Earthquake Prediction Analysis by Python

Earthquake Prediction Analysis by Python

Submitted to:

Ahmed Imran Kabir

Lecturer

School of Business and Economics

Submitted by:

Md Ali Abdullah

ID: 111 182 048

Major: Management Information Systems

Summer 2023

School of Business and Economics

United International University

Date of submission: 28 August 2023

Letter of Transmittal

Date: 28 August 2023

Ahmed Imran Kabir

Lecturer

School of Business & Economics

United International University

Subject: Submission of report on *"Earthquake Prediction Analysis by Python."*

Dear Sir,

This is to inform you that I researched the designated topic, "Python Analysis of Earthquake Predictions," as thoroughly as possible.

As part of my university's bachelor's degree requirements, I completed this assignment. I have made every effort to fulfill the project's objectives and exceed everyone's expectations. I'm hoping you will be kind enough to accept my project work and provide me with constructive feedback on my endeavor. I am certain that the findings and presentation of the report will meet or exceed your expectations.

Sincerely Yours,

Md Ali Abdullah

ID: 111 182 048

Signature:

Date: 28 August 2023

Certification of Similarity Index

Declaration of the Student

I, the undersigned, verify that this project report accurately describes my work from our study with Ahmed Imran Kabir Sir. I certify that my study resulted in the above claims and conclusions.

Furthermore, I can speak to the report's uniqueness, which I achieved under the general supervision and guidance of my teacher. This thesis has not been submitted to any other institution in order to achieve a specific degree, diploma, or certificate from this university, a university in Bangladesh, or any other university in the world. I followed the university's rules while drafting the report. In the report's wording, I always provided credit where credit was due and appropriately referenced any elements (data, a theoretical model, and text) that I borrowed from other sources.

Md Ali Abdullah

ID: 111 182 048

Acknowledgment

It has been an honor to have the proper assistance and inspiration from an exclusive set of people who have either directly or indirectly assisted in the planning of this project.

I want to start off by saying how grateful I am to Almighty Allah for giving me the capacity to finish this entire assignment successfully. Secondly, I want to thank Mr. Ahmed Imran Kabir Sir, who is in charge of my project, for his help. He substantially helped me by offering instructions for correctly finishing the project work and motivated me at different points. The project work and analysis work that have been accomplished here would have been almost impossible without his guidance and help.

Last but not least, I want to express my gratitude to my family and friends for morally supporting me as I finished my project and research. Without their emotional support, finishing this study would have been incredibly difficult for me.

Abstract

In the fields of computer science and the environment, earthquake prediction is a common research challenge. As we are dealing with earthquakes, it is vital to build an effective earthquake system. Nowadays, deep learning and machine learning techniques are introduced to reduce the time and effort required from humans. Since the behavior of our data is similar to that of natural earthquakes, it is conceivable to use the same methodology to anticipate when they will occur. We have gone through various steps such as feature engineering, visualization, applying Artificial Neural Network, and Random Forest Regression.Here, Feature engineering is challenging and complex. Here, ANN achieved 92.42% accuracy and Random Forest Regression achieved a best fit score 87.49%**.** Thus we may draw the conclusion that combining seismic activity with machine learning and deep learning models sounds outstanding as it produces better results.

[Keywords: Python, feature engineering, deep learning, machine learning, Artificial Neural Network, Random Forest Regression model and prediction]

Table Of Content

CHAPTER I: INTRODUCTION

1.1 Background of the Study

Predicting earthquakes is one of the leading causes of structural damage and loss of life across the globe. Seismological research and environmental engineering have grown in popularity in an apparent and dramatic way[1]. In essence, accurate prediction enables us to prepare for the worst-case scenarios and necessary precautions before an earthquake occurs. In the field of seismology and earthquake geology, having the capacity to precisely anticipate the Time, Depth, and Magnitude of earthquakes would be of tremendous assistance. An earthquake prediction may give information about the likelihood that an earthquake will take place, as stated by Jordan et al. [2]. Specific parameters, such as geographic testing, region, magnitude range, authoritative data sets used to evaluate forecasts, and evaluation metrics, must be established before commencing an experiment. Rhoades et al. concluded that future measurements should be compared to the earthquake model's predictions in order to evaluate the model's predictive ability [3].

CSEP has conducted a number of earthquake forecasting experiments at testing centers using autonomous software to evaluate earthquake forecasts over the course of the last decade. [4][5] As a result, pyCSEP was created to provide Python evaluation methods for earthquake forecasts that have been thoroughly tested. It is remarkable that this program can be utilized directly by researchers in their task.

1.2 Statement of the Problem

It is important to point out that the global population is severely confronted with earthquake problems. Because of the event's non-linearity and unreliability, predicting a seismic event is thought to be an impossibility [6]. But machine learning algorithms have turned it into a potentially fascinating phenomenon. Regression and classification models are two different kinds of prediction models that it may construct [7].

In order to find a solution to such a problem, it is necessary to conduct an extensive literature review to predict laboratory earthquakes from continuous seismic data in quasi-periodic laboratory seismic cycles and apply different machine learning and deep learning algorithms in order to detect earthquakes for minimizing the damage and financial loss.

1.3 Objective of the Study

We propose several cutting-edge machine learning and deep learning models in order to detect earthquakes as it reduces time and damage cost. The main contribution of our project is summarized below-

- Two different models Artificial Neural Network and Random Forest Regression model are explored to predict earthquakes.
- We chose accuracy as our evaluation metric and achieved 92.42% for Artificial Neural Network.
- Random Forest Regression model acquired a best fit score 87.49% .
- Specifics of the event; including its size, location, and timing.

Thus, my contributions might lead to earthquake detection which helps to predict earthquakes in the future.

1.4 Conceptual Framework and Research Hypotheses

Our conceptual framework is - Create additional aggregation features, calculated on the segments.Thus, Our proposed method is depicted in Figure 1. It illustrates literature review, study about dataset, feature engineering, process train file, feature importance Machine learning and deep learning algorithm and evaluation metric respectively.

