Prédiction de la pollution de l'air à Londres

Projet Machine Learning pour la prédiction, sous la direction de Yannig Goude

Guillaume Staerman / Amaury Durand

Année universitaire 2017-2018

Table des matières

1 Introduction

Dans le cadre du projet de Machine Learning, nous nous sommes intéressés à la prédiction de la quantité des oxydes d'azote (Nox) présents dans l'air londonien en 2004. Les oxydes d'azote sont parmi les principaux polluants de l'atmosphère et proviennent essentiellement des combustibles fossiles, des moteurs à combustion interne, centrales thermiques, chauffages etc... Pour ce faire, nous disposons, comme base d'apprentissage, d'un certain nombre de variables explicatives (données météorologiques, données de trafic, etc) récupérées à différents lieux en 2002 et 2003.

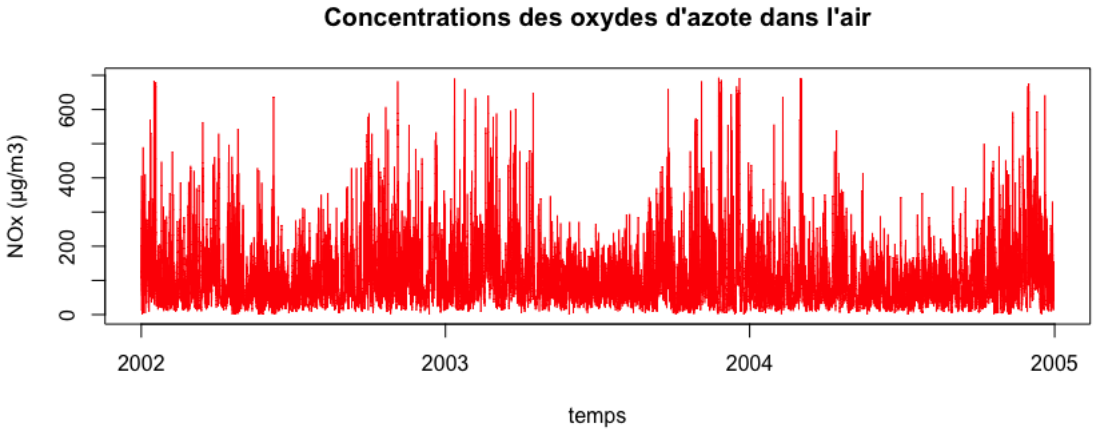
Nous avons mis en place des méthodes usuelles de régression et de prévision de séries temporelles faisant intervenir différents modèles : régression non paramétrique avec des GAMs, arbres de décision et forêts aléatoires, réseaux de neurones, modélisation de processus SARIMA ou lissage exponentiel. Certains modèles peuvent de plus être utilisés pour prévoir différentes composantes du signal : un modèle de régression permet par exemple de prévoir la moyenne tandis qu'un modèle de série temporelle (SARIMA ou lissage exponentiel) permet de prédire des variations plus fines. Enfin, nous explorons de plus des méthodes d'ensemble tels que le boosting et l'agrégation d'experts.

Nous proposons d'effectuer la prédiction de la quantité d'oxydes d'azote en 2004 à horizon d'une année, d'une semaine et d'une journée. Un horizon de prévision d'une journée signifie que l'on prédit chaque journée après avoir observé les précédentes. Les années de 2002 et 2003 représentent donc notre base d'apprentissage qui sera alimentée chaque jour par les données de 2004 pour prédire la quantité d'oxydes d'azote du lendemain. Pour cela, nous nous plaçons dans deux cadres différents. Le premier est un cadre théorique qui n'est pas possible en pratique ; il consiste à considérer que les variables explicatives (ou covariables) sont accessibles pour effectuer la prédiction du lendemain. En pratique, cela n'est pas possible et il est nécessaire soit d'obtenir des prévisions d'experts du domaine pour les variables explicatives, soit de prédire celles-ci avant de prédire la quantité de Nox. Cela constitue le second cadre pour lequel nous développons des modèles permettant de prédire les variables explicatives.

2 Les données

2.1 Présentation des données

Les quantités de Nox ($\mu g/m^3$) ont été mesurées toutes les heures dans le comté de Lewisham qui se situe au sud-est de centre du Londres. Ces mesures sont récupérées sur un site internet du gouvernement qui concentre de nombreuses données de pollution. Nous utilisons la fonction "*importAURN*" (Automatic Urban and Rural Network) directement implémentée sous R.



Nous récupérons ensuite des données de différents aéroports grâce à la fonction "*riem_measures*" sous R.

2.1.1 Variables explicatives obtenues à l'aéroport de Londres

L'aéroport de Londres est situé à quelques kilomètres au nord-est du comté de Lewisham, ce lieu étant très proche du lieu où les mesures de Nox ont été faites, nous y sélectionnerons les principales variables. Les mesures sont prises toutes les 30 minutes.

— X_1 : Température en Fahrenheit (T(F)=1.8*T(C)+32)
— X_2 : Vitesse du vent en nœud (1 nd =1,852 km/h = 0,5144 m/s)
— X_3 : Direction du vent (angle à partir du nord)
— X_4 : Humidité relative dans l'air (en %)

— X_5 : Point de rosée (La température la plus basse à laquelle une masse d'air peut être soumise, à pression et humidité données, sans qu'il ne se produise une formation d'eau liquide par saturation)
— X_6 : Pression atmosphérique (La pression atmosphérique est la pression qu'exerce le mélange gazeux constituant l'atmosphère considérée sur une surface quelconque au contact avec cette atmosphère)

2.1.2 Variables explicatives obtenues à l'aéroport d'Heathrow

L'aéroport d'Heathrow est à l'ouest de Londres, à une certaine distance de Lewisham, nous ne prendrons que deux variables d'intérêt (les mesures sont aussi prises toutes les 30 minutes) :

— X_7 : Vitesse du vent en nœud
— X_8 : Direction du vent (angle à partir du nord)

2.1.3 Variables explicatives relatives au trafic routier

Les quatre dernières variables sont des données du trafic routier directement lié à la production des oxydes d'azote. N'ayant pu trouver de données de trafic dans le comté de Lewisham, nous avons récupéré des données de Marylebone Road (centre de Londres). Nous supposerons qu'elles sont un sous-échantillon représentatif du trafic routier à Londres.

— X_9 : Vitesse moyenne du trafic en km/h.
— X_{10} : Nombre de véhicules de petite taille (motocycles et voitures)
— X_{11} : Nombre de véhicules de grande taille (camions et bus)

2.2 Traitement des données

Nos données sont issues de mesures physiques, ne sont pas traitées au préalable et nécessitent un travail en amont non négligeable.

2.2.1 Traitement de la variable Y

Les mesures des oxydes d'azote comportent 795 valeurs manquantes sur environ 26000 données. Les données manquantes étant regroupées par paquet, une interpolation est difficile. Nous proposons deux idées pour y remédier.
La première est une estimation non-paramétrique, l'histogramme des réalisations de la variable Y suggère une répartition selon une loi exponentielle. Nous pouvons donc simuler aléatoirement une variable exponentielle pour

remplacer les valeurs manquantes. Cette méthode est intéressante car très rapide. Elle demande peu d'investissement en temps au statisticien, quelle que soit la taille du dataset. Cependant elle ne permet pas de respecter les cycles journaliers dans notre cas.
La deuxième consiste à compléter les valeurs manquantes par les valeurs du cycle journalier moyen plus fidèle à nos données. Cependant, elle demande plus d'investissement en temps car il faut le faire manuellement. Par conséquent elle est peu utilisable en grande dimension. Le temps de manipulation étant acceptable pour notre jeu de données, c'est cette méthode que l'on a choisie.
Nous pouvons aussi remarquer quelques valeurs potentiellement aberrantes (10 à 100 fois la quantité moyenne) que l'on remplace par la médiane.

2.2.2 Traitement des données venant de l'aéroport de Londres

Ces données sont en grande partie propres, excepté pour la direction du vent qui compte 1600 valeurs manquantes.
Toutes ces données sont mesurées toutes les 30 minutes avec un décalage par rapport à notre variable Y. (ex : 17H20/17H50/18H20 etc)
Nous allons résoudre ces deux problèmes de la même façon, en procédant à plusieurs interpolations linéaires. Nous pouvons nous poser la question du sens de cette méthode pour compléter les données de la direction du vent. Au commencement du projet, nous n'avions pas trouvé de données aussi propres, nous avions récupéré les mêmes données avec beaucoup plus de valeurs manquantes (25% de l'échantillon). Nous nous sommes servis de ces données pour effectuer une interpolation linéaire et comparer avec les données propres si celles-ci avaient du sens ou non. Il s'est avéré que cette méthode était relativement bonne visuellement dans un premier temps, avec un rmse faible dans un second temps, ce qui a justifié notre utilisation de l'interpolation linéaire pour la direction de vent.

2.2.3 Traitement des données venant de l'aéroport d'Heathrow

Ces données sont aussi propres : une dizaine de valeurs manquantes et quelques valeurs aberrantes ; nous procédons de la même manière que précédemment.

2.2.4 Traitement des données du trafic routier

Les données sont relativement propres : quelques valeurs manquantes isolées complétées par interpolations linéaires. Elles sont données à heure fixe de la même façon que la quantité de Nox.

