ARTIFICIAL INTELLIGENCE -Earthquake prediction model of python

TEAM MEMBER

511821104004– V ARAVINTH KUMAR

Phase – 5 Project Documentation & Submission

PROJECT: ***Conclude the development by building and evaluating the earthquake magnitude prediction model using a neural network. In this phase, we'll document the entire project and prepare it for submission.***

ARTIFICIAL INTELLIGENCE -Earthquake prediction

DEPARTMENT OF COMPUTERSCIENCE and ENGINEERING

YEAR – 3rd **SEMESTER** – 5th

Neural Network Model:

- **A neural network model can be employed to forecast earthquakes by examining diverse elements and trends in seismic data. This model harnesses the capabilities of neural networks, which draw inspiration from the neural connections of the human brain, to analyze intricate data and reveal hidden relationships and patterns.**

- **By training the neural network on historical earthquake data, it can acquire the ability to identify precursor signals and patterns that indicate the probability of an upcoming earthquake.**

Program:

```
from keras.models import Sequential

from keras.layers import Dense

def create_model(neurons, activation, optimizer, loss):

model = Sequential()

model.add(Dense(neurons, activation=activation, input_shape=(3,)))

model.add(Dense(neurons, activation=activation))

model.add(Dense(2, activation='softmax'))

model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

return model
```

```
from keras.wrappers.scikit_learn import KerasClassifier
model = KerasClassifier(build_fn=create_model, verbose=0)
# neurons = [16, 64, 128, 256]
neurons = [16]
# batch_size = [10, 20, 50, 100]
batch_size = [10]
epochs = [10]
# activation = ['relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear', 'exponential']
activation = ['sigmoid', 'relu']
# optimizer = ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax',
'Nadam']
optimizer = ['SGD', 'Adadelta']
loss = ['squared_hinge']
param_grid = dict(neurons=neurons, batch_size=batch_size, epochs=epochs,
activation=activation, optimizer=optimizer, loss=loss)


grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1)
grid_result = grid.fit(X_train, y_train)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
print("%f (%f) with: %r" % (mean, stdev, param))
```

Output:

```
Best: 0.957655 using {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squared_
hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.333316 (0.471398) with: {'activation': 'sigmoid', 'batch_size': 10, 'epochs': 10, 'loss':
'squared_hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.000000 (0.000000) with: {'activation': 'sigmoid', 'batch_size': 10, 'epochs': 10, 'loss':
'squared_hinge', 'neurons': 16, 'optimizer': 'Adadelta'}
0.957655 (0.029957) with: {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squ
ared_hinge', 'neurons': 16, 'optimizer': 'SGD'}
0.645111 (0.456960) with: {'activation': 'relu', 'batch_size': 10, 'epochs': 10, 'loss': 'squ
ared_hinge', 'neurons': 16, 'optimizer': 'Adadelta'}
```

What are ML datasets:

- **A machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions. The common types of data include:**
  - **Text data**
  - **Image data**
  - **Audio data**
  - **Video data**
  - **Numeric data**

Why prepare datasets for machine learning:

- **Preparing and choosing the right dataset is one of the most crucial steps in training an AI/ML model. It can be the determinant between the success and failure of the AI/ML development project.**
- **There are 3 key purposes of an AI/ML dataset:**

1. **To train the model**
2. **To measure the accuracy of the model once it is trained**
3. **To improve the model once it is deployed in a live setting.**

- **If you want to work with an AI data partner, here is out data-driven list of AI data services.**

DATA PRE-PROCESSING STEPS:

- **In this article, you will learn about data preprocessing in Machine Learning: 7 easy steps to follow.**

1. **Acquire the dataset**
2. **Import all the crucial libraries**
3. **Import the dataset**
4. **Identifying and handling the missing values**
5. **Encoding the categorical data**
6. **Splitting the dataset**
7. **Feature scaling**

Data Pre-processing Steps In Machine Learning:

- **Data cleaning, Data transformation, Data reduction, and Data integration are the major steps in data pre-processing.**

1. Data Cleaning:

Data cleaning, one of the major preprocessing steps in machine learning, locates and fixes errors or discrepancies in the data. From duplicates and outliers to missing numbers, it fixes them all. Methods like transformation, removal, and imputation help ML professionals perform data cleaning seamlessly.

2. Data Integration:

Data integration is among the major responsibilities of data preprocessing in machine learning. This process integrates (merges) information extracted from multiple sources to outline and create a single dataset. The fact that you need to handle data in multiple forms, formats, and semantics makes data integration a challenging task for many ML developers.

