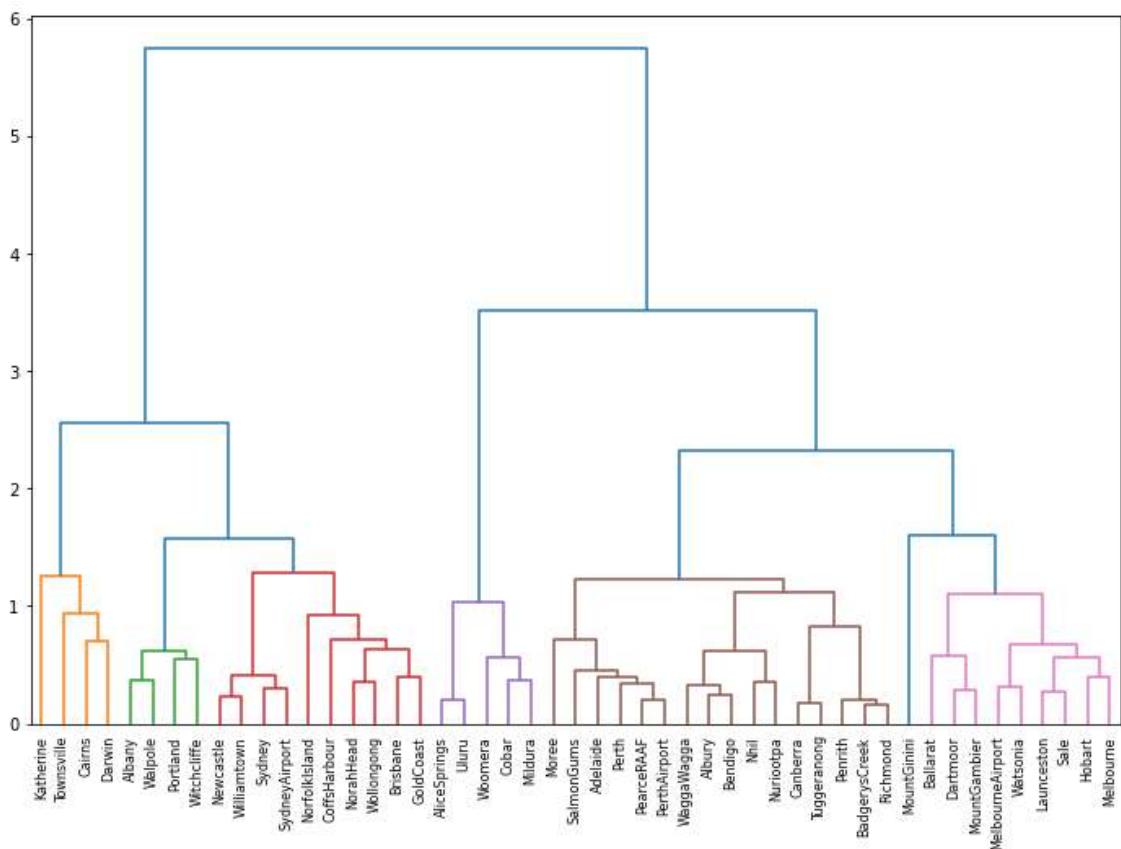
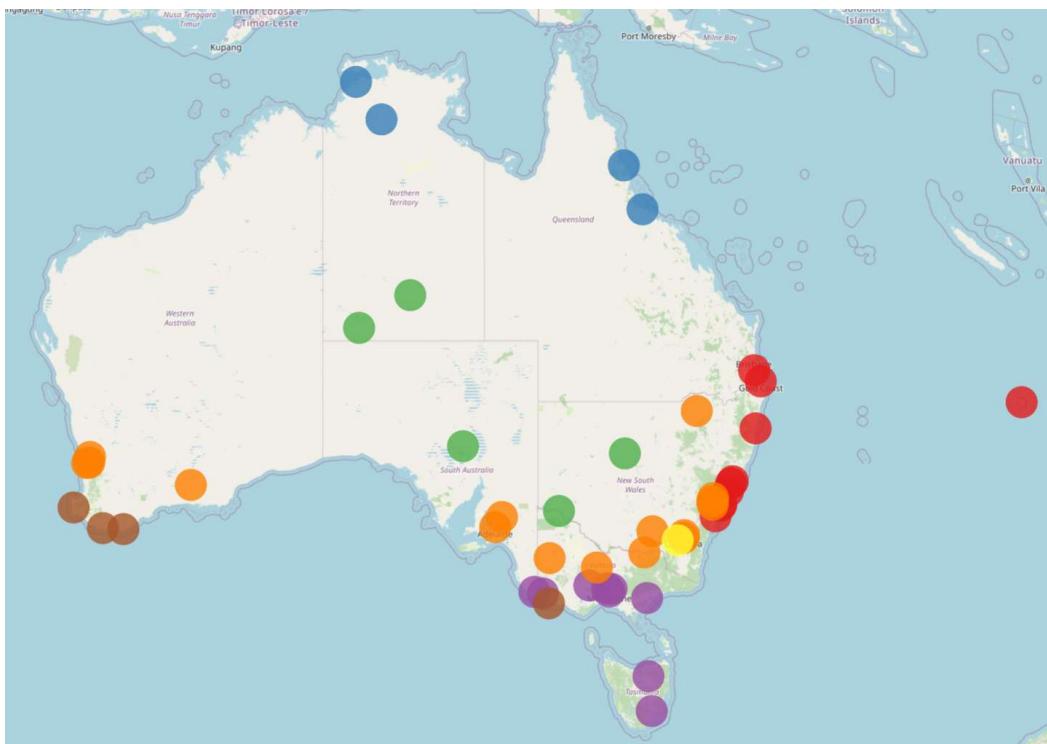


Figure 30 : Résultats de clusterisation par CAH



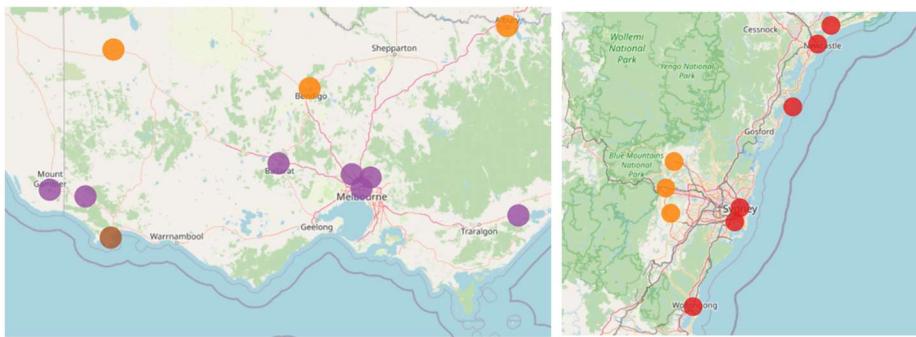
**Figure 31 : La répartition des 7 clusters**

Il saute aux yeux que malgré l'absence de coordonnées géographiques dans les features initiaux – nous les ajouterons ensuite pour nos modèles – il existe une véritable cohérence spatiale de la répartition des *Location*.

Cette cohérence nous permet de nommer chaque zone :

- Climat 0 (rouge) : Côte Est
- Climat 1 (bleu) : Nord
- Climat 2 (vert) : Centre
- Climat 3 (violet) : Sud-Est
- Climat 4 (orange) : Intermédiaire
- Climat 5 (jaune) : Mount Ginini
- Climat 6 (marron) : Côte Sud

Un point marron semble se trouver par erreur au milieu de la zone climatique violette. Il n'en est rien : Portland est bel et bien une ville côtière du sud de l'Australie, ce qui n'est pas le cas des villes alentour, qui sont plus retranchées dans les terres. De même, nous voyons à proximité de Sydney trois point orange pour les stations de Richmond, Penrith et BadgerysCreek : ces trois villes sont à l'intérieur des terres alors que Sydney et les autres villes de la zone 0 sont des villes côtières de l'est.



**Figure 32 : Cas de Portland (en marron) – Cas de Richmond/Penrith/BadgerysCreek (orange)**

Remarquons que Mount Ginini constitue un cluster à elle seule : cette station météorologique est située à 1762m d'altitude, au sommet du mont éponyme. Cette *Location* est bien plus froide que les autres et présente des caractéristiques particulières qui font d'elle un outlier climatique. Nous la conservons dans notre dataset portant sur l'ensemble de l'Australie, mais, dans un objectif de simplification de restitution des travaux, nous l'écartons des modélisations par zone climatique.

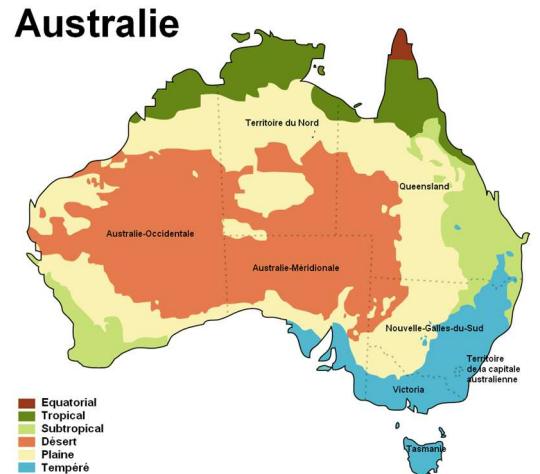
Regardons maintenant une carte représentant les zones climatiques de l'Australie. L'exemple ci-après est issu de la page Wikipédia ‘Climat de l'Australie’.

Zones climatiques australienne (source : Wikipédia : [https://fr.wikipedia.org/wiki/Climat\\_de\\_l'Australie](https://fr.wikipedia.org/wiki/Climat_de_l'Australie) )

Nous retrouvons bien la cohérence des 4 villes nordiques, correspondant à la zone climatique tropicale. Il en va de même pour notre zone centrale qui correspond au désert australien. Les villes des plaines forment également un cluster à part entière. Les zones subtropicales et tempérées correspondent aux autres clusters, mais avec des frontières un peu différentes. En particulier, il est intéressant de constater que notre propre clusterisation a isolé une zone climatique spécifique à la côte est, ce qui semble donc plus fin que la carte climatique de Wikipédia.

Dans un souci de synthèse et de lisibilité, nous ne déclinerons pas systématiquement par la suite les modélisations selon chacune des approches évoquées ci-dessus et nous attacherons plutôt à restituer une représentation assez variée des types d'analyses, en reprenant les résultats les plus intéressants.

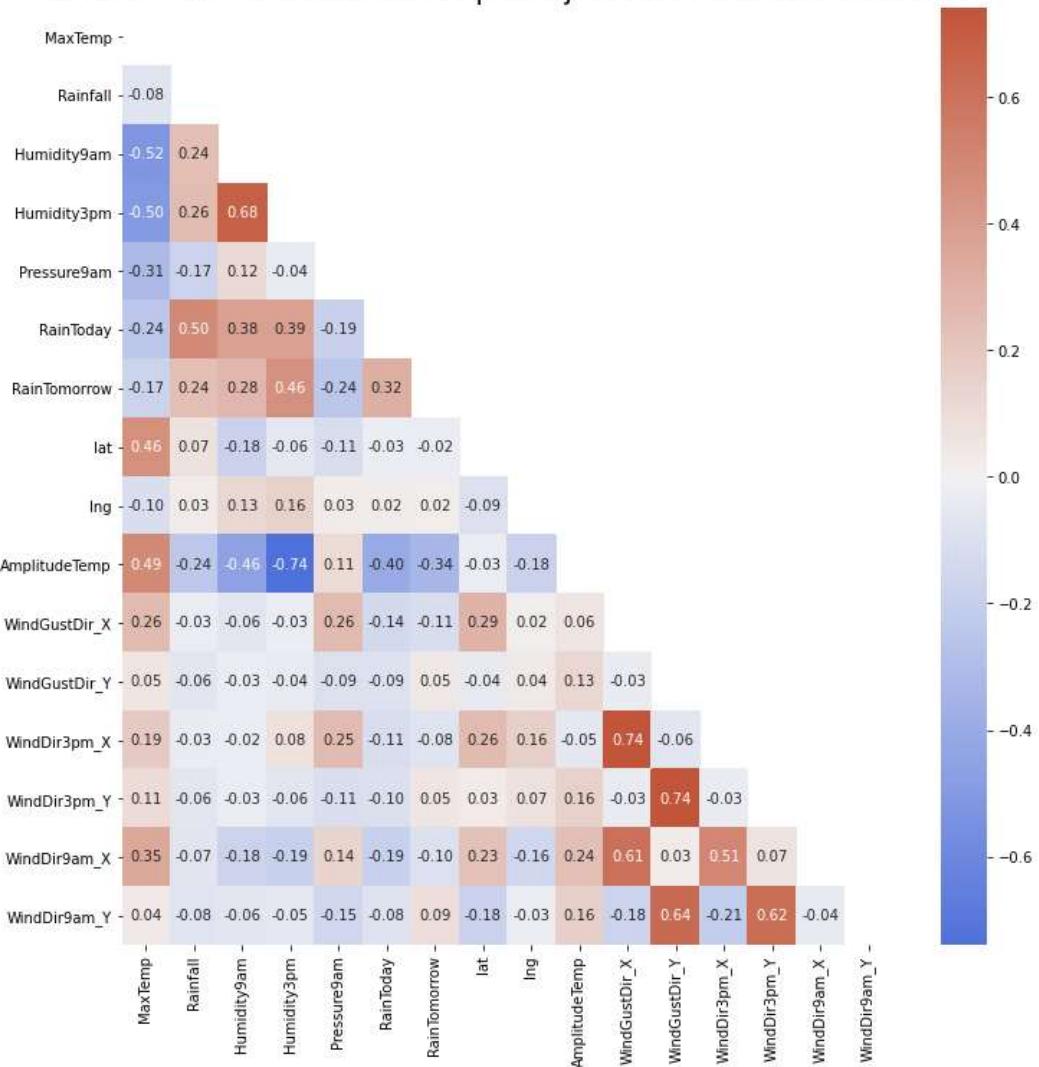
Une force de notre groupe a donc été d'avoir des idées très complémentaires sur les pistes à explorer. Nous nous efforcerons dans ce rapport de conclure sur les intérêts et limites de chaque approche.



### 3.3.4 Corrélations des nouvelles variables

Après ajout des nouvelles features et retrait des variables évoquées, voici la nouvelle matrice de corrélation :

### Corrélations entre variables après ajout des nouvelles variables



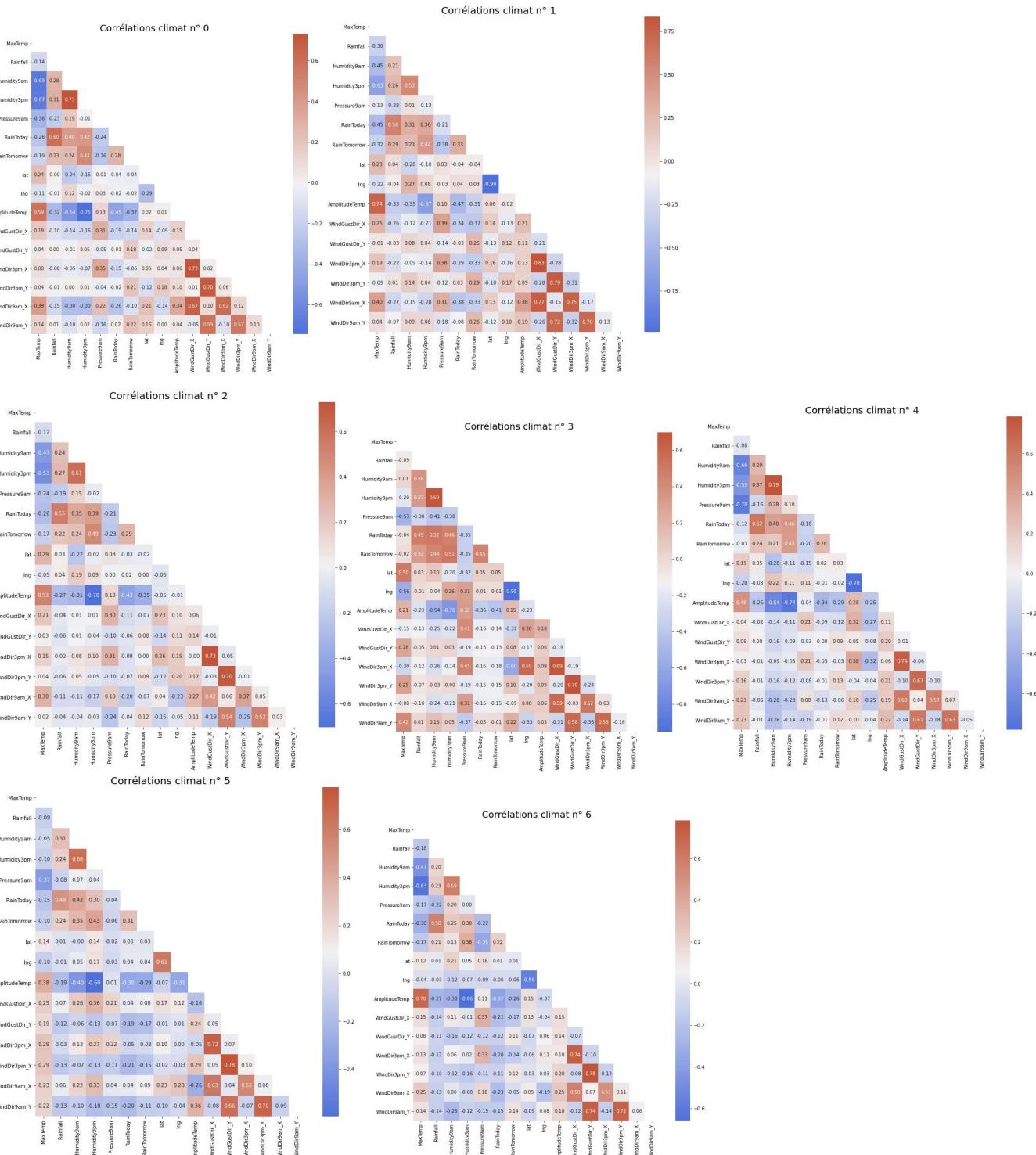
**Figure 33 : Corrélations entre variables après ajout des nouvelles variables**

*RainTomorrow* et *Rainfall* présentent désormais respectivement une corrélation de -0,34 et -0,24 avec la nouvelle variable d'amplitude thermique (*AmplitudeTemp*). Leurs corrélations avec les composantes des vents sont très faibles, de même qu'avec les latitude et longitude des villes.

*MaxTemp* présente des corrélations bien plus élevées avec les nouvelles variables : 0,49 pour l'amplitude thermique, entre 0,19 et 0,35 pour les composantes X du vent.

Notons une forte corrélation (-0,74) entre le taux d'humidité à 15h00 (*Humidity3pm*) l'amplitude thermique.

Par ailleurs, les corrélations varient suivant les climats que nous avons obtenus par clusterisation. Avec le climat n°1, les composantes X et Y des vents sont corrélées entre 0,25 et 0,37 en valeurs absolues. Dans le climat n°0, c'est la composante X des vents qui est corrélée avec *RainTomorrow*.



**Figure 34 : La corrélation entre les variables par zone de climat**

### 3.3.5 Normalisation et standardisation

Nous avons vu que les ordres de grandeur des variables étaient très différents. Il sera donc tout à fait indispensable de standardiser et normaliser les données servant à prédire la variable cible. Notez que cette opération était déjà nécessaire pour la KNN imputation faite plus haut.

## 4 Conclusions sur le preprocessing

Le jeu de données dispose d'informations très variées, avec des taux de données non renseignées assez hétérogènes selon les variables, les dates et les lieux.

La principale difficulté a consisté à analyser les données selon plusieurs axes, notamment temporel et géographique, et à trouver des représentations qui permettent de rendre compte de l'état des données pour les différentes variables selon des spécificités locales.

Des approches assez diverses ont ainsi été mises en œuvre pour proposer des visualisations variées, en prenant également en compte le type de représentation visuelles utilisées par des météorologues (diagramme climatique et représentation polaire des directions des vents), la prise en compte des habitudes professionnelles des interlocuteurs des data scientists étant un élément important dans la bonne communication et compréhension des résultats.

Nos analyses nous ont permis d'arriver à une conclusion sur une méthodologie de gestion des valeurs manquantes, à savoir :

- Télécharger des données complémentaires pour des variables et des lieux ciblés
- Remplacer par des 0 les NA de la variable *Rainfall* d'une plage de date précédent immédiatement une date pour laquelle *Rainfall* est inférieure à 1mm, et mettre à False *RainToday* sur cette même plage et à False *RainTomorrow* la plage un jour plus tôt
- Supprimer les lignes pour lesquelles la variable cible *RainTomorrow* reste non renseignée
- Enlever les lignes sur lesquelles plus de 50% des variables sont non renseignées
- Imputer les NA restants grâce à KNN Imputer

Nous avons également pu enrichir le jeu de données par des variables complémentaires (latitude et longitude des villes, zones climatiques, amplitude thermique, composantes directionnelles du vent) afin d'aider les modèles de prédiction.

Nos données sont désormais nettoyées, enrichies, et prêtes à alimenter nos modèles de prédiction.

## 5 Introduction - Modélisation

### 5.1 Méthodologie

L'objectif principal est ici de prédire le mieux possible la variable '*RainTomorrow*', qui indique s'il pleuvra ou non le lendemain d'une observation donnée, pour l'une des 49 stations météo (*Location*) du jeu de données. Nous utiliserons pour cela les données issues du feature engineering que nous avons effectué lors de la première partie du projet, et allons appliquer des modèles de classification tels que Logistic Regression, Decision Tree, Random Forest ou XGBoost, mais également des modèles de Deep Learning, avec des Réseaux Neuronaux Denses (DNN) et des Réseaux Neuronaux Récursifs (RNN).

Les hyperparamètres de chaque modèle seront optimisés via des tests manuels, des GridSearch, mais aussi à l'aide de la bibliothèque Hyperopt, en cherchant à maximiser diverses métriques telles que l'accuracy, la précision, le recall, le score F1 et le ROC AUC. Nous expliquerons les enjeux portant sur le choix d'une métrique adaptée.

Nous avons également inclus des éléments visuels tels que des matrices de confusion, des graphiques de courbes ROC AUC, et des analyses des caractéristiques les plus importantes pour chaque modèle. Ces éléments permettent une compréhension approfondie de la performance de chaque modèle.

L'utilisation de SHAP (Shapley Additive exPlanations) pour évaluer la contribution de chaque variable explicative a été cruciale.

Dans les parties suivantes, nous tenterons de prédire non seulement s'il pleuvra le lendemain, mais également de voir s'il est possible de prévoir la pluie plusieurs jours à l'avance. Nous essaierons également de prédire d'autres variables, comme la température, avec une problématique de régression et de séries temporelles.

## 5.2 Approches

La phase de feature engineering nous a donné plusieurs pistes sur l'ajout de variables. De premières approches rapides de modélisation n'avaient pas mis en évidence de changements importants de performances selon variables ajoutées. Pour autant, nous ne pouvions à ce stade pas encore arbitrer sur les variables à conserver ou ajouter. Nous avons donc fait le choix d'effectuer les travaux de modélisation au sein de notre groupe de travail non pas avec les mêmes variables explicatives pour chaque personne, mais de nous laisser à chacun la liberté d'explorer différentes pistes afin de pouvoir dans un second temps mieux identifier l'impact des différentes approches. Ce choix d'explorer des pistes différentes explique que, selon les captures d'écran, les variables explicatives peuvent diverger, tout particulièrement pour le vent, ou nous avons fait d'une part un encodage OneHot à partir d'une variable qualitative, mais également d'autre part un encodage trigonométrique pour réduire le nombre de features et transformer ces variables qualitatives en variables quantitatives.

Nous avons également abordé nos problématiques de modélisation selon trois niveaux de finesse :

- Un niveau macro, avec des modèles portant sur l'ensemble des données australiennes du jeu de données
- Un niveau micro, où nous générerons des modèles spécifiques pour chaque *Location*
- Un niveau intermédiaire, dans lequel nous aurons clusterisé l'Australie en plusieurs zones climatiques

Le niveau macro permettra de pouvoir réaliser toutes nos prédictions avec un seul modèle.

Le niveau micro rendra plus complexe l'analyse des performances puisqu'il y aura 49 modèles, mais permettra potentiellement une meilleure adaptation aux spécificités de chaque *Location*.

Le niveau intermédiaire sera un compromis entre les deux approches, puisque nous aurons 6 modèles, qui seront associés aux *Location* en fonction des spécificités climatiques communes détectées lors de la clusterisation.

# 6 Prédiction de la variable *RainTomorrow*

## 6.1 Rappel sur déséquilibre

Il est primordial de garder à l'esprit que les deux classes de *RainTomorrow* sont déséquilibrées, puisque, sur l'ensemble du dataset, seules 22,4% des observations sont positives (= « il pleuvra demain ») et donc 77,6% sont négatives (= « il ne pleuvra pas demain »).

Ce constat nous a amené dans un premier temps à effectuer un rééquilibrage des données par oversampling. Cependant, le Tableau 5 montre que les résultats obtenus n'ont pas indiqué de différence significative sur les performances obtenues. Nous avons donc finalement conservé les données sans oversampling.

Comparaison des modèles selon l'oversampling					
Modèle	accuracy	recall	precision	f1	auc
sans oversampling	0.8642	0.5558	0.7642	0.6436	0.9003
RandomOverSampler	0.8224	0.7997	0.5694	0.6652	0.8988
SMOTE	0.8501	0.6361	0.6684	0.6518	0.8895

Tableau 5 : Comparaison des modèles selon l'oversampling

Notons que ce déséquilibre implique qu'une prédiction systématiquement négative entraînerait une accuracy de 0,776. Par conséquent, nous attendrons de nos modèles qu'ils proposent une accuracy supérieure à ce chiffre. La question de la pertinence de l'accuracy comme critère peut donc se poser.

Comme nous l'avons vu précédemment, le taux de journée pluvieuses diffère de façon significative selon les lieux. Ce ratio de 0,776 vaut donc pour uniquement pour les modèles entraînés sur l'ensemble de l'Australie.

## 6.2 Métriques

Si l'accuracy permet une compréhension simple des résultats, elle n'est pas nécessairement pertinente pour notre jeu de données, du fait du déséquilibre de *RainTomorrow*. Nous la conserverons comme indicateur globale de la qualité, tout en gardant à l'esprit qu'une accuracy inférieure à 0,776 sera mauvaise.

