Random Forests: An Ensemble Method Approach to Earthquake Magnitude Forecasting

Ilyas Khaja

1 Introduction

Earthquake prediction remains a challenging problem due to the scarcity of data points as well as the inherently unpredictable nature of seismic events. In the past, statistical models such as the Epidemic-Type Aftershock Sequence (ETAS) model and physics based models like the Coloumb stress transfer model have been used to great effect. However, these models tend to be limited in that they cannot capture hidden underlying patterns/variance in the data.

In recent years, researchers have explored the use of machine learning techniques by incorporating neural networks, support vector machines, LSTMS, ensemble methods etc. in an effort to advance past this barrier. In this report, we will explore the ensemble method of Random Forests (RF) in more detail and attempt to improve on previous implementations of this algorithm for the task of predicting earthquake magnitude in a particular region.

2 Methods/Case Study

The dataset we will be using for this report is the Significant Earthquakes dataset on Kaggle[(2)] which contains information on significant earthquakes (magnitude > 5) since 1900 and is regularly updated. Tectonic plate data has been sourced from the USGS Earthquake Hazards website[(3)]. The primary machine learning module we will be working with is python's scikit-learn.

2.1 Feature Analysis

Here, we will provide an overview of the features in the dataset as well as explain any additional features we have constructed or decided to omit.

time: ISO 8601 representation of when the earthquake occurred. Will be converted to Unix time in order to facilitate calculating time intervals, among other things.

latitude and longitude: Geographical coordinates of the earthquake's epicenter.

depth: How deep the earthquake's point of origin is, measured in kilometers.

magnitude: The magnitude of the earthquake. Higher magnitudes correspond to stronger earthquakes. The target variable of our model.

magType: The scale used to report the earthquake's magnitude. We will be normalizing these to a single type in order to maintain consistency.

type: The type of seismic event, e.g an earthquake, quarry blast, or explosion. We will only be looking at earthquakes in this report.

horizontal/depth/magError: The calculated error in the latitudinal/longitudinal location, depth, and magnitude estimates.

magNst: The number of seismic stations used to determine the earthquake's magnitude.

status: Indicates whether the earthquake information has been reviewed by a human analyst. We will only be using human reviewed data points in order to maintain the purity of the dataset.

timeSinceLast: Time since the last earthquake occured in the same region.

nearestTectPlate: Distance to the nearest tectonic plate.

We have omitted the following features: nst, gap, dmin and rms due to insufficient data for earlier earthquakes. net, id, and locationSource have also been omitted due to not being highly relevant to our task. We have engineered the following features: timeSinceLast and nearestTectPlate. timeSinceLast will aid in capturing the temporal significance of the earthquake while nearestTectPlate could capture spatial patterns.

2.2 Data Visualization

In this section, we will perform various statistical computations and visualizations in order to better understand the data at hand as well as potentially reveal any underlying patterns.

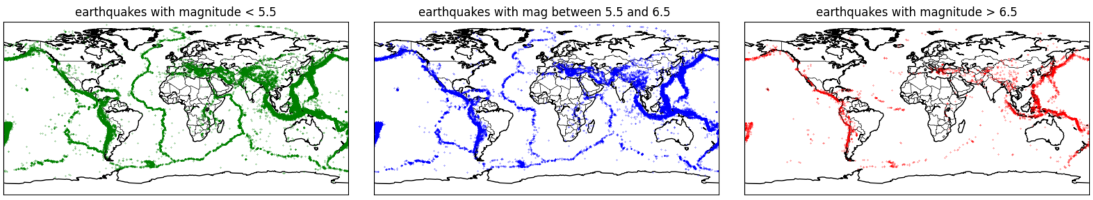

earthquakes with magnitude < 5.5
earthquakes with mag between 5.5 and 6.5
earthquakes with magnitude > 6.5

Figure 1: earthquake magnitudes

Figure 1 shows earthquakes of different magnitudes. The green/blue/red points represent magnitude ranges of m $<$ 5.5, 5.5 $<$ m $<$ 6.5, and m $>$ 6.5 respectively.

From figure 2, we can see that magnitude and depth have an inverse relationship. Figure 3 shows earthquakes with magnitude $>$ 6.0 since 1900, which is clearly trending upwards.

