

**Master of Data Analytics, University of Niagara Falls, Canada**

**DAMO-510-2 Predictive Analytics**

**Spring 2025**

**Assignment 2**

**Group 1:**

**Santiago Bravo Tobón - NF1011277**

**Gabriel Romero Gutierrez - NF 1012849**

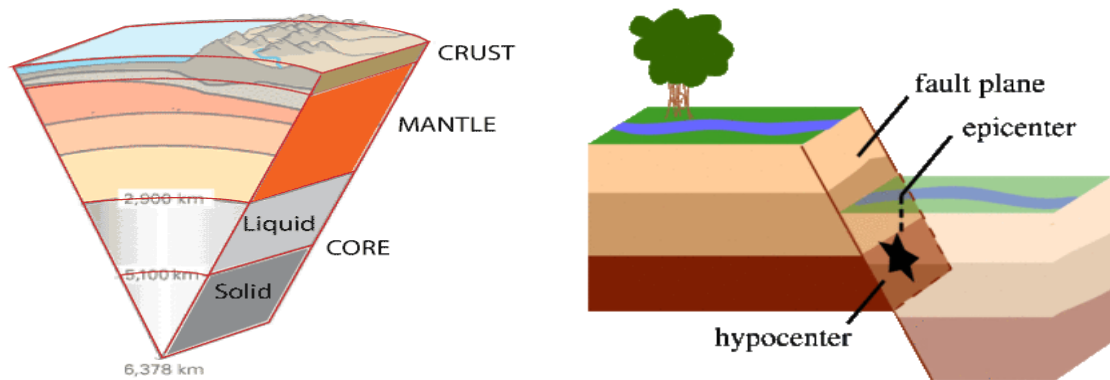
**Vanessa Rueda Nieto - NF1013949**

**Professor: Ph.D. Abbas Yazdinejad**

**June 15, 2025**

## Introduction

Natural events called earthquakes are caused by the abrupt release of energy stored within the Earth. This release is related to Tectonic plates movements at the boundaries of the Earth's lithosphere. When the accumulated stress in the rocks exceeds their resistance, a fracture occurs in the form of a fault, releasing energy in the form of seismic waves. The area under the surface of the earth where the earthquake starts is referred to as the hypocenter; the projection on the surface of the earth is known as the epicenter (Stein & Wysession, 2003).



Source: USGS - Earthquake Science Center

The creation of tsunamis is among the most hazardous consequences of earthquakes. A tsunami is a series of high-energy waves mostly produced by the abrupt disruption of the seabed. Not all earthquakes produce tsunamis; the seismic occurrence must have specific properties such as significant intensity, a shallow depth, and a rupture that causes vertical displacement of the seafloor. After the perturbation, destructive waves can devastate coastal regions thousands of kilometers from the epicenter (NOAA, 2025).

The forecasting of tsunami warnings is an essential element in natural disaster risk management given the catastrophic potential of tsunamis, especially in heavily populated coastal areas. Machine learning models such as regression trees achieve high forecasting accuracy since they describe insights and patterns that provide domain experts with fully explainable and interpretable results, which are a valuable support for environmental scientists and risk mitigation management (Cesario et al., 2024).

Models trained and tested on a historical earthquake database use variables including magnitude, depth, geographical location and other pertinent characteristics. The main goal of this project is to contribute to the development of automated tools that support early decision making for seismic and coastal hazards.

### **Objective**

The primary goal of this project is to develop a machine learning model able to forecast whether a seismic event will create a tsunami alert or not. Since tsunamis are a very complex natural event associated to different geological factors and not only to earthquakes per se, it is crucial to understand that the project will be limited to determine whether a tsunami warning would have been issued based on the recorded data of a seismic event that has already taken place, not the occurrence of the seismic event itself nor the actual generation of a tsunami.

Moreover, this study has an academic and exploratory emphasis; therefore, it aims to assess the practicality of employing machine learning models as a supplementary tool in natural disaster management rather than to substitute formal surveillance and early warning systems.