Un faible recall correspond pour *RainTomorrow* à un nombre élevé de jours de prédictions d'absence de pluie le lendemain alors qu'il pleuvra (faux négatifs). Un modèle naïf prédisant qu'il ne pleuvra jamais aura comme vu précédemment une accuracy de 0,776, mais un recall de 0. C'est une métrique qu'il sera intéressant d'optimiser.

Une faible précision de *RainTomorrow* indiquerait que nous nous trompons souvent lorsque nous prédisons qu'il pleuvra le lendemain (faux positifs).

L'AUC-ROC, que nous noterons simplement AUC par la suite, est une métrique particulièrement intéressante dans le cadre de classe binaire déséquilibrée.

Le choix de la métrique déprendrait normalement de l'objectif à atteindre. Par exemple, pour anticiper des annulations de réservations, un hôtel touristique pourrait vouloir savoir s'il y a un risque de pluie, quitte à avoir beaucoup de faux positifs, alors qu'un agriculteur doit être certain qu'il pleuvra, même si nous devons pour cela prédire parfois à tort qu'il pleuvra.

Dans le premier cas, nous optimiserions le recall pour avoir un modèle très sensible. Dans le second cas, nous optimiserions plutôt la précision pour avoir un modèle avec une grande spécificité.

Voyons maintenant en pratique les différences obtenues selon les métriques optimisées pour chaque modèle.

## 6.3 Résultats de la classification par approches « classiques » via scikit-learn

### 6.3.1 Modèles étudiés

Nous regardons ici des modèles générés avec une approche de machine learning qui n'est pas du deep learning : nous exploitons simplement la variable *RainTomorrow* de chaque observation, indépendamment de la date de l'observation.

### 6.3.2 Optimisation de métriques

Les hyperparamètres optimaux ont été déterminés en optimisant plusieurs métriques.

Attardons-nous sur une problématique incontournable dans le machine learning, à savoir l'overfitting. Une attention permanente a été apporté dans tous les modèles sur ce point, d'une part en nous assurant que l'accuracy de l'ensemble d'entraînement était proche de l'accuracy de l'ensemble de test (inférieure à 2%), mais également via une méthodologie plus complexe via la librairie HyperOpt. Nous avons utilisé la validation croisée *StratifiedKFold(n\_splits=5, random\_state=42, shuffle=True)*. Pour le modèle XGBoost en particulier, nous avons ajouté l'option *early\_stopping\_rounds: int = 50* qui est recommandée pour éviter le problème d'overfitting pour ce modèle. En vérifiant les métriques calculées sur l'ensemble d'entraînement et l'ensemble de test, nous n'avons pas identifié de problème d'overfitting.

### 6.3.2.1 Maximisation de l'accuracy

Les résultats présentés dans le Tableau 6 montrent les performances de quatre modèles différents en termes de cinq métriques d'évaluation, avec l'optimisation basée sur l'accuracy.

	accuracy	recall	precision	f1	auc
LogisticRegression	0.8461	0.5004	0.7278	0.593	0.8694
TreeDecision	0.8389	0.4841	0.7049	0.574	0.8456
RandomForest	0.8465	0.4761	0.7477	0.5818	0.8653
<b>XGBoost</b>	<b>0.8526</b>	0.5407	0.7315	0.6218	0.8844

Tableau 6 : Scores des modèles ayant meilleur accuracy

- Accuracy : XGBoost a la meilleure accuracy parmi les quatre modèles, avec une valeur de 0.8526
- Recall : XGBoost a également le recall le plus élevé, ce qui suggère qu'il a une capacité supérieure à identifier les exemples positifs par rapport aux autres modèles.
- Precision : Random Forest a la precision la plus élevée, indiquant qu'il a moins de faux positifs par rapport aux autres modèles.
- F1-score : XGBoost a le F1-score le plus élevé, ce qui est une combinaison équilibrée de recall et precision.
- AUC : XGBoost a également la meilleure AUC, indiquant de bonnes performances globales pour la classification binaire.

Si l'objectif principal est d'optimiser l'accuracy, le modèle XGBoost semble être le meilleur choix

On peut observer dans la Figure 35e que la ROC courbe du modèle XGBoost est au-dessus des ROC courbes des autres modèles. Cela suggère que le modèle XGBoost a une meilleure capacité à maintenir un taux élevé de vrais positifs tout en limitant le taux de faux positifs, même à différents seuils de classification.

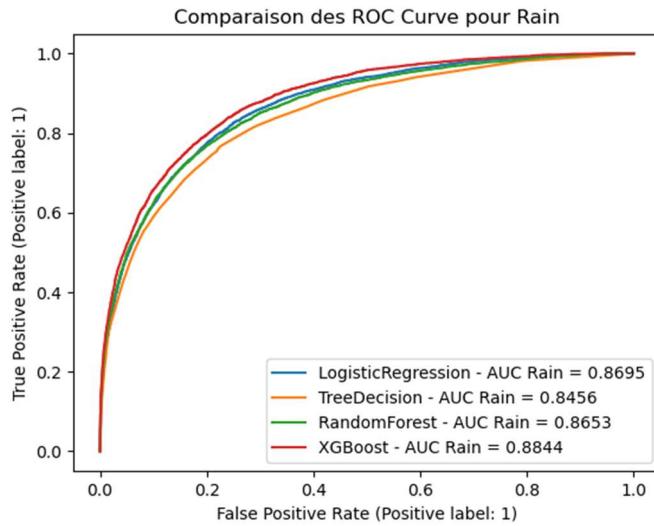


Figure 35 : Les ROC courbes des 4 modèles ayant meilleur accuracy

#### 6.3.2.2 Maximisation de la précision

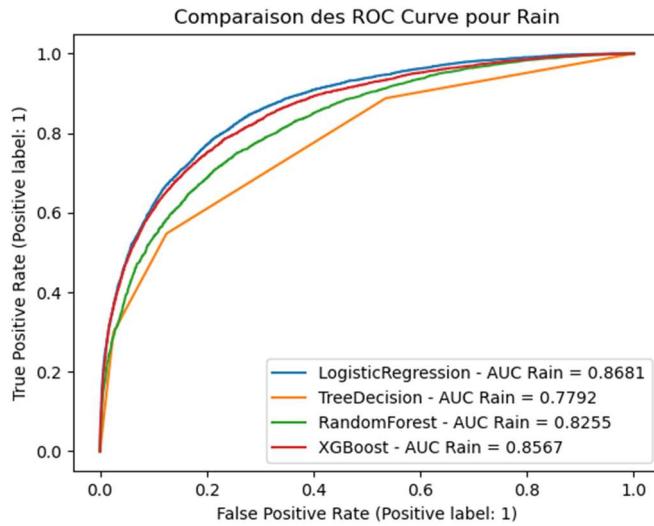
Les résultats présentés dans le Tableau 7 montrent les performances de quatre modèles différents en termes de cinq métriques d'évaluation, avec l'optimisation basée sur la précision.

Scores des modèles ayant meilleur precision					
	accuracy	recall	precision	f1	auc
LogisticRegression	0.8455	0.4894	0.7327	0.5869	0.8681
TreeDecision	0.8233	0.2987	0.7746	0.4311	0.7792
<b>RandomForest</b>	0.783	0.0367	<b>0.8897</b>	0.0705	0.8315
XGBoost	0.841	0.4144	0.7698	0.5388	0.8567

Tableau 7 : Scores des modèles ayant meilleure précision

- Précision : La précision mesure la proportion d'exemples positifs parmi ceux que le modèle a identifiés comme positifs. Dans ce cas, Random Forest a la précision la plus élevée (0.8897), indiquant qu'il a une capacité à minimiser les faux positifs par rapport aux autres modèles.
- Accuracy : La précision seule ne donne pas une image complète de la performance du modèle, car elle ne prend pas en compte les faux négatifs. En termes d'accuracy, Linear Regression a la valeur la plus élevée (0.8455).
- Recall : Random Forest a le recall le plus faible (0.0367), ce qui signifie qu'il identifie très peu d'exemples positifs parmi tous les exemples réellement positifs.
- F1-score : XGBoost a un F1-score relativement équilibré, combinant recall et precision de manière équilibrée.
- AUC : Random Forest a la plus haute AUC dans ce cas (0.8315).

Si la précision est la métrique la plus importante, Random Forest pourrait être le choix préféré en se basant sur ces résultats spécifiques.



**Figure 36: Les ROC courbes des 4 modèles ayant meilleure précision**

#### 6.3.2.3 Maximisation du recall

Les résultats présentés dans le Tableau 8 montrent les performances de quatre modèles différents en termes de cinq métriques d'évaluation, avec l'optimisation basée sur le recall.

Scores des modèles ayant meilleur recall					
	accuracy	recall	precision	f1	auc
LogisticRegression	0.7366	<b>0.8464</b>	0.4531	0.5902	0.864
TreeDecision	0.8313	0.4944	0.6669	0.5679	0.818
RandomForest	0.8397	0.5062	0.6956	0.586	0.8469
XGBoost	0.8292	0.5547	0.6365	0.5928	0.8481

**Tableau 8: Scores des modèles ayant meilleur recall**

- Recall : Recall mesure la proportion d'exemples positifs réellement identifiés par le modèle. Linear Regression a le recall le plus élevé (0.8464), indiquant qu'il a la meilleure capacité parmi les modèles à capturer la majorité des exemples positifs. Cependant, il est important de noter que cette performance peut être associée à une baisse de précision.
- Precision : Linear Regression a la precision la plus faible (0.4531), indiquant qu'il a tendance à identifier un grand nombre de faux positifs.
- Accuracy : Random Forest a la valeur la plus élevée en termes d'accuracy (0.8397), mais cela peut être dû à une balance entre les performances en termes de faux positifs et de faux négatifs.
- F1-score : Le F1-score de Linear Regression est relativement équilibré, étant donné son recall élevé et sa precision basse.
- AUC : AUC mesure la capacité du modèle à discriminer entre les classes positives et négatives. XGBoost a la plus haute AUC dans ce cas (0.8481).

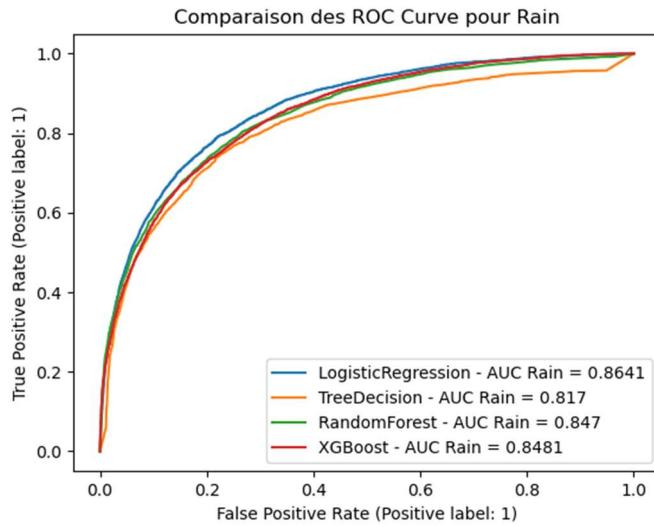


Figure 37: : Les ROC courbes des 4 modèles ayant meilleur recall

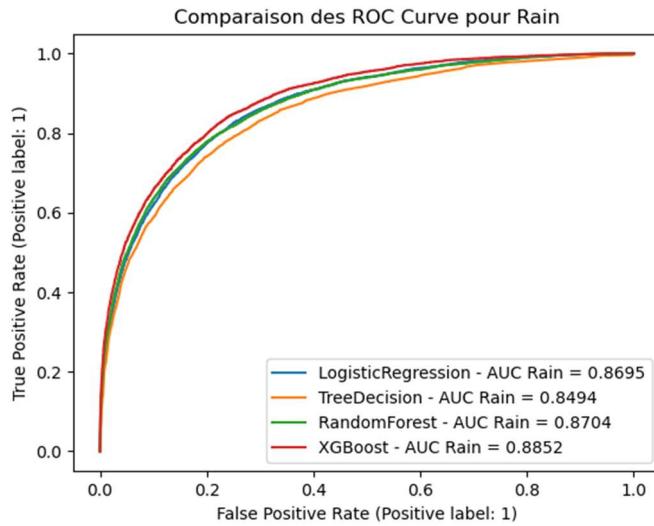
#### 6.3.2.4 Maximisation du F1-score

Les résultats présentés dans le Tableau 9 montrent les performances de quatre modèles différents en termes de cinq métriques d'évaluation, avec l'optimisation basée sur le F1-score

Scores des modèles ayant meilleur f1					
	accuracy	recall	precision	f1	auc
LogisticRegression	0.846	0.5002	0.7277	0.5929	0.8695
TreeDecision	0.8409	0.4883	0.7114	0.5791	0.8498
RandomForest	0.8494	0.5049	0.7409	0.6006	0.8699
<b>XGBoost</b>	<b>0.8554</b>	<b>0.5475</b>	<b>0.7397</b>	<b>0.6292</b>	<b>0.8852</b>

Tableau 9: Scores des modèles ayant meilleur F1-score

- F1-score : Le modèle XGBoost affiche le F1-score le plus élevé (0.6292) parmi tous les modèles. Le F1-score est une métrique qui prend en compte à la fois la precision et le recall. Cela suggère que XGBoost atteint un équilibre optimal entre l'identification des vrais positifs (recall) et la minimisation des faux positifs (precision).
- Accuracy : XGBoost a également une accuracy élevée (0.8554), indiquant une classification correcte d'une grande proportion des exemples.
- Recall et Precision : XGBoost a des valeurs de recall et de precision compétitives par rapport aux autres modèles, ce qui renforce l'idée d'un équilibre entre la sensibilité aux vrais positifs et la limitation des faux positifs.
- AUC : XGBoost présente également la plus haute AUC (0.8852), ce qui confirme sa capacité à bien discriminer entre les classes positives et négatives.



**Figure 38: Les ROC courbes des 4 modèles ayant meilleur F1-score**

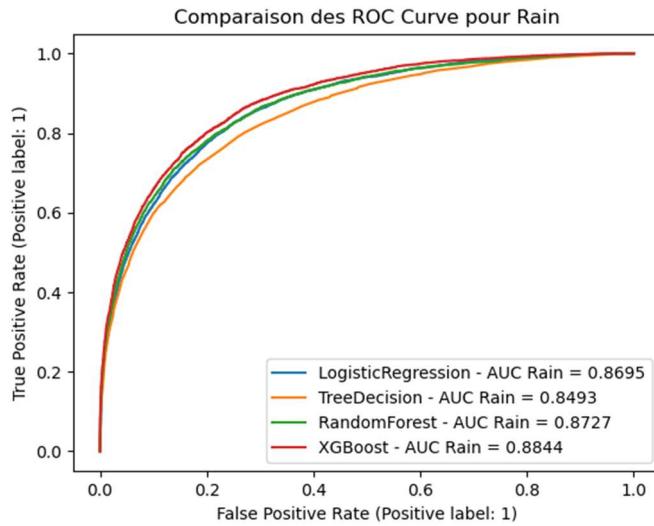
#### 6.3.2.5 Maximisation de l'AUC

Les résultats présentés dans le Tableau 10 montrent les performances de quatre modèles différents en termes de cinq métriques d'évaluation, avec l'optimisation basée sur l'AUC.

Scores des modèles ayant meilleur auc					
	accuracy	recall	precision	f1	auc
LogisticRegression	0.846	0.5004	0.7273	0.5929	0.8695
TreeDecision	0.8399	0.4478	0.7341	0.5563	0.8493
RandomForest	0.8497	0.498	0.7474	0.5978	0.8716
<b>XGBoost</b>	<b>0.8538</b>	<b>0.5358</b>	<b>0.74</b>	<b>0.6216</b>	<b>0.8844</b>

**Tableau 10: Scores des modèles ayant meilleur AUC**

- AUC : L'AUC (Area Under the Curve) est souvent utilisée comme métrique pour évaluer les modèles de classification binaire. Dans ce cas, XGBoost a la valeur d'AUC la plus élevée (0.8844), ce qui indique une performance globale solide pour la classification binaire. Cela suggère que le modèle XGBoost a une bonne capacité à discriminer entre les classes positives et négatives.
- Accuracy : XGBoost a également la meilleure accuracy (0.8538), ce qui signifie qu'il a correctement classé une grande proportion des exemples.
- Recall : XGBoost a le recall le plus élevé, indiquant qu'il a une capacité supérieure à identifier les exemples positifs par rapport aux autres modèles.
- Precision : Random Forest a la precision la plus élevée, ce qui suggère qu'il a moins de faux positifs par rapport aux autres modèles.
- F1-score : XGBoost a le F1-score le plus élevé, ce qui est une combinaison équilibrée de recall et precision



**Figure 39: Les ROC courbes des 4 modèles ayant meilleur AUC**

#### 6.3.2.6 Comparaison avec des modélisations par lieu et par zone climatique

Les résultats dans les Sections 49-53 suggèrent que XGBoost est le modèle qui offre la meilleure performance globale, en particulier en termes d'AUC, accuracy, recall, precision et F1-score. Ces modélisations ont été réalisées avec l'ensemble des données du dataset. Comparons maintenant les performances d'un XGBoost entraîné en optimisant l'AUC-ROC sur l'ensemble du jeu de données avec des modélisations ciblant chaque station météo d'une part (filtrage par la variable *Location*), et chaque zone climatique d'autre part (filtrage via la variable Climat issue de la clusterisation). Le Tableau 11 indiquera donc les performances moyennes de chaque approche ainsi que le modèle offrant les scores les plus intéressants pour une *Location* donnée et pour une zone climatique donnée.

Comparaison des XGBoost selon le périmètre de modélisation					
	accuracy	recall	precision	f1	auc
Macro	0.8642	0.5558	0.7642	0.6436	0.9003
Micro (moyenne)	0,8560	0,4563	0,7757	0,5690	0,8716
Micro (meilleur : PearceRAAF)	0,9167	0,5647	0,8727	0,6857	0,9584
Climat (moyenne)	0,8646	0,5592	0,7667	0,6442	0,8993
Climat (meilleur : Centre)	0,9378	0,4743	0,7742	0,5882	0,9379

**Tableau 11 : Comparaison des XGBoost selon le périmètre de modélisation**

A titre indicatif, les hyperparamètres du modèle global sont un learning rate de 0,23, une profondeur max de 6 et 50 estimateurs. Pour les modèles locaux le learning rate est 0,2, la profondeur max 3 avec 4 estimateurs. Enfin, les modèles climatiques ont également un learning rate de 0,2, mais une profondeur max de 4 et comportent 30 estimateurs.

A première vue, les performances moyennes des modèles par climat semblent très proches du modèle global. Les performances moyennes des modèles locaux ont l'air d'être légèrement inférieures. Toutefois, la réalité est un peu plus complexe. On voit notamment que le modèle de climat le plus performant (recouvrant le centre de l'Australie) performe nettement mieux que le modèle global. De même, le meilleur modèle local, PearceRAAF, est celui qui présente les meilleures performances de tous les modèles confondus au regard de l'AUC.

Afin que les données soient comparables, regardons les performances des trois niveaux de modélisation appliqués sur trois *Location*, représentées dans le Tableau 12.

Comparaison des modèles pour PearceRAAF					
	accuracy	recall	precision	f1	auc
Micro	0,9167	0,5647	0,8727	0,6857	<b>0,9584</b>
Global	0,9053	0,6526	0,7470	0,6966	0,9486
Climat – 4, Intermédiaire	0,9170	0,6818	0,8108	0,7407	0,9463

Comparaison des modèles pour NorfolkIsland					
	accuracy	recall	precision	f1	auc
Micro	0,7555	0,3352	0,7262	0,4586	0,7655
Global	0,7883	0,4837	0,7355	0,5836	<b>0,8094</b>
Climat – 0, Côte Est	0,7783	0,4776	0,7619	0,5872	0,8037

Comparaison des modèles pour Penrith					
	accuracy	recall	precision	f1	auc
Micro	0,8593	0,3879	0,8036	0,5233	0,8284
Global	0,8761	0,5517	0,7901	0,6497	0,8677
Climat – 4, Intermédiaire	0,8693	0,5468	0,8352	0,6609	<b>0,8922</b>

**Tableau 12 : Les performances des trois niveaux de modélisation pour PearceRAAF, NorfolkIsland et Penrith**

Précisons en premier lieu que le taux de journée pluvieuses est très différent pour ces trois lieux et est à rapprocher directement de la valeur de l'accuracy : 18% pour PearceRAAF, 31% pour NorfolkIsland et 20% pour Penrith, soit, avec un modèle prédisant systématiquement qu'il ne pleuvra pas, des accuracy « naïves » respectivement de 0,82, 0,69 et 0,80.

Les résultats dans Tableau 12 illustrent qu'aucune des trois approches, globale, locale ou climatique, n'est meilleure de façon systématique, quelle que soit la métrique. Si nous nous basons sur l'AUC, c'est pour PearceRAAF un modèle entraîné spécifiquement sur les données de la station qui offre les meilleures performances. Pour NorfolkIsland, le modèle global performe mieux que les deux autres. Enfin, on préférera une modélisation par climat pour Penrith.

Ces trois niveaux de granularité semblent donc complémentaires.

De façon plus détaillée, observons, dans la Figure 40, l'accuracy pour chacun des 49 lieux en comparant :

- Un modèle global, entraîné sur l'ensemble du dataset (« ML Global »)
- Un modèle local, entraîné spécifiquement sur les données du lieu concerné (« ML Individuel »)
- Un modèle naïf se contentant de prédire qu'il ne pleuvra jamais, qui nous permettra de relativiser les scores obtenus par les deux premiers modèles (« Pred Bas »)

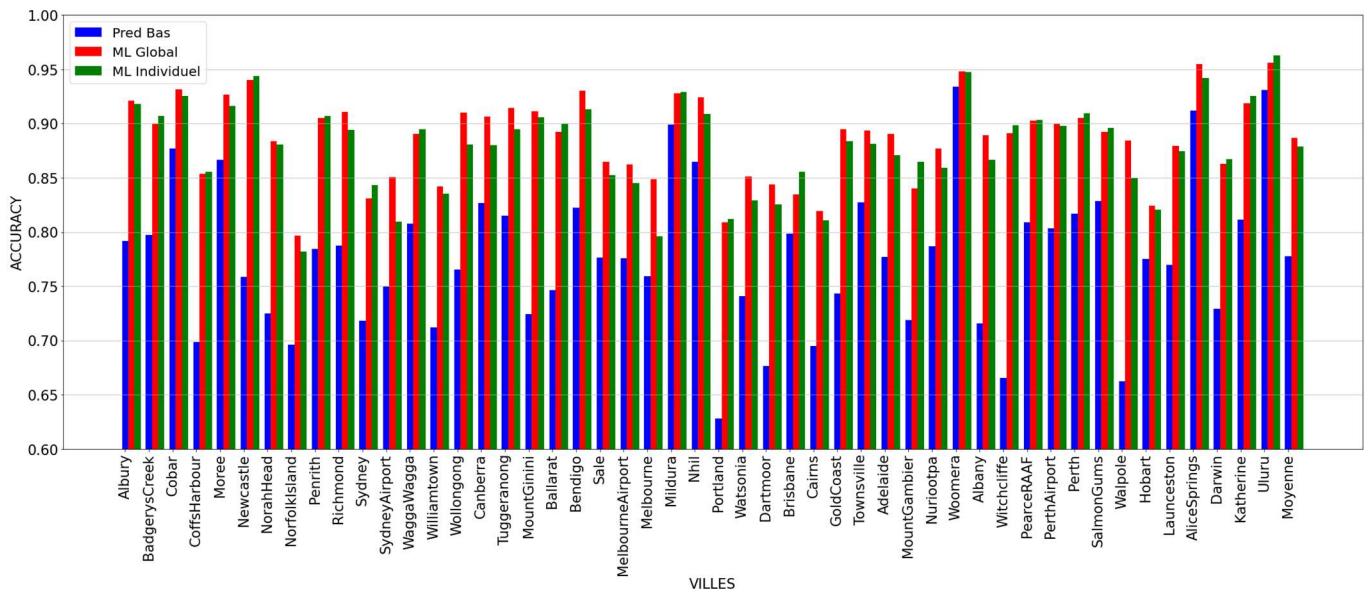


Figure 40 : L'accuracy des trois modèles par *Location*

La moyenne de l'ensemble des *Location* donne un score de 0,879 pour les ML Individuels, 0,886 pour les ML Global et 0,777 pour le modèle « Pred Bas ». L'approche macro fournit donc un score légèrement meilleur en moyenne. La différence est cependant faible, et nous pouvons voir sur le graphique ci-dessus que même en observant chaque *Location*, il n'y a que peu de différences de performances.

### 6.3.3 Impact du feature enginerring

Maintenant que nous avons pu déterminer que le XGBoost performe mieux que les autres modèles de machine learning et que nous avons identifié des hyperparamètres adaptés, regardons quel impact ont eu les travaux de feature engineering.

Le modèle dit « avec features d'origine » effectue des transformations élémentaires sur le dataset *WeatherAUS.csv*, tel le remplacement des ‘Yes’ par des True pour les variables binaires ainsi qu'un encodage OneHot des variables catégorielles. Les valeurs manquantes (NA) sont traitées par suppression pure et simple, sans substitution. Les quatre variables ayant un taux très important de NA (*Sunshine* : 48%, *Evaporation* : 43%, *Cloud3pm* : 41%, *Cloud9am* : 38%) sont supprimées, faute de quoi un *dropna* global entraîne la suppression de plus de la moitié du dataset.

Le modèle dit « avec nouvelles features » est issu des travaux menés dans la première partie du projet. Les valeurs nulles sont gérées par KNN Imputation, les variables catégorielles de direction du vent sont remplacées par des variables quantitatives trigonométriques, les *Location* sont remplacées par la latitude et la longitude, une variable Climat indique le résultat de la clusterisation par zone climatique, l'amplitude thermique est ajoutée (=TempMax-TempMin), et afin deux variables temporelles respectivement égales au cosinus de  $2\pi \times (\text{numéro du jour dans l'année}/365)$  et cosinus  $4\pi \times (\text{numéro du jour dans l'année}/365)$ .

