Analysis of Trends in Earthquake Dataset from 1995-2023 and Tsunami Prediction

Aniket Kulkarni
College of Science and Engineering
University of Minnesota
Minneapolis, Minnesota
kulka389@umn.edu

Evina Novi Thekkudan
College of Science and Engineering
University of Minnesota
Minneapolis, Minnesota
thekk008@umn.edu

Mihir Ashok Momaya
College of Science and Engineering
University of Minnesota
Minneapolis, Minnesota
momay004@umn.edu

Surabhi Sunil
College of Science and Engineering
University of Minnesota
Minneapolis, Minnesota
sunil022@umn.edu

***Abstract*—The project presents a comprehensive analysis of earthquake activity and their relationship to tsunamis. The analysis is based on a broad 1995-2023 earthquake data set. The first phase of the project involved intensive data analysis, revealing an intriguing relationship between seismic activity and tsunamis. This finding prompted further investigation into patterns and trends in the data. Entity-Relationship (ER) diagrams and relationship schema was developed to facilitate efficient data handling and extraction. The data was then stored in a Microsoft SQL Server database, and SQL queries were used to extract valuable information. The project also delved into feature engineering using Python, which played a key role in preparing the data for predictive modeling. Using algorithms like Logistic Regression, Decision Tree, Naive Bayes, Random Forest Classifier and XGB five different predictive models were developed and tested to forecast the likelihood of tsunamis based on seismic activity. The best predictive model was then analyzed. The findings of this work provide valuable insights into the relationship between earthquakes and tsunamis, and demonstrate the potential of data analysis and predictive modeling in disaster prediction and preparedness.**

***Index Terms*—earthquakes, tsunami, feature engineering, prediction, relational schema**

I. Introduction

This topic deals with the analysis of earthquakes from 1995 - 2023 and the tsunamis which were caused due to them. The motivation for this project was to predict tsunamis which can be caused due to earthquakes. We also aim to analyze trends in the earthquake data and try to make sense of various patterns that can be found.

Every year, Federal Emergency Management Agency (FEMA), earthquakes losses are estimated to be around 4.4 billion USD annually in the United States. The casualty numbers due to tsunamis and earthquakes range from 20000-30000 every year on an average.

Our main goal is to predict tsunamis based on earthquake and seismic activity so that the loss of life and damages to property and the environment can be minimized as far as possible. Our project can be used in conjunction with pre-existing systems to help people get to safety and also use appropriate measures for constructing buildings in areas which are earthquake and tsunami prone.

There is a need for this project which will help save lives and also prevent destruction of property. Tsunami prediction models can be leveraged to sound warning systems and get ships, boats and other water transportation mediums away from the tsunami zone or get them to coast as safe as possible. The tsunami prediction system can also help people living in coastal regions move away to safer places in case of a tsunami. For earthquakes as well, with the trends and patterns that we have analyzed will help the governments and public authorities of the most earthquake-prone areas take preventive and corrective measures when earthquakes hit their areas.

The scope for this project extends up until only predicting tsunamis which might occur due to earthquakes. Similarly, we are not predicting earthquakes themselves but only checking the trends and patterns in the earthquakes that have already happened from 1995 - 2023.

II. Data Description

The dataset is taken from Kaggle. It contains the record of earthquakes that happened from 1995 - 2023 and the tsunamis that accompanied the earthquakes. It consists of 20 features and 1000 rows. The dataset columns are described as follows:

- title : name of the earthquake
- magnitude : magnitude of the earthquake
- datetime: the date and time at which the earthquake occurred

