Project : EARTHQUAKE PREDICTION MODEL USING PYTHON

PHASE-1

Abstract: An earthquake is a type of natural disaster that is well-known for the devastation it causes to both naturally existing and artificial structures, including buildings, bungalows, and residential areas, to name a few. Seismometers, which pick up vibrations caused by seismic waves moving through the earth's crust, are used to measure earthquakes.

The damage caused by an earthquake was categorised in this work into damage ratings, which have values ranging from one to five. The damage grade of a certain structure, which is linked to a Unique Identification String, was predicted using a previously gathered data set and a number of criteria. An analysis of current machine learning classifier techniques was used to make the forecast.

Logistic Regression, Support Vector Machine (SVM), Random Forest Classifier, and K-Nearest Neighbours were the machine learning techniques employed in this study. The best algorithm was taken into consideration after a review of a number of attributes. The method used to predict the property underwent a thorough investigation, and the data analysis that followed revealed information that could help future earthquakes' effects be lessened**.**

Keywords: Machine learning, Support Vector Machine (SVM), Random Forest Classifier, Logistic Regression, K Nearest Neighbors, and predictive analysis.

I. INTRODUCTION

A catastrophic event such as an earthquake is harmful to human interests and has negative effects on the environment. Incalculable harm to buildings and other assets has always been done by earthquakes, which have also claimed millions of lives around the world. Numerous national, international, and transnational organisations implement various disaster warning and preventive strategies to lessen the effects of such an incident. Organisation managers have a number of challenges when it comes to allocating the organisation's resources because time and quantity are constraints.To estimate the extent of damage done to buildings after an earthquake, it is possible to use machine learning. This is accomplished by categorising these buildings according to a degree of damage severity based on a number of elements, including their age, foundation, number of floors, kind of material used, and others. Then, ward-by-ward in a district, the number of families and the likely casualties are considered. This enables the proportionate distribution of relief forces by ward and their prioritising according to the severity of the damage.

Such models can contribute to the fastest possible lifesaving and prove to be a successful and affordable option.[1-3] It can be further enhanced by including the distribution of goods like food, clothing, medical care, and money in accordance with the number of fatalities among people and the degree of structural damage.

II. BACKGROUND

From its foundations in databases, statistics, applied science, theory, and algorithms, the discipline of machine learning has developed into a core set of approaches that are applied to a variety of issues. Over the past 20 years, significant advancements have been made in the scientific and technical fields of computational modelling and data gathering.

Additional data repositories have been produced as a result of a combination of sophisticated algorithms, exponentially rising processing power, and precise sensing and measurement tools. Networks with cutting-edge technologies have made it possible to send enormous amounts of data around the globe. This leads to an extreme need for tools and technologies in order to analyse scientific data sets effectively with the aim of deciphering the underlying physical events. Machine Learning applications in geology and geophysics have achieved significant success within the areas as weather prediction, mineral prospecting, ecology, modelling etc and eventually predicting the earthquakes from satellite maps.

An interesting aspect of the numerous of these applications is that they combine both spatial and temporal aspects within the info and within the phenomena that’s being mined. Investigations on earthquake predictions support the concept that each one amongst the regional factors is filtered out and general information about the earthquake precursory patterns is extracted. Feature extraction involves a pre-selection process of varied statistical properties of data and generation of a group of seismic parameters, which correspond to linearly independent coordinators within the feature space. The seismic within the sort of statistic are often analysed by using various pattern recognition techniques. Statistical or pattern recognition methodology usually performs this extraction process. This gives an idea about mining the scientific data.

III. MOTIVATION

Earthquakes are one amongst the foremost destructive natural disasters. They typically occur without notice and do not allow much time for people to react. Therefore, earthquakes can cause serious injuries and loss of life and destroy tremendous buildings and infrastructures, resulting in great economic loss. The prediction of earthquakes is clearly critical to the protection of our society, but it's proven to be a Challenging task to predict beforehand and yet we discover this as a motivating problem to be solved.