Nous comparons dans le Tableau 13 les performances d'un XGBoost avec les mêmes hyperparamètres (learning rate 0,23, max\_depth 6, n\_estimators 50, obtenu par optimisation de l'AUC-ROC) sur ces deux dataset portant sur toute l'Australie dans les deux cas :

### Performances d'un XGBoost en fonction du feature engineering

	accuracy	recall	precision	f1	auc
Avec nouvelles features	0.8642	0.5558	0.7642	0.6436	0.9003
Avec features d'origine	0.8566	0.5299	0.7414	0.618	0.8847

Tableau 13 : Performances d'un XGBoost en fonction du feature engineering

Certes, le modèle bénéficiant des nouvelles features propose de meilleures performances quelle que soit la métrique observée, mais le gain est très faible, tout particulièrement au regard du temps passé pour effectuer le feature engineering.

Nous avons également voulu enrichir les variables pour chaque journée en reprenant également les valeurs de la veille. Ainsi, pour chaque observation, pour prédire *RainTomorrow*, nous disposons des relevés pour le jour J mais également J-1. En plus des trois modélisations décrites plus haut, nous avons donc ici 49 modèles « ML Veille-ind » entraînés avec ces nouvelles features.

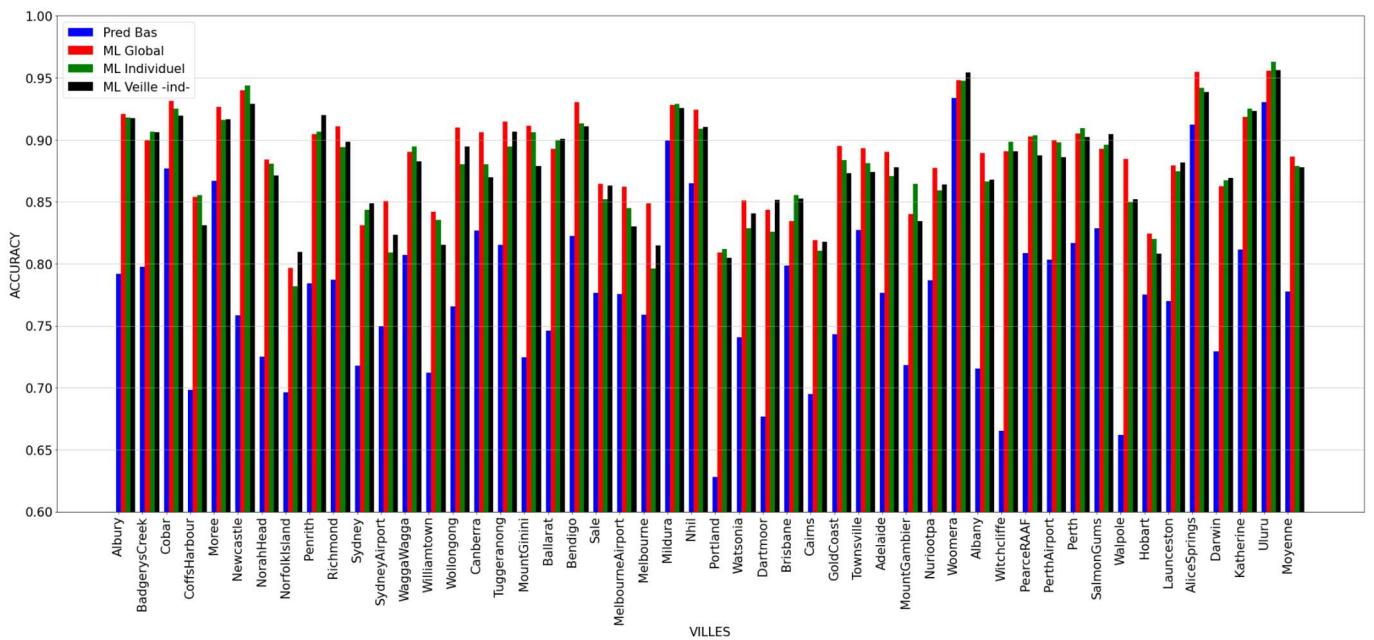


Figure 41 : L'accuracy des quatre modèles, par *Location*, entraînés avec les nouvelles features

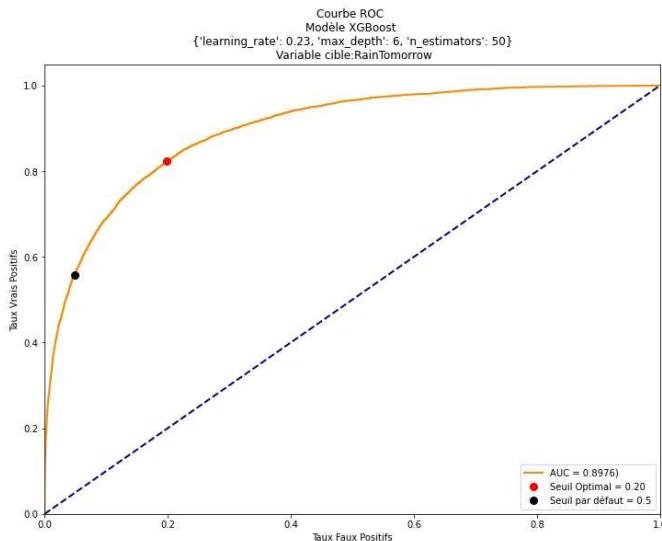
On constate qu'avec les données de la veille, le score du modèle ne s'améliore pas par rapport à un modèle individuel (0.886 contre 0.877). Là aussi, nous obtenons que les différences soient faibles.

La phase de feature engineering, bien que particulièrement chronophage, ne semble donc pas être déterminante dans l'échelle de qualité fournie par les modèles.

#### 6.3.4 Seuil de probabilité

Importance des features sur un modèle XGBoost optimisant l'AUC-ROC.

Jusqu'ici, nous avons exploité directement les prédictions de nos différents modèles XGBoost. Nous allons maintenant profiter du fait que ce modèle nous permet de connaître la probabilité de chaque prédition pour l'affiner.



**Figure 42 : Courbe ROC issue d'un XGBoost entraîné sur toute l'Australie**

Nous l'avons vu dans les tableaux de métriques précédent : un point faible est les performances sur le recall, qui indique globalement que, lorsqu'il pleut, nous ne sommes capables environ qu'une fois sur deux de prévoir la précipitation. Rappelons quand dans le un modèle probabiliste de classification, le seuil est par défaut de 0,5. Ce seuil est représenté par le point noir dans la Figure 42. Quelques remarques :

- Nous confirmons immédiatement visuellement que le taux de vrai positif n'est que d'environ la moitié
- Nous voyons que le taux de faux positif est en revanche particulièrement faible
- Nous disposons donc de modèles ayant une spécificité plutôt bonne, mais une sensibilité assez mauvaise

En d'autres termes, nous sommes assez bons pour garantir qu'il ne pleuvra pas mais assez mauvais pour prédire de façon assez fiable qu'il pleuvra.

Ce constat ne permet pas de juger de la qualité du modèle : pour cela, il faudrait connaître l'objectif du client sur ce modèle. A-t-il besoin d'une garantie absolue d'absence de pluie ? Préfèrerait-il savoir avec certitude lorsqu'il pleuvra ? Souhaite-t-il un compris entre les deux approches ? Encore plus pragmatiquement, quel serait le coût d'une erreur, et donc quel serait le taux de faux positifs maximal ou de vrais positifs minimal pour que le modèle soit économiquement viable ?

La problématique posée ne nous permet malheureusement pas de répondre à ces questions, et donc il ne nous est pas possible de trancher pour savoir s'il fait faire varier le seuil de probabilité.

Nous allons cependant proposer une approche « de compromis », que nous avons nommé pompeusement « Seuil Optimal » qui est le seuil permettant simultanément de maximiser le taux de vrais positifs et de minimiser celui de faux positifs. Dans nos différentes courbes ROC ce point sera représenté par le point rouge et sera généralement sur le coude de la courbe ROC.

Ce changement réduit logiquement légèrement l'accuracy, mais nous offre un gain souvent significatif sur le recall. Ici, nous voyons que les jours de pluie sont d'environ 0,8 et les faux positifs de 0,2, c'est-à-dire que lorsqu'il pleut, nous pouvons le prévoir environ 4 fois sur 5, de même que lorsqu'il ne pleut pas. Dans le cadre d'un bulletin météo dans la presse, par exemple, nous pouvons imaginer qu'il s'agit là de performances plus acceptables pour le grand public que la situation précédente dans laquelle nous ne pouvions prédire que la moitié des jours de pluie, quand bien même l'accuracy était plus élevée de 6%.

Observons plus en détail les performances sur les différentes métriques, présentées dans le Tableau 14, et les matrices de confusions présentées dans le Tableau 15. Notons d'ailleurs qu'au-delà de la légère baisse

d'accuracy, nous avons une baisse importante de la précision, c'est-à-dire que, désormais, nous nous trompons presque une fois sur deux lorsque nous prédisons qu'il pleuvra.

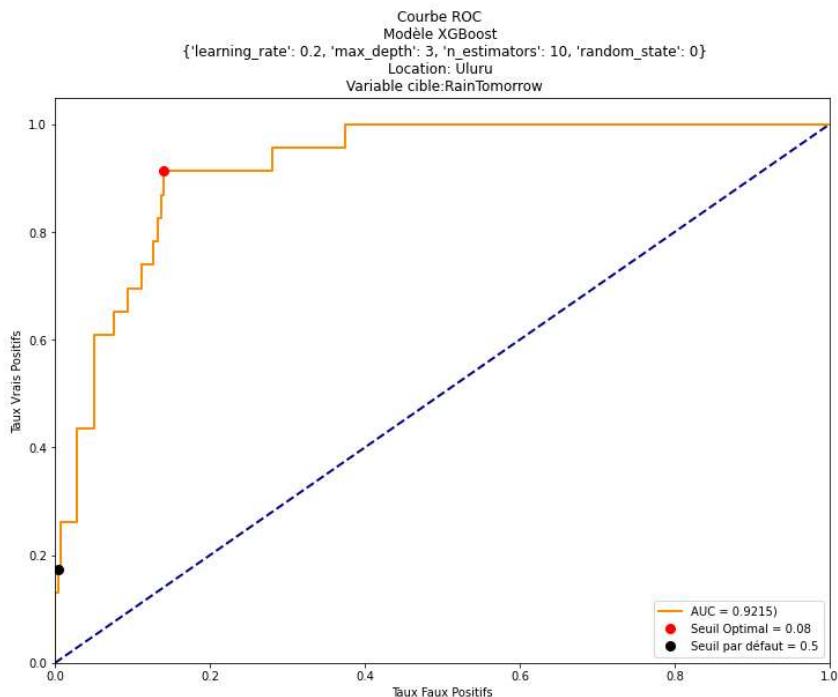
Impact du seuil de probabilité					
	accuracy	recall	precision	f1	auc
Seuil par défaut (0,50)	0.8642	0.5558	0.7642	0.6436	0.9003
Seuil Optimal (0,22)	0.8196	0.814	0.563	0.6656	0.9003

**Tableau 14 : Impact du seuil de probabilité**

		Classe prédictive		Classe prédictive			
		False	True	False	True		
Classe réelle	False	74,2%	3,8%	Classe réelle	False	64,0%	13,9%
	True	9,8%	12,3%		True	4,1%	18,0%
		Seuil par défaut (0,5)		Seuil Optimal (0,2)			

**Tableau 15 : Matrices de confusion**

Regardons le cas tout à fait extrême de la station d'Uluru, située en plein désert (zone Centre) :



**Figure 43 : Courbe ROC des prédictions d'Uluru**

		Classe prédictive		Classe prédictive	
		False	True	False	True
Classe réelle	False	92,0%	0,3%	False	79,4%
	True	6,3%	1,3%	True	13,0%
Seuil par défaut (0,5)			Seuil optimal (0,08)		

En passant du seuil par défaut au seuil optimal (0,08 seulement !), nous passons de prédictions qui nous permettaient de prévoir avec une grande fiabilité lorsqu'il ne pleuvra pas à un modèle dans lequel nous arrivons à capter quasiment l'intégralité des jours où il pleuvra ! Dans une zone aussi aride, nous pouvons imaginer qu'il est beaucoup plus intéressant de pouvoir anticiper les jours où il pleuvra potentiellement plutôt que d'affirmer avec certitude lorsqu'il ne pleuvra pas, mais, là encore, sans contexte économique sur les objectifs à atteindre, il est impossible de trancher sur l'approche la plus pertinente. Il reste très intéressant de voir que les courbes ROC nous permettent de visualiser l'impact de la modification du seuil.

La Figure 44 les 6 courbes ROC des zones climatiques avec les points de seuil par défaut et de seuil optimal : nous voyons que l'impact varie considérablement selon les zones climatiques :

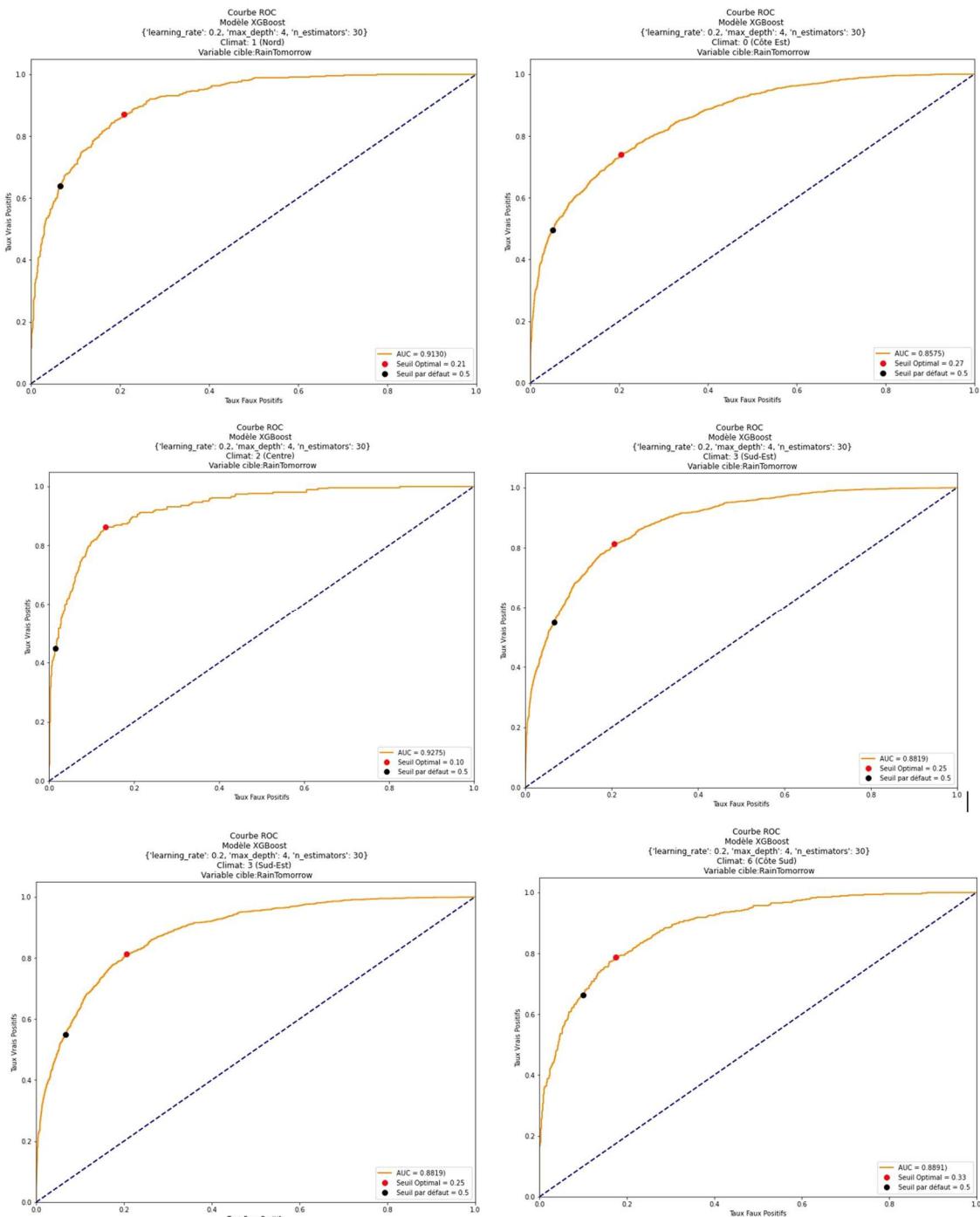


Figure 44: Les courbes ROC des 6 zones climatiques

## 6.3.5 Interprétabilité des modèles

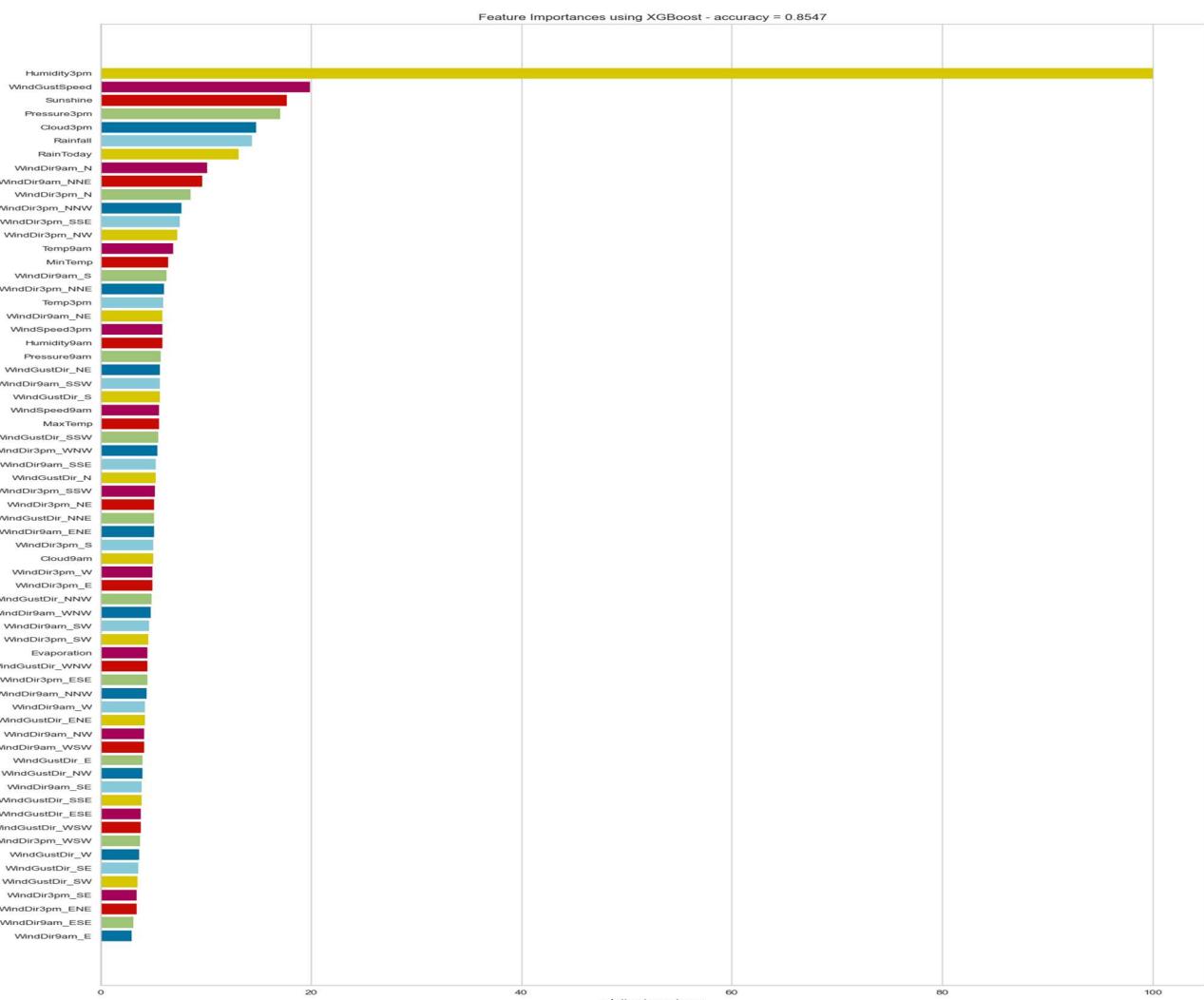
### 6.3.5.1 Importance des features

Sur la quasi-totalité des modèles entraînés, la variable la plus importante semble être de loin *Humidity3pm*, indiquant le taux d'humidité à 15h00. Contre toute attente, *RainToday* semble finalement peu importante dans la prédiction de chute de pluie le lendemain.

Nous avons toutefois pu constater des divergences entre les modèles sur l'importance de chaque variable. Ainsi, si *XGBoost* conserve une importance pour l'ensemble des features, c'est loin d'être le cas des autres modèles. Les *RandomForest* et *TreeClassifier* écartent en particulier la plupart des features pour se concentrer sur quelques-unes. Il est également intéressant d'observer que le poids accordé à chaque variable dépend de la métrique à optimiser. Le cas le plus frappant concerne les *RandomForest* entraînés pour optimiser la précision : dans un contexte de modélisation globale, il s'agit du seul cas où la variable la plus importante devient alors *Cloud3pm*.

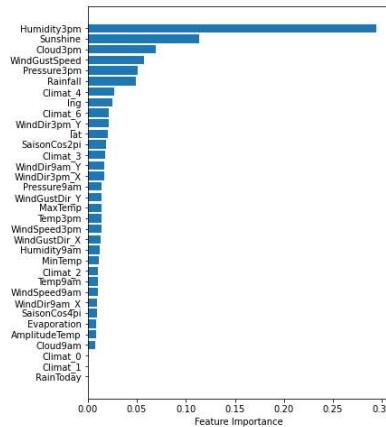
Il est intéressant de relever que *Humidity3pm* était effectivement la variable la plus corrélée avec *RainTomorrow* 0,45, d'une façon symétrique avec *Sunshine* (-0,45). Cette dernière a pourtant un poids nettement inférieur dans les features importances.

Comme nous l'avons indiqué plus haut, nous avons testé des modélisations selon des feature engineering différents, afin de tester plusieurs approches. Les Figure 45 et Figure 46 représentent les features importances sur un modèle entraîné sur un dataset au sein duquel en particulier les variables de vent ont été encodées en OneHot et les features importances d'un second modèle entraîné cette fois avec des variables venteuses trigonométriques et l'ajout de différentes features (zone climatique, coordonnées, variables temporelles).



**Figure 45 : Features importances du modèle XGBoost au niveau macro**

(dataset avec encodage OneHot)

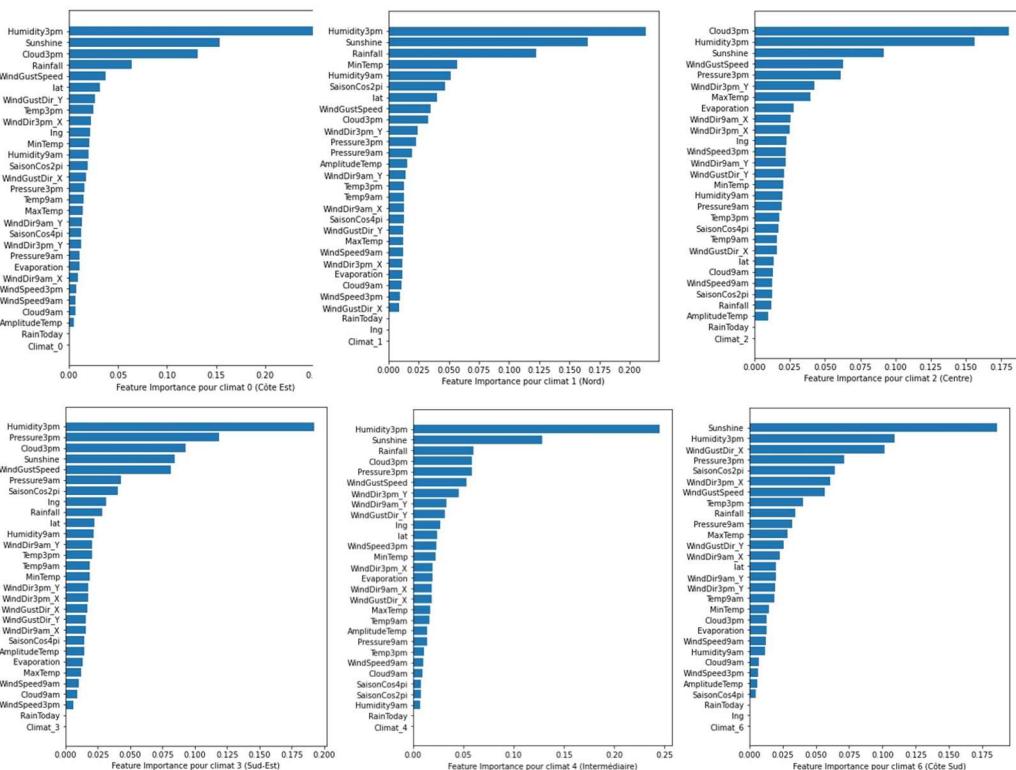


**Figure 46 : Features importances du modèle XGBoost au niveau macro**

(dataset avec features de zones climatiques et variables venteuses converties via sin/cos)

Sur ce second modèle, entraîné sur toute l’Australie, les variables ajoutées de la longitude, la latitude et l’appartenance aux zones climatiques Intermédiaire et Sud-Est sont identifiées parmi le quart des features les plus significatives.

Observons, dans la Figure 47, maintenant les divergences sur des modèles entraînés sur chacune des zones climatiques. Pour toutes les zones climatiques, les variables *Humidity3pm* et *Sunshine* sont très importantes. Il y a par contre des divergences importantes sur le poids de certaines autres variables. Par exemple, si *Cloud3pm* est la feature la plus importante pour la zone Centre, elle se retrouve dans le dernier tiers des variables pour la zone Sud. Nous pouvons voir que la latitude et la longitude se retrouvent généralement dans la moitié des variables les plus importantes, avec toutefois une distinction notable pour la latitude dans la zone Nord qui est carrément mise de côté par le modèle. L’amplitude thermique est assez peu exploitée. SaisonCos2pi est se retrouve dans le premier quart des variables pour la moitié des zones climatiques (Nord, Sud-Est, Sud).



**Figure 47: Features importances des 6 zones climatiques**

### 6.3.5.2 Valeurs de Shapley

Pour déterminer quelles caractéristiques sont généralement les plus importantes pour les prédictions de notre modèle, nous pouvons utiliser un diagramme à barres des valeurs moyennes de SHAP pour toutes les observations. Prendre la moyenne des valeurs absolues garantit que les valeurs positives et négatives ne s'annulent pas.

On observe dans la Figure 48 que la variable avec la valeur SHAP moyenne la plus élevée est *Humidity3pm*, ce qui indique qu'elle a l'impact le plus important sur les prédictions de notre modèle. Ces informations peuvent nous aider à comprendre quelles variables sont essentielles au processus décisionnel du modèle.

