**Task-2:** **Data Normalization Techniques and Mean-Centering with**

**Z-Score**

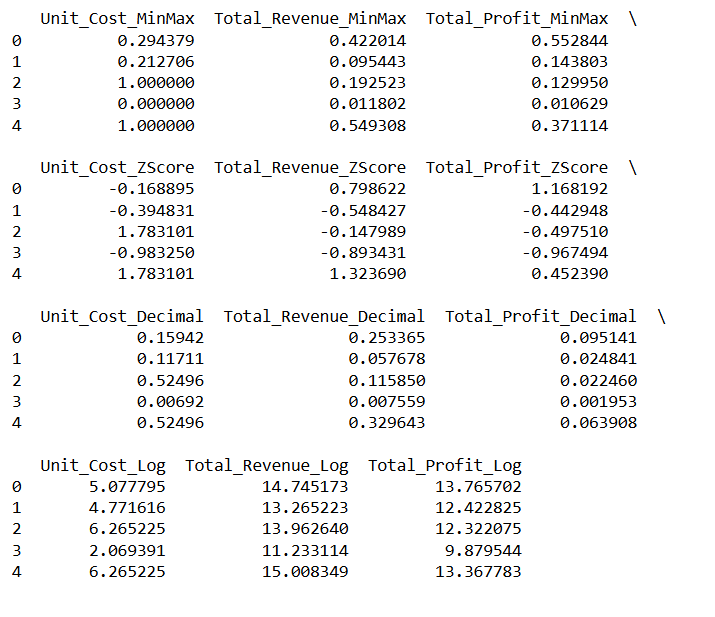
1. **Description**

(Task for normalization and z-score)

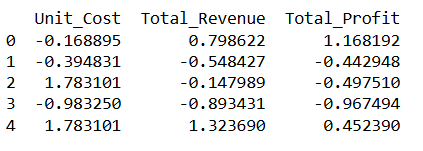
A: Find 10000 row dataset and apply all type of normalization in dataset.   
B: Apply z-score using sklearn library and do mean-centering of Sales dataset.

Dataset: <https://www.kaggle.com/datasets/nishathakkar/100-sales>

1. **Output**

****

**Normalization**

****

**Z-Score**

1. **Algorithm Used in Task**

This code demonstrates four data normalization techniques (Min-Max, Z-Score, Decimal Scaling, Log Transformation) to preprocess numerical data. Each technique adjusts data scales to enhance model performance and data comparability. The final DataFrame combines these normalized results, providing multiple perspectives on the dataset's features. By applying these transformations, this code ensures the dataset is ready for analysis or machine learning tasks.

**Explanation of Libraries and Their Usage:**

1. **Pandas (import pandas as pd)**
   * Used for data manipulation and analysis.
   * Reads the dataset into a DataFrame and selects specific columns for processing.
2. **NumPy (import numpy as np)**
   * Provides mathematical functions for advanced calculations.
   * Used for operations like log transformation and decimal scaling.
3. **Scikit-learn (from sklearn.preprocessing import MinMaxScaler, StandardScaler)**
   * Provides tools for preprocessing and machine learning tasks.
   * **Key Functions Used:**
     + MinMaxScaler: Scales data to a range (0, 1).
     + StandardScaler: Applies Z-score normalization to standardize the data.

**Approach**

1. Load the dataset from a CSV file using Pandas.
2. Select numerical columns for normalization and transformation.
3. Perform the following normalization techniques:
   * **Min-Max Scaling:** Scales data to a range of [0, 1].
   * **Z-Score Normalization:** Standardizes data to have mean 0 and standard deviation 1.
   * **Decimal Scaling:** Scales values by a power of 10, determined by the maximum value in the dataset.
   * **Log Transformation:** Applies logarithmic scaling to reduce skewness.
4. Combine results into a new DataFrame for easy comparison.
5. Display the normalized results.

**Algorithm:**

**Normalization and Transformation:**

1. Import necessary libraries (pandas, numpy, sklearn.preprocessing).
2. Load the dataset from the specified file path.
3. Select the numerical columns for transformation.
4. Apply each normalization method:
   * **Min-Max Scaling:**
     + Use MinMaxScaler().fit\_transform().
   * **Z-Score Normalization:**
     + Use StandardScaler().fit\_transform().
   * **Decimal Scaling:**
     + Divide each value.
   * **Log Transformation:**
     + Apply np.log1p() for log scaling.
5. Store the results in a new DataFrame.
6. Display the normalized DataFrame.

**Z-Score Normalization with Mean-Centering:**

1. Select the numerical columns.
2. Apply Z-score normalization using StandardScaler().fit\_transform().
3. Convert the standardized array into a DataFrame.
4. Display the mean-centered normalized data.