EARTHQUAKE PREDICTION MODEL USING PYTHON

PHASE3 PROJECT

PREPARED BY

S.SNEKHA,
510521205045,
B H A R A T H I D A S A N
ENGINEERING COLLEGE,
PHASE3 PROJECT.

ITDRODUCTION

T

Loading and processing the dataset is a crucial step in earthquake prediction, as it lays the foundation for building accurate models and making informed decisions. In this critical phase of the research, we gather, prepare, and manipulate the data to extract meaningful insights.

The dataset may encompass a wide range of variables, such as seismic activity, geological features, historical earthquake records, and more. Proper data preprocessing involves tasks like cleaning, normalization, and feature engineering, which aim to enhance the dataset's quality and relevance. By preparing the data effectively, we can pave the way for advanced machine learning models and statistical analyses to predict earthquakes and contribute to disaster mitigation efforts.

Loading and processing the dataset is a crucial step in earthquake prediction, as it lays the foundation for building accurate models and making informed decisions. In this critical phase of the research, we gather, prepare, and manipulate the data to extract meaningful insights. The dataset may encompass a wide range of variables, such as seismic activity, geological features, historical earthquake records, and more. Proper data preprocessing involves tasks like cleaning, normalization, and feature engineering, which aim to enhance the dataset's quality and relevance. By preparing the data effectively, we can pave the way for advanced machine learning models and statistical analyses to predict earthquakes and contribute to disaster mitigation efforts.

GIVEN DATASET

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date	Time	Latitude	Longitude	Type	Depth	Depth Erro	Depth Seis	Magnitude	Magnitude	ID	Source	Location So	Magnitude	Status
2	1/2/1965	13:44:18	19.246	145.616	Earthquake	131.6			6	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
3	1/4/1965	11:29:49	1.863	127.352	Earthquake	80			5.8	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
4	1/5/1965	18:05:58	-20.579	-173.972	Earthquake	20			6.2	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
5	1/8/1965	18:49:43	-59.076	-23.557	Earthquake	15			5.8	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
6	1/9/1965	13:32:50	11.938	126.427	Earthquake	15			5.8	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
7	1/10/1965	13:36:32	-13.405	166.629	Earthquake	35			6.7	MW	ISCGEM860	ISCGEM	ISCGEM	ISCGEM	Automatic
8	1/12/1965	13:32:25	27.357	87.867	Earthquake	20			5.9	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
9	1/15/1965	23:17:42	-13.309	166.212	Earthquake	35			6	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
10	1/16/1965	11:32:37	-56.452	-27.043	Earthquake	95			6	MW	ISCGEMSUP	ISCGEMSUP	ISCGEM	ISCGEM	Automatic
11	1/17/1965	10:43:17	-24.563	178.487	Earthquake	565			5.8	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
12	1/17/1965	20:57:41	-6.807	108.988	Earthquake	227.9			5.9	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
13	1/24/1965	0:11:17	-2.608	125.952	Earthquake	20			8.2	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
14	1/29/1965	9:35:30	54.636	161.703	Earthquake	55			5.5	MW	ISCGEM861	ISCGEM	ISCGEM	ISCGEM	Automatic
15	2/1/1965	5:27:06	-18.697	-177.864	Earthquake	482.9			5.6	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic
16	2/2/1965	15:56:51	37.523	73.251	Earthquake	15			6	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic
17	2/4/1965	3:25:00	-51.84	139.741	Earthquake	10			6.1	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic
18	2/4/1965	5:01:22	51.251	178.715	Earthquake	30.3			8.7	MW	OFFICIAL19	OFFICIAL	ISCGEM	OFFICIAL	Automatic
19	2/4/1965	6:04:59	51.639	175.055	Earthquake	30			6	MW	ISCGEMSUP	ISCGEMSUP	ISCGEM	ISCGEM	Automatic
20	2/4/1965	6:37:06	52.528	172.007	Earthquake	25			5.7	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic
21	2/4/1965	6:39:32	51.626	175.746	Earthquake	25			5.8	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic
22	2/4/1965	7:11:23	51.037	177.848	Earthquake	25			5.9	MW	ISCGEMSUP	ISCGEMSUP	ISCGEM	ISCGEM	Automatic
23	2/4/1965	7:14:59	51.73	173.975	Earthquake	20			5.9	MW	ISCGEM859	ISCGEM	ISCGEM	ISCGEM	Automatic