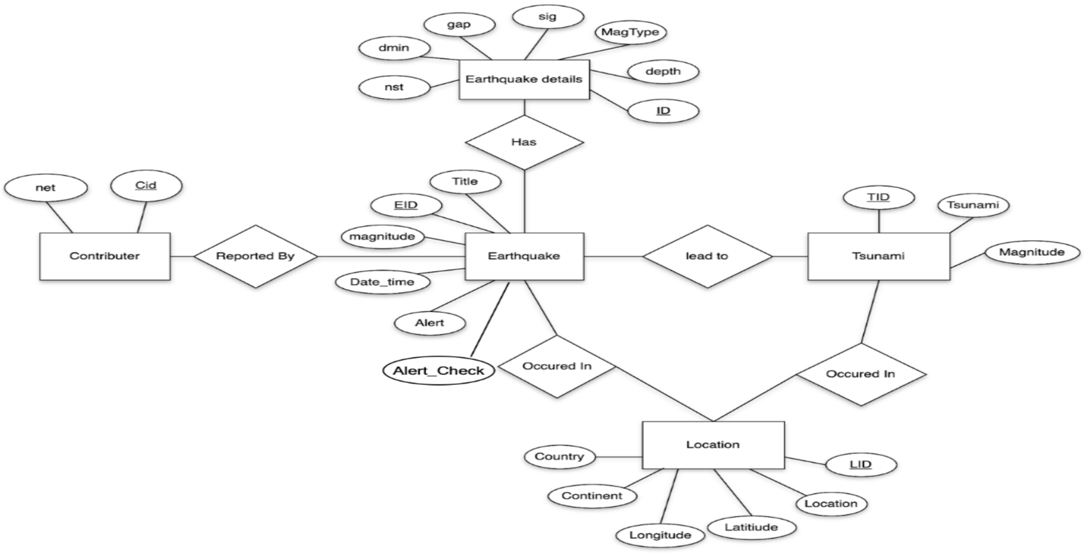


Fig. 1. Entity Relationship Diagram

- cdi : maximum reported intensity for the event range
- mmi: The maximum estimated instrumental intensity for the event
- alert: The alert level - “green”, “yellow”, “orange”, and “red”, depending upon the severity of the disaster
- tsunami: ”1” for events in oceanic regions indicating tsunami occurs after an earthquake and ”0” otherwise indicating tsunami doesn’t occur after the earthquake
- sig: A number describing how significant the event is. Larger numbers indicate a more significant event. This value is determined on a number of factors, including: magnitude, maximum MMI, felt reports, and estimated impact
- net: The ID of a data contributor. Identifies the network considered to be the preferred source of information for this event.
- nst: The total number of seismic stations used to determine earthquake location.
- dmin: Horizontal distance from the epicenter to the nearest station
- gap: The largest azimuthal gap between azimuthally adjacent stations (in degrees). In general, the smaller this number, the more reliable is the calculated horizontal position of the earthquake. Earthquake locations in which the azimuthal gap exceeds 180 degrees typically have large location and depth uncertainties
- magType: The method or algorithm used to calculate the preferred magnitude for the event
- depth: The depth where the earthquake begins to rupture
- latitude, longitude: coordinate system by means of which the position or location of any place on Earth’s surface can be determined and described
- location: location within the country
- continent: continent of the earthquake hit country
- country: affected country

The dataset contains columns having all types of data - object, float, int, datetime, etc. The columns contain categorical and numerical data both.

It is worth mentioning that every earthquake may not give rise to a tsunami, but every tsunami corresponds to an earthquake. This is an important point that we understand from the dataset.

III. Method

The project was divided into two parts. The first part involved creating the database and executing queries on it to understand trends and patterns. The second part entailed feature engineering and predictive modeling for the dataset, specifically for tsunami prediction.

A. Database Design

During the database design phase, our original dataset, which consisted of 20 columns, was divided into five distinct tables. These tables are Earthquake, Tsunami, Contributor, Location, and Earthquake Details. We established suitable relationships between these tables and imposed necessary integrity constraints. An Entity-Relationship (ER) diagram was created to visually represent all entities and their attributes in the database. Additionally, a relational schema was defined to further structure the database.

The Earthquake table contains all the information related to the earthquakes that have taken place. We create a primary key EID for this table which represents the ID of an earthquake. Attributes of the Earthquake table are EID, title, magnitude, alert, datetime, alertcheck.

The Tsunami table contains all the details related to tsunamis that occur due to earthquakes. Similar to the Earthquake table, we have created a primary key, TID, for identifying individual tsunamis. The attributes of this table include 'tsunami' and 'magnitude'. The 'tsunami' attribute indicates whether a tsunami occurred after an earthquake, and the 'magnitude' attribute represents the magnitude of the earthquake that potentially caused the tsunami.

The Location table provides geographical information about the sites where earthquakes have occurred. We create a primary key LID for the location. The other attributes in the table describe various aspects of the location. These attributes are 'country', 'continent', 'location', 'latitude', and 'longitude', the meanings of which are self-explanatory.

The Contributor table is designed to store information about the sources that have contributed to the data and observations related to specific earthquakes and tsunamis. Each contributor is uniquely identified by a primary key, CID, which stands for Contributor ID. The other attribute is 'net' which represents the ID of the data contributor and represents the network associated.

The fifth table is the Earthquake details table. This table contains more information related to the earthquakes. The attributes in this table are nst, min, magType, gap, dmin, depth, ID. All of these attributes are described in the dataset description. here, the ID attribute is considered as the primary key.

