Tech Saksham

Capstone Project Report

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FUNDAMENTALS

“Project Name”

“College Name”

NM ID	**NAME**
au510321214006	VISHALINI P

Trainer Name

Ramar Bose

Sr.AI Master Trainer

ABSTRACT

Earthquakes are natural disasters that can cause significant damage and loss of life. Accurate prediction of earthquakes is essential for developing early warning systems, disaster planning, risk assessment, and scientific research. This project aims to predict the magnitude and probability of Earthquake occurring in a particular region (California, United States) from the historic data of that region using various Machine learning model.

INDEX

CHAPTER 1
INTRODUCTION

1.1 PROBLEM STATEMENT

Earthquakes are one amongst the foremost destructive natural disasters. They typically occur without notice and do not allow much time for people to react. Earthquake can cause serious injuries and loss of life and destroy numerous buildings and infrastructure, leading to great economy loss. Machine learning is a subfield of Artificial Intelligence, which is broadly defined as the capability of machine to imitate the intelligent human behavior.

1.2 PROPOSED SOLUTION

As earthquake is a calamitous occurrence that is detrimental to human interest and has an undesirable impact on the environment. Earthquake prediction is branch of seismology concerned with the specification of the time, location and magnitude of future earthquakes. The prediction of earthquakes is clearly critical to the protection of our society, where we discover this as a motivating problem to be solved.

1.3 FEATURE

In the ongoing pursuit of understanding earthquake, scientists have endeavoured to develop approaches that can be used to predict earthquakes. The predication of earthquakes revolves around three main features:

1. Possible date
2. Time
3. Location
4. Magnitude.

1.4 ADVANTAGES

1. Enable emergency measure to reduce death and destruction.
2. propagate and treat uncertainties.
3. provide computational efficiency.

1.5 SCOPE

While mining this data set through normal EDA process I came across the fact that not all earthquakes are natural and few are indeed caused by humans although very small in numbers. At the end of the analysis I have tried to predict earthquakes and other quakes(seismic activities related to explosion, quarry blast etc.). I have also tried to handle the class imbalance problem because the data set is 98:2 The next steps are pretty usual ones with loading and probing the data. Let's get started with loading the libraries first.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 Services Used

Data collection:

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing:

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

Construction of a Predictive Model:

Machine learning needs data gathering have lot of past data‘s. Data gathering have sufficient historical data and raw data.Before data preprocessing, raw data can‘t be used directly. It‘s used to preprocess then, what kind of algorithm with model. Training and testing this model working and 20 predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

2.2 Tools and Software used

Hardware requirements:

Any personal computer that meets the following specification;

1. Processor- core i3 and above
2. Speed-1.2 GHZ
3. Ram-4 GB
4. Hard disk-20 GB

Software requirements:

Any personal computer that meets the following specifications:

