AXA Data Challenge

Aexandre Carton, Clément Fischer, Alexandre Guinaudeau

March 4, 2016

1 Introduction

In this report, we describe how we constructed a model to predict the number of incoming call to the AXA french center. The dataset on which the prediction is made consists in training data from the years 2011 and 2012, giving the number of calls depending on the date and a number of parameters such as the area of expertise of the call and the state of the call center. It comes up with another complementary set of data concerning weather information in France, which is supposed to have an influence over the number of accidents and thus incoming calls to AXA. We divided our work in 4 parts. First, the dataset has to be preprocessed to be used efficiently. Then we use this data to create features for our model. After this feature engineering step we train a model and use it to predict the number of calls asked in the submission file. We also evaluate our model using cross-validation techniques.

2 Preprocessing

The data comes in two types of csv files : meteo_2011.csv, meteo_2012.csv and train_2011_2012.csv. The main training dataset contains multiple informations on the incoming calls to AXA's centers in France. The number of incoming calls we have to predict, CSPL_RECEIVED_CALLS is one of the 86 columns of the file. For each value of DATE and ASS_ASSIGNMENT (the field of competence to which the call is assigned), the model has to predict the number of incoming calls for the three next days.

We used pandas library to read the files as databases objects. The main advantage of pandas' read_csv function is that it allows us to read only the columns we are interesssted in, thus saving much computation time.

For the training dataset, we keep the columns 'DATE', 'ASS_ASSIGNMENT', 'CSPLRECEIVED_CALLS' and 'DAY_OFF'. We use this last column to eliminate data corresponding to non worked days in AXA. We then group the data by summing the number of calls having the same date and assignment values.

The meteo dataset consists in rows giving information at a given date for a some location - city and departement ($\simeq$region) number - in France. From this,

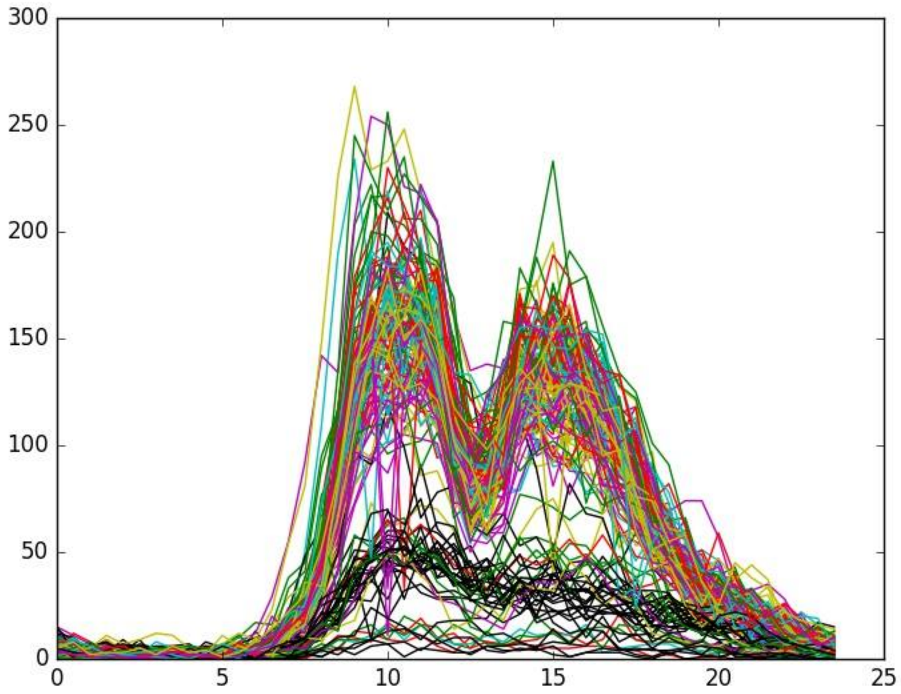

Figure 1: Number of calls depending on the hour of the day.

we extracted the number of french departements undergoing negative temperaures and number of them with rain, the mean of tje lowest temperature over all departements.

3 Feature engineering

The data is then gathered in a matrix. The preprocessed csv files are loaded into a pandas dataframe. We optimized the creation of features by creating a class FeatureFactory with a general wrapper function that works on a column of the feature matrix. To avoid bugs linked with missing values in the preprocessed dataset, we added zeros to the original training set, assuming there is no call when no data is registered. The meteo missing values are also replaced with the mean of the data.

To first have an idea of the features we selected, we plotted the variation of the number of calls depending on what we expected to have an influence. The features that we create are mostly dealing with the date, ie. hour, day in the week, weekend or not, month, year. Indeed, time appears to have essential of the influence over the number of calls. The assignment is also a feature. Finally we added a few meteo features, those we choose to extract at the preprocessng step.

4 Training models

We used the sklearn library to train a model over the features we extracted. Our approach was the following : we try one of the regression models among those provided in the library. From this we compute a cross validation score one one full day. We repeat this until we find out whiwh is the best model to be used.

The best results we got with this aproach are a score of 113 with the submission file. We then realised that this score drops to 75 by simply uploading a mean of all number of call values on the same day of the week/hour/assignment and not using the regression algorithms. Thus, our final algorithm connsists in :

- Computing the means
- Substract them from the data and train a model using this dataset without means
- Finally, add the predictions from the model to the means computed first