### **Data Collection**

The database used in this project was obtained from the Kaggle platform, a repository widely used for the development of data science and machine learning projects. The selected dataset contains detailed information on seismic occurrences recorded around the globe with variables including magnitude, depth, geographic position, time of the event, alert level, etc.

The original data was extracted from EveryEarthquake API, available from RapidAPI. Designed to enable geological studies, academic projects, and artificial intelligence applications, this API provides organized, up-to-date access to seismic information.

## Methodology

### Data Cleaning

The data cleaning process was carried out in seven stages to transform raw data imported from Excel into a structured, standardized, and analyzable dataset. These steps included identifying and converting data types, handling missing values, recalculating fields, and standardizing item names. The objective was to ensure data integrity and consistency for analysis and visualization. This process includes the following steps to guarantee a comprehensive dataset for this project:

#### *Step 1: Identifying useful data attributes*

The dataset contains 44 columns and 1,138 rows, including various information related to earthquake activity. After analyzing this data, we were able to identify which information was relevant to the purpose of the project and could provide realistic insights for the statistical tests and data models we plan to use. Based on this analysis, we proceeded to eliminate 25 columns that did not provide relevant information, such as URL links, updates, titles, networks, etc.

#### *Step 2: Identifying and changing data types to numeric*

After a detailed cleanup of the dataset, the next step was to identify and convert the categorical variables into binary format using encoding with values of 1 and 0. This approach allows the model to better understand the information. In this case, the variables that were converted were **alert** and **magType**, as shown in the image below.

alert	alert cd	magType	mb	mb_lg	MI	mw	mw_b	mwr	mw_w	ml
green	1	mwv	0	0	0	0	0	0	0	1
green	1	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
green	1	ml	0	0	0	0	0	0	0	1
green	1	mb	1	0	0	0	0	0	0	0
	0	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
green	1	mw	0	0	0	1	0	0	0	0
	0	ml	0	0	0	0	0	0	0	1
green	1	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
	0	mw	0	0	0	1	0	0	0	0
	0	ml	0	0	0	0	0	0	0	1
	0	ml	0	0	0	0	0	0	0	1
green	1	mw	0	0	0	1	0	0	0	0
	0	ml	0	0	0	0	0	0	0	1
	0	mw	0	0	0	1	0	0	0	0

### ***Step 3: Handling Missing or Incomplete Data***

The final step was to identify missing values and organize the dataset to ensure consistency in numerical formatting and proper spelling across all entries. We also verified that the dataset contained meaningful and accurate data. After reviewing the data, we observed that approximately 10% of the values were missing, primarily in the *country* column. These were successfully completed. For the numerical columns, we ensured that all values used the same punctuation and were in the correct numeric format.

### **Statistical test and Data Models**

To achieve the project's objective, multiple models were developed and evaluated to compare their performance and determine which one is the most suitable based on the selected variables.

