**Weather in Australia**

Forage de données

Baptiste MARTEL, Antoine DUTEYRAT, Joshua MERCURE, Guillaume

# **Présentation des données**

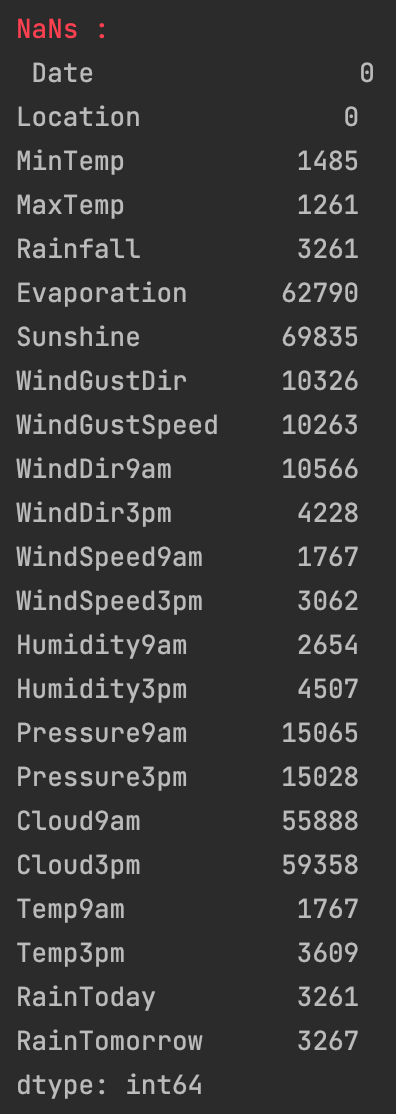
Pour mener à bien TP, nous avons choisi d’étudier un ensemble de données numériques basé sur le thème de la météorologie. Ces données, ci-dessous ont été collectées dans le but de prédire le temps du lendemain : pluvieux ou non, ce qui représente nos deux classes.

En enlevant notre attribut label du dataset, nous avons donc 22 variables distinctes qui nous aiderons à réaliser notre classification :

*Date, Location, MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustDir, WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm, RainToday, RainTomorrow.*

Nous avons pu observer un total de 112 925 instances : 87 906 appartenant à la classe où il ne pleut pas le lendemain et 25 019 appartenant à l’autre classe. Cela crée un déséquilibre de classe.

# **Vérification et prétraitement des données**

Une image contenant table

Description générée automatiquement

On constate sur la capture d’écran de gauche que beaucoup de valeurs manquantes sont présentes dans le dataset de départ. De plus, il faut résoudre le problème d’équilibre.

Nous avons donc effectué trois étapes pour arriver à un dataset exploitable :

* Nous avons d’abord supprimé les attributs que nous trouvions inutiles et/ou qui comportaient trop de données manquantes.
* Nous avons ensuite encodé nos données, car certaines d’entre elles n’étaient pas numériques de base (ex : Wind Direction). Cette étape a globalement rendu les valeurs catégoriques, non-catégoriques.
* Enfin, nous avons supprimé certaines données appartenant à la classe majoritaire pour arriver à un ratio de 50:50.

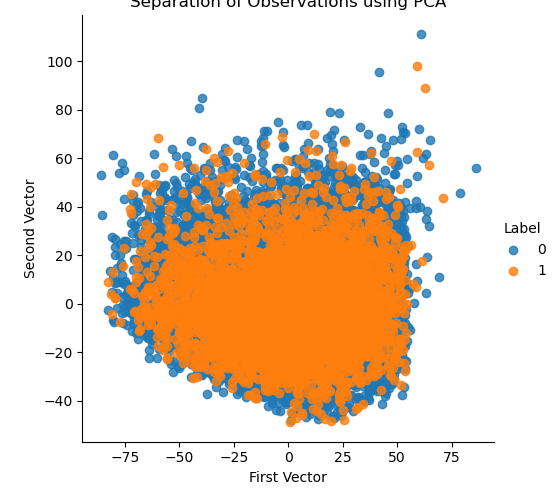
Après tout ce traitement, nous arrivons au dataset décrit dans la capture d’écran de droite.

