**Projet “PyDiction” : Prédiction de la pluie en Australie par les modèles de Machine Learning**

**I- Contexte et Objectifs**

**Contexte** :

Equipe :

Richard Chazal et Nahim Boutemine, lors d’une formation pour compléter leurs compétences professionnelles continue en “Data Science” (RNCP, bloc 03 du titre 36129) avec l’organisme Data Scientest.

Les prédictions de pluie présentent un intérêt pour les organismes publics, les entreprises, ainsi que les citoyens. Les modèles de Machine Learning (ML) peuvent être utilisés pour différents types de prédictions climatiques, parmi d’autres applications.

**Objectifs :**

-objectif : construire un modèle de prédiction de la présence de pluie le lendemain (avec comme critère il pleut si la quantité de pluie est strictement supérieure à 1mm) sur n’importe quel point géographique du territoire australien à partir des données précédemment citées.

-objectif professionnel : valider la capacité : traiter un projet d’analyse des données et de modélisation grâce au langage Python et aux techniques de Data Analyse et de Data Science, en vue d’intégrer ces compétences dans nos activités.

**II- Préparation des données et méthodes employées**

Nous disposons d’un jeu de données de 49 stations de grandes villes australiennes réparties sur tout le territoire australien. Les données climatiques de températures, pression, vent, humidité, quantité de nuages et ensoleillement sont fournies pour différentes dates pour chaque station, pour un total de 145 460 entrées avant la préparation des données (**tableau 1**).

**Tableau 1 : type des variables et pourcentage de valeurs nulles.**

|  | type catégorielle/ numérique | pourcentage de valeurs nulles en % |
| --- | --- | --- |
| Date | catégorielle | 0 |
| Location | catégorielle | 0 |
| MintTemp | numérique | 1.020899 |
| MaxTemp | numérique | 0.866905 |
| Rainfall | numérique | 2.241853 |
| Evaporation | numérique | 43.166506 |
| Sunshine | numérique | 48.009762 |
| WindGustDir | catégorielle | 7.098859 |
| WindGustSpeed | numérique | 7.055548 |
| WinDir9am | catégorielle | 7.263853 |
| WindDir3pm | catégorielle | 2.906641 |
| WindSpeed9am | numérique | 1.214767 |
| WindSpeed3pm | numérique | 2.105046 |
| Humidity9am | numérique | 1.824557 |
| Humidity3pm | numérique | 3.098446 |
| Pressure9am | numérique | 10.356799 |
| Pressure3pm | numérique | 10.331363 |
| Cloud9am | numérique | 38.421559 |
| Cloud3pm | numérique | 40.807095 |
| Temp9am | numérique | 1.214767 |
| Temp3pm | numérique | 2.481094 |
| RainToday | catégorielle | 2.241853 |
| RainTomorrow | catégorielle | 2.245978 |

Une étude approfondie des données par station et un traitement algorithmique nous a permis de montrer que les données présentent des sauts de dates sur tout le jeu.

La méthode globale employée pour prédire la pluie à J+1 à partir des données des jours précédents (**tableau 1**) sur plusieurs années est celle des algorithmes de Machine Learning, comme il a été réalisé dans divers articles scientifiques.

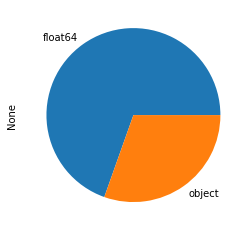
**Modélisation par ML, première étape, la préparation des données :**

Les algorithmes de Machine Learning (notés ML par la suite) demandent généralement une préparation des données. Ainsi les étapes chronologiques de préparation des données avant toute modélisation et les choix effectués pour notre étude spécifiquement sont indiqués ci-dessous. Ces choix sont ceux à utiliser pour tout traitement de nouvelles données de météo avant d’appliquer les modèles sur un dataset similaires selon notre expérience, pour les meilleurs modèles que nous avons retenus (voir partie résultats) :

1. **Valeurs manquantes** : les algorithmes n’acceptant pas ces dernières, il faut donc les supprimer.

Nous posons l’hypothèse que le traitement par une suppression de toutes les lignes contenant au moins une valeur manquante amène à des résultats de prédictions satisfaisants (70%) par les modèles de ML, au regard de la quantité de données restantes après suppression (58 090).

