DATS 6202
INDIVIDUAL REPORT

EARTHQUAKE AND TSUNAMI PREDICTION USING MACHINE LEARNING TECHNIQUES

—

Alejandra Mejia

TABLE OF CONTENT

Introduction

Although tsunamis are generally infrequent, their unpredictable nature makes them a potentially devastating natural hazard. To be able to develop accurate methods to detect and predict them quickly could become key to saving lives. Recent research has focused on the use of artificial intelligence (AI) algorithms (Cardiff University) and deep-learning models (Los Alamos National Laboratory) combined with real-time data.

However, the use of machine learning algorithms in this field still seems novel and a work in progress. Under this project, we aimed to add to the current research by exploring seismic research dataset which contains earthquake information for the past 22 years in order to develop a diverse classification model that can have the capability to predict whether or not an earthquake poses a significant risk of a tsunami.

Our approach focused on using historical data to test such predictions using six different machine learning models and determining their accuracy and performance. As a teammate, the individual task was focused in performing EDA analysis to be able to better select a machine learning approach, the use of geopandas to understand the dimensionality of the events and testing alternate machine learning approaches to the problem like random forest and AdaBoost. Given the slight unbalance of the dataset, metrics such as accuracy, precision, F-score and ROC_AUC were used to test the performance of the models. Although, under the group analysis the XGboost model seems to be the one with better performance, the AdaBoost also does a good job at predicting tsunamis.

The report is structured in four sections:

1. The first section provides a description of the model algorithms individually used and deep dives in the tasks developed individually.
2. Section number two describes the results of the models undertaken individually.
3. The third section provides a summary of the results and conclusions.
4. Section four details the code uniqueness
5. Finally, the last sections provide references

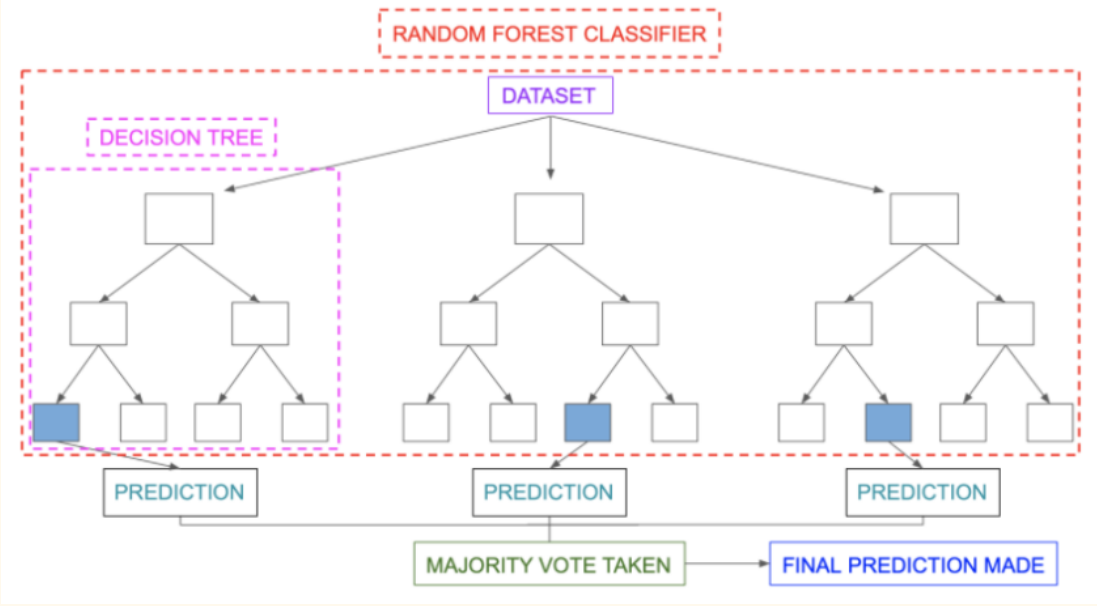
Description of Individual Work

Algorithm development

Random Forest

Given that decision trees and neural network algorithms can fall very easily in the overfitting trap, we decided to test out the ensemble method of Random Forest. Random Forests are known for their good scalability and ease of use, and given that they are able to average the high variance that individual trees suffer, they are able to build more robust and less susceptible to overfitting models (Figure1). Moreover, they are not affected by multicollinearity that much since every model sees a different set of data points. An additional advantage of using random forest relies in not requiring the need of choosing good hyperparameter values given ensemble models are robust to noise. Although not so common in practice optimization of hyperparameters of the Random Forest was undertaken, by using a randomized search. 70% of the reference data were randomly selected for the training stage and the other 30% were used for the testing. The proposed architecture was implemented on the machine learning framework Sklearn. Initializing with 100 n_estimators and max_depth of none (the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.)

