Earthquake Prediction Model using Python

**Team Members :**

**Hariharan K**

**III Year CSE**

**AI101 (IBM Artificial Intelligent Group 2)**

**Team Name : MiddleclassVIP Coder**

**Menter Name : Krisnaveni CSE HOD**

**SRG Engineering College, Namakkal**

**Phase 1: Problem Definition and Design Thinking**

**Problem Statement**

The problem at hand involves developing an earthquake prediction model using a Kaggle dataset. The primary objective is to explore and understand the key features of earthquake data, visualize the data on a world map to gain a global overview, split the dataset into training and testing subsets, and build a neural network model capable of predicting earthquake magnitudes based on the provided features.

**Design Thinking**

**Data Source**

To initiate this project, we will need to select a suitable dataset from Kaggle containing earthquake data. The dataset should include essential features such as date, time, latitude, longitude, depth, and magnitude. We will use this data for analysis and model development.

**Feature Exploration**

1. **Data Acquisition**: Download the selected earthquake dataset from Kaggle. Ensure that it includes the necessary features (date, time, latitude, longitude, depth, magnitude).
2. **Data Inspection**: Load the dataset into a Pandas DataFrame and inspect its structure and content. Ensure that it is clean and well-documented.
3. **Data Analysis**: Perform an exploratory data analysis (EDA) to gain insights into the distribution, correlations, and characteristics of the key features. Calculate summary statistics, identify outliers, and visualize the data using histograms, scatter plots, and correlation matrices.

**Visualization**

1. World Map Visualization: Utilize geospatial visualization libraries (e.g., Folium, Plotly) to create a world map visualization that displays the distribution of earthquakes. Each earthquake's location can be represented by a marker on the map, with the marker's size and color indicating the magnitude and frequency of earthquakes in that region.

**Data Splitting**

1. **Dataset Splitting**: Divide the dataset into two subsets: a training set and a test set. Typically, an 80-20 or 70-30 split ratio is used, with the majority of data allocated for training and a smaller portion for testing.

**Model Development**

1. **Neural Network Architecture**: Design a neural network model for earthquake magnitude prediction. The input layer should correspond to the relevant features (e.g., latitude, longitude, depth), and the output layer should predict earthquake magnitudes.
2. **Model Building**: Implement the neural network using a deep learning framework such as TensorFlow or PyTorch. Define the model architecture, specify activation functions, and choose an appropriate loss function for regression tasks.
3. **Model Training**: Train the neural network model on the training dataset. Monitor the training process, and implement techniques such as batch normalization and dropout to improve model performance and prevent overfitting.

**Training and Evaluation**

1. **Model Evaluation**: After training, evaluate the model's performance on the test dataset using appropriate evaluation metrics for regression tasks. Common metrics include Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).
2. **Fine-Tuning**: If the model's performance is unsatisfactory, consider fine-tuning hyperparameters or experimenting with different neural network architectures to improve predictions.
3. **Visualization of Results**: Visualize the model's predictions on a world map to see how well it aligns with the actual earthquake magnitudes. This can provide valuable insights into the model's accuracy and areas of improvement.
4. **Documentation**: Document the entire process, including data sources, feature selection, model architecture, hyperparameters, and evaluation results. Present findings and insights in a clear and concise manner.

By following this structured approach, we aim to develop a neural network-based earthquake prediction model that can contribute to better understanding and potentially forecasting earthquake magnitudes based on geospatial and temporal data.