Earthquake Data Management, Visualization, Analysis

Authors: Rishabh Narayanan (rrn42) and Siddarth Narayanan (srn79)

Project Type: Applied Project
Date: July 2024
GitHub: https://github.com/Rishabh-Narayanan/earthquake-predictor

Introduction

Thousands of earthquakes occur every year around the world, causing hundreds of deaths and thousands of injuries. The natural phenomenon underlying these tremors are not fully known, and earthquakes still often come as a surprise. From minor tremors to massive quakes, it is crucial to understand the relationships between an earthquake and its destructive power, in order to better prepare and respond to them. In this project, we aim to explore some of these ideas in order to potentially uncover what might constitute a powerful and destructive earthquake. Numerous public datasets containing earthquake data exist, and a similar wealth of tools to process and visualize these geospatial anomalies may be used to interpret this data.

Data

Earthquake data was acquired from the United States Geological Survey (USGS) application programming interface (API). The API exposes various useful metrics about earthquakes, including their geospatial coordinates (latitude and longitude), depth in kilometers, magnitude, date of occurrence, etc. Specifically, we targeted our analysis to all global earthquakes within the last 30 days. At the time of writing, 9436 earthquakes were recorded. This will change over time, but a local copy of the dataset was stored within the GitHub repository for reproducibility.

Data Storage and Cleaning

To first interact with the dataset, it must be converted into a usable tabular format. A relational SQL database is suitable for this purpose, as it can store thousands of records for efficient retrieval. Since the size of the dataset is relatively small (around 10 thousand samples), it can be stored fully in a local SQLite3 database.

To fetch the data, we first hit the USGS API to get JSON data with all earthquakes within the last 30 days, and then create a table within the SQLite3 database called `earthquakes`. This earthquake has 4 different columns (magnitude, latitude, longitude, and depth) which are each real numbers that encode the primary details about the earthquakes.

We then use the Python sqlite3 driver to push the extracted data from the JSON into this table. Because of the SQL database, simple queries like 'select all earthquakes with magnitudes greater than 5.0' become trivially translated into a SQL query

