EARTHQUAKE PREDICTION MODEL USING PYTHON

NAME : NAVEENA M

COLLEGE : R.V.S COLLEGE OF ENGINEERING

PHASE 2 INOVATION

PROJECT *– EARTHQUAKE PREDICTION MODEL*

CONTEXT

The National Earthquake Information Center (NEIC) determines the location and size of all significant earthquakes that occur worldwide and disseminates this information immediately to national and international agencies, scientists, critical facilities, and the general public. The NEIC compiles and provides to scientists and to the public an extensive seismic database that serves as a foundation for scientific research through the operation of modern digital national and global seismograph networks and cooperative international agreements. The NEIC is the national data center and archive for earthquake information.

CONTENT

This dataset includes a record of the date, time, location, depth, magnitude, and source of every earthquake with a reported magnitude 5.5 or higher since 1965.

DATA SOURCE

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

Python libraries make it very easy for us to handle the data and perform typical and complex tasks with a single line of code.

Pandas

– This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.

Seaborn

– This library is used to draw visualizations

Code

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

 import warnings

warnings.filterwarnings('ignore')
```

Now, let's load the dataset into the panda's data frame for easy analysis.

df = pd.read_csv('dataset.csv')

df head()

	Origin Time	Latitude	Longitude	Depth	Magnitude	Location
0	2021-07-31 09:43:23 IST	29.06	77.42	5.0	2.5	53km NNE of New Delhi, India
1	2021-07-30 23:04:57 IST	19.93	72.92	5.0	2.4	91km W of Nashik, Maharashtra, India
2	2021-07-30 21:31:10 IST	31.50	74.37	33.0	3.4	49km WSW of Amritsar, Punjab, India
3	2021-07-30 13:56:31 IST	28.34	76.23	5.0	3.1	50km SW of Jhajjar, Haryana
4	2021-07-30 07:19:38 IST	27.09	89.97	10.0	2.1	53km SE of Thimphu, Bhutan

STEPS

1. *Data Collection*
2. *Data Processing*
3. *Feature Engineering*
4. *Exploratory Data Analysis (EDA)*
5. *Model Development and Evaluation*
6. *Deployment and Monitoring*
7. *Collaboration and Data Sharing*

The earthquake prediction is a complex and challenging field, and accurate predictions are difficult to achieve. However, by gathering and analyzing high-quality data, we can contribute to ongoing research and potentially improve earthquake monitoring and early warning system.

DATA COLLECTION

- *SEISMOMETERS*

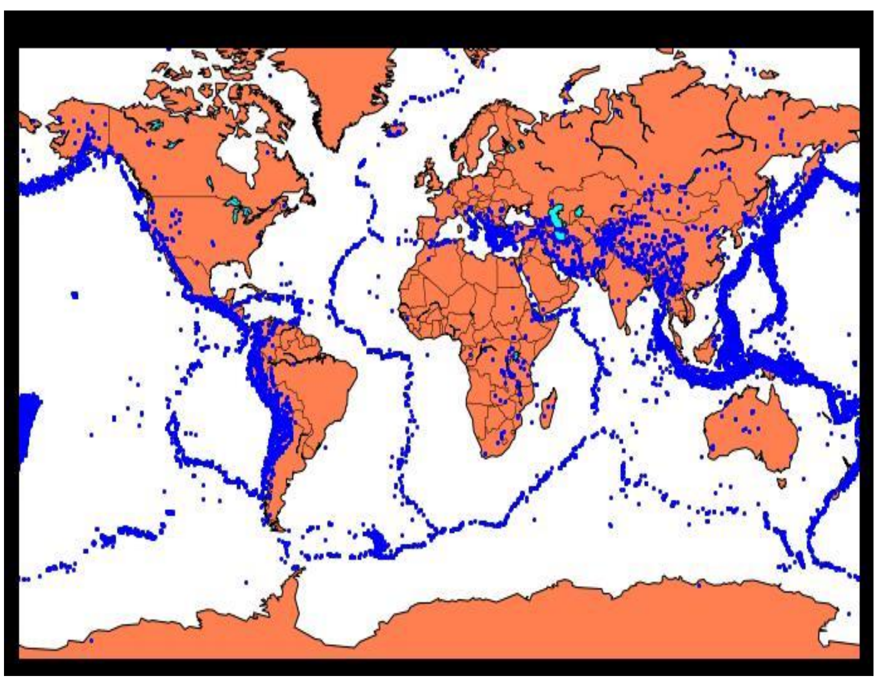
  Obtain seismic data from seismometer networks like USGS or IRIS. You can use their APIs to access real-time and historical earthquake data.

- *GPS DATA*

  Collect GPS data for ground movement measurements. You may need to collaborate with relevant authorities or research institutions to access this data.

- *SATELLITE DATA*

  Satellite imagery and remote sensing data can provide valuable information about surface changes, especially after large earthquakes. Data sources like NASA Earth data can be useful.



DATA PROCESSING

Cleaning and formatting the data to remove noise and inconsistencies, converting data timestamps to a common time zone and combining data from different sources, ensuring alignment in time and location.

- *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*

CODE

df.shape

OUTPUT

(2719, 6)

TYPE OF DATA FORMAT

df.info()

OUTPUT

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2719 entries, 0 to 2718
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Origin Time  2719 non-null   object 
 1   Latitude     2719 non-null   float64
 2   Longitude    2719 non-null   float64
 3   Depth        2719 non-null   float64
 4   Magnitude    2719 non-null   float64
 5   Location     2719 non-null   object 
dtypes: float64(4), object(2)
memory usage: 127.6+ KB
```