The relationships between these entities are :

- Earthquake leads to Tsunami (one-to-one relation)
- Earthquake occurred in Location (one-to-one relation)
- Tsunami occurred in Location (one-to-one relation)
- Earthquake reported by Contributer (one-to-many relation)
- Earthquake has EarthquakeDetails (one-to-one relation)

The relational schema for the database is represented as

- Earthquake Table (<u>EID</u>, Title, Magnitude, DateTime, Alert, Alertcheck, LID, ID, Cid)
- Tsunami Table (<u>TID</u>, Tsunami, Magnitude, EID, LID)
- Location Table (<u>LID</u>, Latitude, Longitude, Location, Continent, Country)
- EarthquakeDetails Table (<u>ID</u>, NST, Dmin, Gap, MagType, Depth, Sig)
- Contributor Table (<u>CID</u>, Net)

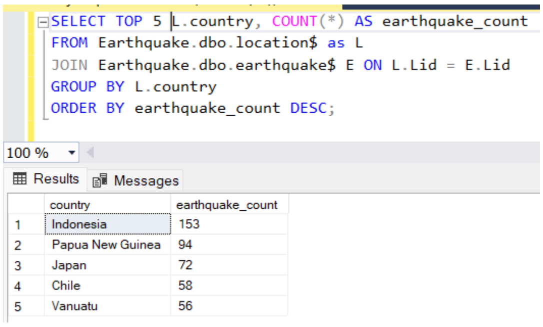
We then load our database with these tables and implement appropriate primary and foreign key constraints to the attributes.

B. SQL Queries and Visualization

We use MySQL queries to analyze 6 trends and patterns primarily. The trends which we analyze are:

- Frequency by Location: Frequency of the earthquake at different locations. How much frequently due earthquakes occur in different parts of the world, primarily in what countries is the frequency highest.
- Frequency over Time: Change in frequency of earthquakes over time. Have the number of earthquakes reduced, or increased and what factors is this increase/reduction dependent on.
- Most Affected Years: Analysis of most intense earthquakes and places affected due to it. What year(s) did the worst earthquakes and tsunamis occur.
- False Alert Analysis: False alerts found over the period of time. When does the magnitude and alert levels issued according to the magnitude not match. How many cases exist where false alarms have been raised or vice-versa.
- Most Affected Season: Earthquake and tsunami frequency patterns over seasons. In which seasons do the earthquakes and tsunamis occur most frequently.
- Intensity and Tsunami: Correlation between earthquake magnitude and tsunami. Over what magnitude of an earthquake will give rise to tsunamis.

For Frequency by Location, it is observed the following five countries have the highest frequency of earthquakes - Indonesia, Papua New Guinea, Japan, Chile, Vanuatu with 152, 94, 72, 58, 56 earthquakes respectively. More details can be seen in the results table below. The MYSQL query used is given in *Figure 2*

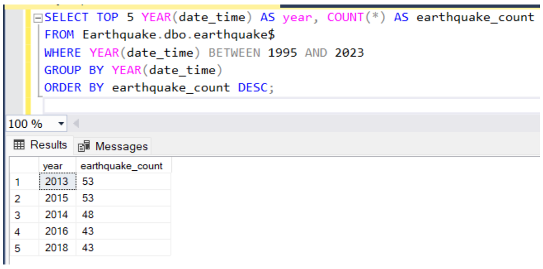


```
SELECT TOP 5 L.country, COUNT(*) AS earthquake_count
FROM Earthquake.dbo.location$ as L
JOIN Earthquake.dbo.earthquake$ E ON L.Lid = E.Lid
GROUP BY L.country
ORDER BY earthquake_count DESC;
```

	country	earthquake_count
1	Indonesia	153
2	Papua New Guinea	94
3	Japan	72
4	Chile	58
5	Vanuatu	56

Fig. 2. Example Query 1 - Frequency by Location

For Frequency by Time, most earthquakes have occurred in the years 2013 and 2015, with a gradual increase in the number of earthquakes. The table below shows the number of earthquakes that occurred every year from 1995 - 2023. The MYSQL query used is given in *Figure 3*.

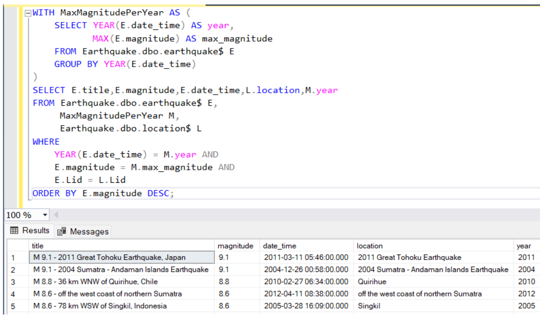


```
SELECT TOP 5 YEAR(date_time) AS year, COUNT(*) AS earthquake_count
FROM Earthquake.dbo.earthquake$
WHERE YEAR(date_time) BETWEEN 1995 AND 2023
GROUP BY YEAR(date_time)
ORDER BY earthquake_count DESC;
```

	year	earthquake_count
1	2013	53
2	2015	53
3	2014	48
4	2016	43
5	2018	43

Fig. 3. Example Query 2 - Frequency over Years

Most Affected Years were 2004 and 2011, where earthquakes having highest magnitude of 9.1 took place off the coast of Japan. The MYSQL query used is given in *Figure 4*.