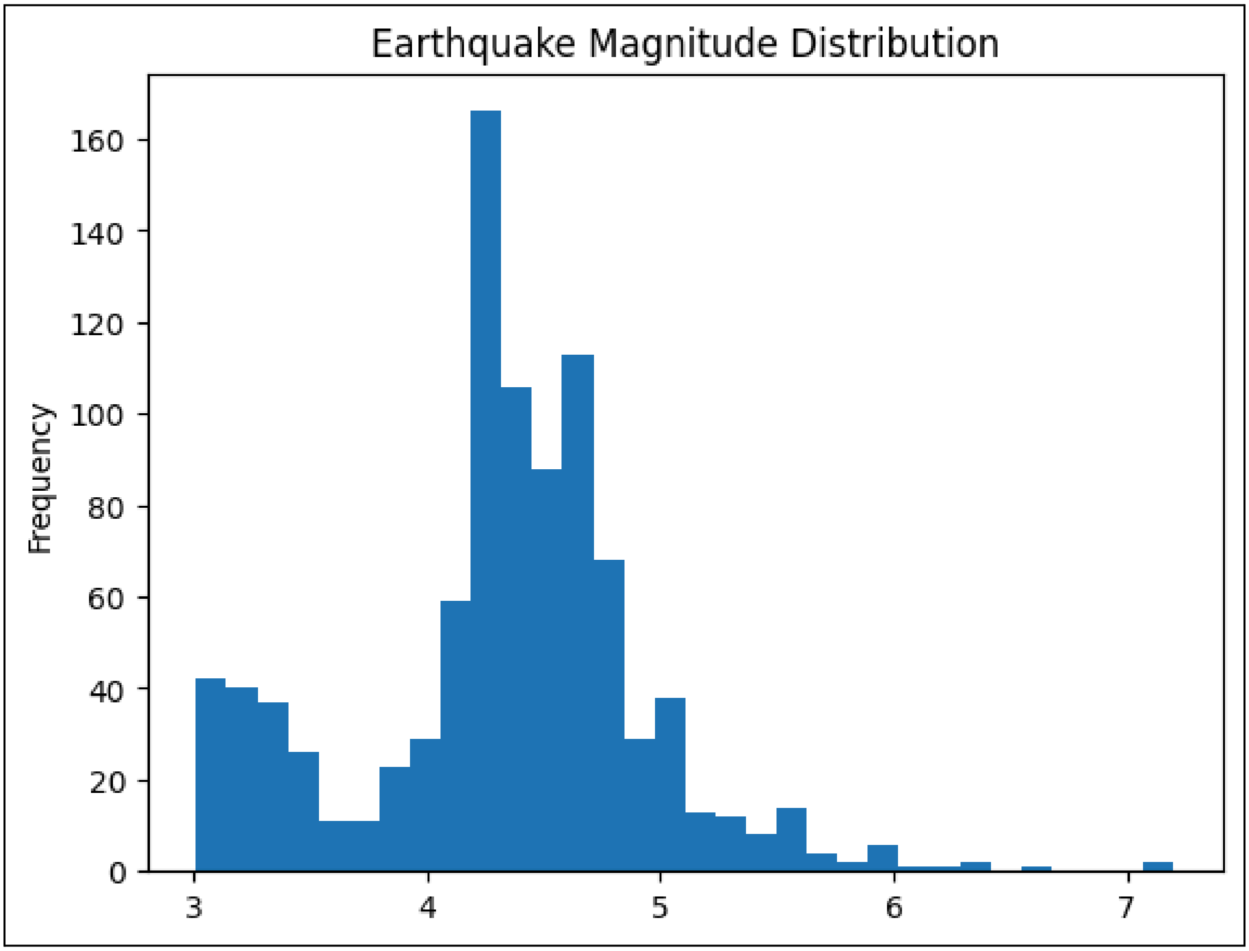
('SELECT * FROM earthquakes WHERE magnitude > 5.0;'). This simplicity is why SQL databases are conducive for data analysis.

However, during analysis, it is much more efficient to store as much data within memory rather than to read directly from the slower disk. Therefore, after storing data into the SQL database, we extract the relevant samples into a Pandas DataFrame. Here we also do some data cleaning by removing any rows with empty values and removing earthquakes with small magnitude. We define small magnitude as that less than or equal to 3 on the Richter scale, as those earthquakes rarely cause significant damage.