Figure 1. Random Forest Classification Algorithm

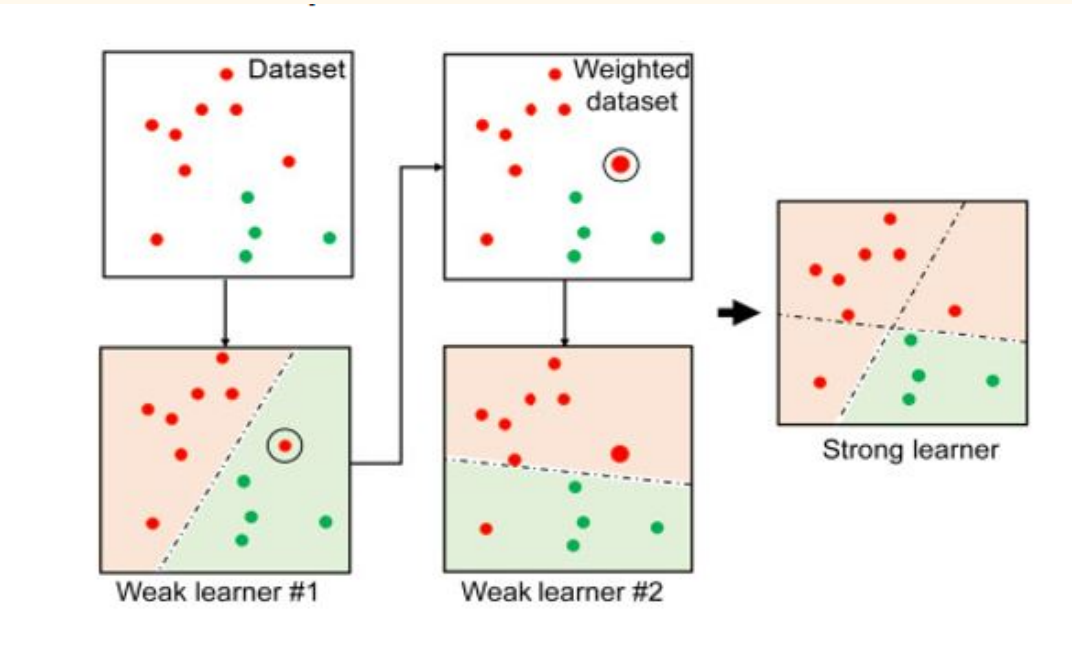


Source: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

AdaBoost

A more powerful approach than Random Forest suggested by the literature is the boosting approach (AdaBoost) which trains weak models and gives higher priority to the observations predicted incorrectly by previous models. Moreover, AdaBoost is often much better at making accurate classifications.The proposed architecture was implemented on the machine learning framework Sklearn. Initializing with 100 n_estimators and using the decision tree like the base_estimator. A grid search was also undertaken to find the best hyperparameters.

Figure 2. AdaBoost Classification Algorithm



Source: https://www.sciencedirect.com/topics/engineering/adaboost

The following section details the individual tasks performed as a team member to the overall project.

Task developed as team member

- Creation of Github Repository
- Gantt Chart and background information proposal
- Google tasks organization

- Data cleaning specifically clean country location based on longitude and latitude information
- Exploratory data analysis
- Plotly visualization of dynamic earthquake occurrence
- Report and ppt layout
- Testing of Random Forest and Adaboost Models
- Write up of the following group report sections:
    - Introduction,
    - data description
    - Introduction of Machine Learning Network and training algorithm section
    - Introduction of Experimental setup section
    - Random forest and AdaBoost Experimental Setup explanation
    - Random forest and AdaBoost Results
    - Summary table of models layout
    - Editing models confusion matrix and metrics
    - Contributed to conclusions and findings write up
- PPT EDA, Random Forest, AdaBoost sections
- Repository Read Me file

Results

The following section includes details of the results from the tasks developed individually

Description of the dataset

Data Overview

The dataset used in the current project was retrieved from Kaggle and contains data records of 782 different earthquakes. The amount of earthquakes recorded was large enough to perform machine learning techniques on. The variables contained in the database include more than 20 features and a binary tsunami variable that we will use as our target. The following section provides further details regarding the data dictionary.

