CMPU–101

Problem Solving and Abstraction

Fall 2021
Assignment 4

Setup and Hand-in

- Copy this file to your `code.pyret.org` folder: `asmt4-code.arr` You can click on the link, open it with Pyret, and save it. This is where you will write the code for this assignment.
- Remember your resources:
  - Table Documentation
  - Campuswire
  - Coaches
  - Your instructor
  - Code Clarity Guide
- Do not change the file name or the names of any of the functions.
- Remember to test your functions thoroughly and to write your code clearly. Your work will be graded on clarity and thorough testing, as well as on whether it works correctly.
- When you are ready, download your `asmt4-code.arr` file and then go to Gradescope to hand it in.

Introduction

We can use tables to store many kinds of data, such as the candy survey responses we worked with in lab or the solar data used on Assignment 3. In other settings, we might want to use a Table to gather observations from an experiment and use data about previous observations to guide future decisions (this is the essence of *machine learning*).

While this assignment uses very simple notions of what machine learning does, it is enough to give you the core idea: we have data on what has happened in the past, and we use statistics about it to make decisions about new situations. If you take a machine learning class, you will learn how to handle more complex notions of similarity between old and new cases, and more nuanced algorithms for deciding what to do based on the data. Still, you have already learned enough in CMPU-101 to see the basic components of such algorithms.

The medical workers in a town where malaria risk is high have a limited budget and would like to give bed-nets, which help prevent malaria, to the people most at risk of getting malaria. They would like to use machine learning to help them decide to whom they should give the nets. They have access to data about people in the town, including the distance of their home from a lake (a Number), their age (a Number) and whether they are pregnant (a Boolean). Living close to a lake is likely to increase malaria risk; children and pregnant people both have less immunity to malaria and are therefore more susceptible.

The medical workers also have data about people from a similar town, including whether those people contracted malaria. In `asmt4-code.arr`, this data is loaded into a Table called `MALARIA-DATA` defined at the top of the file, which includes a column indicating whether each individual contracted malaria. Each person in this table is between 0 and 80 years old lives between 5 and 300 yards from a lake.

In this assignment, you'll implement two techniques for learning from the data from a similar town, in order to help decide who should get nets in the town we're trying to help.

Part 1

In the first part, we will use a machine learning technique called *clustering* where we group similar data points together and predict properties of others that are "close" to those points. Our task plan is to first write a function that determines whether a row in the table is similar to a given person *p* and to use that to find similar people. Next we'll look at the number of similar people who got malaria and the number that did not and will use that to decide whether to give person *p* a net.

Task 1A

Fill in the function `similar-people`, which takes in the distance to lake, age, and pregnancy status for a person *p* and a Table of data like `MALARIA-DATA`. It produces a Table of the individuals who are *similar* to the person *p* we are checking in this sense:

- distance from lake is within 20 yards of *p*'s distance to lake,
- age is within 5 years of *p*'s age, and
- pregnancy status the same as *p*'s status.

You should create smaller tables with a similar format to `MALARIA-DATA` to test this function.

Hint: Write a nested function or a lambda function that returns `true` if and only if a row `r` of table `t` describes a person who is similar to the person whose data is input to `similar-people`. The `similar-people` function should call `filter-with` with this function as one of the parameters.

Task 1B

Fill in the function `give-net`, which takes in a table (similar to the result of `similar-people` and decides based on this past data whether to give people like those in the table a net.

Specifically, **if the number of people in Table *t* who got malaria is at least as large as the number who did not, return `true`. Otherwise, return `false`.** If there are no similar people, we don't have enough information, so we don't want to risk it: return true and give the individual a net. When `give-net` returns `true` it indicates that people who are similar to the person whose data is passed in were likely to get malaria, so we should give them a net.

Hint: One way to count the number of people who got malaria is to filter the table to find those people and use the built-in `length` function to see how many rows that filtered table has. Feel free to declare some names within `give-net` to help keep track of the results of steps in `give-net`'s computation, for example,

```
malaria-count = ...
```

Part 2

In the previous tasks, we used a *clustering* where we group similar data points together and predict properties of others that are "close" to those points. Some other machine learning techniques try to draw a boundary that does a reasonably good (but not necessarily perfect) job of separating one class of data points (e.g., people who got malaria) from another class (e.g., people who did not get malaria). In this part of the assignment, we will focus on non-pregnant adults and try to find a distance d such that most people who got malaria live less than or equal to d meters from the lake and most who did not get malaria live more than d meters from the lake. We can then decide whether a non-pregnant adult should get a net by comparing their distance from the lake to d.

In this part, we will use the `NON-PREG-ADULT-DATA` table, which was obtained by filtering the original `MALARIA-DATA`.

Our task plan is as follows:

- find the average distance from the lake of people who got malaria
- find the average distance from the lake of people who did not get malaria
- use these to compute a threshold that we'll use to decide whether to give someone a net
- write a function that will return true if a person lives close to the lake (relative to the threshold) and returns false otherwise

Task 2A

Fill in the definitions in the `asmt4-code.arr` with code to compute

- A table of non-pregnant-adults who got malaria
- A list of distances of those people from the lake
- The average of those distances. (A function `avg` to compute the average of numbers on a non-empty list of numbers is provided in `asmt4-code.arr`.)

- A table of non-pregnant-adults who did not get malaria
- A list of distances of those people from the lake
- The average of those distances
- `threshold` – the distance the machine learning algorithm will use to determine whether to give someone a net. This number should be the average of the two averages.

Task 2B

Write a function `give-net-2` that gives a net to a person who is less than `ADULT-AGE` years old (a constant defined near the top of the code file) or who is pregnant or who lives "close" to the lake (as determined by the threshold that your program "learned" from the data).

Part 3

To see how well a machine learning algorithm performs, data scientists use one set of data to *train* the algorithm and another set of data to *test* to see if the algorithm make good predictions. `MALARIA-DATA` and `NON-PREG-ADULT-DATA` were training data in parts 1 and 2.

We've provided another small set of *test data* in the second tab of the spreadsheet.

Task 3A

Use `give-net-2` and the data in the first three columns to decide whether each of the first five non-pregnant adults in the test data set should get a net.

Now compare those results to column 4. For which of these people did `give-net-2` make the correct decision (giving them a net if they got malaria; not giving them a net if they didn't get malaria)?

What would be the impact of increasing the threshold on

1. the level of protection provided to people in the town?
2. the number of nets distributed, and hence the cost of the program?

OPTIONAL Task 3B

If you're interested in learning more and doing some more coding, you may complete this part. It is **not** for extra credit and we will not grade this part, but you're welcome to ask questions about it. Find out what *precision* and *recall* mean. Write functions to compute the precision and recall of `give-net` and/or `give-net-2` as follows:

- create a table from the test data
- transform it to create a column(s) giving the results of `give-net` and/or `give-net-2`

- add columns determining which rows had the correct prediction, which are false positives, and which are false negatives.
- use this to compute precision and recall of `give-net` and/or `give-net2`