Methods for Big Data Analytics

Benoît Choffin (benoit.choffin@ensae.fr) Peter Martigny (peter.martigny@ensae.fr)
Geoffrey Chinot (geoffrey.chinot@ensae.fr)

January 2017

Axa Data Challenge - Team San Parisc

Supervised by Michalis Vazirgiannis

Contents

List of Tables

List of Figures

1 Introduction

1.1 The Axa Data Challenge

In this project, we are focusing on 28 different call centers from AXA. We are given the number of incoming calls from these centers every 30 minutes during 3 years, and we wish to build a model able to predict, one week in advance, the number of incoming calls between call centers.

When the company manages its employees and forecasts the repartition of employees between its different centers, it has to be very cautious about the potential losses from a bad forecast. On the one side, if the company predicts too much incoming calls, it will use too many employees, compared to what is needed, and hence it will lose money on labor costs. On the other side, if the company predicts too little number of incoming calls, it will use less employees compared to what is needed, and hence some clients will have a bad experience and eventually the image of the group will be affected.

Hence, we see that there are two kinds of possible losses from mispredicting the number of incoming calls in call centers. The loss is highly non-symmetric, and the strategy of the group will be that providing a bad experience to customers is worse than losing money on labor costs. Hence, we will use as a loss function the following:

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

with y the true value and $\hat{y}$ the predicted value. Specifically, an value of 0.1 for the α hyperparameter is used in the computation of the final loss value.

1.2 Data Exploration

The train set consists of more than 10 millions rows, each providing the number of calls for a particular call center at a particular time slot (every 30 minutes). It covers the years 2011, 2012 and 2013. Some weeks are lacking in the train set, these are in fact the weeks we are trying to predict in the test set *submission.txt*. However, there is another phenomenon at stake here : some time periods are missing in the train set, and these are not the periods we need to predict! In fact, this must come from failures in the data acquisition process of Axa. We will see later how we imputed these missing data.

We reproduce in Appendix the plot of the incoming calls for each center. A first glance at these plots shows that some centers present a visible regular pattern, whereas some others seem to be uneasy to predict. We also represent in Appendix the total number of incoming calls for each center, to know which centers are contributing the most to our predictions. It is also interesting to know when the calls are mostly being made, therefore we reproduced in Appendix the total number of calls by time slots, by day and by month.

We realized that some time slots exist for certain centers, and not for others. Hence, we quantify the percentage of missing values for each call center.

We see on this figure that, while some call centers have a complete record of data, some have a huge amount of missing data, especially those with more than 50 % of missing data.

