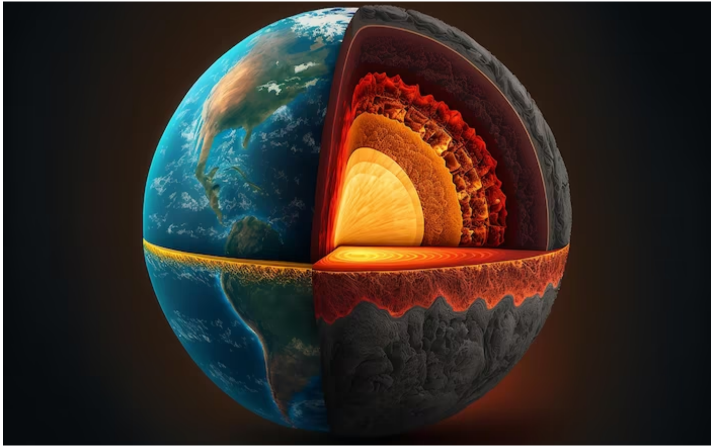
EARTHQUAKE PREDICTION MODEL USING PYTHON

Group 1

Team Members

Akash M
Dinesh R
SM Joseph Clement
S P Siva
E Subash
P Kabilan
D Mani Kandan
B Dinesh Raj



INTRODUCTION

Welcome to the presentation on Unveiling Earthquake Patterns: A Robust Predictive Model Utilizing Python. In this presentation, we will explore the development of a powerful predictive model using Python to analyze earthquake patterns. We will discuss the importance of earthquake prediction, the challenges involved, and how Python can help address these challenges.

UNDERSTANDING EARTHQUAKES

Before delving into the predictive model, it is crucial to understand the basics of earthquakes. Earthquakes are natural phenomena caused by the sudden release of energy in the Earth's crust, resulting in seismic waves. They occur due to tectonic plate movements and can have devastating consequences. By analyzing earthquake patterns, we can gain insights into their occurrence and potentially predict future events.

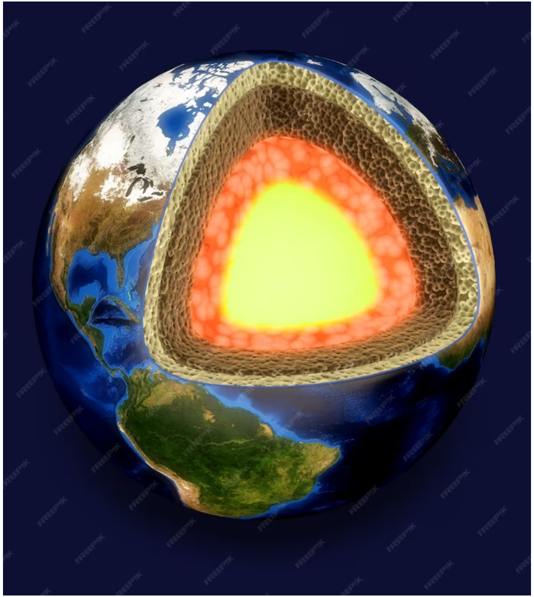
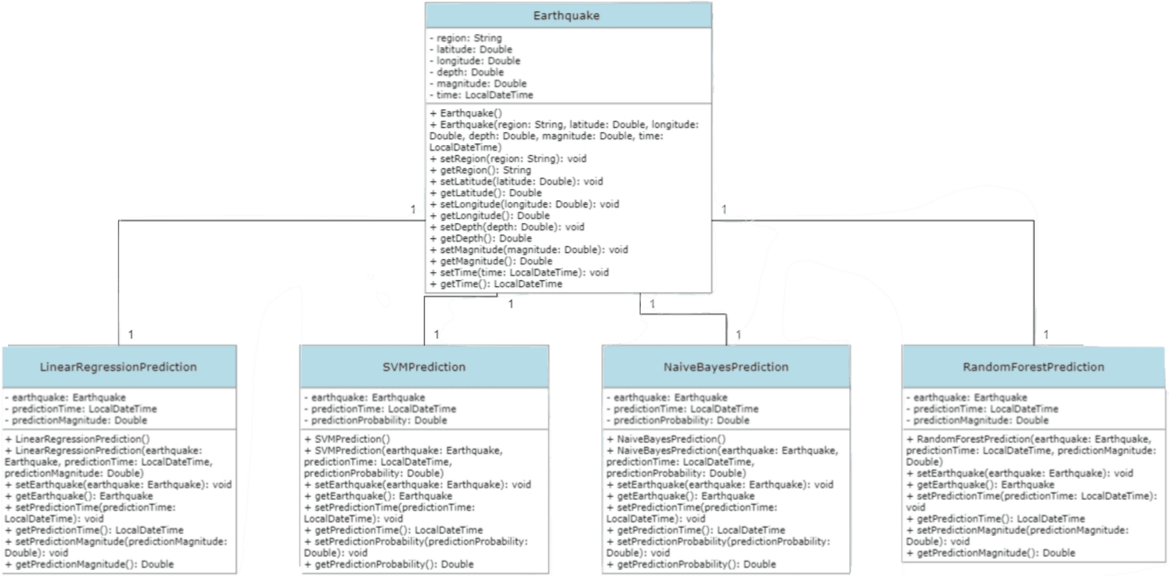
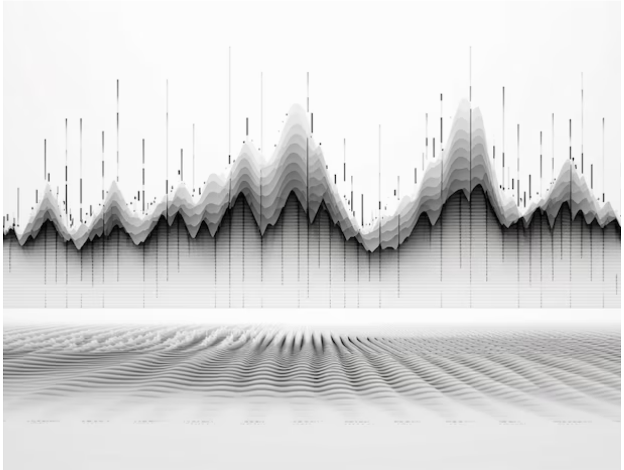


