Predictive Modeling

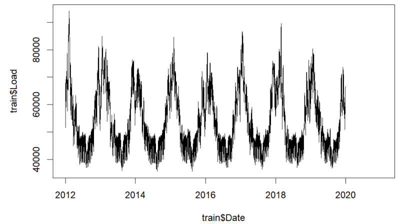
Electricity load Forecasting during the COVID Period

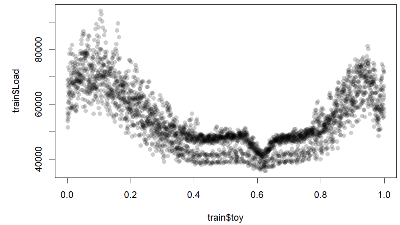
Alex Pierron
Lucas Biechy

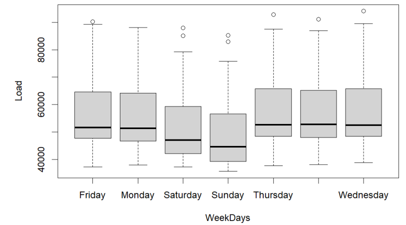
M1 Mathematics and Artificial Intelligence,
Paris Saclay University

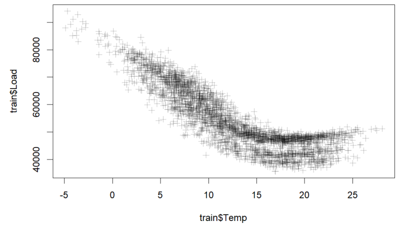
March 8, 2023

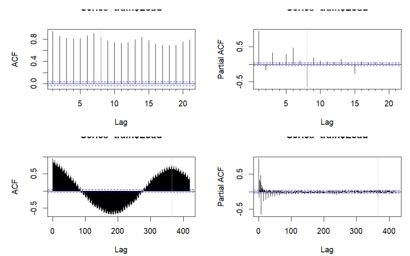
Data
Processing

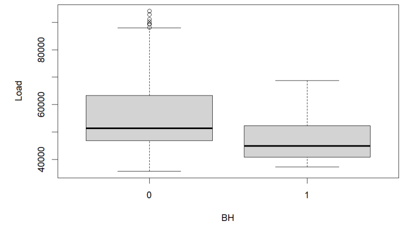










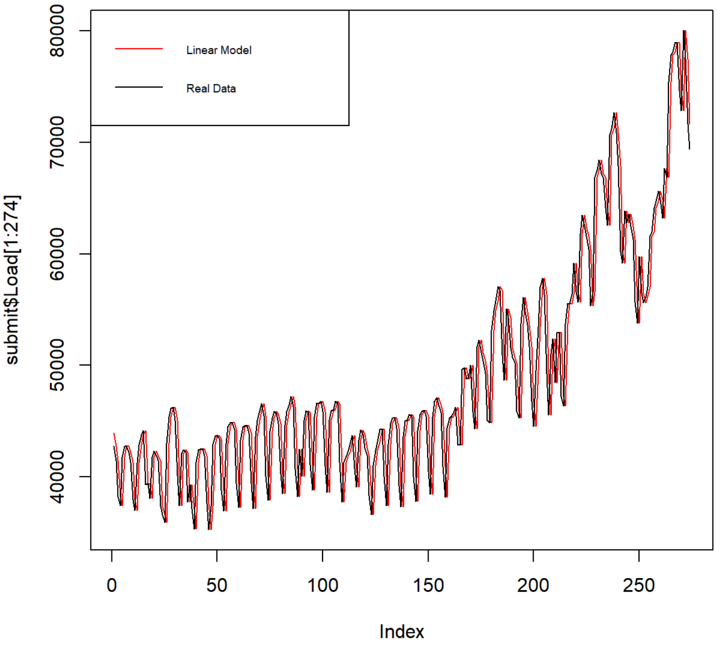


Methods and models used
Linear Model

We first tried to use a linear model as it was the only model we knew back at this time. After using the model seen in course, we quickly moved on the GAM model because of the capability it has compared to the Linear Models.

