Name : Mohammed Sauban Samith P

Nm ID : au513521104026

Email Id : saubansamith021@gmail.com

Phase 3 : Development Part 1

Earthquake Prediction Model Using Python

INTRODUCTION:

- Earthquake prediction is necessary application of machine learning, where we can predict the occurrence of an earthquake based on certain parameters.
- The goal of this project is to predict whether an earthquake will occur within a given time period, allowing for proactive measures and infrastructure enhancements.
- When it comes to predicting earthquakes, it should contain an accurate model is required for minimizing potential damage and loss of life.
- We'll start by understanding the importance of Loading of datasets,data preprocessing and cleansing, and then move on to building the model.

Loading Datasets :

- First, we need to acquire the necessary datasets. A common source for earthquake data will get in the kaggle.
- We can use their API to fetch earthquake data.
- The Python request library can be used to handle API requests, and `pandas` can help to fetch data into a structured format for further analysis.

Program :

```
Import pandas as pd

# Load earthquake data into a Pandas DataFrame

Df = pd.read_csv("earthquake_dataset.csv")
```

Preprocessor Data :

- Preprocessing involves handling missing values, scaling, and transforming data into a usable format.
- In the case of earthquake prediction, this may involve feature engineering,future engineering such as extracting relevant information from timestamps, and scaling numerical data for uniformity purpose.

Data Cleansing :

- Data cleansing involves identifying and handling outliers and inconsistencies within the dataset.
- This ensures that the data used for training the model is reliable and accurate.

- Techniques like removing outliers and handling skewed data can be implemented here.

Historical Data Analysis:

Studying historical earthquake data can help identify trends and potential risk areas.

Early Warning Systems:

AI can be used to develop early warning systems that provide a few seconds to minutes of advance notice before an earthquake strikes.

Input 1:

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

import os
print(os.listdir("../input"))
['database.csv']
```

Input 2:

```
data = pd.read_csv("../input/database.csv")
data.head()
```

Input 3:

```
data.columns
```

Output 3:

```
Index(['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'],
      dtype='object')
```

Input 4:

```
data = data[['Date', 'Time', 'Latitude', 'Longitude', 'Depth',
'Magnitude']]
data.head()
```

Output 4:

	Date	Time	Latitude	Longitude	Depth	Magnitude
0	01/02/1965	13:44:18	19.246	145.616	131.6	6.0
1	01/04/1965	11:29:49	1.863	127.352	80.0	5.8
2	01/05/1965	18:05:58	-20.579	-173.972	20.0	6.2
3	01/08/1965	18:49:43	-59.076	-23.557	15.0	5.8
4	01/09/1965	13:32:50	11.938	126.427	15.0	5.8

Input 5:

```
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.timetuple()))
    except ValueError:
        # print('ValueError')
        timestamp.append('ValueError')
```

Input 6:

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

Input 7:

```
final_data = data.drop(['Date', 'Time'], axis=1)
final_data = final_data[final_data.Timestamp != 'ValueError']
final_data.head()
```

Output 7:

	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

All affected areas