INF554: Machine Learning I
École Polytechnique
AXA Data Challenge - Assignment

Data Science and Mining (DaSciM) Team

October 21, 2016

1 Description of the Assignment

Whether in a contact center or bank branch environment, workforce managers face the constant challenge of balancing the priorities of service levels and labour costs. In the case where the demand (inbound calls, outbound calls, emails, web chats, etc.) is greater than supply (the agents themselves), the price, in the form of reduced service levels, falling customer satisfaction and poor agent morale, rises. On the other side, when supply is greater than demand, service levels tend to improve, but at the cost of idle and unproductive agents. The key to optimising the bottom line performance of a contact centre is to find a harmonious balance between supply and demand. This bottom line performance is directly impacted by the direct costs of hiring and employing your agents, but it is also influenced by client satisfaction, agent morale and other factors. Taking all the above into consideration, the basis of any good staffing plan is an accurate workload forecast. An accurate forecast gives us the opportunity to predict workload in order to get the right number of staff in place to handle it.

The specific project constitutes an AXA data challenge, where its purpose is to apply data mining and machine learning techniques for the development of an *inbound call forecasting system*. The forecasting system should be able to predict the number of incoming calls for the AXA call center in France, on a per "half-hour" time slot basis. The prediction is for seven (7) days ahead in time. More specifically, based on the history of the incoming calls up to a specific time stamp (you cannot use data/features that corresponds to future time slots), the proposed model should be able to predict the number of the calls, received seven (7) days later. In this way, the problem can be seen as a regression problem where the goal is to design a model that achieves to predict the incoming calls of the AXA call center with high accuracy. A detailed description of the dataset that will be used for the training of your proposed models is given in Section 2 . The specific dataset includes telephony data retrieved from AXA call centers, and corresponds to the period spanning the calendar years 2011, 2012 and 2013. Last but not least, the final evaluation of your model will be given by using a *Leaderboard platform* (a detailed description about the *Leaderboard platform* can be found in Section 4).

1.1 Data Challenge Ceremony

As the data are provided by AXA Assistance and this data challenge forms part of the activities of the *AXA-X DaScIS chair* - after the evaluation of your submissions there will be a reception organized by the chair. You will be informed on the details in due time.

2 Dataset Description

In this section, we present the structure of the training dataset (`train.csv` [1]) that will be used for the training of your model. As mentioned previously, the training dataset includes telephony data derived from AXA call centers, and correspond to the calendar years 2011-13. Figure 1 shows how the training dataset has been derived. Each one of the rows of the dataset corresponds to the number of incoming calls for each different combination of values for the following attributes: `DATE` (time stamps in half hour slots), `SPLIT_COD, ACD_COD, ASS_ASSIGNMENT`. Please consider that some combinations may not be present on the dataset. For a detailed description of the attributes refer to the `field_description.xlsx`[2] file.

The *objective of your work* here is to build a model (or a set of models) able to predict the number of incoming calls (`CSPL_RECEIVED_CALLS`) for seven days after the current/given date for each different set of values of the attributes: `Date`(time stamps in half hour slots), `ASS_ASSIGNMENT`.

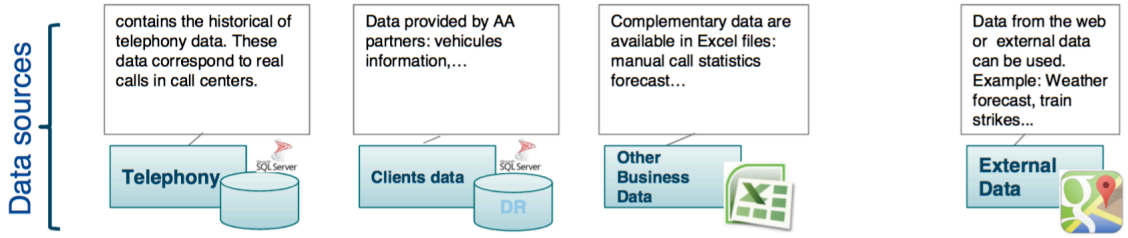

Figure 1: Data Sources of the training dataset.

As you can easily observe, the dataset has some missing values for some attributes (`NULL`). In the preprocessing task, you should take care of a number of similar cases. In the case of features that take numerical values, one approach could be to replace the missing values with the mean value of this feature. Some other features may not be useful at the prediction task. It would be helpful to explore the dataset and try to deal with such cases. Additionally, some of the features take values that correspond to a string (e.g., the `TPER_TEAM` feature takes values `Jours` and `Nuit`). In such cases, we can create two new features (i.e., add two new columns to the data matrix) which correspond to the two possible strings. Thus, if the `TPER_TEAM` feature takes the value `Jours`, the feature that corresponds to `Jours` will become equal to 1, while the feature that corresponds to `Nuit` will become equal to 0.

