# TSUNAMI PREDICTION AND ANALYSIS USING XGBOOST AND NAÏVE BAYES ALGORITM

NURFARAHIN BINTI AMIR HAMZAH

UNIVERSITI TEKNOLOGI MALAYSIA

**UNIVERSITI TEKNOLOGI MALAYSIA**
**DECLARATION OF PROJECT REPORT**

| | | | | | |
|---|---|---|---|---|---|
| Author's full name | : | NURFARAHIN BINTI AMIR HAMZAH | | | |
| Student's Matric No. | : | MCS241020 | Academic Session | : | 2024/2025-3 |
| Date of Birth | : | 16/03/1999 | UTM Email | : | nurfarahin99@graduate.utm.my |
| Project Report Title | : | TSUNAMI PREDICTION AND ANALYSIS USING XGBOOST AND NAÏVE BAYES ALGORITM | | | |

I declare that this project report is classified as:

☒ **OPEN ACCESS** — I agree that my report to be published as a hard copy or made available through online open access.

☐ **RESTRICTED** — Contains restricted information as specified by the organization/institution where research was done. *(The library will block access for up to three (3) years)*

☐ **CONFIDENTIAL** — Contains confidential information as specified in the Official Secret Act 1972)

*(If none of the options are selected, the first option will be chosen by default)*

I acknowledged the intellectual property in the project report bers to Universiti Teknologi Malaysia, and I agree to allow this to be placed in the library under the following terms :

1. This is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of only.
3. The Library of Universiti Teknologi Malaysia is allowed to make copies of this project report for academic exchange.

Signature of Student:

Signature :

Full Name : NURFARAHIN BINTI AMIR HAMZAH
Date : 20 SEPTEMBER 2025

Approved by Supervisor(s)

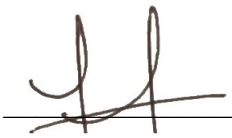
Signature of Supervisor :

Full Name of Supervisor
ASSC.PROF. DR. HAZA NUZLY BIN ABDULL HAMED

Date : 20 SEPTEMBER 2025

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

"I hereby declare that I have read this project report and in my opinion this project report is sufficient in term of scope and quality for the award of the degree of Master in Data Science

| | | |
|---|---|---|
| Signature | : |  |
| Name of Supervisor I | : | ASSC. PROF. DR HAZA NUZLY BIN ABDULL HAMED |
| Date | : | 20 SEPTEMBER 2025 |

# TSUNAMI PREDICTION AND ANALYSIS USING XGBOOST AND NAÏVE BAYES ALGORITM

NURFARAHIN BINTI AMIR HAMZAH

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master in Data Science

Faculty of Computing
Universiti Teknologi Malaysia

SEPTEMBER 2025

**DECLARATION**

I declare that this project report entitled *"TSUNAMI PREDICTION AND ANALYSIS USING XGBOOST AND NAÏVE BAYES ALGORITHM"* is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : NURFARAHIN BINTI AMIR HAMZAH

Date : 20 SEPTEMBER 2025

## ACKNOWLEDGEMENT

Bismillahirrahmanirrahim. In the name of Allah, the Merciful and Compassionate.

With full gratitude, I want to express my highest gratitude to Allah because with His permission, Alhamdulillah I gratefully able to complete these master thesis and journey in a year even though life is tested with various challenges that come like a rollercoaster. I also want to take this opportunity to express my deepest thanks to my supervisor, Associate Professor Dr. Haza Nuzly Bin Abdull Hamed and my master lectures for their irreplaceable advice, support, and positive feedback during this study.

Special thanks to my beloved parents and family who always support, give blessing and being a source of strength and determination. This thesis would not have been possible without their prayers and support which were always there.

I also feel thankful to my fellow postgraduate colleagues and friends who have helped me throughout this process, provided valuable discussions, support and unforgettable memories.

I would also like to give a heartfelt gratitude to Universiti Teknologi Malaysia (UTM) who provided facilities and resources that helped the successful completion of this study.

## ABSTRACT

Earthquakes or seismic activity are devastating geological disasters that can cause great destruction and loss of life. Earthquakes with a high magnitude exceeding moment magnitude (Mw) seven can trigger aftershocks such as tsunamis, especially in coastal areas. The use of traditional geoscience methods in earthquake studies alone is not sufficient to accurately predict and analyse earthquake and tsunami events. Therefore, machine learning approaches such as Extreme Gradient Boosting (XGBoost) and Naïve Bayes (NB) are introduced in this study to improve prediction accuracy. The objective of this study is to perform data pre-processing and exploratory data analysis (EDA) using a global earthquake dataset from Kaggle to make prediction of the tsunami-generating earthquakes occurrences using XGBoost and Naïve Bayes classification algorithm and then the prediction results presented in interactive visualization dashboard of the earthquakes historical trends, modelled prediction and tsunami risk zone for seismic hazard analysis. The methodology approach of data pre-processing, exploratory data analysis (EDA), feature selection and evaluate models were used in this study. The results found that both models produced accuracy and AUC exceeding 90% and 98%. XGboost showed the best performance in identifying earthquakes that produced tsunamis by recording the highest performance with recall of 98.5% compared to NB which was 95%. In conclusion, the XGBoost model is most suitable for tsunami early warning systems by minimizing the risk of false alarms. Furthermore, this study is then visualized using an interactive dashboard that allows the predictions and analysis of this study clearly presented and understandable. The analysis of this study identified that 2021 had the highest seismic activity which concentrated in the Pacific Ring of Fire zone especially in Indonesia, Tonga, and Japan that has highest recorded earthquake while United State and Canada have highest recorded tsunami. The results of this study prove that the integration of machine learning with geophysical and seismological sciences is effective in making disaster predictions and helps support risk mitigation strategies in high-potential disaster areas.

# ABSTRAK

Gempa bumi atau aktiviti sismik merupakan sebuah bencana geologi yang dahsyat dan boleh mengakibatkan kemusnahan yang besar dan kehilangan nyawa. Gempa bumi dengan magnitude tinggi melebihi momen magnitude (Mw) tujuh boleh mencetuskan bencana susulan seperti tsunami terutama di kawasan pesisir pantai. Penggunaan kaedah tradisional geosains dalam kajian gempa bumi sahaja tidak mencukupi untuk meramal and menganalis kejadian gempa bumi dan tsunami dengan tepat. Justeru itu, pendekatan penggunaan pembelajaran mesin (machine learning) seperti Extreme Gradient Boosting (XGBoost) dan Naïve Bayes (NB) diperkenalkan dalam kajian ini bagi meningkatkan ketepatan ramalan. Objektif kajian ini adalah untuk melaksanakan pra-pemprosesan dan analisis data penerokaan (EDA) menggunakan set data gempa bumi global daripada Kaggle untuk membuat ramalan kejadian gempa bumi yang menjana tsunami menggunakan algoritma klasifikasi XGBoost dan NB kemudian keputusan ramalan akan dipaparkan dalam papan pemuka visualisasi interaktif yang menunjukkan trend sejarah gempa bumi, analisis model ramalan seismik dan zon bahaya tsunami. Pendekatan metodologi pra-pemprosesan, eksplorasi analisis data (EDA), pemilihan ciri dan penilaian model digunakan dalam kajian ini. Hasil kajian mendapati kedua-dua model menghasilkan ketepatan dan AUC melebihi 90% dan 98%. XGboost menunjukkan prestasi yang paling baik dalam mengenal pasti gempa yang menghasilkan tsunami dengan merekodkan *recall* tertinggi 98.5% berbanding NB iaitu 95%. Kesimpulannya, model XGBoost paling sesuai untuk sistem amaran awal tsunami dengan meminimumkan risiko amaran palsu. Tambahan pula, kajian ini kemudiannya divisualisasikan menggunakan dashboard interaktif yang membolehkan ramalan dan analisis kajian ini disampaikan secara jelas dan mudah difahami. Analisis kajian ini mendapati bahawa tahun 2021 mempunyai aktiviti sismik yang paling tinggi yang menumpukan kawasan Lingkaran Api Pasifik terutama di Indonesia, Tonga, Jepun yang merekodkan kejadian gempa bumi tertinggi manakala Amerika Syarikat dan Kanada merekodkan kejadian tsunami tertinggi. Hasil kajian membuktikan bahawa integrasi pembelajaran mesin dengan sains geofizikal dan seismologi berkesan dalam membuat ramalan bencana dan membantu menyokong strategi mitigasi risiko di kawasan bencana yang berpotensi tinggi.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| AI | - | Artificial Intelligent |
| AUC | - | Area Under Curve |
| FN | - | False Negative |
| FP | - | False positive |
| FPR | - | False Negative Rate |
| k-NN | - | k- Nearest Neighbour |
| LR | - | Logistic Regression |
| ML | - | Machine Learning |
| Mw | - | Moment magnitude |
| NB | - | Naïve Bayes |
| RF | - | Random Forest |
| ROC | - | Receiver Operating Characteristic |
| SVM | - | Support Vector Machine |
| TN | - | True Negative |
| TP | - | True Positive |
| TPR | - | True Positive Rate |
| XGBoost | - | Extreme Gradient Boosting |

## LIST OF SYMBOLS

| | | |
|---|---|---|
| $A_t$ | - | actual value |
| c | - | constant |
| $h_i(x)$ | - | prediction of the *t*-th tree |
| p | - | AR model. |
| P(C) | - | Prior probability of the class |
| P(X\|C) | - | likelihood of observing the features given the class |
| P(X) | - | evidence. |
| $y_i$ | - | the expected value |
| $\hat{y}$ | - | final prediction. |
| $\mathbf{Y_t}$ | - | predicted value at time |
| $l(y_i, \hat{y_i})$ | - | the loss function |
| $\epsilon_t$ | - | error term |
| $\phi_1$ | - | autoregressive coefficients |
| $\Omega(f_k)$ = $\gamma T + \frac{1}{2}\lambda\sum\omega_j^2$ | - | regularization term that penalizes complexity |
| $\omega_j^2$ | - | leaf weight |

## LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Earthquakes are natural geological disasters that caused the highly damage and threatened the lives and infrastructure. In addition, some earthquakes can trigger other natural hazards such as aftershocks, tsunami, and landslides that have destructive potential as the original seismic movement itself. Most destructive tsunamis are created by high-magnitude offshore earthquakes, which can traverse ocean floors devastating seaside citizens. The occurrence of these events is often unpredictable and this becomes the main challenge for seismologists and scientists to predict these events. Sometimes traditional statistical methods are not suitable for earthquake prediction due to the complex and nonlinear behavior of earthquake that need the advanced methods including machine learning algorithms to generate more accurate earthquake data. This research aims to discover the use of XGBoost and Naïve Bayes (NB) to enhance the prediction accuracy of the earthquake events in identifying the tsunami occurrences generated from the high magnitude earthquake to support more effective disaster preparation and risk mitigation.

## 1.2 Problem Background

Earthquakes or seismic activity are catastrophic natural events that can threats to human life, infrastructure, and socioeconomic stability. The formation of this event due to the releasing energy of geological stress along fault created by tectonic processes including subduction, plate collision or lateral movement of plates that can cause the sudden shaking of lithosphere and can be felt by human (Lay, 2016; Stein & Klosko, 2002)

As the stress exceeds the regional rock, seismic waves are generated and the ground shakes and often initiating secondary hazard such as tsunamis and landslides. Tsunami occurs when a large submarine earthquake displaces huge volumes of water thereby creating thrasher waves that can travel long distance along the ocean to the coasts (Cilia et al., 2021; Ulrich et al., 2019). Although earthquake occurrence is mainly focused in active tectonic zones as the Pacific Ring of Fire, their impact also can be felt in nonactive zone such as Malaysia that affected from the seismic activity and tsunami in Sumatra in 2004 and 2015 Sabah earthquake due to the active movement of fault.

The occurrences of these geohazards are difficult to predict by the traditional prediction methods of the past statistical modelling, so the development of new statistical models and system is necessary to improve the accuracy and timely forecasting systems. As the old methods are limited due to the complexity, unpredictability and constant transformations of earthquakes and tsunamis, recent researchers have used advanced techniques such as machine learning (ML) and artificial intelligence (AI) to help more clearly distinguish underlying patterns in seismic data and improve forecast accuracy.

The research on earthquakes and tsunami mostly relying on the prediction magnitude and the occurrences time rather than spatial perspective of seismic hazards. Risk analysis requires outlining the exact geological areas prone to earthquakes such as areas located on the Pacific Ring of Fire and active fault and the likelihood of new hazards due to earthquakes including aftershocks, tsunami, volcanic activities and landslides. However, the current predictive models do not display spatial generalization and most are built on incomplete datasets which fail to consider the geospatial or tectonic context. The advanced classification techniques of machine learning such as Naïve Bayes, Random Forest, Support Vector Machine (SVM) and XGBoost can be used to classify and point out high-hazard areas by using historical trends, seismic frequency, and geological characteristics that rarely used in geological research (Babu et al., 2024; T. Wang et al., 2023; Yenidoğan, 2024)

Additionally, the existing solutions to earthquake analysis in machine learning involve concentrating on numerical and time series information without explicitly incorporating other geological factors such as fault types, tectonic zone, plate

movement, secondary hazard and post-seismic activity which have decisive influence on seismic activity. This inconsistency between domain and algorithm modelling weakens the interpretability and limits the utility of earthquake and tsunami forecasts especially areas relating to creating disaster plans. The proposed research thus incorporates geospatial and geological parameters in a XGBoost and Naïve Bayes (NB) based model with a two-fold idea of improving predictive performance and the higher interpretability of the areas of high seismic hazard.

Furthermore, the collected seismic data usually has missing values, noise, and class quantity especially data collected from open-source database like Kaggle where the low magnitude seismic event exceeds high-impact seismic event tremendously. These inconsistencies can prevent machine learning models from learning generalizable patterns especially when involving predicting or classifying unusual and disastrous rare events. As a result, appropriate pre-processing is important in scaling features, outlier removal and class balancing before training. Failure to put these considerations into account may enable models such as XGBoost to overfit or to generalize insufficiently across verticals and periods.

## 1.3 Problem Statement

The occurrence of earthquakes and tsunami is still unpredictable and affect the large losses of human lives, infrastructure damage and long-term economic problems. Although seismology and geoscience have greatly improved over the past few decades, there is still a significant challenge in scientific research to predict accurately the timing, location, depth, magnitude, tectonic activities, and secondary hazards of earthquakes. Nonlinearity and the intrinsic complexity of tectonic processes makes the traditional geophysical and statistical models inadequate to provide accurate predictions.

At the same time, the growing availability of seismic data alongside the development of data science and machine learning opens countless possibilities to improve the prediction of earthquakes. Application of machine-learning algorithms

and AI especially to predicting complex and multidimensional temporal correlations among earthquake data may reveal hidden signals and correlations in earthquake data, revealing more profound phenomena of large-scale seismic events.

Thus, an interdisciplinary systematic approach combining the exchange of geoscientific knowledge with the innovative analytical skills of the data science field of knowledge should be developed to overcome these issues. Machine-learning methods applied to large datasets of seismic and geological data have the potential to enhance the accuracy, explicability and dependability of earthquake-prediction models such as the detection of secondary events like tsunamis following high-magnitude earthquakes.

Modern seismic prediction, impact analysis, and risk visualisation methods are usually considered separate tasks in current research, thus delivering incomplete workflow processes that are difficult to scale or execute. It consequently requires the development of a more coherent architecture that incorporates exploratory data analysis, predictive modelling, spatial classification and the visualization in a single consistent pipeline.

This paper will present a comprehensive pipeline consistent with data collection, data pre-processing, model development and deployment of a dashboard into a holistic solution in predicting earthquakes and the evaluation of the hazards and take advantage of the advanced machine learning including XGBoost and NB algorithms.

## 1.4 Research Goal

The aim of this study is to design an interactive and insightful visualization model of machine learning algorithm for analyzing the global earthquake data and make predictions of tsunami-generating earthquake occurrences to improve the understanding of earthquake patterns and facilitating disaster preparedness.

### 1.4.1 Research Questions

Earthquakes and tsunamis are complex, nonlinear, and dynamic random phenomena, where conventional statistical methods often fail to provide accurate analysis. Therefore, the research question arises as to the extent to which machine learning (ML) techniques can improve prediction accuracy, identify hidden patterns, and prove their effectiveness in real earthquake prediction systems. This research aims to address these gaps through the following key questions:

(a) How geosciences and data science knowledge can be integrated in earthquakes analysis and prediction?

(b) How XGBoost and Naïve Bayes (NB) algorithm determine the tsunami-generating earthquake?

(c) How the results of the prediction and analysis of tsunami from XGBoost and NB algorithm visualized?

.

### 1.4.2 Research Objectives

(a) To perform data pre-processing and exploratory data analysis (EDA) for earthquake and tsunami analysis and prediction.

(b) To predict the tsunami-generating earthquakes occurrences using XGBoost and Naïve Bayes classification algorithm.

(c) To develop an interactive visualization dashboard for presenting the earthquakes historical trends, modelled prediction and tsunami risk zone for seismic hazard analysis.

## 1.5 Scope

The scope of this research focuses on the prediction and analysis of the occurrences of global earthquake-generating tsunami from 2015-2024. The dataset available on Kaggle contains information about earthquake events around the world such as magnitude, depth, tsunami, and location. Qualitative and quantitative approaches from the integration of geoscience and data science techniques including data pre-processing, exploratory data analysis (EDA) and machine learning models are used in this study to improve the performances of predictions and analysis of tsunami events. However, there are several limitations of the study that need to be acknowledged, including limited availability of high-quality data in some areas, class imbalance between tsunami-generating and non-tsunami-generating earthquakes. In addition, model results may be affected by limitations in integrating other geological factors such as rock type or incomplete geotechnical conditions in the dataset.

## 1.6 Significance of Research

This work is an important contribution to the scientific research and has practical consequences in respect to the initiatives of alleviating the risks of disasters. This study aims at improving tsunami prediction models by incorporating the concepts of machine learning with geoscientific knowledge that breaks the traditional boundaries of the few factors that are well understood due to inadequate understanding of the intricate, nonlinear patterns of seismic data. The effectiveness of the early warning systems due to the optimization of these predictive models can be increased, whether in the more seismically sensitive and less sensitive areas providing the communities with more lead time to react and counteract the destructive secondary effects of an earthquake. Simultaneously, this research further the understanding in the emerging field of geoinformatics, with a synergistic combination of geological sciences and data science. Since urbanization and climate changes entail the increased likelihood of earthquakes, the increased accuracy of prediction may substantially reduce human and economic cost in the face of these events. Further, the results of this study may contribute to the improvement of predictive systems applicable to various

natural catastrophes, therefore extending the scope and impact of this future seismology study.

## 1.7 Chapter Summary

In summary, this chapter is the fundamental stage of research where it explains the challenges of analysing and predicting high magnitude earthquakes triggering the tsunami and the limitations of traditional methods due to the complexity of seismic activity. The application of machine learning highlighted in this study to improve earthquake by exposing hidden patterns in seismic data. The chapter also outlines the approach of researching and comparing machine learning models for disaster preparedness and risk reduction especially in locations with a limited of earthquakes history.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

This chapter reviews the past study of earthquake events and the machine learning tools used in earthquake analysis and predictions. The strength and limitations of past study in various predictive approaches and highlighting their relevance to current development discussed in this chapter.

## 2.2 Tsunami Formation

Tsunamis are known as secondary hazard generated primarily from the high magnitude earthquake or volcanic activities. The word of "Tsunami" comes from Japanese language, *'tsu'* = harbour and *'nami'* = wave which literally means as "waves in harbour". This named based on the Japanese experience of large waves caused by earthquakes that appear calm in the open sea but are devastating when entering a harbor or narrow bay (Paris, 2015). The term was officially recognized by the international scientific community after the 20th century to distinguish it from tidal waves and then this name was used by international bodies such as UNESCO-IOC and NOAA (Suppasri et al., 2024).

Tsunamis mostly occurred in the subduction zone where seafloor suddenly moved and pushes huge amounts of water up or down thus large waves. For example tsunami 2004 in Sunda Trench of Aceh Indonesia (Cilia et al., 2021). According to E. Yousif, (2022) , earthquakes often cause the sea to pull back first, exposing the seabed. In some cases, cracks may open and let water rush down toward hot magma. The water

instantly turns to steam, expands about 1700 times and blasts back upward, making the returning tsunami waves much stronger. This helps explain why tsunamis after events like the 2004 Indian Ocean earthquake and the 2011 Japan earthquake became so powerful with waves reaching 30–40 m high on land.

Once the waves form, they travel very fast across the ocean. Tandel et al., (2022) show that in deep ocean water ~4000 m, tsunami waves move at speeds of 700–900 km/h but usually less than 1 m high that ships may not notice them. But as the waves approach the coast, the ocean gets shallower. The waves slow down to about 30–50 km/h, but their height grows dramatically because the energy gets squeezed into a smaller space. That is why when tsunamis reach shorelines, they can rise into walls of water that flood inland for kilometres, as happened in Sumatra (2004) and Japan (2011) (Cilia et al., 2021; Suppasri et al., 2024).

In addition, tsunamis can be triggered by underwater landslides. For example, Palu tsunami in 2018 due to a strike-slip earthquake triggered a liquefaction phenomenon that caused a large portion of sediment and land to slide into the sea, generating a local but very destructive tsunami wave (Sassa & Takagawa, 2019). Moreover, tsunamis can also be triggered by volcanic activity, whether from an eruption or a mountainside collapse or an underwater eruption. As example, the Anak Krakatau tsunami in 2018 claimed hundreds of lives indicates of how a volcanic collapse can generate large waves without a tectonic earthquake (Heidarzadeh et al., 2019)

Also, high-risk giant underwater eruptions that rarely occur such as the Hunga Tonga-Hunga Ha‘apai eruption in 2022 produced tsunamis that spread as far as Japan, Chile, and North America (Dogan et al., 2023). Less commonly discussed mechanisms are non-seismic sources such as meteorite falls into the sea or a combination of several factors at once. Although these events are rare, researchers emphasize the need to understand the potential of this type of tsunami because it is high-risk and difficult to predict (Firoozi & Firoozi, 2023)

### 2.2.1 Factors of Tsunami

The size of a tsunami depends not only on the triggering mechanism, but also on factors that influence its intensity and impact. First, the magnitude and depth of the earthquake are the main factors that triggered tsunami. Shallow earthquakes with large magnitudes have the potential to produce stronger tsunamis than deep earthquakes (Lay, 2016). Second, the shape of the seabed and coastline also play a major role where waves will be larger when entering narrow bays or areas with topography that can focus on wave energy (Sassa & Takagawa, 2019). Next, the distance from the epicentre to the land also can be the factor of this event. If the epicentre is close to the coastline, the early warning period becomes very short, so that the population does not have time to save their lives (Suppasri et al., 2021).

Moreover, tsunami frequently occur in the Pacific Ring of Fire, a belt around the Pacific Ocean that is the most seismically and volcanically active location in the world. The collision and shifting of tectonic plates in this area produces large earthquakes and tsunamis that frequently cause major damage, such as in Sumatra, Chile, and Japan (Dogan et al., 2023; Dragoni & Santini, 2022).

In addition, type of fault involved in an earthquake also determines the potential for tsunamis. Thrust faults in subduction zones are more dangerous than strike-slip faults because thrust faults involve large vertical displacements of the seafloor. In contrast, strike-slip faults do not usually cause tsunamis unless they trigger underwater landslides such as occurred in Palu in 2018. Thus, recent studies have highlighted the necessity of source mechanism analysis of an earthquake to determine the probability of a tsunami with the latest technologies such as artificial intelligence (AI) and machine learning (Handayani et al., 2024).

## 2.3 Earthquake Generating Tsunami

Most tsunami events started with the occurrences of earthquake events. Earthquake happens due to a sudden movement of a tectonic plate caused by the energy release along a fault zone, leading to a vibration in lithosphere of the earth and failure, slipping and shifting of the tectonic plate (Pwavodi et al., 2024).

According to Lay (2016), earthquakes started at subduction zones such as the Atlantic ridges and transform fault zones. Subduction zone is a tectonic plate boundary where an oceanic plate is forced under another oceanic or continental plate based on gravitational and slab pull force, producing major seismic activity due to accumulated strain energy (Figure 2.1). As the subducting plate sinks into mantle, stress accumulates at the megathrust interface due to friction and resistance at this interface until a point when a rapid rupture occurs and triggers an earthquake and formed tsunami (Lu et al., 2017).

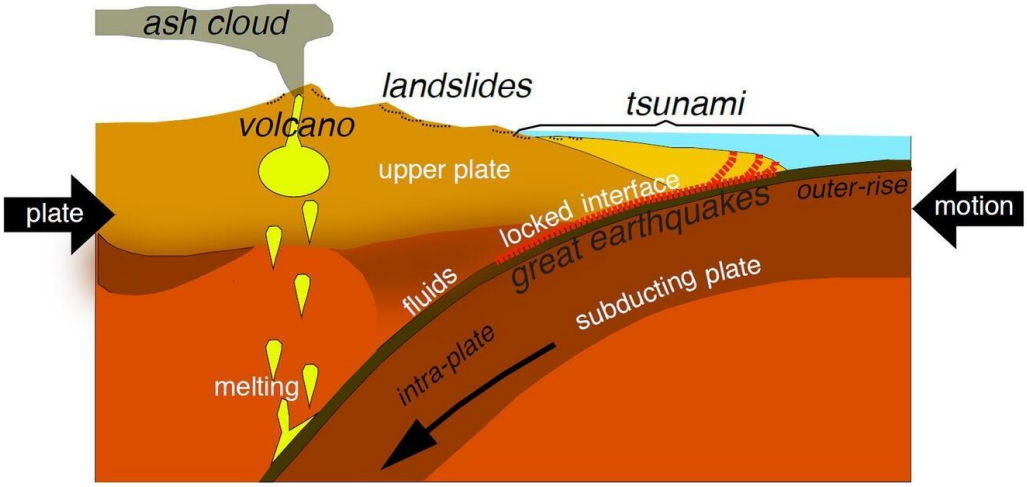



Figure 2.1 Schematic cross section of a typical subduction zone (USGS, 2016)

Subduction zones are most tectonic active areas that induce tsunamis. It occurrence commonly along convergent boundaries with various collision of tectonic plates. Plate collision classified into three major types which are oceanic-continental plate, oceanic-oceanic plate and continental-continental plate. Subduction zone mostly occurs at active seismic activity known as “Pacific Ring of Fire” (Figure 2.2) that also has the active volcano activities and deep oceanic trenches (Newbrough, 2025).

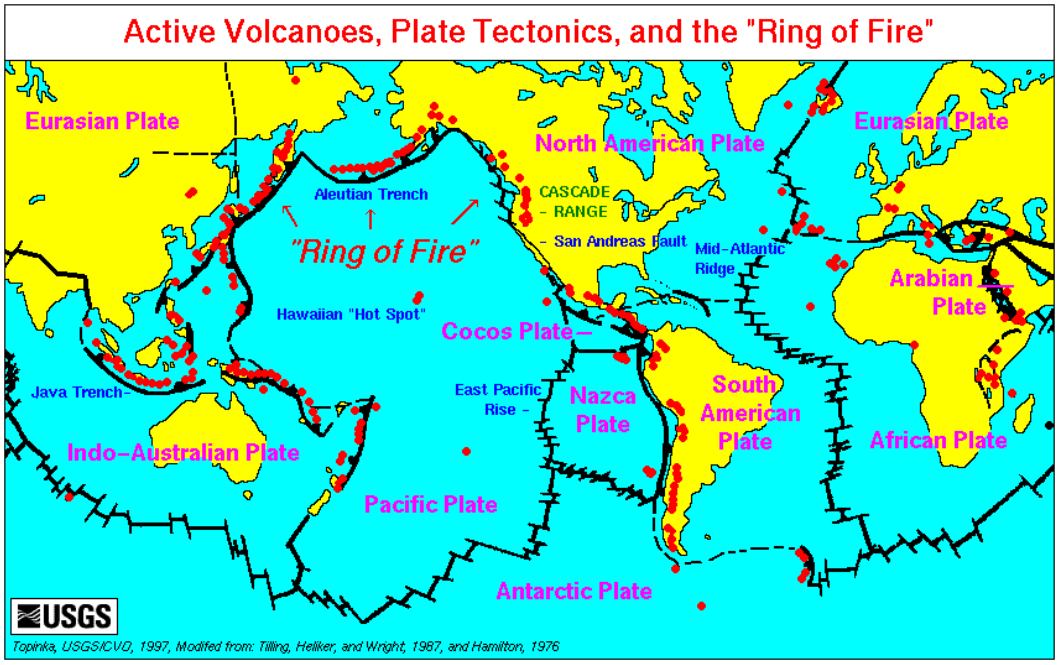



Figure 2.2 Location of active volcanoes, plate tectonics and the Ring of Fire (USGS,1997)

The presence of active fault also related to the formation of earthquake due the continuous crustal movement that generated the frictional resistance between plates and accumulation of the tectonic stress. Over the time, when the stress exceeds the frictional resistance, the earth crust fractured and sudden slip formed then releasing energy in form of seismic wave. The larger and deeper the slip occur, the highest magnitude of earthquake generated with some can triggered tsunami especially in ocean. This process called as elastic rebound (Biswas et al., 2023).

The active faults such as East Anatolian Fault and San Andreas Fault are prone to earthquake due to active fault interaction and accumulation of stress at faster rate but this fault has low possibility to generate tsunamis due to the horizontal nature of the faults, the predominantly terrestrial location and the absence of oceanic subduction zones. Although large earthquakes of magnitudes approaching 7.0 moment magnitude (Mw) can occur, the tectonic energy is released in the form of horizontal shaking of the land and not uplift or subsidence of the seafloor. Thus, the main risk in this region is from land-based earthquakes rather than from large tsunamis as is common in the Pacific Ring of Fire..  (Güvercin et al., 2022).

## 2.4 Types of Plate Boundary

Earth consists of combination of plates that interact with each other resulting in the energy release, shaking to the earth and the creation of faults. These natural activities are very important to generate new geological features such as lands, rocks, islands, mountains and volcanoes.  Plate boundaries can be divided into several types including convergent plate, divergent plates and transform plates shown in Figure 2.3  (Duarte & Schellart, 2016).

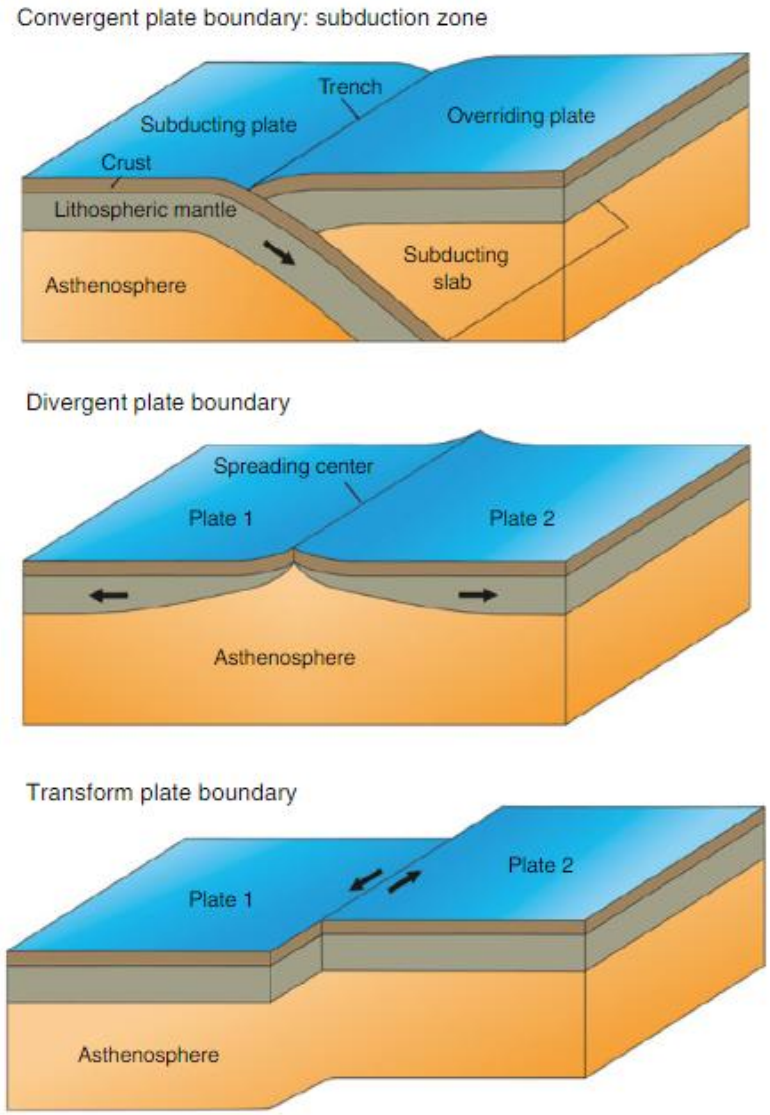



Figure 2.3 Schematic representation of three types of plate boundaries: convergent (top), divergent (centre) and transform (bottom) modelled from (Duarte & Schellart, 2016)

### 2.4.1 Convergent Plates

Most earthquakes generated at these plates where two plates collide often causing one plate to be beneath the other plate that is known as subduction zone (Figure 2.1). This process creates most powerful earthquake, volcano activities and tsunamis, such as those along the Pacific Ring of Fire. Convergent plates consist of three categories which are continental-continental boundary, oceanic-continental boundary and oceanic-oceanic boundary (USGS, 2016).

#### 2.4.1.1.1 Continental - Continental Boundary

In continental-continental boundaries, two tectonic plates of continental crusts meet and collide into each other producing a large compression that deform the crust resulting in earthquakes of large magnitude. The best example is the collision of the Indian and Eurasian plate which has led to the formation of Himalayas ridge that uplift and produce large earthquake due to crustal shortening and the formation of fault (Bao et al., 2022). Earthquakes in these boundaries are pointed out at shallow depth and triggered the thrust faulting amongst complex fault systems that did not form a deep subduction since neither the plate can subduct easily enough to the mantle with low possibility of tsunami. (Luo et al., 2022).

Another example is the Zagros Mountains that have been formed as a resulted from the Arabic-Eurasian continental collision have moderate to strong earthquakes. In this area significant seismic activity results from the sharp release of the accumulated tectonic stress by fault failure with no tsunami alarm (Manaman & Shomali, 2010). It can be concluded that crustal compression and fault reactivation within thickened lithosphere is the major cause of earthquakes at continental-continental margins with low possibility of tsunami as secondary hazard (James, 2021).

#### 2.4.1.1.2 Oceanic – Continental boundary

Oceanic–continental convergence boundary occurs when a high density of oceanic plate subducts under a continental plate leading to intensive tension and pressure causing deep-focus earthquakes and inland volcanism (Duarte & Schellart, 2016). This mechanism creates deep-sea trenches, forearc basins, back arcs basin and volcanic arcs such as those found in the Andes and Cascadia regions which also generated the intermediate to high magnitude of deep. These large earthquakes in these zones capable to create the tsunami event due to the subduction of plates (Bilek & Lay, 2018).

Moreover, these boundaries also generate characteristic seismic anomalies consist of a double seismic zone where the dehydration of the sinking slab increases the fluid pressures, weakens the fault strength, and causes intermediate earthquakes (Geersen et al., 2022). Some of major earthquakes occurred within the outer rise, seaward of the trench, or on plate-bending related faults below the megathrusts, while others are originated within the upper plate near to the zone of high slip. Even though the deeper megathrust ruptured are rarely happened, but some intraplate aftershock may occur due to the significant afterslip in the upper plate that resulting from the complicated faults and high activity aftershocks that can triggered formation of secondary hazard like tsunami (Bilek & Lay, 2018; Lay et al., 2020).

#### 2.4.1.1.3 Oceanic – Oceanic boundary

Oceanic-oceanic plate boundaries formed when two oceanic plate collides where one plate subduct under another plate produced the deep ocean trenches of large earthquakes due to the frictional locking and massive energy released throughout the subduction zone. The subduction initiate when the older dense oceanic lithosphere being forced beneath the younger and less dense oceanic lithosphere that basically facilitated by transform faults (Wu et al., 2019). For example, Tonga-Kermadec subduction zone is the example of megathrust earthquake caused by the accumulation of tectonic stress (Wang et al., 2023). Subduction at these boundaries also lead to

volcanic eruptions and tsunamis often associated with the sudden rise of the seafloor (Buforn & Udías, 2010).

In these regions, seismic activity usually traces the geometry of the descending slab and exhibits constant subduction dynamics. Further, geophysical studies show that there is an internal deformation of the down going slab in ocean-ocean boundaries, which explains the complexity of the earthquake (Gurnis et al., 2000) . Seismic profiles and focal mechanisms recorded in these areas indicate that reverse and thrust faulting are the dominant patterns of rupture, consistent with the overall compressional characteristics of tectonic processes (Sartori et al., 1994).

### 2.4.2 Divergent Plate

Divergent plate boundaries are zones occur when two tectonic plates move separately formed new seafloor and mid-ocean ridges or continental rift zones due to upwelling magma and induced shallow earthquakes from stretching crust (Duarte & Schellart, 2016). Earthquake in this plate less prone to high magnitude compared to convergent plate due to tensional stress that pulls the plate apart along rift zones and primarily occur at the normal fault where the hanging wall moves down relative to the footwall which generate small tension and seismic activity (Olive, 2023).

Divergent plate also has low possibility to triggered tsunami even though it experience the large seismic activity due to dynamic stresses transfer from distant seismic events, magmatic processes such as dike intrusion that spread plates apart and large tensional stresses and faulting movement (Hill & Prejean, 2015). As example, earthquake of 5.0 magnitude at Reykjanes Peninsula in Iceland on 2021 of the divergent Mid-Atlantic Ridge are associated to both intense seismic swarms and magma intrusion into formed the 9 km dyke beneath Fagradalsfjall area but no tsunami event recorded (Fischer et al., 2022).

Similarly, the earthquake near the East African Rift Zone with magnitude 5.9 affect disruption in North Tanzania demonstrated the impact of divergent zone quakes due to the magma intrusion of Fentale diking resulting the seismic activity and volcanic events as secondary hazard (Macheyeki, 2024). These events highlight the need of rigorous monitoring of rift regions especially in areas of higher population growth.

