# EARTHQUAKE PREDICTION MODEL USING PYTHON

PHASE4

PROJECT

# PREPARED BY

S.SNEKHA,
510521205045,
BHARATHIDASAN ENGINEERING COLLEGE,
PHASE4 PROJECT SUBMISSION.

# INTRODUCTION

Earthquake prediction using advanced techniques is a vital area of research aimed at mitigating the devastating impact of seismic events on human lives and infrastructure. Traditional earthquake prediction methods have often been limited in their accuracy and reliability. However, recent advancements in seismology, geodesy, and data analytics have opened up new avenues for more precise earthquake forecasting. In this discussion, we will explore the cutting-edge technologies and methodologies that are revolutionizing earthquake prediction, offering hope for better preparedness and risk reduction in earthquake-prone regions.

# DATASET

| Date | Time | Latitude | Longitude | Type | Depth | Depth Error | Depth Seis | Magnitude | Magnitude | ID | Source | Location So | Magnitude | Statu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/2/1965 | 13:44:18 | 19.246 | 145.616 | Earthquake | 131.6 | | | 6 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/4/1965 | 11:29:49 | 1.863 | 127.352 | Earthquake | 80 | | | 5.8 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/5/1965 | 18:05:58 | -20.579 | -173.972 | Earthquake | 20 | | | 6.2 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/8/1965 | 18:49:43 | -59.076 | -23.557 | Earthquake | 15 | | | 5.8 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/9/1965 | 13:32:50 | 11.938 | 126.427 | Earthquake | 15 | | | 5.8 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/10/1965 | 13:36:32 | -13.405 | 166.629 | Earthquake | 35 | | | 6.7 | MW | ISCGEM860 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/12/1965 | 13:32:25 | 27.357 | 87.867 | Earthquake | 20 | | | 5.9 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/15/1965 | 23:17:42 | -13.309 | 166.212 | Earthquake | 35 | | | 6 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/16/1965 | 11:32:37 | -56.452 | -27.043 | Earthquake | 95 | | | 6 | MW | ISCGEMSUP | ISCGEMSUP | ISCGEM | ISCGEM | Auto |
| 1/17/1965 | 10:43:17 | -24.563 | 178.487 | Earthquake | 565 | | | 5.8 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/17/1965 | 20:57:41 | -6.807 | 108.988 | Earthquake | 227.9 | | | 5.9 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/24/1965 | 0:11:17 | -2.608 | 125.952 | Earthquake | 20 | | | 8.2 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 1/29/1965 | 9:35:30 | 54.636 | 161.703 | Earthquake | 55 | | | 5.5 | MW | ISCGEM861 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/1/1965 | 5:27:06 | -18.697 | -177.864 | Earthquake | 482.9 | | | 5.6 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/2/1965 | 15:56:51 | 37.523 | 73.251 | Earthquake | 15 | | | 6 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 3:25:00 | -51.84 | 139.741 | Earthquake | 10 | | | 6.1 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 5:01:22 | 51.251 | 178.715 | Earthquake | 30.3 | | | 8.7 | MW | OFFICIAL19 | OFFICIAL | ISCGEM | OFFICIAL | Auto |
| 2/4/1965 | 6:04:59 | 51.639 | 175.055 | Earthquake | 30 | | | 6 | MW | ISCGEMSUP | ISCGEMSUP | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 6:37:06 | 52.528 | 172.007 | Earthquake | 25 | | | 5.7 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 6:39:32 | 51.626 | 175.746 | Earthquake | 25 | | | 5.8 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 7:11:23 | 51.037 | 177.848 | Earthquake | 25 | | | 5.9 | MW | ISCGEMSUP | ISCGEMSUP | ISCGEM | ISCGEM | Auto |
| 2/4/1965 | 7:14:59 | 51.73 | 173.975 | Earthquake | 20 | | | 5.9 | MW | ISCGEM859 | ISCGEM | ISCGEM | ISCGEM | Auto |

# OVERVIEW

FEATURE ENGINEERING

MODEL TRAINING

EVALUATION

## FEATURE ENGINEERING:

Feature engineering in earthquake prediction involves selecting, creating, or transforming relevant attributes or characteristics (features) from raw data that can improve the accuracy of earthquake prediction models. These features might include seismic data, geological information, historical earthquake records, and various other parameters. The goal is to extract meaningful patterns and relationships that enable machine learning models to better predict when and where earthquakes might occur.

## MODEL TRAINING:

Model training in earthquake prediction refers to the process of using historical earthquake data, as well as relevant features and attributes, to teach a machine learning or statistical model how to make predictions about future seismic events. During training, the model learns the patterns and relationships within the data, adjusting its internal parameters to minimize prediction errors. Once the model is trained, it can be used to make forecasts and predictions about the likelihood, location, and magnitude of future earthquakes based on new, unseen data. This training process is a crucial step in developing accurate earthquake prediction models.