LOADING AND PREPROCESSING DATASET STEPS

Loading and preprocessing a dataset for earthquake prediction involves several essential steps. Here's an overview of the process:

1. Data Collection: Gather earthquake-related data, which typically includes seismic sensor readings, geological information, and historical earthquake records. This data can be obtained from various sources like government agencies, research institutions, or public databases.
2. Data Cleaning:
   - Remove duplicates and irrelevant data points.
   - Handle missing data by imputing or removing them.
   - Check for outliers and decide whether to correct or remove them.
3. Feature Selection/Engineering:
   - Select relevant features for earthquake prediction, such as seismic readings, fault line locations, geological features, and historical earthquake data.
   - Create new features if needed, like earthquake frequency in a region over time.
4. Data Transformation:
   - Standardize or normalize numerical features to have a consistent scale.
   - Encode categorical variables if present.
   - Perform time-series-specific data transformations if your data contains temporal information.

5. Splitting the Data:

- Divide the dataset into training, validation, and testing sets to assess the model's performance accurately.

6. Data Augmentation (Optional):

- Generate additional data points or variations to improve the model's robustness, especially if the dataset is small.

7. Balancing the Data (Optional):

- If your dataset is imbalanced (e.g., rare earthquakes), consider techniques like oversampling, undersampling, or using weighted loss functions.

8. Data Preprocessing for Time Series(If applicable):

- Ensure that the data is sorted by timestamp.
- Consider creating sequences or windows of data for training recurrent neural networks (RNNs) or similar models.

9. Data Visualization:

- Visualize the dataset to gain insights into its characteristics and potential patterns.

10. Data Splitting:

- Split the data into training, validation, and test sets.

11. Scaling:

- Apply scaling to ensure that features have similar scales for modeling.

12. Model-Specific Preprocessing:

- Some models may require specific preprocessing steps. For example, a convolutional neural network (CNN) might need image data, while an LSTM might need sequence data.

13. Save Preprocessing Steps:

- Save preprocessing information (e.g., mean and standard deviation for scaling) to apply to future data during inference.

These steps are crucial for preparing your dataset for earthquake prediction. The choice of techniques and tools may vary based on the specific requirements of your project and the machine learning model you plan to use.

loading and preprocessing the dataset program

```
import pandas as pd
Preprocessing steps
# loading the dataset into a dataframe
df = pd.read_csv(r'/kaggle/input/usgs-dataset-for-earthquake-30-days/Earthquake_of_last_30 days.csv')
# Display the first 5 rows of data
df.head()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...
	updated	place	type	horizontalError	depthError				magError	magNst	status
	locationSource	magSource									
0	2023-02-14T21:31:52.124Z		60.828300		-151.841200		85.00	2.20	ml	NaN	
	NaN	NaN	1.6100	...	2023-02-14T21:35:21.982Z		33 km WNW of Nikiski, Alaska				
	earthquake	NaN	2.10	NaN	NaN	automatic			ak	ak	
1	2023-02-14T20:45:56.420Z		19.254333		-155.410828		31.32	2.27	ml	41.0	
	139.00	NaN	0.1500	...	2023-02-14T20:51:26.040Z		9 km NE of Pahala, Hawaii				
	earthquake		0.66	0.81	2.790	10.0	automatic		hv	hv	
2	2023-02-14T20:45:12.919Z		38.146900		-117.982000		7.30	1.90	ml	11.0	
	110.46	0.02000	0.1385	...	2023-02-14T21:04:41.699Z		Nevada	earthquake		NaN	
	1.30	0.210	9.0	reviewed	nn	nn					
3	2023-02-14T20:43:53.796Z		63.898700		-148.655300		82.40	1.30	ml	NaN	
	NaN	NaN	0.5700	...	2023-02-14T20:46:28.820Z		15 km ENE of Healy, Alaska				
	earthquake	NaN	1.50	NaN	NaN	automatic			ak	ak	
4	2023-02-14T20:43:40.220Z		33.324167		-116.757167		12.42	0.89	ml	23.0	
	67.00	0.08796	0.1700	...	2023-02-14T21:22:42.029Z		9km N of Lake Henshaw, CA				
	earthquake		0.26	1.00	0.133	8.0	reviewed	ci	ci		

5 rows × 22 columns