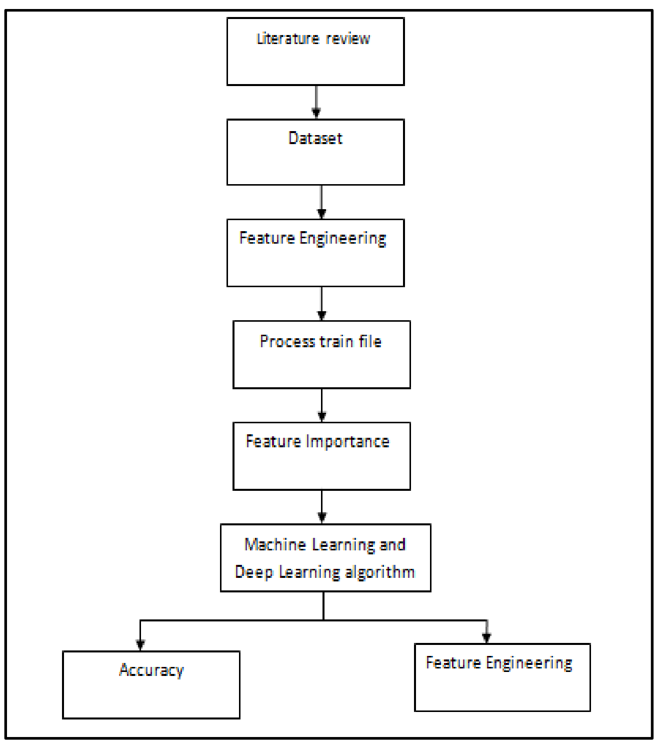


Fig.1.Proposed Method

1.5 Motivation of the Study

I love to introduce myself as a "forever learner" in this world. I started my study to know the topic thoroughly. I was motivated to study for a variety of reasons such as -

- Willing to resolve unresolved issues.
- Having a desire to be of service to society by delivering amazing research.
- Early warning of ground shaking leads to preventing the huge loss of life and property.
- Guide to designing infrastructure that can stand during earthquakes.
- Understand real-life issues and act necessarily.
- Growing knowledge for further study.
- Develop research skills to improve academic knowledge.
- Explore a new world regarding earthquakes.

1.6 Scope and Limitation of the Study

We should implement more deep learning models with raw data and choose Mean absolute error(MAE) as evaluation metric. Thus, it can result in better and more accurate results. We find out through extensive experiments that we should generate a larger dataset.

It takes a lot of effort and extensive subject expertise to analyze feature engineering which includes feature scaling in machine learning models. The result is still not ready to be transferable to the real world. If the result is transferable from the lab to the real world, researchers can take necessary actions to improve earthquake hazards, which could save millions of lives and billions worth of property. Yet it could not predict earthquake fully in real life.

1.7 Definition of Key Terms

CNN - Convolutional Neural Network
LSTM - Long short-term memory
GRU - Gated recurrent units
Bi-LSTM - Bi-directional long short term memory
SVR - Support Vector Regression
ANN- Artificial Neural Network

CHAPTER II: LITERATURE REVIEW

2.1 Descriptive Form

Seismometers and GNSS are used to measure Rapid ground-surface displacements. Both types of measurements help to find out the characteristics of the deformation source by investigating the inverse issue [8]. According to Tarantola et al. [9], parameter uncertainties should be quantified and brought on by measurement and theory errors. Moreover, Dettmer et al. [10] introduce theory errors whereas discretization of the deformation source causes the earthquake fault plane to be divided into a predetermined number of patches. Pratiksha et al. [11] chose the combination of Random Forest and Support Vector Machine Algorithms for earthquake prediction and obtained 83% accuracy.

In accordance with the findings of Wenrui et al's study, it has been shown that significant seismic events are often accompanied by subsequent aftershocks. The determination of the location of aftershocks may be achieved by the analysis of the onset timings of the P- and S-waves. The seismic data from a total of 16 earthquake stations is recorded and stored in the SAC file format. Subsequently, the recorded data is processed by trimming and eliminating any extraneous noise, resulting in the extraction of the desired waveform. The user did not provide any text to rewrite. In their study, Khawaja et al. employed tree-based ensemble classifiers, including rotboost, random forest, and rotation forest, to perform binary classification and transform magnitude values into binary classifications. They integrate characteristics based on three factors: the Gutenberg-Richter relationship, seismic rate variation, and foreshock frequency distribution. By attaining 95.9% accuracy for Rotation forests, they distinguished themselves as the top model among the others. [13].

Kumari et al. designed a classification model using Bagging and Boosting for batch processing and online processing whereas online processing performs much better than batch processing. It is proven for larger datasets. This model is efficient as it helps to improve precision and it is a remarkable ensemble method. [14] Ant´onioE et al. depict failure of event detection which is dependent on STA/LTA ratio. It is obvious that bogus alarms can be generated from PVAQ and the PESTR station which is the main seismic source of their research. They come forward with a new detector using an SVM classifier which is capable of separating alarm and event perfectly. Thus, they tried to reduce the time for the detection system which requires a lot of experimental study and finally they obtained 1.3 and 1.8 respectively. To detect correctly, it is mandatory to increase recall and specificity value.[15] Also, Olha et al. chose a dataset based on laboratory earthquakes which is the simulation of seismological fault. It appears exactly as a natural earthquake and we have applied random forest technique and predicted 1.61 Mean Absolute Error(MAE).[16] Vasyura et. al explores open-source Python software The Bayesian Earthquake Analysis Tool (BEAT) for earthquake detection by calculating Green's functions. It offers a framework for geodetic and seismic data and considers different aspects of source. It is also helpful for further geophysical study. [17]