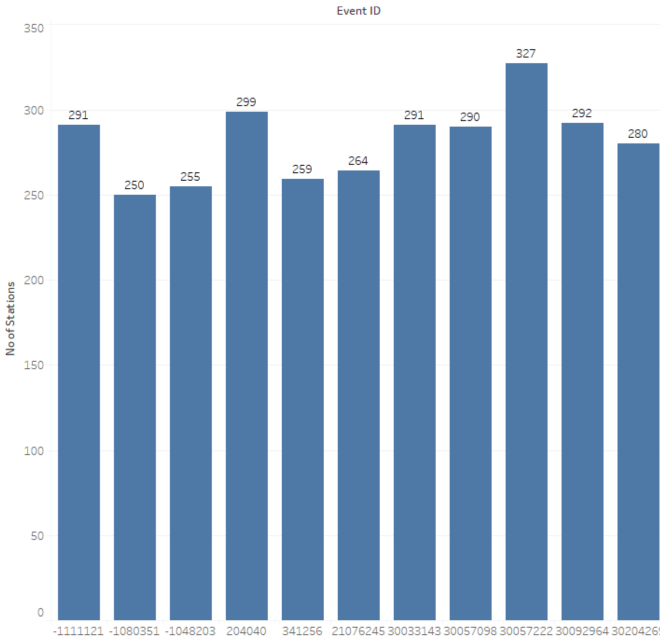
Diagram :


