Axa Challenge Report

Ahmed MAZARI – Hafed RHOUMA
Université Paris Sud
Big Data

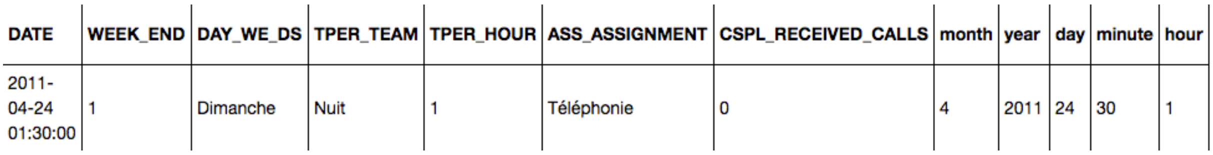
1. Data Preprocessing / Feature Engineering

2. Best Score and used algorithm (with and without multiplying by 1.7)

3. Other methods that…. FAILED

1 - Data Preprocessing / Feature Engineering

From the training dataset, we kept some features and added noew ones :



DATE	WEEK_END	DAY_WE_DS	TPER_TEAM	TPER_HOUR	ASS_ASSIGNMENT	CSPL_RECEIVED_CALLS	month	year	day	minute	hour
2011-04-24 01:30:00	1	Dimanche	Nuit	1	Téléphonie	0	4	2011	24	30	1

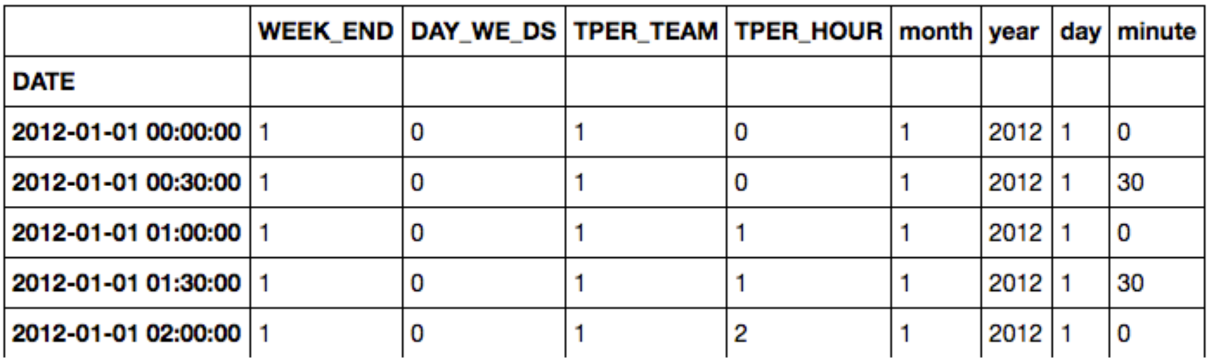
The Columns DATE, WEEK_END, DAY_WE_DS, TPER_TEAM, TPER_HOUR, ASS_ASSIGNEMENT and CSPL_RECEIVED_CALLS are the column that we took from the training dataset.

Then we added the time variable month, year, day, minute and hour using the DATE column.

All the string variables had to be transformed into categorical ones.

We did this thanks to sklearn.preprocessing functions, transforming :

Finally, we separated the features column from the calls column to have our training datas :



	WEEK_END	DAY_WE_DS	TPER_TEAM	TPER_HOUR	month	year	day	minute
DATE								
2012-01-01 00:00:00	1	0	1	0	1	2012	1	0
2012-01-01 00:30:00	1	0	1	0	1	2012	1	30
2012-01-01 01:00:00	1	0	1	1	1	2012	1	0
2012-01-01 01:30:00	1	0	1	1	1	2012	1	30
2012-01-01 02:00:00	1	0	1	2	1	2012	1	0

The submission file :

Obviously, we had to transform the submission file to be able to prediction the calls, as it has to be the same shape of the training dataset for some classifier that we used (e.g decision tree, naivebayes, gradient boost)

Algorithm we used :

The best score was obtained with combined models : We used decision tree on some Assignement, and time series on other assignement.

Before going through the explication of our models, we must inform you that we multiplied the number of calls by 2 for the training, because we observed that the linex function penalizes a lot when we underestimate the number of incoming calls, meaning that we had to surrestimate the predictions.

Decision tree :

For all the ASSIGNEMENTS except for Tech. Axa, Tech. Inter and RENAULT, w e used the decision tree classifier from scikit-learn.
For the training set we kept only the data from 2012 and 2013.

Time – Series :

We used time series for only 3 categories because of the beautiful shape of the data :
Tech. Axa, Tech. Inter, RENAU

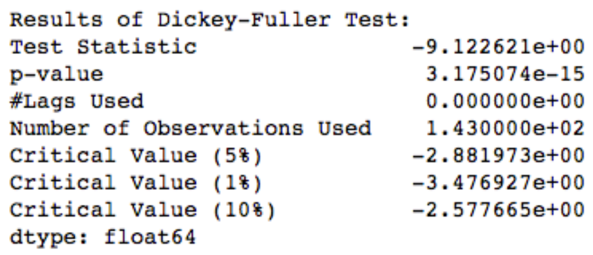
First we decomposed the data by the day of week. So we did a separate model for each single day of the week and for each hour, with and without half hour, meaning that for a single assignement **we trained 48 models**. That means that we had weekly datas to train the time series model. There were some missing values obviously, so what we did Is that we filled them in with the mean of the calls of the month.

