# NAAN MUDHALVAN PROJECT

## EARTHQUAKE PREDICTION MODEL USING PYTHON

**Phase -2: Algorithm for Design to Innovation**

## Step 1: Data Collection

In this phase, we will collect the earthquake dataset from Kaggle (we got the link of dataset). Our dataset contains the features like: Date, Time, Latitude, Longitude, Type, Depth, Depth Error, Depth Seismic Stations, Magnitude and Magnitude Type. It's crucial to ensure data quality and completeness during this stage.

## Step 2: Data Preprocessing

### 2.1 Data Loading

- Load the earthquake dataset into a **Pandas Data Frame** or a suitable data structure for analysis.

### 2.2 Initial Data Exploration

- Check the dataset for missing values and handle them by imputing or removing rows with missing data.
- Convert the date and time columns into a unified **timestamp** format (e.g., datetime).

### 2.3 Exploratory Data Analysis (EDA)

- Explore the distribution of each feature using statistical summaries, histograms, and box plots.
- Examine correlations between features using correlation matrices or scatter plots.

## Step 3: Data Visualization

### 3.1 Geospatial Visualization

- Utilize **geospatial libraries** (e.g., Folium, Plotly, or Matplotlib with basemap) to create a world map visualization.
- Plot earthquake occurrences on the map, using color-coding to represent magnitude.
- Generate descriptive visualizations to understand the global distribution of earthquakes.

## Step 4: Data Splitting

### 4.1 Data Preparation

- Prepare the dataset for machine learning by selecting relevant features (e.g., latitude, longitude, depth) and the target variable (magnitude).

### 4.2 Data Splitting

- Split the dataset into two subsets: a **training set (80%) and a test set (20%)**.
- Ensure that the splitting maintains the distribution of magnitudes to avoid bias.

## Step 5: Model Development

### 5.1 Neural Network Architecture

- Design a neural network model for **earthquake magnitude prediction**.
- Define the input layer with the appropriate number of neurons (based on selected features).
- Add one or more hidden layers with appropriate activation functions (e.g., ReLU).

- Use a linear activation function in the output layer for regression tasks.

### 5.2 Model Compilation

- Compile the neural network model by specifying the loss function (e.g., mean squared error), optimizer (e.g., Adam), and evaluation metrics (e.g., MSE, MAE).

### 5.3 Model Training

- Train the model on the training dataset using the compiled model, specifying the number of epochs and batch size.
- Monitor the training progress, observing loss convergence and model performance on validation data.

## Step 6: Model Evaluation

### 6.1 Model Testing

- Evaluate the trained neural network model on the test dataset to assess its ability to make accurate earthquake magnitude predictions.

### 6.2 Performance Metrics

- Calculate evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared ($R^2$) score.
- Visualize the model's predictions compared to the actual earthquake magnitudes.

## Step 7: Model Tuning

### 7.1 Hyperparameter Tuning

- Consider tuning hyperparameters like learning rate, batch size, and the number of hidden layers and neurons.

- Use techniques like grid search or random search to find optimal hyperparameters.

## 7.2 Model Complexity

- Experiment with different neural network architectures, including deep and shallow networks, to find the best model complexity for the task.