### 2.4.3 Transform Plate

Transform plate boundaries formed when tectonic plates move and slide alongside each other horizontally produced large shear stress which often generate earthquakes (Duarte & Schellart, 2016). The characteristics of these boundaries include strike slip faults such as California's San Andreas Fault where the Pacific plate moves northward to the North American plate. If the motion of tectonic plates is hindered by friction, built-up stress may eventually be abruptly discharged through short earthquakes causing extensive damage (Liu & Buck, 2018).

The East Anatolian Fault in Turkey is the example of major transform fault that triggered fatal earthquakes in 2023 with the loss of tens of thousands of individuals (Biswas et al., 2023). This case demonstrates the huge seismic risks involved in transform boundaries especially in areas where large cities are located close to fault zones. Transform boundaries has moderate possibility on creating tsunami due to vertical displacement or the plates linked to other tectonic activities. For example, the 7.7 Mw of 2012 Haida Gwaii, Canada formed the significant 7-meter tsunami in several inlets on the west coast of Moresby Island (several over 6 m, with a maximum of 13 m). This event generated from a thrust event on a predominantly strike-slip transform fault and there is a component of oblique convergence between the Pacific and North America plates off Haida Gwaii (Leonard & Bednarski, 2015).

## 2.5 Pacific Ring of Fire

Approximate 90% of global earthquakes and tsunami and 75% of dormant world's active volcano located at active seismic zone known as "Ring of Fire" mainly due to active interactions between the Pacific Plate and the nearby plates such as the Philippine, Cocos, and Nazca plates shown in Figure 2.1 and Figure 2.4. These zones have high seismicity, active volcanoes and large oceanic trenches resulting from the intense tectonic forces (Pwavodi et al., 2024). Most of these interactions take place at subduction zones where oceanic lithosphere subducts under continental or other oceanic plates causing slabs to be circulated into the mantle and the epirogenic of high heat and magma that feeds explosive volcanic arcs (Wu et al., 2019).

The Ring of Fire is bounded by subduction zones that have deep ocean trenches with volcanism occurred along the arc systems and high earthquake activities at convergence and subduction zones (X. Wang et al., 2023; F. Y. Wu et al., 2019). As example, Japan Trench and Peru-Chile Trench located at major subduction zones where the Pacific Plate subducted under Eurasian and South American Plates (Dragoni & Santini, 2022; Nakamura et al., 2023). These processes of slab pull, and mantle convection are important for developing the dynamics of tectonic motion and the ongoing transformation of the Ring of Fire.

However, 10% possibility of the earthquake's events occur outside this zone can also lead to the massive disaster due to the active movement of plate boundaries such 7.8 magnitude of earthquake with 17.9-km depth in Turkey and Syria 2023 which killed more than 45, 000 peoples (Biswas et al., 2023). The collision of Eurasian and Africa plates (Figure 2.3) triggered by several active faults including North Anatolian Fault, the East Anatolian Fault and the Dead Sea Transform Fault has triggered varying magnitude of earthquakes ranging from minor to major (Biswas et al., 2023). These events summaries in Table 2.1 where some highest magnitude does not generate tsunami due to its geolocation outside the subduction zone and ring of fire.

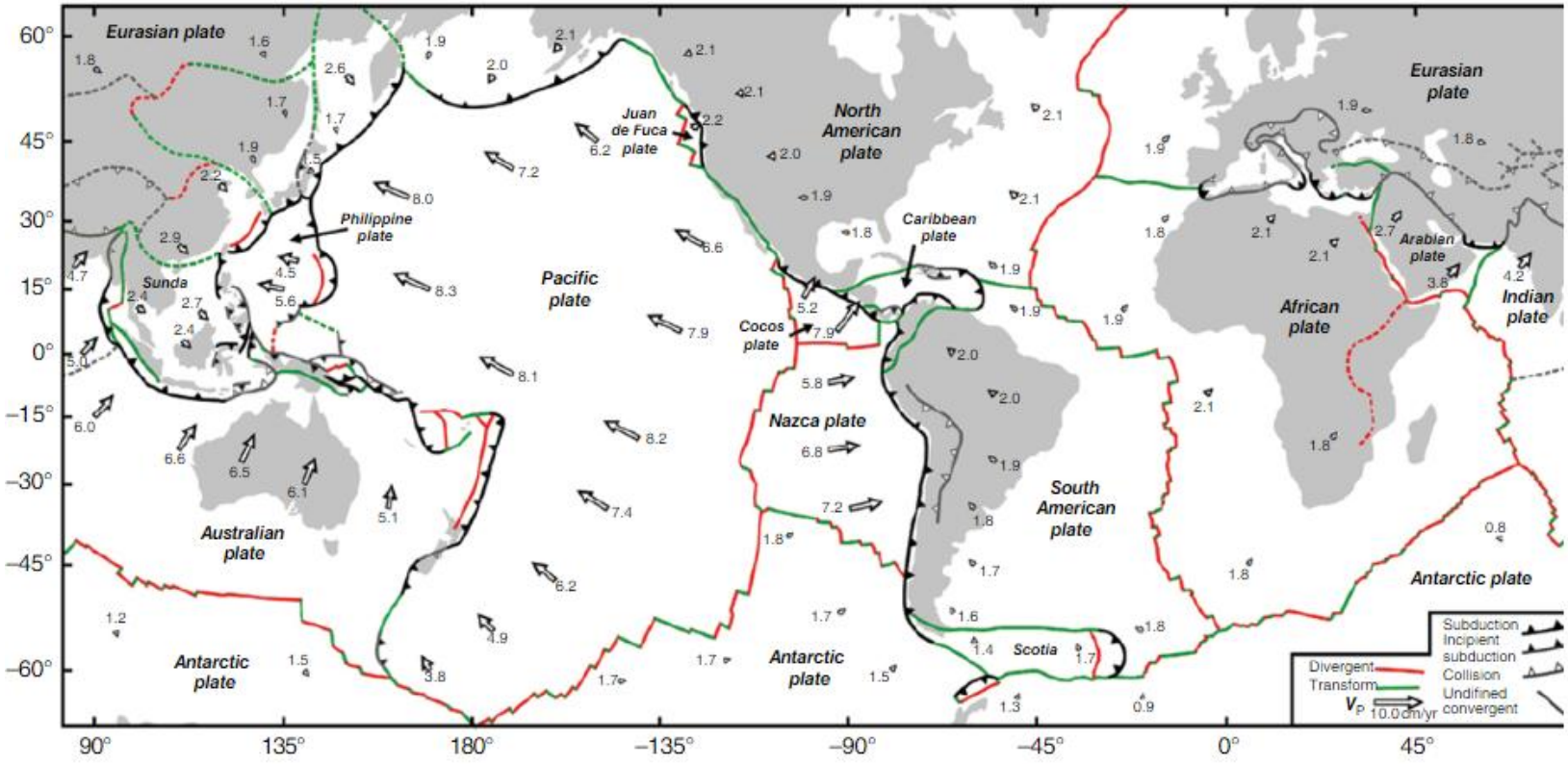



Figure 2.4 Global plate tectonic map illustrating of major tectonic plates modified from (Duarte & Schellart, 2016)

Table 2.1 History of high magnitude earthquake in from 2015-2025

| Year | Citation | Location | Characteristics of Earthquake | Risk Factor (Outcome) | Impact |
|---|---|---|---|---|---|
| 2015 | Quantifying The Benefit of Risk Mitigation Strategies on Present and Future Seismic Losses in Katmandu Valley, Nepal.<br><br>(Mesta et al., 2023) | Nepal | Mw 7.8<br>Thrust fault<br><br>No tsunami | Himalaya fold zone, old rock structure, poor infrastructure and high population density caused the earthquake but it not triggered the tsunami. | Massive destruction in Kathmandu killed nearly 8,800, 23,000 injuries, extensive property damage. |
| 2016 | April 2016 Ecuador Earthquake of magnitude 7.8 Mw: Overview and Damage Report.<br><br>(Mera et al., 2017) | Ecuador | Mw 7.8,<br>Subduction zone<br><br>No tsunami | Subduction of the Nazca Plate under the South American Plate caused the crustal earthquakes and volcanism on the Andes Mountains. However, no tsunami hazard recorded. | 670 killed, 28,000 injured and massive destruction |
| 2018 | Field Insights and Analysis of the 2018 Mw 7.5 Palu, Indonesia Earthquake, Tsunami and Landslides<br><br>(Cilia et al., 2021) | Indonesia, Sulawesi | Mw 7.5,<br>Strike-slip fault and Tsunami | Palu-Koro left-lateral strike-slip fault triggered tsunami due to earthquake-induced soil liquefaction and landslides. | Destroy Palu city with more than 2,245 people killed, thousands missing, 10,000 injured and 4000 severe injured, 75,000 displaced. |

| | | | | | |
|---|---|---|---|---|---|
| 2023 | February 6, 2023, Earthquakes and Preliminary Assessment of Building Damage Based on Field Surveys.<br><br>(Yenidoğan, 2024) | Turkey-Syria, 2023 | Mw 7.8 + Mw 7.6,<br>Strike-slip<br><br>No tsunami | Extensive rupture length (~560 km total), multiple large magnitude events in close succession, and the presence of seismic gaps along the fault segments. No tsunami has recorded. | Over 53,000 killed, 11 provinces affected and extensive property damage |
| 2024 | The 2024 Noto Peninsula Earthquake: Preliminary Observations and Lessons to Be Learned.<br><br>(Suppasri et al., 2024) | Noto Peninsula, Japan | Mw 7.5,<br>Active fault and Cascading hazard<br><br>Tsunami | Cascading hazard including geological uplift, liquefaction, landslides, fires and tsunami. | 240 deaths, severe infrastructure damage |
| 2025 | In the wake of the March 28, 2025, Myanmar earthquake: A detailed examination.<br><br>(Shahzada et al., 2025) | Myanmar-Thailand, 2025 | Mw 7.7,<br>Strike-slip (Sagaing fault)<br><br>No tsunami | A major dextral strike-slip boundary between the Burma Microplate and Sunda Plate caused supershear rupture propagated over 460 and surface displacements exceeding 6 m and violent shaking in urban centers like Mandalay, Sagaing, and Naypyidaw. These events did not generate tsunami cause it happen at land. | More than 4390 killed, over ~ 4900 fatalities, ~ 6000 injuries, and widespread destruction of infrastructure. |

## 2.6 Supervised Machine Learning for Earthquake Analysis and Prediction

Supervised machine learning is the labelled training dataset that implemented to make a prediction of input data with chosen output. The application of supervised ML models in earthquake studies such as simulate the movement and analysis of tsunami movement through neural network-based hazard predictions which compared outputs of the model and observed wave heights (Pham et al., 2018). These interlocking mechanisms has improved the insights of researchers and disaster management organizations to measure and manage tectonic events in complex geologies such as in Sulawesi (Velarde et al., 2024).

Some earthquakes triggered the occurrence of tsunami resulting the massive catastrophic impact of that area such as in Japan, 2011 and Indonesia, 2018. The diverse earthquakes analysis on earthquake dataset from past event using machine learning such as Random Forest and Logistic Regression to perform binary classification of tsunami events based on seismic, geospatial features enhanced the accuracy and lead time of tsunami forecasting (Satish et al., 2025). The performance of ML model improves the predictive performance in both tsunami and earthquake magnitude forecasting enabling the extraction of complex temporal and spatial trend from high-dimensional seismic data (Kaftan, 2025).

### 2.6.1 Extreme Gradient Boosting (XGBoost) Algorithm

Extreme Gradient Boosting algorithm is a powerful machines for earthquake analysis through statistical boosting methods and build classification ad regression trees by integrating multiple trees (Figure 2.5) into a consensus prediction framework to improve the model performance on tabular and structured data (Chen and Guestrin, 2016, J. Wang et al., 2025 ). Extreme Gradient Boosting or XGBoost model is generating from a simple decision tree to a more complex system tree that is designed to fix the error from the original decision tree to create new tree. The outcomes from the trees will become predictions results. This model can measure based on equation (2.1):

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \tag{2.1}$$

Where:

- $\widehat{y_i}$ = predicted value for instance *i*,
- $f_k$ = function (a regression tree) added at the *k*-th step
- *K* = total number of trees.

XGBoost constructs several important innovations such as a sparse-aware algorithm for missing values, a weighted-quantile sketch procedure that facilitates approximate tree learning, and superior regularization methods that rely on preventing overfitting and improving generalization. All these technical innovations allow XGBoost to not only be more efficient in terms of computation than traditional gradient boosting algorithms but also to perform better in terms of prediction, which makes it highly applicable to large-scale data in the real-world situation (Velarde et al., 2024).

The versatility and adaptability given through the ability to implement different objective functions and assessment metrics makes the algorithm flexible enough to suit different constraints making it useful to both research and industries. Empirical studies have consistently demonstrated the excellence of XGBoost across various domains including classifying disaster forecasting including earthquakes, tsunami, landslides and flood by Airlangga, (2025) shown the high accuracy of more than 85% compared to SVM and Neural Network algorithm.

This model demonstrates the high performance accuracy and efficiency for earthquake analysis, can handle large and complex dataset while prevent overfitting since the algorithm incorporates L1 (Lasso) and L2 (Ridge) regulation techniques (Babu et al., 2024). According to Wang et al., (2023), XGBoost well performed in classifying seismic event including earthquakes, explosion and mining induced earthquakes which shown the high accuracy of more than 90% compared to SVM algorithm that indicates the high possibility of seismic events discrimination.

However, XGBoost algorithm need careful handling especially when dealing with a sparse, high-dimensional and complex data. It is extremely sensitive to the hyperparameter tuning process, which means that to find its best performance on minority classes, precision may be swamped and false positives may increase (Velarde et al., 2024). As an example, this model can reduce the sensitivity to imbalance classes in differentiating the mining-induce earthquakes from explosion leading to skewed precision and recall that can cause the bias and affect the prediction accuracy especially for nonlinear structural responses (Babu et al., 2024; Wang et al., 2023).

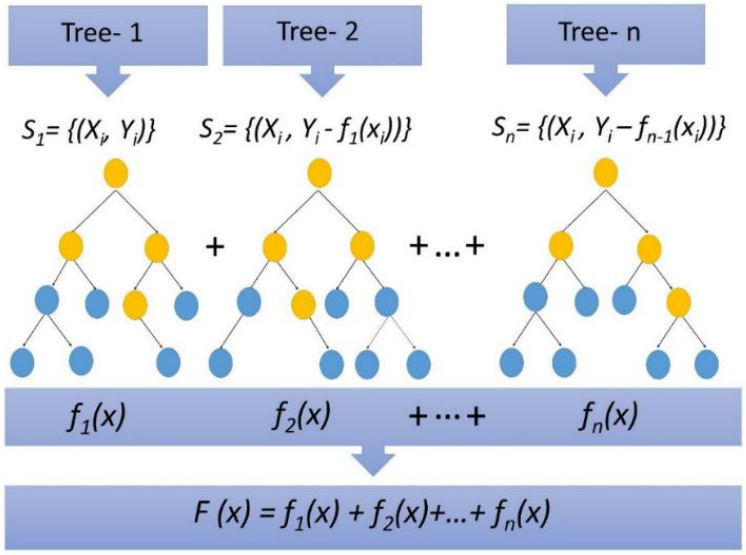



Figure 2.5 Schematic diagram of XGBoost algorithm

### 2.6.2 Naïve Bayes (NB) Algorithm

Naive bayes classifier is a probabilistic model that uses Bayes theorem with a simplifying assumption that the input features are conditionally independent with the class label. This assumption means that all features are independent regarding the likelihood of an outcome although there can be dependencies between features. The model computes the posterior probability to predict the likelihood of a single-class (C) using a given a set of features $X=(x_1,x_2,\ldots,x_n)$) that can be calculated using equation (2.2) and illustrated like Figure 2.6:

$$P\ (\mathbf{X} \mid \mathbf{C})\ = \frac{\mathbf{P}(\mathbf{X} \mid \mathbf{C}).\mathbf{P}(\mathbf{C})}{\boldsymbol{P}(\boldsymbol{X})} \tag{2.2}$$

Where:

- P(C) = the prior probability of the class
- P(X|C) = the likelihood of observing the features given the class
- P(X) = evidence.

In classification problems, the denominator is the same in all classes. Therefore, the decision rule involves choosing that class, which maximizes the numerator that the product of the prior probability and the likelihood. The type of Naive Bayes variant selected is determined by the characteristics of the features such as the multinomial formulation of Naive Bayes works with count data, the Gaussian formulation with continuous attributes and the Bernoulli model with binary variables.

Naive Bayes algorithm has been utilized in research about disasters and earthquakes as a simple but useful classification tool. This approach was used by Chandu, (2025) to categorize news reports that are related to disasters as either highly or low newsworthiness. Naive Bayes is a computationally efficient reference point, even though deep learning methods like Multi-Layer Perceptrons were found to be more accurate, but it is suitable in real-time monitoring applications where fast processing is crucial.

Likewise, Karimzadeh et al., (2019) used the Naive Bayes classifier to predict aftershock distributions after a significant earthquake using data such as slip distribution, Coulomb changes in stress, and features of proximity to faults. The Naive Bayes model performed better than the other Random Forest and Support Vector Machine models even though its performance was superior but the results were interpretable and could be obtained quickly hence supporting the rapid mapping of hazards.

Moreover, Ertuncay & Costa, (2021) have recently developed a multivariate Naive Bayes classifier to discriminate between impulsive and non-impulsive near-fault ground-motion signals. Their model assigned probabilities to hazardous impulsive signals by including earthquake magnitude, rupture geometry and site distance, which is important especially for tsunami-generating earthquakes, where such impulsive signals are often indicative of rupture directivity into the ocean.

All these studies together indicate that even though Naive Bayes is not the most sophisticated algorithm, it still has significant applicability in disaster management. Its advantage is that it is simple, interpretable, and computational efficient, which makes it a practical option in real-time earthquake and tsunami early-warning systems (Chandu, 2025; Karimzadeh et al., 2019; Ertuncay and Costa, 2021).

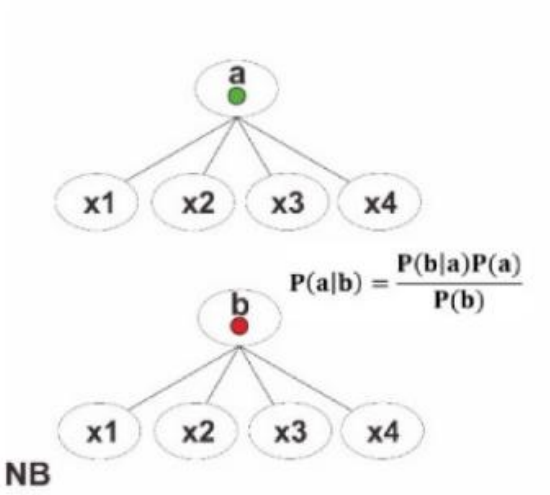



Figure 2.6 Schematic diagram of Naïve Bayes algorithm (Karimzadeh et al., 2019)

### 2.6.3 Random Forest Algorithm

Random Forest (RF) algorithm widely used in predicting earthquake magnitude based on diverse seismic indicator and time series features with high accuracy from the historical pattern of seismic data (Kukartsev & Degtyareva, 2024; Novick & Last, 2023) and to identify the occurrence of tsunami based on binary classification (Satish et al., 2025). RF is developed based on the empirical studies with multiple decision trees and works on the over fitting problem of decision trees by combining the decision with the votes of multiple decision trees (Figure 2.7).

In contrast to boosting and adaptive bagging, Random Forest does not adjust the training set iteratively. Rather, it is based off specific random inputs and features and would provide better classification results as opposed to normal regression (Breiman, 2001). The good performance of random forest classifier is likely due to independence of individual trees in the forest. This independence causes the overall decisions of the trees to be more correct than the decisions each tree makes independently. The high correlation between the trees will inherently have a positive effect on the accuracy of the results of the algorithm (Chang et al., 2023)

In addition, Sukamana et al. 2024 recorded the higher model prediction performances of RF model compared to Support Vector Machine (SVM) model in analyse local seismic features influence on damage of the 6 February 2023 Türkiye Earthquakes. The results that indicate the performance of higher accuracy of 61.15%, precision of 50.00%, higher recall of 36.07%, F1-score of 41.90%, and ROC AUC of 63.84% in predicting.

Even though RF show the high performance accuracy, sometimes it can be overfitting when the model is too complex that tend to capturing noise data than the underlying trends in the data (Z.-N. Wu et al., 2024). This model also struggles with the highly nonlinear relationship of seismic data where sometimes the other machine learning like Gradient Boosting Machine (GBM) and Support Vector Machine (SVM) shown better performance in determining these nonlinearities.(Babu et al., 2024; Cornely & Wang, 2023).

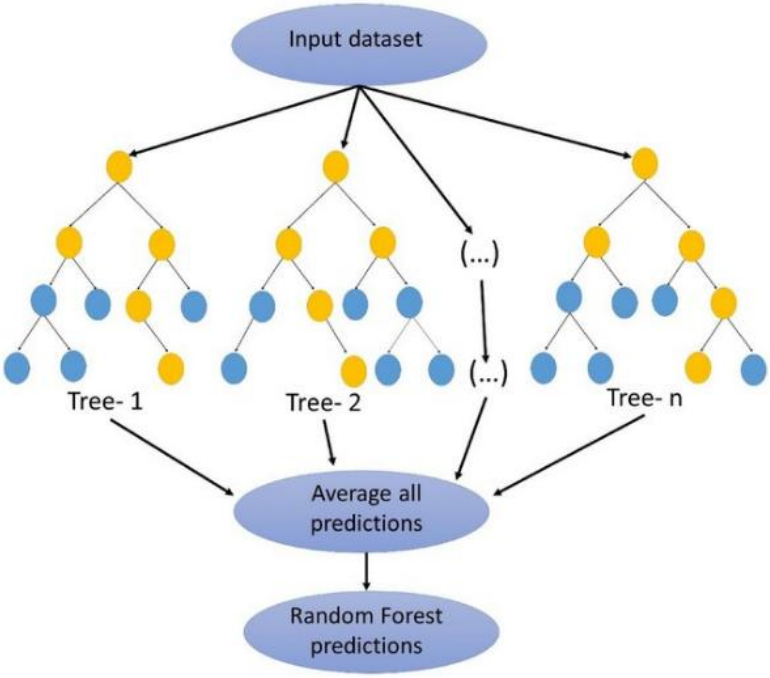



Figure 2.7 Schematic diagram of Random Forest algorithm

### 2.6.4 Support Vector Machine (SVM) Algorithm

Support Vector Machine (SVM) is another supervised machine learning that can be used for analyzing earthquakes due to its ability to robust the high dimensional data, identifying the complex pattern, capability of modelling non-linear relationships through kernel techniques and suitable for classification and regression for predicting the earthquake features like earthquake magnitude (Mahmoud et al., 2025; Satish, Gonaygunta, Yadulla, et al., 2025).

According to Navarro et al., (2025), SVM shows the stable performance with limited features but high accuracy for small dataset rather than Random Forest and LSTM. However, SVM models are not suitable for handling nonlinear dataset without optimal kernel. As SVM heavily depends on selection of kernel functions and parameters this models will become complex and time consuming when deal with massive datasets that require modifications to improve scalability and efficiency (Figure 2.8) (Hoque et al., 2020; Mahmoud et al., 2025; Navarro et al., 2025).

Based on performance in earthquake analysis, SVM shows the accuracy until 98% for classification of earthquake magnitude with consistent result of binary classification. Kazbekova et al., (2025) indicates that SVM resulted 72.34% of accuracy higher than Logistic Regression and Decision Tree but lower than accuracy of XGBoost and Random Forest that has accuracy of 74.57% and 73.18% respectively.

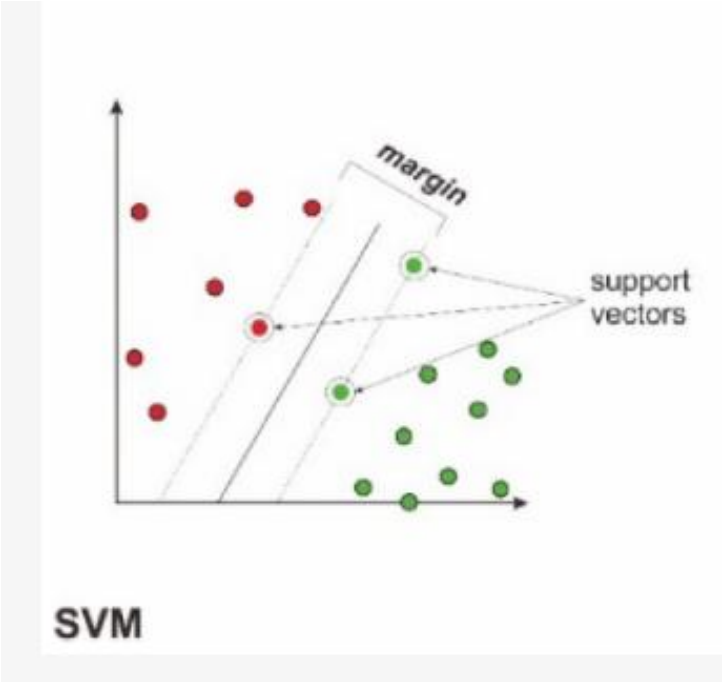



Figure 2.8 SVM schematic diagram by Karimzadeh et al., (2019)

### 2.6.5 Logistic Regression (LR) Algorithm

Logistic Regression (LR) is a classical statistical method widely used in data science and machine learning especially for binary classification. It models the probability of an event based on a linear combination of input features, thus allowing the assessment of the probability of an earthquake triggering a tsunami (tsunami-genic) or not. The main strengths of LR are its simplicity and extensibility, which make it a suitable choice for researchers and policymakers. According to Satish et al. (2025), LR serves as a standard model due to its ability to provide clear interpretation and as a benchmark point for evaluating the performance of more complex algorithms such as Random Forest or XGBoost.

Karimzadeh et al. (2019) compared LR with SVM and KNN in aftershock prediction and found that LR, although less superior in terms of accuracy, still provides advantages in terms of interpretation and computational speed. Similarly, Maazallahi et al., (2025) confirmed the role of LR in the classification of seismic event parameters, where it is more suitable as a baseline model before using complex algorithms.

Overall, Logistic Regression remains important in tsunami prediction research. Although its performance may not be comparable to modern models, it offers advantages in interpretability, efficiency, and integration of multivariate data. Therefore, LR not only serves as a basic tool in tsunami classification, but also as an important reference in the development of more complex disaster prediction models.

### 2.6.6 K-Nearest Neighbor (K-NN) Algorithm

K-Nearest Neighbour (k-NN) is a simple supervised machine learning algorithm and non-parametric that used for combination of classification and regression to detect similarity or close average distance between data points in a multidimensional environment (Figure 2.9). using Euclidean distances of the response variables, and an inverse-squared distance method to generate weights (Karimzadeh et al., 2019).

Meanwhile, KNN is widely used in tsunami early warning classification due to its simplicity and ability to detect patterns based on the distance between data points. Narne, (2023) showed that KNN provides competitive tsunami prediction results, although it faces the challenge of large data scale that requires computational optimization. Furthermore, Bustos et al., (2024) reported that KNN is effective in classifying seismic event features for tsunami early warning systems although its accuracy is strongly influenced by the choice of k parameter and distance metric.

Truong et al., (2025) applied KNN algorithm for evaluating the seismic safety classification of steel frame related to ground motion by classifying the “safe” or “unsafe” region based on the close distance of seismic point response that likely similar to other known designs. The performance of this method not good compared to other supervised algorithm, but the simplicity of this algorithm that does not require

predefined areas or normalization making it statistically robust for analyzing the seismic intensity field in two seismogenic regions like Kuril Island and Japan by (Pisarenko & Pisarenko, 2022) .

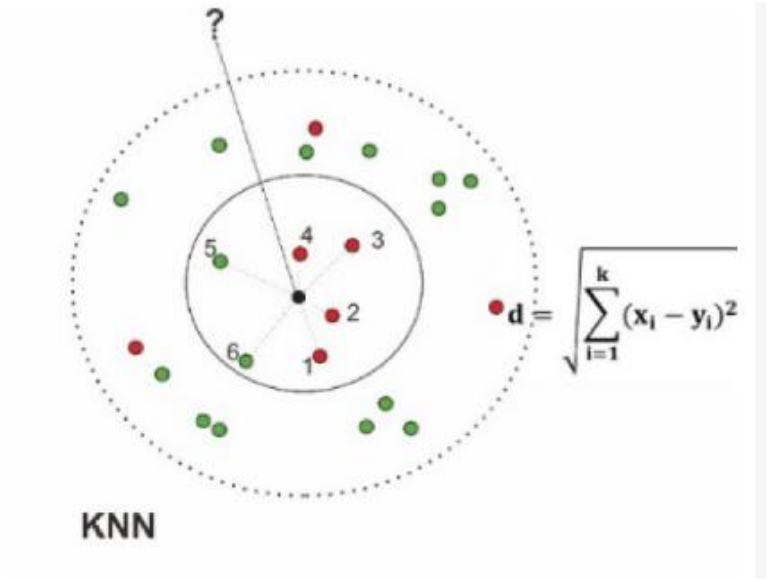



Figure 2.9 k-NN schematic diagram by Karimzadeh et al., (2019)

### 2.6.7 Conclusion of Supervised ML

Table 2.2 shows the summary of the application of multiple supervised ML in earthquake and tsunami prediction using Kaggle dataset. Based on this study, there are a just few studies on the XGBoost and Naïve Bayes model compared to Random Forest and SVM that most popular in tsunami classification. Although rarely applied, both models have their own advantages: XGBoost offers high accuracy, the ability to handle large-scale data and nonlinear relationships better, while Naïve Bayes is lightweight, easy to interpret and suitable for real-time forecasting applications. Therefore, this selection not only provides an opportunity to evaluate the effectiveness of two potential alternative models but also fills a research gap by expanding the variety of tsunami prediction methods using machine learning.

Table 2.2 Summary of machine learning algorithms used for earthquake and tsunami analysis

| Tittle | Model used | Dataset | Outcomes |
|---|---|---|---|
| Forecasting the Unseen: Enhancing Tsunami Occurrence Predictions with Machine-Learning-Driven Analytics<br><br>(Satish et al. 2025) | - Random Forest<br>- Logistic Regression | Kaggle dataset of earthquake events from 1995–2023. | The Random Forest model showed the best performance with an accuracy of 90% and a precision of 88% compared to the Logistic Regression model which recorded an accuracy of 89% and a precision of 87%. These results highlight the usefulness of the algorithm of Random Forest in dealing with imbalanced data. |
| Comparative Analysis of SVM and RF Algorithms for Tsunami Prediction: A Performance Evaluation Study<br><br>(Sukmana et al., 2024) | - Random Forest<br>- Support Vector Machine (SVM) | Kaggle dataset contain comprehensive collection of earthquake records from 1 January 2001 to 1 January 2023. | The comparison of Support Vector Machine (SVM) and Random Forest (RF) models to predict tsunami proved that SVM achieved a high precision of 70.591% but a low recall of 19.671% whereas RF had a high recall of 36.071% at the expense of a low precision of 50%. These results mean that SVM is more efficient at reducing false alarms, and RF is more efficient at identifying actual tsunami events, thus highlighting the trade-off that necessarily exists when choosing the right model to disaster management. |

| | | | |
|---|---|---|---|
| Disaster Prediction Using Appropriate Machine Learning Techniques<br><br>(Bangar et al., 2024) | XGBoost<br>SVM<br>Neural Network | Open-source dataset of global disaster. | XGBoost was found to be effective in prediction of various natural calamities such as tsunamis, landslides, floods, and earthquakes. The findings indicate that XGBoost was more effective than support vector machines and neural network models in predicting all the disaster types with an overall accuracy, recall, precision, ROC, and AUC more than 90.5% in disaster prediction of floods, tsunamis, earthquakes, and landslides. |
| Classification of earthquakes, explosions and mining-induced earthquakes based on XGBoost algorithm<br><br>(T. Wang et al., 2023) | XGBoost<br>SVM | Earthquake catalogue, the unnatural seismic event catalogue and waveform dataset from December 2007 to December 2016 collected from Beijing Digital Seismic Network and Data Management Centre of the China National Seismic Network. | This paper shows that feature extraction significantly improves seismic event discrimination with an accuracy of 97.48 and 95.12 when discriminating between earthquakes and explosions. XGBoost outperformed SVM in all tasks with high accuracy in the classification of earthquakes (96.41 %), explosions (90.38ʿ%), and earthquakes caused by mining (94.04ʿ%), thus confirming its immense potential in the classification of seismic events. |

## 2.7 Research Gap

Studies of tsunamis since the early 20th century have largely focused on traditional geoscience approaches using geological, seismological, and tectonic mapping methods to explain the formation of tsunami. Most studies have focused on tsunami events in the Pacific Ring of Fire of subduction zone where large and high-magnitude earthquakes often occur. However, recent cases such as the 2018 Palu tsunami caused by a horizontal fault earthquake and submarine landslide and the 2018 Anak Krakatau tsunami caused by volcanic eruptions have proven that tsunamis are not necessarily caused by shallow tectonic earthquakes alone. This creates a gap in the literature because previous studies still place excessive emphasis on classical tectonic tsunami while non-tectonic tsunami sources are still poorly explored systematically.

In addition, there are also limitations in traditional geoscience-based tsunami forecasting systems. Early warning systems still rely heavily on parameters such as magnitude, depth, and epicenter location. While these parameters are important, they are not sufficient to distinguish between earthquakes that can generate tsunamis and those that do not. Furthermore, time delays in calculating magnitude and fault mechanisms often result in late forecasts while in certain areas tsunamis can reach the coast in less than ten minutes. This gap indicates that traditional systems are not fast enough to protect coastal areas close to the epicenter.

In the context of non-seismic tsunami sources, the research gap is becoming increasingly clear. Tsunamis from underwater volcanic eruptions, such as in Tonga in 2022, fail to be detected by traditional systems because they do not involve large earthquakes. The same goes for tsunamis from volcanic flank collapses and underwater landslides, which are difficult to predict with only using seismic data. Although modern satellite technologies such as InSAR and ocean altimetry are capable of detecting changes in the shape of the earth and sea surface, their uses is still limited and has not been fully integrated into global early warning systems. This creates a research gap because the integration of data from various sources, such as satellite deformation, acoustic data, and volcano monitoring, is still rarely applied operationally.

Furthermore, in the era of big data, thousands of earthquake events are recorded annually by global seismographs. However, most tsunami studies still use conventional statistical methods that are slow and inflexible. The development of machine learning (ML) opens up great opportunities to analyse hidden patterns in earthquake data, but its application is still in its infancy. Algorithms such as Naïve Bayes (NB) and XGBoost have proven effective in handling high-dimensional, unbalanced, and noisy data in the field of data science but their uses in earthquake and tsunami prediction is still limited. Moreover, research only focuses on seismic signal analysis or magnitude prediction, without considering other geological factors such as fault systems, rock types, and tectonic stress regimes. This lack of integration of geological factors makes predictions less effective and difficult to interpret scientifically.

Additionally, small dataset sizes, incomplete data, and the lack of detailed geological information in most studies have limited the ability of ML models to produce accurate predictions. This lack of data prevents the use of important geological concepts such as stress buildup, rock bending, and fault connections in ML analyses. This highlights a significant gap between geoscience expertise and data science expertise. The lack of collaboration between experts in both fields has limited the more holistic application of ML in the context of earthquake and tsunami prediction.

Previous studies have also shown an imbalance in geographical focus. Majority of tsunami research has focused on the Ring of Fire regions such as Japan, Indonesia, and Chile, while areas outside the Ring of Fire such as Turkey, Myanmar, and the Mediterranean have received less attention. In fact, large earthquakes occurring outside the subduction zone, such as in Turkey (2023–2024) and Myanmar (2025), also have the potential to pose a tsunami risk, although they are rare. The lack of a robust warning system in these regions further adds to the gap in the literature, especially in terms of the application of ML to predict tsunami risk in non-subduction regions.

Therefore, it can be concluded that there are several major gaps in the tsunami and earthquake prediction literature. These include an over-focus on classical tectonic tsunamis, traditional prediction systems that are too slow, a lack of non-seismic tsunami studies, limited datasets, and the use of ML that is still limited and does not fully integrate geological factors. To address these gaps, future studies need to develop a more

comprehensive approach by combining geoscience, data science, and hybrid ML algorithms. This approach will enable the development of more accurate, reliable, and real-time prediction models. More importantly, it will assist in disaster mitigation strategies, urban planning, and community risk management in tsunami-prone areas.

## 2.8 Chapter Summary

This chapter highlights the comprehensive review of earthquake formation and tectonic settings that highlights the main mechanism at various earthquakes and the worldwide distribution of seismic activity especially at the Pacific Ring of Fire such as Indonesia that commonly generated the tsunami due to high magnitude earthquake. This chapter also describe the supervise machine learning model applications used for earthquake analysis and prediction using multiple ML for predicting and classifying earthquakes that generate secondary hazard of tsunami. There is a clear gap in research for combining geological information and machine learning indicating the necessity of using several disciplines to enhance prediction and decision-making abilities.

# CHAPTER 3

# RESEARCH METHODOLOGY

## 3.1 Introduction

This chapter discussed the comprehensive framework applied in this research to analysing and predicting the patterns of earthquake occurrences using machine learning algorithm of XGBoost and NB. This research is built on earthquake events dataset collected from Kaggle. The research methodology presents the detailed research process from the problem identification and initial study of the topic to the evaluation of the developed model. The evaluation performance of the machine learning model used also discussed in this chapter.

## 3.2 Research Framework

The methodology is designed into seven stages to analyze and predict earthquake events including problem identification and initial study, data collection, data pre-processing, exploratory data analysis (EDA), feature selection, model development, evaluation and validation and data visualization. The stages of this research framework are illustrated in flow diagram Figure 3.1. The work schedule of this research framework in a year was shown in Appendix A.

