CIS8395 - Final Report
LANL EARTHQUAKE PREDICTION OF UPCOMING LABORATORY EARTHQUAKES

Group Members –
Blesson Thomas
Karen Zhang
Shekha Saxena
Shivam Namdeo

Submitted By -
Shekha Saxena

Introduction:
Earthquakes are one of the major catastrophes and their unpredictability causes even more destruction in terms of human life and financial losses. Earthquake prediction remained an unachieved objective due to several reasons. One of the reasons is the lack of technology in accurately monitoring the stress changes, pressure and temperature variations deep beneath the crust through scientific instruments, which eventually results in unavailability of comprehensive data about seismic features. The second probable cause is the gap between seismologists and computer scientist for exploring the various venues of technology to hunt this challenging task.

Forecasting earthquakes is one of the most important problems in Earth science. Current scientific studies related to earthquake forecasting focus on three key points: when the event will occur, where it will occur, and how large it will be.

In this project, we will address when the earthquake will take place. Specifically, we will predict the time remaining before laboratory earthquakes occur from real-time seismic data.

We try to achieve good prediction accuracy and the physics are ultimately shown to scale from the laboratory to the field, researchers will have the potential to improve earthquake hazard assessments that could save lives and billions of dollars in infrastructure.

Project Description
Predicting the timing and magnitude of an earthquake is a fundamental goal of geoscientists. In a laboratory setting, we show we can predict "labquakes" by applying new developments in machine learning (ML), which exploits computer programs that expand and revise themselves based on new data. What we did in this project is to use seismic signals to predict the timing of laboratory earthquakes. Here we show that by listening to the acoustic signal emitted by a laboratory fault, machine learning can predict the time remaining before it fails with great accuracy. We use ML to identify acoustic signal —much like a squeaky door— that predict when a quake will occur. The experiment closely mimics Earth timing, so the same approach may work in predicting timing, but not size, of an earthquake.

About the Data:
The goal of this project is to use seismic signals to predict the timing of laboratory earthquakes. The data comes from a well-known experimental setup used to study earthquake physics. The laboratory apparatus uses steel blocks to closely mimic the physical forces at work in a real

earthquake, and also records the seismic signals and sounds that are emitted. The seismic signals come from grain fracture, rotation, and displacement, or brittle failure of adhesive grain contact junctions within the laboratory fault gouge. The acoustic_data input signal is used to predict the time remaining before the next laboratory earthquake (time_to_failure).

The primary source of our data is Kaggle:

https://www.kaggle.com/c/LANL-Earthquake-Prediction/data

Training Data:

The training data is a single sequence of signal and comes from one experiment alone. In total we have 600 million records with two variables the acoustic_data and time_to_failure.

- acoustic_data - the seismic signal
- time_to_failure - the time (in seconds) until the next laboratory earthquake

Sample training data Training Data:

acoustic_data	time_to_failure
12	1.469099983
6	1.469099982
8	1.469099981
5	1.46909998
8	1.469099979
8	1.469099978
9	1.469099977
7	1.469099976
-5	1.469099974

Before applying any machine learning models, it is important to perform exploratory data analysis and visualization so that we can understand the dataset, remove outliers and clean the dataset.
In the below table we describe the Acoustic data variable, we get the mean, standard deviation, minimum and maximum value.

```
count    6.29145480e+08
mean     4.51946757e+00
std      1.07357072e+01
min     -5.51500000e+03
25%      2.00000000e+00
50%      5.00000000e+00
75%      7.00000000e+00
max      5.44400000e+03
Name:    acoustic_data, dtype: float64
```

The time to failure, is given in seconds with a max value of 16 seconds and minimum value close to zero (1e-5)

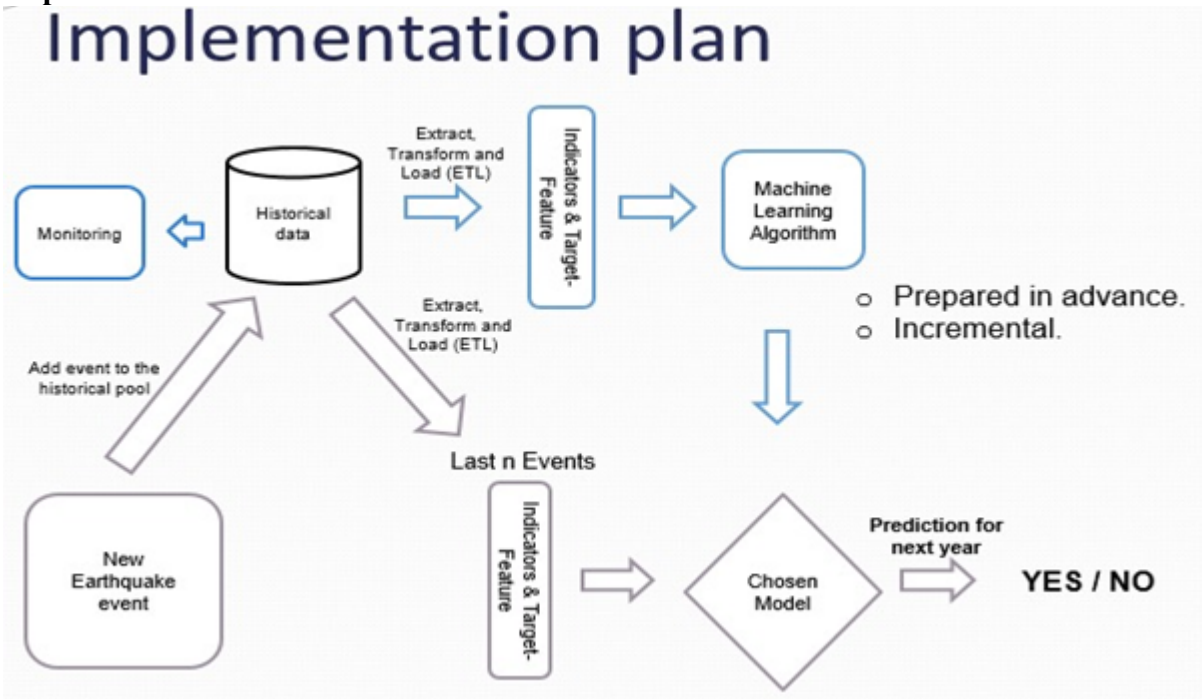
Test Data:

- The test data consists of several different sequences, called segments, that may correspond to different experiments.
- For each test data segment with its corresponding seg_id we will be predicting it's single time until the lab earthquake takes place
- seg_id - the test segment ids for which predictions should be made (one prediction per segment)

ETL Introduction

ETL is an abbreviation of Extract, Transform and Load. It is an important part of today's business intelligence (BI) processes and systems. It is the IT process from which data from disparate sources can be put in one place to programmatically analyze and discover business insights. An ETL tool extracts the data from different RDBMS source systems then transforms the data like applying calculations, concatenations, etc. and then load the data into the Data Warehouse system.