1. **Traitement des données catégorielles** :

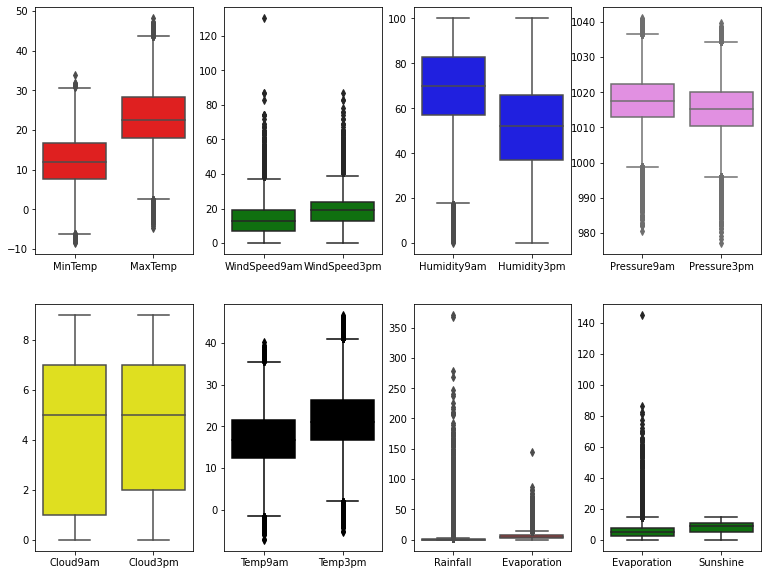


**Figure 1 : Pourcentage de données catégorielles (en orange).** En orange, les données catégorielles.

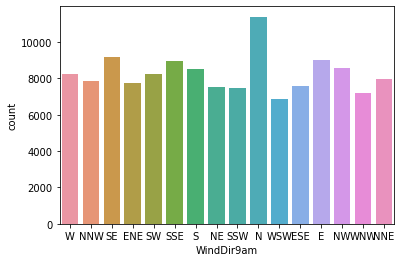
Nous remarquons un nombre conséquent de valeurs catégorielles. Leur traitement demande un encodage et parfois des traitements particuliers ; ici ‘Date’ a été séparée en (‘An’, ‘Mois’, ‘Jour’) pour déterminer leur effet et encoder en même temps.

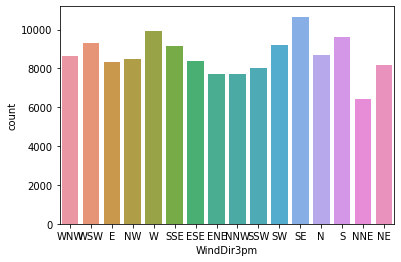
Les autres variables catégorielles ont été encodées directement, sans séparer en créant des booléennes afin de ne pas augmenter la dimensionnalité du jeu de données, en posant l’hypothèse que ce traitement serait le plus adapté et le plus rapide.

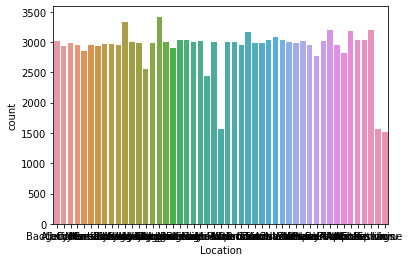
1. **Enlever les doublons** : les données en double n’apportent pas d’informations supplémentaires, il faut donc les supprimer. Il n’y en avait pas dans le dataset.
2. **Repérage et corrections éventuelles des valeurs extrêmes et des variables trop peu “gaussiennes”** : les valeurs extrêmes peuvent nuire à la précision des modèles de ML car ils sont parfois sensibles à ses paramètres. Il en est de même pour les distributions des variables peu gaussiennes. Les valeurs extrêmes et les distributions gaussiennes sont visibles dans les **figures 2 et 3**.



