MODÉLISATION ET PRÉDICTION CONSOMMATION ÉLECTRIQUE D'UN FOYER

29 mars 2020

Harol Filliat
Omar El Mellouki
ENSTA Institut Polytechnique de Paris

Table des matières

Résumé

Dans ce projet, nous avons mis en place différents modèles statistiques afin de modéliser et prédire la consommation d'un foyer. Nous avons pour ce faire analysé le jeu de données à notre disposition, composé de différentes variables comme le nombre de lumières allumées, l'humidité relative, la température intérieure et extérieure, ou l'instant de la journée. Une fois que nous avons réussi à mettre en exergue certains sous-ensembles de variables explicatives pertinents, nous avons entraîné trois modèles : un modèle de régression linéaire, un modèle non linéaire, ainsi qu'un modèle de forêt aléatoire. Nous avons finalement testé chacun de ces modèles sur un jeu de données test et nous l'avons évalué à l'aide du critère des moindres carrés. Ce rapport présentera donc nos résultats, en complément de la plateforme Kaggle sur laquelle nous avons soumis à l'évaluation nos prédictions, et du code R grâce auquel l'analyse la modélisation et la prévision ont été faites.

1 Analyse descriptive du jeu de données

La première chose par laquelle on commence est l'analyse des variables et de leur signification. Le jeu de données est constitué de 42 variables explicatives et de la variable cible *Appliances*. Ces variables représentent entre autres les mesures du temps, de la température, de l'humidité relative, le nombre de lumière allumées, ainsi que des données plus globales à l'échelle de la Belgique, et ce avec un pas de 10 minutes sur une durée globale de quatre mois et demi. Certaines sont redondantes, ou avec des formats particuliers comme des "levels" ou des formats date.

Nos données sont séparées en deux parties : un fichier d'entraînement nommé train, contenant 13 964 observations, sur lequel on construira nos modèles, et un fichier de test, contenant 5 771 observations, permettant de tester l'adéquation de notre modèle. Nous n'avons pas accès aux *Appliances* du fichier test. Une partie du fichier test se situe à des relevés qui ont été effectués entre d'autres relevés utilisés pour le train, et le reste du fichier test se situe dans le futur du fichier train.

1.1 Analyse de la variable cible

Nous commençons par visualiser la variable cible sur tout le jeu de données d'entraînement (Fig. 1), puis sur quelques jours (Fig. 2).

On observe sur la figure 1 que la consommation électrique varie énormément. En effet, elle varie de quelques dizaines de Wh à plusieurs centaines. La figure 2 permet quand à elle d'observer clairement une consommation moyenne entrecoupée régulièrement de pics que l'on se doute être la consommation aux heures de présence des membres de la famille. On voit également une périodicité dans les pics de consommation représentant le cycle journalier de la famille.

Finalement, la figure 3 représentant la consommation sur une journée permet de voir clairement que les pics de consommation sont le matin et le soir, tandis que la consommation est particulièrement faible pendant les heures de nuit entre 2h et 7h du matin, ce qui est cohérent avec l'activité d'un foyer pendant un jour commun.

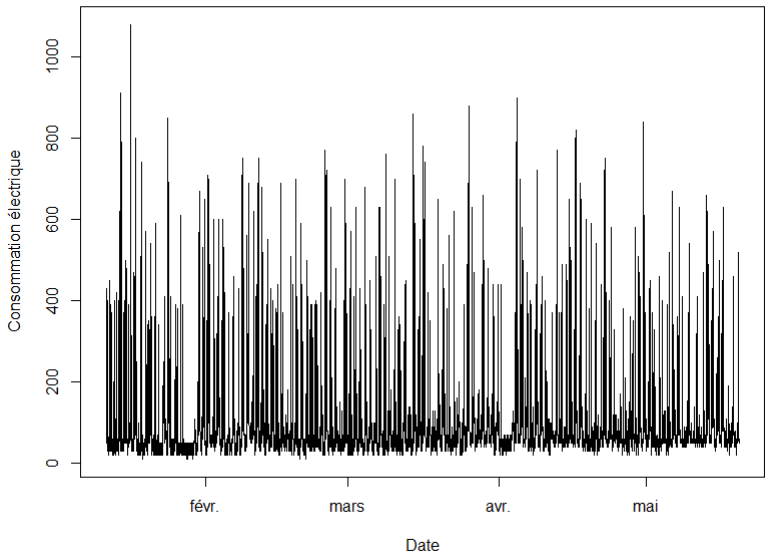

FIGURE 1 – La consommation électrique totale en Wh

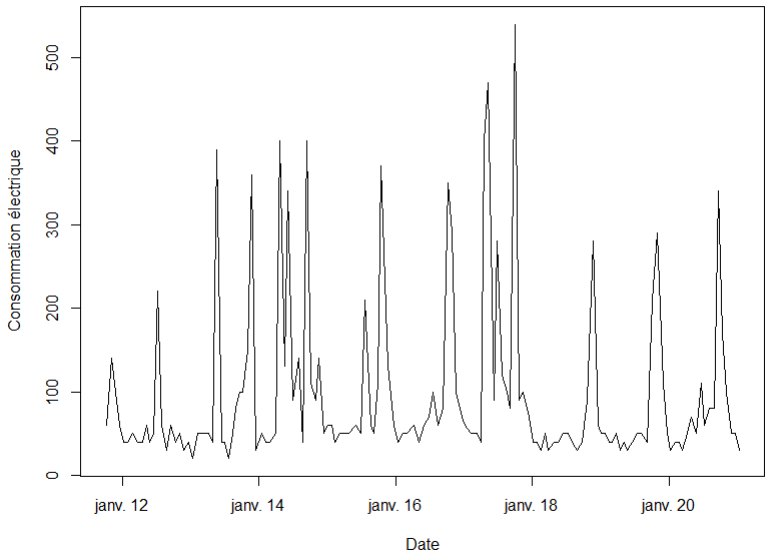

FIGURE 2 – La consommation électrique sur une semaine en Wh

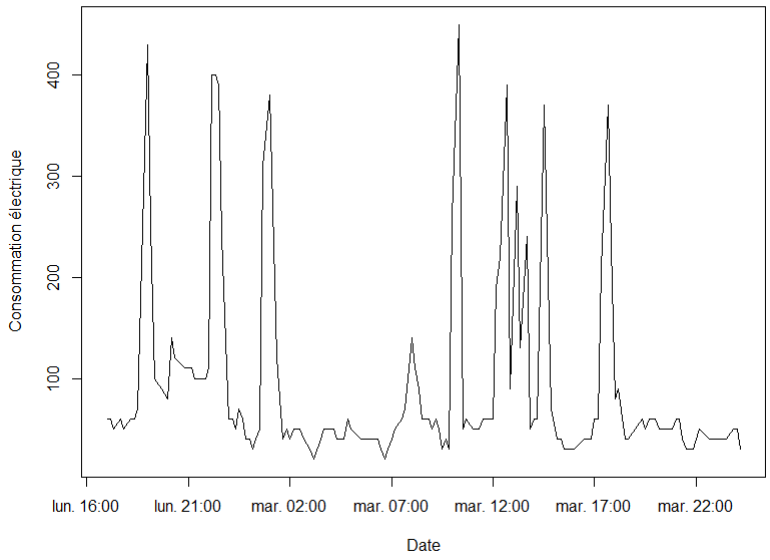

FIGURE 3 – La consommation électrique sur deux jours

Afin d'en savoir plus sur la consommation électrique nous avons également représenté figure 4 les boxplots mensuels, ainsi que journaliers, afin d'analyser les dépendances temporelles et exhiber des tendances et/ou des saisonnalités sur la durée des relevés.