2.3 Analyse descriptive des variables

Une analyse descriptive de la variable à prédire et des variables explicatives permet de mieux comprendre l'ampleur de la tâche. En effet, l'écart-type des Oxydes d'azote est relativement élevé.

	Mean	Median	σ	min	max
NOx	112.1	91.32	87.2	0	693
Température	54.3	53.6	10.3	26.6	98.6
Vitesse trafic	39.2	39.4	7.15	14.9	51.5
Trafic petit	470	537.3	158	2.16	719
Trafic gros	59.6	48	36.5	0.3	154.8
Humidité	73	75	14.3	20.5	100
Point de rosée	45.1	45.3	8.3	18.5	67.1
Pression	30	30	0.3	28.7	30.8
Vitesse vent Heathrow	8.24	7.8	4.2	0	39.5
Direction du vent Heathrow	187	201	90.1	0	360
Vitesse du vent AL	7.3	6.9	4.44	0	33.5
Direction du vent AL	168	194	96	0	360

3 Premier cadre : on connaît les variables explicatives sur les données de validation

Commençons par introduire deux mesures d'erreur, le RMSE (Root square mean error) et le MAE (Mean absolute error) qui vont nous permettre de comparer les performances des différentes méthodes de ML. Nous séparons nos données en deux parties : les années 2002/2003 composent l'échantillon d'entraînement et l'année 2004 compose l'échantillon de validation. Nous entraînons nos algorithmes sur l'échantillon d'entraînement et nous regardons

l'efficacité de la méthode en regardant l'erreur de prédiction sur l'échantillon de validation.

3.1 Présentation des modèles et résultats à horizon d'un an

Nous évaluons tout d'abord les modèles sur une prédiction de l'année entière. Cela signifie que les modèles ne sont entraînés qu'une seule fois sur les données de 2002 et 2003 et on prédit les données de 2004.

3.1.1 Modèles Gam

On se propose d'utiliser des modèles additifs généralisés. On va chercher à obtenir les fonctions f_1,..,f_{11} telles qu'on ait :

$$Y_t = \beta + f_1(X_{1,t}) + ... + f_{11}(X_{11,t}) + \epsilon_t$$

où les f_j j=1,...,11 sont estimées par régression sur une base de splines et ϵ étant le bruit. Les f_j s'écrivent donc $f_j(x) = \sum_{i=1}^{k_j} a_{j,i}\beta_{j,i}$. On doit donc choisir les bases de splines adaptées ainsi que le nombre de nœuds k_j associé à chaque base de splines. On propose d'effectuer une V-Fold Cross-Validation pour choisir les k_j optimaux pour les quatre bases de splines les plus courantes pour chaque variable séparément. Les splines par régression cubique "cr", leur version shrinkage "cs", les *Thin Plate Splines* "tp" et les B-splines pénalisés "ps". Notons qu'on aurait aussi pu tester les différentes bases et k_j avec deux variables en même temps. Cependant, au delà de deux variables, les degrés de liberté explosent. On propose cette modélisation :

- X_1 Température : k=10 "cs"
- X_2 Vitesse du vent (Aéroport de Londres) : k=10 "cs"
- X_3 Direction du vent (Aéroport de Londres) k=10 "cs"
- X_4 Humidité relative : k=10 "cs"
- X_5 Point de rosée : k=10 "cs"
- X_6 Pression atmosphérique : k=5 "ps"
- X_7 Vitesse du vent (Heathrow) : k=10 "cs"
- X_8 Direction du vent (Heathrow) : k=10 "cs"
- X_9 Vitesse du trafic : k=10 "cs"
- X_{10} Nombre de véhicules de petite taille : k=10 "cr"
- X_{11} Nombre de véhicules de grande taille : k=10 "cr"

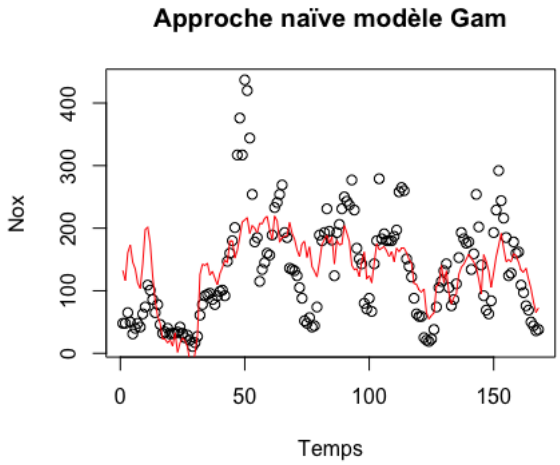


FIGURE 1: Modèle GAM sur la première semaine de l'échantillon de validation à horizon de prévision d'un an (en rouge le modèle, en noir les données)

NB : Quand il n'y a pas de différence entre les différentes bases, et k_{opt} appartient à un intervalle contenant 10, on prend k égal à 10 (standard).

Une approche naïve consiste à utiliser directement ce modèle. On obtient un rmse de 62 et un mae de 43 pour un horizon de prévision d'une année. Cependant il peut rester des corrélations dans les résidus. En regardant la première semaine de prévision sur le Dataset de test, on obtient le résultat présenté en figure 1.

3.1.2 Arbres de régression et leurs dérivés

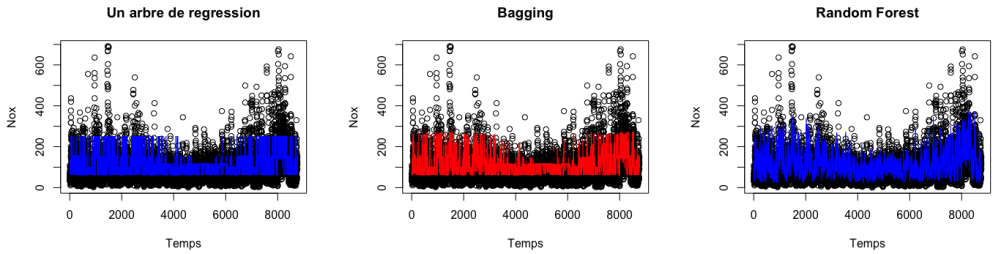
Nous proposons de modéliser la quantité de Nox dans l'air par des méthodes d'ensemble : un arbre de régression simple, une méthode de bagging et une méthode de forêts aléatoires. On peut deviner que les forêts aléatoires seront plus efficaces. Néanmoins il nous semble intéressant de tester les différentes méthodes en parallèle, ne serait-ce que pour visualiser les améliorations. Nous les réutiliserons lorsque nous essayerons d'augmenter les performances de prédiction par agrégation de nos différentes méthodes. Un arbre simple de régression est très grossier avec un rmse de 72 et un mae de 53.

On utilise ensuite une méthode dite de bagging qui consiste à agréger un certain nombre d'arbres de régression et qui apporte de la stabilité. A partir de 200 arbres, il n'y a plus d'amélioration du modèle. On choisit donc d'effectuer un bagging avec 200 arbres. On choisit une profondeur maximale de 30. On constate une petite amélioration avec un rmse de 70 et un mae de 52.

Passons maintenant aux forêts aléatoires qui sont reconnues pour leur efficacité même si peu de garanties théoriques sont encore présentes sur cette méthode. Intéressons-nous dans un premier temps au choix des paramètres :

- Le nombre d'arbres `ntree` : La performance du modèle augmente avec le nombre d'arbres choisi. Cependant, cela augmente linéairement le temps de calcul de la procédure. Sur nos données, l'amélioration du modèle est très faible au-delà 90 arbres. Nous choisirons donc `ntree`=90.
- La taille du sous-échantillon `mtry` : La pratique usuelle est de prendre le tiers du nombre de variables. N'ayant que 11 variables, il nous est facile de tester toutes les tailles puis de prendre celle qui apporte la meilleure performance. celle-ci est donné pour `mtry`=2.
- Le nombre de valeurs maximales dans chaque feuille `nodesize` : les différentes valeurs possibles que nous avons testées n'ont pas influé sur les performances. On choisira donc le paramètre par défaut.
- `maxnodes` qui limite le nombre de nœuds terminaux. Le temps de calcul augmente avec ce paramètre. Si il est faible, la performance l'est aussi. Il est infini par défaut. Le temps de calcul est acceptable avec notre jeu de données, on le laisse donc ainsi.

Nos paramètres étant ajustés, on obtient un rmse de 60 et un mae de 40 ce qui ne laisse présager que du bon. La figure 2 représente les prévisions obtenues pour les trois méthodes. On peut aussi regarder l'importance des différentes covariables sur la prédiction. On remarque ci-dessous, que la variable la plus importante est la densité du trafic des gros véhicules. Cela concorde avec une approche purement intuitive. Le modèle Random Forest explique 62 % de la variance, ce qui n'est pas énorme mais se justifie par des données fortement bruitées venant d'appareils de mesures physiques.



