LANL EARTHQUAKE PREDICTION

Sofiene Omri

Hamza Ferchichi

Taieb Jlassi

Hakim Gomri

Under the supervision of:

Florian Yger

2019

PARIS DAUPHINE UNIVERISTY

1. Introduction

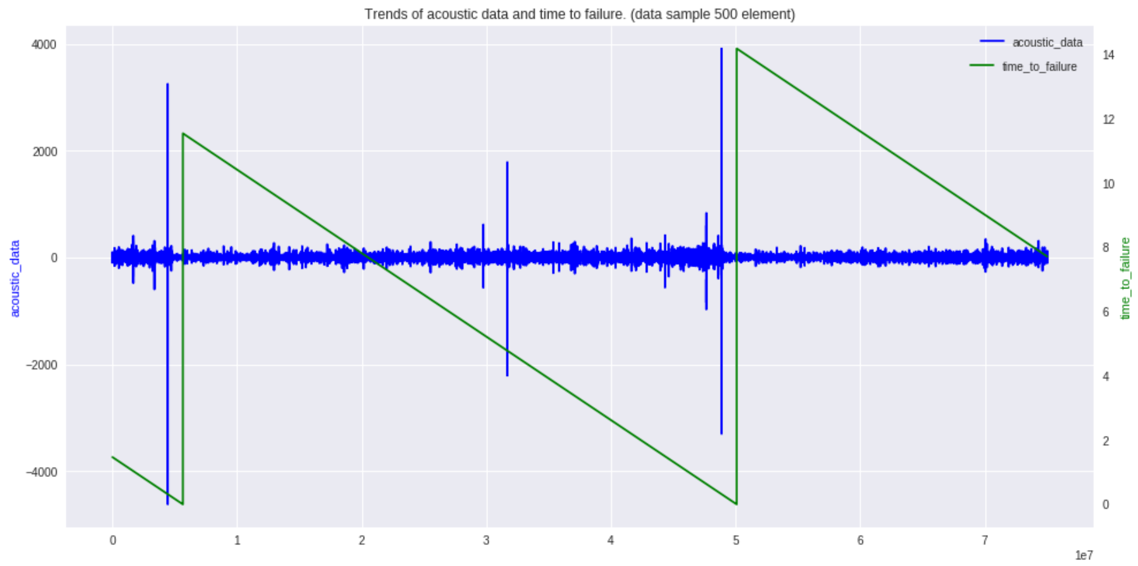
The goal of this project is to build a reliable training model for predicting when the earthquake will take place. Specifically, real-time seismic data will be used to predict the time remaining before a laboratory earthquake takes place. The Kaggle dataset as provided by Los Alamos National Laboratory consists of training set of a single, continuous segment of experimental data and a test set of small segments of 150000 observation each.

In order to build a reliable ML algorithm, we have adopted two main approaches after dividing the training set into a discrete set of 150000 long segments. In the first approach we have amid to extract a set of predefined statistical features that best describe the data sequences. We have relied mainly on the tsfresh package developed at the MIT to extract more than 500 statistical features. Form this set we have selected the top features using Kendall test of independency, and further dimensionality reduction using the scikit learn lib. In the second approach, we rely mainly on the LSTM capacity to use the sequences as they are to train the model and solve for the best prediction.

In this project we have used three main ML algorithms; SVM regression, CatBoost regression and LSTM network. The LSTM model proved to be the most efficient even though the experiment is lacking due to limited hardware resources to train the model over higher number of sequences. Finally, we have tried to combine weighted model prediction to see if the error rate can be improved.

2. Exploratory analysis

Given that the data is too big we have only extracted 500 sequence of 150000 observation. Now plotting both the acoustic and time to failure we get the below graph.



We can note that the acoustic data oscillates around zero with high peaks of fluctuations just before failure occurs and this looks a cyclic effect. Another take away would be that visually failures can be expected if the huge acoustic fluctuation is followed by lower oscillations.

