Group 11

QUAKE

ANALYSIS OF NEAR FAULT NON-PULSE GROUND MOTIONS USING ARTIFICIAL NEURAL NETWORK MODELS

Meet The Team

ADVAITH
Team Leader

SRUTILEKA
Design Lead

SRIJA REDDY
Front-End Developer

DIVYA
Front-End Developer

PRANAV
Back-End Developer

Project Outline

- This project aims to train a model to predict whether earthquake is possible or not when inputted with parameters like **longitude, latitude, depth, azimuthal gap and root mean square.**

- This project aims to showcase the efficacy of deep learning in earthquake recognition and prediction by leveraging the increasing number of seismic monitoring stations and reduces causality and recovery funds for the destruction caused by earthquakes by being warned about the severity.

INDEX

What we are trying to build

- We are trying to build an active model which utilizes deep learning techniques in earthquake seismology by employing neural networks to analyze seismic data.
- The neural networks are trained on labeled data to enhance earthquake recognition and prediction, resulting in faster and more precise identification.
- The project has significant implications for mitigating earthquake hazards, providing timely warnings, and improving decision-making in earthquake preparedness and response.

Abstract

- Earthquakes in nutshell acts as nexus between tectonic rocks and various factors causing **Quake**. Developing accurate earthquake prediction models is crucial for effective disaster preparedness and risk management. The proposed model leverages historical earthquake data, seismic activity patterns and various geophysical features to predict the occurrence, magnitude, and location of earthquakes. **This model is not only limited to the usual factors rather it emphasises on wide range of factors for more fidelity and accuracy.** The model includes data collection, preprocessing, model training, and evaluation but widespread to more factors.

Business Use Case

- **Insurance companies -** Insurance companies often face significant financial losses due to earthquake-related claims. By implementing an earthquake prediction model, insurance companies could better assess the risk associated with insuring properties in earthquake-prone areas.

- **Infrastructure and planning -** Government agencies and civil engineering firms can leverage such models to make informed decisions regarding the design, construction and maintenance in earthquake prone areas.

- **Supply chain management -** Supply chain disruptions can have significant impacts on businesses.

Important Use Case

- The predominant use case is the **early warning system.**
- **Warning Generation:** Based on the analysis, if the system detects an earthquake, it generates an alert or warning message. The warning can be transmitted through various channels, including mobile phone notifications, television and radio broadcasts, and smartphone applications.

Why Quake?

- **Detecting Ground Motion, Seismometer Deployment** etc. are few ways seismologists predict the earthquake.
- But, in Quake we **Forecast** the earthquake (just like how we forecast weather) by previous earthquakes and the under-looked attributes of an earthquake.
- In the methods mentioned above, the seismologists analyze the rocks, the land around ad many more factors including **tectonic gap.**
- But our model uses the old data of previous earthquakes.
- Earthquakes is **directly proportional** to it's frequency.
- Hence by training the model, we can give the probable answer.