(a) Arbre de régression (b) Bagging (c) Forêt aléatoire

FIGURE 2: Méthodes basées sur les arbres, prévision à horizon d'un an

	%IncMSE	IncNodePurity
WVitesseNE	2264.1275	16088699
WDirectionNE	1691.9013	12503183
WVitesseO	2160.2307	16695001
WDirectionO	1362.2600	11554630
Temperature	2317.5520	13428808
Pression	844.8716	8248262
humidite	1119.6916	9800717
Point_de_rosee	1425.7541	8932812
TrafficPetit	963.7256	7374199
TrafficGros	2796.5950	14373653
VitesseT	1096.3853	7893001

3.1.3 Algorithmes de Boosting

Nous allons maintenant nous intéresser aux méthodes de boosting. Les algorithmes de boosting sont des méthodes d'ensemble, qui proviennent d'une agrégation séquentielle d'arbres simples.

Generalized boosted regression : Nous commençons par utiliser un modèle "Generalized boosted regression". Comme avec les arbres aléatoires, on regarde dans un premier temps le nombre d'arbres nécessaire avec une méthode d'out of bag grâce à la fonction `gbm.perf`. L'amélioration du modèle est négligeable après 460 arbres. C'est un nombre d'arbres conséquent mais nos données le permettent, les temps de calcul sont faibles. Ensuite nous tentons d'optimiser le paramètre de rétrécissement appliqué à chaque arbre. Si sa valeur est petite (< 0.1) cela conduit à augmenter le nombre d'arbres mais entraîne généralement une amélioration de la qualité de prévision (voir figure 3). Au vu du graphique et du temps de calcul nécessaire on gardera

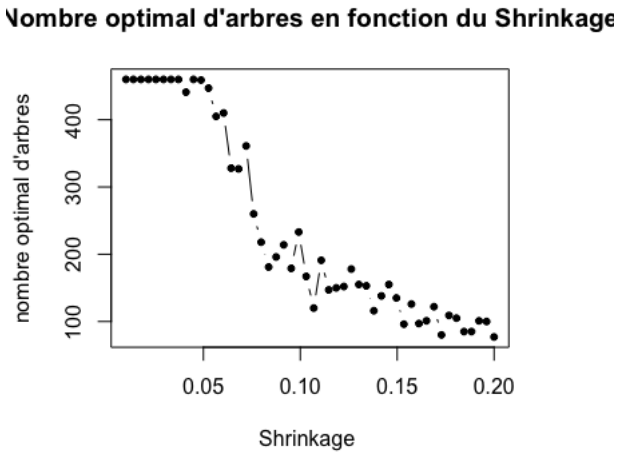


FIGURE 3: Influence du shrinkage

460 arbres. Pour la profondeur maximale des arbres, on gardera les valeurs par défaut. On obtient un rmse de 61 et un mae de 44.

XGboost : Nous allons maintenant utiliser l'algorithme Xgboost. L'extrême Gradient Boosting est une très bonne méthode utilisée fréquemment dans les compétitions Kaggle. On prend donc le temps d'optimiser les différents paramètres.

- La profondeur maximale des arbres `max.depth` : Le modèle donne de meilleurs résultats pour `max.depth` $\in [8, 10]$. On prendra donc une profondeur maximale égale à 8.
- Le Learning rate `eta` : une valeur faible de ce paramètre améliore les prévisions mais augmente le temps de calcul, il permet aussi d'éviter l'over-fitting (il ne faut pas le choisir trop petit non plus). Encore une fois nos données ne sont pas si grandes, on prend `eta`=0.01.
- La proportion des sous-échantillons `subsample` : une proportion plus faible réduit évidemment le temps de calcul. On prendra `subsample`=0.9.

On obtient avec les paramètres ci-dessus un rmse de 58 et un mae de 41 (voir figure 5).

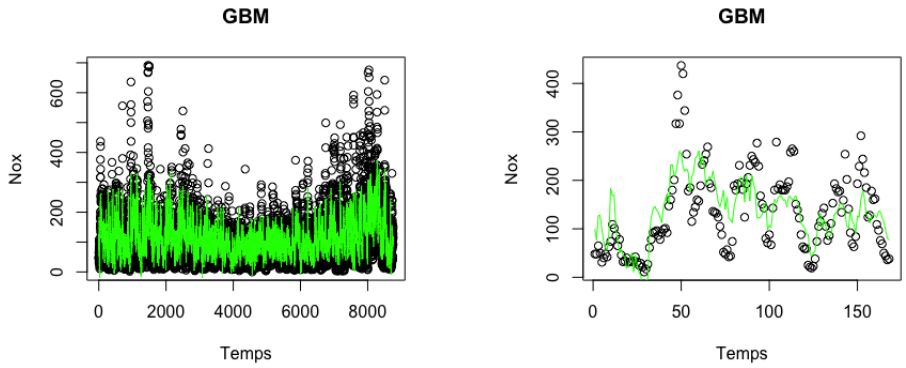


FIGURE 4: Generalized boosted regression sur la première semaine et l'année entière de l'échantillon de validation à horizon de prévision d'une année (en vert le modèle, en noir les données).

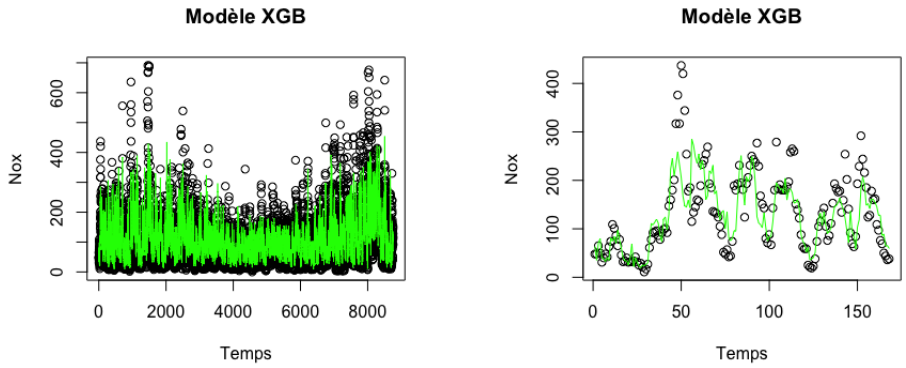


FIGURE 5: XGB sur la première semaine et l'année entière de l'échantillon de validation (en vert le modèle, en noir les données).

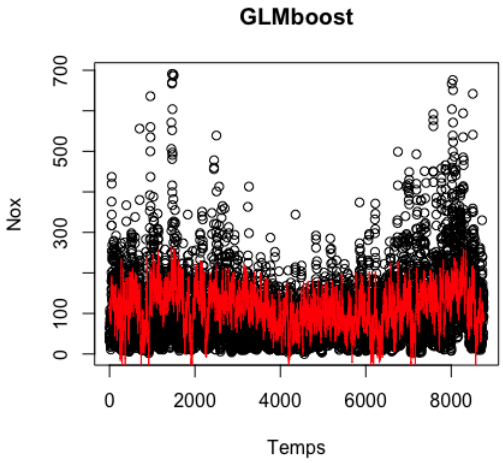


FIGURE 6: Gradient boosting avec modèle linéaire, horizon de prévision d'un an (en noir les données, en rouge le modèle)

Gradient boosting avec modèle linéaire : Nous avons testé un modèle de gradient boosting linéaire. On obtient un rmse de 72 et un mae de 55. Cet algorithme n'est pas très approprié pour notre jeu de données à première vue, car difficile pour lui de prendre les pics en compte (voir figure 6). Nous n'irons pas plus loin.

Gamboost : Nous proposons d'utiliser les méthodes de boosting associées aux modèles Gam. Nous allons optimiser les paramètres du modèle :

- Le `Baselearner` : on choisit d'utiliser P-splines avec une base de B-splines.
- `dfbase` : le nombre de degrés de liberté des "baselearner". La performance du modèle est la même de 4 à 10, on choisira 4 degrés de liberté.
- les paramètres du boosting : nous choisirons les mêmes paramètres que pour xgboost, qui donnent de bons résultats.

On obtient ces prédictions en ajustant les paramètres ci-dessus avec un rmse de 62 et un mae de 44 (voir figure 7).