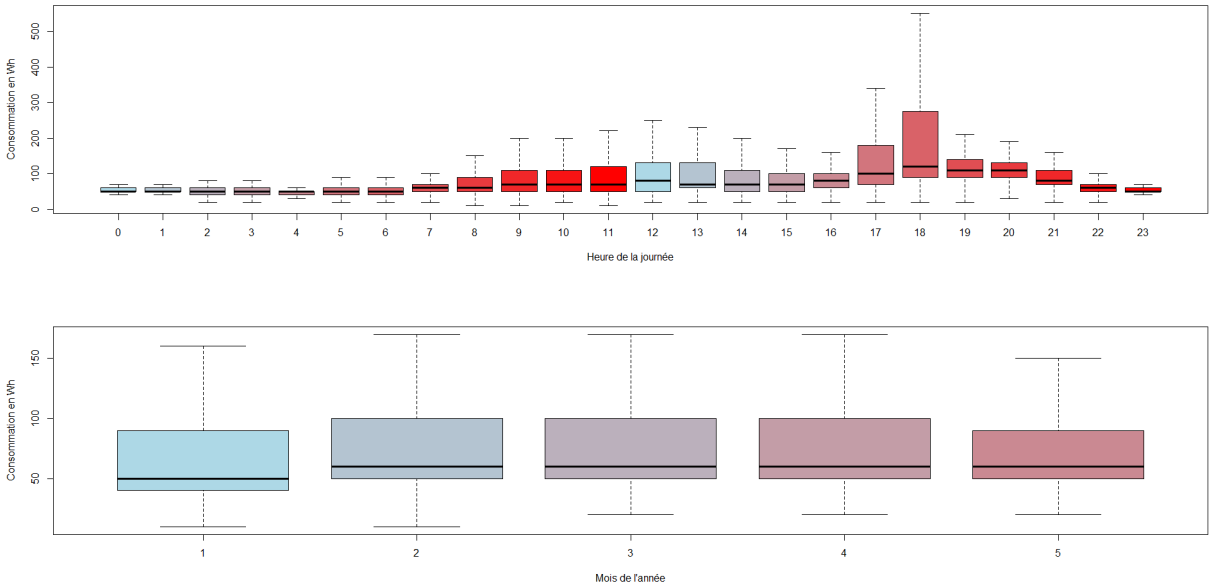

FIGURE 4 – Boxplots de la consommation horaire et de la consommation mensuelle

L'on remarque clairement que la consommation varie selon l'époque de l'année, signe que la température et les jours plus courts ou plus longs selon la saison influent

très fortement sur la consommation électrique, mais également de manière plus intéressante au niveau du graphe horaire les heures de consommation les plus fortes, qui confirment encore une fois ce qui a été observé lorsque l'on a tracé la série temporelle de la consommation électrique. Les pics de consommations sont enregistrés en moyenne en milieu de journée et fin de soirée, probablement en raison de l'activité des occupants de la maison et des différents appareils qui sont en route à ce moment là.

Nous sommes passés ensuite à une analyse des variables explicatives.

1.2 Analyse des variables explicatives

Afin d'observer les contributions des différentes variables à la consommation électrique, nous avons procédé à l'affichage de la matrice de corrélation. Une corrélation (en valeur absolue) égale à 1 signifie que deux variables sont complètement corrélées. Une corrélation proche de 0 indique que les variables sont décorrélées.

La figure 5 suivante représente les corrélations des premières variables. On voit immédiatement que notre intuition est vérifiée : les lights sont corrélées positivement à l'*Appliances*, mais nous observons également d'autres variables qui à première vue ne paraissent pas particulièrement intéressantes, comme NSM, et d'autres qui se révèlent inutiles comme Visibility.

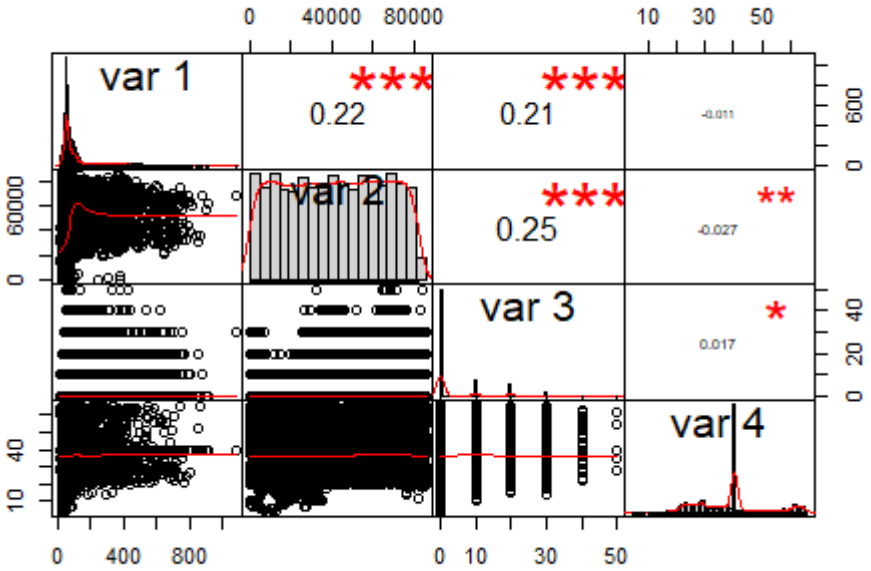

FIGURE 5 – Corrélations de Appliances, NSM, lights, Visibility

Par ailleurs, on peut étudier un groupe de variables, comme les températures (°C), ou bien les humidités relatives :

— *T1* température et *RH_1* humidité relative dans la cuisine
— *T2* température et *RH_2* humidité relative dans le salon
— *T3* température et *RH_3* humidité relative dans la buanderie
— *T4* température et *RH_4* humidité relative dans le bureau

— *T5* température et *RH_5* humidité relative dans la salle de bain
— *T6* température et *RH_6* humidité relative à l'extérieure de la maison
— *T7* température et *RH_7* humidité relative dans la salle du fer à repasser
— *T8* température et *RH_8* humidité relative dans la chambre d'adolescent
— *T9* température et *RH_9* humidité relative dans la chambre parentale
— *T_out* température et *RH_out* humidité relative à la station météorologique
— *Tdewpoint* température à la station météorologique

On peut présenter la matrice de corrélation entre les températures et *Appliances*, ainsi que celle entre les humidités relatives et *Appliances*. Nous n'affichons que la matrice des corrélations pour lesquelles les variables explicatives ne sont pas fortement corrélées entre elles (*correlation* < 0.85) et pour lesquelles les corrélations avec *Appliances* sont les plus élevées.

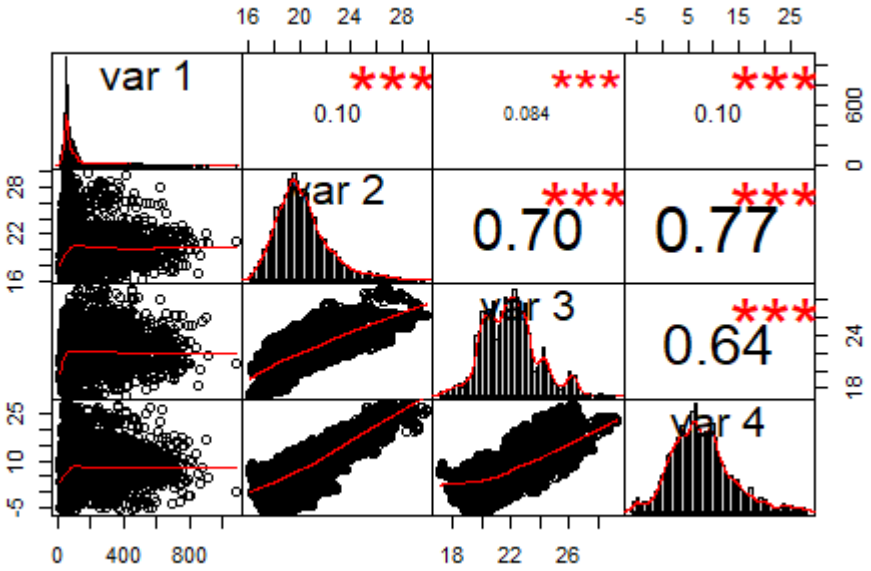

FIGURE 6 – Corrélations de Appliances avec T2, T3 et T6

En annexe, on pourra constater que *T6* est très fortement corrélé à *T_out*, ce qui est cohérent étant donné que ce sont les températures extérieures.