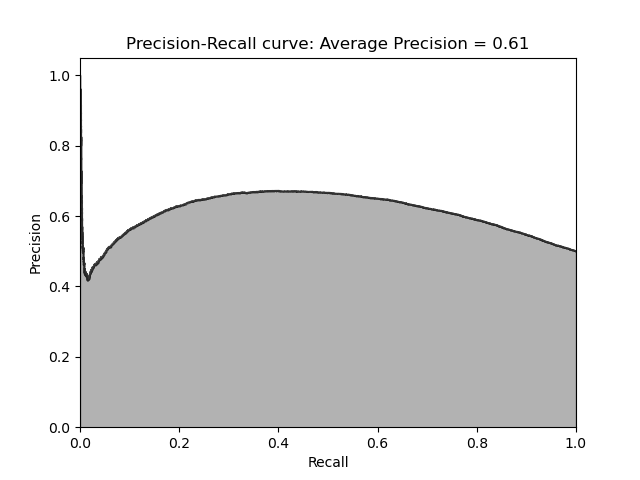
# **Phase de tests**

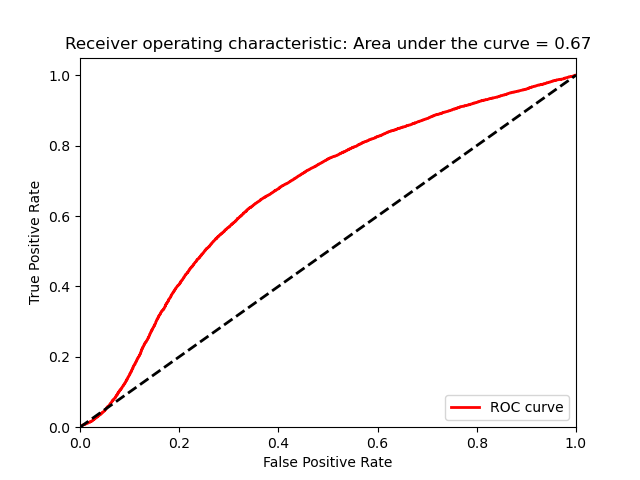
Voici ci-dessous les résultats des tests effectués pour tous les algorithmes à notre disposition. Les algorithmes sont les suivants : PCA, Sparse PCA, Dictionnaire, Fast ICA, Kernel PCA et Random Gaussian Projection.

Les résultats recueillis sont la répartition et la séparation des instances après la réduction de dimensionnalité, la précision globale de la reconstruction ainsi que le ratio vrai-positifs/faux-positifs.

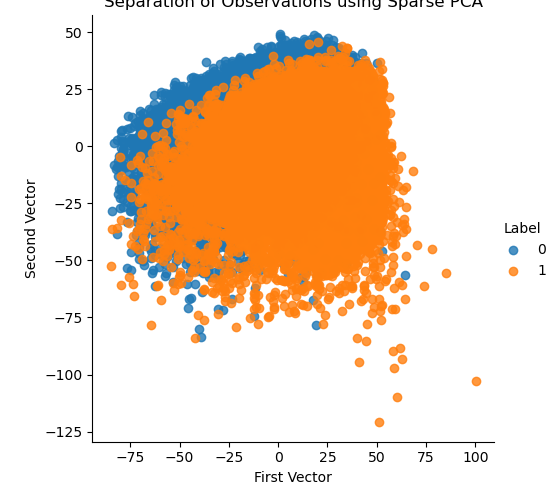
PCA :