**Figure 2 : Distribution des variables numériques**



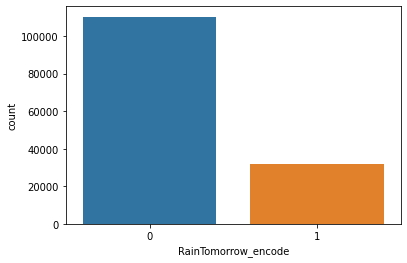




**Figure 3 : Distribution des variables catégorielles.**

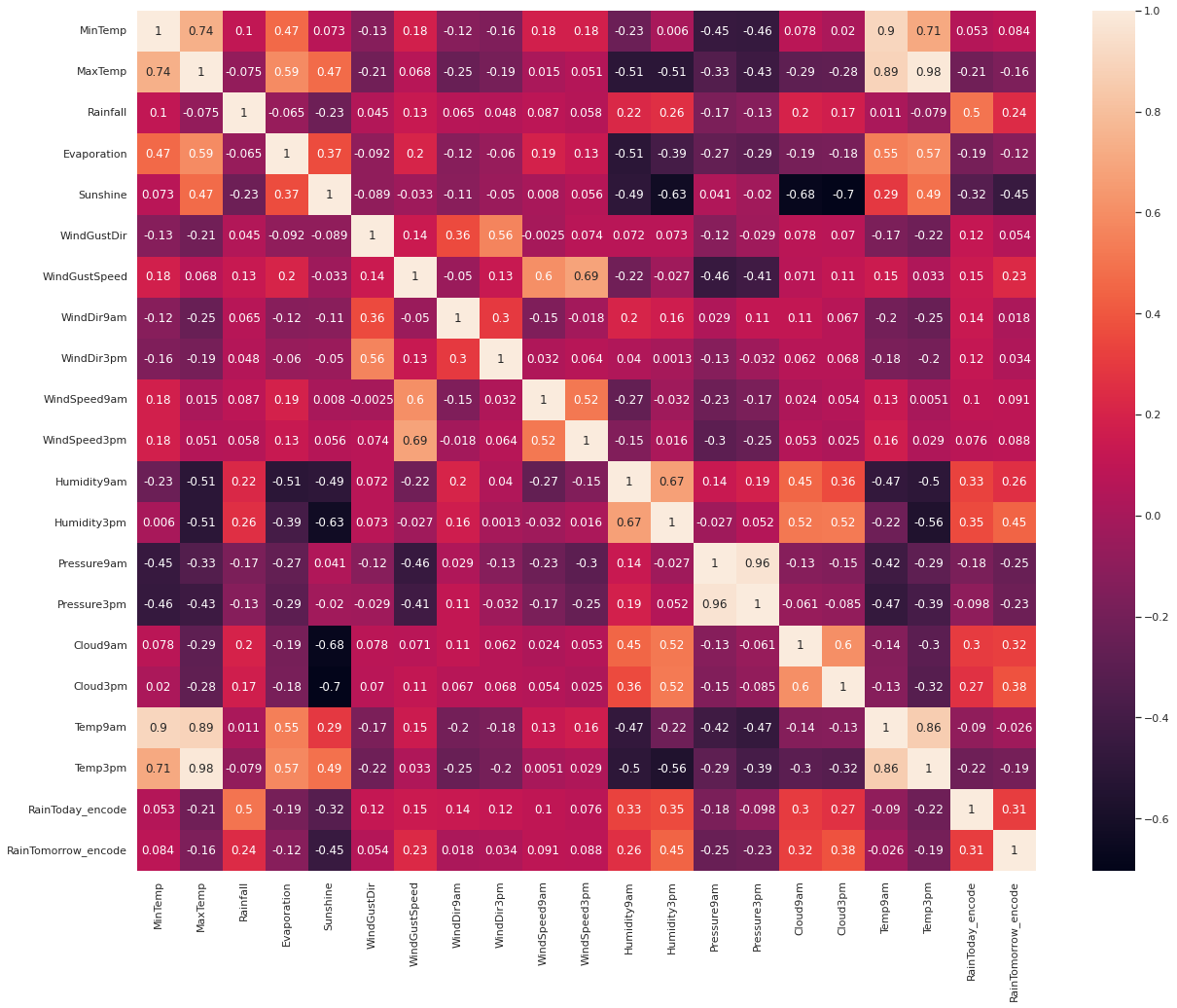
Les données étant des données météorologiques, et le modèle devant pouvoir prédire la pluie avec les nouvelles valeurs extrêmes qui viendront à l’avenir selon les experts climatiques, nous choisissons donc de les conserver. Cette hypothèse sera vérifiée par la suite.

1. **Suppression des variables constantes ou possédant trop de données manquantes** : il n’y a pas de variable constante, de même au regard du nombre de données manquantes (toutes ont un nombre de données manquantes inférieur à 50%, le seuil choisi).
2. **Repérage des déséquilibres de répartition dans la variable cible et corrections éventuelles (rééchantillonnage)** : La répartition des données dans la variable cible RainTomorrow étant déséquilibrées (**figure 4**) (environ un quart des données dans une catégorie contre environ trois quart dans l’autre), nous avons posé l’hypothèse d’un gain de précision grâce aux méthodes de rééchantillonnage.



