EARTHQUAKE PREDICTION

Vijay Kumar Kodari | Sri Krishna Tirthala | Dharma Teja Bandaru

[CS519 APPLIED MACHINE LEARNING I] [Open Machine Learning Project]

1.Introduction:

Man has always looked to computer systems for help in every area of human endeavours and even in understanding very complex systems as the human ability is limited in handling the amount of data generated every day. As the size of this data has increased considerably, the need to explore the use of computer system to learn from the data became imperative.

Machine learning [1], provides computer systems the ability to learn autonomously from experience and improve as well without being explicitly programmed. Natural disasters cause massive casualties, damages and leave many injure. Human beings cannot stop them, but timely prediction and due safety measures can prevent human life losses and many precious objects can be saved. Earthquake is one of the major catastrophes. Unlike other disasters, there is no specific mechanism for earthquake prediction, which makes it even more destructive. Though many of them say that it is impossible to make earthquake predictions, few Scientists declare that it is a predictable phenomenon.

2.Motivation:

Believing that earthquakes could be predicted, from the past experiences we can say that we can avoid life loss and property damage if we have an estimation on severity of earthquake that will occur in near future.

On a yearly basis, there are approximately 14,000—16,000 fatalities due to earthquakes [2]. The count of fatalities per year due to earthquakes when compared to other disasters and accidents might be less. Whereas, an abrupt and a huge earthquake in a city could take away thousands of lives at once with huge loss of property as well. The solution mentioned in the project doesn't directly help in predicting the earthquake but it forms a good resource for the people who are actually working on it since the dataset used in the project is artificially created in the lab environment.

3.Problem Definition:

3.1 What could be the Solution?

Predicting the time left for the quake to occur when the acoustic signal is passed as input from the artificial created lab environment.

3.2 What are Solution Benefits?

As mentioned in the motivation the results from this project form a very good resource for the actual earthquake prediction.

4.How would I solve the problem?

4.1 Exploratory Data Analysis and Data Visualisation:

The data set is huge with 600 million rows of data with two columns namely “acoustic data”, and “time to failure”.

Acoustic data: It is the acoustic signal measured in the laboratory experiment.

Time to failure: The time left for a failure/next failure to occur.

In the below graph, we can see visualize the pattern for the 1% data of the whole dataset between the acoustic data and the time to failure. If we observe, whenever there is a spike in the activity of seismic data there is also a rise in the time to failure.

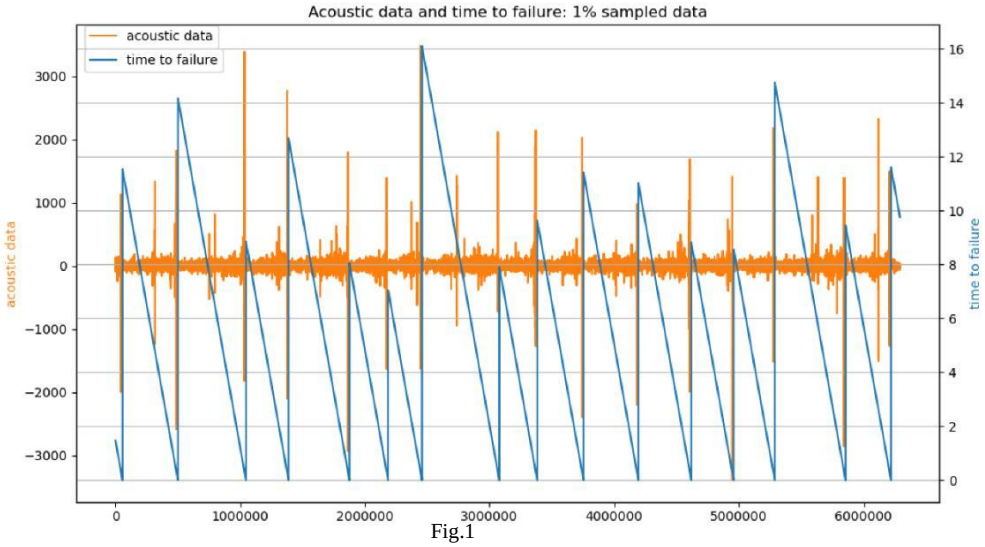


Fig.1

Here as we analyse the data further, we can find some interesting relation between the variables:

The below figures(Fig.2 & Fig.3) shows us how the variables change over time .