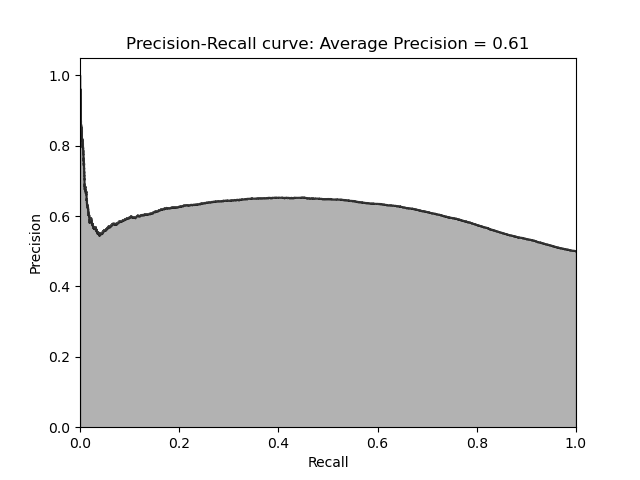


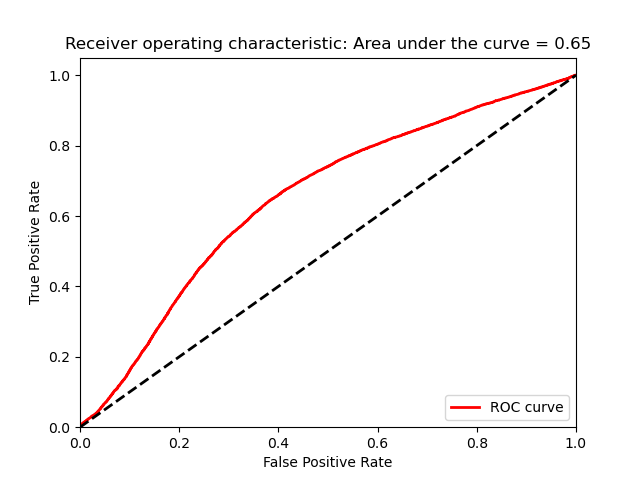




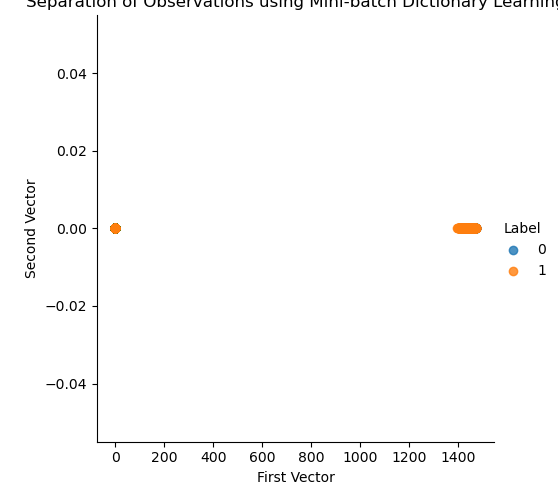
Sparse PCA :

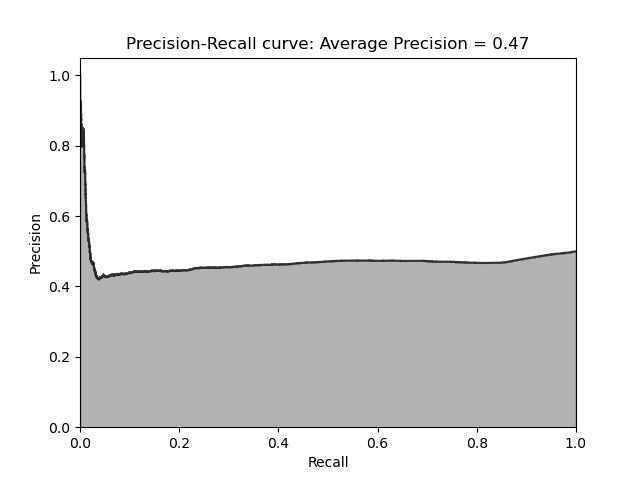


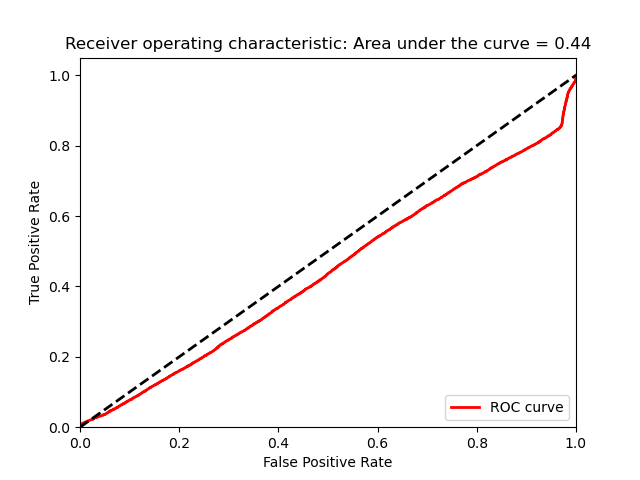




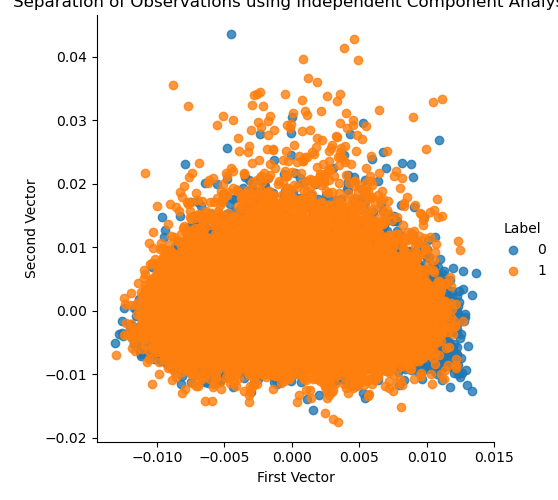
Dictionnaire :