Data Dictionary

Our dataset consisted of 18 variables detailed below:

1. Title: title name given to the earthquake
2. Magnitude: The magnitude of the earthquake
3. Date_time: date and time
4. cdi: The maximum reported intensity for the event range
5. mmi: The maximum estimated instrumental intensity for the event
6. alert: The alert level - “green”, “yellow”, “orange”, and “red”
7. tsunami: "1" for events in oceanic regions and "0" otherwise
8. sig: A number describing how significant the event is. Larger numbers indicate a more significant event. This value is determined on a number of factors, including: magnitude, maximum MMI, felt reports, and estimated impact
9. net: The ID of a data contributor. Identifies the network considered to be the preferred source of information for this event.
10. nst: The total number of seismic stations used to determine earthquake location.
11. dmin: Horizontal distance from the epicenter to the nearest station
12. gap: The largest azimuthal gap between azimuthally adjacent stations (in degrees). In general, the smaller this number, the more reliable the calculated horizontal position of the earthquake. Earthquake locations in which the azimuthal gap exceeds 180 degrees typically have large location and depth uncertainties
13. magType: The method or algorithm used to calculate the preferred magnitude for the event
14. depth: The depth where the earthquake begins to rupture
15. latitude / longitude: coordinate system by means of which the position or location of any place on Earth's surface can be determined and described
16. location: location within the country
17. continent: continent of the earthquake hit country
18. country: affected country

Prior to starting any analysis, some data cleaning was required in order to ensure that our machine learning methods would be able to produce robust results.

Exploratory Data Analysis

Given that our research question is focused on assessing the predictability of tsunamis based on earthquake occurrence, it became necessary to understand the distribution of earthquake magnitude in our sample. Especially, given that according to the National Weather Service, an earthquake must exceed magnitude 8.0 to generate a dangerous distant tsunami. From the data

we were able to depict that the most common magnitude in earthquakes is actually 6.5 and only a few number of instances a magnitude greater than or equal to 8.0 has been observed (Figure 1).

Figure 3. Distribution of earthquake magnitude in the sample

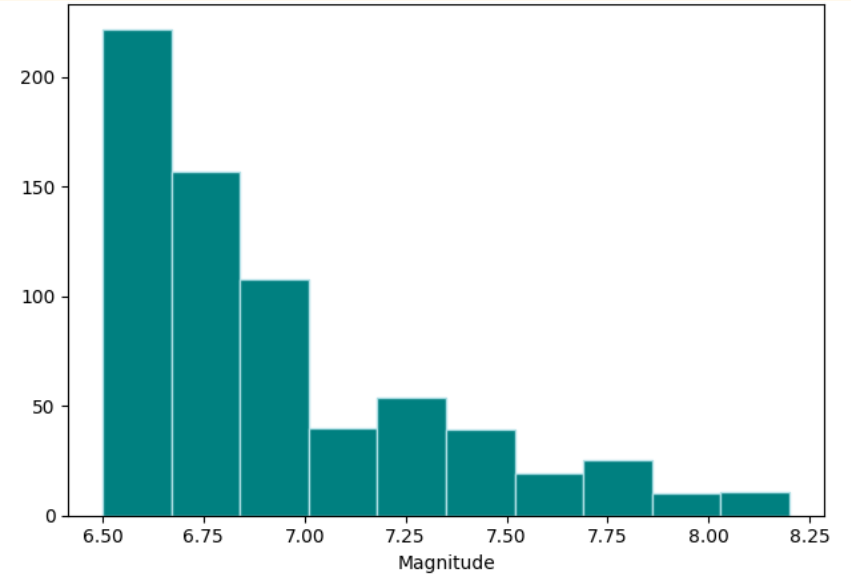
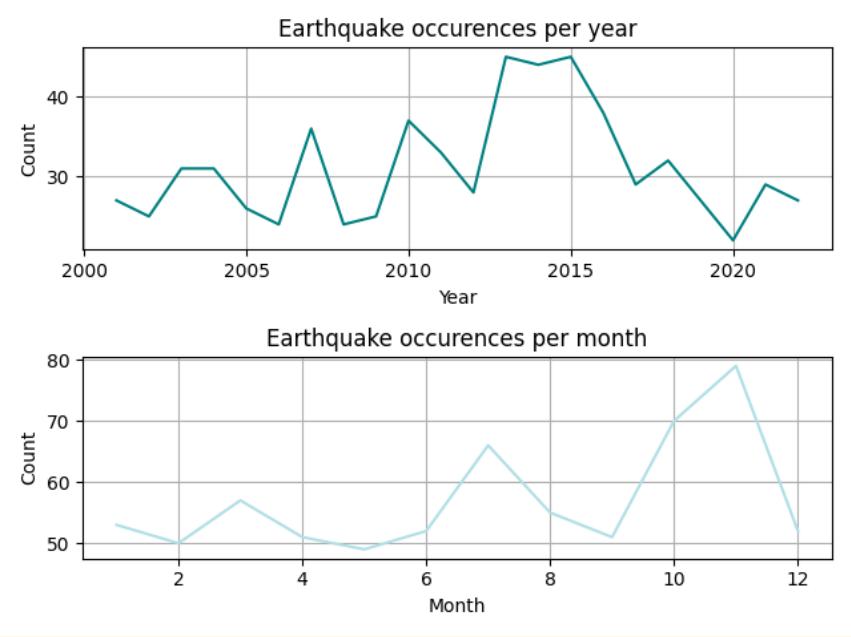