Note also that we kept only the datas from 2012/12 and all 2013 as the submission file dates are in this range.

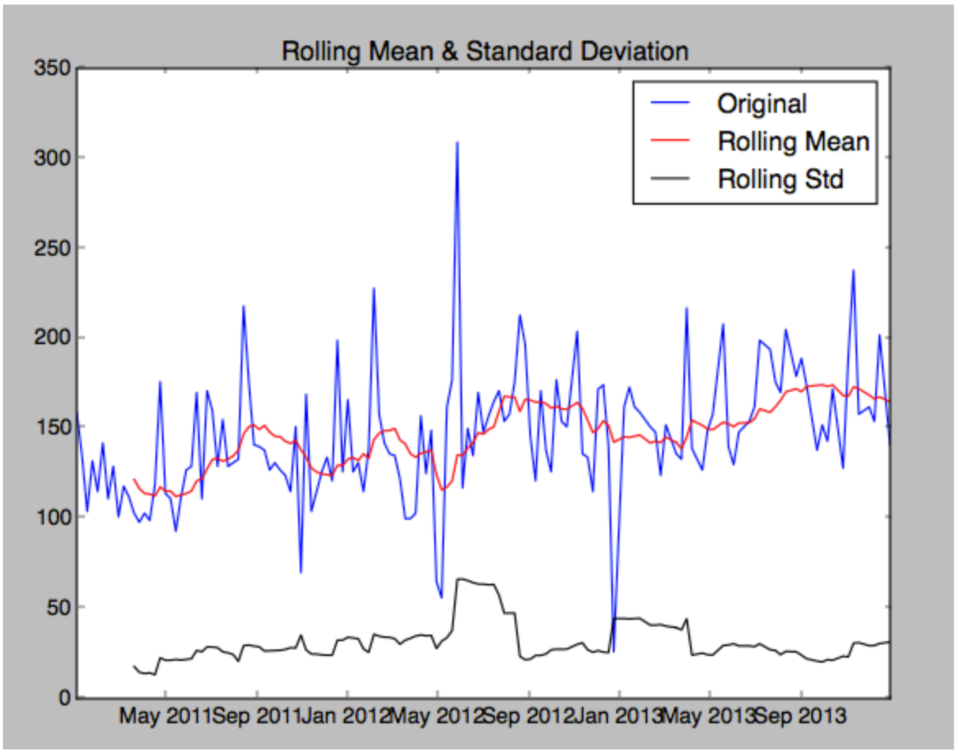
The model we used for the time series is LSTM from Keras library, with 4 input neurons and one output neuron, and we used 1000 epochs and 1 batch_size as parameters.

So, we came to the idea that these assignement were adequate for times series because we ploted their shape day by day and hour by hour first, and secondly because we did a statistical test for testing the stationarity of the data, thanks to to the statsmodels module, including a function for stationarity test.

Here Is the output of the function for ASS_ASSIGNEMENT = 'Tech. Axa', hour = 16h30, day = Thursday.



```
Results of Dickey-Fuller Test:
Test Statistic                 -9.122621e+00
p-value                         3.175074e-15
#Lags Used                      0.000000e+00
Number of Observations Used     1.430000e+02
Critical Value (5%)            -2.881973e+00
Critical Value (1%)            -3.476927e+00
Critical Value (10%)           -2.577665e+00
dtype: float64
```



The test statistic Is equal to -9 wich Is below to the critivalue, proving that the data are stationnary.
This Is almost the same for all the other hours and days. But even when it's not the case, the LSTM model shown good results.

<u>Comments on the score on the leader board :</u>

Without muliplying the calls by 2, and training a decision tree on 2012/2013 data, and using the DT for all assignement we got a score of 16.
Then after muliplying the calls by 2, we got 0.8.

After we combined DT and Time series, we obtained our final score : 0.63

We are almost convinced that if we run the time series model for 1 or 2 more ASSIGNEMENT, we could get a better result, but even with a postpone of the due date, we were short of time. Indeed, a single run for the time series takes more than 30 minutes, so testing many parameters took us days and days.

Other methods that failed :

We also tried the naive bayes classifier which gave us very bad results, and we tried gradient boosting that never ended runiing, so we tried xgboost, it was faster but not for all assignement, so we couls not end the tests with this classifier and we didn't get any result.

Also, we would like to mention that our first try was using LSTM for all the assignements, for all hours and half hours, for every single day of the week, so it was **48*7*26 = 8736** differents run that took us 3 days to be achieved and at the end, it was a big FAIL, so we were a bit angry, and were even more angry when we observed that a simple decision tree classifier was giving excellent results.
But we learned a lot from that experience. Never trust completely neural networks.