3. Data Transformation:

ML programmers must pay close attention to data transformation when it comes to data preprocessing steps. This process entails putting the data in a format that will allow for analysis. Normalization, standardization, and discretisation are common data transformation procedures. While standardization transforms data to have a zero mean and unit variance, normalization scales data to a common range. Continuous data is discretized into discrete categories using this technique.

4. Data Reduction:

Data reduction is the process of lowering the dataset's size while maintaining crucial information. Through the use of feature selection and feature extraction algorithms, data reduction can be accomplished. While feature extraction entails translating the data into a lower-dimensional space while keeping the crucial information, feature selection requires choosing a subset of pertinent characteristics from the dataset.

FEATURE EXTRACTION TECHNIQUES:

- **Data scientists use many feature extraction methods to tap into the value of raw data sources. Let's look at three of the most common and how they're used to extract data useful for machine learning applications.**

1. *Image processing:*

- **Feature extraction plays an important role in image processing. This technique is used to detect features in digital images such as edges, shapes, or motion. Once these are identified, the data can be processed to perform various tasks related to analyzing an image.**

2. *Bag of words:*

- **Used in natural language processing, this process extracts words from text-based sources such as web pages, documents, and social media posts and classifies them by frequency of use. The bag-of-words technique supports the technology that enables computers to understand, analyze, and generate human language.**

3. *Autoencoders:*

- **Autoencoders are a form of unsupervised learning designed to reduce the noise present in data. In autoencoding, input data is compressed, encoded, and then reconstructed as an output. This process leverages feature extraction to reduce the dimensionality of data, making it easier to focus on only the most important parts of the input.**

Innovative uses of machine learning:

- Here are the top **3 most innovative use cases of machine learning** and also areas where machine learning is thriving in recent years.

1. Healthcare:

Using **advanced analytics** through machine learning changed the way physicians are informed about the patient's medical condition. Processing data that shows patient's race, gender, socioeconomic status, family history, blood pressure readings, lab analysis test results, and latest clinical trials can result in much more useful and comprehensive information about patient's risk for stroke, kidney failure, and coronary artery disease. Also, getting this kind of results in a fraction of time with a high level of accuracy leads to increased patient satisfaction, lower cost of care and ultimately leads to a better outcome. Of course, the processes that are standardized or repetitive are more suitable for the use of machine learning than others. **Radiology**, **pathology** and **cardiology** are disciplines with large image datasets, which makes them pretty strong candidates. This way, it's possible to bring value from the application of this technology in healthcare. Introducing machine learning to daily clinical practice should be incremental so the medical workers can adjust to a new landscape along with improving their own efficiency.

2. Autonomous vehicles:

Companies have been racing to develop fully autonomous vehicles, but we can already see the benefits of using **semi-autonomous cars** developed by Tesla. Tesla Autopilot has an **advanced driver-assistance** feature that includes lane centering, adaptive cruise control, self-parking, automatic lane change and the ability to summon the car from a parking spot or garage. An excellent example of the impact **AI** is making on our everyday life. But what is behind all of these incredible features? There are multiple AI subfields that have to be put together to make a vehicle that navigates itself such as deep learning, voice search, motion detection, image recognition, processing etc. They also gather a combination of high-tech sensors and innovative algorithms in order to detect and consequently respond to their surroundings. That includes radar, laser light, GPS, drive-by-wire control systems and computer vision. All these networked components provide data for self-driving cars and the intellect for making autonomous decisions.

3. Cybersecurity:

Machine learning plays an important role in cybersecurity since most attacks come from software, not individuals; these are few and sporadic. It is why cybersecurity is looking for assistance from this innovative technology. There is too much volume of **malware attacks** for humans to handle and machine learning has the ability to sort through millions of files and identify potentially dangerous ones. It is increasingly being used to reveal threats and eliminate them before doing damage. For instance, security is being **a critical concern for self-driving cars**. The good news is that machine learning can be deployed to protect them from cyberattacks and malware. Even regular automobiles have millions of lines of code and electrical components communicating via an internal network. The first step to deploying machine learning to avoid security risks is collecting and storing the correct data. If a car's internal network uses a platform for monitoring, capable of storing and analyzing logs, the vehicle is able to detect malicious activity and prevent it. Since autonomous vehicles require a lot of processing power to make decisions based on sensory input, the second-best solution is to alert the driver, at least. Once the self-driving car is set to collect and store user logs, machine learning is called to detect anomalies in communication behaviour or unusual commands like activating parking mode while the car is on a highway. Machine learning algorithms can run 24/7, waste-free, which makes them a perfect tool for maintaining a high level of protection.