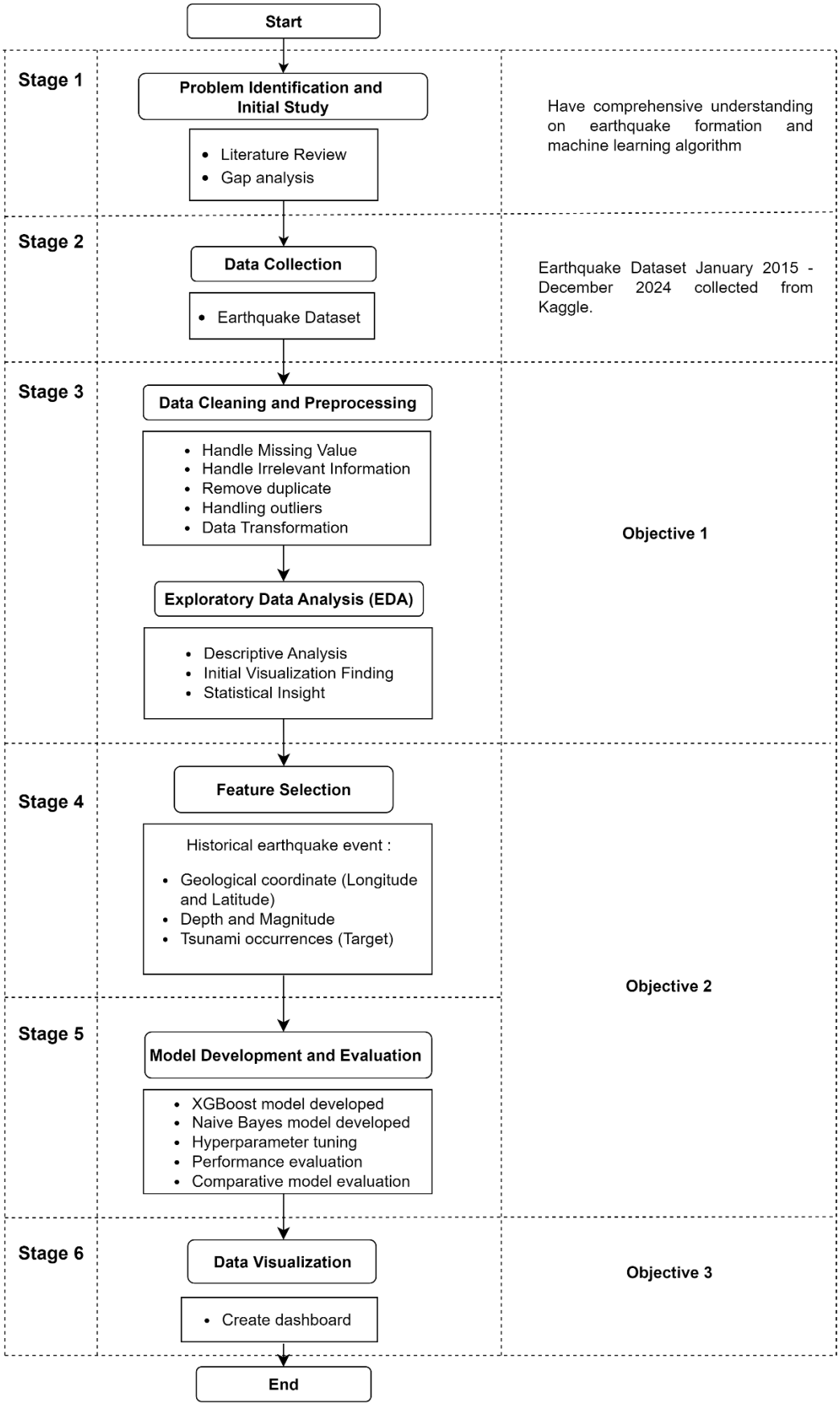



Figure 3.1 Flow diagram framework

### 3.2.1 Stage 1: Problem Identification and Initial Study

This primary stage is a foundation of research study which is important before the research can be conducted. In this stage, the research problem was identified based on literature review of the related study of earthquakes and machine learning to gain information and understanding of the research field. The occurrences of earthquakes are unpredictable and become the main challenge for experts to minimize damage and casualties through data-driven predictions. This study aimed to predict the tsunami-generating earthquake.

### 3.2.2 Stage 2: Data Collection

The global earthquake dataset from Kaggle is used in this research. The main source of this dataset is from USGS Earthquake API (Real-time GeoJSON feed). The USGS dataset used by many researchers to train machine learning models for several algorithms such as Decision Tree, KNN, Random Forest, Gradient Boost, XG Boost, SVM, and Ridge Regression to predict earthquake magnitude (Airlangga, 2025; Satish, Gonaygunta, Yudalla, et al., 2025; Sukmana et al., 2024)

This extracted dataset consist of recorded earthquake event from January 2015 until December 2024 consists of 72583 records and 11 features with magnitude 4.5 and above. This dataset saved as .csv file with total of 72583 records and 11 features, which include date and time of the earthquakes, geographical location (latitude and longitude ) on earth and the location of earthquakes to the epicentre with the factors that influence this activity such as magnitude, depth_km, alert, type of seismic and possibility of tsunami such in Table 3.1.

Table 3.1 Key earthquake dataset

| Attribute | Description | Data Type |
|---|---|---|
| time | Timestamp of the earthquake (UTC) | Object |
| latitude | Geographic latitude of the earthquake epicenter | Float64 |
| longitude | Geographic longitude of the earthquake epicenter | Float64 |
| depth_km | Depth of the earthquake in kilometers | Float64 |
| magnitude | Earthquake magnitude | Float64 |
| place | Specific earthquake location description (such as 100km NW of Anchorage, Alaskaâ) | Object |
| type | Type of seismic event like earthquake, quarry blast | Object |
| alert | Alert level issued (green/yellow/orange/red), if any | Integer |
| tsunami | Binary classification (1 is tsunami was triggered, else 0) | Float |
| status | Review status of the data (reviewed, automatic) | Object |
| id | USGS event identifier(s) | Object |

### 3.2.3 Stage 3: Data Cleaning and Pre-processing

Data cleaning and pre-processing is an important step to make sure the quality and reliability of the dataset before operating machine learning models. The cleaning process includes deleting the unnecessary column, remove duplicate value and filled the missing rows with mean, median and mode. The missing values, outliers, inconsistent formats of the earthquake dataset must check to improve the accuracy of model. The outliers with unrealistic values such as negative or extreme need to remove to avoid skewed model training (Harirchian et al., 2021).

Exploratory Data Analysis (EDA) is the early procedure of analyzing datasets to discover and grasp the key features through visualization and statistics before modelling. This process will uncover patterns, trends, visualize distribution, correlations and detect anomalies, missing value and outliers in the earthquake dataset then summarize it using

descriptive statistics through plots such as plotting magnitude and depth distributions using histograms, scatterplots, and summary statistics (mean, median, skewness). This helps detecting imbalances data such as rare high-magnitude earthquakes, which may require resampling techniques (Senkaya et al., 2024)

The benefits of this process allow data scientists or seismologists to see the entire dataset clearly and handle the data cleaning, selecting the best features and the most effective modelling strategies. The application of EDA reduces the errors, enhances the accuracy and reduces time consuming in later analysis. Effectively, EDA will verify the fitness of the data for further use in the machine learning flow (Cui et al., 2024).

### 3.2.4 Stage 4: Feature Selection

Feature selection and dimensionality reduction methods are utilized to improve and optimize the feature set. Features like earthquake magnitude, depth, longitude, latitude and target feature of tsunami occurrences are chosen to test and train the models. Correlation analysis, redundant features such as overlying temporal variables with high collinearity are identified and removed to increase model efficiency and decrease overfitting. Tree-based XGBoost provide strong feature importance scores allowing elimination of non-informative predictors to produce more reliable and accurate results by choosing the best splitting classes and geospatial trends (Kaftan, 2025).

### 3.2.5 Stage 5: Model Development and Evaluation

This stage involved the model development and evaluation of XGBoost and NB models. The cleaned dataset and the relevant selected feature were split into training and test set. The training and testing split with range 80% training to ensures model has enough data to learn the pattern, relationships and trends within the dataset encompassing features and the target variable while the 20% test for evaluating the performance accuracy of the hidden data (Senkaya et al., 2024). This range is enough for the model to perform well with sufficient data without having overfit or underfitting. Randomization is used in the process of splitting data that allowed both the training and testing sets to be

representative of the overall distribution of the dataset, hence ruling out the possible biases of ordered or non-random datasets.

Naive Bayes model training is based on estimating the class priors and class-conditional probabilities of the features under the assumption of conditional independence. The model then compares the posterior probabilities of each of the available classes based on the Bayes rule at prediction time and chooses the one with the greatest posterior probability (Jiang et al., 2020). Meanwhile, the XGBoost model built from the simple decision tree to more complex tree with the aim of fixing the errors of the previous gradient boosting. The hyperparameters of the number of trees, learning rate, and tree depth are modified either through cross-validation to demonstrate the best-performing configuration. (Babu et al., 2024).

Hyperparameter tuning is the optimization of extrinsic settings of a model values that are not directly learned during data training that used to increase performance of a model on a task. The parameters regulate the behaviour of the learning algorithm and help avoid the underfitting and overfitting problems while ensuring both robust model optimization and reliable performance evaluation. (Chen & Guestrin, 2016). The combination of techniques such as grid search, k-fold CV, and stratified k-fold CV with evaluation metrics allow the comprehensive evaluation of classifiers and ensemble learning algorithms including XGBoost (Chen & Guestrin, 2016).

As the data of tsunami in this dataset is imbalanced, the hyperparameter tuning of GridSearchCV cross-validation and Stratified 10-Fold Cross-Validation applied to provides a robust framework for ensuring fair evaluation across imbalanced data (C. Wang et al., 2020). GridSearchCV searches through all possible combinations of hyperparameters including learning rate, depth and strength of regularization and stratified splitting make sure that each fold contains the same percentage of cases affected by a tsunami and other cases, as shown in Figure 3.2.

As the dataset is large, 10 fold were used to give more stable and less biased especially for this imbalanced dataset while can reduce variance and enhance the robustness evaluation (Raschka, 2020; Wong & Yang, 2017; Yadav & Shukla, 2016). This stratification will be essential to avoid biased outcomes since these would otherwise

tend to favour the majority group. This method provides unbiased estimates of generalizability and estimates the best hyperparameter set that maximizes recall or F1-score, a much more descriptive measure of rare-event prediction than accuracy (Pedregosa et al., 2011; Saito & Rehmsmeier, 2015a; Senkaya et al., 2024).

This enables ensemble models XGBoost and NB to be systematically calibrated to early warning systems in which false negatives (missed tsunami event) are the most important metric to minimize. Afterward, XGBoost and NB model is evaluated using 20% test set to measure its generalization performance on different data. The confusion matric and key metrics of accuracy, precision, recall, and F1-score used for classification. Both models provide the importance features to identifying the best performance of the key metrics for the predictions. In tsunami detection, the more important evaluation metric is Recall and F1-score that used on an imbalanced task rather than accuracy or AUC, which can be misleading when applied to a skewed dataset (Saito & Rehmsmeier, 2015).

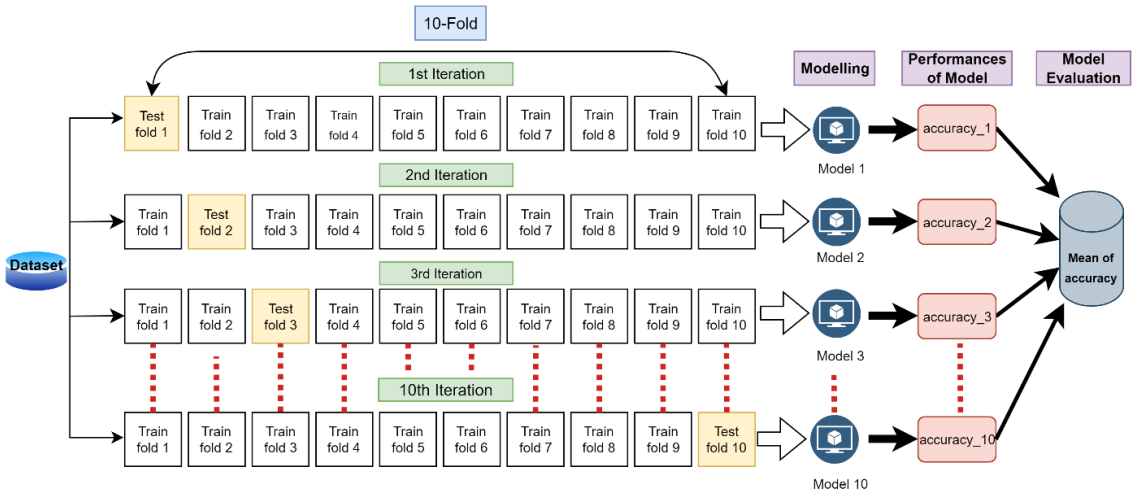



Figure 3.2 10-fold Stratified Cross Validation illustration

### 3.2.6 Stage 6: Data Visualization

The final step for this research is data visualization using dashboard. The results of the earthquake analysis and prediction from the modelling visualized using dashboard in Power BI. The output from tsunami prediction models of XGboost and Naïve Bayes in google colab exported into CSV file and saved in Google Drive. This dataset is linked to Power BI using default connector and can be updated automatically or periodically through an API.

Then, the visualization dashboard was built using information like magnitude, depth, location and status of tsunami in the dataset for further prediction and analyzing of tsunami-generating earthquakes. In addition, Power BI dashboards also support real time monitoring of streaming datasets support or DirectQuery without requiring manual uploads. This interactive dashboard allows the users to explore model outputs and key insights from the data for better understanding and enhance the accuracy of decision-making.

Graphical features such as bar chart and time series plots assist users understand the results of the tsunami-generating earthquake and identify the factors influence the predictions. The spatial of earthquake data visualized into maps to demonstrate predicted risk areas or historical earthquake allocations, presenting the analysis more understandable and practical for decision-making.

## 3.3 Performance Metrics:

### 3.3.1 Confusion Matrix

Confusion matrix used to determine model performance for earthquakes events in binary classification for model assessment. In classification tasks, reliability of a certain model is traditionally evaluated using the measure of a few performance indicators. One of the major tools involved in these evaluations is the confusion matrix represented in Table 3.1 that indicates the true positives, stands for the false positives, represents false negatives, and is equivalent to true negatives.

Table 3.2 Confusion Matrix

| **Actual / Predicted** | **No Tsunami (Class 0)** | **Tsunami (Class 1)** |
|---|---|---|
| **No Tsunami (Class 0)** | TN | FP |
| **Tsunami (Class 1)** | FN | TP |

### 3.3.2 Classification Performance Metrics

These measurements counting the precision, recall, accuracy and F1- Score were used to assess the efficacy of the specific model. The equations demonstrated in equations (3.1) until (3.4):

i. **Accuracy:** accuracy works to determine the overall effectiveness of a learning algorithm in the classification task. This may be expressed as a general equation as given in equation (3.1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

ii. **Precision**: represents the number of correct positive result divided by the total positive results. Increased of precision means there will be fewer cases of positive results falsely classified into a negative result category. Equation (3.2) provides the essential calculation for precision.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

iii. **Recall**: Sensitivity or True Positive Rate. It is computed by dividing the number of correctly labeled positive samples in the data set by the total number of positive samples in the data set such as equation (3.3) high percentage implies that less positive samples were not classified.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

**iv. F-1 Score :** F1-score is a balance mean of recall and precision in equation (3.4). It is different to accuracy in that it does not depend on the TN (true negative) result and is not symmetrical when there are changes of classes. It has values between 0 and 1 with 0 being the worst performance while 1 indicating the absence of false negatives (FN) and false positives (FP), which is the best performance (Chicco and Jurman, 2020).

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.4}$$

Where:

- TP = true positive of predicted number
- TN = true negative of predicted number
- FP = false positive of predicted number
- FN = false negative of predicted number

### 3.3.3 ROC and AUC

**Roc and AUC:** Receiver operating characteristic curves (ROC) are two-dimensional graphs where the True Positive (TP) rate is shown on the Y-axis, and the False Positive (FP) rate is shown on the X-axis such in Figure 3.3. The graph generated with equation (3.5) and (3.6) (Fawcett, 2004). (Pedregosa et al., 2011; Saito & Rehmsmeier, 2015a; Senkaya et al., 2024)

$$TPR = \frac{TP}{TP + FN} \tag{3.5}$$

$$FPR = \frac{FP}{FP + FN} \tag{3.6}$$

There are two very important situations in ROC analysis. The former is the ability to correctly predict the relevant condition known as sensitivity and specificity, which is the ability to assign it appropriately. Sensitivity or specificity can have different relationships with respect to threshold value. (Fawcett, 2006; Rish, 2001; Saito & Rehmsmeier, 2015; Zhang, 2011).

The classification performance of various machine learning algorithms in a binary environment is indicated by the AUC values obtained after ROC analysis. Ideally, a TPR of 1 and a False Positive Rate of 0 would give a model an AUC of 0.5, meaning it can no longer distinguish between positives and negatives, but a model that distinguishes positives and negatives would have a TPR value of 1 and a False Positive rate of 0. Practically, AUC over 0.5 expresses a better-than-chance performance, and closer to 1.0 indicating a well-separated suitability and inaccurately underneath 0.5 expressing inappropriate setup or some inappropriate assessment.(Fawcett, 2006; Pedregosa et al., 2011).

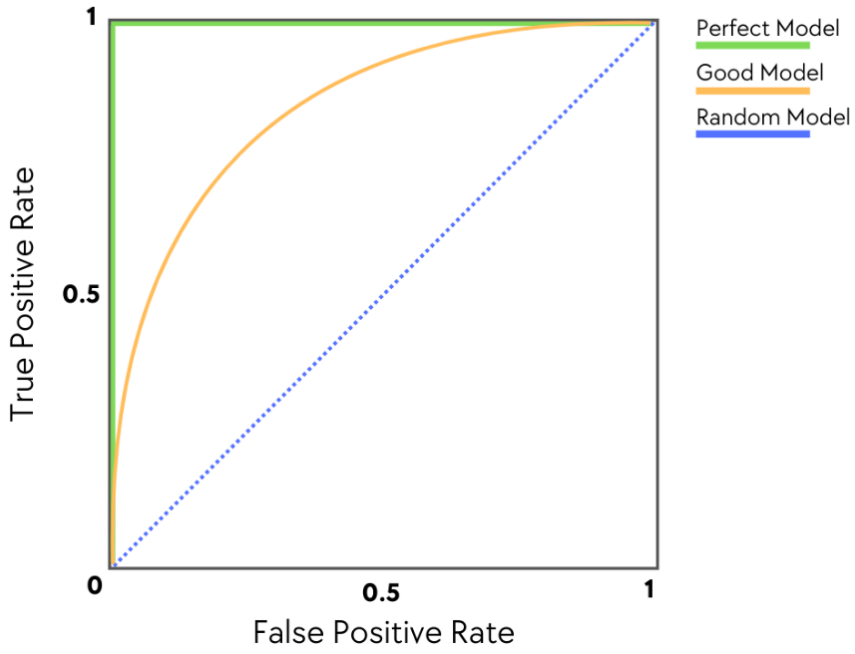



Figure 3.3 Roc and AUC curve

## 3.4 Chapter Summary

This chapter outlines the methodology for analyzing and predicting tsunami-generating earthquakes events using XGBoost and NB model based on global earthquake dataset. The details of the steps discussed in this chapter include the primary study and problem identification, data collection, data pre-processing, exploration data analysis, feature selection, model development and evaluation and visualization of results. The use of statistical and machine learning measuring performance matrices such as accuracy, precision, Recall and F1-score discussed in this chapter to validate the effectiveness in predicting seismic activities.

# CHAPTER 4

# EXPLORATORY DATA ANALYSIS (EDA)

## 4.1 Introduction

This chapter presents the initial findings from the exploratory data analysis (EDA) conducted on the Global Earthquake Dataset. The analysis aims to uncover patterns, relationships, and trends related to earthquake occurrences and their triggers, focusing on global patterns. The findings are presented on statistical summaries, visualizations, and machine learning techniques.

## 4.2 Overview of the Dataset

The dataset named as `df` for earthquake dataset and read using prompt `.read_csv` (Figure 4.1). The information of data presented in Figure 4.2 which used command `.info()`. The details of this dataset explained in 3.2.2.

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Earthquakes data/EarthquakeData (2015-2024) (1).csv')
print("\nEarthquake dataset:")
df.head()
```

```
Earthquake dataset:
```

| | time | place | magnitude | depth_km | longitude | latitude | type | alert | tsunami | status | id |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2015-01-01 05:01:10.640 | near the east coast of Honshu, Japan | 4.8 | 41.39 | 142.0405 | 38.8957 | earthquake | NaN | 0 | reviewed | ,usc000tb3v, |
| 1 | 2015-01-01 06:48:29.670 | 93 km N of Isangel, Vanuatu | 4.6 | 223.61 | 169.1795 | -18.7052 | earthquake | NaN | 0 | reviewed | ,usc000tb42, |
| 2 | 2015-01-01 06:54:20.570 | central Mid-Atlantic Ridge | 4.7 | 10.00 | -31.7641 | 3.4769 | earthquake | NaN | 0 | reviewed | ,usc000tb46, |
| 3 | 2015-01-01 07:12:44.230 | 120 km SSE of Kirakira, Solomon Islands | 4.6 | 26.24 | 162.4998 | -11.3818 | earthquake | NaN | 0 | reviewed | ,usc000tb4a, |
| 4 | 2015-01-01 08:49:53.200 | 70 km W of F?r?z?b?d, Iran | 5.1 | 10.10 | 51.8580 | 28.7280 | earthquake | NaN | 0 | reviewed | ,usc000tb4f,iscgem606436879, |

Figure 4.1 Earthquake dataset

```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72583 entries, 0 to 72582
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   time       72583 non-null  object
 1   place      72583 non-null  object
 2   magnitude  72583 non-null  float64
 3   depth_km   72583 non-null  float64
 4   longitude  72583 non-null  float64
 5   latitude   72583 non-null  float64
 6   type       72583 non-null  object
 7   alert      6427 non-null   object
 8   tsunami    72583 non-null  int64
 9   status     72583 non-null  object
 10  id         72583 non-null  object
dtypes: float64(4), int64(1), object(6)
memory usage: 6.1+ MB
```

Figure 4.2 Earthquake info

## 4.3 Data Cleaning

Data cleaning is important to avoid hopeless inconsistency, error and misleading information in the data that will be analysed into machine learning algorithms. Incomplete and invalid data can lead to lack accuracy, biased models and poor performance affecting the credibility of the decision-making (Harirchian et al., 2021). The datasets of both earthquake and fault cleaned differently before both data merged and undergoes pre-processing step.

### 4.3.1 Check the Column

Prompt of `.columns` used to check the columns by printing the columns names of a data frame in Pandas which is a two-dimensional data structure. The DataFrame output of `df`, contains several columns relevant that are related to earthquake occurrences. This type of columns, for example `'time'`, `'latitude'`, `'longitude'`, `'depth_km'`, `'magnitude'`, `'type'`, `'place'`, `'alert'`, `'tsunami'`, `'status'`,`'id'`, that stores different types of data. The used of `dtype=object` indicate the columns may containing string, integer or date values such in Figure 4.3.

```
df.columns
```

```
Index(['time', 'place', 'magnitude', 'depth_km', 'longitude', 'latitude',
       'type', 'alert', 'tsunami', 'status', 'id'],
      dtype='object')
```

Figure 4.3 Earthquake columns checked

### 4.3.2 Identify Unique Value

Figure 4.4 indicates the identifying unique and non-unique values within a DataFrame. `.nunique()` function used to calculates the number of unique values in DataFrame with variable named `unique_counts`. One column with a single unique value (indicating that there are no unique values per row) is removed after that, and the `unique_counts` list is filtered. Lastly, the names of these non-non-informative columns are also added to a list stated as `non_unique_cols` and are printed out.

```
# Identify unique and non-unique values

# Get the number of unique values in each column
unique_counts = df.nunique()

print("Number of unique values per column:")
print(unique_counts)

# Identify columns with only one unique value (non-unique)
non_unique_cols = unique_counts[unique_counts == 1].index.tolist()

print("\nColumns with only one unique value (non-unique):")
print(non_unique_cols)
```

Figure 4.4 Identify unique and non-unique

### 4.3.3 Dropping Irrelevant Column

Then, the unnecessary columns such as : `'id'`, `'status'` and `'alert'`, dropped from the original data frame using `prompt .drop ()`. Then, the prompt .isnull().sum() used to identify the missing value to avoid from less accuracy of the model. These presented in Figure 4.5 and Figure 4.6.

```
df.drop(['alert','status', 'id'],axis=1, inplace=True)
df.head()
```

Figure 4.5 Remove irrelevant column of earthquakes

```
df.isnull().sum()
```

| | 0 |
|---|---|
| time | 0 |
| place | 0 |
| magnitude | 0 |
| depth_km | 0 |
| longitude | 0 |
| latitude | 0 |
| type | 0 |
| tsunami | 0 |

dtype: int64

Figure 4.6 Identify the missing value

### 4.3.4 Change Datetime Format

The `'time'` column changed to the datetime data type shown in Figure 4.7 The `pd.to_datetime()` function used as conversion function that is significant when performing date-based analysis and operations.

```
df['time'] = pd.to_datetime(df['time'])
```

Figure 4.7 Change datetime format of earthquake dataset

## 4.4 Pre-Processing

Pre-processing stage is important for maintaining the reliability and accuracy of modelling from the cleaned earthquake raw data. After cleaning process, this step will generated new features for providing additional information which can enhanced the analysis and model performance (Harirchian et al., 2021).

### 4.4.1 Adding New Column of Country

To provide a geographical context to the earthquake data, a new column of country created using reverse geocoding enhancement of the same latitude and longitude data. The `reverse_geocoder` library used to enabled conversion of geographic coordinates to respective country codes. It was decided to write a custom function to retrieve the coordinates in every row and do the geocoding lookup and extract the two letters country code related to each event. This was then used over all the data set using the `.apply()` method and the output in the form of the country codes were saved under a new column of `'Country'`. Figure 4.8 indicates the codes and the output of the conversion based on longitude and latitude coordinate. This preprocessing activity allows analysing the data according to the location and makes the dataset easier to interpret.

```
#  generate new column of country based on longitude and latitude

!pip install reverse_geocoder

import reverse_geocoder as rg

def get_country(row):
  coordinates = (row['latitude'], row['longitude'])
  result = rg.search(coordinates)
  return result[0]['cc']

df['Country'] = df.apply(get_country, axis=1)
```

Figure 4.8 New column of country

### 4.4.2 Rename The Country Code

Figure 4.9 shows the step of data preprocessing was replacing country codes in the column of the `'Country'` with full country names to make the data more readable and clearer in interpretation. A dictionary, henceforward referred to as `country_code_to_name`, was created containing each two-letter code country such as 'US', 'JP', 'ID' and the full designation are 'United States', 'Japan', 'Indonesia'. This replacement was implemented by using the column `Country` in DataFrame earthquakes and `.replace()` function and resulted in the dataset in which the full name of countries replaced the shorter codes of countries such in Figure 4.10.

```python
# Mapping country codes to country names
country_code_to_name = {
    'AF': 'Afghanistan', 'AX': 'Aland Islands', 'AL': 'Albania', 'DZ': 'Algeria', 'AS': 'American Samoa',
    'AD': 'Andorra', 'AO': 'Angola', 'AI': 'Anguilla', 'AQ': 'Antarctica', 'AG': 'Antigua and Barbuda',
    'AR': 'Argentina', 'AM': 'Armenia', 'AW': 'Aruba', 'AU': 'Australia', 'AT': 'Austria', 'AZ': 'Azerbaijan',
    'BS': 'Bahamas', 'BH': 'Bahrain', 'BD': 'Bangladesh', 'BB': 'Barbados', 'BY': 'Belarus', 'BE': 'Belgium',
    'BZ': 'Belize', 'BJ': 'Benin', 'BM': 'Bermuda', 'BT': 'Bhutan', 'BO': 'Bolivia', 'BQ': 'Bonaire, Sint Eustatius and Saba',
    'BA': 'Bosnia and Herzegovina', 'BW': 'Botswana', 'BV': 'Bouvet Island', 'BR': 'Brazil', 'IO': 'British Indian Ocean Territory',
    'BN': 'Brunei Darussalam', 'BG': 'Bulgaria', 'BF': 'Burkina Faso', 'BI': 'Burundi', 'CV': 'Cabo Verde',
    'KH': 'Cambodia', 'CM': 'Cameroon', 'CA': 'Canada', 'KY': 'Cayman Islands', 'CF': 'Central African Republic',
    'TD': 'Chad', 'CL': 'Chile', 'CN': 'China', 'CX': 'Christmas Island', 'CC': 'Cocos (Keeling) Islands',
    'CO': 'Colombia', 'KM': 'Comoros', 'CG': 'Congo', 'CD': 'Congo, The Democratic Republic of the',
    'CK': 'Cook Islands', 'CR': 'Costa Rica', 'CI': 'Cote d\'Ivoire', 'HR': 'Croatia', 'CU': 'Cuba', 'CW': 'Curacao',
    'CY': 'Cyprus', 'CZ': 'Czech Republic', 'DK': 'Denmark', 'DJ': 'Djibouti', 'DM': 'Dominica',
    'DO': 'Dominican Republic', 'EC': 'Ecuador', 'EG': 'Egypt', 'SV': 'El Salvador', 'GQ': 'Equatorial Guinea',
    'ER': 'Eritrea', 'EE': 'Estonia', 'ET': 'Ethiopia', 'FK': 'Falkland Islands (Malvinas)', 'FO': 'Faroe Islands',
    'FJ': 'Fiji', 'FI': 'Finland', 'FR': 'France', 'GF': 'French Guiana', 'PF': 'French Polynesia',
    'TF': 'French Southern Territories', 'GA': 'Gabon', 'GM': 'Gambia', 'GE': 'Georgia', 'DE': 'Germany',
    'GH': 'Ghana', 'GI': 'Gibraltar', 'GR': 'Greece', 'GL': 'Greenland', 'GD': 'Grenada', 'GP': 'Guadeloupe',
    'GU': 'Guam', 'GT': 'Guatemala', 'GG': 'Guernsey', 'GN': 'Guinea', 'GW': 'Guinea-Bissau', 'GY': 'Guyana',
    'HT': 'Haiti', 'HM': 'Heard Island and McDonald Islands', 'VA': 'Holy See (Vatican City State)', 'HN': 'Honduras',
    'HK': 'Hong Kong', 'HU': 'Hungary', 'IS': 'Iceland', 'IN': 'India', 'ID': 'Indonesia', 'IR': 'Iran',
    'IQ': 'Iraq', 'IE': 'Ireland', 'IM': 'Isle of Man', 'IL': 'Israel', 'IT': 'Italy', 'JM': 'Jamaica', 'JP': 'Japan',
    'JE': 'Jersey', 'JO': 'Jordan', 'KZ': 'Kazakhstan', 'KE': 'Kenya', 'KI': 'Kiribati', 'KP': 'North Korea',
    'KR': 'Korea, Republic of', 'KW': 'Kuwait', 'KG': 'Kyrgyzstan', 'LA': 'Lao People\'s Democratic Republic',
    'LV': 'Latvia', 'LB': 'Lebanon', 'LS': 'Lesotho', 'LR': 'Liberia', 'LY': 'Libya', 'LI': 'Liechtenstein',
    'LT': 'Lithuania', 'LU': 'Luxembourg', 'MO': 'Macao', 'MK': 'Macedonia, The Former Yugoslav Republic of',
    'MG': 'Madagascar', 'MW': 'Malawi', 'MY': 'Malaysia', 'MV': 'Maldives', 'ML': 'Mali', 'MT': 'Malta',
    'MH': 'Marshall Islands', 'MQ': 'Martinique', 'MR': 'Mauritania', 'MU': 'Mauritius', 'YT': 'Mayotte',
    'MX': 'Mexico', 'FM': 'Micronesia, Federated States of', 'MD': 'Moldova, Republic of', 'MC': 'Monaco',
    'MN': 'Mongolia', 'ME': 'Montenegro', 'MS': 'Montserrat', 'MA': 'Morocco', 'MZ': 'Mozambique',
    'MM': 'Myanmar', 'NA': 'Namibia', 'NR': 'Nauru', 'NP': 'Nepal', 'NL': 'Netherlands', 'NC': 'New Caledonia',
    'NZ': 'New Zealand', 'NI': 'Nicaragua', 'NE': 'Niger', 'NG': 'Nigeria', 'NU': 'Niue', 'NF': 'Norfolk Island',
    'MP': 'Northern Mariana Islands', 'NO': 'Norway', 'OM': 'Oman', 'PK': 'Pakistan', 'PW': 'Palau',
    'PS': 'Palestine, State of', 'PA': 'Panama', 'PG': 'Papua New Guinea', 'PY': 'Paraguay', 'PE': 'Peru',
    'PH': 'Philippines', 'PN': 'Pitcairn', 'PL': 'Poland', 'PT': 'Portugal', 'PR': 'Puerto Rico', 'QA': 'Qatar',
    'RE': 'Reunion', 'RO': 'Romania', 'RU': 'Russian Federation', 'RW': 'Rwanda', 'BL': 'Saint Barthelemy',
    'SH': 'Saint Helena, Ascension and Tristan da Cunha', 'KN': 'Saint Kitts and Nevis', 'LC': 'Saint Lucia',
    'MF': 'Saint Martin (French part)', 'PM': 'Saint Pierre and Miquelon', 'VC': 'Saint Vincent and the Grenadines',
    'WS': 'Samoa', 'SM': 'San Marino', 'ST': 'Sao Tome and Principe', 'SA': 'Saudi Arabia', 'SN': 'Senegal',
    'RS': 'Serbia', 'SC': 'Seychelles', 'SL': 'Sierra Leone', 'SG': 'Singapore', 'SX': 'Sint Maarten (Dutch part)',
    'SK': 'Slovakia', 'SI': 'Slovenia', 'SB': 'Solomon Islands', 'SO': 'Somalia', 'ZA': 'South Africa',
    'GS': 'South Georgia and the South Sandwich Islands', 'SS': 'South Sudan', 'ES': 'Spain', 'LK': 'Sri Lanka',
    'SD': 'Sudan', 'SR': 'Suriname', 'SJ': 'Svalbard and Jan Mayen', 'SZ': 'Swaziland', 'SE': 'Sweden',
    'CH': 'Switzerland', 'SY': 'Syria', 'TW': 'Taiwan', 'TJ': 'Tajikistan',
    'TZ': 'Tanzania', 'TH': 'Thailand', 'TL': 'Timor-Leste', 'TG': 'Togo', 'TK': 'Tokelau',
    'TO': 'Tonga', 'TT': 'Trinidad and Tobago', 'TN': 'Tunisia', 'TR': 'Turkey', 'TM': 'Turkmenistan',
    'TC': 'Turks and Caicos Islands', 'TV': 'Tuvalu', 'UG': 'Uganda', 'UA': 'Ukraine', 'AE': 'United Arab Emirates',
    'GB': 'United Kingdom', 'US': 'United States', 'UM': 'United States Minor Outlying Islands', 'UY': 'Uruguay',
    'UZ': 'Uzbekistan', 'VU': 'Vanuatu', 'VE': 'Venezuela, Bolivarian Republic of', 'VN': 'Viet Nam',
    'VG': 'Virgin Islands, British', 'VI': 'Virgin Islands, U.S.', 'WF': 'Wallis and Futuna', 'EH': 'Western Sahara',
    'YE': 'Yemen', 'ZM': 'Zambia', 'ZW': 'Zimbabwe'
}
```

Figure 4.9 Define country code

```python
# Replace country codes with full names
df['Country'] = df['Country'].replace(country_code_to_name)
```

Figure 4.10 Rename country code to full name country

### 4.4.3 Adding A New Column of Continent

New column of continent generated based on countries name. The `pycountry_convert` package is utilized to convert country names to their respective continents names making it easy to absorb common semantics flaws such the United States is renamed North America. Function `get_continent` used to convert the country names to ISA Alpha-2 codes, which are then mapped to continent codes and mapped to continent labels. This process provides a dictionary that can be used to assign individual countries to the proper continent and to create a new column of Continental into the DataFrame to allow geographical analysis (Figure 4.11).

```
!pip install pycountry_convert
import pycountry_convert as pc

# Function to map country name to continent
def get_continent(country_name):
    try:
        # Handle name discrepancies manually
        rename_dict = {
            "Russian Federation": "Russia",
            "Iran, Islamic Republic of": "Iran",
            "Venezuela, Bolivarian Republic of": "Venezuela",
            "Korea, Republic of": "South Korea",
            "Korea, Democratic People's Republic of": "North Korea",
            "Syrian Arab Republic": "Syria",
            "Taiwan, Province of China": "Taiwan",
            "Viet Nam": "Vietnam",
            "United States": "United States of America",
            "Micronesia, Federated States of": "Micronesia",
            "Tanzania, United Republic of": "Tanzania",
            "Macedonia, The Former Yugoslav Republic of": "North Macedonia",
            "Congo, The Democratic Republic of": "Democratic Republic of the Congo"
        }
        if country_name in rename_dict:
            country_name = rename_dict[country_name]
        country_code = pc.country_name_to_country_alpha2(country_name)
        continent_code = pc.country_alpha2_to_continent_code(country_code)
        continent_name = pc.convert_continent_code_to_continent_name(continent_code)
        return continent_name
    except:
        return "Unknown"

# Map countries to continents
continent_mapping = {country: get_continent(country) for country in countries}
continent_mapping_sorted = dict(sorted(continent_mapping.items(), key=lambda x: x[1]))
df['Continental'] = df['Country'].map(continent_mapping_sorted)
```

Figure 4.11 New column of continental

### 4.4.4 Dropping All the Duplicated Rows

Duplicate rows were deleted in a dataset (Figure 4.12) to maintain data quality and integrity by using the `drop_duplicates()` method of Pandas library. This method searches the whole DataFrame and drops the row that is a complete copy of another in all columns. With prompt of `inplace=True`, the process is done in place of changing the original DataFrame rather than returning a copy of it. The command `df.info()` was then entered to show a short summary of the augmented DataFrame with number of non-null values, data types of columns as well as memory consumption. The result showed that few duplicates were left with final number of 72571 rows and 10 columns, which implied that the dataset was cleaned and no duplicated data. The step is essential in avoidance of data distortions in further analyses and model training.