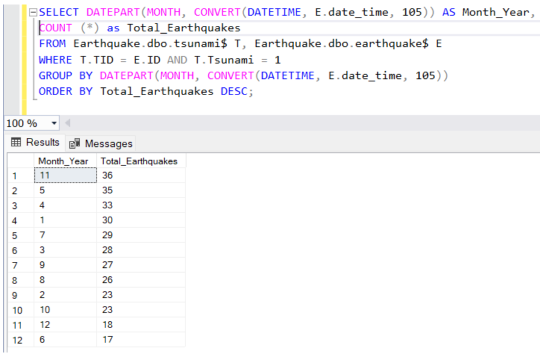
```
WITH MaxMagnitudePerYear AS (
    SELECT YEAR(E.date_time) AS year,
           MAX(E.magnitude) AS max_magnitude
    FROM Earthquake.dbo.earthquake$ E
    GROUP BY YEAR(E.date_time)
)
SELECT E.title,E.magnitude,E.date_time,L.location,M.year
FROM Earthquake.dbo.earthquake$ E,
     MaxMagnitudePerYear M,
     Earthquake.dbo.location$ L
WHERE
    YEAR(E.date_time) = M.year AND
    E.magnitude = M.max_magnitude AND
    E.Lid = L.Lid
ORDER BY E.magnitude DESC;
```

	title	magnitude	date_time	location	year
1	M 9.1 - 2011 Great Tohoku Earthquake, Japan	9.1	2011-03-11 05:46:00.000	2011 Great Tohoku Earthquake	2011
2	M 9.1 - 2004 Sumatra - Andaman Islands Earthquake	9.1	2004-12-26 00:58:00.000	2004 Sumatra - Andaman Islands Earthquake	2004
3	M 8.8 - 36 km WNW of Quirihue, Chile	8.8	2010-02-27 06:34:00.000	Quirihue	2010
4	M 8.6 - off the west coast of northern Sumatra	8.6	2012-04-11 08:38:00.000	off the west coast of northern Sumatra	2012
5	M 8.6 - 78 km WSW of Singkil, Indonesia	8.6	2005-03-28 16:09:00.000	Singkil	2005

Fig. 4. Example Query 3 - Most Affected Years

False Alert Analysis was conducted, and the results are that for 404 earthquakes, the data for alert levels and the earthquake magnitude was not in sync. To elaborate on what a false alert means, we can say that if the alert type is 'red' but the magnitude is between 6 and 7, then it is a false alert as for alert type 'red' the magnitude should be 8.5 and above.

For Most Affected Seasons, the earthquakes in November, May and April were far more than earthquakes in other months. The number of earthquakes for these months was 36, 35 and 33 respectively. The MYSQL query used is given in *Figure 5*.

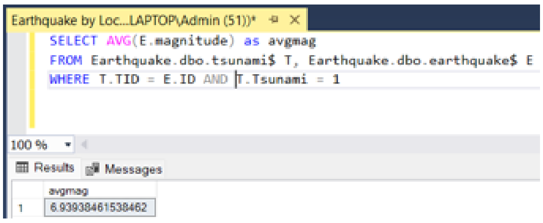


```
SELECT DATEPART(MONTH, CONVERT(DATETIME, E.date_time, 105)) AS Month_Year,
COUNT (*) as Total_Earthquakes
FROM Earthquake.dbo.tsunami$ T, Earthquake.dbo.earthquake$ E
WHERE T.TID = E.ID AND T.Tsunami = 1
GROUP BY DATEPART(MONTH, CONVERT(DATETIME, E.date_time, 105))
ORDER BY Total_Earthquakes DESC;
```

	Month_Year	Total_Earthquakes
1	11	36
2	5	35
3	4	33
4	1	30
5	7	29
6	3	28
7	9	27
8	8	26
9	2	23
10	10	23
11	12	18
12	6	17

Fig. 5. Example Query 4 - Most Affected Months

For Intensity and Tsunami, there were 325 earthquakes which were followed by tsunamis. Majority of these took place at a local time 6pm to 12 am in the affected regions. The average magnitude of earthquake which increases the likelihood of Tsunami occurring was found to be 6.98. The MYSQL query used is given in *Figure 6*.



```
SELECT AVG(E.magnitude) as avgmag
FROM Earthquake.dbo.tsunami$ T, Earthquake.dbo.earthquake$ E
WHERE T.TID = E.ID AND T.Tsunami = 1
```

	avgmag
1	6.93938461538462

Fig. 6. Example Query 5 - Relation between Intensity of Earthquake and Tsunami

C. Feature Engineering

We perform feature engineering in order to pick the best features to train the dataset on for tsunami prediction. The correlation matrix was used for the same. It was observed that some features are highly correlated with each other. Depending on this, the features were picked to train our prediction model. However, before that the data has been cleaned and pre-processed in order to obtain accurate results. We convert the datetime column to the datetime64 [ns] format. We also check if any duplicate or NaN data is present in the dataset and get rid of it.

