HarvardX: PH125.9x Data Science
Capstone IDV Earth Quake Prediction Project

Srinivasan Menon

June 10, 2019

Table of Contents

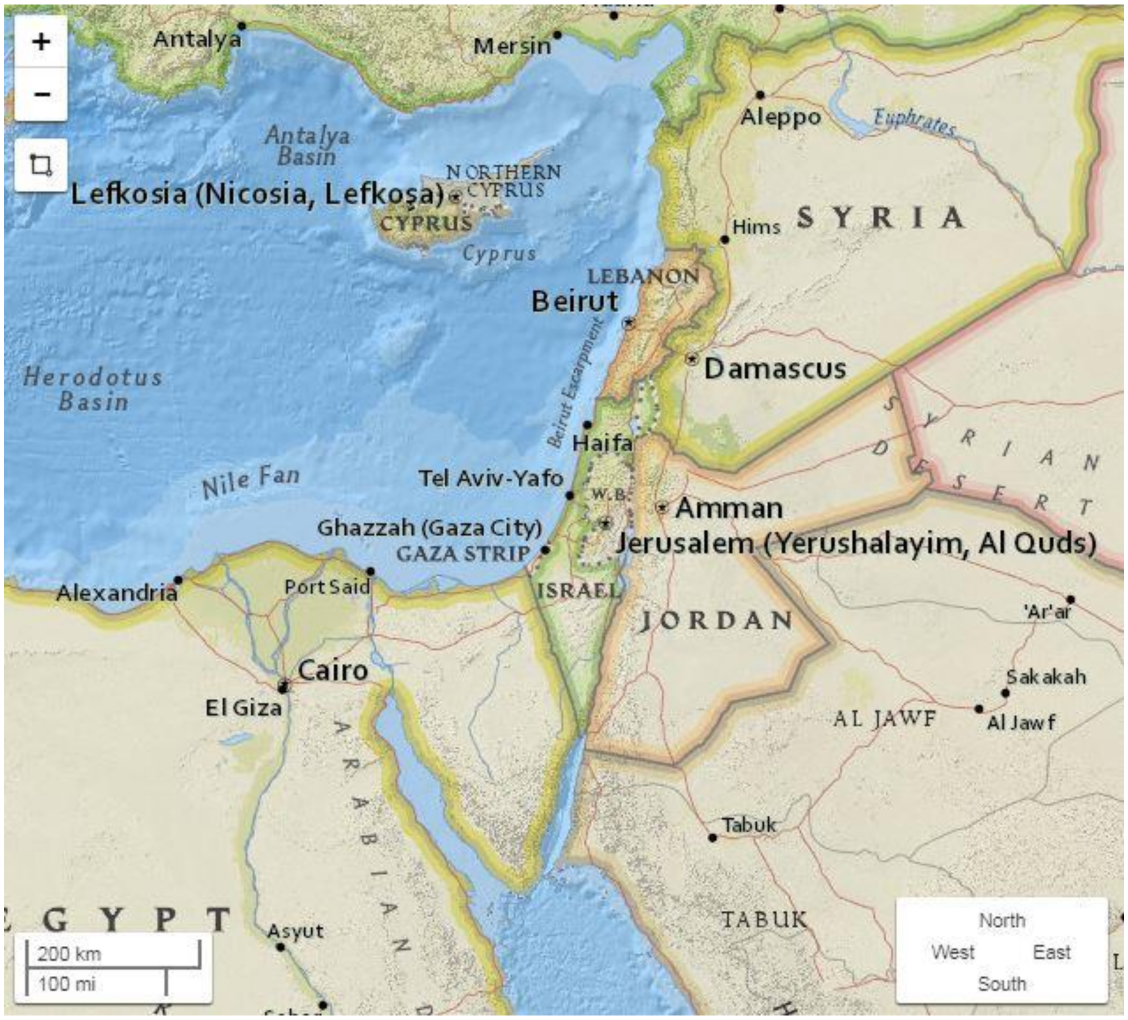
Earth Quake Prediction ML project Overview

This project is related to the HervardX: PH125.9x Data Science: Capstone course ML Earth Quake Prediction using seismic data. The present report start with a general idea of the project and by representing its objectives.

It has long been observed that the relationship between the frequency and the magnitude of seismic events in a given region follows the power law. In other words, when observing earthquake occurrences over time we expect the earthquakes of small magnitude to be much more frequent than the earthquakes of large magnitude. Following the study of California earthquakes by, this relationship is named the Gutenberg-Richter inverse power law and it is defined mathematically as follows:

$$Log_{10}N(M) = a - bM$$

The data set for this project was taken from the catalog of USGS (United States Geological Survey) website. The data contains regional (see Figure 1) recorded seismic network information such as earthquake location, magnitude, date, depth, and more (total of 23 variables and 4071 observations).



Dataset area

Characterization of the problem

The objective of this project is **to Predict the Maximum Earthquake Magnitude of next year**. This kind of prediction is based on a research article published on 2016 (Last et al). All the formulas and synthesized features are based on this article. The Prediction of this project is a binary classification task based on the median of maximum yearly magnitudes (threshold value). Possible models that were applied in this task are Decision Tree, K-Nearest Neighbor (KNN), Support vector machine (SVM), and Neural Network (NNET). The next sections are written in accordance to the CRISP-DM data science life cycle.

```
#Loading relevant packages (Installation needed if package not exist in
running computer)
```

```
library(plyr)
library(ggplot2)

## Registered S3 methods overwritten by 'ggplot2':
##   method         from 
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

library(data.table)
library(lubridate)

## 
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
## 
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following object is masked from 'package:plyr':
## 
##     here

## The following object is masked from 'package:base':
## 
##     date

library(Hmisc) #For Lag function

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## 
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':
## 
##     is.discrete, summarize

## The following objects are masked from 'package:base':
## 
##     format.pval, units

library(zoo)#For rollmeanr

## 
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

library(hydroGOF)#For MSE calculation

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

library(class)#For knn model
library(nnet)#For Neural-Network model
library(rpart)#For Decision Tree model
library(rattle)#For Decision Tree model

## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)#For Decision Tree model
library(RColorBrewer)#For Decision Tree model
library(e1071)#For SVM model

##
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':
##
##     impute

library(ROCR)#For ROC curves

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

library(RCurl)#For nnet plotting

## Loading required package: bitops
```

Data Understanding

AS mentioned above, the raw data consists of 23 variables and 4071 observations:

```
#Loading data
rawcsv<-read.csv("Israel 2.5.csv", check.names=TRUE, na.strings=c("","NA"))
str(rawcsv)
```

```
## 'data.frame':	4071 obs. of  22 variables:
##  $ time           : Factor w/ 4071 levels "1918-09-29T12:07:14.000Z",..: 1
2 3 4 5 6 7 8 9 10 ...
##  $ latitude       : num  35.1 36 31.9 32.4 36.5 ...
##  $ longitude      : num  35.6 29.5 35.6 33.5 35.9 ...
##  $ depth          : num  15 15 15 15 15 15 15 15 15 20 ...
##  $ mag            : num  6.6 6.8 6.3 5.9 5.8 5.2 5.3 5.7 5.5 6.3 ...
##  $ magType        : Factor w/ 10 levels "m","mb","md",..: 6 6 6 6 6 6 6 6
6 6 ...
##  $ nst            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ gap            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ dmin           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ rms            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ net            : Factor w/ 3 levels "iscgem","iscgemsup",..: 1 1 1 1 1
2 2 1 1 1 ...
##  $ id             : Factor w/ 4071 levels "iscgem17290377",..: 21 20 19 17
18 26 27 16 14 15 ...
##  $ updated        : Factor w/ 4049 levels "2013-11-15T03:20:29.000Z",..:
3886 3887 3887 3888 3888 3907 3907 3889 3890 3890 ...
##  $ place          : Factor w/ 170 levels "0km N of Ypsonas, Cyprus",..:
163 170 155 157 153 156 170 153 168 154 ...
##  $ type           : Factor w/ 2 levels "earthquake","explosion": 1 1 1 1 1
1 1 1 1 1 ...
##  $ horizontalError: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ depthError     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ magError       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ magNst         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ status         : Factor w/ 2 levels "automatic","reviewed": 1 1 1 1 1 1
1 1 1 1 ...
##  $ locationSource : Factor w/ 14 levels "ath","bhl","csem",..: 8 8 8 8 8 8
8 8 8 8 ...
##  $ magSource      : Factor w/ 16 levels "ath","bhl","csem",..: 10 10 10 10
10 10 10 10 10 10 ...

#Adding Column of Date derived from the time column
rawcsv$Date <- as.Date(rawcsv$time)
```

For the purposes of this project and for preserving the basic information of each earthquake event (observation), only 6 variables were extracted from the raw dataset:

- Date- Date of earthquake event (extracted from the Time variable).
- Latitude and Longitude (two seperate variables)- Location of an earthquake event.
- Depth- Depth to earthquake origin location (from surface).
- Magnitude(mag)- Richter magnitude scale of an earthquake event.

