INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-1563)
Available online at: https://www.ijariit.com

Earthquake Prediction using Machine Learning

Madhumita Kulkarni
madhukulkarni158@gmail.com
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra

Chaitralee Mulay
chaitralee99@gmail.com
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra

Swarali Marathe
marathe.swarali99@gmail.com
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra

Prashant Itankar
pyi.cm.dmce@gmail.com
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra

ABSTRACT

Earthquake forecasting is one of the most significant issues in Earth science because of its devastating consequences. Current earthquake forecasting scientific studies focus on three key points: when the disaster will occur, where it will occur and how big it will be. Scientists can predict where an earthquake will occur but it has been a major challenge to predict when it will occur and how powerful it will be. When the earthquake happens, we must fix this project. Specifically, you predict the time left before laboratory earthquakes occur from real-time seismic data that will have the potential to improve earthquake hazard assessments that could save lives and billions of dollars in infrastructure.

Keywords— *Ad boost classifier (Random Forest Classifier and Decision Tree Classifier), XG Boost, Machine Learning, Earthquake Dataset*

1. INTRODUCTION

Countless dollars and entire scientific careers have been dedicated to predicting where and when the next big earthquake will strike. But unlike weather forecasting, which has significantly improved with the use of better satellites and more powerful mathematical models, earthquake prediction has been marred by repeated failure due to highly uncertain conditions of earth and its surroundings. Now, with the help of artificial intelligence, a growing number of scientists say changes in the way they can analyze massive amounts of seismic data can help them better understand earthquakes, anticipate how they will behave, and provide quicker and more accurate early warnings. This helps in hazard assessments for many builders and real estate business for infrastructure planning from business perspective. Also, many lives can be saved through early warning. This project aims a simple solution to above problem by predicting or forecasting likely places to have earthquake in next 7 days. For user-friendly part, this project has a web application that extracts live data updated every minute by USGS.gov and predicts next likely place world-wide to get hit by an earthquake, hence a real-time solution is provided.

2. PROBLEM STATEMENT AND APPROACH TO SOLUTION

Anticipating seismic tremors is a pivotal issue in Earth science because of their overwhelming and huge scope outcomes. The goal of this project is to predict where likely in the world and on what dates the earthquake will happen. Application and impact of the project includes potential to improve earthquake hazard assessments that could spare lives and billions of dollars in infrastructure and planning. Given geological locations, magnitude and other factors in dataset from https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php for 30 days past which is updated every minute, we predict or forecast 7 days' time in future that is yet to come, the places where quake would likely happen. Since this is event series problem type, proposed solution in this project follows considering binary classification of earthquake occurance with training period includes fixed rolling window moving averages of past days while for which its labels, a fixed window size shifted ahead in time. The model will be trained with Adaboost classifier (RandomForestClassifier and DecisionTreeClassifier) and compared with XGBoost based on AUC ROC score and recall score due to the nature of problem (i.e binary classification). Model with better AUC score and recall will be considered for web app that uses Google maps api to predict places where earthquake might occur.

3. METRICS

The problem addressed above is about binary classification, Earthquake occur = 1 and Earthquake not occur = 0 and with these predictions we try to locate co-coordinates corresponding to the predictions and display it on the google maps API web app. More suitable metrics for binary classification problems are ROC (Receiver operator

characteristics), AUC (Area Under Curve), Confusion matrix for Precision, recall, accuracy and sensitivity**.** One important thing about choosing metrics and model is what exactly we need from predictions and what not. To be precise, we need to minimize or get less False negative predictions since we don't want our model to predict as 0 or no earthquake occurred at particular location when in reality it had actually happend as this is more dangerous than the prediction case in which prediction is true/1 or earthquake occurred but in reality it did not because its always better safe than sorry!!!**.** Hence apart from roc_auc score, I have considered Recall as well for evaluation and model selection with higher auc_roc score and recall, where recall = (TP/TP+FN).

4. DATASET

Real time data that updates every minute on **https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php** for past 30 days. Below is the feature description of the dataset with 22 features and 14150 samples at the time of training.