Implementation Plan



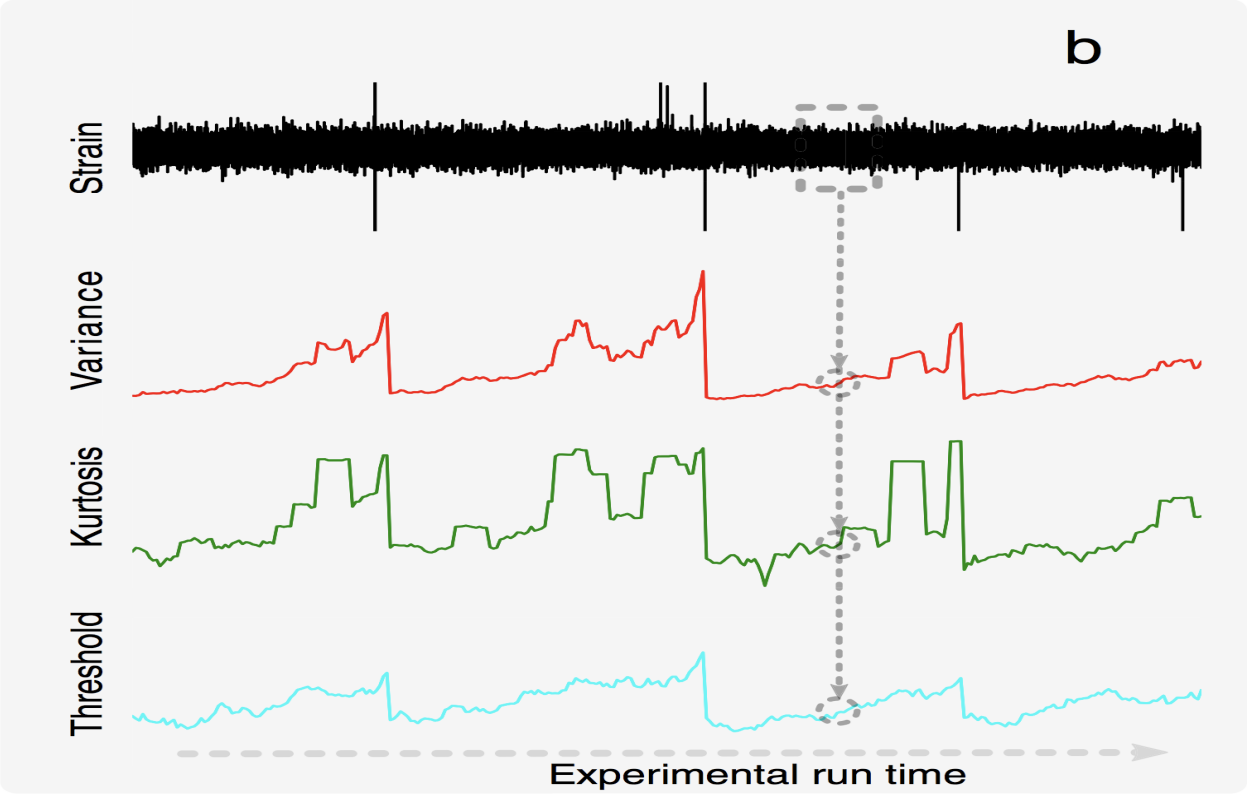
Transformation

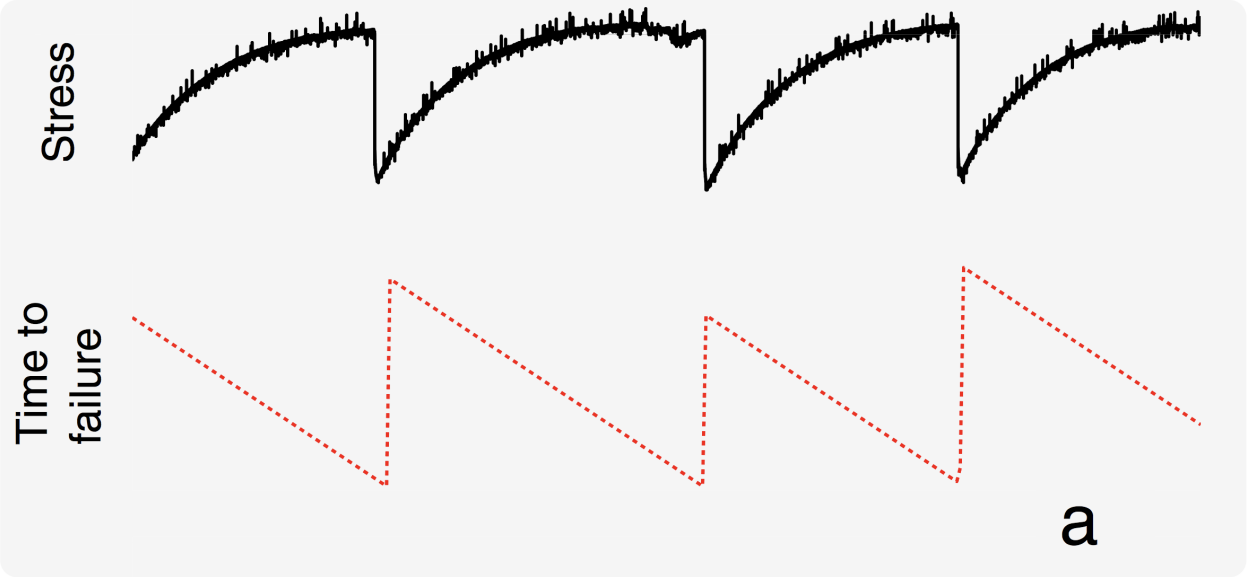
After the extraction step, we have around 600 million rows worth of data. Now our primary task is to clean, map and transform the data. We also have to remove the missing values and null values.

In our case, we are dealing a lot with the acoustic signals. We are analyzing these acoustic signals to predict the earthquakes. In each time window of the acoustic signal (also known as strain in terms of earthquake forces), we compute a set of approximately 100 potentially relevant statistical

features (e.g., mean, variance, kurtosis, and autocorrelation). We use these statistical features to observe patterns, i.e. how does the mean or variance curve look like of the acoustic signal when there is about to be a time of failure. In order to understand why we need to transform the acoustic signal (strain) and stress measures, let us understand what role they play in an earthquake.

The data comes from a well-known experimental setup used to study earthquake physics. In total, we have 600 million records with two variables the acoustic_data and time_to_failure. The acoustic_data input signal is used to predict the time remaining before the next laboratory earthquake (time_to_failure). In the ETL step, we firstly exacted the data from Kaggle and secondary sources from external contributor while created a changing table to track data changes and check timestamps. Based on the researches that we did on seismology, we also computed a group of potentially relevant statistical features, such as mean, variance, kurtosis, Amax, Amin, STD, autocorrelation and etc. After that, we loaded this data into our data warehouse.

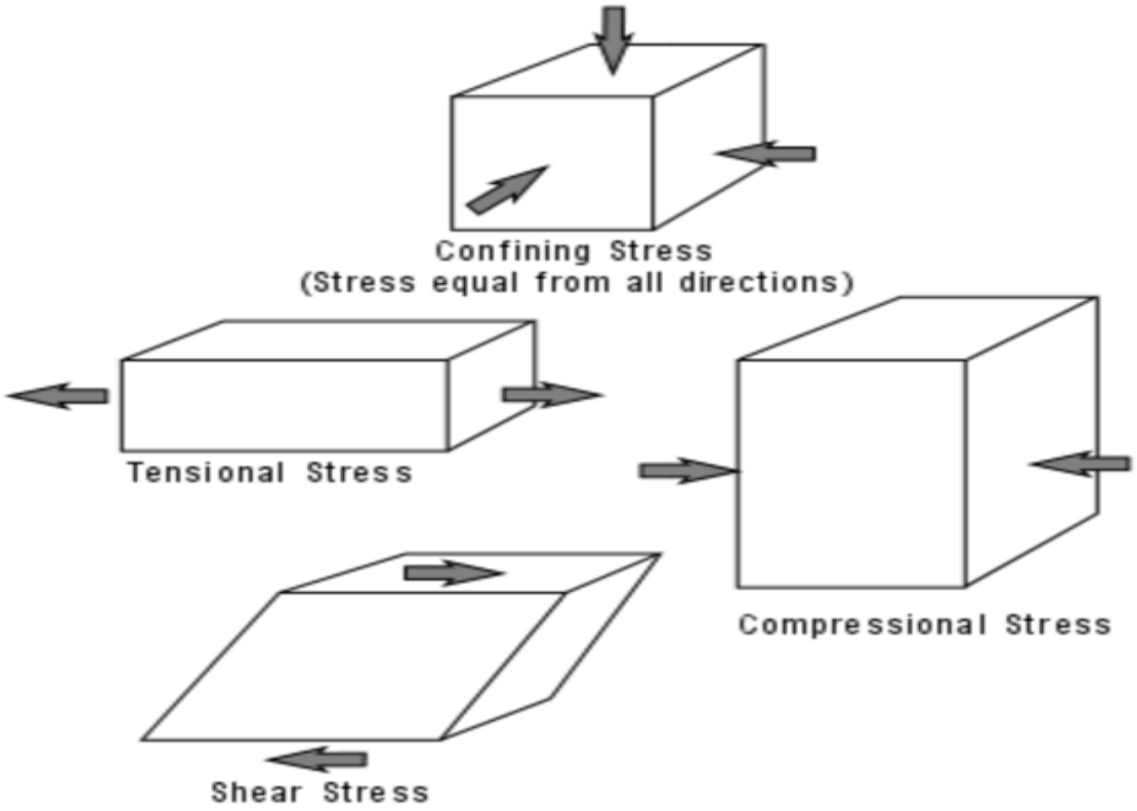


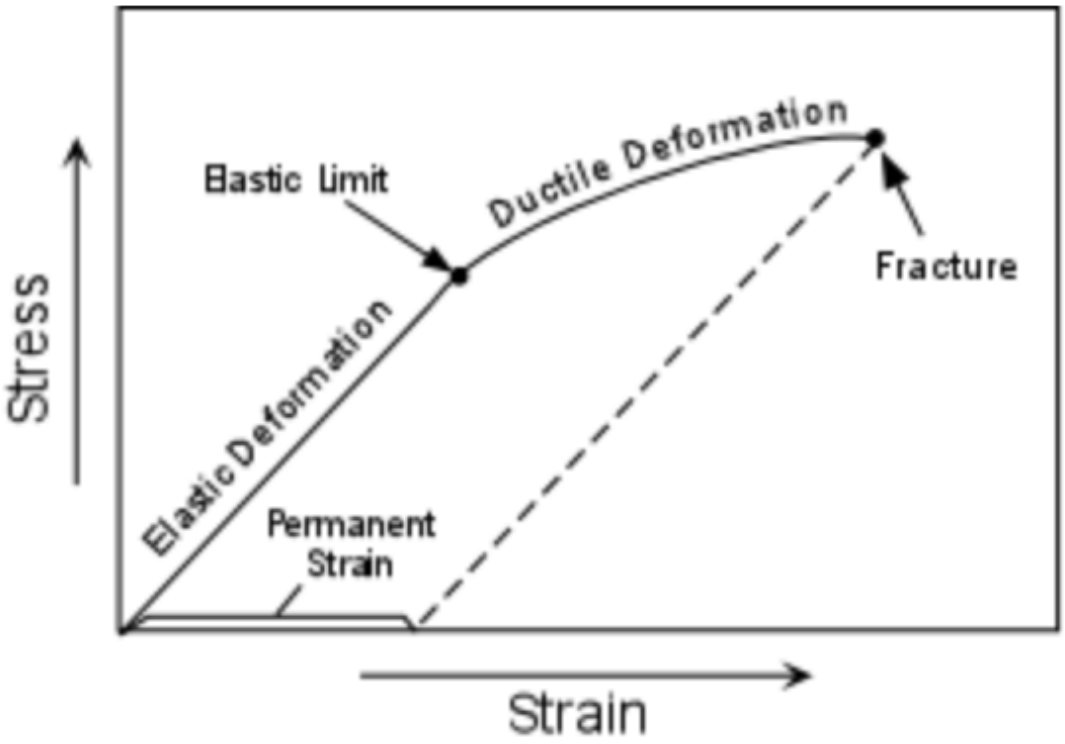


Causes of Earthquakes
Within the Earth, rocks are constantly subjected to forces that tend to bend, twist, or fracture them. When rocks bend, twist or fracture they are said to deform. The strain is a change in shape, size, or volume. The forces that cause deformation are referred to as stresses. To understand the causes of earthquakes we must first explore stress and strain.

Stress and Strain
Recall that stress is a force applied over an area. Uniform stress is where the forces act equally from all directions. The pressure is uniform stress and is referred and is also called confining stress or hydrostatic stress. If stress is not equal from all directions, then the stress is differential stress. There are three types of stress -
1> Tensional stress (or extensional stress), which stretches rock
2> Compressional stress, which squeezes rock and
3> Shear stress, which results in slippage and translation.

When a rock is subjected to increasing stress it changes its shape, size or volume. Such a change in shape, size or volume is referred to as a strain. When stress is applied to rock, the rock passes through 3 successive stages of deformation. There are three types of strain -
1>Elastic Deformation -- wherein the strain is reversible,
2> Ductile Deformation -- wherein the strain is irreversible and
3> Fracture -- irreversible strain wherein the material breaks.