IV. SOFTWARE REQUIREMENT SPECIFICATION

i. Anaconda Navigator 2.3.2
ii. Jupyter Notebook 6.0.3
iii. Tkinter
iv. OpenCV
v. Pycharm 2022.2.4
vi. Language: Python 3.7
vii. Environment: Keras and Tensorflow environment
viii. OS: Windows 7 or higher

V. METHODOLOGY

Dataset Link:

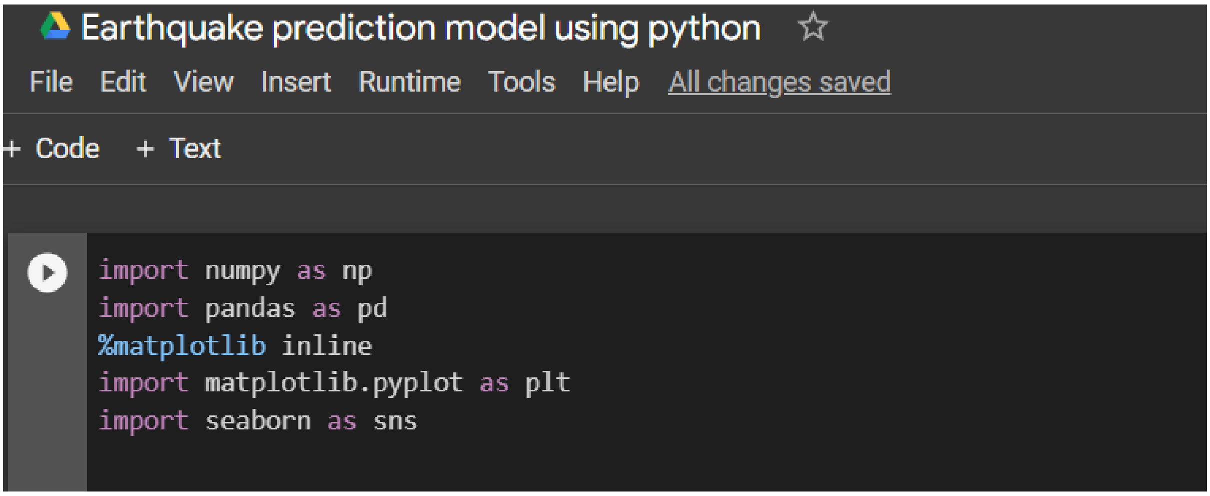
https://www.kaggle.com/datasets/usgs/earthquake-database

DESIGN THINKING

1.Data Source: Choose a suitable Kaggle dataset containing earthquake data with features like date, time, latitude, longitude, depth, and magnitude.

A. Importing Libraries :

Figure 1 shows the Python code to import libraries. We have used four libraries :-
• Python has a library called Numpy that is used for scientific computing. This library is utilised throughout the project and is imported as np.
• Pandas are used for data analysis and manipulation. An open source, BSD-licensed library called pandas offers simple data structures and tools for data analysis. It is imported as pd.
• matplotlib is a python library. The command-style utilities in pyplot enable matplotlib to behave similarly to MATLAB. It is imported as plt.
• Seaborn is a matplotlib-based Python data visualisation package for aesthetically pleasing and educational statistical visuals. It is imported as sns.



B. Importing data :

The below figure displays the Python code for importing data from the appropriate directory or file and allocating it to a DataFrame. It imports the data that is kept in CSV format.

[12]
```
from google.colab import files
upload=files.upload()
```

Choose Files database.csv.zip

- **database.csv.zip**(application/x-zip-compressed) - 604367 bytes, last modified: 9/29/2023 - 100% done

```
Saving database.csv.zip to database.csv.zip
```

```
data = pd.read_csv("database.csv.zip")
data.head()
```

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Type	...	Magnitude Seismic Stations	Azimuthal Gap	Horizontal Distance	Horizontal Error	Root Mean Square	ID	Source	L
0	01/02/1965	13:44:18	19.246	145.616	Earthquake	131.6	NaN	NaN	6.0	MW	...	NaN	NaN	NaN	NaN	NaN	ISCGEM860706	ISCGEM	I
1	01/04/1965	11:29:49	1.863	127.352	Earthquake	80.0	NaN	NaN	5.8	MW	...	NaN	NaN	NaN	NaN	NaN	ISCGEM860737	ISCGEM	I
2	01/05/1965	18:05:58	-20.579	-173.972	Earthquake	20.0	NaN	NaN	6.2	MW	...	NaN	NaN	NaN	NaN	NaN	ISCGEM860762	ISCGEM	I
3	01/08/1965	18:49:43	-59.076	-23.557	Earthquake	15.0	NaN	NaN	5.8	MW	...	NaN	NaN	NaN	NaN	NaN	ISCGEM860856	ISCGEM	I
4	01/09/1965	13:32:50	11.938	126.427	Earthquake	15.0	NaN	NaN	5.8	MW	...	NaN	NaN	NaN	NaN	NaN	ISCGEM860890	ISCGEM	I

