Methods for Big Data Analytics: Axa Challenge Project

Team San Parisc
Geoffrey Chinot
Benoit Choffin
Peter Martigny

January 18, 2017

1. Introduction
   - Introduction to the Axa Data Challenge
   - Data Exploration
2. Data Cleaning and Feature Engineering
3. Algorithms
   - Time series modeling
   - Self training algorithm
   - Final model
4. Conclusion

Axa Challenge

- Goal : Predict the number of incoming calls in Axa's call centers
- Constraint : the potential losses are not symmetric :

Loss function

$$LinEx(y, \hat{y}) = exp(\alpha(y - \hat{y})) - \alpha(y - \hat{y}) - 1$$

- $\alpha = 0.1$
- Under-estimations are severely penalized, compared to over-estimations

The dataset

- The training file
  - More than 10M rows
  - Each row : number of incoming calls in a specific center at a specific time slot (one time slot is 30 minutes)
  - 28 different call centers
  - Some full weeks are missing...
- The submission file
  - Correspond to the missing weeks of the training set
  - Only 2 features : DATE and ASS_ASSIGNMENT
  - Need for building features !

Data Exploration

Data Exploration

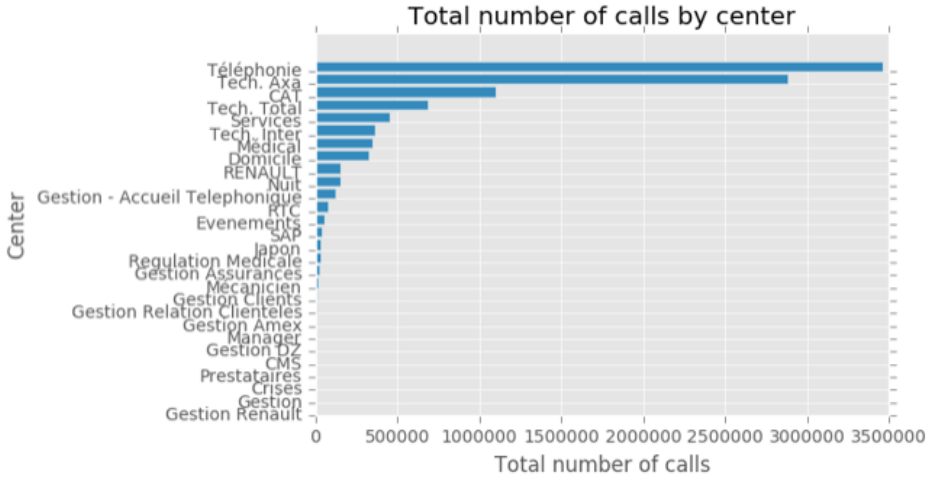


FIGURE – Total number of calls by center

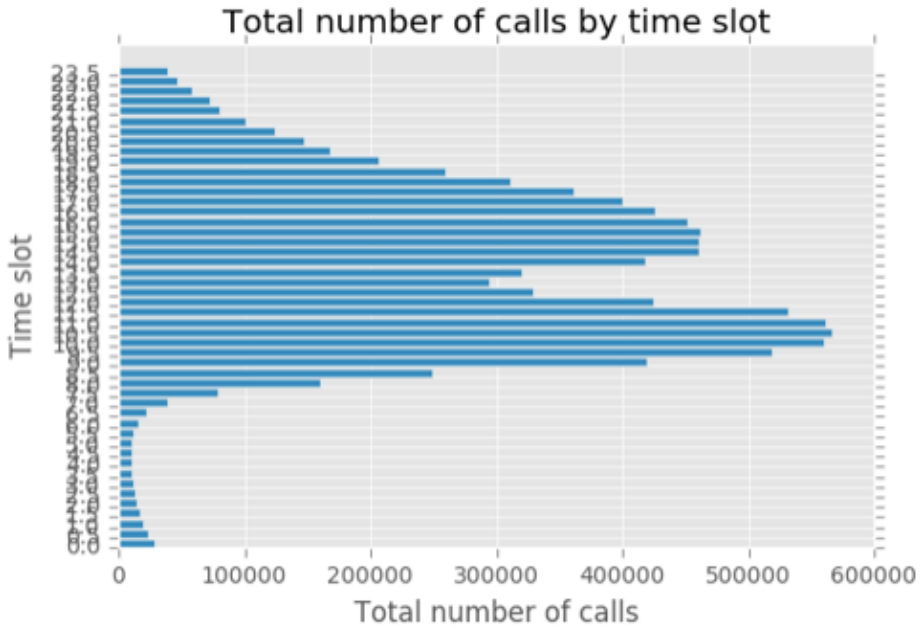


FIGURE – Total number of calls by time slot

Data Exploration

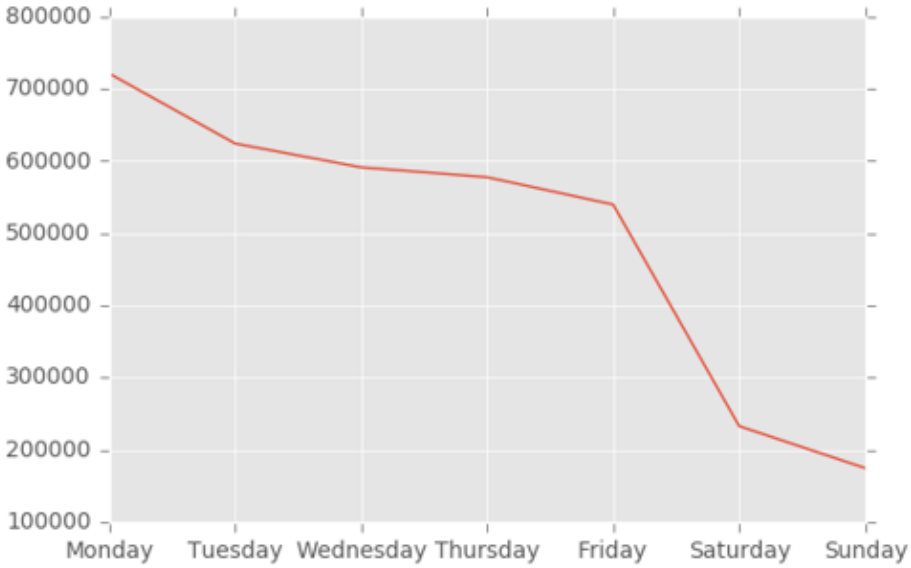


FIGURE – Total number of calls by day

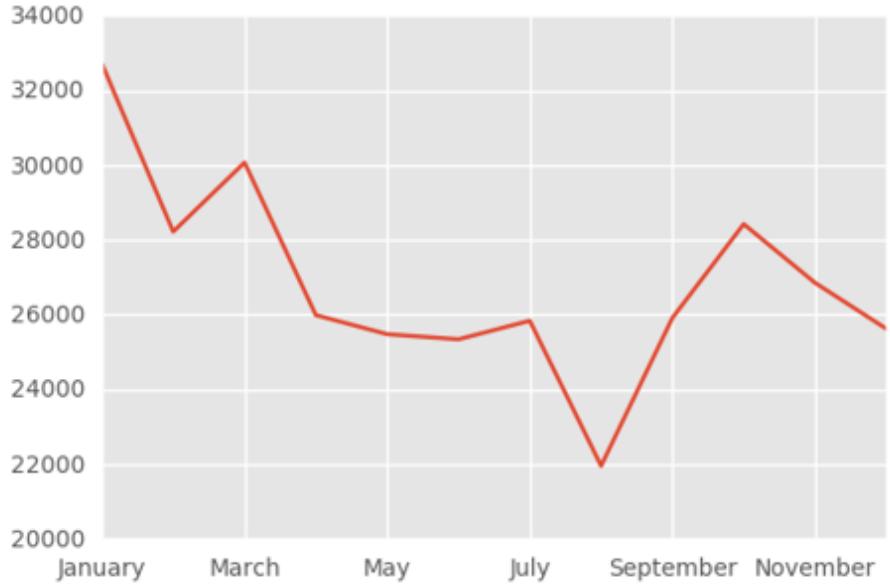
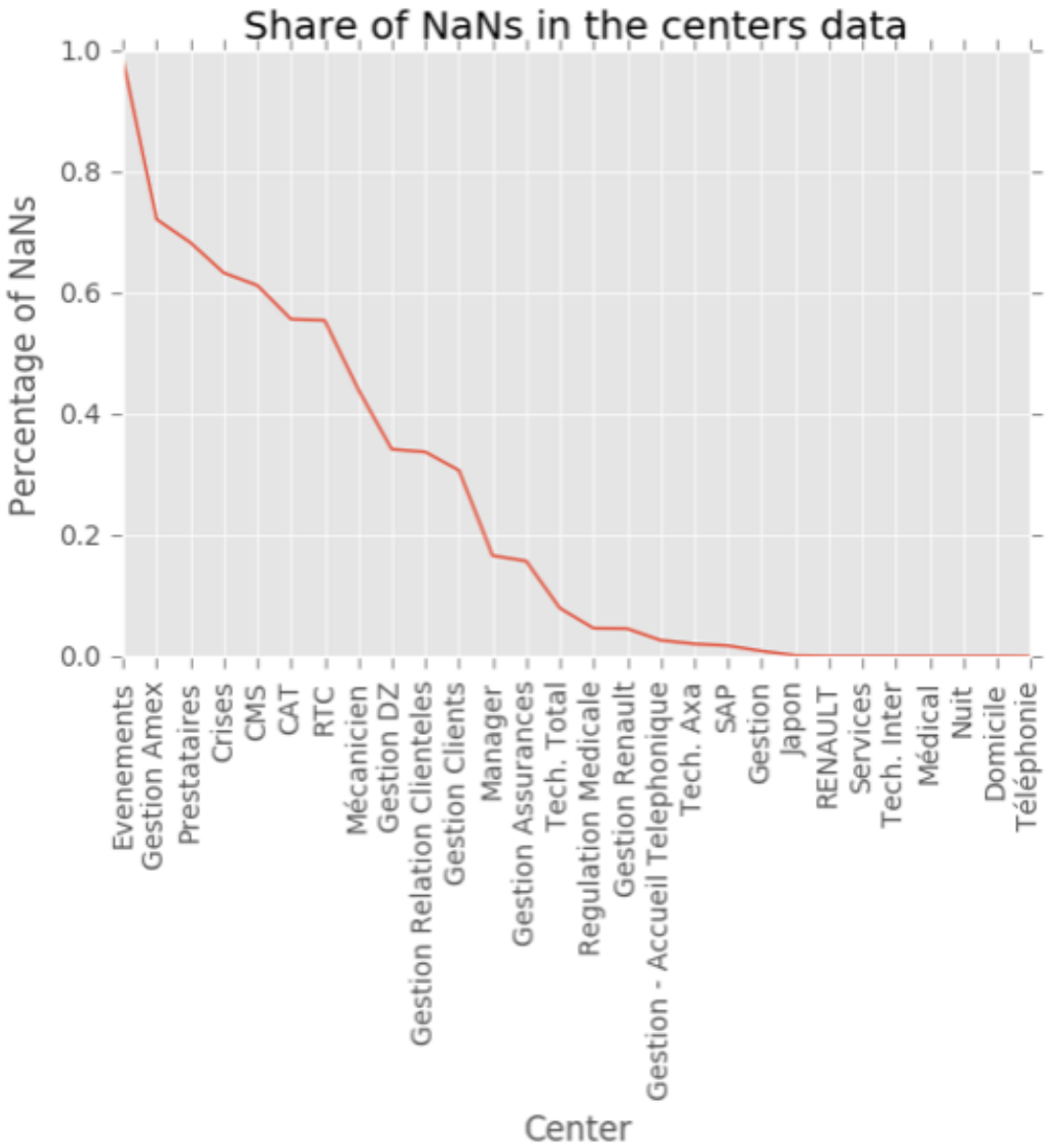