- time: Time when the event occurred. Times are reported in milliseconds since the epoch
- latitude: Decimal degrees latitude. Negative values for southern latitudes.
- longitude: Decimal degrees longitude. Negative values for western longitudes.
- depth: Depth of the event in kilometers.
- mag: Magnitude of event occurred.
- magType: The method or algorithm used to calculate the preferred magnitude
- nst: The total number of seismic stations used to determine earthquake location.
- gap: The largest azimuthal gap between azimuthally adjacent stations (in degrees).
- dmin: Horizontal distance from the epicenter to the nearest station (in degrees).
- rms: The root-mean-square (RMS) travel time residual, in sec, using all weights.
- net: The ID of a data source contributor for event occured.
- id: A unique identifier for the event.
- types: A comma-separated list of product types associated to this event.
- place: named geographic region near to the event.
- type: Type of seismic event.
- locationSource: The network that originally authored the reported location of this event.
- magSource: Network that originally authored the reported magnitude for this event.
- horizontalError: Uncertainty of reported location of the event in kilometers.
- depthError: The depth error, three principal errors on a vertical line.
- magError: Uncertainty of reported magnitude of the event.
- magNst: The total number of seismic stations to calculate the magnitude of earthquake.
- status: Indicates whether the event has been reviewed by a human.

5. IMPLEMENTATION

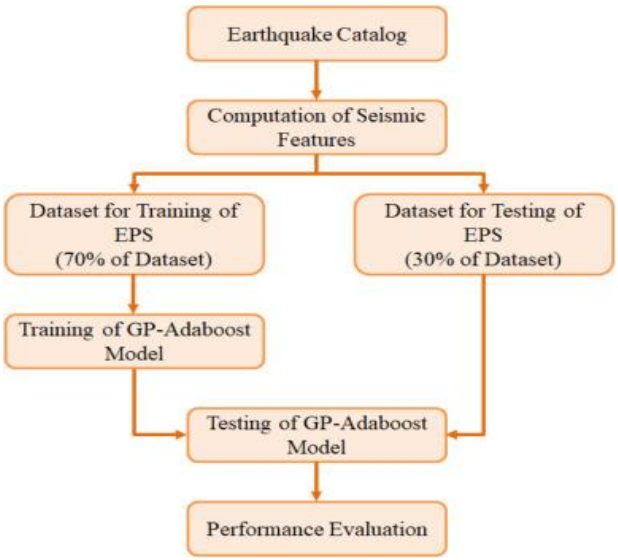

Fig. 1: Proposed System Design Flowchart

5.1 Exploratory Dataset Analysis

Null values Input to model from dataset has many important features to consider as time, latitude & longitude, depth of quake, magnitude, place, rest other features are error and non-supporting features for classification, below shows the null value counts for.

We can see lots of null values of certain features, but as part of prediction most of the features that address 'error' in measurement have missing values, thus for feature selection we consider only certain features in final data frame, hence I choose simply drop or ignore the null values.

```
time                0
latitude            0
longitude           0
depth               0
mag                 1
magType             1
nst              4190
gap              2987
dmin             3958
rms                 0
net                 0
id                  0
updated             0
place               0
type                0
horizontalError  5621
depthError          0
magError         5178
magNst           4844
status              0
locationSource      0
magSource           0
dtype: int64
```

Figure 2

Apart from features in dataset we focus on, I have done some feature Engineering based on some considerations on my model as follows:

Set rolling window size for future prediction based on past values with fixed window size in past.

I have created 6 new features based on rolling window size on average depth and average magnitude.

A final outcome 'mag_outcome' has been defined as target values and the output is considered as shifted values from set rolling window of past days eg: '7'. New features include : avg_depth, magnitude_avg for 22,15,7 days rolling window period for training.

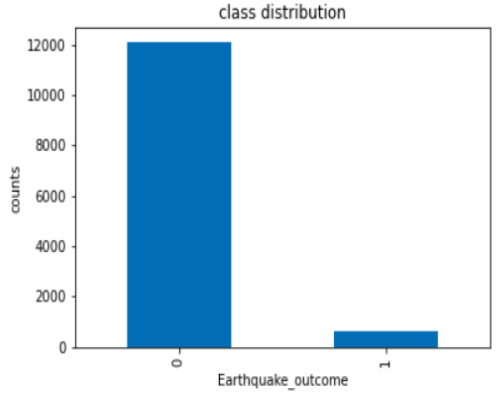

Figure 3

Accuracy is not the metric to use when working with an imbalanced dataset. We have seen that it is misleading. There are metrics that have been designed to tell you a more truthful story when working with imbalanced classes. such as collect more data, change metrics, resampling data, cross-validation dataset etc. For the project I have considered the metrics for treating this imbalance nature with-

(a) Confusion Matrix: A breakdown of predictions into a table showing correct predictions (the diagonal) and the types of incorrect predictions made (what classes incorrect predictions were assigned).
(b) Recall: A measure of a classifier's completeness
(c) ROC Curves: Like precision and recall, accuracy is divided into sensitivity and specificity and models can be chosen based on the balance thresholds of these values.

