DS2001 - CS Practicum

Spring 2023

Practicum 5 – Data Analysis

Practicum is DUE at the end of your scheduled practicum.

We'll be working in pairs, and today you can choose your own partner instead of being assigned. You may not finish every problem every week, but we expect your best effort.

We will let you know when there are 15 minutes left in class – at that point, it is time to wrap up your code and move on to the self-reflection portion of the assignment! Read the self-reflection questions below, write your answers, and save them in a PDF.

Submit both your code and your self-reflection PDF on Gradescope.

Grading Policy
You will receive full credit if you submit your self-reflection (with thoughtful answers) and any code by the deadline. All questions in the self-reflection must be answered for full credit.

Feedback
We will provide feedback on your code if requested in your self-reflection. Otherwise, we will not look at your code. Our feedback will mirror the expectations of your DS200 homeworks, and we will use the same criteria for grading your projects in DS2001. So, make sure you ask for feedback when helpful, and read our notes!

Today's Goals

- Open and read data from a file into a 2d list
- Find nearest neighbors

Working in Pairs

Each pair will work at one computer, and everyone contributes. Here's how we split up the work:

- Navigator: Dictates the code to be written. Explains the *why* as we go. Checks for syntax errors.
- Driver: Writes the code. Listens closely to the navigator. Asks questions whenever there is a lack of clarity.
- Both driver and navigator are responsible for contributing to the work, and for making space for the other person to make contributions.

Programming Assignment

1. **Gather Data**

   Download cereal.csv from the course website and save it in the same folder as the python file you will write today. This datafile contains nutrition facts panel information for 80 different cereals. It was obtained from this website: https://www.kaggle.com/datasets/crawford/80-cereals?resource=download.

   The data contained in cereal.csv is:

   ```
   name (cereal name)
   mfr (manufacturer)
   type (hot or cold)
   calories
   protein
   fat
   sodium
   fiber
   carbo (carbohydrates)
   sugars
   potass (potassium)
   vitamins (% daily value)
   shelf (display shelf)
   weight (oz per recommended serving)
   cups (cups per recommended serving)
   rating
   ```

   Define a function with a parameter for csv name. The function should read in the csv passed to it when called and return a 2d list of data. Call your function to read in cereal.csv. Remember to save the result as a variable.

2. Computations

Your goal today is to find cereals most similar to Ab's favorite cereal:

name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
Cracklin' Oat Bran	K	C	110	3	3	140	4	10	7	160	25	3	1	0.5	40.448772

Subsetting Data

To start, define a function named `subset_data`. This function should have parameters for a dataset represented as a 2d list, and three column indices. The function should return a subset of the dataset (as a 2d list) with only the three specified columns.

For example, say I have the following dataset:

```
headers = ["animal", "legs", "wings", "fins", "coolness"]
dataset = [["dog",  4, 0, 0, 99],
           ["cat",  4, 0, 0, 100],
           ["fish", 0, 0, 2, 90],
           ["bird", 2, 2, 0, 10]]
```

If I call `subset_data(dataset, 0, 1, 4)` the following should be returned:

```
[["dog",  4, 99],
 ["cat",  4, 100],
 ["fish", 0, 90],
 ["bird", 2, 10]]
```

Calculating Similarity

There are many ways to calculate similarity between two points (i.e. rows) in a dataset. For this assignment, we will define similarity as Euclidean distance. In other words, we will say the similarity between point Z and point W is equal to the Euclidean distance between them. The shorter the Euclidean distance, the more similar the points are.

Here is the formula for calculating Euclidean distance between 2d points:

Let d be the distance between point A (a, a_2) and point B (b_1, b_2)

$$d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Define a function named `similarity`. This function should take as parameters two points and should return the Euclidean distance between the points. For example, calling `similarity(a_1, a_2, b_1, b_2)` should return the Euclidean distance between the points (a_1, a_2) and (b_1, b_2).

Finding Most Similar

Define a function named `find_closest`. This function should take as parameters (1) Ab's favorite cereal, subset to 3 columns, and (2) the cereals dataset, subset to 3 columns. The function should find the cereal in the cereals dataset most similar to Ab's favorite cereal and return its name.

For example, say we subset to look at protein and fat:

```
abs_fave = [["Cracklin' Oat Bran", 3, 3]

subset_cereal = [["100% Bran", 4, 1],
                 ["100% Natural Bran", 3, 5],
                  …]
```

If I call `find_closest(abs_fave, subset_cereal)` 100% Bran should be returned.

3. Communication

In `main`, use the functions you found above to report to the user:

- Which cereal is most similar to Ab's favorite in terms of sodium and fiber
- Which cereal is most similar to Ab's favorite in terms of carbohydrates and rating
- Which cereal is most similar to Ab's favorite in terms of protein and vitamins
- Which cereal is most similar to Ab's favorite in terms of fiber and cups

At this point, please complete the self-reflection! If you still have time afterwards, scroll down for a little extra/bonus work you can do on this assignment.

Self-Reflection

Create a PDF with answers to the following questions (each should be answered collaboratively with your partner unless instructed otherwise):

- Does this work reflect the best effort of both partners?
- In what way(s) did each partner (1) make contributions to the assignment, and (2) give the other person space to make contributions? (Each partner should respond to this question separately.)
- Which problem(s) would you like feedback on and why?

Keep Going (if time!)

Here are some ways you can improve and build on what we've done so far. Try any/all of them in the time we have left.

- Expand your subset and similarity functions so that you can find the most similar cereal to Ab's favorite based on 3, 4, and 5 columns instead of only two.