**PREDICTING HOUSE PRICE USING MACHINE LEARNING**



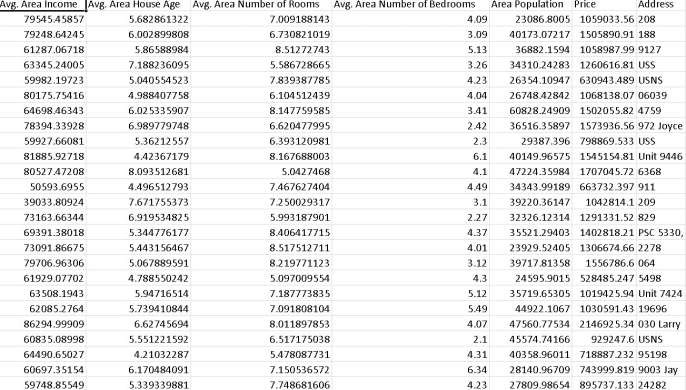
TEAM MEMBER

511521106xx : Earth quake pretegtor using python Phase 2 Submission Document

Project: House Price Prediction

# Introduction:

1. Data Collection: Begin by gathering seismic data from various sources, such as seismographic stations or online databases.
2. Data Preprocessing in Python: Utilize Python’s data processing libraries, like NumPy and Pandas, to clean and organize the seismic data, ensuring it’s ready for analysis.
3. Machine Learning Implementation: Apply machine learning techniques using Python’s scikit-learn or TensorFlow to create a predictive model based on patterns observed in the preprocessed seismic data



# Content for Project Phase 2 :

Prepare seismic data, utilize Python with scikit-learn or TensorFlow to build a regression model predicting earthquake magnitudes, and then train and evaluate the model for enhanced seismic risk assessment.

**Data Source**

A good data source for house price prediction using machine learning should be

Accurate, Complete, Covering the geographic area of interest, Accessible.

Dataset Link: (https://www.kaggle.com/code/gpreda/lanl-earthquake-eda-and-prediction?kernelSessionId=12610502)

*Output*

*# This Python 3 environment comes with many helpful analytics libraries installed*

*# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python*

*# For example, here's several helpful packages to load*

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

*# Input data files are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory*

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

*# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"*

*# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session*

/kaggle/input/earthquake-dataset/earthquake\_data.csv

/kaggle/input/earthquake-dataset/earthquake\_1995-2023.csv

## Earthquakes with Python

## 

## September 2023

## Objectives

## To discover the areas of the world mostly affected by earthquakes

## To identify patterns in the timeline of earthquakes

## To determine if the year 2022 was significantly different to the previous years.

## Areas covered.

## 

## Areas covered[¶](https://www.kaggle.com/code/karuppusamyn/fork-of-earthquakes-with-python/notebook?scriptVersionId=145977446#Areas-covered)

* **Structuring & Cleaning data**
* **Plotting**
* **Scatter maps**
* **Hypothesis Testing**

## Problems

## Duplicate and null report

## Datetime transformations

## Data Visualisation

## Hypothesis Testing

## Input

import seaborn as sns

import matplotlib.pyplot as plt

*# finding the number of earthquakes per country, to determine the most seismogenic countries:*

countries = df['country'].value\_counts()

print(countries)

country

Indonesia 110

Papua New Guinea 56

Chile 34

Vanuatu 27

Solomon Islands 22

Japan 21

Mexico 20

Peru 20

Philippines 17

United States of America 17

Russia 15

People's Republic of China 12

Fiji 9

New Zealand 9

Afghanistan 6

Taiwan 6

Ecuador 6

Myanmar 5

United Kingdom of Great Britain and Northern Ireland (the) 5

Iran 5

India 5

Greece 5

Pakistan 4

Colombia 4

Panama 4

Nepal 4

Turkey 4

Brazil 3

Bolivia 3

Costa Rica 3

Antarctica 3

Argentina 2

Haiti 2

Kyrgyzstan 1

Martinique 1

Mozambique 1

Algeria 1

Tonga 1

Canada 1

Tanzania 1

South Georgia and the South Sandwich Islands 1

Nicaragua 1

Tajikistan 1

Italy 1

Botswana 1

Guatemala 1

Venezuela 1

Mongolia 1

El Salvador 1

Name: count, dtype: int64

top\_10\_countries = (df['country'].value\_counts()).iloc[:10]

print(top\_10\_countries)

## Duplicate and null report

## Input

df.duplicated()

## output

0 False

1 False

2 False

3 False

4 False

...

