**EARTHQUAKE PREDICTION MODEL**

**1. Problem Statement:**

The objective of this project is to predict earthquakes based on historical seismic data. Earthquake occurrences are natural events, and understanding when and where they may happen is of great importance for disaster preparedness and risk mitigation.

**2. Design Thinking Process:**

The project followed a structured development process, including data collection, preprocessing, feature engineering, model selection, and evaluation. This process was essential to ensure a systematic approach to earthquake prediction.

**3. Phases of Development:**

The development process can be divided into several phases: data import, data preprocessing, feature extraction, data visualization, data splitting, model selection, and model evaluation.

**4. Dataset Description:**

The earthquake data was obtained from a CSV file named "database.csv."

The dataset includes various attributes such as Date, Time, Latitude, Longitude, Depth, Magnitude, etc.

The source of the dataset is not mentioned in the code.

**5. Data Preprocessing:**

Data preprocessing steps include handling missing values and converting Date and Time to Unix timestamps for use as numerical input.

Records with erroneous Date or Time values are dropped from the dataset.

**6. Feature Extraction:**

The selected features for earthquake prediction are Date, Time, Latitude, Longitude, Depth, and Magnitude.

**7. Visualization:**

The code uses Basemap to visualize earthquake occurrences on a world map. This provides a clear representation of the regions with a higher frequency of earthquakes.

**8. Splitting the Data:**

The dataset is split into input (X) and output (y) variables.

Input variables (X) include Timestamp, Latitude, and Longitude.

Output variables (y) include Magnitude and Depth.

The data is split into training and testing datasets with an 80-20 ratio.

**9. Machine Learning Model (Random Forest Regressor):**

A RandomForestRegressor model is used to predict earthquake Magnitude and Depth.

The model is trained and tested, but the prediction results are not satisfactory.

**10. Hyperparameter Tuning with GridSearchCV:**

GridSearchCV is used to find the best hyperparameters for the RandomForestRegressor.

The best-fit model is chosen based on the mean test score.

**11. Neural Network Model**

A neural network model is constructed using Keras.

The neural network has three dense layers with 16, 16, and 2 nodes, respectively, and uses the ReLU activation function.

**12. Hyperparameter Tuning for Neural Network:**

Hyperparameter tuning is performed for the neural network, including parameters like batch size, epochs, activation functions, optimizers, and loss functions.

**13. Model Evaluation:**

The neural network model is evaluated on both the training and testing datasets.

The evaluation results include loss and accuracy metrics.