Methods and models used

Linear Model



Methods and models used

Generalized Additive Model (GAM)

Let y be the response variable we are trying to model and $x_1, x_2, ..., x_p$ the explanatory variables. The GAM model takes the following general form:

$$y = f_1(x_1) + f_2(x_2) + ... + f_p(x_p) + \epsilon$$

where ϵ is the random error and f_i are non linear functions of the explanatory variables. The f_i functions are often represented as splines, which are piecewise polynomial functions.
GAM Model is flexible and can capture non-linear relationships between the explanatory variables and the response variable.

Methods and models used

Generalized Additive Model (GAM)

For tuning our equation, we used several approaches:

```
eq<-as.formula(paste("Load~ WeekDays2 + s(toy,Temp,k=12) + I(Temp^2)+
                      s(Load.1,Load.7)+Summer_break +
                      s(Temp_s95,Temp_s99) + BH +
                      Temp_s95_min + Temp_s99_min +
                      Temp_s95_max + Temp_s99_max",sep=""))
```

```
equation = Load ~ WeekDays + s(toy,Temp,k=12) +
  te(Load.1,Load.7,k=7)+ s(Load.1,by =Summer_break) +
  s(Load.1,by =Christmas_break)+ s(Load.7, by = Summer_break,bs='cr') +
  s(Temp_s95,Temp_s99) + BH + s(Load.1,by = WeekDays2) +
  Temp_s95_min  + s(Load.7, by = WeekDays2) +
  Temp_s95_max  + s(Load.1,by = Month) + s(Load.7,by = Month)
```

```
equation = Load ~ WeekDays + s(toy,Temp,k=12) +
  te(Load.1,Load.7,k=7)+ s(Load.1,by =Summer_break, bs='cr') +
  s(Load.1,by =Christmas_break,bs='cr')+ s(Load.7, by = Summer_break,bs='cr') +
  s(Temp_s95,Temp_s99) + BH + s(Load.1,by = WeekDays2,bs='cr') +
  Temp_s95_min  + s(Load.7, by = WeekDays2,bs='cr') + s(Load.1,by = Month,bs='cr') + s(Load.7,by = Month,bs='cr')
```

Figure. Three milestones equations found for each approach: Scores reached on Kaggle test set were of 965 , 904 and 870 (from top to bottom)

,

Methods and models used

Generalized Additive Model (GAM)

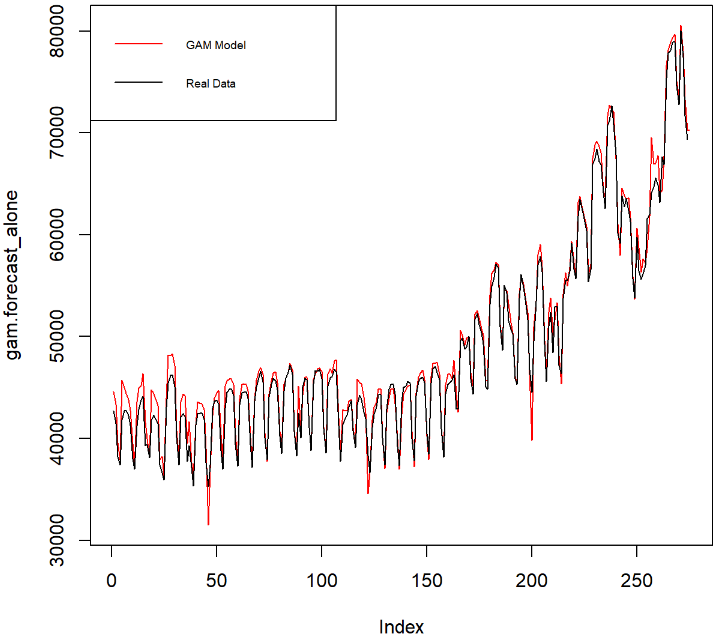


Figure. Result for the GAM model on the test set

ARIMA

Theory

AR : $y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$
wherey is the time series, c is a constant, p is the order of theautoregressive regression, ϕ_i are the AR coefficients, and ϵ_t is the error term.
MA : $y_t = c + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t$
where y is the time series, c is a constant, q is the order of the moving average, θ_i are the MA coefficients, and ϵ_t is the error term.
The I : $\Delta y_t = y_t - y_{t-1}$
where Δy_t is the difference between the current value of the time series and the previous value.