After this, we draw up the correlation matrix for the attributes sig, Magnitude, cdi, mmi, dmin, depth, tsunami, nst. Looking at the matrix, we observe that the highest positive correlation exists between magnitude and sig features. sig and cdi also show high positive correlation between them. Similarly, tsunami and dmin also have a good amount of correlation. These are the features that we would consider while building the prediction model.

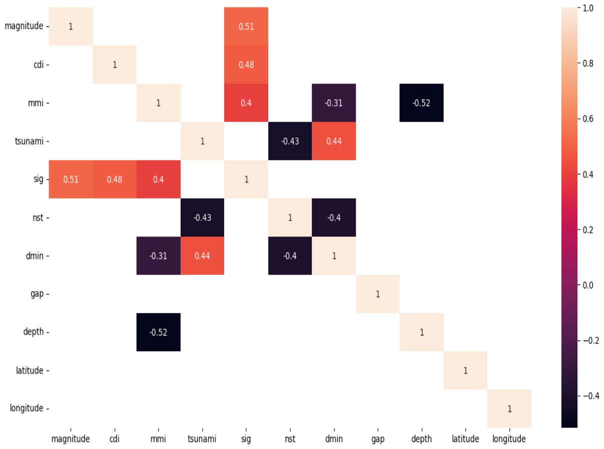


Fig. 7. Correlation Heat-Map

On the other hand, the mmi and depth features show the highest negative correlation between them. tsunami and nst are also negatively correlated, as are nst and dmin. dmin and mmi are negatively correlated too.

Another important aspect of feature engineering was to handle the challenges of imbalanced classes. In the dataset, as we mentioned previously, not every earthquake would give rise to a tsunami. Hence, it was also important to handle these imbalanced classes. We observe that the dataset contained about 30 percent records of Tsunami occurring after an earthquake. This would have given rise to lower accuracies if the same dataset was to be used directly for model prediction. It would have created skewed results as the input dataset is also skewed towards tsunamis not occurring after an earthquake. In the figure given, we see that there are around 670 result classes where tsunami does not occur as compared to 300 result classes where tsunami occurs after earthquakes.

To tackle this challenge, we implement SMOTE. The full-form of SMOTE is Synthetic Minority Oversampling Technique. SMOTE is a data augmentation technique for a minority class. SMOTE over samples the minority class and hence helps abolish the skewness of the result classes. Using SMOTE, the samples for the minority class have been increased and made equal to the samples of the majority class. We ensure that there are 528 samples of both classes after implementing SMOTE.

We also try under sampling the data to reduce the skew and hence try to improve the dataset needed for model prediction. The samples of the majority class have been reduced and made equal to the samples of the minority class. We ensure that there are 295 samples of both classes after implementing under sampling. Both the under sampling and over sampling methods have been analysed to see which one gave the best prediction accuracies.

D. *Predictive Modelling*

Two approaches are used for predictive modelling. The first approach is using data obtained by oversampling to train the model and the second approach is to use the data obtained using under sampling. Five ML models are trained for both the approaches. They are:

- Random Forest: Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result.
- Logistic Regression: Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false.
- XG Boost: XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.
- Decision Tree: A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.
- Naive Bayes: A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

We split the dataset into train and test sets first with 800 training set and 200 test sets. We then train the dataset for each ML model and compute the train and test accuracy. The support, f-1, precision score and recall scores are also calculated which help us understand the performance of the models.

In the case of oversampling, the Random Forest and XG Boost models gave us the best results in terms of accuracies. Decision trees, Logistic Regression and Naive Bayes do not perform as well as the other models. Similarly for under sampling as well, the same trend is observed.

It is observed that Random Forest and XG Boost are best suited for this task. Using Random Forest and XGBoost for both oversampling and under sampling we get similar results.

It can also be observed that the random forest classifier with the under sampling method will be better for this by looking at the confusion matrix as we are more interested in making a prediction where an earthquake lead to a tsunami. If we falsely made this prediction, it might be acceptable. However, if we fail to make a prediction of the actual occurrence, this might have a deeper implication. Therefore, from the confusion matrix the False Positive and the False negative is lesser in Random forest classifier with under sampling.

The confusion matrices obtained and the results of all algorithms with under sampling technique are shown below.