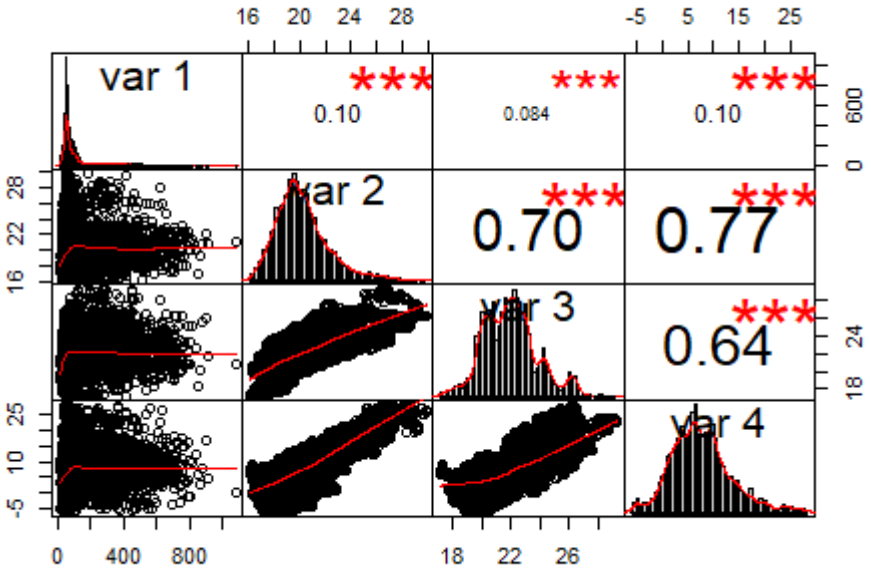

FIGURE 7 – Corrélations de Appliances avec T2, T3 et T6

Par ailleurs, on constate en annexe que la variable RH_5 est très peu corrélée aux autres humidités relatives, ce qui est cohérent car c'est la mesure effectuée dans la salle de bain. De plus, il faut noter que, comme ce sont des humidités relatives, elles sont liées à la température ambiante par la relation suivante :

$$RH(\%) = \frac{P_{vap}}{P_{sat}(T)} \times 100$$

Où la pression de vapeur saturante est croissante avec la température comme le montre la figure 8 :

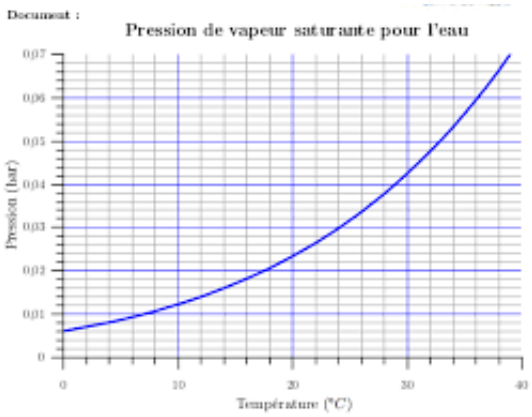

FIGURE 8 – Pression de vapeur saturante de l'eau en fonction de la température

Nous avons également effectué une analyse par composantes principales du jeu de données afin de faire apparaître des compressions pertinentes des 42 variables à disposition, dans le but de réduire la dimension du modèle. Cette approche à la fois

géométrique et statistique nous permet de représenter dans le plan des directions principales les variables dans ce système de coordonnées qui maximise l'inertie, afin notamment de faire apparaître des regroupements pertinents. La figure 9 montre la représentations des variables explicatives du modèle dans le premier plan des deux directions de plus grands pourcentages d'inertie (29.32% et 18.11%).

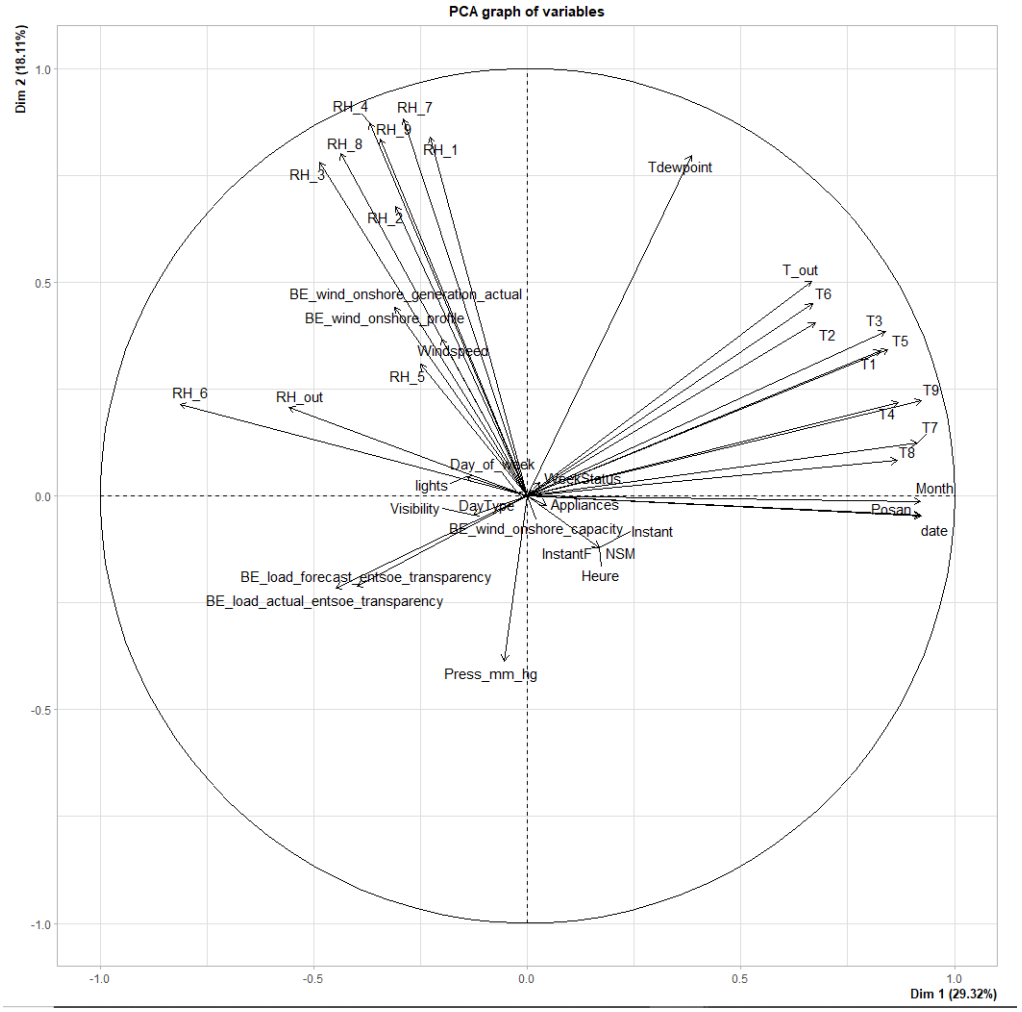

FIGURE 9 – Représentation du nuage des variables dans le premier plan principal

Finalement, la dernière analyse préliminaire que nous avons effectuée est celle se basant sur l'algorithme de Boruta. Cet algorithme est un procédé de classement des variables selon leur importance qui se base sur la création de variables copies "mélangées" de celles dont nous disposons dans le jeu de données et de maximisation du "Z-score", un critère comparant la distance d'une population à sa moyenne. La sortie est représentée figure 10 et représente dans l'ordre croissant l'importance des variables dans le jeu de données.

Pour plus de lisibilité nous donnerons ici la liste des variables dans l'ordre croissant d'importance selon l'algorithme de Boruta : rv1, rv2, WeekStatus, BE-wind-onshore-profile, Month, Windspeed, BE-wind-onshore-generation-actual, RH-8, T6, BE-load-forecast-entsoe-transparency, RH-out, Visibility, T1, RH-7, RH-2, RH-1, T4, T9, T-out, Posan, RH-5, RH-4, DayType, Day-of-week, RH-6, T2, T7, Heure,

RH-9,T8, BE-load-actual-entsoe-transparency, Press-mm-hg, date, Tdewpoint, T5, T3,RH-3, lights, InstantF, Instant, et NSM. Nous voyons immédiatement que l'algorithme est cohérent avec la matrice de corrélation. En effet, la variable la plus importante semble être NSM, qui est aussi celle avec la corrélation la plus importante avec Appliances. On voit également que les variables les moins importante sont rv1 et rv2, ce qui est logique étant considéré que ces dernières sont des variables complètement aléatoires qui n'ont par conséquent aucun lien avec la variable cible. Les boîtes à moustache représentent les valeurs de l'importance calculée à chacune des 100 itérations de l'algorithme.

