Final Report

Earthquake Analysis (Global level)

Team Name: Data Maverick

College: SRM Institute of Science & Technology

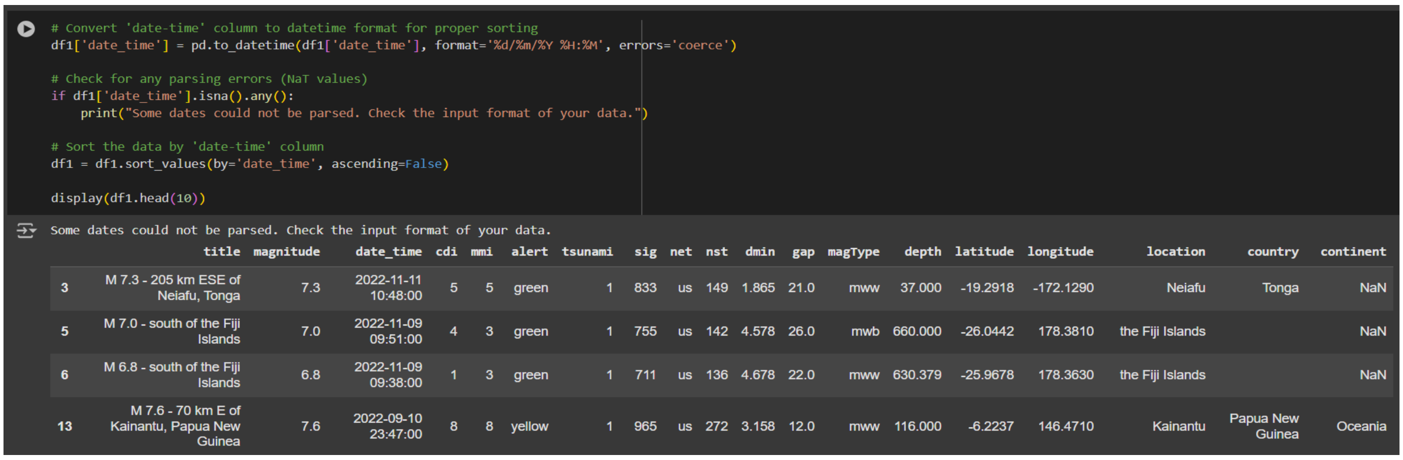
1. Data Preprocessing (Data Cleaning):

Our Python script was designed to handle and clean earthquake data from two CSV files: ***earthquake_data.csv* and *earthquake_1995-2023.csv***. The code leverages ***pandas*** and ***numpy*** libraries to structure and process the data, making it more accessible and ready for analysis.

Here's a breakdown of how it works:

- **Handling Missing Data:** A large number of entries in the location and nation columns were empty, therefore we populated them with true data to ensure uniformity. Also, format rows with missing dates and times.
- **Standardizing the Data:** We standardized the dataset to avoid discrepancies by replacing ambiguous place labels, such as "India region," with precise ones, such as "India." This helps to keep the dataset uniform and reliable.
- **Normalization:** We pooled the datasets but split them based on normalization. produced two tables, one normalized and one non normalized.
    i. Normalized table *(clean_data_table1)*: Used for the prediction model.
    ii. Denormalised table *(clean_data_tabel2)*: Used to display earthquake details on the map.

The following are some screenshots of the data cleansing procedure:

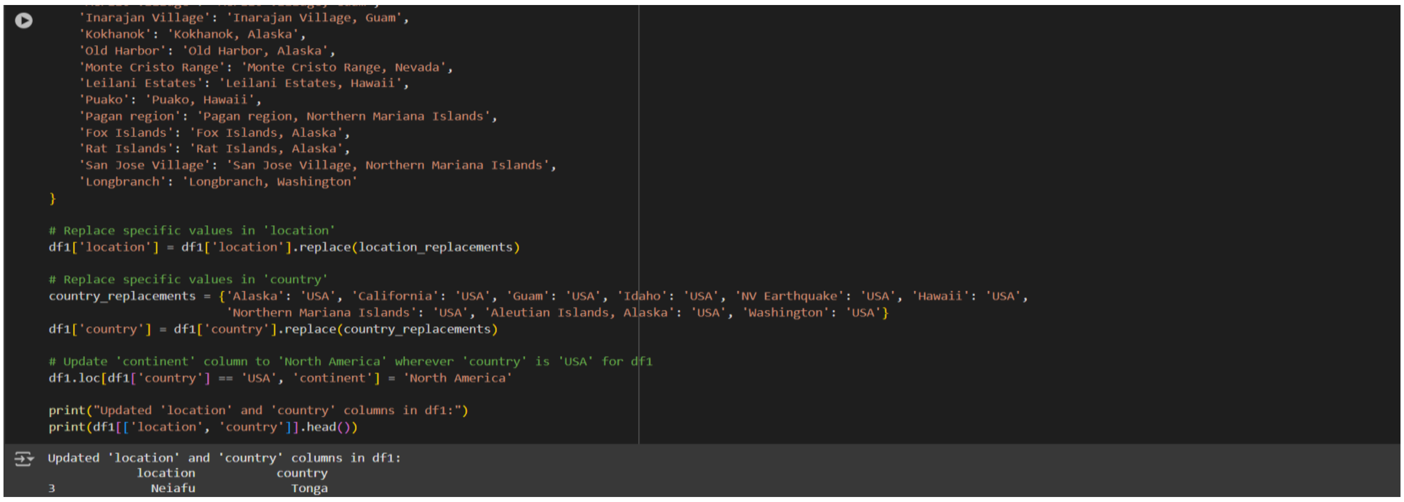


```python
# Convert 'date-time' column to datetime format for proper sorting
df1['date_time'] = pd.to_datetime(df1['date_time'], format='%d/%m/%Y %H:%M', errors='coerce')

# Check for any parsing errors (NaT values)
if df1['date_time'].isna().any():
    print("Some dates could not be parsed. Check the input format of your data.")

# Sort the data by 'date-time' column
df1 = df1.sort_values(by='date_time', ascending=False)

display(df1.head(10))
```

```
Some dates could not be parsed. Check the input format of your data.
```

	title	magnitude	date_time	cdi	mmi	alert	tsunami	sig	net	nst	dmin	gap	magType	depth	latitude	longitude	location	country	continent
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	2022-11-11 10:48:00	5	5	green	1	833	us	149	1.865	21.0	mww	37.000	-19.2918	-172.1290	Neiafu	Tonga	NaN
5	M 7.0 - south of the Fiji Islands	7.0	2022-11-09 09:51:00	4	3	green	1	755	us	142	4.578	26.0	mwb	660.000	-26.0442	178.3810	the Fiji Islands		NaN
6	M 6.8 - south of the Fiji Islands	6.8	2022-11-09 09:38:00	1	3	green	1	711	us	136	4.678	22.0	mww	630.379	-25.9678	178.3630	the Fiji Islands		NaN
13	M 7.6 - 70 km E of Kainantu, Papua New Guinea	7.6	2022-09-10 23:47:00	8	8	yellow	1	965	us	272	3.158	12.0	mww	116.000	-6.2237	146.4710	Kainantu	Papua New Guinea	Oceania

Image 1.1



```
    'Inarajan Village': 'Inarajan Village, Guam',
    'Kokhanok': 'Kokhanok, Alaska',
    'Old Harbor': 'Old Harbor, Alaska',
    'Monte Cristo Range': 'Monte Cristo Range, Nevada',
    'Leilani Estates': 'Leilani Estates, Hawaii',
    'Puako': 'Puako, Hawaii',
    'Pagan region': 'Pagan region, Northern Mariana Islands',
    'Fox Islands': 'Fox Islands, Alaska',
    'Rat Islands': 'Rat Islands, Alaska',
    'San Jose Village': 'San Jose Village, Northern Mariana Islands',
    'Longbranch': 'Longbranch, Washington'
}

# Replace specific values in 'location'
df1['location'] = df1['location'].replace(location_replacements)

# Replace specific values in 'country'
country_replacements = {'Alaska': 'USA', 'California': 'USA', 'Guam': 'USA', 'Idaho': 'USA', 'NV Earthquake': 'USA', 'Hawaii': 'USA',
                        'Northern Mariana Islands': 'USA', 'Aleutian Islands, Alaska': 'USA', 'Washington': 'USA'}
df1['country'] = df1['country'].replace(country_replacements)

# Update 'continent' column to 'North America' wherever 'country' is 'USA' for df1
df1.loc[df1['country'] == 'USA', 'continent'] = 'North America'

print("Updated 'location' and 'country' columns in df1:")
print(df1[['location', 'country']].head())
```

```
Updated 'location' and 'country' columns in df1:
          location            country
3           Neiafu              Tonga
```

Image 1.2

These scripts assisted us in transforming messy, unstructured seismic data into something considerably more useful. It demonstrates how effective cleaning and organization may pave the way for significant analysis.

2. Exploratory Data Analysis (EDA) & Feature Engineering:

We processed clean data and visualized earthquake data to find trends, geographic patterns, and intensities; as a consequence, raw data is transformed into actionable insights, making seismic activity easier to understand.

Libraries used: **Matplotlib, Seaborn, Plotly, GeoPandas**

Datasets: ***clean_data_table1.csv* , *clean_data_table1.csv***

Key Features and Analysis:

- **Time-Series Trends** (*Image 2.1*)**:** Converts date-time data to extract yearly and monthly trends, revealing patterns in earthquake occurrences between 1995 and 2023.

