

Assignment: Data Normalization in E-commerce

Objective:

Develop a Python program that takes a list of dictionaries representing features in an e-commerce dataset and normalizes the numerical values within each dictionary using appropriate techniques. The program should identify the data type (numerical or otherwise) and apply the corresponding normalization method.

Data Structure:

The input data will be provided as a list of dictionaries, where each dictionary represents a feature with the following structure:

Python

```
feature = {  
    "value": numerical_value, # Numerical value of the feature  
    "type": "numerical"      # Data type of the feature (always "numerical" in this case)  
}
```

Normalization Methods:

The program should be able to handle various normalization techniques based on the characteristics of the data:

StandardScaler: Suitable for numerical features with a Gaussian (normal) distribution. It transforms data to have a mean of 0 and a standard deviation of 1.

MinMaxScaler: Useful for numerical features with a known and bounded range. It scales data to a fixed range (typically [0, 1]).

Log Transformation: Effective for highly skewed data. It applies a logarithmic transformation ($\log(\text{value} + 1)$) to reduce skewness.

Implementation:

Define Functions:

Create a function `get_normalization_method(data_type)` that takes a data type as input and returns the appropriate normalization method (e.g., `StandardScaler`, `MinMaxScaler`, `log_transform`).

Implement separate functions for each normalization method:

`standard_scale(data)`

`min_max_scale(data)`

`log_transform(data)`

Main Loop:

Iterate through each dictionary in the input list.

For each dictionary:

Extract the numerical value using `feature["value"]`.

Call the `get_normalization_method` function to determine the suitable method based on the data type.

Apply the chosen normalization method to the value using the corresponding function (e.g., `standard_scale(value)`).

Update the dictionary's value with the normalized value.

Return Normalized Data:

After processing all dictionaries, return the modified list containing the normalized feature values.

Example Usage:

Python

```
input_data = [  
    {"value": 123456, "type": "numerical"},  
    {"value": 500, "type": "numerical"},  
    {"value": "Some Text", "type": "text"} # Non-numerical data ignored  
]
```

```
normalized_data = normalize_data(input_data)
```

```
print(normalized_data)
```

Considerations:

You may need to import additional libraries like `sklearn.preprocessing` for standard and min-max scalers.

Handle cases where the data type is not "numerical" gracefully (e.g., ignore the dictionary or raise an error).

You can extend the program to include more normalization techniques as needed.

This assignment provides a framework for normalizing numerical data in a list of dictionaries based on data types.