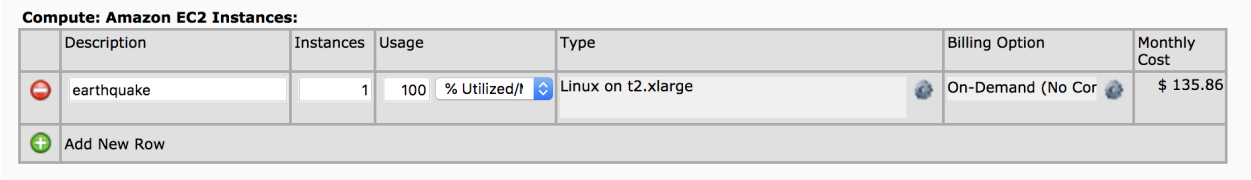

Confining Stress
(Stress equal from all directions)
Tensional Stress
Compressional Stress
Shear Stress


Stress
Elastic Limit
Ductile Deformation
Fracture
Elastic Deformation
Permanent
Strain
Strain

Loading

Since our dataset is humungous, it was impossible for us to run any queries or perform manipulation on the data without using external sources. In our case we harnessed the power of EC2 instance on AWS and created a jupyter notebook there to work on our data and then ran machine learning models.

Our dataset was 9GB which made it difficult for us to choose a lower configuration EC2 instance of AWS. In our case we used t2.xlarge instance, providing us 4 vCPU's computational power and 16GB Memory. As shown in the image below was our estimated monthly cost, which we saved upon by following the standards mentioned below to save on AWS monthly cost.



Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
⊖	earthquake	1	100 % Utilized/I	Linux on t2.xlarge	On-Demand (No Cor	$ 135.86
⊕	Add New Row					

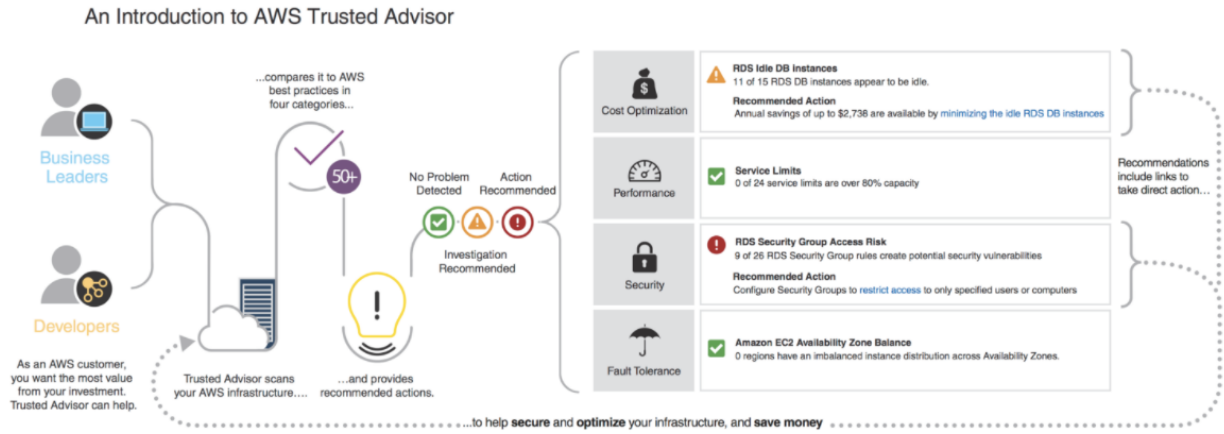
The advantage of using t2 is that it is an on-demand instance and we can increase and decrease the configuration very easily depending upon the use. With On-Demand instances, you pay for compute capacity by per hour or per second depending on which instances you run. No longer-term commitments or upfront payments are needed. You can increase or decrease your compute capacity depending on the demands of your application and only pay the specified per hourly rates for the instance you use.

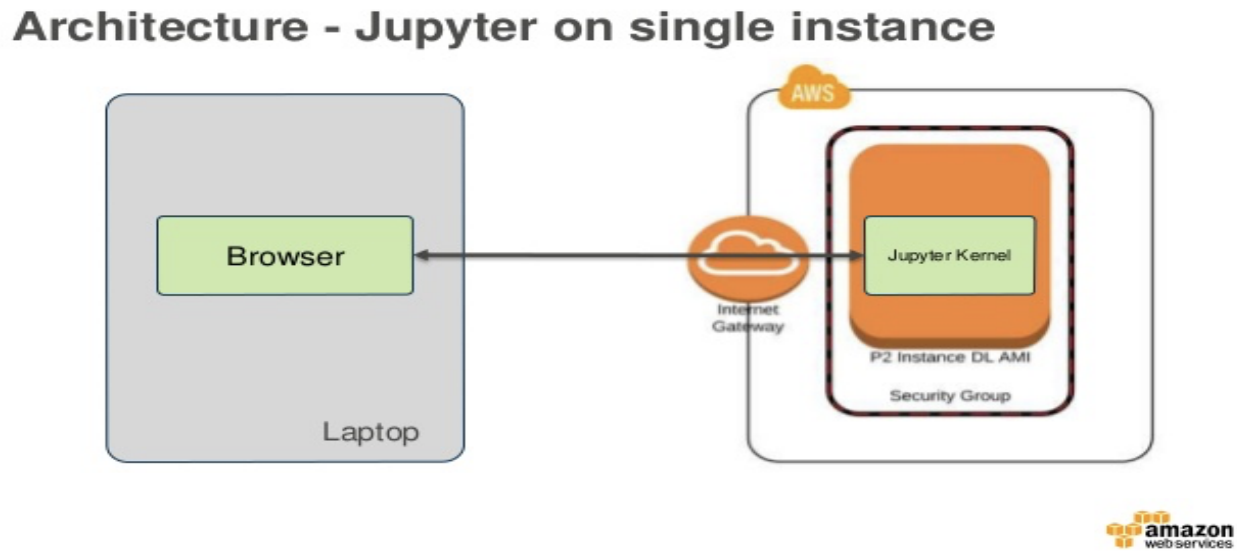
On-Demand instances are recommended for:

- Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on Amazon EC2 for the first time

Steps we followed to save cost of our EC2 instances:

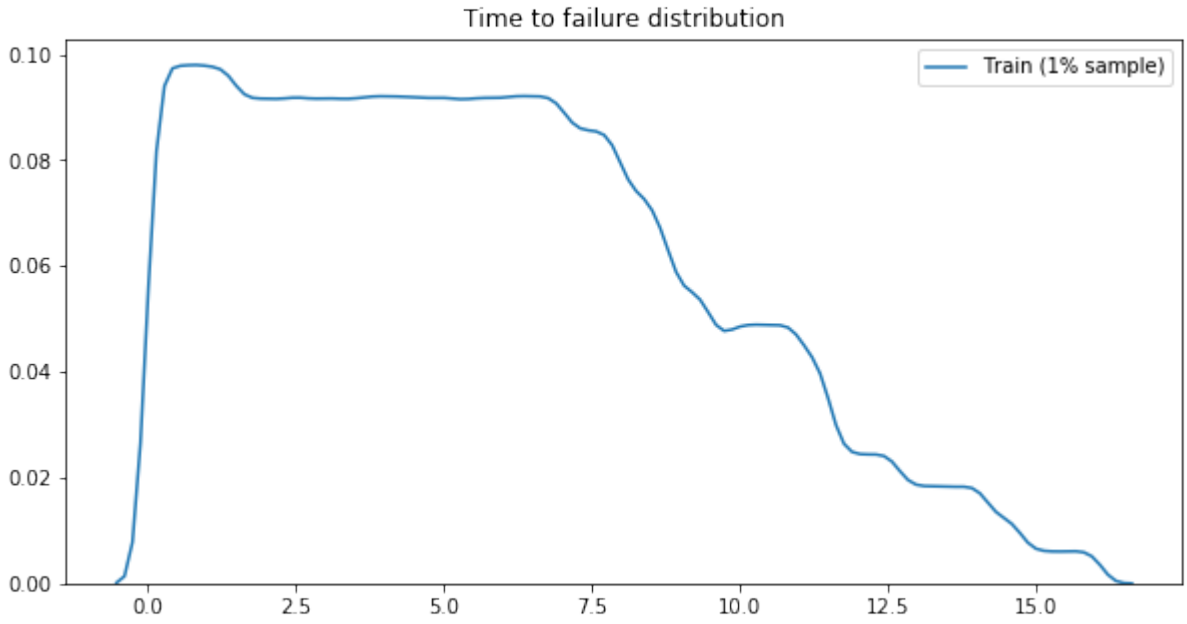
1. Shutdown unused instances
2. Select the right instance size for your workload
3. Select the appropriate S3 storage class
4. Use Cloudwatch and Trusted Advisor to monitor costs
5. Use Auto Scaling to align your resources with demand
6. Consolidated Billing provides cost savings
7. Take advantage of Reserved and Spot Instances





Data Visualization

For all the below plots we will be using a 1% random sample (~6M rows):
First, we will plot a graph to determine the data distribution of acoustic signal and time to failure.



In the above graph the acoustic signal are represented by red color and time to failure in green. There are 16 earthquakes in the training data. We can see that signal peaks usually happen right before earthquakes. This is not a perfect rule though: there are some peaks far from earthquakes (e.g. between the 14º and 15º). Another interesting thing to check is the time between these high levels of seismic signal and the earthquakes.