```
df.drop_duplicates(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 72571 entries, 0 to 72582
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   time         72571 non-null  datetime64[ns]
 1   place        72571 non-null  object
 2   magnitude    72571 non-null  float64
 3   depth_km     72571 non-null  float64
 4   longitude    72571 non-null  float64
 5   latitude     72571 non-null  float64
 6   type         72571 non-null  object
 7   tsunami      72571 non-null  int64
 8   Country      72571 non-null  object
 9   Continental  72571 non-null  object
dtypes: datetime64[ns](1), float64(4), int64(1), object(4)
memory usage: 6.1+ MB
```

Figure 4.12 Removed duplicate

## 4.5 Descriptive Statistics

Figure 4.14 consists of descriptive statistics which provide valuable information concerning the distributional characteristics and the scales of measurement of each of the numeric variables. The variability of depth of earthquakes is quite strong with the minimum of 0 km to the maximum of 683.36 km and the standard deviation being the highest of 115.08 km. This indicates strong disparity of seismic depth where the earthquake event can occur in various and unpredictable depth. Comparatively, magnitude has a considerably medium to higher range 4.5 to 8.3 and a mean magnitude of 4.7 suggesting that most of the events fall into the moderate to mildly intense category. In addition, tsunami relate to earthquake with binary 1 recorded with 1004 cases while the earthquake that not generating tsunami with binary 0 that indicate the events are recorded with 71,567 cases.

| | longitude | latitude | depth_km | magnitude | tsunami |
|---|---|---|---|---|---|
| count | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 |
| mean | 33.335054 | -1.808635 | 63.131648 | 4.803313 | 0.013835 |
| std | 123.129062 | 29.755335 | 115.081476 | 0.371198 | 0.116805 |
| min | -179.999700 | -79.983700 | 0.000000 | 4.500000 | 0.000000 |
| 25% | -72.265150 | -22.231350 | 10.000000 | 4.500000 | 0.000000 |
| 50% | 92.482800 | -5.119700 | 14.200000 | 4.700000 | 0.000000 |
| 75% | 141.492850 | 19.313400 | 57.735000 | 4.900000 | 0.000000 |
| max | 179.999300 | 87.386000 | 683.360000 | 8.300000 | 1.000000 |

```
Number of tsunami events: 1004
Number of non-tsunami events in the dataset: 71567
```

Figure 4.14 Descriptive statistics

### 4.5.1 Depth and Magnitude Distribution

Figure 4.15 shows the histogram of key features distribution of earthquake activities including earthquake latitude, longitude, magnitude and depth. The longitude graph shows the distribution of earthquake higher at longitude -150 to -120 (Pacific region including coast Americas like Alaska) and 120 to 160 (Japan and Southeast Asia like Indonesia and Indonesia). These clearly demonstrate that this region located along tectonic plate of Ring of Fire reflecting the geodynamic processes of strong seismic activities.

Similarly, latitude graph illustrates the distribution of majority earthquake located at the central graph range between -40 to 40 near the equatorial region where most major tectonic interactions occur such as Pacific and Indo-Australian Plates (Indonesia, Philippine, and Australia). These spatial trends confirm that earthquakes are not randomly distributed across the world but focused in the tectonically active plates.

The magnitude graph illustrates that most earthquakes occur in the moderate range 4.5–6.0. The left-skewed magnitude shows that magnitudes occur often at magnitude 4.5 while the high magnitude earthquake (more than magnitude 6) occur less often. This correlates with worldwide seismic trends of moderate earthquakes strike with greater frequency than catastrophic earthquakes.

The depth (km) graph shows that most earthquakes happen at shallow depths, within the first 100 km of the Earth's crust. The frequency decreases as depth increases because shallow earthquakes occur more often and easier to detect. This is important for disaster studies since shallow earthquakes are usually more damaging compared to deep earthquakes.

These graphs also help identify potential outliers, but in geoscience, many of these extreme values (like very deep or very strong quakes) are not errors. Instead, they represent rare but important geological events that provide critical insights for earthquake hazard modelling and risk assessment.

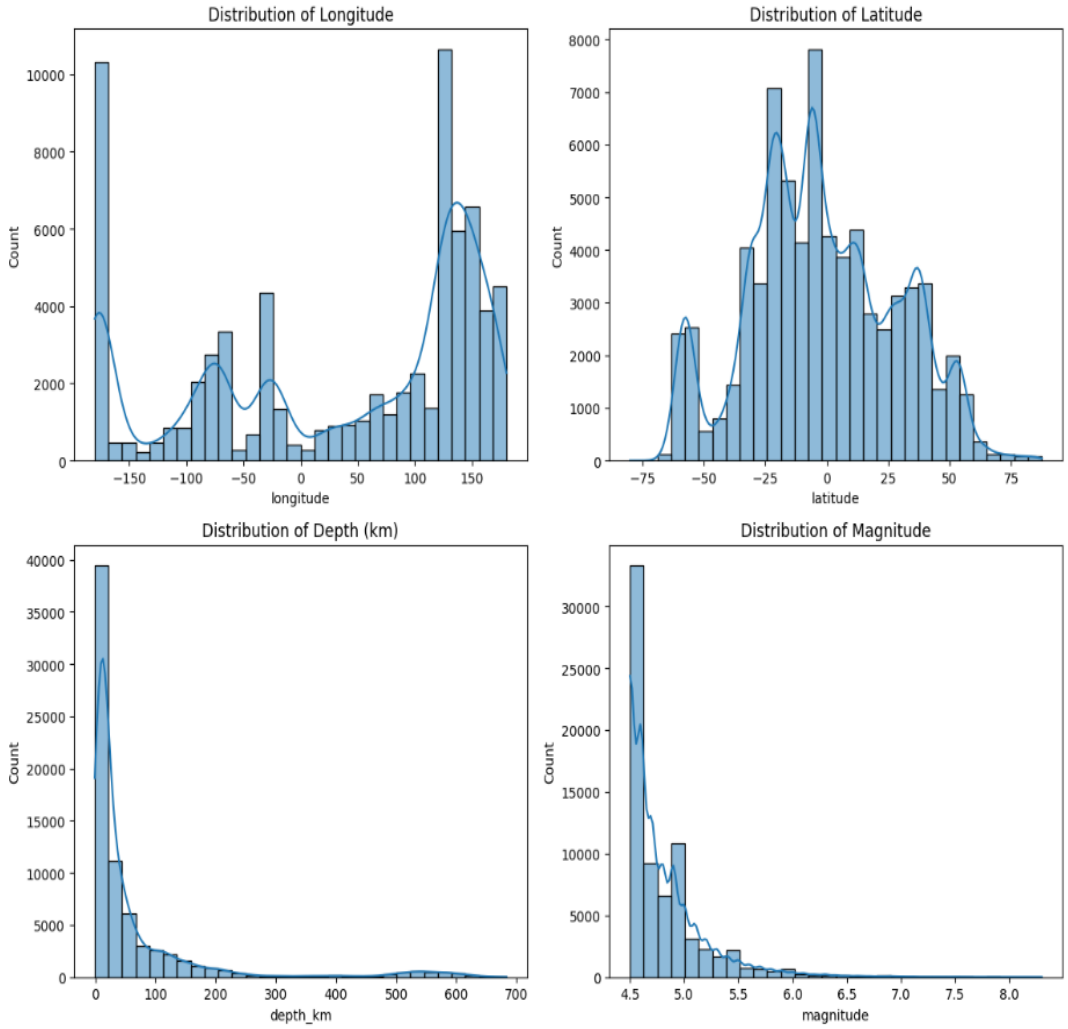



Figure 4.15 Earthquake features distribution

### 4.5.2 Location of Highest Magnitude

The strongest earthquakes recorded worldwide from 2015 to 2024 shown in Figure 4.16. Chile recorded the strongest magnitude of 8.3 earthquake in 2015 due to the subduction of the Nazca Plate underneath South America. Additionally, 8.2 magnitude earthquake recorded at United State Peninsula (likely Alaska), Fiji, and near Mexico's Chiapas region which are located at the major subduction zone that caused the creation of tsunami.

Meanwhile, 8.1 magnitude recorded at the South Sandwich Islands does not creating the tsunami even though it's located at subduction zone. This is because the seismic activity in southern part of the South Atlantic can be attributed mainly to its location in the South Atlantic seismic belt and not in the Pacific Ring of Fire where the subduction zones are wider and more geodynamical active. Therefore, the South Sandwich subduction system is smaller and tectonically lesser than the Pacific margin despite having the massive megathrust earthquakes and tsunamis.

Top 5 Highest Magnitude Earthquakes:

| | time | place | magnitude | depth_km | longitude | latitude | type | tsunami | Country | Continental |
|---|---|---|---|---|---|---|---|---|---|---|
| **4845** | 2015-09-16 22:54:32.860 | 48 km W of Illapel, Chile | 8.3 | 22.44 | -71.6744 | -31.5729 | earthquake | 1 | Chile | South America |
| **46898** | 2021-07-29 06:15:49.188 | Alaska Peninsula | 8.2 | 35.00 | -157.8876 | 55.3635 | earthquake | 1 | United States | North America |
| **24950** | 2018-08-19 00:19:40.670 | 267 km E of Levuka, Fiji | 8.2 | 600.00 | -178.1530 | -18.1125 | earthquake | 1 | Wallis and Futuna | Oceania |
| **18557** | 2017-09-08 04:49:19.180 | near the coast of Chiapas, Mexico | 8.2 | 47.39 | -93.8993 | 15.0222 | earthquake | 1 | Mexico | North America |
| **47257** | 2021-08-12 18:35:17.231 | South Sandwich Islands region | 8.1 | 22.79 | -25.2637 | -58.3753 | earthquake | 0 | South Georgia and the South Sandwich Islands | South America |

Figure 4.16 The highest magnitude of earthquake

## 4.6 Initial Findings Visualization

The initial finding results visualized using matplotlib and seaborn library to generate various plots and graphs that help to understand the earthquake data visually. These visualizations provide extensive information on earthquake features in terms of magnitude and depth distribution, temporal dynamics, impacts based on the location and cross-correlation between different parameters.

### 4.6.1 Correlation Matrix

The Spearman correlation matrix in Figure 4.17 explains the monotonically dependent nature of different attributes of earthquake that related with tsunami occurrences. Overall correlation shows the weak correlation between all the earthquake features where the maximum positive correlation recorded is 0.14 between the magnitude and tsunami and 0.14 between latitude and tsunami.

These results support that earthquakes have a slightly higher tendency to cause tsunamis, which corresponds to geological predictions because higher magnitude earthquakes often cause more seafloor displacement. The relatively poor correlation between latitude and tsunami is consistent with a large proportion of tsunamis focusing within subduction zone that localized to the Pacific Ring of Fire. However, the low magnitude of these correlations indicates that none of these variables or factors can determine causally the formation of tsunami.

Despite the low correlation between depth and occurrence of tsunamis (0.01), the correlation emphasizes that even though very shallow earthquakes can generate most dangerous tsunami, but the fault depth is not a strong predictive variable. Altogether, the heatmap reveals that the process of tsunami generation is inherently multifactorial, the relationship with the fault magnitude, tectonic setting, and fault type is complex rather than linear or monotonic.

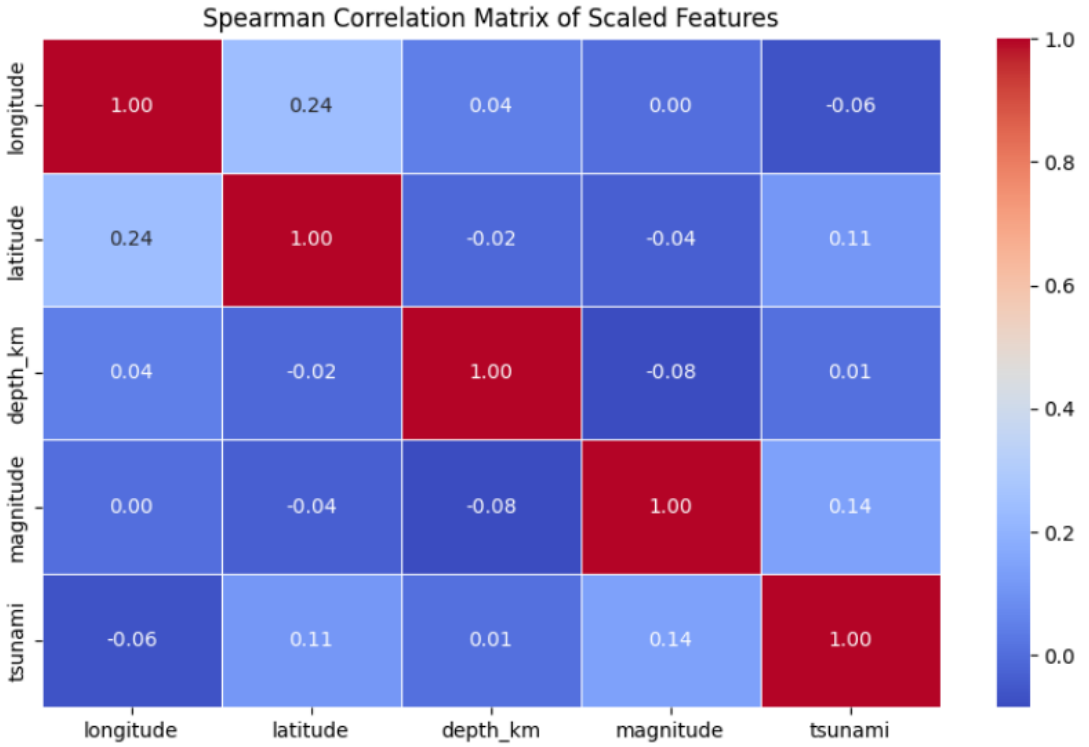



Figure 4.17 Spearman correlation matrix of important features

### 4.6.2 Scatterplot of Magnitude, Depth and Tsunami

Figure 4.18 shows the scatterplot of relationship between earthquake depth, magnitude and tsunamis occurrences that marked as red. The existed of tsunami shown that most tsunamis are generated by shallow earthquakes with magnitudes greater than 6.0 with depth less than 100 km. This trend is consistent with geological laws where shallow depth with high magnitude earthquake create energetic processes in the inshore or ocean floor can transmit adequate energy to move large volumes of water to the coastal area. Moreover, earthquakes that are located deeper than 300km even though with high magnitude rarely cause tsunamis since the seismic energy dies out before it reaches the ocean floor. Overall, the distribution of magnitude and depth of an earthquake are key factors that define tsunami potential as the greatest threat occurs when an earthquake is shallow and of a high magnitude.

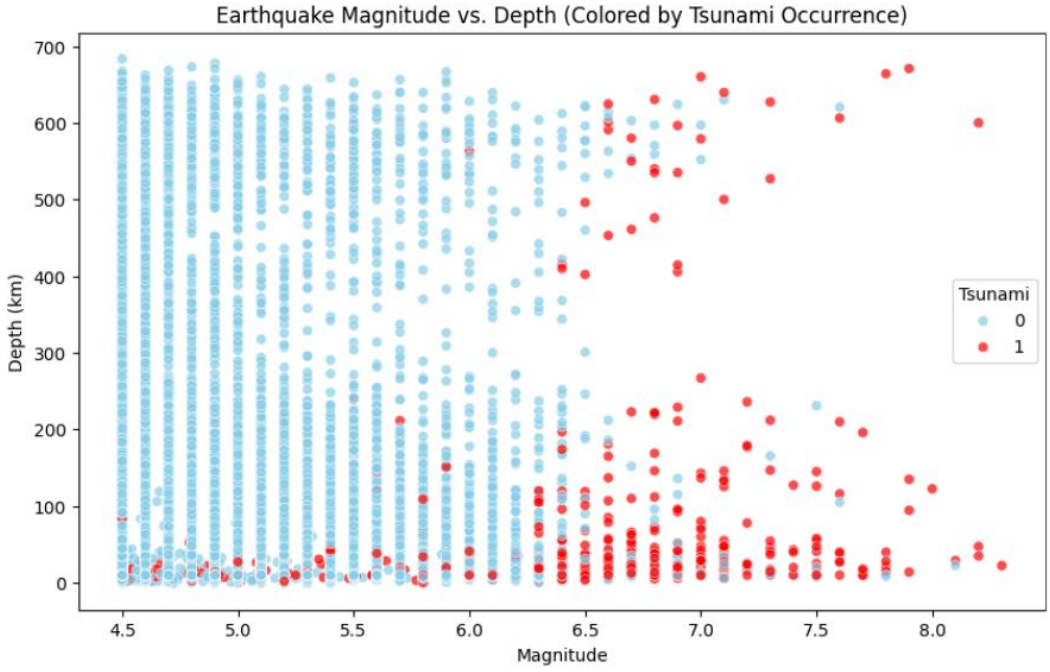



Figure 4.18 Scatterplot of relationship earthquake magnitude vs depth for tsunami occurrence

### 4.6.3 Trend of Earthquake and Tsunami from 2015-2024

The bar chart in Figure 4.19 shows the pattern of earthquake variations from 2015 to 2024. The number of earthquakes experienced in these years within a range of around 6,000 to 8,000 earthquakes. It is remarkable that 2021 had an unusual increase of seismic activity with the largest number exceeding 8,000 earthquakes. 2018, 2023 and 2022 shows relatively high numbers and other years were moderate compared to 2020 and 2024.

In addition, the annual trend of Tsunami in 2015-2024 is shown in Figure 4.20 where there is a significant change in tsunamis during the period 2015-2024. The highest tsunami occurrence was experienced in 2018 with more than 120 tsunamis while 2024 recorded lower occurrences with more than 80 cases. This trend shows that the tsunami events have been decreasing slowly, especially after 2018 with a relatively constant trend then become lower in recent years. This trend illustrates that seismic activity is not increasing continuously, there are spikes in some years but not in overall trend which reflects the randomness of tsunami occurrences.

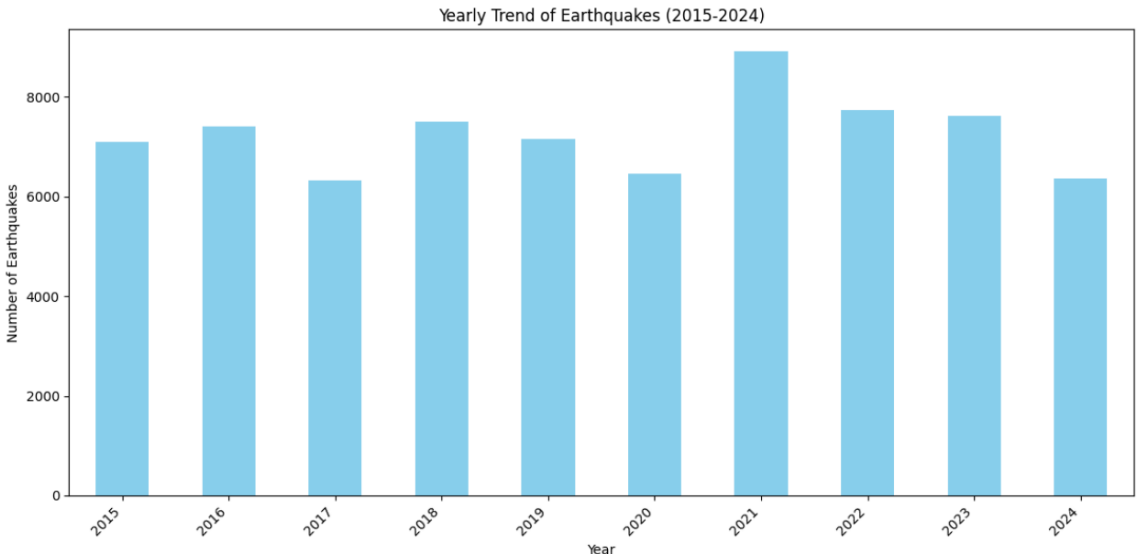



Figure 4.19 Annual trend of earthquake

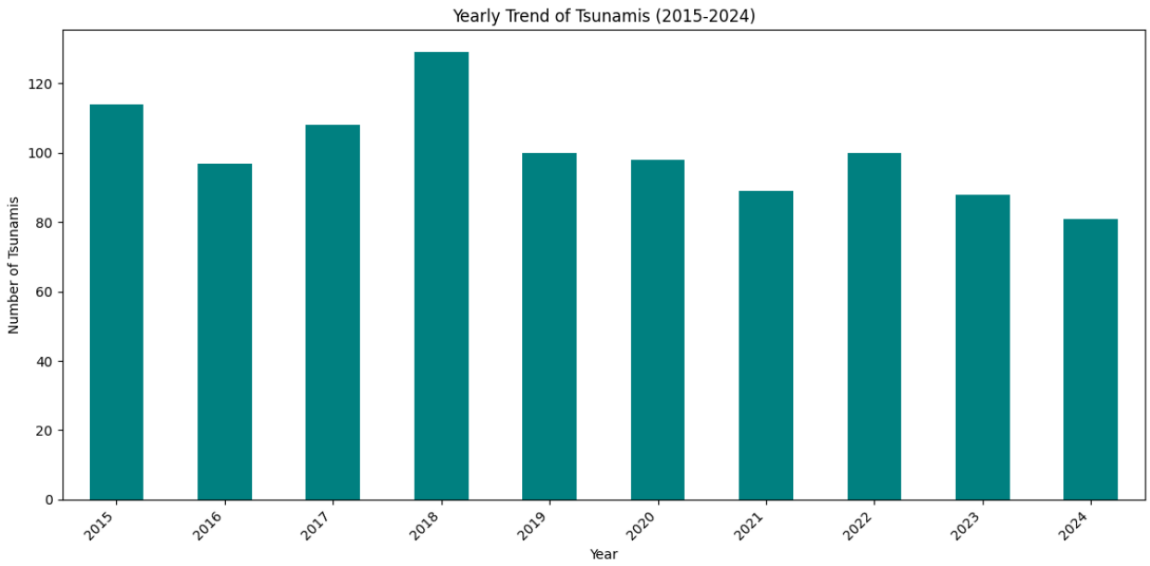



Figure 4.20 Annual trend of tsunami

### 4.6.4 Countries of Most Recorded Earthquake and Tsunami From 2015-2024

Figure 4.21 presents the country with the highest earthquake and record in a decade. Indonesia and Tonga recorded the highest number of earthquakes with more than 7000 events in a decade reflecting this country located on a major tectonic zone of active seismic activities. Japan, Papua New Guinea and Philippines recorded more than 3,000 earthquakes and others recorded less than 3,000 events. This data proved that the areas located at the Ring of Fire have frequent seismic activity.

However, United State shown the most recorded tsunami occurrences with more than 400 cases while the other country like Canada, Russia, Tonga and Papua New Guinea recorded less than 100 cases. This indicates that United States located at the most active subduction zone that highly exposed to tsunami-generating earthquake and underwater landslide. This collected information allows authorities to identify the frequent seismic activity and tsunami occurrences for the mitigation preparation and reduce the impact of earthquakes on the human population and infrastructures.

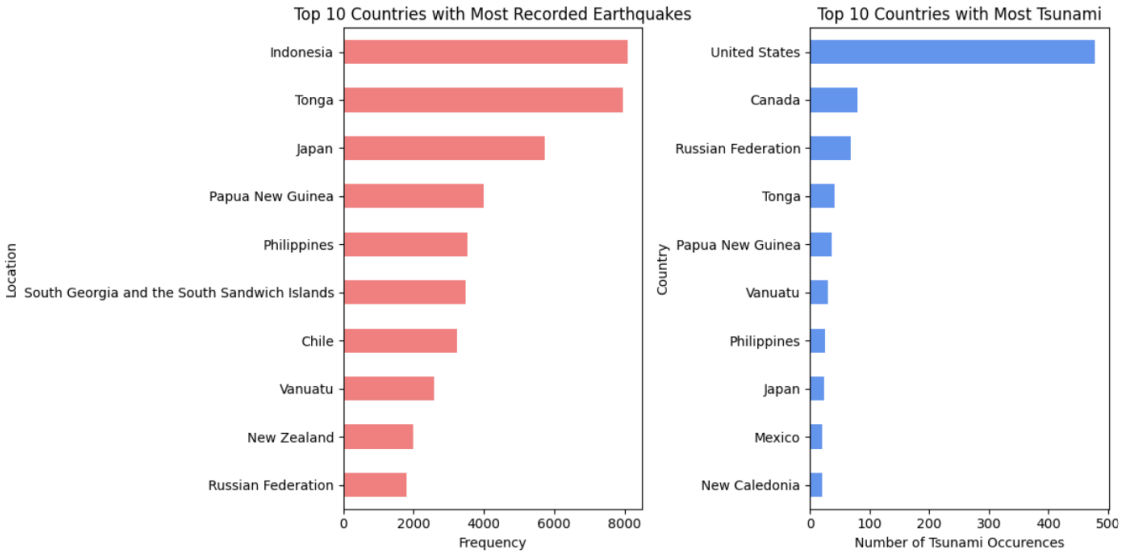



Figure 4.21 Countries with most recorded earthquake and tsunami from 2015-2025

### 4.6.5 Country of Highest Magnitude Earthquakes generated Tsunami

Figure 4.22 demonstrates the countries with the highest magnitude of earthquake generated tsunami in a decade. Chile recorded with the highest magnitude followed by Fiji, Mexico, United State and Tonga region with overall magnitude more than 8.0. These countries proved that highest magnitude generated tsunami located at Ring of Fire with most major subduction zone. Most intense and high energy earthquakes on earth are found in these countries such as 1960 Valdivia quake in Chile (M 9.5) being the most megathrust events. This happened due to tectonic structure characterised by oceanic plates that are subducted under either continental or other oceanic plates, which makes them prone to the occurrence of megathrust earthquakes and the associated tsunamis.

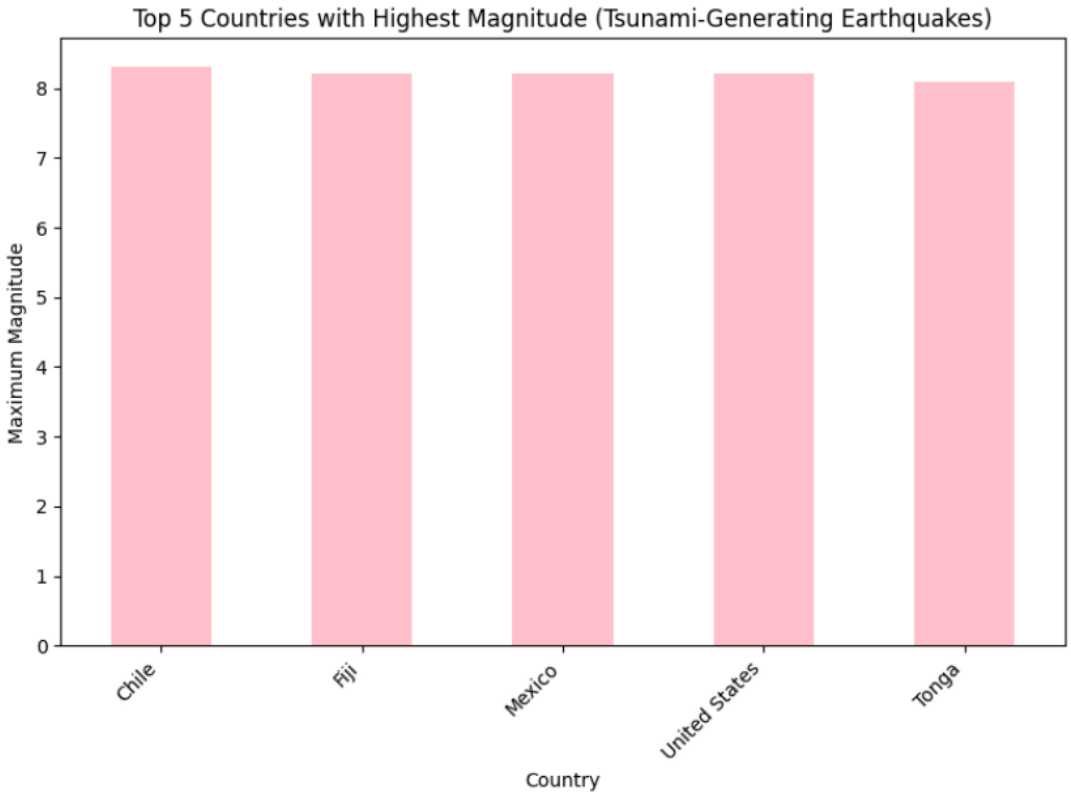



Figure 4.22 Countries with highest earthquake magnitude generated tsunami

### 4.6.6 Tsunami Occurrences

Figure 4.23 illustrated pie chart of the distribution of tsunami occurrence with only 1.4% earthquake generate the tsunami (1004 cases) while 98.6% (71,567) of the recorded earthquake has no tsunami recorded. The small percentage of tsunami due to the high abundance of earthquakes occurring at land or on strike-slip faults with horizontally oriented displacement with a relatively small amount of underground seawater raised. Conversely, the major tsunamis are typically triggered by the strikes of the large-magnitude megathrust earthquakes along subduction zones, which occur when one tectonic plate is forced beneath another, producing sudden vertical motion of the seafloor. Since these conditions are rare as compared to the general occurrence of earthquakes, tsunami events are not common but are very calamitous when realized.

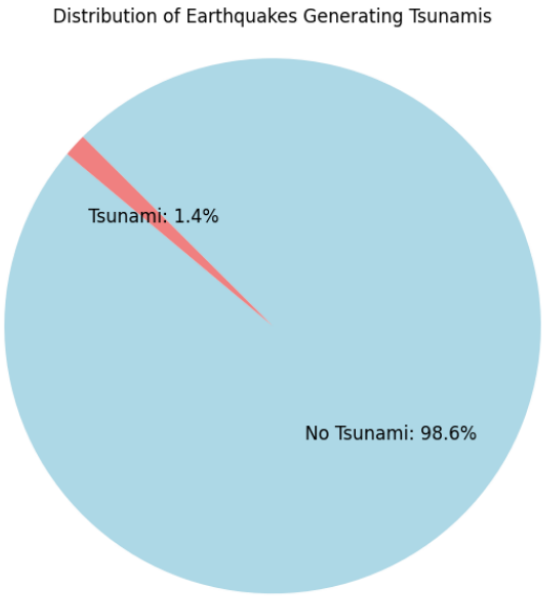



Figure 4.23 Pie chart of distribution of earthquake-generating tsunami

### 4.6.7 Earthquake and Tsunami Distribution Based on Continent

The pie chart in Figure 4.24 demonstrates the distribution of earthquakes across different continents. The largest proportion of 33.8 % of global earthquakes is occupied by Oceania and is followed by Asia (32.8 %). South America has an equally large contribution of 15 %. These indicates the occurrences of high seismic activity in these regions lying in the boundaries of active tectonic plates.

Other continents have relatively lower percentages of the incidences of earthquakes. North America is 7.8 %, Europe 5.2 %, Africa 4.7 %, and the least is Antarctica where it is 0.7 %. These relative values imply that continents like Antarctica and Africa have less seismicity which could be ascribed to their location being far aside along tectonic plate boundaries or major fault lines. Overall, the chart highlights the asymmetrical worldwide seismic activity distribution by indicating that some areas of the world are much more prone to earthquakes as compared to others.

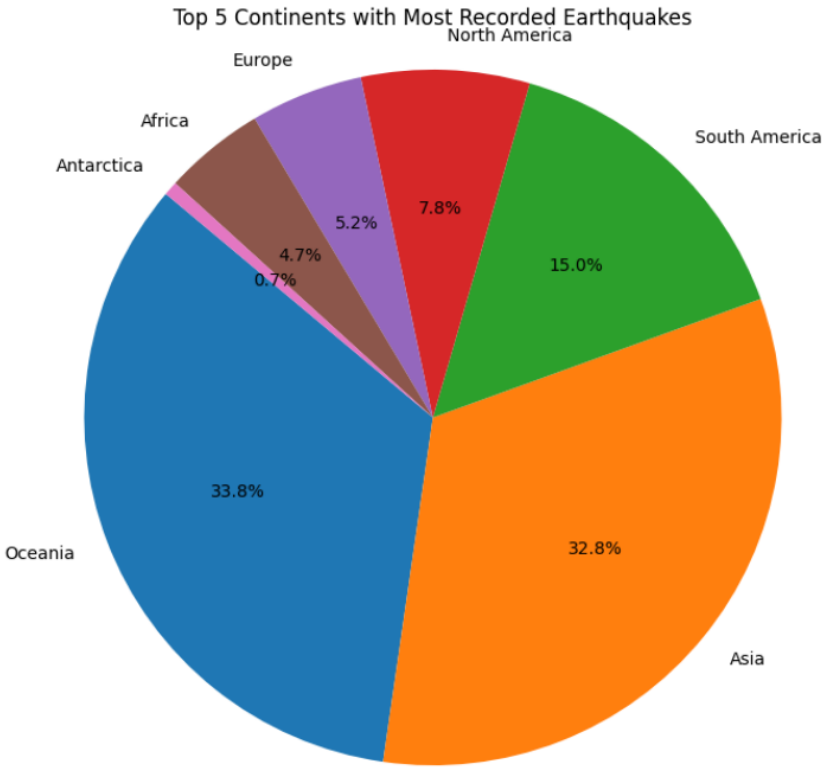



Figure 4.24 Earthquake distribution by continent

## 4.7 Chapter Summary

In conclusion, this chapter discussed about the exploratory data analysis (EDA) steps started from data cleaning and pre-processing until the early findings of the earthquake from 2015 until 2024. Analysis of the earthquake events indicate the most frequent earthquake occur at the active seismic zone of the Ring of Fire like Indonesia, Tonga and Japan. Overall correlation matrix shows the low to moderate relationship between all features. Further analysis will perform in feature selection to identify the earthquake that influences the tsunami occurrences.

# CHAPTER 5

# MODEL DEVELOPMENT AND EVALUATION

## 5.1 Introduction

This chapter describes how the predictive models of earthquakes occurrence and risk assessment are created, trained, and validated based on the discovered patterns during exploratory data analysis (EDA). The steps involve applicable feature selection, handling imbalanced and the use of algorithms including XGBoost and NB before the hyperparameters are tuned and the results are measured using confusion matrix and classification results of accuracy, precision, recall, F1-score and the ROC and AUC for the model performance. After verification, the models are incorporated into an interactive dashboard which enables visualisation, storytelling and efficient communication of results to assist in decision making for earthquake monitoring and mitigation.

## 5.2 Feature Selection

In this feature selection process, a correlation-based feature selection method was applied to identify the most relevant variables for predicting the target variable of tsunami. The seismically relevant features in Table 5.1 were chosen for the training while the features that less important were removed to reduce noise, reduce overfitting and harnessing the dimensionality (Satish et.al 2025; Sukmana et.al, 2024)

Table 5.1 Selected Features

| Features | Detail |
|---|---|
| Latitude | Define the earthquake's location. |
| Longitude | Define the earthquake's location. |
| Magnitude | Measures the earthquake's strength. |
| Depth | Affects the severity of surface shaking. |

Figure 5.1 shows a correlation cut off 0.1 was applied which implies that only features with an absolute correlation beyond this cut off value with any of the variables were retained. The target variable was excluded in the selected features to avoid data leakage. The final selected features including geographical attributes of longitude and latitude with seismic attributes of depth and magnitude.

```
threshold = 0.1
selected_features = correlation_matrix[abs(correlation_matrix) >= threshold].index.tolist()

# Remove the target from the list
selected_features.remove('tsunami')

print("Selected features based on correlation threshold:")
print(selected_features)
```

```
Selected features based on correlation threshold:
['longitude', 'latitude', 'depth_km', 'magnitude']
```

Figure 5.1 Feature selection

## 5.3 Model Development and Evaluation

### 5.3.1 Split and Train

Figure 5.2 demonstrate the dataset was prepared and split into training and testing sets. The selected features are transformed into the feature matrix `X` and the target vector `y` is obtained in the form of a binary number of tsunami values. `train_test_split` function imported form Scikit-learn library of `sklearn.model_selection` and applied to split the 80% of the data into 58,056 samples model train and 20% testing of 14,515 samples of the feature X and target y to prevent unbiased and show real performance of the model. The parameter `test_size =0.2` and random sampling technique using `random _state=42` prompt used to ensure all the random selected data split equally with the same possibility of data be chosen.