2.2 Summary Form

Attribute	Definition	Scholars
Rapid ground-surface displacements	Using seismometers and GNSS to measure. By analyzing the inverse problem, both sets of measurements can be used to learn more about the features of the source of the deformation.	Kikuchi, M. and Kanamori, H. (1982). Inversion of complex body waves. Bull. Seismol. Soc. Am., 72(2):491–506.] [Yagi, Y. and Fukahata, Y. (2011). Introduction of uncertainty of Green's function into waveform inversion for seismic source processes. Geophys, J. Int., 186(2):711–720
Theory errors	The quantification of parameter uncertainties is crucial, since they arise due to inaccuracies in both measurement and theory. The process of discretizing the deformation source results in the partitioning of the earthquake fault plane into a pre-established quantity of patches.	Dettmer, J., Benavente, R., Cummins, P. R., and Sambridge, M. (2014). Trans-dimensional finite fault inversion. Geophys, J. Int., 199(2):735–751. Pratiksha Bangar, Deeksha Gupta, Sonali Gaikwad, Bhagyashree Marekar, Jyoti Patil. Earthquake Prediction using Machine Learning Algorithm

Aftershocks	By examining the arrival times of the P- and S-waves, we may determine the location of these aftershocks.	W. Li, N. Narvekar, N. Nakshatra, N. Raut, B. Sirkeci and J. Gao, “Seismic Data Classification Using Machine Learning,” 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, 2018, pp. 56-63.
Ensemble Classifiers	rotboost, random forest, rotation forest and convert magnitude into binary classes	K. M. Asim, A. Idris, F. Mart´ınez-A´ lvarez and T. Iqbal, “Short Term Earthquake Prediction in Hindukush Region Using Tree Based Ensemble Learning,” 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, 2016, pp. 365-370.
Ensemble method Bagging and Boosting	batch processing and online processing	Kumari, G. P. (2012). A Study of Bagging and Boosting approaches to develop meta-classifiers. *Eng. Sci. Technol. An Int. J*, *2*(5), 850-855.
SVM classifier	Distinguish alarm and event, increase recall and specificity value.	VINOD, M., REDDY, P. B., HEMALATHA, K., PRIYA, C. L., & NAVNEETHA, Y. (2022). CLASSIFYING EARTHQUAKE DAMAGE TO BUILDINGS USING MACHINE LEARNING.

random forest technique	predicted 1.61 Mean Absolute Error(MAE)	Machine Learning Predicts Aperiodic Laboratory Earthquakes,Olha Tanyuk, Daniel Davieau, Charles South and Daniel W. Engels
The Bayesian Earthquake Analysis Tool (BEAT)	framework for geodetic and seismic data and considers different aspects of source	The Bayesian Earthquake Analysis Tool H. Vasyura-Bathke1,2, J. Dettmer3, A. Steinberg5, S. Heimann4, M. Isken5,O. Zielke1, P.M. Mai1, H. Sudhaus5, S. Jónsson1

Table1. Summary Form

2.3 Literature Breakdown

It is proven that earthquake waves are very common nowadays. Multiple research projects were explored to find out the forecasting and reasons behind earthquakes.

2.3.1 Rapid ground-surface displacements: Most studies examined the inverse issue, seismometers and GNSS readings may be used to examine the features of the deformation source.

2.3.2 The presence of an unlimited number of parameters may lead to inaccuracies in both measurement and theory. Furthermore, the process of discretizing the deformation source leads to the partitioning of the earthquake fault plane into a pre-established quantity of segments, which afterwards introduces a theoretical inaccuracy.

2.3.3 Aftershocks: It is obvious that we want to determine the location of aftershocks.

2.3.4 Ensemble methods: It efficiently compared rotboost, random forest, rotation forest and thus rotboost stood top among them. Bagging and Boosting is known as a popular ensemble method that is useful for larger datasets.

2.3.5 Failure of event detection: A SVM classifier is applied for distinguishing fake alarm and event accurately.

2.3.6 The Bayesian Earthquake Analysis Tool (BEAT) provides a broad framework that can consider source aspects. Researchers can utilize it to study the geophysical condition of the earthquake.

CHAPTER III: METHODOLOGY

In order to perform earthquake forecasting, the whole workflow is separated into three parts such as dataset, feature engineering, machine learning and deep learning models. We have tested one machine learning model and one deep learning model such as Artificial Neural Network and Random Forest Regression.

3.1 Identifying Data:

The dataset we used is about earthquake prediction and is obtained from [18] by The National Earthquake Information Center (NEIC). It is based on all recorded earthquakes of magnitude 5.5 or greater since 1965. Every earthquake's record of the 'Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude' is listed in this dataset.

3.2 Capturing Data:

The data are derived from experiments conducted on rocks with a double direct shear geometry. Earth's tectonic fissures are modeled in the laboratory. Two fault gouge layers are simultaneously severed under a constant normal load and a certain shear velocity. Despite depicting the majority of the physical characteristics of an actual earthquake, this is a simplification of the actual event. In addition, the data is periodic with a realistic behavior and contains earthquakes that occur sporadically.

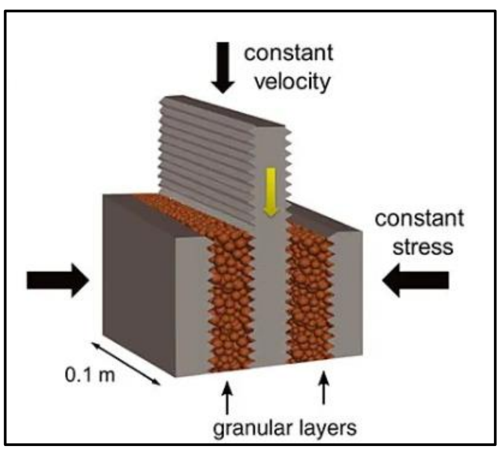


Fig.2.Simulation of earthquakes

3.3 <u>Storing the Data</u>:

Over 600 million rows of data make up the data dimension, which is fairly huge. The two columns in the train dataset are-

- accoustic_data: the acoustic signal measured during the investigation.
- time to failure: the amount of time until a failure occurs.

The acoustic data appears to have complex oscillations with changing amplitudes, as seen in the following plot. We can see that there is an increase in the amplitude of the acoustic data Just before each failure.