Looking at the descriptive statistical measures also gives us some idea regarding the distribution of the data.

CODE

df.describe()

OUTPUT

	Latitude	Longitude	Depth	Magnitude
count	2719.000000	2719.000000	2719.000000	2719.000000
mean	29.939433	80.905638	53.400478	3.772196
std	7.361564	10.139075	68.239737	0.768076
min	0.120000	60.300000	0.800000	1.500000
25%	25.700000	71.810000	10.000000	3.200000
50%	31.210000	76.610000	15.000000	3.900000
75%	36.390000	92.515000	82.000000	4.300000
max	40.000000	99.960000	471.000000	7.000000

From the above description of the dataset, we can conclude that:

- *The maximum magnitude of the Earthquake is 7.*
- *The maximum depth at which the earthquake started is 471 km below the ground.*

FEATURE ENGINEERING

Feature Engineering helps to derive some valuable features from the existing ones. These extra features sometimes help in increasing the performance of the model significantly and certainly help to gain deeper insights into the data.

CODE

splitted = df['Origin Time'].str.split(' ', n=1,expand=True)

df['Date'] = splitted[0]

df['Time'] = splitted[1].str[:-4]

df.drop('Origin Time',axis=1, inplace=True)

df.head()

OUTPUT

	Latitude	Longitude	Depth	Magnitude	Location	Date	Time
0	29.06	77.42	5.0	2.5	53km NNE of New Delhi, India	2021-07-31	09:43:23
1	19.93	72.92	5.0	2.4	91km W of Nashik, Maharashtra, India	2021-07-30	23:04:57
2	31.50	74.37	33.0	3.4	49km WSW of Amritsar, Punjab, India	2021-07-30	21:31:10
3	28.34	76.23	5.0	3.1	50km SW of Jhajjar, Haryana	2021-07-30	13:56:31
4	27.09	89.97	10.0	2.1	53km SE of Thimphu, Bhutan	2021-07-30	07:19:38

Now, let's divide the date column into the day, month, and year columns respectively.

CODE

splitted = df['Date'].str.split('-', expand=True)

df['day'] = splitted[2].astype('int')

df['month'] = splitted[1].astype('int')

df['year'] = splitted[0].astype('int')

df.drop('Date', axis=1,inplace=True)

df.head()

OUTPUT

	Latitude	Longitude	Depth	Magnitude	Location	Time	day	month	year
0	29.06	77.42	5.0	2.5	53km NNE of New Delhi, India	09:43:23	31	7	2021
1	19.93	72.92	5.0	2.4	91km W of Nashik, Maharashtra, India	23:04:57	30	7	2021
2	31.50	74.37	33.0	3.4	49km WSW of Amritsar, Punjab, India	21:31:10	30	7	2021
3	28.34	76.23	5.0	3.1	50km SW of Jhajjar, Haryana	13:56:31	30	7	2021
4	27.09	89.97	10.0	2.1	53km SE of Thimphu, Bhutan	07:19:38	30	7	2021

EXPLORATORY DATA ANALYSIS

EDA is an approach to analyzing the data using visual techniques. It is used to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations.