5 rows × 21 columns

[14]
```
data.columns
```

```
Index(['Date', 'Time', 'Latitude', 'Longitude', 'Type', 'Depth', 'Depth Error',
       'Depth Seismic Stations', 'Magnitude', 'Magnitude Type',
```

C. Checking for NaN :

Checking for NaN is a critical step in the pre-processing of data. We were only able to identify a few NaNs in this test. The Python code to check for NaN is displayed below .

[17]
```
timeStamp = pd.Series(timestamp)
data['Timestamp'] = timeStamp.values
```

```
data.isnull()
```

	Date	Time	Latitude	Longitude	Depth	Magnitude	Timestamp
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
23407	False	False	False	False	False	False	False
23408	False	False	False	False	False	False	False
23409	False	False	False	False	False	False	False
23410	False	False	False	False	False	False	False
23411	False	False	False	False	False	False	False

23412 rows × 7 columns

D. Manipulating NaN values :

It is essential to remove the NaN values. This can be done by
• Removing the entire column containing many NaN values
• Forward fillna method
• Backward fillna method
• Mean method

Here, the data is random; we need to scale according to inputs to the model. In this, 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.

In [5]:

```
import datetime
import time

timestamp = []
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:
        # print('ValueError')
        timestamp.append('ValueError')
```

In [6]:

```
timeStamp = pd.Series(timestamp)
data['Timestamp'] = timeStamp.values
```

2.Feature Exploration: Analyse and understand the distribution, correlations, and characteristics of the key features.

Distribution :

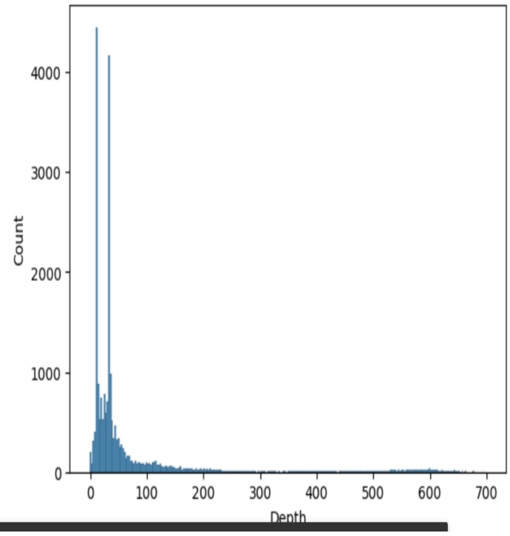
+ Code + Text

[77]

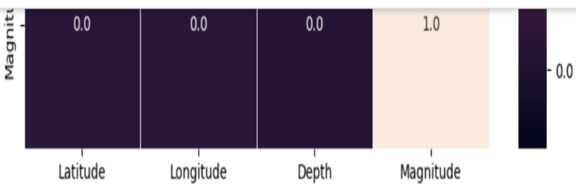
2	01/05/1965	18:05:58	-20.579	-173.972	20.0	6.2	-157355642.0
3	01/08/1965	18:49:43	-59.076	-23.557	15.0	5.8	-157093817.0
4	01/09/1965	13:32:50	11.938	126.427	15.0	5.8	-157026430.0

```
sns.histplot(data['Depth'])
```

```
<Axes: xlabel='Depth', ylabel='Count'>
```



[65]