Figure 4. Earthquakes occurrence per year and month

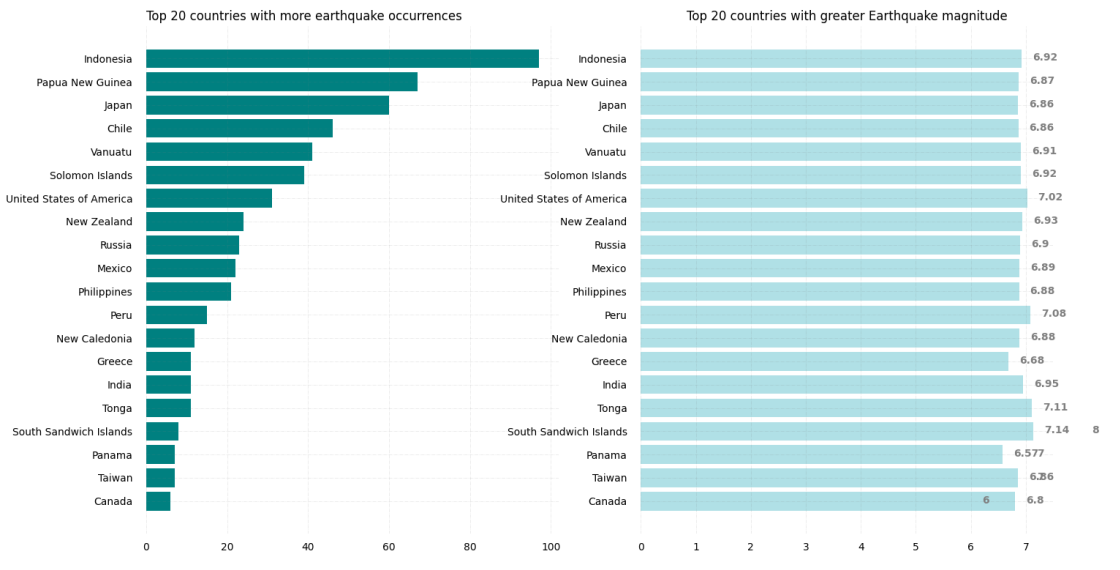


Similarly, we assess if the frequency of events followed a specific trend over the years. According to the available data the years 2013 and 2015 saw the most number of earthquakes, while the month of November seems to be the one with more recorded occurrences (Figure 2). This provides an interesting insight into adding variables in future research that could

be related to specific environmental factors that occur during those years or in that month, like sea level rise.