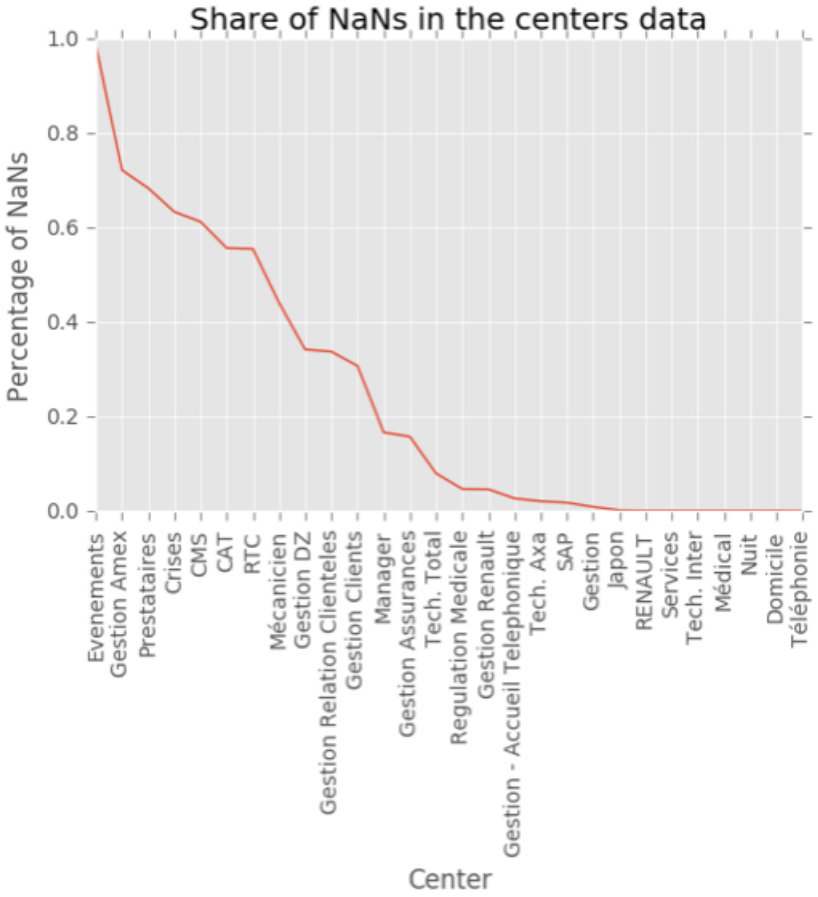


Figure 1: Share of missing values for each call center

Finally, we reproduce in Appendix the total number of missing values by time slots and by day (and see that a large part of them are due to week-ends).

2 Pipeline

2.1 Duplicates and management of missing data

Given the fact that the submission set contains only the features $ASS\_ASSIGNMENT$ and $DATE$, we only keep from the initial train set the variables $ASS\_ASSIGNMENT$, $DATE$ and $CSPL\_RECEIVED\_CALLS$. We remark that some couples ($ASS\_ASSIGNMENT$, $DATE$) appear several time (corresponding to different administrative codes), we decide then to group by the key ($ASS\_ASSIGNMENT$, $DATE$) and aggregate by using a sum.

There are two ways to deal with the problem of missing values. On the one side, we could infer that a missing data would mean that the timeslot is not important, and therefore fill them with zeros. However, because of the assymetry of the loss function, missing values wrongly filled with zeros will lead to surestimations, and then the loss will be high. Hence, we choose to fill the missing values, by using various methods, presented in the following paragraph. Each of the following methods is center-specific.

We proceeded in an iterative way so as to impute missing data in this challenge : we indeed selected a few ways to impute missing data that seemed relevant to us, ranked them by order of importance, and applied them to our problem. At first, we built a list of day-offs for France, and if a date was in this list, with no value for $CSPL\_RECEIVED\_CALLS$, we would impute it by 0. Then, for regular centers, we imputed the rest by the number of calls for the same day (e.g. Monday) and the same hour in the preceding week. There were still issues because for some centers, the missing values were periodic and occurred each week at the same hour. Therefore, we completed this approach by taking the median over the last 7 days. Finally, we interpolated and put the rest to 0.

2.2 Feature Engineering

In order to create the features needed to predict the output variable, we first selected a few features based on prior knowledge we had on the subject. It is also important to notice that for the dummification of categorical variables, we only used $d-1$ variables (with d the number of categories), in order to prevent the features matrix to be non-invertible:

- a dummy variable for each day label (Monday, Thursday,...) except one
- $time\_since\_start\_of\_day$ which indicates the hour of the day
- $week-end$ a dummy variable telling if this is the week-end or not
- $night$ a dummy variable telling if this is the night or not (we considered that it was the night if the call occured between 19:30 PM and 8:30)
- $holiday$ a dummy variable telling if the date is in the holidays or not
- a dummy variable for each month (except one)
- $last\_nb\_of\_calls$, which indicates the number of calls exactly one week before
- $past\_7\_days\_avg$, which gives the average number of incoming calls of the week $W-2$

We can also take as features all values from the week $W-2$. Taking more features would lead to computational difficulties, and the correlation would be less decisive to make our predictions. In the last model that we used to make our predictions, we did not take into account all values from week $W-2$, but only the value at the same time slot one week before, and an average value of the week $W-2$. The reason is that, while we carry out a dimension reduction algorithm such as PCA, the features that are created (containing the most significant parts of te variance) are made of highly correlated features, the correlation resulting from the temporal correlations. We observed that, in the end, in addition to produce a higher computational cost, the result were worst than the ones obtained with less initial features. Hence, we decided to only t ake into account the features listed before.