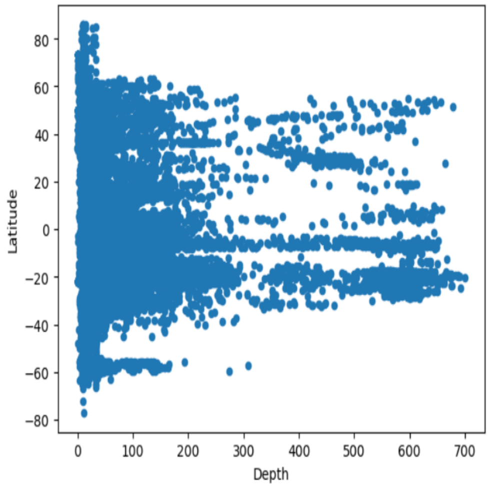

```
[77] data.head()
```

	Date	Time	Latitude	Longitude	Depth	Magnitude	Timestamp
0	01/02/1965	13:44:18	19.246	145.616	131.6	6.0	-157630542.0
1	01/04/1965	11:29:49	1.863	127.352	80.0	5.8	-157465811.0
2	01/05/1965	18:05:58	-20.579	-173.972	20.0	6.2	-157355642.0
3	01/08/1965	18:49:43	-59.076	-23.557	15.0	5.8	-157093817.0
4	01/09/1965	13:32:50	11.938	126.427	15.0	5.8	-157026430.0

Correlation :

```
import matplotlib.pyplot as plt

data.plot(x ='Depth', y='Latitude', kind='scatter')
plt.show()
```



Characteristics:

Earthquake prediction typically requires the information regarding the magnitude, location, and time of occurrence.
These can be categorized into short-term and long-term processes.

3.Visualization: Create a world map visualization to display earthquake frequency distribution.

Here, all the earthquakes from the database are visualised on to the world map which shows clear representation of the locations where frequency of the earthquake will be more.

E. Plotting a Basemap :

Basemaps serve as a reference map on which you overlay data from layers and visualise geographic information. An individual basemap can be made of multiple feature, raster, or web layers. Basemaps are the foundation for your maps and provide context for your work. A sample 3D basemap over Athens, Greece, is shown.

```
[39] Installing collected packages: basemap-data, basemap
Successfully installed basemap-1.3.8 basemap-data-1.3.2
WARNING: The following packages were previously imported in this runtime:
  [mpl_toolkits]
You must restart the runtime in order to use newly installed versions.
```

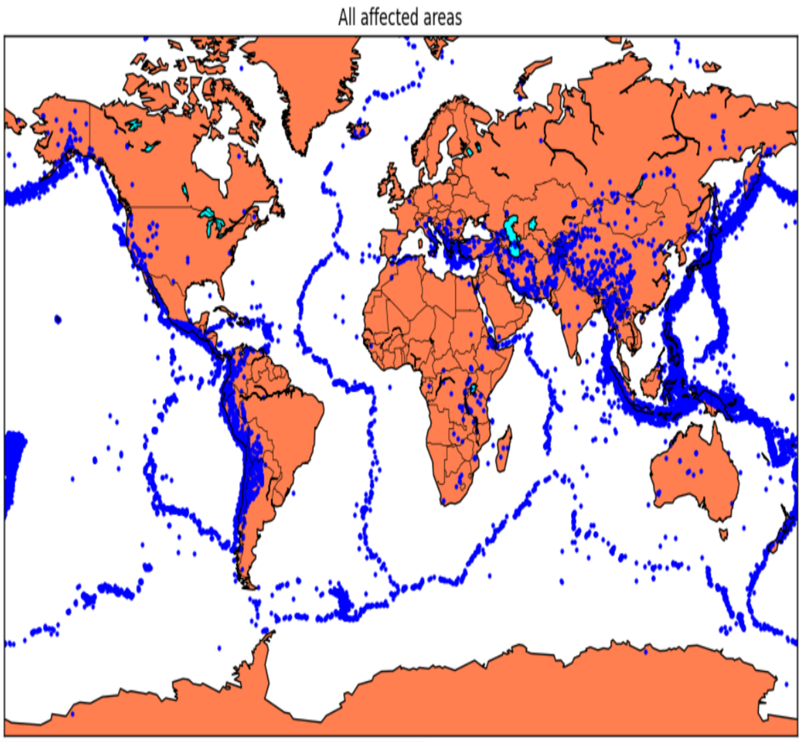
RESTART RUNTIME

```
[42] from mpl_toolkits.basemap import Basemap

m = Basemap(projection='mill',llcrnrlat=-80,urcrnrlat=80, llcrnrlon=-180,urcrnrlon=180,lat_ts=20,resolution='c')

longitudes = data["Longitude"].tolist()
latitudes = data["Latitude"].tolist()
#m = Basemap(width=12000000,height=9000000,projection='lcc',
            #resolution=None,lat_1=80.,lat_2=55,lat_0=80,lon_0=-107.)
x,y = m(longitudes,latitudes)
```