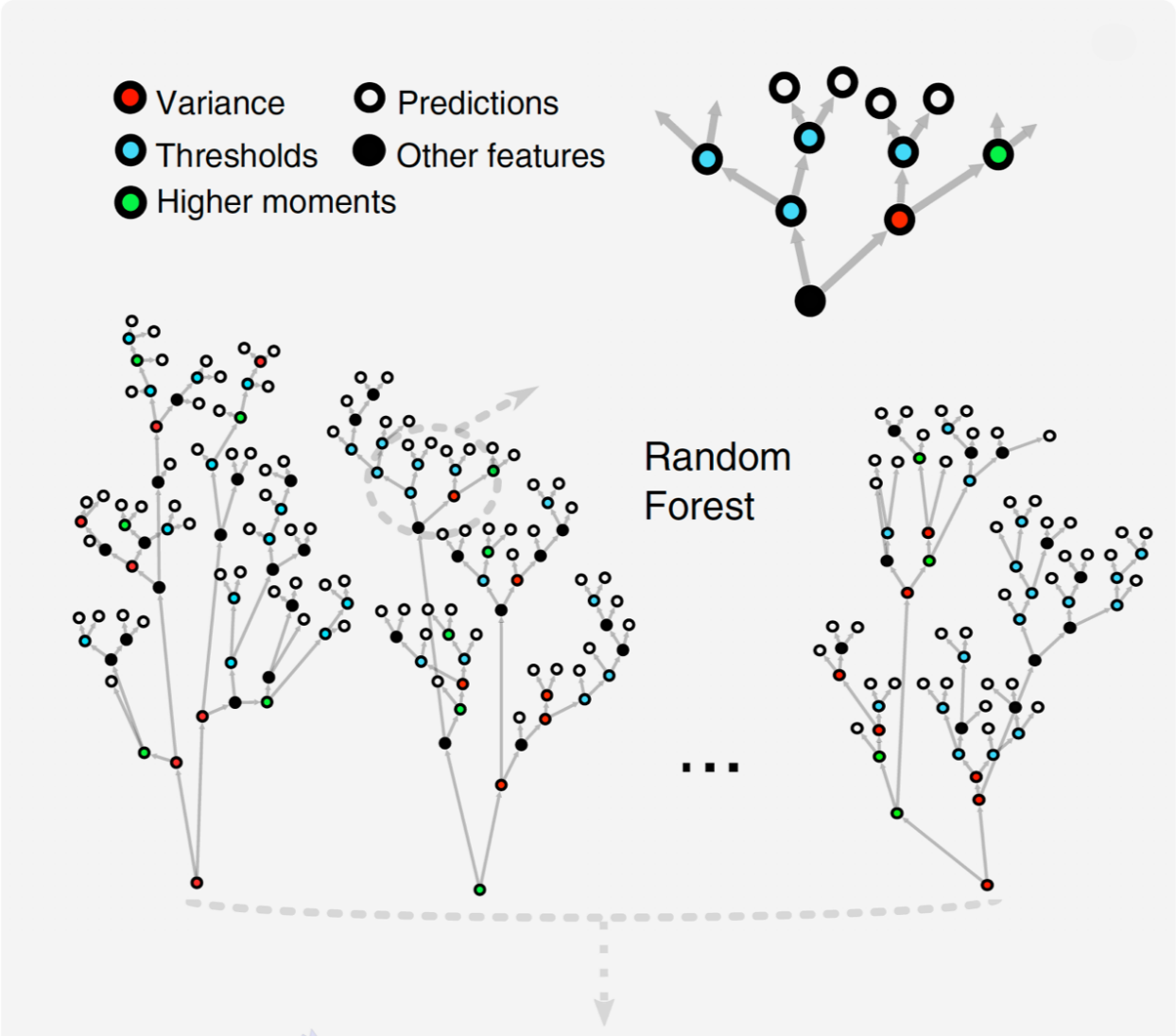
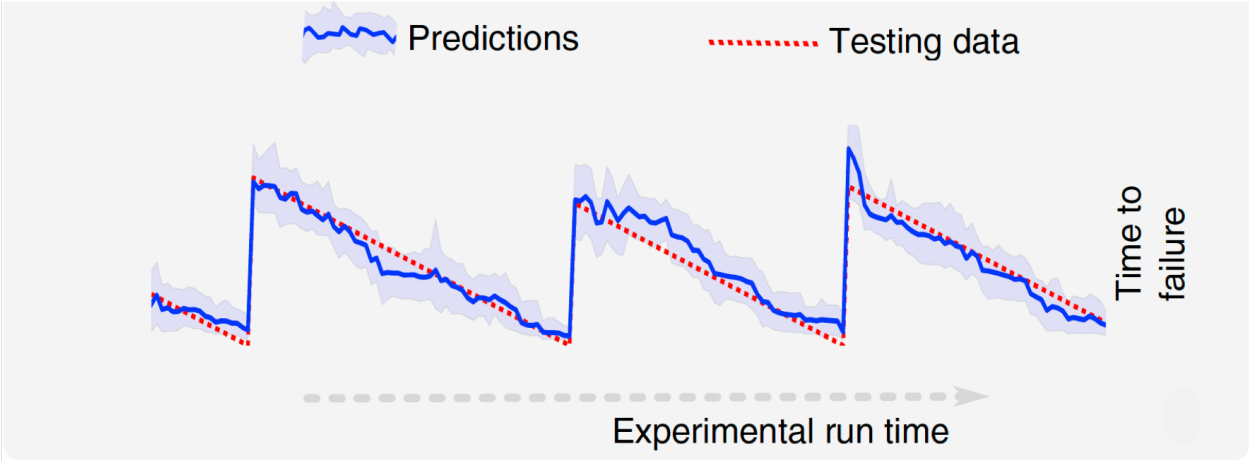


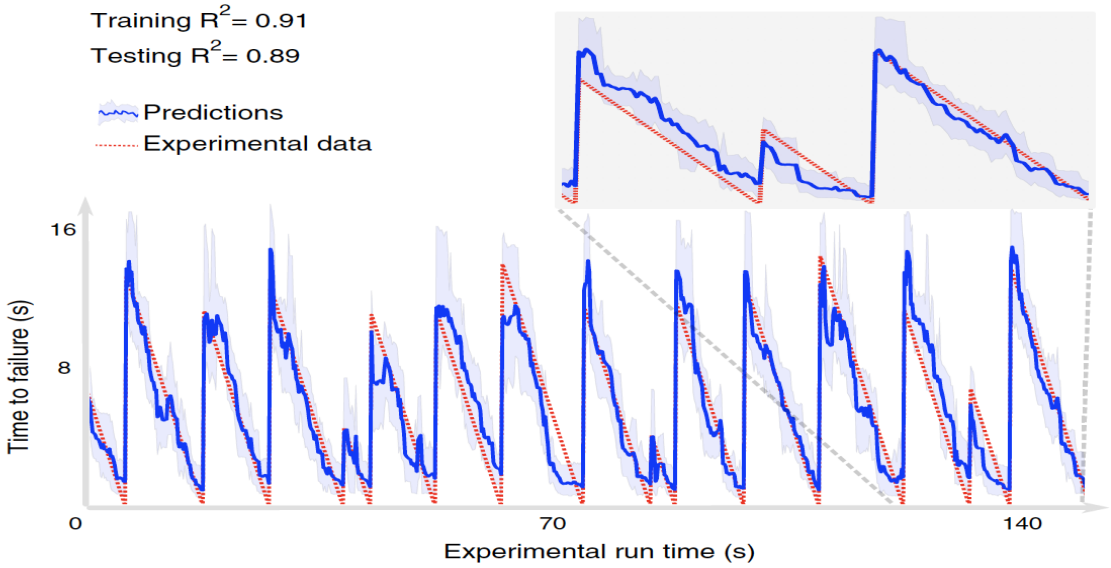
Figure. Random Forest (RF) approach for predicting time remaining before failure.

Our goal is to predict the time remaining before the next failure using only local, moving time windows of the AE data. We apply a machine learning technique, the random forest (RF), to the continuous acoustic time series data recorded from the fault. The RF model is an average over a set of decision trees. Each decision tree predicts the time remaining before the next failure using a sequence of decisions based on statistical features derived from the time windows shows the laboratory shear stress exhibiting multiple failure events during an experiment.



We wish to predict the time remaining before the next failure derived from the shear stress drops using the acoustic emission (dynamic strain) data. The RF model predicts the time remaining before the next failure by averaging the predictions of 1,000 decision trees for each time window. Each tree makes its prediction (white leaf node), following a series of decisions (colored nodes) based on features of the acoustic signal during the current window. The RF prediction (blue line) on data it has never seen (testing data) with 90% confidence intervals (blue shaded region). The predictions agree remarkably well with the actual remaining times before failure (red curve). We emphasize that the testing data are entirely independent of the training data and were not used to construct the model.

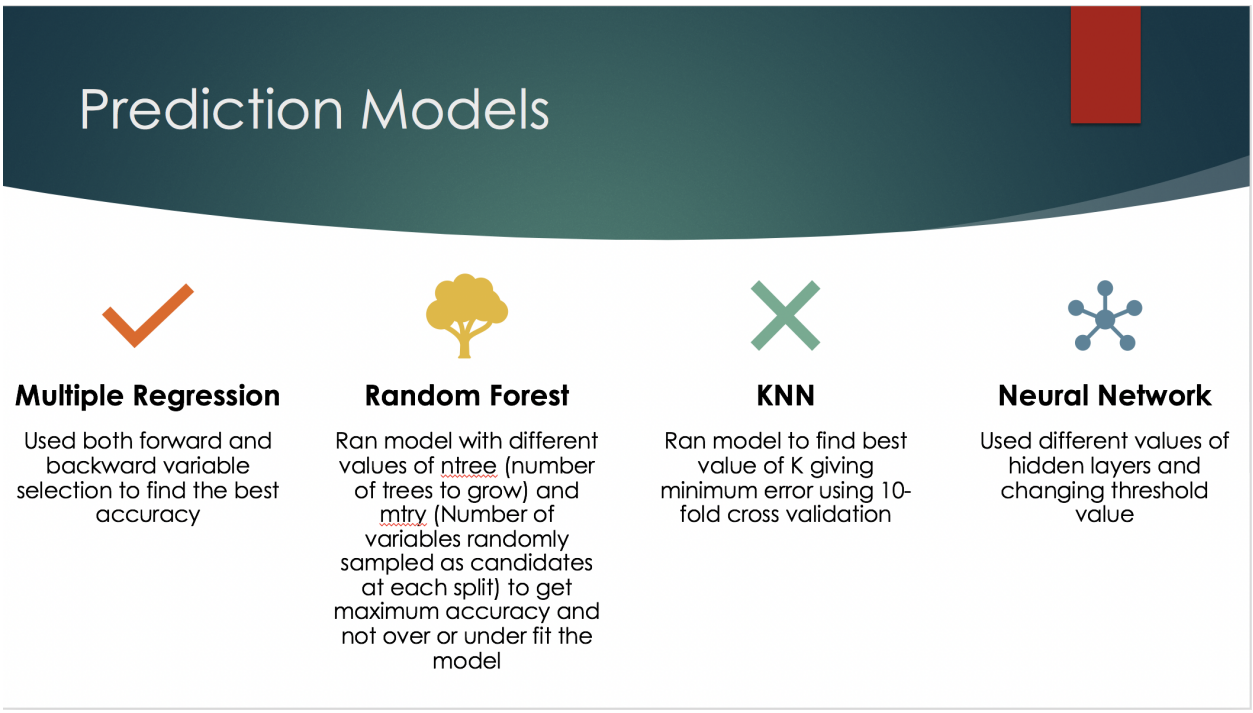
When making a prediction (blue curve): each prediction uses only the information within one single time window of the acoustic signal. Thus, by listening to the acoustic signal currently emitted by the system, we predict the time remaining before it fails—a "now" prediction based on the instantaneous physical characteristics of the system that does not make use of its history. We quantify the accuracy of our model using R2, the coefficient of determination. The time to failure predictions from the RF model are highly accurate, with an R2 value of 0.89.