### **Variables definition**

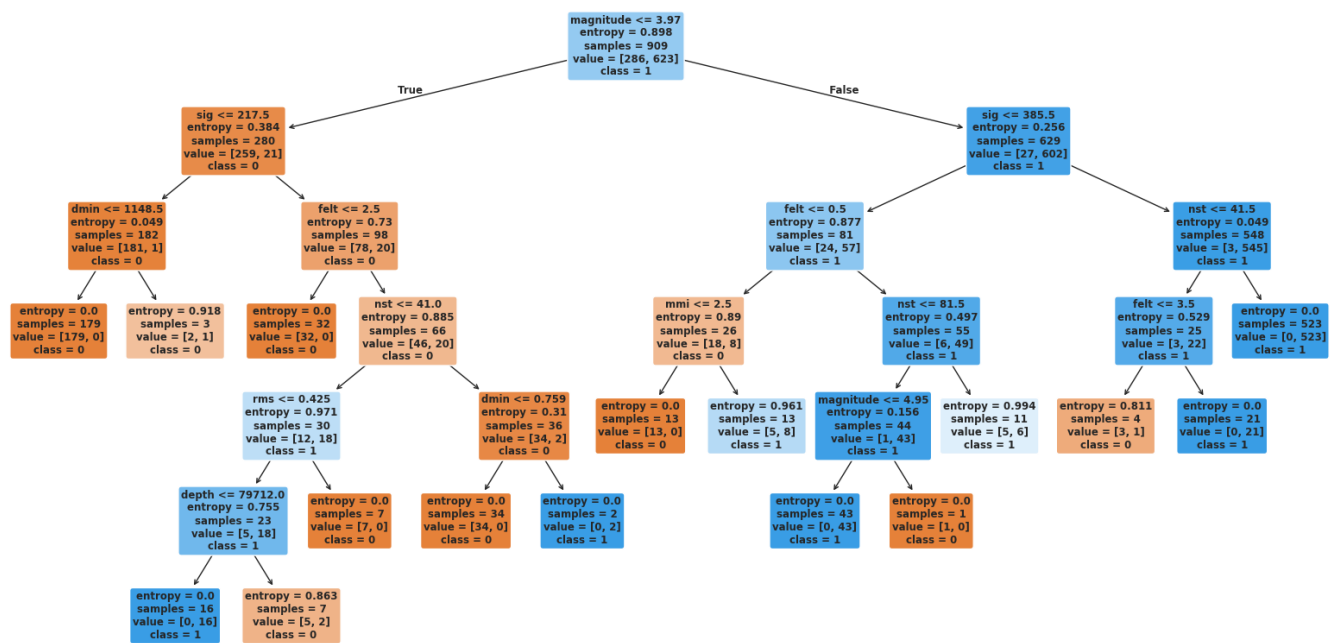
To build the models, the following variables from the dataset were selected as dependent and independent variables:

- Dependent variable [**Alert Code**: built variable based on Alert]
- Independent variables:

<b>Variable</b>	<b>Description</b>
<b>Magnitude</b>	The strength of the earthquake
<b>MagType</b>	Type Magnitude Scale Magnitude (one-hot encoding)
<b>Distance</b>	Distance from the nearest populated place in kilometers
<b>Depth</b>	Depth of the hypocenter in kilometers
<b>Gap</b>	Data gap between stations detecting the earthquake
<b>Rms</b>	Root mean square of signal, used to measure earthquake intensity
<b>Dmin</b>	Minimum distance to the earthquake event;
<b>Nst</b>	Number of seismic stations that recorded the event;
<b>Sig</b>	Significance of the earthquake, based on magnitude and impact
<b>Mmi</b>	Modified Mercalli Intensity, scale used to measure earthquake intensity;
<b>Cdi</b>	Community Determined Intensity, how strongly the event was felt;
<b>Felt</b>	Number of people who reported feeling the earthquake

## Model 1: Decision Tree

The decision tree model was created using the “**Decision Tree Classifier**” from **scikit-learn**. The model was configured with the following parameters: *criterion='entropy'*, *max\_depth=7*, *min\_samples\_split=20*, *class\_weight='balanced'*, and *random\_state=42*. These settings allowed the construction of a tree with controlled depth, balanced class handling, and consistent results across runs.



Decision Tree Structure

## Model 2: Logistic Regression

The logistic regression model was built using the “**Logistic Regression**” function from **scikit-learn**. The model was initialized with *class\_weight='balanced'* to account for class imbalance and *max\_iter = 10000* to ensure convergence during training. After training, the coefficients were extracted and transformed into odds ratios to better interpret the effect of each feature.

$$\text{Odds Ratio} = e^{\beta_i}$$

### ***Model 3: K-Nearest Neighbors (KNN)***

The KNN model was developed by calculating the optimal number of neighbors (k) where values from 1 to 30 were evaluated using *5-fold cross-validation*. For each k, the average validation accuracy was calculated, and the value that yielded the highest score was selected as the optimal k. With the best k determined, a final KNN model was trained, and predictions were made on the test set.

$$P\left(\text{Class} = \frac{c}{x}\right) = \frac{\text{Number of neighbors with class } c}{k}$$

Where  $P(\text{Class}=c|x)$  is the probability that instance x belongs to class c and k is the total number of neighbors considered.

