NAAN MUDHALVAN

PROJECT: Earthquake Prediction Model using Python

PHASE 3

TEAM MEMBERS:

Srijeet Goswami - 2021504044

Akash R - 2021504501

Gokul E – 2021504514

Balamurugan M – 2021504507

PROBLEM DEFINITION:

The problem is to develop an earthquake prediction model using a Kaggle dataset. The objective is to explore and understand the key features of earthquake data, visualize the data on a world map for a global overview, split the data for training and testing, and build a neural network model to predict earthquake magnitudes based on the given features.

STEPS TAKEN SO FAR TO ACHIEVE THE GOAL:

1.Loading the dataset

We have successfully uploaded the dataset that we have chosen for our project. The dataset contains relevant and reliable information that will help me answer our research question.

Dataset link: ***https://www.kaggle.com/datasets/usgs/earthquake-database***

Procedure to load the dataset:

1. Go to Google Drive and sign in with your Google account.
2. At the top left, click New > File upload or Folder upload.
3. Choose the file or folder you want to upload and wait for the upload to complete.
4. Open a new notebook in Google Colab and import pandas as pd.
5. To access your uploaded file or folder, you need to mount your google drive to colab.
6. This will ask you to open a URL in your browser to grant access to Colab. Copy the authorization code and paste it into the space given in the notebook.
7. Now you can access all your files and folders on your drive directly on Colab. To load the dataset, you need to know its path on your drive. You can use the file explorer on the left pane of Colab to find it, or use the !ls command to list the files and folders in a directory.
8. Once you have the path, you can use the pd.read csv() function to load the dataset as a pandas data frame.
9. You can now use the data frame for further analysis and visualization using various tools and techniques.

2.Data manipulation

By viewing the dataset, we find that the following parameters are provided for different earthquakes that have occurred in the past:

'Date', 'Time', 'Latitude', 'Longitude', 'Type', 'Depth', 'Depth Error', 'Depth Seismic Stations', 'Magnitude', 'Magnitude Type', 'Magnitude Error', 'Magnitude Seismic Stations', 'Azimuthal Gap', 'Horizontal Distance', 'Horizontal Error', 'Root Mean Square', 'ID', 'Source', 'Location Source', 'Magnitude Source', 'Status'

These labels are printed by using the ‘data.columns’ command.

Next, we analyse the data we need and the ones we don’t need.

Then, we retain the selected parameters and the first few rows are displayed using the data.head command.

3.Visualization

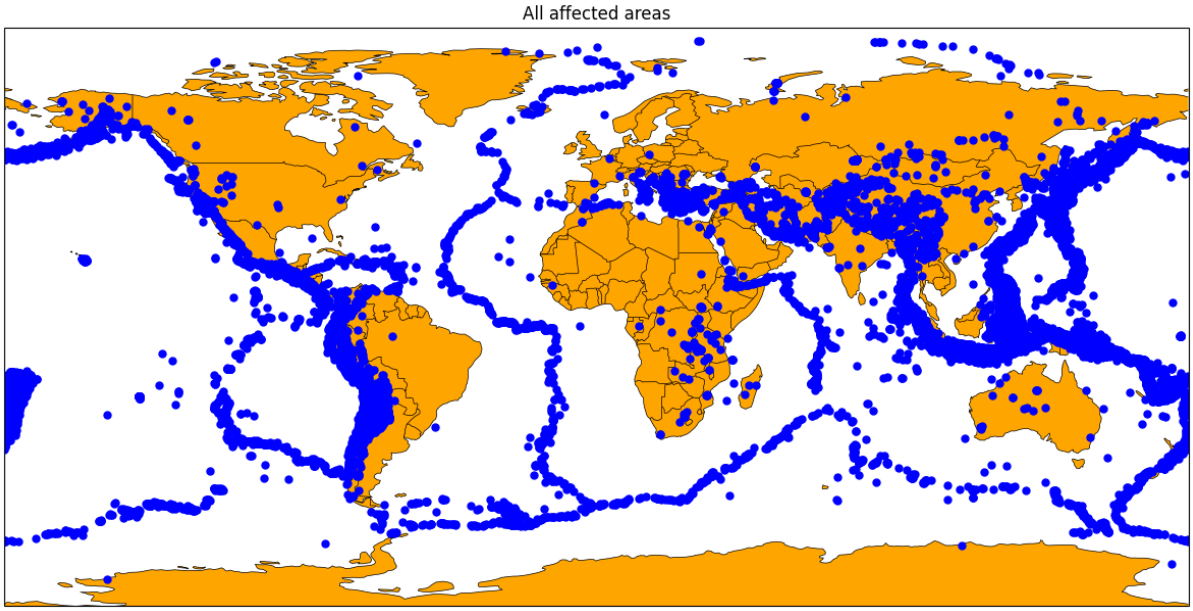
A world map visualization to created display earthquake frequency distribution.

The following libraries are imported for this purpose:

- cartopy.crs
  Cartopy.crs is a Python library that provides tools for defining and transforming coordinate reference systems (CRS) for cartographic visualization. It is part of the Cartopy package, which is designed to make drawing maps for data analysis and visualisation easy. Cartopy.crs allows users to create and manipulate various types of CRS objects, such as geodetic, geocentric, projected, and rotated CRS. It also enables users to transform points, lines, polygons, images, and vectors between different CRS using the Proj.4 library.
- cartopy.feature
  Cartopy.feature is a Python library that provides tools for creating and manipulating geographic features for cartographic visualization. Cartopy.feature allows users to access various types of features, such as country borders, coastlines, lakes, rivers, and states, from Natural Earth or GSHHS shapefiles. It also enables users to customize the appearance and style of the features using Matplotlib.
- cartopy.mpl.patch
  Cartopy.mpl.patch is a Python library that provides tools for converting shapely geometries into matplotlib paths for cartographic visualization.

- matplotlib.pyplot
  Matplotlib.pyplot is a Python library that provides a state-based interface to Matplotlib, a comprehensive library for creating static, animated, and interactive visualizations in Python1. Matplotlib.pyplot allows users to make plots using an implicit, MATLAB-like, way of plotting, and also opens figures on the screen and acts as the figure GUI manager
- matplotlib.patches
  Matplotlib.patches is a Python library that provides tools for creating and manipulating 2D shapes for data visualization. It is part of the Matplotlib package, which is a comprehensive library for creating static, animated, and interactive visualizations in Python

Given below is a sample plot highlighting all the affected areas, which are represented by blue colour.



4.Data splitting

The dataset is divided into two subsets: a training set and a testing (or validation) set. The training set is used to train a machine learning model, while the testing set is used to assess the model's performance and generalization to unseen data.

This is done using the train_test_split library.