**Figure 4 : Répartition des valeurs dans les catégories - variable cible.**

1. **Tri des variables explicatives trop peu corrélées à la variable cible:**



**Figure 5 : Heatmap du jeu de données (test de Pearson)**

Nous supprimons les variables non significativement corrélées à la cible soit ayant une p-value de Pearson supérieur à 5% (voir la liste ci-dessous) et dont la corrélation est inférieure à 5% (**figure 5**). Ce tri nous indique qu’il faut conserver les variables suivantes : Sunshine, Evaporation, Cloud3pm, Cloud9am, Pressure9am, Pressure3pm, WindGustSpeed, Humidity3pm, Temp3pm, WindSpeed3pm, Humidity9am, Rainfall , WindSpeed9am, MinTemp, MaxTemp, WindGustDir, RainToday\_encode, RainTomorrow\_encode. A noter, ces résultats ont été confirmés par des tests bivariés ANOVA et Cramer.

**Données de p-value :**

-Sunshine : 47.692924

-Evaporation : 42.789026

-Cloud3pm : 40.152469

-Cloud9am : 37.735332

-Pressure9am : 9.855619

-Pressure3pm : 9.832411

-WindGustSpeed : 6.519308

-Humidity3pm : 2.538803

-Temp3pm : 1.917113

-WindSpeed3pm : 1.849599

-Humidity9am : 1.247600

-Rainfall : 0.988797

-WindSpeed9am : 0.948007

-MinTemp : 0.447983

-MaxTemp : 0.226453

-WindGustDir : 0.000000

-RainToday\_encode : 0.000000

-RainTomorrow\_encode : 0.000000

1. **Tri des variables par d’autres méthodes :**

Nous posons l’hypothèse que les méthodes de tri des variables par ACP, sélection k-best et … n’apportent pas d’amélioration des résultats finaux et nous avons décidé de vérifier cela en dernier lieu si le temps nous le permettait. Nous l’avons au final eu et présenté en dernier dans ce rapport au vu de l’ordre chronologique.

**III- Modélisation : test, évaluation et sélection de modèles de ML :**

**III-1) Choix des modèles**

Nous décidons de tester les modèles suivants en se basant premièrement sur la méthode de scikit-learn. Le temps de calcul des modèles est pris en compte en deuxième paramètre de choix.

-1) SVM linéaire car nous avons moins de 100 000 données, notre variable cible est de type booléenne (0 et 1).

-2) KNN : il vient ensuite selon la méthode scikit-learn. De plus, il est utilisable en météorologie selon nos sources.

-3) arbre de décision car c’est un modèle utilisable en météorologie et assez simple d’utilisation.

-4) régression logistique : adaptée car notre cible est catégorielle et ce modèle est potentiellement rapide.

-5) forêt aléatoire : une méthode ensembliste recommandée par la méthode scikitlearn au temps de calcul généralement performant.

**III-2) Indicateurs d’évaluation des modèles**

-score de précision (accuracy) : indique la proportion de bonnes prédictions, c'est donc un indicateur de la performance globale capable de comparer les modèles. Il sera calculé sur les jeux d'entraînement et de test pour évaluer l'overfitting et la précision globale des prédictions.

-Le f1score et matrice de confusion : la précision ne permet pas de comparer le classement correct des données positives (“il pleut aujourd’hui” dans notre cas) et négatives séparément, ce qui est nécessaire pour une analyse complète. Le f1-score permet cela, il intègre les informations de précision et de rappel (taux de données positives manquées parmi les données positives) puisqu’il est la moyenne harmonique de ces deux paramètres. Il augmente avec la performance de ces derniers (un f1-score de 1 indique une précision et un rappel parfaits).

La matrice de confusion permet d’avoir le compte des vrais et faux positifs et négatifs, selon leurs classes et de confirmer le f1-score.

-AUC : c'est l'aire sous la courbe ROC. Plus le score “AUC” est grand, mieux le modèle prédit les vrais positifs. Le meilleur modèle aura donc l'”AUC” la plus haute possible.

Interprétation classique du score AUC : entre 0.9 et 1 : très bons ; entre 0.8 et 0.9 : bon ; entre 0.7 et 0.8 : acceptable ; en dessous de 0.6 : trop juste.

