**TITLE : EARTHQUAKE PREDICTION MODEL USING PYTHON**

**INTRODUCTION:**

**An earthquake is shaking the surface of the Earth, which is caused as a result of the movable plate boundary.**

**Earthquakes are measured using remarks from seismometers with a Richter magnitude scale. Ground rupture, Landslides, Soil liquefaction and Tsunami are the main effects created by earthquakes. Today's earthquake warning systems used to provide regional notification of an earthquake in progress.**

**Developing an earthquake prediction model is a complex and ongoing scientific endeavor. While I can't provide real-time or the most up-to-date information, I can highlight some innovative techniques and approaches that were commonly used in earthquake prediction models up to my last knowledge update in January 2022. Please note that research in this field might have advanced since then.**

**Objective:**

**The objective of an earthquake prediction model using Python is to develop a system that can forecast the occurrence, magnitude, and location of earthquakes with improved accuracy.**

**Problem Understanding:**

**Data Source:**

**Creating a dataset for an earthquake prediction model using Python typically involves collecting and preparing earthquake-related data from reliable sources.**

**Data Preprocessing:**

**Data preprocessing is a crucial step in building an earthquake prediction model using Python. Properly preparing your data ensures that it's in a suitable format for training machine learning models and can significantly impact the model's performance.**

**Association analysis:**

**Association analysis is typically used for finding patterns, associations, and correlations within large datasets, which is different from earthquake prediction. Earthquake prediction typically involves predictive modeling, time series analysis, and geospatial analysis rather than association analysis.**

**Insight generation:**

**Once an association rule is created, it is important to scrutinize it to identify the most important rules. Creating an earthquake prediction model is a complex task that involves a combination of seismological, geological, and machine learning techniques.**

**Visualize:**

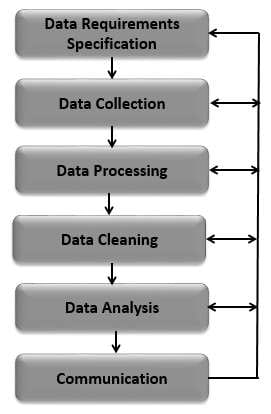
**Visualizing an earthquake prediction model can help you understand its performance and gain insights from the data.**

**Recommendations for business:**

**Building a business around an earthquake prediction model using Python can be a valuable endeavor, but it's essential to approach it with caution and responsibility due to the complexity and sensitivity of earthquake prediction.**

**Existing System**

**The existing system addresses novel methodology to predict the next earthquake. Apache hadoop is designed to run in a distributed environment and it manages the collection of various nodes running map and reduce functions. In this system data analysis performed on earthquake data year wise and location wise.**

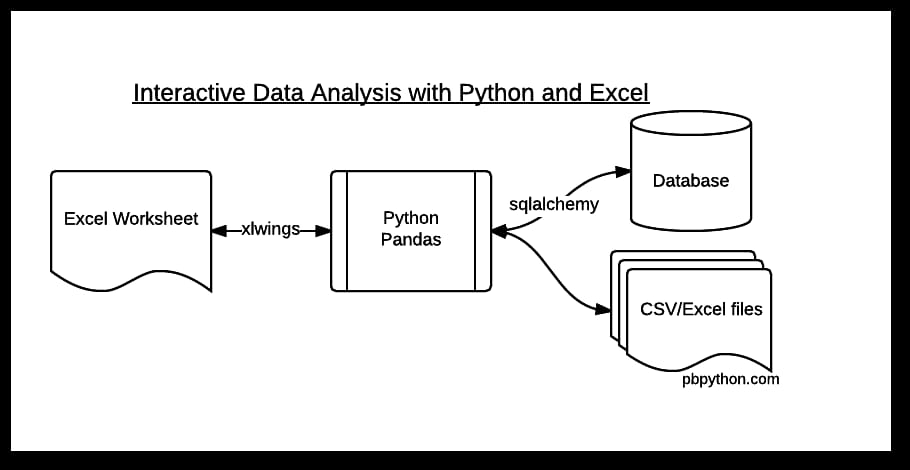
****

**Proposed system**

**In the future, the same Mapper and Reducer class will be implemented with pandas and matplotlib framework components working effectively. Pandas handle with a data is easy that's the proposed system runs effectively visualization method is matplotlib framework working with overcome the existing system. data parsing & data format conversion is easy way to pandas.**

**Design and Implementation**

**The prediction methodology itself is composed of three integral steps: data preprocessing, model selection and the final prediction process. In the dataset, a building was uniquely identified by 4 attributes: Building Identification, District Identification, Municipality Identification, Ward Identification. These attributes were added to the training data for identifying the building damage grade.**