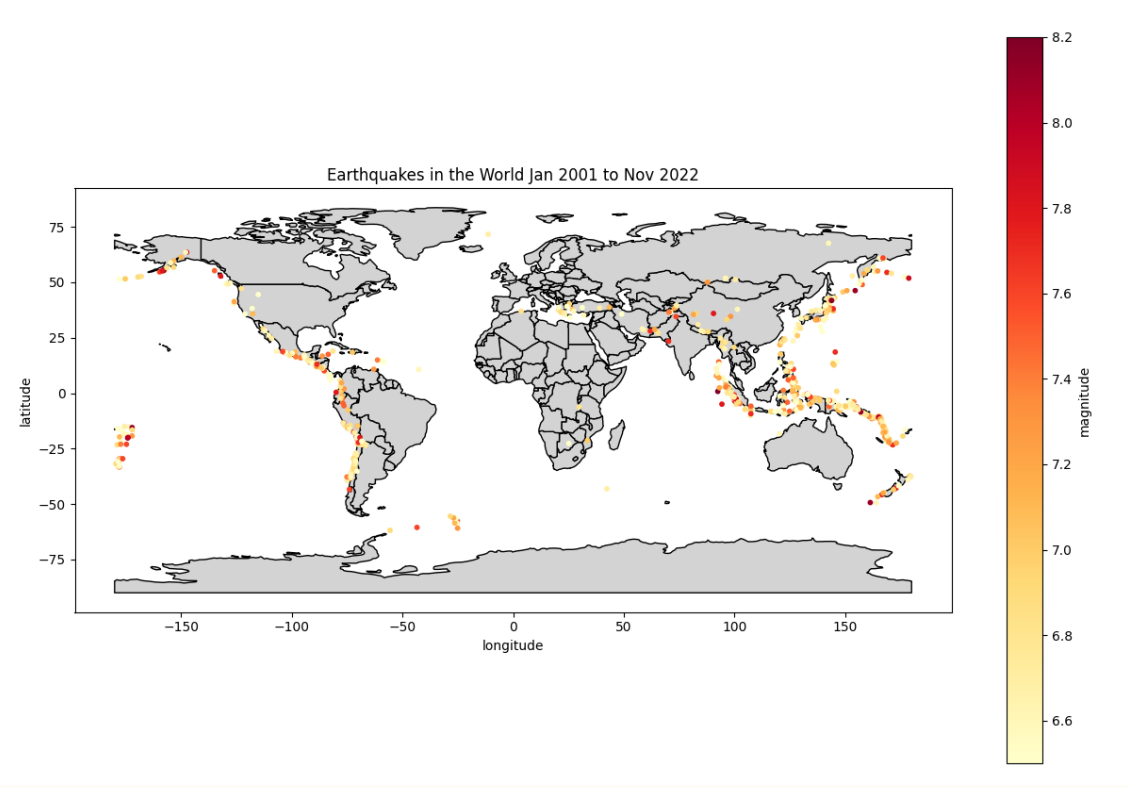
Moreover, our dataset included information on the location of the events. We could see from the data that the events are spread out in different continents , with Indonesia being the country with the highest number of earthquakes. However, it is relevant to notice that on average a higher number of earthquakes does not mean a higher magnitude.

Figure 5. Top 20 countries with more earthquakes and their magnitudes



Moreover, with the latitude and longitude data available we were able to map the reported earthquakes and be able to distinguish a pattern. A great majority of the earthquakes follow the path of the Ring of Fire, which is a path along the Pacific Ocean characterized by active volcanoes.