FIGURE – Total number of calls by month

Missing Values



Missing Values

FIGURE – Total missing values by hour of the day

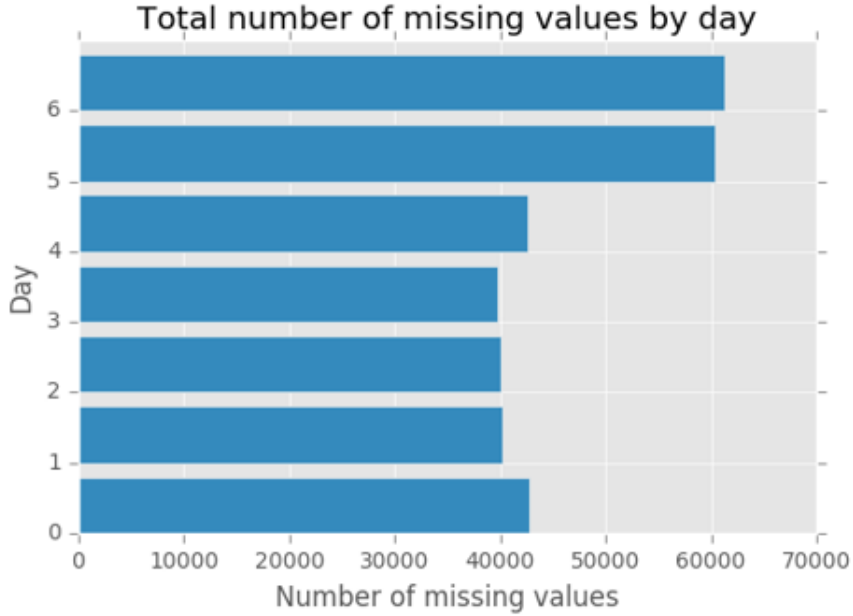


FIGURE – Total missing values by day of the week

Cleaning the database

- Dealing with duplicates
  - Original dataset contained multiple rows with the same multi-index ['ASS_ASSIGNMENT','DATE']
  - Reason ? They had different administrative codes $\rightarrow$ group by code
- Dealing with missing values
  - **Major issue** : which process lead to these missing values ?
  - Center-specific and iterative imputation
  - Days off $\rightarrow$ 7-days lag $\rightarrow$ median over last week $\rightarrow$ interpolation

Feature engineering

- Dummy variables for each day (except 1)
- Hour of the day (numerical)
- Week-end dummy variable
- Night dummy variable (Night = between 19 :30 PM and 8 :30)
- Holiday dummy variable
- Dummy variables for each month (except 1)
- Number of calls exactly one week before
- Average number of incoming calls of the week $W - 2$

Time series modeling

- Decomposition of each time series into three components : a trend, a season and a noise
- Fit an ARMA on the noise : $ARMA(p, q)$:

  $X_t = \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_i \epsilon_{t-j}$, with $(\epsilon_t)_t$ Gaussian terms
- Different problems :
  - How to choose q and p ? Cross validation is too expensive for each center
  - Is the residual term stationary ? Not really for some centers ("Téléphonie")
- It appears that this method gave poor results

Self training algorithm

- Goal : Capture the behaviour of "Téléphonie"'s center in 2013
- We train only on the last few months our model
- We cannot predict one year only with few months in our training dataset $\rightarrow$ Self-training
- At each iteration we add our last predictions and some new observations in the training set.
- However we have not improved our results with that method $\rightarrow$ Propagation of the error ?