-MAE : cet indicateur évalue la moyenne des distances absolues entre les données réelles et les prédictions. Il permet donc de comparer les modèles sur le taux d'erreurs de prédiction, et doit donc être le plus bas possible et donc de confirmer les précédents.

-Temps de calcul : il doit être le plus court possible et dans tous les cas inférieur à 1h pour pouvoir prédire assez rapidement RainTomorrow.

**III-3) Résultats de l’évaluation des modèles**

Premièrement, nous déterminerons pour chaque modèle les meilleurs hyperparamètres avant d’évaluer un modèle optimisé sur le jeu de données dont les manquantes ont été supprimées par dropna.

Dans un second temps, nous testerons différentes pistes de préparation des données pour confirmer nos hypothèses de travail précédentes. Nous ne ferons ce travail que sur le meilleur modèle sélectionné précédemment (si le temps le permet, nous vérifierons cette hypothèse en testant sur les autres modèles également l'intérêt de ces modifications).

Nous vérifions ensuite nos hypothèses de travail précédentes (réduction de dimension, rééchantilonnages, sélection de variables par seuil, normalisation, jeu de données initial avec fillna par la moyenne des variables) sur le meilleur modèle.

L'ensemble de ces résultats sont regroupés dans les **tableaux 2 et 3**.

**a) SVM linéaire :**

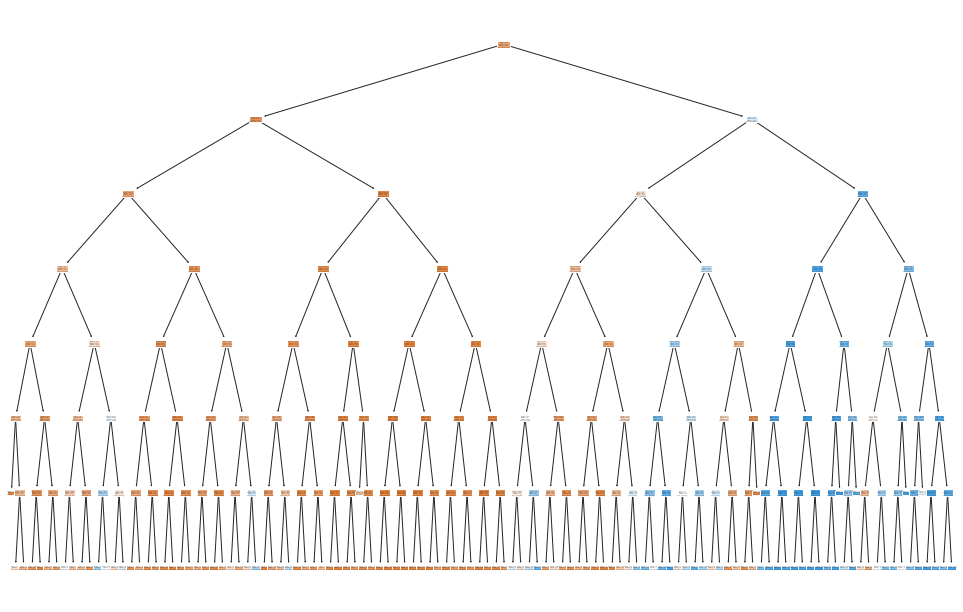
Avec ses meilleurs hyperparamètres (C = 1000, gamma = 0.001) , nous n'avons pas un temps de calcul du résultat (supérieur à 24H) satisfaisant. Ce modèle est donc écarté.

**b) KNN :**

Le calcul des hyperparamètres optimaux et du résultat prend environ 1H30. Les hyperparamètres sont : metric='manhattan', n\_neighbors=26, weights='distance'. Les résultats de prédiction sont visibles dans le **Tableau 2**.

**c) Arbre de décision :**

Les meilleurs hyperparamètres sont également recherchés par gridsearchCV : criterion = 'entropy', max\_depth = 7, min\_samples\_leaf = 40, random\_state = 123. La structure de l’arbre est en **figure 7**. Les performances que nous obtenons sont diminuées par rapport à ce que nous observions avec le KNN optimisé.

**Figure 7** : structure de l’arbre de décision.

**d ) Régression logistique :**

Les meilleurs hyperparamètres (par gridsearch) sont : C=0.01, penalty= 'l2'. Les performances sont sensiblement les mêmes que celles observées pour le KNN optimisé. Néanmoins, cet algorithme est le plus rapide car le temps de calcul (hyperparamétrage compris) est de quelques minutes.