### ***Model 4 & 5: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)***

The LDA and QDA models were created using “**Linear Discriminant Analysis**” and “**Quadratic Discriminant Analysis**” respectively from *scikit-learn*. The models were trained, and predictions were made on the test set. Both predicted class labels and class probabilities were generated in all cases.

### **Evaluation Metrics**

Performance metrics including *accuracy, precision, recall, F1 score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC)* were used to compare and evaluate the final performance of each model. In all models, 80% of the data was assigned to the training set and 20% to the testing set; They were also evaluated on a test set (20% hold-out).

## Model Results

### Preliminary Data Assessment

Before fitting any classification models, an initial exploratory analysis was conducted to assess the suitability of the dataset. A class balance check was performed to verify whether the target variable (alert code) was evenly distributed. Although slight imbalance was observed, the distribution did not warrant complex resampling techniques; instead, class weights were adjusted in some models (e.g., Logistic Regression, Decision Tree) to mitigate any bias.



In addition, a correlation matrix was generated to examine multicollinearity among the numerical features. While some moderate correlations were present (e.g., between *cdi*, *mmi*, and *sig*), no features showed a level of collinearity that would require removal or dimensionality reduction. This step ensured that input variables would not distort the training process, particularly for models sensitive to multicollinearity like Logistic Regression and LDA.

### Performance Comparison

The following table summarizes the performance of each model in accordance with the previously selected metrics:

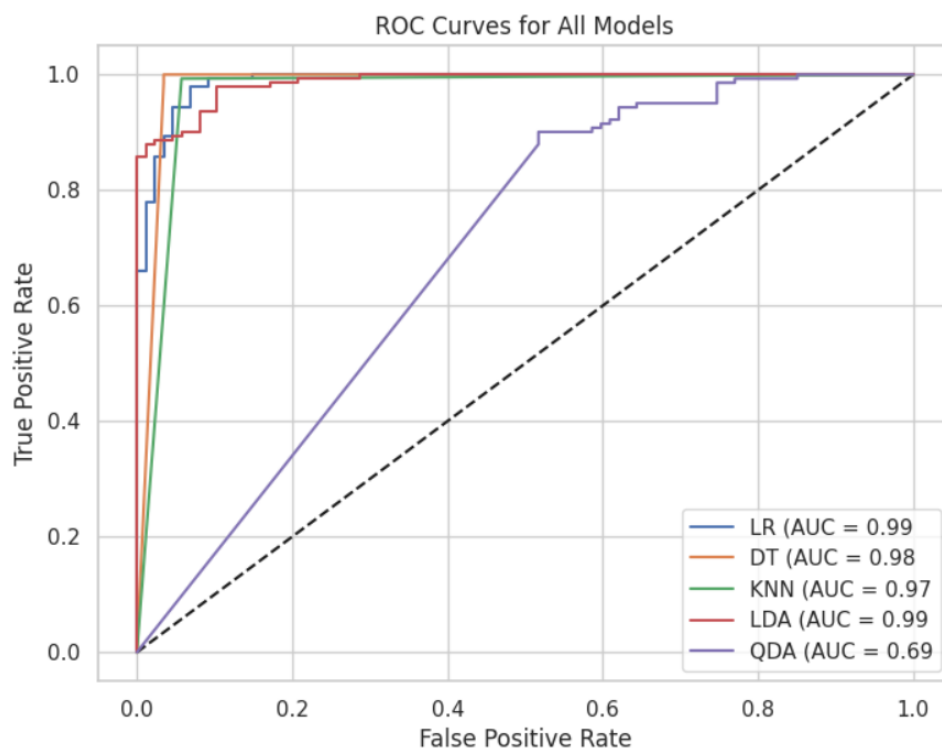
**Table 1**

*Model performance comparison*

Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.96	0.96	0.97	0.96	0.99
Decision Tree	0.99	0.98	1.00	0.99	0.98
KNN (k=1)	0.97	0.97	0.99	0.98	0.97
LDA	0.92	0.95	0.92	0.94	0.99
QDA	0.71	0.71	0.92	0.80	0.69