```
# Display the last 5 rows of data
df.tail()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...
	updated	place	type	horizontalError	depthError	magError	magNst	status			
	locationSource	magSource									
10148	2023-01-15T21:45:41.552Z			41.507200	19.986200	10.000	4.10	mb	32.0		
	40.0	0.1850	0.66	...	2023-01-26T02:26:34.175Z		Albania	earthquake	4.68		
	1.972	0.301	3.0	reviewed	us	us					
10149	2023-01-15T21:42:53.540Z			33.323167	-116.900333	19.470	1.28	ml	42.0		
	39.0	0.1282	0.16	...	2023-01-17T23:04:14.863Z		5km SW of Palomar Observatory,				
CA	earthquake	0.18	0.500	0.223	23.0	reviewed	ci	ci			
10150	2023-01-15T21:42:37.119Z			62.524300	-149.442100	50.900	1.10	ml	NaN		
	NaN	NaN	0.48	...	2023-01-30T21:47:30.436Z		35 km ENE of Chase, Alaska				
	earthquake	NaN	0.600	NaN	NaN	reviewed	ak	ak			
10151	2023-01-15T21:39:04.248Z			63.160500	-150.495600	112.900	1.70	ml	NaN		
	NaN	NaN	0.40	...	2023-01-30T21:47:30.191Z		74 km SE of Denali National Park,				
Alaska	earthquake	NaN	0.400	NaN	NaN	reviewed	ak	ak			
10152	2023-01-15T21:37:21.608Z			41.451300	19.999500	16.295	4.90	mb	105.0		
	37.0	0.1460	0.74	...	2023-02-12T22:12:07.662Z		9 km SW of Klos, Albania				
	earthquake	4.32	3.590	0.072	60.0	reviewed	us	us			

5 rows × 22 columns

```
# Display the shape of the data (number of rows and columns)
df.shape
(10153, 22)
# Display the column names
df.columns
Index(['time', 'latitude', 'longitude', 'depth', 'mag', 'magType', 'nst',
       'gap', 'dmin', 'rms', 'net', 'id', 'updated', 'place', 'type',
       'horizontalError', 'depthError', 'magError', 'magNst', 'status',
       'locationSource', 'magSource'],
      dtype='object')
 # Display the data types of each column
df.dtypes
time              object
latitude         float64
longitude        float64
depth            float64
mag              float64
magType           object
nst              float64
gap              float64
dmin             float64
rms              float64
net               object
id                object
```

For bivariate analysis, we can look at the relationship between two variables in the dataset.

Bivariate Analysis:

```
import matplotlib.pyplot as plt
# Scatter plot of depth vs magnitude
plt.scatter(df['depth'], df['mag'])
plt.xlabel('Depth')
plt.ylabel('Magnitude')
plt.title('Scatter plot of Depth vs Magnitude')
plt.show()
# Box plot of earthquake magnitude by type
df.boxplot(column='mag', by='type')
plt.title('Box plot of Earthquake Magnitude by Type')
plt.suptitle('')
plt.show()
```