****

**1)Pandas**

**The python library matplotlib, pandas, os are used in project.**

**In particular, it offers data structures and operations for manipulating numerical tables and time series**

**2)Matplotlib**

**Matplotlib is a comprehensive library for**

**creating static, animated, and interactive visualizations**

**in Python**

**3)Matplotlib**

**Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object oriented API for embedding plots into applications using general-purpose GUI toolkits likeTkinter, wxPython, Qt, or GTK+**

**4)Numpy**

**NumPy is the fundamental package for**

**scientific computing in Python NumPy arrays facilitate**

**advanced mathematical and other types of operations on**

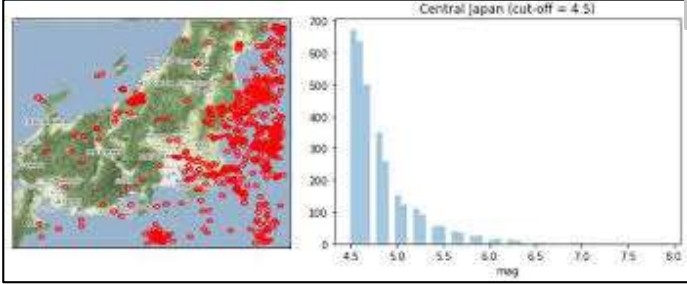
**large numbers of data.**

**5)Pycharm**

**In this tutorial, you operate in Scientific Mode and use Matplotlib and NumPy packages to run and debug a Python code with data visualization.**

**6) OS**

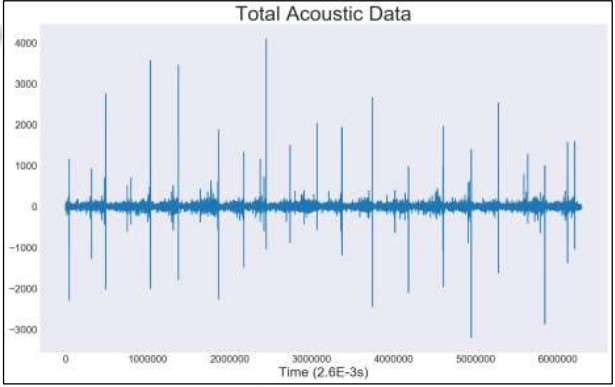
**The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux.**

****

**(Sample output for matplotlib)**

**The Matplot library used to find the targeted**

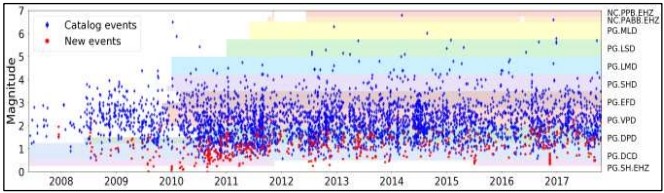
**graphical output using numpy.and derive data and clear output in user.**

****

**(Sample output for pandas library)**

**The library used to evaluvate a data and determind**

**and formatting a datasets.it provided a statical represented a output**

****

**( Sample output of the magnitude in matplotlib)**

**Furthermore, the analysis of earthquake prediction**

**results are carried. Past earthquake magnitude data are used as an input for the network.**

**STEPS TAKEN**

**1)Preprocessing of Dataset**

**2)Splitting the Datasets**

**3)Building models such as Linear Regression, Decision Tree and KNN**

**4)Visualization with Matplotlib and Seaborn**

**5)Prediction**

**IMPORTING LIBRARIES**

**# Importing Libraries**

**import numpy as np**

**import pandas as pd**

**import pandas as gdp**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import folium**

**from folium import Choropleth**

**from folium.plugins import HeatMap**

**import datetime**

**1)Loading Data**

**Python Code:**

**Loading data for tectonic plate boundaries**

**#loading data for tectonic plate boundaries**

**tectonic\_plates=pd.<aonclick="parent.postMessage({'referent':'.pandas.read\_csv'},'\*')">read\_csv("../input/tectonic-plate-boundaries/all.csv" )**

**tectonic\_plates.head()**

**2)Data Preprocessing**

**Steps to preprocess the data**

* **Date parsing: Parsing date to dtype datetime64(ns)**
* **Time Parsing: Parsing time to dtype timedelta64**
* **Adding Attributes: ” Date\_Time ” and ” Days “**