2.3 Model Architecture and Implementation

Figure 5 shows the architecture of our Random Forest model. We first began by transforming the input tensor X: $[lat, long, depth]$ to the higher dimensional space phi(X): $[lat, long, depth, unix - time, timeSinceLast, nearestTectPlate]$ by running subroutines to convert to unix time, compute time (in seconds) since the last earthquake within a given distance threshold, and to

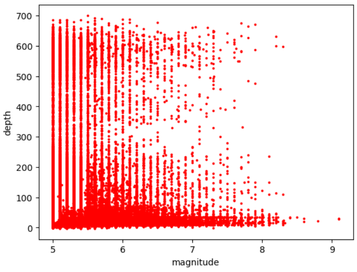


Figure 2: Magnitude vs depth plot

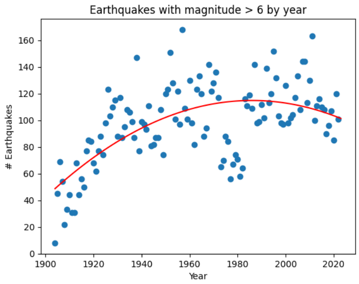
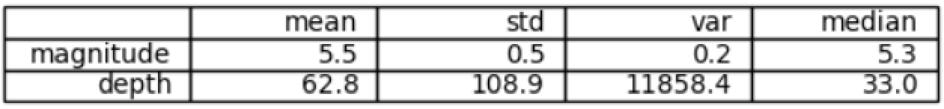


Figure 3: Earthquakes with magnitude > 6.0 since 1900



	mean	std	var	median
magnitude	5.5	0.5	0.2	5.3
depth	62.8	108.9	11858.4	33.0

Figure 4: magnitude/depth statistics

calculate the distance to the nearest tectonic plate based on the data from USGS Earthquake Hazards website, then adding these features to our input tensor.

Phi(X) was used to fit a RandomForestRegressor() instance from scikit-learn's ensemble class. A typical grid search algorithm was implemented to determine the optimal hyperparameters, using R^2 from sklearn.metrics as a measurement of model performance (since this a regression problem, our choices for loss are either mean-squared-error or R^2, and we have chosen R^2 in this case as it provides a more interpretable metric for evaluating model performance).

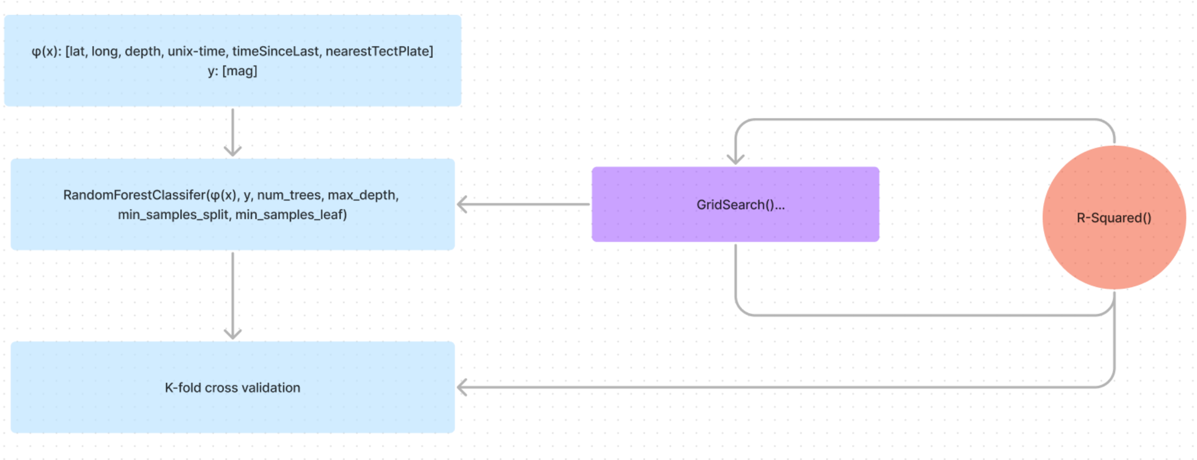

φ(x): [lat, long, depth, unix-time, timeSinceLast, nearestTectPlate]
y: [mag]
RandomForestClassifer(φ(x), y, num_trees, max_depth, min_samples_split, min_samples_leaf)
GridSearch()...
R-Squared()
K-fold cross validation