## EVALUATION:

Evaluation in earthquake prediction involves assessing the performance and accuracy of a prediction model to determine how well it can forecast seismic events. This typically includes measuring the model's ability to correctly predict earthquakes in terms of their timing, location, and magnitude. Common evaluation metrics in earthquake prediction might include precision, recall, F1 score, and mean squared error. The goal of evaluation is to gauge the model's reliability and effectiveness, helping researchers and authorities understand its strengths and limitations in providing early warnings or forecasts of earthquakes.

# STEPS FOR FEATURE ENGINEERING

1. Data Collection
2. Data Preprocessing
3. Feature Selection
4. Feature Extraction
5. Feature Engineering
6. Dimensionality Reduction
7. Data Splitting
8. Model Building andTraining
9. Model Evaluation
10. Iterative Refinement
11. Validation and Cross-Validation
12. Hyperparameter Tuning
13. Final Model Selection

# STEPS FOR MODEL TRAINING

1. Data Preprocessing
2. Select a Model
3. Hyperparameter Selection
4. Feature Scaling and Normalization
5. Model Training
6. Cross-Validation
7. Performance Metrics
8. Model Evaluation
9. Hyperparameter Tuning
10. Final Model Selection
11. Testing
12. Deployment
13. Monitoring and Maintenance

# STEPS FOR EVALUATION

1. Test Data Selection
2. Model Prediction
3. Performance Metrics
4. Evaluate Location and Magnitude
5. Evalute Timing
6. Visual Inspection
7. Analyze False Positives and False Negatives
8. Cross-Validation
9. Threshold Tuning
10. Compare to Baselines
11. Report Results
12. Decision-Making
13. Iterative Refinement
14. Deployment Decision

# PROGRAM

## **Importing Libraries:**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
print(os.listdir("../input"))
```

**OUTPUT:**

```
['database.csv']
```

## Read the Dataset:

```
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')
timeStamp = pd.Series(timestamp)
data['Timestamp'] = timeStamp.values
final_data = data.drop(['Date', 'Time'], axis=1)
final_data = final_data[final_data.Timestamp != 'ValueError']
final_data.head()
```

| | Latitude | Longitude | Depth | Magnitude | Timestamp |
|---|---|---|---|---|---|
| **0** | 19.246 | 145.616 | 131.6 | 6.0 | -1.57631e+08 |
| **1** | 1.863 | 127.352 | 80.0 | 5.8 | -1.57466e+08 |
| **2** | -20.579 | -173.972 | 20.0 | 6.2 | -1.57356e+08 |
| **3** | -59.076 | -23.557 | 15.0 | 5.8 | -1.57094e+08 |
| **4** | 11.938 | 126.427 | 15.0 | 5.8 | -1.57026e+08 |

## Splitting The Dataset:

```
from sklearn.model_selection import GridSearchCV
parameters = {'n_estimators':[10, 20, 50, 100, 200, 500]}
grid_obj = GridSearchCV(reg, parameters)
grid_fit = grid_obj.fit(X_train, y_train)
best_fit = grid_fit.best_estimator_
best_fit.predict(X_test)
```

**OUTPUT:**

```
array([[  5.8888 ,  43.532  ],
       [  5.8232 ,  31.71656],
       [  6.0034 ,  39.3312 ],
       ...,
       [  6.3066 ,  23.9292 ],
       [  5.9138 , 592.151  ],
       [  5.7866 ,  38.9384 ]])
```

**Neural Network Model:**

```
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
from keras.wrappers.scikit_learn import KerasClassifier
model = KerasClassifier(build_fn=create_model, verbose=0)
# neurons = [16, 64, 128, 256]
neurons = [16]
# batch_size = [10, 20, 50, 100]
batch_size = [10]
epochs = [10]
# activation = ['relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear', 'exponential']
activation = ['sigmoid', 'relu']
```

```
# optimizer = ['SGD', 'RMSprop', 'Adagrad', 'Adadelta',
'Adam', 'Adamax', 'Nadam']
optimizer = ['SGD', 'Adadelta']
loss = ['squared_hinge']

param_grid = dict(neurons=neurons,
batch_size=batch_size, epochs=epochs,
activation=activation, optimizer=optimizer, loss=loss)

grid = GridSearchCV(estimator=model,
param_grid=param_grid, n_jobs=-1)
grid_result = grid.fit(X_train, y_train)

print("Best: %f using %s" % (grid_result.best_score_,
grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

