University Paris-Saclay

AXA Data Challenge - Assignment

Team DataMiners:
Yuxiang WANG
Hong-lin LI
firstname.lastname@u-psud.fr

Supervisor:
Nikolaos Tziortziotis
Jesse Read

January 11, 2017

Abstract

This project constitutes an AXA data challenge, whose purpose is to apply data mining and machine learning techniques for the development of an inbound call forecasting model. Our target is to predict the number of incoming calls for the AXA call center in France, in a half-hour time slot, by analyzing three years' history data. This is a regression prediction problem.

1 Introduction

Our project consists of four parts: Data pre-processing, Feature-engineering, Learning algorithm and Evaluation. Each part plays a very important role in this project and will affect the final result.

2 Data pre-processing

2.1 Data analysis

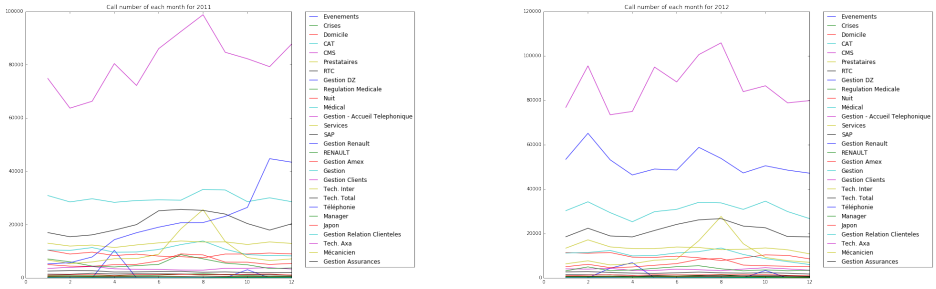
Our raw data is very complex. Each record of data contains 86 features. In fact, most of features do not help our prediction. And if we read all the features at once, our home notebook may be overloaded. So we decide to choose same features among them.

2.2 Visualization

For helping view the data distribution features, we have built many statistic figures such as Figure 1. And there are many other interesting figures in our visualization.ipynb file.

2.3 Data separation

If we process the data directly by one program, it will be to overloaded. And from figure 1, we can see that the call number distributions of each assignment are very different. And we find they are totally independent among them. So for each type of ASS_ASSIGNMENT, we generate a csv



(a) 1a (b) 1b

Figure 1: call numbers of each month of 2011 & 2012 for each assignment.

file. And we read it when we build a model for it. In the submission file, we find that there are 26 cases to be predicted, so we generate 26 csv files for next step.

3 Feature engineering

3.1 Feature selection

1. DATE: most important feature. But it is a type of string as "2011-01-01 00:00:00.000".It contains the year, month, day and the "half-hour" time.
2. DAY_WE_DS: the weekday from Monday to Sunday.
3. WEEK_END : whether it is on weekend.
4. ASS_ASSIGNMENT: the assignment for the calls.
5. CSPL_RECEIVED_CALLS: the number of the received calls.

3.2 Feature extraction

1. DATE: We need several features from "DATE", so we transform it to "year","month","day" and "time" with type int by using Pandas Python library(data.DATE.str.split() and data.str.get()). But the "time" is hard to represent, so we mark it with a consecutive integers(from 0 to 47).

2. DAY_WE_DS: We get the values of weekday which corresponds to the "DATE" and we mark them with integers(from 1 to 7)

3. WEEK_END: The same as the last case. And we mark it with 1 and 0 (1 present weekend).

4. ASS_ASSIGNMENT: At the beginning of our project, we consider to build a model from all assignments, but after testing, we found it was not good enough. The figure 1 shows that different assignments have distinct distribution. So we decide to separate the assignments and build a special model for each assignment.

5. CSPL_RECEIVED_CALLS: For each "$ASS\_ASSIGNMENT$", we calculate the sum of the received calls for each "half-hour" time.

6. ferie: we search some information of the french off-day from internet, and we make it as a feature in our project.

7. Count: the sum of the received numbers.

4 Learning algorithms

4.1 Decision tree

At first, we try the Decision tree regression. The operation time is small, but the loss function of cross-validation is very bad, especially for the group "Telephonie", it is more than 1e10.

4.2 Xgboost and RandomForest

Then, we try Xgboost and RandomForest, except $ASS =$ "$telephonie$", the 25 others' results are good enough, most of them are less than 0.1, and all of them less than 0.5, the results are acceptable. But for $ASS =$ "$telephonie$", the value of loss function is not stable. Sometimes it can be 1 or 2, sometimes it change to thousands.
Last two cases, we tried to make a submission, the result is incredible, it is always more than thousands.

4.3 Gradient boosting

After that, we try the Gradient boosting. It gives us some hope to continue our project. We set the $max\_depth = 10$ and $n\_estimators = 500, loss =$ "*huber*" and $alpha = 0.999$. We make a grid_search and determine max depth and number of estimators. and then Huber function is more similar to our loss function(better than "ls"), after some tests, $alpha = 0.999$ will be better than 0.1. After cross validation, the average of loss function is smaller than before, less than 10. and we make several submissions, the results were not stable. sometimes it could be 200, and sometimes it could be less than 2. And the 1.2 was the best among them.

4.4 Bagging

In order to get a better result, we try to use Bagging. At first we make bagging based on XgboostRegressor, but if the $n\_estimators$ is too small, the submission is bad, if we turn the $n\_estimators$ to a big number, our computer will be overloaded. The same situation for randomforest and gradient boosting. So, we try decision tree as the basic model, and make bagging. After modifying some parameters($max\_depth = 30$ for decision tree and $n\_estimators = 50$ for bagging) and making some submission, we get our best result 0.84. But the local loss function was not the best.

4.5 Remarks

After some tests, we found that the "tech.axa", "CAT" and "telephonie" are very bad by cross validation, so we separate them from the others, and we try to find the best model for each of them.

4.6 Autoregressive models

Finally, to improve our accuracy, we try to use the autoregressive models. It uses a set of methodologies of time series analysis. And with the help of pandas and the library statsmodels.tsa.arima_model, we test two data set, Telephonie and Domicile. The call number distribution of Domicile is very regular and we can get a great regression model of ARIMA. For the data Telephonie, it depends also on the year. So we think it needs more data preprocessing to increase the stability of time series which is necessary to use

time series analysis. But the problem is that we don't have enough time to go on our study and test.

5 Conclusion

5.1 Problems

1. We do not clearly know how to determine a basic model according to a specific loss function. For same model, such as xgboost, RandomForest, they have some options for loss function, but we can not define a specific loss function.

2. Because of the loss function, there are too much difference between local test and the online test. Because if it exists a number too distinct that is far from the original number, the result will become too bad, it is exponential increase. So we do not know how to build a model which is perfect for local cross-validation and online-submission.
(Our local cross validation, the gradient is the best, but online test the decision tree is the best)

5.2 Improvements

1. We think we can improve our feature engineering, maybe our feature is too simple, we should find more information for the original data.

2. We have seen some hope using autoregressive models because it runs well generally on the time series forecasting problems.

3. If use neural network, maybe it will be good for our project, because we can define the loss function and determine the structure of the neural network.