```
X = analysis[selected_features]
y = analysis['tsunami']
```

```
from sklearn.model_selection 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: ', X_train.shape)
print('y-train: ', y_train.shape)
print()
print('X-test: ', X_test.shape)
print('y-test: ', y_test.shape)
```

```
X-train:  (58056, 4)
y-train:  (58056,)

X-test:  (14515, 4)
y-test:  (14515,)
```

Figure 5.2 Splitting the samples of dataset

Then, Table 5.2 shows the results of training and testing datasets are left with 4 features of class each so that the model can learn the patterns with the training set and verified on the testing set. 80% or 58,056 of training data consists of 57,256 class "No Tsunami" and 800 of class "Tsunami" while another 20% or 14,515 of testing data consist of 14,311 class "No Tsunami" and 204 class of "Tsunami".

Table 5.2 Train-Test Original Dataset

| Class | Training (80%) | Testing (20%) |
|---|---|---|
| Class 0 = No tsunami | 57,256 | 14,311 |
| Class 1 = Tsunami | 800 | 204 |

### 5.3.2 Handling Imbalanced

The result of splitting clearly shows the imbalance of tsunami events which are only 1.4% compared to non-tsunami events. The earthquake-generating tsunami only recorded 1004 cases compared to earthquakes that do not generate tsunami of 71,567 cases. This high imbalance data will give the challenge to machine learning to learn and tend to create biased to minority class of tsunami resulting low recall and poor sensitivity in detecting tsunami events. Hence, two complimentary methods applied in this study to overcome it including Synthetic Minority Oversampling Technique (SMOTE) and class weighting method. However, both techniques only applied during training model to

maintain the validation and testing set with original unbalanced class distribution to ensures the model evaluation reflects real-world scenarios where tsunami events are rare.

SMOTE is applied in the preprocessing phase, specifically on the training set. Unlike random oversampling methods that simply duplicate minority class samples, SMOTE generates new synthetic samples by interpolating between existing tsunami examples and their nearest neighbours. This approach expands the representation of tsunami classes in feature space thus provides more meaningful model of decision boundaries and reduces the risk of overfitting. Table 5.3 indicates the number of tsunamis in training model before and after SMOTE sampling. This technique creates artificial tsunami data that is similar to the original data, thus the number of tsunami samples increases from 800 to 57,256 and is balanced with the non-tsunami class. Balancing the dataset with oversampling on this minority class can improve the model's performance in detecting tsunamis without discarding important information from both events and avoiding bias against the majority class.

Table 5.3 Training set before and after SMOTE sampling

| Class | Training Before | Training After |
|---|---|---|
| Class 0 = No tsunami | 57,256 | 57,256 |
| Class 1 = Tsunami | 800 | 57,256 |

In parallel, positive class weighting also used in the XGBoost model by the scale_pos_weight parameter. This method does not directly modify the dataset instead of adjusts the learning algorithm to provide a higher penalty for misclassification of the tsunami class. The weight value is determined in equation (5.1) using the ratio between the number of negative samples (No Tsunami) and the number of positive samples (Tsunami) in the training data indicate that approximately 72 that indicate the tsunami is 72 times rare than non-tsunami events. Each recorded tsunami given a greater influence during the training process, allowing the model to focus more on detecting tsunami events without having to add synthetic data to the original set.

$$scale\ pos\ weight = \frac{57{,}256}{800} = 71.57 \quad (5.1)$$

### 5.3.3 Model Implementation

Figure 5.3 and Figure 5.4 involves training and testing two machine learning models of XGBoost and NB for baseline, SMOTE and class weight method for prediction of tsunami-generating earthquake. First, the `XGBClassifier` library imported from Scikit-learn. The XGBoost Classifier is set with `binary:logistic` and a fixed `random_state=42` to ensure repeatability and then trained on the training data (`X_train, y_train`) and (`X_smote, y_smote`) for both SMOTE and class weight. Then, (`X_test`) applied to make classifications for the test dataset. Similarly, the `GaussianNB` imported from Sckit-learn library to train and test the splitting dataset and it was applied with equal number of estimators and random state to predict outcomes for the test set. Figure 5.5 shows the XGBoost class weight method used parameter of `scale_pos_weight` to handle minority class of tsunami.

```
import xgboost as xgb
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate an XGBClassifier model
xgb_model = xgb.XGBClassifier(objective='binary:logistic',
                              use_label_encoder=False,
                              eval_metric='logloss',
                              random_state=42)

# Train the XGBoost model on the original training data
xgb_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb = xgb_model.predict(X_test)
y_pred_proba_xgb = xgb_model.predict_proba(X_test)[:, 1]


from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate and train the Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# Make predictions
y_pred_nb = nb_model.predict(X_test)
y_pred_proba_nb = nb_model.predict_proba(X_test)[:, 1]
```

Figure 5.3 Model of XGBoost and NB baseline algorithms

```
# Instantiate an XGBClassifier model
xgb_smote_model = xgb.XGBClassifier(objective='binary:logistic',
                                    use_label_encoder=False,
                                    eval_metric='logloss',
                                    random_state=42)

# Train the XGBoost model using the SMOTE-augmented training data
xgb_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_xgb_smote = xgb_smote_model.predict(X_test)
y_pred_proba_xgb_smote = xgb_smote_model.predict_proba(X_test)[:, 1]
```

```
# Instantiate a GaussianNB model
nb_smote_model = GaussianNB()

# Train the Naive Bayes model using the SMOTE-augmented training data
nb_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_nb_smote = nb_smote_model.predict(X_test)
y_pred_proba_nb_smote = nb_smote_model.predict_proba(X_test)[:, 1]
```

Figure 5.4 XGBoost and NB algorithms using SMOTE

```
# Instantiate and train the XGBoost model with scale_pos_weight
scale_pos_weight = (y_train == 0).sum() / (y_train == 1).sum()

# Instantiate an XGBClassifier model with scale_pos_weight
xgb_scaled_model = xgb.XGBClassifier(objective='binary:logistic',
                                     use_label_encoder=False,
                                     eval_metric='logloss',
                                     random_state=42,
                                     scale_pos_weight=scale_pos_weight
)

# Train the XGBoost model on the original training data
xgb_scaled_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb_scaled = xgb_scaled_model.predict(X_test)
y_pred_proba_xgb_scaled = xgb_scaled_model.predict_proba(X_test)[:, 1]
```

Figure 5.5 XGBoost algorithm using class weight

The evaluation result of testing models shown in Table 5.4. The results of the analysis demonstrate the imbalance control data have different effects on the performance of the XGBoost and Naïve Bayes models. Both SMOTE and positive class weighting was applied in XGboost model. The analysis results of both methods successfully improve the model's ability to detect tsunamis compared to XGBoost baseline model with recall values of 0.90686 for SMOTE and 0.88235 for class weight respectively. Although XGBoost with SMOTE provides a slight advantage in terms of recall, it requires a large increase in synthetic data and can increase the computational cost.

In addition, XGBoost with class weight provides almost equivalent performance without changing the dataset size by adjusting the model's loss function that giving a higher penalty to misclassified tsunami cases. This combination give stability and maintains balance between recall, precision, and F1-score thus making it more practical to apply to large-scale datasets.

Besides, Naïve Bayes model using the original data without any balancing method only resulted 0.51961 of recall, which means that almost half of the tsunami events could not be detected. When SMOTE was applied, the recall value increased significantly to 0.94118, hence allowing the model to detect almost all tsunami events. However, the accuracy decreased drastically indicating that there were many false alarms detected.

Yet, recall is much more important in the context of disaster early warning system because the risk of failure to detect a tsunami (false negative) can result in significant loss of life and damage. Nevertheless, the existence of false positives or false alarms can still be handled through further validation mechanisms. Therefore, the selection of Naïve Bayes with SMOTE is considered more suitable for this study than using only the original data.

Table 5.4 Evaluation results models

| Models | Accuracy | Precision | Recall | F1-Score | AUC | Confusion Matrix |
|---|---|---|---|---|---|---|
| XGBoost baseline algorithm | 0.99208 | 0.70892 | 0.74020 | 0.72422 | 0.99500 | $\begin{bmatrix} 14249 & 62 \\ 53 & 151 \end{bmatrix}$ |
| Naïve Bayes baseline algorithm | 0.98188 | 0.39114 | 0.51961 | 0.44632 | 0.97522 | $\begin{bmatrix} 14146 & 165 \\ 98 & 106 \end{bmatrix}$ |
| XGBoost + SMOTE | 0.98932 | 0.57632 | 0.90686 | 0.70476 | 0.99560 | $\begin{bmatrix} 14175 & 136 \\ 19 & 185 \end{bmatrix}$ |
| XGBoost + Class Weight | 0.98918 | 0.57508 | 0.88235 | 0.69632 | 0.99575 | $\begin{bmatrix} 14178 & 133 \\ 24 & 180 \end{bmatrix}$ |
| Naïve Bayes + SMOTE | 0.90947 | 0.12851 | 0.94118 | 0.22615 | 0.97494 | $\begin{bmatrix} 13009 & 1302 \\ 12 & 192 \end{bmatrix}$ |

Overall, the selection of the best imbalance handling method for each model is different. The method of class weight is more suitable for XGBoost model because it can maintain performance stability without burdening the computational process. While SMOTE is a more suitable choice for Naïve Bayes model by increase the representation of very small minority classes. The combination of these two approaches allows this study to achieve the main objective of improving the model's ability to predict tsunami events by prioritizing recall while maintaining reasonable prediction accuracy in real-world scenarios.

### 5.3.4 Model Validation and Optimization (Hyperparameter Tuning)

Figure 5.6 and Figure 5.7 illustrated both model of XGboost and NB classifier tuned with the objective of maximizing the recall metric that is very crucial for imbalanced datasets in detecting the minority class using `GridSearchCV`**.** XGboost tuned using parameters of `n_estimators` (number of trees), `max_depth` (tree depth), `learning_rate` (step size), `subsample` (sample fraction per tree), and `colsample_bytree` (feature fraction per tree). The grid allows a thorough search of the most possible parameters settings by testing several values of each parameter. In the process, XGBoost classifier is initialised with parameters such as `use_label_encoder=False` to suppress warnings, `eval_metric='logloss'` to optimise the model and `scale_pos_weight` to handling imbalanced. The scoring criterion in the search is recall that aim to minimize the false negatives and maximize the detection of positive cases. The optimal configuration of hyperparameters for these models are as per Table 5.5 below.

Table 5.5 Best hyperparameter for XGBoost and NB models

| Parameter XGBoost | Parameter Naïve Bayes |
|---|---|
| colsample_bytree = 0.8<br>learning_rate = 0.1<br>'max_depth': 3<br>n_estimators = 100<br>subsample = 1.0 | var_smoothing = np.float64(1e-05) |

Similarly, NB tuned with parameter `var_smoothing` using a log-spaced range. Then, the NB classifier initialized using this parameter. To maintain credible estimation, 10-fold stratified cross-validation applied in both data to maintains the ratio of classes in each fold. `GridSearchCV` will then fit and score the model with all combinations of parameters and recall is the scoring metric, to prioritise the identification of true positives. Once searching is completed, the optimal model saved in `grid_search.best_estimator_` and this is used to predict the data in the test set. Then, last predictions of `y_pred_tuned_xgb_scaled` and `y_pred_tuned_nb_smote` is the performance of XGBoost and NB model that are tuned to achieve the high levels of recall.

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold

# Define the parameter grid for XGBoost
param_grid = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1],
    'max_depth': [3, 5],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}

# Instantiate StratifiedKFold
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Instantiate GridSearchCV
grid_search_xgb_scaled = GridSearchCV(estimator=xgb.XGBClassifier(objective='binary:logistic',
                                                                  use_label_encoder=False,
                                                                  eval_metric='logloss',
                                                                  random_state=42,
                                                                  scale_pos_weight=scale_pos_weight),
                                      param_grid=param_grid,
                                      cv=skf,
                                      scoring='recall',
                                      verbose=1,
                                      n_jobs=-1)

# Fit GridSearchCV to the original training data
grid_search_xgb_scaled.fit(X_train, y_train)

# Make predictions on the original test set
y_pred_tuned_xgb_scaled = xgb_tuned_scaled_model.predict(X_test)
y_pred_proba_tuned_xgb_scaled = xgb_tuned_scaled_model.predict_proba(X_test)[:, 1]
```

Figure 5.6 XGBoost model validation and optimization

```
# Define the parameter grid for Naive Bayes
param_grid_nb = {
    'var_smoothing': np.logspace(0, -9, 10)
}

# Instantiate a GaussianNB model
nb_tuned_smote_model = GaussianNB()

# Instantiate StratifiedKFold
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Instantiate GridSearchCV
grid_search_nb_smote = GridSearchCV(estimator=nb_tuned_smote_model,
                                    param_grid=param_grid_nb, cv=skf, scoring='recall',
                                    verbose=1, n_jobs=-1)

# Fit GridSearchCV to the SMOTE-augmented training data
grid_search_nb_smote.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_tuned_nb_smote = nb_tuned_smote_model.predict(X_test)
y_pred_proba_tuned_nb_smote = nb_tuned_smote_model.predict_proba(X_test)[:, 1]
```

Figure 5.7 NB model validation and optimization

### 5.3.5 Model Evaluation

The application of the evaluation of confusion matrix process shown in Figure 5.8. The tuned XGBoost `confusion_matrix` calculated using (`y_test, y_pred_tuned_xgb_scale`). Meanwhile, NB were calculated via `confusion_matrix` function (`y_test, y_pred_tuned_smote`).

Figure 5.9 describes the performance evaluation models including the accuracy of training and testing using the Scikit-learn `train_aacuracy,test_accuracy` and `classification_report` including precision which measures the proportion of true tsunami predictions, recall that measures the proportion of actual tsunamis that were detected correctly, and f1-score to balances both the procedure calculated using Scikit-learn functions on both the original and tuned models.

In addition, performance was reported as ROC and AUC score using `(roc_auc_score(y_test, model.predict_proba(X_test)[:, 1])` is to assess the capacity of the models to differentiate between tsunami and non-tsunami events in a probability ranking order. Collectively, these lines of codes offered a systematic structure on which to compare their performance to determine the predictive ability of the XGBoost and NB algorithm for hyperparameter optimization.

```
# Calculate the confusion matrix
cm_tuned_xgb_scaled = confusion_matrix(y_test, y_pred_tuned_xgb_scaled)
cm_tuned_nb_smote = confusion_matrix(y_test, y_pred_tuned_nb_smote)
```

Figure 5.8 Confusion matrix

```python
# Collect metrics for each model
all_metrics_list = []

# Function to get accuracy for training and testing
def get_accuracy(model, X_train, y_train, X_test, y_test):
    train_accuracy = model.score(X_train, y_train)
    test_accuracy = model.score(X_test, y_test)
    return train_accuracy, test_accuracy


# Tuned XGBoost on Original Data with scale_pos_weight
train_acc_xgb_scaled_tuned, test_acc_xgb_scaled_tuned = get_accuracy(xgb_tuned_scaled_model,
                                                                     X_train, y_train,
                                                                     X_test, y_test)
report_xgb_scaled_tuned = classification_report(y_test, y_pred_tuned_xgb_scaled, output_dict=True)
all_metrics_list.append({
    'Model': 'XGBoost',
    'Dataset/Method': 'Scale Pos Weight (Tuned)',
    'Accuracy (Training)': round(train_acc_xgb_scaled_tuned, 5),
    'Accuracy (Testing)': round(test_acc_xgb_scaled_tuned, 5),
    'Precision (Class 0)': round(report_xgb_scaled_tuned['0']['precision'], 5),
    'Recall (Class 0)': round(report_xgb_scaled_tuned['0']['recall'], 5),
    'F1-Score (Class 0)': round(report_xgb_scaled_tuned['0']['f1-score'], 5),
    'Precision (Class 1)': round(report_xgb_scaled_tuned['1']['precision'], 5),
    'Recall (Class 1)': round(report_xgb_scaled_tuned['1']['recall'], 5),
    'F1-Score (Class 1)': round(report_xgb_scaled_tuned['1']['f1-score'], 5),
    'AUC': round(roc_auc_score(y_test, y_pred_proba_tuned_xgb_scaled), 5)
})
# Tuned Naive Bayes on SMOTE Data
train_acc_nb_smote_tuned, test_acc_nb_smote_tuned = get_accuracy(nb_tuned_smote_model,
                                                                 X_smote, y_smote, X_test, y_test)

report_nb_smote_tuned = classification_report(y_test, y_pred_tuned_nb_smote, output_dict=True)
all_metrics_list.append({
    'Model': 'Naive Bayes',
    'Dataset/Method': 'SMOTE (Tuned)',
    'Accuracy (Training)': round(train_acc_nb_smote_tuned, 5),
    'Accuracy (Testing)': round(test_acc_nb_smote_tuned, 5),
    'Precision (Class 0)': round(report_nb_smote_tuned['0']['precision'], 5),
    'Recall (Class 0)': round(report_nb_smote_tuned['0']['recall'], 5),
    'F1-Score (Class 0)': round(report_nb_smote_tuned['0']['f1-score'], 5),
    'Precision (Class 1)': round(report_nb_smote_tuned['1']['precision'], 5),
    'Recall (Class 1)': round(report_nb_smote_tuned['1']['recall'], 5),
    'F1-Score (Class 1)': round(report_nb_smote_tuned['1']['f1-score'], 5),
    'AUC': round(roc_auc_score(y_test, y_pred_proba_tuned_nb_smote), 5)
})

# Create the DataFrame
all_performance_df = pd.DataFrame(all_metrics_list)

# Display the DataFrame
display(all_performance_df)
```

Figure 5.9 Model evaluation

## 5.4 Results and Discussion

### 5.4.1 Confusion Matrix

Figure 5.10 shows the final confusion matrix result of the XGBoost classifier model in tsunami prediction. XGBoost achieved highest true positive result that can correctly identify 201 tsunami events from 204 cases with only 3 misclassified tsunami events (false negatives). This indicate almost all of tsunamis can be detected in real world scenario. Although there were 334 false alarms, this number is small compared to the 13,977 of "No Tsunami" cases that were correctly classified. This kind of a trade-off can indicate the deliberate trade-off made during hyperparameter tuning, where a preference to reduce the number of missed tsunami incidents is made at the cost of an increased number of false alarms. This result proved that XGBoost can maintain very high recall and a very low false negative rate thus increasing its reliability.

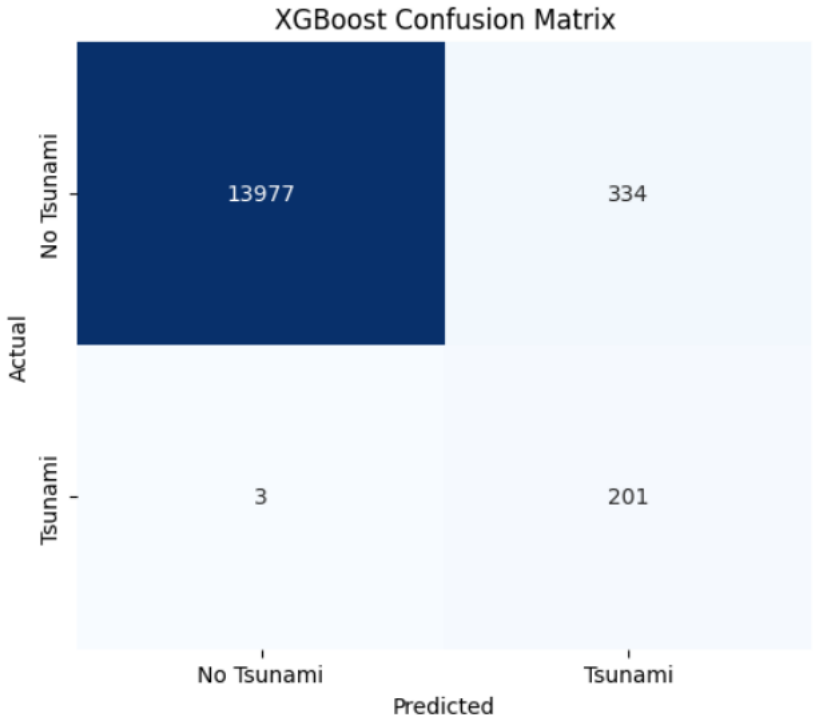



Figure 5.10 XGBoost confusion matrix

In comparison, Naïve Bayes confusion matrix results in Figure 5.11 shows that this model also capture high true positive of tsunami events with 194 from 204 events and only 10 misclassify tsunami event (false negative). Also, this model indicates good tsunami detection ability but less stable compared to XGBoost due to its higher false positive rate of non-tsunami events that 1,067 cases incorrectly classified as tsunamis. Even though the recall value of this model is still high, the large number of false alarms can reduce the practical effectiveness of the system if used for early warning. This contrast shows that XGBoost is more suitable because its performance is more balanced with high recall and lower false alarm rate than Naïve Bayes.

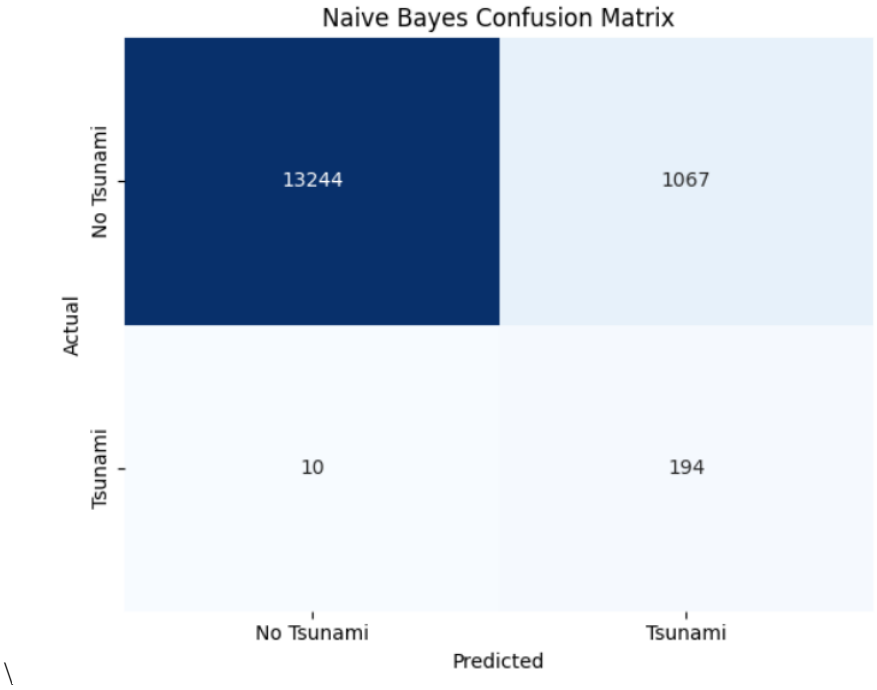



Figure 5.11 Naïve Bayes confusion matrix

Overall, the findings indicate that XGBoost is more vigorous and recall-based, which makes it a good disaster early warning system where failure to identify a genuine tsunami might lead to catastrophic effects. XGBoost also is more aligned to the precautionary principle of tsunami prediction, whereby the necessity to balance detection of rare but important events prevails over the inconvenience of more false positives (Saito and Rehmsmeier, 2015).

### 5.4.2 Classification Performance Metrics

Based on Table 5.6, the model performance analysis shows that XGBoost and Naïve Bayes (NB) have significant differences in their ability to classify tsunami events. XGBoost recorded an accuracy of 0.97833 on the training set and 0.97678 on the test set. The very small difference between these two values proves that XGBoost achieved a good fitting condition means that it not only able to learn patterns from the training data but also maintains an optimal balance between bias and variance with consistent performance on new data and thus avoiding overfitting or underfitting issues.

In contrast, NB recorded an accuracy of 0.94223 for training and 0.92580 for testing. Although these values exceed 90% and the gap is still small, but the overall performance is low indicates that this model tends to underfit and fail to capture complex patterns in the data due to its too simple nature. This indicates that NB is not powerful enough to describe the deep relationships between the imbalance features of the tsunami data.

The precision of both model, XGBoost (0.99979) and NB (0.99925) perform almost perfectly in detecting the “No Tsunami” class. However, the “Tsunami class” of NB's precision performance drops sharply until 0.15385, much lower than XGBoost precision of 0.37570. This low precision value indicates a high rate of false positives, where the model often predicts the occurrence of a tsunami when the actual event does not occur. This situation can have major implications in real-world applications because false predictions may cause panic or overreaction. In this context, although XGBoost is still imperfect, it is more effective in reducing the risk of false positives than NB.

Meanwhile, the ability to identify real tsunamis measured using recall. Recall of both models provide good performance with high values where XGBoost is 0.98529 and NB is 0.95098. This shows that both models are more likely to detect real tsunami events by reducing false negatives. The practical implications are more important because failure to detect real tsunamis can cause loss of life and major damage. In this case, high recall is essential but needs to be balanced with good precision to avoid false warnings. This trade-off reflects a purposely readjusted balance towards maximizing the detection of infrequent yet critical tsunami events, which the literature recognizes as the essential

strategy in making high-stakes decisions, in which false negatives should be minimized (Saito and Rehmsmeier, 2015; Wang et al., 2020).

F1-score is a measure of the balance between precision and recall. XGBoost is more stable and effective with a tsunami score of 0.54398 compared to NB which only recorded 0.26485. This large difference indicates that XGBoost is more successful in balancing the ability to identify real tsunamis (recall) with the accuracy of predictions made (precision). This result also confirms that XGBoost can reduce bias and adapt well to the data without being too complex to cause overfitting. This results is in line with the simplifying assumptions of Naive Bayes that often tend to generate biased estimates of probabilities in complex imbalanced datasets (Rish, 2001; Zhang, 2011).

In conclusion, this comparison proves that XGBoost is more suitable for use in tsunami early warning systems because it provides high accuracy, excellent recall, and a more robust performance balance through F1-score. The ability to detect almost all tsunami events makes it ethically and operationally favorable in high-stakes situations where false detections can cause disastrous effects. This strategy aligns with established research of the theory of imbalanced learning, which prioritizes recall and minimise false negatives over precision in dealing with life-critical implication (Chen & Guestrin, 2016; Fawcett, 2006). Although NB offers advantages in terms of simplicity and computational speed, it is not effective enough to be applied in critical situations such as tsunami detection that require accurate and reliable predictions.

Table 5.6 Results of performance metrics

| Model / Results | Accuracy | | Precision | | Recall | | F1- Score | |
|---|---|---|---|---|---|---|---|---|
| | Training | Testing | No Tsunami | Tsunami | No Tsunami | Tsunami | No Tsunami | Tsunami |
| **XGBoost** | 0.97833 | 0.97678 | 0.99979 | 0.37570 | 0.97666 | 0.98529 | 0.98809 | 0.54398 |
| **Naïve Bayes (NB)** | 0.94223 | 0.92580 | 0.99925 | 0.15385 | 0.92544 | 0.95098 | 0.96093 | 0.26485 |

### 5.4.3 ROC Curve and AUC

Figure 5.8 compares Naive Bayes and XGBoost algorithms that both demonstrate better discriminative ability. Naïve Bayes resulted in an area under the curve (AUC) of 0.9850 and a TPR of values close to unity while XGBoost slightly outperformed the Naïve Bayes with an AUC of 0.9963 and a similar approach to TPR. These results suggest that both models are robustly able to differentiate between tsunami and non-tsunami events in nearly all thresholds with XGBoost showing a slight benefit.

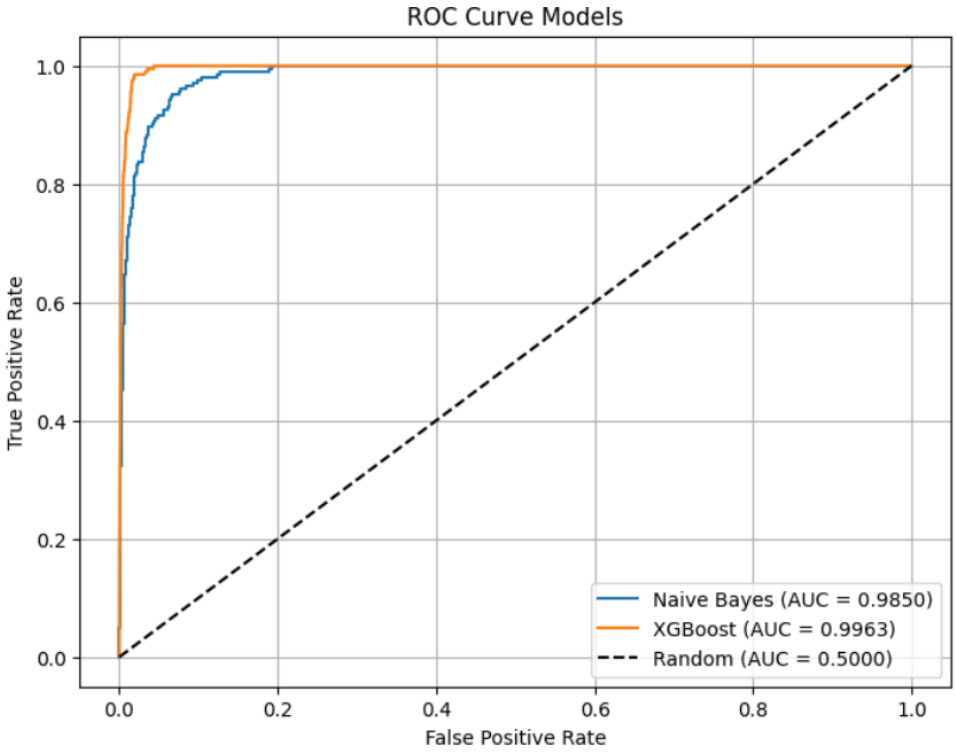



Figure 5.12 Roc and AUC curves of models

The ROC curves of the two classifiers fall on the upper-left quadrant, indicating that they have high recalls with a relatively low false positive rate. This coincidence supports the literature that more sophisticated models especially XGBoost models, perform better on skewed data since they can capture non-linear relationships (Chen and Guestrin, 2016). Even though XGBoost shows a slightly higher AUC, it is not an operationally significant difference, which means that both models have a high discriminatory ability.

Thus, the selection of these models must not only be based on AUC values but must be concentrate by the application-specific priorities like the trade-off between accuracy and recall. False negatives reduction is the highest priority in disaster prediction tasks like tsunami detection which implies that XGBoost might be more reliable than Naive Bayes is a more reliable for lightweight classifier.

### 5.4.4 Comparative Models Performance

This study was compared with previous study in Table 5.7. Sukmana et al. (2024) used Support Vector Machine (SVM) and a Random Forest (RF) classifier to work with earthquake data between 2001-2023 while Satish et al. (2025) applied RF and Logistic Regression (LR) on a longer earthquake dataset from 1995 – 2023 which only focusing on accuracy and precision only.

Based on this comparison of model performance, XGBoost is clearly the most dominant model with the best results in almost all metrics. This model achieved the highest accuracy of 0.97678, outperforming NB of 0.92580, RF of 0.6115 (Sukmana et al. 2024) and 0.90 (Satish et al. 2025) and SVM of 0.6561. Precision of XGBoost is 0.37570 better than NB of 0.15385, but it still lower than SVM (0.7059), Satish et. al 2025 of RF (0.88) and LR (0.87) which tends to be biased because of its high precision but very low recall. Overall, these results show that XGBoost able to reduce the rate of false positives with more balanced performance.

The recall of XGBoost achieved 0.98529, almost perfect in detecting real tsunami events and far exceeding RF of Sukmana et al. 2024 of 0.3607, SVM (0.1967), and NB (0.95098). This capability is very important in early warning systems because high recall reduces the risk of detection failures (false negatives). In addition, the F1-score of XGBoost (0.54398) is more stable than NB (0.26485), RF of Sukmana et al. 2024 (0.4190;), and SVM (0.3077). The almost perfect AUC value of 0.99632 also confirms the advantage of XGBoost indicating excellent class separation capability in unbalanced data.

Compared to previous studies, the performance of Random Forest in Sukmana et al. (2024) and Satish et al. (2025) is still much lower than XGBoost. Although Satish et al. reported an accuracy of 0.90 and a precision of 0.88 for RF, these values do not match the advantages of XGBoost in terms of recall, F1-score, and AUC. Meanwhile, the SVM reported by Sukmana et al. (2024) has a relatively high precision but fails to maintain recall making it unstable and less suitable for tsunami early warning.

Overall, this contrast proves that XGBoost the best model that achieves a good fitting level and is most suitable for application in tsunami prediction systems because it maintains a critical balance of performance to reduce the risk of false positives and false negatives. However, the choice of a model depends on the purpose of operations like to build an early warning system, high-recall models like XGBoost are the best option whereas RF, SVM or LR can be selected in case precision and efficiency are the top priority and to make comparison of the performances.

Table 5.7 Comparative Testing Model Performances

| Models | Accuracy | Precision | Recall | F1-score | AUC | Key Insights |
|---|---|---|---|---|---|---|
| **XGBoost** | 0.97678 | 0.37570 | 0.98529 | 0.54398 | 0.99632 | XGBoost focuses on recall that correctly classify almost all tsunami events at the cost of a high false alarm rate. |
| **Naïve Bayes (NB)** | 0.92580 | 0.15385 | 0.95098 | 0.26485 | 0.98503 | NB model has very low precision but high recall that indicate it identify a significant number of tsunami events correctly but also produced very high false alarm rate compared to XGBoost. |
| **Sukmana et al. (2024)** | | | | | | |
| **Random Forest (RF)** | 0.6115 | 0.5000 | 0.3607 | 0.4190 | 0.6384 | RF has better recall than SVM but with lower precision hence moderate overall performance. |
| **Support Vector Machine (SVM)** | 0.6561 | 0.7059 | 0.1967 | 0.3077 | 0.6215 | SVM model can prevent false positives effectively, but this model is weak in detecting actual tsunami events. |
| **Satish et al. (2025)** | | | | | | |
| **Random Forest (RF)** | 0.90 | 0.88 | - | - | - | RF has a strong balance and resilient in handling imbalanced data. |
| **Logistic Regression** | 0.89 | 0.87 | - | - | - | Although this model slightly less efficient than RF, however it extremely interpretable. |

### 5.4.5 Analysis of Feature Importance

The findings of the feature importance shown in Figure 5.13 indicate that magnitude is the most significant variable in forecasting the occurrence of tsunamis with an importance score of 0.4431 which indicates its significant contribution in tsunami prediction. This phenomenon is in line with geophysical evidence, where larger earthquakes typically cause more displacement of the seafloor which is the major process behind tsunami production. Precisely, earthquakes that are stronger than 7.0 Mw have the greatest potential of causing tsunami waves (Cilia et al., 2021).

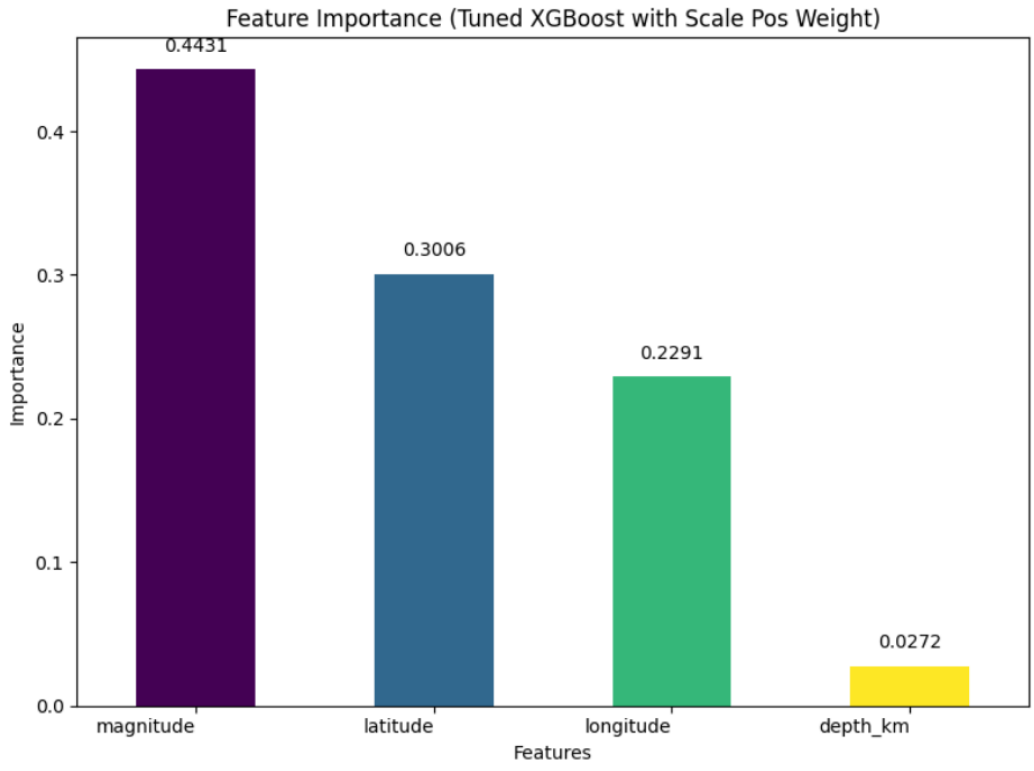



Figure 5.13 Features importance in predicting tsunami-generating earthquake

Earthquake latitude has the second strongest influence with a score of 0.3006. Therefore, the spatial distribution of earthquakes especially in tectonically active regions including the Ring of Fire and subduction zones has a strong impact on the likelihood of destructive, tsunami-generating earthquake events (Lay, 2016; F. Y. Wu et al., 2019).

However, the longitude shows a rather low value of the importance of 0.2291, which means that it has a contributory significance on the prediction of tsunamis, but not as significant as latitude and magnitude. Finally, depth of earthquake is the least important with the importance score of 0.0272. Although shallow earthquakes have more chances to cause tsunamis because they produce more powerful shaking of the seafloor, depth is not enough but must be accompanied by magnitude and geographic location (Tandel et al., 2022).

In general, these findings reinforce well-known seismological concepts where the magnitude and epicentral location of the earthquake are the most important factors when it comes to tsunami generation, and depth plays a minor role. This observation also shows that the machine-learning model can capture the underlying geophysical processes that generated the tsunami.

## 5.5 Chapter Summary

This chapter reveals the XGBoost and Naïve Bayes models demonstrated excellent performance in tsunami prediction with accuracy more than 90% and AUC more than 98% whereas the XGBoost model is more effective on the high-stakes application. This model's performance achieved a recall of 98.5%, missing 3 tsunami events with a reduced precision of 37.6% compared to 15.4% of Naïve Bayes. This highlights the significance of prioritizing recall in disaster forecast, where false negatives are very important to avoid. The importance of features further attributed to the importance of the earthquake magnitude and location as predictors which makes XGBoost is the best model to use in tsunami early warning systems. The results of this prediction model then saved into CSV file to further visualization.

# CHAPTER 6

# VISUALIZATION

## 6.1 Introduction

This chapter used Power BI dashboards with ArcGIS Maps to provide a multidimensional analysis of earthquake activity in the world between 2015 and 2024. The prediction results of XGBoost and NB model that saved into CSV file then linked to Power BI using default connector to generate the interactive dashboard for better visualization. The dashboards have been structured into three main sections, which are Overview, Analysis, and Map. Each having a different function to unveil the patterns, distributions, and trends. All these dashboards make up a comprehensive decision-support system to researchers, policymakers, and disaster management authorities.

## 6.2 Earthquake Overview

Figure 6.1 shows a dashboard that offers a complete view of earthquake activity around the globe based on statistics and visual information. Key performance indicators (KPIs) are presented at the top containing a summary of important metrics where 72,560 earthquakes were recorded in total, 1,004 of them result in the tsunamis, the largest magnitude 8.30, and the smallest depth 0 km which is surface-level earthquakes. These headline numbers provide the user with a quick overview of the magnitude and effect of seismic activity during the time period under analysis.