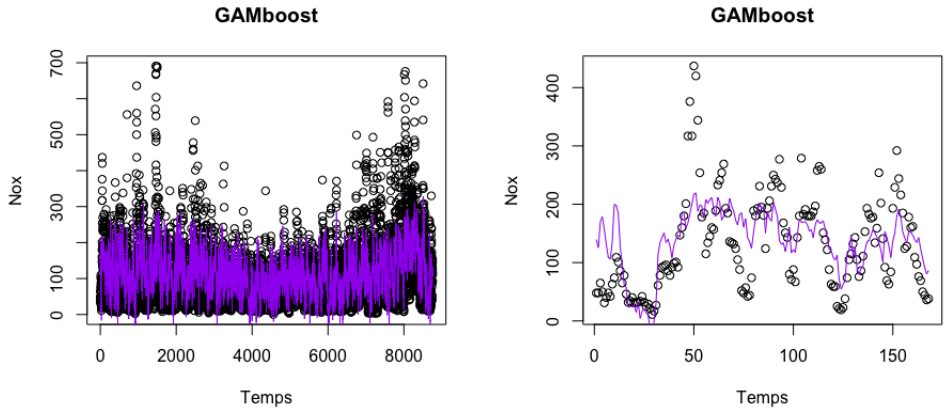


FIGURE 7: Gamboost, horizon de prévision d'un an (en violet le modèle, en noir les données)

3.1.4 Extreme Machine Learning Neural Network

L'Extreme Machine Learning correspond à un type particulier de réseau de neurones qui ne possède qu'une seule couche de nœuds cachés. Les poids et les biais des neurones sont aléatoires et ne sont pas mis à jour. Considérons K neurones cachés, tel que K< nombre d'observations, les poids $w_i \in \mathbb{R}^n$ qui connectent le i-ème neurone caché et les neurones d'entrée, et $\beta_i \in \mathbb{R}^n$ qui connecte le i-ème neurone caché avec les neurones de sortie.
Le problème consiste à minimiser

$$\sum_{j=1}^{n}(\sum_{i=1}^{K}\beta_i g(w_i X_i + b_i) - Y_j)^2$$

où g est la fonction d'activation du neurone. En dérivant en β, minimiser ce problème revient à trouver β tel que

$$\sum_{i=1}^{K}\beta_i g(w_i X_i + b_i) = Y_j \quad j = 1, ..., n$$

c'est-à-dire trouver β tel que $H\beta = T$ sous forme matricielle. Ainsi

$$\beta^* = H^{\dagger}T$$

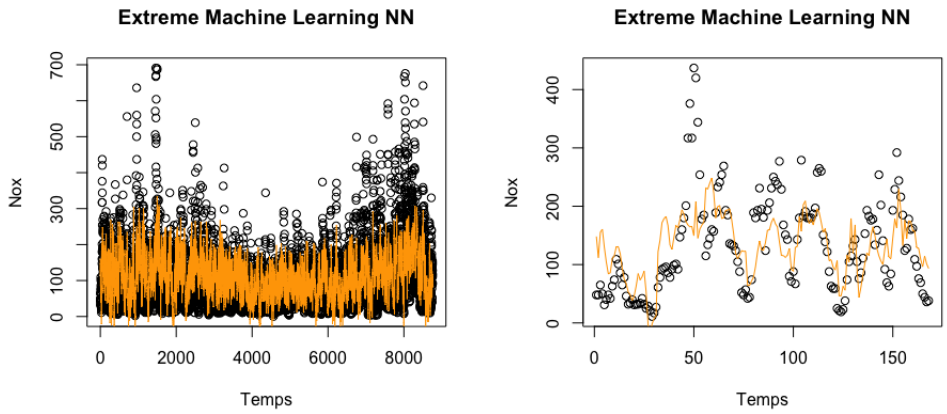


FIGURE 8: Extreme Machine Learning Neural Network , horizon de prévision d'un an (en noir les données, en orange le modèle)

où $H^{\dagger}$ représente l'inverse généralisé de Moore-Penrose.
N'ayant qu'une seule couche de neurones, le temps de calcul de cette méthode est très rapide (voir figure 8 pour les prévisions).

3.2 Méthodes à horizon d'une heure

Pour améliorer les résultats, deux méthodes peuvent être utilisées : des modèles de séries temporelles sur les résidus ou l'agrégation des prédicteurs.

Les modèles de série temporelle modélisent les dépendances entre les variables à un instant t et leurs valeurs aux instants précédent. Afin de prédire sur un horizon plus grand que le pas de nos données (ici une heure), il est nécessaire de réintroduire séquentiellement dans le modèle les prévisions faites précédemment. Cela a pour conséquence un mauvais fonctionnement de ces méthodes à horizon de prévision long. Mais l'horizon d'une heure est trop faible pour avoir un intérêt pratique. A titre indicatif, nous proposons tout de même d'étudier l'influence d'un tel lissage à un horizon d'une heure. Les résultats présentés ci-dessous sont obtenus en appliquant d'abord le modèle de base à horizon annuel auquel on ajoute les prévisions du lissage exponentiel à pas horaire.

Les méthodes d'agrégation mettent à jour les poids associés aux prédicteurs de manière séquentielle. Ici de même, les poids sont mis à jour à pas

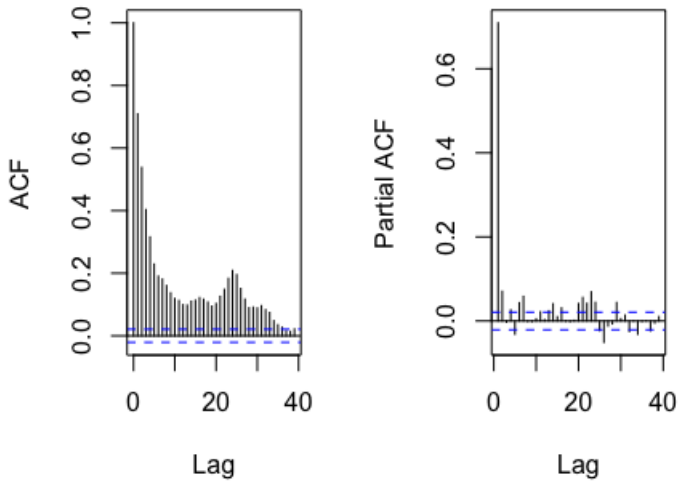


FIGURE 9: ACF et PACF des résidus du modèle GAM

égal à celui des données. Pour effectuer des prédictions à horizon supérieur à une heure, on doit alors soit moyenner les données sur l'horizon qu'on veut (exemple une journée) ou bien considérer plusieurs séries temporelles, une pour chaque heure de l'horizon (par exemple pour une journée, on considère 24 séries temporelles). Ici, encore une fois on garde le pas horaire à titre indicatif.

3.2.1 Analyse des résidus et méthodes de séries temporelles

L'analyse des résidus obtenus par les différents modèles permet de se rendre compte qu'ils ne modélisent pas complètement l'information présente dans les données. Par exemple, la figure 9 présente l'auto-corrélation et l'auto-corrélation partielle des résidus obtenus à partir du modèle GAM. Il semble qu'il reste une composante de saisonnalité et des corrélations dans les résidus.

Ces composantes peuvent être captées par des méthodes de séries temporelles, on propose un lissage exponentiel de Holt-Winters additif avec tendance linéaire et saisonnalité ou un ETS (Error,Trend,Seasonal) sur les résí-

dus.

Soit y_t un processus, la méthode de Holt-Winters prédit $\hat{y}_{t+h}$ connaissant l'information à la date t par : $y_{t+h} = l_t + hb_t + st$ où s_t est la composante saisonnière à la date t. Les différentes composantes sont mises à jour par :

$$
\begin{aligned}
y_{t+h} &= l_t + hb_t + s_{t-m} \\
l_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\
b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma^*(y_t - l_{t-1} - b_{t-1}) + (1-\gamma^* s_{t-m})
\end{aligned}
$$

Les modèles ETS sont une généralisation des lissages exponentiels avec un grand nombre de variantes (dépend de la présence de saisonnalité/tendance, si elle sont multiplicatives/additives). La fonction *ets()* permet de choisir automatiquement ces paramètres.

Les visualisations des prévisions sont présentées en figure 10 et les résultats en section 3.5

3.2.2 Agrégation

On se propose maintenant d'agréger nos différents modèles à horizon d'une année à l'aide du package `opera`. On utilise différents cortèges d'experts (composés de tous les modèles précédents). Après expérimentation (en testant à la main avec EWA/BOA), Il s'avère que le meilleur groupe est celui composé des 7 experts suivants :

— XGBoost, GamBoost, GBM, GAM, RF,ElmNN,GLM

Nous allons ensuite choisir les poids les plus performants. On obtient les rmse suivants :

— EWA : 56.8, BOA : 57.2, FS : 52, OGD : 57.7, MLpol : 57.1, Ridge : 55.3.

On remarque que peu importe le type d'agrégation, on obtient un gain de performance par rapport aux meilleurs experts. L'agrégation Fixed share semble être bien meilleur que les autres. Ce type d'agrégation consiste à donner beaucoup de poids au meilleur expert et très peu aux autres. Le modèle XGBoost étant vraiment meilleur que les autres experts cela explique ce résultat. Le package *opera* permet d'obtenir différents graphiques permettant de mieux comprendre le processus d'agrégation et notamment de voir quels ont été les différents poids choisis au cours du processus et de comparer la perte quadratique entre l'agrégat et les différents modèles (voir figure 11).