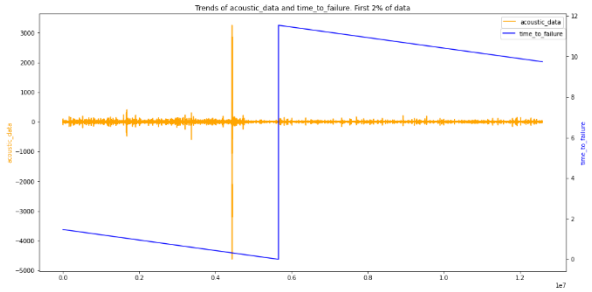


Fig.2

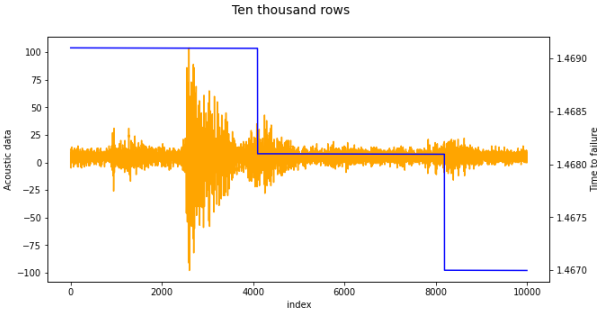


Fig.3

Fig.2 shows us the first 2% of the whole dataset, how the “time to failure” varies when the “acoustic data” has got a spike.

In Fig.3We can observe a fall in the time to failure for every 4000 samples around, even if the difference is in microseconds it tells us that the data is recorded from categorised samples.

4.2 Processing the data:

As we need to predict “time to failure” only with “acoustic data” present in test datasets, the data in the training dataset i.e.; acoustic data, time to failure alone won’t train the model accurate enough. So, we use **feature engineering** and add few common statistical features like Mean, Min, Max, Standard Deviation, Kurtosis, Skew, Quantile to the data in dataset (**train.CSV**.) Doing feature engineering will allow us to accurately represent the underlying structure of the data and therefore create the best model.

4.2.1 Features Added:

Mean: Mean or Average is a central tendency of the data i.e. a number around which a whole data is spread out.

Median: Median is the value which divides the data in 2 equal parts i.e. number of terms on right side of it is same as number of terms on left side of it when data is arranged in either ascending or descending order
Standard Deviation: Standard deviation is the measurement of average distance between each quantity and mean.
Kurtosis: It is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. It also helps us in finding outliers.
Skew: Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or undefined.
Quantile: Cut points dividing the range of a probability distribution into continuous intervals with equal probabilities or dividing the observations in a sample in the same way.

4.2.2 Splitting the data considering the Temporal component (time to failure) in the dataset:

We are not splitting up the data into training and testing sets but performing some pre-processing and modifying the dataset considering the temporality of "time to failure" data.
The process of modifying the dataset is:
Though there is abundance of data in the dataset, we are considering first 60 lakhs rows of data to train the model. Then dividing the data into 40 separate chunks of data leaving 1.5 lakhs rows of data in each chunk. We find all the above-mentioned special features for all the 40 chunks(6million/1.5million); Now each chunk has 1.5 million data rows, For each chunk we calculate all the feature engineered statistics like(mean, median---etc) of the acoustic data and the new row is formed with all the feature engineered statistics and the time to failure for the new row is considered to be the time to failure of the last row of the each chunk. Hence, we now have 40 new rows of data with columns mean, min, max, median, standard deviation, Kurtosis, Skew and quantile. Now as we have a better data to predict, we train the model, predict and obtain the MSE, R2 score values.
Why not the mean of "time to failure" values instead of last row data??
We really can't consider the mean of "time to failure" values because the value of a mean of two different set of values will never be equal, hence all the 40 values will be unique, and no two values are same. As we are considering the "time to failure" values to be as a label to predict we can't use mean value and hence we go with the last row data.