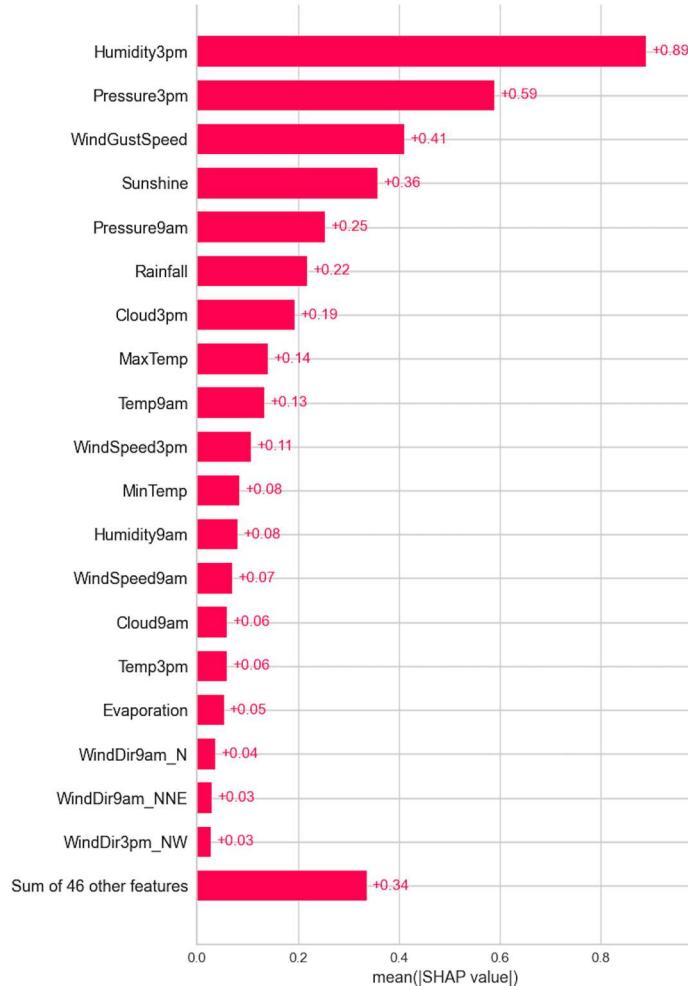


Figure 48: Valeurs de Shapley

### 6.3.5.3 Beeswarm Plot

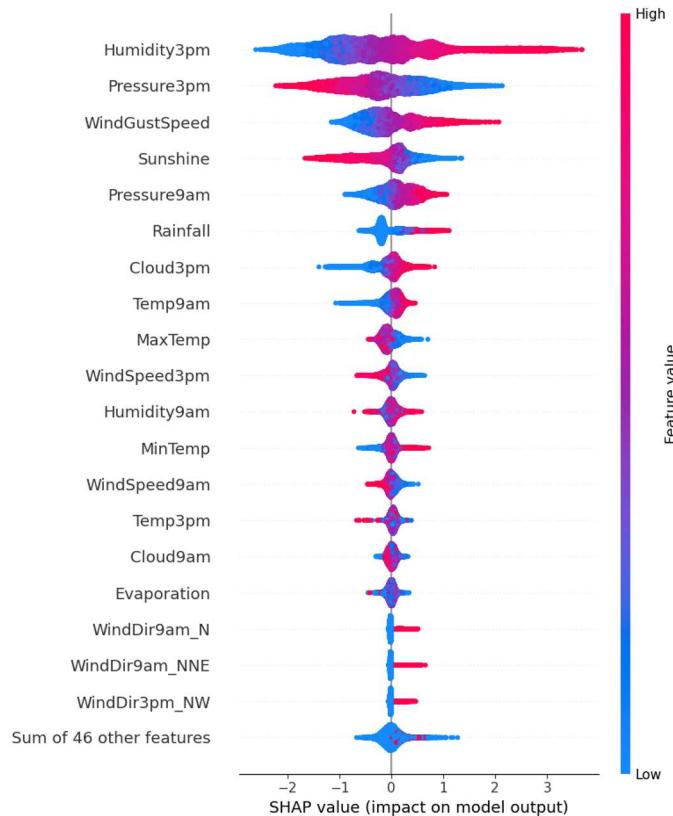
Le graphique Beeswarm est une visualisation utile pour examiner toutes les valeurs SHAP pour chaque entité. L'axe Y regroupe les valeurs SHAP par caractéristique, la couleur des points indiquant la valeur de la caractéristique correspondante. En règle générale, les points les plus rouges représentent des valeurs de caractéristiques plus élevées.

Le beeswarm plot peut aider à identifier les relations importantes entre les caractéristiques et les prédictions du modèle. Dans ce graphique, les caractéristiques sont classées selon leurs valeurs SHAP moyennes.

En examinant les valeurs SHAP dans la Figure 49, nous pouvons commencer à comprendre la nature des relations entre les variables et la pluie du lendemain. Par exemple, pour *Humidity3pm* et *WindGustSpeed*,

nous observons que les valeurs SHAP augmentent à mesure que la valeur de la fonctionnalité augmente. Cela suggère que des valeurs plus élevées de *Humidity3pm* et *WindGustSpeed* contribuent à la probabilité qu'il va pleuvoir demain plus élevée.

En revanche, pour la *Pressure3pm* et la *Sunshine*, nous remarquons la tendance inverse, où des valeurs de caractéristiques plus élevées conduisent à des valeurs SHAP plus faibles. Cette observation implique que des valeurs de *Pressure3pm* et de *Sunshine* plus élevées sont associées à la probabilité qu'il va pleuvoir demain plus faible.

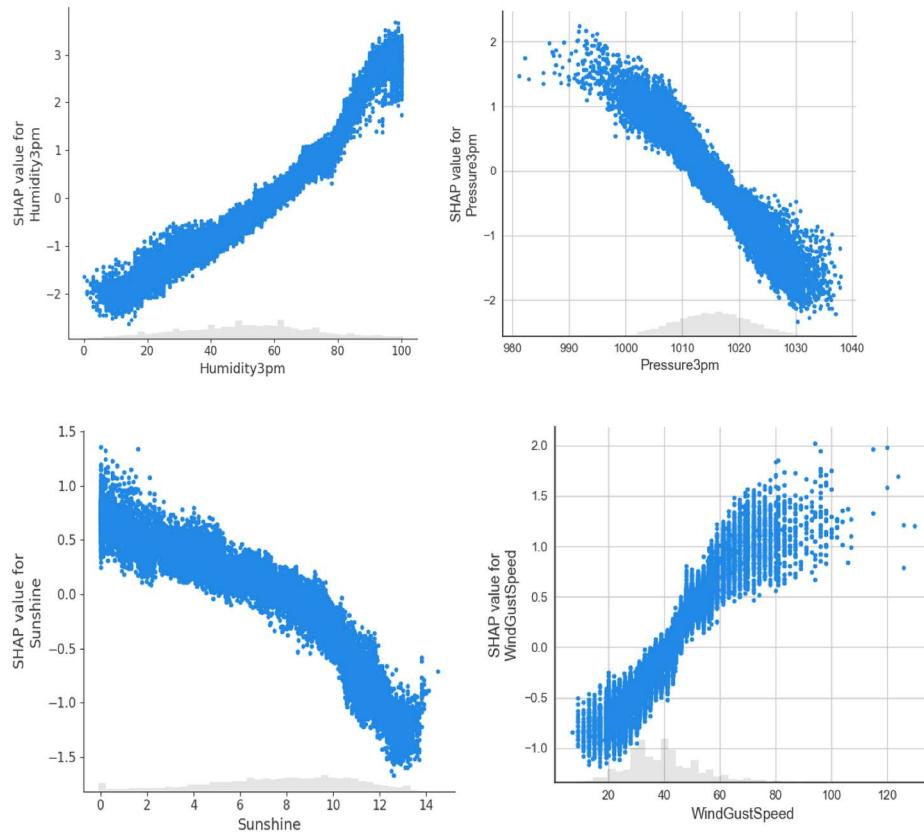


**Figure 49: Beeswarm Plot**

#### 6.3.5.4 Dependence Plots

Pour mieux comprendre les relations entre les caractéristiques individuelles et leurs valeurs SHAP correspondantes, nous pouvons créer des tracés de dépendance. Un diagramme de dépendance est un nuage de points qui montre la relation entre la valeur SHAP et la valeur de caractéristique pour une seule caractéristique.

En analysant les diagrammes de dépendance, la Figure 50, nous pouvons confirmer les observations faites dans le beeswarm plot. Par exemple, lorsque nous créons un diagramme de dépendance pour *Humidity3pm* et *WindGustSpeed*, nous observons une relation positive entre les valeurs de ces variables et les valeurs SHAP. En d'autres termes, des valeurs de ces deux variables plus élevées entraînent des prévisions de la pluie demain plus élevées.



**Figure 50: Dépendances plots**

#### 6.3.5.5 Waterfall Plot

Ce graphique nous aide à visualiser les valeurs SHAP de chaque échantillon dans nos données individuellement. La Figure 51 visualise les valeurs SHAP du premier échantillon de test.

En ignorant les signes, l'ampleur de la valeur SHAP pour la *Humidity3pm*, 0.9, est supérieure à celle des autres variables. Cela impliquait que *Humidity3pm* avait l'impact le plus significatif sur cette prédiction particulière.

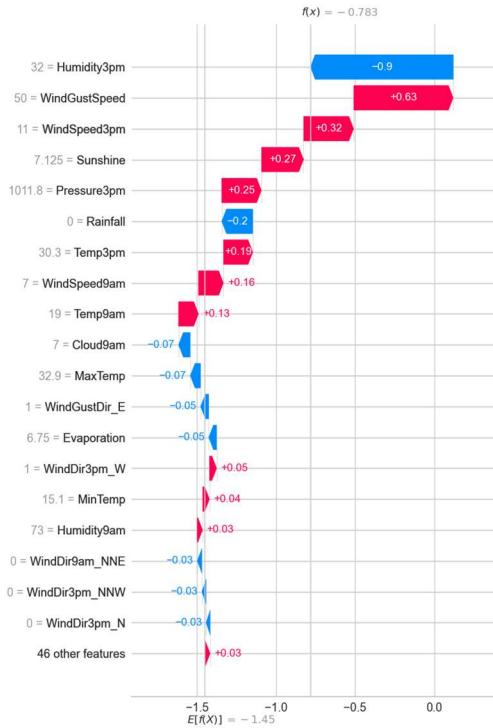


Figure 51: Waterfall plot, exemple 1

Tout comme nous avons visualisé les valeurs SHAP du premier échantillon, nous pouvons également visualiser les valeurs SHAP du deuxième échantillon de test, comme dans la Figure 52.

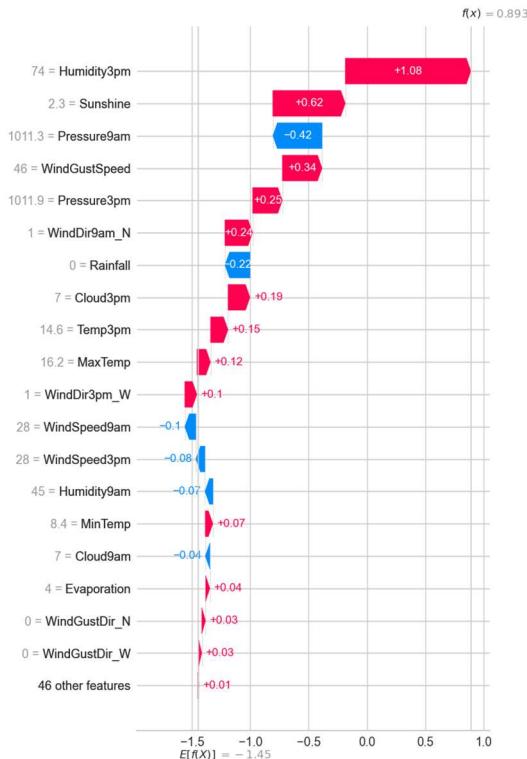


Figure 52: Waterfall plot, exemple 2

## 6.4 Deep Learning avec Keras et TensorFlow

### 6.4.1 DNN

#### 6.4.1.1 Nombre de couches et nombre de neurones

Notre première problématique va consister à dimensionner correctement notre réseau de neurones dense, tant en nombre de couches cachées qu'en nombre de neurones par couches.

Pour cela, nous allons procéder itérativement par couche. Nous allons débuter par un « réseau » sans couche cachée, constitué du seul neurone de la couche de sortie.

Nous allons ensuite ajouter une première couche cachée, avec un faible nombre de neurones (5), un grand (200) et un nombre intermédiaire (50). Nous avons en réalité tester beaucoup d'autres combinaisons : nous allons retenir ici les cas extrêmes pour synthétiser les résultats obtenus.

Nous allons ensuite ajouter une deuxième couche en testant plusieurs combinaisons, puis nous regarderons l'apport d'une troisième couche.

Les modèles ont tous été entraînés avec un *learning rate* de 0.001, un *batch\_size* de 512, des fonctions d'activation *tanh* et sur 300 époques. Ces paramètres seront affinés par la suite.

Notre fonction loss est une *binary\_crossentropy*, et notre métrique une *binary\_accuracy*. Enfin, le neurone de la couche de sortie sera activé par une fonction *sigmoïde*.

Comparaison des réseaux de neurones					
	accuracy	recall	precision	f1	auc
Aucune couche cachée	0,8533	0,5198	0,7375	0,6098	0,8773
Couche cachée (CC) de 5 Neurones (n)	0,8612	0,5484	0,7554	0,6355	0,8925
CC 200 neurones	0,8637	0,5626	0,7574	0,6456	0,8982
CC 50 neurones	0,8638	0,5798	0,7461	0,6525	0,8986
CC 200 n, CC 200 n	0,8561	0,6257	0,6924	0,6574	0,8922
CC 200 n, CC 50 n	0,8660	0,5391	0,7863	0,6396	0,9018
CC 50 n, CC 200 n	0,8655	0,5621	0,7657	0,6483	0,9003
CC 50 n, CC 50 n	0,8656	0,6211	0,7293	0,6709	0,9020
CC 10 n, CC 10 n	0,8644	0,5988	0,7370	0,6607	0,8976
CC 50 n, CC 50 n, CC 50 n	0,8598	0,5772	0,7305	0,6449	0,8939
CC 50 n, CC 50 n, CC 5 n	0,8637	0,5922	0,7381	0,6571	0,9011

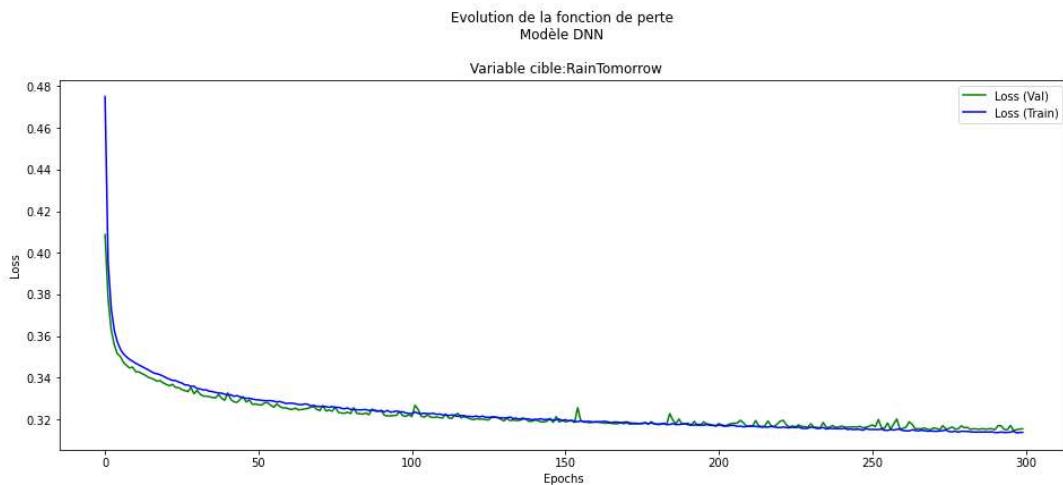
Tableau 16 : Les scores des différents modèles de réseau de neurones

Toutes ces métriques sont souvent très proches. Malgré des réseaux très différents, allant d'un perceptron à un réseau dense de 3 couches de 50 neurones par couche, les performances ne se distinguent parfois que par la troisième décimale de la métrique observée. Il semble de prime abord difficile de trancher sur une topologie de réseaux en se basant uniquement sur ces métriques. Et pour cause : le comportement de ces différents réseaux évolue différemment tout au long de ces 300 époques. Certains d'entre eux divergent même après quelques dizaines d'époques seulement, indiquant qu'il faudrait stopper l'apprentissage et ne justement surtout pas le laisser perdurer sur 300 époques. En particulier, de façon systématique, un trop grand nombre de neurones entraîne rapidement un accroissement de la fonction de perte en validation, voire

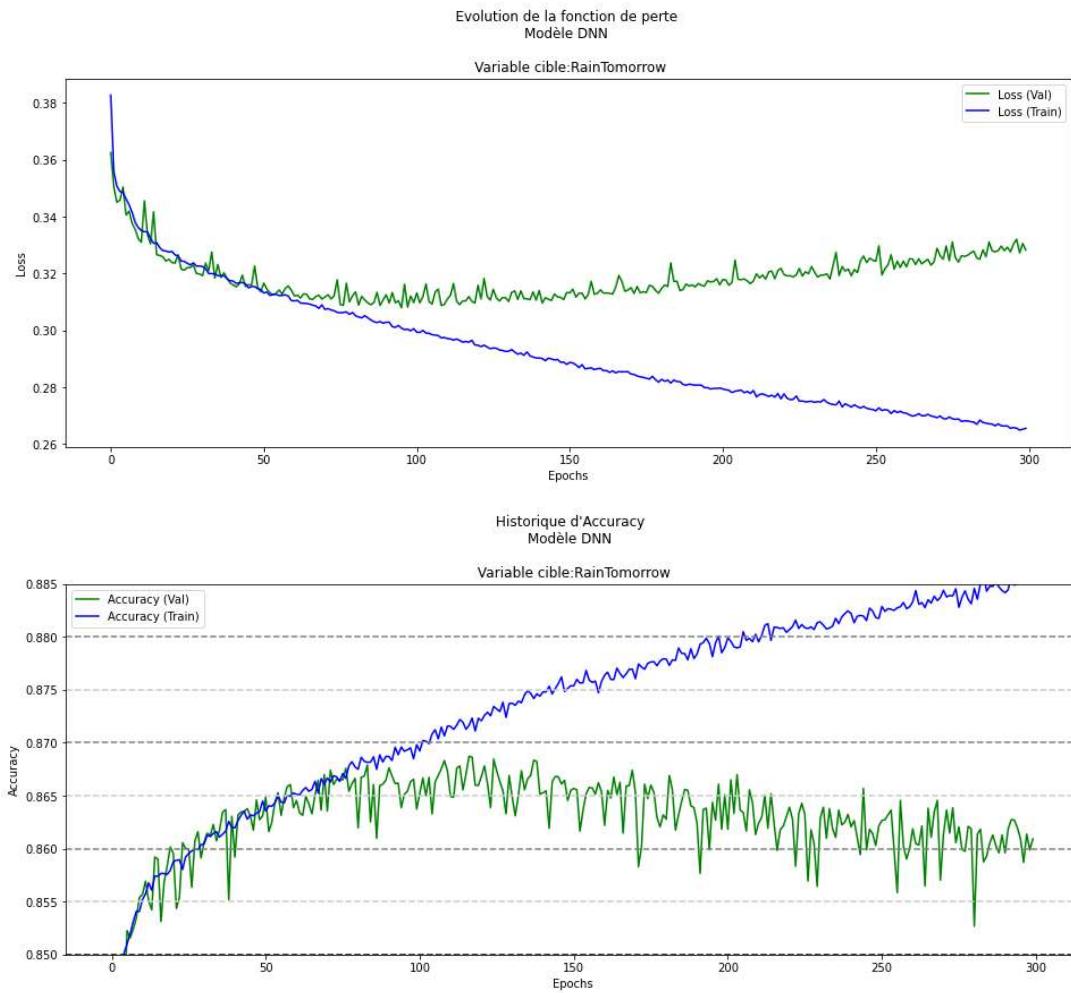
une baisse de l'accuracy sur l'échantillon de validation. Il en va de même lorsque nous introduisons la troisième couche. Les réseaux à une seule couche cachée ne semblent pas souffrir de ces défauts, mais apprennent moins bien que ceux possédant une seconde couche cachée.

La conclusion de l'observation de ces courbes d'apprentissage ne nous permet donc que d'écartier timidement certains modèles. En effet, les courbes nous montrent surtout que la présence d'un grand nombre de neurones ou celle d'une troisième couche cachée rend le réseau beaucoup plus sensible au surapprentissage, et nous poussera donc à redoubler de vigilance sur le nombre d'époques pour ce type de réseau. On peut voir sur la courbe d'accuracy du réseau (200,200) (Figure 54 : Evolution de la fonction de perte pour (200,200) – l'apprentissage diverge au-delà de l'époque 60 ) que l'accuracy maximale atteinte lors de l'apprentissage vers l'époque 60 est très légèrement plus faible que celle du réseau (50,50). On pourrait donc privilégier ce dernier. Toutefois, étant donné qu'aucun hyperparamètre n'a été encore optimisé à ce stade, il est prématûré de tirer des conclusions définitives sur le réseau optimal.

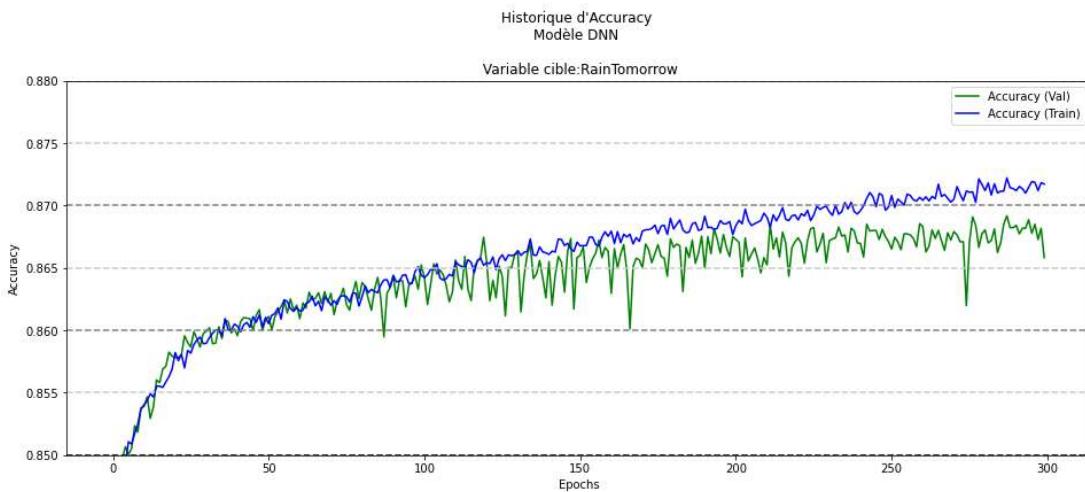
Pour la suite, nous faisons le choix de conserver un réseau de deux couches cachées de 50 neurones chacune car ce modèle offre des performances intéressantes tout en étant assez stable et moins sensible au surapprentissage que des réseaux plus complexes. De plus, nous avons également réalisé les travaux de recherche de paramètres optimaux décrits ci-après sur des réseaux plus complexes sans obtenir *in fine* de meilleures performances y compris avec des entraînement sur plusieurs milliers d'époques.