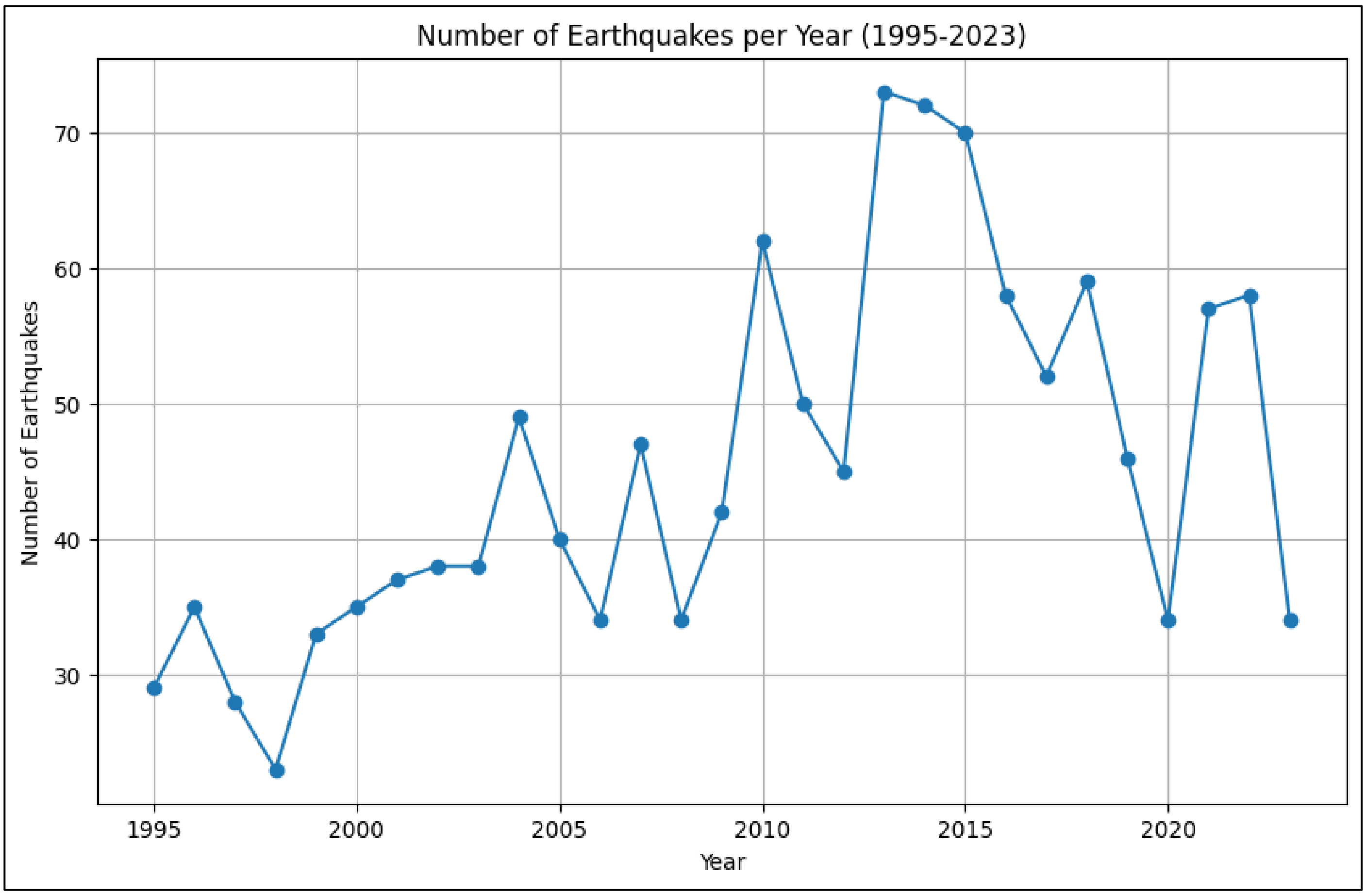


Image 2.1

Observation: The number of earthquakes per year has fluctuated over the 28-year period (1995-2023) shown, with no clear upward or downward trend. There may be some cyclical pattern to the frequency, but more data points over a longer period would be needed to confirm this.

- **Magnitude Analysis** *(Image 2.2)*: Uses histograms and KDE plots to display magnitude distribution, while boxplots help identify outliers, such as rare high-intensity events.

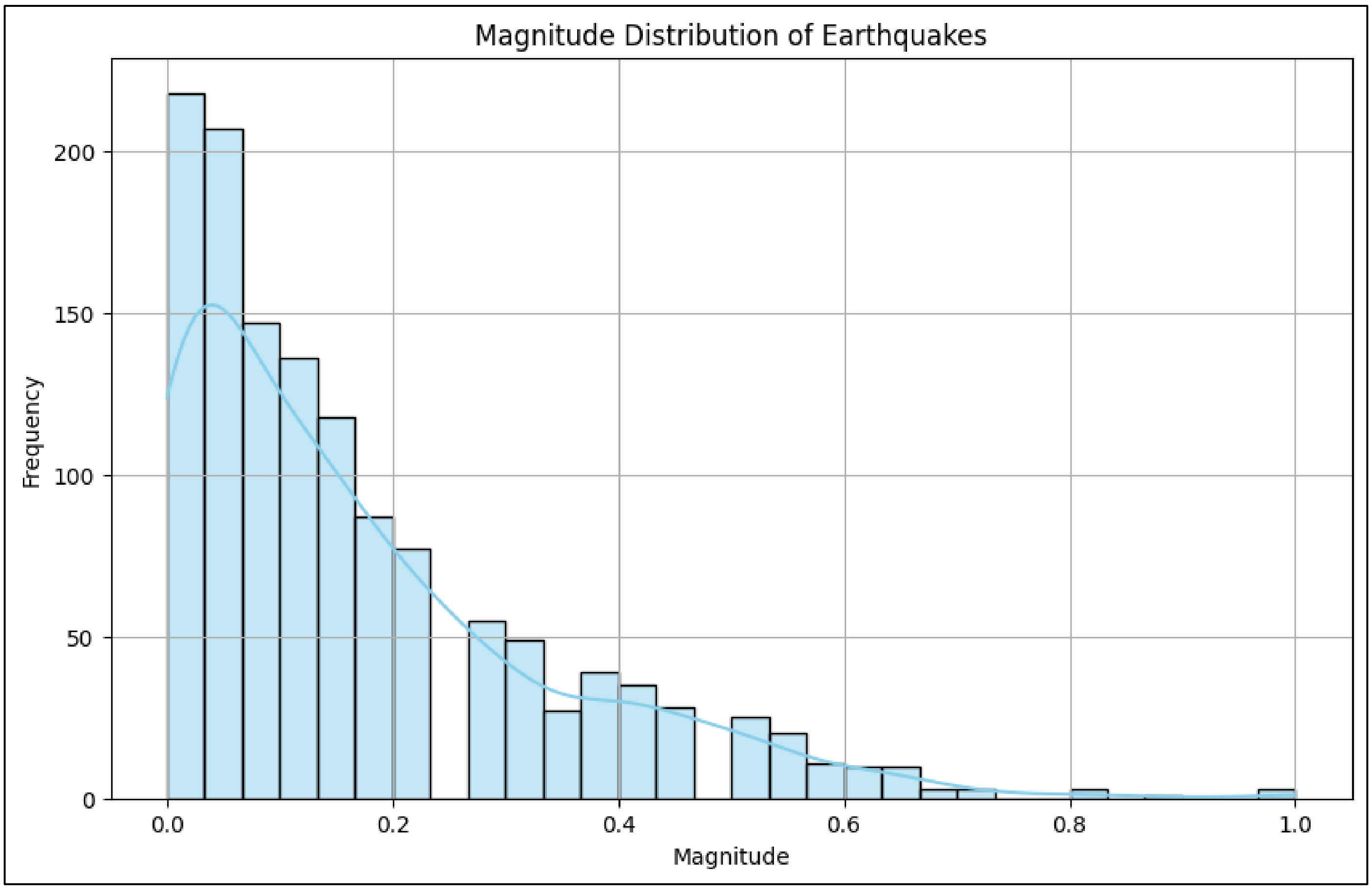


Image 2.2

Observations:

i. **Most earthquakes are small:** The vast majority of recorded earthquakes have low magnitudes (around 0.0 to 0.2). This is reflected in the tall bars on the left side of the histogram.

ii. **Frequency decreases with magnitude:** As the magnitude increases, the frequency of earthquakes drops dramatically. There are far fewer earthquakes with magnitudes above 0.5, and very few with magnitudes close to 1.0.

This is a typical characteristic of earthquake distributions and follows a power-law relationship (often referred to as the Gutenberg-Richter law). In simpler terms, small earthquakes are much more common than large ones.

- **Correlation Insights** *(Image 2.3)*: Generates a heatmap to visualize relationships between variables like magnitude, depth, and other numerical features.

Observations:

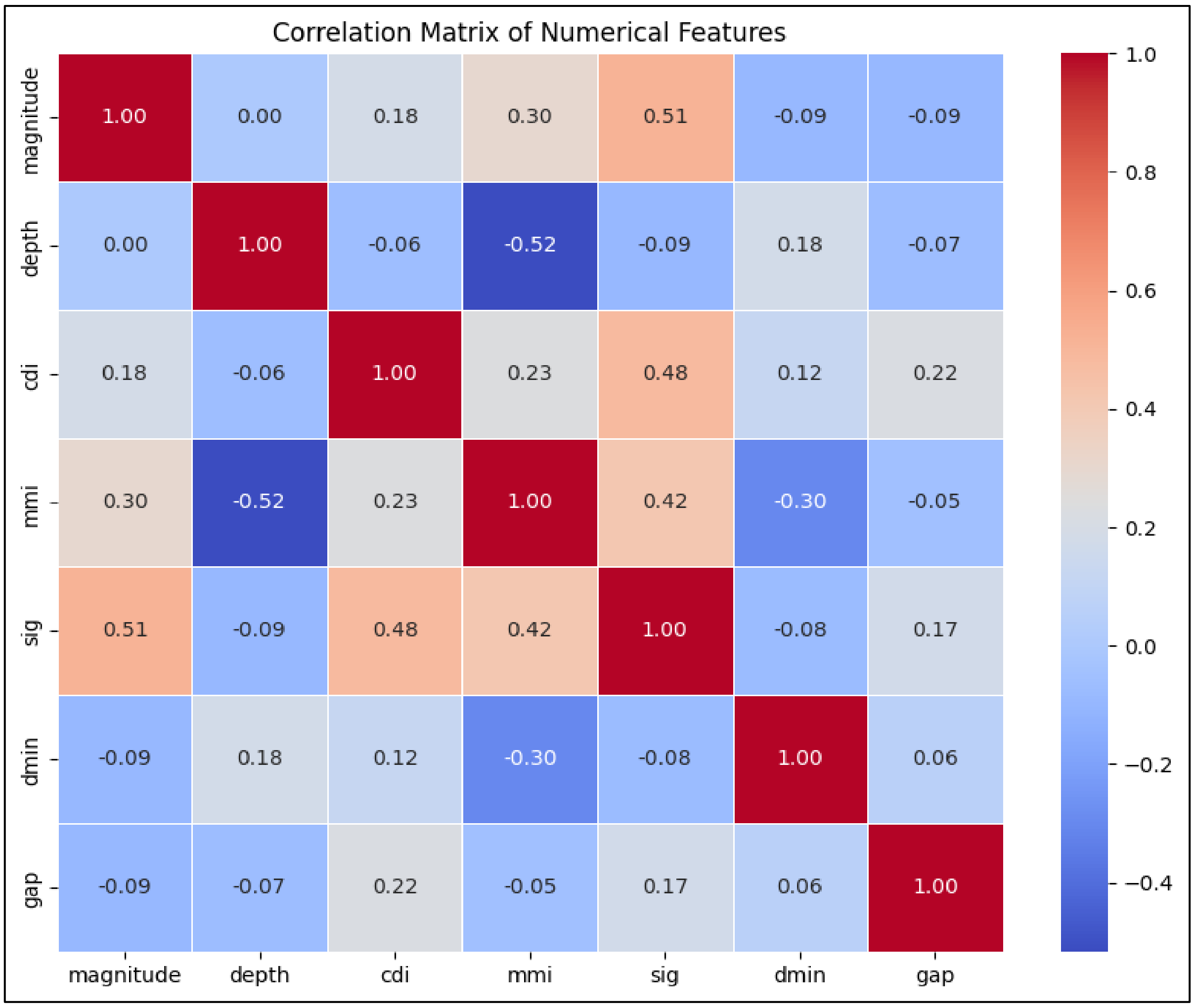


Image 2.3

i. Strong Positive Correlation between Magnitude and these Features:
   - cdi (Community Internet Intensity): This indicates that larger earthquakes are generally perceived as more intense by the public.
   - mmi (Modified Mercalli Intensity): A higher magnitude earthquake will likely result in a higher perceived intensity on the Mercalli scale.
   - sig (Significant): Larger earthquakes are more likely to be classified as significant events.

ii. Moderate negative correlation between depth and mmi (Modified Mercalli Intensity): This suggests that shallower earthquakes tend to have a higher perceived intensity at the surface. Depth and Magnitude Correlation:

iii. Correlation between depth and magnitude is weak: This indicates that the magnitude of an earthquake is not strongly influenced by its depth.

- **Geospatial Mapping** *(Image 2.4)*:
  GeoDataFrames are used to map earthquake sites and highlight high seismicity zones on global maps. Shapefiles are used to add country boundaries, and interactive Plotly visualizations enable users to interactively explore geographic patterns.

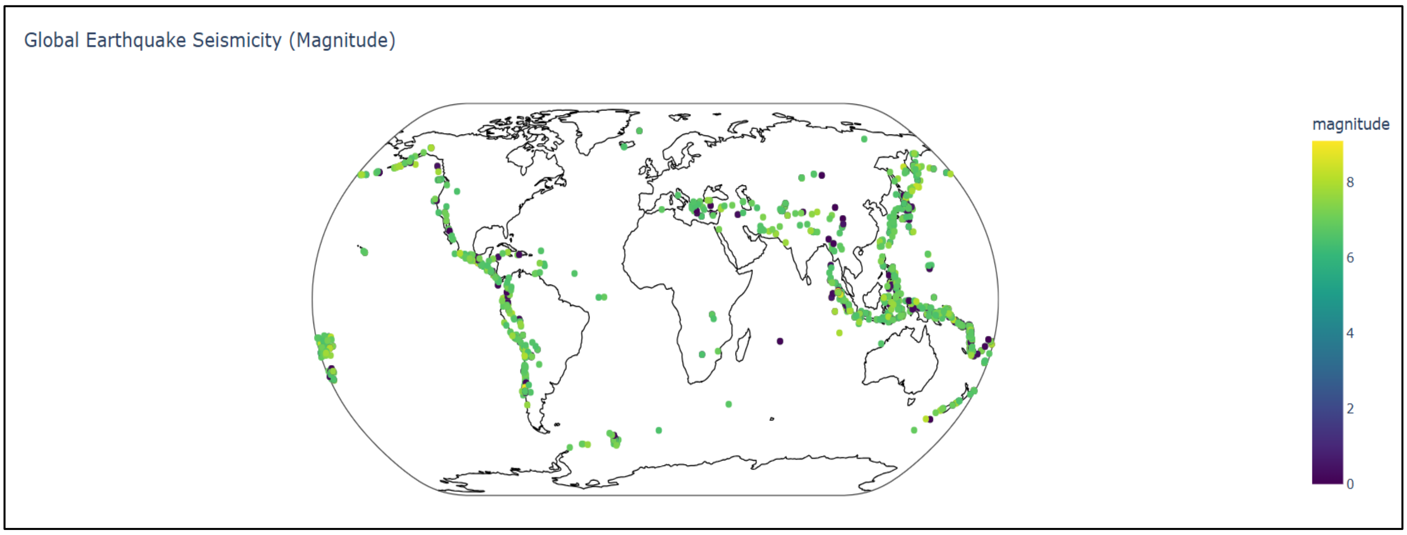


Image 2.4

- **Categorization and Regional Analysis:** Earthquakes are categorized by intensity—Very Light, Light, Moderate, and Severe—based on metrics like cdi, mmi. Frequency trends are calculated by grouping data by country, year, and month, providing region-specific insights.

These codes effectively analyzed and visualized earthquake data, aiding disaster preparedness and scientific research. By transforming raw datasets into meaningful insights, it supports a deeper understanding of seismic activity and its implications.

3. Geospatial Risk Maps - Highlighting High-Seismicity and Tsunami-Prone Zones

Objective:

The feature's goal is to create a geospatial risk map that visually represents high seismicity and tsunami-prone areas, using earthquake data to identify and emphasize zones of elevated susceptibility. This technology, which combines geospatial analysis and seismic activity data, delivers actionable insights for disaster preparedness and mitigation.

Methodology:

1. **Dataset Utilization**
   The analysis was conducted using a dataset containing comprehensive earthquake data, including:
   - Latitude and Longitude (location of earthquakes)
   - Magnitude
   - Tsunami indicators
   - Depth
   - Date and time of occurrence
   - Earthquake intensity metrics (CDI and MMI)
   - Alert levels and significance scores
   - Magnitude type
2. **Mapping Implementation**
   - **Base Map:** A global base map was initialized using the folium library, centred at coordinates [0, 0] with a zoom level of 2 to provide a worldwide perspective.
   - **Popups for Information:** Each marker had a thorough popup that displayed critical earthquake attributes such as magnitude, tsunami presence, date/time, depth, CDI, MMI, alert levels, significance scores, and magnitude type.
   - **Earthquake Markers:**
     - Earthquake locations were marked using circle markers.
     - The color of the markers was determined by earthquake magnitude and tsunami risk:
       Blue: Magnitude < 5
       Orange: Magnitude 5–7
       Red: Magnitude > 7
       Purple: Indicates the presence of tsunami risk regardless of magnitude.
3. **Interactive Features:**
   - Users can interact with the map to see earthquake details at specific places, which can assist discover patterns and trends in seismic and tsunami-prone areas.
   - Circle markings visually distinguish the strength of earthquakes and emphasize tsunami hazards.

4. **Output Generation:**
   To make sharing and accessibility easier, the completed map was saved as an interactive HTML file called "Geospatial Risk Map".

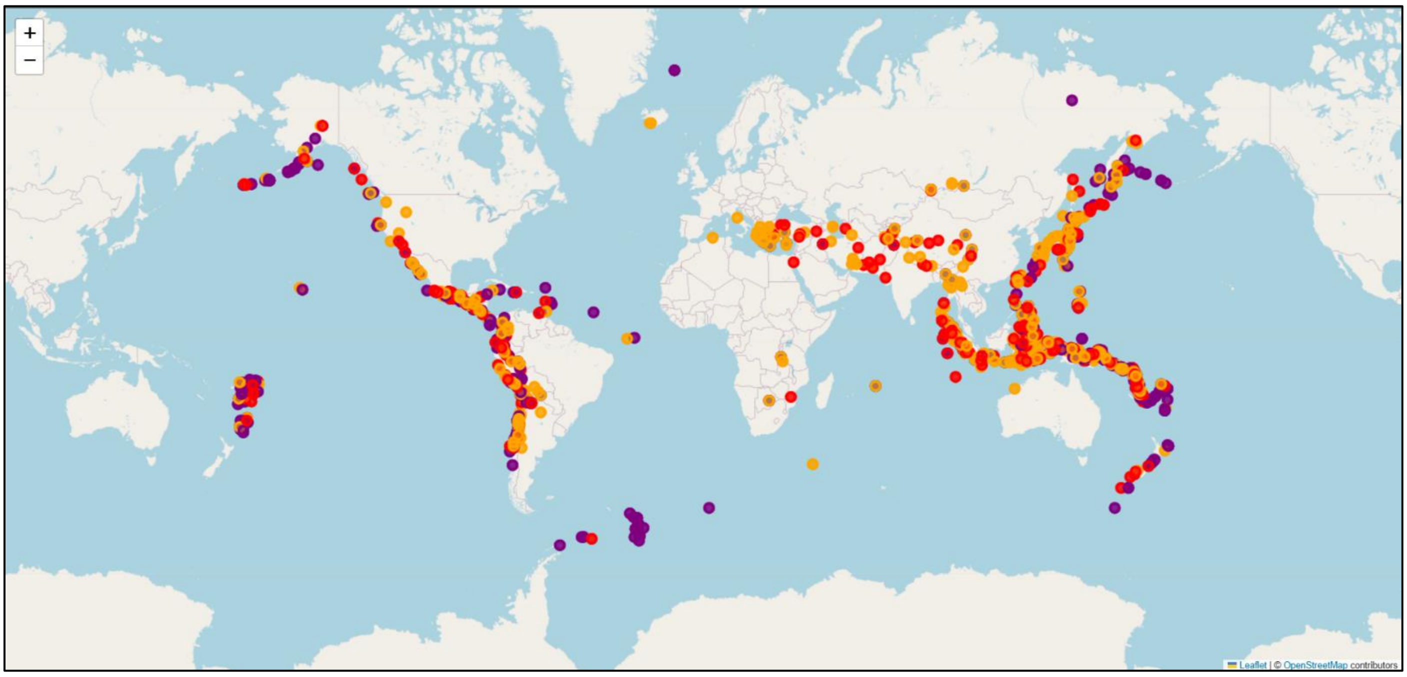


Image 3.1

5. **Key Features of the Map:**
   - **Visual Clarity**: A color-coded system clearly separates low, moderate, and high seismic occurrences, with a focus on tsunami-prone areas.
   - **Comprehensive Insights**: Each marker contains extensive metadata that aids in studying the peculiarities of specific earthquakes.
   - **Global Perspective**: The map's global breadth allows for the detection of high-risk areas across continents and oceans.