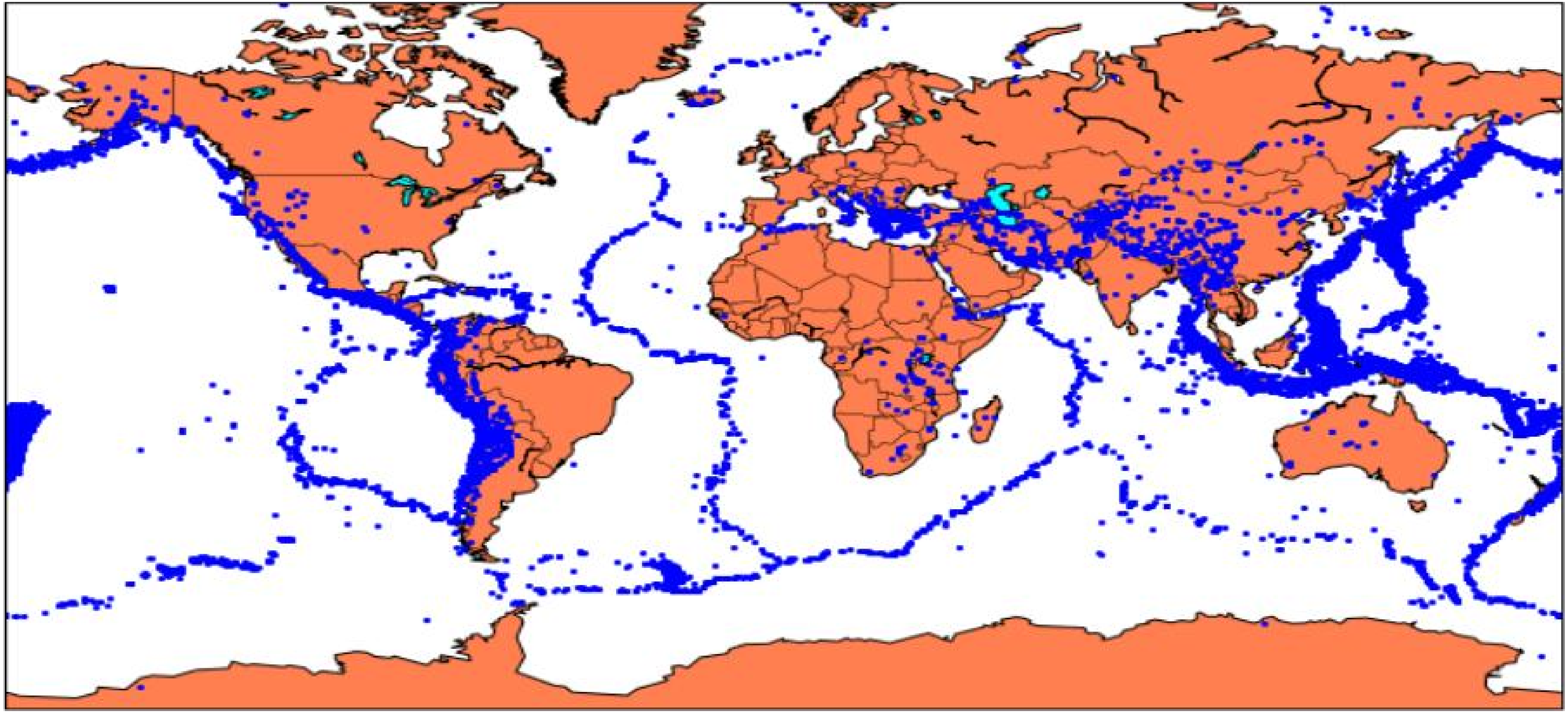
**OUTPUT**:

```
Best: 0.957655 using {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squared_
hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.333316 (0.471398) with: {'activation': 'sigmoid', 'batch_size': 10, 'epochs': 10, 'loss':
'squared_hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.000000 (0.000000) with: {'activation': 'sigmoid', 'batch_size': 10, 'epochs': 10, 'loss':
'squared_hinge', 'neurons': 16, 'optimizer': 'Adadelta'}
0.957655 (0.029957) with: {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squ
ared_hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.645111 (0.456960) with: {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squ
ared_hinge', 'neurons': 16, 'optimizer': 'Adadelta'}
```

## Visualization:

```
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()
```

All affected areas



# CONCLUSION

Understanding earthquakes and effectively responding to them remains a complex and challenging task, even with the latest technological advancements. However, leveraging the capabilities of machine learning can greatly enhance our comprehension of seismic events. By employing machine learning techniques to analyze seismic data, we can uncover valuable insights and patterns that contribute to a deeper understanding of earthquakes. These insights can subsequently inform more effective strategies for mitigating risks and responding to seismic events.

In summary, an earthquake prediction project is a critical undertaking that holds the potential to mitigate the devastating effects of earthquakes. While the pursuit of precise prediction remains challenging, the project's efforts contribute to a growing body of knowledge and technology aimed at enhancing our understanding and preparedness for these natural disasters. The project's conclusion marks not the end but a milestone in a continuous journey toward more effective earthquake prediction and risk reduction.

# Thank you so much for your watching