**#Parsing datetime**

**#exploring the length of date objects**

**lengths = data["Date"].str.len()**

**lengths.value\_counts()**

****

**#having a look at the fishy datapoints**

**wrongdates = np.<a onclick="parent.postMessage({'referent':'.numpy.where'}, '\*')">where([lengths == 24])[1]**

**print("Fishy dates:", wrongdates)**

**data.loc[wrongdates]**

****

**#fixing the wrong dates and changing the datatype from numpy object to datetime64[ns]**

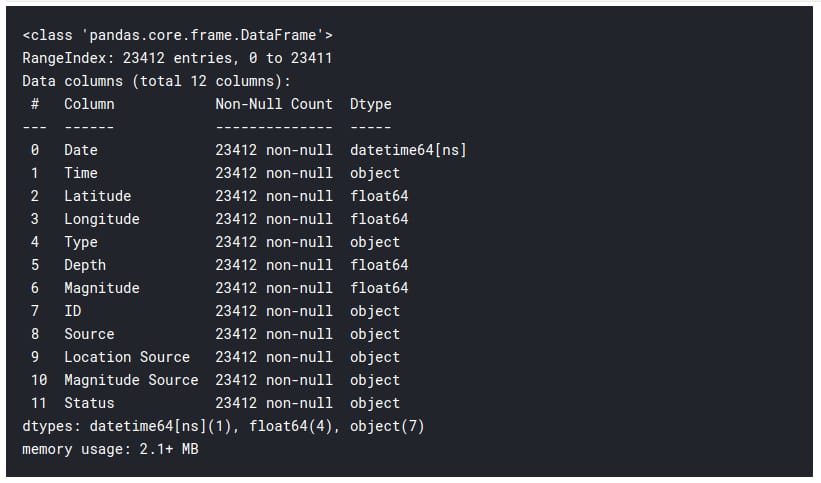
**data.loc[3378, "Date"] = "02/23/1975"**

**data.loc[7512, "Date"] = "04/28/1985"**

**data.loc[20650, "Date"] = "03/13/2011"**

**data['Date']= pd.<a onclick="parent.postMessage({'referent':'.pandas.to\_datetime'}, '\*')">to\_datetime(data["Date"])**

**data.info()**

****

**#We have time data too. Now that we are at it,lets parse it as well.**

**#exploring the length of time objects**

**lengths = data["Time"].str.len()**

**lengths.value\_counts()**

****

**#Having a look at the fishy datapoints**

**wrongtime = np.where([lengths == 24])[1]**

**print("Fishy time:", wrongtime)**

**data.loc[wrongtime]**

****

**#Ah! Is it deja vu or are those the same datapoints**

**#fixing the wrong time and changing the datatype from numpy object to timedelta64[ns]**

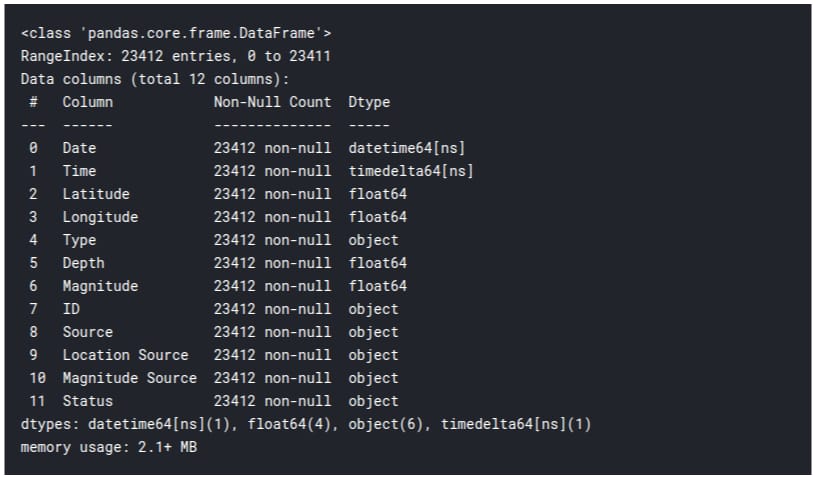
**data.loc[3378, "Time"] = "02:58:41"**

**data.loc[7512, "Time"] = "02:53:41"**

**data.loc[20650, "Time"] = "02:23:34"**

**data['Time']= pd.<a onclick="parent.postMessage({'referent':'.pandas.to\_timedelta'}, '\*')">to\_timedelta(data['Time'])**

**data.info()**

****

**#I don't think there is any point in doing this step, but why not!**

**data["Date\_Time"]=data["Date"] +data["Time"]**

**#Now that we have Date, Time (and Date\_Time) we totally should have Days of week too.**

**data["Days"]= data.Date.dt.strftime("%A")**

**#Lets have a look at data**

**data.head()**

**3) Data analysis**

**First, we make a line plot of magnitudes with dates. This will show a time series of various earthquakes around the world from 1965 to 2016. An initial inference that can be drawn by looking at the plot is that there was relatively high seismic activity from 1965 to the early 1970s.**