Figure 6. Earthquakes in the World Jan 2001 to Nov 2022



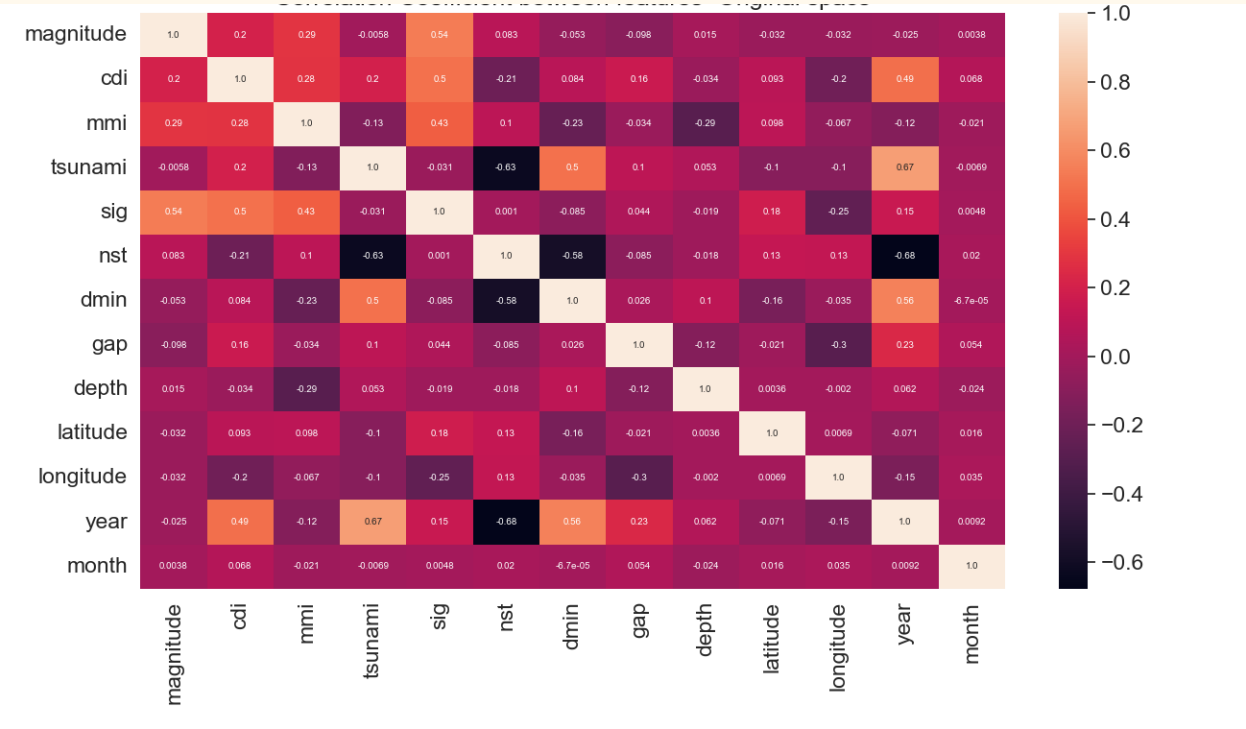
Additionally, we produced a pairplot that allowed us to understand the best set of features to explain the relationship between two variables and to determine how unbalanced our dataframe was. Specifically, given the low occurrence of tsunamis as previously mentioned, our dataset seems slightly unbalanced with 37 percent of tsunamis occurrences against 63 percent of occurrence with earthquakes but no tsunamis. Given this characteristic and that classification accuracy is almost universally inappropriate for imbalanced classification we will focus on sensitivity and precision metrics to better evaluate our models.

Figure 7. Pairplot

Moreover, in order to make sure our features were not correlated between them or with our target variable we produced a correlation matrix. We were able to confirm that the correlation

was fairly low, even in variables that based on the dataset dictionary description could been perceived as highly correlated (i.e: sig).

Figure 8. Correlation Matrix original space

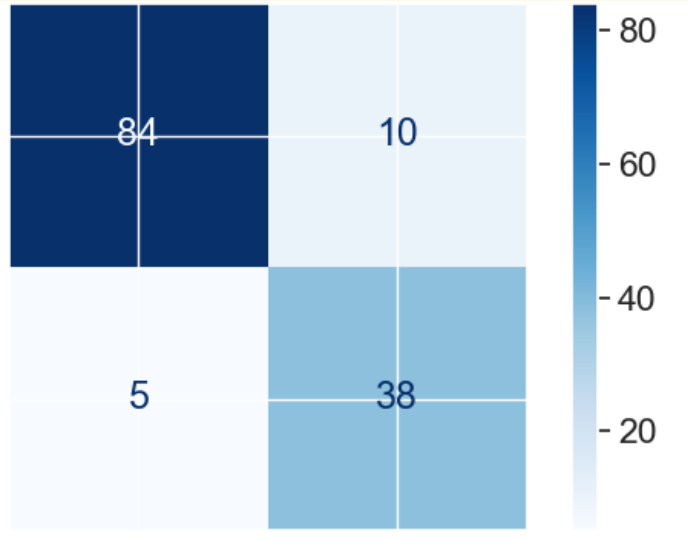


Modeling Alternative Machine Learning Algorithms

Random Forest

After performing the randomized search it was determined that the best hyperparameters for the model were a max depth of 19 and 495 as number of estimators. Under the best model scenario the performance metric shows a precision of 0.7966, a recall of 0.9216, a ROC_AUC of 0.8910 and aF1 score of 0.8545 (Table1). As it can be seen in the confusion matrix (Figure 9), even though the precision is high, given the unbalance in the data the previously mentioned metrics are a more accurate indicator of the model performance.