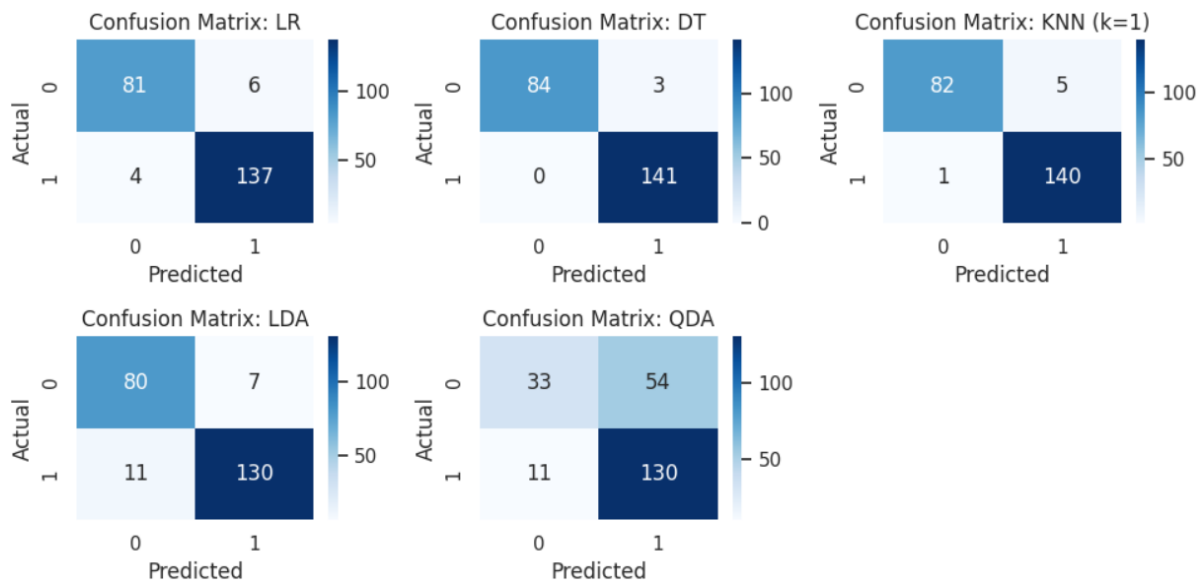
### ROC Curves and Confusion Matrices

Figure 1 displays the ROC curves for all five models. Logistic Regression achieved the highest AUC, indicating strong class separation capability.



ROC curves for all models

The confusion matrices provide insight into the types of classification errors made by each model.



*Models' confusion matrices*

## Observations and Limitations

Several observations and limitations emerged during the modeling process:

- **Class Balance:** Although the dataset was not severely imbalanced, the use of class-weight adjustments in models like Logistic Regression and Decision Tree helped maintain fairness across classes and avoided bias toward the majority class;
- **Feature Selection and Engineering:** Most features were numeric and domain-specific, including seismic measurements like magnitude, cdi, mmi, and sig. Feature scaling was necessary for KNN and LDA/QDA to function properly. However, feature interactions were not explored in depth, and more complex engineered variables (e.g., non-linear transformations or interactions) might enhance model performance;
- **Model Assumptions:** While LDA and Logistic Regression assume linear relationships and normally distributed predictors, the Decision Tree and KNN are non-parametric and more flexible. QDA's assumption of differing covariance structures likely contributed to its poor performance;
- **Generalizability:** All models were evaluated using a single random train-test split. Although cross-validation was used for KNN parameter tuning, future iterations could

benefit from full cross-validation across all models to better assess generalization error and reduce potential variance in the results;