**#plotting a lineplot with magnitudes with respectto dates**

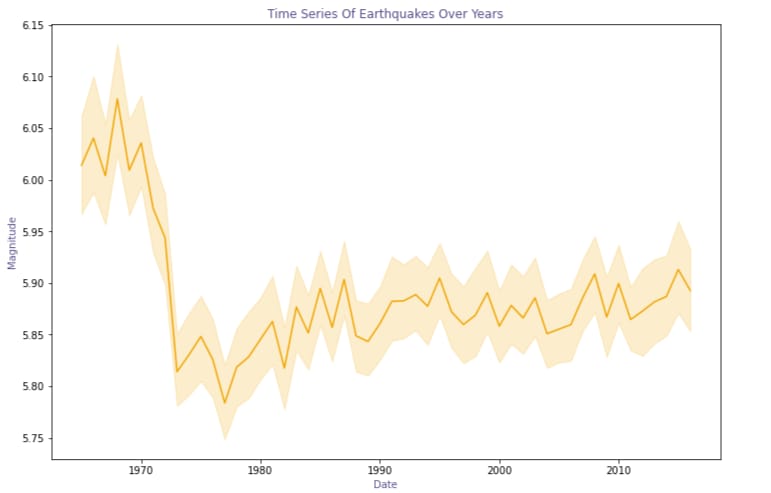
**plt.<a onclick="parent.postMessage({'referent':'.matplotlib.pyplot.figure'}, '\*')">figure(figsize=(12,8))**

**Time\_series=sns.<a onclick="parent.postMessage({'referent':'.seaborn.lineplot'}, '\*')">lineplot(x=data['Date'].dt.year,y="Magnitude",data=data, color="#ffa600")**

**Time\_series.set\_title("Time Series Of Earthquakes Over Years", color="#58508d")**

**Time\_series.set\_ylabel("Magnitude", color="#58508d")**

**Time\_series.set\_xlabel("Date", color="#58508d")**

****

**However, our dataset contains earthquakes caused by other factors like normal explosions, Nuclear explosions, and rockburst. In order to draw a clear relation between seismic activities and earthquakes, we have to rule out the other factors.**

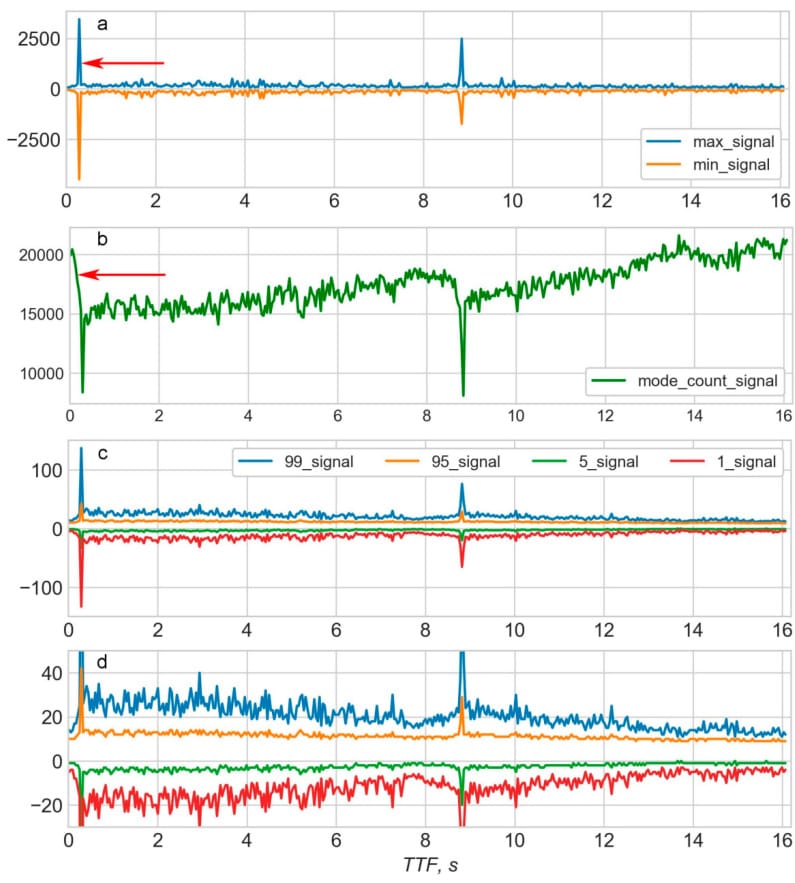
**Feature Engineering**

**The following key approach was used for feature engineering. In the first step, it is assumed that the distribution of AD is the source of useful features. This assumption is based on “common sense” suggestions, observation of changes in AD distribution over time and also on results published in related works .**