As part of the preprocessing step, you can also apply feature selection techniques to keep a subset with the most informative features or dimensionality reduction methods (e.g., Linear Discriminat Analysis) to create a representation of the data in a new space preserving some of the underlying properties of the data. It is also possible to create new features that do not exist in the dataset, but can be useful in the forecasting task. Thus, you can create a new feature (i.e., add a new column to the data matrix) to represent this information (this is known as feature engineering or generation).

3 Summary of the Pipeline

The pipeline that will be followed in the project is similar to the one followed in the labs. In the following, we briefly describe each part of the pipeline.

- *Data pre-processing*: After loading the data, a preprocessing task should be done to transform the data into an appropriate format. In the previous section, we discussed some of these points.

[1] Training dataset:
`https://moodle.polytechnique.fr/pluginfile.php/59386/mod_assign/introattachment/0/train_2011_2012_2013.7z.001?forcedownload=1`.
`https://moodle.polytechnique.fr/pluginfile.php/59386/mod_assign/introattachment/0/train_2011_2012_2013.7z.002?forcedownload=1`

[2] Dataset description:
`https://moodle.polytechnique.fr/pluginfile.php/59386/mod_assign/introattachment/0/field_description.zip?forcedownload=1`.

- *Feature engineering - Dimensionality reduction*: The next step involves the feature engineering task, i.e., how to select a subset of the features that will be used in the learning task (feature selection) or how to create new features from the already existing ones (see also previous section). Moreover, it is possible to apply dimensionality reduction tecniques in order to improve the performance of the algorithms.
- *Learning algorithm*: The next step of the pipeline involves the selection of the appropriate learning (i.e., regression) algorithm for the problem. At this point, you can test the performance of a number of different algorithms and choose the best one. Additionally, you can follow an ensemble learning approach, combining many regression algorithms.
- *Evaluation*: In Section 4, we describe in detail how the evaluation will be performed.

4 Evaluation

You will build your model based on the training data contained in the `train.csv` file. To do this, you can apply *cross-validation* techniques[3]. The goal of cross-validation is to define a dataset to test the model in the training phase, in order to limit problems like overfitting and have an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, like the test dataset that will be used to assess your model).

In k-fold cross-validation (assuming your model allows this type of validation), the original sample is randomly partitioned into k equal size subsamples. On the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation (i.e., test) data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation (average accuracy of the model).

4.1 How to evaluate your model?

Of course, having a good model that achieves good accuracy under cross validation does not guarantee that the same accuracy will be also achieved for the test data. Thus, the final evaluation of your model will be done on the test dataset contained in the `submission.txt`[4] file. So, after having a model that performs well under cross-validation, you should train the model using the whole training dataset and test it on the test dataset as described below.

Submission file

For the final evaluation of your model, you have to predict the number of calls that will be received at a number of different combination of values of the following attributes: `DATE` (corresponding to half hour slots), `ASS_ASSIGNMENT`. More specifically, get the predicted number of calls for each instance (row) contained in the `submission.txt` file. Each row of the `submission.txt` file corresponds to a different combination `DATE` (corresponding to half hour slots) and `ASS_ASSIGNMENT` (Table 1 presents a snapshot of the `submission.txt` file). In the `submission.txt` example file, all the prediction values are set equal to zero. You must replace those values with your predicted ones. Do not change the format of the file (fields separated by tab). The final evaluation of your model will be made based on *LinEx* loss function (see Section 4.2 for a detailed description).

The data corresponding to the required dates (the listed dates in the `submission.txt` file) are omitted from the dataset. Moreover the data on a 6-day window a priori to those dates listed in the `submission.txt` file, are also omitted to ensure that you will not use them for the predictions of your submission.

[3] Wikipedia's lemma for *Cross-validation*: `http://en.wikipedia.org/wiki/Cross-validation_(statistics)`.

[4] Testing dataset:
`https:https://moodle.polytechnique.fr/pluginfile.php/59386/mod_assign/introattachment/0/submission.txt?forcedownload=1`

DATE	ASS_ASSIGNMENT	Prediction
2012-01-03 00:00:00.000	CAT	0
2012-01-03 00:00:00.000	Tlphonie	0
2012-01-03 00:00:00.000	Tech. Inter	0
2012-01-03 00:00:00.000	Tech. Axa	0
2012-01-03 00:00:00.000	Services	0

Table 1: First five rows of the `submission.txt` example file

Leaderboard Platform

The final evaluation of your model will be done using a private leaderboard platform, which is available at the following link: `http://moodle.lix.polytechnique.fr/data_chalenge/`. The specific platform will evaluate your predictions and the evaluation score as well as your position (with respect to the rest users) will appear in the Leaderboard. In order to make a new submission you just need to login to the platform (by using the identifier of your team along with your password) and upload the submission.txt file. Your final score will be the *best* one that you have achieved. Finally, **note that you can submit up to 10 entries per day**. Please be careful with the submission process as the submission counter resets 24 hours after your last submission.

