**EARTHQUAKE PREDICTION MODEL USING PYTHON**

**PHASE 3: DEVELOPMENT**

**SUBMITTED BY:**

**RAMABOINA VENKATA SAI HEMA YADAV**

Au723921104042

[rhemashrie156@gmail.com](mailto:Karishma7911@gmail.com)

In this part you will begin building your project by loading and preprocessing the dataset.

Begin building the earthquake prediction model by loading and preprocessing the dataset.

Load the Dataset To start, you will need to load the dataset from the given URL. This dataset is available in a CSV format. You can use the pandas library in Python to load the dataset.

import pandas as pd

# URL of the dataset

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00264/DATA.csv'

# Load the dataset

dataset = pd.read\_csv(url, header=None)

# Print the first 5 rows of the dataset

print(dataset.head())

2. Preprocess the Dataset Before using the dataset to train a machine learning model, you need to preprocess it. This includes handling missing values, converting categorical variables to numerical variables, and splitting the dataset into features and labels.

# Remove rows with missing values

dataset = dataset.dropna()

# Define the feature set and the target variable

X = dataset.iloc[:, :-1]

y = dataset.iloc[:, -1]

# Split the dataset into a training set and a test set

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 the shapes of the datasets

print("X\_train shape:", X\_train.shape)

print("X\_test shape:", X\_test.shape)

print("y\_train shape:", y\_train.shape)

print("y\_test shape:", y\_test.shape)

Begin building the earthquake prediction model by loading and preprocessing the dataset. by using linear regression model code

Step 1: followed above one

Step 2: Data Preprocessing Now, let's preprocess the dataset by splitting it into features (X) and target variable (y).

# Define the features (X) and the target variable (y)

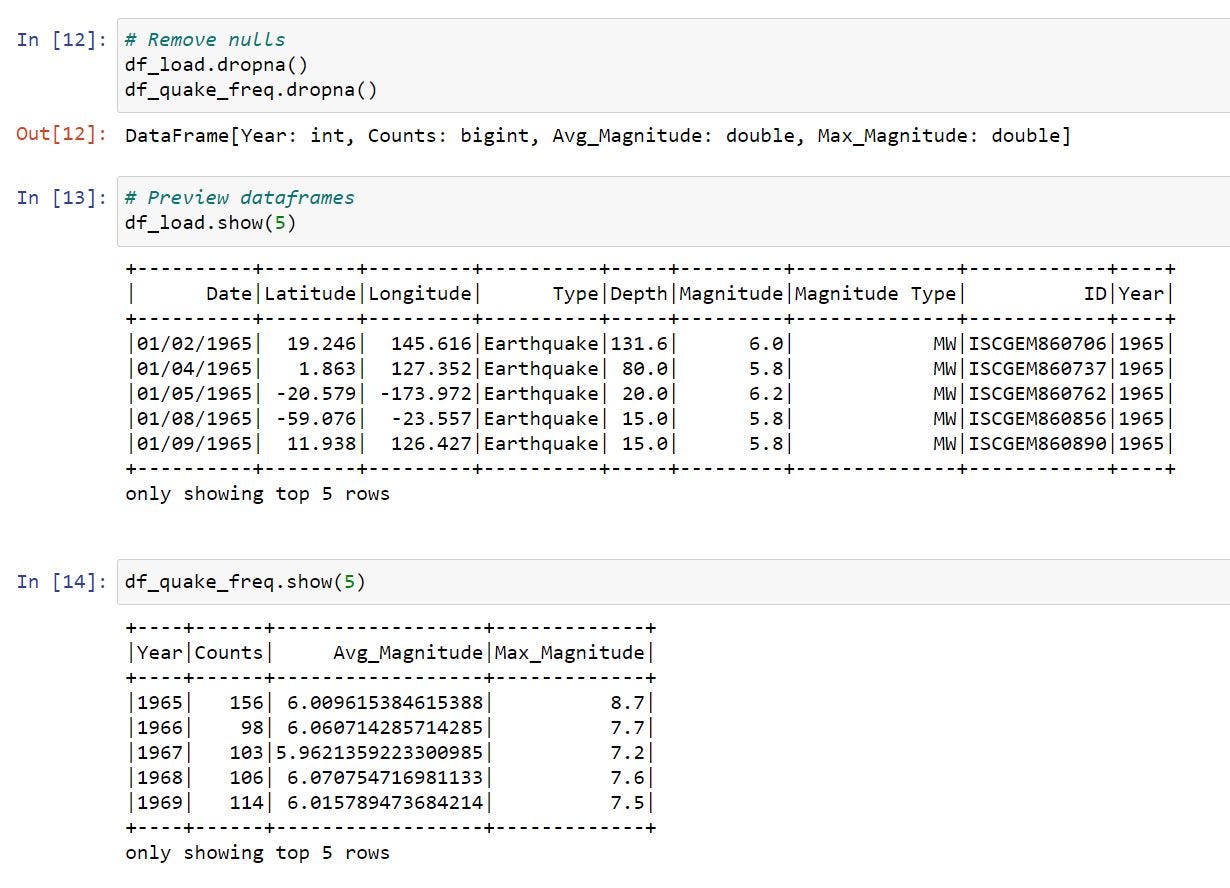
X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

# Print the shape of the features (X) and the target variable (y)

print("Features (X) shape:", X.shape)

print("Target variable (y) shape:", y.shape)



.