Figure 9. Confusion Matrix Random Forest

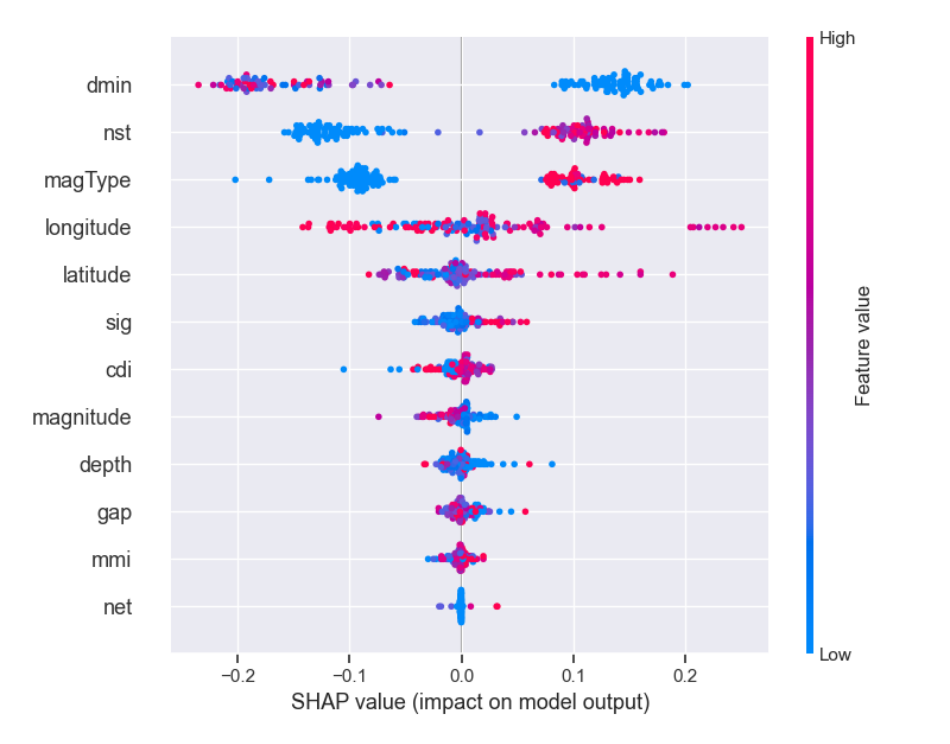


In addition to common performance metrics, it also became relevant to understand the feature importance of the model. In order to do so SHAP values were used. The usefulness of SHAP values relies on them being model-agnostic and being able to get an overview of which features are most important for a model. Figure 10 sorts features by the sum of SHAP value magnitudes over all samples, and uses SHAP values to show the distribution of the impacts each feature has on the model output. The color represents the feature value (red high, blue low). This reveals for example that a high dmin (horizontal distance from the epicenter to the nearest station) lowers the predicted tsunami probability.

Table1. Performance Metrics Random Forest

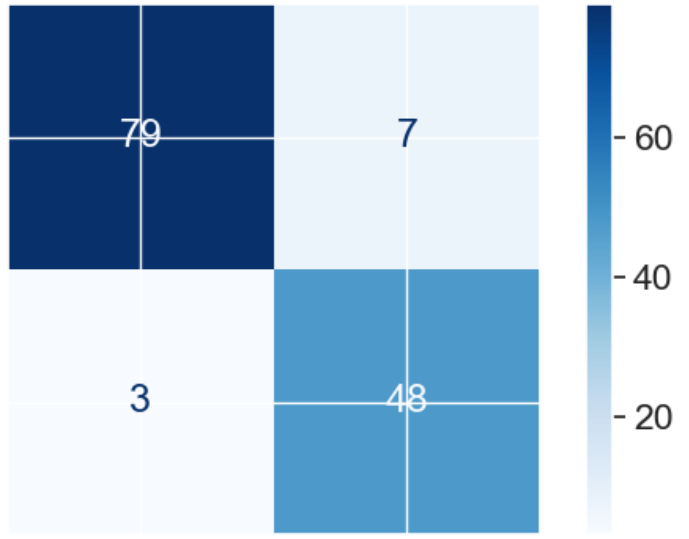
Precision	Recall	ROC_AUC	F1 Score
0.7966	0.9216	0.8910	0.8545