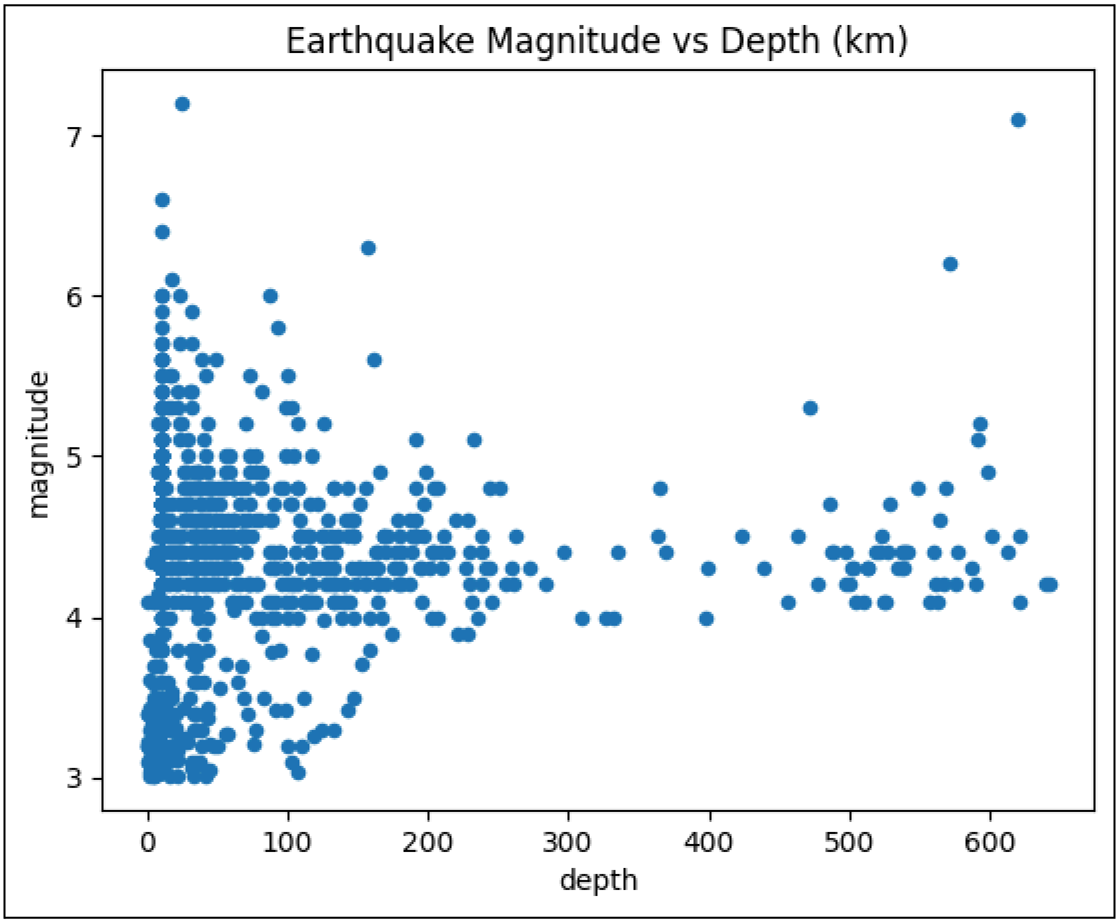
Exploratory Analysis

To perform analysis, it is important to identify the key characteristics of the features and the target class. Here, we are trying to predict magnitude using the other features, and so therefore magnitude is our target class, and location and depth constitute our feature space.

To explore the dataset, we first visualized the distribution of earthquake magnitude to understand which types of earthquakes were most common within the dataset. Interestingly, we see here that the distribution for this dataset happens to be bimodal with a peak near 3 and a peak near 4.5.