Figure 6.1 Earthquake overview dashboard

In the upper part of the dashboard, the filter controls are implemented to allow users to narrow the visualization to the Countries, Year/Month, Magnitude, Tsunami occurrence and Depth. As an example, users may filter high-magnitude earthquakes (those with magnitudes above 7.0), filter shallow earthquakes and deep earthquakes, and filter earthquakes that have caused a tsunami. This flexibility will turn the map into a moving image of a visualization to a powerful analytical tool.

Moreover, this dashboard also offers a detailed breakdown of the earthquake patterns using several visualizations. The Trend by Year chart shows the trend of earthquake activity, with the highest number of 8.91k in 2021 and then decreasing in the following years thus showing the change in seismic activity with time. The Total Earthquakes by Month bar chart shows that the frequency of earthquake is not much fluctuating over the months though the highest level of earthquake activity is experienced in the month of December with 6.9k cases. On the other hand, the Total Earthquakes by Magnitude chart indicates that most earthquakes have a moderate magnitude of 4 with 55.73k recorded cases, which is a significant number to indicate that most of the seismic events are moderate in intensity.

Further information is provided on the distribution of depths, where most earthquakes occur at shallow depths within the 0-50 km depth range with 52.20k events. These shallower occurrences are usually more devastating because of the closeness of energy release to the earth surface. Lastly, the donut chart illustrating tsunami-generating earthquakes shows that only 1.4% (1,004 events) of all earthquakes recorded caused tsunamis, while 98.6% events not generate tsunami. In general, these insights can help stakeholders such as researchers, disaster-management officials, and policymakers to improve the comprehension of the occurrences of earthquakes, assess risks better, and develop preparedness and resilience strategies.

## 6.3 Earthquake Analysis

The Earthquake Analysis dashboard presented in Figure 6.2 provides an in-depth overview of the activity of the global seismic activity split by continent, country, and location and further interrogates the seismic properties and the predictive model functionality.

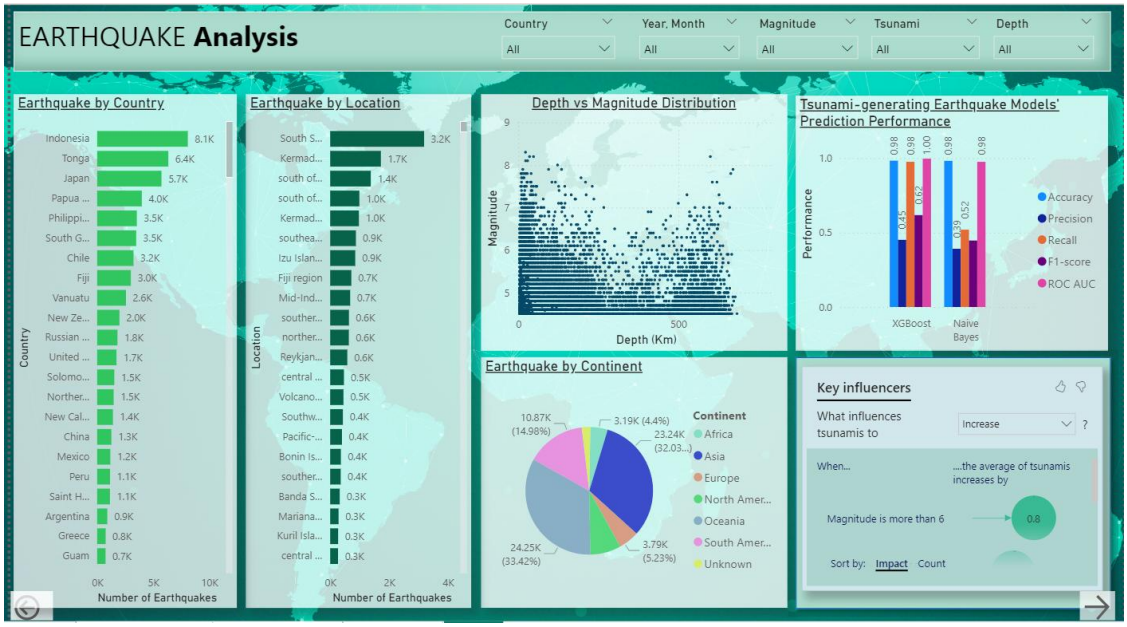



Figure 6.2 Earthquake analysis dashboard

The pie chart at the bottom left suggests that the highest number of earthquakes occurs in Oceania with 33.42% (24.25k), then followed by Asia by 32.03 % (23.24k), South America of 14.98% (10.87k), North America and Europe present large shares as well, but Africa presents the least events. This distribution highlights that most of the occurrence of earthquakes takes place along tectonic boundaries especially along the Pacific Ring of Fire.

The main panels represent the frequency of earthquakes by country and specific locations. Indonesia has the largest number of events recorded at national level with 8.1k earthquakes, Tonga by 6.4k cases, 5.7k cases in Japan and Papua New Guinea recorded 4.0k cases. At specific location, the bar graph shows highest seismic activities located in South Sandwich Islands with 3.2k events, Kermadec Islands recorded 1.7k events, and in the South of Tonga, Fiji and Japanese neighboring area that all located at active subduction zones. These observations underscore the disposition of the clustering of earthquakes around island arcs, in trenches, and in boundary zones to determine those regions that are most in need of the strongest monitoring and preparedness strategies.

The Depth vs Magnitude scatterplot demonstrates that most earthquakes have a hypocentral depth of between 0 to 200km and a moderate magnitude of 4 to 6, thus, present a potentially higher devastating hazard even with their relative occurrence. Conversely, deeper seismic events are less common but it can have higher magnitudes but generate least impact of surface risk. The dashboard on the right side evaluates those earthquake models which can produce tsunamis. It shows both XGBoost and Naïve Bayes algorithm has high accuracy but XGBoost has highest recalls by 0.9755 compared to NB classifier even though it has higher precision of 0.6842.

Moreover, the panel of Key Influencers reveals that the likelihood of a tsunami will be higher, when the magnitude of an earthquake is greater than 6. Overall, the dashboard effectively consolidates geographic, geophysical, predictive data and is a powerful tool to use in scientific research, disaster preparedness, and policy development.

## 6.4 Earthquake Map

The Earthquake Map Dashboard in Figure 6.3, is the spatial focal point of the seismic reporting system which is achieved by integrating ArcGIS and Power BI. In contrast with earlier dashboards, which focus more on statistical computations and categoric disaggregation the current dashboard focuses on the geographic distribution of seismic events between 2015 and 2024. It represents the earthquake events in the form of color-coded spots on the world map, superimposed by the tectonic plate boundaries thus creating a good visual analogy between seismicity and geological formations.

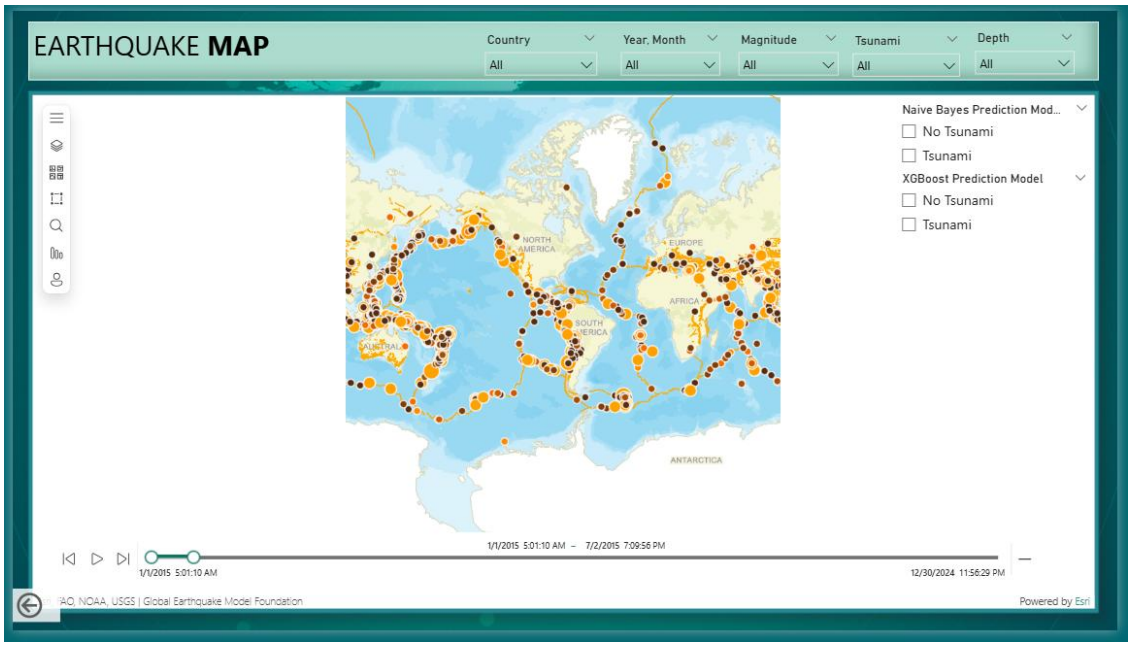



Figure 6.3 Earthquake map dashboard of seismic activity and tectonic plate

This map enables comparison between predictive models, and checkboxes enable users to switch between Naïve Bayes and XGBoost classifications of tsunami and non-tsunami events. This allows not only to visualize the location of seismic events, but also to visualize how each of the models views its tsunami potential. The Pacific Ring of Fire, the Mid-Atlantic Ridge, and other regions of tectonic collisions have been the epicenter of seismic events, and other areas of continental interiors are relatively less active.

An interactive chronology is provided by a temporal slider at the bottom that allows viewers to explore the development of seismic activity over a period of approximately ten years. This possibility allows determining clusters, aftershock sequences, and regional patterns to evolve over time. With the combination of spatial visualization and a combination of model predictions and time-filtering data, the Earthquake Map Dashboard transforms raw seismic data into an interactive story that informs disaster preparedness, scientific research, and policy making.

## 6.5 Chapter Summary

This chapter explains visualization of datasets and predictions of tsunami generating earthquake predictions using Power BI dashboard that emphasizes the importance of presenting data clearly, concisely, and consistently to ease the understanding and support accurate decision-making. The development of dashboards and interactive maps used in this study can understand global earthquake activity from 2015-2024 by displaying a summary of data, trends, magnitude distribution, depth, and tsunami events. Through filters, geographic visualization, and tsunami prediction models such as XGBoost and Naïve Bayes, this dashboard helps researchers, governments, and related parties assess risk, monitor seismic patterns, and strengthen disaster mitigation strategies.

# CHAPTER 7

# CONCLUSION

## 7.1 Conclusion

This finding proved that machine learning can be used to improve the prediction of tsunami through the analysis of global seismic characteristics including magnitude, depth, latitude, and longitude. XGBoost worked best of the tested models with 97.7% accuracy and 98.5% recall, meaning that almost all earthquakes that generate tsunami were detected although with high false alarm compared to Naïve Bayes. A comparative analysis with past studies strengthened the idea that the recall-based ensemble methods are more appropriate in disaster preparedness since it is important to limit missed events. In addition to predictive performance, exploratory analysis indicated that there were important spatiotemporal patterns, the highest seismic activity was in 2021 and was concentrated along the Pacific Ring of Fire especially in Indonesia, Tonga, Japan, United State and Canada. All these results indicate the importance of combining advanced machine learning with seismological data to enhance early warning systems, better risk management, and protect vulnerable populations at risk.

## 7.2 Limitation

Although this research offers important information regarding earthquake and tsunami prediction, there are several limitations associated with it. First, the data employed from 2015-2024 is limited in time and might not be able to fully reflect long-term seismic cycles or occasionally rare extreme events, which might do have an impact on the model generalizability. Next, the data was skewed having significantly fewer tsunami compared to non-tsunami earthquakes, which made it harder to use models and decreased the ability to recall with recall-centric models like XGBoost.

In addition, a small number of seismic variables such as magnitude, depth, latitude, and longitude were incorporated, excluding other geological/geophysical variables of interest like wave information, fault structures, tectonic stress regimes or seafloor deformation that might be predictive. Then, the models were trained using historical data and not on real-time seismic streams which means that their ability to operate in real-time early warning systems has not been tested. Finally, computational resources limited the detection of deeper deep learning or hybrid ensemble models that could potentially achieve better predictive performance and false alarms.

## 7.3 Achievement

The study has been able to meet its targets by establishing and assessing machine learning models to improve prediction of earthquakes that produce a tsunami including:

(a) Objective 1 achieved and explained in Chapter 4: Initial finding of a wide scope of EDA revealing the important spatiotemporal trends like the highest number of earthquakes in 2021 and the presence of seismic activity concentration along the Pacific Ring of Fire especially in Indonesia, Tonga, Japan, United State and Canada that also generated most of the tsunami events.

(b) Objective 2 achieved and explained in Chapter 5: XGBoost model was proven to be most effective model by providing high accuracy of 97.7% and recall 98.5%, which makes it possible to reduce missed tsunami events. Latitude and magnitude are the most important features in detecting and predicting tsunami events.

(c) Objective 3 also achieved and explained in Chapter 6: Generating an interactive dashboard providing convenient visualisation of seismic activities, user-friendly for predictive guidelines and high-risk areas and thus proving the viability of machine-learning applications in disaster preparedness. In addition to the contributions to the body of knowledge in geoinformatics and machine learning, this study provides a large-scale of practical implications to design

early warning systems and improve risk management procedures especially in the population at risk due to seismic hazards.

## 7.4 Future Improvement

Further improvements in the future should be made to enlarge the dataset to capture more geological and geophysical variables such as fault geometries, stress accumulation measures and seafloor deformation that can increase strength and interpretability. These models would be deployed more quickly in real-time with more reliable notifications using real-time seismic data streams as a component of operational early warning systems. Hybrid ensembles that combine boosting algorithms, random forests, and deep learning designs could also balance recall and precision and thus minimize missed events and false positives. Finally, predictive models could be incorporated in interactive decision- support dashboards to enhance their relevance by providing timely and actionable information to the policy makers and agencies involved in disaster management.

# REFERENCES


Airlangga, G. (2025). Machine Learning For Tsunami Prediction: A Comparative Analysis Of Ensemble And Deep Learning Models. *Kesatria : Jurnal Penerapan Sistem Informasi (Komputer Dan Manajemen)*, *6*, 302–311. Https://Doi.Org/10.30645/Kesatria.V6i1.572

Babu, D. B., Revathi, M. L., Senthil, M., Parvathi, A. L., Sceenilai, B., & Sheema, Sk. (2024). Earthquake Prediction Model Using Random Forest And Gradient Boosting Algorithms. *2024 9th International Conference On Communication And Electronics Systems (ICCES)*, 1597–1607. Https://Doi.Org/10.1109/ICCES63552.2024.10859534

Bangar, A., Ansari, S., Ali, S. S., Baderao, V., & Alhat, B. (2024). Disaster Prediction Using Appropriate Machine Learning Techniques. *2024 IEEE Pune Section International Conference (Punecon)*, 1–7. Https://Doi.Org/10.1109/Punecon63413.2024.10895154

Bao, H., Xu, L., Meng, L., Ampuero, J.-P., Gao, L., & Zhang, H. (2022). Global Frequency Of Oceanic And Continental Supershear Earthquakes. *Nature Geoscience*, *15*(11), 942–949. Https://Doi.Org/10.1038/S41561-022-01055-5

Bilek, S., & Lay, T. (2018). Subduction Zone Megathrust Earthquakes. *Geosphere*, *14*. Https://Doi.Org/10.1130/GES01608.1

Biswas, S., Kumar, D., & Bera, U. K. (2023). *Prediction Of Earthquake Magnitude And Seismic Vulnerability Mapping Using Artificial Intelligence Techniques: A Case Study Of Turkey*. In Review. Https://Doi.Org/10.21203/Rs.3.Rs-2863887/V1

Buforn, E., & Udías, A. (2010). Chapter 3—Azores–Tunisia, A Tectonically Complex Plate Boundary. In R. Dmowska (Ed.), *Advances In Geophysics* (Vol. 52, Pp. 139–182). Elsevier. Https://Doi.Org/10.1016/S0065-2687(10)52003-X

Bustos, K., Maazallahi, A., Salari, M. A., Snir, E., Norouzzadeh, P., & Rahmani, B. (2024). *Classifying And Forecasting Seismic Event Characteristics Using Artificial Intelligence*. In Review. Https://Doi.Org/10.21203/Rs.3.Rs-4249733/V1

Chandu, H. S. (2025). Accessing The Effectiveness Of Disaster Management Strategies Through Advance Techniques. *2025 IEEE International Conference On Interdisciplinary Approaches In Technology And Management For Social Innovation (IATMSI)*, 1–6. Https://Doi.Org/10.1109/IATMSI64286.2025.10984535

Chen, T., & Guestrin, C. (2016). Xgboost: A Scalable Tree Boosting System. *Proceedings Of The 22nd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 785–794. Https://Doi.Org/10.1145/2939672.2939785

Cilia, M. G., Mooney, W. D., & Nugroho, C. (2021). Field Insights And Analysis Of The 2018 Mw 7.5 Palu, Indonesia Earthquake, Tsunami And Landslides. *Pure And Applied Geophysics*, *178*(12), 4891–4920. Scopus. Https://Doi.Org/10.1007/S00024-021-02852-6

Cornely, P.-R., & Wang, J. (2023). Advancing Earthquake Prediction: A Comprehensive Review Of Data Science Techniques. *2023 6th International Conference On Computing And Big Data (ICCBD)*, 9–16. Https://Doi.Org/10.1109/ICCBD59843.2023.10607190

Cui, B., Guo, J., Han, G., & Liu, X. (2024). Earthquake Magnitude And Depth Prediction Based On Machine Learning And Multiple Linear Regression Models. *2024 IEEE 2nd International Conference On Sensors, Electronics And Computer Engineering (ICSECE)*, 1056–1060. Https://Doi.Org/10.1109/ICSECE61636.2024.10729410

Dogan, G. G., Yalciner, A. C., Annunziato, A., Yalciner, B., & Necmioglu, O. (2023). Global Propagation Of Air Pressure Waves And Consequent Ocean Waves Due To The January 2022 Hunga Tonga-Hunga Ha'apai Eruption. *Ocean Engineering*, *267*, 113174. Https://Doi.Org/10.1016/J.Oceaneng.2022.113174

Dragoni, M., & Santini, S. (2022). Contribution Of The 2010 Maule Megathrust Earthquake To The Heat Flow At The Peru-Chile Trench. *Energies*, *15*(6), Article 6. Https://Doi.Org/10.3390/En15062253

Duarte, J., & Schellart, W. (2016). Introduction To Plate Boundaries And Natural Hazards. In *Plate Boundaries And Natural Hazards* (P. 352 Pages). Https://Doi.Org/10.1002/9781119054146.Ch1

E. Yousif, M. (2022). *The Tsunami Mechanism*. Https://Doi.Org/10.1002/Essoar.10509559.1

Ertuncay, D., & Costa, G. (2021). Determination Of Near-Fault Impulsive Signals With Multivariate Naïve Bayes Method. *Natural Hazards*, *108*(2), 1763–1780. Https://Doi.Org/10.1007/S11069-021-04755-0

Fawcett, T. (2006). An Introduction To ROC Analysis. *Pattern Recognition Letters*, *27*(8), 861–874. Https://Doi.Org/10.1016/J.Patrec.2005.10.010

Firoozi, A. A., & Firoozi, A. A. (2023). Non-Seismic And Complex Source Tsunami: Unseen Hazard. *Intech Open*. Https://Doi.Org/DOI: 10.5772/Intechopen.1002308

Fischer, T., Hrubcová, P., Salama, A., Doubravová, J., Ágústsdóttir, T., Gudnason, E. Á., Horálek, J., & Hersir, G. P. (2022). Swarm Seismicity Illuminates Stress Transfer Prior To The 2021 Fagradalsfjall Eruption In Iceland. *Earth And Planetary Science Letters*, *594*, 117685. Https://Doi.Org/10.1016/J.Epsl.2022.117685

Geersen, J., Sippl, C., & Harmon, N. (2022). Impact Of Bending-Related Faulting And Oceanic-Plate Topography On Slab Hydration And Intermediate-Depth Seismicity. *Geosphere*, *18*. Https://Doi.Org/10.1130/GES02367.1

Gurnis, M., Zhong, S., & Toth, J. (2000). On The Competing Roles Of Fault Reactivation And Brittle Failure In Generating Plate Tectonics From Mantle Convection. *Geophysical Monograph Series*, *121*, 73–94. Https://Doi.Org/10.1029/GM121p0073

Güvercin, S. E., Karabulut, H., Konca, A. Ö., Doğan, U., & Ergintav, S. (2022). Active Seismotectonics Of The East Anatolian Fault. *Geophysical Journal International*, *230*(1), 50–69. Https://Doi.Org/10.1093/Gji/Ggac045

Handayani, T., Wijayanto, Wijaya, A., Himantara, L., Saputro, A. H., & Djuhana, D. (2024). Machine Learning Implementation For Estimation Of Earthquake Magnitude Using Strong-Motion Data. *2024 4th International Conference On Robotics, Automation And Artificial Intelligence (RAAI)*, 351–355. Https://Doi.Org/10.1109/RAAI64504.2024.10949525

Harirchian, E., Kumari, V., Jadhav, K., Rasulzade, S., Lahmer, T., & Raj Das, R. (2021). A Synthesized Study Based On Machine Learning Approaches For Rapid Classifying Earthquake Damage Grades To RC Buildings. *Applied Sciences*, *11*(16), 7540. Https://Doi.Org/10.3390/App11167540

Heidarzadeh, M., Muhari, A., & Wijanarto, A. B. (2019). Insights On The Source Of The 28 September 2018 Sulawesi Tsunami, Indonesia Based On Spectral

Analyses And Numerical Simulations. *Pure And Applied Geophysics*, *176*(1), 25–43. Https://Doi.Org/10.1007/S00024-018-2065-9

Hill, D. P., & Prejean, S. G. (2015). Dynamic Triggering. In *Treatise On Geophysics: Second Edition* (Vol. 4, Pp. 273–304). Scopus. Https://Doi.Org/10.1016/B978-0-444-53802-4.00078-6

Hoque, A., Raj, J., Saha, A., & Bhattacharya, D. (2020). *Earthquake Magnitude Prediction Using Machine Learning Technique*.

James, D. E. (2021). Lithosphere, Continental. In H. K. Gupta (Ed.), *Encyclopedia Of Solid Earth Geophysics* (Pp. 866–872). Springer International Publishing. Https://Doi.Org/10.1007/978-3-030-58631-7_32

Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image Recognition Of Four Rice Leaf Diseases Based On Deep Learning And Support Vector Machine. *Computers And Electronics In …*, *Query Date: 2024-12-01 16:19:06*. Https://Www.Sciencedirect.Com/Science/Article/Pii/S016816992030795X

Kaftan, I. (2025). *Machine Learning Applications For Earthquake Magnitude Prediction In Western Türkiye* (SSRN Scholarly Paper No. 5234889). Social Science Research Network. Https://Doi.Org/10.2139/Ssrn.5234889

Karimzadeh, S., Matsuoka, M., Kuang, J., & Ge, L. (2019a). Spatial Prediction Of Aftershocks Triggered By A Major Earthquake: A Binary Machine Learning Perspective. *ISPRS International Journal Of Geo-Information*, *8*(10), 462. Https://Doi.Org/10.3390/Ijgi8100462

Karimzadeh, S., Matsuoka, M., Kuang, J., & Ge, L. (2019b). Spatial Prediction Of Aftershocks Triggered By A Major Earthquake: A Binary Machine Learning Perspective. *ISPRS International Journal Of Geo-Information*, *8*(10), 462. Https://Doi.Org/10.3390/Ijgi8100462

Kazbekova, G., Aben, A., Amanov, A., Zhunissov, N., & Abibullayeva, A. (2025). Effectiveness Of Machine Learning Methods In Determining Earthquake Probable Areas: Example Of Kazakhstan. *Scientific Journal Of Astana IT University*. Https://Doi.Org/10.37943/21KUXZ6354

Kukartsev, V., & Degtyareva, K. (2024). Forecasting Seismic Activity Using Machine Learning Algorithms. *E3S Web Of Conferences*, *592*, 05002. Https://Doi.Org/10.1051/E3sconf/202459205002

Lay, T. (2016). Great Earthquakes On Plate Boundaries. *Oxford Research Encyclopedia Of Natural Hazard Science*.

Https://Oxfordre.Com/Naturalhazardscience/Display/10.1093/Acrefore/9780199389407.001.0001/Acrefore-9780199389407-E-32

Lay, T., Ye, L., Wu, Z., & Kanamori, H. (2020). Macrofracturing Of Oceanic Lithosphere In Complex Large Earthquake Sequences. *Journal Of Geophysical Research: Solid Earth*, *125*(10). Scopus. Https://Doi.Org/10.1029/2020JB020137

Leonard, L. J., & Bednarski, J. M. (2015). The Preservation Potential Of Coastal Coseismic And Tsunami Evidence Observed Following The 2012 Mw 7.8 Haida Gwaii Thrust Earthquake. *Bulletin Of The Seismological Society Of America*, *105*(2B), 1280–1289. Https://Doi.Org/10.1785/0120140193

Liu, Z., & Buck, W. R. (2018). Magmatic Controls On Axial Relief And Faulting At Mid-Ocean Ridges. *Earth And Planetary Science Letters*, *491*, 226–237. Https://Doi.Org/10.1016/J.Epsl.2018.03.045

Lu, R., Xu, X., He, D., John, S., Liu, B., Wang, F., Tan, X., & Li, Y. (2017). Seismotectonics Of The 2013 Lushan Mw 6.7 Earthquake: Inversion Tectonics In The Eastern Margin Of The Tibetan Plateau. *Geophysical Research Letters*, *44*(16), 8236–8243. Https://Doi.Org/10.1002/2017GL074296

Luo, S., Yao, H., Zhang, Z., & Bem, T. S. (2022). High-Resolution Crustal And Upper Mantle Shear-Wave Velocity Structure Beneath The Central-Southern Tanlu Fault: Implications For Its Initiation And Evolution. *Earth And Planetary Science Letters*, *595*, 117763. Https://Doi.Org/10.1016/J.Epsl.2022.117763

Maazallahi, A., Bustos, K., Salari, M. A., Snir, E., Norouzzadeh, P., & Rahmani, B. (2025). Classifying Seismic Event Parameters Using Artificial Intelligence. *Academia Environmental Sciences And Sustainability*, *2*(2). Https://Www.Academia.Edu/2997-6006/2/2/10.20935/Acadenvsci7719

Macheyeki, A. S. (2024). Present-Day Fault Kinematics And Their Reactivation Likelihood Within And South Of The North Tanzania Divergence (NTD), East African Rift System: Implication For Geo-Hazards Assessment. *Journal Of The Geological Society Of India*, *100*(1), 127–138. Scopus. Https://Doi.Org/10.17491/Jgsi/2024/172989

Mahmoud, A., Alrusaini, O., Shafie, E., Aboalndr, A., & S.Elbelkasy, M. (2025). Machine Learning-Based Earthquake Prediction: Feature Engineering And Model Performance Using Synthetic Seismic Data. *Applied Mathematics &*

*Information Sciences*, *19*(3), 695–702. Https://Doi.Org/10.18576/Amis/190317

Manaman, N. S., & Shomali, H. (2010). Upper Mantle S-Velocity Structure And Moho Depth Variations Across Zagros Belt, Arabian–Eurasian Plate Boundary. *Physics Of The Earth And Planetary Interiors*, *180*(1), 92–103. Https://Doi.Org/10.1016/J.Pepi.2010.01.011

Mera, W., Vera, X., Antonio, L. T., & Ponce, G. (2017). April 2016 Ecuador Earthquake Of Moment Magnitude Mw7.8: Overview And Damage Report. *Key Engineering Materials*, *747 KEM*, 662–669. Scopus. Https://Doi.Org/10.4028/Www.Scientific.Net/KEM.747.662

Mesta, C., Kerschbaum, D., Cremen, G., & Galasso, C. (2023). Quantifying The Potential Benefits Of Risk-Mitigation Strategies On Present And Future Seismic Losses In Kathmandu Valley, Nepal. *Earthquake Spectra*, *39*(1), 377–401. Scopus. Https://Doi.Org/10.1177/87552930221134950

Nakamura, Y., Kodaira, S., Fujie, G., Yamashita, M., Obana, K., & Miura, S. (2023). Incoming Plate Structure At The Japan Trench Subduction Zone Revealed In Densely Spaced Reflection Seismic Profiles. *Progress In Earth And Planetary Science*, *10*(1), 45. Https://Doi.Org/10.1186/S40645-023-00579-7

Narne, H. (2023). Analyzing Tsunami Occurrence And Predictive Techniques: Enhancing Early Warning Systems With Machine Learning. *International Journal Of Science And Research (IJSR)*. Https://Doi.Org/10.21275/SR241112205044

Navarro, A., Castro-Artola, O., García-Guerrero, E., Aguirre-Castro, O., Tamayo Pérez, U., López-Mercado, C., & Inzunza Gonzalez, E. (2025). Recent Advances In Early Earthquake Magnitude Estimation By Using Machine Learning Algorithms: A Systematic Review. *Applied Sciences*, *15*, 1. Https://Doi.Org/10.3390/App15073492

Newbrough, M. (2025, July 11). *Understanding Plate Motions*. United State Geological Survey (USGS). Https://Pubs.Usgs.Gov/Gip/Dynamic/Understanding.Html

Novick, D., & Last, M. (2023). Using Machine Learning Models For Earthquake Magnitude Prediction In California, Japan, And Israel. *Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence*

*And Lecture Notes In Bioinformatics)*, *13914 LNCS*, 151–169. Scopus. Https://Doi.Org/10.1007/978-3-031-34671-2_11

Olive, J.-A. (2023). Mid-Ocean Ridges: Geodynamics Written In The Seafloor. In *Dynamics Of Plate Tectonics And Mantle Convection* (Pp. 483–510). Scopus. Https://Doi.Org/10.1016/B978-0-323-85733-8.00018-4

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Louppe, G., Prettenhofer, P., Weiss, R., Weiss, R. J., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-Learn: Machine Learning In Python. *Journal Of Machine Learning Research*. Https://Doi.Org/10.5555/1953048.2078195

Pham, B. T., Bui, D. T., & Prakash, I. (2018). Bagging Based Support Vector Machines For Spatial Prediction Of Landslides. *Environmental Earth Sciences*. Https://Doi.Org/10.1007/S12665-018-7268-Y

Pisarenko, V. F., & Pisarenko, D. V. (2022). A Modified K-Nearest-Neighbors Method And Its Application To Estimation Of Seismic Intensity. *Pure And Applied Geophysics*, *179*(11), 4025–4036. Scopus. Https://Doi.Org/10.1007/S00024-021-02717-Y

Pwavodi, J., Ibrahim, A. U., Pwavodi, P. C., Al-Turjman, F., & Mohand-Said, A. (2024). The Role Of Artificial Intelligence And Iot In Prediction Of Earthquakes: Review. *Artificial Intelligence In Geosciences*, *5*, 100075. Https://Doi.Org/10.1016/J.Aiig.2024.100075

Raschka, S. (2020). *Model Evaluation, Model Selection, And Algorithm Selection In Machine Learning* (No. Arxiv:1811.12808). Arxiv. Https://Doi.Org/10.48550/Arxiv.1811.12808

Rish, I. (2001). *An Empirical Study Of The Naïve Bayes Classifier*. 41–46. Https://Www.Researchgate.Net/Publication/228845263_An_Empirical_Study_Of_The_Naive_Bayes_Classifier

Saito, T., & Rehmsmeier, M. (2015a). The Precision-Recall Plot Is More Informative Than The ROC Plot When Evaluating Binary Classifiers On Imbalanced Datasets. *PLOS ONE*, *10*(3), E0118432. Https://Doi.Org/10.1371/Journal.Pone.0118432

Saito, T., & Rehmsmeier, M. (2015b). The Precision-Recall Plot Is More Informative Than The ROC Plot When Evaluating Binary Classifiers On Imbalanced

Datasets. *PLOS ONE*, *10*(3), E0118432. Https://Doi.Org/10.1371/Journal.Pone.0118432

Sartori, R., Torelli, L., Zitellini, N., Peis, D., & Lodolo, E. (1994). Eastern Segment Of The Azores-Gibraltar Line (Central-Eastern Atlantic): An Oceanic Plate Boundary With Diffuse Compressional Deformation. *Geology*, *22*(6), 555–558. Https://Doi.Org/10.1130/0091-7613(1994)022<0555:ESOTAG>2.3.CO;2

Sassa, S., & Takagawa, T. (2019). Liquefied Gravity Flow-Induced Tsunami: First Evidence And Comparison From The 2018 Indonesia Sulawesi Earthquake And Tsunami Disasters. *Landslides*, *16*(1), 195–200. Https://Doi.Org/10.1007/S10346-018-1114-X

Satish, S., Gonaygunta, H., Yadulla, A. R., Kumar, D., Maturi, M. H., Meduri, K., Cruz, E. D. L., Nadella, G. S., & Sajja, G. S. (2025). Forecasting The Unseen: Enhancing Tsunami Occurrence Predictions With Machine-Learning-Driven Analytics. *Computers 2025*, *14*(5), 175. Https://Doi.Org/10.3390/Computers14050175

Satish, S., Gonaygunta, H., Yudalla, A. R., Kumar, D., Mohan, H. M., Meduri, K., & Cruz, E. D. L. (2025). Forecasting The Unseen: Enhancing Tsunami Occurrence Predictions With Machine-Learning-Driven Analytics. *Computers 2025*, *14*(5), 175. Https://Doi.Org/10.3390/Computers14050175

Senkaya, M., Silahtar, A., Erkan, E. F., & Karaaslan, H. (2024). Prediction Of Local Site Influence On Seismic Vulnerability Using Machine Learning: A Study Of The 6 February 2023 Türkiye Earthquakes. *Engineering Geology*, *337*, 107605. Https://Doi.Org/10.1016/J.Enggeo.2024.107605

Shahzada, K., Noor, U. A., & Xu, Z.-D. (2025). In The Wake Of The March 28, 2025 Myanmar Earthquake: A Detailed Examination. *Journal Of Dynamic Disasters*, *1*(2), 100017. Https://Doi.Org/10.1016/J.Jdd.2025.100017

Singh, A. (2025). Global Earthquake Dataset (2015–2024). Kaggle.com. https://www.kaggle.com/adi2606

Stein, S., & Klosko, E. (2002). 7—Earthquake Mechanisms And Plate Tectonics. *International Geophysics*, *81*, 69–78. Https://Doi.Org/10.1016/S0074-6142(02)80210-8

Sukmana, H. T., Derachman, Y., Amri, A., & Supardi, S. (2024). Comparative Analysis Of SVM And RF Algorithms For Tsunami Prediction: A Performance

Evaluation Study. *Journal Of Applied Data Sciences*, *5*(1), 84–99. Https://Doi.Org/10.47738/Jads.V5i1.159

Suppasri, A., Kitamura, M., Alexander, D., Seto, S., & Imamura, F. (2024). The 2024 Noto Peninsula Earthquake: Preliminary Observations And Lessons To Be Learned. *International Journal Of Disaster Risk Reduction*, *110*. Scopus. Https://Doi.Org/10.1016/J.Ijdrr.2024.104611

Tandel, P., Patel, H., & Patel, T. (2022). Tsunami Wave Propagation Model: A Fractional Approach. *Journal Of Ocean Engineering And Science*, *7*(6), 509–520. Https://Doi.Org/10.1016/J.Joes.2021.10.004

Truong, V.-H., Tangaramvong, S., Nguyen, M.-C., & Pham, H.-A. (2025). Machine Learning-Based Safety Assessment Of Steel Frames Under Seismic Loadings Using Nonlinear Time-History Analysis. *Steel And Composite Structures*, *54*(4), 295–312. Scopus. Https://Doi.Org/10.12989/Scs.2025.54.4.295

Ulrich, T., Vater, S., Madden, E. H., Behrens, J., Dinther, Y. Van, Zelst, I. Van, Fielding, E. J., Liang, C., & Gabriel, A.-A. (2019). *Coupled, Physics-Based Modeling Reveals Earthquake Displacements Are Critical To The 2018 Palu, Sulawesi Tsunami*. Https://Eartharxiv.Org/Repository/View/1030/

USGS. (2016, July). *Earthword–Subduction | U.S. Geological Survey*. Https://Www.Usgs.Gov/News/Science-Snippet/Earthword-Subduction

Velarde, G., Weichert, M., Deshmunkh, A., Deshmane, S., Sudhir, A., Sharma, K., & Joshi, V. (2024). Tree Boosting Methods For Balanced And Imbalanced Classification And Their Robustness Over Time In Risk Assessment. *Intelligent Systems With Applications*, *22*, 200354. Https://Doi.Org/10.1016/J.Iswa.2024.200354

Wang, C., Deng, C., & Wang, S. (2020). Imbalance-Xgboost: Leveraging Weighted And Focal Losses For Binary Label-Imbalanced Classification With Xgboost. *Pattern Recognition Letters*, *136*, 190–197. Https://Doi.Org/10.1016/J.Patrec.2020.05.035

Wang, J., Shahani, N. M., Zheng, X., Hongwei, J., & Wei, X. (2025). Machine Learning-Based Analyzing Earthquake-Induced Slope Displacement. *PLOS ONE*, *20*(2), E0314977. Https://Doi.Org/10.1371/Journal.Pone.0314977

Wang, T., Bian, Y., Zhang, Y., & Hou, X. (2023). Classification Of Earthquakes, Explosions And Mining-Induced Earthquakes Based On Xgboost Algorithm.