- Magnitude type(magType)- The method or algorithm used to calculate the preferred magnitude for the event.

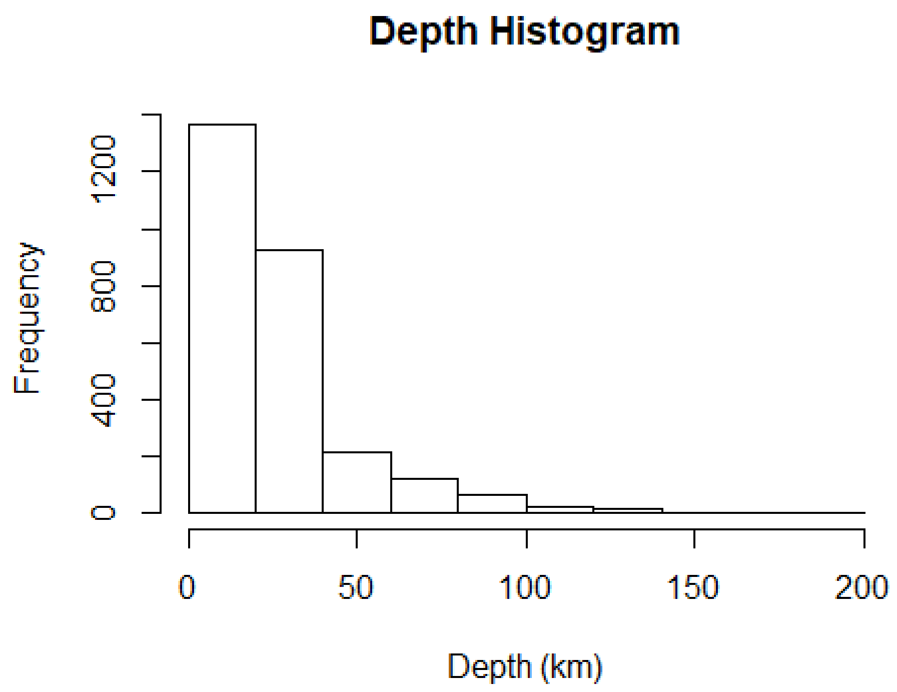
Before further investigating the data, there is a need to filter out all foreshock and aftershock events. To obtain that, all recorded events from the same date were combined by the maximum magnitude value of that day. It is important to mention that this kind of filter is definitely not enough to include only mainshocks but for this project educational purpose it is okay.

```
#removing Foreshocks & Aftershocks - (extracting only maximum values from
each date)
csv <- as.data.table(rawcsv)
csv<-csv[csv[, .I[which.max(mag)], by=Date]$V1]

#Defining a threshold value (median of all years maximums)
threshold<-median(tapply(csv$mag,year(csv$Date),max))

csv<-csv[,c("Date","latitude","longitude","depth","mag","magType")]
summary(csv)
```

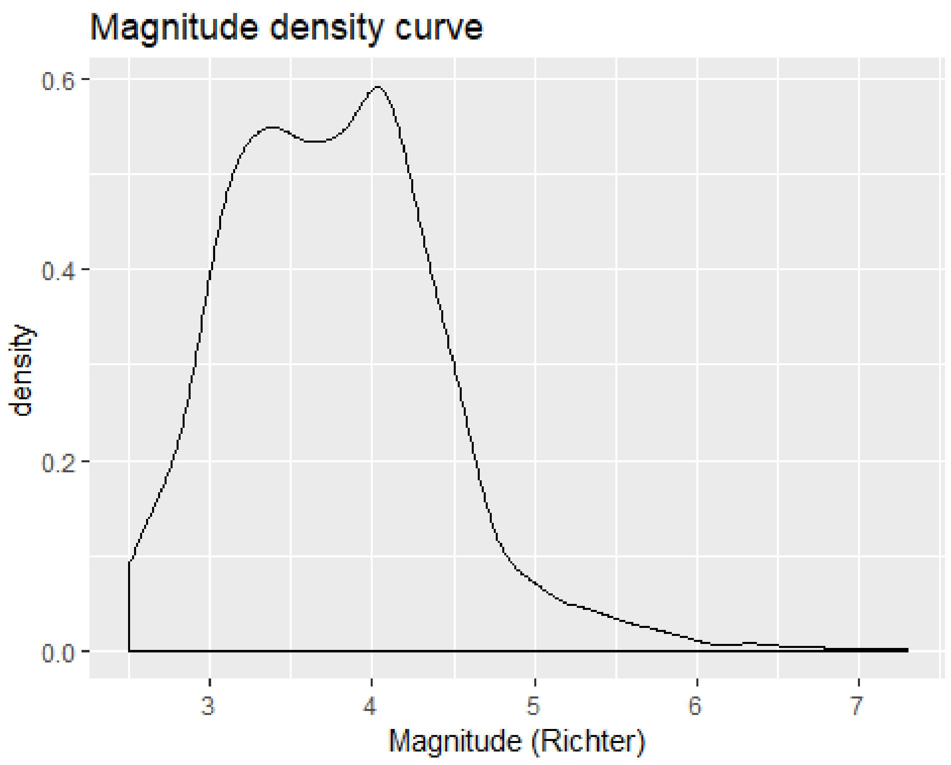
```
##       Date               latitude        longitude         depth
##  Min.   :1918-09-29   Min.   :26.55   Min.   :27.77   Min.   :  0.00
##  1st Qu.:1993-11-04   1st Qu.:34.55   1st Qu.:28.92   1st Qu.: 10.00
##  Median :2000-09-28   Median :35.48   Median :31.36   Median : 20.00
##  Mean   :1999-05-17   Mean   :34.95   Mean   :31.29   Mean   : 25.71
##  3rd Qu.:2006-03-05   3rd Qu.:36.50   3rd Qu.:33.13   3rd Qu.: 33.00
##  Max.   :2017-09-01   Max.   :37.07   Max.   :41.50   Max.   :191.60
##
##       mag           magType
##  Min.   :2.50   mb     :968
##  1st Qu.:3.30   ml     :858
##  Median :3.70   md     :758
##  Mean   :3.77   mw     : 50
##  3rd Qu.:4.20   mwc    : 36
##  Max.   :7.30   m      : 33
##                 (Other): 19
```

The information above is relating to the filtered data (total of 2722 observations). From the maximum and average values of mag it is clear that in general, earthquakes in the region are not deadly from one hand but still cause damage on the other hand. Hence, the threshold value, 5.6, calculated as the median of the yearly maximums magnitudes of the dataset, represents somehow the limit from which earthquakes may cost a heavy economic price.
Below are graphs that reveal some information regarding the distribution of the numeric variables. Also presented is the mean magnitude per year. The variation in the average magnitude in this graph is probably due to measurement improvement along the years (the early years are characterized by strong earthquakes documentation only). The year distribution graph supports this - very few observations in the early years.