RESULTS

The key takeaways from this project can be classified into two categories. The first category is of analyzing the trends and patterns in the dataset related to earthquakes. We describe the dataset and use SQL queries to understand various trends. Out of the trends which we identify, we present the results of trends like Countries Most Affected by Earthquakes, Most Earthquakes Which Took Place in What Time Period, Years having the Highest Severity of Earthquakes, Total Number of False Alerts, Severity according to Months and Earthquakes giving rise to Tsunamis.

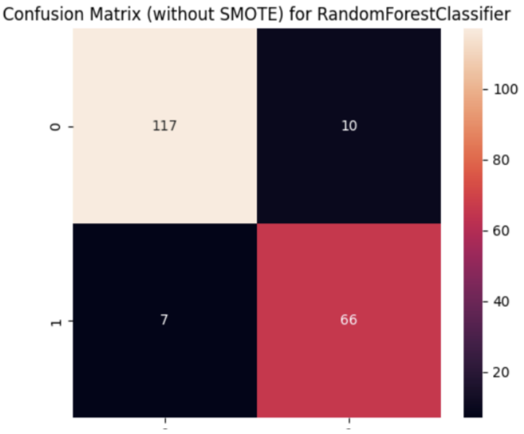


Fig. 8. Confusion Matrix with under sampling

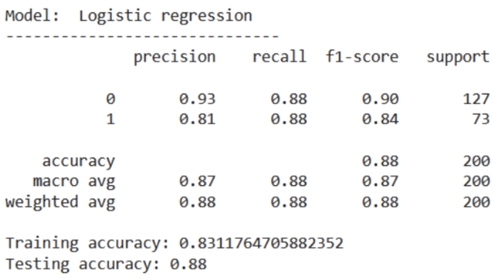

```
Model:  Logistic regression
-----------------------------
              precision    recall  f1-score   support

           0       0.93      0.88      0.90       127
           1       0.81      0.88      0.84        73

    accuracy                           0.88       200
   macro avg       0.87      0.88      0.87       200
weighted avg       0.88      0.88      0.88       200

Training accuracy: 0.8311764705882352
Testing accuracy: 0.88
```

Fig. 9. Logistic regression with under sampling

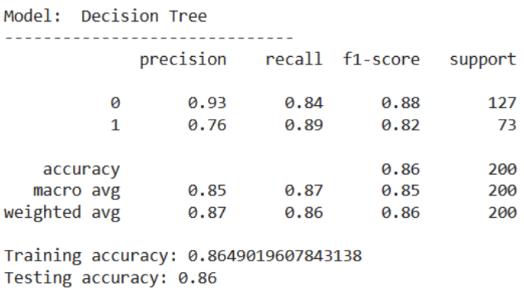

```
Model:  Decision Tree
-----------------------------
              precision    recall  f1-score   support

           0       0.93      0.84      0.88       127
           1       0.76      0.89      0.82        73

    accuracy                           0.86       200
   macro avg       0.85      0.87      0.85       200
weighted avg       0.87      0.86      0.86       200

Training accuracy: 0.8649019607843138
Testing accuracy: 0.86
```

Fig. 10. Decision Tree with under sampling

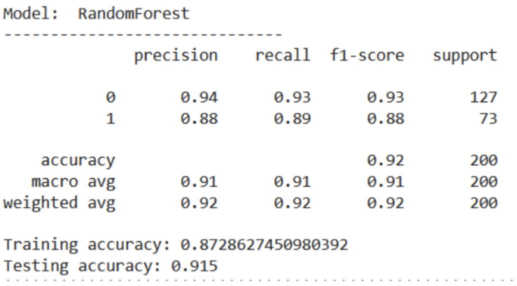

```
Model:  RandomForest
-----------------------------
              precision    recall  f1-score   support

           0       0.94      0.93      0.93       127
           1       0.88      0.89      0.88        73

    accuracy                           0.92       200
   macro avg       0.91      0.91      0.91       200
weighted avg       0.92      0.92      0.92       200

Training accuracy: 0.8728627450980392
Testing accuracy: 0.915
```

Fig. 11. Random Forest with under sampling

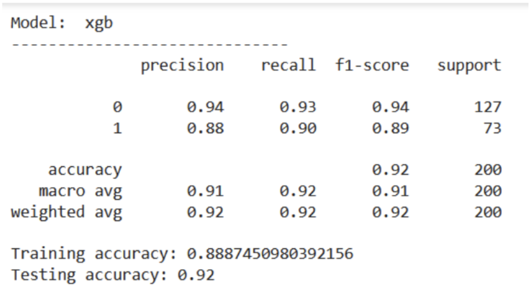

```
Model:  xgb
-----------------------------
              precision    recall  f1-score   support

           0       0.94      0.93      0.94       127
           1       0.88      0.90      0.89        73

    accuracy                           0.92       200
   macro avg       0.91      0.92      0.91       200
weighted avg       0.92      0.92      0.92       200

Training accuracy: 0.8887450980392156
Testing accuracy: 0.92
```