Multivariate Analysis:

```
import seaborn as sns
import matplotlib.pyplot as plt
# Select the numerical columns for correlation analysis
numeric_cols = ['latitude', 'longitude', 'depth', 'mag', 'nst', 'gap', 'rms', 'horizontalError', 'depthError']
# Create correlation matrix
corr_matrix = df[numeric_cols].corr()
# Plot heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

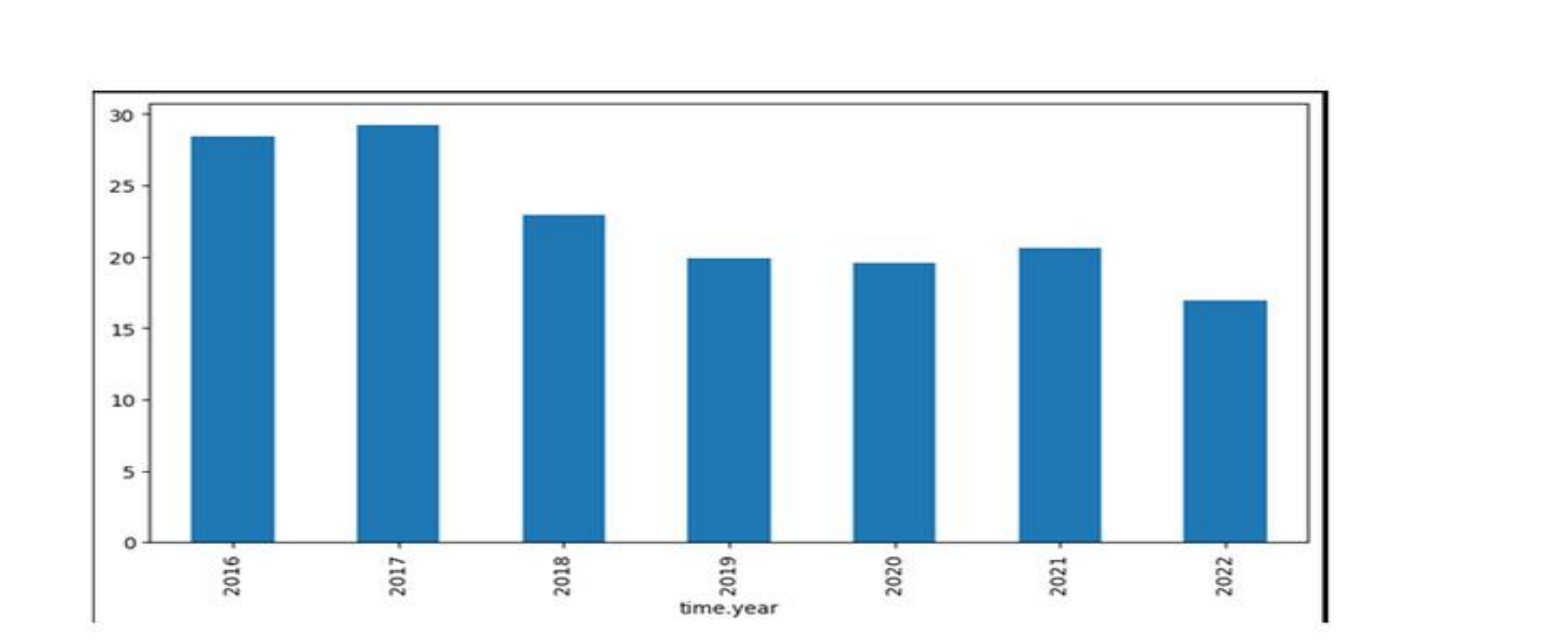
Data Visualization:

```
import seaborn as sns
import matplotlib.pyplot as plt
# Set style for all visualizations
sns.set_style("darkgrid")
# Scatter plot to show relationship between magnitude and depth
sns.scatterplot(data=df, x="mag", y="depth")
# Bar plot to show distribution of magnitudes
sns.histplot(data=df, x="mag")
# Box plot to show distribution of magnitudes by type
sns.boxplot(data=df, x="magType", y="mag")
# Heatmap to show correlation between variables
corr = df.corr()
sns.heatmap(corr, annot=True)
# Pairplot to show scatterplots of all possible variable combinations
sns.pairplot(df)
# Show all visualizations
plt.show()
```

DATA VISUALIZATION

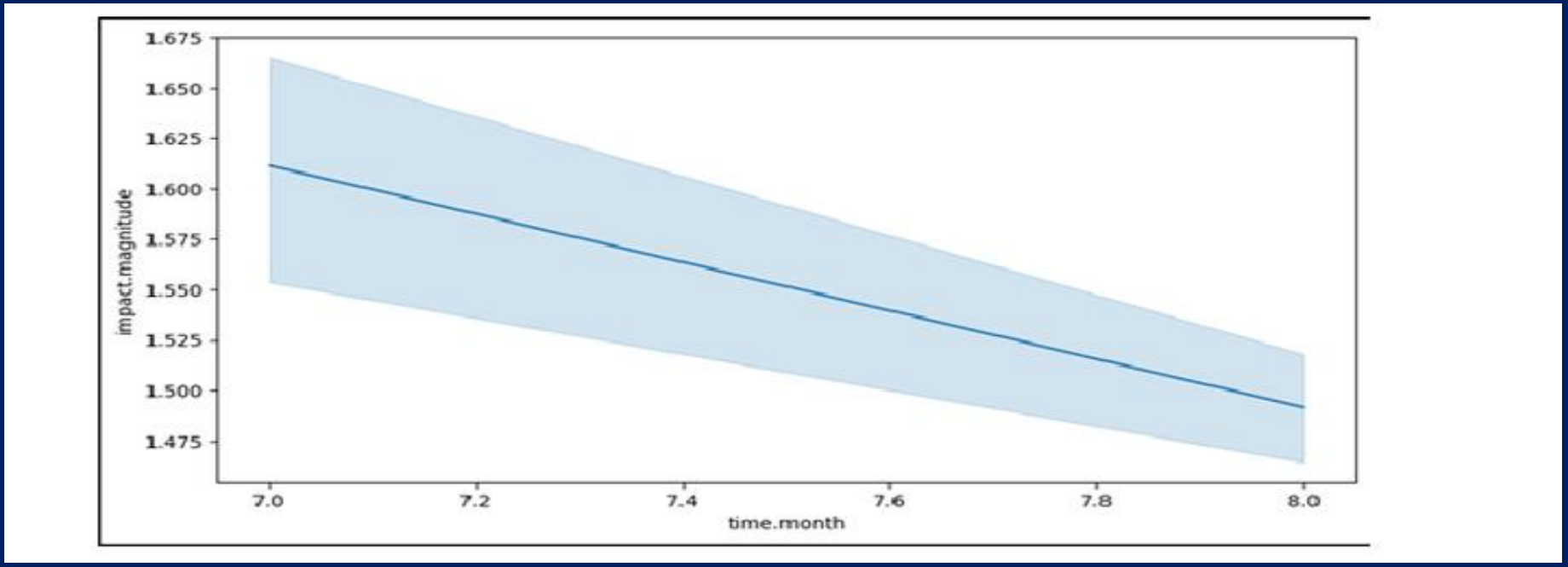
```
import pandas as pdd
import numpy as npp
import matplotlib.pyplot as pltt
import seaborn as sbb
pltt.figure(figsize=(10, 5))
x1 = df1.groupby('time.year').mean()['location.depth']