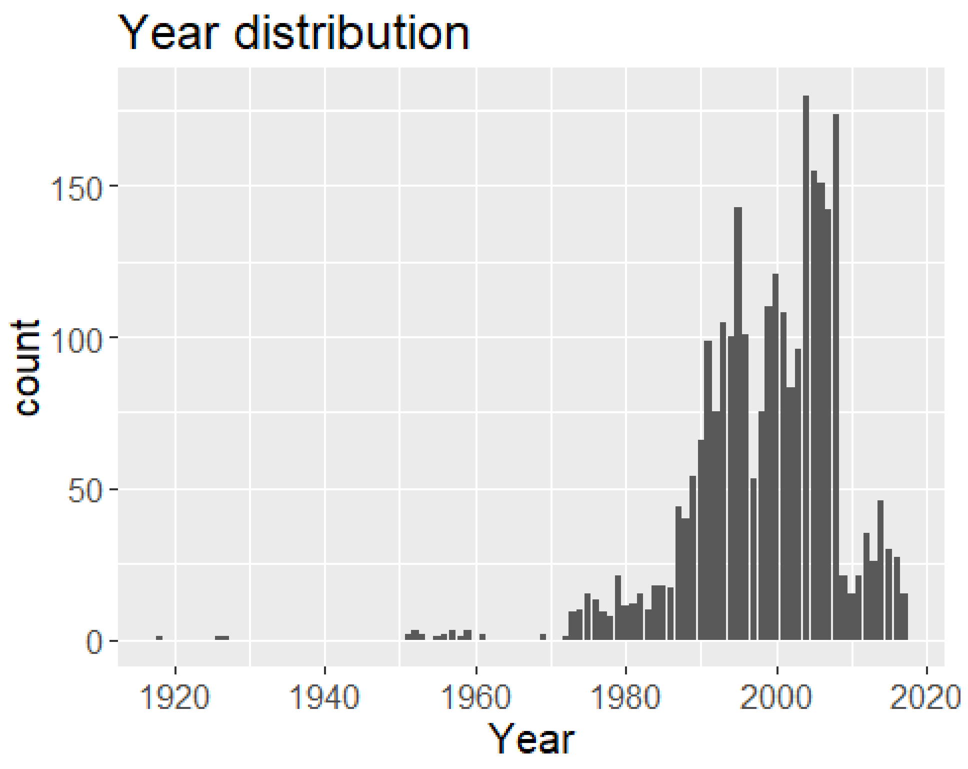
```r
hist(csv$depth, main = "Depth Histogram", xlab = "Depth (km)") #Depth
```



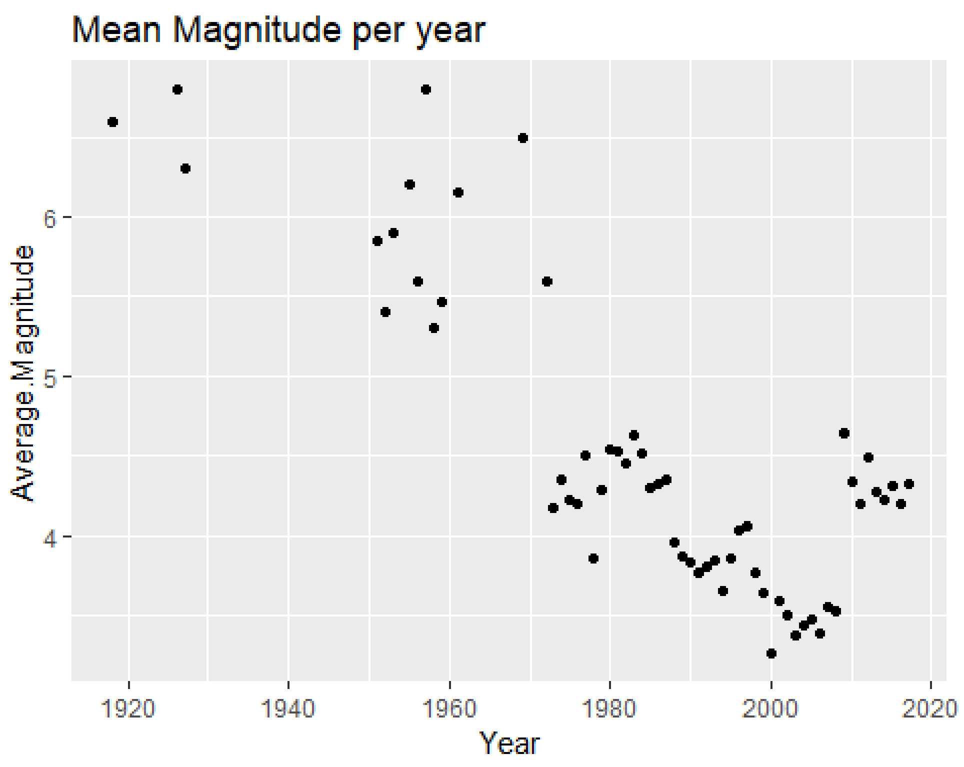
```r
gplot<-ggplot(csv, aes(x=mag)) + ggtitle("Magnitude density curve")
gplot + labs(x="Magnitude (Richter)") + geom_density() #Magnitude
```



```
ggplot(csv, aes(x=year(csv$Date)))+ggtitle("Year distribution") +
  xlab("Year")+theme(text = element_text(size=15))+geom_bar()#Year of date
```



```
#calculating average magnitude per year + graph
avg_mag_ByYear<-setnames(aggregate(csv$mag, list(year(csv$Date)),
mean),c("Year","Average.Magnitude"))
ggplot(avg_mag_ByYear, aes(x=Year,y=Average.Magnitude)) + geom_point() +
ggtitle("Mean Magnitude per year")
```



- This section can be easily broaden on the investigated variables as well as additional variables.

Data Preparation

Due to the lack of correlation between the magnitude and the other variables, new features are required in order to construct the mentioned models as well the target-feature which represents the binary output:

1. Target-feature – Boolean variable - the forecasted year is labeled as 1 (“Yes“) if the maximum earthquake magnitude exceeds the median of maximum yearly magnitudes (threshold value) or 0 (”No“) if it is below the median.

Seismicity Indicators :

2. TimeDiff - Time elapsed over the last n events (in this project n=50).
3. Mean Magnitude – mean magnitude of the last n events.
4. dE1/2 -The rate of square root of seismic energy released over TimeDiff. Calculated by $\frac{\sum E^{1/2}}{TimeDiff}$. The energy component is calculated by $E = 10^{11.8+1.5M} ergS$. If the release of

seismic energy is disrupted for significantly long periods of time, the accumulated energy will be released abruptly in the form of a major seismic event when the stored energy reaches a threshold.

5. b coefficient – b value of the Gutenberg-Richter inverse power law.