Moreover, the reason for choosing this metrics not only helps me improve class imbalance confirmation bias but also due to my nature of problem to be solved of earthquake prediction False negative must be penalized more.

5.2 XG Boost Model

With Estimators = 500, and learning rate =0.03 as we can see this significantly gives higher AUC score of almost 0.98 and also False negative = 37 which is similar Random Forest adaboost but xgboost has higher True positive and less False Positve compared to Random forest adaboost. i.e Recall score = 0.805 which is similar adaboost Random Forrest tree. But XGboost is really good at classifying positive and negative classes and also better aur_roc_score = 0.98193. We can see above that xgboost algorithm has higher auc score (0.9819) than adaboost decision tree and random forest, as it is evident from the ROC curve. Since Xgboost model having higher recall & auc_score than other algorithms, it can be considered more robust as it has ability to handle class imbalance with recall score, and deal good with False negative values and penalize it which is important for our task. i.e., reduce False Negative values. Hence, we consider xgboost for prediction of live data and deployment in the application.

6. CONCLUSION AND FUTURE SCOPE

Though XGboost model has given Higher roc_auc and better recall, I believe any work given always has some scope for improvement and in here we could also use RNN or LSTM for time series or rather event series forecasting. LSTMs have hidden memory cells that help in remembering and handling time series or event series data well. Moreover, for xgboost I have just used hyper parameters from already tuned Adaboost models, but we can also tune xgboost hyper parameter and find best parameters using GridSearchCV or RandomSearch.

- So far, the model looks good with xgboost as chosen model for predictions in web app haveing higher auc score and higher recall_score as I have explained under XGBoost result section why auc and recall score are chosen.
- Our main Aim is to predict whether earthquake will happen or not at a given day and place. So we definitely would not like the model with higher False Neagtive values, since its more dangerous to predict as no earthquake while in reality earthquake happend than predicting earthquake will happen given in reality it did not. We can allow False positive more than False negative.
- After seeing these comparision on auc_roc score, confusion matrix, and recall score, since all the above algorithm have given similar result with slightly different recall scores, Xgboost with FN=37 but with higher auc_score 0f 0.98 performs over-all better. Hence for webapplication deployment, I have chosen Xgboost as it also faster than adaboost.

Hence with all the mentioned implementation, the web application was successfully deployed and necessary project walktrhough can be accessed from Data and models directory.

7. ACKNOWLEDGMENT

We as a group want to extend our gratitude to all the sources of motivation. We would like to grab this opportunity to thank Prof. A. P. Pande, Head of Department who has been the constant driving force behind the completion of this project. We wish to express our heartfelt appreciation and deep sense of gratitude to our project guide Prof Prashant Itankar for his encouragement, invaluable support, timely help, lucid suggestions and excellent guidance which helped us to understand and achieve the project goal. His concrete directions and critical views have greatly helped us in successful completion of this work. We extend our sincere appreciation to all the professors for their valuable insights and tips during the designing of the project.

8. REFERENCES

[1] Numerical earthquake prediction' | The Seismological Society of China, Institute of Geophysics, China Earthquake Administration,2014 https://link.springer.com/article/10.1007/s11589-014-0082-z
[2] 'Machine Learning Predicts Laboratory Earthquakes' | https://www.researchgate.net/post/Introduction_to_earthquake_prediction_with_machine_learning
[3] 'Earthquake prediction with machine learning'
[4] https://towardsdatascience.com/earthquake- prediction-faffd7160f98
[5] 'Machine Learning Methods for Earthquake Prediction: a Survey'
[6] http://ceur-ws.org/Vol-2372/SEIM_2019_paper_31.pdf

