Earthquake prediction using python

| DATE | 25 october 2023 |
| --- | --- |
| TEAM ID | proj-212172\_TEAM\_1 |
| PROJECT NAME | Earthquake prediction using python |
| MAXIMUM MARKS |  |

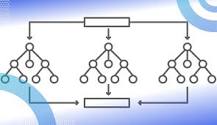
# 

# 

# 

# Random forest regressor:

An earthquake's latitude, longitude, magnitude, and depth may be predicted using the random forest algorithm. A random forest with multioutput technique is employed, with variables being each station's recorded value and geographic position.



Random forest regression is a supervised learning algorithm and bagging technique that uses an ensemble learning method for regression in machine learning. The trees in random forests run in parallel, meaning there is no interaction between these trees while building the trees.

Random Forest is a popular machine learning algorithm used for classification and regression tasks due to its high accuracy, robustness, feature importance, versatility, and scalability. Random Forest reduces overfitting by averaging multiple decision trees and is less sensitive to noise and outliers in the data.

# 

# Import:

from sklearn.ensemble import RandomForestRegressor

model=RandomForestRegressor()

# Code:

#Train the model

x=df[['Magnitude','Latitude','Longitude','Depth']]

x=x[:int(len(df)-1)]

y=df['Root Mean Square']

y=y[:int(len(df)-1)]

model.fit(x,y)

#This is how scikit-learn calculates model.score(X\_test,y\_test):

#x = ((y\_test - y\_predicted) \*\* 2).sum()

#y = ((y\_test - y\_test.mean()) \*\* 2).sum()

#score = 1 - (x/y)

#test the model

predictions=model.predict(x)

print('the model score is:',model.score(x,y))

# Output:

the model score is: 0.8849358321362218

# make the predictions

new\_data=df[['Magnitude','Latitude','Longitude','Depth']].tail(1)

prediction=model.predict(new\_data)

print('the model predicts the last row:',prediction)

print('actual value is:',df[['Root Mean Square']].tail(1).values[0][0])

# Output:

the model predicts the last row: [0.80072784] actual value is: 0.91