**Earthquake Prediction Model using Python**

**Problem Definition and Design Thinking:**

Explore the key features of earthquake data and design an object for those features, such as date, time, latitude, longitude, depth, and magnitude. Before developing the prediction model, visualize the data on a world map to display a complete overview of where the earthquake frequency will be higher. Split the data into a training set and a test set for validation. Lastly, build a neural network to fit the data from the training set.

**Problem Definition:**

To work with earthquake data, design an object that encapsulates essential attributes such as date, time, latitude, longitude, depth, and magnitude. Afterward, visualize the data on a world map to gain insights into regions with higher earthquake frequency. To prepare for model development, split the dataset into training and test sets for validation purposes. Finally, construct a neural network to train on the training set and make predictions based on this structured earthquake data.

Design Thinking:

1.Define the object of earthquake data: To represent earthquake data, create a python class that include date, time, latitude, longitude, depth and magnitude.

2.Data visualization on world map: To visualize earthquake data on a world map, use libraries like “matplotlib” and “basemap”. First, extract latitude and longitude data from the earthquake dataset and plot it on a map.

3.Split Data into Training and Test Sets: Before building a neural network for prediction, split the earthquake data into training and test sets. The training set will be used to train the model, while the test set will be used to evaluate its performance. Use Python libraries like `scikit-learn`.

4: Build a Neural Network Model: To build a neural network model for earthquake magnitude predicting a deep learning library like TensorFlow and Keras are used.

**Phases of Development:**

Earthquake prediction is a complex task that involves various factors. While I can't provide a complete design for such a system, I can outline a basic approach you might take:

**1. Data Collection:**

* Gather earthquake data from reliable sources like USGS.
* Collect additional relevant data, such as tectonic plate movements, historical seismic activity, and geological features.

**2. Data Pre-processing:**

* Clean and pre-process the collected data.
* Handle missing values and outliers.
* Convert data into a suitable format for analysis.

**3. Feature Engineering:**

* Extract relevant features from the data.
* Consider features like magnitude, depth, location, and temporal patterns. - Create additional features that might capture important information.

**4. Exploratory Data Analysis (EDA):**

* Analyse the distribution of seismic events.
* Explore correlations between different features.
* Visualize data to identify patterns or anomalies.

**5. Model Selection:**

- Choose a machine learning model suitable for time-series prediction, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), or traditional regression models. - Consider the trade-offs between model complexity and interpretability.

**6. Training the Model:**

* Split the data into training and testing sets.
* Train the chosen model on the training data. - Validate the model using the testing data.

**7. Hyper parameter Tuning:**

* Optimize the model's hyper parameters to improve performance.
* Use techniques like grid search or random search for this purpose.

**8. Evaluation:**

* Evaluate the model's performance using appropriate metrics (e.g., mean squared error for regression).
* Assess the model's ability to predict earthquakes accurately.

**9. Deployment:**

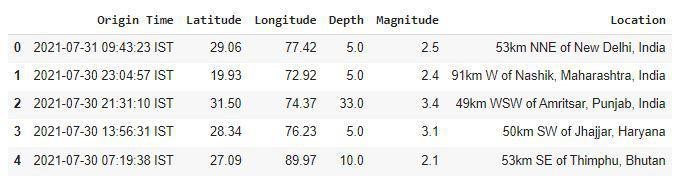
* Deploy the trained model to a suitable environment.
* Integrate the model into a system that can continuously update and retrain itself with new data.

**10. Monitoring and Maintenance:**

* Implement monitoring mechanisms to detect model degradation.
* Regularly update the model with new earthquake data to ensure its relevance.

Earthquake prediction is a highly challenging task, and the accuracy of predictions may vary. This outline provides a general guide, but you might need domain-specific expertise and collaboration with seismologists for a more accurate and reliable system.

**Dataset Used:**



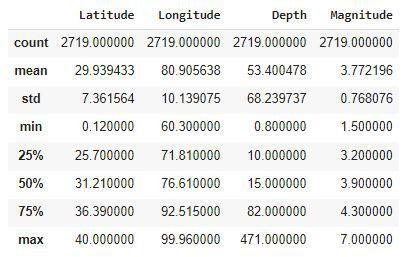
**Analysis of the dataset:**

**Describing the CSV file**

Code:

df.describe()

Output:



**Feature Engineering:**

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:



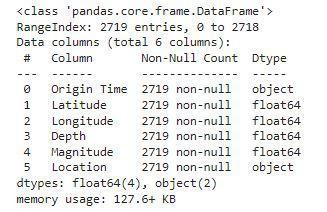
**Info Gathering:**

Code:

df.shape df.info()

