EARTHQUAKE PREDICTION USING AI, MACHINE LEARNING AND PYTHON PROGRAMMING LANGUAGE

DATE	11 October 2023
TEAM ID	PROJ-212172_TEAM_I
PROJECT NAME	Earthquake Prediction Model using PYTHON
MAXIMUM MARK	

Introduction:

Data preprocessing is an important step in data mining process.It refers to the cleaning,transforming and integrating of data in order to make it ready for the analysis.

The goal of data preprocessing is to improve the quality of data and make it more suitable for the specific data mining task.

Importing libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

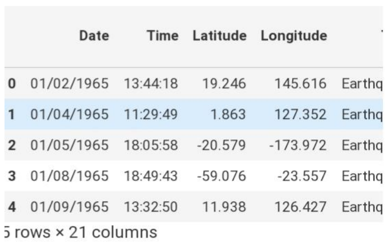
Dataset:

```
df=pd.read_csv("database.csv")
```

Code:

```
df.head()
```

- The head() method returns a specified number of rows, string from the top. The head() method returns the first 5 rows if a number is not specified. Note: The column names will also be returned, in addition to the specified rows.
- If we didn't mention the parameter then all the rows are returned.



	Date	Time	Latitude	Longitude	
0	01/02/1965	13:44:18	19.246	145.616	Earthq
1	01/04/1965	11:29:49	1.863	127.352	Earthq
2	01/05/1965	18:05:58	-20.579	-173.972	Earthq
3	01/08/1965	18:49:43	-59.076	-23.557	Earthq
4	01/09/1965	13:32:50	11.938	126.427	Earthq

5 rows × 21 columns

df.isnull().sum()

The output :

```
Date                            0
Time                            0
Latitude                        0
Longitude                       0
Type                            0
Depth                           0
Depth Error                 18951
Depth Seismic Stations      16315
Magnitude                       0
Magnitude Type                  3
Magnitude Error             23085
Magnitude Seismic Stations  20848
Azimuthal Gap               16113
Horizontal Distance         21808
Horizontal Error            22256
Root Mean Square             6060
ID                              0
Source                          0
Location Source                 0
Magnitude Source                0
Status                          0
```

```
dtype: int64
```

Code:

```
df['Depth Error'].fillna(df['Depth Error'].mean(),inplace=True)
df['Magnitude Error'].fillna(df['Magnitude Error'].median(),inplace=True)
df['Depth Seismic Stations'].fillna(df['Depth Seismic
Stations'].min(),inplace=True)
df['Magnitude Seismic Stations'].fillna(df['Magnitude Seismic
Stations'].max(),inplace=True)
df.isnull().sum()
```

Fillna:

- The fillna() method replaces the NULL values with a specified value.
- The specified value find by using the mode() method.
- The fillna() method returns a new DataFrame object unless the inplace parameter is set to True , in that case the fillna() method does the replacing in the original DataFrame instead.

The output for the above code is,

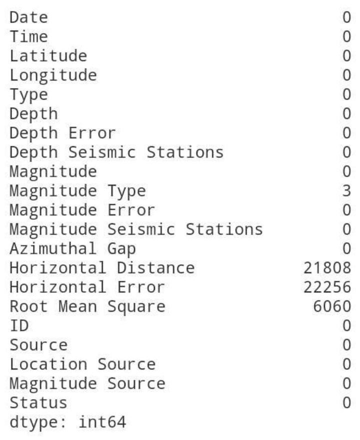
```
Date                              0
Time                              0
Latitude                          0
Longitude                         0
Type                              0
Depth                             0
Depth Error                       0
Depth Seismic Stations            0
Magnitude                         0
Magnitude Type                    3
Magnitude Error                   0
Magnitude Seismic Stations        0
Azimuthal Gap                 16113
Horizontal Distance           21808
Horizontal Error              22256
Root Mean Square               6060
ID                                0
Source                            0
Location Source                   0
Magnitude Source                  0
Status                            0
 dtype: int6
```

Code:

```
df['Azimuthal Gap'].fillna(df['Azimuthal Gap'mean(),inplace=True)
```

```
df.isnull().sum()
```

The output is,



```
Date                             0
Time                             0
Latitude                         0
Longitude                        0
Type                             0
Depth                            0
Depth Error                      0
Depth Seismic Stations           0
Magnitude                        0
Magnitude Type                   3
Magnitude Error                  0
Magnitude Seismic Stations       0
Azimuthal Gap                    0
Horizontal Distance          21808
Horizontal Error             22256
Root Mean Square              6060
ID                               0
Source                           0
Location Source                  0
Magnitude Source                 0
Status                           0
dtype: int64
```