Earthquake
- region: String
- latitude: Double
- longitude: Double
- depth: Double
- magnitude: Double
- time: LocalDateTime
+ Earthquake()
+ Earthquake(region: String, latitude: Double, longitude: Double, depth: Double, magnitude: Double, time: LocalDateTime)
+ setRegion(region: String): void
+ getRegion(): String
+ setLatitude(latitude: Double): void
+ getLatitude(): Double
+ setLongitude(longitude: Double): void
+ getLongitude(): Double
+ setDepth(depth: Double): void
+ getDepth(): Double
+ setMagnitude(magnitude: Double): void
+ getMagnitude(): Double
+ setTime(time: LocalDateTime): void
+ getTime(): LocalDateTime
1
1
1
1
1
1
1
1
LinearRegressionPrediction
- earthquake: Earthquake
- predictionTime: LocalDateTime
- predictionMagnitude: Double
+ LinearRegressionPrediction()
+ LinearRegressionPrediction(earthquake: Earthquake, predictionTime: LocalDateTime, predictionMagnitude: Double)
+ setEarthquake(earthquake: Earthquake): void
+ getEarthquake(): Earthquake
+ setPredictionTime(predictionTime: LocalDateTime): void
+ getPredictionTime(): LocalDateTime
+ setPredictionMagnitude(predictionMagnitude: Double): void
+ getPredictionMagnitude(): Double
SVMPrediction
- earthquake: Earthquake
- predictionTime: LocalDateTime
- predictionProbability: Double
+ SVMPrediction()
+ SVMPrediction(earthquake: Earthquake, predictionTime: LocalDateTime, predictionProbability: Double)
+ setEarthquake(earthquake: Earthquake): void
+ getEarthquake(): Earthquake
+ setPredictionTime(predictionTime: LocalDateTime): void
+ getPredictionTime(): LocalDateTime
+ setPredictionProbability(predictionProbability: Double): void
+ getPredictionProbability(): Double
NaiveBayesPrediction
- earthquake: Earthquake
- predictionTime: LocalDateTime
- predictionProbability: Double
+ NaiveBayesPrediction()
+ NaiveBayesPrediction(earthquake: Earthquake, predictionTime: LocalDateTime, predictionProbability: Double)
+ setEarthquake(earthquake: Earthquake): void
+ getEarthquake(): Earthquake
+ setPredictionTime(predictionTime: LocalDateTime): void
+ getPredictionTime(): LocalDateTime
+ setPredictionProbability(predictionProbability: Double): void
+ getPredictionProbability(): Double
RandomForestPrediction
- earthquake: Earthquake
- predictionTime: LocalDateTime
- predictionMagnitude: Double
+ RandomForestPrediction(earthquake: Earthquake, predictionTime: LocalDateTime, predictionMagnitude: Double)
+ setEarthquake(earthquake: Earthquake): void
+ getEarthquake(): Earthquake
+ setPredictionTime(predictionTime: LocalDateTime): void
+ getPredictionTime(): LocalDateTime
+ setPredictionMagnitude(predictionMagnitude: Double): void
+ getPredictionMagnitude(): Double

CHALLENGES IN EARTHQUAKE PREDICTION

Predicting earthquakes is a complex task due to several challenges. Earthquake occurrence is influenced by various factors, including geological characteristics, fault lines, and historical seismic data. Additionally, earthquakes often exhibit unpredictable behavior, making accurate forecasting difficult. Overcoming these challenges requires advanced data analysis techniques and the utilization of powerful computational tools like Python.





Sum of No of Stations for each Event ID. The data is filtered on No of Stations, which includes values greater than or equal to

Linear Regration

Linear regression is a type of supervised machine learning algorithm that is used to model the linear relationship between a dependent variable (in this case, earthquake magnitude) and one or more independent variables (in this case, latitude, longitude, depth, and the number of seismic stations that recorded the earthquake).The basic idea behind linear regression is to find the line of best fit through the data that minimizes the sum of the squared residuals (the difference between the predicted and actual values of the dependent variable). The coefficients of the line of best fit are estimated using a method called ordinary least squares, which involves minimizing the sum of the squared residuals with respect to the coefficients.