2.3 Split, Cross-Validation

Time series differ from other usual tasks in the fact that the temporal dimension has a physical significance: the data are usually not iid. Hence, it is not possible to use future data to predict past data. The usual K-Fold Cross Validation is therefore obsolete for time series forecasts.

Instead, we use a cross validation method adapted to time series' temporal dependency, called *TimeSeriesSplit* in *sklearn*. This method is very similar to K-fold CV, but at the $k-1$-th split, it takes the first $k-1$ folds as train set, and the k-th fold as test set.

2.4 Algorithms

Once the training set is built and the splits being done with time series specific cross-validation, we use machine lerning regressors to predict the number of incoming calls.

We tried random forests and boosting methods, as well as several dimension reduction transformation (PCA, k-PCA), however it appeared that a simple Lasso was both behaving better on test set, and moreover it had a way cheaper computational time.

Time series modeling

As we will see, the results for the center "Téléphonie" are bad : that's why we decided to develop other models for this center. We tried to fit some time series models. To begin we have to transform our series in order to have a stationary process. To do that, we decompose our series in 3 signals: a trend, a seasonality and what it remains (cf. appendix). After doing that we fit an ARMA model on the rest. However the results were not good because it was not a stationary process. We have tried also other models but the results were also bad.

Self-learning algorithm

Another idea was to try a self-learning algorithm. To do that, in a first time, we train the model on all data before the date of the first prediction. Then we predict the full week we have to. In a second time we consider a new train dataset composed of the last one plus the predictions and the data between the two weeks we have to predict and so on. The idea was to capture the singular behaviour of certain centers such as "Téléphonie".

Lasso Regression

Our best model combines the advantage of feature selection and sparsification with the advantage of overfitting risk reduction. Indeed, we used a Lasso regression, with various penalization factors. We chose this hyperparameter by cross-validating with time-series specific K-Fold. For the centers whose confidence interval for the LinEx score was below 1, we considered that the initial value of α was good. But for some centers, the LinEx score was extremely high (e.g. for "Téléphonie") and therefore, they were our major focus. For most cases, decreasing the penalization strength was a good solution.

Even with that, our final predictions were systematically below the true values. This leads to systematically extreme values for the LinEx scores, due to the construction of the loss. To address this issue, we multiplied our scores by a factor in order to rescale our predictions.

3 Results

We found that, while the regressors were providing good results for most call centers, they had huge loss for the centers "Téléphonie", "CAT" and "Tech. Axa". These results are explained by

the fact that these call centers have large amount of incoming calls. Hence, even if the relative error is small in average, the absolute error may be large, and if this large absolute error produces an under-estimation of the real number of calls, this will result in huge losses, as it is in our case.

In order to take this issue into account, we carry out an additionnal transformation to the results produced by the regressor, in order to lower the loss for these call centers.

4 Conclusion and future work

The task of predicting the number of incoming calls from Axa call centers is both a business and a technical challenge. Indeed, if the regression task is not so much complicated when dealing with a symmetric loss measure like $RMSE$, the dyssymetry in our task included by the business problem of prioritizing the customer experience over the labor costs makes the regression harder.

We have chosen to use algorithms given from usual libraries, which do not enable to choose a custom loss function. Hence, we had to be able to correct afterwards the prediction made by the $RMSE$-optimization based algorithms. The use of *xgboost* enables to customize a loss function, but the results given on cross validation were poor. As a future work, it would be necessary to implement the algorithms we used (penalized linear regression, SVR, Gradient Boosting...) ourselves with a custom loss measure, using its gradients to derive the gradient-based optimization algorithms.