777 False

778 False

779 False

780 False

781 False

Length: 782, dtype: bool

## Input

df.duplicated().sum()

## output

## Good. No duplicates!

## Null fields in 'alert' --> will replace null with 'Unknown' 'location' 'continent' 'country' --> will not use these 3 fields directly

## Datetime transformations

## Input

df['date\_time'] = pd.to\_datetime(df['date\_time'], dayfirst=True)

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 782 entries, 0 to 781

Data columns (total 19 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 title 782 non-null object

1 magnitude 782 non-null float64

2 date\_time 782 non-null datetime64[ns]

3 cdi 782 non-null int64

4 mmi 782 non-null int64

5 alert 782 non-null object

6 tsunami 782 non-null int64

7 sig 782 non-null int64

8 net 782 non-null object

9 nst 782 non-null int64

10 dmin 782 non-null float64

11 gap 782 non-null float64

12 magType 782 non-null object

13 depth 782 non-null float64

14 latitude 782 non-null float64

15 longitude 782 non-null float64

16 location 777 non-null object

17 continent 206 non-null object

18 country 484 non-null object

dtypes: datetime64[ns](1), float64(6), int64(5), object(7)

memory usage: 116.2+ KB

## Output

## 

## Data Visualisation

## Input

import seaborn as sns

import matplotlib.pyplot as plt

*# finding the number of earthquakes per country, to determine the most seismogenic countries:*

countries = df['country'].value\_counts()

print(countries)

country

Indonesia 110

Papua New Guinea 56

Chile 34

Vanuatu 27

Solomon Islands 22

Japan 21

Mexico 20

Peru 20

Philippines 17

United States of America 17

Russia 15

People's Republic of China 12

Fiji 9

New Zealand 9

Afghanistan 6

Taiwan 6

Ecuador 6

Myanmar 5

United Kingdom of Great Britain and Northern Ireland (the) 5

Iran 5

India 5

Greece 5

Pakistan 4

Colombia 4

Panama 4

Nepal 4

Turkey 4

Brazil 3

Bolivia 3

Costa Rica 3

Antarctica 3

Argentina 2

Haiti 2

Kyrgyzstan 1

Martinique 1

Mozambique 1

Algeria 1

Tonga 1

Canada 1

Tanzania 1

South Georgia and the South Sandwich Islands 1

Nicaragua 1

Tajikistan 1

Italy 1

Botswana 1

Guatemala 1

Venezuela 1

Mongolia 1

El Salvador 1

Name: count, dtype: int64

top\_10\_countries = (df['country'].value\_counts()).iloc[:10]

print(top\_10\_countries)

country

Indonesia 110

Papua New Guinea 56

Chile 34

Vanuatu 27

Solomon Islands 22

Japan 21

Mexico 20

Peru 20

Philippines 17

United States of America 17

Name: count, dtype: int64

top\_10\_countries.plot(kind='bar')

for index, value **in** enumerate(top\_10\_countries):

plt.text(index, value, str(value), ha='center', va='bottom')

plt.title('Countries with the most Earthquakes since 2001')

plt.xticks(rotation=45, horizontalalignment='right')

plt.show()

## 

## Input

continents = (df['continent'].value\_counts())

print(continents)

continent

Asia 100

South America 55

North America 34

Europe 10

Oceania 4

Africa 3

Name: count, dtype: int64

In [25]:

continents.plot()

for index, value **in** enumerate(continents):

plt.text(index, value, str(value), ha='center', va='bottom')

plt.title('Earthquake count per Continent since 2001')

plt.xticks(rotation=45, horizontalalignment='right')

Out[25]:

(array([-1., 0., 1., 2., 3., 4., 5., 6.]),

[Text(-1.0, 0, 'Africa'),

Text(0.0, 0, 'Asia'),

Text(1.0, 0, 'South America'),

Text(2.0, 0, 'North America'),

Text(3.0, 0, 'Europe'),

Text(4.0, 0, 'Oceania'),

Text(5.0, 0, 'Africa'),

Text(6.0, 0, '')])

## 

Data on Continents has a large number of missing values (74%)

However, all earthquake information is attached to a latitude and longitude, corresponding to its geographical location.

Therefore, to further evaluate this data with additional visual evidence, we will introduce the function express from plotly to create an interactive map.