5. Implementation:

We can apply the regression and bring out the "time to failure" values. **Regression analysis** is a form of predictive modelling technique which investigates the relationship between a dependent and independent variable (s). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.
The regression methods [3] we are trying to implement are below:

5.1 Linear Regression:

Linear regression models predict a continuous target when there is a linear relationship between the target and one or more predictors. In this method, the variable to be predicted depends on only one other variable. This is calculated by using the formula that is generally used in calculating the slope of a line.

5.2 Decision Tree Regressor:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes.

5.3 Random Forest Regressor:

A Random Forest [4] is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap

Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

5.4 Catboost Regressor:

CatBoost is a machine learning algorithm that uses gradient boosting on decision trees. It is available as an open source library. It's particularly useful for predictive models that analyse ordered (continuous) data and categorical data. Catboost model one of the most efficient ways to build ensemble models. The combination of gradient boosting with decision trees provides state-of-the-art results in many applications with structured data.

5.5 Support Vector Regression:

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. The main Idea is to minimize error, individualizing the hyperplane which maximizes the margin. The kernel functions transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

Table: Comparison between the regression performances of the models. Data Considered is 6 Million rows

Regression Model	MSE-Train	MSE-Test	R^2-Train	R^2- Test	Time-Train	Time-Test
Linear Regression	0.353	0.102	0.311	0.948	0.003	0.002
Decision Tress Regressor	0.005	0.000	0.989	1.000	0.003	0.002
Random Forest Regressor	0.101	0.115	0.802	0.941	1.012	0.880
CatBoost Regressor	0.136	0.000	0.968	1.000	9.622	7.828
Support Vector Regressor	0.528	1.566	-0.030	0.194	0.167	0.139

Table: Comparison between the regression performances of the models. Data Considered is 60 Million rows

Regression Model	MSE-Train	MSE-Test	R^2-Train	R^2 -Test	Time-Train	Time-Test
Linear Regression	0.481	0.368	0.515	0.638	0.004	0.003
Decision Tress Regressor	0.324	0.226	0.672	0.778	0.003	0.003
Random Forest Regressor	0.061	0.059	0.939	0.942	1.220	0.994
CatBoost Regressor	0.619	0.128	0.966	0.993	74.966	39.385
Support Vector Regressor	0.409	0.375	0.587	0.632	0.448	0.217

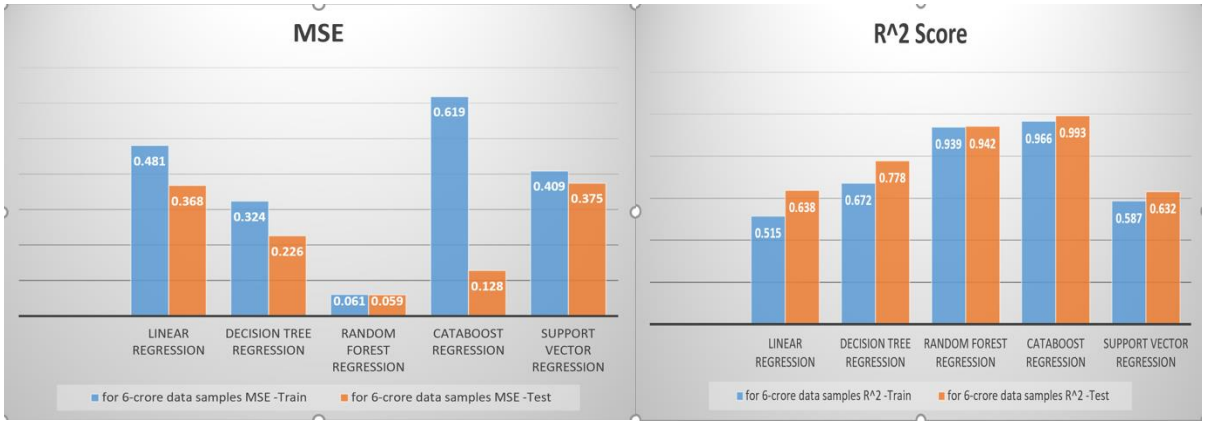
Reason for taking specific regressors:

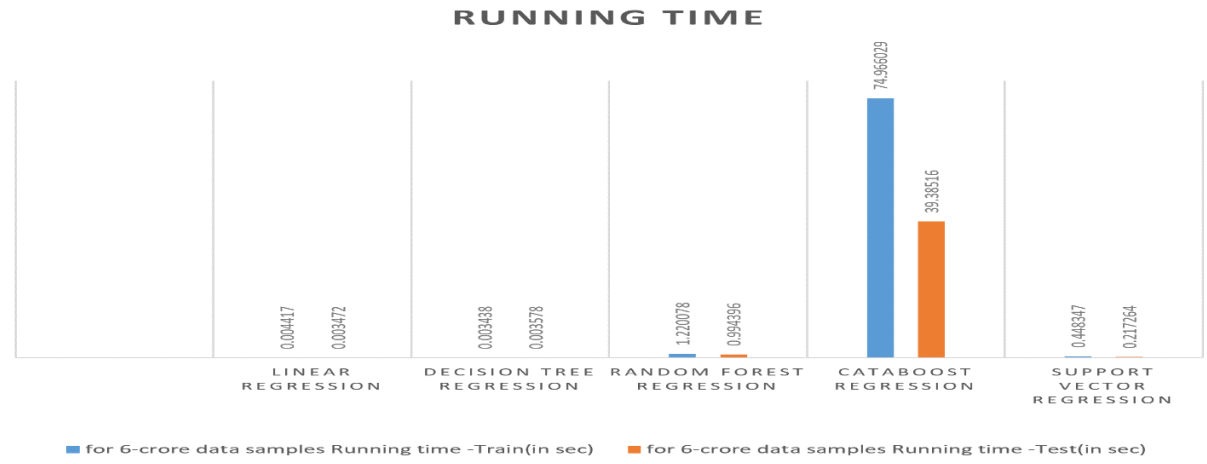
The reason behind selection of the models Linear regression, Decision Tree regressor and Random forest regressor is, these models are discussed in the class so we would like to test how these go on the dataset. We choose Catboost Regressor [8] since it is going very well in the current data science trend and reason for choosing Support Vector Regression is since support vectors are well defined and used in many regressions.

Take away points from the comparison tables:

When the data is considered is 6 million rows, we have tuned the data in such a way that 6 million rows are compressed to 40 rows(Using the Technique mentioned in 4.2.2), In that 40 rows (70% Training and 30% Testing Data)31 rows are used for training the data and 9 rows are used for testing the data, since the test data and train data are very minimal two of the models have reached maximum possible performance. But this doesn't happen when we considered 60 million rows since these are compressed to 400 rows training and testing samples significantly increased which results tough time to the models to give maximum performance possible as like the before case.

Analysis: The below graphs show us the comparison between the regression models





Above three figures Shows the comparison between performances of all Five regression techniques and for the both training and testing data considering the MSE, R^2 Score and running time values. **In conclusion**, considering all the statistics **Random Forest Regressor** preformed the best but **Catboost Regressor** is also considered to be best in general because it doesn't need standardisation to be done and wont use feature engineering, yet generate the best values.

6. Proposed solution:

Predicting the time remaining before the next laboratory earthquake i.e.; "time to failure", for which we are using the "acoustic data" from the training data and can accomplish the task and Comparison of regression techniques performances.

7. Future Implementation:

Furtherly, we can also use classification techniques and work on generating seismic signals in "acoustic data", so that the **WHERE** issue can also be solved as the values depend upon the laboratory locations. As it was said the WHERE issue can also be predicted, it is done using the **Seismic data** provided in the dataset. The seismic data is generally gathered when geoscientists visualize the subsurface of the earth using waves of sound to "map" geologic structures. So, the value varies from one location to another. Predicting the time left using the seismic data would obviously give information about how much time left before an earthquake in a specific location from where the seismic data is obtained.

8. REFERENCES:

1. https://expertsystem.com/machine-learning-definition/
2. https://www.researchgate.net/publication/307951466_Earthquake_magnitude_prediction_in_Hindukush_region_using_machine_learning_techniques
3. https://developer.ibm.com/technologies/data-science/tutorials/learn-regression-algorithms-using-python-and-scikit-learn/
4. https://www.geeksforgeeks.org/random-forest-regression-in-python/
5. https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db
6. https://medium.com/@hanishsidhu/whats-so-special-about-catboost-335d64d754ae
7. https://www.kaggle.com/c/LANL-Earthquake-Prediction/discussion/89909
8. https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/

The End