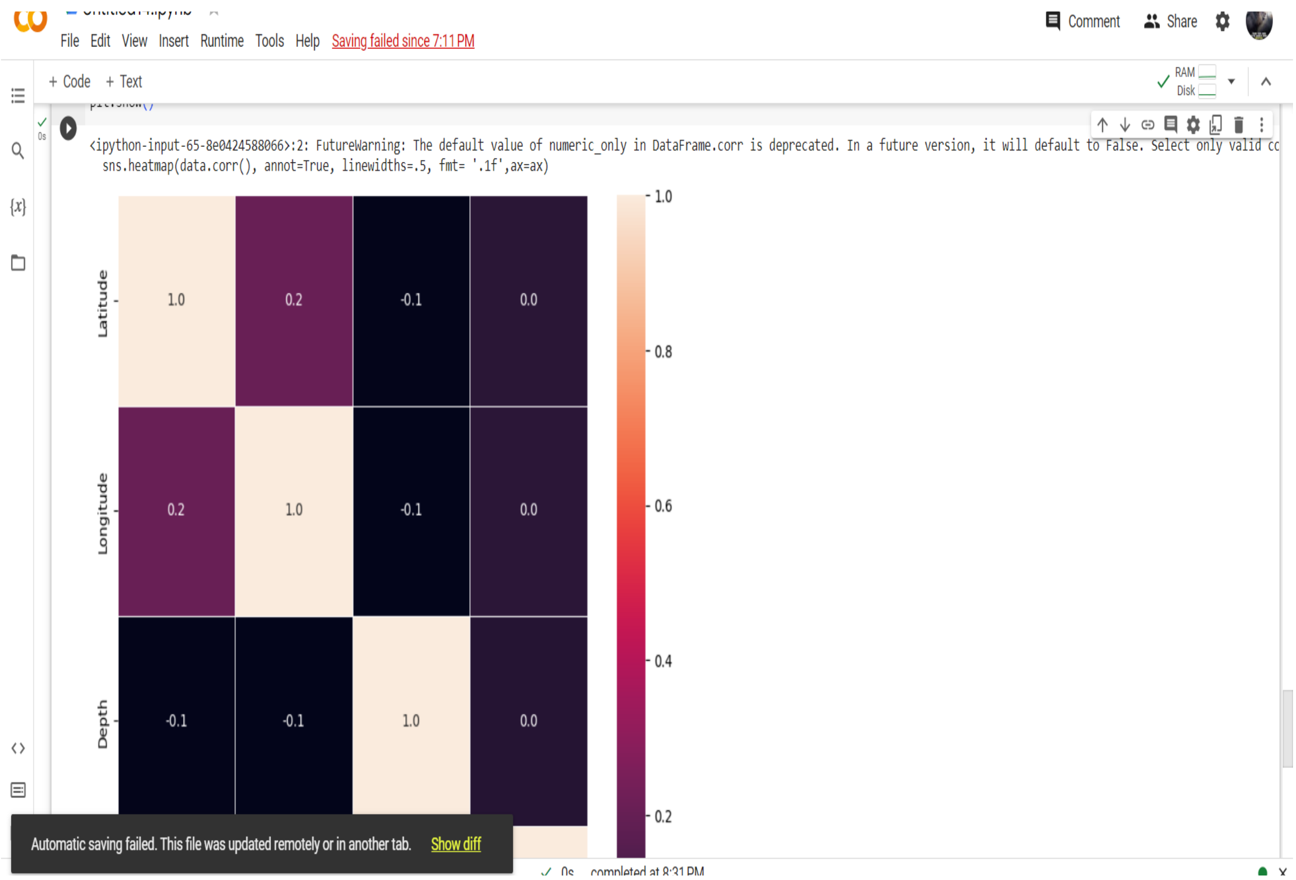
```
fig = plt.figure(figsize=(12,10))
plt.title("All affected areas")
m.plot(x, y, "o", markersize = 2, color = 'blue')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
m.drawmapboundary()
m.drawcountries()
plt.show()
```



F.Plotting the Heatmap :

A heatmap is used to assess the correlation between the fields of the collecting data.When developing various AI prediction models,the magnitude of the values along with the sign is crucial.