**e) Forêt aléatoire :**

Les meilleurs hyperparamètres sont : max\_depth = 8, n\_estimators = 200, criterion = gini, max\_features = sqrt. Le temps de calcul de la gridsearchCV est de 2H30 environ.

Les performances sont en tous points similaires à celles du KNN, sauf l’AUC qui est dégradée.

**f) Bilan et choix du meilleur modèle :**

**Tableau 2** : comparaison des scores et temps de calcul des différents modèles pour la prédiction de RainTomorrow.

| modèle optimisé | SVM | KNN | arbre de décision | régression logistique | Forêt aléatoire |
| --- | --- | --- | --- | --- | --- |
| acc sur jeu test |  | 0,85 | 0,84 | 0,84 | 0.85 |
| f1 score au jeu test |  | 0,75 | 0,74 | 0,74 | 0.74 |
| MAE |  | 0,15 | 0,16 | 0,16 | 0.15 |
| AUC |  | 0,72 | 0,71 | 0,72 | 0.71 |
| temps GRIDSEARCH | trop long >24h --> éliminé | environ 45 minutes | environ 2h30 | quelques minutes | environ 2h30 |
| temps pour arriver au résultat | plus d'une heure | quelques minutes | quelques minutes | quelques secondes | quelques minutes |
| conclusion sur le meilleur modèle | le plus lent | le meilleur |  | le plus rapide |  |

Les modèles SVM et forêts aléatoires sont écartés en raison de leurs temps de calculs, en plus de n’apporter aucun gain de performances.

L’arbre de décision est également écarté au regard de son AUC et MAE dégradées par rapport aux autres modèles.

Les modèles régression logistique sont les deux meilleurs modèles et aux performances comparables, nous avons choisi le modèle KNN car il présente une performance légèrement supérieure.

Nous notons que tous les modèles classent moins bien la classe positive (matrice de confusion et f1-score), ce qui s’explique par le déséquilibre du jeu de données.

**III-4) Vérification des hypothèses de travail (rééchantillonnages, réductions de dimensions, sélection k-best) :**

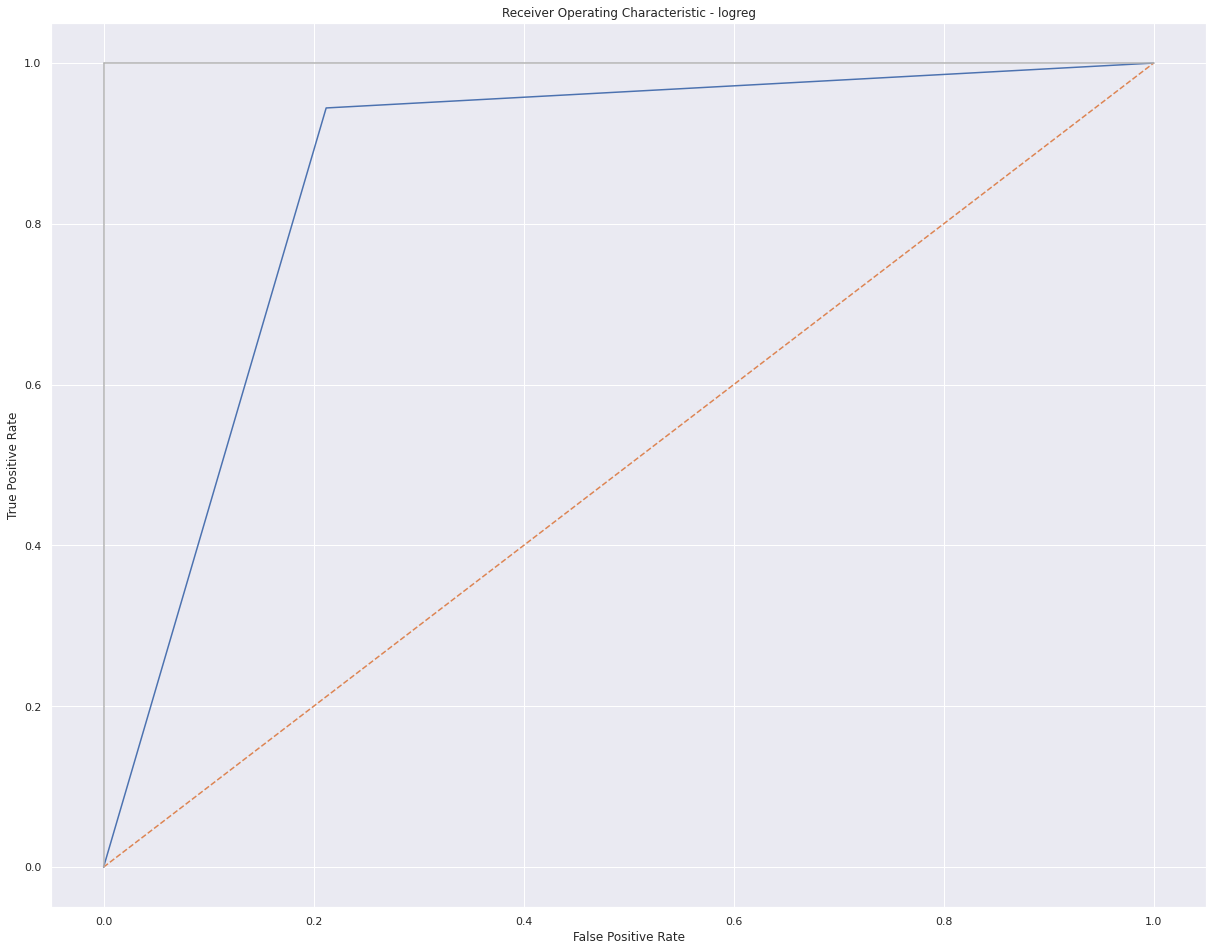
Si les performances en termes d’accuracy, de f1score, et de MAE sont acceptables, nous ne sommes pas satisfaits du classement des positifs. Dans l’optique d’améliorer ce classement, nous avons tenté différentes approches de manipulation du jeu de données.