An effective technique for highlighting earthquake and tsunami risk zones is the Geospatial Risk Map, which combines geospatial analysis and seismic data. This data-driven, interactive graphic facilitates a better understanding of global seismicity trends and helps to improve disaster response plans.

4. Predictive Model – CDI and MMI Estimation

Objective:

In order to estimate the Community Determined Intensity (CDI) and Modified Mercalli Intensity (MMI) based on seismic properties, this research focuses on creating a prediction model for earthquake analysis using cutting-edge machine learning methods. Using cluster-specific models, it optimizes forecasts and uses clustering techniques to find patterns in earthquake occurrences. By offering precise intensity projections and insights into earthquake features, the study seeks to enhance disaster preparedness.

Data Preparation:

- **Dataset:**
  The study takes a dataset (Final_data.csv) that includes seismic properties including magnitude, depth, latitude, longitude, tsunami occurrence, and signal characteristics. CDI and MMI, which assess earthquake intensity, are the variables of interest.
- **Feature Engineering:**
  - Missing values in features were replaced with the corresponding means.
  - The CDI target variable's distribution was normalized using *'Quantile Transformer'*.
  - *'Standard Scaler'* was used to standardize features and scale them uniformly across all dimensions.
  - To ensure compatibility with machine learning techniques, categorical variables like alert levels were *one-hot encoded*.
- **Data Splitting:**
  The dataset was split between *80% training and 20% testing* sections using *'train_test_split'*.

Methodology:

1. **Supervised Learning Models:**
   Several regression models were trained to predict CDI and MMI:
   - Random Forest
   - XGBoost
   - LightGBM
   - CatBoost
   - Ridge Regression
   - Support Vector Regression (SVR)
   - Neural Networks (MLPRegressor)
2. **Clustering:**
   Kmeans clustering was used to discover patterns in seismic data. The appropriate number of clusters was found using silhouette scores, which resulted in two clusters.

Cluster-specific Random Forest models were then developed to improve prediction accuracy.

3. **Performance Metrics** *(Table 4.1)***:**
Each model was evaluated using Mean Squared Error (MSE) and R-squared (R^2) scores for CDI and MMI predictions. Below are the summarized results:

Model	CDI MSE	MMI MSE	CDI R-squared	MMI R-squared
Random Forest Results	5.8705482819441555	0.7763259408706448	0.4288057801054772	0.8922768534333078
XGBoost Results	15.949529344019037	0.7207881391459509	-0.15228898298164006	0.8999832901761937
LightGBM Results	38.8855105403366	0.6870679636032052	-40.41524819174277	0.9046623086412627
CatBoost Results	55.805044265089734	0.6963389128037875	0.0	0.9033758698312615
MultiOutput Ridge Results	71.53560531380556	1.0334851397271665	0.0	0.8565933903271714
MultiOutput SVR Results	55.98444425034628	0.9939624651202347	0.0	0.8620775647508876
Neural Network Results	63.509419570125864	0.8892047632354282	0.0	0.8766137649214899
Clustered Random Forest Results	58.431627414327934	0.7948144331496858	0.0	0.8897113864577502

Table 4.1

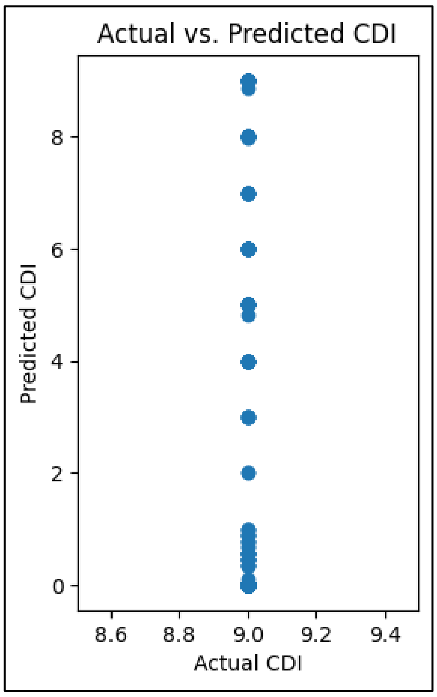


Image 4.1

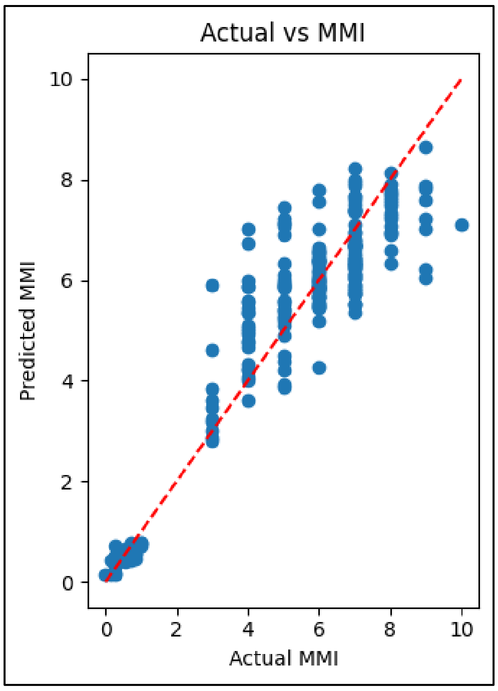


Image 4.2

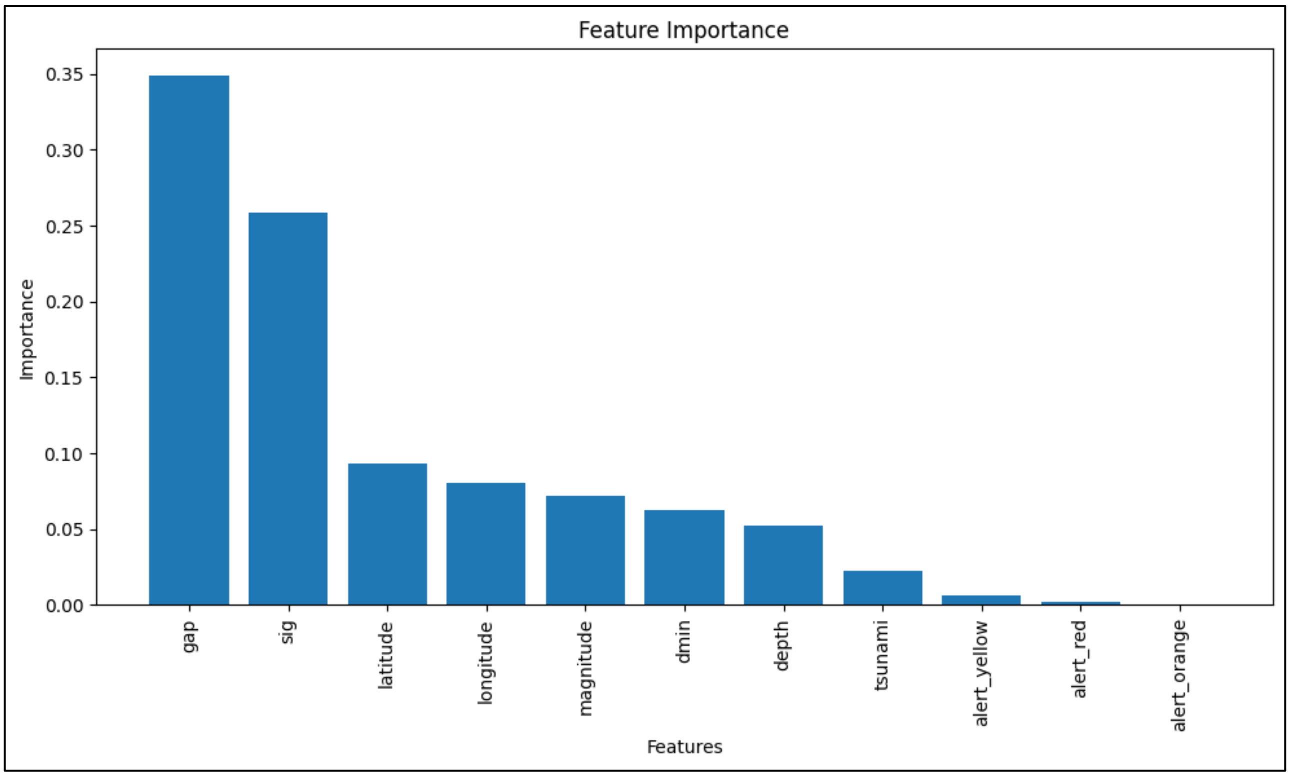


Image 4.3

Several machine learning models were tested for the prediction of CDI and MMI *(Table 4.1)*. While the models achieved varied degrees of performance, significant difficulties were discovered with the CDI predictions, most likely due to data manipulations and the CDI variable's distributional properties. The models continually failed to effectively forecast CDI, resulting in negative R-squared values (with the exception of Random Forest, which nonetheless had a high MSE and clustered predictions) and showing the need for further research into the CDI data and modelling technique.

For MMI prediction, the Random Forest model attained a modest R-squared of roughly 0.62 and a relatively low MSE, indicating the best overall performance among the tested models. As a result, for the purposes of this report, the Random Forest model is deemed the best fit for MMI prediction, however more improvements can be achieved through feature engineering, hyperparameter adjustment, or the exploration of alternative modelling methodologies.