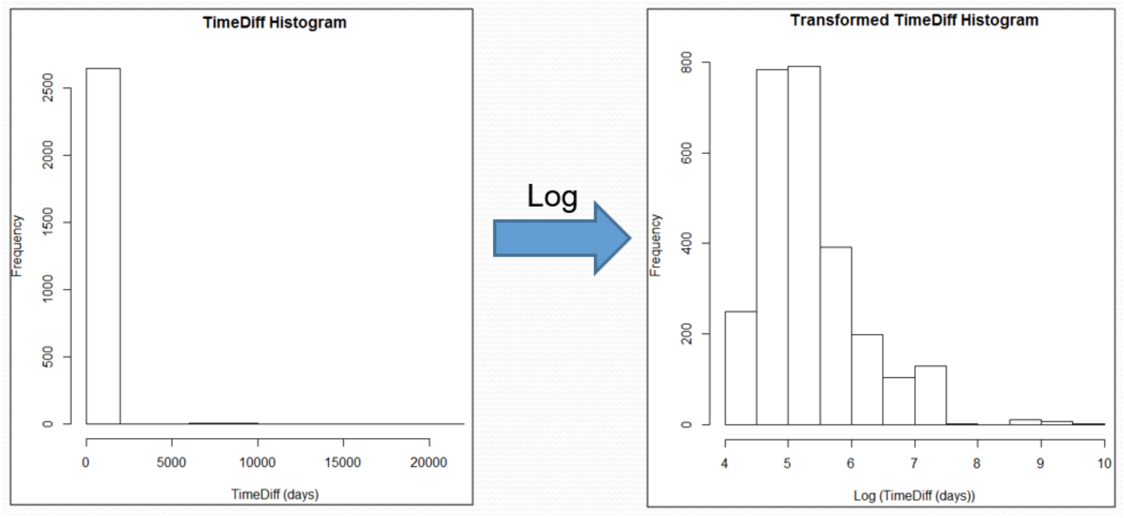
6. η value – The Mean Squared Error (MSE) of the regression line based on the Gutenberg-Richter inverse power law: $\eta = \frac{\sum(log_{10}N_i-(a-bM_i))^2}{n-1}$. N_i is the number of events in the sliding window with magnitude Mi or greater. This is a measure of the conformance of the observed seismic data to the Gutenberg-Richter inverse power-law relationship.

7. a/b coefficients - Difference between the largest observed magnitude and the largest expected magnitude based on the Gutenberg-Richter relationship.

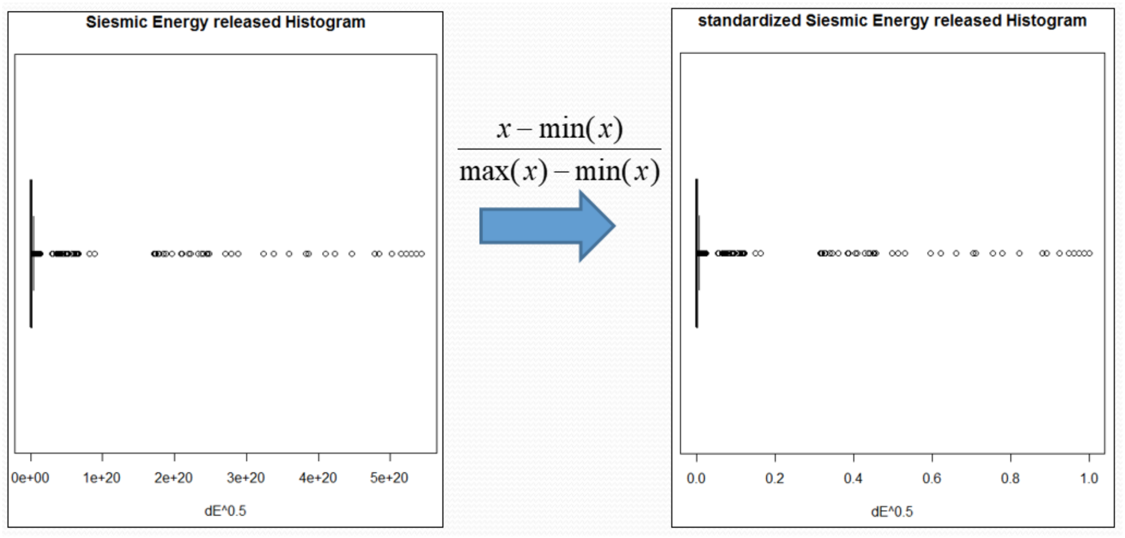
```
#Creating Target-Feature - maximum of last year (360 days) of observations
called "maxLastYear"
for (i  in 1:nrow(csv)){ # i=1
  csv$maxLastYear[i]=max(csv[csv$Date>(csv$Date[i]-360) &
csv$Date<=csv$Date[i],"mag"])
}
#Creating INDICATORS for the models
#Calculating time difference between every 50 events
csv<-mutate(csv,"TimeDiff"=Date - Lag(Date,50))
csv$TimeDiff<-as.integer(csv$TimeDiff)#convert outcome as numeric

#Calculating Magnitude average for last 50 events
csv<-mutate(csv,"MagAvg"=rollmeanr(mag,50,fill=NA))

# Calculating The rate of square root of seismic energy released (dE1/2) for
last 50 events
E<-10^(11.8+(1.5*csv$mag)) #The Energy calculation formula
csv<-
mutate(csv,"dE0.5"=(rollapplyr(E,50,sum,fill=NA))/as.numeric(csv$TimeDiff))

#Calculating the a&b coeffecients of the Gutenberg-Richter inverse power law
##log of total observations of observed mag and above in the last 50 events
csv<-
mutate(csv,"log10N(M)"=rollapplyr(mag,50,function(i){log10(sum(i>=tail(i,1)))
},fill=NA))
csv<-csv[complete.cases(csv)==TRUE,] #filter to have only complete
observations (no NA) in order to be able to apply coefficients extraction
Coefa<-function(df){coef(lm(`log10N(M)`~mag, as.data.frame(df)))[1]}
Coefb<-function(df){coef(lm(`log10N(M)`~mag, as.data.frame(df)))[2]}
csv<-
mutate(csv,"a"=rollapplyr(csv[,c("mag","log10N(M)")],50,Coefa,fill=NA,by.colu
mn = F))
csv<-
mutate(csv,"b"=rollapplyr(csv[,c("mag","log10N(M)")],50,Coefb,fill=NA,by.colu
mn = F))

#Calculating ni (MSE of the created regression line)
csv<-mutate(csv,"a-(b*mag)"=a-(b*mag))
ni<-function(df){mse(df[,"log10N(M)"],df[,"a-(b*mag)"])}
csv<-mutate(csv,"ni"=rollapplyr(csv[,c("log10N(M)","a-
(b*mag)")],50,ni,fill=NA,by.column = F))

#Calculating Magnitude deficit (a/b)
csv<-mutate(csv,"ab.Ratio"=a/b)
```

All of these indicators combine 50 earthquake events information (moving stats - similar to moving average by definition) which will be used in the construction of the models.
In order to get better evaluated models it is essential first to check what is the distribution of the indicators:

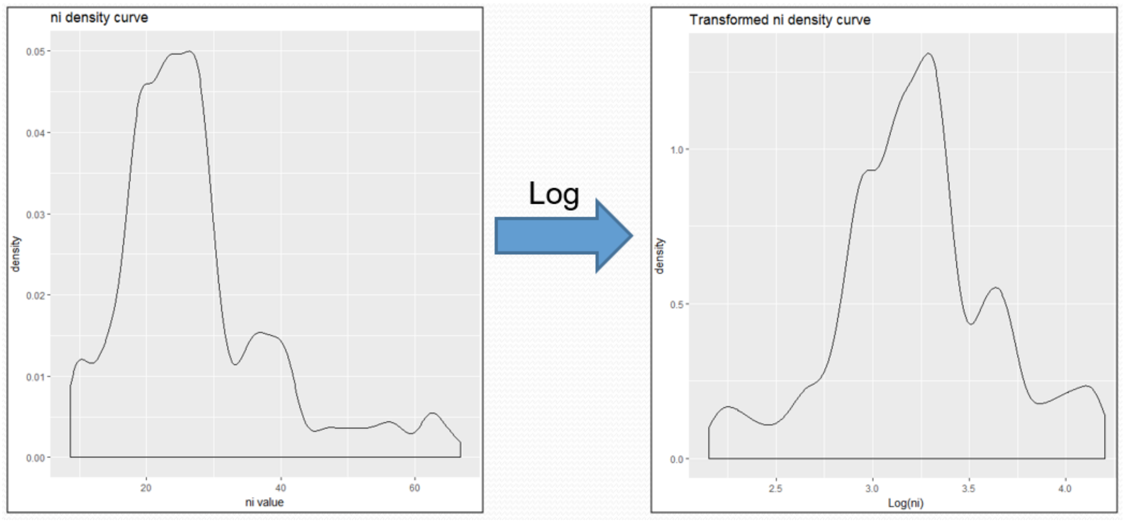
```
##hist(csv$TimeDiff, main = "TimeDiff Histogram", xlab = "TimeDiff (days)")
#TimeDiff
##gplot<-ggplot(csv, aes(x=MagAvg)) + ggtitle("Average Magnitude density
curve")
##gplot + labs(x="Average Magnitude (Richter)") + geom_density() #Avg
Magnitude
##boxplot(csv$dE0.5, main = "Siesmic Energy released Histogram", xlab =
"dE^0.5", horizontal = T) #Energy
##gplot1<-ggplot(csv, aes(x=b)) + ggtitle("b coefficient density curve")
##gplot1 + labs(x="b value") + geom_density() #b coefficient
##gplot2<-ggplot(csv, aes(x=ni)) + ggtitle("ni density curve")
##gplot2 + labs(x="ni value") + geom_density() #ni value
##gplot3<-ggplot(csv, aes(x=ab.Ratio)) + ggtitle("a/b Ratio")
##gplot3 + labs(x="a/b Ratio") + geom_density() #a/b Ratio
```