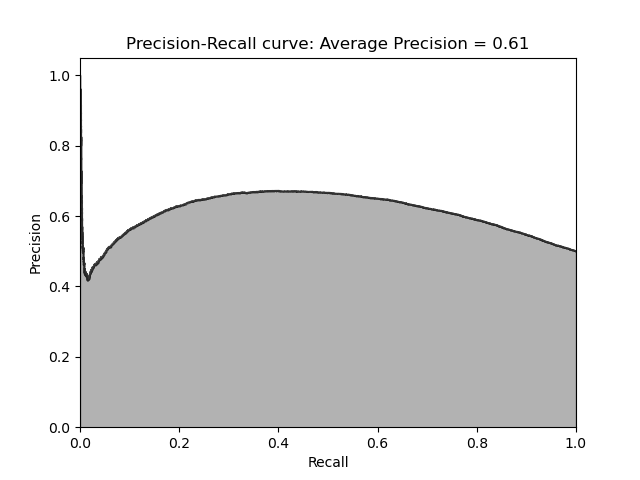


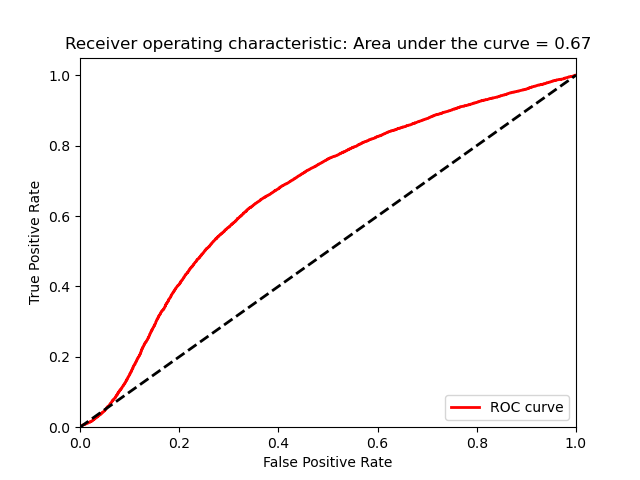




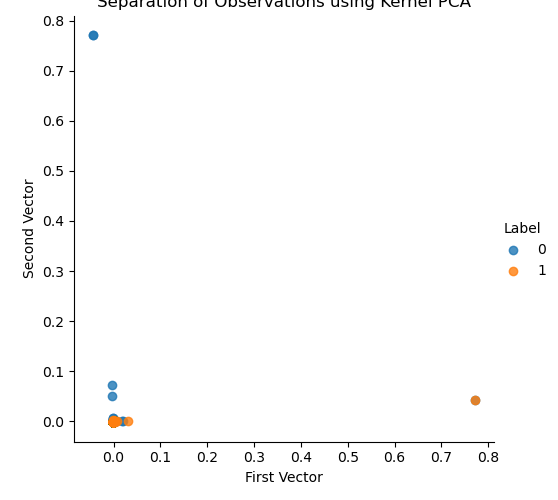
Fast-ICA :

