EARTHQUAKE PREDICTION MODEL USING PYTHON

Batch member:

- DINESH.G
- 510521104006

PHASE-1 SUBMISSION DOCUMENT:

INTRODUCTION:

It is well known that if a disaster occurs in one region, it is likely to happen again. Some regions have frequent earthquakes, but this is only a

comparative amount compared to other regions.

So, predicting the earthquake with date and time, latitude and longitude from previous data is not a trend that follows like other things, it happens naturally.

Machine learning has the ability to advance our knowledge of earthquakes and enable more accurate forecasting and catastrophe response. It's crucial to remember that developing accurate and dependable prediction models for earthquakes still needs more study as it is a complicated and difficult topic.

DEFINITION:

In order to anticipate earthquakes, machine learning may be used to examine seismic data trends. Seismometers capture seismic data, which may be used to spot changes to the earth's surface, like seismic waves brought on by earthquakes. Machine learning algorithms may utilize these patterns to forecast the risk of an earthquake happening in a certain region by studying these patterns and learning to recognize key traits that are linked to seismic activity.

Neural Network Model:

A neural network model can be employed to forecast earthquakes by examining diverse elements and trends

in seismic data. This model harnesses the capabilities of neural networks, which draw inspiration from the neural connections of the human brain, to analyze intricate data and reveal hidden relationships and patterns. By training the neural network on historical earthquake data, it can acquire the ability to identify precursor signals and patterns that indicate the probability of an upcoming earthquake

The following concepts are used to create an earthquake prediction model:

1. importing libraries
2. read the dataset

3.visualization

4.splitting datasets

We will be predicting the earthquake fromDate and Time, Latitude, and Longitude from previous data is not a trend that follows like other things. It is naturally occurring.

The dataset we are using here contains data for the following columns:

- ✓ Origin time of the Earthquake
- ✓ Latitude and the longitude of the location.

- ✓ Depth – This means how much depth below the earth's level the earthquake started.
- ✓ The magnitude of the earthquake
- ✓ Location

The some other conLocation machine learning used in the earthquake prediction model are given below:

- ✓ Featured engineering
- ✓ Exploratory data analysis (EDA)

DESIGN THINKING:

The above mentioned concepts are implemented in order to create an earthquake prediction model.

- ***Importing libraries:***

create a model for earthquake prediction by importing the necessary python libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import os
print(os.listdir("../input")
```

OUTPUT:

(database.csv)

➢ ***Read the datasets:***

Now we will read the dataset and look for the various features in the dataset

PROGRAM:

```
Data=pd.read_csv("../input/database.v)
Data.head()
Data.columns
Data = data[['Date', 'Time',
'Latitude','Longitude', 'Depth', 'Magnitude']]
Data.head()
```

Consider the following code to create the table of values that contains information about the earthquake factors.

PROGRAM:

```
Import datetime
Import time

Timestamp = []
For d, t in zip(data['Date'], data['Time']):
Try:
 Ts = datetime.datetime.strptime(d+' '+t,
'%m/%d/%Y %H:%M:%S')
Timestamp.append(time.mktime(ts.timetul
e()))
 Except ValueValueErro
# print('ValueError')
Timestamp.append('ValueError')