- **Scalability and Runtime:** KNN can be computationally expensive with large datasets due to its instance-based nature. In contrast, the Decision Tree and Logistic Regression models train and predict efficiently, which is beneficial in real-time applications;

### **Final model selection**

Based on the evaluation metrics, the Decision Tree classifier emerged as the best-performing model. It achieved the highest accuracy (0.99), perfect recall (1.00), and an excellent F1 score (0.99), indicating a strong ability to correctly identify all relevant alert cases while maintaining precision. Additionally, its AUC of 0.98 shows excellent discrimination between alert and non-alert classes. Another key advantage of the decision tree is its interpretability, which is essential in critical applications like seismic alert systems.

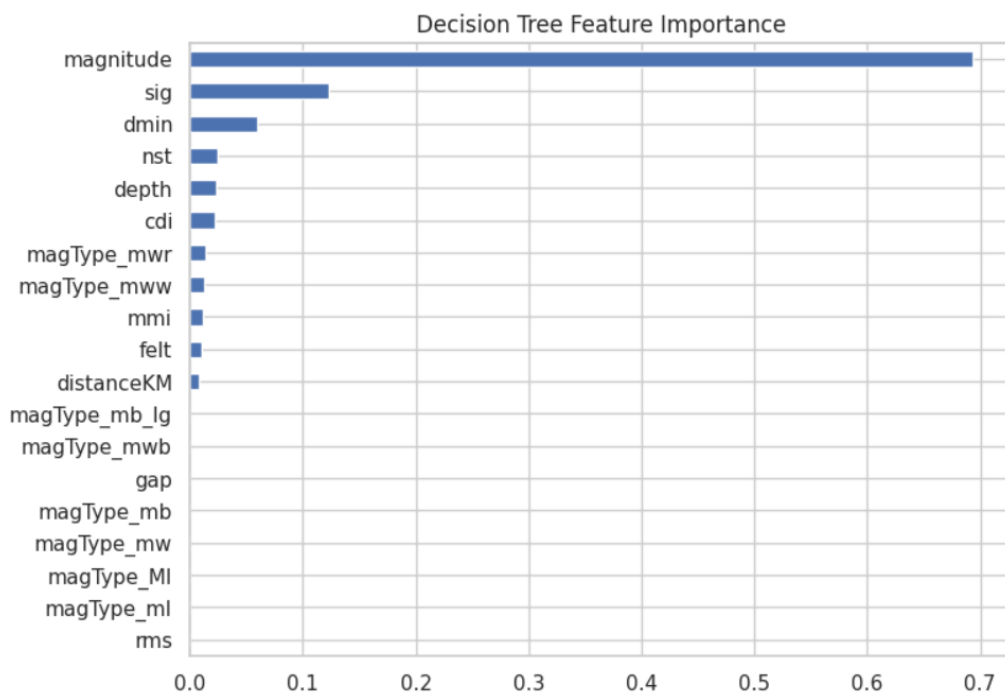
However, Logistic Regression and KNN also demonstrated strong and consistent performance, both achieving high AUC and balanced F1 scores. Logistic Regression remains a valuable model due to its simplicity and interpretability and could serve as a reliable baseline in future applications.

On the other hand, since LDA and QDA do not satisfy all the assumptions required for their application, these models were presented in this report as a way to enrich the model selection strategy, adding more information to the comparison and offering a broader perspective on how to correctly define the best model to apply in a business context.

Ultimately, the Decision Tree was selected as the final model and an analysis of feature importance was conducted. This analysis identifies which variables contributed the most to the model's predictions by measuring the decrease in entropy across splits.