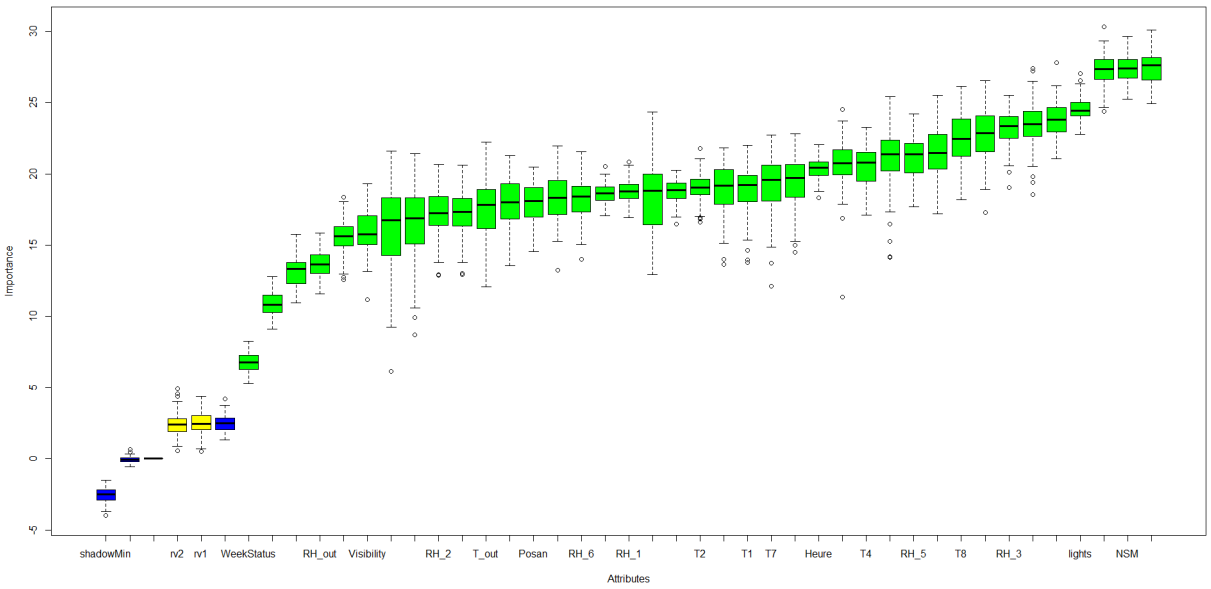

FIGURE 10 – Sortie de l'algorithme de Boruta au bout de 100 itérations avec 500 arbres chacune

Nous disposons finalement au bout de cette analyse d'une première idée des variables les plus pertinentes dans notre étude. Nous allons donc nous baser sur cela, ainsi que sur d'autres algorithmes de sélection de subset que nous détaillerons plus tard, afin de construire nos différents modèles.

2 Traitement des données

2.1 Valeurs manquantes

La première étape dans le traitement de nos données a été de compléter les valeurs manquantes de certains relevés. En effet, entre les variables RH_6 et $Visibility$, il y a environ 200 valeurs manquantes, que l'on remarque en appelant la fonction *summary*, de base en R. Nous allons donc extrapoler ces valeurs manquantes à l'aides des autres valeurs dont nous disposons. Une première approche évidente est celle de remplacer naïvement les valeurs manquantes par la moyenne. Cette méthode

a l'avantage d'être très simple à implémenter mais ne permet pas de prévoir les fortes variations autour de la moyenne. Cependant, nous pouvons imaginer une méthode plus pertinente qui consiste à exploiter la forte corrélation entre l'humidité en extérieur et l'humidité à l'intérieur du bâtiment et à établir un modèle à base de forêts d'arbres décisionnels (que nous expliquerons plus en détail dans le chapitre consacré) basé sur les autres variables d'humidité. De même pour la visibilité, que nous approximerons par le biais de d'autres variables pertinentes dans ce cas là. Nous avons également retiré les deux variables $rv1$ et $rv2$ qui ne sont en fait que des variables complètement aléatoires.

2.2 Étude du domaine à prévoir

Lorsque l'on regarde notre jeu de données, nous remarquons que le pas de temps de dix minutes n'est pas toujours respecté. Il suffit d'afficher les premières lignes de la data frame pour remarquer que par exemple l'entrée correspondant au 11 janvier à 17h30 n'est pas présente. Ce manque dans les données est cependant de nature différente de celui que nous avons observé pour l'humidité ou la visibilité, où certains relevés d'une variable manquaient alors que le reste était disponible. Cette fois-ci, c'est toute la ligne de la data frame qui n'est pas présente. Cela s'explique cependant en s'intéressant au jeu de données test sur lequel il faudra faire la prédiction, car celui-ci présente exactement les lignes manquantes du jeu de données d'entraînement. On peut donc logiquement supposer que le jeu de données test a été construit en deux parties : une première partie de prédiction pure de la consommation électrique sur la semaine suivante, et une partie de prédiction de données choisies aléatoirement dans le jeu de données d'entraînement, comme l'on ferait pour une vérification par validation croisée. On doit donc travailler de deux manières différentes sur ces deux parties à prévoir. Sur la partie construite par sélection aléatoire d'individus du jeu de données d'entraînement, il faut extrapoler les valeurs manquantes en prenant en compte les valeurs environnantes. Nous avons traité cette problématique de deux manières différentes, la première étant une interpolation naïve, par la droite liant les points existants, et une deuxième par un modèle de forêts aléatoires en rajoutant comme variable explicative la série temporelle de l'Appliance décalée de plus et moins 1. Nous présenterons cette méthode plus tard dans la section implémentation du modèle de forêts aléatoires.

En ce qui concerne l'interpolation, nous avons complété la série temporelle des Appliances du jeu de données d'entraînement par les entrées du jeu de données test, créant ainsi une série temporelle avec des données manquantes. Nous avons ensuite interpolé linéairement les données manquantes de la consommation à l'aide des données disponibles. L'avantage de cette méthode est qu'elle est facile à implémenter et qu'elle prend en compte les contributions des appliances environnants au point manquant, et par conséquent implicitement les variables explicatives. Elle est également plus précise qu'une méthode qui consisterait à remplacer simplement par la moyenne globale de l'appliance et qui comporterait une grosse erreur d'approximation. Cependant, cette méthode a le désavantage de ne pas traiter efficacement les valeurs très éloignées de la moyenne. De plus, dans le cas où il faudrait approximer deux individus successifs ou plus, cette méthode n'est pas efficace. On peut le voir

illustré par exemple dans la figure où l'on observe soudainement un pic. Malheureusement, ce genre de phénomènes est commun : par exemple, entre 18h et 18h20, les membres de la famille reviennent de l'école et du travail et la consommation saute soudainement d'une valeur relativement faible à une valeur bien plus élevée. On a représenté figure 11 un exemple d'une situation où cette approximation est imprécise.