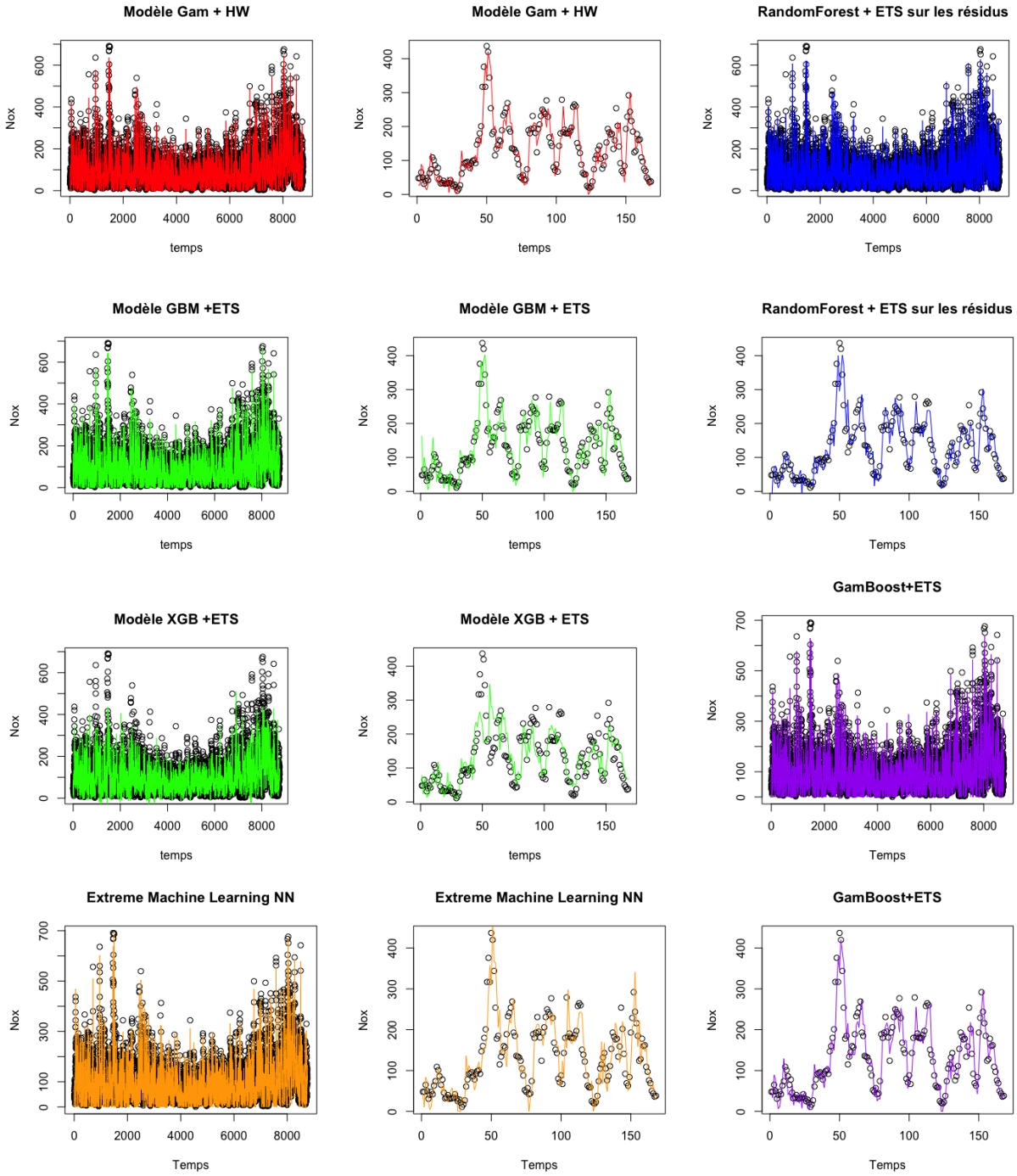


FIGURE 10: Prévisions avec lissage exponentiel (horizon d'une heure)

On effectuera une deuxième agrégation pour les horizons de prévision d'un jour (1) dans la partie des moyennes journalières, on utilisera le cortège d'experts suivant :

— RF+ETS, GamBoost+ETS, GBM+ETS, GAM+ETS, ElmNN+ETS

Il n'y a pas vraiment de modèle meilleur que les autres sur ces experts, la meilleure agrégation est la descente de gradient en ligne `OGD`.

3.3 Alternatives pour améliorer la performance

3.3.1 Enlever la saisonnalité et la tendance des Oxydes d'azote au préalable

On propose dans un second temps de commencer par prédire la saisonnalité et la tendance de la variable Y avec des modèles GAM, puis de les enlever de nos données pour ensuite appliquer nos algorithmes de ML. On obtient de meilleurs résultats, cela permet de mettre en exergue le fait que les algorithmes de ML, aussi puissants soient-il, ont du mal à prendre en compte toutes les composantes des données. Il est donc plus intéressant d'utiliser des modèles additifs ou multiplicatifs.

3.3.2 Moyenner les données

Comme on a pu le voir dans certaines publications de chercheurs sur le sujet, ces derniers ne s'attaquent que très rarement au même problème que nous. Bien souvent, ils simplifient le problème en moyennant les valeurs journalières et en découpant les données en fonction des saisons (été,hiver..). Nous n'avons que deux années pour entraîner nos modèles. Découper les données en fonction des saisons nous semble peu pertinent ici, néanmoins nous allons moyenner de façon journalière nos variables. On obtient donc 729 observations sur l'échantillon d'entraînement et 365 sur l'échantillon de test ce qui reste correct pour faire de la prédiction. Notons qu'en faisant cela on enlève beaucoup de variance sur les variables ce qui simplifie le problème,. On obtient d'ailleurs de bien meilleurs résultats.

3.4 Horizons de prévision plus raisonnables

Nous avons vu jusque-là les modèles tels quels avec un horizon de prévision d'une année. Deux méthodes ont de plus été présentées pour améliorer la prédiction (lissage exponentiel et agrégation). Néanmoins, ces méthodes

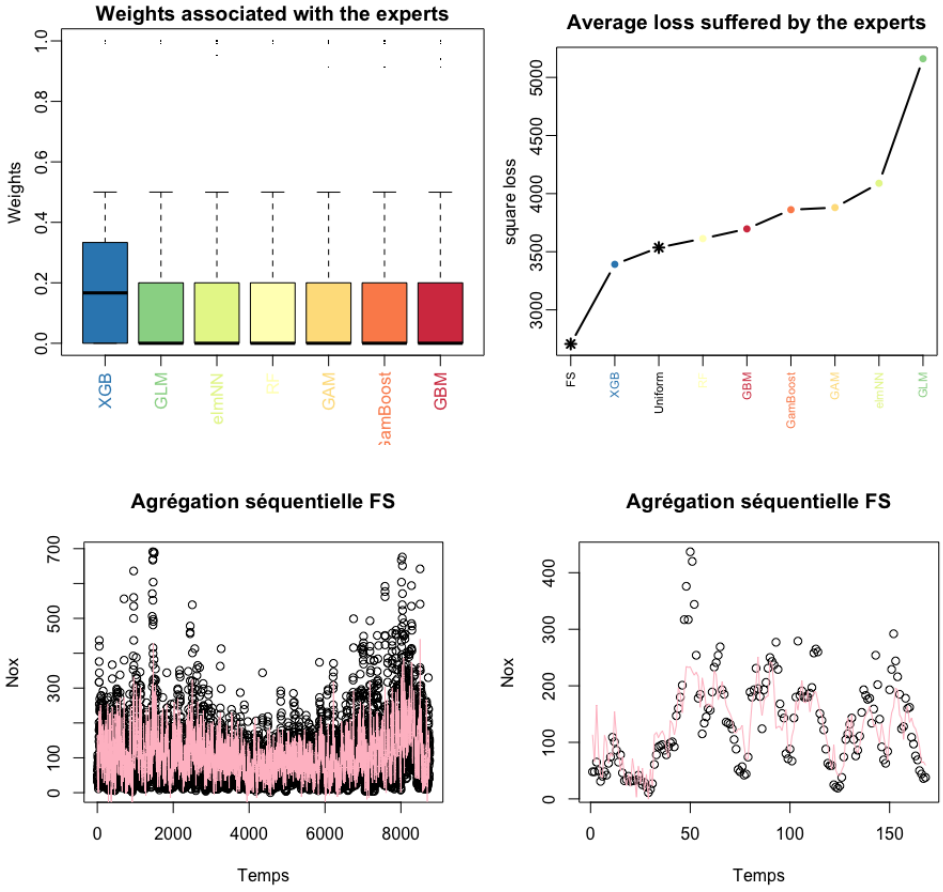


FIGURE 11: Agrégation

ont été testées à horizon de prévision d'une heure (le pas de nos données) uniquement pour avoir une indication de la manière dont elles affectent la prévision. Les deux horizons vus jusqu'ici sont peu représentatifs d'un cas concret, l'horizon annuel semble trop difficile étant donné notre pas horaire alors que l'horizon d'une heure semble trop facile. Nous proposons alors d'évaluer les modèles à horizon d'une semaine et d'une journée. Par exemple pour un horizon d'une semaine, nous allons d'abord entraîner le modèle sur les deux années d'entraînement, puis prédire une semaine dans le futur. Une fois les données de cette semaine observées, celles-ci sont ajoutées aux données d'entraînement. Le modèle est alors réentraîné pour prédire la prochaine semaine. Cela est fait jusqu'à avoir prédit l'année entière.

Nous allons voir que les prévisions sont, en général, bien meilleures que l'horizon annuel, cependant le temps de calcul sera bien plus grand puisqu'il faudra entraîner 52 (ou 365 pour l'horizon d'une journée) fois le modèle.

Les études faites précédemment à horizon annuel et horaire permettent d'avoir une idée des meilleurs modèles. Ainsi, nous n'allons utiliser que les meilleurs modèles précédents.

Il est important de noter que les erreurs de prévision peuvent varier entre les différentes semaines (ou journées). A titre d'exemple, les RMSE calculés pour chaque semaine de prévision pour un modèle GAM avec lissage exponentiel sont représentés en figure 12. On s'aperçoit que pour la plupart des semaines les performances sont bonnes, cependant pour quelques semaines le modèle semble mauvais. Ces valeurs correspondent aux semaines où il y a présence de grands pics de pollution. Afin d'avoir un résultat global sur l'année, nous présentons dans la section 3.5 les résultats moyens et médians sur l'année.