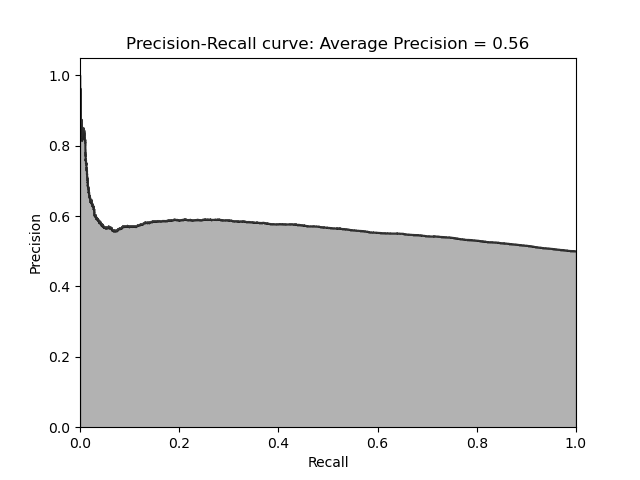


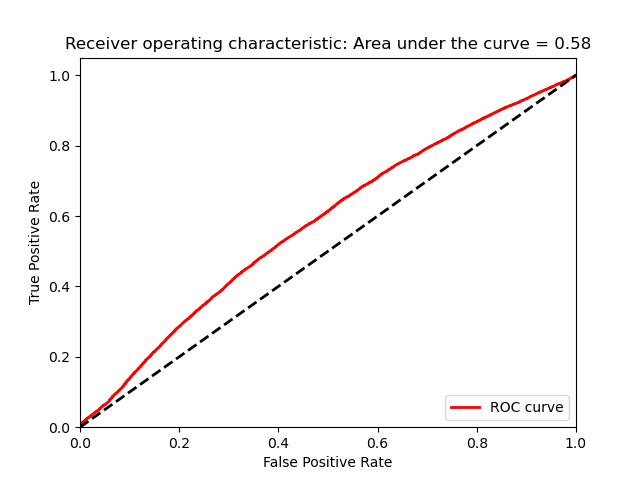




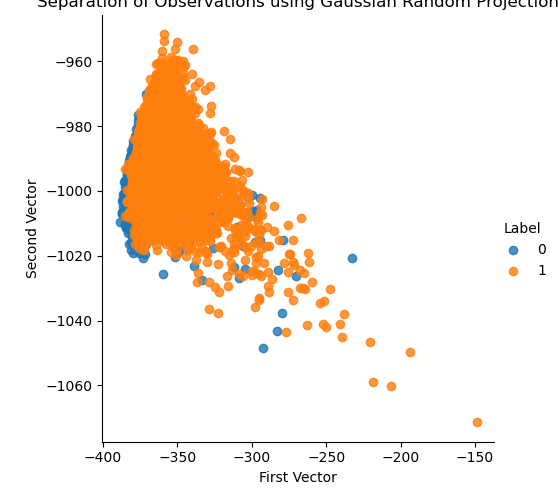
Kernel PCA :

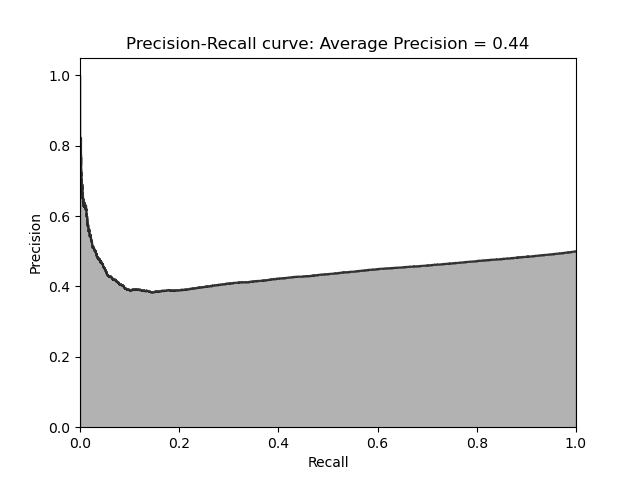






Gaussian-Random projection :

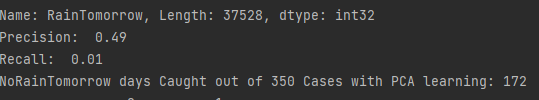




Après avoir réalisé les tests sur l’entièreté des algorithmes, on se rend compte que les résultats les plus prometteurs sont ceux du PCA standard, du sparse PCA et du fast ICA. En effet, les résultats laissent paraître qu’ils s’agirait, dans notre cas, des algorithmes les plus précis.

Bien sûr, dans le dataset présent, le rapport d’anomalie n’est pas aussi significatif que dans l’exemple du tutoriel car le dataset n’est pas débalancé (de par son rééquilibrage du départ), il n’y a donc pas de différence marquée dans la difficulté de reconstruction des données.

Tout ceci nous donne pour l’instant les résultats suivants et ce pour l’entièreté des algorithme utilisé:



Nous continuerons donc l’étude avec les trois algorithmes les plus prometteurs pour l’instant.

Par ailleurs, pour obtenir de meilleurs résultats, nous allons essayer d’appliquer un modèle de classification basique au dataset, tel que le Decision Tree.

Après l’essai de 100 profondeurs d’arbres différentes, on trouve une précision maximale de 0.79 pour une profondeur de 8, comme en témoignent ces captures d’écran :





On obtient donc une précision bien plus significative en utilisant un modèle de classification en plus de la réduction de dimensionnalité.