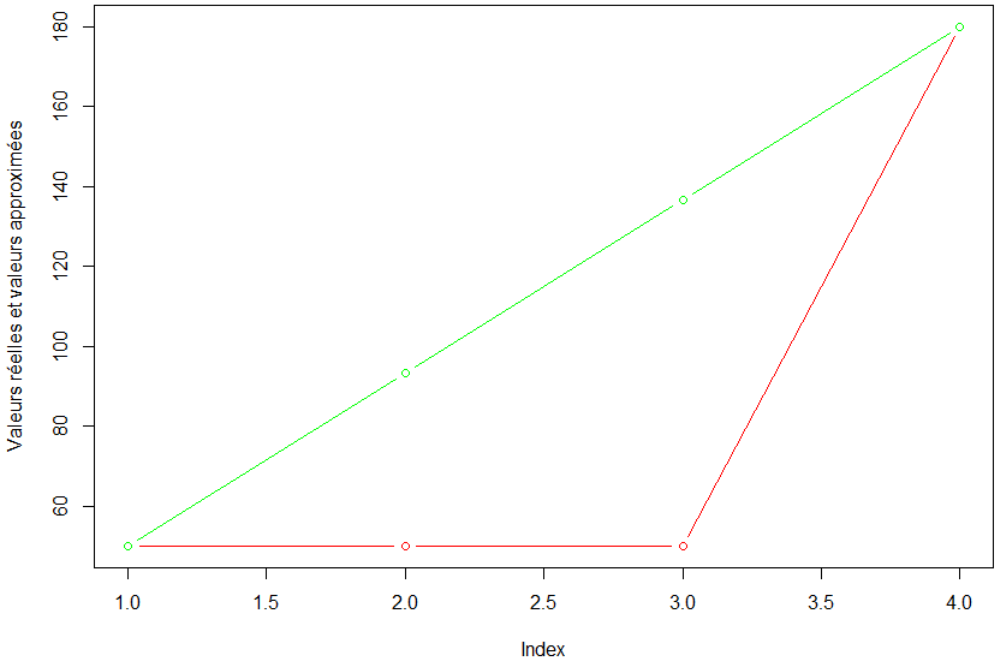

FIGURE 11 – En rouge, les valeurs réelles, et en vert les valeurs extrapolées par interpolation

3 Modèles linéaires

Le modèle par lequel l'on a commencé notre étude est un modèle linéaire. Nous n'attendions pas beaucoup de ces modèles dont la prédictivité est assez limitée pour des problèmes aussi complexes, mais il était nécessaire de passer par ces modèles simples afin de nous faire une première idée.

3.1 Théorie de la Modélisation linéaire

Le principe est le suivant : étant donné une variable cible Y de dimension 1 et, par exemple, deux variables explicatives X_1 et X_2 de dimension 1 également, on pose ϵ qui suit une loi normale centrée de variance σ^2 et on a alors le modèle :

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \epsilon$$

Les observations de Y, X_1 et X_2 permettent alors de déterminer les coefficient b_0 (l'intercept), b_1 et b_2. Plus précisément, ils sont issus de la minimisation de la somme des carrés des résidus entre l'observation Y et son estimation $\hat{Y} = b_0 + b_1 X_1 + b_2 X_2$:

$(\hat{b_0}; \hat{b_1}; \hat{b_2}) = \underset{b_0,b_1,b_2}{argmin} \sum_{i=1}^{n} (Y_i - b_0 - b_1 \times X_{1i} - b_2 \times X_{2i})^2$

Posons g la fonction qui, pour un vecteur Y et une matrice X donnés et fixés, correspondant aux observations respectivement de la variable cible et des variables explicatives, associe le coût $g : \beta \longrightarrow g(\beta) = ||Y - X\beta||^2$.

Alors on sait que g est convexe, et que lorsque X est de rang plein, la solution au problème de minimisation de g est $\hat{\beta} = (X'X)X'Y$ et cette solution vérifie $\frac{dg}{d\beta}(\hat{\beta}) = 0$ avec $\frac{dg}{d\beta}(\beta) = -2X'Y + 2X'X\beta$. Par ailleurs $\hat{\beta}$ est un estimateur sans biais de variance minimale. Ainsi, pour réaliser une prédiction y sur la base de nouvelles observations x, on obtient $y = x\hat{\beta}$.

Pour savoir si notre modèle est significatif, lors de l'appel de *lm*, nous avons une statistique de Fischer *F-statistic* qui teste l'hypothèse H_0 : "absence de significativité globale du modèle". Par exemple, en se donnant une p-value seuil de 5% alors si on obtient une p-value $< 5\%$, dans ce cas on rejette H_0 et donc notre modèle est globalement significatif. Dans le cas contraire, on rejette une par une les variables non adaptées. Pour cela, on regarde dans le tableau des coefficients la p-value $Pr(> |t|)$ du test de Student sur la significativité de la variable en question. Pour un seuil de 5%, si cette probabilité $Pr(> |t|)$ est supérieure à ce seuil de 5%, alors on rejette l'hypothèse de significativité de la variable : elle n'est pas significative, donc on peut la supprimer du modèle. Par ailleurs, la fonction *predict* permet de prédire la variable Y sur la base du modèle déterminé et de nouvelles observations de X_1 et X_2. Enfin, nous pouvons récupérer les résidus par la commande *residuals*. On peut ainsi tracer un *qq plot* permettant de visualiser la répartition des quantiles estimés en fonction de la répartition des quantiles théoriques de la loi normale. Un bon modèle doit, en plus, avoir un *qq plot* proche de la première bissectrice ($residuals = fitted$).

Cependant, le choix de la régression linéaire via *lm* pour l'hyptohèse des modèles linéaires présente l'inconvénient de prendre le plus de variables explicatives possibles, ce qui en grande dimension amène au phénomène de sur-apprentissage (le modèle colle aux observations de la variable cible, mais est incapable d'interpoler) et à la lenteur de l'algorithme. Pour contrer cela, on peut s'orienter vers le critère de sélection AIC (Akaike Information Criterion). On peut augmenter la vraisemblance d'un modèle, c'est-à-dire sa qualité en ajoutant des paramètres. Mais comme dit précédemment, on souhaite éviter le sur-apprentissage et donc pénaliser un trop grand nombre de variables. Le critère AIC à minimiser est le suivant :

$$AIC = 2k - 2ln(L)$$

où k est le nombre de paramètres du système et L la vraisemblance du modèle. Le critère AIC est un critère de qualité relative du modèle (et non absolue). Ce critère est similaire au critère BIC (Bayesian Information Criterion) à la différence que ce dernier pénalise également la taille d'échantillon n :

$$BIC = -2ln(L) + kln(n)$$

3.2 Présentation des résultats

Afin d'ajuster notre modèle linéaire il a fallu choisir un subset de variables explicatives. Nous avons pour cela procédé de manière combinatoire, en explorant les différentes combinaisons de variables avec comme critère la minimisation de l'AIC. Nous avons pour cela utilisé comme jeu de données d'entraînement une partition de 85% du jeu données de base puis nous avons validé le modèle sur les 15% restants.

L'algorithme, executé avec la fonction **stepAIC** du package **MASS**, va alors produire au bout de 10 itérations un modèle linéaire et un jeu de variables qui minimisent le critère AIC à 113015,1. Nous avons ensuite testé ce modèle sur le jeu de données témoins.

Nous observons sur la figure un extrait de la prédiction ainsi que les valeurs réelles que nous avions préalablement retirées.

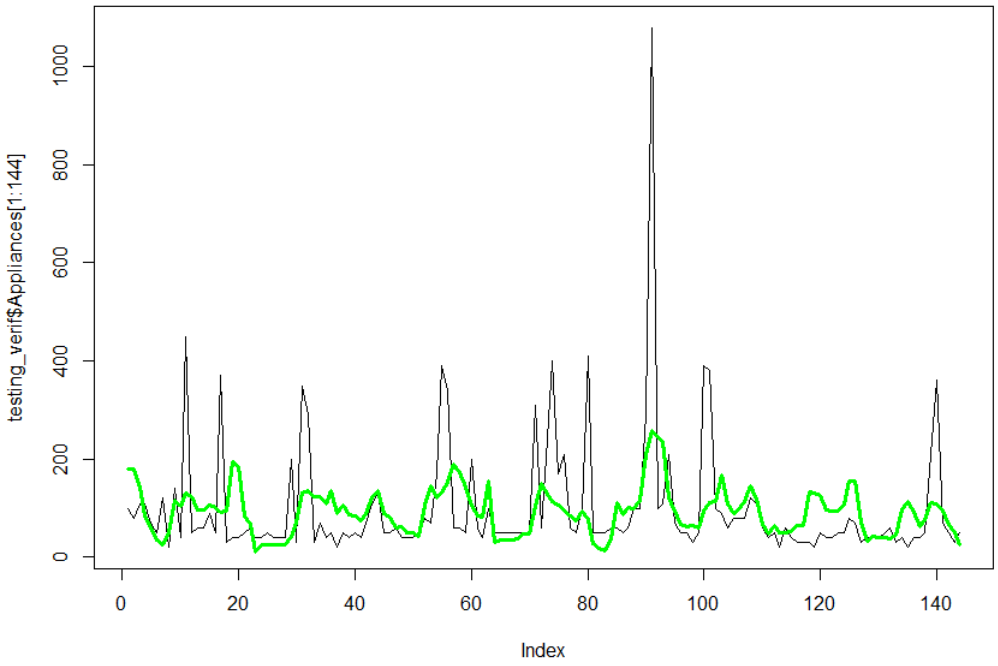

FIGURE 12 – Représentation d'un jeu de données ainsi que de sa prédiction par un modèle linéaire