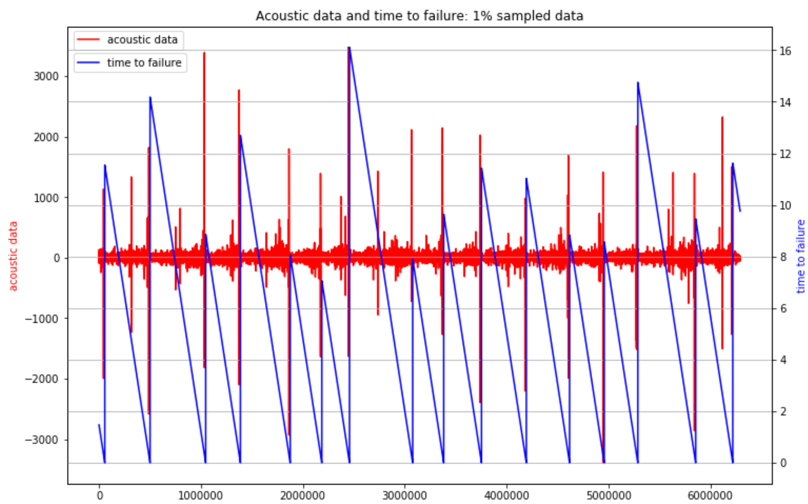


Fig.3.Acoustic data and time to failure

It is clear that the significant oscillation before the breakdown is not happening at the very last second. Intense oscillation trains can be found both before and after the major one, as well as some oscillations with lesser peaks. After a few brief oscillations, the failure occurs.

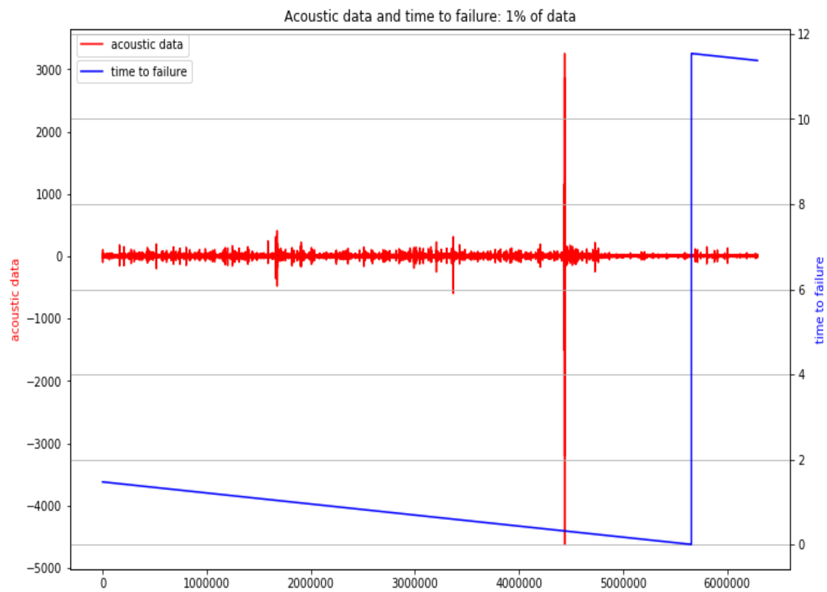


Fig.4.Time to failure

The first ten rows of the data are shown in Table 2.

	acoustic_data	time_to_failure
0	12	1.4690999984741211
1	6	1.4690999984741211
2	8	1.4690999984741211
3	5	1.4690999984741211
4	8	1.4690999984741211
5	8	1.4690999984741211
6	9	1.4690999984741211
7	7	1.4690999984741211
8	-5	1.4690999984741211
9	3	1.4690999984741211

Table 2:Ten rows of data

CHAPTER IV: ANALYSIS AND DISCUSSION

Feature engineering:
Our dataset contains many columns but we need to choose the necessary features to forecast earthquakes. In that case, a minimum number of necessary features has been taken which can contribute to prediction. Each test section is 150,000. We divided train data into segments of the same dimensions as the test sets. We develop further aggregation features that are based on the segments.

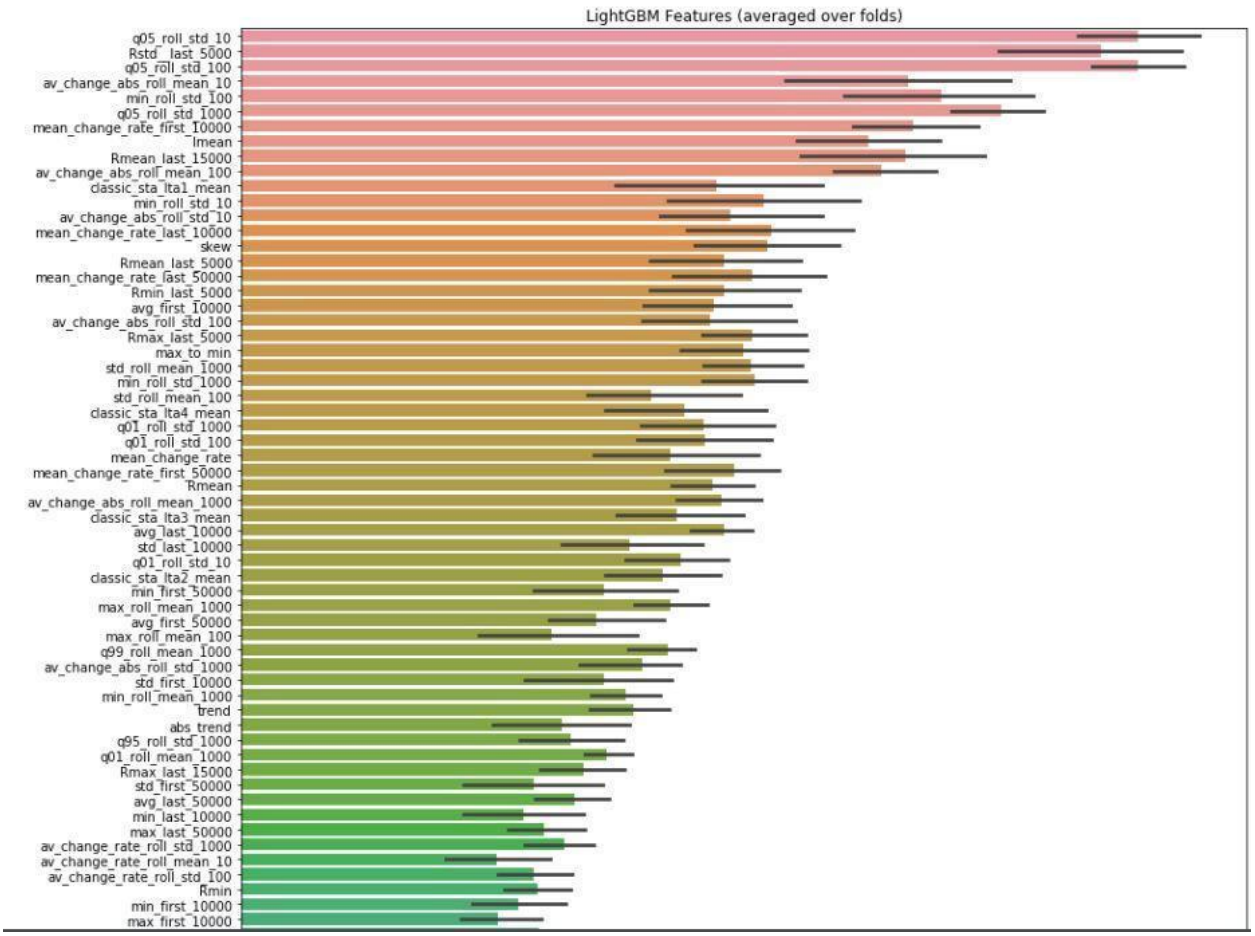
Process train file:
Model is trained by choosing batch_size = 128 and 10 epochs for every model. In order to determine the most efficient model, we apply scaling training data where scaling helps to normalize feature range and thus we obtain the following table.