Heatmaps are used in various forms of analytics but are most commonly used to show user behavior on specific web pages or webpage templates. Heatmaps can be used to show where users have clicked on a page, how far they have scrolled down a page, or used to display the results of eye-tracking tests.



G. Train/Test split :

Creating train and test sets from the data is our next step towards developing a Machine Learning model. The Python code to divide the data set into train and test data is shown below

In [10]:

```
X = final_data[['Timestamp', 'Latitude', 'Longitude']]
y = final_data[['Magnitude', 'Depth']]
```

In [11]:

```
from sklearn.cross_validation import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, X_test.shape)
```

```
(18727, 3) (4682, 3) (18727, 2) (4682, 3)

/opt/conda/lib/python3.6/site-packages/sklearn/cross_validation.py:41: DeprecationWarning:
This module was deprecated in version 0.18 in favor of the model_selection module into whi
ch all the refactored classes and functions are moved. Also note that the interface of the
new CV iterators are different from that of this module. This module will be removed in 0.
20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

5.Model Development :

Build a neural network model for earthquake magnitude prediction.
In the above case it was more kind of linear regressor where the predicted values are not as expected. So, Now, we build the neural network to fit the data for training set. Neural Network consists of three Dense layer with each 16, 16, 2 nodes and relu, relu and softmax as activation function.

In [16]:

```
from keras.models import Sequential
from keras.layers import Dense

def create_model(neurons, activation, optimizer, loss):
    model = Sequential()
    model.add(Dense(neurons, activation=activation, input_shape=(3,)))
    model.add(Dense(neurons, activation=activation))
    model.add(Dense(2, activation='softmax'))

