**ANN - Adaptive Difference-Based Prediction Model**

*Author:* [*Madubuike Nzubechukwu David*](mailto:david.nzube.official22@gmail.com?subject=Contact%20David%20on%20ANN)

**Table of Contents:**

**1. Introduction**

**2. Usage**

**3. Class Methods**

**3.1. load\_train\_data**

**3.2. calculate\_adaptive\_differences**

**3.3. adjust\_data**

**3.4. predict**

**3.5. calculate\_error\_margin**

**3.6. calculate\_accuracy**

**4. Example Usage**

**5. Conclusion**

**1. Introduction:**

**ANN is a Python class that implements an adaptive difference-based prediction model. It is designed to work with time-series data, specifically monthly data, to predict the most likely class (month) for a given dataset. The model calculates adaptive differences for each class (month) in the training data, adjusts the data based on these differences, and uses this adjusted data to make predictions.**

**2. Usage:**

**To use ANN, follow these steps:**

**- Create an instance of the ANN class.**

**- Load your training data (a dictionary of monthly data) using the `load\_train\_data` method.**

**- Calculate adaptive differences for each month in the training data using the `calculate\_adaptive\_differences` method.**

**- Adjust the training data based on adaptive differences using the `adjust\_data` method.**

**- Make predictions for test data using the `predict` method.**

**- Calculate error margin and accuracy for the predictions using the provided methods.**

**3. Class Methods:**

**3.1. `load\_train\_data(data: dict)`**

**- Load training data from a dictionary. The dictionary should have months as keys and lists of data as values.**

**3.2. `calculate\_adaptive\_differences()`**

**- Calculate adaptive differences for each month in the training data. These adaptive differences indicate the variability within each month.**

**3.3. `adjust\_data()`**

**- Adjust the training data based on the calculated adaptive differences. The adjusted data is used for making predictions.**

**3.4. `predict(test\_data: list) -> str`**

**- Make predictions for a given test dataset (list). The method returns the most likely class (month) for the test data.**

**3.5. `calculate\_error\_margin(predictions: list, actual: list) -> float`**

**- Calculate the error margin for the predictions. This metric measures the percentage of incorrect predictions.**

**3.6. `calculate\_accuracy(predictions: list, actual: list) -> float`**

**- Calculate the accuracy of the predictions. This metric measures the percentage of correct predictions.**

**4. Example Usage:**

**A complete example of how to use ANN is provided in the code. It demonstrates how to load training data, calculate adaptive differences, adjust the data, make predictions, and calculate error margin and accuracy.**

**5. Conclusion:**

**ANN is a powerful tool for predicting the most likely class or category for a given dataset. It is particularly well-suited for time-series data analysis and can be applied in various domains where adaptive differences play a crucial role in understanding and predicting data trends.**