1. Operating System-Linux, Windows 7 and Above

2. Language- python, Machine Learning

3.Tools- Jupyter Notebook 5.7.8 or higher

4. Framework-Flask

CHAPTER 3

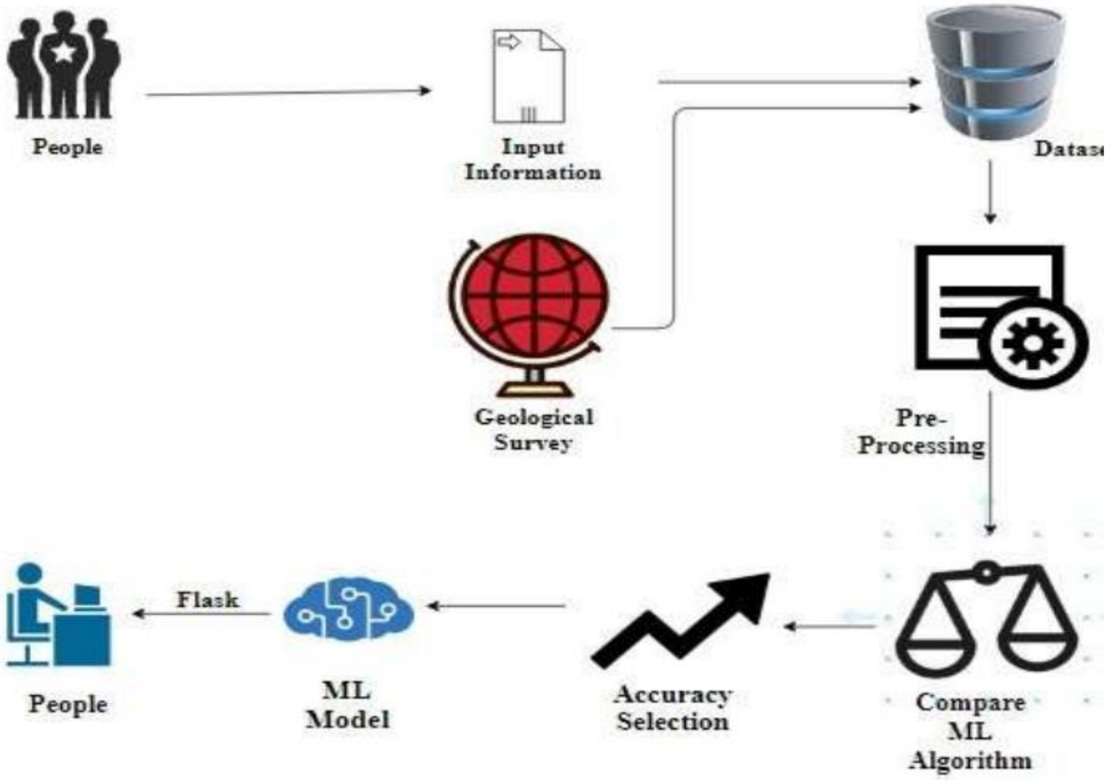
PROJECT ARCHITECTURE

3.1 Architecture

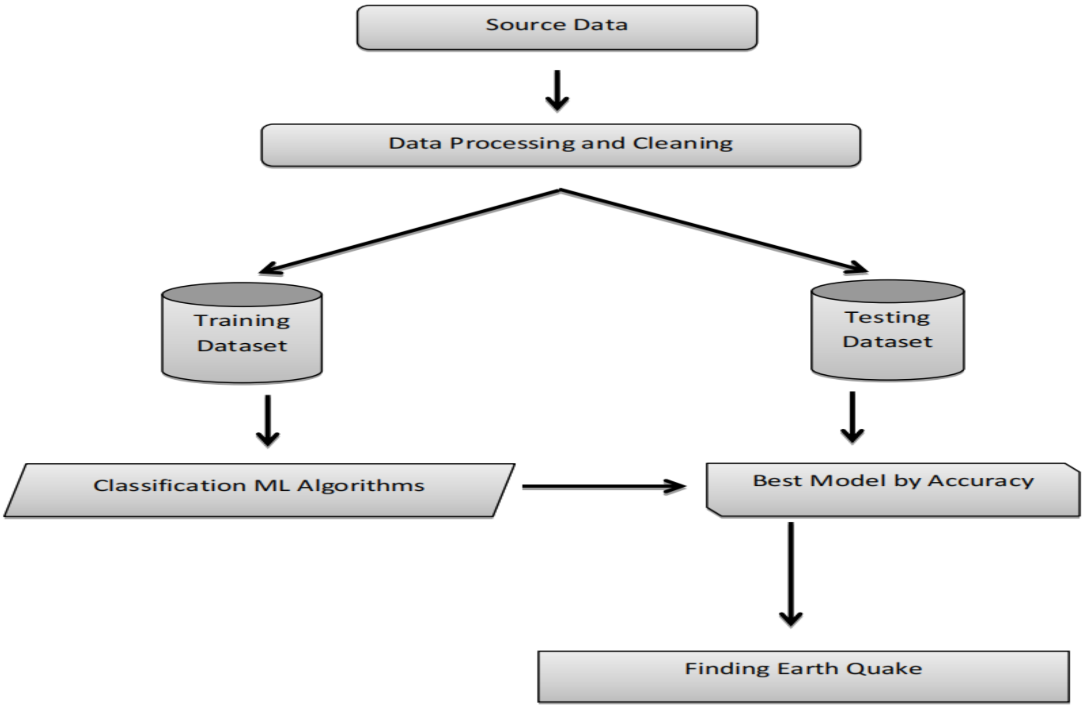
Earthquake prediction

1. **System flow diagram -User interface will work**
2. **Data flow diagram- How data is flow in your project**
3. **Module explain- Submodule u have do the diagram**

