Ta’Quawn Watts

Plotting, Salting, and Smoothing Data in Java

For Project 2 of CSCI-3327 Probability and Applied Statistics course at Stockton University, my task was to create a program that produce various CSV files. The program should be able to create a CSV file containing the input and output of a specific function. The program should be able to customize the range of input values of the function.

After creating this file, the program should be able to take a CSV file of input and output values and creates another CSV file with “salted” output values. Salting, or adding noise, to data means to add randomness to input data. Salting data is very common with areas like machine learning and neural networks as it benefits neural networks and models performance with variations in data. In this case, the method adds and subtracts numbers within a customizable range randomly to the output values.

In addition, the program should also be able to take a CSV file of input and output values and creates another CSV file with “smoothed” output values. Smoothing reduces the randomness from data set typically by using an algorithm. Smoothed data is usually used to find and predict trends within the data set. In this case, the method takes each output value and averages it with a specified number of y values around it. This range is known as the window size.

The function that I chose to use was . I chose this function because it was a parabola. I was interested to see how drastically the salted graph would change and how accurate the smoothed graph would be to the original. The functionSolver method has a parameter of integer x representing the variable x in the function and returns the output of .

The writeToCSV method creates a CSV file titled “data.csv”. This file will contain the input and output values of our function. writeToCSV takes two parameters: integer min and integer max. These parameters represents the range of input values input in the function. Using functionSolver, writeToCSV writes the input and output values to data.csv and prints the path to data.csv. Using the range of -25 and 25, writeToCSV created a file with values that produced this graph:

The saltCSV method creates a file titled “saltedData.csv”. This file will contain the input and salted output values of the input CSV file. saltCSV contains three parameters: String filePath, integer min, and integer max. The filePath parameter is the file path to the CSV file that will get salted values. The min and max parameter is the range of the randomized numbers to be added to the output values. saltCSV uses filePath and copies the input and output values to saltedData.csv. saltCSV then generates a random positive or negative number and adds or subtracts it to each y value. It replaces the original y value with the new value and writes it to saltedData.csv. After finishing this process, it prints out the path of saltedData.csv. Using the range of 5 and 100, saltCSV created a file with values that produced this graph:

Notice that, instead of the smooth curves of the parabola, the salted graph has jagged unpredictable lines. It looks significantly different than the original function’s graph.

The smoothData method takes each element of a list and averages it with a specified range of elements around it. smoothData takes two parameters: Double type arraylist data and integer windowValue. The data parameter is the list of numbers being smoothed. The windowValue is the number of points being taken right and left of the index point. This method goes through the data arraylist and takes the average of the values around the initial value. It then adds each average to the smoothedData arraylist. After finishing the process for each element in the arraylist, it returns the smoothedData arraylist.

The smoothCSV creates a file titled “smoothedData.csv”. This file will contain the input and smoothed output of the input CSV file. smoothCSV takes two parameters: String filePath and integer windowValue. The filePath parameter is the file path to the CSV file that will get smoothed values. The windowValue is the number of points being taken right and left of the index point. After copying the input and output values of the input CSV file, it saves the output values to an arraylist. Using smoothData, smoothCSV replaces the original output with the smoothed outputs. Using the windowValue of 5, smoothCSV produces this graph:

Notice that the smoothed graph is very similar, but not exact, to the function’s graph. However, the more a data is smoothed, the straighter the graph will be. This is what the graph will look like if you smoothed the data 10 times:

Notice that the smoothed graph begins to stop resembling the original function’s graph and more resemble a straight line. Smoothing data removes outliers from the dataset. Repeating the process removes more outliers, creating a straighter trend.

Works Cited

Dhir, R. (2022, November 20). *Data smoothing: Definition, uses, and methods*. Investopedia. Retrieved April 24, 2023, from https://www.investopedia.com/terms/d/data-smoothing.asp

TutorMaster, A. I. (2023, February 2). *What is gaussian noise in deep learning? how and why it is used?* Medium. Retrieved April 24, 2023, from https://ai.plainenglish.io/what-is-gaussian-noise-in-deep-learning-how-and-why-it-is-used-af3730449e3a