Earthquake Prediction Model using Python

Creating an earthquake prediction model involves several steps, starting with loading and preprocessing the dataset. However, it's important to note that earthquake prediction is a complex and challenging task, and there's no guaranteed method to predict earthquakes accurately. Still, we can attempt to build a model using historical earthquake data and various features.

Following our the steps involved in data processing:

1.1 Import Necessary Libraries:

Let’s start by importing the necessary Python libraries for data manipulation, analysis, and machine learning for our dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

1.2 Loading our Dataset :

The data which we use is from kaggle. In order to use that dataset with help of Pandas DataFrame.

```
dataset = pd.read_csv("/kaggle/input/earthquake-database/database.csv")
dataset.head(5)
```

1.3 Exploring and Preprocessing our Dataset :

Explore the dataset to understand its structure, features, and target variable. Preprocess the data accordingly.

In order to check for any missing values, the following code is implemented.

```
print(dataset.isnull().sum())
```

Adding a new column known as Binary, which we will use later for prediction.

```
for mag in df["Magnitude"]:
   if round(mag) > float(5.5) :
      df["Binary"] = 1
   else:
      df["Binary"] = 0
```

Don’t consider the unwanted column with help drop keyword and the column which we want to predict is when an earthquake occurs .

```
data = dataset.drop([
   "Root Mean Square",
   "Depth Seismic Stations",
   "Magnitude Type",
   "Magnitude Seismic Stations",
   "ID",
   "Source",
   "Magnitude Source",
   "Location Source",
   "Horizontal Error",
   "Horizontal Distance",
   "Azimuthal Gap",
], axis=1)
```

```
y = data[‘Location’]
```

Now we will be dividing our dataset into two testing and training dataset.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

1.4 Visualizing our data

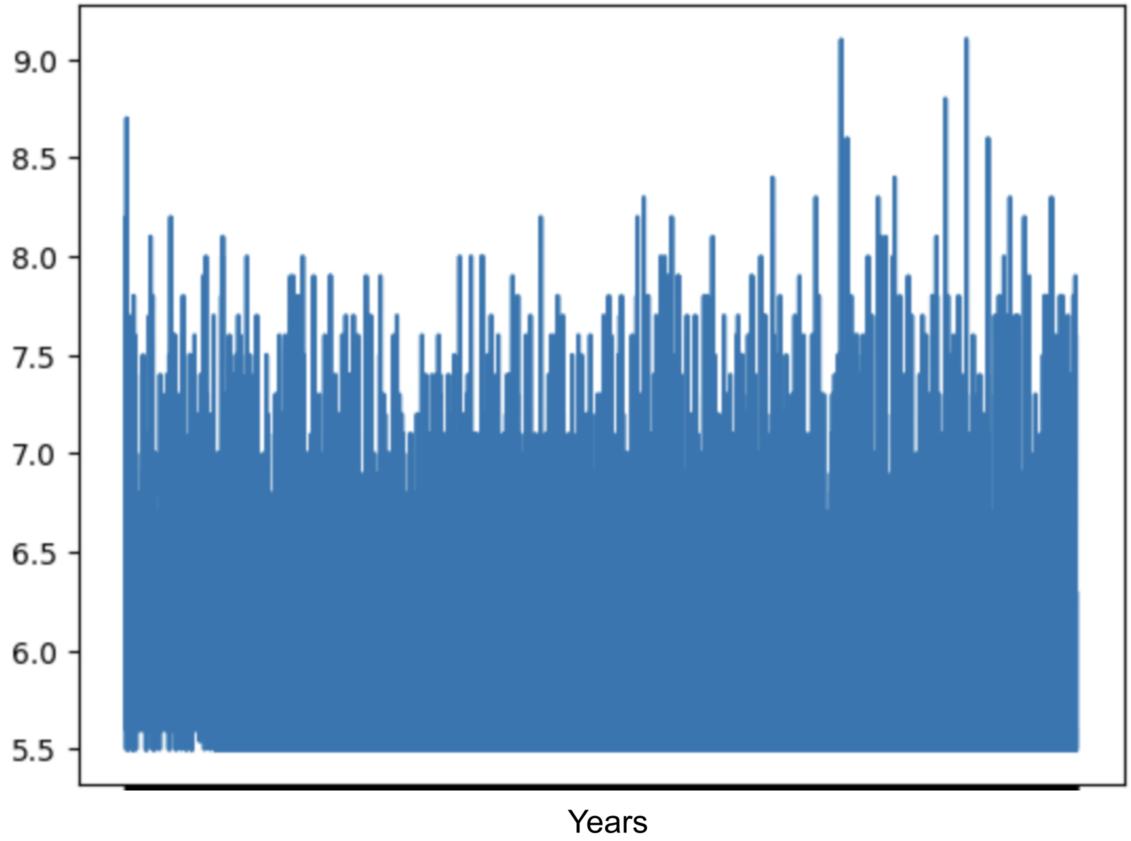
We will extract year from the dataset date column and assign the value to a new column of the dataset

```
data[“Date”] = pd.to_datetime(data[“Date”])
```

Now we will be using this data to plot against earthquake occurrence.

```
plt.bar(earthquake_counts_per_year.index, earthquake_counts_per_year.values,
color='skyblue') plt.xlabel('Year')
```

```
plt.ylabel('Number of Earthquakes')
plt.title('Earthquake Occurrences (1990 to 2023)')
plt.xticks(range(1990, 2024, 1))
plt.show()
```



Once we have preprocessed the data, we can proceed to build and train a model for earthquake prediction.The model which we used is a deep neural network which is based on logistic regression.The output of the model is then converted to a map, which contains the location of the earthquake