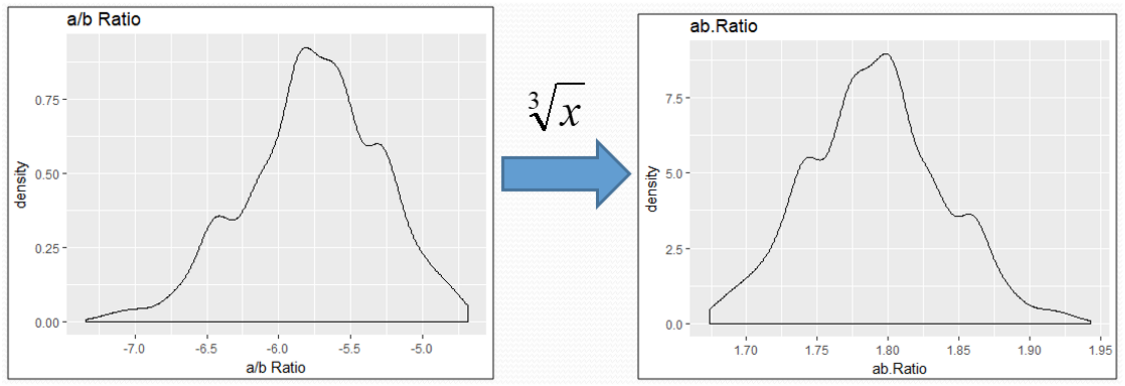
TimeDiff Transformation



Energy Transformation



η Transformation



a/b ratio transformation

Several features were transformed, according to the figures above:

- TimeDiff and η - log transformation due to right skewed data.
- dE0.5 - data standardization to the range between 0-1 due to a very wide range of data.
- ab.Ratio - cube root transformation due to left skewed data.

Below are the transformed distributions:

```r
#Log-Transformation and standardisation to some indicators for better models
csv[,"TimeDiff"] <- sapply(csv[,"TimeDiff"], log)
csv[,"ab.Ratio"] <- sapply(csv[,"ab.Ratio"], function(x) abs(x)^(1/3))
csv[,"ni"] <- sapply(csv[,"ni"], log)
csv[,"dE0.5"] <- sapply(csv[,"dE0.5"], function(x) (x-min(x))/(max(x)-
min(x)))

#graphics of transformed data
##hist(csv$TimeDiff, main = "TimeDiff Histogram", xlab = "TimeDiff (days)")
#TimeDiff
##gplot<-ggplot(csv, aes(x=ab.Ratio)) + ggtitle("ab.Ratio")
##gplot + labs(x="ab.Ratio") + geom_density() #ab.Ratio
##boxplot(csv$dE0.5, main = "Siesmic Energy released Histogram", xlab =
"dE^0.5", horizontal = T) #Energy
##gplot2<-ggplot(csv, aes(x=ni)) + ggtitle("ni density curve")
##gplot2 + labs(x="ni value") + geom_density() #ni value
```

Modeling

In this section the mentioned models (Decision tree, KNN, SVM, NNET) will be constructed and evaluated. The models will be based on a training-set which represents the first 2/3 complete observations while the remaining 1/3 will be used as test-set. The model evaluation will be first rely on accuracy values and eventually concluded based on calculated F-measures and ROC curves.

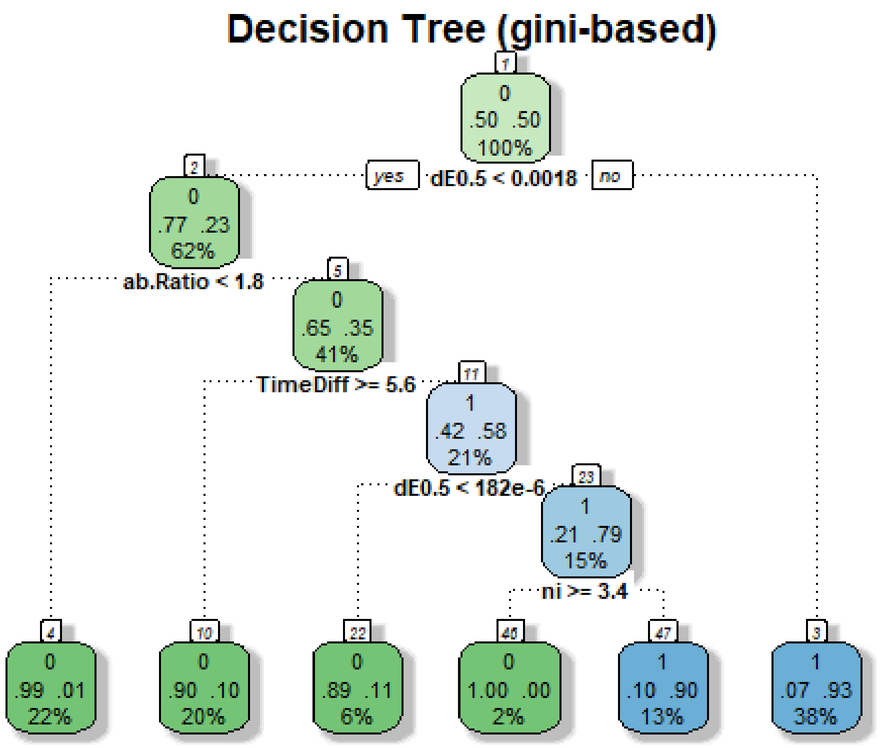
```r
#Defining AboveMedian column as factor (True=1, False=0)
csv<-mutate(csv, "AboveMedian"=as.factor(ifelse(maxLastYear>=threshold,1,0)))
csv<-csv[complete.cases(csv)==TRUE,]#filtering all non-completed observations
csv<-csv[,-c("latitude", "longitude","magType")]


#Arange sets and correlations for testing
train_set <- csv[1:(nrow(csv)*(2/3)),]#Defining training set (2/3 of events
by appearance order)
test_set <- csv[(nrow(csv)*(2/3)):nrow(csv),]#Defining test set (the last 1/3
events)
target_feature <- "AboveMedian"
fmla <-
as.formula(paste(target_feature,"~TimeDiff+dE0.5+b+ni+ab.Ratio+MagAvg",sep=""
))
```

```
#fmla2 <-
as.formula(paste(target_feature,"~depth+TimeDiff+dE0.5+b+ni+ab.Ratio+MagAvg",
sep=""))
numeric_features <- sapply(csv, is.numeric) #assign variable to all numeric
features
```

For commparison purpose, the fraction of true target-feature (AboveMedian) values (value of "1") is 0.4674037.
Below are the constructed models and their confusion matrices and accuracy:

Decision Tree

```
#Simple Decision Tree
fit <- rpart(fmla,data=train_set, cp=0.03, method = 'class')
fancyRpartPlot(fit, cex=0.7,tweak=1, sub = "")
title("Decision Tree (gini-based)",line = 3.25)
```

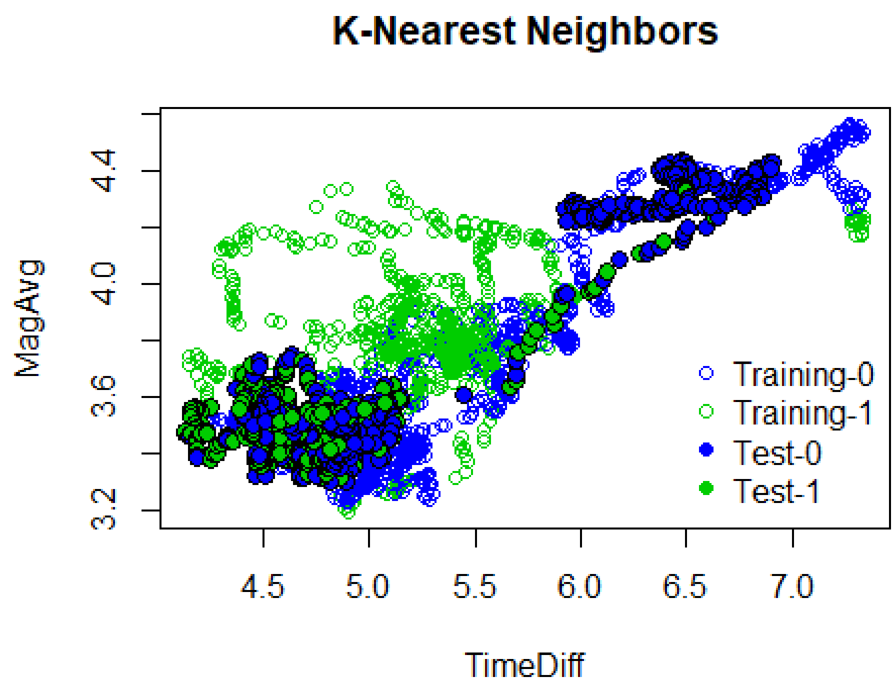


```
DT_pred<-predict(fit,test_set, type = "prob")
DT_predClass<-predict(fit,test_set, type = "class")
confusion_matrix_DT <- table(test_set$AboveMedian,DT_predClass)
confusion_matrix_DT

##    DT_predClass
##       0   1
##   0 441  73
##   1 146 215
```

The generated tree uses 4 out of the 6 indicators. The advantage of the decision tree is that the model selects the most discriminatory features which means that in this case the most contributing indicator is dE0.5 (released energy) since it serves as a first and secondary classifier. Another advantage of this model is its ease of interpretation. On the other hand this model is greedy and not necessarily represents the best fit to the data. The accuracy of this model is 0.7497143 which shows a moderate fit to the data.

KNN

```
#knn on all numeric features
#knn execution k=3 found to be most usefull
knn_3 <- knn(train = train_set[,..numeric_features], test =
test_set[,..numeric_features], cl = train_set$AboveMedian, k=3, prob = T)
knn_3_prob<-attr(knn_3, "prob")
knn_3_prob <- 2*ifelse(knn_3 == "-1", 1-knn_3_prob, knn_3_prob) - 1
#confusion matrix of the created knn and its % of corrected predictions
confusion_matrix_knn_3 <- table(test_set$AboveMedian,knn_3)
confusion_matrix_knn_3

##    knn_3
##       0   1
##   0 399 115
##   1 153 208

sum(diag(confusion_matrix_knn_3))/sum(confusion_matrix_knn_3) # % of true
cases

## [1] 0.6937143

#plotting the model with the test set
plot(train_set[,c("TimeDiff","MagAvg")],
     col=c(4,3)[train_set$AboveMedian], main = "K-Nearest Neighbors")
points(test_set[,c("TimeDiff","MagAvg")], bg=c(4,3)[as.numeric(knn_3)],
       pch=c(21,21)[as.numeric(knn_3)],cex=1.2, col=grey(0))
legend("bottomright", pch = c(1,1,21,21), col=c(4,3), pt.bg=c(4,3),
       legend = c("Training-0", "Training-1", "Test-0", "Test-1"), bty = "n")
```

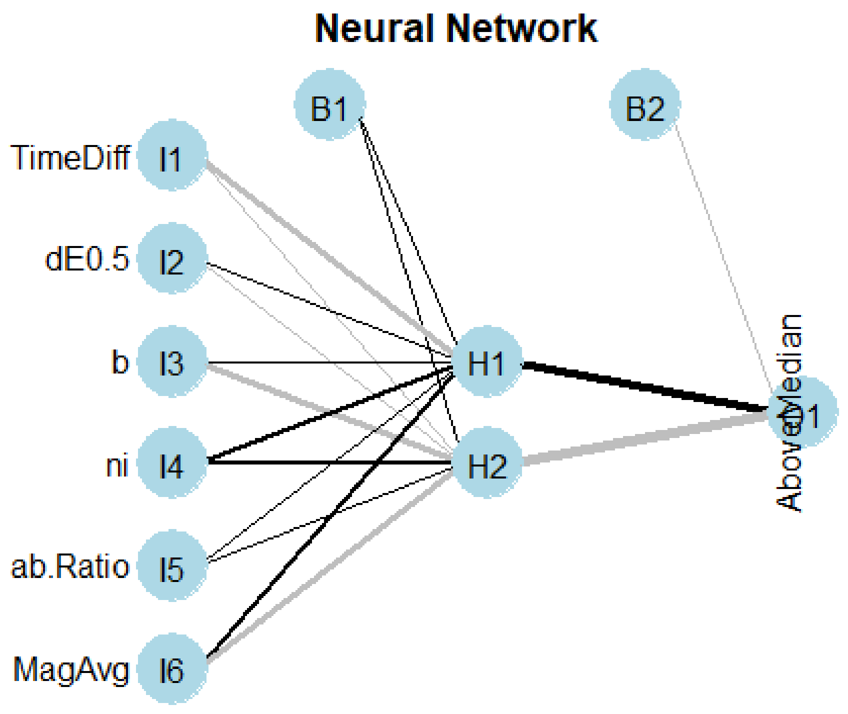


The shown KNN model present the scattering of both training and test sets on the dimension (axes) of TimeDiff and average magnitude (MagAvg). The calculated accuracy of 0.6937143 show less fitting in relation to the decision tree. This kind of model is robust to noisy training data but also require selection of K and attributes. In this case best KNN model found was with K=3 (achieved by trial and error).