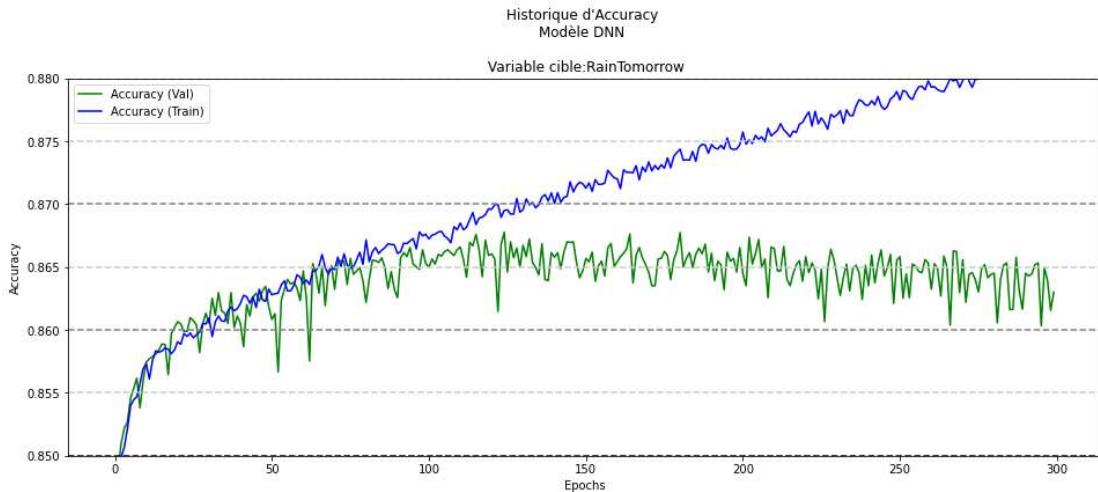
**Figure 53 : Evolution de la fonction de perte pour (50) – bon apprentissage**



**Figure 54 : Evolution de la fonction de perte pour (200,200) – l'apprentissage diverge au-delà de l'époque 60**



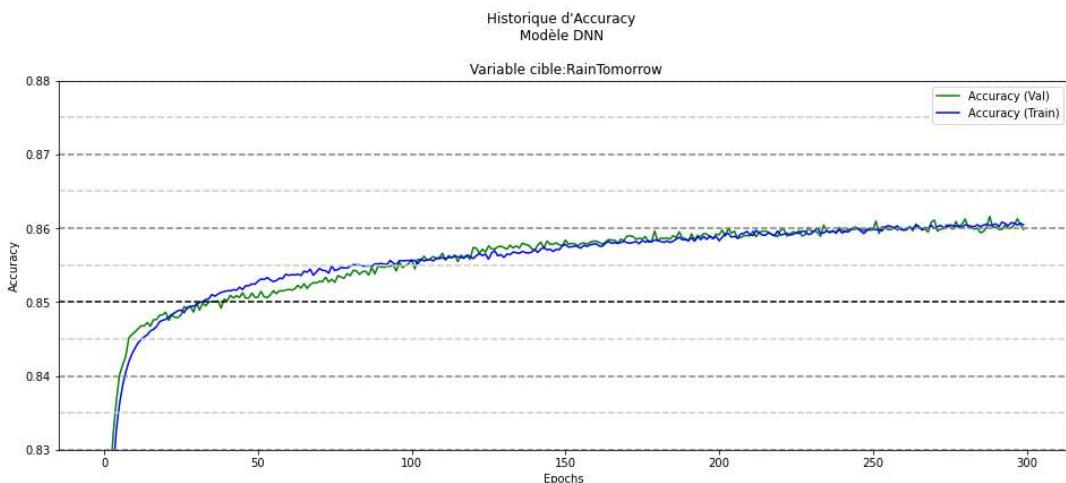
**Figure 55 : Evolution de l'accuracy pour (50,50) – overfitting croissant à partir de l'époque 150**



**Figure 56 : Evolution de l'accuracy pour (50,50,50) – overfitting entraînant une dégradation de la validation dès l'époque 110**

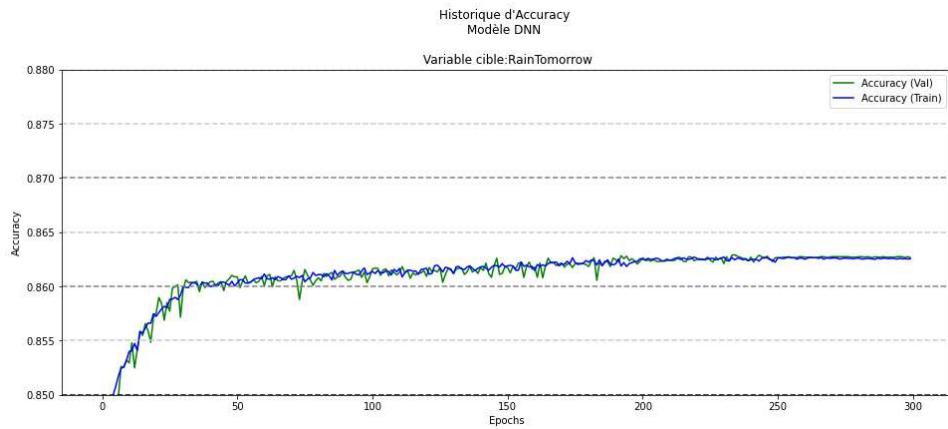
#### 6.4.1.2 Learning Rate

Le taux d'apprentissage de  $10^{-3}$  utilisé ci-dessus n'est pas assez fin : nous voyons des sauts assez brusques d'une époque à l'autre. L'apprentissage avec un taux plus faible ( $10^{-4}$ ) ne présente plus ces défauts. En revanche, nous voyons qu'il faudrait davantage d'époques pour que le modèle converge. Un taux de  $10^{-5}$  ne permet pas même d'atteindre la limite de 0,850 à l'issue des 300 époques.

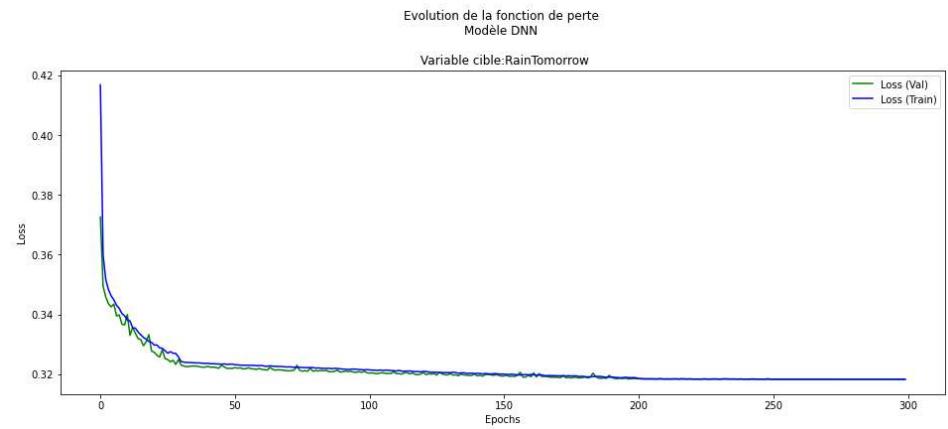


**Figure 57 : Learning Rate de  $10^{-4}$**

Nous allons ajouter une callback via `LearningRateScheduler` qui adaptera le learning rate en fonction de l'époque. Nous allons débuter par un learning rate de  $10^{-3}$  pendant les 30 premières époques afin d'atteindre rapidement une accuracy de 0,86. Ensuite, nous allons passer sur une accuracy de  $10^{-4}$  (Figure 57 : Learning Rate de  $10^{-4}$ ) pour poursuivre l'apprentissage de façon plus précise jusqu'à l'époque 200, au-delà de laquelle nous passons à  $10^{-5}$ . Enfin, nous finaliserons sur les 50 dernières époques sur un learning rate de  $10^{-6}$  seulement. La courbe d'apprentissage ainsi obtenue converge de façon harmonieuse (Figure 58 : Learning rate dynamique avec callback). A dire vrai, il serait même possible d'arrêter l'apprentissage dès l'époque 200.



**Figure 58 : Learning rate dynamique avec callback**



**Figure 59 : Fonction de perte avec Learning rate dynamique**

La fonction de perte évolue cette fois sans pics, en convergeant rapidement vers son minimum.

Ce learning rate dynamique ne permet pas directement d'apporter un gain sur les résultats, car il suffirait d'entraîner sur beaucoup plus d'époques un modèle avec un faible taux d'apprentissage. Mais il nous permet d'obtenir les résultats optimums des modèles avec beaucoup moins d'époques, et donc de temps de calcul lors de l'apprentissage.

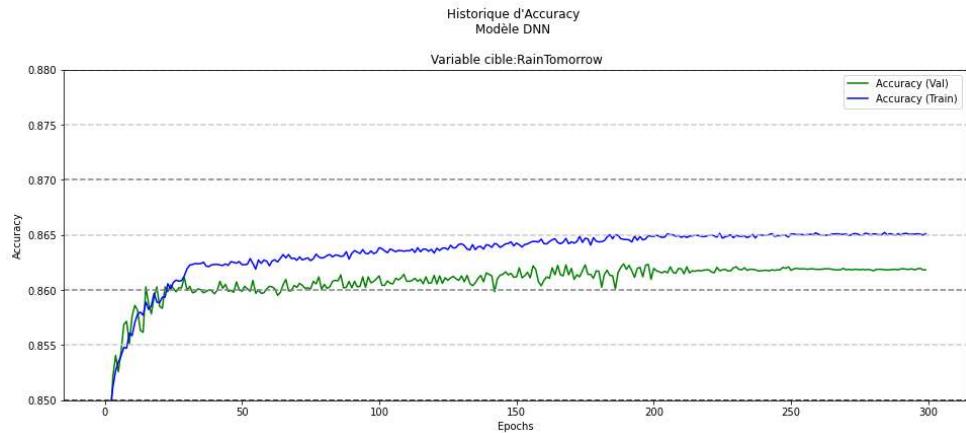
#### 6.4.1.3 Fonctions d'activation

Comparaison des fonctions d'activation					
	accuracy	recall	precision	f1	auc
Tanh / Tanh	0,8631	0,5617	0,7551	0,6442	0,8957
ReLU / ReLU	0,8621	0,5594	0,7518	0,6415	0,8970
ReLU / Tanh	0,8629	0,5664	0,7511	0,6458	0,8958
Tanh / ReLU	0,8637	0,5588	0,7598	0,6440	0,8967

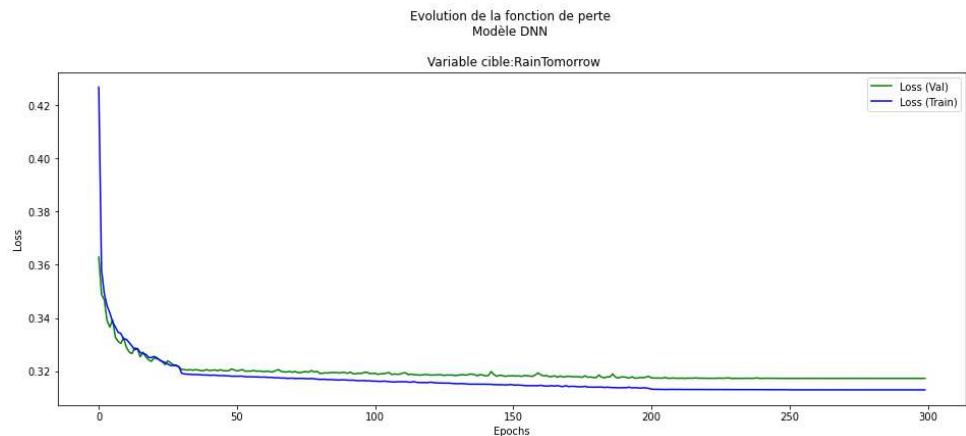
**Tableau 17 : Comparaison des fonctions d'activation**

Les historiques nous montrent que l'apprentissage d'un réseau ReLU/ReLU (la Figure 60) est plus dispersé qu'un apprentissage Tanh / Tanh (Figure 60-Figure 61), permettant potentiellement de sortir de minimum local. C'est un constat intéressant qui nous a ensuite incité à tester des combinaisons ReLU / Tanh et Tanh / ReLU. Bien nous en a pris : il semble que cette dernière configuration nous permette un léger gain. Nous

conserverons donc ensuite un réseau avec une première couche de 50 neurones activés par Tanh et une seconde couche de 50 neurones activés par ReLU.



**Figure 60 : Accuracy du réseau ReLU / ReLU**



**Figure 61 : Loss du réseau ReLU / ReLU**

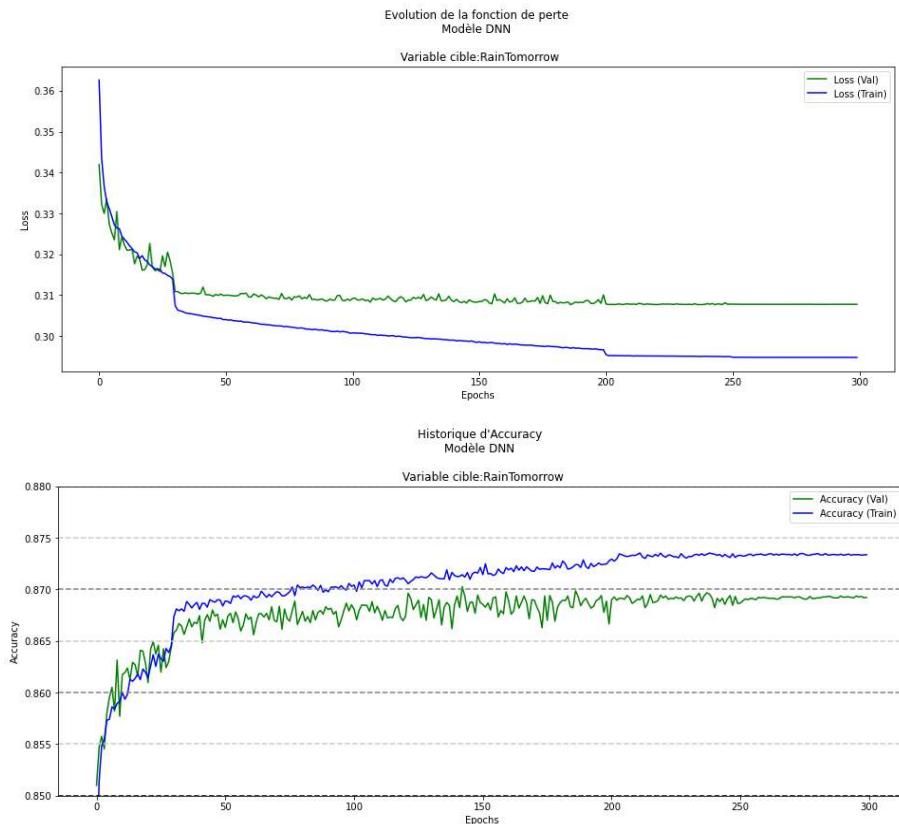
#### 6.4.1.4 Batch size

Nous avions pris un batch size relativement grossière de 512 échantillons. Le Tableau 18 représente l'impact d'une variation de cette quantité d'une part sur les métriques, mais également sur les temps de calculs.

Variation du batch size						
Batch_size	accuracy	recall	precision	f1	auc	Temps
2048	0,8597	0,5567	0,7430	0,6365	0,8914	0 mn 33s
512	0,8637	0,5588	0,7598	0,6440	0,8967	1 mn 23s
128	0,8660	0,5655	0,7656	0,6505	0,9006	3 mn 48
16	0,8666	0,5767	0,7609	0,6561	0,9024	13 mn 25s

**Tableau 18 : Variation du batch\_size**

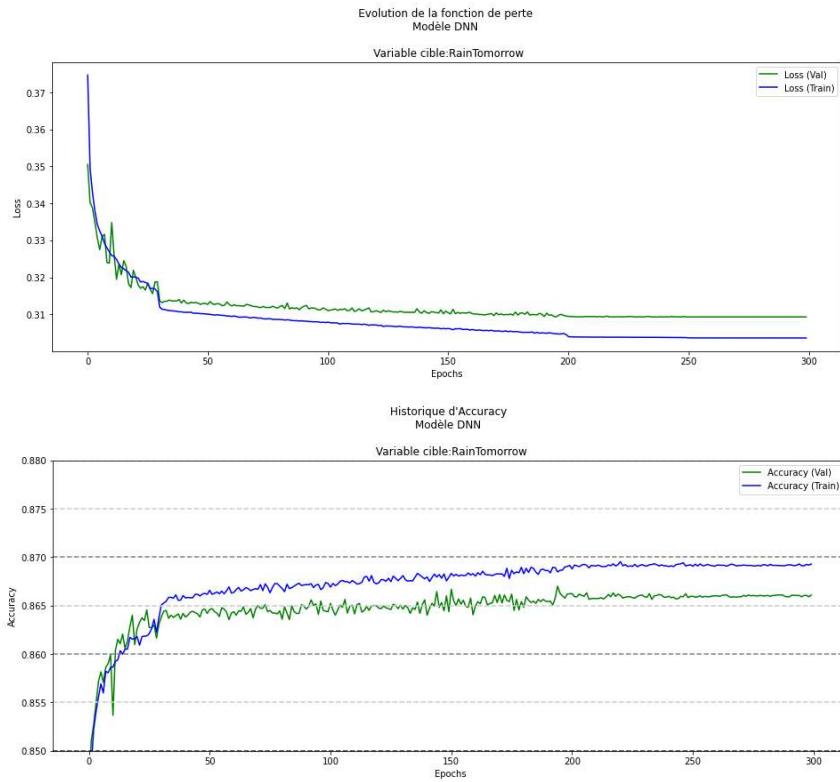
Le modèle avec *batch\_size* de 16 est le plus performant, mais également de loin le plus lent. On remarque une différence relativement marquée entre les échantillons de train et de validation, tant pour l'évolution de la *loss* que de l'*accuracy*. Il est amusant de remarquer qu'on peut constater une baisse de la *loss* à chaque changement de learning rate par notre callback aux époques 30, 200 et 250. Au final, étant donné le faible écart de performances, le temps bien plus important pour entraîner le modèle de *batchsize*=16 et l'écart entre l'échantillon de train et de test, notre préférence se portera sur un *batch\_size* de 128.



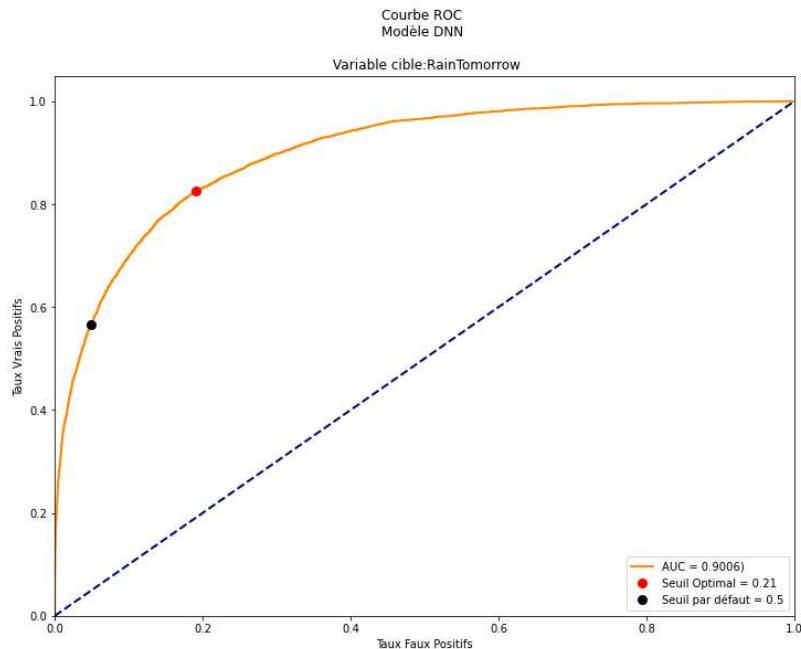
**Figure 62 : Evolutions de l'apprentissage pour modèle avec batch\_size = 16**

#### 6.4.1.5 Comparaison des performances

Maintenant que nous avons obtenu notre meilleur modèle de DNN (learning rate dynamique via notre callback personnalisée, batch size=128, 1<sup>ère</sup> couchée cachée de 50 neurones avec fonction d'activation tanh, 2<sup>ème</sup> couche cachée de 50 neurones avec fonction d'activation ReLU, 300 époques), observons ses courbes d'apprentissage présentées dans la Figure 63, puis comparons ses performances avec notre meilleur XGBoost. Nous allons profiter du fait que notre neurone de sortie applique une fonction continue pour afficher sa courbe ROC. Comme nous l'avons fait précédemment, nous regarderons l'impact de la variation du seuil de classification dans les métriques.



**Figure 63 : Evolutions de l'apprentissage de notre modèle final**



**Figure 64 : Courbe ROC du modèle DNN final**

<b>Comparaison du DNN avec XGBoost</b>					
	<b>accuracy</b>	<b>recall</b>	<b>precision</b>	<b>f1</b>	<b>auc</b>
XGBoost - Seuil par défaut (0,50)	0.8642	0.5558	0.7642	0.6436	0.9003
XGBoost - Seuil Optimal (0,22)	0.8196	0.8140	0.5630	0.6656	0.9003
DNN – Seuil par défaut	0,8660	0,5655	0,7656	0,6505	0,9006
DNN – Seuil Optimal	0,8131	0,8263	0,5510	0,6611	0,9006

Avec le seuil par défaut, le DNN est plus performant que le XGBoost sur toutes les métriques, mais dans des proportions infimes. Avec le seuil optimal, le DNN est meilleur sur le recall et l'AUC, là aussi de façon peu significative.

#### 6.4.1.6 Interprétabilité

Si *Humidity3pm* reste une feature importante, remarquons que *Pressure3pm* revêt aux yeux du DNN une importance bien plus élevée qu'avec nos modèles XGBoost. L'appartenance aux zones climatiques, la latitude et la longitude ont une importance notable.

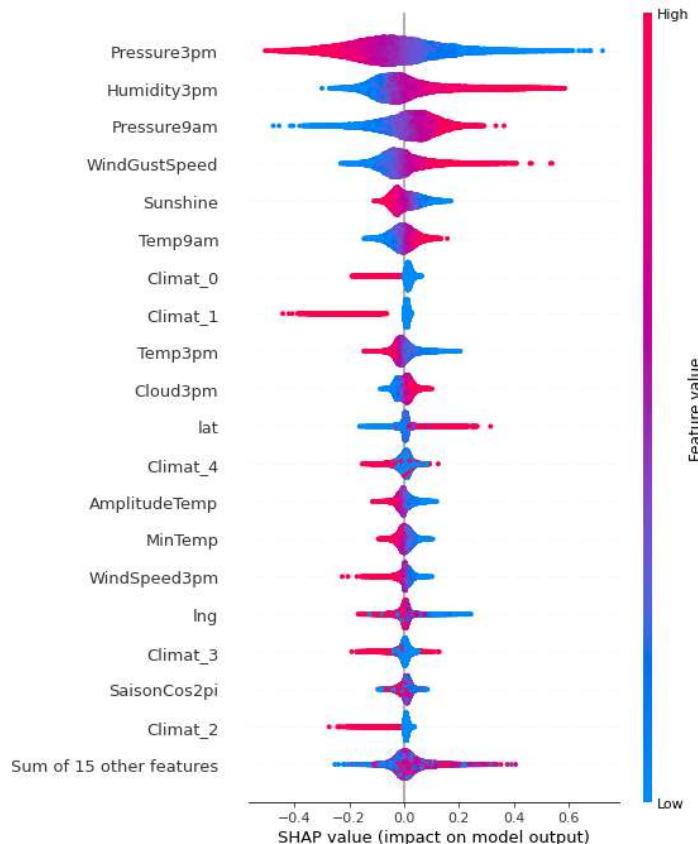


Figure 65 : beeswarm des valeurs de Shapley de notre réseau dense

Regardons le poids des variables explicatives pour quatre prédictions par notre DNN représentés dans la Figure 66

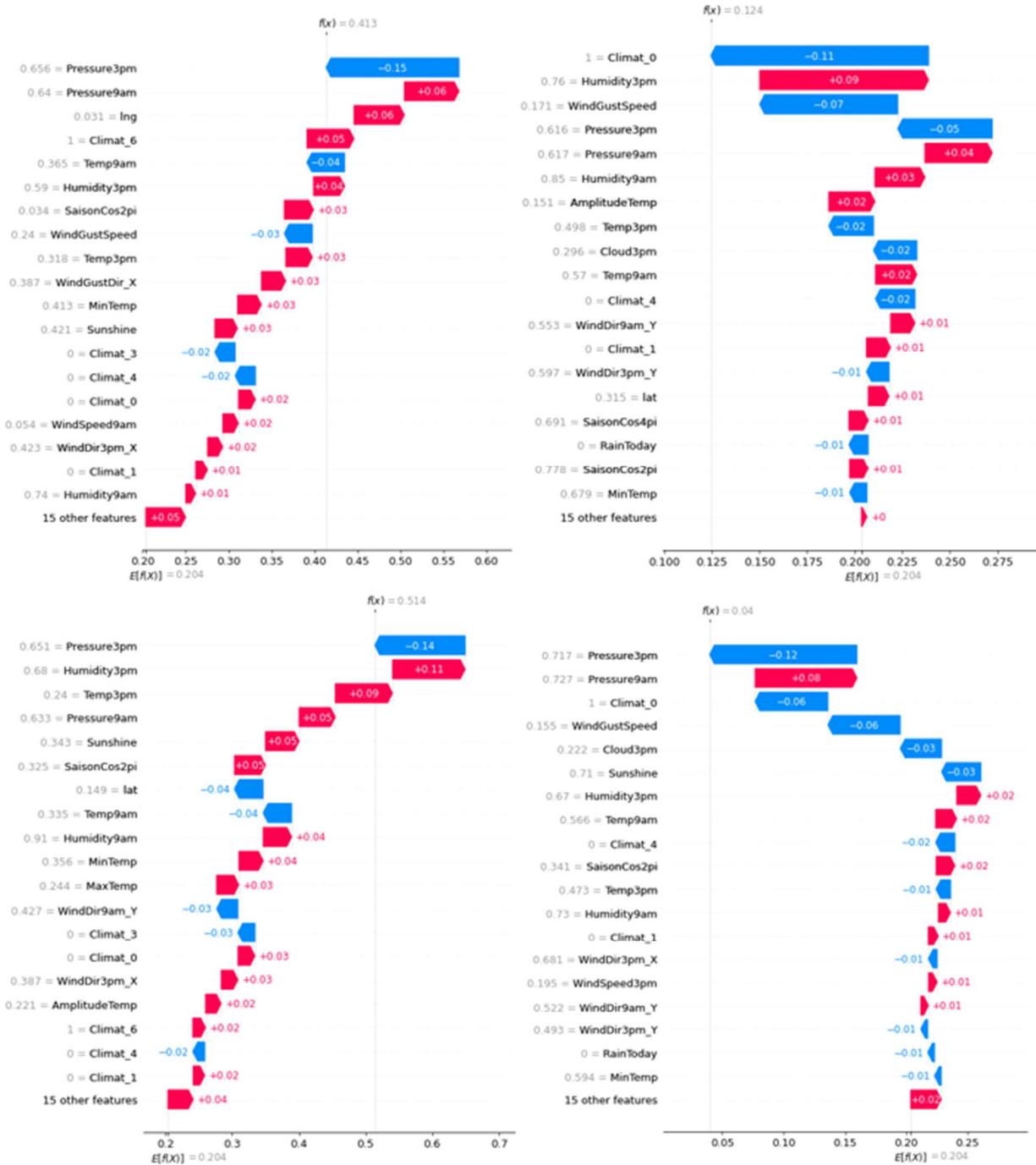


Figure 66: Exemples de quatre prédictions par le modèle DNN

Dans la première prédition, *Pressure3pm* pèse beaucoup pour indiquer qu'il ne pleuvra pas, contrairement à *Pressure9am*. La latitude et l'appartenance à la zone climatique de la côte sud jouent en la faveur de la pluie également.

C'est l'appartenance à la zone climatique de la côte Est qui a le plus de poids dans notre seconde prédition en haut à droite, devant l'*Humidity3pm* qui, elle, incite à prévoir la pluie.

Les variables de saisonnalité *SaisonCos2pi* et *SaisonCos4pi* n'influent que modérément les décisions. Bien que le poids des features explicatives varie beaucoup d'une prédition à l'autre, nous remarquerons que *Pressure3pm* est en tête de trois d'entre elle dans l'importance de la prise de décision.

#### 6.4.1.7 Conclusion

L'entraînement de nos DNN a été parfois très long lors de nos tâtonnements. A de nombreuses reprises, nous avons entraîné des modèles prenant plusieurs dizaines de minutes, voire plus d'une heure. Les XGBoost, quant à eux, ne prennent chaque fois que quelques secondes à être entraînés. De plus, il y a un nombre bien plus important de paramètres à définir sur un DNN que sur un XGBoost, ce qui implique une recherche également plus importante en temps humain. Certes, notre DNN offre légèrement de meilleures performances que notre XGBoost, mais, dans le cadre d'un projet avec un budget défini, cette expérience nous montre qu'il faudra être particulièrement attentif sur le temps total que nous serons prêts à consacrer à un DNN et qu'il faudra bien peser le rapport du bénéfice de performances par rapport au coût de développement.