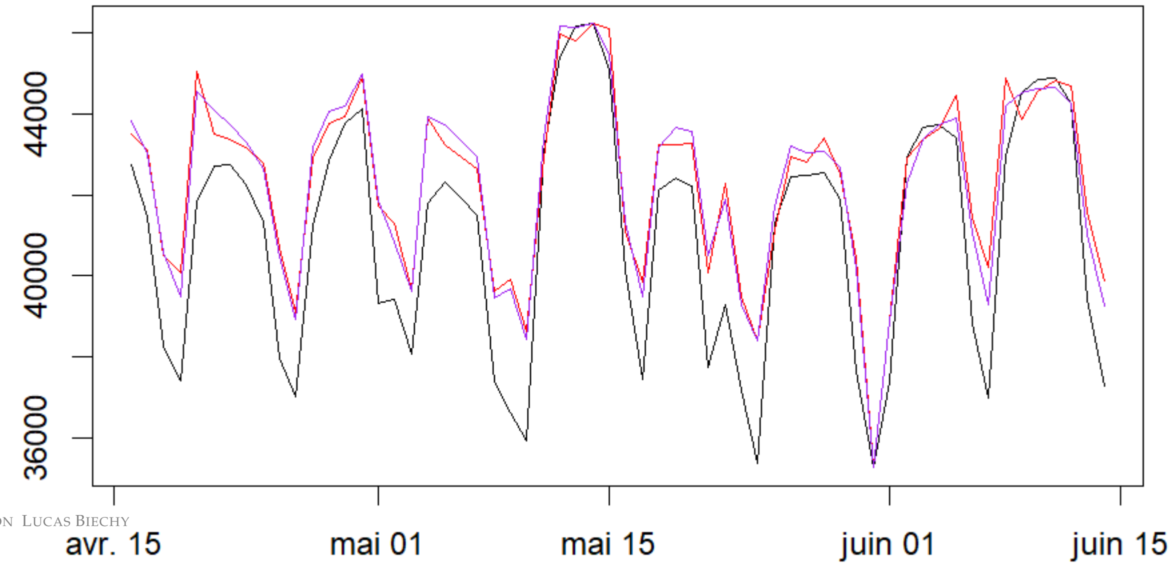
ARIMA(p,d,q) model, where p is the order of autoregression, d is the order of differentiation, and q is the order of the moving average.

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d y_t = c + (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon_t$$

where L is the lag operator, and y_t is the time series.

ARIMA

PLOT



Methods and models used
Online GAM Arima

We tried an online learning with a GAM model and also an ARIMA correction for residuals. Several Remarks can be done:

- For each new day to forecast: we used all the prior days to create a GAM model with the equation found before. Then we adjust an ARIMA model for the residuals upon this set. We can make a forecast by adding the GAM prediction and the ARIMA predicted residual. Then we pass to the next day and we repeat the process.
- Residuals for fitting the Arima model are not calculated with Cross-Validation because it is too long (around 10 minutes for one day and only for the Cross-Validation). Therefore our model is perfectible because of this hardware limit.
- To calculate the residuals for the day we are predicting: we use the actual load consumption of the previous day. We assume some sort of continuicity between days with this hypothesis.