MODEL PROGRAM

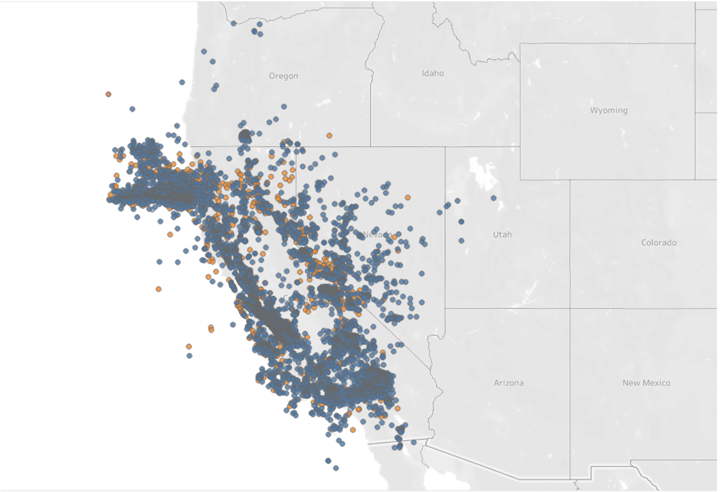
```
import numpy as np
from sklearn.model_selection import cross_val_score
from catboost import CatBoostRegressor

# set output float precision
np.set_printoptions(precision=3
# init model
model = CatBoostRegressor(random_seed=0,
verbose=False)
# calculate mae on folds
mae = cross_val_score(model, data[best_features],
data['target'],
cv=5, scoring='neg_mean_absolute_error', n_jobs=8)
# print the results
print('folds: {}'.format(abs(mae)))
print('total: {:.3f}'.format(np.mean(abs(mae))))
```

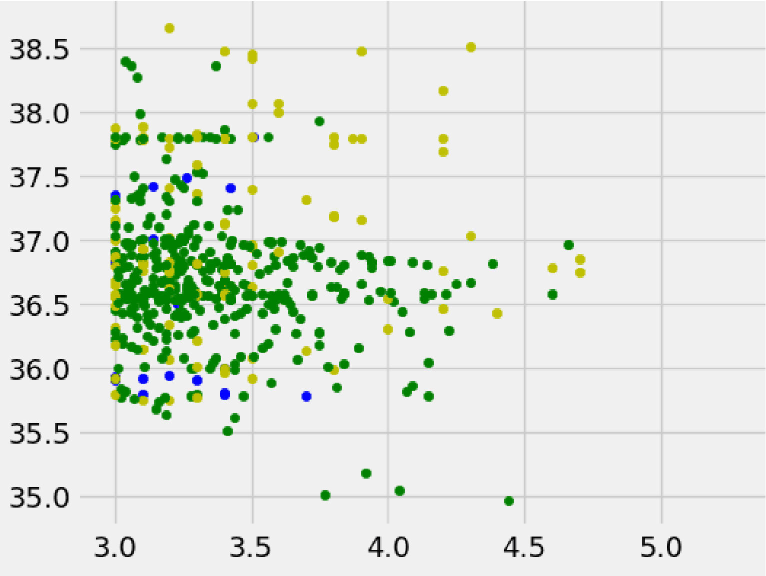
REAL-TIME EARTHQUAKE PREDICTION

Real-time earthquake prediction is a significant advancement in the field. By continuously analyzing incoming data using our predictive model, we can provide timely warnings and alerts. Python's ability to handle streaming data and perform real-time analysis makes it an ideal choice for implementing such systems. Real-time prediction empowers us to take proactive measures and save lives during seismic events.

gnitude type



ıde (generated). Color shows details about Magnitude type. Details are shown for Latitude(deg) and Longitude(deg).



MODEL EVALUATION AND VALIDATION

Evaluating and validating the predictive model is crucial to ensure its effectiveness. Python offers a range of evaluation metrics and techniques to assess the model's performance, such as accuracy, precision, recall, and F1-score. Cross-validation and train-test splits are commonly used for validation. Through rigorous evaluation and validation, we can fine-tune the model and enhance its predictive capabilities.

APPLICATIONS AND IMPACT

The applications and impact of accurate earthquake prediction are profound. Early warnings can save lives, enable timely evacuations, and facilitate effective disaster management. Infrastructure planning can be enhanced to withstand seismic events, reducing damage and economic losses. Governments, organizations, and communities can make informed decisions and allocate resources strategically. By leveraging Python's predictive capabilities, we can make a significant positive impact on earthquake-prone regions worldwide.

LIMITATIONS AND FUTURE SCOPE

While our predictive model using Python is a significant step forward, it is essential to acknowledge its limitations. Earthquake prediction is a complex and evolving field, and our model may not capture all the intricacies. Enhancements can be made by incorporating more data sources, refining algorithms, and exploring new machine learning techniques. Continuous research and development will pave the way for more accurate earthquake predictions in the future.

THANKING YOU !