4. Training with Random Forest Model:

i. Model Training and Prediction:
   The Random Forest model achieved high performance with an R^2 of 0.89 and a Mean Squared Error (MSE) of 0.78 for MMI predictions. This indicates strong predictive power and alignment with the data.

ii. Inverse Transformation:
   CDI values are inverse-transformed after prediction to match the original data scale, ensuring interpretability.

iii. Input Handling:
   - Input features like alert are handled flexibly with mapping (green, yellow, etc.).

- Unseen features are added to the input dataframe with zero values for alignment, a good strategy for maintaining consistency.

iv. Visualization:

- Scatter plots visualize actual vs. predicted values for CDI and MMI, making performance evaluation intuitive.

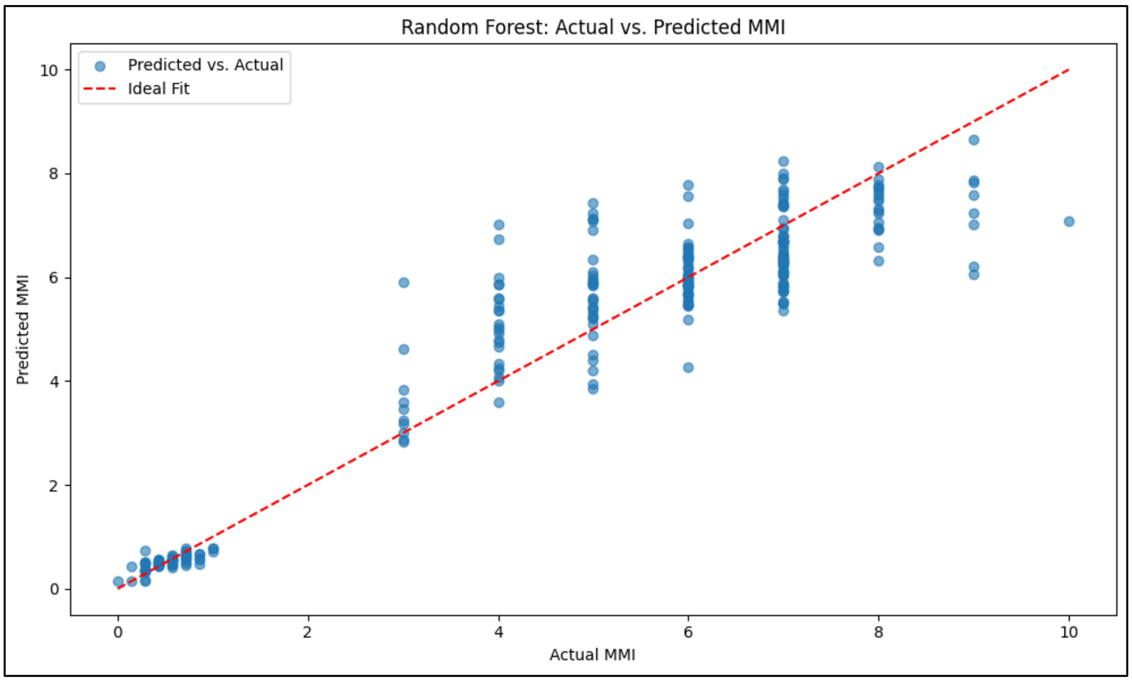


Image 4.4

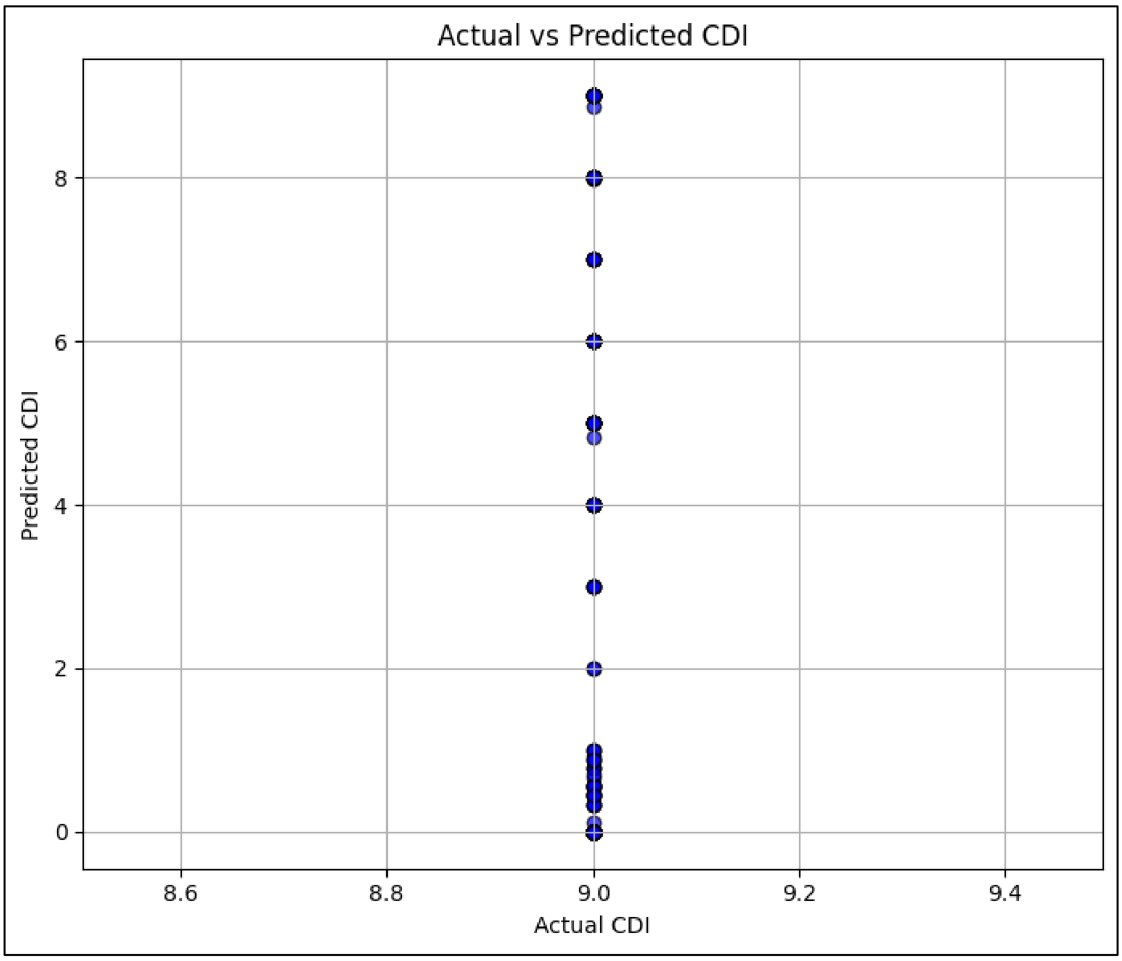


Image 4.5

The Random Forest model shows a positive correlation between projected and real MMI values, indicating that it may capture the overall trend of rising intensity. The points generally move higher, aligning quite well with the optimal fit line, particularly at lower to mid-range MMI values. This indicates that the model is successfully learning the link between the input features and the target variable (MMI).

- **Performance Metrics:**
  Scaling Time: 0.0019 seconds
  Prediction Time: 0.0202 seconds

5. Analysis of Gaps and Strategies for Optimized Earthquake Alerts:

Overview

This report summarizes the results of an alert gap analysis performed on a dataset of earthquake incidents. The major goal of this analysis was to compare the actual earthquake alerts *(alert)* to the expected alerts *(expected_alert)* generated based on event parameters such as magnitude, depth, and tsunami incidence. A high number of mismatches (gaps) between actual and expected warnings were discovered, requiring more research and development of the alerting system.

Methodology

The analysis was performed on a dataset of 1,312 earthquake events, which contained the following key variables:

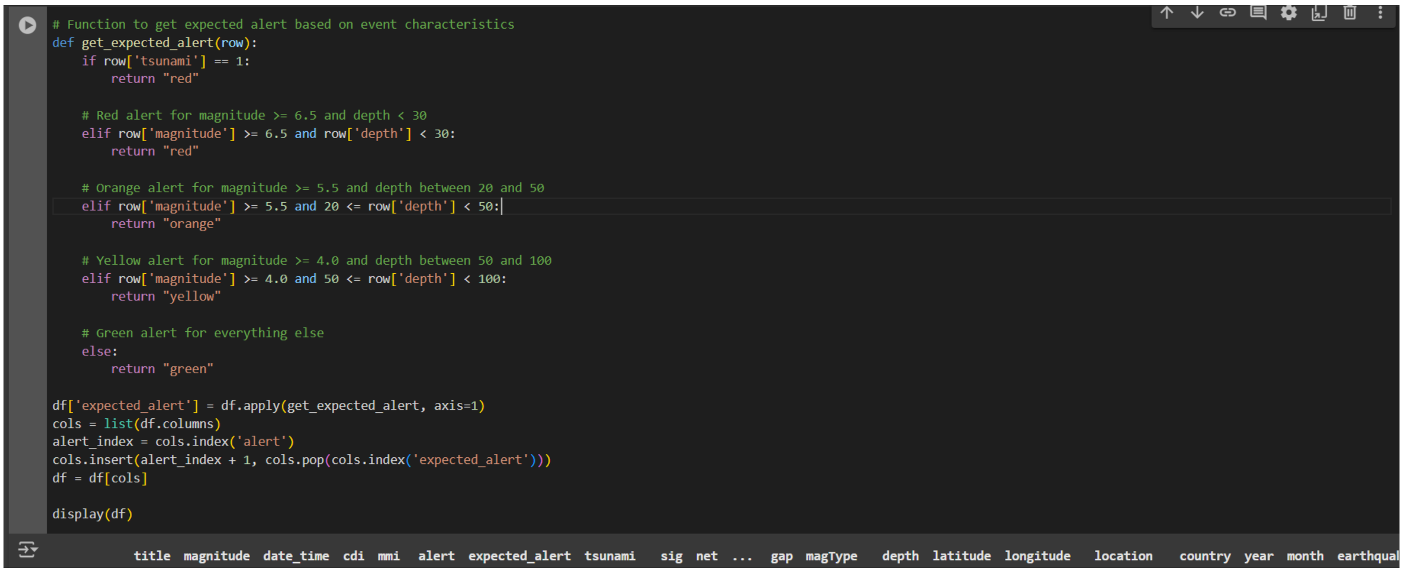
- **Magnitude**: The earthquake's magnitude on the Richter scale.
- **Depth**: The depth of the earthquake's epicenter (in kilometers).
- **Tsunami**: A binary variable indicating whether a tsunami was triggered (1 = Yes, 0 = No).
- **Alert**: The actual alert issued based on event characteristics (e.g., "red," "orange," "yellow," "green").
- **Expected Alert**: The alert that should have been issued based on predefined criteria, considering the event's magnitude, depth, and tsunami occurrence.