Online GAM ARIMA

Results on the test set

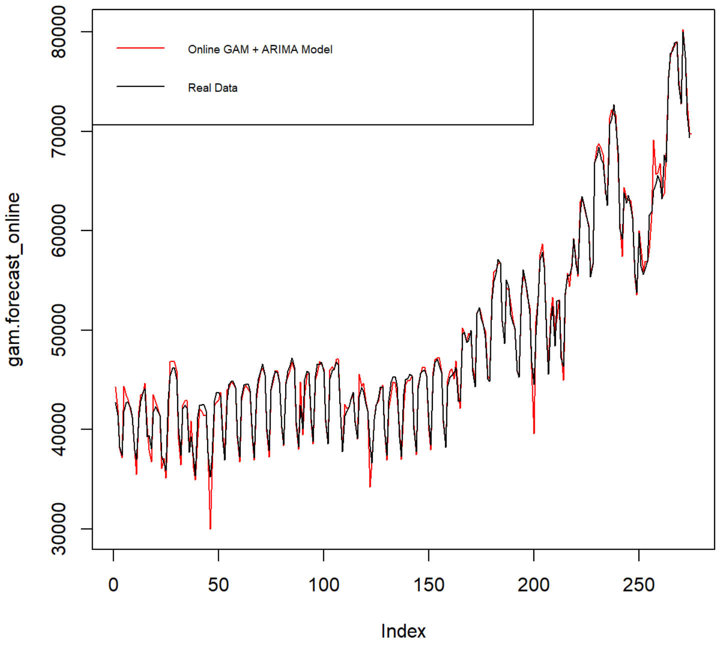


Figure. Result on the test set for the Online GAM + ARIMA model

Methods and models used

Online GAM Arima

- We used all the previous model introduced (except the linear model) and their forecast to make an aggregation of experts.
- For a day n, We are looking for producing the best aggregation possible of our experts based on the labels and forecast available for the n-1 days before. The quality of the result depends on the chosen loss function.
- For the final day, we use the prediction from our best individual model (which is the online GAM Arima model).

Ensemble Methods

Random Forest and Boosting

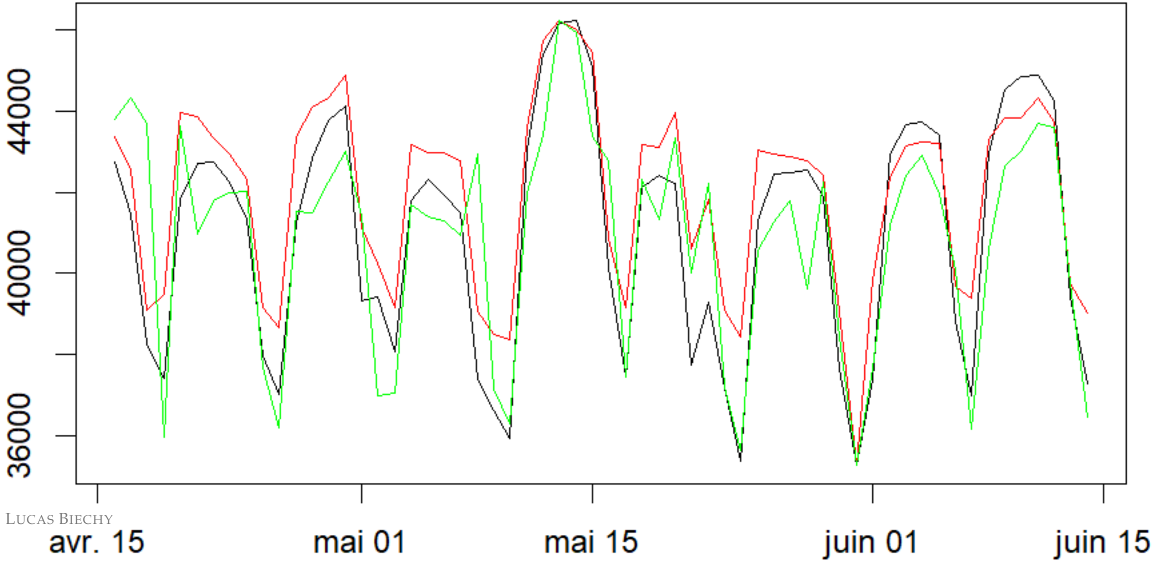
B independent decision trees. Each tree $b = 1, ..., B$ is obtained by drawing n observations with replacement from the training dataset.
Then, for each node in a tree, a random subset of m predictors is selected for splitting, where m is a number set by the user.
The final prediction is obtained by averaging the predictions of all the constructed trees:

$$\hat{f}RF(x) = \frac{1}{B} \sum b = 1^B \hat{f}_b(x)$$

where $\hat{f}_b(x)$ is the prediction of tree b for observation x.

In contrast, Boosting builds a set of successive models, with each model built on the residual errors of the previous model, giving greater weight to examples that were misclassified in the previous steps.