### 6.4.2 RNN

#### 6.4.2.1 Prédiction monovariée

Jusque-là, pour effectuer la prédiction à  $J+1$ , nous ne disposions que des relevés météo à  $J$ . Les réseaux récurrents vont nous permettre de bénéficier également de l'apport des relevés météorologiques de plusieurs jours précédents la prédiction. Une des difficultés consistera d'ailleurs à déterminer le nombre de journées optimum à reprendre.

Contrairement aux DNN et aux modèles de machine learning classiques, l'entraînement d'un RNN nécessite que les données d'entraînement soient triées chronologiquement et qu'il n'y ait que peu de trous dans la chronologie.

Cette chronologie implique que notre RNN ne pourra être entraîné qu'au niveau micro, et non sur des zones climatiques ou sur l'ensemble de l'Australie, puisqu'une seule observation ne sera utilisée par jour.

Les modèles monovariés de prédiction de *RainTomorrow* donnent des résultats particulièrement mauvais : quels que soient les propriétés de notre RNN, les résultats sont à peine meilleurs qu'un tirage aléatoire, et donc sans commune mesure avec les résultats obtenus précédemment avec nos modèles de machine learning classique. Et pour cause : n'oublions pas que contrairement aux modèles précédents, nous n'exploitons ici que la seule feature *RainTomorrow*. Nous ne bénéficions donc plus de l'apport des autres features. Or, cette variable ne présentant que peu de régularité, nous comprenons ces résultats médiocres.

Il semble indispensable de disposer d'un modèle multivarié pour prédire *RainTomorrow* avec un RNN.

L'architecture que nous avons retenue pour les schémas suivants possède les propriétés suivantes :

- 5 LSTM, activation tanh
- 10 LSTM, activation relu
- 10 Dense
- Sortie : 1 Dense

Nous avons relativement peu de neurones car il n'y a ici qu'une seule feature exploitée.

Nous avons un learning rate de  $10^{-3}$ , un batch\_size de 1 et une fenêtre d'apprentissage de 3 journées. Là encore, nous avons tenté de nombreuses autres combinaisons (LR de  $10^{-1}$  à  $10^{-4}$ , fenêtre de 2 à 15 jours, variation du nombre de neurones et de couches) sans jamais arriver à des performances intéressantes.

L'apprentissage ci-dessous a été fait sur 10 époques. Là encore, nous avons tenté avec bien davantage d'époques, mais il n'y avait aucun intérêt à poursuivre l'apprentissage : comme nous pouvons le voir sur les courbes d'apprentissage ci-dessous, nous stagnons assez vite. Nous voyons que l'apprentissage se déroule assez mal : la fonction de perte converge rapidement vers 0,52, ce qui est très mauvais pour une binary\_crossentropy. L'accuracy reste stable à 0,75 sans s'améliorer avec les époques.

Les performances finales du modèle sont une accuracy de 0,7526, un recall de 0,5185, une précision de 0,5176 et un F1 de 0,5180.

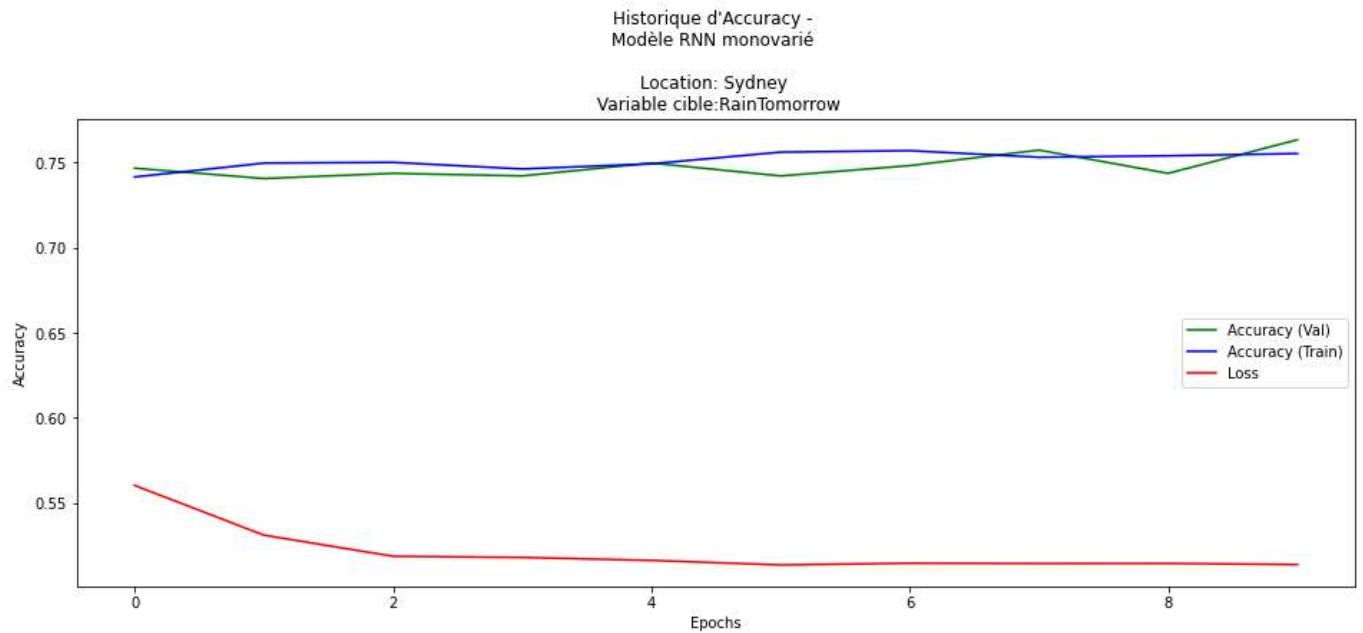


Figure 67: Apprentissage du RNN monovarié pour Sydney sur RainTomorrow

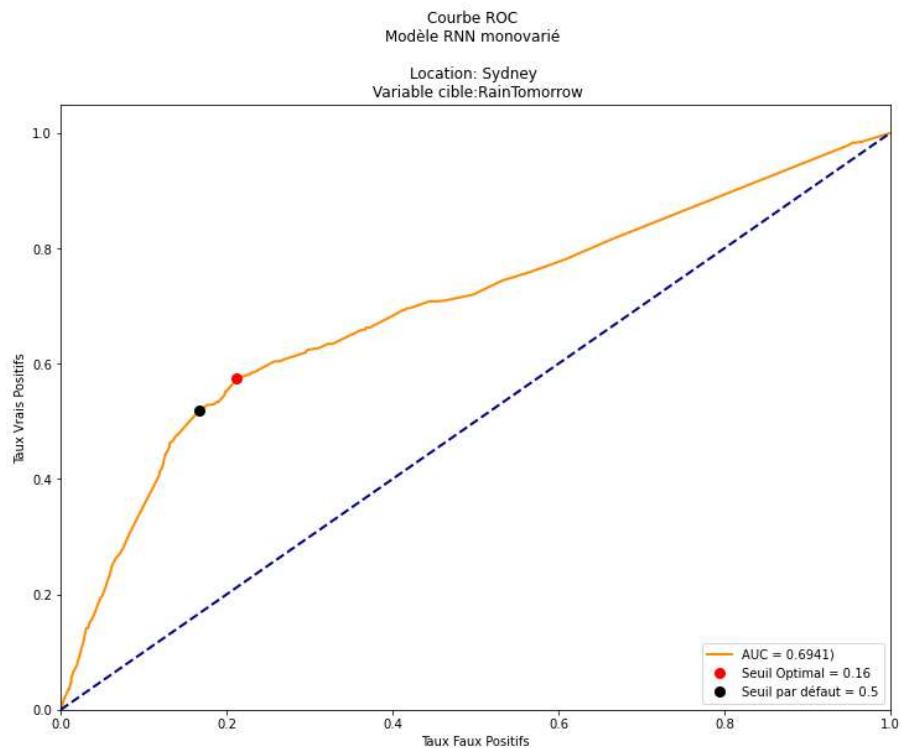


Figure 68: Courbe ROC du RNN monovarié pour Sydney sur RainTomorrow

#### 6.4.2.2 Prédiction multivariée

L'approche multivariée ne nous a pas permis d'améliorer les performances par rapport à l'approche monovariée.

Nous reprenons ici l'ensemble des features de notre dataset, et nous avons autant de neurones en couche de sortie afin de prédire chaque variable. Tout comme pour l'approche monovariée, nous avons tenté de très nombreuses combinaisons de paramètres, sans jamais arriver à des résultats probants.

L'architecture retenue pour les schéma suivants est :

- 50 LSTM, activation tanh
- 50 LSTM, activation relu
- 50 Dense
- Sortie : 27 (=autant de neurones que de features) Dense

Comme nous avons à la fois des variables quantitatives et qualitatives, nous utilisons non pas une mais deux fonctions loss (binary\_crossentropy pour les variables binaires, MSE pour les variables quantitatives).

Tout comme pour l'approche monovariée, nous voyons que l'apprentissage se déroule plutôt mal, avec une loss qui converge rapidement vers 0,56 et l'accuracy qui stagne. La métrique s'appliquant ici sur l'ensemble des variables, elle n'est pas à exploiter directement pour l'évaluation des performances, mais il s'agit tout de même d'un critère montrant que le modèle n'arrive pas à généraliser.

Les performances finales du modèle sur RainTomorrow sont une accuracy de 0,7659, un recall de 0,4631, une précision de 0,5520 et un F1 de 0,5036.

Nous espérions que l'approche multivariée nous permettrait de gagner en performance, mais il n'en est rien. Nous supposons que cet échec provient de la difficulté de prédire certaines variables telles que Rainfall, rendant la tâche particulièrement complexe pour le modèle.

L'utilisation des RNN, en monovarié et multivarié, est donc un échec pour prédire la pluie. Nous verrons en revanche plus bas qu'ils présentent davantage d'intérêt pour prédire la température.

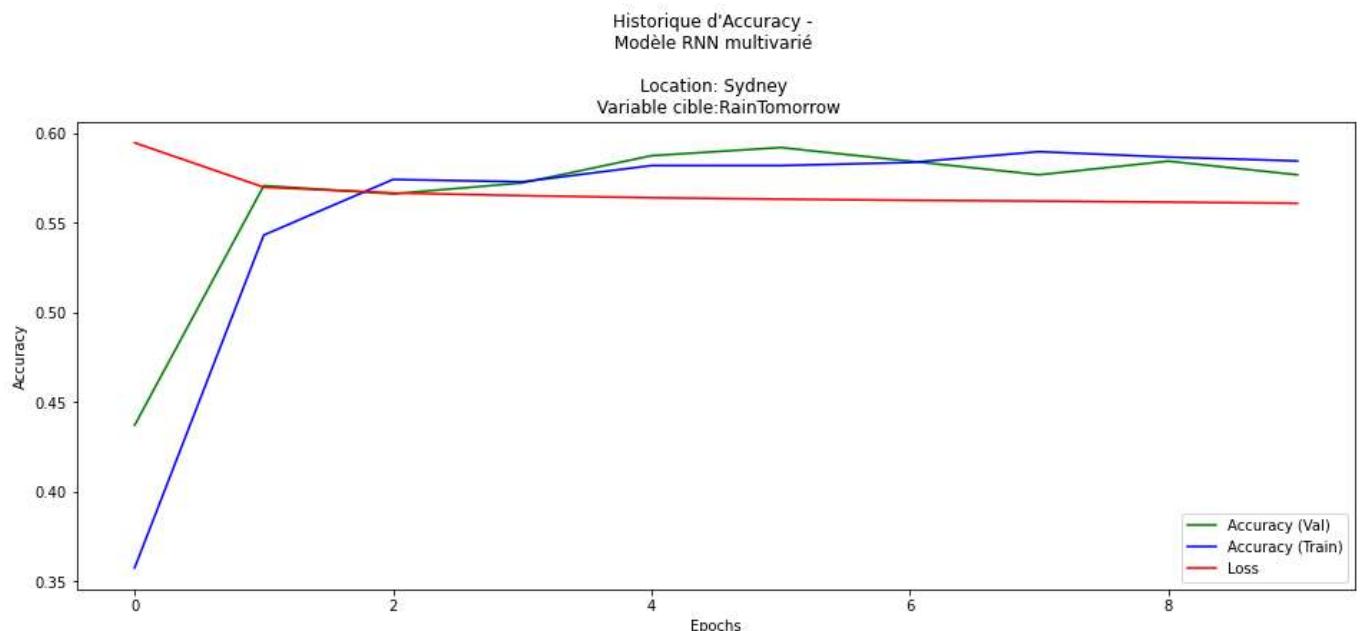


Figure 69: Apprentissage du RNN multivarié pour Sydney sur RainTomorrow

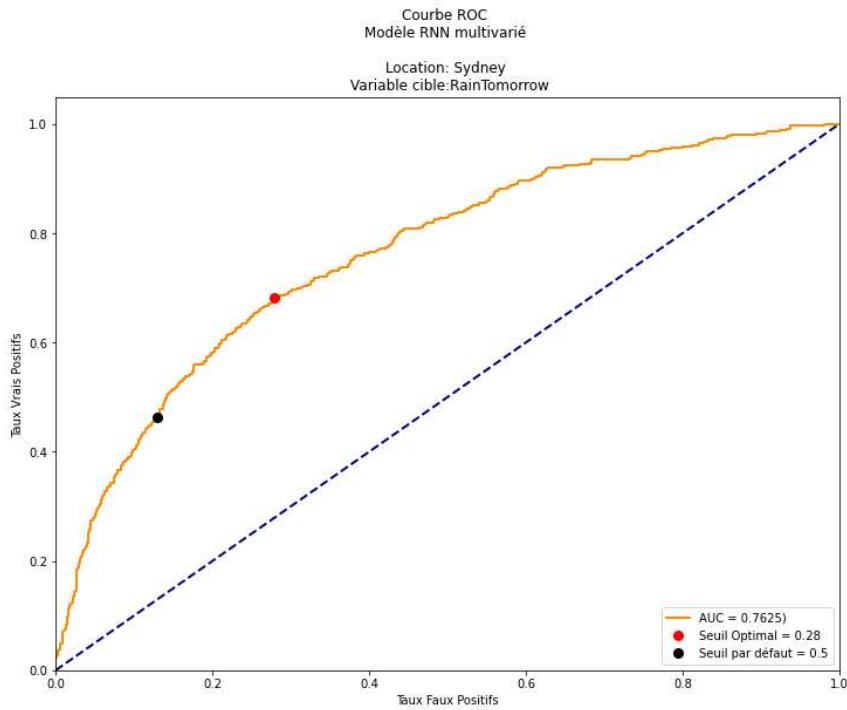


Figure 70: Courbe ROC du RNN multivarié pour Sydney sur RainTomorrow

## 7 Prédition de la pluie à un horizon de temps

### 7.1 Objectif et méthodologie

Nous avons tenté jusqu’ici de prédire s’il pleuvra à  $J+1$ . Voyons voir maintenant s’il est possible de prévoir la pluie sur davantage de jours dans le futur.

Pour cela, nous allons créer de nouvelles variables cibles, nommées  $Rain\_J\_h$ , où  $h$  est le nombre de jours dans le futur et dont  $Rain\_J\_h$  vaudra 1 s’il pleuvra dans  $h$  jours et 0 sinon.

Afin d’obtenir ces variables, nous partons de  $RainToday$ , à laquelle nous allons appliquer des shifts successifs à partir de notre Dataframe trié chronologiquement, après avoir rempli les dates manquantes avec des données vierges. Ce décalage est appliqué pour chaque *Location*, et non sur le dataframe globale pour que le shift n’impute pas la valeur de la dernière date d’une *Location* sur la valeur de la première date d’une autre *Location*.

$Rain\_J\_1$  est donc égale à  $RainTomorrow$ ,  $Rain\_J\_2$  indique s’il pleuvra après-demain, etc. Les taux de répartition entre les classes restent donc identiques.

Notre approche va consister à entraîner spécifiquement un modèle pour chaque variable cible  $Rain\_J\_h$ . Dans les graphes qui suivront, lorsque nous regardons l’évolution des qualités de prédiction suivant le nombre de jours dans le futur prédit, il s’agira donc d’autant de modèles qu’il y a de jours distincts prédits. Nous utiliserons des XGBoost avec les mêmes hyperparamètres que vus précédemment, en fonction de la granularité micro, macro ou climatique.

### 7.2 Limite théorique

Les sites de prévision météorologique proposent des prédictions maximales sur 2 semaines. Nous nous attendons donc que la qualité des prédictions se réduisent progressivement jusqu’à ne plus pouvoir se

distinguer du hasard avant cette limite théorique de 15 jours. Observons ici l'accuracy, le recall et l'AUC-ROC pour des modèles entraînés sur l'ensemble de l'Australie. Dans la Figure 71, nous avons conservé le seuil par défaut de 0,5. Dans la Figure 72, nous avons pris le seuil optimal au regard de la courbe ROC. Regardons comment les métriques évoluent en fonction du nombre de journées dans le futur de prédictions :

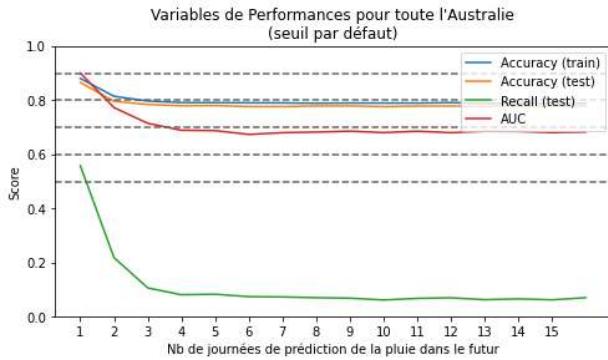


Figure 71: Scores du modèle macro, seuil par défaut

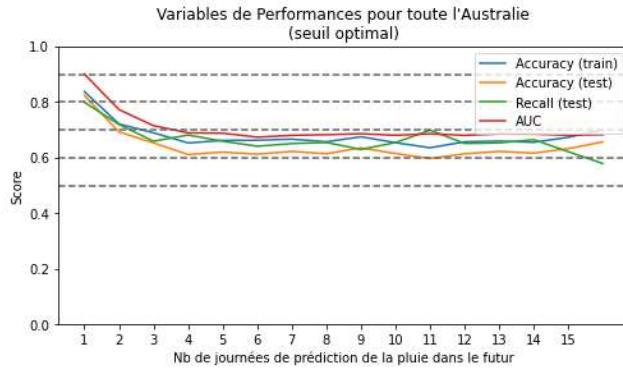


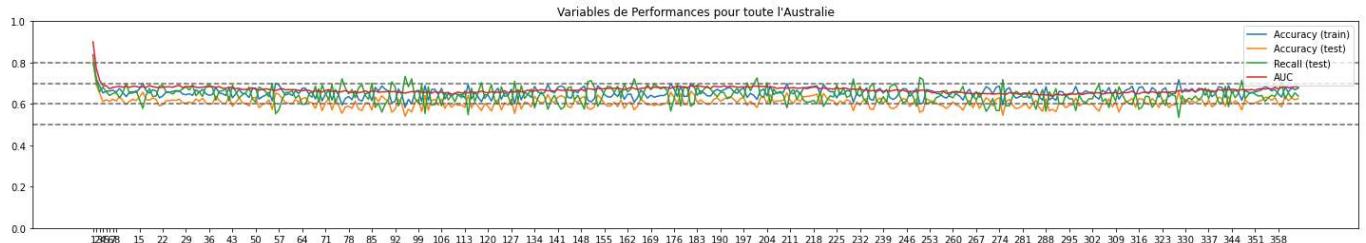
Figure 72 : Scores du modèle macro, seuil optimal

Plusieurs constats s'imposent :

- L'accuracy faiblit rapidement dans les deux cas les quatre premiers jours
- Elle converge vers une valeur légèrement inférieure à 0,8 avec le seuil par défaut (rappelons qu'il y a un taux de journées non pluvieuses de 0,77), et un peu supérieure à 0,6 avec le seuil optimal
- Le recall s'effondre à moins de 0,1 là aussi avec le seuil par défaut, ce qui fait supposer que les modèles ne doivent que rarement prédire de la pluie à partir de J+4
- Avec le seuil optimal, le recall reste supérieur à 0,6 et montre qu'on arrive encore à capter plus de 60% des journées pluvieuses à J+15 dans nos prédictions positives
- Enfin, et surtout, l'AUC ne converge pas vers 0,5, mais plutôt vers une valeur proche de 0,7 !

Ce dernier constat est particulièrement surprenant : si, comme nous l'avions supposé, la qualité des prédictions s'étaient dégradée jusqu'à devenir impossible, nous aurions dû avoir une AUC-ROC qui aurait dû se rapprocher de 0,5, de sorte à être comparable avec des prédictions aléatoires.

Plus surprenant encore : même en élargissant nos prédictions sur une année, l'AUC reste très au-dessus de 0,5. Nous représentons ci-après les métriques utilisant le seuil optimal de chaque modèle :



**Figure 73 : Métriques de performance de 360 modèles entraînés chacun à prédire la pluie à J+n**

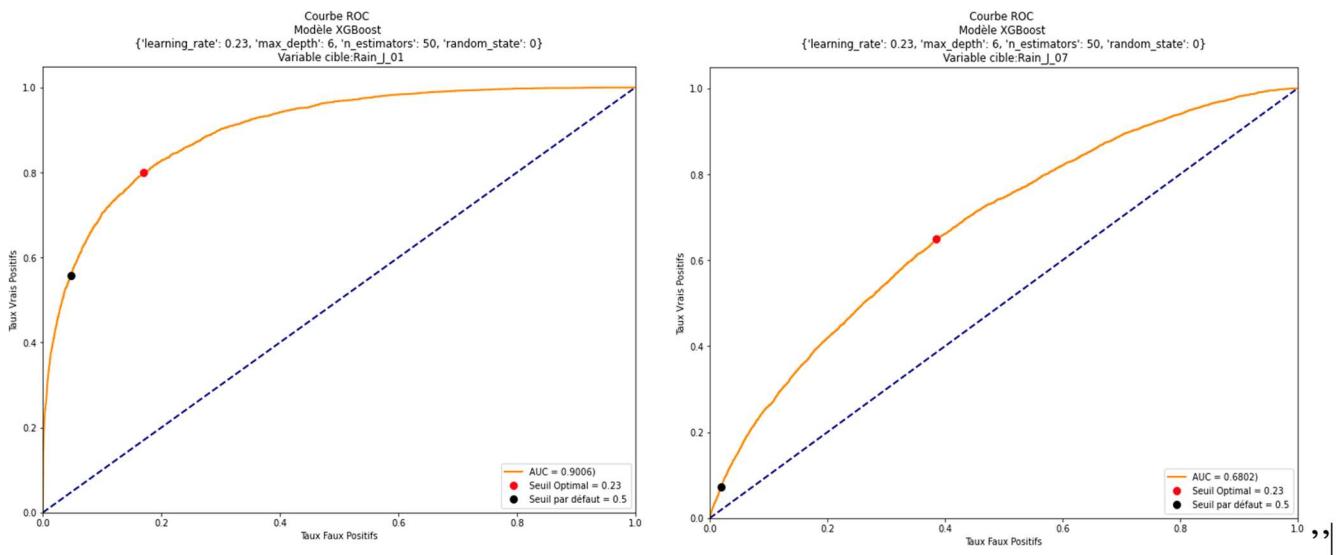
Bien entendu, l'accuracy étant d'environ 0,6 pour des prévisions dès J+4, nous avons bien conscience de la faible qualité de ces prédictions. Il n'empêche qu'il reste étonnant de constater que les seuls relevés météorologiques d'un jour donné permettent de prédire mieux que le hasard s'il pleuvra un certain nombre de journées dans le futur.

### 7.3 Comportement détaillé

Essayons de comprendre nos modèles. La courbe ROC de gauche illustre les résultats à J+1 (ce qui correspond donc à *RainTomorrow*). La courbe de droite trace les résultats à J+7. Les différences entre ces deux courbes sont représentatives de l'évolution dans le temps :

- la courbe ROC s'aplatit rapidement les 4 premiers jours
- le point noir, représentant le seuil par défaut, se rapproche de l'origine
- le point rouge, représentant le seuil optimal, reste au centre de la courbe

Ces simples visualisations corroborent les métriques vues plus haut : en conservant le seuil par défaut, nous allons très vite prédire presque systématiquement qu'il ne pleuvra pas. En revanche, le seuil optimal nous permet d'améliorer d'une façon frappante le taux de vrais positifs.

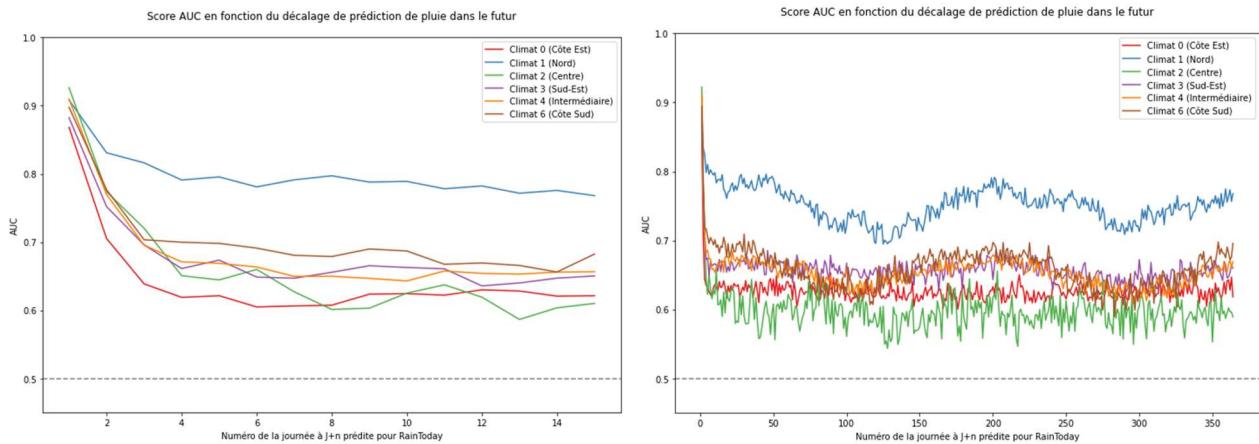


**Tableau 19 : Les courbes ROC du modèle XGBoost qui prédit la pluie du lendemain (gauche) et la pluie dans 7 jours (droite)**

### 7.4 Analyse par zone climatique

Le graphique de gauche ci-dessous nous présente l'évolution de l'AUC-ROC pour chaque zone climatique pour les 15 premiers jours de prédiction. Le graphique de droite les représente sur une année.