In [26]:

*# plotting number of earthquakes on world map, using latitude and longitude, magnitude and depth*

import plotly.express as px

fig = px.scatter\_geo(

df,

lat='latitude',

lon='longitude',

color='magnitude',

size='depth',

hover\_name='location',

projection='natural earth',

title='Earthquake Occurrences Worldwide'

)

fig.show()

6.577.588.59magnitudeEarthquake Occurrences Worldwide

## 9

Earthquake hotspots:

* Western Pacific Faultline
* Andes faultline in Latin America, extending into the Caribbean plate Very few major events since 2001 in Africa and Australia

Events in Europe only in the Eastern Mediterranean

In [27]:

*# finding the years with the largest average magnitude*

avg\_magnitude\_per\_year = df.groupby('year')['magnitude'].mean().reset\_index().sort\_values(by='magnitude', ascending=False)

avg\_magnitude\_per\_year

Out[27]:

|  | year | magnitude |
| --- | --- | --- |
| 8 | 2009 | 7.161538 |
| 11 | 2012 | 7.070968 |
| 6 | 2007 | 7.054054 |
| 20 | 2021 | 7.052381 |
| 0 | 2001 | 7.028571 |
| 9 | 2010 | 7.004878 |
| 10 | 2011 | 6.988235 |
| 3 | 2004 | 6.959375 |
| 17 | 2018 | 6.953488 |
| 15 | 2016 | 6.944186 |
| 4 | 2005 | 6.942857 |
| 5 | 2006 | 6.942308 |
| 19 | 2020 | 6.911111 |
| 7 | 2008 | 6.900000 |
| 1 | 2002 | 6.900000 |
| 14 | 2015 | 6.898113 |
| 12 | 2013 | 6.890566 |
| 2 | 2003 | 6.889032 |
| 18 | 2019 | 6.860606 |
| 13 | 2014 | 6.843750 |
| 21 | 2022 | 6.812500 |
| 16 | 2017 | 6.811111 |

## 

df\_2001\_2021['year'].value\_counts()

Out[33]:

year

2015 53

2013 53

2014 48

2018 43

2016 43

2021 42

2010 41

2007 37

2017 36

2011 34

2019 33

2004 32

2012 31

2003 31

2005 28

2001 28

2020 27

2009 26

2006 26

2008 25

2002 25

Name: count, dtype: int64

In [34]:

df\_2001\_2021['year'].value\_counts().mean()

Out[34]:

35.333333333333336

In [35]:

df\_2022.count()

Out[35]:

title 40

magnitude 40

date\_time 40

year 40

month 40

cdi 40

mmi 40

alert 40

tsunami 40

sig 40

net 40

nst 40

dmin 40

gap 40

magType 40

depth 40

latitude 40

longitude 40

location 37

continent 17

country 18

dtype: int64

There were 40 earthquakes over 6.5 in 2022 > mean of 35.333333333333336 Average magnitude in 2022 was 6.812500 < mean of 2001 - 2021: 6.948

As standard deviation for both samples is quite low (0.28 for 2022, 0.45 for 2001-2021), my prediction is that the difference will be more than 2σ, therefore statistically significant.

In [36]:

from scipy import stats

*# setting significance level*

significance\_level = 0.05

significance\_level

Out[36]:

0.05

In [37]:

*# computing p-value*

stats.ttest\_ind(a=df\_2022['magnitude'], b=df\_2001\_2021['magnitude'], equal\_var=False)

Out[37]:

TtestResult(statistic=-2.824790324997883, pvalue=0.006768655020004039, df=50.24143813494711)

A t-statistic of -2.82 suggests that there is a significant difference between the means of the two groups,  
and the negative sign indicates that the second group's mean is larger than the first group's mean: 6.812 < 6.948

p = 0.6% < 5%. As our p-value is far less than our chosen significance level, it indicates statistical significance.

Therefore, we reject our Hypothesis 0.

There is a significant difference in magnitude between the year 2022 and the previous years (2001 - 2021),  
which isn't due to sampling variability or chance.

## Conclusion and Future Work (Phase 2):

## Project Conclusion:

* The data analysis together with the Scatterplot identified **The Asian Pacific** and **Latin America** as the most seismic areas in the world.
* The three countries with the most earthquakes were:

1. Indonesia (110)
2. Papua New Guinea (56)
3. Chile (34)

* **2009** was the year with the most earthquakes over 6.5 degrees of the Richter scale
* Finally, our **Hypothesis Testing** determined that indeed **2022** was significantly different from the previous years.
* In the Phase 2 conclusion, we will summarize the key findings and insights from the advanced regression techniques. We will reiterate the impact of these techniques on improving the accuracy and robustness of house price predictions.
* Future Work: We will discuss potential avenues for future work, such as incorporating additional data sources (e.g., real-time economic indicators), exploring deep learning models for prediction, or expanding the project into a web application with more features and interactivity.