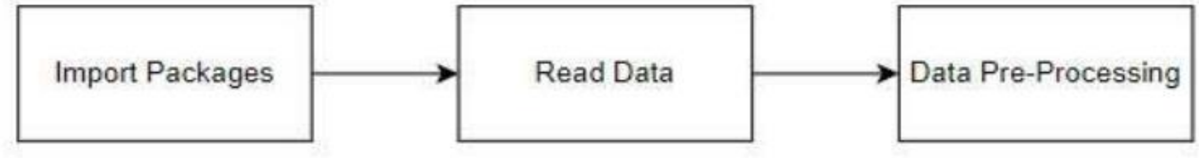
1. System flow diagram - User interface will work



2. Data flow diagram - How data is flow in your project



3. Module explain - Submodule you have do the diagram



CHAPTER 4

PROJECT OUTCOME

Implementation

We will use four models in this project:

1. Linear regression
2. Support Vector Machine(SVM)
3. NaiveBayes
4. Random Forest

Linear Regression

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.

In this situation, we have used multiple linear regression to model the relationship between earthquake magnitude and latitude, longitude, depth, and the number of seismic stations that recorded the earthquake. The multiple linear regression model assumes that there is a linear relationship between the dependent variable (magnitude) and each of the independent variables (latitude, longitude, depth, and number of seismic stations), and that the relationship is additive (i.e., the effect of each independent variable on the dependent variable is independent of the other independent variables).

Once the model has been fit to the data, we can use it to predict the magnitude of a new earthquake given its latitude, longitude, depth, and the number of seismic stations that recorded it. This can be useful for earthquake monitoring and early warning systems, as well as for understanding the underlying causes of earthquakes and improving our ability to predict them in the future.

SVM

Support Vector Machines (SVM) is a type of supervised machine learning algorithm that can be used for both regression and classification tasks. The basic idea behind SVM is to find the best boundary that separates the data into different classes or predicts a continuous output variable (in this case, earthquake magnitude).

In SVM, the data points are mapped to a higher-dimensional space where the boundary can be easily determined. The best boundary is the one that maximizes the margin, which is the distance between the boundary and the closest data points from each class. This boundary is called the "hyperplane."

For regression tasks, SVM uses a similar approach but instead of a hyperplane, it finds a line (or curve in higher dimensions) that best fits the data while maximizing the margin. This line is the "support vector regression line."

SVM can handle both linear and non-linear data by using different kernels that transform the data into a higher-dimensional space. Some commonly used kernels include linear, polynomial, and radial basis function (RBF) kernels.

Once the SVM model has been trained on the data, it can be used to predict the magnitude of a new earthquake given its features (latitude, longitude, depth, and number of seismic stations). This can be useful for predicting the magnitude of earthquakes in real-time and for better understanding the factors that contribute to earthquake occurrence.

Naive Bayes

In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models,[1] but coupled with kernel density estimation, they can achieve high accuracy levels.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[3]: 718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the code, we used the Naive Bayes classifier to predict the magnitude of earthquakes based on their latitude, longitude and number of monitoring stations.

Random Forest

Random forest is a machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to create a more accurate and robust model.

The basic idea behind random forest is to create multiple decision trees, each trained on a subset of the data and a random subset of the features. Each tree makes a prediction, and the final prediction is the average (for regression) or the mode (for classification) of the individual tree predictions. By creating many trees and taking their average, random forest can reduce the impact of overfitting and improve the accuracy and stability of the model.