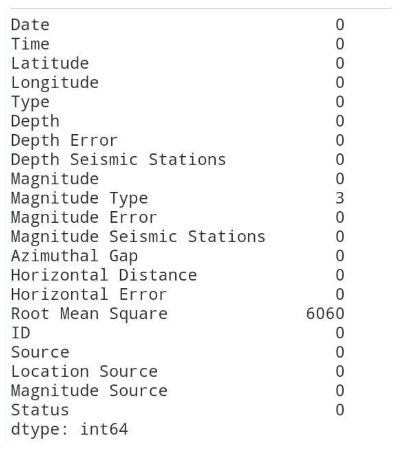
Code:

```
df['Horizontal Distance'].fillna(df['Horizontal
Distance'].min(),inplace=True)

df['Horizontal Error'].fillna(df['Horizontal Error'].max(),inplace=True)

df.isnull().sum()
```

Isnull:

The `isnull()` method returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False.

Output:



```
Date                            0
Time                            0
Latitude                        0
Longitude                       0
Type                            0
Depth                           0
Depth Error                     0
Depth Seismic Stations          0
Magnitude                       0
Magnitude Type                  3
Magnitude Error                 0
Magnitude Seismic Stations      0
Azimuthal Gap                   0
Horizontal Distance             0
Horizontal Error                0
Root Mean Square             6060
ID                              0
Source                          0
Location Source                 0
Magnitude Source                0
Status                          0
dtype: int64
```

Code:

df['Root Mean Square'].fillna(df['Root Mean Square'].mean(),inplace=True) df.isnull().sum()

Output:

```
Date                          0
Time                          0
Latitude                      0
Longitude                     0
Type                          0
Depth                         0
Depth Error                   0
Depth Seismic Stations        0
Magnitude                     0
Magnitude Type                3
Magnitude Error               0
Magnitude Seismic Stations    0
Azimuthal Gap                 0
Horizontal Distance           0
Horizontal Error              0
Root Mean Square              0
ID                            0
Source                        0
Location Source               0
Magnitude Source              0
Status                        0
dtype: int64
```

Code:

df.duplicated()

Duplicate:

The duplicated() method returns a Series with True and False values that describe which rows in the DataFrame are duplicated and not. Use the subset parameter to specify which columns to include when looking for duplicates. By default all columns are included.

Output:

```
0        False
1        False
2        False
3        False
4        False
         ...
23407    False
23408    False
23409    False
23410    False
23411    False
Length: 23412, dtype: bool
```

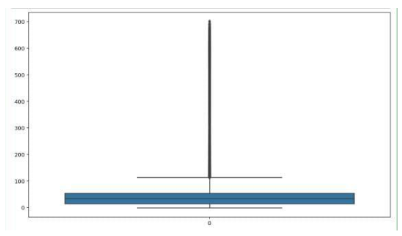
Code:

sns.boxplot(df['Depth'])
plt.show()

Boxplot:

A **Box Plot** is also known as **Whisker plot** is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

Output:



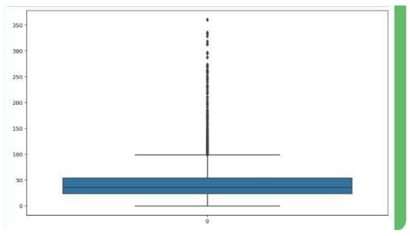
Code:

```
Q1 = df['Depth'].quantile(0.25)
Q3 = df['Depth'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = (df['Depth'] < lower_bound) | (df['Depth'] > upper_bound)
outlier_data = df[outliers]

print("Outliers:")
print(outlier_data)
```

Outliers:

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors. The analysis for outlier detection is referred to as outlier mining. There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the panda’s data frame.

Output:



Drop:

The drop() method removes the specified row or column. By specifying the column axis (axis='columns'), the drop() method removes the specified column. By specifying the row axis (axis='index'), the drop() method removes the specified row.

Code:

```
column_to_drop = 'Magnitude Type'
df = df.drop(columns=[column_to_drop])

df.isnull().sum()
```

Output:

```
Date                          0
Time                          0
Latitude                      0
Longitude                     0
Type                          0
Depth                         0
Depth Error                   0
Depth Seismic Stations        0
Magnitude                     0
Magnitude Error               0
Magnitude Seismic Stations    0
Azimuthal Gap                 0
Horizontal Distance           0
Horizontal Error              0
Root Mean Square              0
ID                            0
Source                        0
Location Source               0
Magnitude Source              0
Status                        0
dtype: int64
```

Explanation:

- **Step 1:** Remove irrelevant data
- **Step 2:** Deduplicate your data
- **Step 3:** Fix structural errors
- **Step 4:** Deal with missing data

- **Step 5:** Filter out data outliers
- **Step 6:** Validate your data

Conclusion:

After this process the cleaned data set is obtained.