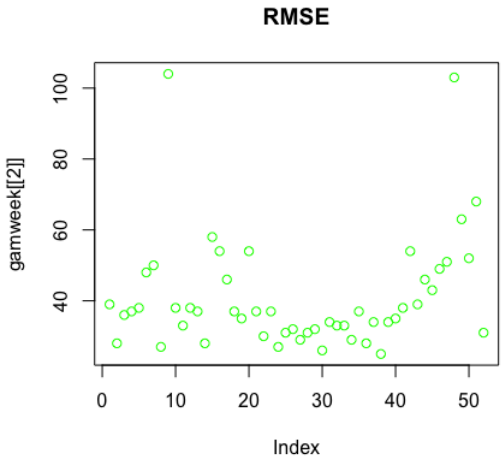


FIGURE 12: RMSE à horizon d'une semaine pour GAM+ETS

3.5 Résultats

Nous présentons dans les tableaux 1 et 2 les résultats des différents modèles aux différents horizons de prévision. Nous rappelons ici comment les différents cas sont étudiés :

- **Modèles seuls :** Ici, les modèles ont été testés tels quels.
- **Modèles + lissage ou agrégation :** Ici, il faut distinguer l'horizon d'un an des autres. Pour l'horizon d'un an, on utilise le modèle pour prédire l'année complète puis on applique le lissage ou l'agrégation à pas horaire sur les résidus. Les résultats ne sont présents qu'à titre indicatif. Pour les autres horizons, les prévisions sont réellement faites à l'horizon décrit : on utilise le modèle et le lissage pour prédire l'horizon entier.
- **Avec modélisation de la tendance et de la saisonnalité :** Ici, on utilise un modèle additif où l'on prédit d'abord la tendance et la saisonnalité avant d'utiliser le modèle.

Les résultats du tableau 2 sont obtenus en moyennant les données sur une journée.

Horizon de prévision	Année		Semaine		Jour	
Mesures d'erreurs	RMSE	MAE	Mean RMSE	Med RMSE	Mean RMSE	Med RMSE
Modèles seuls						
Arbre simple	72	53	-	-	-	-
Bagging	70	52	-	-	-	-
Random Forest	60	40	55	48	50	42
Gam	61	44	57	50	53	46
GBM	61	44	55	48	51	44
XGboost	58	41	50	43	44	36
GLMboost	72	55	-	-	-	-
GAMboost	62	44	57	52	53	47
ELMNN	64	45	60	55	57	50
Modèle + lissage ou agrégation						
RF+ETS	43	26	39	35	-	-
Gam+ETS	44	28	41	38	-	-
GBM+ETS	43	27	40	36	-	-
XGboost+ETS	46	29	-	-	-	-
GAMboost+ETS	44	27	39	34	-	-
ELMNN+ETS	45	29	40	36	-	-
Agrégation	52	36	-	-	-	-
Avec modélisation de la tendance et de la saisonnalité						
Random Forest	56	38	51	43	47	40
Gam	58	41	54	45	50	44
GBM	57	40	52	44	48	42
GAMboost	57	40	53	45	50	43
ELMNN	59	42	56	50	53	45
Xgboost	55	37	50	42	44	36

TABLE 1: Résultats avec variables explicatives accessibles. On rappelle que les données de NOx vont de 0 à 693.

Horizon de prévision	Année		Semaine		Jour	
Mesures d'erreurs	RMSE	MAE	Mean RMSE	Med RMSE	Mean RMSE	Med RMSE
Random Forest	41	31	34	30	28	21
Random Forest+ETS	-	-	-	-	26	20
Gam	43	32	36	32	29	21
Gam+ETS	-	-	-	-	27	21
GBM	45	32	36	31	28	19
GBM+ETS	-	-	-	-	26	18
GAMboost	45	33	35	32	28	20
GAMboost+ETS	-	-	-	-	26	20
ELMNN	43	33	38	36	30	23
Xgboost	43	31	35	29	-	-
Agrégation	-	-	-	-	25	-

TABLE 2: Résultats en moyennant les données sur une journée

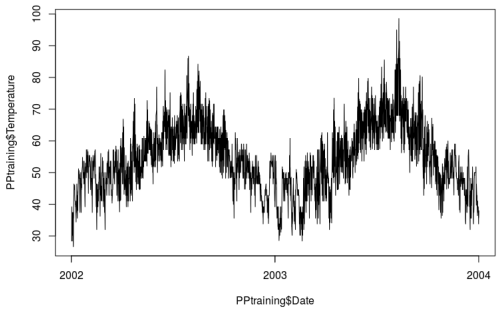
4 Second cadre : on ne connaît pas les variables explicatives sur les données de validation

4.1 Prédiction des covariables

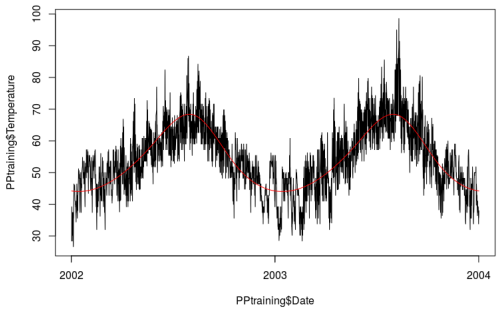
Dans un cas concret, on n'a pas accès aux valeurs des covariables pour faire la prédiction, il est d'abord nécessaire de prédire celles-ci avant d'utiliser les modèles développés ci-dessus. Pour prédire les covariables nous avons utilisé une méthode usuelle de série temporelle (comme on ne peut utiliser que les données relatives au temps et au passé de la covariable à prédire). On cherche d'abord à estimer la tendance puis la ou les saisonnalités si elles existent et enfin on cherche à estimer un modèle de type SARIMA sur les résidus. Néanmoins, nous avons privilégié des modèles SARIMA simples pour des raisons de temps de calcul (d'autant plus que les différences de performances n'étaient pas flagrantes).

4.1.1 Température

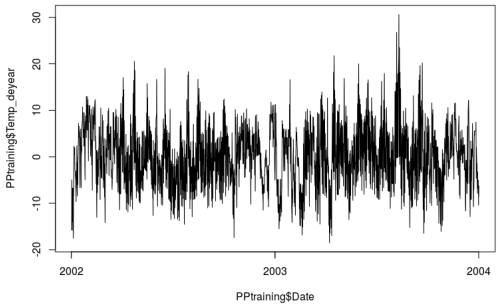
La température présente deux saisonnalités : une première sur l'année et une seconde sur la journée. Pour modéliser cela, nous avons utilisé un modèle GAM avec spines cycliques pour la saisonnalité annuelle et un second pour la saisonnalité journalière auquel nous avons ajouté la température décalée de 24 heures dans le passé afin d'ajouter une composante auto-régressive au modèle.



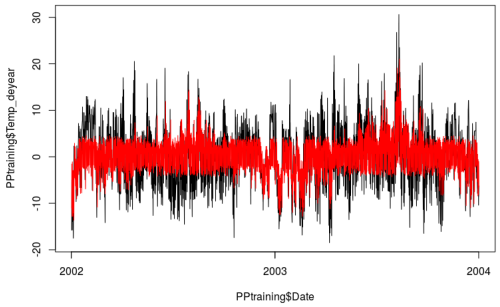
(a) Température



(b) Estimation de la saisonnalité annuelle



(c) Température sans la saisonnalité annuelle



(d) Estimation saisonnalité journalière + composante auto-régressive

FIGURE 13: Modèle pour la température

4.1.2 Données relatives au trafic routier

Le trafic routier est lié à l'activité humaine, donc le fait d'être en week-end ou en vacances (essentiellement les vacances de Noël) est une donnée importante dans sa modélisation (voir figure 14). De plus, une saisonnalité journalière est en général présente lorsque l'on observe des données liées à l'activité humaine. Le modèle utilisé est similaire pour les trois types de données de trafic. Tout d'abord un modèle GAM avec peu de degrés de liberté mais prenant en compte Noël et les week-ends. Ensuite un modèle GAM à splines cycliques pour capter la saisonnalité journalière.