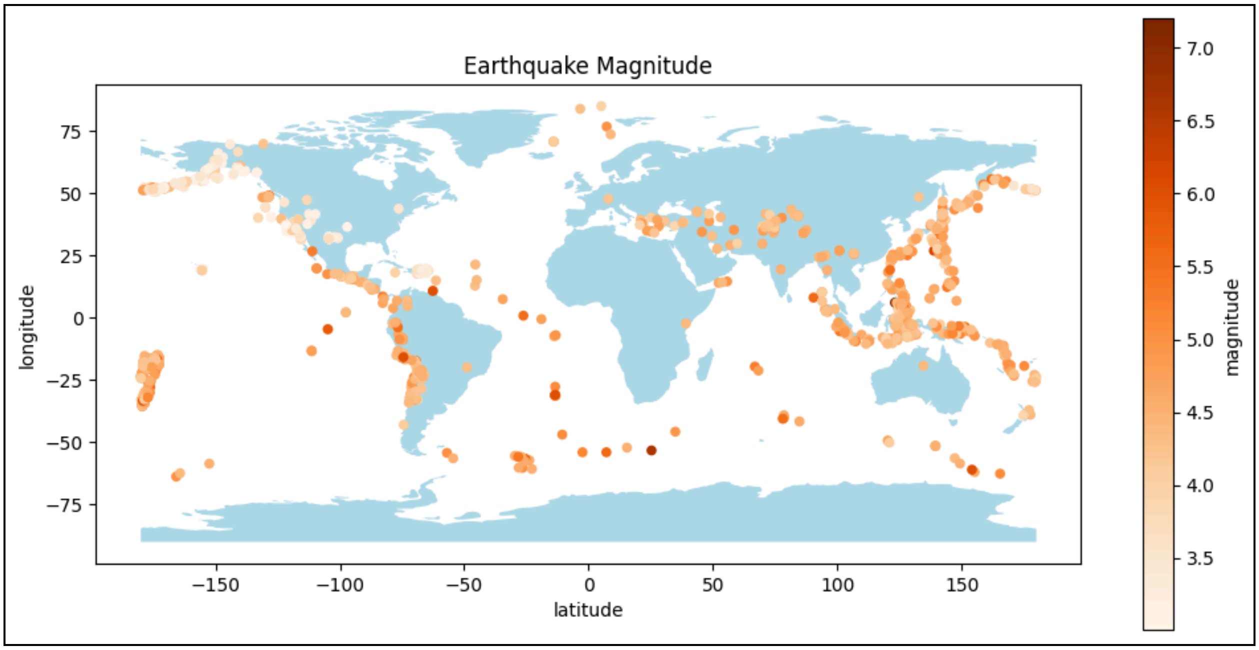


We also decided to see whether depth had a noticeable effect on the magnitude. Here, although no strong correlation is found, we do see small patterns. Namely, low depth seems to be a required yet insufficient metric for low earthquake magnitude. In other words, earthquakes with low magnitudes seem to occur only at shallow depths.