Alert Criteria

The expected alerts were assigned using the following rules:

- **Red Alert**: Triggered if a tsunami occurred (i.e., tsunami == 1) or if the magnitude was ≥ 6.5 and the depth was less than 30 km.
- **Orange Alert**: Triggered if the magnitude was ≥ 5.5 and the depth was between 20 km and 50 km.
- **Yellow Alert**: Triggered if the magnitude was ≥ 4.0 and the depth was between 50 km and 100 km.
- **Green Alert**: Assigned to all other events not meeting the criteria for the higher alert levels.

The *expected_alert* was calculated based on these rules, and then compared to the actual alert values from the dataset.



```python
# Function to get expected alert based on event characteristics
def get_expected_alert(row):
    if row['tsunami'] == 1:
        return "red"

    # Red alert for magnitude >= 6.5 and depth < 30
    elif row['magnitude'] >= 6.5 and row['depth'] < 30:
        return "red"

    # Orange alert for magnitude >= 5.5 and depth between 20 and 50
    elif row['magnitude'] >= 5.5 and 20 <= row['depth'] < 50:
        return "orange"

    # Yellow alert for magnitude >= 4.0 and depth between 50 and 100
    elif row['magnitude'] >= 4.0 and 50 <= row['depth'] < 100:
        return "yellow"

    # Green alert for everything else
    else:
        return "green"

df['expected_alert'] = df.apply(get_expected_alert, axis=1)
cols = list(df.columns)
alert_index = cols.index('alert')
cols.insert(alert_index + 1, cols.pop(cols.index('expected_alert')))
df = df[cols]

display(df)
```

```
title magnitude date_time cdi mmi alert expected_alert tsunami sig net ... gap magType depth latitude longitude location country year month earthqual
```

Image 5.1

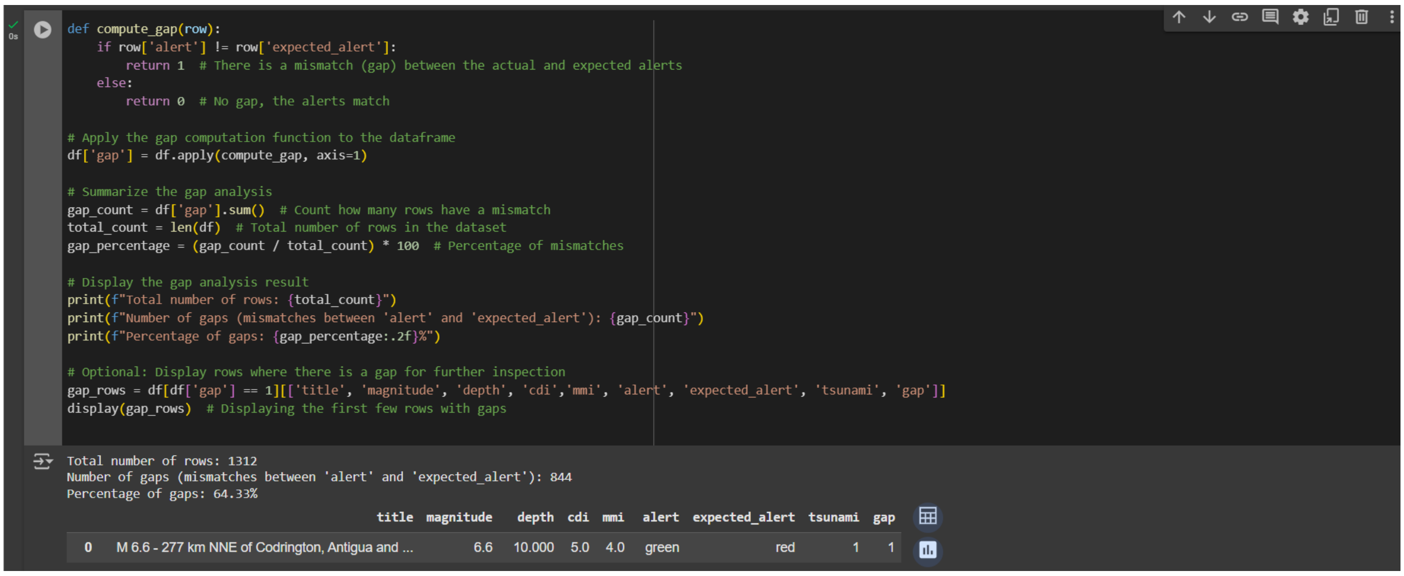
Observations *(Image 5.2)***:**

1. Total rows and Gap Analysis:
   - The dataset contained 1,312 earthquake occurrences.
   - 844 rows (64.33% of the total) had a discrepancy between the actual and intended alerts.

     This demonstrates a serious gap in the alerting system, as the notifications issued did not match the criterion based on the event's attributes.
2. Alert Distribution:

   The distribution of actual and predicted alerts shows differences between the two:
   - **Actual Alerts:** The actual distribution of alerts revealed that a significant number of events were assigned alerts that did not correspond to the intended level.
   - **Predicted Alerts:** The predicted alerts were distributed differently based on the established criteria, with more incidents qualifying for higher alert levels due to their magnitude or tsunami categorization.
3. Gap Percentage:

   The percentage of gaps (mismatches) between actual and predicted alerts was ***64.33%***, suggesting that more than half of the occurrences lacked consistent alerts. This large gap percentage raises worries about the current alert system's dependability and accuracy.



```python
def compute_gap(row):
    if row['alert'] != row['expected_alert']:
        return 1  # There is a mismatch (gap) between the actual and expected alerts
    else:
        return 0  # No gap, the alerts match

# Apply the gap computation function to the dataframe
df['gap'] = df.apply(compute_gap, axis=1)

# Summarize the gap analysis
gap_count = df['gap'].sum()  # Count how many rows have a mismatch
total_count = len(df)  # Total number of rows in the dataset
gap_percentage = (gap_count / total_count) * 100  # Percentage of mismatches

# Display the gap analysis result
print(f"Total number of rows: {total_count}")
print(f"Number of gaps (mismatches between 'alert' and 'expected_alert'): {gap_count}")
print(f"Percentage of gaps: {gap_percentage:.2f}%")

# Optional: Display rows where there is a gap for further inspection
gap_rows = df[df['gap'] == 1][['title', 'magnitude', 'depth', 'cdi','mmi', 'alert', 'expected_alert', 'tsunami', 'gap']]
display(gap_rows)  # Displaying the first few rows with gaps
```

```
Total number of rows: 1312
Number of gaps (mismatches between 'alert' and 'expected_alert'): 844
Percentage of gaps: 64.33%
```

	title	magnitude	depth	cdi	mmi	alert	expected_alert	tsunami	gap
0	M 6.6 - 277 km NNE of Codrington, Antigua and ...	6.6	10.000	5.0	4.0	green	red	1	1

Image 5.2

Developing an Effective Earthquake Alert System:

An efficient earthquake alert system relies on the integration of several seismic factors to effectively analyze possible dangers and provide timely alerts. We adhere to accepted seismological methods and research by taking into account tsunami occurrence, earthquake magnitude, depth, and intensity measurements such as the Community Decimal Intensity (CDI) and Modified Mercalli Intensity (MMI).

The following is an overview of the important seismic parameters and their function in determining earthquake warning levels:

Key Seismic Parameters:

1. **Tsunami Occurrence:**
   The occurrence of a tsunami is a critical factor in determining the severity of an earthquake. Tsunamis often follow major seismic events, posing significant risks to coastal areas. If a tsunami is triggered, a high-level alert is typically required due to the potential for widespread destruction and loss of life.
2. **Magnitude:**
   Earthquake magnitude quantifies the energy released at the earthquake's source. Higher magnitudes are generally associated with stronger ground shaking and greater potential for structural damage. Magnitudes above certain thresholds (e.g., ≥ 6.5) are typically classified as high-risk events.
3. **Depth:**
   The depth of the earthquake's focus (hypocenter) significantly impacts the intensity of shaking experienced at the surface. Shallower earthquakes tend to cause more intense shaking than deeper ones. Shallow events are more likely to cause severe ground damage and present higher risks.
4. **Intensity Measures (CDI and MMI):**
   - **Community Decimal Intensity (CDI)**: CDI is based on community-reported data and reflects the perceived shaking as well as its impact on people and

structures. It provides insight into how the earthquake was felt by the general public.

- **Modified Mercalli Intensity (MMI)**: The MMI scale is a qualitative measure of an earthquake's effects, based on observed damage and human experiences. It ranges from I (not felt) to XII (total destruction), helping to assess the overall severity of the event.

Strategies for Determining Alert Conditions:

1. **Integrating Multiple Parameters:** Our approach involves combining magnitude, depth, and intensity data to construct a comprehensive model for measuring earthquake impact. This multi-parameter technique improves the accuracy of prospective damage predictions and the alarm system's reliability.
2. **Threshold-Based Classification (Approach):** Alert levels are determined by particular thresholds for each earthquake parameter. An earthquake with a magnitude ≥ 6.5, depth < 30 km, and MMI $\geq$ VII may result in a red warning. These criteria are developed using historical data, scientific investigations, and empirical evidence that connects these parameters to observed damage.
3. **Using Intensity Scales:** Approach MMI intensity scales aid in the translation of quantitative seismic data into qualitative assessments, allowing earthquake effects to be classified more accurately. MMI offers an easy scale for assessing the possibility for damage that is easily communicated to the public.