	mean	std	max	min	Rmean
0	1.424140499795022	-0.170213582943171	-0.218193508692627	0.193218185209325	1.199316205489370
1	0.805716032556442	0.004734017014316	0.063936007992510	-0.018037271219829	0.078885810556771
2	1.511155259373638	0.049252196732055	-0.086288799593083	0.163038834290874	0.078885810556771
3	1.494934375107487	0.043949637951170	0.122560323147863	-0.187796120136112	0.078885810556771
4	1.520242078970568	0.088495070341074	-0.067968701107035	0.087590456994748	1.199316205489370
5	1.538962489544185	-0.131053986788160	-0.078960760198664	0.019686917428235	1.519439175470111
6	1.313015726042317	-0.101218785996422	-0.159569193537274	0.268666562505451	-0.721421614395085
7	-0.054824008985421	-0.081617152532347	-0.089952819290292	0.057411106076298	-0.241237159423971
8	0.774732321036825	0.145943916470089	0.016303751928785	-0.025582108949441	-0.561360129404713
9	0.825998647072221	0.040204715451199	-0.042320563226568	0.087590456994748	0.078885810556771

Table 3 Scale training data

Both training and test data go through the same procedure. As a result, we read the submission file and generate the test file. We have chosen n fold cross-validation,where n= 5.

Feature importance:
Feature importance depicts the relevant features. Here is the feature importance depicted in the below figure 5.


LightGBM Features (averaged over folds)
q05_roll_std_10
Rstd__last_5000
q05_roll_std_100
av_change_abs_roll_mean_10
min_roll_std_100
q05_roll_std_1000
mean_change_rate_first_10000
Imean
Rmean_last_15000
av_change_abs_roll_mean_100
classic_sta_lta1_mean
min_roll_std_10
av_change_abs_roll_std_10
mean_change_rate_last_10000
skew
Rmean_last_5000
mean_change_rate_last_50000
Rmin_last_5000
avg_first_10000
av_change_abs_roll_std_100
Rmax_last_5000
max_to_min
std_roll_mean_1000
min_roll_std_1000
std_roll_mean_100
classic_sta_lta4_mean
q01_roll_std_1000
q01_roll_std_100
mean_change_rate
mean_change_rate_first_50000
Rmean
av_change_abs_roll_mean_1000
classic_sta_lta3_mean
avg_last_10000
std_last_10000
q01_roll_std_10
classic_sta_lta2_mean
min_first_50000
max_roll_mean_1000
avg_first_50000
max_roll_mean_100
q99_roll_mean_1000
av_change_abs_roll_std_1000
std_first_10000
min_roll_mean_1000
trend
abs_trend
q95_roll_std_1000
q01_roll_mean_1000
Rmax_last_15000
std_first_50000
avg_last_50000
min_last_10000
max_last_50000
av_change_rate_roll_std_1000
av_change_rate_roll_mean_10
av_change_rate_roll_std_100
Rmin
min_first_10000
max_first_10000