timeStamp = pd.Series(timestamp)
data['Timestamp'] = timeStamp.values
```

OUTPUT:

	Latitude	Longitude	Depth	Magnitude	Timestamp
0	19.246	145.616	131.6	6.0	-1.57631e+08
1	1.863	127.352	80.0	5.8	-1.57466e+08
2	-20.579	-173.972	20.0	6.2	-1.57356e+08
3	-59.076	-23.557	15.0	5.8	-1.57094e+08
4	11.938	126.427	15.0	5.8	-1.57026e+08

- ***Visualization***:

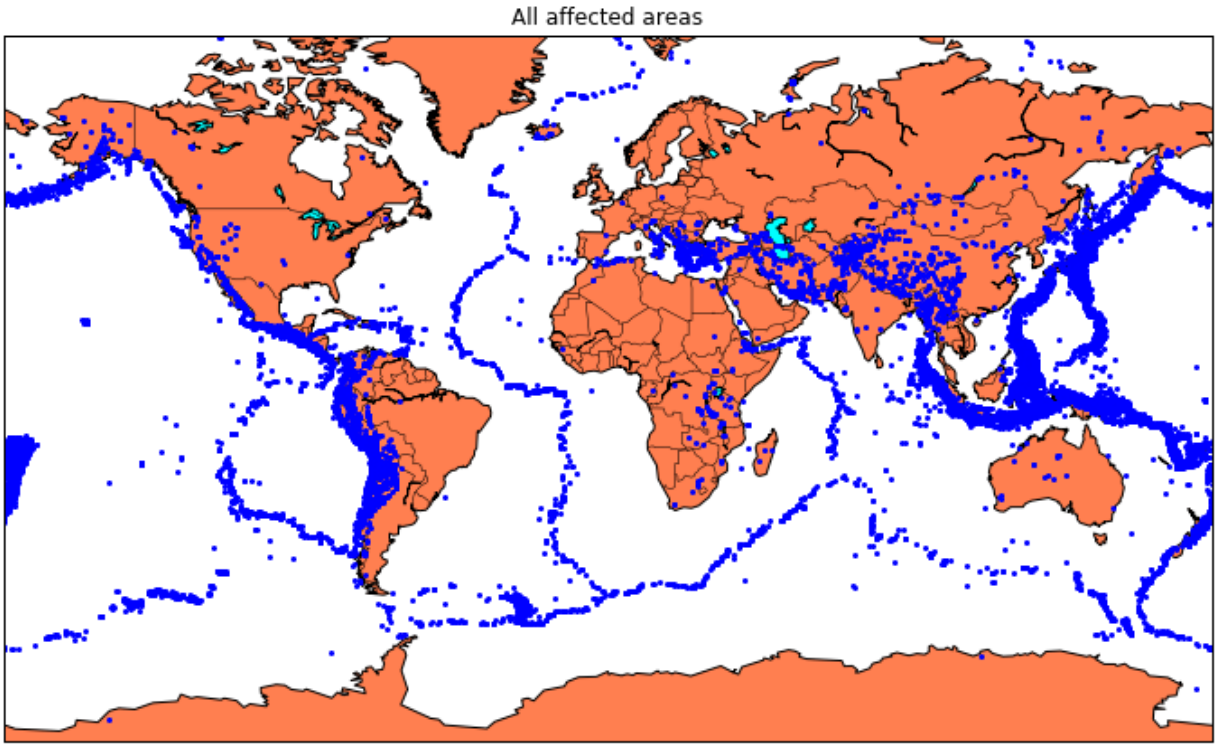
Here, we will visualize the earthquakes that have occurred all around the world.

PROGRAM:

```
from mpl_toolkits.basemap import Basemap

m=Basemap(projection='mill',llcrnrlat=-80,urcrnrlat=80,llcrnrlon=180,urcrnrlon=180,lat_ts=20,resolution='c')

longitudes = data["Longitude"].tolist()
latitudes = data["Latitude"].tolist()
#m = Basemap(width=12000000,height=9000000,projection='lcc',

#resolution=None,lat_1=80.,lat_2=55,lat_0=80,lon_0=-107.)
X,y = m(longitudes,latitudes
```

```
Fig = plt.figure(figsize=(12,10))
Plt.title("All affected areas")
m.plot(x, y, "o", markersize = 2, color = 'blue')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aq
ua')
m.drawmapboundary()
m.drawcountries()
plt.show()
```

OUTPUT:



➢ ***Splitting the datasets:***

Now we will split the dataset into a training and testing set.

PROGRAM:

X = final data[[‘Timestamp’, ‘Latitude’, ‘Longitude’]]
Y = final data[[‘Magnitude’, ‘Depth’]]

From sklearn.cross validation import train test split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2, random state=42)

Print(X train.shape,X test.shape,y train.T, dataset)

OUTPUT:

(18767,3) (34587,3) (45309,3) (48965,2)

We will be using the Random Forest Regressor model to predict the earthquake, here will look for its accuracy

PROGRAM:

```
Reg=RandomForestRegressor(random_state
=42)
Reg.fit(X train, y train)
Reg.predict(X test)
```

OUTPUT:

```
array([[  5.96,  50.97],
       [  5.88,  37.8 ],
       [  5.97,  37.6 ],
       ...,
       [  6.42,  19.9 ],
       [  5.73, 591.55],
       [  5.68,  33.61]])
```

Best fit.score(X test, y test)

OUTPUT:

0.876549876244

NEURAL NETWORK MODEL:

The use of neural network model is to find the accurate velue for the earthquake prediction.

PROGRAM:

```
From keras.models import Sequential
From keras.layers import Dense

Def create_model(neurons, activation, optimizer, loss):
 Model = Sequential()
 Model.add(Dense(neurons, activation=activation, input_shape=(3,)))
 Model.add(Dense(neurons, activation=activation))
 Model.add(Dense(2, activation='softmax'))

 Model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])
```

Return model

CONCLUSION:

Understanding earthquakes and effectively responding to them remains a complex and challenging task, even with the latest technological advancements. However, leveraging the capabilities of machine learning can greatly enhance our comprehension of seismic events. By employing machine learning techniques to analyze seismic data, we can uncover valuable insights and patterns that contribute to a deeper understanding of earthquakes. These insights can subsequently inform more effective

strategies for mitigating risks and responding to seismic events.

As we head towards the future, we might see new technologies that will precisely predict the place and time of the earthquake that will happen.