x1.plot.bar()
pltt.show()
```

Output:



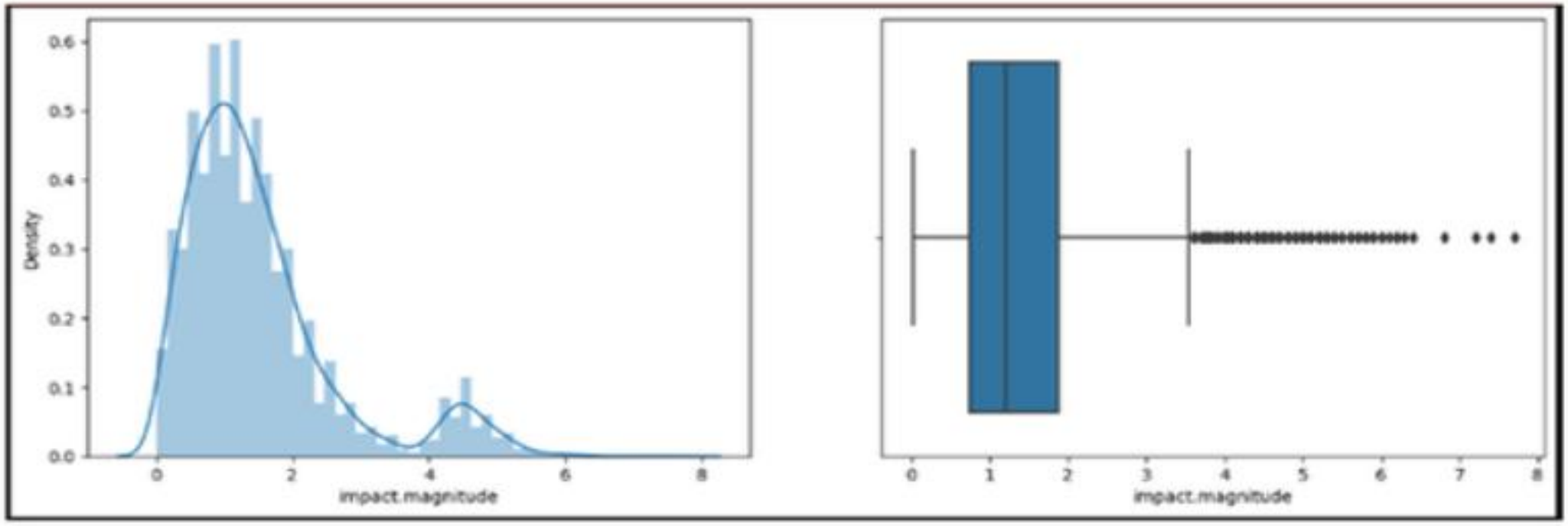
```
pltt.figure(figsize=(10, 5))
sbb.lineplot(data=df1,
   x='time.month',
   y='impact.magnitude')
pltt.show()
```

output:



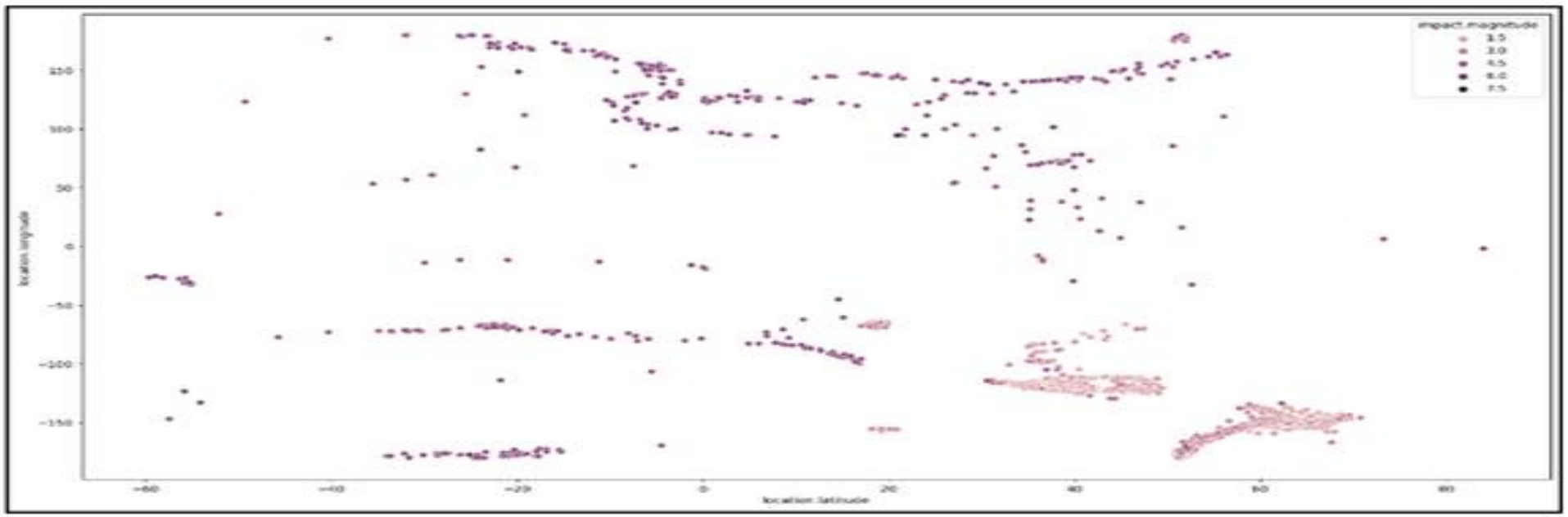
```
pltt.subplots(figsize=(15, 5))
pltt.subplot(1, 2, 1)
sbb.distplot(df1['impact.magnitude'])
pltt.subplot(1, 2, 2)
sbb.boxplot(df1['impact.magnitude'])
pltt.show()
```

output:



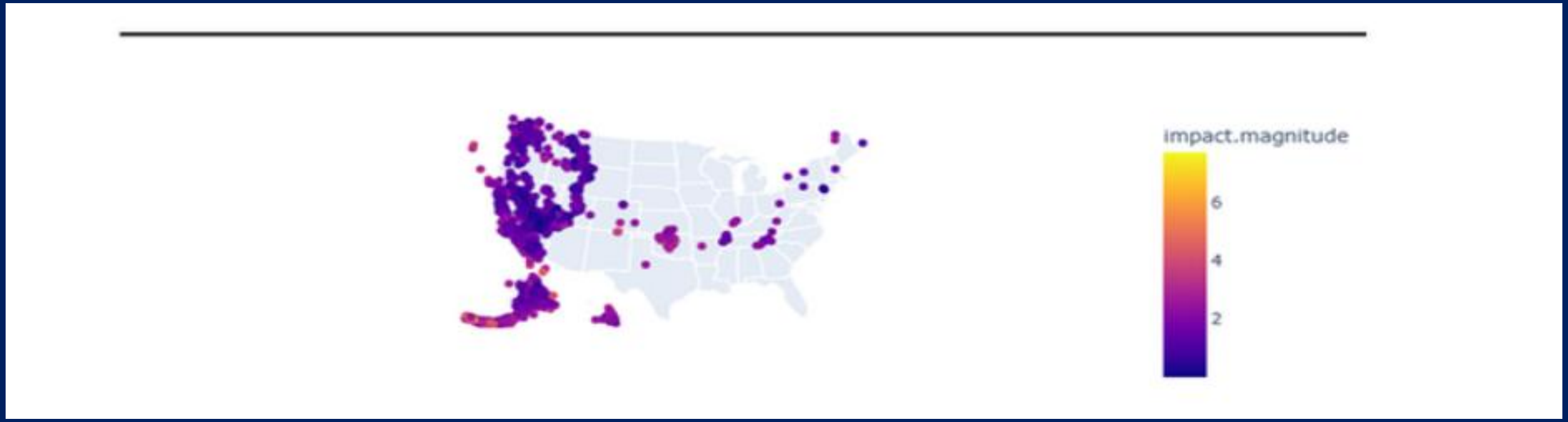
```
pltt.figure(figsize=(20, 10))
sbb.scatterplot(data=df1,
   x='location.latitude',
   y='location.longitude',
   hue='impact.magnitude')
pltt.show()
```

output:



```
import plotly.express as pxx
import pandas as pdd
 fig_w = pxx.scatter_geo(df1, lat='location.latitude',
   lon='location.longitude',
   color="impact.magnitude",
   scope='usa')
fig_w.show()
```

output:



CONCLUSION

In conclusion, the loading and preprocessing of datasets for earthquake prediction is a foundational process in developing accurate and reliable predictive models. It involves several key steps, including data collection, cleaning, integration, feature extraction, normalization, splitting, and model training. The dataset must be carefully prepared to ensure data quality and suitability for machine learning models.

The specific tools and techniques used in this process will depend on the nature of the data sources, the chosen infrastructure (whether IoT or cloud-based), and the project's requirements. Effective data loading and preprocessing are essential for the successful development and deployment of earthquake prediction models. These models have the potential to provide valuable insights and early warnings, contributing to disaster management and public safety.

THANKS