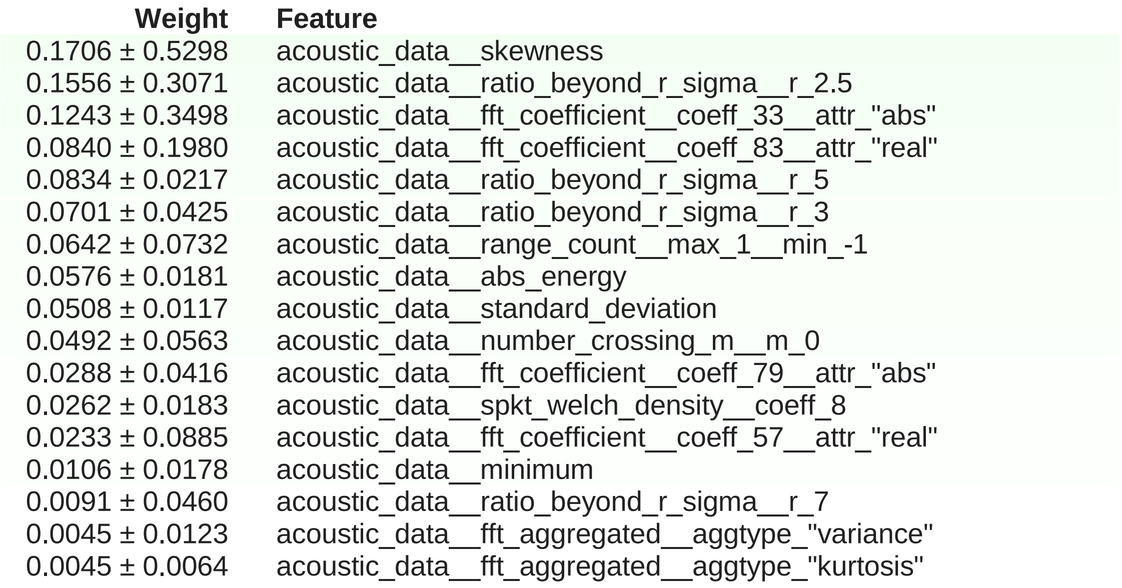
3. Feature Extraction

In order to deal with huge sequences, we have created a set of statistical features that best describe each sequence. We have relied mainly on the tsfresh package that over a variety of feature. From the provided features we have selected the least computational costly to extract from the given data. This includes the following:

- Usual aggregations: mean, std, min, max, median, sum, variance, kurtosis
- Description of the additional statistical features from the tsfresh package can be found here:https://tsfresh.readthedocs.io/en/latest/text/_generated/tsfresh.feature_extraction.feature_calculators.html?fbclid=IwAR1hfvnEeld8MVMChLllqQZ68nLUc-GyyAOYWHi_AnDFkIoN3h3_9M13DrI
- We have only selected the feature listed in the notebook in the appendix

4. Feature Selection

Out of the generated features some are relevant and can have an impact on the regression outcome while others do not. In order to select the most relevant features we have opted for the Kendall rank statistical test to establish whether a given attribute and target value (time to failure) may be regarded as statistically dependent. The test is non-parametric with null hypothesis H0 means independency. We have selected feature for alpha equal to 0.01 resulting in selection of 50 features. On these features we have used the importance permutation module in Scikit learn:



Weight	Feature
0.1706 ± 0.5298	acoustic_data__skewness
0.1556 ± 0.3071	acoustic_data__ratio_beyond_r_sigma__r_2.5
0.1243 ± 0.3498	acoustic_data__fft_coefficient__coeff_33__attr_"abs"
0.0840 ± 0.1980	acoustic_data__fft_coefficient__coeff_83__attr_"real"
0.0834 ± 0.0217	acoustic_data__ratio_beyond_r_sigma__r_5
0.0701 ± 0.0425	acoustic_data__ratio_beyond_r_sigma__r_3
0.0642 ± 0.0732	acoustic_data__range_count__max_1__min_-1
0.0576 ± 0.0181	acoustic_data__abs_energy
0.0508 ± 0.0117	acoustic_data__standard_deviation
0.0492 ± 0.0563	acoustic_data__number_crossing_m__m_0
0.0288 ± 0.0416	acoustic_data__fft_coefficient__coeff_79__attr_"abs"
0.0262 ± 0.0183	acoustic_data__spkt_welch_density__coeff_8
0.0233 ± 0.0885	acoustic_data__fft_coefficient__coeff_57__attr_"real"
0.0106 ± 0.0178	acoustic_data__minimum
0.0091 ± 0.0460	acoustic_data__ratio_beyond_r_sigma__r_7
0.0045 ± 0.0123	acoustic_data__fft_aggregated__aggtype_"variance"
0.0045 ± 0.0064	acoustic_data__fft_aggregated__aggtype_"kurtosis"

