

To begin, make sure you have your earthquake dataset available. Depending on its format (CSV, Excel, etc.), you can use libraries like pandas in Python to load the data. Once loaded, consider normalizing numerical features, handling missing values, and encoding categorical variables if necessary. This ensures your data is ready for training the prediction model.

1. Load the Dataset:

Use a library like pandas to read your earthquake dataset. For example:

```
```python
import pandas as pd

data = pd.read_csv('your_dataset.csv')
```
```

2. Explore the Data:

Get a sense of your data by checking its structure, columns, and summary statistics:

```
```python
print(data.head())

print(data.info())
```
```

3. Handle Missing Values:

Address any missing values in your dataset. You can drop or impute them based on your analysis:

```
```python
data = data.dropna() # or use fillna() for imputation
```
```

4. Normalize Numerical Features:

If you have numerical features with different scales, normalize them to ensure consistent training:

```
```python
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data[['feature1', 'feature2']] = scaler.fit_transform(data[['feature1', 'feature2']])
```
```

``5. Encode Categorical Variables:

If your dataset contains categorical variables, encode them for the model to understand:

``python

```
data = pd.get_dummies(data, columns=['categorical_feature'])
```

``These steps set the foundation for building your earthquake prediction model. Next, you can move on to selecting a suitable model, splitting the data for training and testing, and initiating the model training process.