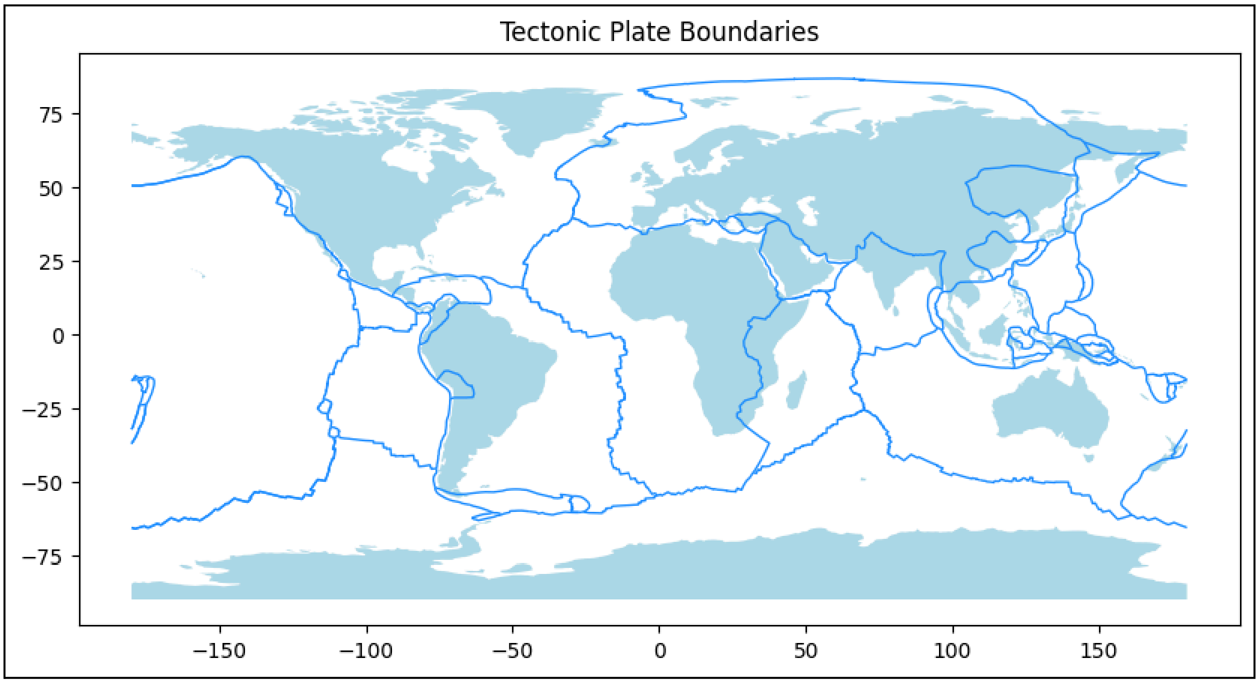


Geospatial Visualization

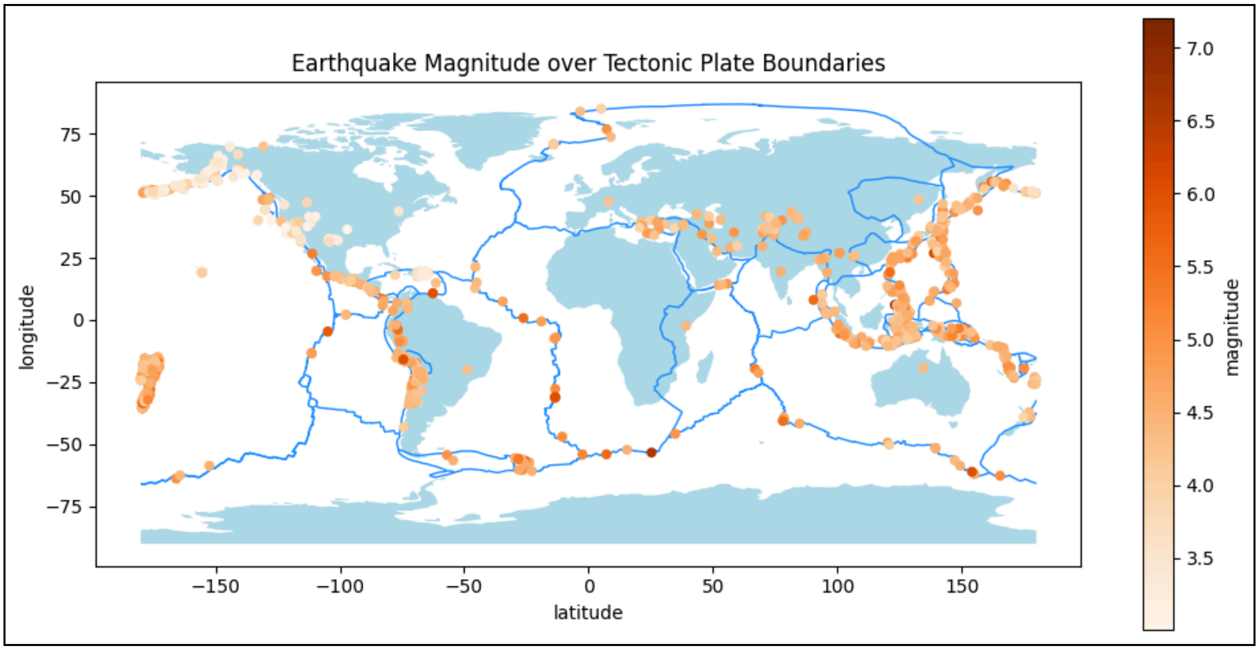
Since two of our features encode geospatial data (latitude and longitude) it is useful to visualize the data on a map. Here we plot the magnitude of each earthquake on a map, using color to indicate the strength. Plotting is done using Geopandas and Matplotlib.



What is interesting to see here is that the earthquakes seem to follow very distinct lines, with some being clustered over the ocean. Using domain knowledge of earthquakes, we decided to explore the relationship between these earthquakes and the boundaries of tectonic plates. Here, we plot the tectonic boundaries using publicly available tectonic plate data. This data can be accessed in the GitHub repository.



Interestingly, the points seem to line up almost exactly with these lines. Even the cluster in the middle of the pacific and atlantic oceans line up exactly. Here are both plots overlaid on top of each other for better clarity. This result stresses the importance of using previously known domain knowledge when analyzing data.



Feature Engineering

To better understand if distance to the tectonic plate boundary plays a role in the magnitude of the event, we decided to engineer a feature that represents this data. To our DataFrame, we added a column using the tectonic plate dataset which represents the distance to the closest tectonic plate boundary. We then compared the efficacy of machine learning techniques with and without the additional feature. This comparison would enable us to determine whether the distance really mattered or if it was just naturally correlated because of the nature of earthquakes.

