**EARTHQUAKE PREDICTION MODEL USING PYTHON**

**AI\_PHASE3**

**NAME: JOSWA S**

**REG.NO: 610821205303**

**DEVELOPMENT – PART 1**

**LOADING AND PREPROCESSING THE DATASET**

**1. Dataset Loading and printing dataset**

## *import numpy as np*

## *import pandas as pd*

## *import matplotlib.pyplot as plt*

## *# Dataset from https://www.kaggle.com/datasets/usgs/earthquake-database*

## *data = pd.read\_csv("earthquake-database/database.csv")*

## *data*

# **[OUTPUT]**

**2. DATASET PREPROCESSING**

**THE COLUMNS THAT ARE NEED FOR MODEL AND PREDICTION**

## *# THE COLUMNS THAT ARE NEED FOR MODEL AND PREDICTION*

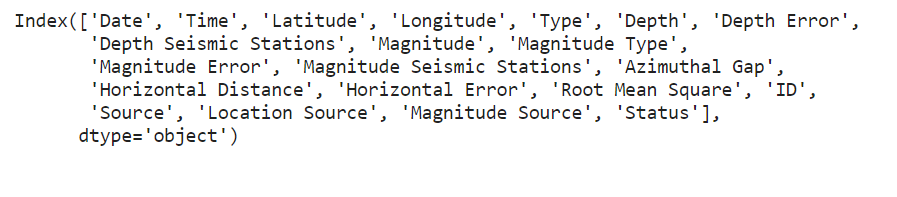
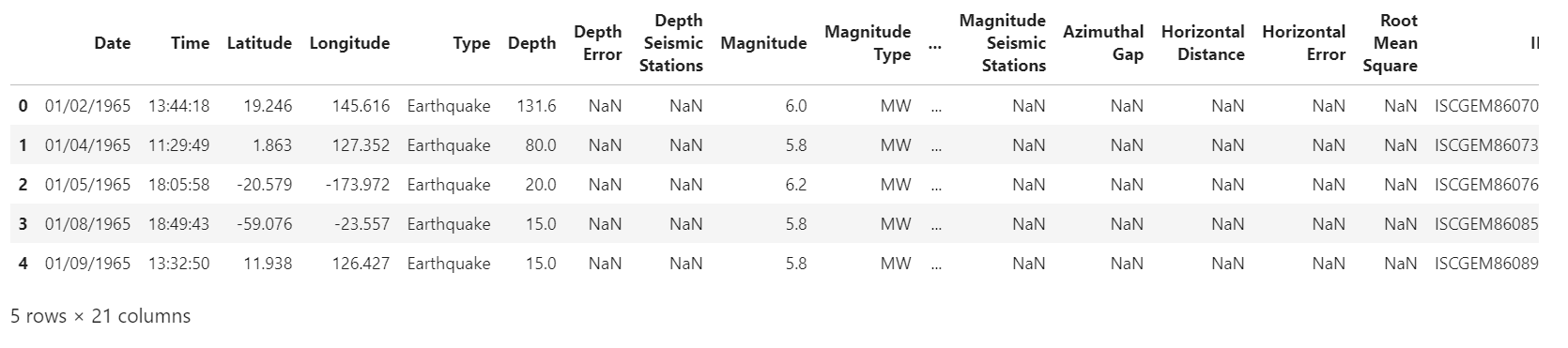
## *data.head()*

## **[OUTPUT]**

**Column Names**

## #Column Names

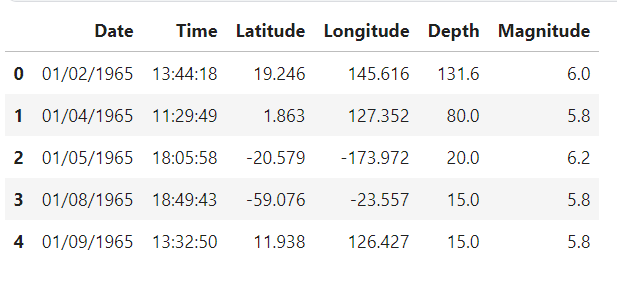
## data.columns

**[OUTPUT]**

# The main features from earthquake data creating a object namely, Date, Time, Latitude, Longitude, Depth, Magnitude

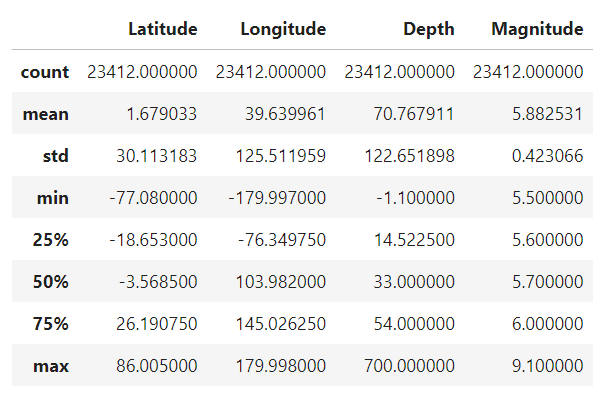
data = data[['Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude']]

data.head()

**[OUTPUT]**

data.describe()

**[OUTPUT]**



**Here, the data is random we need to scale according to inputs to the model. So, we convert given Date and Time to Unix time which is in seconds and a numeral. This can be easily used as input for the network we built**

import datetime

import time

# Create a list to store Unix timestamps

timestamp = []

# Iterate through the "Date" and "Time" columns

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:

timestamp.append('ValueError')

# Create a Pandas Series from the timestamp list

timeStamp = pd.Series(timestamp)

# Add the "Timestamp" column to the DataFrame

data['Timestamp'] = timeStamp.values

# Drop the "Date" and "Time" columns

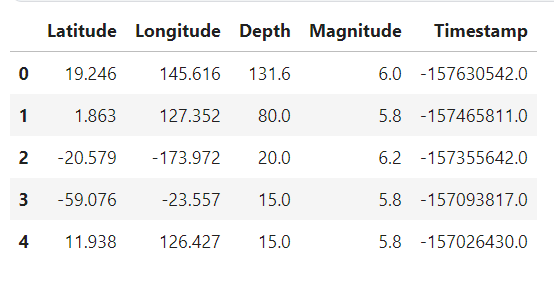
final\_data = data.drop(['Date', 'Time'], axis=1

# Remove rows with 'ValueError' in the "Timestamp" column

final\_data = final\_data[final\_data['Timestamp'] != 'ValueError']

# Display the first few rows of the final dataset

final\_data.head()

 **[OUTPUT]**