Figure 5: The architecture of the RF model

Table 1: R^2 score on training/validation sets

	R^2 score
Training set	0.466
Validation set	0.353

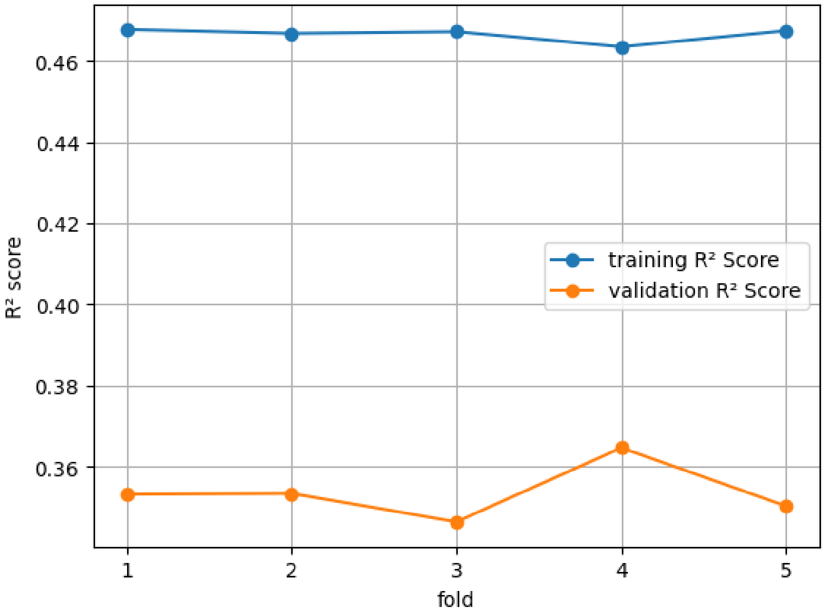


Figure 6: Training/validation R^2 curves vs k-folds

K-Fold Cross-Validation was incorporated to determine the average performance of the model. We plot the R^2 score vs each fold (Figure 6) then computed the average score for both training/validation across folds as seen in Table 1

3 Results and Discussion

We obtained an average R^2 of 0.353 by implementing a grid search on the num-trees, max-depth, min-samples-split, min-samples-leaf hyperparameters. We could not use the constructed 'timeSinceLast' feature due to insufficient compute. This is a substantial improvement over the R^2 found in Mondol, Manaswi. (2021). Analysis and Prediction of Earthquakes using different Machine Learning techniques[(6)], which reported a R^2 score in the range of -.125 to 0 for magnitude predictions.

We found that the 'nearestTectPlate' feature made negligible difference in both the train/test regression scores, so in future iterations this feature could be removed to simplify the model.

The results from K-fold cross validation showed an average R^2 score of approximately 0.466.

The difference between training and validation R^2 scores suggests that the model could be improved in terms of generalization to unseen data (i.e possibly suffers from overfitting). In future iterations of the model, techniques to reduce overfitting such as pruning or regularization can be implemented to alleviate this.

4 Conclusion

In this report, we aimed to develop a machine learning model that predicts the magnitude of earthquakes in specific regions, an area of research that has been highly challenging for a multitude of reasons. We utilized a dataset containing information on significant earthquakes worldwide, dating back to 1900, and employed a Random Forest algorithm to build our regression model. Over the course of development, we have learned that engineering features is a much more delicate task than it may seem, and adding features frivolously will not provide any additional representation power to the model.

Ultimately, it is imperative that we continue to improve and build upon these models, as more accurate and reliable earthquake prediction models have the potential to mitigate the severe consequences on human life, infrastructure and the environment.

Reference

(1) Lombardi, A.M. The epistemic and aleatory uncertainties of the ETAS-type models: an application to the Central Italy seismicity. Sci Rep 7, 11812 (2017). https://doi.org/10.1038/s41598-017-11925-3
(2) https://www.kaggle.com/datasets/usamabuttar/significant-earthquakes
(3) https://www.usgs.gov/programs/earthquake-hazards/google-earthtmkml-files
(4) Breiman, L. Random forests. Mach. Learn. 45, 1(Oct. 2001), 5–32
(5) Ridzwan, N.S.M., Yusoff, S.H.M. Machine learning for earthquake prediction: a review (2017–2021). Earth Sci Inform (2023). https://doi.org/10.1007/s12145-023-00991-z
(6) Mondol, Manaswi. (2021). Analysis and Prediction of Earthquakes using different Machine Learning techniques. 10.13140/RG.2.2.15085.10727.