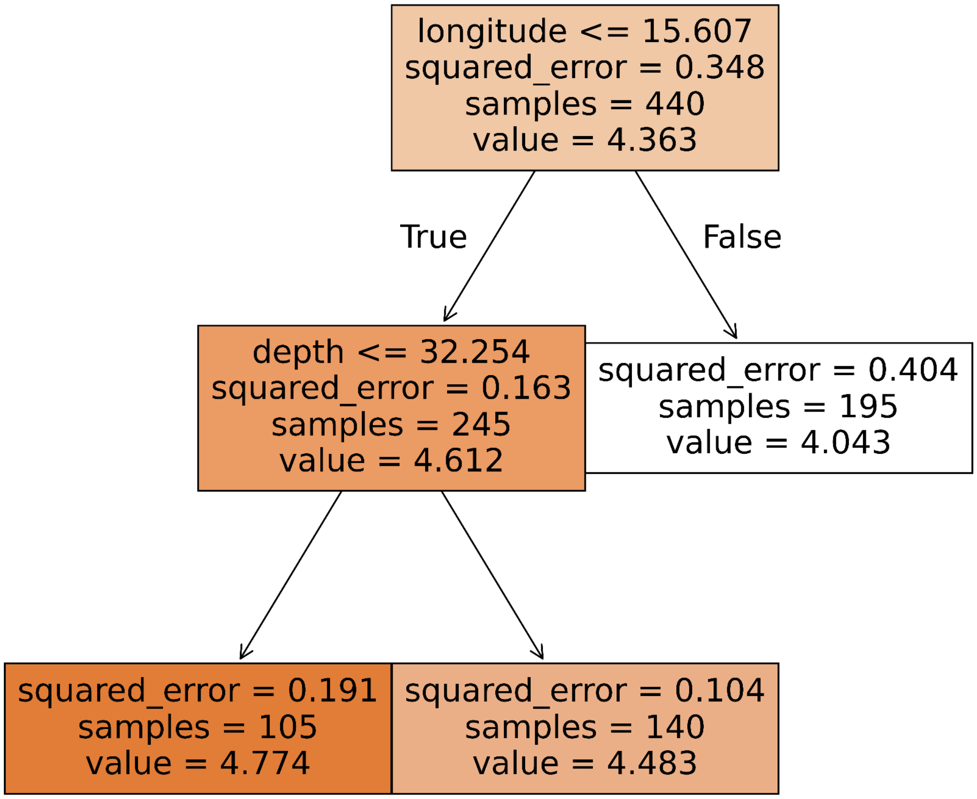
Machine Learning

When deciding how to build our prediction model, we had the choice of using a more statistics-based machine learning algorithm or one that was based on deep learning and neural networks. We ultimately chose the Random Forest (RF) algorithm, a statistical ML technique, as we valued explainability over raw accuracy. A deep neural network may have resulted in higher predictive accuracy but our ability to interpret what factors result in larger magnitude earthquakes would be impaired.