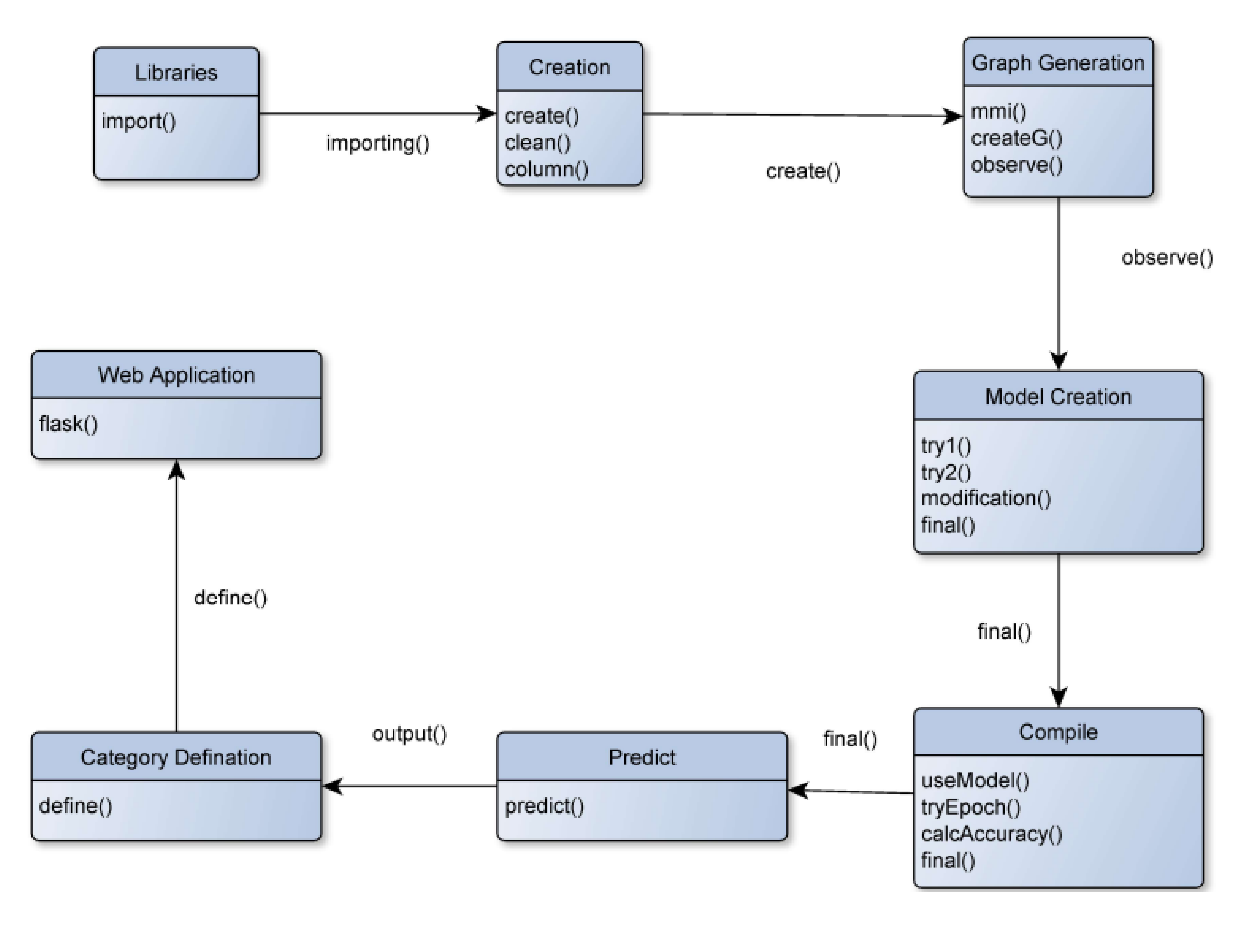
Concepts Used

- The main concept used here is **Seismic wave Analysis incorporated with Deep Learning algorithms.**

- This model utilizes deep learning techniques in earthquake seismology by employing neural networks to analyze seismic data.

- The neural networks are trained on labeled data to enhance earthquake recognition and prediction, resulting in faster and more precise identification.

System Model



Our Data

- The dataset contains 75000 rows of data. Each row is described with **Date, Time, Longitude, Latitude, RMS, Depth, Magnitude and it's type (MW) and Azimuthal Gap, etc**

- The data was obtained from a server NCEDC - **Northern California Earthquake Data Center & some part from Kaggle to maintain accuracy**

How we Processed Data

- Pre-processed using Pandas(read data), created a new column for MMI values, NumPy for data cleaning(taking off null values).
- Data is collected from the **NCEDC** network by configuring parameters such as time, radius, and magnitude threshold. The code connects to the NCEDC client, retrieves **station information, and obtains seismic events for each station.**
- Events are filtered based on magnitude and radius, and positive events are stored in a list.

Comparison of Models

- Random Forest - Disadvantageous as it requires higher computational power as it requires numerous trees to combine their outputs. This model outputted an accuracy of 73%. Although this model reduces variance and improves accuracy, it is not employed as it is a very high .

- Logistic Regression - The major disadvantage of this model was that it works well when the parameters are independent of each other, due to this disadvantage the model had an accuracy of 81%.

- SVM- Linear SVM world better if the number of parameters are less. The dataset had many features and hence was difficult to train and resulted in an accuracy of 83%.