Weight	Feature
0.0039 ± 0.0083	acoustic_data__ar_coefficient__k_10__coeff_0
0.0030 ± 0.0490	acoustic_data__fft_coefficient__coeff_54__attr_"real"
0.0011 ± 0.0028	acoustic_data__quantile__q_0.6
0.0007 ± 0.0009	acoustic_data__quantile__q_0.3
0.0007 ± 0.0018	acoustic_data__linear_trend__attr_"stderr"
0.0006 ± 0.0032	acoustic_data__fft_aggregated__aggtype_"skew"
0 ± 0.0000	acoustic_data__quantile__q_0.1
0 ± 0.0000	acoustic_data__quantile__q_0.9
0 ± 0.0000	acoustic_data__quantile__q_0.4
-0.0004 ± 0.0010	acoustic_data__quantile__q_0.7
-0.0018 ± 0.0024	acoustic_data__quantile__q_0.2
-0.0025 ± 0.0068	acoustic_data__agg_autocorrelation__f_agg_"mean"__maxlag_40
-0.0041 ± 0.0097	acoustic_data__mean_abs_change
-0.0076 ± 0.0248	acoustic_data__number_peaks__n_50
-0.0081 ± 0.0039	acoustic_data__fft_aggregated__aggtype_"centroid"
-0.0114 ± 0.0238	acoustic_data__ratio_beyond_r_sigma__r_1
-0.0121 ± 0.0330	acoustic_data__variance
-0.0252 ± 0.1632	acoustic_data__ratio_beyond_r_sigma__r_6
-0.0409 ± 0.0264	acoustic_data__ratio_value_number_to_time_series_length
-0.0418 ± 0.4892	acoustic_data__number_peaks__n_10
-0.0426 ± 0.0973	acoustic_data__number_peaks__n_5
-0.0568 ± 0.1276	acoustic_data__ratio_beyond_r_sigma__r_2
-0.0625 ± 0.1939	acoustic_data__sum_of_reoccurring_values
-0.0822 ± 0.0183	acoustic_data__absolute_sum_of_changes
-0.1277 ± 0.0577	acoustic_data__range_count__max_1000000000000.0__min_0
-0.1620 ± 0.1560	acoustic_data__number_crossing_m__m_-1
-0.1938 ± 0.6845	acoustic_data__spkt_welch_density__coeff_2
-0.2243 ± 0.4233	acoustic_data__ratio_beyond_r_sigma__r_0.5
-0.4984 ± 0.7880	acoustic_data__percentage_of_reoccurring_values_to_all_values
-0.5561 ± 0.3476	acoustic_data__quantile__q_0.8
-0.5649 ± 0.5024	acoustic_data__maximum
-1.4494 ± 0.5071	acoustic_data__spkt_welch_density__coeff_5
-1.6328 ± 0.4824	acoustic_data__number_peaks__n_3
-4.2495 ± 0.7691	acoustic_data__number_peaks__n_1

This module selects features with the highest importance weights on the prediction. We have used a threshold of 0 resulting in final 25 selected features. However, in the upcoming experimental phase we have used all the 50 and the 25 features to compare the SVM and Catboost performance with different number of features.