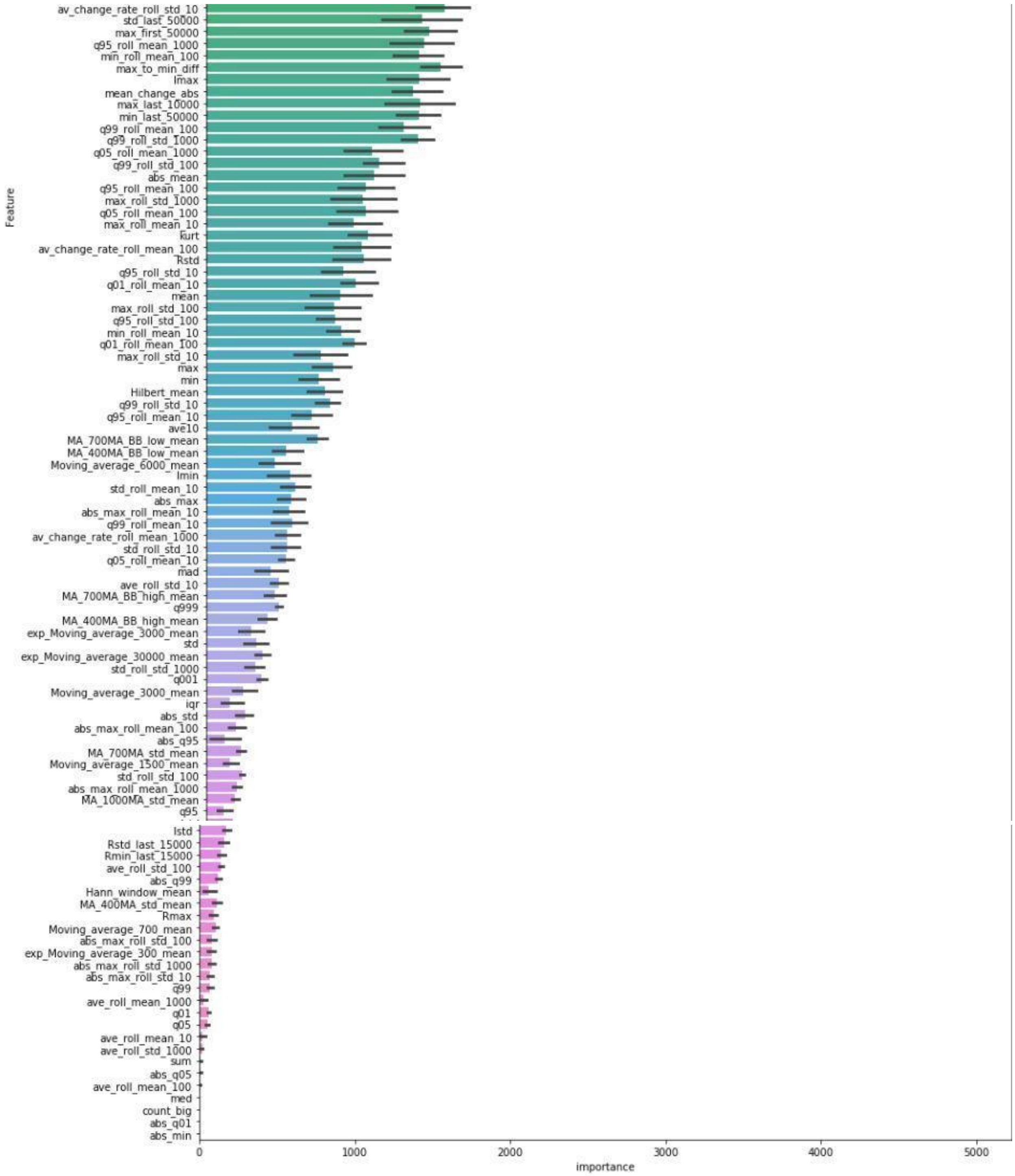


Fig.5.Feature importance

Thus,in the submission file, we are setting the predicted time to failure.

Visualization:
On the global map, every earthquake from the database is displayed. It depicts areas where earthquakes will occur more often.

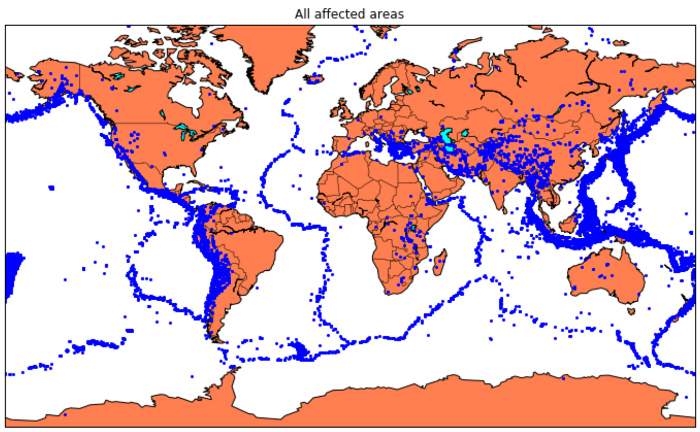


Fig.6.vulnerable earthquake area

Splitting the Data:
Our dataset file is in .CSV format. 20% of the dataset is the test dataset, while 80% is the training dataset. Therefore, all of our models were trained on Google Colab and have TensorFlow backend. We read the dataset and take the necessary features. We convert the given Dateand and Time in the dataset to numeral. We marked the affected area as blue and the coral area as aqua.

Machine learning and deep learning model:

Several decision trees are combined as part of the Random Forest ensemble learning process to generate a more accurate and reliable model. Used to forecast earthquakes, Random Forest employs bagging, which performs well when dealing with a large number of input characteristics and can manage absent data adequately.

Random Forest is a machine learning ensemble technique used for classification and regression tasks. It combines multiple decision trees, each with its own strengths and weaknesses, to produce a more accurate model. Bagging, or Bootstrap Aggregating, is a technique used to create multiple subsets of training data, reducing overfitting and increasing model accuracy. Random Forest is particularly useful for datasets with large input features and handling missing data without imputation. It has been applied to geophysical prediction tasks, such as forecasting earthquakes based on historical data.

We have used grid search for random forest which helps to define a parameter grid that includes a set of parameters into a matrix. Thus, our built model is trained on parameters. It is obvious that hypermeter tuning can lead to better performance of the model.

K-fold is the most prevalent cross-validation technique. In this method, the dataset is divided into training and testing sets, and the training set is divided into k segments. Each pleat must be repeated k times. Thus, after all iterations, we calculate the average performance of all folds and discover the validation metric.

Consequently, the Random Forest Regression model is implemented. Unexpectedly, a prediction with a score greater than 80% can be presumed to be the best fit even if its predicted values are not the best fit. Thus, we might conclude that the predicted values are not as accurate as anticipated. Then, we construct the neural network to improve its performance.

ANNs are a popular machine learning algorithm used for forecasting earthquakes. ANNs are composed of layers of interconnected nodes which collaborate to forecast the outcome. Moreover, ANNs are often considered a "black box" algorithm, as it is difficult to interpret the results.Our neural networks have three dense layers, each with 16 nodes,2 nodes, and applied activation functions relu, relu, and softmax.