Fig. 12. XGBoost with under sampling

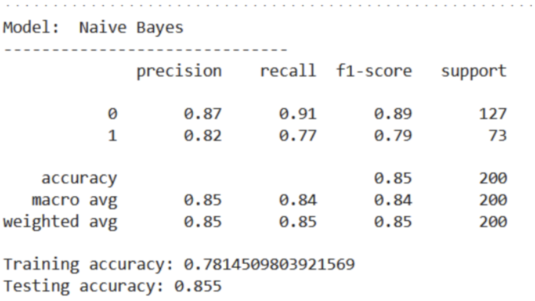

```
Model:  Naive Bayes
-----------------------------
              precision    recall  f1-score   support

           0       0.87      0.91      0.89       127
           1       0.82      0.77      0.79        73

    accuracy                           0.85       200
   macro avg       0.85      0.84      0.84       200
weighted avg       0.85      0.85      0.85       200

Training accuracy: 0.7814509803921569
Testing accuracy: 0.855
```

Fig. 13. Naive Bayes with under sampling

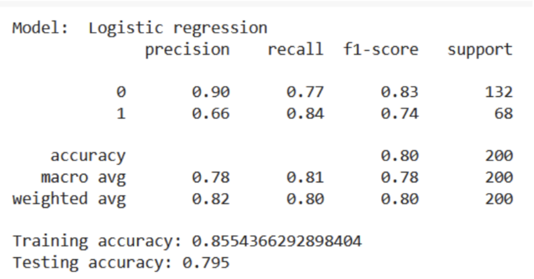

```
Model:  Logistic regression
              precision    recall  f1-score   support

           0       0.90      0.77      0.83       132
           1       0.66      0.84      0.74        68

    accuracy                           0.80       200
   macro avg       0.78      0.81      0.78       200
weighted avg       0.82      0.80      0.80       200

Training accuracy: 0.8554366292898404
Testing accuracy: 0.795
```

Fig. 14. Logistic regression with oversampling

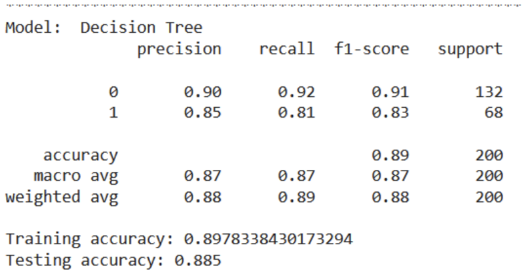
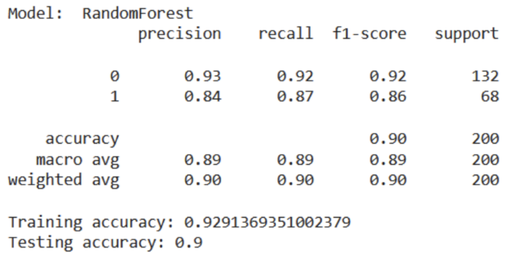

```
Model:  Decision Tree
              precision    recall  f1-score   support

           0       0.90      0.92      0.91       132
           1       0.85      0.81      0.83        68

    accuracy                           0.89       200
   macro avg       0.87      0.87      0.87       200
weighted avg       0.88      0.89      0.88       200

Training accuracy: 0.8978338430173294
Testing accuracy: 0.885
```

Fig. 15. Decision Tree with oversampling

The second category of results presented is in accordance with the models trained for predicting whether a tsunami follows an earthquake, depending on various factors. The results for this method can be split into two sections - results of ML models with under sampling and results of the same ML models with oversampling using SMOTE.

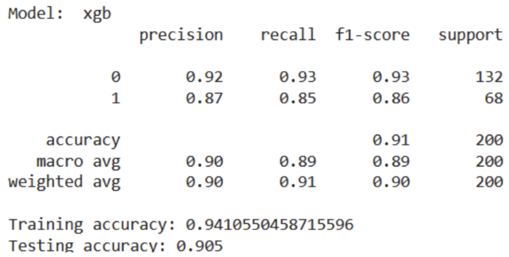
We implemented five ML models - Naive Bayes, Logistic Regression, XG Boost, Random Forest and Decision Tree to check which is the best.