**It is evident that stick-slip failure is preceded by a number of spikes . These spikes appear as a result of micro failure events and may predict TTF. Generally, the shorter the TTF the more frequent the AD spikes. Hence, the statistical characteristics of AD may serve as features for modeling.**

**A total of 18 statistical features were derived from each of the 150 K pieces of AD in this work. Nine of these statistical features were maximum, minimum, mean, standard deviation, (standard deviation)/(mean), skewness, kurtosis, mode, number of mode appearance. The remaining nine features were percentiles at the following percent levels: 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th and 99th. The “maximum” and “minimum” features were calculated but not used for modeling because these features were only used to indicate the main EQ event due to their outstandingly large values.**

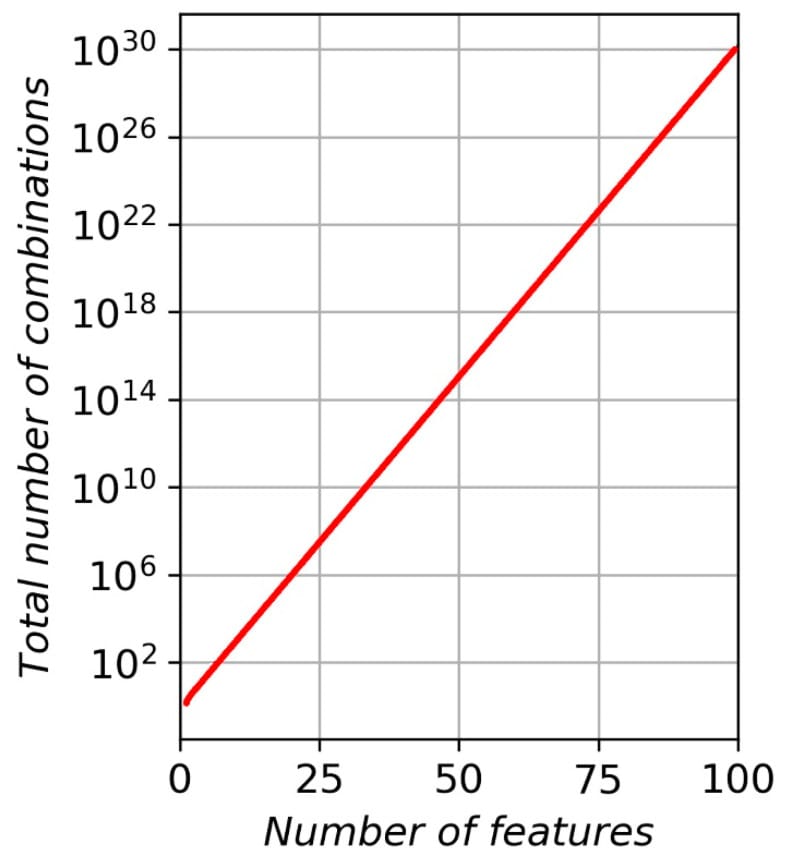
**If values recorded after the EQ were incorporated, then an additional error would appear in the model. This is due to the similarity in feature values at the beginning and the end of a cycle.**

****

**In order to increase the model accuracy, all tail rows which correspond to the period after the EQ should be deleted from the database of statistical features. Another rationale for deleting data after an EQ is that the main goal of modelling using data from laboratory EQs is to identify features that would be useful to predict real EQs. It is obvious that in reality only data before an EQ would be used for prediction. Any data after an EQ has neither a logical nor practical sense for prediction of that particular EQ.**

**Due to the development of modelling tools such as Python and appropriate libraries, training a model can be performed rapidly using only several lines of code. The major challenge in training any model is determining which features should be used.**

**In the current work, the final selection of features was based on the building of different models to compare MAEs and picking the best combination of features that gives the lowest MAE. However, according to the well-known curse of dimensionality, the total number of possible combinations of features increases far faster than the number of features in the set.**