Output:

(2719, 6)



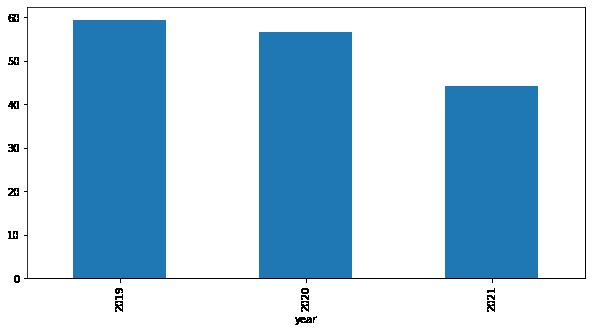
**Data Visualization:**

Code:

plt.figure(figsize=(10, 5)) x = df.groupby('year').mean()['Depth'] x.plot.bar()

plt.show()

Output:



**Monthly Analysis:**

Code :

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

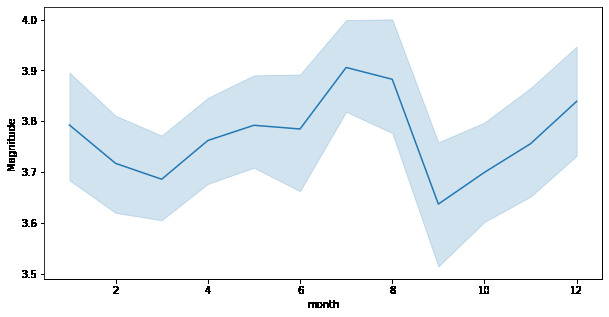
sb.lineplot(data=df,

x='month',

y='Magnitude')

plt.show()

Output:



Here we can observe that the changes of an earthquake with higher magnitude are more observed during the season of monsoon.

**Distribution Graph:**

Code:

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

plt.subplot(1, 2, 1)

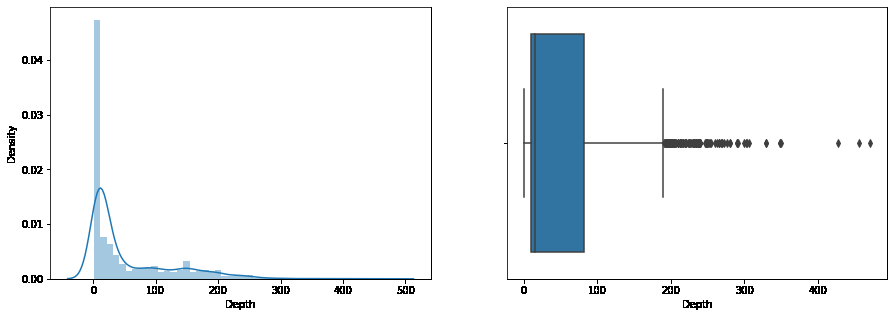
sb.distplot(df['Depth'])

plt.subplot(1, 2, 2)

sb.boxplot(df['Depth'])

plt.show()

Output:



From the distribution graph, it is visible that there are some outliers that can be confirmed by using the boxplot. But the main point to observe here is that the distribution of the depth at which the earthquake rises is left-skewed.

**Normal Distribution:**

Code:

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

plt.subplot(1, 2, 1)

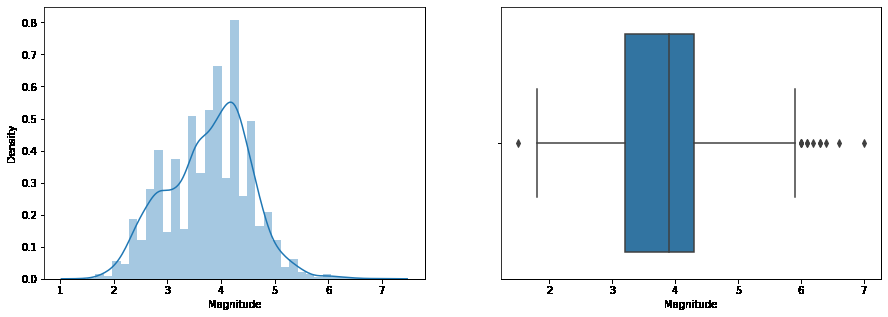
sb.distplot(df['Magnitude'])

plt.subplot(1, 2, 2)

sb.boxplot(df['Magnitude'])

plt.show()

Output:



**Scatterplot:**

Code:

plt.figure(figsize=(10, 8))

sb.scatterplot(data=df,

x='Latitude',

y='Longitude',

hue='Magnitude')

plt.show()

Output:

