IT166 Lab 8

In this lab, we will practice data scaling (one type of data transformtion) using List comprehensions.

2:49

Objectives

- Be able to write Python functions to implement different data scaling.
- Be able to use List Comprehensions to apply the scaling functions to the data in a list

Preparation

- Launch the Jupyter notebook.
- Rename the notebook page as “Lab8”

Background:

Data transformation is the process of converting raw data into a format or structure that would be more suitable for (statistical or ML) model building and also data discovery in general. It is an imperative step in feature engineering that facilitates discovering insights.

Scaling Transformation

Scaling is often used before fedding data to any ML models. This is because data often consists of many different input variables or features (columns) and each may have a different range of values or units of measure, such as feet, miles, kilograms, dollars, etc.

If there are input variables that have very large values relative to the other input variables, these large values can dominate or skew some machine learning algorithms. The result is that the algorithms pay most of their attention to the large values and ignore the variables with smaller values.

As such, it is normal to scale input variables to a common range as a data preparation technique prior to fitting a model. There are many data scaling methods, including Min-Max-Scaler (aka normalization), Standard Scaler (aka standardization), and Robust Scaler(aka clipping).

In this lab, we will develop three data scaling functions as described below.

For all problems below, use List Comprehensions to do the data transformation (scaling).

You can use any functions (e.g., minMaxList, etc.) developed in the last lab.

Problem 1: Develop a MinMaxScaler() function to conduct data normalization, which normalizes the data (in a List) into a range between 0 and 1 based on:
$x' = (x - min(x))/(max(x) - min(x))$

Problem 2: Develop a stdScaler() function to conduct data standardization, which converts the data (in a List) into the standard form of $mean = 0$ and $variance = 1$ based on:
$x' = (x - mean(x))/stdev$

Problem 3: Develop a dataClipping() function to conduct data clipping in order to handle outliers. Clipping method sets up the upper and lower bound, and all data points will be contained within the range. We can use quantile() to find out what is the range of the majority amount of data (between 0.05 or 5th percentile, and 0.95 or 95th percentile). Any numbers below the lower bound (defined by 5th percentile) will be rounded up to the lower bound. Similarly, the numbers above upper bound (defined by 95th percentile) will be rounded down to upper bound.

For example, the 95th percentile is the highest value remaining after the top 5% of a data set is removed.

Here are the steps to calculate nth percentile of a given list:

- Sort the list;
- $index = round(len(list) \times n/100) - 1$
- $the\ nth\ percentile = list[index]$

You also need to develop a function called clipping(itemValue, low, high), which can be applied to each item in the list such that:

- if itemValue < low (outlier), rerun low;
- if itemValue > high (outlier), rerun high;
- otherwise, rerun itemValue;