Bibliography

[1] Christoph Bergmeir, Rob J. Hyndman, Bonsoo Koo, *A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction* (2015)

5 Appendix

5.1 Data Exploration

Figure 2: Incoming calls time series for each call center

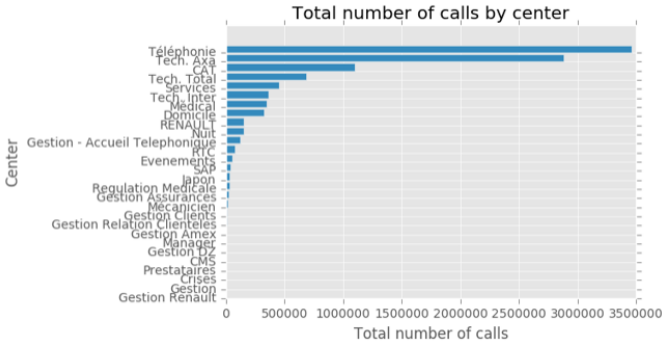


Figure 3: Total number of calls by center

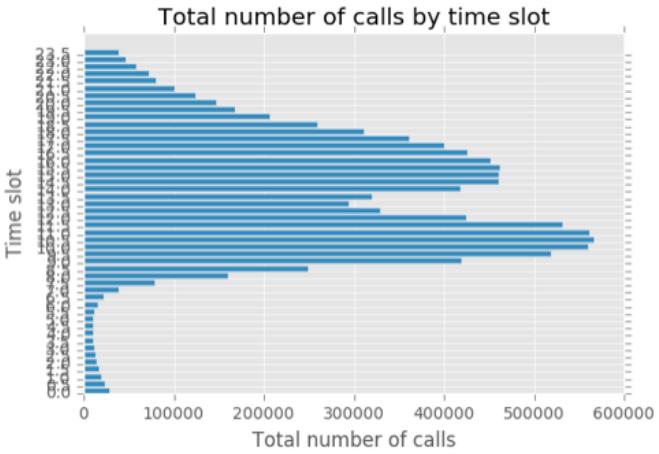


Figure 4: Total number of calls by time slot

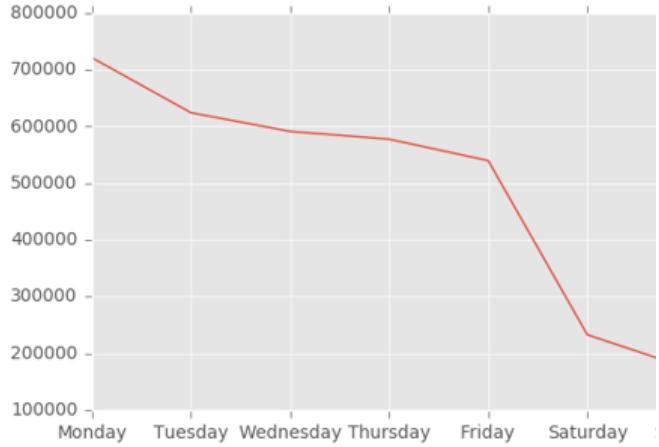


