ARTIFICIAL INTELLIGENCE -Earthquake prediction model of python

TEAM MEMBER

511821104037– N. THARUNRAJ

Phase – 3 Development Part-1

PROJECT: ***Begin building the earthquake prediction model by loading and pre-processing the dataset.***

ARTIFICIAL INTELLIGENCE -Earthquake prediction

DEPARTMENT OF COMPUTERSCIENCE and ENGINEERING

YEAR – 3rd **SEMESTER** – 5th

- **Certainly, building an earthquake prediction model is a complex task that requires access to Up-to-date and extensive datasets, as well as substantial computing resources. To get started, You'd typically follow these steps:**

1. **"Data Collection": Acquire a comprehensive earthquake dataset. You can find earthquake data from sources like the US Geological Survey (USGS) or international earthquake databases.**

2. **"Data Cleaning": Remove duplicates, handle missing values, and ensure data integrity.**

3. **"Feature Selection/Extraction": Identify relevant features like earthquake magnitude, location, depth, and time.**

4. **"Feature Engineering": Create new features if necessary, such as distance from fault lines, historical seismic activity, or geological data.**

5. **"Labelling": Define your target variable, such as predicting whether an earthquake will occur in a certain time frame.**

6. **"Data Splitting": Divide the dataset into training, validation, and test sets to evaluate the model's performance.**

7. **"Normalization/Standardization": Scale your data to ensure that all features have the same range or distribution. This can improve model training.**

8. **"Model Selection": Choose an appropriate machine learning or deep learning model for earthquake prediction. Some options include decision trees, random forests, support vector machines, or neural networks.**

9. **"Model Training": Train your selected model on the training data, adjusting hyper parameters and optimizing for performance.**

10. **"Model Evaluation": Use the validation set to assess the model's performance, considering metrics like accuracy, precision, recall, or F1-score.**

11. **"Hyper parameter tuning": Fine-tune your model by adjusting hyperparameters to improve its performance.**

12. **"Model Testing": Evaluate the final model on the test set to ensure its generalization to new unseen data.**

13. **"Deployment": If the model performs well, deploy it in a real-time or batch prediction environment.**

- **Remember that building an accurate earthquake prediction model is a complex task, and its success largely depends on the quality and quantity of data, as well as the choice of modelling techniques. Additionally, working on earthquake prediction models should be done in Collaboration with experts in seismology and geophysics, as the domain knowledge is crucial for Meaningful results and practical applications.**

DATA PRE-PROCESSING:

- **This subsection is foccused on preparing the data to give the models the best opportunity of identifying the signal in the data. The following steps will be applied:**

1. **Label Encode Target Variable**
2. **Ensure variables have the correct data types**
3. **Covert nomial variables to numeric with onehot encoding**
4. **Restrain outliers**
5. **Centre and scale all numerical variables**
6. **Remove variables with either no or minimal variance**

- **Firstly we encode the objects y_train and y_test to be value labels ranging from 0 to 4, corresponding to Grades 1 - 5.**

Program:

```
# Target Variable Transformer
preprocessor_tar = LabelEncoder()
y_train = preprocessor_tar.fit_transform(y_train)
y_test = preprocessor_tar.fit_transform(y_test)

# Visualise Counts
pd.DataFrame(y_train).value_counts(normalize = True)

# Clean objects
del preprocessor_tar
```

- **The second step is to create a pipeline of processing steps which can be applied to the predictor features. These are all combined into a single pipe which easily allows for the same processing steps to be applied to different dataframes.**

- **The print out below shows that with have significantly increased the dimensionality of our dataset as a result of one hot encoding the categorical variables. I did not venture into it this time, but it would have been sensible to apply some dimensionality reduction to the columns created by one hot encoding.**

- **Before preprocessing there were 609675 rows and 31 columns**
- **After preprocessing there are 609675 rows and 69 columns**