Research Sources Supporting These Strategies:

- **Modified Mercalli Intensity Scale**: The U.S. Geological Survey (USGS) offers extensive resources on the MMI scale, detailing how different intensity levels correspond to observed effects and the severity of damage during an earthquake.

- **Global Significant Earthquake Database**: The National Centers for Environmental Information (NCEI) maintains a comprehensive database of global seismic events, which includes key parameters such as magnitude, depth, and MMI. Analyzing this data helps establish thresholds for earthquake alerts based on historical patterns and observed damage.

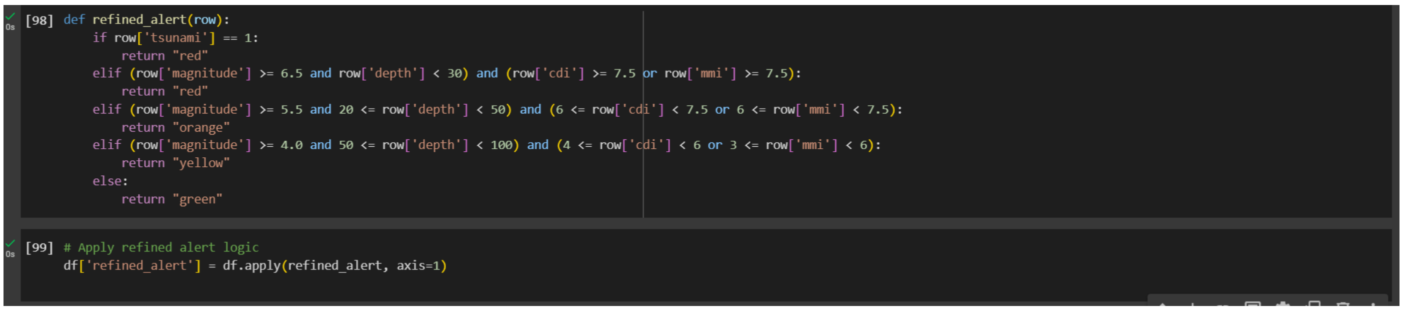
- **Tsunami Alert Models**: The Global Disaster Alert and Coordination System (GDACS) provides insights into tsunami alert systems, underscoring the importance of earthquake magnitude and depth in predicting tsunami potential. This research is instrumental in creating accurate tsunami risk assessments.

Stragically Derived Condition:

1. **Red Alert:**
    - If a tsunami occurred (**tsunami** = 1), mark as **Red Alert**.
    - Alternatively, if:
        - The earthquake has a **magnitude of 6.5 or higher**.

        - The earthquake's **depth is less than 30 km**.
        - The earthquake's impact (either **CDI** or **MMI**) is **7.5 or higher**.

2. **Orange Alert:**
    - If:
        - The earthquake has a **magnitude of 5.5 or higher**.
        - The earthquake's **depth is between 20 and 50 km**.
        - The earthquake's impact (either **CDI** or **MMI**) is **between 6 and 7.5**.
3. **Yellow Alert:**
    - If:
        - The earthquake has a **magnitude of 4.0 or higher**.
        - The earthquake's **depth is between 50 and 100 km**.
        - The earthquake's impact (either **CDI** or **MMI**) is **between 4 and 6**.
4. **Green Alert:**
    - For all other cases that do not meet the above conditions, mark as **Green Alert**.

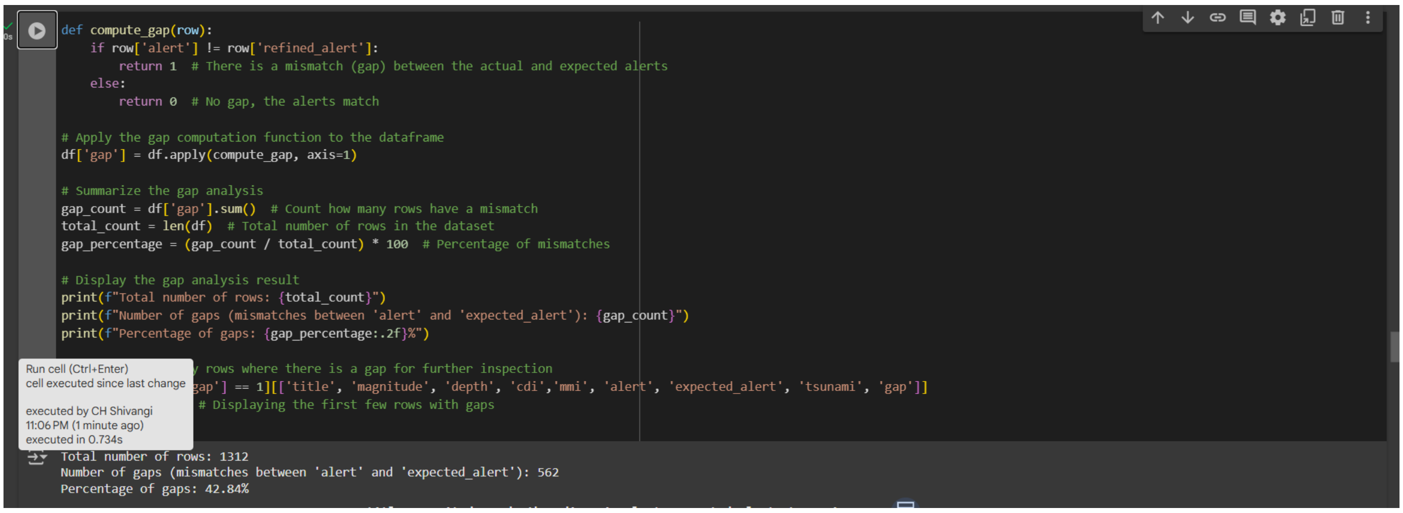


```
[98] def refined_alert(row):
         if row['tsunami'] == 1:
             return "red"
         elif (row['magnitude'] >= 6.5 and row['depth'] < 30) and (row['cdi'] >= 7.5 or row['mmi'] >= 7.5):
             return "red"
         elif (row['magnitude'] >= 5.5 and 20 <= row['depth'] < 50) and (6 <= row['cdi'] < 7.5 or 6 <= row['mmi'] < 7.5):
             return "orange"
         elif (row['magnitude'] >= 4.0 and 50 <= row['depth'] < 100) and (4 <= row['cdi'] < 6 or 3 <= row['mmi'] < 6):
             return "yellow"
         else:
             return "green"
```

```
[99] # Apply refined alert logic
     df['refined_alert'] = df.apply(refined_alert, axis=1)
```

Image 5.3

Observations *(Image 5.4)*:



```
def compute_gap(row):
    if row['alert'] != row['refined_alert']:
        return 1  # There is a mismatch (gap) between the actual and expected alerts
    else:
        return 0  # No gap, the alerts match

# Apply the gap computation function to the dataframe
df['gap'] = df.apply(compute_gap, axis=1)

# Summarize the gap analysis
gap_count = df['gap'].sum()  # Count how many rows have a mismatch
total_count = len(df)  # Total number of rows in the dataset
gap_percentage = (gap_count / total_count) * 100  # Percentage of mismatches

# Display the gap analysis result
print(f"Total number of rows: {total_count}")
print(f"Number of gaps (mismatches between 'alert' and 'expected_alert'): {gap_count}")
print(f"Percentage of gaps: {gap_percentage:.2f}%")

y rows where there is a gap for further inspection
gap'] == 1][['title', 'magnitude', 'depth', 'cdi','mmi', 'alert', 'expected_alert', 'tsunami', 'gap']]
 # Displaying the first few rows with gaps
```

```
Total number of rows: 1312
Number of gaps (mismatches between 'alert' and 'expected_alert'): 562
Percentage of gaps: 42.84%
```

Image 5.4

Key Findings (Post-Refinement)

1. Total Rows and Gap Analysis:
    - The dataset contained 1,312 earthquake occurrences.
    - 562 rows (42.84% of the total) contained a discrepancy between the actual alert and the refined expected_alert.
    - This is a significant drop in the amount of gaps compared to the prior analysis, which had a gap percentage of 64.33%.
2. Gap Reduction:
    - The difference between actual and expected alerts dropped by about 21.49%, indicating that the modified alert criteria that included CDI and MMI helped to eliminate discrepancies.
    - While the gap percentage has narrowed, 42.84% is a very high value, indicating that more progress might be done in matching the alerts with predicted levels.
3. Alert Distribution:
    The distribution of actual alerts and expected alerts post-refinement showed a more aligned pattern. The refined criteria, which consider both magnitude, depth, and intensity measures (CDI and MMI), were able to capture more realistic event characteristics and adjust alerts accordingly.

Analysis of Gap Rows (Post-Refinement)

The rows with mismatches (gaps) were examined further to understand the remaining discrepancies:

- Magnitude and Depth Thresholds:
  Despite the refinement, some events still had mismatches due to complex interactions between magnitude, depth, and the CDI/MMI values. Events close to the thresholds (e.g., magnitude = 6.5 or depth = 50 km) continue to pose challenges in assigning the correct alert level.
- Tsunami Events:
  Events with tsunami occurrences were generally well-classified as "red" alerts. However, there were a few cases where the expected "red" alert did not match the actual alert, which may indicate other factors affecting alert issuance, such as operational constraints or manual overrides.
- CDI and MMI Values:
  The inclusion of CDI and MMI values improved the model's accuracy, but certain events with moderate intensity measures (e.g., CDI between 4 and 6 or MMI between 3 and 6) still caused mismatches, likely due to inconsistencies or subjective classifications of intensity.