Figure 5: Total number of calls by day of the week

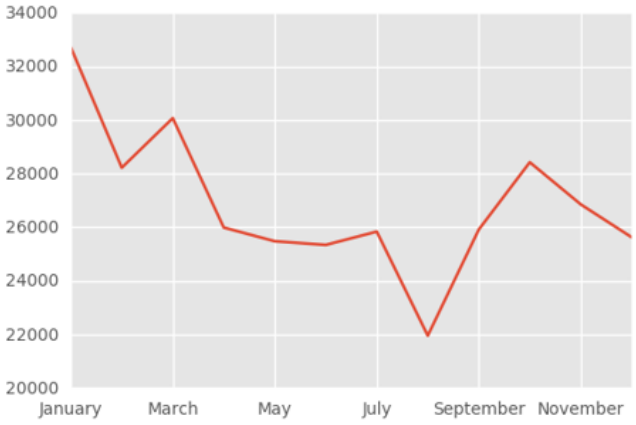


Figure 6: Total number of calls by month of the year

5.2 Analysis of missing values

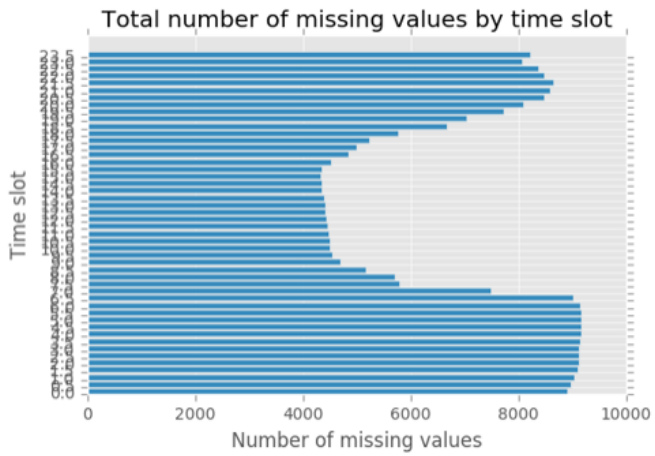


Figure 7: Total missing values by hour of the day

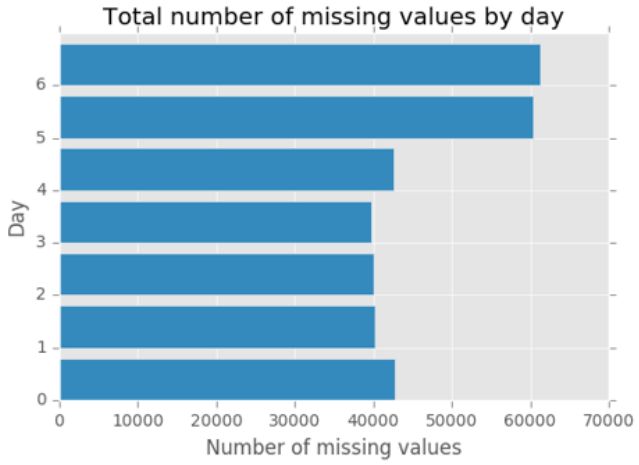
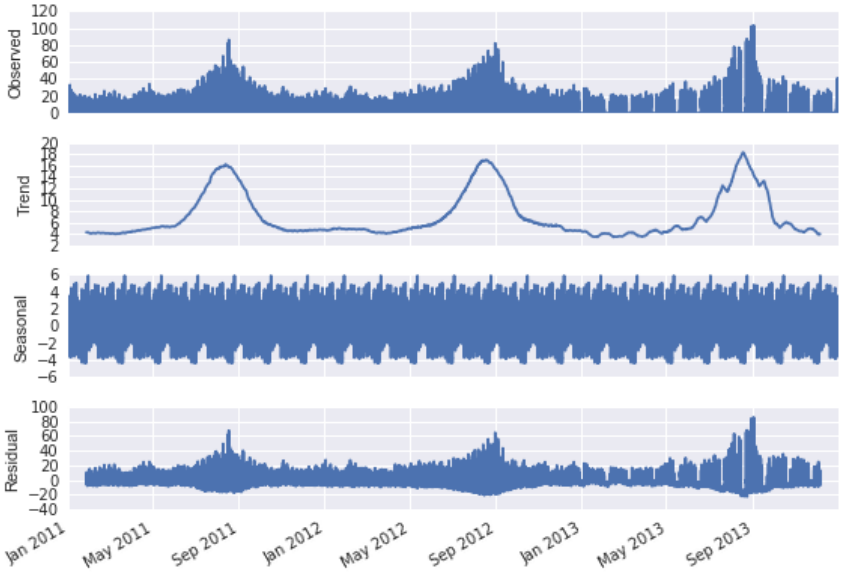


Figure 8: Total missing values by day of the week

5.3 Decomposition of Time Series



[H]