ArrayLists and Data

Haverford CS 106 - Introduction to Data Structures

Lab 2 (one week)

Make sure to complete the pre-lab exercises before beginning this lab. If you are starting early (i.e. before lab on Feb 14) then you do not need to check in with an instructor.

Also make sure to **TAG** your Lab 1 repository before starting Lab 2, since they are in the same repository. There is a short video accompanying this lab about how to tag your repo.

1 Finishing Your Data Storage

In the previous lab you created a class to store a single row of the dataset. Now, you should add a class to store the full dataset:

1. Create a class (separate file in the same package) that will hold the full CSV of data, using within it the class you made to store rows. Think carefully about what to name it, based on your understanding of the data, so that it correctly indicates what data is being stored.
2. Make a field in your dataset class that holds an `ArrayList` of rows.
3. Thinking carefully about names, add any constructors, getters, and setters and be sure to add comments and Javadocs.

2 Reading in Data

OpenCSV is a library that allows you to read in data. You can read in data from "compas-scores.csv" using the below code. You'll also need to add the appropriate imports - Eclipse can help you find them. I would recommend using try/catch to deal with any issues reading in the file.

```
CSVReaderHeaderAware reader = new CSVReaderHeaderAware(
                                    new FileReader("compas-scores.csv"));
ArrayList<String[]> myEntries = new ArrayList<String[]>(reader.readAll());
reader.close();
```

This will give you an `ArrayList` containing `String` arrays, where each `String` array is a row from the CSV, and each entry in the `String` array is one cell from that row.

To read in the data from the CSV:

1. In the `main` method of your Main class, read the data from the ArrayList of String arrays (myEntries) into the data structure you developed earlier in this lab. Think carefully about how to add to your data structure design to do this.

2. Catch any thrown exceptions due to invalid data entries so that the data continues to be read in, skipping any rows with errors.

Answer the following questions in your **README.md** file (add a section below your name and above the questions about difficulty/timing) to the main method:

1. If any of your validation methods indicate that your precondition assumptions were not met by the data, document what preconditions were violated and in what way. Update your validation methods one error at a time, documenting all issues, until you can read in all the data. If this did not happen, mention that.

2. Describe a person who would generate data that would *not* pass your (updated) precondition assumptions.

3 Analysis

Now that you have the data read into your data structure we can reproduce the analysis ProPublica shows in their chart, "Prediction Fails Differently for Black Defendants."

1. Add, modify, and/or implement any methods that you need to replicate the ProPublica analysis that were missing from your previous lab. This should include method(s) added to your dataset class that perform the full analysis.

2. Have your main method call the relevant methods to calculate and print out the data from the chart. You should be able to check your work by making sure it matches the numbers in the ProPublica chart. Make sure you can run the `VerifyFormat_Lab2.java` file and produce the same table (+/- 0.1 percent is okay). Note that a small change is needed to this file (see Piazza for more info). Please change the number 44.7 to 47.7 in the `VerifyFormat_Lab2.java` file.

3. Find charges that, based on the `r_charge_desc`, you think should not count as recidivism. Change your analysis using the `two_year_recid` outcome indicator accordingly and rerun the analysis. Do this using an optionally run method so that you can still run the previous version of the analysis (your final submission should run only the original analysis so it can be checked by the autograder). Describe the choices you made and what the resulting analysis shows in your **README.md**.