L'ensemble de ces résultats est regroupé dans le **tableau 3**.

**a) Evaluation de l'intérêt du rééquilibrage du jeu de données par undersampling et oversampling SMOTE**

L’undersampling offre une amélioration significative du classement des vrais positifs qui est amélioré, au prix d’une dégradation de tous les autres indicateurs.

L’oversampling SMOTE permet d'améliorer les performances globales du modèle KNN car tous les indicateurs d'évaluation sont améliorés, y compris le classement des vrais positifs (**figure 8)**.



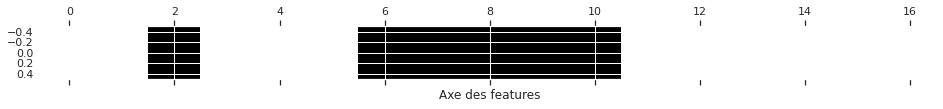
**Figure 8** **: courbe ROC du KNN après oversampling SMOTE.**

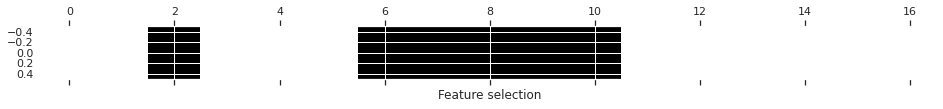
**b) Evaluation de l'intérêt de la réduction de dimension via acp :**

Nous retenons les 6 premières composantes, qui permettent d’expliquer 90% de la variance totale du jeu de données. Nous observons une stagnation voire une légère dégradation de tous les indicateurs de performance.

**c) Evaluation de l'intérêt du threshold via un VarianceThreshold :**

Afin d’évaluer cette méthode, nous avons choisi un threshold de 50, afin de retenir la moitié des variables (**figure 9**). Nous n’observons aucune différence de performance significative, excepté une dégradation de l’AUC.





**figure 9** : axe et fetaure selection.

**d) Evaluation de l'intérêt de la normalisation :**

Classiquement, les jeux de données numériques doivent être normalisés afin de s’assurer que tous les ordres de grandeur des variables n’altèrent pas le poids qu’elles auront dans le modèle.

Ainsi, nous avons réalisé une standard normal normalisation (appelée aussi normalisation vectorielle) sur le jeu de données des données dont les manquantes ont été supprimées par dropna. Nous n’observons aucune différence de performance significative, excepté une très faible dégradation de l’AUC.

**e) : Evaluation de l'intérêt du jeu de données traité par fillna généralisé au lieu de dropna généralisé :**

L'élimination de toutes les lignes ayant au moins une donnée manquante (dropna généralisé) a eu pour résultat une réduction de ⅔ du jeu de données, pour aboutir à un dataset réduit nommé “df\_bourrin”.