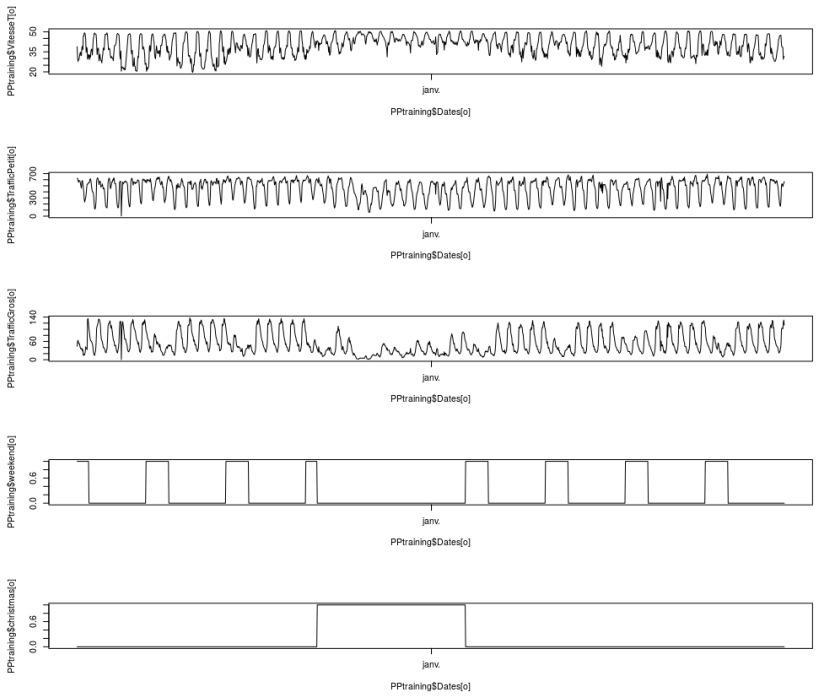
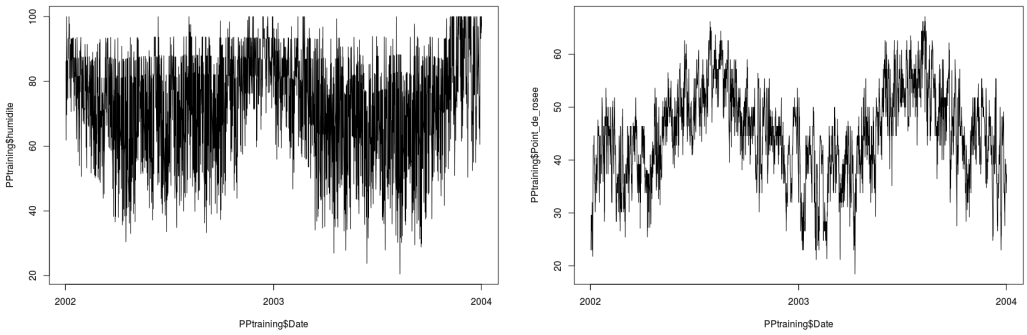


FIGURE 14: Dépendance des données de trafic avec les week-end et la période de Noël. De haut en bas : vitesse, trafic petit, trafic gros, week-end, Noël.

4.1.3 Humidité et point de rosée

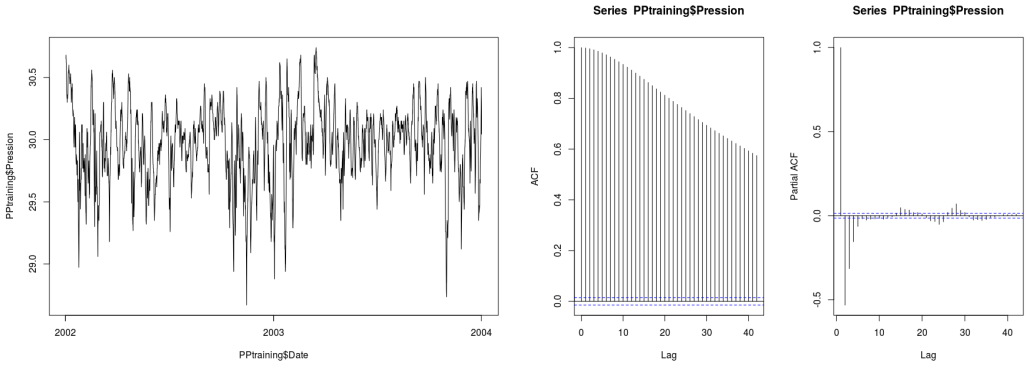
L'humidité a un comportement similaire à la température : une saisonnalité annuelle et une autre journalière toutes deux modélisées par des modèles GAM auxquels on ajoute une composante AR(1). Le point de rosée a une saisonnalité annuelle uniquement à laquelle on ajoute un composante AR(5).



(a) Humidité (b) Point de rosée

4.1.4 Pression

La pression n'a ni tendance ni saisonnalité mais une analyse des auto-corrélations et auto-corrélations partielles laissent penser qu'un AR(5) est un bon modèle.



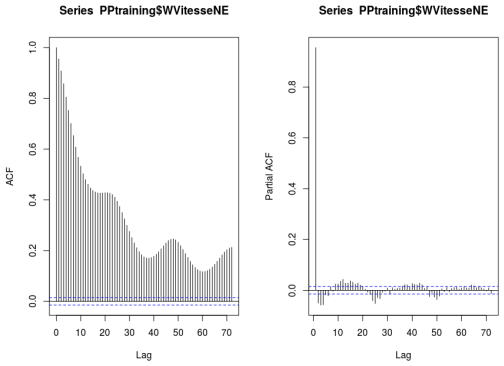
(a) Données (b) Auto-corrélation et auto-corrélation partielle

FIGURE 16: Pression

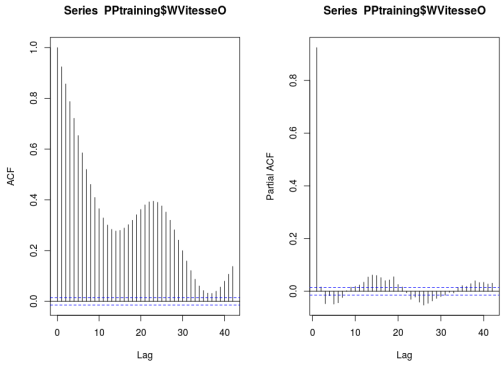
4.1.5 Vitesse du vent

La vitesse du vent aux deux aéroports présente les mêmes caractéristiques : une saisonnalité journalière et une composante auto-régressive. La

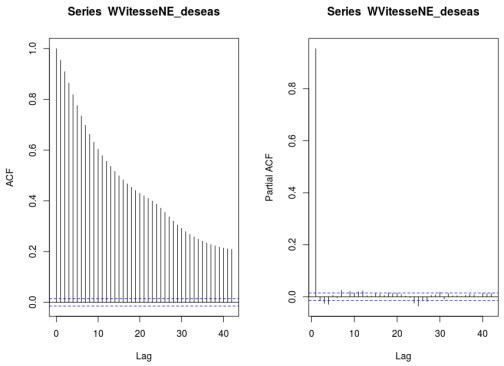
saisonnalité est captée par un modèle GAM à spline cyclique et la composante auto-régressive par un AR(1) pour Heathrow et AR(3) pour London Airport.



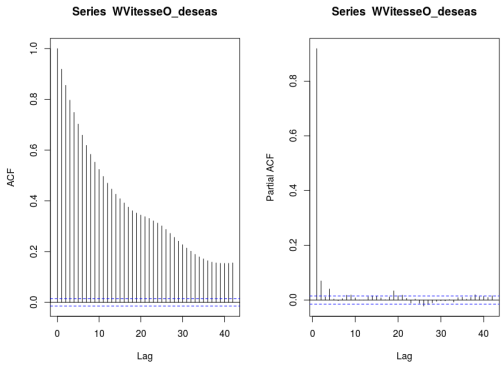
(a) London Airport



(b) Heathrow



(c) London Airport désaisonnalisé

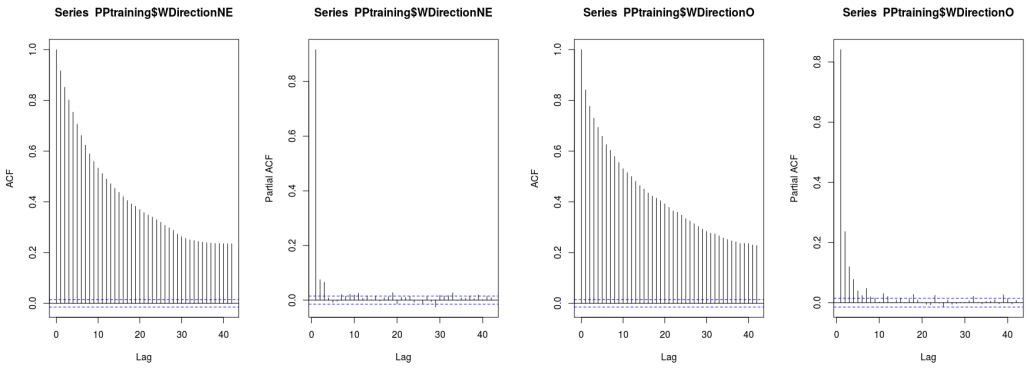


(d) Heathrow désaisonnalisé

FIGURE 17: ACF et PACF de la vitesse du vent

4.1.6 Direction du vent

La direction du vent n'a pas de saisonnalité apparente et les auto-corrélations et auto-corrélations partielles laissent penser à un processus AR(5) pour Heathrow et AR(3) pour London Airport.