CODE

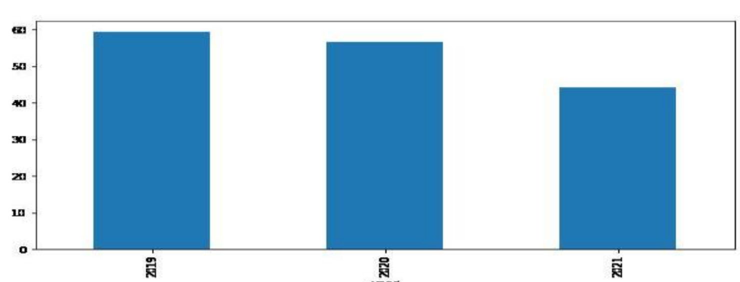
plt.figure(figsize=(10, 5))

x = df.groupby('year').mean()['Depth']

x.plot.bar()

plt.show()

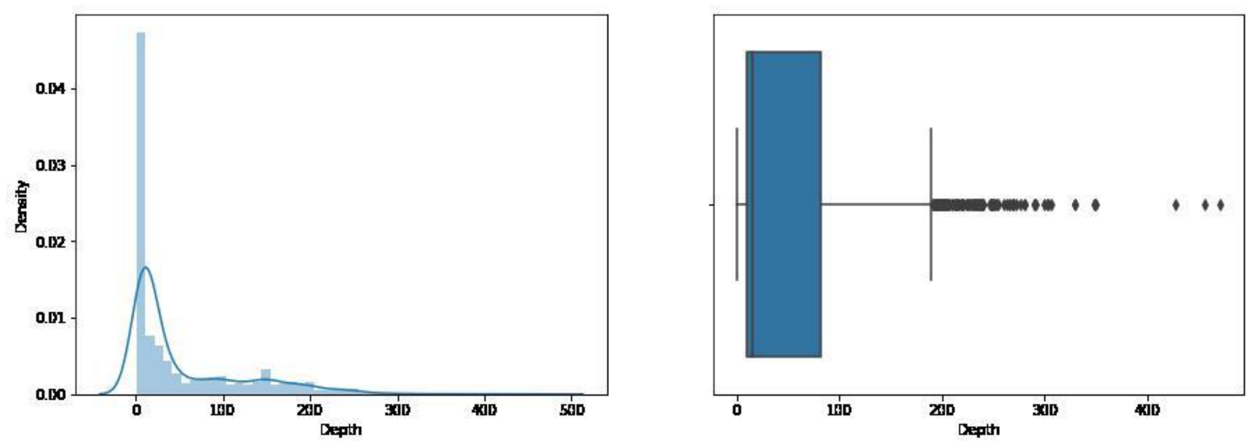
OUTPUT



The depth from which earthquakes are starting is reducing with every passing year.

```
plt.figure(figsize=(10, 5))

sb.lineplot(data=df,x='month',y='Magnitude')

plt.subplots(figsize=(15, 5))

 plt.subplot(1, 2, 1)

sb.distplot(df['Depth'])

 plt.subplot(1, 2, 2)

sb.boxplot(df['Depth'])

plt.show()
```

OUTPUT



MODEL DEVELOPMENT AND EVALUATION

- *Our Python model for earthquake prediction. Options include regression, time series analysis, or neural networks.*
- *Splitting the data into training and testing sets.*
- *Training the model on the training data.*
- *Evaluating the model's performance using metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE)*
- *Adjusting hyper parameters and the model architecture as needed to improve performance.*

DEVELOPMENT AND MONITORING

- *Deploying the trained model in a production environment.*
- *Continuously monitoring incoming data and model performance.*
- *Retraining the model periodically with new data to maintain accuracy.*

COLLABORATION AND DATA SHARING

- *Collaborating with experts in the field for domain-specific insights.*
- *Consider sharing the findings and models with the scientific community.*

PROJECT STRUCTURE

```
.
├── ...
├── data
│   ├── train.csv          # Original training set decomposed into feature set
```

```
│   ├── test.csv          # Testing signal decomposed into feature set
│   └── results.csv       # Modeling results prepared for submission
├── notebooks
│   └── earthquake.ipynb  # Misc
├── earthquake
│   ├── config.py         # Configuration parameters
│   ├── ga.py             # GA for feature selection
│   ├── generator.py      # Feature engineering
│   ├── submission.py     # Make prediction and prepare file for submission
│   └── utils.py          # Helpers
└── ...
```

PHASE 2 CONCLUSION

In the Phase 2, we have summarized the advanced techniques such as hyper parameter tuning and feature engineering to improve the prediction model's performance.