Comme prévu cette prévision n'est pas très bonne. Tout d'abord de manière absolue, on remarque un grand manque de précision au niveau de la prédiction. Les pics de consommation ne sont pas bien représentés, et la prévision ne suit pas bien les variations du jeu de données. Globalement, le modèle semble aussi avoir surestimé la prévision. De manière relative, nous avons calculé un RMSE de **86.024** en validation croisée. Nous allons voir plus tard que ce score est loin d'être aussi bon que celui des modèles suivants. De même sur le jeu de données test, nous calculions un RMSE de 95.92813, score que nous améliorerons énormément par la suite à l'aide de modèle plus poussés.

4 Modèles non linéaires

4.1 Théorie de la Modélisation Non Linéaire

Le modèle *GAM*, generalized additive models, est une généralisation du modèle linéaire à des modèles non linéaires, mais additifs :

$$Y = b_0 + f(X) + \epsilon, \text{ où } \epsilon \sim N(0, \sigma^2)$$
$$ln(Y) = b_0 + f(X) + c\gamma + \epsilon \text{ où } \gamma \sim N(0, \Sigma^2) \text{ et } \epsilon \sim P(\lambda)$$

En général, on choisit la fonction f parmi une base de fonctions prédéfinie, qu'on appelle base de splines, comme sur la figure 13. De même on peut utiliser la fonction *predict* pour un modèle gam.

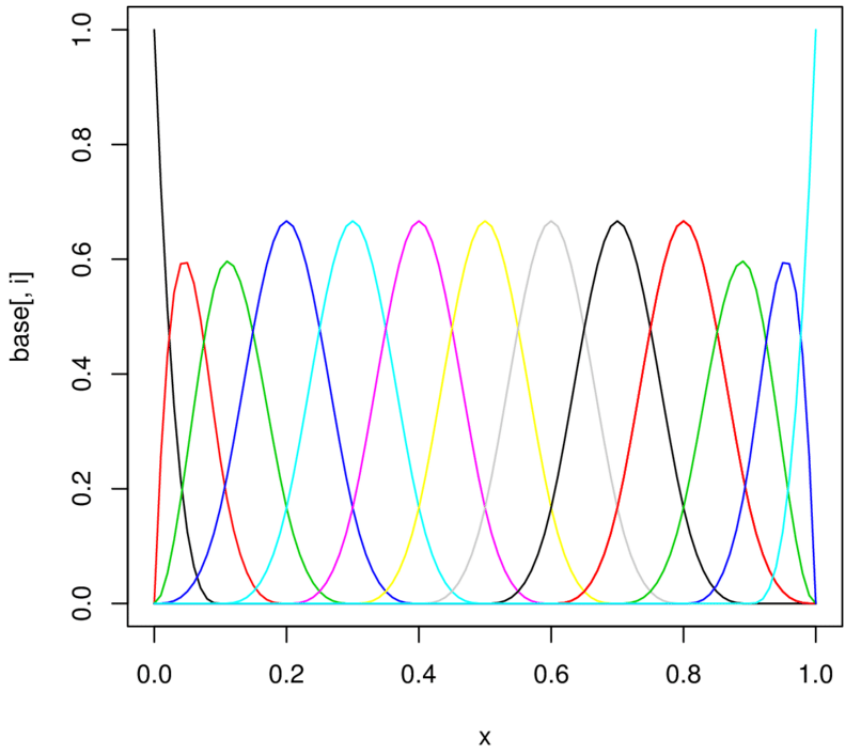

FIGURE 13 – Base de splines cubique avec les noeuds équidistants

4.2 Présentation des résultats

Pour le modèle non linéaire nous nous sommes basés sur une approche plus heuristique. En effet, nous avons trouvé les modèles non linéaires plus difficiles à appréhender d'un point de vue pratique car leur ajustement requiert de l'expérience que nous n'avons pas forcément encore. Nous avons essayé de mettre en place un

modèle par choix de variables pertinentes et en essayant de trouver des regroupements pertinents. Nous avons donc proposé le modèle dont voici une prédiction sur le même jeu de validation croisée que le dernier.

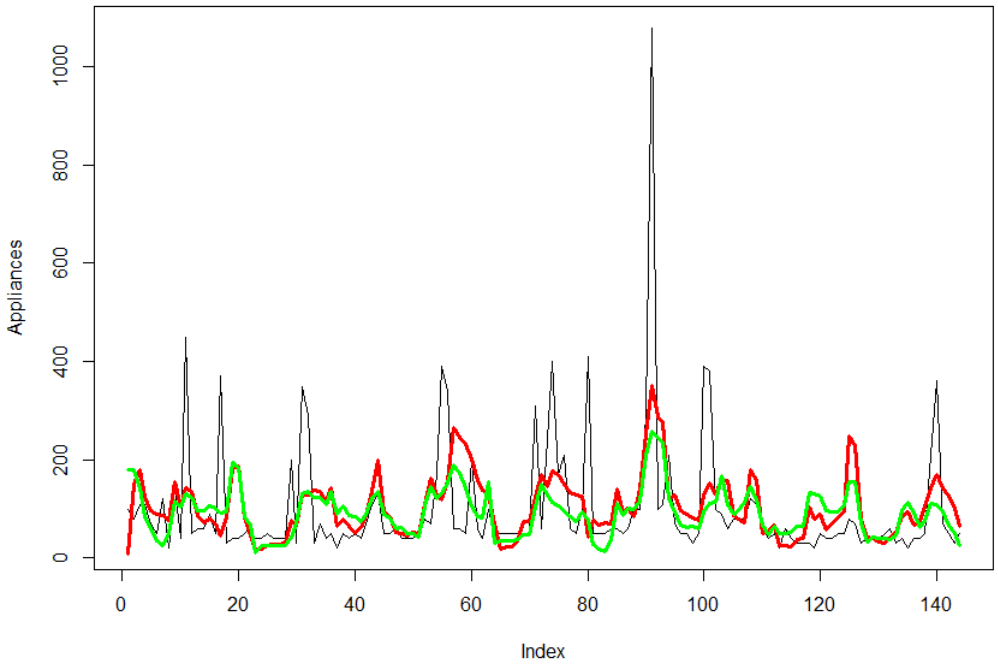

FIGURE 14 – En rouge le graphique d'une prédiction par un modèle gam

On observe une légère amélioration du RMSE sur le jeu de données de validation croisée, passant de **90** pour le modèle linéaire à **88** pour le modèle non linéaire. Ce score n'est sans doute pas optimal, car nous avons eu du mal avec ce modèle là, qui requiert une plus grande maîtrise afin d'ajuster tout les paramètres. Nous nous sommes enfin intéressés à un modèle de forêts aléatoires qui a été beaucoup plus performant. C'est celui sur lequel nous nous sommes concentrés dans cette étude.

5 Modèles de forêts aléatoires

Le principe de la modélisation par forêts aléatoire repose sur la technique dite des arbres de décisions en tant que modèle prédictif. Au noeud n, si ce n'est pas la feuille, on choisit une variable explicative via un certain processus de choix et on la teste, puis la branche correspond au résultat de test. Les feuilles sont les valeurs de la variable ciblée Y ou une ditribution de probabilité sur les valeurs possibles de la variable cible Y. On peut paramétrer le nombre d'arbres générés (*ntree*) et le nombre de variables testées à chaque séparation (*mtry*).

5.1 Approche Théorique

Il y a deux types de variable cible : les variables qualitatives (exemple : une variable avec des levels) et on dit que l'arbre est dit arbre de classification, puis les variables quantitatives (comme ici *Appliances*) pour lesquels l'arbre est dit arbre de régression.