*Computers & Geosciences*, *170*, 105242. Https://Doi.Org/10.1016/J.Cageo.2022.105242

Wang, X., Cao, L., Zhao, M., Cheng, J., & He, X. (2023). What Conditions Promote Atypical Subduction: Insights From The Mussau Trench, The Hjort Trench, And The Gagua Ridge. *Gondwana Research*, *120*, 207–218. Scopus. Https://Doi.Org/10.1016/J.Gr.2022.10.014

Wong, T.-T., & Yang, N.-Y. (2017). Dependency Analysis Of Accuracy Estimates In K-Fold Cross Validation. *IEEE Transactions On Knowledge And Data Engineering*, *29*(11), 2417–2427. Https://Doi.Org/10.1109/TKDE.2017.2740926

Wu, F. Y., Wang, J. G., Liu, C. Z., Liu, T., Zhang, C., & Ji, W. Q. (2019). Intra-Oceanic Arc: Its Formation And Evolution. *Yanshi Xuebao/Acta Petrologica Sinica*, *35*(1), 1–15. Scopus. Https://Doi.Org/10.18654/1000-0569/2019.01.01

Wu, Z.-N., Li, Z.-Q., Dong, Y., Han, X.-L., Zhang, G., Feng, R., & Zhu, K. (2024). Seismic Intensity Measure Selection Incorporating Interaction Effects For Damage Assessment Across Different Structural Sensitive Regions. *Structures*, *67*, 106917. Https://Doi.Org/10.1016/J.Istruc.2024.106917

Yadav, S., & Shukla, S. (2016). Analysis Of K-Fold Cross-Validation Over Hold-Out Validation On Colossal Datasets For Quality Classification. *2016 IEEE 6th International Conference On Advanced Computing (IACC)*, 78–83. Https://Doi.Org/10.1109/IACC.2016.25

Yenidoğan, C. (2024). February 6, 2023 Earthquakes And Preliminary Assessment Of Building Damage Based On Field Surveys. *Turkish Journal Of Civil Engineering*, *35*(5), 75–113. Scopus. Https://Doi.Org/10.18400/Tjce.1335742

Zhang, H. (2011). Exploring Conditions For The Optimality Of Naïve Bayes. *International Journal Of Pattern Recognition And Artificial Intelligence*, *19*, 183–198. Https://Doi.Org/10.1142/S0218001405003983

## Appendix A Gantt Chart Project 1 and Project 2

| Task | Month | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Jan | Feb | Mar | Apr | May | June | July | Aug | Sep |
| **1. Desk Study** | | | | | | | | | |
| **2. Proposal Development** | | | | | | | | | |
| 2.1 Research Question and Objective | X | | | | | | | | |
| 2.2 Literature Review | X | | | | | | | | |
| 2.3 Proposal Submission and Approval | | X | | | | | | | |
| **Project 1 :** | | | | | | | | | |
| **3. Data Collection** | | | | | | | | | |
| 3.1 USGS Global Earthquake Data Catalog | X | X | | | | | | | |
| **4. Data Pre-processing** | | | | | | | | | |
| 4.1 Data Cleaning | | | X | X | | | | | |
| **5. Exploratory Data Analysis (EDA)** | | | | | X | | | | |
| **6. Thesis Writing** | X | X | X | X | X | X | | | |
| **7. Thesis Submission** | | | | | X | X | | | |
| **8. Thesis Presentation** | | | | | | X | | | |
| **9. Thesis Revision and Final Submission** | | | | | | | X | | |
| **Project 2:** | | | | | | | | | |
| **1. Feature Engineering** | | | | | X | X | X | | |
| **2. Model Development and Evaluation** | | | | | | | | | |
| 2.1 Random Forest and XGBoost | | | | | | X | X | X | |
| 2.2 Training Machine Learning Model | | | | | | | X | X | X |
| 2.3 Model Validation and Testing | | | | | | | X | X | X |
| 3. **Results Analysis** | | | | | | | | | |
| 3.1 Performance Metrics Analysis | | | | | | | X | | |
| 4.2 Visualization of Results | | | | | | | | X | |
| **4. Thesis Writing** | X | X | X | X | X | X | X | X | X |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **5. Thesis Submission** | | | | | | | | | X |
| **6. Thesis Presentation** | | | | | | | | | X |
| **7. Thesis Revision and Final Submission** | | | | | | | | | X |

# Appendix B Python Code

DATA Loading

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Earthquakes data/EarthquakeData (2015-2024) (1).csv')
print("\nEarthquake dataset:")
df.head()
```

Earthquake dataset:

| | time | place | magnitude | depth_km | longitude | latitude | type | alert | tsunami | status | id |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2015-01-01 05:01:10.640 | near the east coast of Honshu, Japan | 4.8 | 41.39 | 142.0405 | 38.8957 | earthquake | NaN | 0 | reviewed | ,usc000tb3v, |
| 1 | 2015-01-01 06:48:29.670 | 93 km N of Isangel, Vanuatu | 4.6 | 223.61 | 169.1795 | -18.7052 | earthquake | NaN | 0 | reviewed | ,usc000tb42, |
| 2 | 2015-01-01 06:54:20.570 | central Mid-Atlantic Ridge | 4.7 | 10.00 | -31.7641 | 3.4769 | earthquake | NaN | 0 | reviewed | ,usc000tb46, |
| 3 | 2015-01-01 07:12:44.230 | 120 km SSE of Kirakira, Solomon Islands | 4.6 | 26.24 | 162.4998 | -11.3818 | earthquake | NaN | 0 | reviewed | ,usc000tb4a, |
| 4 | 2015-01-01 08:49:53.200 | 70 km W of F?r?z?b?d, Iran | 5.1 | 10.10 | 51.8580 | 28.7280 | earthquake | NaN | 0 | reviewed | ,usc000tb4f,iscgem606436879, |

```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72583 entries, 0 to 72582
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   time       72583 non-null  object
 1   place      72583 non-null  object
 2   magnitude  72583 non-null  float64
 3   depth_km   72583 non-null  float64
 4   longitude  72583 non-null  float64
 5   latitude   72583 non-null  float64
 6   type       72583 non-null  object
 7   alert      6427 non-null   object
 8   tsunami    72583 non-null  int64
 9   status     72583 non-null  object
 10  id         72583 non-null  object
dtypes: float64(4), int64(1), object(6)
memory usage: 6.1+ MB
```

| | magnitude | depth_km | longitude | latitude | tsunami |
|---|---|---|---|---|---|
| count | 72583.000000 | 72583.000000 | 72583.000000 | 72583.000000 | 72583.000000 |
| mean | 4.803389 | 63.121136 | 33.304390 | -1.804892 | 0.013832 |
| std | 0.371220 | 115.074867 | 123.142047 | 29.754361 | 0.116796 |
| min | 4.500000 | -1.010000 | -179.999700 | -79.983700 | 0.000000 |
| 25% | 4.500000 | 10.000000 | -72.288800 | -22.229850 | 0.000000 |
| 50% | 4.700000 | 14.196000 | 92.451800 | -5.115800 | 0.000000 |
| 75% | 4.900000 | 57.718000 | 141.491750 | 19.325833 | 0.000000 |
| max | 8.300000 | 683.360000 | 179.999300 | 87.386000 | 1.000000 |

```
df.shape
```

```
(72583, 11)
```

```
df.columns
```

```
Index(['time', 'place', 'magnitude', 'depth_km', 'longitude', 'latitude',
       'type', 'alert', 'tsunami', 'status', 'id'],
      dtype='object')
```

```
# Identify unique and non-unique values

# Get the number of unique values in each column
unique_counts = df.nunique()

print("Number of unique values per column:")
print(unique_counts)

# Identify columns with only one unique value (non-unique)
non_unique_cols = unique_counts[unique_counts == 1].index.tolist()

print("\nColumns with only one unique value (non-unique):")
print(non_unique_cols)
```

```
df.isnull().sum()
```

| | 0 |
|---|---|
| **time** | 0 |
| **place** | 0 |
| **magnitude** | 0 |
| **depth_km** | 0 |
| **longitude** | 0 |
| **latitude** | 0 |
| **type** | 0 |
| **alert** | 66156 |
| **tsunami** | 0 |
| **status** | 0 |
| **id** | 0 |

**dtype:** int64

## Cleanning

```
] df['time'] = pd.to_datetime(df['time'])
```

```
] df.drop(['alert','status', 'id'],axis=1, inplace=True)
  df.head()
```

| | time | place | magnitude | depth_km | longitude | latitude | type | tsunami |
|---|---|---|---|---|---|---|---|---|
| **0** | 2015-01-01 05:01:10.640 | near the east coast of Honshu, Japan | 4.8 | 41.39 | 142.0405 | 38.8957 | earthquake | 0 |
| **1** | 2015-01-01 06:48:29.670 | 93 km N of Isangel, Vanuatu | 4.6 | 223.61 | 169.1795 | -18.7052 | earthquake | 0 |
| **2** | 2015-01-01 06:54:20.570 | central Mid-Atlantic Ridge | 4.7 | 10.00 | -31.7641 | 3.4769 | earthquake | 0 |
| **3** | 2015-01-01 07:12:44.230 | 120 km SSE of Kirakira, Solomon Islands | 4.6 | 26.24 | 162.4998 | -11.3818 | earthquake | 0 |
| **4** | 2015-01-01 08:49:53.200 | 70 km W of F?r?z?b?d, Iran | 5.1 | 10.10 | 51.8580 | 28.7280 | earthquake | 0 |

## Pre-processing

```
#  generate new column of country based on longitude and latitude

!pip install reverse_geocoder

import reverse_geocoder as rg

def get_country(row):
  coordinates = (row['latitude'], row['longitude'])
  result = rg.search(coordinates)
  return result[0]['cc']

df['Country'] = df.apply(get_country, axis=1)
```

```python
# Mapping country codes to country names
country_code_to_name = {
    'AF': 'Afghanistan', 'AX': 'Aland Islands', 'AL': 'Albania', 'DZ': 'Algeria', 'AS': 'American Samoa',
    'AD': 'Andorra', 'AO': 'Angola', 'AI': 'Anguilla', 'AQ': 'Antarctica', 'AG': 'Antigua and Barbuda',
    'AR': 'Argentina', 'AM': 'Armenia', 'AW': 'Aruba', 'AU': 'Australia', 'AT': 'Austria', 'AZ': 'Azerbaijan',
    'BS': 'Bahamas', 'BH': 'Bahrain', 'BD': 'Bangladesh', 'BB': 'Barbados', 'BY': 'Belarus', 'BE': 'Belgium',
    'BZ': 'Belize', 'BJ': 'Benin', 'BM': 'Bermuda', 'BT': 'Bhutan', 'BO': 'Bolivia', 'BQ': 'Bonaire, Sint Eustatius and Saba',
    'BA': 'Bosnia and Herzegovina', 'BW': 'Botswana', 'BV': 'Bouvet Island', 'BR': 'Brazil', 'IO': 'British Indian Ocean Territory',
    'BN': 'Brunei Darussalam', 'BG': 'Bulgaria', 'BF': 'Burkina Faso', 'BI': 'Burundi', 'CV': 'Cabo Verde',
    'KH': 'Cambodia', 'CM': 'Cameroon', 'CA': 'Canada', 'KY': 'Cayman Islands', 'CF': 'Central African Republic',
    'TD': 'Chad', 'CL': 'Chile', 'CN': 'China', 'CX': 'Christmas Island', 'CC': 'Cocos (Keeling) Islands',
    'CO': 'Colombia', 'KM': 'Comoros', 'CG': 'Congo', 'CD': 'Congo, The Democratic Republic of the',
    'CK': 'Cook Islands', 'CR': 'Costa Rica', 'CI': 'Cote d\'Ivoire', 'HR': 'Croatia', 'CU': 'Cuba', 'CW': 'Curacao',
    'CY': 'Cyprus', 'CZ': 'Czech Republic', 'DK': 'Denmark', 'DJ': 'Djibouti', 'DM': 'Dominica',
    'DO': 'Dominican Republic', 'EC': 'Ecuador', 'EG': 'Egypt', 'SV': 'El Salvador', 'GQ': 'Equatorial Guinea',
    'ER': 'Eritrea', 'EE': 'Estonia', 'ET': 'Ethiopia', 'FK': 'Falkland Islands (Malvinas)', 'FO': 'Faroe Islands',
    'FJ': 'Fiji', 'FI': 'Finland', 'FR': 'France', 'GF': 'French Guiana', 'PF': 'French Polynesia',
    'TF': 'French Southern Territories', 'GA': 'Gabon', 'GM': 'Gambia', 'GE': 'Georgia', 'DE': 'Germany',
    'GH': 'Ghana', 'GI': 'Gibraltar', 'GR': 'Greece', 'GL': 'Greenland', 'GD': 'Grenada', 'GP': 'Guadeloupe',
    'GU': 'Guam', 'GT': 'Guatemala', 'GG': 'Guernsey', 'GN': 'Guinea', 'GW': 'Guinea-Bissau', 'GY': 'Guyana',
    'HT': 'Haiti', 'HM': 'Heard Island and McDonald Islands', 'VA': 'Holy See (Vatican City State)', 'HN': 'Honduras',
    'HK': 'Hong Kong', 'HU': 'Hungary', 'IS': 'Iceland', 'IN': 'India', 'ID': 'Indonesia', 'IR': 'Iran',
    'IQ': 'Iraq', 'IE': 'Ireland', 'IM': 'Isle of Man', 'IL': 'Israel', 'IT': 'Italy', 'JM': 'Jamaica', 'JP': 'Japan',
    'JE': 'Jersey', 'JO': 'Jordan', 'KZ': 'Kazakhstan', 'KE': 'Kenya', 'KI': 'Kiribati', 'KP': 'North Korea',
    'KR': 'Korea, Republic of', 'KW': 'Kuwait', 'KG': 'Kyrgyzstan', 'LA': 'Lao People\'s Democratic Republic',
    'LV': 'Latvia', 'LB': 'Lebanon', 'LS': 'Lesotho', 'LR': 'Liberia', 'LY': 'Libya', 'LI': 'Liechtenstein',
    'LT': 'Lithuania', 'LU': 'Luxembourg', 'MO': 'Macao', 'MK': 'Macedonia, The Former Yugoslav Republic of',
    'MG': 'Madagascar', 'MW': 'Malawi', 'MY': 'Malaysia', 'MV': 'Maldives', 'ML': 'Mali', 'MT': 'Malta',
    'MH': 'Marshall Islands', 'MQ': 'Martinique', 'MR': 'Mauritania', 'MU': 'Mauritius', 'YT': 'Mayotte',
    'MX': 'Mexico', 'FM': 'Micronesia, Federated States of', 'MD': 'Moldova, Republic of', 'MC': 'Monaco',
    'MN': 'Mongolia', 'ME': 'Montenegro', 'MS': 'Montserrat', 'MA': 'Morocco', 'MZ': 'Mozambique',
    'MM': 'Myanmar', 'NA': 'Namibia', 'NR': 'Nauru', 'NP': 'Nepal', 'NL': 'Netherlands', 'NC': 'New Caledonia',
    'NZ': 'New Zealand', 'NI': 'Nicaragua', 'NE': 'Niger', 'NG': 'Nigeria', 'NU': 'Niue', 'NF': 'Norfolk Island',
    'MP': 'Northern Mariana Islands', 'NO': 'Norway', 'OM': 'Oman', 'PK': 'Pakistan', 'PW': 'Palau',
    'PS': 'Palestine, State of', 'PA': 'Panama', 'PG': 'Papua New Guinea', 'PY': 'Paraguay', 'PE': 'Peru',
    'PH': 'Philippines', 'PN': 'Pitcairn', 'PL': 'Poland', 'PT': 'Portugal', 'PR': 'Puerto Rico', 'QA': 'Qatar',
    'RE': 'Reunion', 'RO': 'Romania', 'RU': 'Russian Federation', 'RW': 'Rwanda', 'BL': 'Saint Barthelemy',
    'SH': 'Saint Helena, Ascension and Tristan da Cunha', 'KN': 'Saint Kitts and Nevis', 'LC': 'Saint Lucia',
    'MF': 'Saint Martin (French part)', 'PM': 'Saint Pierre and Miquelon', 'VC': 'Saint Vincent and the Grenadines',
    'WS': 'Samoa', 'SM': 'San Marino', 'ST': 'Sao Tome and Principe', 'SA': 'Saudi Arabia', 'SN': 'Senegal',
    'RS': 'Serbia', 'SC': 'Seychelles', 'SL': 'Sierra Leone', 'SG': 'Singapore', 'SX': 'Sint Maarten (Dutch part)',
    'SK': 'Slovakia', 'SI': 'Slovenia', 'SB': 'Solomon Islands', 'SO': 'Somalia', 'ZA': 'South Africa',
    'GS': 'South Georgia and the South Sandwich Islands', 'SS': 'South Sudan', 'ES': 'Spain', 'LK': 'Sri Lanka',
    'SD': 'Sudan', 'SR': 'Suriname', 'SJ': 'Svalbard and Jan Mayen', 'SZ': 'Swaziland', 'SE': 'Sweden',
    'CH': 'Switzerland', 'SY': 'Syria', 'TW': 'Taiwan', 'TJ': 'Tajikistan',
    'TZ': 'Tanzania', 'TH': 'Thailand', 'TL': 'Timor-Leste', 'TG': 'Togo', 'TK': 'Tokelau',
    'TO': 'Tonga', 'TT': 'Trinidad and Tobago', 'TN': 'Tunisia', 'TR': 'Turkey', 'TM': 'Turkmenistan',
    'TC': 'Turks and Caicos Islands', 'TV': 'Tuvalu', 'UG': 'Uganda', 'UA': 'Ukraine', 'AE': 'United Arab Emirates',
    'GB': 'United Kingdom', 'US': 'United States', 'UM': 'United States Minor Outlying Islands', 'UY': 'Uruguay',
    'UZ': 'Uzbekistan', 'VU': 'Vanuatu', 'VE': 'Venezuela, Bolivarian Republic of', 'VN': 'Viet Nam',
    'VG': 'Virgin Islands, British', 'VI': 'Virgin Islands, U.S.', 'WF': 'Wallis and Futuna', 'EH': 'Western Sahara',
    'YE': 'Yemen', 'ZM': 'Zambia', 'ZW': 'Zimbabwe'
}
```

```python
# Replace country codes with full names
df['Country'] = df['Country'].replace(country_code_to_name)
df.head()
```

| | time | place | magnitude | depth_km | longitude | latitude | type | tsunami | Country | Continental |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2015-01-01 05:01:10.640 | near the east coast of Honshu, Japan | 4.8 | 41.39 | 142.0405 | 38.8957 | earthquake | 0 | Japan | Asia |
| 1 | 2015-01-01 06:48:29.670 | 93 km N of Isangel, Vanuatu | 4.6 | 223.61 | 169.1795 | -18.7052 | earthquake | 0 | Vanuatu | Oceania |
| 2 | 2015-01-01 06:54:20.570 | central Mid-Atlantic Ridge | 4.7 | 10.00 | -31.7641 | 3.4769 | earthquake | 0 | Brazil | South America |
| 3 | 2015-01-01 07:12:44.230 | 120 km SSE of Kirakira, Solomon Islands | 4.6 | 26.24 | 162.4998 | -11.3818 | earthquake | 0 | Solomon Islands | Oceania |
| 4 | 2015-01-01 08:49:53.200 | 70 km W of F?r?z?b?d, Iran | 5.1 | 10.10 | 51.8580 | 28.7280 | earthquake | 0 | Iran | Asia |

```
!pip install pycountry_convert
import pycountry_convert as pc

# Function to map country name to continent
def get_continent(country_name):
    try:
        # Handle name discrepancies manually
        rename_dict = {
            "Russian Federation": "Russia",
            "Iran, Islamic Republic of": "Iran",
            "Venezuela, Bolivarian Republic of": "Venezuela",
            "Korea, Republic of": "South Korea",
            "Korea, Democratic People's Republic of": "North Korea",
            "Syrian Arab Republic": "Syria",
            "Taiwan, Province of China": "Taiwan",
            "Viet Nam": "Vietnam",
            "United States": "United States of America",
            "Micronesia, Federated States of": "Micronesia",
            "Tanzania, United Republic of": "Tanzania",
            "Macedonia, The Former Yugoslav Republic of": "North Macedonia",
            "Congo, The Democratic Republic of": "Democratic Republic of the Congo"
        }
        if country_name in rename_dict:
            country_name = rename_dict[country_name]
        country_code = pc.country_name_to_country_alpha2(country_name)
        continent_code = pc.country_alpha2_to_continent_code(country_code)
        continent_name = pc.convert_continent_code_to_continent_name(continent_code)
        return continent_name
    except:
        return "Unknown"
```

```
# Map countries to continents
continent_mapping = {country: get_continent(country) for country in countries}
continent_mapping_sorted = dict(sorted(continent_mapping.items(), key=lambda x: x[1]))
df['Continental'] = df['Country'].map(continent_mapping_sorted)
```

```
df.isnull().sum()
```

| | ø |
|---|---|
| time | 0 |
| place | 0 |
| magnitude | 0 |
| depth_km | 0 |
| longitude | 0 |
| latitude | 0 |
| type | 0 |
| tsunami | 0 |
| Country | 0 |
| Continental | 0 |

**dtype:** int64

```
df.drop_duplicates(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 72571 entries, 0 to 72582
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype         
---  ------       --------------  -----         
 0   time         72571 non-null  datetime64[ns]
 1   place        72571 non-null  object        
 2   magnitude    72571 non-null  float64       
 3   depth_km     72571 non-null  float64       
 4   longitude    72571 non-null  float64       
 5   latitude     72571 non-null  float64       
 6   type         72571 non-null  object        
 7   tsunami      72571 non-null  int64         
 8   Country      72571 non-null  object        
 9   Continental  72571 non-null  object        
dtypes: datetime64[ns](1), float64(4), int64(1), object(4)
memory usage: 6.1+ MB
```

## Descriptive

```
df.describe()
```

| | time | magnitude | depth_km | longitude | latitude | tsunami |
|---|---|---|---|---|---|---|
| **count** | 72571 | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 |
| **mean** | 2020-01-17 12:47:01.361239040 | 4.803313 | 63.131648 | 33.335054 | -1.808635 | 0.013835 |
| **min** | 2015-01-01 05:01:10.640000 | 4.500000 | 0.000000 | -179.999700 | -79.983700 | 0.000000 |
| **25%** | 2017-08-12 09:34:29.515000064 | 4.500000 | 10.000000 | -72.265150 | -22.231350 | 0.000000 |
| **50%** | 2020-02-10 05:22:56.635000064 | 4.700000 | 14.200000 | 92.482800 | -5.119700 | 0.000000 |
| **75%** | 2022-06-11 10:05:27.760000 | 4.900000 | 57.735000 | 141.492850 | 19.313400 | 0.000000 |
| **max** | 2024-12-30 23:56:29.977000 | 8.300000 | 683.360000 | 179.999300 | 87.386000 | 1.000000 |
| **std** | NaN | 0.371198 | 115.081476 | 123.129062 | 29.755335 | 0.116805 |

## Initial Findings

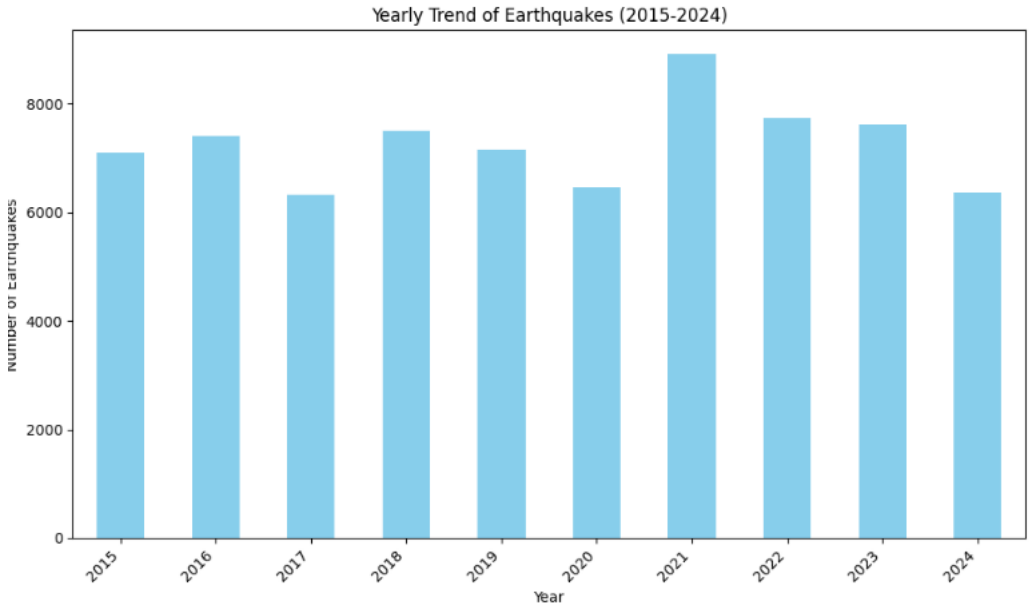
```
# Feature Distribution

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 10))

plt.subplot(2, 2, 1)
sns.histplot(df['longitude'], kde=True, bins=30)
plt.title('Distribution of Longitude')

plt.subplot(2, 2, 2)
sns.histplot(df['latitude'], kde=True, bins=30)
plt.title('Distribution of Latitude')

plt.subplot(2, 2, 3)
sns.histplot(df['depth_km'], kde=True, bins=30)
plt.title('Distribution of Depth (km)')

plt.subplot(2, 2, 4)
sns.histplot(df['magnitude'], kde=True, bins=30)
plt.title('Distribution of Magnitude')

plt.tight_layout()
plt.show()
```

```
# 5 highest earthquake

# Sort by 'Earthquake Magnitude' in descending order and select the top 5
top_5_earthquakes =df.sort_values(by='magnitude', ascending=False).head(5)

print("Top 5 Highest Magnitude Earthquakes:")
display(top_5_earthquakes)
```

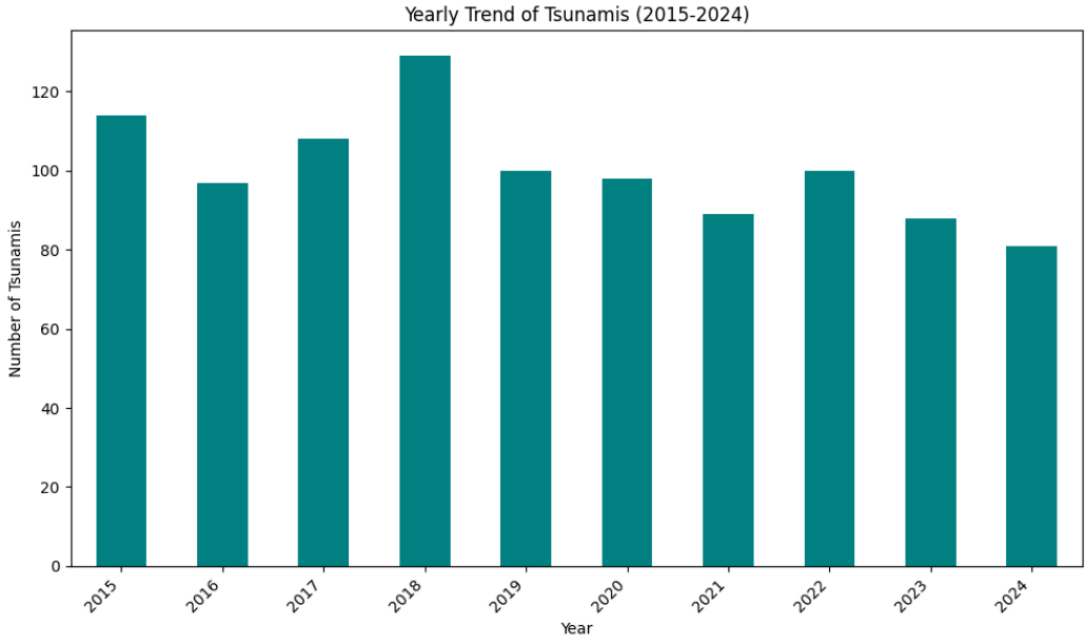
```
Top 5 Highest Magnitude Earthquakes:
```

| | time | place | magnitude | depth_km | longitude | latitude | type | tsunami | Country | Continental |
|---|---|---|---|---|---|---|---|---|---|---|
| **4845** | 2015-09-16 22:54:32.860 | 48 km W of Illapel, Chile | 8.3 | 22.44 | -71.6744 | -31.5729 | earthquake | 1 | Chile | South America |
| **46898** | 2021-07-29 06:15:49.188 | Alaska Peninsula | 8.2 | 35.00 | -157.8876 | 55.3635 | earthquake | 1 | United States | North America |
| **24950** | 2018-08-19 00:19:40.670 | 267 km E of Levuka, Fiji | 8.2 | 600.00 | -178.1530 | -18.1125 | earthquake | 1 | Wallis and Futuna | Oceania |
| **18557** | 2017-09-08 04:49:19.180 | near the coast of Chiapas, Mexico | 8.2 | 47.39 | -93.8993 | 15.0222 | earthquake | 1 | Mexico | North America |
| **47257** | 2021-08-12 18:35:17.231 | South Sandwich Islands region | 8.1 | 22.79 | -25.2637 | -58.3753 | earthquake | 0 | South Georgia and the South Sandwich Islands | South America |

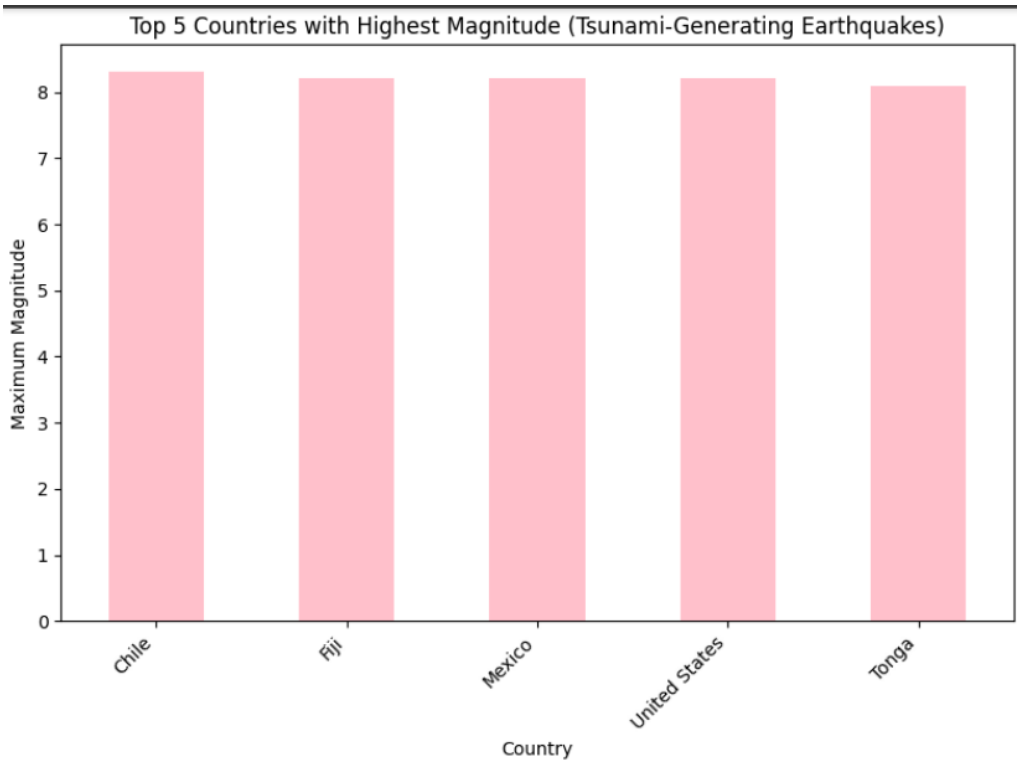
```
import matplotlib.pyplot as plt
# Plotting
plt.figure(figsize=(10, 6))
yearly_frequency.plot(kind='bar', color='skyblue')
plt.xlabel("Year")
plt.ylabel("Number of Earthquakes")
plt.title("Yearly Trend of Earthquakes (2015-2024)")
plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```



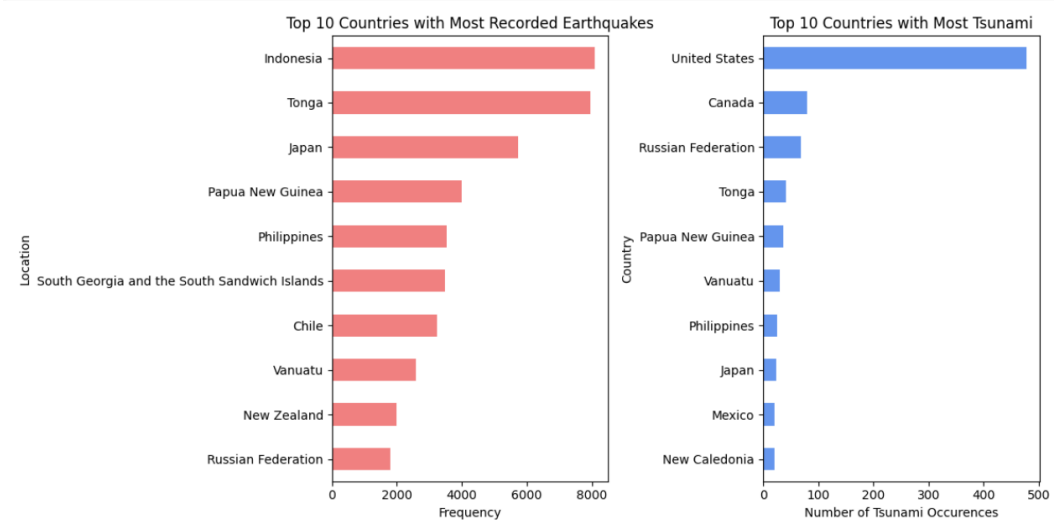

```python
import matplotlib.pyplot as plt

# Filter for rows where 'tsunami' is 1 (indicating a tsunami occurred)
tsunami_events = df[df['tsunami'] == 1]

# Group by year and count the number of tsunami events
yearly_tsunami_counts = tsunami_events['Year'].value_counts().sort_index()

# Create a bar chart
plt.figure(figsize=(10, 6))
yearly_tsunami_counts.plot(kind='bar', color='teal')
plt.xlabel("Year")
plt.ylabel("Number of Tsunamis")
plt.title("Yearly Trend of Tsunamis (2015-2024)")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



```python
# Filter only tsunami-generating earthquakes
tsunami_events = df[df['tsunami'] == 1]

# Get the maximum magnitude per country
max_mag_by_country = tsunami_events.groupby('Country')['magnitude'].max()

# Sort and get the top 5
top_5_countries = max_mag_by_country.sort_values(ascending=False).head(5)

# Plot
plt.figure(figsize=(8, 6))
top_5_countries.plot(kind='bar', color='pink')
plt.xlabel('Country')
plt.ylabel('Maximum Magnitude')
plt.title('Top 5 Countries with Highest Magnitude (Tsunami-Generating Earthquakes)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

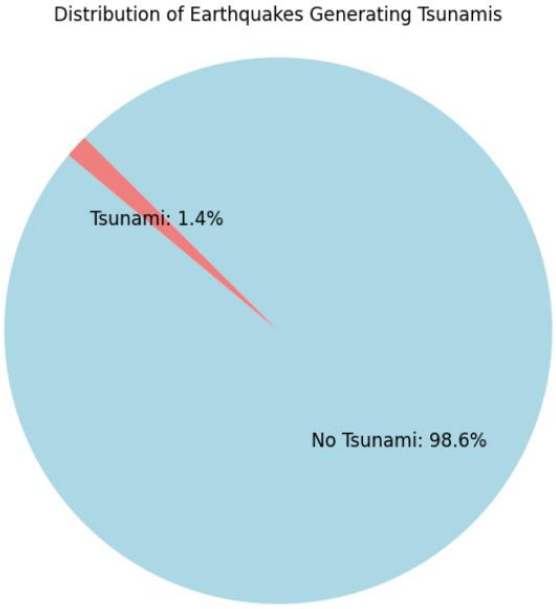


```
# Visualize
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
country_counts.plot(kind='barh', color='lightcoral')
plt.xlabel('Frequency')
plt.ylabel('Location')
plt.title('Top 10 Countries with Most Recorded Earthquakes')
plt.gca().invert_yaxis()

plt.subplot(1, 2, 2)
country_counts_tsunami.plot(kind='barh', color='cornflowerblue')
plt.xlabel('Number of Tsunami Occurences')
plt.ylabel('Country')
plt.title('Top 10 Countries with Most Tsunami')
plt.gca().invert_yaxis()

plt.tight_layout()
plt.show()
```

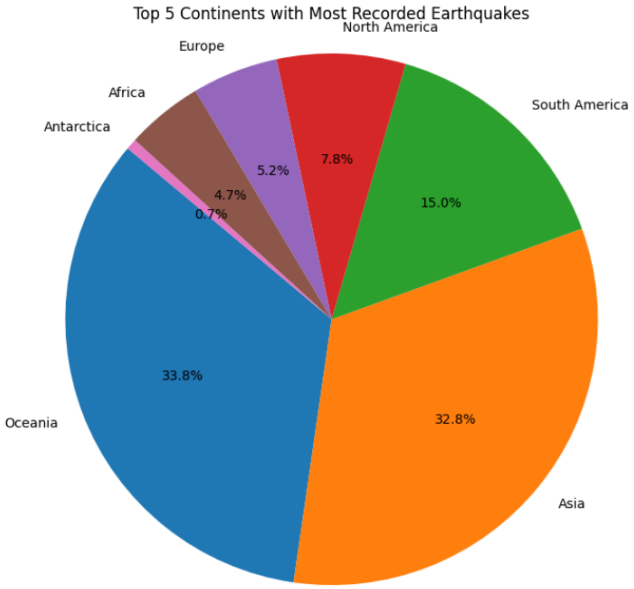


```python
import matplotlib.pyplot as plt

# Count frequency of earthquakes that generated tsunamis vs not
tsunami= df['tsunami'].value_counts()

print("\nFrequency of earthquakes that generated tsunami:")
print(tsunami)

# Map labels (0 = No tsunami, 1 = Tsunami generated)
labels = tsunami.index.map({0: "No Tsunami", 1: "Tsunami"})

# Prepare data
sizes = tsunami.values
colors = ['lightblue', 'lightcoral']

# Create the pie chart with labels inside
plt.figure(figsize=(8, 6))
wedges, texts, autotexts = plt.pie(
    sizes,
    labels=None,  # We'll add custom labels
    autopct='%1.1f%%',
    startangle=140,
    colors=colors,
    textprops={'fontsize': 12, 'color': 'black'}
)

# Customize each label inside the pie slice
for i, a in enumerate(autotexts):
    a.set_text(f"{labels[i]}: {a.get_text()}")  # Example: "Tsunami: 15.0%"

plt.title('Distribution of Earthquakes Generating Tsunamis')
plt.axis('equal')  # Ensures pie is a circle
plt.tight_layout()
plt.show()
```