NNET

```
# nnet
#nnet model
nnet_model <- nnet(fmla, data = train_set, size = 2 ,maxit=1000, decay=5e-1)
#nnet prediction on test data
nnet_predClass <- predict(nnet_model,test_set, type = "class")
nnet_pred <- predict(nnet_model,test_set, type = "raw")
#confusion matrix of the created nnet and its % of corrected predictions
confusion_matrix_nnet <- table(nnet_predClass,test_set$AboveMedian)
confusion_matrix_nnet
sum(diag(confusion_matrix_nnet))/sum(confusion_matrix_nnet)

#Plotting the model
#import the function from Github
library(devtools)
source_url('https://gist.githubusercontent.com/fawda123/7471137/raw/466c1474d
0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')

## SHA-1 hash of file is 74c80bd5ddbc17ab3ae5ece9c0ed9beb612e87ef
```

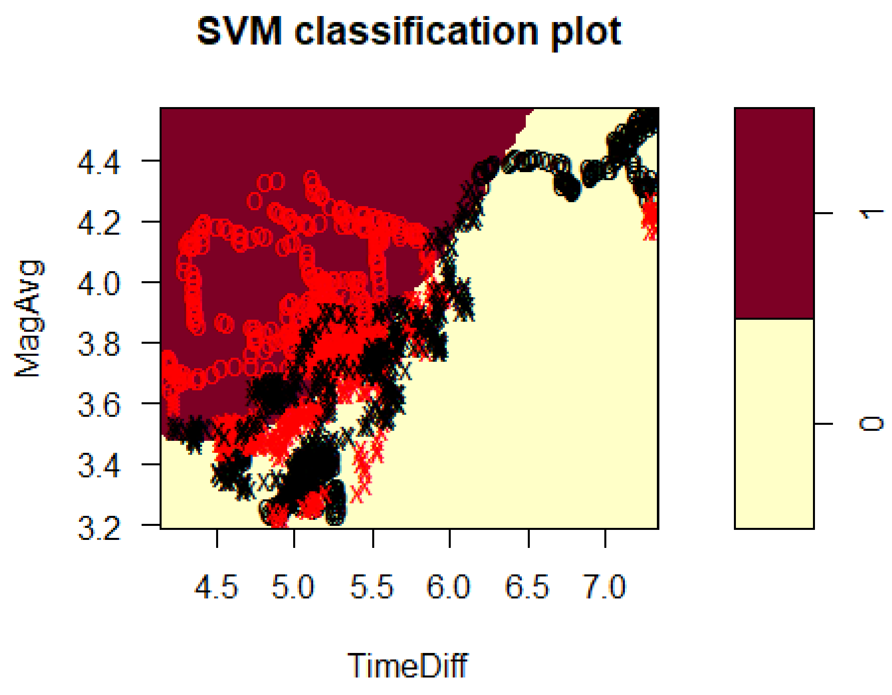
```r
par(mar=c(0,0,1,1))
plot.nnet(nnet_model, main = "Neural Network", y.lab = "")

## Loading required package: scales

## Loading required package: reshape

##
## Attaching package: 'reshape'

## The following object is masked from 'package:class':
##
##     condense

## The following object is masked from 'package:lubridate':
##
##     stamp

## The following object is masked from 'package:data.table':
##
##     melt

## The following objects are masked from 'package:plyr':
##
##     rename, round_any

mtext("AboveMedian", side=4, line=-4, cex.lab=2,las=3)
```



The constructed NNET model show two hidden nodes between the input and the output layers and was defined with a maximum number of iterations of 1000 and a decay value of 5e-1. These parameters found to be the optimal for this data-set. The accuracy value of the model is 0.6765714 which is also lower than the accuracy level of the decision tree. The adventage of the NNET model is its capability to do non-linear separation and that it is easy to synthesize.

SVM

```
# SVM
#SVM model and testing
SVM_model <- svm(fmla,train_set, kernel="radial", scale = F)
SVM_Pred <- predict(SVM_model,test_set, type = "class", decision.values = T)
svm.probs<-attr(SVM_Pred,"decision.values")
SVM_class <- predict(SVM_model,test_set, type = "class")
#confusion matrix of the created SVM and its % of corrected predictions
confusion_matrix_SVM <- table(test_set$AboveMedian,SVM_Pred)
confusion_matrix_SVM

##    SVM_Pred
##       0   1
##   0 410 104
##   1 227 134

sum(diag(confusion_matrix_SVM))/sum(confusion_matrix_SVM)

## [1] 0.6217143

par(mar=c(5.1, 4.1, 4.1, 2.1))
plot(SVM_model, train_set, MagAvg ~ TimeDiff,
     slice=list(dE0.5=0.0164983, b=-0.6761, ni=3.225, ab.Ratio=1.791))
```



The figure above shows the SVM model in the same dimension as the KNN model (TimeDiff vs MagAvg) given the average value of all other indicators (dE0.5=0.0164983, b=-0.6761, η=3.225, ab.Ratio=1.791).
The SVM model was constructed using radial basis function (which can be seen clearly on the model graph). The separation between the two classifications can be seen in the graph. Accuracy value of this model was found to be the lowest among the constructed models (0.6217143). The benefits using such model is the capability of the SVM to find non-linear separation in a highly dimensional space. On the other hand, such model is not intuitive to interpret because of its highly algorithmic complexity.

Evaluation Measures

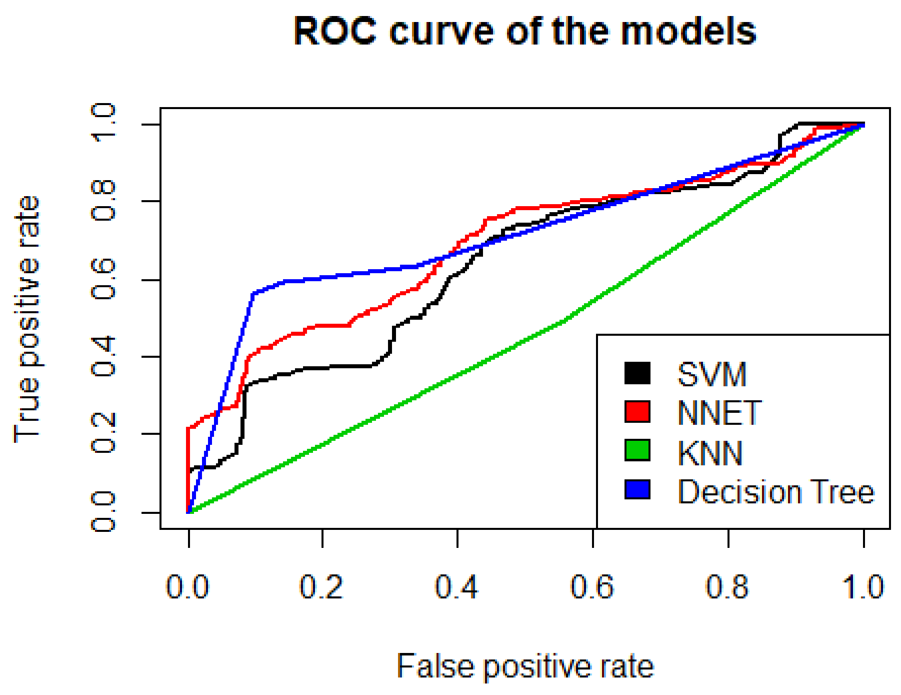
As mentioned at the begining of the modeling task, the constructed models will be also evaluated based on their calculated F-measure and ROC curve.
The F-Measure Contains two basic parameters derived from the confusion matrix (Precision & Recall). Below is a table containing the F-measure values of each model. It is also clear from this table that the decision tree is more reliable (preferred) than the other models regarding this set of data.

```
#F-Measure
treePrecision<-confusion_matrix_DT[2,2]/sum(confusion_matrix_DT[,2])
treeRecall<-confusion_matrix_DT[2,2]/sum(confusion_matrix_DT[2,])
FDT<-(2*treePrecision*treeRecall/(treeRecall+treePrecision))

knnPrecision<-confusion_matrix_knn_3[2,2]/sum(confusion_matrix_knn_3[,2])
```

```
knnRecall<-confusion_matrix_knn_3[2,2]/sum(confusion_matrix_knn_3[2,])
Fknn<-(2*knnPrecision*knnRecall/(knnRecall+knnPrecision))

nnetPrecision<-confusion_matrix_nnet[2,2]/sum(confusion_matrix_nnet[,2])
nnetRecall<-confusion_matrix_nnet[2,2]/sum(confusion_matrix_nnet[2,])
Fnnet<-(2*nnetPrecision*nnetRecall/(nnetRecall+nnetPrecision))

SVMPrecision<-confusion_matrix_SVM[2,2]/sum(confusion_matrix_SVM[,2])
SVMRecall<-confusion_matrix_SVM[2,2]/sum(confusion_matrix_SVM[2,])
FSVM<-(2*SVMPrecision*SVMRecall/(SVMRecall+SVMPrecision))

FTable<-matrix(c(FDT, Fknn, Fnnet, FSVM), ncol = 4)
colnames(FTable)<-c("Decision Tree", "KNN", "NNET", "SVM")
rownames(FTable)<-"F-Measure"
FTable

##            Decision Tree       KNN      NNET       SVM
## F-Measure     0.6625578 0.6081871 0.5398374 0.4474124
```