Pour revenir sur le choix de la variables explicative sur un noeud, l'algorithme teste les différentes variables explicatives possibles, puis sélectionne sur la base d'un critère. On cite pour les arbres de classification les critères de l'entropie de Shanon, ou l'indice de diversité de Gini qui mesure le nombre de mauvais classement de variable cible Y dans les catégories possible. Il faut alors minimiser ce critère. Concernant notre cas, les arbres de régression, le critère utilisé est celui du test du chi-deux. Il s'agit de maximiser la variance entre différentes classes de modèles de l'arbre (les valeurs de la variable cible doivent être éloignées entre différentes classes). En effet, les modèles par forêts aléatoires sont instables dans le sens où une modification du jeu de données change beaucoup l'arbre, en particulier si l'on change des valeurs des données proches de la racine de l'arbre d'origine (exemple : changer certaines valeurs de la température T6, alors que T6 est au premier noeud d'un arbre de décision). Les arbres ont alors des variances élevées entre eux, et on souhaite réduire cette variance, pour stabiliser l'arbre généré par rapport au jeu de données. Le but est donc de moyenner différents arbres en effectuant un tirage parmis les données (on choisit une partie des données aléatoirement), puis on construit un arbre sur ces données et enfin on effectue une moyenne des arbres générés.

Les arbres d'apprentissage présentent le problème d'un sur-apprentissage du modèle, c'est-à-dire qu'en prenant trop de variables explicatives en compte, le modèle va coller à la variable cible parfaitement mais être incapable d'extrapoler à de nouvelles données et le modèle risquera d'être instable. Par exemple sur notre jeu de données train, contenant 13 964 observations, alors un nombre de feuilles de 13 964 provoquera un sur-apprentissage maximum : le modèle sera parfaitement égal aux valeurs de la variable cible sur le fichier train, mais instable sur le fichier test, et la prédiction issue du modèle sera mauvaise car le modèle sera difficile à interpoler. Les algorithmes de forêts aléatoire permettent de gérer ce problème issu des modèles à arbres de décisions, en infligeant une pénalisation sur le nombre de variables prises en compte.

5.2 Présentation des résultats

Nous avons pour le modèle d'arbres aléatoires utilisé deux modèles. Le tout premier avait pour équation celle du modèle linéaire calculé par sélection StepWise : le but était pour nous de nous faire une idée de la différence de performance entre le modèle linéaire et le modèle par arbres aléatoires. Le second modèle avait pour but d'améliorer la partie de prédiction pure en choisissant des variables plus pertinentes, basées sur l'étude préliminaire.

5.2.1 Prédiction sur la première partie du jeu de données test

Comme expliqué dans la partie traitement des données, nous avons extrapolé de deux manières les données :

Premièrement de manière naïve à l'aide d'une interpolation linéaire. Nous obtenions des résultats satisfaisants qui nous ont permis de grandement réduire notre erreur de prédiction.

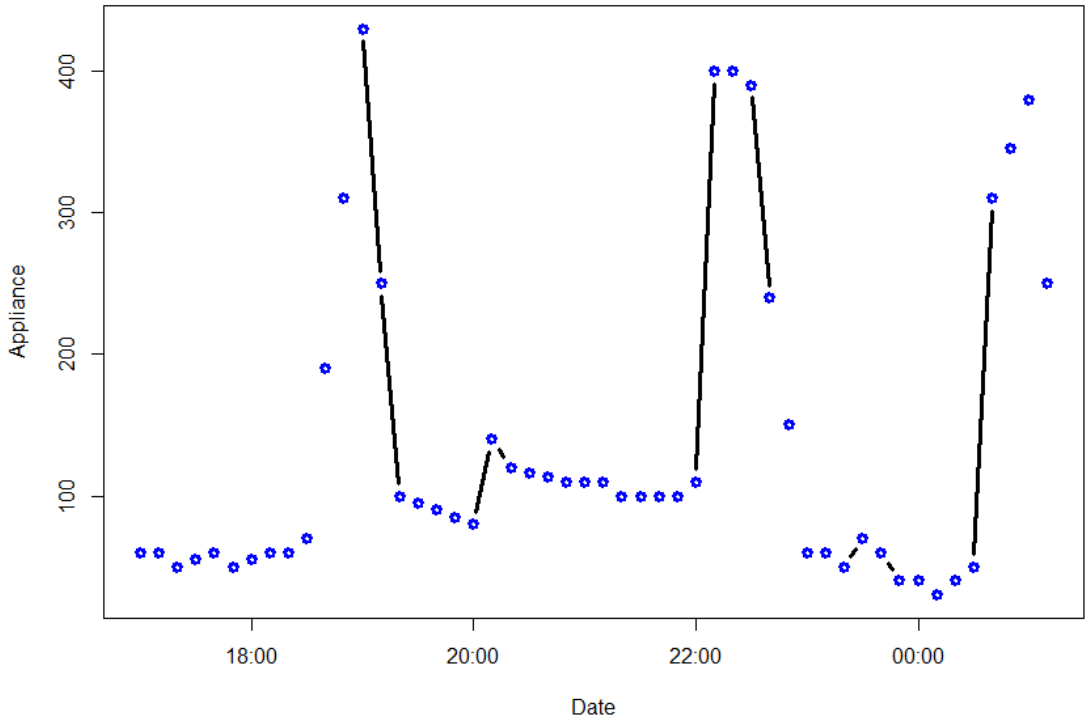

FIGURE 15 – Tracé sur la partie interpolation du jeu de données d'un extrait de l'extrapolation linéaire naïve avec en bleu les points extrapolés

Les résultats semblent satisfaisant, bien qu'ils présentent les mêmes approximations qu'évoqué précédemment. Nous avons quand même grâce à cela réussit à améliorer notre prédiction pour passer d'un RMSE de 79 sans cette interpolation, c'est à dire par une prédiction à l'aide de la forêt aléatoire sur tout le jeu de données test, à 74 avec l'interpolation.

Voici le tracé de ces valeurs :

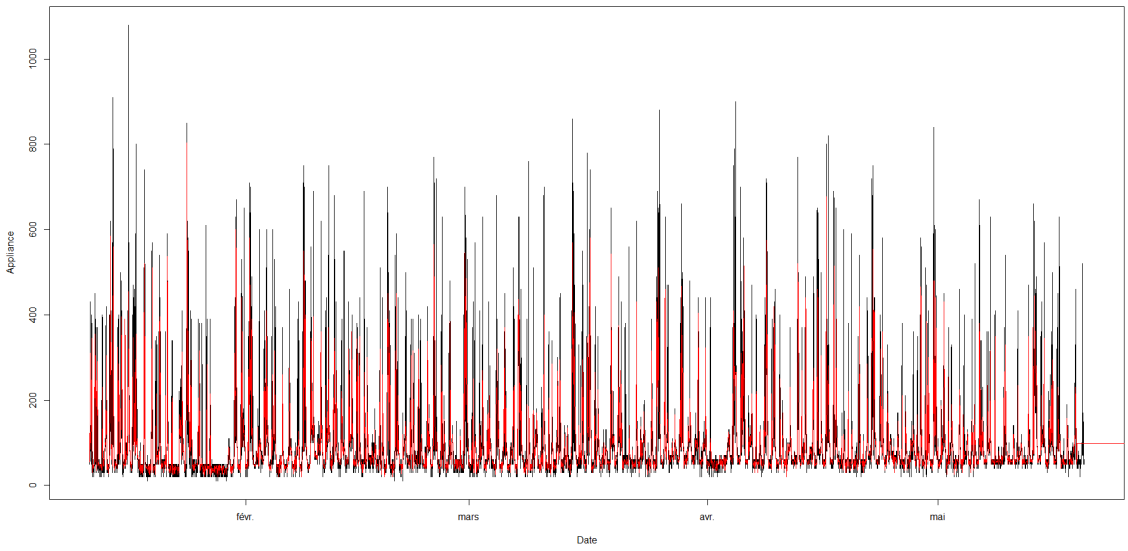

FIGURE 16 – Le jeu de données d'entraînement complété par les valeurs prédites par interpolation

Cependant, nous avons également implémenté une méthode de prédiction moins naïve permettant de considérer en plus des valeurs précédente et suivante les contributions des variables explicatives. Cependant, cette méthode n'a pas vraiment amélioré notre prédiction, car elle était légèrement moins performante que la méthode naïve. Nous n'avons pas vraiment compris pourquoi cependant.