5. Training Phase

5.1. Feature based training

5.1.1. SVM

Before running the SVM model we have standardized the features between one and zero to account for the widely variable range of values of the features. Now given the high number of extracted features, a natural choice of ML algorithm would be linear SVM. In order to solve for the optimal parameters, we have used grid search to find the following parameters: C, Nu and the kernel to use. Following combinations of feature selection and SVM models are considered:

- SVM with the 50-features selected based on the Kendall test
- SVM with the 25-features selected based on the permutation importance

Under the first combination the best parameters are: {'C': 0.1, 'gamma': 1, 'kernel': 'linear', 'nu': 0.95} with an RMSE Test Score of 2.3872. However, under the second combination the best parameters {'C': 0.1, 'gamma': 1, 'kernel': 'linear', 'nu': 0.85} with an RMSE Test Score of 2.5954. We can note that SVM performs better under high dimensionality

5.1.2. Catboost

Boosting algorithm proved to be very successful and performant predictors with Catboost being easy to train compared to its Xgboost counterpart. We will use the Catboost regressor to predict the time of failure for lab earthquakes. In fact, Catboost reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting which leads to better generalized. We have also used a simple grid search to find some of the optimal parameters.

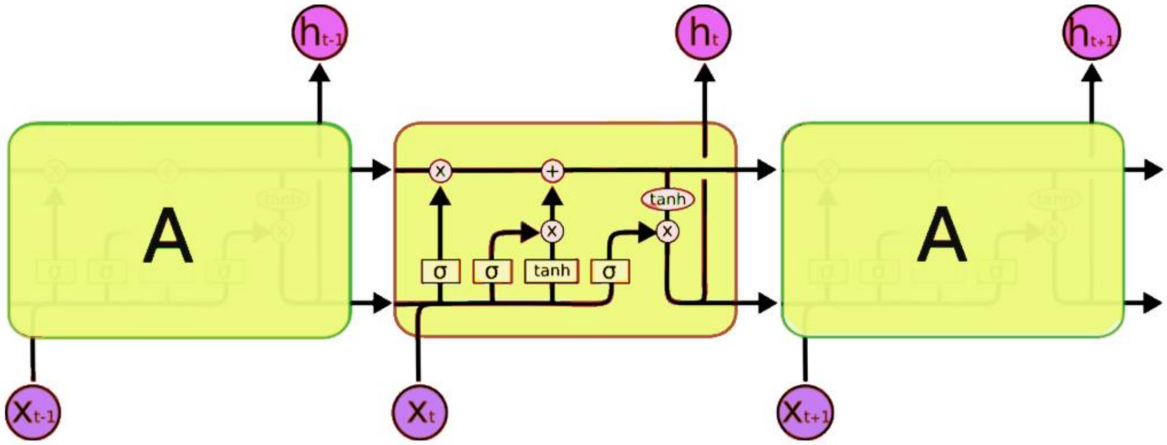
Similar to the SVM approach we will use the same combinations.

- Catboost with the 50-features selected based on the Kendall test
- Catboost with the 25-features selected based on the permutation importance

Under the first combination the best parameters are: {'depth': 10, 'learning_rate': 0.001} with an RMSE Test Score of 4.0367. However, under the second combination only the score worsened a bit for a Test Score of 4.4590 RMSE. We can note that similar to SVM Catboost performs better with high number of features.

5.2. LSTM with full sequence

Given that we are dealing with sequence of historical data, LSTM based nets seems a good choice. This is due to the network capacity to selectively remember and forget certain information thus capable of learning long-term dependencies. The LSTM architecture is composed of three main gates; Input, forget and output gates.



The forget gate (sigmoid layer) is used by the LSTM to decide what information to throw away from the cell state. Then the input gate (sigmoid layer) is used to point which values to update and finally the output gate (sigmoid layer to be multiplied by tanh) decides what parts of the cell state to output.

In our architecture we have used one LSTM layer, followed by dense layer, then a dropout layer and finally a one unit dense. The dropout layer aims to regularize the network by randomly killing neurons as per the dropout rate set. As we can see in the appendix and the attached notebook the LSTM model achieved the best RMSE of 0.7416.

Its also worth mentioning that LSTM RMSE degrades in case the output value is not scaled to over 2.3.

6. Conclusion

In this project we have dealt with sequence prediction through the usage of three different models under tow main approaches. An approach in which we extract various features then use SVM and CATboost algorithms for regression and in the second approach we used LSTM straight forward on sequences.