From each time window, we compute a set of approximately 100 potentially relevant statistical features (e.g., mean, variance, kurtosis, and autocorrelation). We find that statistics quantifying the signal amplitude distribution (e.g., its variance and higher-order moments) are highly effective at forecasting failure. The variance, which characterizes overall signal amplitude fluctuation, is the strongest single feature early in time. As the system nears failure, other outlier statistics such as the kurtosis and thresholds become predictive as well. These outlier statistics are responding to the impulsive precursor AE typically observed as a material approaches failure, including those under shear conditions in the laboratory and in Earth. These signals are due to
small, observable shear failures within the gouge immediately preceding the laboratory earthquake.

Machine learning Models:

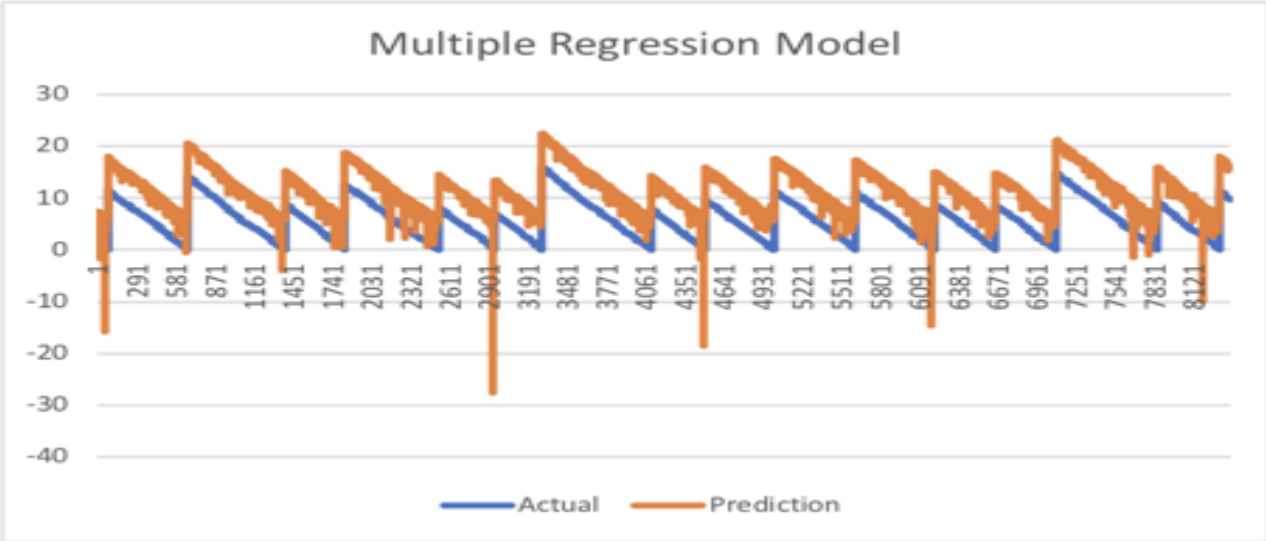
We applied recent advances in machine learning to data using R Studio, Jupyter notebook at EC2 instance (AWS). We choose NN-Artificial neural network model, Multiple Regression, KNN, Random Forest to conduct analysis. We ran model to find best value of K giving minimum error using 10-fold cross validation. We ran Multiple regression, which is an extension of simple linear regression, to predict the value of a variable based on the value of the other variables. We also run KNN model, which is a non-parametric, lazy learning algorithm, to use the database in which the data points are separated into several classes to predict the classification of a new sample point. The RF model is an average over a set of decision trees. Each decision tree predicts the time remaining before the next failure using a sequence of decisions based on statistical features derived from the time windows. Among those models, KNN has the best prediction accuracy.



Multiple Regression:

```
# partition
mydata <- read.csv("newTrain2.csv")
set.seed(1)
train.index <- sample(c(1:dim(mydata)[1]), dim(mydata)[1]*0.8)
train <- mydata[train.index, ]
valid <- mydata[-train.index, ]

#train model
regmodel <- lm (time_to_failure ~.,data = train)
summary(regmodel)

library(forecast)
regpred <- predict(regmodel, valid)
accuracy(regpred, valid$time_to_failure)

#Variable Selection
stepPred <- step(regmodel, direction = "both")
summary(stepPred)  # Which variables did it drop?
stepPred.pred <- predict(stepPred, valid)
accuracy(stepPred.pred, valid$time_to_failure)

linearModel = data.frame(Actual = valid$time_to_failure, Prediction = regpred)
library(tidyverse)
write_csv(linearModel, path = "linear.csv")
```

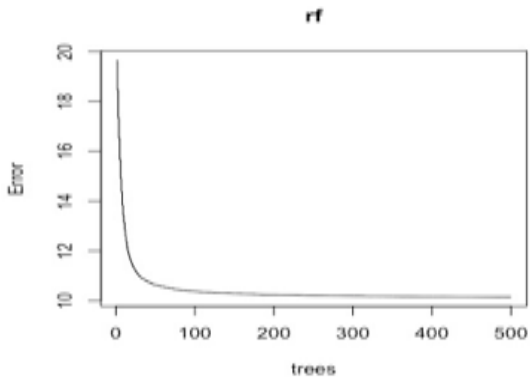
```
Call:
lm(formula = time_to_failure ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-20.141  -2.781  -0.329   2.312  31.879

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.996e+00  3.133e-01  28.715  < 2e-16 ***
Variance     7.132e-04  1.743e-05  40.918  < 2e-16 ***
Kurtosis    -8.079e-03  1.699e-03  -4.756 1.98e-06 ***
Mean        -3.813e-01  1.077e-01  -3.540 0.000401 ***
Median       2.555e-02  6.385e-02   0.400 0.689022
Percentile  -1.294e-02  7.062e-03  -1.832 0.066903 .
Amax         1.247e-02  9.612e-04  12.977  < 2e-16 ***
Amin        -4.948e-03  9.427e-04  -5.248 1.54e-07 ***
STD         -4.265e-01  3.098e-02 -13.769  < 2e-16 ***
Skew        -2.159e-02  7.339e-02  -0.294 0.768594
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.554 on 33545 degrees of freedom
Multiple R-squared:  0.07015,   Adjusted R-squared:  0.0699
F-statistic: 281.2 on 9 and 33545 DF,  p-value: < 2.2e-16
```

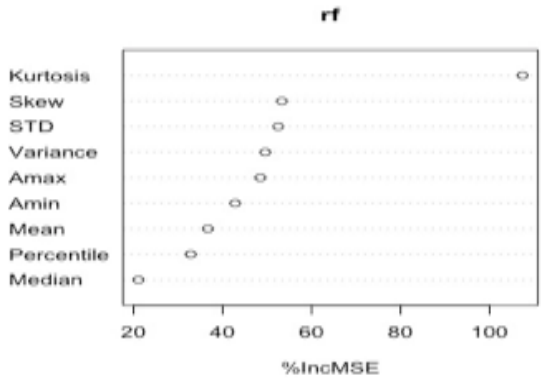


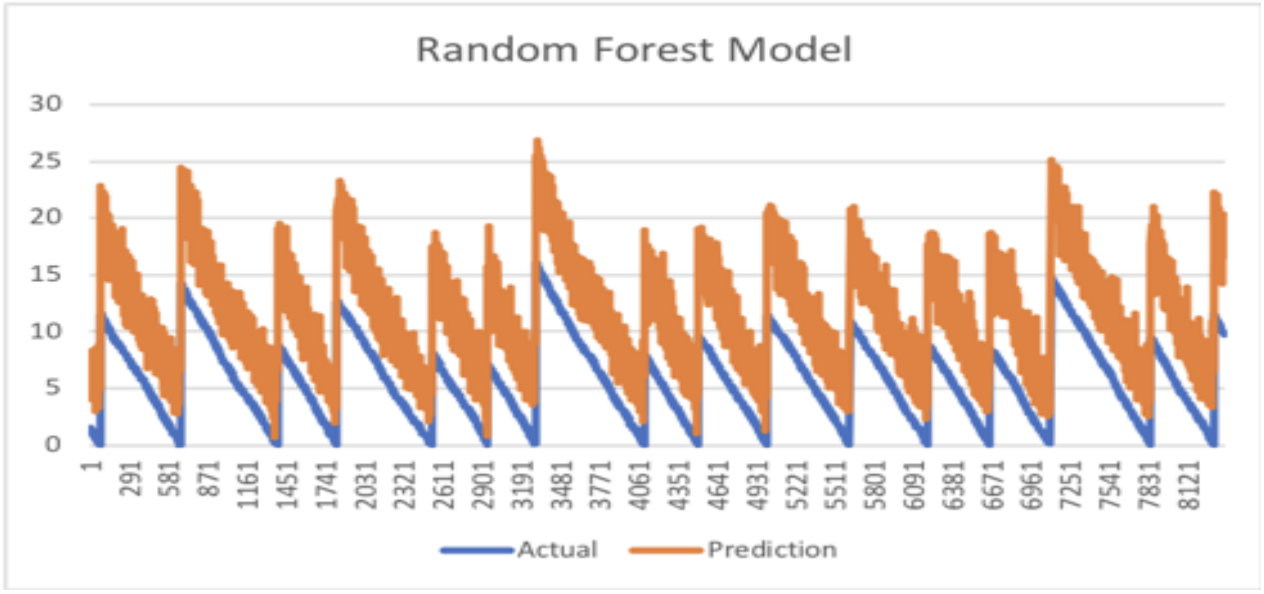
Random Forest:

```
# partition
mydata <- read.csv("newTrain2.csv")
set.seed(1)
train.index <- sample(c(1:dim(mydata)[1]), dim(mydata)[1]*0.8)
train <- mydata[train.index, ]
valid <- mydata[-train.index, ]
library(randomForest)
rf <- randomForest(time_to_failure ~ ., data = train, ntree = 500,
                   mtry = 4, nodesize = 5, importance = TRUE)

library(devtools)
library(reprtree)
tree <- getTree(rf, k=1, labelVar=TRUE)
realtree <- reprtree:::as.tree(tree, rf)

summary(rf)
summary(tree)
head(rf$votes,10)

## Plot forest by prediction errors
plot(rf, type = "simple")

## variable importance plot
varImpPlot(rf, type = 1)

## confusion matrix
rf.pred <- predict(rf, valid)
library(forecast)
```

```
accuracy(rf.pred, valid$time_to_failure)
randomForestModel = data.frame(Actual = valid$time_to_failure, Prediction = rf.pred)
library(tidyverse)
write_csv(randomForestModel, path = "rf.csv")
```