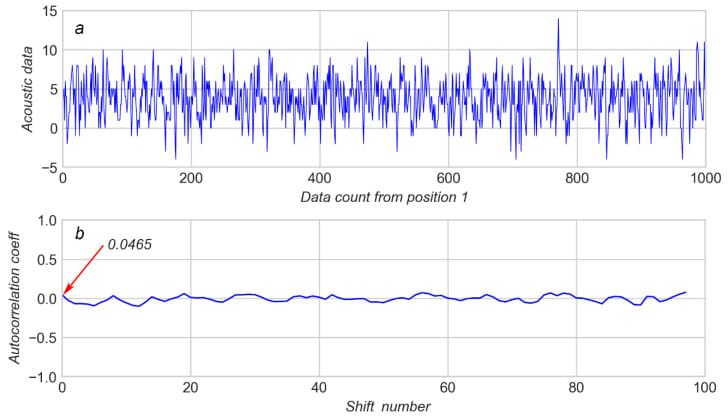
****

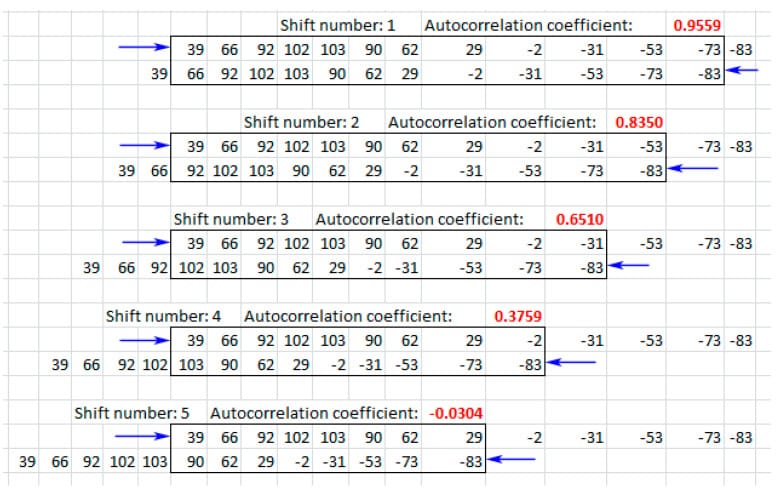
**In the paper only two best features were chosen from 43 for prediction of TTF. This means that most of the features are either excessive or not suitable for modelling. Therefore, it can be concluded that the first step in feature engineering is to exclude all non-significant features from the set. Every feature excluded can significantly decrease the total amount of combinations to examine during the BF approach. In our, case excluding only two features decrease the number of combinations from 262,143 to 64,995 (16 features instead of 18).**

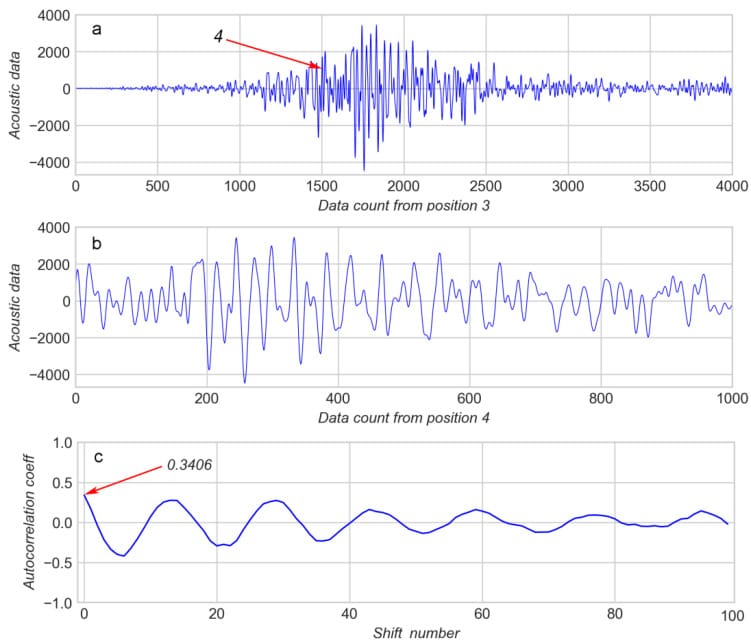
**The “maximum” and “minimum” features can be excluded based on the following reasoning: The maximum values of AD in 150 K pieces is equivalent to the 100th percentile value. This study uses the “99th percentile” feature which is close to the 100th percentile; therefore the 100th percentile (i.e., “maximum”) is superfluous and can be excluded. Similarly, the “minimum” feature is equivalent to the “0 percentile” and can be excluded as the “1st percentile” feature has already been considered. The only reason for calculating “maximum” and “minimum” features is because they are needed for correct identification of the 150 K piece which contains the EQ. It also helps to correctly delete tail rows containing AD after the EQ. After excluding “maximum” and “minimum” features, 16 features remain in the model, giving a total of 64,995 possible combinations.**