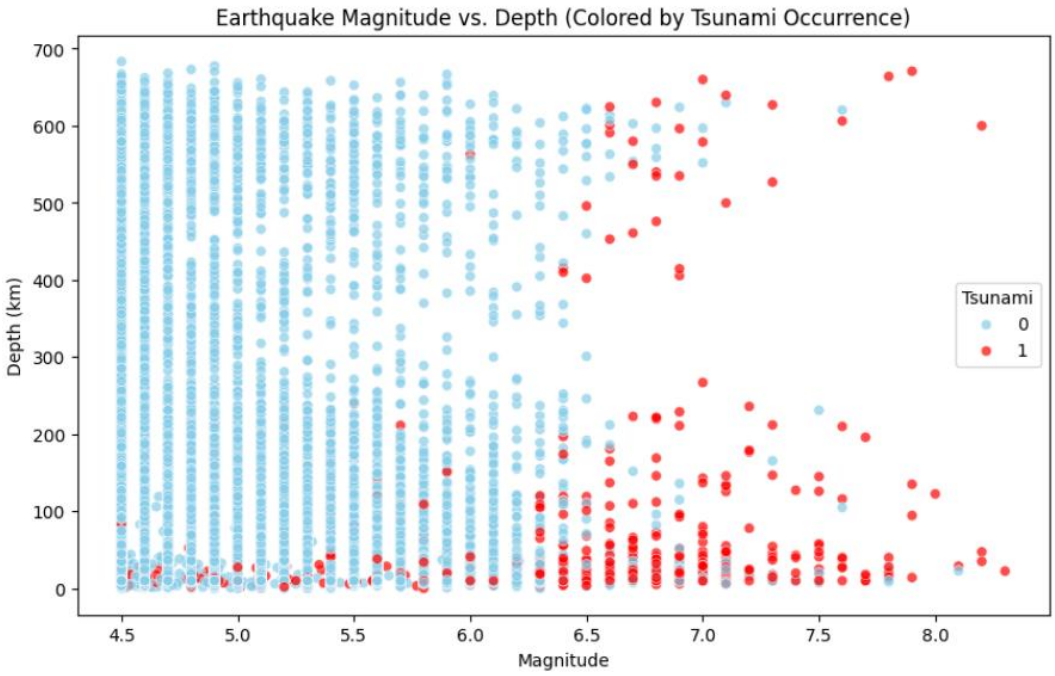
```
Frequency of earthquakes that generated tsunami:
tsunami
0    71579
1     1004
Name: count, dtype: int64
```



```
import matplotlib.pyplot as plt
import seaborn as sns

# Pie chart for continents with the highest number of earthquakes
earthquake_counts_by_continent = df['Continental'].value_counts().head(7)
plt.figure(figsize=(10, 8))
plt.pie(earthquake_counts_by_continent, labels=earthquake_counts_by_continent.index, autopct='%1.1f%%', startangle=140)
plt.title('Top 5 Continents with Most Recorded Earthquakes')
plt.axis('equal')
plt.show()
```




```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df,
    x='magnitude',
    y='depth_km',
    hue='tsunami',       # color by tsunami (0 or 1)
    palette={0: 'skyblue', 1: 'red'},
    alpha=0.7
)
plt.xlabel('Magnitude')
plt.ylabel('Depth (km)')
plt.title('Earthquake Magnitude vs. Depth (Colored by Tsunami Occurrence)')
plt.legend(title='Tsunami')
plt.show()
```



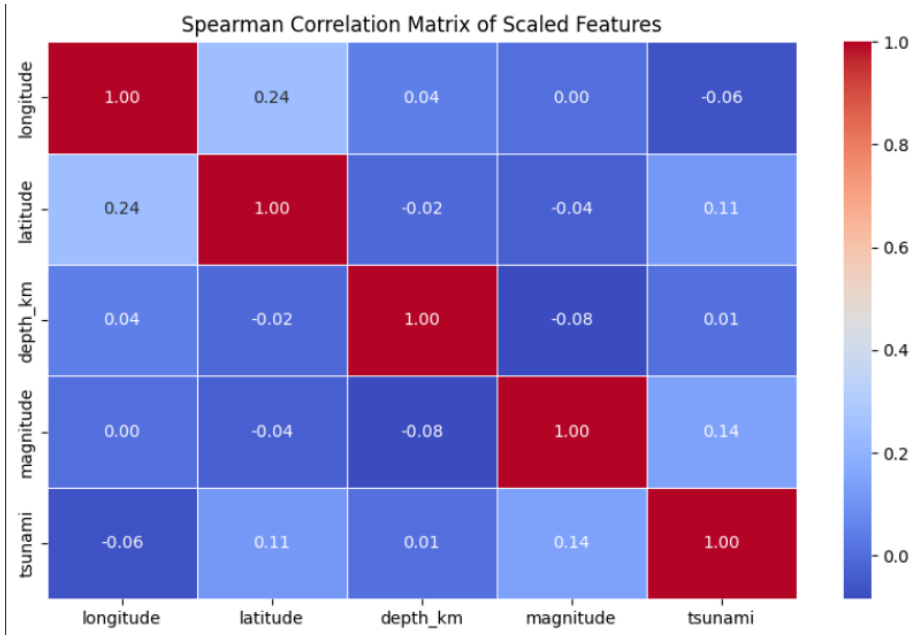

## Feature Selection

```
analysis = df[['longitude', 'latitude','depth_km', 'magnitude', 'tsunami']]
display(analysis.describe())
```

| | longitude | latitude | depth_km | magnitude | tsunami |
|---|---|---|---|---|---|
| **count** | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 | 72571.000000 |
| **mean** | 33.335054 | -1.808635 | 63.131648 | 4.803313 | 0.013835 |
| **std** | 123.129062 | 29.755335 | 115.081476 | 0.371198 | 0.116805 |
| **min** | -179.999700 | -79.983700 | 0.000000 | 4.500000 | 0.000000 |
| **25%** | -72.265150 | -22.231350 | 10.000000 | 4.500000 | 0.000000 |
| **50%** | 92.482800 | -5.119700 | 14.200000 | 4.700000 | 0.000000 |
| **75%** | 141.492850 | 19.313400 | 57.735000 | 4.900000 | 0.000000 |
| **max** | 179.999300 | 87.386000 | 683.360000 | 8.300000 | 1.000000 |

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the Spearman correlation matrix
correlation_matrix = analysis.corr(method='spearman')

# Print the correlation matrix
print("Spearman Correlation Matrix:")
print(correlation_matrix)

# Optional: Visualize the correlation matrix using a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Spearman Correlation Matrix of Scaled Features')
plt.show()
```

```
Spearman Correlation Matrix:
           longitude  latitude  depth_km  magnitude   tsunami
longitude   1.000000  0.239795  0.044346   0.001038 -0.063586
latitude    0.239795  1.000000 -0.015054  -0.036528  0.112511
depth_km    0.044346 -0.015054  1.000000  -0.084707  0.006108
magnitude   0.001038 -0.036528 -0.084707   1.000000  0.140068
tsunami    -0.063586  0.112511  0.006108   0.140068  1.000000
```



```
threshold = 0.1
selected_features = correlation_matrix[abs(correlation_matrix) >= threshold].index.tolist()

# Remove the target from the list
selected_features.remove('tsunami')

print("Selected features based on correlation threshold:")
print(selected_features)
```

```
Selected features based on correlation threshold:
['longitude', 'latitude', 'depth_km', 'magnitude']
```

```
analysis['tsunami']
```

| | tsunami |
|---|---|
| **0** | 0 |
| **1** | 0 |
| **2** | 0 |
| **3** | 0 |
| **4** | 0 |
| **...** | ... |
| **72578** | 0 |
| **72579** | 0 |
| **72580** | 0 |
| **72581** | 0 |
| **72582** | 0 |

72571 rows × 1 columns

## Split and Train

```
X = analysis[selected_features]
y = analysis['tsunami']
```

```
from sklearn.model_selection 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: ', X_train.shape)
print('y-train: ', y_train.shape)
print()
print('X-test: ', X_test.shape)
print('y-test: ', y_test.shape)
```

```
X-train:  (58056, 4)
y-train:  (58056,)

X-test:  (14515, 4)
y-test:  (14515,)
```

```
print("Number of tsunami in training set:", y_train.sum())
print("Number of tsunami in testing set:", y_test.sum())
```

```
Number of tsunami in training set: 800
Number of tsunami in testing set: 204
```

## XGBOOST BASELINE

+ Code + Text

```
import xgboost as xgb
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate an XGBClassifier model
xgb_model = xgb.XGBClassifier(objective='binary:logistic',
                              use_label_encoder=False,
                              eval_metric='logloss',
                              random_state=42)

# Train the XGBoost model on the original training data
xgb_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb = xgb_model.predict(X_test)
y_pred_proba_xgb = xgb_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (XGBoost on Original Data):\n", classification_report(y_test, y_pred_xgb))
print("AUC Score (XGBoost on Original Data):", roc_auc_score(y_test, y_pred_proba_xgb))
```

```
/usr/local/lib/python3.12/dist-packages/xgboost/training.py:183: UserWarning: [07:04:03] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
Classification Report (XGBoost on Original Data):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     14311
           1       0.71      0.74      0.72       204

    accuracy                           0.99     14515
   macro avg       0.85      0.87      0.86     14515
weighted avg       0.99      0.99      0.99     14515
```

## NB BASELINE

+ Code

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate and train the Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# Make predictions
y_pred_nb = nb_model.predict(X_test)
y_pred_proba_nb = nb_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (Naive Bayes on Original Data):\n", classification_report(y_test, y_pred_nb))
print("AUC Score (Naive Bayes on Original Data):", roc_auc_score(y_test, y_pred_proba_nb))
```

```
Classification Report (Naive Bayes on Original Data):
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     14311
           1       0.39      0.52      0.45       204

    accuracy                           0.98     14515
   macro avg       0.69      0.75      0.72     14515
weighted avg       0.98      0.98      0.98     14515

AUC Score (Naive Bayes on Original Data): 0.9752237069798222
```

```
# Calculate the confusion matrix
cm_nb_original
```

```
array([[14146,   165],
       [   98,   106]])
```

```
# Calculate the confusion matrix for XGBoost on Original Data
cm_xgb_original
```

```
array([[14249,    62],
       [   53,   151]])
```

SMOTE XGBOOST

```
import xgboost as xgb
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate an XGBClassifier model
xgb_smote_model = xgb.XGBClassifier(objective='binary:logistic',
                                    use_label_encoder=False,
                                    eval_metric='logloss',
                                    random_state=42)

# Train the XGBoost model using the SMOTE-augmented training data
xgb_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_xgb_smote = xgb_smote_model.predict(X_test)
y_pred_proba_xgb_smote = xgb_smote_model.predict_proba(X_test)[:, 1]
```

```
# Print the classification report
print("Classification Report (XGBoost on SMOTE-augmented Data):\n", classification_report(y_test, y_pred_xgb_smote))

# Print the AUC score
print("AUC Score (XGBoost on SMOTE-augmented Data):", roc_auc_score(y_test, y_pred_proba_xgb_smote))
```

```
Classification Report (XGBoost on SMOTE-augmented Data):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     14311
           1       0.58      0.91      0.70       204

    accuracy                           0.99     14515
   macro avg       0.79      0.95      0.85     14515
weighted avg       0.99      0.99      0.99     14515

AUC Score (XGBoost on SMOTE-augmented Data): 0.9956027586074608
```

SMOTE NB

```
from sklearn.naive_bayes import GaussianNB

# Instantiate a GaussianNB model
nb_smote_model = GaussianNB()

# Train the Naive Bayes model using the SMOTE-augmented training data
nb_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_nb_smote = nb_smote_model.predict(X_test)
y_pred_proba_nb_smote = nb_smote_model.predict_proba(X_test)[:, 1]

# Print the classification report
print("Classification Report (Naive Bayes on SMOTE-augmented Data):\n", classification_report(y_test, y_pred_nb_smote))

# Print the AUC score
print("AUC Score (Naive Bayes on SMOTE-augmented Data):", roc_auc_score(y_test, y_pred_proba_nb_smote))
```

```
Classification Report (Naive Bayes on SMOTE-augmented Data):
              precision    recall  f1-score   support

           0       1.00      0.91      0.95     14311
           1       0.13      0.94      0.23       204

    accuracy                           0.91     14515
   macro avg       0.56      0.93      0.59     14515
weighted avg       0.99      0.91      0.94     14515

AUC Score (Naive Bayes on SMOTE-augmented Data): 0.9749368372881961
```

```
# confusion matrix for Naive Bayes on SMOTE-augmented Data
cm_nb_smote
```

```
array([[13009,  1302],
       [   12,   192]])
```

```
#  confusion matrix for XGBoost on SMOTE-augmented Data
cm_xgb_smote
```

```
array([[14175,   136],
       [   19,   185]])
```

## CLASS WEIGHT XGboost

+ Code + Text

```
scale_pos_weight = (y_train == 0).sum() / (y_train == 1).sum()
print("Calculated scale_pos_weight:", scale_pos_weight)
```

```
Calculated scale_pos_weight: 71.57
```

```
import xgboost as xgb
from sklearn.metrics import classification_report, roc_auc_score

# Instantiate and train the XGBoost model with scale_pos_weight
scale_pos_weight = (y_train == 0).sum() / (y_train == 1).sum()

# Instantiate an XGBClassifier model with scale_pos_weight
xgb_scaled_model = xgb.XGBClassifier(objective='binary:logistic',
                                     use_label_encoder=False,
                                     eval_metric='logloss',
                                     random_state=42,
                                     scale_pos_weight=scale_pos_weight
)

# Train the XGBoost model on the original training data
xgb_scaled_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb_scaled = xgb_scaled_model.predict(X_test)
y_pred_proba_xgb_scaled = xgb_scaled_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (XGBoost on Original Data with scale_pos_weight):\n", classification_report(y_test, y_pred_xgb_scaled))
print("AUC Score (XGBoost on Original Data with scale_pos_weight):", roc_auc_score(y_test, y_pred_proba_xgb_scaled))
```

```
/usr/local/lib/python3.12/dist-packages/xgboost/training.py:183: UserWarning: [07:04:14] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
Classification Report (XGBoost on Original Data with scale_pos_weight):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     14311
           1       0.58      0.88      0.70       204

    accuracy                           0.99     14515
   macro avg       0.79      0.94      0.85     14515
weighted avg       0.99      0.99      0.99     14515

AUC Score (XGBoost on Original Data with scale_pos_weight): 0.9957538147674695
```

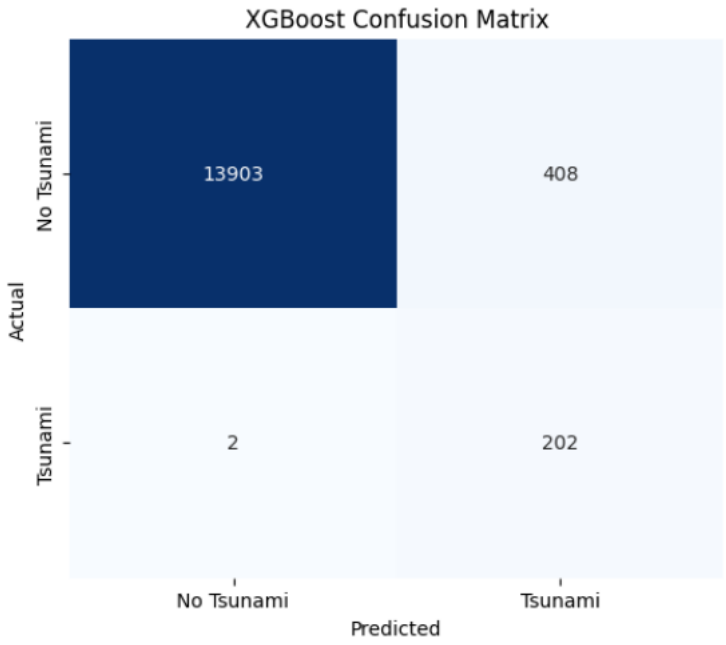
```
#  confusion matrix for XGBoost on Original Data with scale_pos_weight
cm_xgb_scaled
```

```
array([[14178,   133],
       [   24,   180]])
```

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
import xgboost as xgb # Import xgboost

# Define the parameter grid for XGBoost
param_grid = {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1],
    'max_depth': [3, 5],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}

# Instantiate StratifiedKFold
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Instantiate GridSearchCV
grid_search_xgb_scaled = GridSearchCV(estimator=xgb.XGBClassifier(objective='binary:logistic',
                                                                  use_label_encoder=False,
                                                                  eval_metric='logloss',
                                                                  random_state=42,
                                                                  scale_pos_weight=scale_pos_weight),
                                      param_grid=param_grid,
                                      cv=skf,
                                      scoring='recall',
                                      verbose=1,
                                      n_jobs=-1)

# Fit GridSearchCV to the original training data
grid_search_xgb_scaled.fit(X_train, y_train)

# Instantiate a new XGBoost model with the best hyperparameters found from grid search
xgb_tuned_scaled_model = xgb.XGBClassifier(objective='binary:logistic',
                                           use_label_encoder=False,
                                           eval_metric='logloss',
                                           random_state=42,
                                           scale_pos_weight=scale_pos_weight,
                                           **grid_search_xgb_scaled.best_params_)

# Train the new XGBoost model on the original training data
xgb_tuned_scaled_model.fit(X_train, y_train)

# Make predictions on the original test set
y_pred_tuned_xgb_scaled = xgb_tuned_scaled_model.predict(X_test)
y_pred_proba_tuned_xgb_scaled = xgb_tuned_scaled_model.predict_proba(X_test)[:, 1]
```

```
# Instantiate a new XGBoost model with the best hyperparameters found from grid search and scale_pos_weight
xgb_tuned_scaled_model = xgb.XGBClassifier(objective='binary:logistic',
                                           use_label_encoder=False,
                                           eval_metric='logloss',
                                           random_state=42,
                                           scale_pos_weight=scale_pos_weight,
                                           **grid_search_xgb_scaled.best_params_)

# Train the new XGBoost model on the entire training dataset
xgb_tuned_scaled_model.fit(X_train, y_train)

# Make predictions on the original test set
y_pred_tuned_xgb_scaled = xgb_tuned_scaled_model.predict(X_test)
y_pred_proba_tuned_xgb_scaled = xgb_tuned_scaled_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (Tuned XGBoost Model with scale_pos_weight):\n", classification_report(y_test, y_pred_tuned_xgb_scaled))
print("AUC Score (Tuned XGBoost Model with scale_pos_weight):", roc_auc_score(y_test, y_pred_proba_tuned_xgb_scaled))

# Calculate and plot the confusion matrix
cm_tuned_xgb_scaled = confusion_matrix(y_test, y_pred_tuned_xgb_scaled)
disp_tuned_xgb_scaled = ConfusionMatrixDisplay(confusion_matrix=cm_tuned_xgb_scaled, display_labels=y_test.unique())

plt.figure(figsize=(6, 5))
sns.heatmap(cm_tuned_xgb_scaled, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['No Tsunami', 'Tsunami'],
            yticklabels=['No Tsunami', 'Tsunami'])
plt.title('XGBoost Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
Classification Report (Tuned XGBoost Model with scale_pos_weight):
              precision    recall  f1-score   support

           0       1.00      0.97      0.99     14311
           1       0.33      0.99      0.50       204

    accuracy                           0.97     14515
   macro avg       0.67      0.98      0.74     14515
weighted avg       0.99      0.97      0.98     14515

AUC Score (Tuned XGBoost Model with scale_pos_weight): 0.99633115072596
Text(45.72222222222221, 0.5, 'Actual')
```



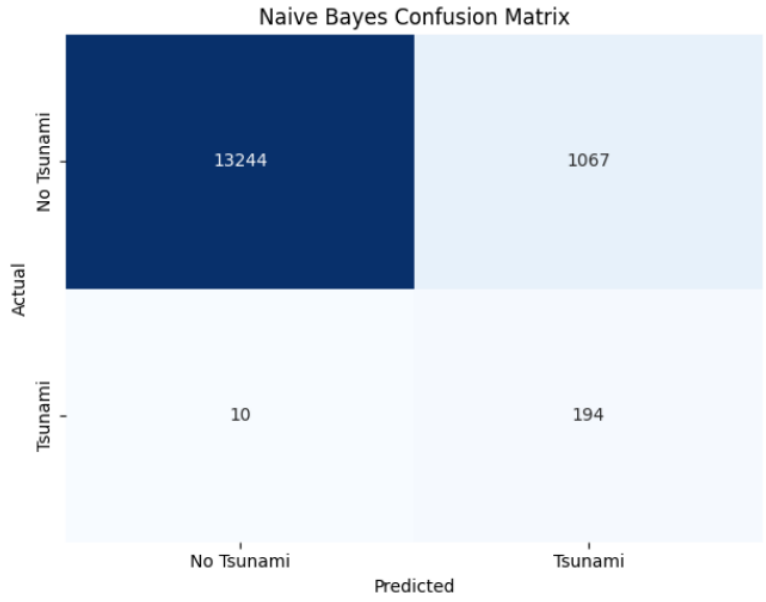

```python
# Instantiate a new GaussianNB model with the best hyperparameter found from grid search on SMOTE data
nb_tuned_smote_model = GaussianNB(**grid_search_nb_smote.best_params_)

# Train the new Naive Bayes model on the entire SMOTE-augmented training dataset
nb_tuned_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_tuned_nb_smote = nb_tuned_smote_model.predict(X_test)
y_pred_proba_tuned_nb_smote = nb_tuned_smote_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (Tuned Naive Bayes Model on SMOTE-augmented Data):\n", classification_report(y_test, y_pred_tuned_nb_smote))
print("AUC Score (Tuned Naive Bayes Model on SMOTE-augmented Data):", roc_auc_score(y_test, y_pred_proba_tuned_nb_smote))

# Calculate and plot the confusion matrix
cm_tuned_nb_smote = confusion_matrix(y_test, y_pred_tuned_nb_smote)
disp_tuned_nb_smote = ConfusionMatrixDisplay(confusion_matrix=cm_tuned_nb_smote, display_labels=y_test.unique())

# Visualize confusion matrices
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.heatmap(cm_tuned_nb_smote, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['No Tsunami', 'Tsunami'],
            yticklabels=['No Tsunami', 'Tsunami'])
plt.title('Naive Bayes Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.tight_layout()
plt.show()
```

```python
import numpy as np
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import GridSearchCV, StratifiedKFold

# Define the parameter grid for Naive Bayes
param_grid_nb = {
    'var_smoothing': np.logspace(0, -9, 10)
}

# Instantiate a GaussianNB model
nb_tuned_smote_model = GaussianNB()

# Instantiate StratifiedKFold
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Instantiate GridSearchCV
grid_search_nb_smote = GridSearchCV(estimator=nb_tuned_smote_model,
                                    param_grid=param_grid_nb, cv=skf, scoring='recall',
                                    verbose=1, n_jobs=-1)

# Fit GridSearchCV to the SMOTE-augmented training data
grid_search_nb_smote.fit(X_smote, y_smote)

# Print the best parameters and best score
print("Best parameters found for Naive Bayes on SMOTE data: ", grid_search_nb_smote.best_params_)
print("Best cross-validation AUC score for Naive Bayes on SMOTE data: ", grid_search_nb_smote.best_score_)
```

```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
Best parameters found for Naive Bayes on SMOTE data:  {'var_smoothing': np.float64(1e-05)}
Best cross-validation AUC score for Naive Bayes on SMOTE data:  0.9593055441584927
```

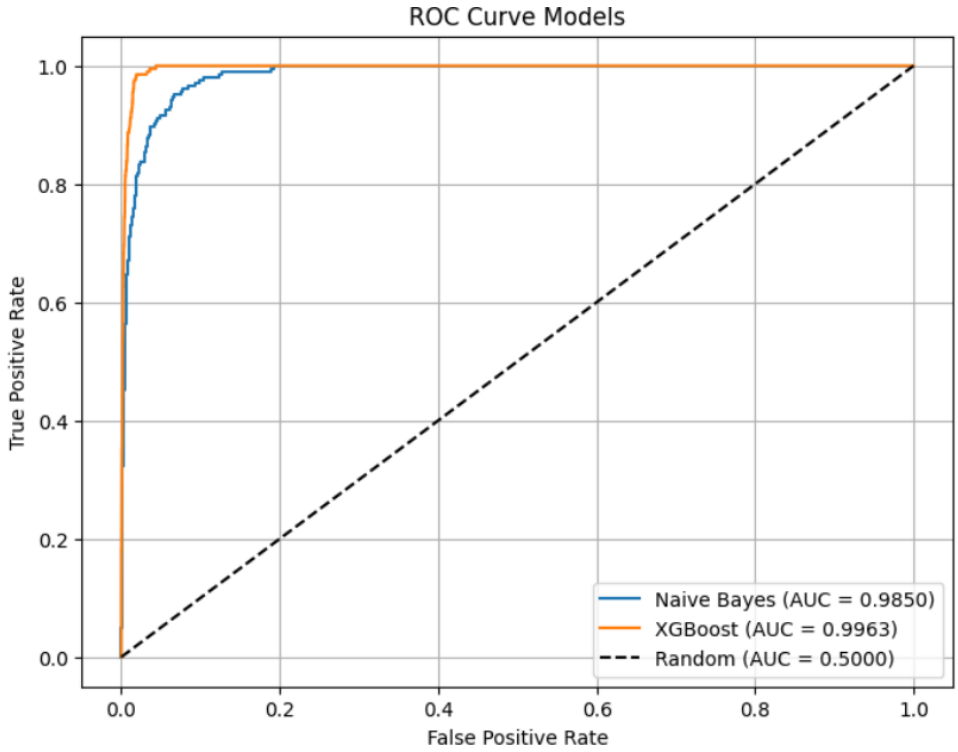
```python
# Instantiate a new GaussianNB model with the best hyperparameter found from grid search on SMOTE data
nb_tuned_smote_model = GaussianNB(**grid_search_nb_smote.best_params_)

# Train the new Naive Bayes model on the entire SMOTE-augmented training dataset
nb_tuned_smote_model.fit(X_smote, y_smote)

# Make predictions on the original test set
y_pred_tuned_nb_smote = nb_tuned_smote_model.predict(X_test)
y_pred_proba_tuned_nb_smote = nb_tuned_smote_model.predict_proba(X_test)[:, 1]

# Evaluate the model
print("Classification Report (Tuned Naive Bayes Model on SMOTE-augmented Data):\n", classification_report(y_test, y_pred_tuned_nb_smote))
print("AUC Score (Tuned Naive Bayes Model on SMOTE-augmented Data):", roc_auc_score(y_test, y_pred_proba_tuned_nb_smote))

# Calculate and plot the confusion matrix
cm_tuned_nb_smote = confusion_matrix(y_test, y_pred_tuned_nb_smote)
disp_tuned_nb_smote = ConfusionMatrixDisplay(confusion_matrix=cm_tuned_nb_smote, display_labels=y_test.unique())

# Visualize confusion matrices
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.heatmap(cm_tuned_nb_smote, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['No Tsunami', 'Tsunami'],
            yticklabels=['No Tsunami', 'Tsunami'])
plt.title('Naive Bayes Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.tight_layout()
plt.show()
```

```
Classification Report (Tuned Naive Bayes Model on SMOTE-augmented Data):
              precision    recall  f1-score   support

           0       1.00      0.93      0.96     14311
           1       0.15      0.95      0.26       204

    accuracy                           0.93     14515
   macro avg       0.58      0.94      0.61     14515
weighted avg       0.99      0.93      0.95     14515

AUC Score (Tuned Naive Bayes Model on SMOTE-augmented Data): 0.9850320814511256
```



## ROC AND AUC

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))

# Tuned Naive Bayes on SMOTE Data
fpr_nb_smote, tpr_nb_smote, _ = roc_curve(y_test, y_pred_proba_tuned_nb_smote)
auc_nb_smote = auc(fpr_nb_smote, tpr_nb_smote)
plt.plot(fpr_nb_smote, tpr_nb_smote, label=f'Naive Bayes (AUC = {auc_nb_smote:.4f})')


# Tuned XGBoost on Original Data with scale_pos_weight
fpr_xgb_scaled, tpr_xgb_scaled, _ = roc_curve(y_test, y_pred_proba_tuned_xgb_scaled)
auc_xgb_scaled = auc(fpr_xgb_scaled, tpr_xgb_scaled)
plt.plot(fpr_xgb_scaled, tpr_xgb_scaled, label=f'XGBoost (AUC = {auc_xgb_scaled:.4f})')

plt.plot([0, 1], [0, 1], 'k--', label='Random (AUC = 0.5000)')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Models')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```



# COMPARATIVE PERFORMANCE

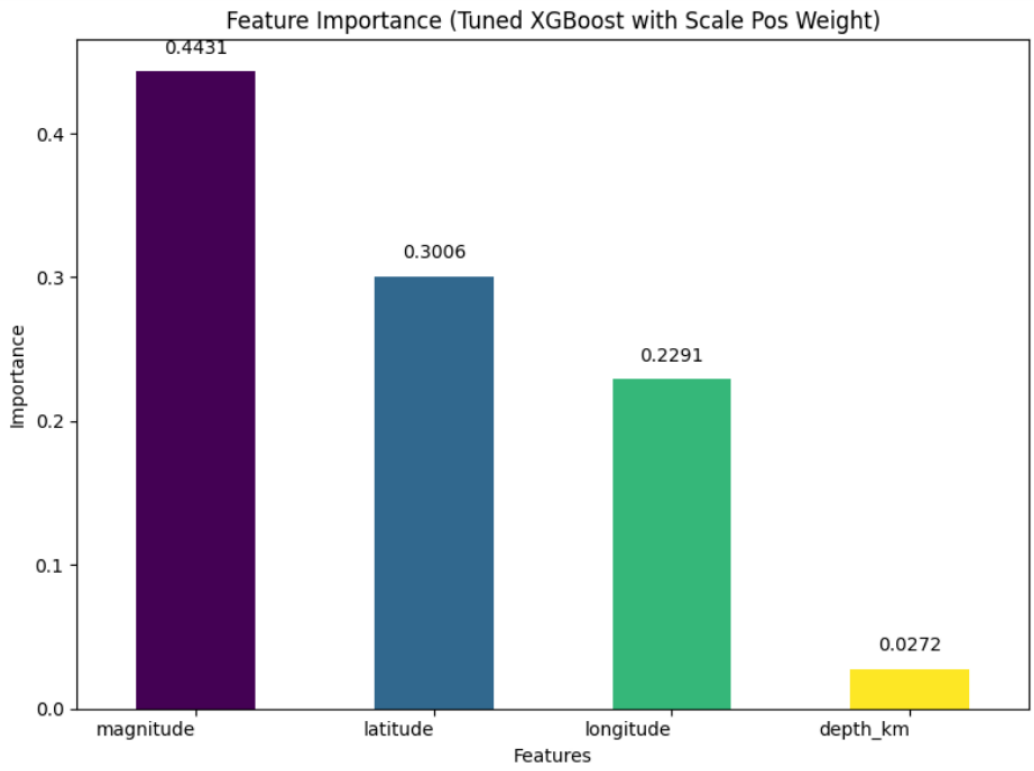
```
# Collect metrics for each model
all_metrics_list = []

# Function to get accuracy for training and testing
def get_accuracy(model, X_train, y_train, X_test, y_test):
    train_accuracy = model.score(X_train, y_train)
    test_accuracy = model.score(X_test, y_test)
    return train_accuracy, test_accuracy


# Tuned XGBoost on Original Data with scale_pos_weight
train_acc_xgb_scaled_tuned, test_acc_xgb_scaled_tuned = get_accuracy(xgb_tuned_scaled_model,
                                                                     X_train, y_train,
                                                                     X_test, y_test)
report_xgb_scaled_tuned = classification_report(y_test, y_pred_tuned_xgb_scaled, output_dict=True)
all_metrics_list.append({
    'Model': 'XGBoost',
    'Dataset/Method': 'Scale Pos Weight (Tuned)',
    'Accuracy (Training)': round(train_acc_xgb_scaled_tuned, 5),
    'Accuracy (Testing)': round(test_acc_xgb_scaled_tuned, 5),
    'Precision (Class 0)': round(report_xgb_scaled_tuned['0']['precision'], 5),
    'Recall (Class 0)': round(report_xgb_scaled_tuned['0']['recall'], 5),
    'F1-Score (Class 0)': round(report_xgb_scaled_tuned['0']['f1-score'], 5),
    'Precision (Class 1)': round(report_xgb_scaled_tuned['1']['precision'], 5),
    'Recall (Class 1)': round(report_xgb_scaled_tuned['1']['recall'], 5),
    'F1-Score (Class 1)': round(report_xgb_scaled_tuned['1']['f1-score'], 5),
    'AUC': round(roc_auc_score(y_test, y_pred_proba_tuned_xgb_scaled), 5)
})
```

```
# Tuned Naive Bayes on SMOTE Data
train_acc_nb_smote_tuned, test_acc_nb_smote_tuned = get_accuracy(nb_tuned_smote_model,
                                                                 X_smote, y_smote, X_test, y_test)

report_nb_smote_tuned = classification_report(y_test, y_pred_tuned_nb_smote, output_dict=True)
all_metrics_list.append({
    'Model': 'Naive Bayes',
    'Dataset/Method': 'SMOTE (Tuned)',
    'Accuracy (Training)': round(train_acc_nb_smote_tuned, 5),
    'Accuracy (Testing)': round(test_acc_nb_smote_tuned, 5),
    'Precision (Class 0)': round(report_nb_smote_tuned['0']['precision'], 5),
    'Recall (Class 0)': round(report_nb_smote_tuned['0']['recall'], 5),
    'F1-Score (Class 0)': round(report_nb_smote_tuned['0']['f1-score'], 5),
    'Precision (Class 1)': round(report_nb_smote_tuned['1']['precision'], 5),
    'Recall (Class 1)': round(report_nb_smote_tuned['1']['recall'], 5),
    'F1-Score (Class 1)': round(report_nb_smote_tuned['1']['f1-score'], 5),
    'AUC': round(roc_auc_score(y_test, y_pred_proba_tuned_nb_smote), 5)
})

# Create the DataFrame
all_performance_df = pd.DataFrame(all_metrics_list)

# Display the DataFrame
display(all_performance_df)
```

| | Model | Dataset/Method | Accuracy (Training) | Accuracy (Testing) | Precision (Class 0) | Recall (Class 0) | F1-Score (Class 0) | Precision (Class 1) | Recall (Class 1) | F1-Score (Class 1) | AUC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | XGBoost | Scale Pos Weight (Tuned) | 0.97380 | 0.97175 | 0.99986 | 0.97149 | 0.98547 | 0.33115 | 0.99020 | 0.49631 | 0.99633 |
| 1 | Naive Bayes | SMOTE (Tuned) | 0.94223 | 0.92580 | 0.99925 | 0.92544 | 0.96093 | 0.15385 | 0.95098 | 0.26485 | 0.98503 |

## Feature Importance

```
# Get feature importances from the best Random Forest model
feature_importances = best_xgb_model.feature_importances_

# Get the names of the features
feature_names = X_train.columns

# Create a DataFrame for easier plotting
importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': feature_importances})

# Sort features by importance
importance_df = importance_df.sort_values(by='Importance', ascending=False)

# Create a bar plot with different colors
plt.figure(figsize=(8, 6))
bars = plt.bar(importance_df['Feature'], importance_df['Importance'],
               color=plt.cm.tab20.colors[:len(importance_df)])  # unique colors

# Add value labels above each bar
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height,
             f'{height:.3f}', ha='center', va='bottom', fontsize=9)

# Customize plot
plt.ylabel('Importance')
plt.xlabel('Feature')
plt.title('Feature Importances')
plt.xticks(rotation=0, ha='right')
plt.tight_layout()
plt.show()
```



SAVE FOR POWER BI

```
# Assuming df and pred_df are available from previous steps

# Reset the index of both dataframes to ensure proper alignment
df_reset = df.reset_index(drop=True)
pred_df_reset = pred_df.reset_index(drop=True)


# Combine the original data with the prediction results
df_with_predictions = pd.concat([df_reset, pred_df_reset], axis=1)


# Display the first few rows of the combined DataFrame
display(df_with_predictions.head())

# Optional: Save the combined DataFrame to a new CSV
combined_filename = 'earthquakes_with_predictions.csv'
df_with_predictions.to_csv(combined_filename, index=False)
print(f"\nCombined data with predictions saved to {combined_filename}")
files.download(combined_filename)
```

| | time | place | magnitude | depth_km | longitude | latitude | type | tsunami | Country | Continental | Actual | Naive Bayes Tuned Prediction | XGBoost Tuned Prediction |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2015-01-01 05:01:10.640 | near the east coast of Honshu, Japan | 4.8 | 41.39 | 142.0405 | 38.8957 | earthquake | 0 | Japan | Asia | 0.0 | 0.0 | 0.0 |
| 1 | 2015-01-01 06:48:29.670 | 93 km N of Isangel, Vanuatu | 4.6 | 223.61 | 169.1795 | -18.7052 | earthquake | 0 | Vanuatu | Oceania | 0.0 | 0.0 | 0.0 |
| 2 | 2015-01-01 06:54:20.570 | central Mid-Atlantic Ridge | 4.7 | 10.00 | -31.7641 | 3.4769 | earthquake | 0 | Brazil | South America | 0.0 | 0.0 | 0.0 |
| 3 | 2015-01-01 07:12:44.230 | 120 km SSE of Kirakira, Solomon Islands | 4.6 | 26.24 | 162.4998 | -11.3818 | earthquake | 0 | Solomon Islands | Oceania | 0.0 | 0.0 | 0.0 |
| 4 | 2015-01-01 08:49:53.200 | 70 km W of F?r?z?b?d, Iran | 5.1 | 10.10 | 51.8580 | 28.7280 | earthquake | 0 | Iran | Asia | 0.0 | 0.0 | 1.0 |

```
Combined data with predictions saved to earthquakes_with_predictions.csv
```