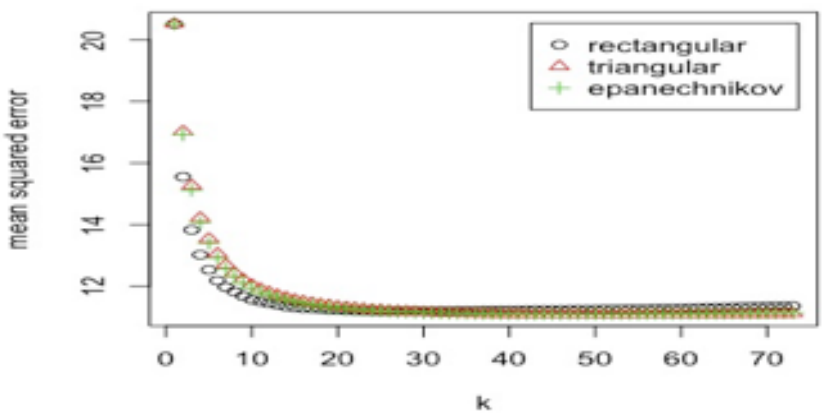


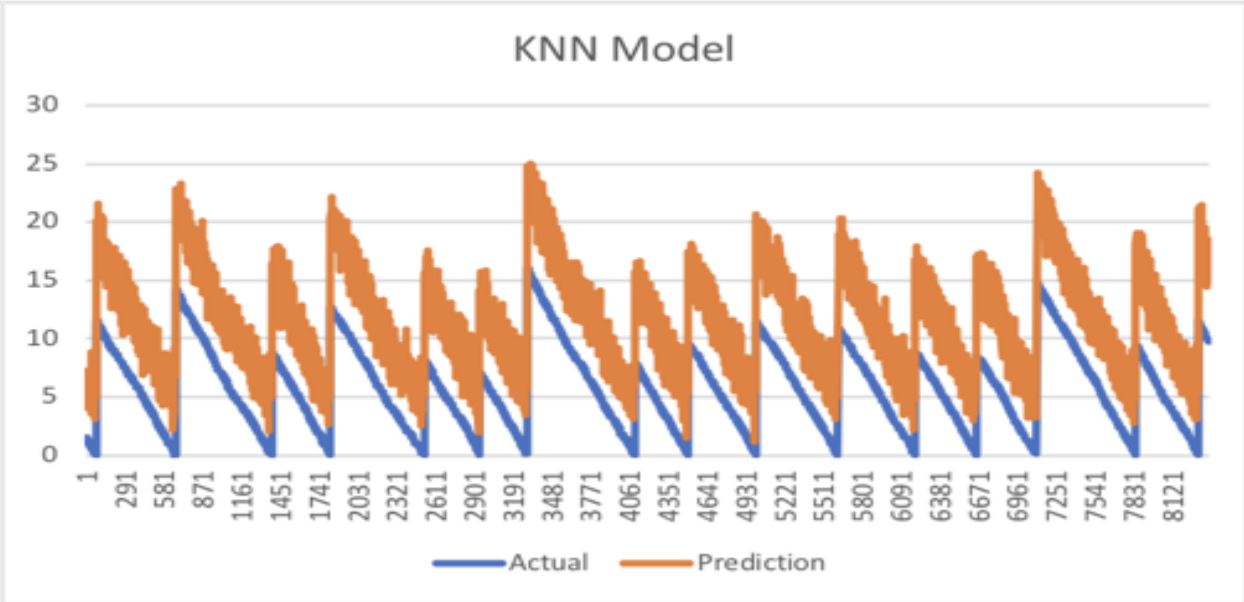
KNN Model:

```
# partition
mydata <- read.csv("newTrain2.csv")
set.seed(1)
train.index <- sample(c(1:dim(mydata)[1]), dim(mydata)[1]*0.8)
train <- mydata[train.index, ]
valid <- mydata[-train.index, ]

library(DMwR)
set.seed(502)
```

```
grid1 <- expand.grid(.k = seq(2, 150, by = 10))
control <- trainControl(method = "cv")
knn.train <- train(time_to_failure ~ ., data = train,
                   method = "knn",
                   trControl = control,
                   tuneGrid = grid1)
knn.train

knn.pred <- predict(knn.train, newdata = valid)

library(kknn)
set.seed(123)
kknn.train <- train.kknn(time_to_failure ~ ., data = train, kmax = 80,
                         distance = 10,
                         kernel = c("rectangular", "triangular", "epanechnikov"))
plot(kknn.train)

kknn.train
kNN.pred <- kNN(time_to_failure ~ .,train,valid,norm=TRUE,k=73)

kknn.pred <- predict(kknn.train, newdata = valid)

KnnModel = data.frame(Actual = valid$time_to_failure, Prediction = kknn.pred)
accuracy(kknn.pred, valid$time_to_failure)
library(tidyverse)
write_csv(KnnModel, path = "knn.csv")
```

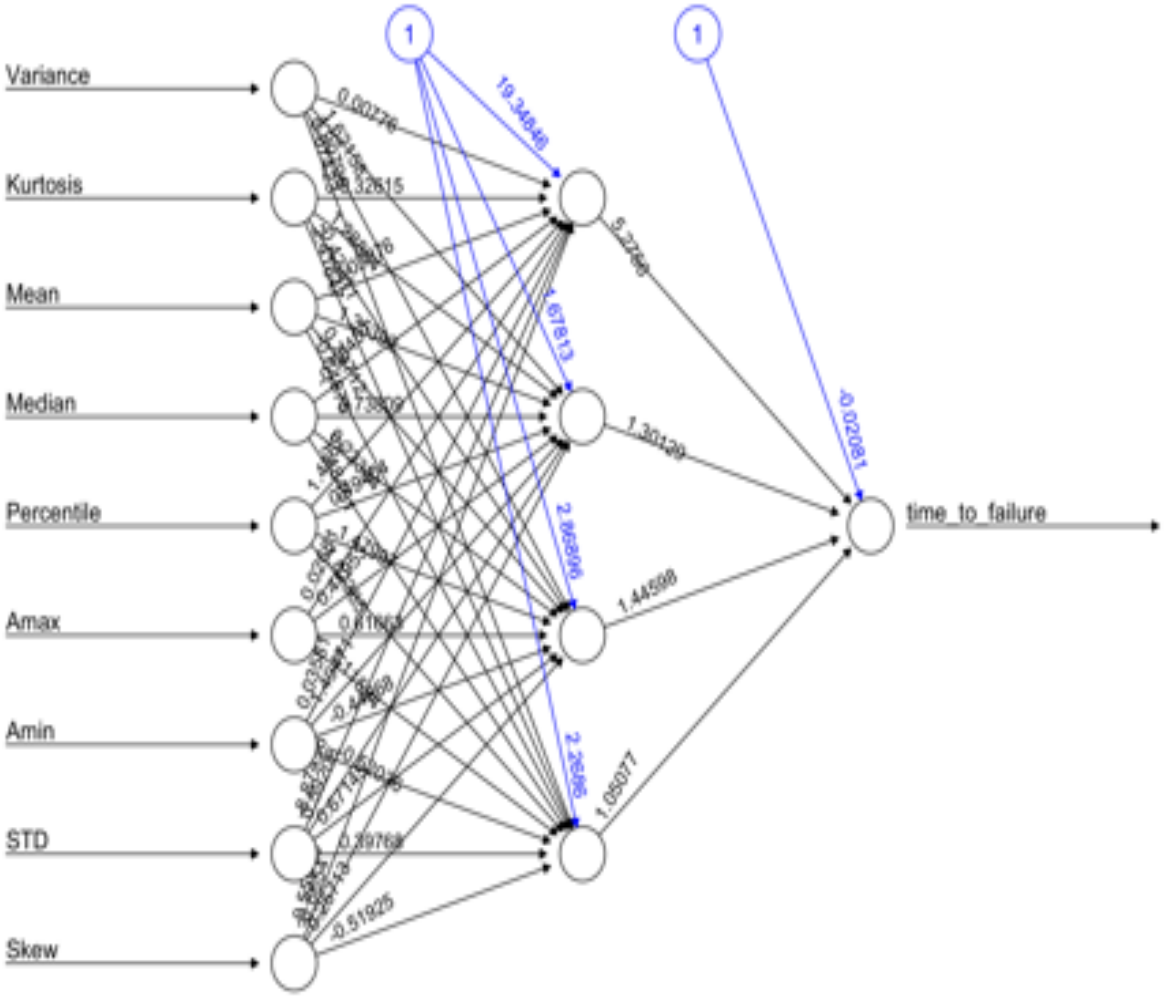
```
Type of response variable: continuous
minimal mean absolute error: 2.673926
Minimal mean squared error: 11.08715
Best kernel: triangular
Best k: 53
>
```



