

**NAAN MUDHALVAN PROJECT(IBM)**

**IBM AI 101 ARTIFICIAL INTELLIGENCE-GROUP 1**

**Title : EARTHQUAKE PREDICTION MODEL USING PYTHON**

**Team name**: Proj\_224824\_Team\_1

**Team members:** SARAVANA P 113321106086

SARATH KUMAR M 113321106085

YEKOLLA KESAVA 113321106118

VEMURU VINAY KUMAR REDDY 113321106113

PULA GNANA CHAITANYA 113321106071

# Problem Statement :

# Explore the key features of earthquake data and design an object for those features, such as date, time, latitude, longitude, depth, and magnitude. Before developing the prediction model, visualize the data on a world map to display a complete overview of where the earthquake frequency will be higher. Split the data into a training set and a test set for validation. Lastly, build a neural network to fit the data from the training set.

# Empathize:

# Understand the problem: Research and gather information about earthquakes, their causes, and historical earthquake data.

# Identify stakeholders: Determine who will benefit from earthquake predictions, such as local communities, governments, or researchers.

# Define:

# Clearly define the problem you want to solve: In this case, it's predicting earthquakes to mitigate their impact.

# Set specific goals: Determine what you aim to achieve with your prediction model, such as early warning or risk assessment.

# Ideate:

# Brainstorm potential data sources: Identify the data you need, such as seismic data, geological data, and historical earthquake records.

# Explore machine learning techniques: Research different algorithms like neural networks, decision trees, or support vector machines that can be used for prediction.

# Prototype:

# Gather data: Acquire and preprocess relevant datasets. You can use libraries like Pandas for data manipulation.

# Feature engineering: Extract meaningful features from your data, such as seismic activity trends, geological characteristics, and historical earthquake patterns.

# Choose a machine learning algorithm: Implement and train a model on your data. Libraries like Scikit-Learn can be helpful.

# Evaluation: Evaluate your model's performance using metrics like accuracy, precision, recall, and F1-score.

# Test:

# Test the model with real-time or historical data to assess its accuracy and reliability.

# Gather feedback from experts and stakeholders to improve the model.

# 

# Implement:

# Integrate the model into a user-friendly interface, such as a web application or mobile app.

# Develop a system for real-time data collection and prediction.

# Establish communication channels for disseminating earthquake predictions.

# Iterate:

# Continuously improve the model based on feedback and new data.

# Stay updated with the latest research and technologies in earthquake prediction.

# Feedback and Insights:

# Monitor the model's performance in real-world scenarios.

# Collect feedback from users and stakeholders to make necessary adjustments.

# 