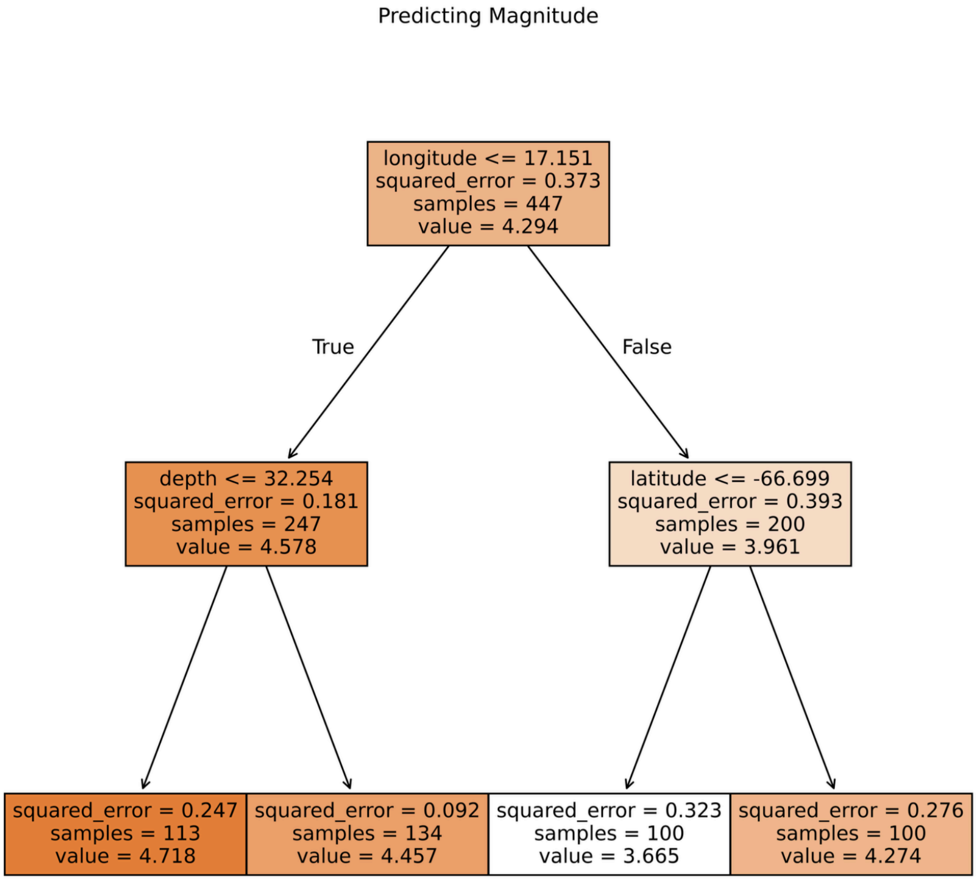
Once we settled on a random forest approach, we also had to tweak a few hyperparameters. We chose to use 100 individual decision trees in our random forest with the minimum number of samples in each life of the tree being 100 as well. We landed upon this configuration as we wanted a balance between complexity and accuracy. A random forest with a smaller minimum leaf size would result in a more complex, highly nested tree structure that may result in greater accuracy, but also lead to irrelevant or uninterpretable splits. Additionally, more decision trees/estimators helps improve accuracy while being generalizable, which helps prevent overfitting.

In our experiment, we ran two random forests. One random forest regressor was fit on the training data **without** the added ‘distance to tectonic plate’ feature mentioned earlier in *Feature Engineering*. We took this approach to determine the effect of the feature on the random forest’s error. Without this feature present, the random forest with 100 estimators and a minimum leaf size of 100 samples achieved a Mean Squared Error (MSE) of 0.1984. A random forest with the same configuration fitted on the training data **with** the distance feature achieved a MSE of 0.1951, which is a slight improvement. Although we saw this slight improvement, exploring one of the estimators of these random forests gives us a better picture of the effects of each feature on the final predicted magnitude.

Predicting Magnitude with Distance to Tectonic Plate



For the model with the distance to tectonic plate feature, the estimator actually used longitude and depth as the primary features for explaining magnitude. We see the same phenomenon for the model without the distance to tectonic plate feature.



Even though both estimators **did not** use the distance to tectonic plate feature, the variable did affect where the estimator made each split. For example, without the feature, the estimator made its initial split on longitude = 15.607. With the distance feature, the estimator made the split at 17.151.

Conclusion

Understanding earthquakes and why they occur is vital in humanity's efforts to prepare for these natural disasters. Each year thousands of people die from earthquakes and many predict these natural phenomena will become more frequent over the next few decades. Analytical models like these can help scientists assess what causes high magnitude earthquakes and what regions in the world are the most susceptible to them. We explored a miniature scale experiment of earthquake analysis and understanding using the limited dataset and simple RF model at our disposal. We believe that future research using more vast and abundant data with technically superior model architectures can help the public truly understand these natural phenomena and support governmental action and preparedness all over the world.