Modélisation Prédictive Rapport

Valentin Gölz, Laura Fuentes

February 14, 2023

Contents

Introduction **2**

Notre jeu de données **2**

Choix de variables: **2**

Choix du type de modèle **2**

Introduction

Nous avons un jeu de données regroupant différents variables en rapport avec la consommation énérgétique française pendant la période de 2012 à 2021. Le but ici est de construire un modèle qui permettant de prédire la consommation française en énergie pendant la période du Covid. \ Le premier reflèxe est ici de télécharger l'ensemble des packages dont on fera usage lors du développement des différents modèles. Nous avons également divisé le set train en deux pour pouvoir tester nos modèles avant de les soumettre. Nous avons ainsi choisi la période de 2012 - 2019 comme train et 2019-2020 (mars) comme test.

Notre jeu de données

On télécharge ensuite les deux jeux de données train et test déjà modifiés. Ces modifications consistent en des ajouts de variables. Nous avons d'une part recupéré les différents mouvements sociaux et le pourcentage de population mobilisée. Ensuite nous avons créée une variable mesurant la température ressentie. Le problème de cette dernière variable concernait les nombreuses valeurs NA's, ainsi que la représentativité au niveau national des stations météorologiques constituant le jeu de données. Nous avons également, la variable WeekDays2. Il s'agit d'une version modifiée de la variable WeekDays qui distingue les jours laborales, samedis et dimanches. Enfin, nous avons pensé à comment implémenter l'effet du Covid sur le modèle. En effet, en rajoutant la variable GovernementResponseIndex on avait des très mauvais résultats. Ceci s'explique du fait que la variable comprends des valeurs nulles pendant des années, et celles-ci explosent dans une courte période d'un mois, laissant peu de temps d'entrainement sur la pandémie. Pour simuler le comportement de la population pendant le confinement avec les données que l'on avait déjà, nous avons pensé aux samedis. En effet, nous avons mis l'hypothèse qu'un jour de confinement était comparable en termes de consommation à un jour de weekend comme un samedi. Dans cet esprit, nous avons créé la variable WD, qui modifie le jour de la semaine à samedi s'il y a confinement (la GovernementResponseIndex>=70), et maintien des jours de la semaine sinon.

```
load("Data/Data0.Rda")
load("Data/Data1.Rda")

sel_a <- which(Data0$Year<=2019)
sel_b <- which(Data0$Year>2019)
```

Choix de variables:

Pour comprendre quelles variables sont plus significatives, et argumenter le choix des variables, nous allons effectuer une random forest, et regarder l'importance des variables.

```
### METTRE CODE ICI
```

Nous pouvons ainsi bien remarquer que les variables à plus forte importance sont: Load.1, Load.7, les variables relatives à la temperature, WeekDays, WD, BH toy, Summer_break, DLS and Christmas_break. Nous avons vérifié ce résultat à l'aide de ANOVA. !!!!!!!!!!

Choix du type de modèle

Pour comprendre et apprehender le cadre d'étude on commencera par effectuer un modèle simple. C'est-à-dire un modèle linéaire avec les covariables choisies précédamment. Nous avons considéré que la consommation

de la veille changeait en fonction du jour de la semaine. C'est pour cela que nous avons décidé de créer une fonction de la consommation de la veille en fonction de chaque catégorie de WeekDays.

```
mod1 = lm(Load~WeekDays2+Temp, data=Data0[sel_a,])
summary(mod1)
```

```
##
## Call:
## lm(formula = Load ~ WeekDays2 + Temp, data = Data0[sel_a, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16071.4  -3480.3    -21.9   3256.1  18573.0
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       76284.60     294.73  258.825  < 2e-16 ***
## WeekDays2Monday   -1153.50     324.92   -3.550 0.000391 ***
## WeekDays2Saturday -5298.50     326.07  -16.250  < 2e-16 ***
## WeekDays2Sunday   -8014.38     325.88  -24.593  < 2e-16 ***
## WeekDays2WorkDay    450.09     265.74    1.694 0.090423 .
## Temp              -1572.77      14.36 -109.510  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4708 on 2939 degrees of freedom
## Multiple R-squared:  0.8191, Adjusted R-squared:  0.8188
## F-statistic:  2661 on 5 and 2939 DF,  p-value: < 2.2e-16
```

Nous avons mis en place des transformations polynomiales sur le modèle linéaire comme une première approche de compléxification du modèle.

```
#mettre
```

Dans la suite, nous avons ainsi considéré de mettre en place des Modèles Additifs généralisés. Pour cela, nous avons d'abord distingué les variables à mettre dans la partie linéaire du modèle, puis dans la partie spline. Nous avons intégré ainsi les variables qualitatives ainsi que la consommation de la veille en fonction du jours de la semaine dans la partie linéaire. On a ajouté ainsi dans la partie spline les variables ayant une notion de temporalité comme la consommation de la semaine ou les températures. Nous avons également regroupé dans un même spline des variables ayant une relation logique, comme c'est le cas de la température et le Temps ou les Temperatures_s99 min et max. Nous avons également crée une fonction spline pour chaque jour de la semaine pour la variable toy pour ne pas négliger l'effet des jours de la semaine sur la consommation annuelle.

```
#mettre
```

Pour améliorer le rendement du modèle gam, nous avons utilisé la fonction gam.check. Celle-ci nous a permis d'ajuster la dimension des bases spline. Nous avons ainsi incrémenté les valeurs de k quand la p-value était très petite. Pour la variable toy, on a !!!!!!

```
#mettre code
```

Dans la suite on utilisera le package qgam, et en particulier la fonction qgam. Celle-ci est performe une régression quantile. En fixant le quantile à 0.4, on a changé la fonction de perte. On a ainsi introduit un biais, qui permet de s'ajuster mieux aux données du covid.

```
#mettre
```

On a ainsi décidé de garder notre équation sur la qgam et de l'implémenter ensuite sur d'autres modèles. On a ainsi d'étudier les résidus. Ceci va ainsi nous permettre de ????

```
#mettre
```

Après avoir vu en cours les forêts aléatoires, on a décidé de l'implémenter sur .....

```
#mettre
```

Nous avons enfin appliqué arima sur les résultats obtenus, pour gagner en pécision.

```
#ts_res_forecast <- ts(c(Block_residuals.ts, Data_test$Load-gam9.forecast),  frequency= 7)
#refit <- Arima(ts_res_forecast, model=fit.arima.res)
#prevARIMA.res <- tail(refit$fitted, nrow(Data_test))
#gam9.arima.forecast <- gam9.forecast + prevARIMA.res
```

Comme dernière méthode, nous avons décidé de mettre en place un aggrégation d'experts pour extraire une combinaison de prédicteurs qui puissent améliorer davantage la performance du modèle. Pour cela, nous avons regroupé les différents prédicteurs dans une variable experts, et nous avons.... Nous avons également décidé d'ajouter un modèle additionnel utilisant le filtre kalman. Nous avions déjà essayer un tel élément, mais les résultats n'étaient pas tellement satisfaisants.