4.2 Evaluation metric

The objective of this task is the development of an accurate inbound call forecasting system, able to keep the cost (human resources) at an affordable level and at the same time achieve a high level of customer satisfaction. Nevertheless, it is very difficult to define a standard metric in order to evaluate the customers' satisfaction. Intuitively speaking, an underestimate of the number of incoming calls in the call center is usually much more serious than an overestimate. In this direction, we adopt the *LinEx* loss as evaluation metric instead of the standard mean square error (MSE), which is given by:

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

where the true number of calls is y and that the predicted number (by your model) is $\hat{y}$; and $\alpha = -0.1$ which gives a relatively higher penalty to underestimating the number of calls. The final loss is averaged over all examples. This penalty is illustrated in Figure 2. Having such a loss function should encourage the design of algorithms towards building models that are not underestimate the number of calls.

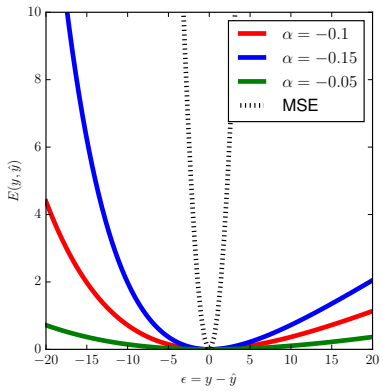

Figure 2: LinEx (for various α) compared with MSE, where y is the true number of calls, $\hat{y}$ is corresponding *predicted* number of calls.

5 Useful Python Libraries

In this section, we briefly discuss some useful tools that can be useful in the project and you are encouraged to use.

- For the preprocessing task which also involves some initial data exploration, you may use the `pandas` Python library for data analysis[5].
- A very powerful machine learning library in Python is `scikit-learn`[6]. It can be used in the preprocessing step (e.g., for feature selection) and in the calls forecasting task (a plethora of regression algorithms have been implemented in `scikit-learn`.). Recall that we have already used the `scikit-learn` in the labs.
- Finally, you are always encouraged to propose and develop your own learning algorithms or use the ones developed in the labs.

6 Details about the Submission of the Project

As part of the project, you have to submit the following:

1. Your **final submission file** (`submission.txt`), which contains the estimated number of calls.
2. A 2-5 **pages report**, in which you should describe the approach and the methods that you used in the project. Since this is a real data science task, we are interested to know how you dealt with each part of the pipeline, e.g., if you have created new features and why, which algorithms did you use for the calls forecasting task and why, their performance (accuracy and training time), approaches that finally didn't work but is interesting to present them, and in general, whatever you think that is interesting to report). Also, in the report, please provide the names and the emails of the team members, and the identifier of your team (e.g., `INF554`).
3. A directory with the code of your implementation.
4. Create a `.zip` file with the `team_identifier.zip` (the identifier of your team), containing the code and the report and **submit it to Moodle platform (one submission per team).**
5. **Deadline: Friday, December 9, 23:59.**

7 Oral presentation

Each team will give an oral presentation on **Thursday, December 15**. More details will be announced later.

8 Project evaluation

Your final evaluation for the project will be based:

1. on the leaderboard score (according to the *LinEx* loss function, Section 4.2) of the proposed model,
2. on the report and code that you will submit,
3. on your oral presentation.

[5] `http://pandas.pydata.org/`.
[6] `http://scikit-learn.org/`.

Appendix

Even though the specifics of the problem defined in this document are unique, the general task has been dealt in the past before. Therefore , we provide you here a list of approaches (features and models) which were among the top ones in the previous versions of this assignment.

Feature engineering BEYOND the raw data:

- Considering the date/time aspect past feature have included :
  - time since epoch
  - time since start of day
  - time as a categorical feature
  - month
  - week day
  - week end
  - night/day
  - "day off"
  - holidays: Taken from outside sources which indicate holidays or periods of vacation.
- average values of the target variable over various windows sizes (average on the past values)
- dummy variables from existing categorical ones

Models: The most prominent models were:

- Tree regressors
- Random Forest regressors
- Gradient boosting regressors
- Autoregressive models.

You are encouraged to explore these options but more importantly to explore solutions beyond them!