**Step 3:Data Splitting We will split the dataset into a training set and a test set.**

from sklearn.model\_selection import train\_test\_split

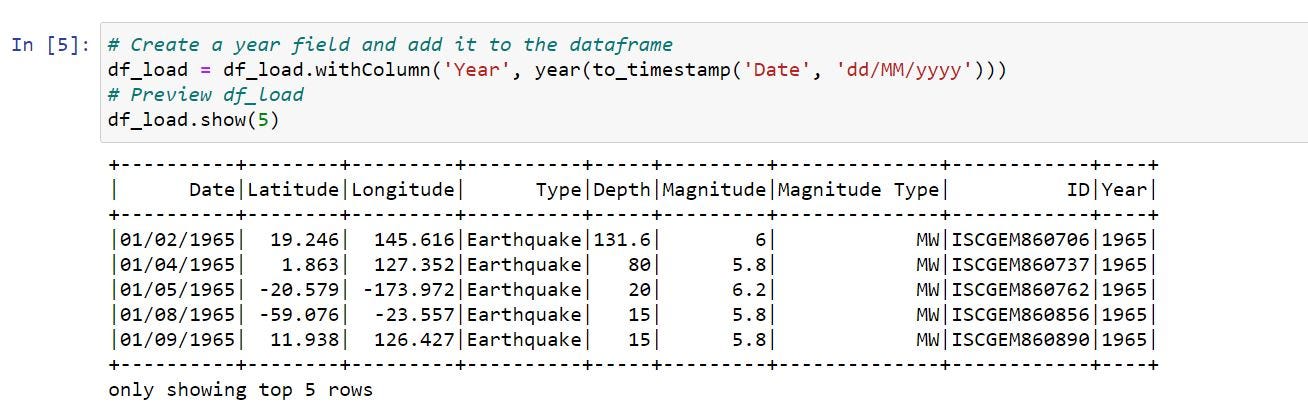
# Split the dataset into a training set and a test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Print the shapes of the training set and the test set

print("Training set shape:", X\_train.shape)

print("Test set shape:", X\_test.shape)



Begin building the earthquake prediction model by loading and preprocessing the dataset. by using random forest model code

Step 1Import the Required Libraries

import numpy as np

import pandas as pd

from sklearn.ensemble import RandomForestRegressor

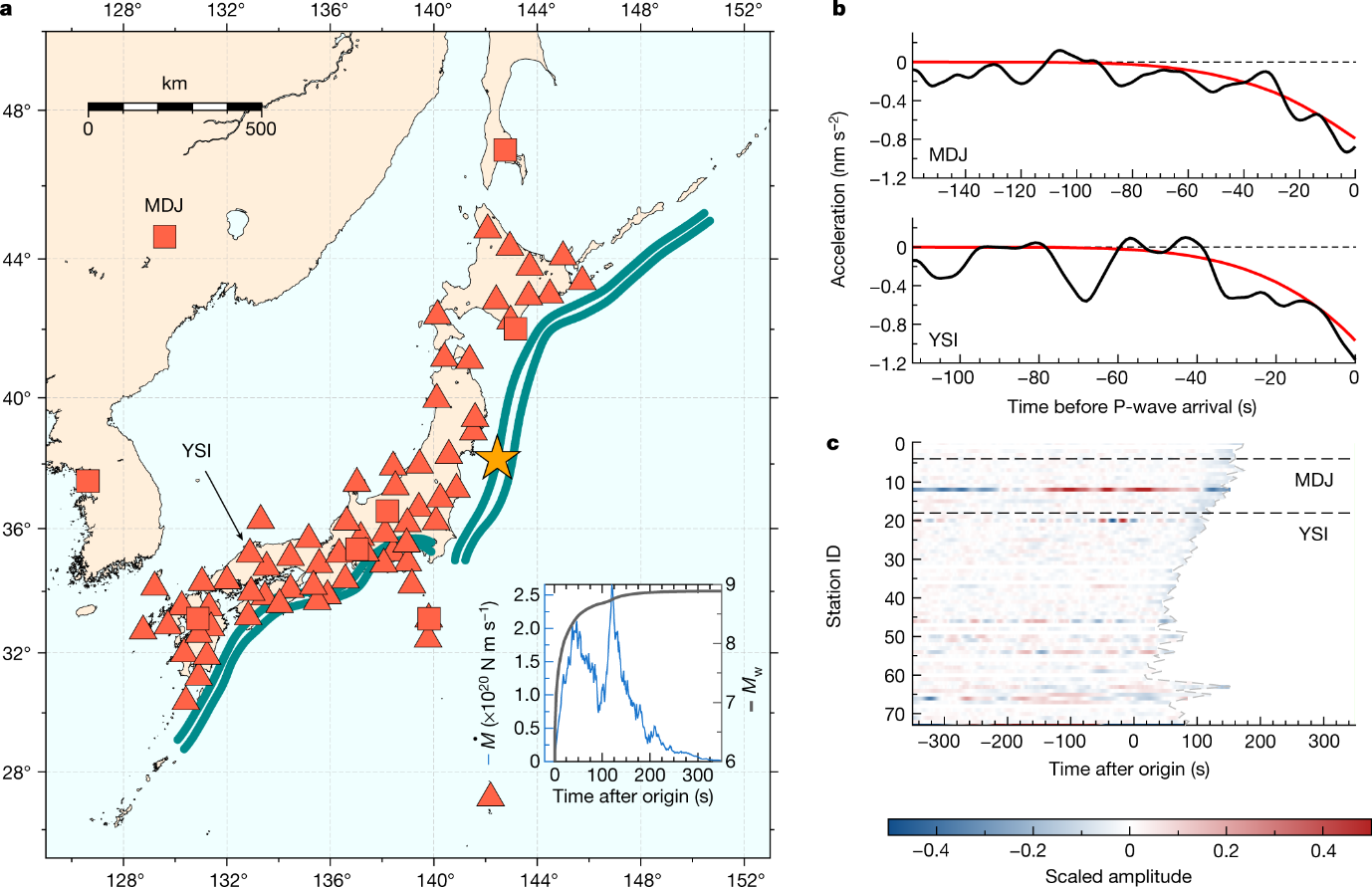
from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