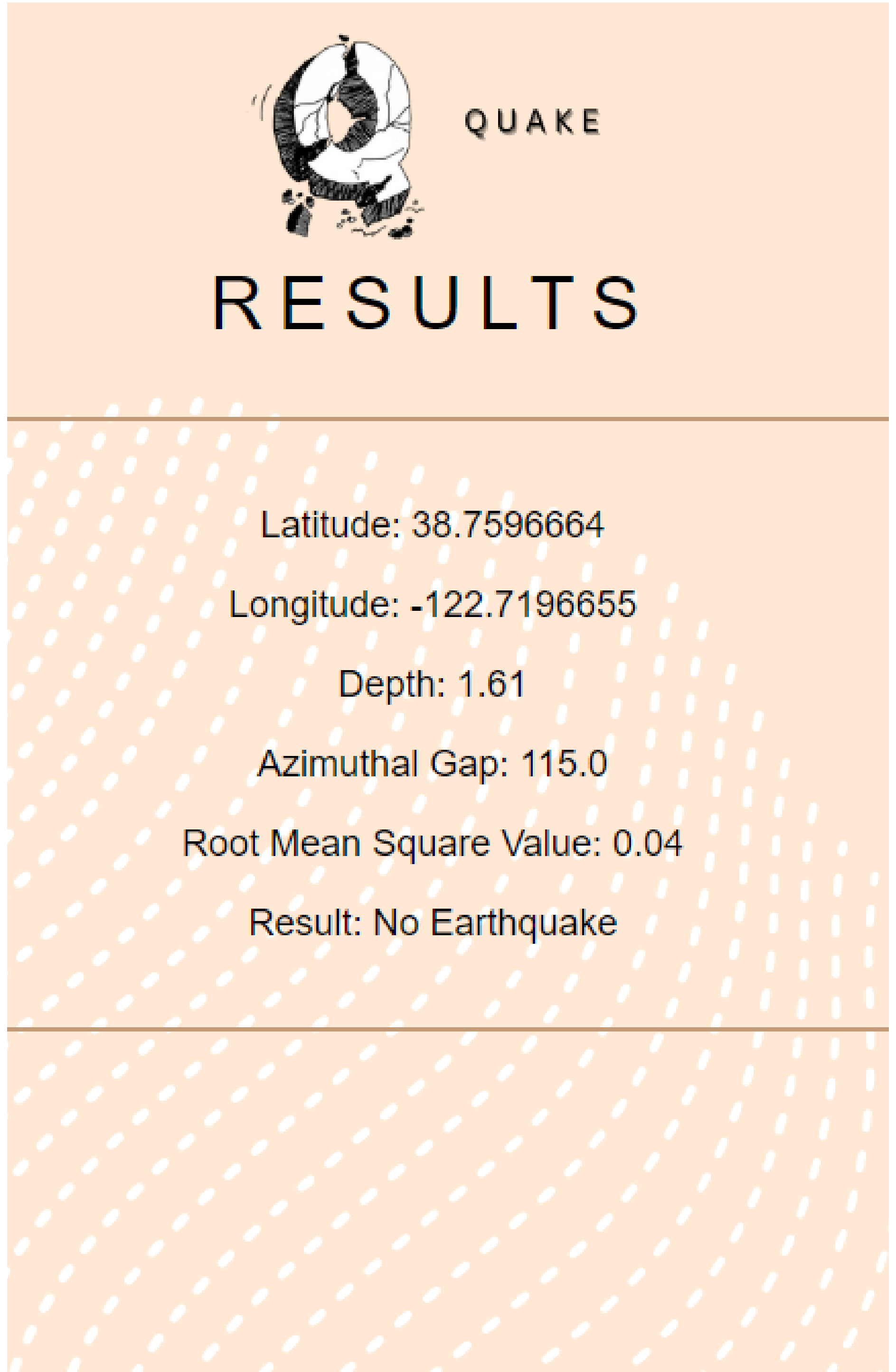
Algorithm Used

- This paper presents an overview of an earthquake prediction model utilizing **Deep Learning algorithms. (ANN)**
- The paper excludes date and time data as earthquakes are not predictable and do not depend on these two features.Then, a **Sequential model** is used to train the model.
- The loss function is computed using a **Binary cross entropy** model.
- The activation function uses **'ReLu' and 'Sigmoid'** function. The model uses 2 hidden layers and 1 sigmoid output layer.
- **Ngrok** is used for server hosting.