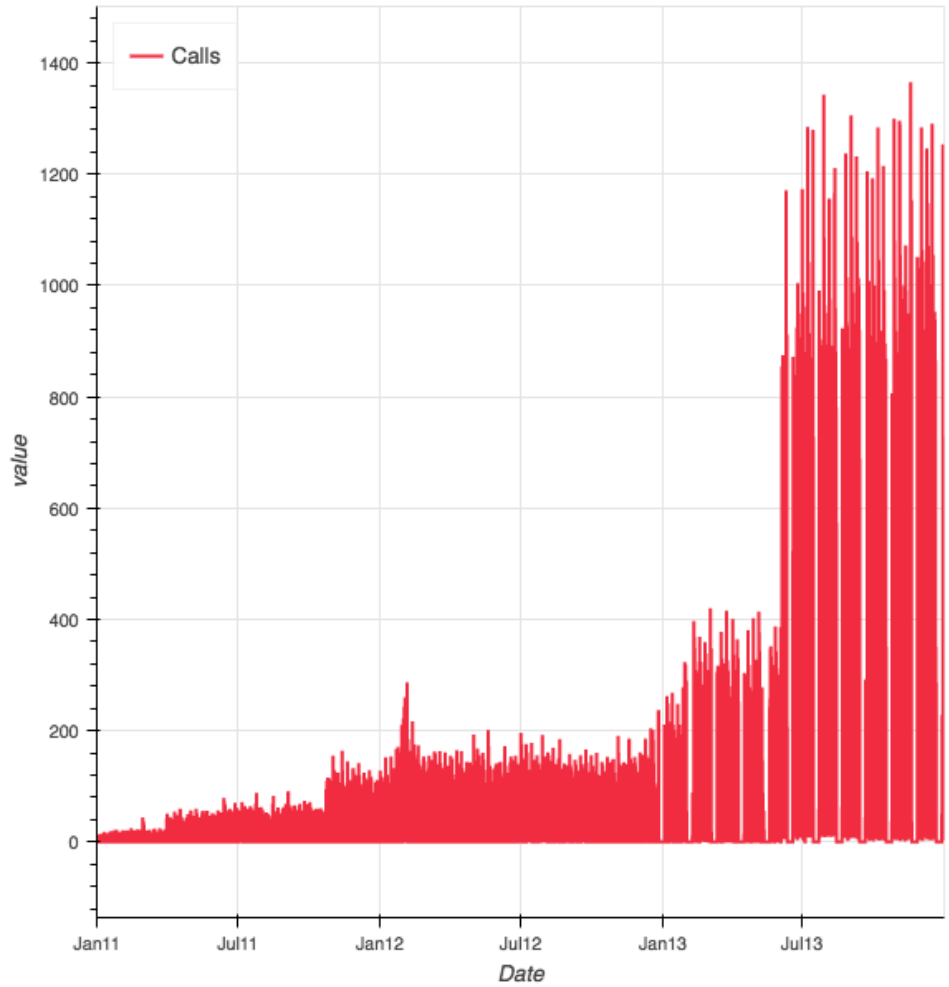


FIGURE – Behaviour of "Téléphonie"

Final model

- Lasso regression :
  - Feature selection/sparsification (*l*1-norm)
  - Overfitting risk reduction
- Hyperparameter selection : K-Fold cross-validation
  - We fine-tuned the penalization strength for the centers whose baseline LinEx score was above a certain threshold
  - **Result** : we almost systematically decreased the penalization strength
- **Final result** : 0.82 on the public leaderboard

Conclusion

- Both a technical and a business challenge
- First approach of a real-life business case
- Further possible improvements :
  - Implementing our algorithms with a custom loss function to be minimized
  - Change of paradigm : State-Space Models as a natural way to deal with sequential data