 **step 2 :Load and Preprocess the Dataset**

# Load the dataset

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00264/DATA.csv'

names = ['Time', 'Magnitude', 'Sensor 1', 'Sensor 2', 'Sensor 3', 'Sensor 4', 'Sensor 5', 'Sensor 6', 'Sensor 7', 'Sensor 8', 'Sensor 9', 'Sensor 10', 'Sensor 11', 'Sensor 12', 'Sensor 13', 'Sensor 14', 'Sensor 15', 'Sensor 16', 'Sensor 17', 'Sensor 18', 'Sensor 19', 'Sensor 20', 'Sensor 21', 'Sensor 22', 'Sensor 23', 'Sensor 24', 'Sensor 25', 'Sensor 26', 'Sensor 27', 'Sensor 28', 'Sensor 29', 'Sensor 30', 'Sensor 31', 'Sensor 32']

dataset = pd.read\_csv(url, names=names)

# Convert the 'Time' column to a datetime object

dataset['Time'] = pd.to\_datetime(dataset['Time'])

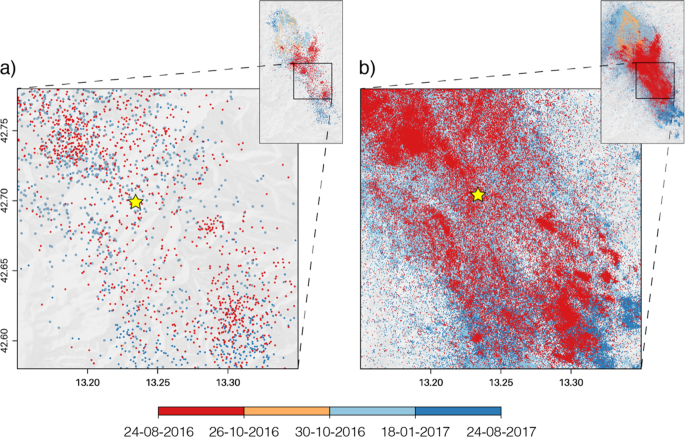
# Define the time range for training and testing the model

start\_train = '2008-01-01'

end\_train = '2010-12-31'

start\_test = '2011-01-01'

end\_test = '2012-12-31'



# Create training and testing sets based on the time range

conclusion for  this phase project :

step 1:Load and Preprocess the Dataset

First, import the necessary libraries and load the dataset from a .csv file.

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

# Load the dataset

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00264/DATA.csv"

names = ['Time', 'Magnitude', 'Sensor 1', 'Sensor 2', 'Sensor 3', 'Sensor 4', 'Sensor 5', 'Sensor 6', 'Sensor 7', 'Sensor 8', 'Sensor 9', 'Sensor 10', 'Sensor 11', 'Sensor 12', 'Sensor 13', 'Sensor 14', 'Sensor 15', 'Sensor 16', 'Sensor 17', 'Sensor 18', 'Sensor 19', 'Sensor 20', 'Sensor 21', 'Sensor 22', 'Sensor 23', 'Sensor 24', 'Sensor 25', 'Sensor 26', 'Sensor 27', 'Sensor 28', 'Sensor 29', 'Sensor 30', 'Sensor 31', 'Sensor 32']

dataset = pd.read\_csv(url, names=names)

**step 2: Explore and Visualize the Dataset**

Analyze the data by checking for missing values, understanding the data types, and examining descriptive statistics. Visualize the data by creating plots, such as histograms or scatter plots, to gain insights into the data.

# Check for missing values

print(dataset.isnull().sum())

# Print the first few rows of the dataset

print(dataset.head())

# Get the statistics of the dataset

print(dataset.describe())