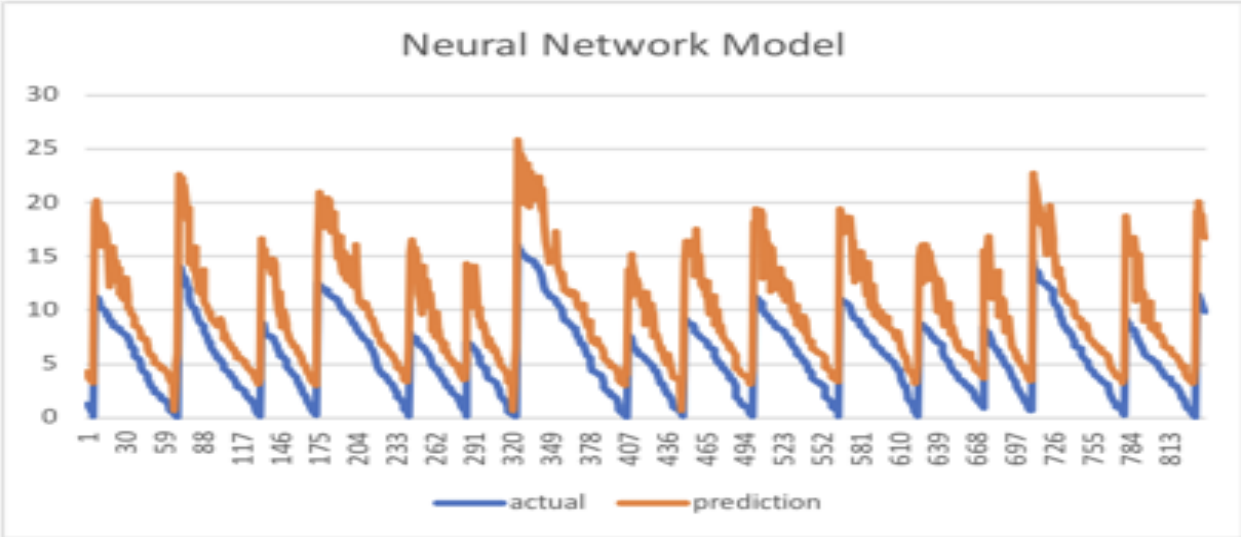


Neural Network Model:

```
# partition
mydata <- read.csv("newTrain2.csv")
set.seed(1)
train.index <- sample(c(1:dim(mydata)[1]), dim(mydata)[1]*0.8)
train <- mydata[train.index, ]
valid <- mydata[-train.index, ]
library(neuralnet)
library(nnet)
library(caret)
nn <- neuralnet(time_to_failure ~ ., data = train, hidden = 4, linear.output = TRUE, threshold = 0.1)
nn$result.matrix
plot(nn)

nn.results <- neuralnet::compute(nn, valid[, -c(8)])

results <- data.frame(actual = valid$time_to_failure, prediction = nn.results$net.result)

accuracy(results$prediction, valid$time_to_failure)

library(tidyverse)
write_csv(results, path = "nn.csv")
```



Error: 169953.154286 Steps: 94004



Comparison of Error values of all models:

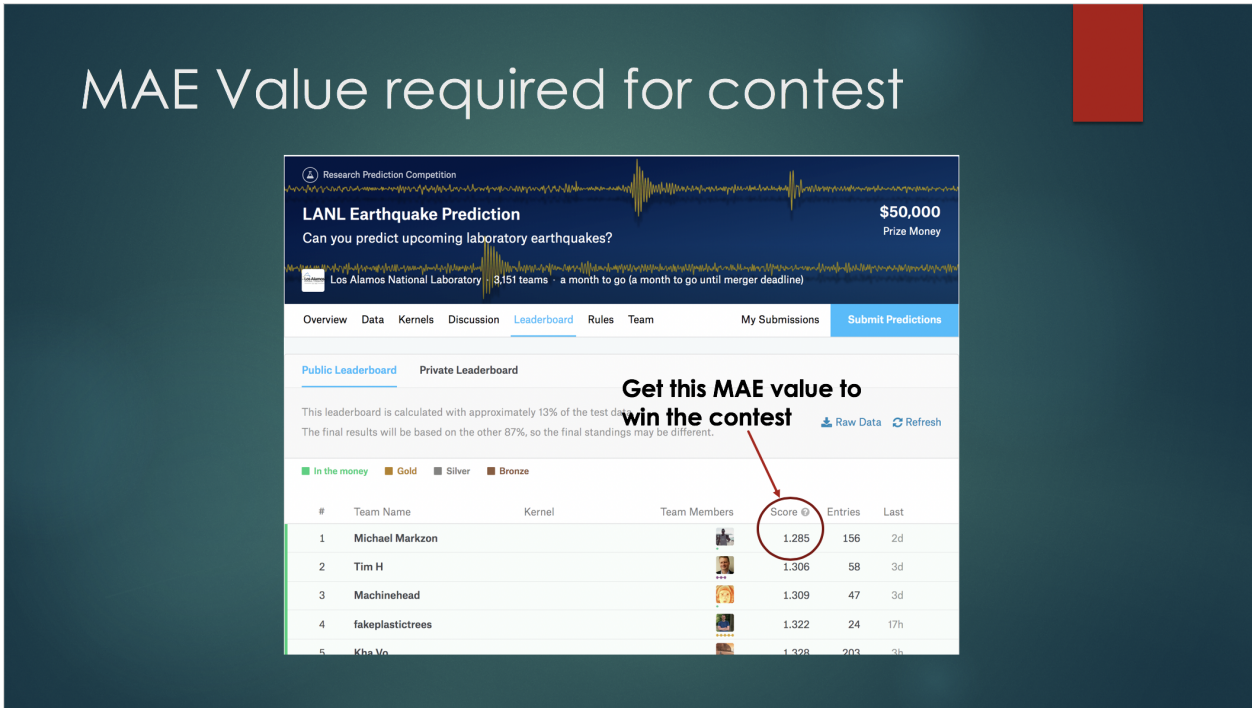
```
> accuracy(regpred, valid$time_to_failure)
                    ME     RMSE      MAE       MPE     MAPE
Test set -0.02572933 3.49162 2.857612 -348.9805 385.2563
>
```

```
> accuracy(rf.pred, valid$time_to_failure)
                    ME     RMSE      MAE       MPE     MAPE
Test set -0.04710793 3.130435 2.486949 -414.3376 440.607
>
```

```
> accuracy(kknn.pred, valid$time_to_failure)
                    ME     RMSE      MAE       MPE     MAPE
Test set -0.009574229 3.263729 2.625079 -389.4562 416.4801
```

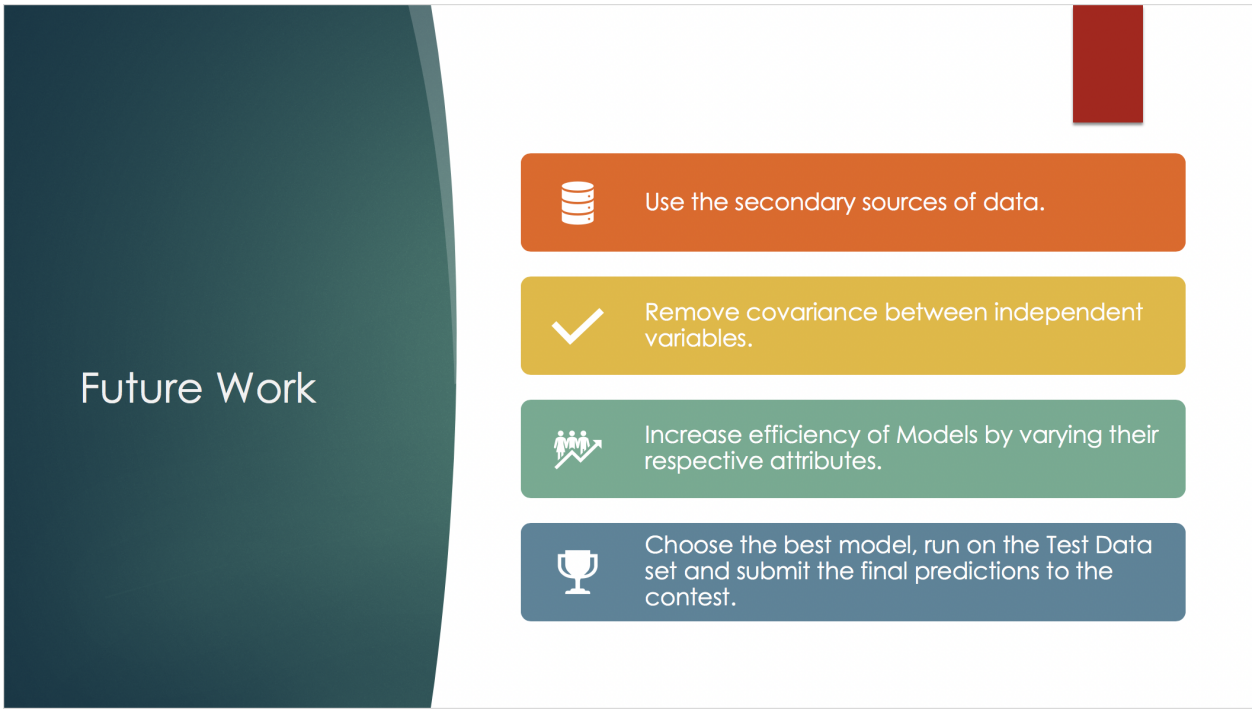
```
> accuracy(results$prediction, valid$time_to_failure)
                   ME     RMSE      MAE       MPE     MAPE
Test set 1.626368 3.177401 2.437937 -148.2669 199.5417
```

Our Target:



Future Work:

Since the deadline to submit this project on Kaggle is 3rd June, our work does not end with end of this academic coursework. Below are the details for our future work.



Conclusion:

To summarize, we show that ML applied to this experiment provides accurate failure forecasts based on the instantaneous analysis of the acoustic signal at any time in the slip cycle and reveals a signal previously unidentified. These results should suffice to encourage ML analysis of seismic signals in Earth. In particular, ML-based approaches mitigate human bias by automatically searching for patterns in a large space of potentially relevant variables. Our current approach is to progressively scale from the laboratory to the Earth by applying this approach to Earth problems that most resemble the laboratory system. An interesting analogy to the laboratory may be faults that exhibit small repeating earthquakes. For instance, fault patches located Repeaters at these fault patches may be emitting chattering in analogy to the laboratory. If so, can this signal be recorded by borehole and surface instruments? We are currently studying this problem. Whether ML approaches applied to continuous seismic or other geophysical data succeed in providing information on timing of earthquakes (not to mention the challenge of predicting earthquake magnitude), this approach may reveal unidentified signals associated with undiscovered fault physics. Furthermore, this method may be useful for failure prediction in a broad spectrum of industrial and natural materials. Technology is at a confluence of dramatic advances in instrumentation, machine learning, the ability to handle massive data sets and faster computers. Thus, the stage has been set for potentially marked advances in earthquake science.