Optimal Case

LATITUTUDE: 38.75
LONGITUDE: -122.72
DEPTH: 1.61
AZIMUTHAL GAP: 0.05
RMS: 0.04

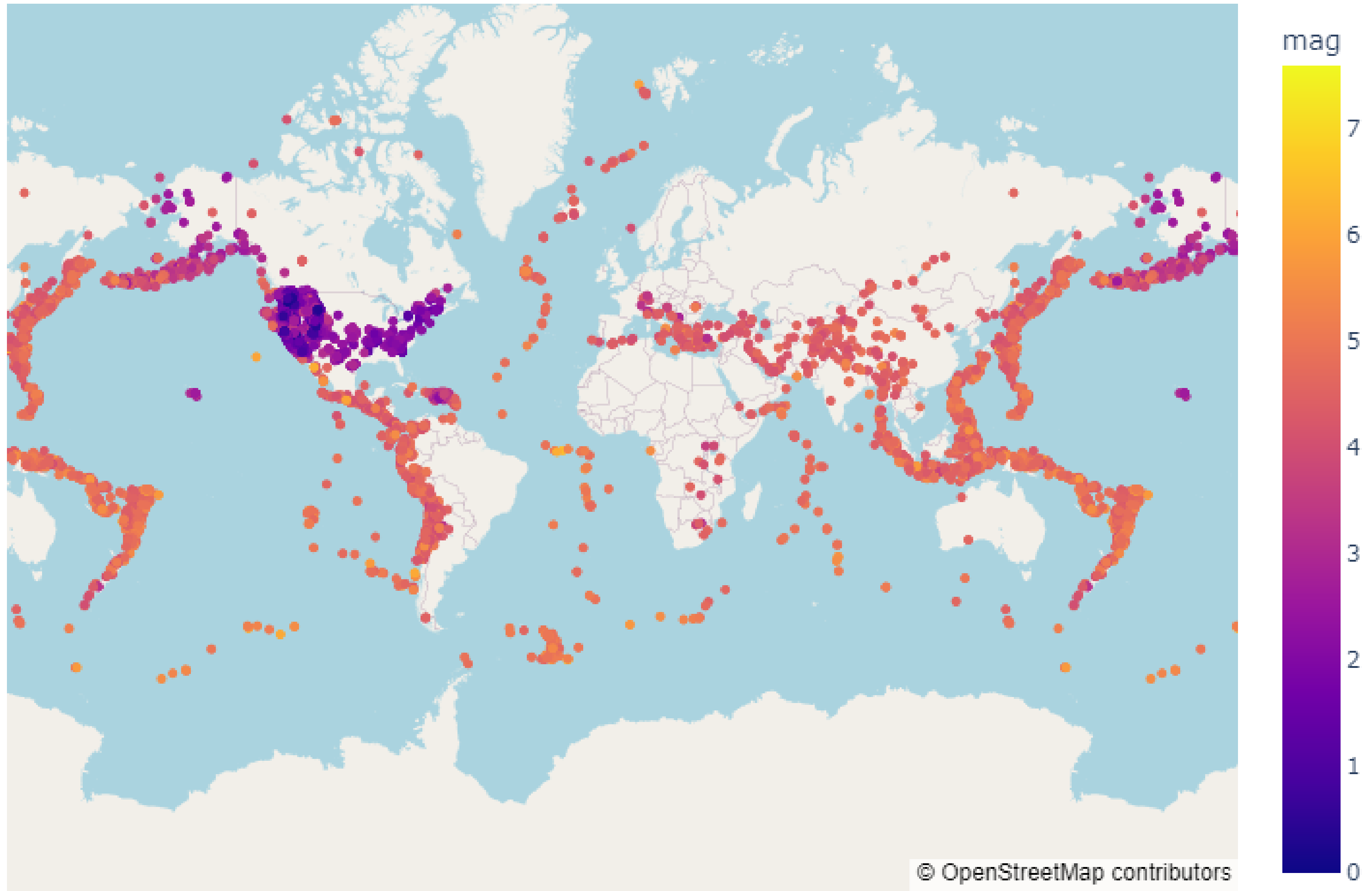
PREDICTED OUTPUT: NO EARTHQUAKE
ACTUAL OUTPUT: NO EARTHQUAKE



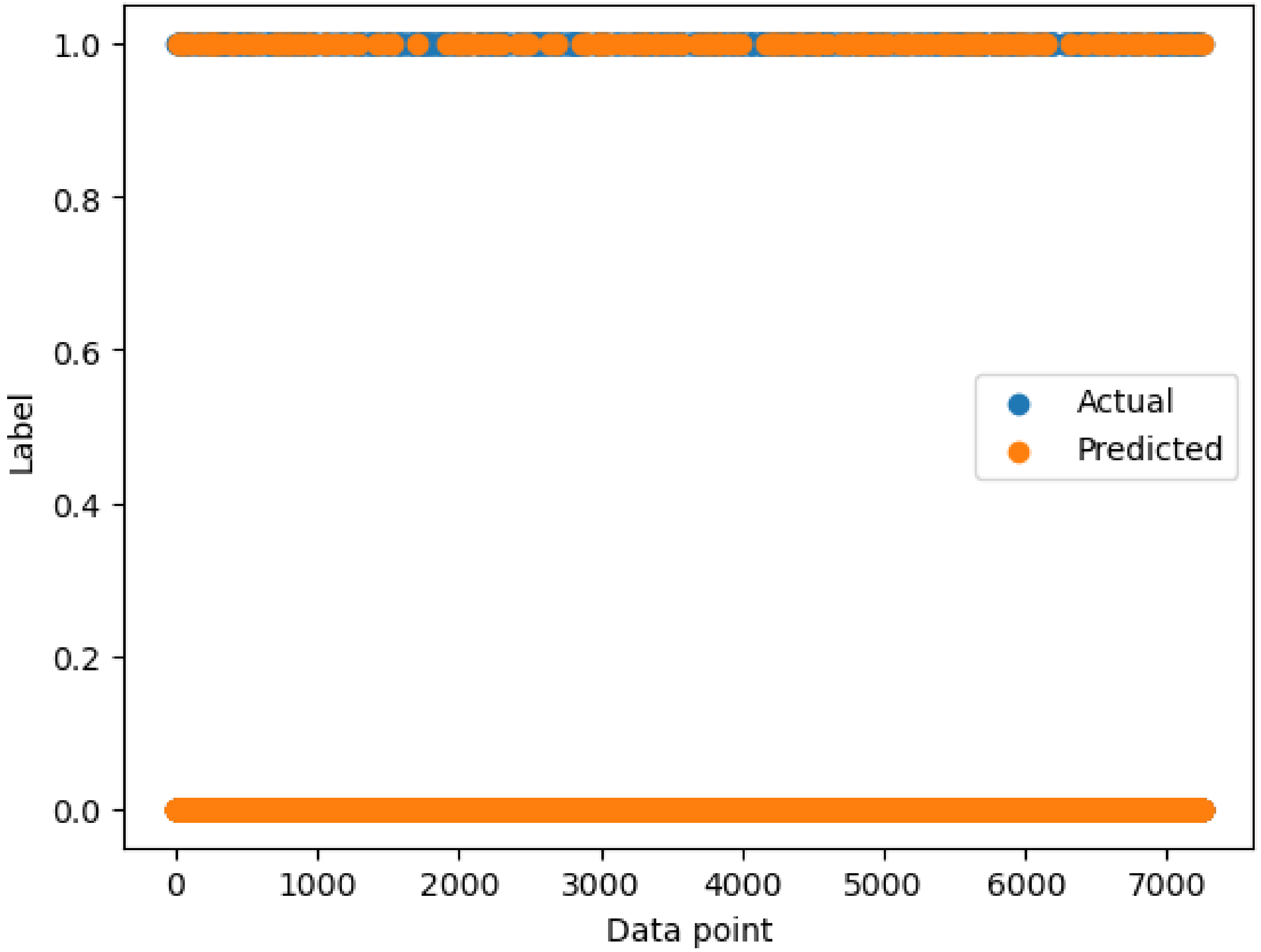
Our Output

- Once the attributes are provided, the model gives either:
  - Earthquake
  - No Earthquake
- It **will not** give the probability of when the earthquake will occur, because there is no correlation between **Time/Date and Occurrence of Earthquake.**
- So, our model just gives a forecast whether earthquake is possible in that area or not, in the near future.

Visualization



Earthquakes around the world scaled based on their magnitude



Actual vs Predicted graph

Challenges Faced

- **Data Quality**: Seismic data can be affected by various sources of noise and interference, including null values, negative magnitude values, etc.

- **Detection Threshold**: Detecting smaller-magnitude earthquakes is more challenging than detecting larger ones. Smaller events often produce weaker signals that are harder to distinguish from background noise.

- **Near-Field vs. Far-Field Events**: Detecting earthquakes that occur in close proximity to the monitoring stations (near-field events) is generally easier than detecting distant earthquakes (far-field events)

Our Learning

- Learnt how to implement and work around earthquake data.
- More attributes can enhance the quality of results.
- Remember to handle errors gracefully, and consider implementing alert systems or notifications for significant earthquakes.

Future Enhancements

- Incorporating the model onto a mobile app can make the application more accessible.

- Attempting to get a correlation between date and time of earthquakes.

DEMO

THANK YOU