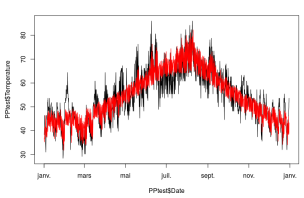


(a) London Airport (b) Heathrow

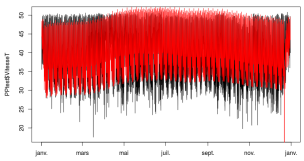
FIGURE 18: ACF et PACF de la direction du vent

4.1.7 Résultats des prévisions des covariables

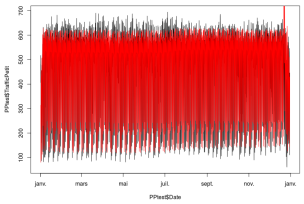
Pour évaluer les modèles sur les données de test, on utilise des prévisions à pas d'une journée. Cela signifie qu'on entraîne d'abord le modèle sur les données d'entraînement puis on prédit la première journée des données de test. Ensuite on prédit les prochaines journées des données de test en ajoutant les journées précédentes aux données d'entraînement et en réentraînant le modèle sur ces nouvelles données au préalable.



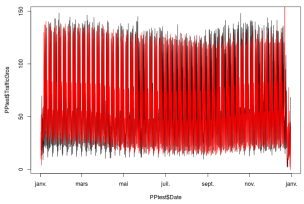
(a) Température



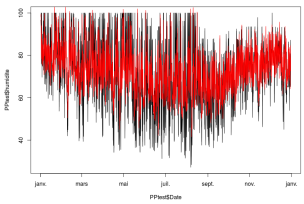
(b) Vitesse trafic



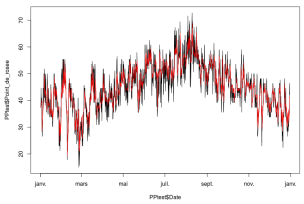
(c) Trafic petit



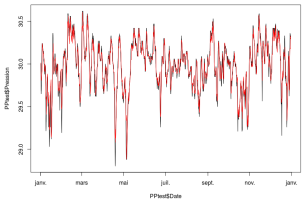
(d) Trafic gros



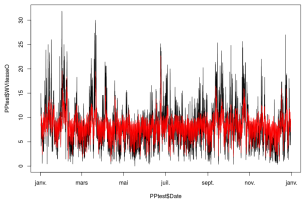
(e) Humidité



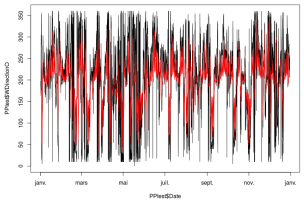
(f) Point de rosée



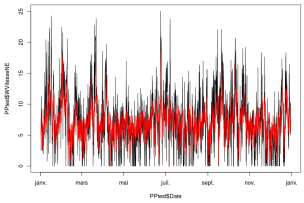
(g) Pression



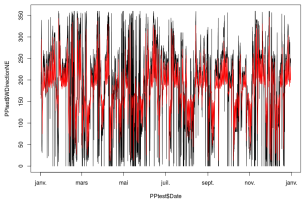
(h) Vitesse du vent à Heathrow



(i) Direction du vent à Heathrow



(j) Vitesse du vent à London Airport



(k) Direction du vent à London Airport

FIGURE 19: Résultats des prévisions des covariables

	RMSE		MAE		Ordre de grandeur	
	Mean	Median	Mean	Median	min	max
Température	4.3	3.9	3.8	3.3	26.6	98.6
Vitesse trafic	4.0	3.6	3.3	2.9	14.9	51.5
Trafic petit	72.2	63.4	60.0	51.1	2.16	719
Trafic gros	18.5	12.2	14.7	9.4	0.3	154.8
Humidité	10.6	9.8	8.9	8.1	20.5	100
Point de rosée	3.7	3.4	3.1	2.8	18.5	67.1
Pression	0.07	0.05	0.06	0.04	28.7	30.8
Vitesse vent Heathrow	3.1	2.7	2.6	2.3	0	39.5
Direction du vent Heathrow	64.3	57.1	55.7	47.8	0	360
Vitesse du vent AL	2.9	2.6	2.4	2.1	0	33.5
Direction du vent AL	71.4	65.7	62.3	56.8	0	360

TABLE 3: Erreurs sur les prévisions des covariables.

4.2 Prédiction des Oxydes d'azote

Nous allons maintenant faire des prédictions sur notre échantillon-test en utilisant les prédictions des covariables ci-dessus. Évidemment les erreurs seront plus élevées que celles présentées dans le premier cas. L'intérêt est de voir à quel point elles le sont puisqu'en pratique on n'a pas accès aux covariables au préalable. Les modèles utilisés pour prédire les variables météorologiques en pratique (chez Météofrance par exemple) sont sûrement plus complexes et plus performants que nos modèles simplement mathématiques. On obtient de meilleures performances lorsqu'on enlève la tendance et les saisonnalités au préalable. On utilisera donc cette méthode pour effectuer les prévisions.

Horizon de prévision	Jour	
Mesures d'erreurs	Mean RMSE	Med RMSE
Random Forest	72	60
Gam	74	59
GBM	72	59
GAMboost	77	61
ELMNN	78	65
Xgboost	87	75

TABLE 4: Résultats de la prévision de Nox avec prévision des covariables

5 Package

Nous avons construit un package `PjML` qui nous permet de réduire la densité de lignes de codes du programme principal et de réutiliser ces fonctions dans d'autres problèmes pour celles qui se généralisent. Ce package est composé d'un certain nombre de fonctions dont celles qui ont été faites en TP (par vous) et que l'on ne détaillera pas.

- Les fonctions `Load_data` et `Load_prevision_covariables` qui permettent respectivement de charger les données (à séparer ensuite en entraînement et test) et de charger les prévisions des covariables pour la dernière année. Elles sont présentes dans le fichier `data_import.r`
- La fonction `Analysedescriptive` qui permet pour une data.frame ou une matrice donnée de calculer les principales statistiques descriptives de chaque variable en supposant que les dates sont dans la première

colonne. Cette fonction est très simple mais se généralise bien et peut être réutilisé pour n'importe quelle étude statistique.

— La fonction `enleve_saison_tendance` qui permet d'enlever la saison et la tendance de la variable Y. Cette fonction est propre à notre problème et n'est pas généralisable.

— La fonction `moyenne_journaliere` qui permet de transformer une data.frame ayant des données heure par heure en data.frame contenant les moyennes journalières pour chaque variable. Cette fonction est généralisable pour n'importe quel problème à condition que la 1ère colonne de la data.frame contienne les dates.

— la fonction `eval_model` qui est la "grosse" fonction du package. Elle prend en paramètres un modèle de Machine Learning choisi (parmi ceux que l'on a utilisés), un horizon de prévision, un échantillon d'entrainement, un échantillon de test et la fonction de prédiction. Elle permet de calculer les prévisions à horizon donné sur l'échantillon de validation avec le modèle choisi couplé à un modèle ETS (à horizon 1) optionnel. Elle renvoie une liste avec les prédictions et les rmse de chaque horizon de prévision. Cependant cette fonction est difficilement généralisable puisqu'elle utilise les paramètres optimaux de chaque modèle pour nos données. Cette fonction nécessite les deux fonctions suivantes pour fonctionner.

— la fonction `res.block` qui permet calculer les résidus de prévision sur un block donné en s'entraînant sur le complémentaire du block. Cette fonction a été faite en TP.

— la fonction `respred` qui compile toute la série des résidus prédits pour un modèle donné, dérivée d'une fonction utilisée en TP.

— la fonction `generic_eval_model` qui permet d'évaluer un modèle entrainé au préalable en faisant des prévisions à un horizon de temps donné. Cette fonction a été utilisée pour les prédictions des covariables.

— Les fonctions `rmse` et `mae` qui sont les mesures d'erreurs que l'on a choisies et qui sont elles généralisables.

6 Conclusion

Au cours de cette étude, nous avons testé un nombre important de modèles et nous avons vu voir qu'il est en général nécessaire de diviser la prédic-

tion en plusieurs étapes. Il est intéressant de modéliser les parties tendance et saisonnalité à part par exemple ou d'ajouter des méthodes de série temporelle (SARIMA ou lissage exponentiel). Nous nous sommes de plus confrontés à la question de l'horizon de prédiction et du pas de nos données. Un bon choix d'horizon est essentiel pour la prédiction car certains modèles fonctionnent mal à horizon trop élevé alors qu'un horizon trop faible n'est pas forcément très réaliste. Nous avons de plus vu que prédire la moyenne de la quantité de NOx sur une journée était plus facile que de prédire la journée entière heure par heure. Enfin, nous nous sommes confrontés à la question de la prédiction réelle pour laquelle les covariables ne sont pas observables. Dans ce cas nous avons mis en place nos propres modèles pour prédire ces covariables et avons observé, comme attendu, que cela affecte nettement la prédiction de la variable d'intérêt.

Références

[1] Yves Aragon, " Séries temporelles avec R", edpsciences 2016

[2] Erwan Scornet, "Tuning parameters in Random Forest" ESAIM Procs, 2017,Vol. 60 pp. 144-162

[3] Pierre Gaillard "Prévision de la consommation électrique par agrégation séquentielle de prédicteurs spécialisés"

[4] Jiangshe Zhang, Weifu Ding "Prediction of Air Pollutants Concentration based on an Extreme Learning Machine : The Case of Hong Kong" 2017

[5] Rachel H. Keeler " A machine learning model of Manhattan air pollution at high spatial resolution" 2014

[6] Benjamin Hofner, Andreas Mayr, Nikolay Robinz, Matthias Schmid "Model-based Boosting in R"