Ensemble Methods
Plot



Expert Aggregation
Different methods

```
## Aggregation rule: MLpol
## Loss function:  squareloss
## Gradient trick:  TRUE
## Coefficients:
##    gam gamarima gamarimaonline     rf boosting
##  0.201    0.235          0.212 0.0995    0.252
##
## Number of experts:  5
## Number of observations:  274
## Dimension of the data:  1
##
##         rmse   mape
## MLpol    834 0.0122
## Uniform  841 0.0131
```

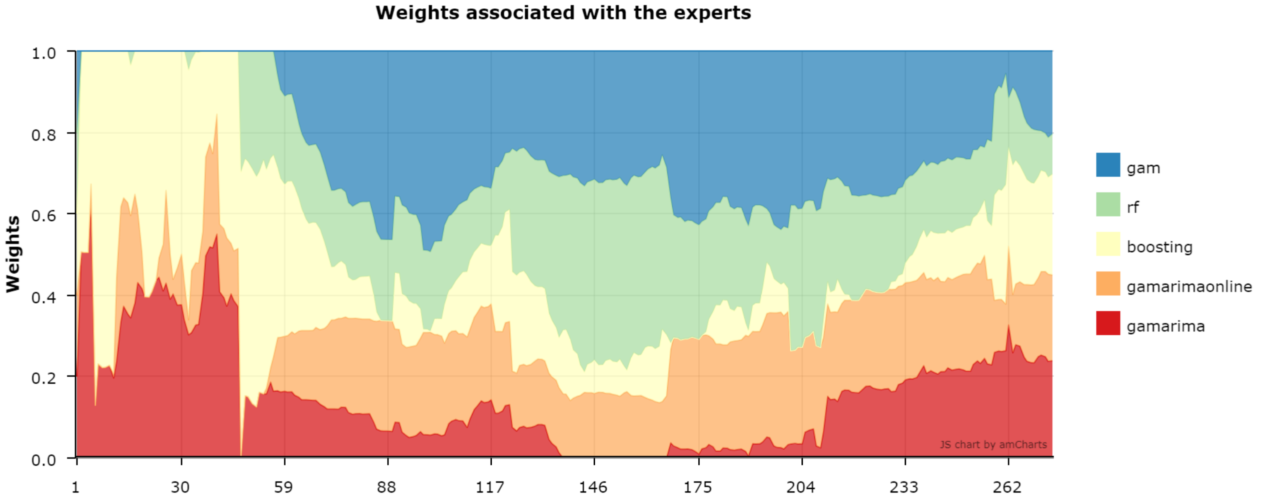
```
## Aggregation rule: BOA
## Loss function:  squareloss
## Gradient trick:  TRUE
## Coefficients:
##    gam gamarima gamarimaonline    rf boosting
##  0.179    0.291          0.257 0.146    0.127
##
## Number of experts:  5
## Number of observations:  274
## Dimension of the data:  1
##
##         rmse   mape
## BOA      849 0.0126
## Uniform  841 0.0131
```

Figure. Two aggregation rules for expert aggregation

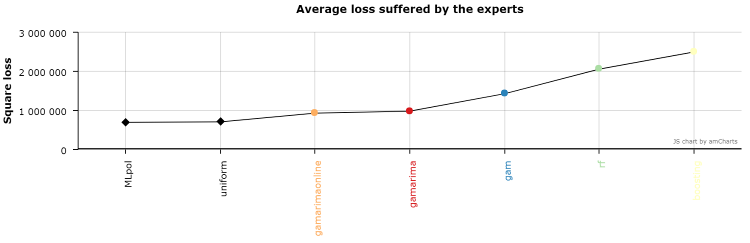
,

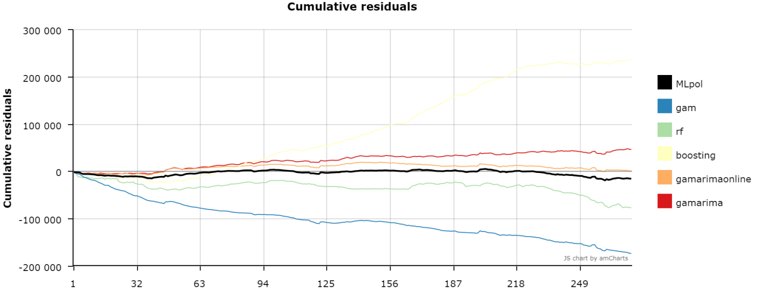
Expert Aggregation
Graphs

We use the package opera and the mixture fonction to have the following graphs:



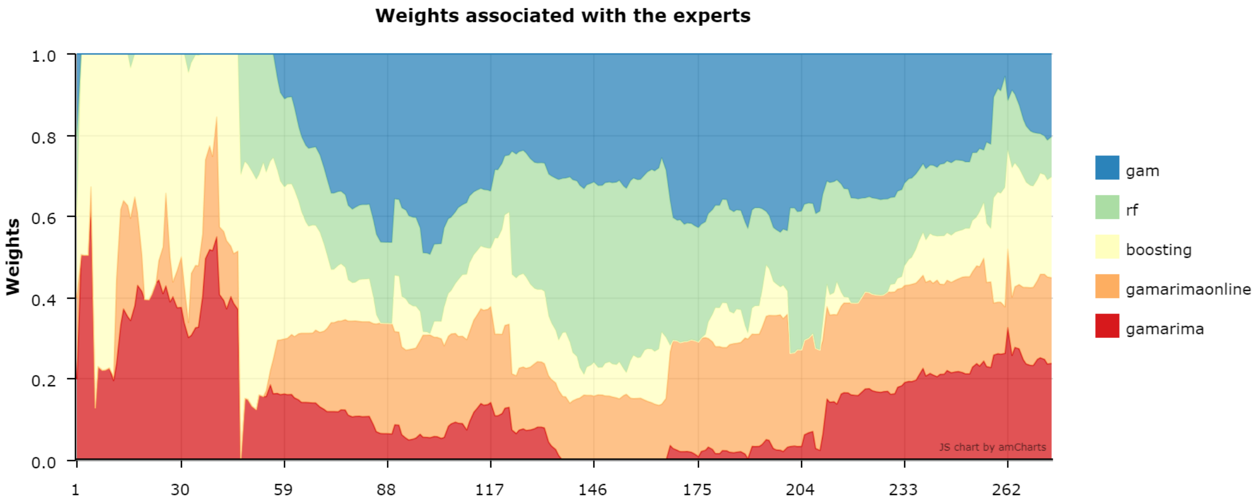
Expert Aggregation
Graphs





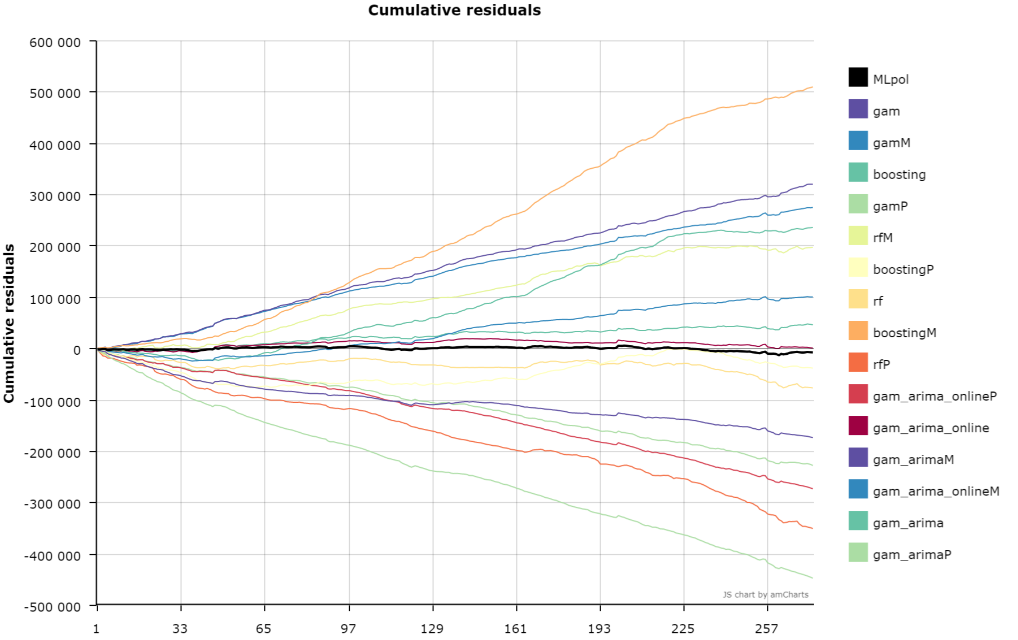
Expert Aggregation
Graphs

We use the package opera and the mixture fonction to have the following graphs:



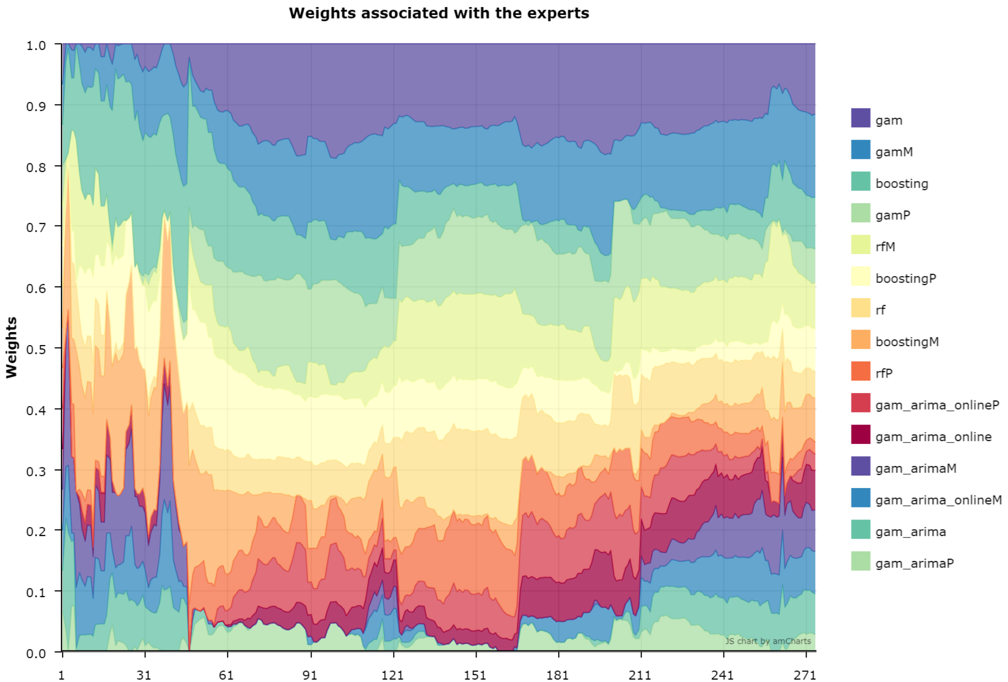
Biaised Expert Aggregation

Cumulative residuals graph



Biaised Expert Aggregation

Experts Weights



Biaised Expert Aggregation

Final Result

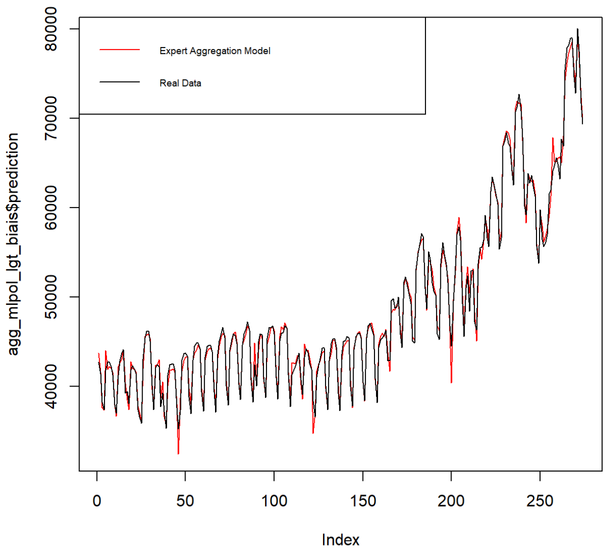


Figure. Final model with biaised expert aggregation on the test set

END

Possibility of improvement :

- Cross Validation during the online GAM- Arima model
- Fine-tune the equation used for all the GAM models
- Include additionnal data

Thank you

Questions ?