### ***Feature Importance Analysis***

The feature importance plot derived from the Decision Tree model provides valuable insights into the relative contribution of each predictor variable to the classification task. In this context, variables such as sig (event significance), cdi (Community Determined Intensity), and mmi (Modified Mercalli Intensity) emerged as the most influential features in determining whether a tsunami alert would be issued. These variables capture the perceived and measured intensity of seismic events, suggesting that both the physical magnitude of the event and its impact on the population are key indicators in triggering tsunami warnings. The tree structure uses these features in its top-level splits, confirming their relevance in early classification decisions. The feature importance plot is shown in figure below:



Then, the table with the respective values of importance for each variable is shown below

(variables with importance equal or nearly equal to zero were omitted):

Feature	Importance	Relative %
magnitude	6.932848e-01	6.932848e+01
sig	1.227381e-01	1.227381e+01
dmin	5.981445e-02	5.981445e+00
nst	2.398747e-02	2.398747e+00
depth	2.290776e-02	2.290776e+00
cdi	2.214914e-02	2.214914e+00
magType_mwr	1.415368e-02	1.415368e+00
magType_mww	1.219165e-02	1.219165e+00
mmi	1.103666e-02	1.103666e+00
felt	9.746000e-03	9.746000e-01
distanceKM	7.990295e-03	7.990295e-01
magType_mb_lg	9.712229e-17	9.712229e-15

The feature importance values obtained from the Decision Tree model quantify the relative contribution of each variable to the predictive performance of the classifier. These values are calculated based on the reduction in entropy (or impurity) each feature provides when used to split nodes within the tree. A higher importance score indicates that a feature was frequently used in decision-making and produced substantial information gain. The corresponding percentage values represent each feature's proportion of the total importance, allowing for an intuitive comparison across variables.

### Discussion and Conclusion

Through the application of multiple Machine Learning algorithms, including Decision Tree, Logistic Regression, KNN, LDA, and QDA, it was possible to evaluate and compare the strengths and weaknesses of each method using a comprehensive historical earthquake dataset. Among the evaluated models, the Decision Tree classifier outperformed the others in terms of predictive performance, interpretability, and suitability for this specific classification task. The perfect recall (1.00) ensures that all tsunami-related alerts were correctly identified, minimizing the risk of missing a critical event. The interpretability of the model also makes it a strong candidate for use in real-world applications.

From a business and operational perspective, the potential application of this model could provide valuable support for emergency management agencies and environmental

monitoring systems. An early indication of whether a tsunami alert might be issued based on real-time seismic parameters could help reduce decision-making latency, improve public communication, and assist in the pre-deployment of emergency resources. Although this model is not intended to replace official early warning systems, it can serve as a complementary decision-support tool, especially in contexts with limited access to fully automated detection infrastructure.

Nevertheless, the study has several limitations. First, the model was trained on historical data, which may not fully capture the dynamic and evolving nature of seismic patterns. Second, the dataset used lacked detailed oceanographic variables and geospatial tectonic attributes, which could improve model accuracy. Third, no ensemble methods or neural networks were tested, which could be explored in future iterations to potentially improve performance further.

Some additional recommendations to consider for improving the model performance such as expand the dataset with additional geophysical and oceanographic variables to improve predictive accuracy and evaluate more complex models (e.g., random forests, gradient boosting, deep learning) are suggested for potential performance improvements.

In conclusion, the results of this project demonstrate that machine learning can be a powerful ally in disaster risk reduction, providing timely and accurate insights to support decision-makers in high-stakes environments.

## References

- Cesario, E., Giampá, S., Baglione, E., Cordrie, L., Selva, J., & Talia, D. (2024). Machine learning for tsunami waves forecasting using regression trees. *Journal of Computational Science*, 72, 102202. <https://doi.org/10.1016/j.jocs.2023.102202>
- National Oceanic and Atmospheric Administration. (2025, February 25). Tsunamis. <https://www.noaa.gov/education/resource-collections/ocean-coasts/tsunamis>
- Stein, S., & Wysession, M. (2003). *An introduction to seismology, earthquakes, and earth structures*. Blackwell Publishing.