CHAPTER V: FINDINGS AND RECOMMENDATIONS

To develop an efficient model for earthquake prediction, we choose a machine learning model and deep learning models. It can be assessed by evaluation metrics accuracy and best fit score.

Therefore, accuracy is a remarkably popular method that is defined as a measure of correctness with respect to ground truth. It is a great measure to evaluate the correct prediction of all observations. [19]

In this project, we explored efficient machine learning and deep learning models such as Artificial Neural Network and Random Forest Regression which are developed to detect earthquakes. Here,the Artificial Neural Network achieved 92.42% in terms of accuracy. On the other hand, Random Forest Regression achieved a best fit score 87.49% .

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$$

Therefore, it is worth mentioning that our proposed approach ANN is much more efficient than Random Forest Regression as it consists of a dense layer of neural network.

In the future, we can discover more models such as Convolution Neural Network (CNN), Long short-term memory(LSTM), Gated recurrent units(GRU), Bi-directional long short term memory(Bi-LSTM), Support Vector Regression (SVR) and Bagging and Boosting. Also, we can explore many larger datasets and different types of data so that we can explore different types of earthquakes and geological conditions of our earth. Thus, different Machine learning and deep learning model prediction could lead to better detection of earthquakes in the foreseeable future.   It is noted that feature engineering plays a vital role in our prediction. It is important to choose the feature according to necessity and choosing algorithms is also vital. Also, we could use many more evaluation metrics such as precision and recall to get better results and performance. All these things are basically dependent on the preparation of the dataset. We should focus more on the preparation of the dataset which is essential for good research. Also, it should be noted that research is a continuous process. We need to go further work on it.We can also split the dataset into 70-30,60-40 for better output.

CHAPTER VI: CONCLUDING REMARKS

As our safety is dependent on the prevention and early prediction of earthquakes, researchers are trying to find new ways to predict them. In this paper, we presented Exploratory Data Analysis (EDA), feature engineering, machine learning, and deep learning models in order to predict earthquakes for saving life and financial loss. We express through extensive experiments that our best-performing model ANN achieved 92.42% accuracy and Random Forest Regression achieved a best fit score 87.49% . It is evident that experimental earthquake prediction is similar to real earthquakes. But there are still a lot of research opportunities to make it similar to real-life earthquakes. That time is very near when geophysicists and environmental researchers could come up with a solution to use this experiment in real life. In that case, it is relevant to understand the geology of the earth and the simulation experiment to predict earthquakes. This research field is the most demandable research field as the world is prone to earthquakes in recent times.

On the other hand, we can discover more models such as Bi-directional long short term memory(Bi-LSTM), Support Vector Regression (SVR), and Bagging and Boosting. Also, we can explore many larger datasets and address other feature engineering,cross validation processes too.

Reference

[1] Li, C., & Liu, X. (2016, May). An improved PSO-BP neural network and its application to earthquake prediction. In *2016 Chinese Control and Decision Conference (CCDC)* (pp. 3434-3438). IEEE.

[2] Field, E. H., Jordan, T. H., Page, M. T., Milner, K. R., Shaw, B. E., Dawson, T. E., ... & Thatcher, W. R. (2017). A synoptic view of the third Uniform California Earthquake Rupture Forecast (UCERF3). *Seismological Research Letters*, *88*(5), 1259-1267.

[3] Rhoades, D. A., Schorlemmer, D., Gerstenberger, M. C., Christophersen, A., Zechar, J. D., & Imoto, M. (2011). Efficient testing of earthquake forecasting models. *Acta Geophysica*, *59*, 728-747.

[4] Michael, A. J., & Werner, M. J. (2018). Preface to the focus section on the Collaboratory for the Study of Earthquake Predictability (CSEP): New results and future directions. *Seismological Research Letters*, *89*(4), 1226-1228.

[5] Schorlemmer, D., Gerstenberger, M. C., Wiemer, S., Jackson, D. D., & Rhoades, D. A. (2007). Earthquake likelihood model testing. *Seismological Research Letters*, *78*(1), 17-29.

[6] Asim, K. M., Idris, A., Martínez-Álvarez, F., & Iqbal, T. (2016, December). Short term earthquake prediction in Hindukush region using tree based ensemble learning. In *2016 International conference on frontiers of information technology (FIT)* (pp. 365-370). IEEE.

[7] Minaie, N. (2019). A Beginner‟ s Guide to Selecting Machine Learning Predictive Models in Python. *Towardsdatascience, Medium*, *16*.

[8] Kikuchi, M., & Kanamori, H. (1982). Inversion of complex body waves. *Bulletin of the Seismological Society of America*, *72*(2), 491-506.

[9] Tarantola, A., & Valette, B. (1982). Inverse problems= quest for information. *Journal of geophysics*, *50*(1), 159-170.

[10] Dettmer, J., Benavente, R., Cummins, P. R., & Sambridge, M. (2014). Trans-dimensional finite-fault inversion. *Geophysical Journal International*, *199*(2), 735-751..

[11]Nandwani, D. T., & Buradkar, V. Implementing Machine Learning Models in Predicting Earthquake. *Journal homepage: www. ijrpr. com ISSN*, *2582*, 7421.

[12] Li, W., Narvekar, N., Nakshatra, N., Raut, N., Sirkeci, B., & Gao, J. (2018, March). Seismic data classification using machine learning. In *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 56-63). IEEE.

[13] Asim, K. M., Idris, A., Martínez-Álvarez, F., & Iqbal, T. (2016, December). Short term earthquake prediction in Hindukush region using tree based ensemble learning. In *2016 International conference on frontiers of information technology (FIT)* (pp. 365-370). IEEE.

[14][Kumari, G. P. (2012). A Study of Bagging and Boosting approaches to develop meta-classifier. *Eng. Sci. Technol. An Int. J*, *2*(5), 850-855..]