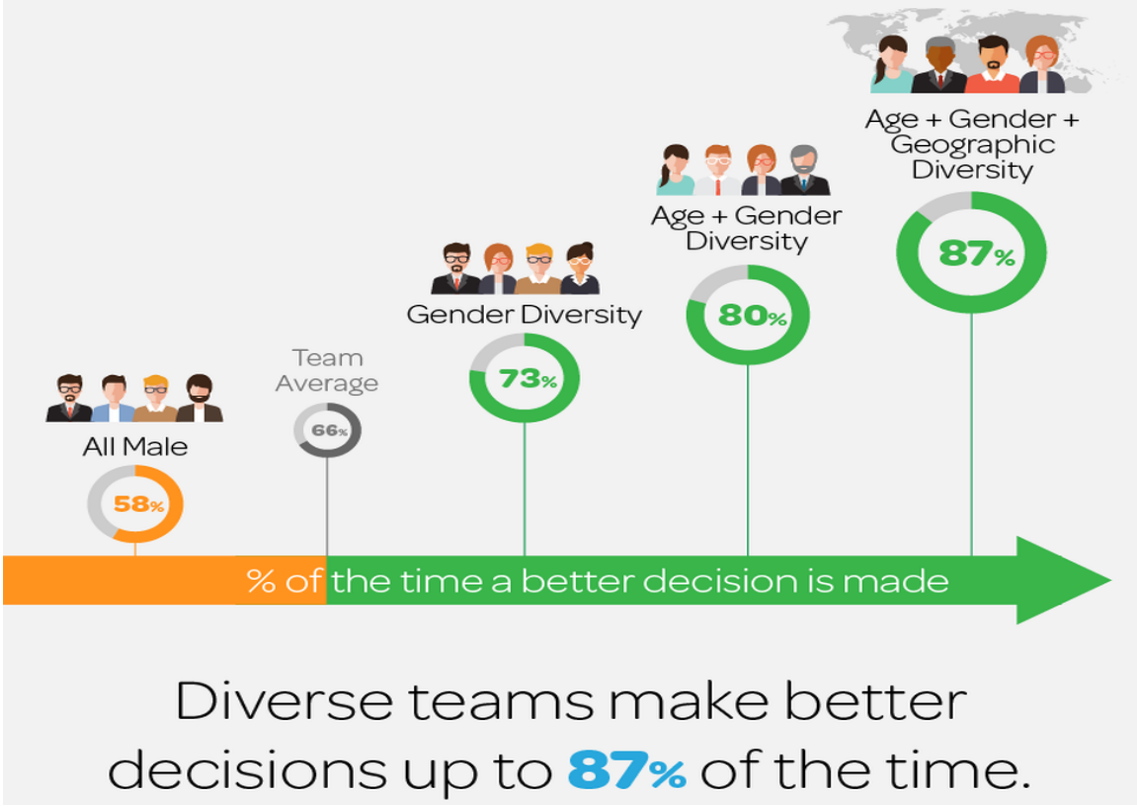
Key Takeaways:

The project was a significant part not only for the coursework but also from the learning prospective. It was a great opportunity to for me to be able to take up a big project as a part of coursework and take part in the Kaggle challenge as well. I would like to take this opportunity to thank our **Professor Vijay** for streamlining the coursework for our projects. The slides and guidance given to us by our professor was insightful and the way he related our coursework to his personal experience in the industry and his guidelines how each step will come into picture when we will be working in a corporate environment was motivational. I would also like to thank our external collaborator **Swasti Saxena** - Geotechnical Engineering, Ph.D. student at University of Nevada, Reno. She is currently working on deterministic ground motion simulations using large scale wavefield propagation for her project, 'Toward development of site-specific vertical earthquake ground motion for resiliency of nuclear power plants'.

Few key points which everyone should consider while taking a big project are mentioned below:

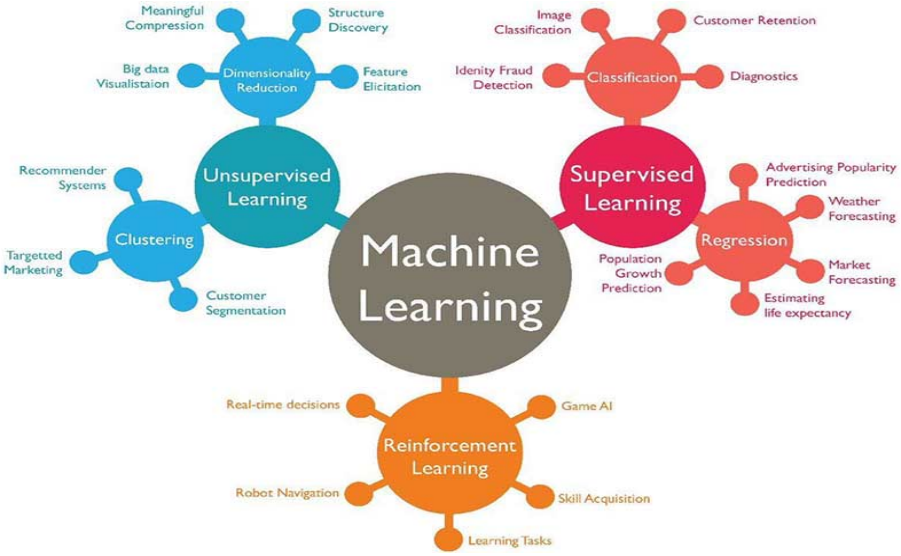
1. **Selection of group members:**

   Selection of correct team members plays a significant role in any project. Different minds with different ideas bring out the best results in a project. It is very important to have disagreements in a group as this is how best work comes up into the picture. As in our group we all come from different backgrounds of work we had done previously, which helped a lot during the progress of our work as everyone's input belonging to their respective fields brought the best in our work.



2. **Selection of type of Big Data project:**

When one is asked to choose a project, it is important that you consider all options and choose the type of project you are most confident you can do it along with your team. While choosing a project you have 3 options, firstly, to challenge yourself and take up a project type you have never done before secondly, to challenge yourself in a with a more difficult project in your field, thirdly, to take the basic project which you know you will be able to complete. We choose the second option and challenged ourselves.



3. **Selection of Project:**

As mentioned earlier we challenged ourselves with the kind of projects we had done before. We did that by taking up the Earthquake project. Having no prior knowledge of earthquake data was a challenge and to overcome this we found an external collaborator who helped us throughout our journey. One also needs to be smart while selecting your project e.g. We not only challenged ourselves with an unknown field, but our project is a 50,000$ competition on Kaggle. Since, we knew are going to put in our best into the upcoming project then why not participate in a competition and try to win it.

4. **Analyzing Data (Training and Testing):**

Before one start working on the model selection and the planning of how to start, the journey needs to begin with the analysis of the data. It is very important to understand that what training data has been given, what are the input variables (Independent variables) and output variables (Dependent Variables), finding the correlation between the independent variables, cleaning data if necessary.

5. **Making ETL Plan and bringing it in action:**

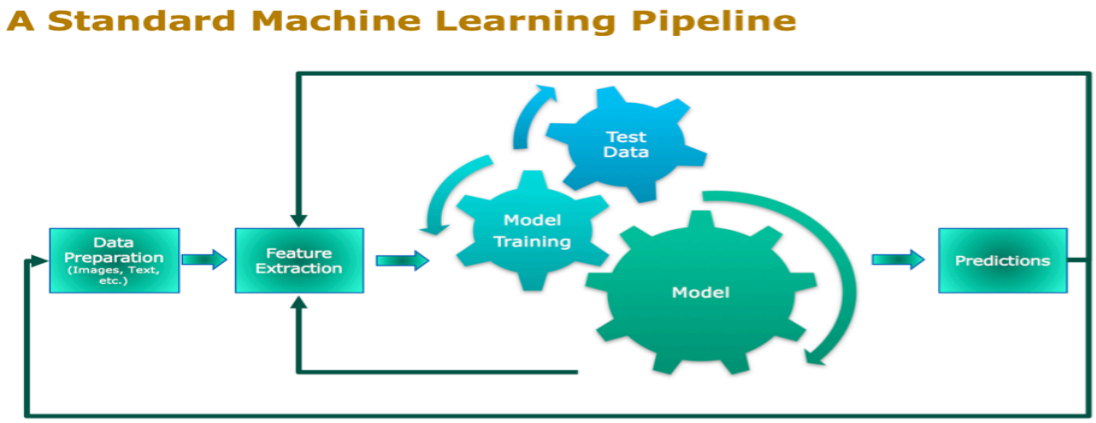
One you are familiar with the data, make a plan how do you need to perform ETL on your data i.e. how will you extract, transform and load your data. Since, it is a big data project you will obviously be extracting your data from some source, but you also need to ponder on the fact that will you need secondary sources to get more data. Once you know how you will be extracting your data, you need to ponder do you need to transform your data, clean it up, merge data, create new variables using existing data, etc. As it is a big data project your data records are going to be so high that you will need plan where will you be loading your data for future work, which type of server will be able to take the load of computation you are about to perform.

6. **Data Visualization:**

Data visualization plays a very significant role in Big Data projects because you have such large data and it is sometimes difficult to find pattern in humungous dataset. When you visualize, the picture of data becomes clear and you can see how your data is behaving and much more.

7. **Choosing Prediction Models:**

Every dataset is unique, and one needs to identify what kind of output they are trying to predict and what will be their input variable and which models can help in prediction with the dataset you have chosen. After working on many big data projects, I can say from my experience that you cannot guarantee that any one type of model will give you best results. As mentioned earlier, every dataset is unique and every model analyzes data differently, so there so no fixed model that will give you best result, you will have to hit and trial and that too not once but many times. **Patience is the key to good modeling**. Be patient, try different models, change their attributes and run again and again until you find best accuracy.



Challenges faced during this project work:

References:

- arxiv.org/abs/1702.05774: Machine Learning Predicts Laboratory Earthquakes

- https://www.technologyreview.com/s/603785/machine-learning-algorithm-predicts-laboratory-earthquakes/

- Thi Lu, Ngo & Rodkin, Mikhail & Viet Phuong, Tran & Thi Thu Hang, Phung & Quang, Nguyen & Hoan, Vu. (2016). Algorithm and program for earthquake prediction based on the geological, geophysical, geomorphological and seismic data. VIETNAM JOURNAL OF EARTH SCIENCES. 38. 10.15625/0866-7187/38/3/8708.

- Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. Geophysical Research Letters, 44, 9276– 9282. https://doi.org/10.1002/2017GL074677

- Earthquakes and EQ Prediction. (2015). *Earthquake Prediction with Radio Techniques,*1-17. doi:10.1002/9781118770368.ch1

- Nanjo, K. Z., Schorlemmer, D., Woessner, J., Wiemer, S., & Giardini, D. (2010). Earthquake detection capability of the Swiss Seismic Network. *Geophysical Journal International*. doi:10.1111/j.1365-246x.2010.04593.x

- Joseph, A. (2011). Earthquake Detection and Monitoring for Early Warnings of Seismogenic Tsunamis. *Tsunamis,*109-114. doi:10.1016/b978-0-12-385053-9.10007-9

- Stefánsson, R. (2011). Application of earthquake prediction to other earthquake-prone regions. *Advances in Earthquake Prediction,*207-222. doi:10.1007/978-3-540-47571-2_8