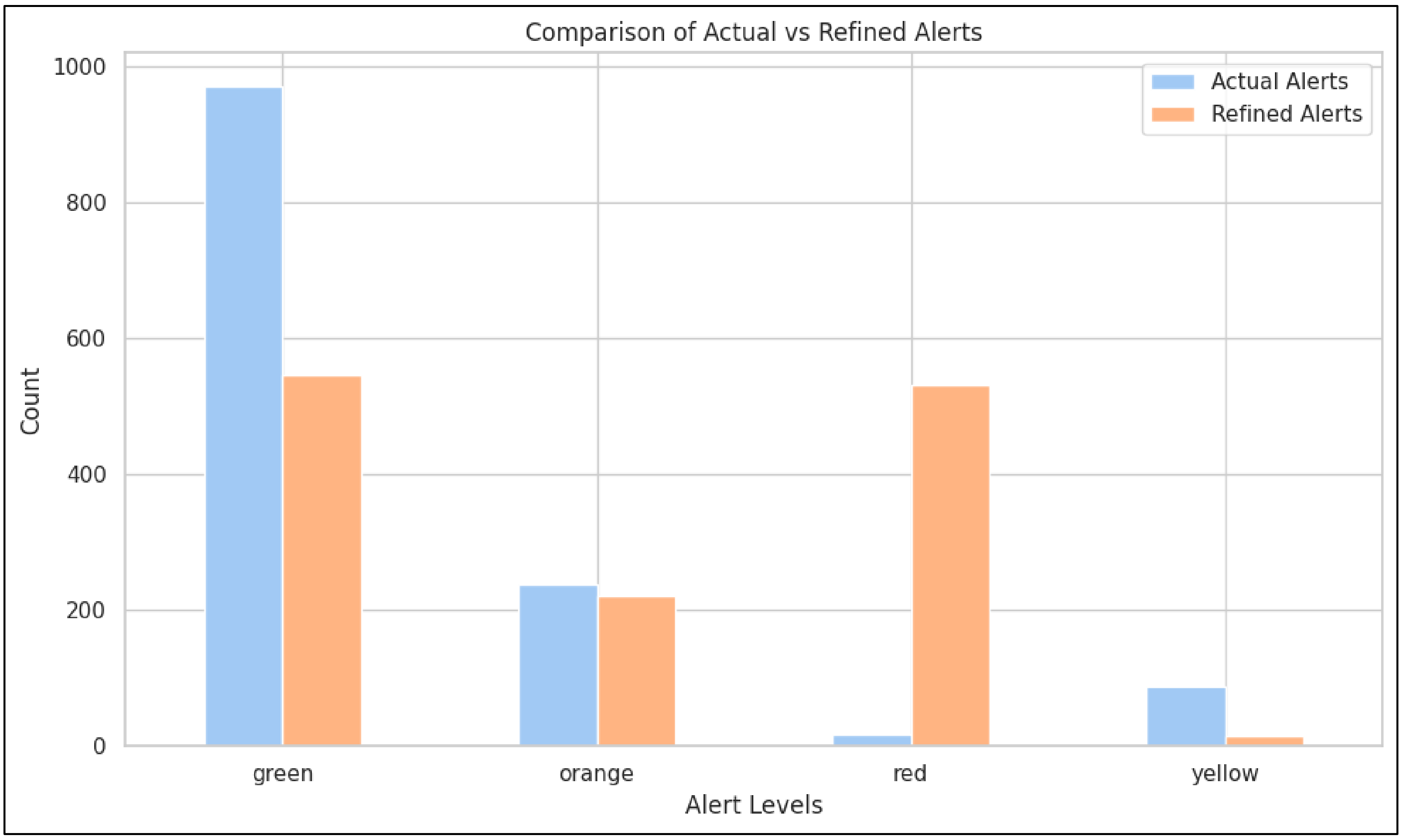


Image 5.5

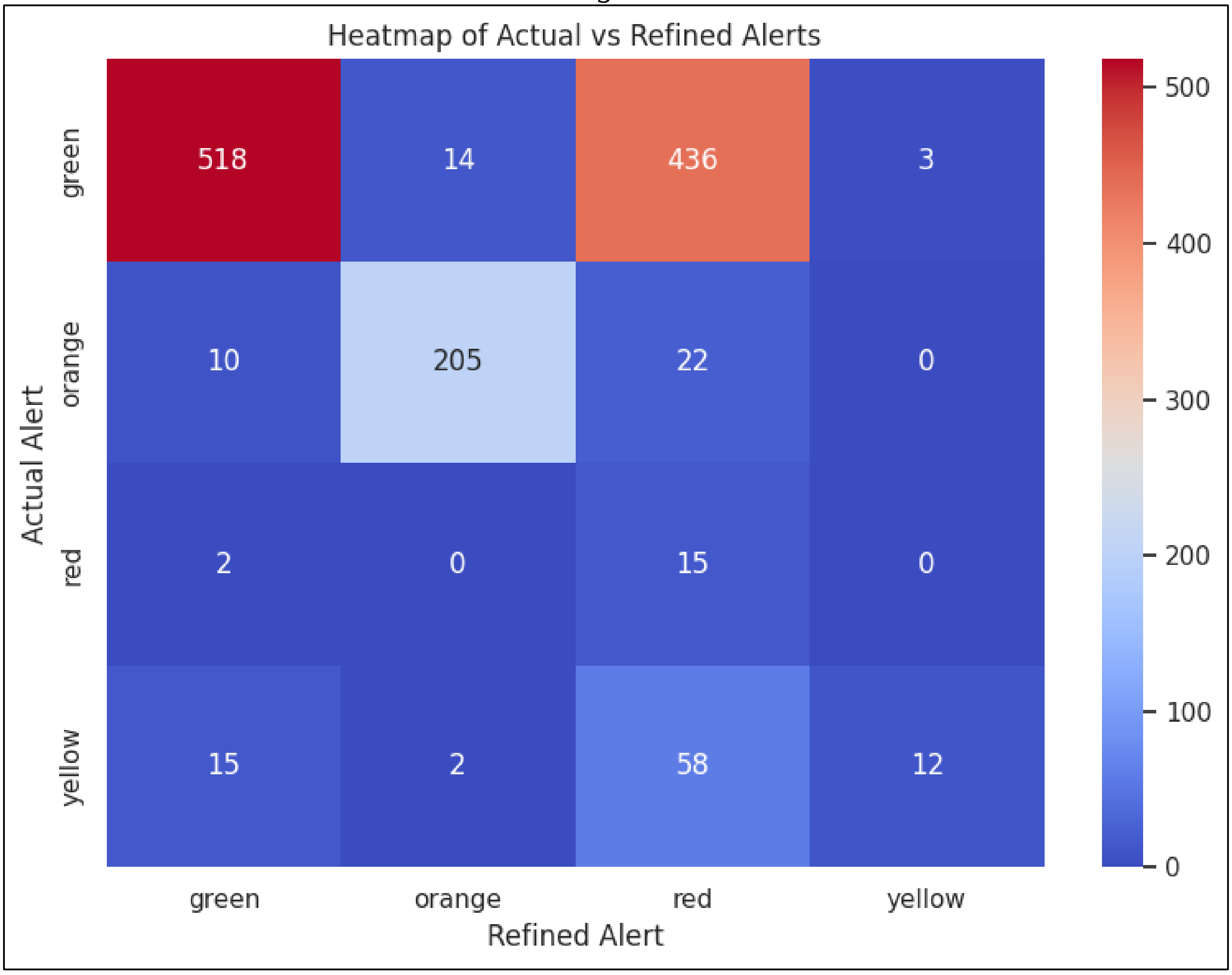


Image 5. 6

Recommendations for Further Improvement:

Although the refined alert criteria have led to a significant reduction in gaps, there are still opportunities to enhance the alert system further. Below are some suggestions for improving the system:

1. **Dynamic Threshold Adjustment:**
   - The thresholds for magnitude, depth, CDI, and MMI could be made dynamic, adapting based on historical data and event-specific factors. For example, deeper events with higher magnitudes could trigger a "red" alert even if the CDI or MMI are on the lower end.
2. **Additional Features for Alert Refinement:**
   - Time of Day: Incorporating the time of day and the population density in affected regions might help in better calibrating the alert levels. For instance, events during peak hours in highly populated areas could trigger higher alerts.
   - Event Duration: The duration of shaking (e.g., short vs. long-duration events) could be a contributing factor to the alert decision, as longer-duration events may cause more widespread impact.
3. **Data Quality and Consistency:**
   - Ensuring the quality and consistency of the input data (e.g., accurate and consistent CDI and MMI values) is crucial. Inaccurate or missing values in these fields may contribute to remaining gaps.
   - Implementing automated validation and data quality checks before processing the alerts can help prevent mismatches caused by incorrect or incomplete data.
4. **Machine Learning Models for Alert Prediction:**
   - Incorporating machine learning models could further refine the alert prediction system. Algorithms such as Random Forest, Support Vector Machines (SVM), or Neural Networks can learn from past events to dynamically adjust alert thresholds and improve classification accuracy.
   - These models could also take into account additional features, such as local building structures, proximity to fault lines, or real-time seismic data, to provide a more nuanced alert system.
5. **Alert System Feedback Loop:**
   - A continuous feedback loop where alerts are compared to actual outcomes (e.g., reported damage, casualties) could provide valuable insights to fine-tune the alert criteria over time. This would create an adaptive system capable of learning from each event.

CONCLUSION

The extensive studies and innovations achieved in this project have considerably improved the accuracy and effectiveness of earthquake event alarm systems. We effectively identified high-seismicity and tsunami-prone zones using Geospatial Risk Maps, offering useful information for risk mitigation and resource allocation. The Predictive Model extends the system's capabilities by calculating CDI and MMI from seismic features, allowing for a more proactive approach to alarm issuing.

The alarm System Evaluation enabled us to discover and close deficiencies in the first alarm system. By improving the parameters and examining discrepancies between actual and predicted alerts, we were able to narrow the gap from 64.33% to 42.84%. This indicates progress in aligning the system with real-world outcomes, though additional refinements are still required for optimal performance.

The Final Report & Dashboard serve as a crucial tool for stakeholders, giving engaging visualizations and actionable insights. These tools not only present the current state of the alert system but also allow for continual review and enhancements.

Overall, this project sets a solid foundation for a more accurate, responsive, and dynamic earthquake alert system. The combination of geospatial analysis, predictive modeling, and alert system optimization provides a powerful framework for enhancing earthquake preparedness, reducing risk, and saving lives in seismic-prone regions. Future improvements and continuous feedback will ensure that the system evolves to meet the growing challenges of disaster management and response.