5.2.2 Prédiction sur la semaine suivante

Une fois la première partie prédite, nous avons utilisé notre modèle de forêt aléatoires sur la deuxième partie de prédiction pure. Les résultats étaient très satisfaisants, et le modèle présentait en validation croisée un RMSE de **29**, par conséquent meilleur que tout les modèles que nous avions jusque là.

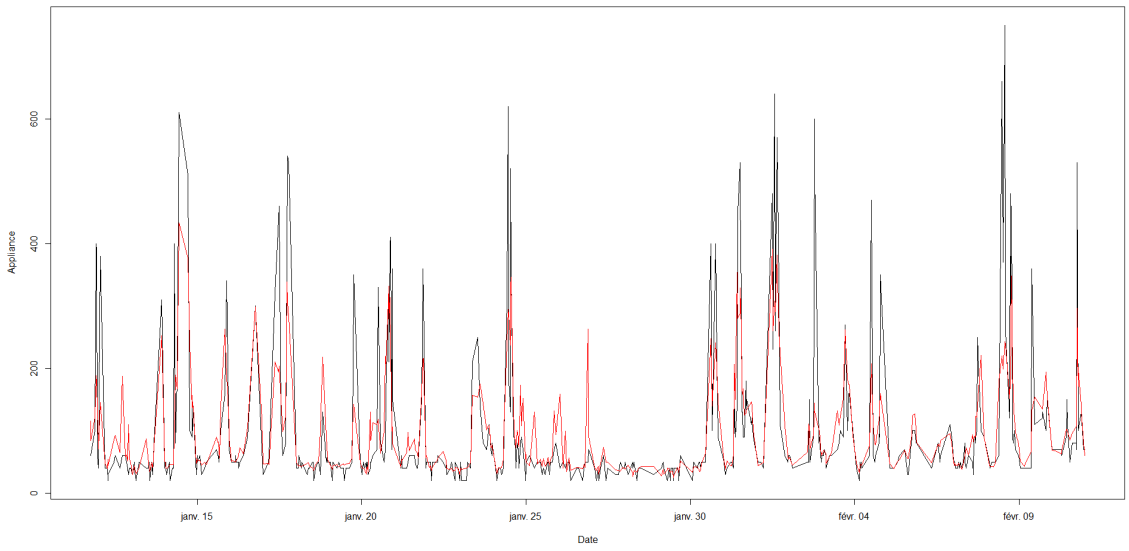

FIGURE 17 – Prédiction (en rouge) sur un échantillon de validation croisée de l'Appliance à l'aide d'un Random Forest

On trouve un RMSE de **29** sur le jeu de données de validation croisée, un score bien meilleur que ceux que nous avons eu jusque là. Nous voyons que la courbe prédite suit bien les variations réelles, bien qu'elle ait tendance à sous estimer les pics de consommation. Nous pouvons également observer le résultat de la prédiction sur le jeu de données test dans la figure suivante :

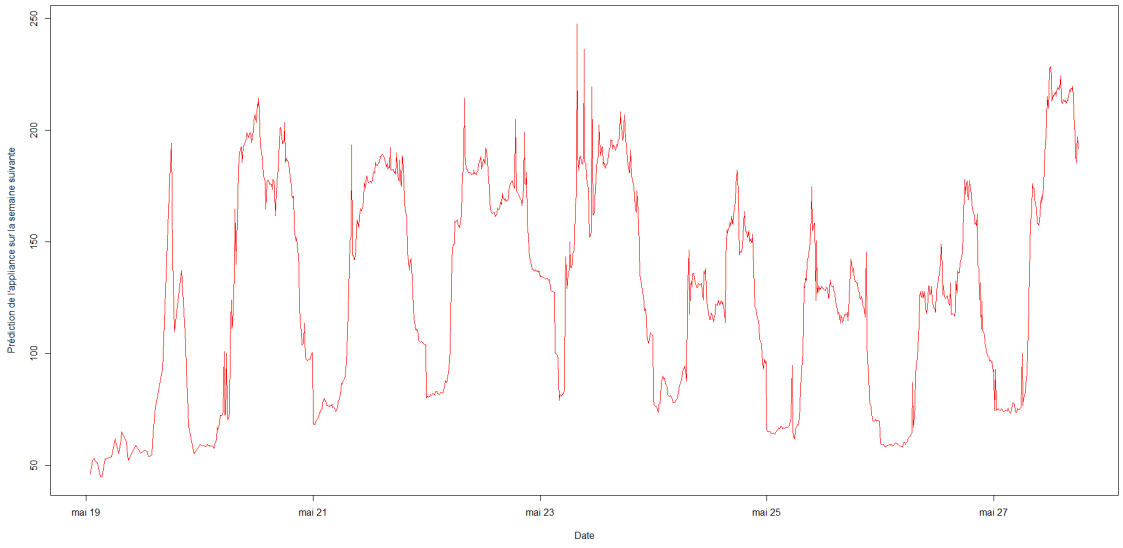

FIGURE 18 – Notre prédiction de la consommation électrique du foyer sur la semaine du 19 au 27 mai

6 Conclusion et résumé de nos résultats

Le tableau suivant résume nos résultats :

Modèle	Équation	RMSE
Linéraire	lights + T1 + RH_1 + T2 + RH_2 + T3 + RH_3 + RH_4 + RH_5 + T7 + RH_7 + T8 + RH_8 + T9 + RH_9 + T_out + Press_mm_hg + Windspeed + Tdewpoint + Day_of_week + InstantF	95.92813
Non Linéaires	InstantF + lights + s(T3,T9,T8,T2)+ Day_of_week + s(RH_3,RH_2,RH_1,RH_8,RH_5,RH_4)+ T_out + Press_mm_hg + Windspeed + Tdewpoint + Day_of_week+ InstantF +NSM	95.59870
Forêt Aléatoire	lights+T1+RH_1+T2+RH_2+T3 +RH_3+ T4+RH_4+T5+RH_5+T6+RH_6 +T7+ RH_7+T8+RH_8+T9+RH_9+T_out+ Press_mm_hg+RH_out+Windspeed+ Tdewpoint+NSM+ WeekStatus+ Day_of_week	73.35837

En conclusion, nous avons énormément appris lors de ce projet. Nous avons eu l'occasion de découvrir la démarche du data scientist et de l'appliquer en partant d'un jeu de données inconnu, avec plusieurs dizaines de variable et plusieurs dizaines de milliers de mesures, et en arrivant à atteindre un objectif qui nous semblait très lointain au début du projet. Nous avons également découvert plusieurs aspects très importants du traitement de problèmes en lien avec des données, ainsi que beaucoup des difficultés que l'on y rencontrait.

Nous avons lors de cette étude étudié énormément de méthodes en plus de celles proposées en cours, et nous avons notamment exploré les méthodes de gradient boosting ou d'ajustement répété de modèles, par le biais de la fonction **train** du package **caret**. Nous avons cependant préféré nous concentrer sur les méthodes du cours que nous comprenions mieux, et améliorer notre maîtrise de ces outils déjà bien assez performants.

Nous avons également identifié un certain nombre de voies d'amélioration. Nous n'avons par exemple pas pu complètement exploité l'analyse par composantes principales, notamment car nous venons de commencer le cours. Nous avons également essayé d'implémenter des variables qui résumeraient les variables corrélées de température et d'humidité, mais devant les difficultés rencontrées lors de l'implémentation nous avons préféré nous concentrer sur la mise en place d'un modèle avec simplement les variables de base. De même pour le traitement de la variable InstantF, dont les facteurs de nuit sont apparus comme non significatif lors de l'étude.

Enfin, nous sommes très contents, et même fiers, du travail que nous avons fourni et espérons à l'avenir pouvoir améliorer nos compétences et continuer à viser les scores les plus faibles.