    model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

    return model
```

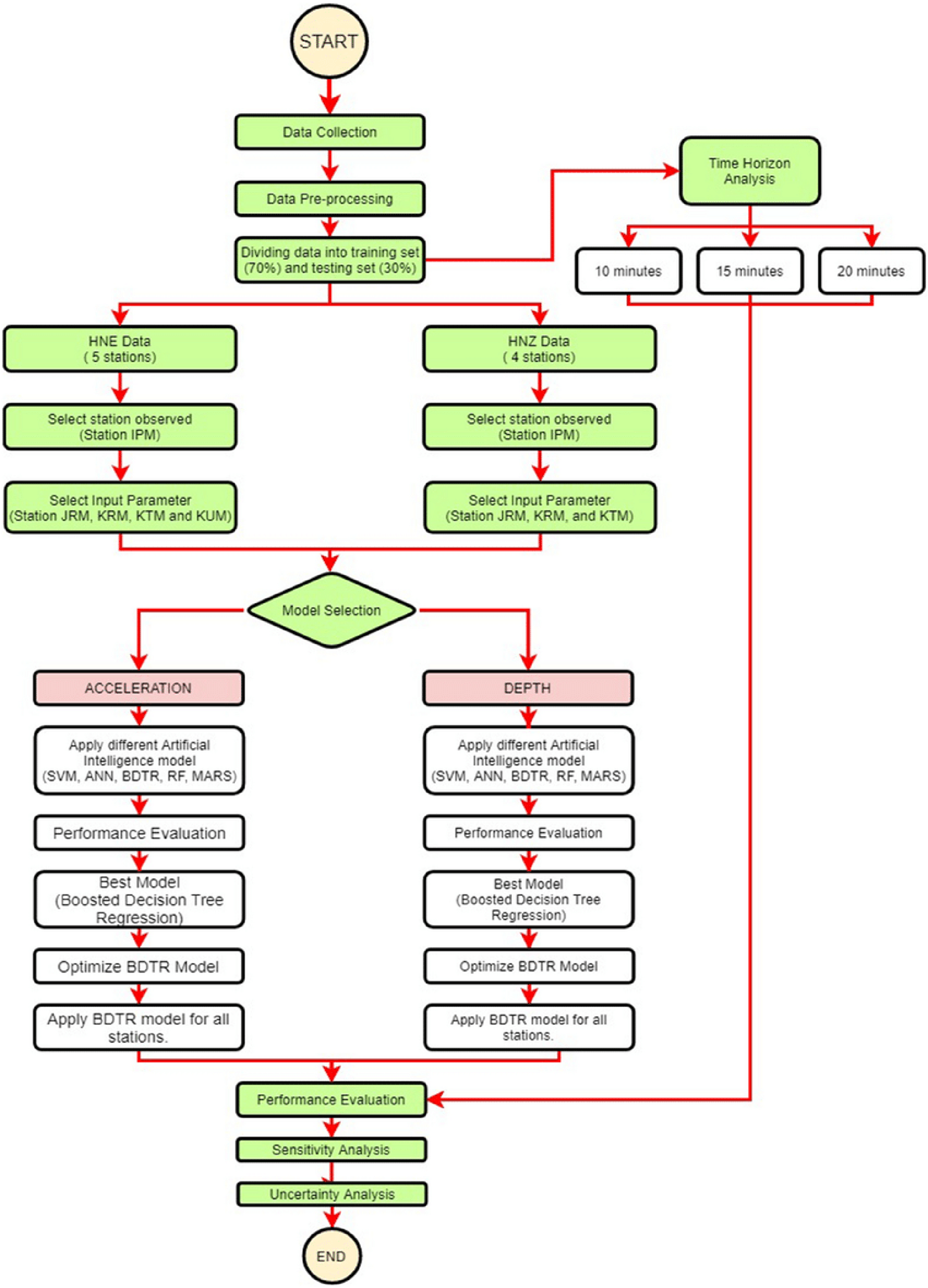
SOLUTION ARCHITECTURE

- Solution architecture is a structured approach to designing complex systems or projects,
- Outlining the components , relationships , and processes to achieve specific goals or solve problems efficiently .
- It provides a high - level blueprint for project development and implementation .
- Gather and preprocess data from various sources , ensuring data quality .
- Train AI models for earthquake prediction and evaluate their performance.
- Deploy models in a secure , scalable environment with a user - friendly interface .
- Continuously monitor and update the system while complying with regulations .
- Collaborate with professionals and gather user feedback for improvements .

AI based earthquake prediction system solution architecture:

Machine learning algorithms can analyze patterns in seismic activity, geological data, and other factors to identify the probability of an earthquake occurring . This technology could potentially provide timely and accurate predictions, allowing authorities to prepare and mitigate the impact of earthquakes.
Seismometers capture seismic data, which may be used to spot changes to the earth's surface, like seismic waves brought on by earthquakes.
In this architecture ,

- The data is collected , processed and divided into train and test sets.
- Then they are separated into HNE and HNZ data , and they are processed and analysed by the visualization tools .


START
Data Collection
Data Pre-processing
Dividing data into training set (70%) and testing set (30%)
Time Horizon Analysis
10 minutes
15 minutes
20 minutes
HNE Data (5 stations)
HNZ Data (4 stations)
Select station observed (Station IPM)
Select station observed (Station IPM)
Select Input Parameter (Station JRM, KRM, KTM and KUM)
Select Input Parameter (Station JRM, KRM, and KTM)
Model Selection
ACCELERATION
DEPTH
Apply different Artificial Intelligence model (SVM, ANN, BDTR, RF, MARS)
Apply different Artificial Intelligence model (SVM, ANN, BDTR, RF, MARS)
Performance Evaluation
Performance Evaluation
Best Model (Boosted Decision Tree Regression)
Best Model (Boosted Decision Tree Regression)
Optimize BDTR Model
Optimize BDTR Model
Apply BDTR model for all stations.
Apply BDTR model for all stations.
Performance Evaluation
Sensitivity Analysis
Uncertainty Analysis
END

Conclusion:

In this Machine Learning Project, we explore an earthquake prediction system. In this model , we used some of design thinking process such as data source,exploration of dataset features and also visualise the earthquake frequency using visualisation tools like numpy , pandas , seaborn , matplotlib ,etc.,

References :

1. https://github.com/akash-r34/Earthquake-prediction-using-Machine-learning-models
2. https://www.researchgate.net/publication/313858017_Machine_Learning_Predicts_Laboratory_Earthquakes
3. http://earthquake.usgs.gov/data/comcat/data-eventterms.php#time
4. DriveData, Richter's Predictor: Modeling Earthquake Damage (2021).
5. Sameer, Earthquake History (1965–2016): Data Visualization and Model Development.
6. World Health Organization, Earthquakes, Health topics.