COS126: Final Project Report Template

Fall 2022

Instructions: Make a copy of this Google doc and respond to all required questions. Then, download this Google doc as a PDF and change its name to `report.pdf`. Finally, upload `report.pdf` to TigerFile.

Note: Unlike the proposal template, this template is just a suggestion of the minimum information you should include. You can and should feel free to modify this template to suit the needs of your project, for example, by adding sections or questions.

1. Basic Information

Name #1: Matthew Banaag
NetID #1: mb6611

Name #2: Xuan Lin
NetID #2: xl2524

Project Advisor: Prachi Sinha
Link to YouTube video presentation: https://youtu.be/CF74HVQXvAI

2. Project/Course Feedback

Before moving on*, fill up the following form (https://forms.gle/MSQC4cswCJ3PBszY8) to provide feedback on the project and on the course.*

We filled up the feedback form: (X)

3. Project Description

What is your project title? Bridge

Please summarize your project in 1 paragraph. Only describe things you implemented.

It is a program that creates teams from individuals based on their strengths in hard skills and soft skills. It first grades each individual based on how many keywords of the respective categories appear in their resume (Individuals Class). It then uses the KNN algorithm to return the predicted label and sort individuals using merge sort based on their distances to others.

Finally, the TeamMatcher class classifies individuals using the KNN algorithm and adds them to the correct categories of high, medium, low skill. It then generates random teams that are balanced using our RandomNumber RNG class.

Please list all the relevant files (.java files, datasets, .jar libraries) from your project. Include a brief explanation of their contents.

1. Individual.java
   a. The ADT for members that includes their Soft and Hard skill score, team contribution calculated based on # of keyword occurrences in resume
2. Knn.java
   a. The implementation of the KNN algorithm that then sorts the individuals based on their distance ((x,y) = (hard skills score, soft skills score))
3. Merge.java
   a. Merge sort algorithm to sort individuals by their Euclidean distances to one individual
4. RandomNumber.java
   a. Generates random number based on the Wichmann-Hill pseudorandom number generator
5. TeamMatcher.java
   a. Classifies individuals into high, medium, and low skill levels and then randomly generates balanced teams with a Wichmann-Hill pseudorandom number generator
6. Gui.java
   a. The GUI for the project. It includes a top section that allows input of the files, middle section that displays the roster and generated teams, and bottom section that shows statistics.

Describe instructions on how to compile and run your project (e.g. compilation commands).

Using the COS 126 starting file, compile the Gui.java file using normal procedure (javac-introcs Gui.java then java-introcs Gui). Enter the files in the exact order (case sensitive): the training file (e.g. training.txt), testing file (e.g. testing.txt) and integer value for team size (e.g. 7). With regards to the final input, the number must be evenly divisible by the total number of members.

4. Features and Project Requirements

For each of the following, mention the .java files, classes, and methods that implement the things you are describing. If you used any external libraries, mention those as well.

Feature #1:

In 2-3 sentences, please describe your feature. Only describe things you implemented.
This feature implements the K Nearest Neighbor algorithm to prepare a group of individuals to be grouped into ideal teams. The algorithm clusters the individuals into three classifications (high skill, medium skill, low skill) based on their calculated worth from their hard and soft skill metrics, and prepares individuals to be sorted into teams in the TeamMatcher class.

In 1-2 sentences, please describe how you tested your feature.
We tested our feature by passing in files for different individuals which we had already computed values for, and checking that the algorithm sorted individuals by their Euclidean distance from another individual. We assigned all of the resumes in our database a label, and verified that the predict method consistently produced the same output as the label.

What was the agreed category of your feature using the buckets paradigm?
( ) Standard
( ) Sprinkle
(x) Sparkle

Feature #2:

In 2-3 sentences, please describe your feature. Only describe things you implemented.
It will form random teams within a certain tolerance (for the team contribution score). The program will then select the ideal team based on a certain tolerance for the team balance score. It places classified individuals into skill levels and then randomly selects from the skill levels to form teams of even compositions of different skill levels.

In 1-2 sentences, please describe how you tested your feature.
It prints out the elements in the individuals arraylist to see if it was populated correctly and then we compare the individuals that were sorted to hand-calculated values to see if they matched up. To test the teams, we manually checked the overall stats of the team and compared it to see if they fall within the tolerance range.

What was the agreed category of your feature using the buckets paradigm?
( ) Standard
( ) Sprinkle
(X) Sparkle

Feature #3:

In 2-3 sentences, please describe your feature. Only describe things you implemented.
The GUI consists of three main areas: the area for input, the area that displays the roster and built teams, and a dropdown menu that contains the stats and information of individual members. The text field allows for three inputs before the button is disabled to ensure no extra inputs are provided. The roster and team display lists the members in a numbered list and the team stat-viewer has a drop-down menu with every team member after they have been placed into a team and has a text area at the bottom that displays their stats.

In 1-2 sentences, please describe how you tested your feature.
We entered in three files and manually tested if the button became disabled, we compared the text area for both the roster, teams, and drop down to the total number of team members and manually tested if the numbers matched up. For the stats themselves, we compared them to our hand calculated values.

What was the agreed category of your feature using the buckets paradigm?
( ) Standard
( ) Sprinkle
(X) Sparkle

Other Project Information:

What would you try to add to your project if you had more time to implement extra features?
A graphical representation of each team member stats, much like the chart provided in video games. We would also add a menu bar that allows for selecting all three files at once along with various quality of life improvements such as allowing for hotkeys.

What did you learn by working on this project?
We learned how to implement the KNN algorithm, and how to build GUIs using Java Swing along with coding problem-solving skills.

What advice would you give to students next year, to help them have a better experience working on their projects?
Please don’t work solo, and make sure you use YouTube to learn new stuff!

5. Extra credit

You should use this section to mention any of the extra credit items you did. Please provide justification for each one (for example, for the version control item you should have a link to your project on a website like GitHub)

1. *Having a project partner: Matthew and Xuan*
2. *Order of growth analysis:*
3. *Creating your own dataset*