With under sampling, the best results were obtained



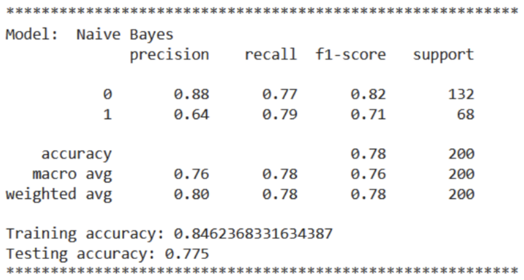
```
Model:  RandomForest
               precision    recall  f1-score   support

           0       0.93      0.92      0.92       132
           1       0.84      0.87      0.86        68

    accuracy                           0.90       200
   macro avg       0.89      0.89      0.89       200
weighted avg       0.90      0.90      0.90       200

Training accuracy: 0.9291369351002379
Testing accuracy: 0.9
```

Fig. 16. Random Forest with oversampling



```
Model:  xgb
               precision    recall  f1-score   support

           0       0.92      0.93      0.93       132
           1       0.87      0.85      0.86        68

    accuracy                           0.91       200
   macro avg       0.90      0.89      0.89       200
weighted avg       0.90      0.91      0.90       200

Training accuracy: 0.9410550458715596
Testing accuracy: 0.905
```

Fig. 17. XGBoost with oversampling



```
************************************************************
Model:  Naive Bayes
               precision    recall  f1-score   support

           0       0.88      0.77      0.82       132
           1       0.64      0.79      0.71        68

    accuracy                           0.78       200
   macro avg       0.76      0.78      0.76       200
weighted avg       0.80      0.78      0.78       200

Training accuracy: 0.8462368331634387
Testing accuracy: 0.775
************************************************************
```

Fig. 18. Naive Bayes with oversampling

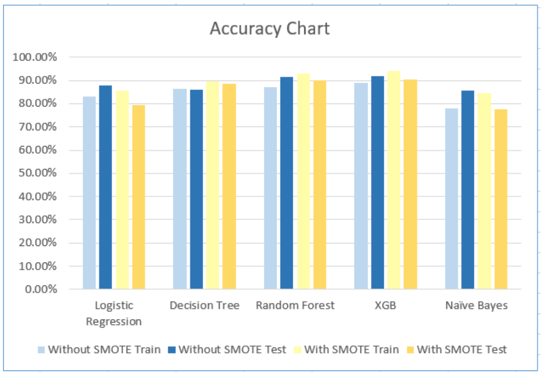


Fig. 19. Comparison of Train and Test Accuracy of Algorithms with both approaches

by Random Forest and XGBoost models, with a test accuracy of 0.915 and 0.92 respectively. The next best results were given with Logistic Regression, Decision Trees and Naive Bayes algorithms with test accuracies of 0.88, 0.86 and 0.855 respectively. Other measures like precision, recall and f1-score can be found in the images/tables below.

For oversampling, we used the SMOTE technique. The test accuracy for Naive Bayes is 0.775, Logistic Regression is 0.795, XG Boost is 0.941, Random Forest is 0.9291 and Decision Tree is 0.885. Hence, we can see that with SMOTE, XGBoost and Random Forest give the best results.

Conclusion

In conclusion, our analysis of earthquake data from 1995 to 2023, conducted using Python and SQL, has shown some insights into seismic patterns and potential links with tsunamis. After carrying out data cleaning, pre-processing, and feature engineering, we visualized trends and also made use of predictive modeling to predict the possibility of a tsunami. Through feature engineering, we were able to enhance the predictive power of the model. Integration of SQL further enhanced the analytical capabilities, helping in efficient querying and summarizing of the data set.

Acknowledgment

We express our gratitude to Jaideep Srivatsava, Professor, Director of Undergraduate Studies for Data Science, Department of Computer Science and Engineering, University of Minnesota, for this opportunity to use the knowledge acquired in class in practical setting. We extend our sincere thanks to our Teaching Assistant Jiacheng Liu for unwavering support and clearing doubts in timely manner which helped us to get through classes seamlessly.

The dataset is taken from the Kaggle website. The languages used are Python and SQL. The software used are MySQL server, Google Collab. The Python libaries used to carry out the project are seaborn, pandas, numpy, sklearn, matplotlib.

The work was divided between four members after identifying the data set and analysing the problem statement. We came up with 12 research questions which will help in analysing the trends in the data set and was divided equally between us to find the answers and conduct exploratory data analysis. After understanding the data set Aniket and Surabhi made the ER diagram and relational schema which was loaded into the MYSQL server by Mihir. The database was then queried by Evina and Mihir to extract relevant information from the tables. Feature engineering and prediction model was done by Evina and Surabhi to use the relevant features for the prediction of tsunami if an earthquake occurs.

References

[1] https://www.kaggle.com/datasets/warcoder/earthquake-dataset/data

[2] https://riteshuppal1402.medium.com/part-1-earthquake-data-analysis-2b556f023e61

[3] https://earthquake.usgs.gov/earthquakes

[4] https://datascienceplus.com/earthquake-analysis-1-4-quantitative-variables-exploratory-analysis/