In the code we provided earlier, we used the random forest algorithm to predict the magnitude of earthquakes based on their latitude, longitude, depth, and number of monitoring stations. We split the data into training and testing sets, trained the random forest model on the training data, and evaluated its performance on the test data using the mean squared error (MSE) and R-squared (R2) score.

Procedure:

1. Environment Setup
2. Data Acquisition
3. Data Loading and Preprocessing
4. Feature Engineering (Optional)
5. Data Splitting
6. Model Selection and Training
7. Model Evaluation
8. Prediction

Steps to implement

1. Import necessary libraries:

```
importpandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor# Example model
```

2. Load the downloaded data from a panda data frame

```
data = pd.read_csv(″earthquake_data.csv″)
```

3. Explore the data to understand its structure and identify potential issues:

```
data.head()  # View the first few rows
data.describe()  # Get summary statistics
data.isnull().sum()  # Check for missing values
```

4. Divide the data into training and testing sets for model training and evaluation:

```
X = data[[″feature1″, ″feature2″, ...]]  # Select features (predictors)
y = data[″magnitude″]  # Target variable (magnitude)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

5. Create and train the model:

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

6. Evaluate the model's performance on the testing data using metrics like:

Mean Squared Error (MSE):

```
from sklearn.metrics import mean_squared_error
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f″Mean Squared Error: {mse}″)
```

R-squared (coefficient of determination):

```
from sklearn.metrics import r2_score
r2 = r2_score(y_test, y_pred)
print(f″R-squared: {r2}″)
```

7. Use the trained model to predict earthquake magnitudes for new, unseen data:

```
new_data = pd.DataFrame([[feature1_value, feature2_value, ...]])  # Create a
DataFrame for the new data point
predicted_magnitude = model.predict(new_data)[0]
print(f″Predicted magnitude: {predicted_magnitude}″)
```

output:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

```python
data = pd.read_csv("/content/earthquake_1995-2023.csv")
```

```python
data.head()  # view the first few rows
data.describe()  # get summary statistics
data.isnull().sum()
```

- title 0
- magnitude 0
- date_time 0
- cdi 0
- mmi 0
- alert 551
- tsunami 0
- sig 0
- net 0
- nst 0
- dmin 0
- gap 0
- magType 0
- depth 0
- latitude 0
- longitude 0
- location 6
- continent 716
- country 349
- dtype: int64

```python
x = data[["latitude", "longitude"]]  # select features (predictors)
y = data["magnitude"] # Target variable (magnitude)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
model = RandomForestRegressor(n_estimators=100,
random_state=42)
model.fit(x_train, y_train)
```

- RandomForestRegressor
- `RandomForestRegressor(random_state=42)`

```
from sklearn.metrics import mean_squared_error
y_pred = model.predict(x_test)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error:_{mse}")
```

- `Mean Squared Error:_0.22899961034999994`

```
from sklearn.metrics import r2_score
r2 = r2_score(y_test, y_pred)
print(f"R-squared: {r2}")
```

- `R-squared: -0.17207443632208497`

```
new_data = pd.DataFrame([[52.7772, 158.484]])  # create a
dataframe for the new data point
predicted_magnitude = model.predict(new_data)[0]
print(f"predicted magnitude: {predicted_magnitude}")
```

- predicted magnitude: 6.64
- /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature names
- warnings.warn(

CONCLUTION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score is will be find out. This application can help to find the Prediction of Earth Quake.

FUTURE SCOPE

1. Earth Quake prediction to connect with AI model.

2. To automate this process by show the prediction result in web application or desktop application.

3. To optimize the work to implement in AI environment.

.

REFERENCE

1. Project Github link, VISHALINI P, 2024.

2. Project vedio recorded link (you tube/github), VISHALINI P, 2024.

3. Project ppt and Report github link,VISHALINI P, 2024.

GITHUB LINK FOR ALL:

https://github.com/Vishalinipalani/nm-project-earthquake-prediction/tree/main

GIT Hub link for project code:

https://github.com/Vishalinipalani/nm-project-earthquake-prediction/blob/main/earthquake_prediction.ipynb