Figure10. Random Forest Feature Importance Shap values



AdaBoost

Figure 11. Confusion Matrix AdaBoost



In the case of the AdaBoost after performing the grid search it was determined that the best hyperparameters for the model were a learning rate of 0.1 and 100 as number of estimators. Under the best model scenario the performance metric shows a precision of 0.8727, a recall of 0.9412, a ROC_AUC of 0.9299 and a F1 score of 0.9057 (Table2). Similarly to the Random Forest model, it can be seen in the confusion matrix (Figure 10), that even though the precision is high, given the

unbalance in the data the previously mentioned metrics are a more accurate indicator of the model performance.

Table2. Performance Metrics AdaBoost

Precision	Recall	ROC_AUC	F1 Score
0.8727	0.9412	0.9299	0.9057

Summary and conclusions

As it can be seen in Table 3, from the two models implemented AdaBoost have the best performance metrics. However, the Random Forest feature importance provides us useful information regarding how the different features interact with our model that provide valuable information to tweak the model in future efforts. Moreover, for AdaBoost to be the right model we would need to ensure the quality of the dataset and be aware that this type of model is more sensitive to outliers and noise. However, it also becomes relevant to point out that this approach has numerous limitations, including a lack of expertise from the authors to develop more specialized deep learning algorithms, a small dataset that could be enhanced with more relevant features and real time information, verification of consistency and specificity of the variables in the dataset. Given so this research could be improved by the addition of geospatial data, sea-level rise information. Moreover, the approach used in this research could also be adopted to classify other hazards such as earthquakes, landslides, and floods. Additionally, this research could be expanded with the use of convolutional neural networks and geospatial analysis.

Table3. Summary Performance Metrics

Method	Precision	Recall	ROC_AUC	F1 Score
Random Forest	0.7966	0.9216	0.8910	0.8545
AdaBoost	0.8727	0.9412	0.9299	0.9057

Code Uniqueness calculation

Code lines= 300

Code from internet= 150

Modified code lines= 100

Own code lines= 150

Uniqueness formula

$$\frac{150-100}{150+150} = 17\%$$

References

Chamola, V., Hassija, V., & Gupta, S. (2020). *Disaster and Pandemic Management Using Machine Learning: A Survey*. NCBI. Retrieved April 29, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8768997/

Guha, S., K. Jana, R., & K. Sanyal, M. (2022, July 29). *https://www.sciencedirect.com/science/article/abs/pii/S2212420922004952*. Science Direct. Retrieved April 29, 2023, from https://www.sciencedirect.com/science/article/abs/pii/S2212420922004952

Kainthura, P., & Sharma, N. (2022, July 29). *Hybrid machine learning approach for landslide prediction, Uttarakhand, India*. https://www.nature.com/articles/s41598-022-22814-9. Retrieved April 29, 2023, from https://www.nature.com/articles/s41598-022-22814-9

Linardos, V., Drakaki, M., Tzionas, P., & Karnavas, Y. L. (2022). *Machine Learning in Disaster Management: Recent Developments in Methods and Applications*. MDPI. Retrieved April 29, 2023, from https://www.mdpi.com/2504-4990/4/2/20

NWS. (n.d.). *NWS JetStream - Tsunami Generation: Earthquakes*. National Weather Service. Retrieved April 29, 2023, from https://www.weather.gov/jetstream/gen_earth

scikit learn. (n.d.). *scikit.learn manual*. scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation. Retrieved April 29, 2023, from https://scikit-learn.org/stable/index.html

Yousefi, S., & Reza Pourghasemi, H. (2022, July 29). *A machine learning framework for multi-hazards modeling and mapping in a mountainous area*. https://www.nature.com/articles/s41598-020-69233-2. Retrieved April 29, 2023, from https://www.nature.com/articles/s41598-020-69233-2