Another evaluation measure is the ROC curve. The clear advantage of this measure in relation to the F-measure is its graphic visualisation which shows in a comprehensive way the differences between the evaluated models.
Below are the generated ROC curves:

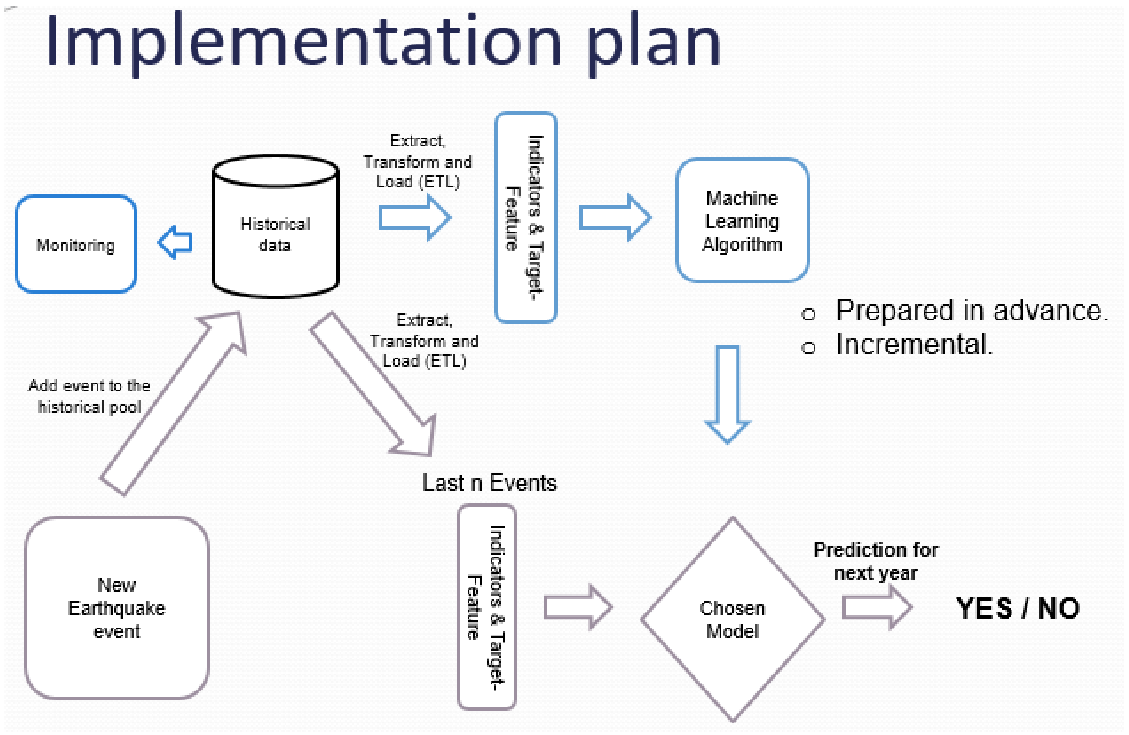
```
#ROC
SVM_Predict <-
prediction(as.numeric(svm.probs),as.numeric(test_set$AboveMedian))
nnet_Pred <-
prediction(as.numeric(nnet_pred),as.numeric(test_set$AboveMedian))
knn3_Pred <-
prediction(as.numeric(knn_3_prob),as.numeric(test_set$AboveMedian))
DT_Pred <-
prediction(as.numeric(DT_pred[,2]),as.numeric(test_set$AboveMedian))
roc_SVM<-performance(SVM_Predict,"tpr","fpr")
roc_nnet<-performance(nnet_Pred,"tpr","fpr")
roc_knn3<-performance(knn3_Pred,"tpr","fpr")
roc_DT<-performance(DT_Pred,"tpr","fpr")

roc.tot<-plot(roc_SVM, lwd=2, main="ROC curve of the models")
roc.tot<-plot(roc_nnet, add=T, col=2, lwd=2)
roc.tot<-plot(roc_knn3, add=T, col=3, lwd=2)
roc.tot<-plot(roc_DT, add=T, col=4, lwd=2)
legend("bottomright",legend = c("SVM", "NNET", "KNN","Decision Tree"),fill =
1:4)
```



This graph shows the clear inferiority of the KNN model in relation to the other models (despite its relative high F-measure). Not only that, but this model also found to be worse than a total random model (described in the literature as the diagonal line connecting between 0,0 and 1,1 points).Also worth noting is the similarity between the other 3 models in the higher range of the false positive rate and their divergence in the lower range. Taking into account all evaluation measures, the conclusion regarding the best fitted model for the task in question is to use the decision tree model.

Implementation plan and further examination

The implementation plan in the figure below describes in a basic way how the chosen model should fit into the corporation's system (considered as a general preliminary plan). The purpose of the model is to supply answer regarding the predicted value (in this case: yes/no output) which in turn dictate the response of the corporation to the tested situation (for example: the amount to be collected for force majeure insurance).


Implementation plan
Monitoring
Historical data
Extract, Transform and Load (ETL)
Indicators & Target-Feature
Machine Learning Algorithm
Prepared in advance.
Incremental.
Extract, Transform and Load (ETL)
Add event to the historical pool
Last n Events
New Earthquake event
Indicators & Target-Feature
Chosen Model
Prediction for next year
YES / NO

Implementation plan

Conclusion

In conclusion, the purpose of this project was to supply basic model evaluation regarding earthquake data. The tested models show different functionality and qualities. In order to maximize the process of modeling and understanding, further examiniation is required such as:

- Expand and elaborate data understanding.
- Check other transformations.
- Check models with more effective indicators (exclude indicators that has little contribution).
- Improve models according to discovered patterns.
- Check the significance of predications.
- Check additional models.
- Check other configurations of implementations.

- Apply model updating according to new data.
- Combine additional models from related fields (Force majeure).

References

1. Wikipedia “Earthquake” Page. URL: http://en.wikipedia.org/wiki/Earthquake. Accessed 1/31/2013.
2. USGS “Measuring the size of an earthquake” Page. URL: http://earthquake.usgs.gov/learn/topics/measure.php. Accessed 1/31/2013
3. R Core Team (2012). ”R: A language and environment for statistical computing.” URL: http://www.R-project.org
4. Seber, George AF, and Alan J. Lee. Linear regression analysis. Vol. 936. Wiley, 2012.
5. Ferguson, Thomas S. A Course in Large Sample Theory: Texts in Statistical Science. Vol. 38. Chapman & Hall/CRC, 1996.
6. R Markdown Page. URL: http://www.rstudio.com/ide/docs/authoring/using_markdown. Accessed 1/31/2013
7. USGS Real Time Data and Feeds Page. URL: http://earthquake.usgs.gov/earthquakes/feed/. Accessed 1/31/2013.
8. Last M, Rabinowitz N, Leonard G (2016) Predicting the Maximum Earthquake Magnitude from Seismic Data in Israel and Its Neighboring Countries. PLoS ONE11(1): e0146101. https://doi.org/10.1371/journal.pone.0146101

Appendix - Enviroment

```
print("Operating System:")

## [1] "Operating System:"

version

##                _                           
## platform       x86_64-w64-mingw32          
## arch           x86_64                      
## os             mingw32                     
## system         x86_64, mingw32             
## status                                     
## major          3                           
## minor          6.0                         
## year           2019                        
## month          04                          
## day            26                          
## svn rev        76424                       
## language       R                           
## version.string R version 3.6.0 (2019-04-26)
## nickname       Planting of a Tree
```