**The backward feature elimination technique (BFE) was employed for reducing the number of features. The rationale behind using this method in the current study is that, if there is a total of n features in a set, then there are n possible combinations of (n−1) features in the subset. Assuming that the vast majority of features are either bad or neutral for model quality, it is highly probable that the MAE for the model—which uses all n features—would be bigger than the least MAE for n models using (n−1) features. If so, then only one model is needed such that it uses all n features, and MAEn is then calculated; thereafter, n models are required, each of which uses one of the possible subsets of (n−1) features. It is also important to choose the subset which results in the least MAEn−1. If a full set of n features contains at least one feature that is bad or excessive for the model, then MAEn would be greater than or equal to the least MAEn−1. This bad or excessive feature should be absent in the subset that generates the model with the least MAEn−1. Thus, this feature can be excluded from the set of features. BFE takes about a minute in semiautomatic mode to exclude one feature and can be fully automated if necessary. BFE is consistently used to reduce the number of features from n to about 10. Thereafter, the straight BF method is used to find the combination of features that gives a model with the least MAE.**

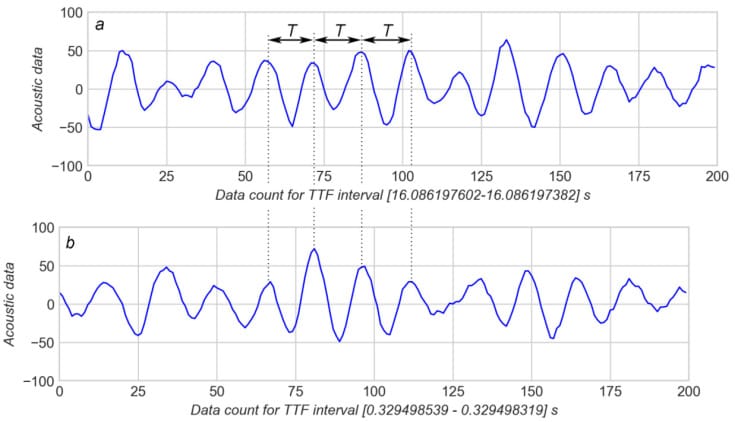
**The next important point to consider is how many CV cycles are necessary for every step of the work. Each single CV cycle returns the mean MAE for only six calculations in total. Therefore, the resulting MAE varies at the second decimal point from one run to the other. In order to decrease the variance of MAE the number of repetitions (cycles) of CV should be increased. Two cycles (CV-2) were used for BFE and 500 cycles (CV-500) were used in the final calculation of MAE for the best combination of features. Using CV-500 enables MAEs that are stable to the third decimal point to be obtained.**

****

****

****

**The frequency of AD oscillation was considered during a spike as an additional feature which can be used for modelling. However, the comparison of AD of early and late spikes shows that periods of oscillation T for both cases are approximately equal .Therefore, the frequency of AD oscillation during a spike was not used for modelling.**

****

**This way, three major parameters were used for modelling which are: acoustic data (AD); the first value of AC on every “sliding window” (AC\_first); the amplitude of AC on every “sliding window” (AC\_ampl). Each 150 K piece of AD contains 150 sliding windows and, therefore, 150 values of AC\_first and 150 values of AC\_ampl. Because 16 statistics were calculated for each of three parameters, the overall number of features considered was 48.**

**These features were calculated for every separate seismic cycle in the database provided.For every portion containing 150 K of AD, the features were calculated and recorded with the corresponding TTF in the separate file of features. Since the TTF change during 150 K of AD was just 0.04 s, the TTF value is considered a constant during any given 150 K piece. The last value of TTF in the 150 K piece was used as this constant time. Maximum and minimum values of AD were also recorded; they allowed the row that contained the EQ event to be located. All rows after the row with EQ event were deleted according to the reasons explained above. It should be noted that in all files TTF for the EQ event was approximately the same (near 0.3 s).**

**Earthquake prediction model training**

**It is well known that if a disaster occurs in one region, it is likely to happen again. Some regions have frequent earthquakes, but this is only a comparative amount compared to other regions.**

**So, predicting the earthquake with date and time, latitude and longitude from previous data is not a trend that follows like other things, it happens naturally.**

**I will start this task to create a model for earthquake prediction by importing the necessary python libraries:**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**data = pd.read\_csv("database.csv")**

**data.columns**

**Index(['Date', 'Time', 'Latitude', 'Longitude', 'Type', 'Depth', 'Depth Error',**

**'Depth Seismic Stations', 'Magnitude', 'Magnitude Type',**

**'Magnitude Error', 'Magnitude Seismic Stations', 'Azimuthal Gap',**

**'Horizontal Distance', 'Horizontal Error', 'Root Mean Square', 'ID',**

**'Source', 'Location Source', 'Magnitude Source', 'Status'],**

**dtype='object')**

**Now let’s see the main characteristics of earthquake data and create an object of these characteristics, namely, date, time, latitude, longitude, depth, magnitude:**