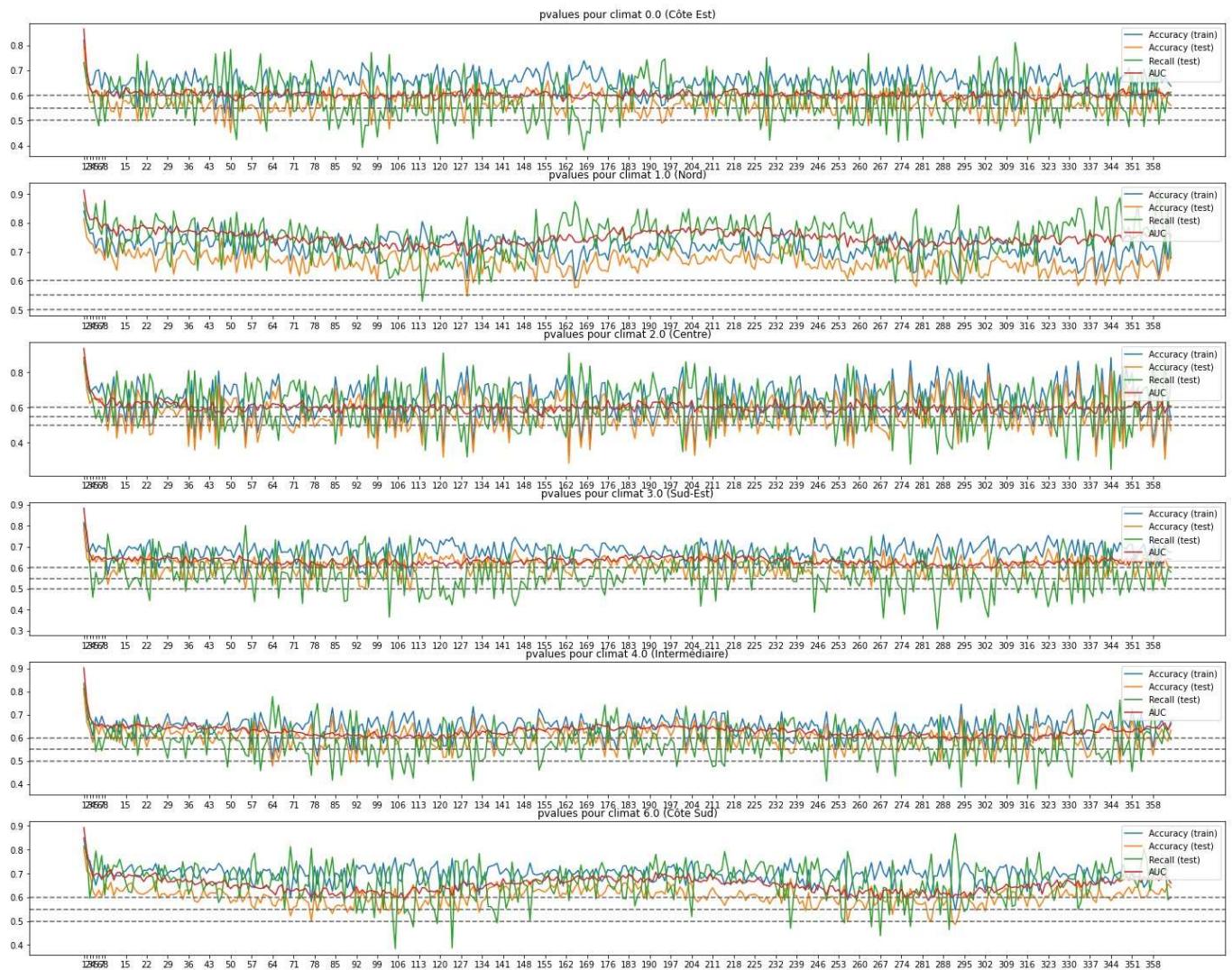
Sur les deux premières semaines, nous pouvons voir des différences dans l'évolution de l'AUC selon les zones climatiques. La zone septentrionale conserve un AUC proche de 0,8 après deux semaines, alors que la zone du désert central baisse plus rapidement vers 0,6.



**Tableau 20 : L'évolution de l'AUC-ROC pour chaque zone climatique pour les 15 premiers jours de prédiction (gauche) et pour une année (droite)**

Lorsqu'on observe les métriques sur une plage de prédiction d'un an, les différences de prévisibilité à long terme se confirment selon les zones climatiques. Notez qu'on remarque une saisonnalité de l'AUC. Il s'agit d'un phénomène que nous avons remarqué de façon encore plus marquée pour la prévision de la température maximale dans le futur. Cette tendance saisonnière semble suivre une courbe ayant deux cycles au cours de l'année, c'est pourquoi nous avons ajouté une variable temporelle SaisonCos4pi.

Eclatons cette représentation sur une année pour chaque zone climatique, et enrichissons-là de l'accuracy (avec les données d'entraînement et de test, afin de repérer d'éventuels overfitting) et du recall :

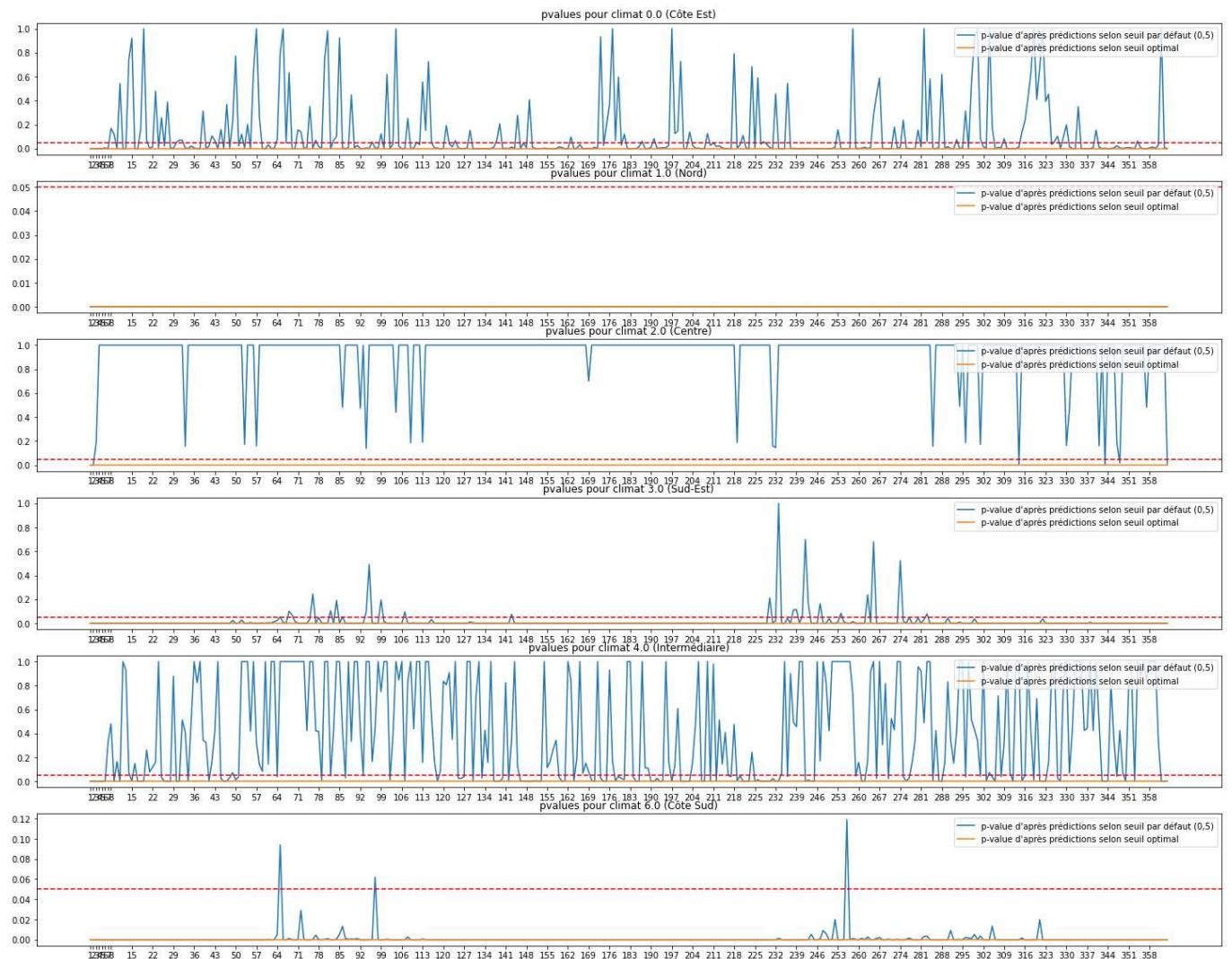


**Figure 74 : Métriques de 360 modèles prédisant la pluie à J+n pour chaque zone climatique**

Nous pouvons légitimement nous demander si ces modèles sont finalement plus performants que le hasard de façon systématique ou non. On effet, nous savons qu'un score AUC de 0,5 correspond à des prédictions aléatoires, mais est ce que nos scores souvent proches de 0,6 sont réellement pertinents ?

Pour éclairer ce point, nous allons réaliser des tests Chi<sup>2</sup> entre les valeurs observées et les valeurs prédictives de la variable cible, pour chaque journée de prédiction dans le futur, et nous allons afficher la pvalue obtenue. Notre hypothèse nulle est que nos prédictions ne sont pas corrélées avec la réalité.

Les courbes bleues dans la Figure 75 issues des prédictions avec le seuil par défaut, montrent que H0 ne peut souvent pas être rejetée, car étant supérieur au seuil de 0,05. En revanche, les courbes orange de prédictions tenant compte du seuil optimal sont systématiquement inférieures à 0,05. Les prédictions prenant en compte le seuil optimal prédisent donc véritablement mieux que le hasard d'une façon significative.

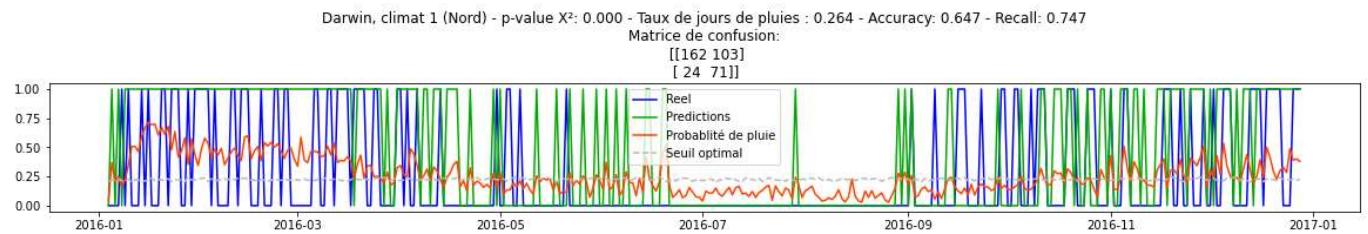


**Figure 75 : pvalue issues du test  $\chi^2$  de corrélation entre les prédictions de 360 modèles prédisant la pluie à  $J+n$  et les observations à ces dates, pour chaque zone climatique**

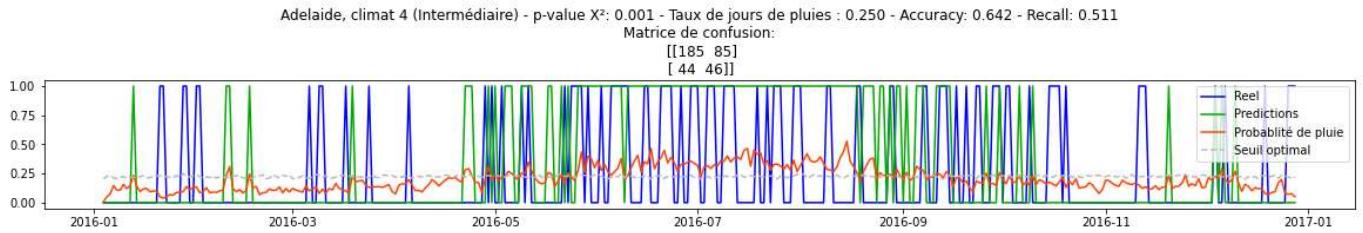
## 7.5 Prédictions

Puisqu'il semble possible de prédire s'il pleuvra ou non un an à l'avance, jetons-nous à l'eau sur quelques cas concrets.

Les graphiques Figure 76 - Figure 77 montrent les résultats de la prédiction d'une journée pluvieuse ou non sur l'ensemble de l'année 2016 avec comme unique observation les features du 1<sup>er</sup> janvier 2016 ! Les 360 modèles entraînés pour effectuer cette prédiction l'ont bien entendu été uniquement avec des données antérieures à 2016.



**Figure 76 : Prédictions de la pluie sur l'année 2016 pour Darwin à partir de la seule observation du 1<sup>er</sup> janvier 2016**



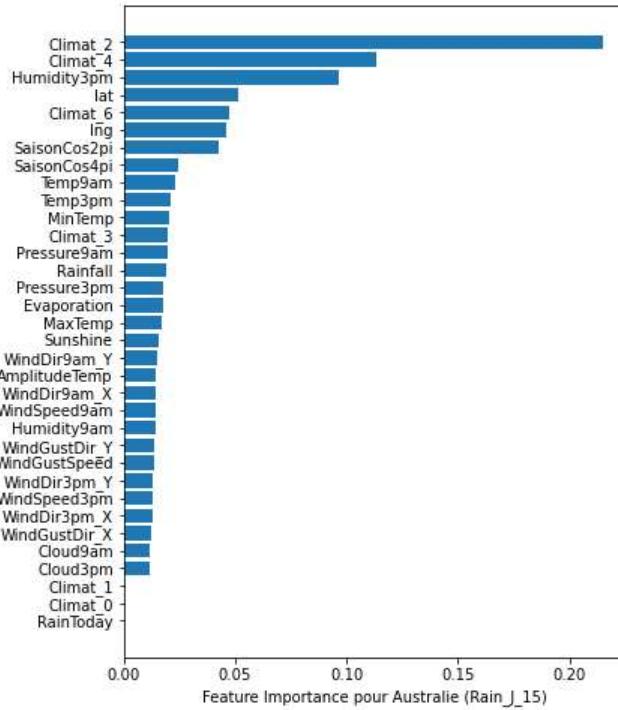
**Figure 77: Prédictions de la pluie sur l'année 2016 pour Adelaide à partir de la seule observation du 1<sup>er</sup> janvier 2016**

Nous avons retenu ici deux villes, Darwin et Adélaïde, situées dans des zones climatiques différentes. Dans les deux cas, on voit que le modèle prévoit qu'il pleuvra sur une vaste période au cours de laquelle les épisodes réellement pluvieux ont effectivement été nombreux (de janvier à avril pour Darwin, de juin à septembre pour Adélaïde). Les modèles semblent aussi avoir capté la saison sèche de Darwin de juillet à septembre. Bien sûr, ces modèles comprennent nos deux features de temporalité, SaisonCos2pi et SaisonCos4pi, mais même sans ces informations temporelles les modèles sont capables d'identifier ces tendances à partir des celles observations du 1<sup>er</sup> janvier.

Les courbes de pluie réelle et de prédiction sont cependant loin de se confondre, et de nombreuses erreurs sont visibles, ainsi que l'attestent les métriques indiquées dans le titre de ces deux graphiques.

## 7.6 Interprétabilité

Nous avons vu que pour *RainTomorrow*, les features les plus importantes étaient *Humidity3pm*, *Sunshine*, *Cloud3pm*, *WindGustSpeed*. Les features que nous avions ajoutées étaient exploitées de façon modérée par les modèles. La situation change considérablement lorsque nous regardons les variables exploitées par XGBoost pour prédire s'il pleuvra plusieurs jours dans le futur. Regardons ci-dessous les features importance sur un modèle entraîné sur toute l'Australie pour déterminer s'il pleuvra à J+15 :



**Figure 78: Feature Importance de la prévision de la pluie à J+15 avec un modèle macro**

L'appartenance aux zones climatiques 2 (Centre), 4 (Intermédiaire) et dans une moindre mesure 6 (Côte Sud) est une information importante. Nous retrouvons d'autres features que nous avons créées parmi les plus importantes, telles la latitude et la longitude, ainsi que nos deux variables de temporalité SaisonCos2pi et SaisonCos4pi. L'importance du feature engineering, qui était relativement modérée pour prédire *RainTomorrow*, semble bien plus marquée pour des prévisions plus lointaines.

De plus, si nous regardons cette fois les features exploitées sur un modèle micro (la Figure 79) nous verrons que des spécificités géographiques qui n'étaient pas déterminantes pour *RainTomorrow* deviennent particulièrement intéressantes ici. Sur ce modèle entraîné sur les données de Darwin uniquement pour prédire s'il pleuvra dans 100 jours, nous voyons par exemple que c'est *WindDir3pm\_Y* qui est la variable la plus importante. Or, Darwin est située tout au Nord de l'Australie, dans une zone touchée par la mousson : on comprend ici que connaître les vents venant du nord permet au modèle d'anticiper sur 100 jours si la mousson touchera la ville.

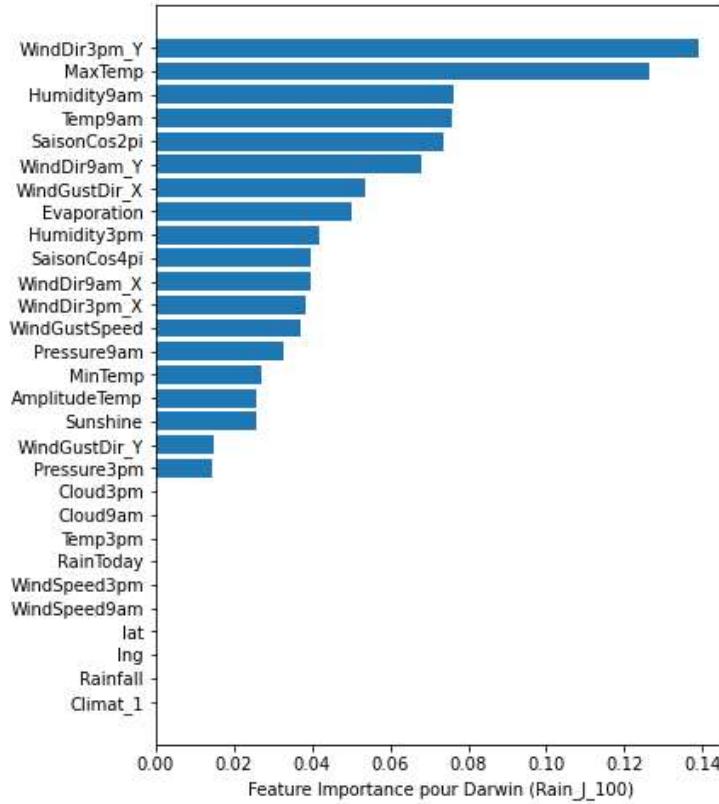


Figure 79 : Feature Importance de la prévision de la pluie à J+100 avec un modèle micro sur Darwin

## 7.7 Conclusions

Du fait des faibles scores sur l'accuracy, la prédiction plusieurs jours à l'avance avec des XGBoost n'est pas directement exploitable. Seules les prédictions jusqu'à J+3 sont exploitables en pratique. Cependant, savoir que les observations d'une journée donnée peuvent permettre de prédire s'il pleuvra un an à l'avance, même avec un taux d'erreur important, reste intéressant.

# 8 Prédiction de la variable *MaxTemp*

## 8.1 Présentation

Après nous être attelés à prévoir s'il pleuvra le lendemain, voire dans plusieurs jours, nous avons tenté de prévoir la température maximale du lendemain.

Nous avons choisi d'effectuer cette prédiction car il s'agit là d'une variable classiquement présentée dans les bulletins météorologiques, et elle présente l'intérêt d'être une variable non seulement continue mais saisonnière. Nous pourrons donc utiliser des approches de régressions et de séries temporelles.

## 8.2 Résultats de la régression par approches « classiques » via scikit-learn

Tout comme nous avions créé des variables *Rain\_J\_h* pour tenter de prédire s'il pleuvra dans h jours, nous avons créé dans notre dataset des variables *MaxTemp\_J\_h*. Regardons ici les performances obtenues par un XGBoost de regression pour prédire *MaxTemp*, c'est-à-dire prédire la température maximale de la journée à partir des autres features, mais aussi *MaxTemp\_J\_01*, c'est-à-dire la température maximale du lendemain à partir des features de la journée ainsi que la température dans 2 jours :

Prédiction des températures avec XGBoost		
	RMSE	MAE
MaxTemp	0,71	0,51
MaxTemp_J_01	2,65	1,93
MaxTemp_J_02	3,35	2,50

On voit qu'il y a beaucoup plus d'erreurs dans les prédictions de température du lendemain que dans celles de la journée même, et davantage encore dans celle du surlendemain. Comparons avec d'autres approches.

### 8.3 Séries Temporelles par SARIMAX

La courbe des températures présente une saisonnalité apparente. Nous pouvons donc tenter une modélisation monovariée avec SARIMAX.

Nous effectuerons là des analyses micro afin de tenter de prédire l'évolution des températures maximales pour une *Location* donnée. Prenons l'exemple de Mildura, qui a l'avantage de présenter une amplitude thermique assez importante au cours de l'année :

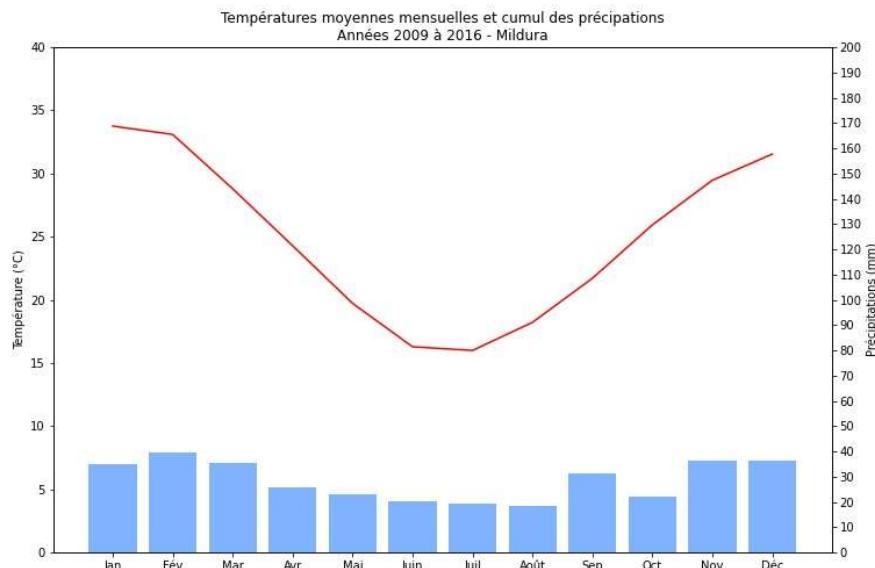
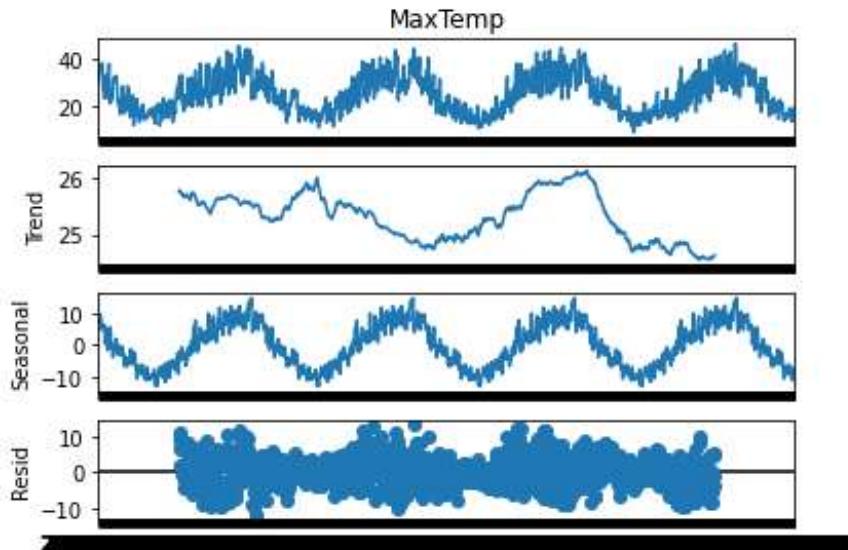


Figure 80 : Températures moyennes mensuelles et cumul des précipitations, années 2009 à 2016 - Mildura

La décomposition de la variable *MaxTemp* nous apprend plusieurs choses :

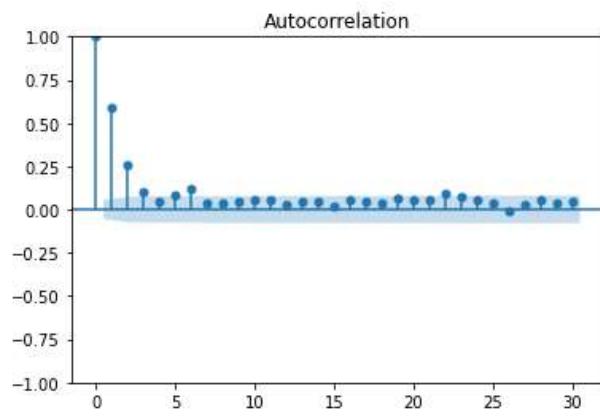
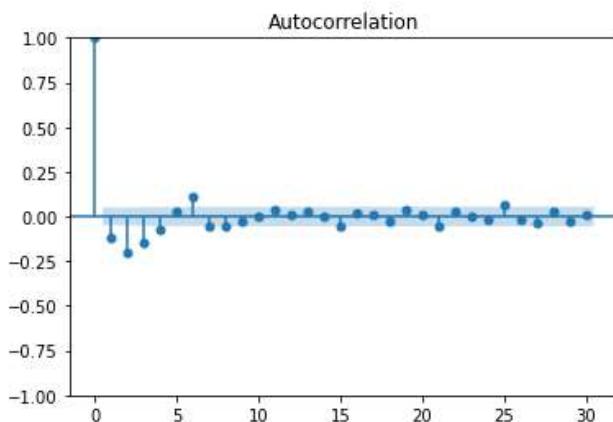
- Il existe bien, comme nous nous en doutions, une saisonnalité, qui explique des variations d'une amplitude de 20°
- La tendance reste dans une fourchette de variation d'un peu plus de 1° seulement
- Les résidus sont étalés sur une amplitude de 20°



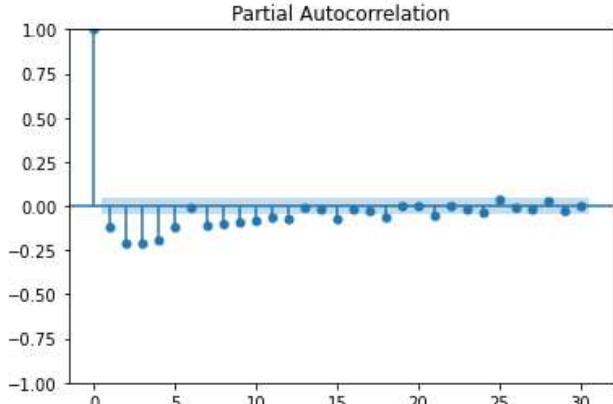
**Figure 81 : La décomposition de la variable *MaxTemp***

Si les deux premiers constats sont encourageants, le dernier l'est nettement moins : l'existence d'un bruit aussi important implique souvent des prévisions de qualité réduite.