Afin d’étudier l’impact de ses lignes absentes sur la prédiction de RainTomorrow, nous avons décidé de tester le modèle avec le jeu complet. Les valeurs manquantes, au lieu d’être supprimées, sont remplacées par la moyenne, et ce pour chaque variable.

Non seulement nous observons une stagnation voire une légère dégradation de tous les indicateurs de performance, mais, en plus, nous augmentons drastiquement le temps de calcul du modèle KNN optimisé (1 minute avec df\_bourrin contre 1 heure environ avec df\_subtil\_2).

**f) Bilan :**

**Tableau 3** : comparaison des scores et temps de calcul des différents tests de resampling, réduction de dimensions, sélection des variables, normalisation, et prédiction à partir du jeu entier.

| **hypothèse testée** | **undersampling** | **oversampling SMOTE** | **acp** | **threshold** | **normalisation** | **jeu entier** |
| --- | --- | --- | --- | --- | --- | --- |
| accuracy sur jeu test | 0.79 | 0.87 | 0.84 | 0.84 | 0.85 | 0.85 |
| f1 score au jeu test | 0.79 | 0.87 | 0.74 | 0.73 | 0.74 | 0.74 |
| MAE | 0.21 | 0.13 | 0.15 | 0.16 | 0.15 | 0.15 |
| AUC | 0.79 | 0.87 | 0.71 | 0.70 | 0.71 | 0.71 |
| temps pour arriver au résultat | 1min | 1min | 1min | 1min | 1min | + de 30 minutes |
| conclusion sur le meilleur modèle |  | le meilleur |  |  |  |  |

L’undersampling augmente l’AUC, au prix de la dégradation des autres indicateurs. Elle est donc écartée.

l’ACP tout comme la sélection des variables par seuil ne permet aucun gain de performance. Ces hypothèses sont elles aussi écartées.

La normalisation n’apporte aucun gain de performances par rapport aux résultats du KNN voir tableau 2). De même pour le jeu de données traité par fillna généralisé, qui en plus dégrade fortement le temps de calcul de l’algorithme.

La seule hypothèse permettant une amélioration significative des performances est l'oversampling SMOTE.

Le traitement des données et le modèle optimal sont :

* encodage des catégorielles et filtre des variables à p-value de pearson > 5% et à corrélation de pearson <5%.
* dropna généralisé.
* séparation en jeu de données test et train,
* oversampling SMOTE.
* KNN optimisé par grid-search.

**Tentative de création et prédiction de RainIn3Days**

Dans l'optique d'enrichir les capacités de notre modèle, nous avons tenté de créer la variable RainIn3Days, en nous servant de Rain\_Today.

Pour cela, nous avons tenté de processer la variable de RainIn3Days à partir de RainToday, au moyen d’une boucle prenant en compte la date et la localisation (en effet, il y a plusieurs stations météos, et seulement 256 dates qui ne sont pas en doublons : cela signifie qu’il y a plusieurs bulletins météos émis le même jour par ces stations).

Malheureusement, nous n'avons pas pu mener cette étude à son terme par manque de temps notamment suite au départ d’une personne de l’équipe.

**IV- Conclusion :**

Nous avons pu sélectionner les variables les plus pertinentes grâce aux tests statistiques. Des modèles de classification simples offrent des performances similaires à celles offertes par des modèles ensemblistes. Au vu de la répartition de la population cible, un resampling par oversampling SMOTE est nécessaire et son efficacité a été montrée. Ainsi, nous confirmons notre capacité à prédire Rain-Tomorrow avec une marge d'erreur acceptable.

**Limites**

**Traitement des outliers** : nous n’avons pas vérifié l’hypothèse de l’élimination des outliers en entier, même si nous sommes convaincus que les résultats ne seraient pas améliorés par la suppression de ces extrêmes ainsi que par nos raisons de ce choix. La rigueur indique tout de même de vérifier cette hypothèse dans une étude ultérieure.

**Autres modèles :** Nous aurions pu utiliser d’autres modèles tels que les réseaux de neurones, les méthodes de séries temporelles. A deux personnes au lieu de trois et au vu des alternatives et du nombre de modèles testés, nous sommes satisfaits de la quantité de résultats. De même pour la pluie dans trois jours.