**data = data[['Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude']]**

**data.head()**

**Since the data is random, so we need to scale it based on the model inputs. In this, we convert the given date and time to Unix time which is in seconds and a number. This can be easily used as an entry for the network we have built:**

**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()**

**Any innovative technologies or approaches used during the development:**

**1)Machine Learning and Deep Learning:**

**Implementing machine learning and deep learning algorithms, such as neural networks and support vector machines, to analyze seismic data and make predictions based on patterns and historical earthquake data.**

**2)Seismic Data Preprocessing:**

**Advanced data preprocessing techniques to clean and enhance the quality of seismic data, which can include denoising, filtering, and data augmentation.**

**3)Feature Engineering:**

**Feature selection and engineering to extract relevant features from seismic data, such as P-wave and S-wave arrival times, spectral features, and spatial characteristics.**

**4)Ensembling Methods:**

**Using ensemble methods like Random Forest, Gradient Boosting, or stacking to combine the predictions of multiple models for improved accuracy.**

**5)Transfer Learning:**

**Leveraging pre-trained models from other domains, like computer vision or natural language processing, and fine-tuning them for earthquake prediction tasks.**

**6)Data Fusion:**

**Integrating various data sources, including seismic data, satellite imagery, geological data, and social media information, to improve the model's predictive capabilities.**

**7)Real-time Data Streaming:**

**Building systems that can handle real-time data streams from seismometers and other sensors, allowing for immediate analysis and alerts.**

**8)Geographic Information Systems (GIS):**

**Incorporating geographic and geological information through GIS tools to model the impact of terrain, fault lines, and tectonic plate movements.**

**9)Uncertainty Quantification:**

**Developing methods to estimate and communicate the uncertainty associated with earthquake predictions, which is crucial for decision-makers and the public.**

**11)Continuous Model Improvement:**

**Implementing online learning techniques that allow the model to adapt and improve over time as new earthquake data becomes available.**

**12)Public Engagement and Education:**

**Developing tools and applications that provide real-time earthquake information and educational resources to increase public awareness and preparedness.**

**INSAR**

**The main technology used in earthquake prediction is InSAR (Interferometric Synthetic Aperture Radar), which is a satellite-based remote sensing technique. InSAR works by measuring the differences in the Earth's surface over time and detecting changes in the Earth's crust.**

**Other methods used to predict an earthquake:**

**In order to understand how to forecast earthquakes, several approaches have been explored. Seismicity changes, changes in seismic wave speed, electrical changes, and groundwater alterations are some of the most serious techniques that have been investigated.**

**New Technologies That Can Revolutionize Earthquake Prevention Strategies:**

**From early warning systems to smart infrastructure, these technologies are helping communities better prepare for the inevitable shaking of the earth and minimize the damage caused by these earthquakes.**

**Drones and satellites are also being deployed to aid relief efforts.**

**Technology is used to monitor earthquakes:**

**Seismographs are instruments used to record the motion of the ground during an earthquake. They are installed in the ground throughout the world and operated as part of a seismographic network.**

**Technology improved our understanding of earthquakes:**

**The GPS measurements will also be useful during and after earthquakes. Scientists can measure ground motions from earthquakes, and identify the fault that ruptured and help evaluate regional deformation and stress changes in near-real-time with an automated system.**

**CONCLUSION :**

**This work presents that the Random Forest Classifier algorithm has the highest accuracy in predicting the damage due to earthquakes, based on the F1 score calculated for each of the four algorithms previously mentioned in this work. K Nearest Neighbors has been observed to be the second most preferred algorithm for earthquake damage prediction. On analysis of the materials that help curb damage to buildings during an earthquake, the work concludes that Reinforced Concrete is the material most suited to the cause. Earthquakes are well known to excite electromagnetic pulse, that cause tremors under the Earth’s crust. These electromagnetic pulses are shielded effectively by Reinforced Concrete. Reinforced concrete has a low tensile strength, and hence Steel bars are used, which are embedded in the concrete sets. This provides Reinforced Concrete with immense ability to withstand natural calamities such as Earthquakes. The applications of this work can be further extended earthquakes to predict damage caused by Earthquakes in areas & time also possible for which a similar and relevant dataset can be obtained.**

**Remember that earthquake prediction is a complex and challenging problem, and reliable predictions are difficult to achieve. You may need to explore various algorithms and continuously update your model as new data becomes available. Additionally, consider the ethical and practical implications of earthquake prediction, as false alarms can have serious consequences.**

**Earthquake prediction remains a challenging field, and accurate long-term prediction is still a subject of ongoing research. Additionally, the success of any model depends on the quality and quantity of data available and the expertise of the research team involved in its development. Always refer to the latest research and publications for the most current techniques and approaches in earthquake prediction.**