[15][VINOD, M., REDDY, P. B., HEMALATHA, K., PRIYA, C. L., & NAVNEETHA, Y. (2022). CLASSIFYING EARTHQUAKE DAMAGE TO BUILDINGS USING MACHINE LEARNING. *Journal of Engineering Sciences*, *13*(7).]

16][Tanyuk, O., Davieau, D., South, C., & Engels, D. W. (2019). Machine learning predicts aperiodic laboratory earthquakes. *SMU Data Science Review*, *2*(2), 11.]

17][Vasyura-Bathke, H., Dettmer, J., Steinberg, A., Heimann, S., Isken, M. P., Zielke, O., ... & Jónsson, S. (2020). The Bayesian earthquake analysis tool. *Seismological Research Letters*, *91*(2A), 1003-1018.]

[18] https://www.kaggle.com/datasets/usgs/earthquake-database

[19] Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

Appendix

Python Codes Used for The Analysis:

#Import the necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import os
print(os.listdir("../input"))
['database.csv']
```

#Read the data from csv and also columns

```
data = pd.read_csv("../input/database.csv")

data.head()

data.columns


Index(['Date', 'Time', 'Latitude', 'Longitude', 'Type', 'Depth', 'Depth Error',

       'Depth Seismic Stations', 'Magnitude', 'Magnitude Type',

       'Magnitude Error', 'Magnitude Seismic Stations', 'Azimuthal Gap',

       'Horizontal Distance', 'Horizontal Error', 'Root Mean Square', 'ID',

       'Source', 'Location Source', 'Magnitude Source', 'Status'],

      dtype='object')
```

#Figure out the main features from earthquake data and create a object of that features, namely, Date, Time, Latitude, Longitude, Depth, Magnitude.

```
data = data[['Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude']]

data.head()
```

#Convert given Date and Time to Unix time which is in seconds and a numeral.

```
import datetime

import time

timestamp = []

for d, t in zip(data['Date'], data['Time']):

    try:

        ts = datetime.datetime.strptime(d+' '+t, '%m/%d/%Y %H:%M:%S')

        timestamp.append(time.mktime(ts.timetuple()))

    except ValueError:

        # print('ValueError')

        timestamp.append('ValueError')


timeStamp = pd.Series(timestamp)

data['Timestamp'] = timeStamp.values
```

linkcode

```
final_data = data.drop(['Date', 'Time'], axis=1)

final_data = final_data[final_data.Timestamp != 'ValueError']

final_data.head()
```

Visualization

Here, all the earthquakes from the database are visualized on to the world map which shows clear representation of the locations where frequency of the earthquake will be more.

```
from mpl_toolkits.basemap import Basemap

m = Basemap(projection='mill',llcrnrlat=-80,urcrnrlat=80,
llcrnrlon=-180,urcrnrlon=180,lat_ts=20,resolution='c')
```

```
longitudes = data["Longitude"].tolist()

latitudes = data["Latitude"].tolist()

#m = Basemap(width=12000000,height=9000000,projection='lcc',

            #resolution=None,lat_1=80.,lat_2=55,lat_0=80,lon_0=-107.)

x,y = m(longitudes,latitudes)


fig = plt.figure(figsize=(12,10))
plt.title("All affected areas")
m.plot(x, y, "o", markersize = 2, color = 'blue')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
m.drawmapboundary()
m.drawcountries()
plt.show()
```

Splitting the Data

Firstly, split the data into Xs and ys which are input to the model and output of the model respectively. Here, inputs are TImestamp, Latitude and Longitude and outputs are Magnitude and Depth. Split the Xs and ys into train and test with validation. Training dataset contains 80% and Test dataset contains 20%.

```
X = final_data[['Timestamp', 'Latitude', 'Longitude']]

y = final_data[['Magnitude', 'Depth']]
```

```
from sklearn.cross_validation import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, X_test.shape)
```

Here, we used the RandomForestRegressor model to predict the outputs, we see the strange prediction from this with a score above 80% which can be assumed to be best fit but not due to its predicted values.

```
from sklearn.ensemble import RandomForestRegressor
reg = RandomForestRegressor(random_state=42)
reg.fit(X_train, y_train)
reg.predict(X_test)

best_fit.score(X_test, y_test)

from keras.models import Sequential
from keras.layers import Dense

def create_model(neurons, activation, optimizer, loss):
    model = Sequential()
    model.add(Dense(neurons, activation=activation, input_shape=(3,)))
    model.add(Dense(neurons, activation=activation))
    model.add(Dense(2, activation='softmax'))

    model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

    return model
```

#Define the hyperparameters with two or more options to find the best fit.

```
from keras.wrappers.scikit_learn import KerasClassifier

model = KerasClassifier(build_fn=create_model, verbose=0)

# neurons = [16, 64, 128, 256]
neurons = [16]
# batch_size = [10, 20, 50, 100]
batch_size = [10]
epochs = [10]
# activation = ['relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear', 'exponential']
activation = ['sigmoid', 'relu']
# optimizer = ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']
optimizer = ['SGD', 'Adadelta']
loss = ['squared_hinge']

param_grid = dict(neurons=neurons, batch_size=batch_size, epochs=epochs,
activation=activation, optimizer=optimizer, loss=loss)

grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1)
grid_result = grid.fit(X_train, y_train)

print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
```

```
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

#The best fit parameters are used for same model to compute the score with training data and testing data.

```
model = Sequential()
model.add(Dense(16, activation='relu', input_shape=(3,)))
model.add(Dense(16, activation='relu'))
model.add(Dense(2, activation='softmax'))

model.compile(optimizer='SGD', loss='squared_hinge', metrics=['accuracy'])

model.fit(X_train, y_train, batch_size=10, epochs=20, verbose=1,
validation_data=(X_test, y_test))

[test_loss, test_acc] = model.evaluate(X_test, y_test)
print("Evaluation result on Test Data : Loss = {}, accuracy = {}".format(test_loss,
test_acc))

4682/4682 [==============================] - 0s 39us/step
Evaluation result on Test Data : Loss = 0.5038455790406056, accuracy =
0.9241777017858995

model.save('earthquake.h5')
```