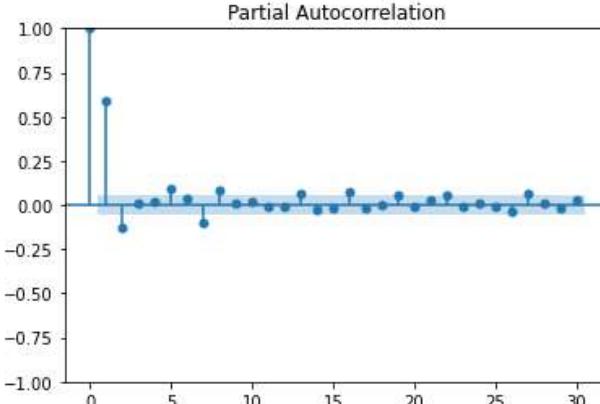
Passons aux décompositions ACF et PACF, sur la base d'une différenciation d'une journée d'une part et de 365 jours (=un an) d'autre part :



**Partial Autocorrelation**



**Partial Autocorrelation**



**Figure 82 : ACF et PACF – différenciation d'un journeé**

**Figure 83 : ACF et PACF – différenciation d'une année (=365 journées)**

La série est stationnaire après une différenciation d'une journée (graphes de gauche), de même que sur 365 journées (graphes de droite). Les décompositions ACF et PACF permettent de déterminer les valeurs max de  $(p,d,q)(P,D,Q)$  comme étant  $(5,1,4)(2,1,3)(365)$ .

Malheureusement, il nous a été impossible d'obtenir des résultats avec cette approche. En effet, quelle que soit la *Location* sélectionnée pour modéliser la variable *TempMax*, la modélisation n'était pas terminée après 12h de calculs, y compris lorsque nous avions restreint les données sur 4 ans seulement au lieu de 10.

## 8.4 Deep Learning

### 8.4.1 RNN

#### 8.4.1.1 Prédiction monovariée sur une journée

Nous avons développé ici un réseau récurrent afin de tenter de prédire la température d'une journée à l'aide de la température des jours précédents. L'analyse est ici monovariée.

Parmi les paramètres, le plus simple à déterminer a été le `batch_size` : dès lors qu'il était supérieur à 1, les performances étaient réduites. Nous avons donc retenu un `batch_size` de 1, malgré les temps plus importants.

La topologie du réseau de neurone est la suivante :

- Une première couche cachée de 30 neurones LSTM, avec une fonction d'activation ReLU
- Une seconde couche cachée de 10 neurones LSTM, avec une fonction d'activation ReLU
- Une couche dense de sortie de 1 neurone, sans fonction d'activation

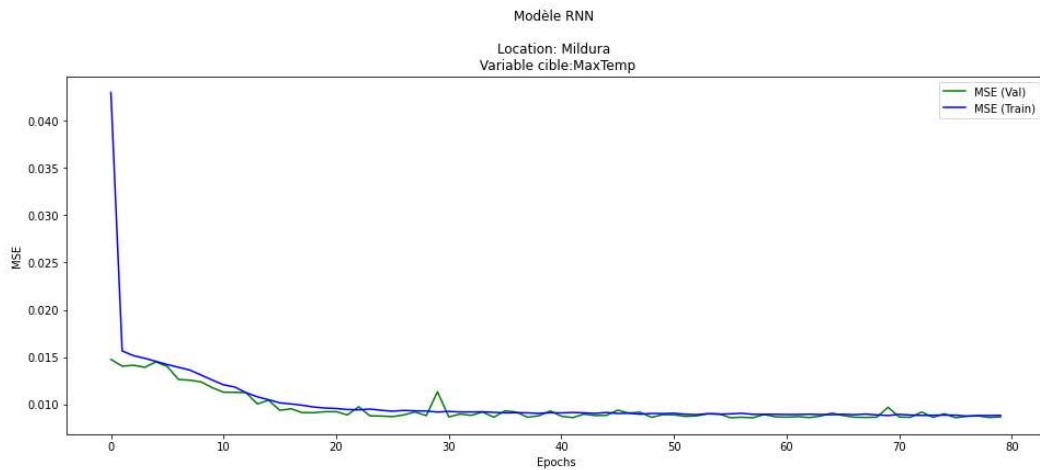
La fonction de perte est une MSE.

Pour savoir combien de jours dans le passé il convient de reprendre, regardons les performances obtenues en faisant varier cette valeur. Il semble qu'une fenêtre de 15 jours soit optimale.

<b>Impact du nombre de journées dans le passé</b>				
	<b>RMSE (train)</b>	<b>RMSE (valid)</b>	<b>MAE (train)</b>	<b>MAE (valid)</b>
30 jours	3,568	3,556	2,732	2,723
15 jours	3,507	3,500	2,679	2,669
7 jours	3,576	3,572	2,726	2,743
3 jours	3,841	3,894	2,948	3,019

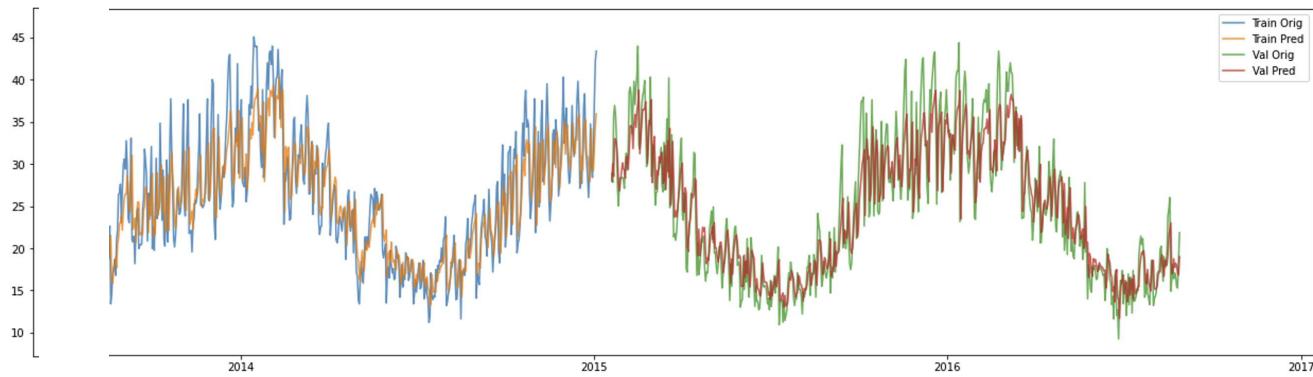
**Tableau 21 : Impact du nombre de journées dans le passé**

L'évolution de la fonction de perte se déroule correctement au fil des 80 époques d'entraînement :



**Figure 84 : Evolution de la loss (MSE) lors de l'apprentissage**

Les prédictions des températures sont assez proches des données réelles, bien sûr sur les dates d'entraînement (bleu) mais aussi de validation (vert). En revanche, on constate que le modèle reste chaque fois dans une fourchette plus réduite que la réalité. Il n'arrive jamais à prédire des pics de chaleurs ou des vagues de froid.

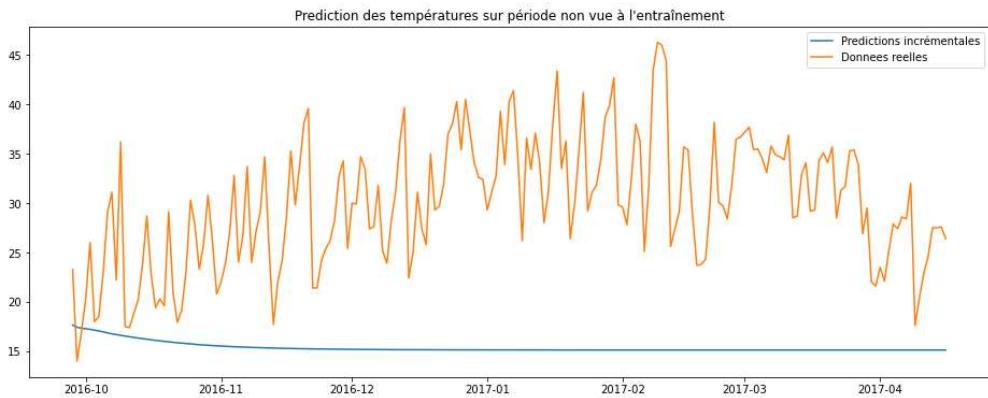


**Figure 85 : Comparaison des prédictions (sur train et validation) avec les données réelles**

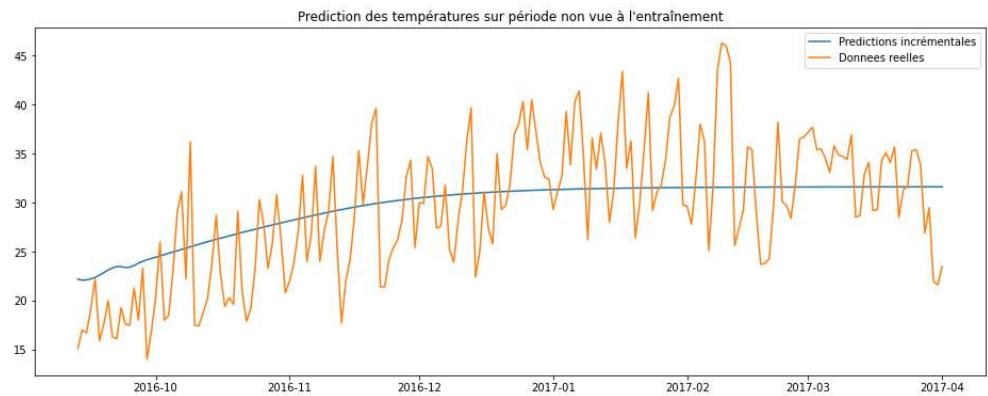
#### 8.4.1.2 Prédiction monovariée sur plusieurs jours

A partir du meilleur modèle entraîné, essayons maintenant de prédire de façon itérative la température maximale sur plusieurs jours de suite. Notre modèle a été entraîné sur des données de train et de validation. Nous allons ici effectuer les prédictions sur des données jamais vues par le modèle.

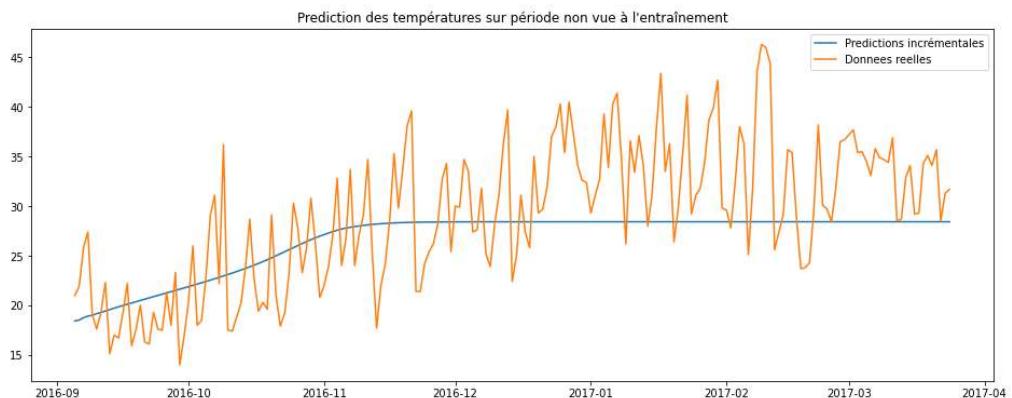
Nous avons vu plus haut qu'une fenêtre de 15 jours était optimale au regard des métriques observées. Comparons maintenant visuellement les conséquences sur le forecast des températures pour ces 4 plages de fenêtres :



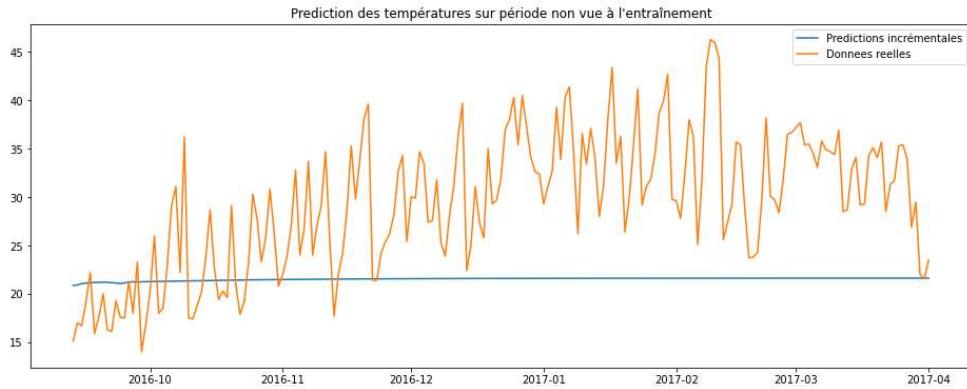
**Figure 86 : Fenêtre de 30 jours**



**Figure 87 : Fenêtre de 15 jours**



**Figure 88 : Fenêtre de 7 jours**



**Figure 89 : Fenêtre de 3 jours**

Le modèle avec une fenêtre de 30 jours ne part pas dans la bonne direction pour prédire les températures. A l'instar du modèle entraîné avec une fenêtre de 3 jours, il converge rapidement vers une valeur fixe dont il ne s'écarte plus. Les deux autres modèles finissent également par converger, mais sur une durée plus longue : environ 2 mois pour le modèle avec fenêtre de 7 jours et 3 mois pour le modèle d'une fenêtre de 15 jours.

Les prédictions présentent globalement assez peu d'intérêt : si la tendance générale est respectée par le modèle avec fenêtre de 15 jours, les variations soudaines d'une journée à l'autre ne sont en rien prédictes par le modèle. L'utilité d'un forecast des températures basé sur un modèle univarié semble donc avoir un intérêt limité.

#### 8.4.1.3 Prédiction multivariée

Maintenant que nous avons prédit MaxTemp en monovarié, nous allons écrire un modèle multivarié afin de bénéficier de l'apport de toutes les variables. Notre réseau aura une couche de plus et davantage de neurones que pour l'approche monovariée, dans la mesure où il y a par définition davantage de features en entrées :

- Une première couche cachée de 30 neurones LSTM, avec une fonction d'activation ReLU
- Une seconde couche cachée de 100 neurones LSTM, avec une fonction d'activation ReLU
- Une couche dense de 100 neurone, avec une fonction d'activation ReLU
- Une couche dense de sortie de 1 neurone, sans fonction d'activation

Contrairement à RainTomorrow, nous avons là une amélioration significative des performances avec l'approche multivariée. L'apprentissage se déroule mieux, et nous voyons sur les comparaisons de performances finales (toujours sur la ville de Mildura avec une fenêtre de 15 jours de données passées utilisées) un gain sur toutes les métriques. Nous pouvons également voir sur le graphique de comparaison des prédictions par rapport aux valeurs réelles que les prédictions sont maintenant bien plus étendues, alors que les prédictions en monovariées ne permettaient pas d'appréhender les valeurs très élevées ou très basses. Pour autant, nous restons loin des performances de XGBoost, qui permet d'obtenir une RMSE de 0,61 et une MAE de 0,47 pour Mildura.

### Comparaison mono/multivarié (Mildura, MaxTemp, 15 jours)

	RMSE (train)	RMSE (valid)	MAE (train)	MAE (valid)
Monovarié	3,507	3,500	2,679	2,669
Multivarié	2,815	2,900	2,1184	2,142

Tableau 22 : Comparaison des modèles RNN mono et multivariés

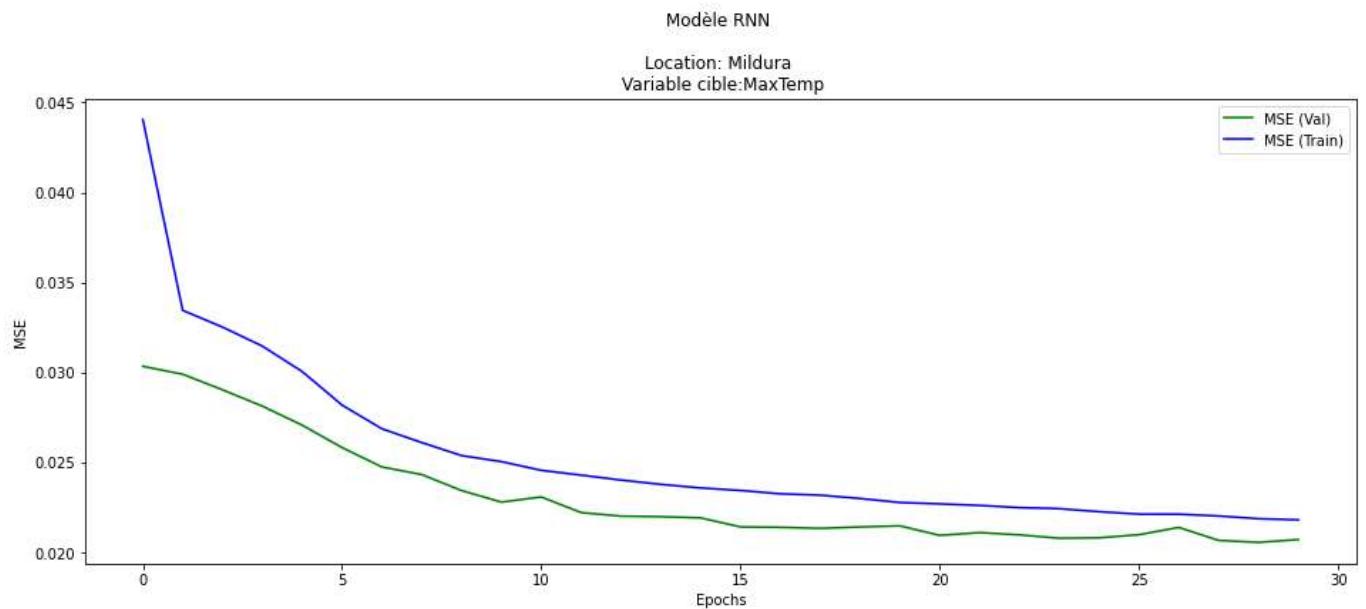


Figure 90 : Evolution de la loss (MSE) lors de l'apprentissage

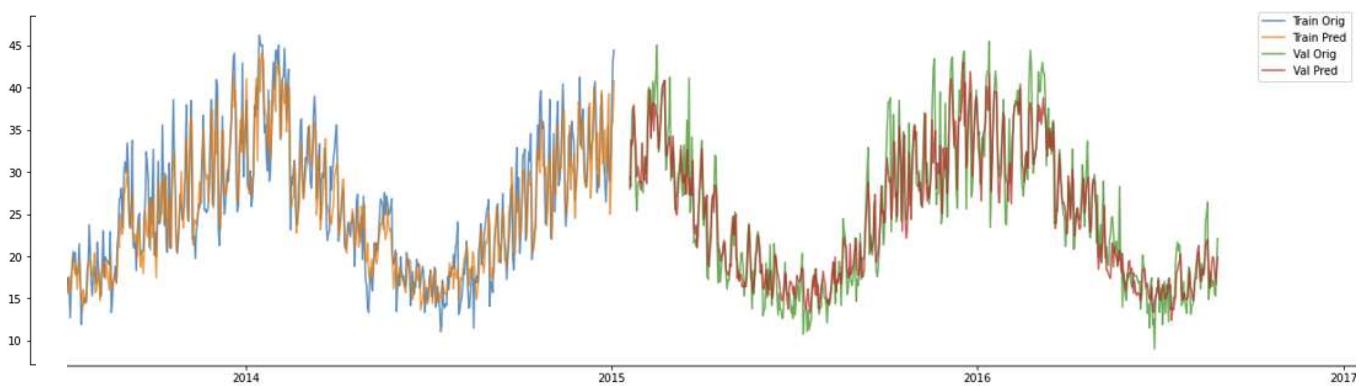


Figure 91 : Comparaison des prédictions (sur train et validation) avec les données réelles

## 9 Autres variables cibles

Il aurait été intéressant de pouvoir prédire *Rainfall*, laquelle nous aurait permis dans un second temps de déduire *RainTomorrow*. Cependant, la dispersion et l'irrégularité de cette variable font qu'il semble particulièrement complexe de prédire cette donnée. Par ailleurs, souvenons-nous que le site du Bureau of Meteorology indiquait que le niveau des pluviomètres n'était parfois pas relevé pendant quelques jours, et que le cumul des précipitations de ces journées était de ce fait rattaché à la prochaine journée de relève. Nous cumulons donc ici une problématique de distribution et un problème de récolte de données.

## 10 Conclusion

### 10.1 Constats

Nos modélisations nous ont réservé plusieurs surprises lors de nos explorations. La première d'entre elle a été la robustesse et l'intérêt du XGBoost, qu'il s'agisse de prédire *RainTomorrow* ou *MaxTemp* : pour notre problématique, ce modèle, très rapide à entraîner, rivalise y compris avec des réseaux de neurones bien plus complexes.

La seconde surprise a été que malgré le déséquilibre de nos classes sur la variable cible, la plupart des modèles entraînés nous ont offert de très belles performances.

La troisième surprise est l'apport limité du feature engineering dans notre problématique. Même s'il présente bel et bien un intérêt, en particulier pour effectuer des prédictions plus éloignées dans le futur, nous avons été particulièrement surpris par le faible écart de performances une modélisation effectuée avec les données initiales et celles obtenues après plusieurs mois de feature engineering. C'est une vraie leçon en termes de gestion de projet sur le temps à budgéter sur cet aspect. Il est en effet assez aisément de se perdre dans d'innombrables conjectures pour un intérêt potentiellement infime.

La quatrième surprise a été la possibilité de prédire la pluie sur une année à partir de l'observation d'une seule journée ! Encore une fois, les performances sont faibles, mais cette possibilité est particulièrement intrigante. Il s'agit là d'un aspect qu'il serait intéressant d'approfondir avec le regard d'un météorologue australien, qui pourrait potentiellement comprendre ce phénomène à partir de son expertise métier.

### 10.2 Limites et perspectives

Pour autant, nous restons avec une satisfaction mitigée sur les performances finales de notre meilleur modèle. Nous espérions en effet obtenir des prédictions quasiment fiables à 100%. Or, nous en sommes très loin avec notre accuracy de 86,6% sur l'ensemble de l'Australie, en particulier au regard du taux de journées non pluvieuses de 77,6%.

Nous avons toutefois pu identifier au fil du projet plusieurs pistes qui nous permettraient d'améliorer potentiellement les performances. Tout d'abord, nous avons vu dès la phase exploratoire que la feature *Sunshine* était absente sur environ la moitié des observations alors qu'elle présentait une forte corrélation avec la variable cible. L'importance de *Sunshine* a d'ailleurs été confirmée dans les analyses d'interprétabilité des différents modèles. Par conséquent, la première action à mener pourrait être de récolter les valeurs de *Sunshine* pour le maximum d'observations.

Deux autres variables sont très importantes : *Humidity3pm* et *Pressure3pm*. Une autre piste pourrait être de récolter le taux d'humidité et la pression atmosphérique à d'autres heures de la journée. Celles, existantes, de 9h du matin sont bien moins importantes, mais peut-être que les modèles gagneraient à ajouter ces taux à 14h ou 16h, par exemple. D'ailleurs, n'oublions pas que l'Australie s'étale sur plusieurs fuseaux horaires, allant de GMT+8 à GMT+11 : même si *Pressure3pm* est relevée à 15h pour chaque *Location* sur tout le pays, il ne s'agit en réalité pas de la même heure par rapport au soleil.

*Rainfall* est une variable dont la qualité de renseignement est discutable, alors même qu'elle est la source de notre variable cible *RainTomorrow* : contrairement aux autres variables, qui peuvent simplement être inexistantes pour une journée, *Rainfall* s'accumule en réalité dans le pluviomètre pendant plusieurs jours jusqu'à ce que le niveau soit relevé par un des bénévoles en charge. Il est probable que les modèles gagneraient en précision si les relevés étaient réellement quotidiens et, *a minima*, que les valeurs ne se cumulent pas en cas d'absence de relevé.

Au-delà des features elles-mêmes, il conviendrait aussi de disposer d'un dataset avec le plus de dates possibles renseignées. Pour rappel, il nous manque pour l'intégralité des lieux trois mois complets, et, pour certains lieux tels Melbourne, il manque plus d'un an et demi en cumulé sur la plage de dates. Ces trous sont de nature à perturber les modélisations par série temporelle. Il serait donc bénéfique de disposer de dates intégralement observées.

Notre dataset porte sur dix années, ce qui peut sembler conséquent, mais qui est en réalité assez peu à l'échelle des possibilités du machine learning. Il serait intéressant de pouvoir disposer de relevés sur une période de plusieurs décennies.

Etant donnée l'immensité de l'Australie, il pourrait être profitable de disposer de relevés d'autres stations météorologiques afin d'avoir d'une part plus de données, mais également un meilleur maillage géographique, qui permettrait peut-être de toutes nouvelles approches de prédictions basées sur les villes voisines.

Il serait aussi peut-être possible d'obtenir de meilleurs résultats avec les RNN en disposant de machines nettement plus puissantes que les nôtres afin de pouvoir multiplier les affinement d'hyperparamètres. Enfin, nous n'avons pas eu le temps d'explorer les transformer, qui pourraient potentiellement proposer des résultats encore meilleurs que les RNN.