Project Proposal

Name	Claire (Yumeng) Luo	Sang Jun Chun
UNI	yl4655	sc4658
Team Name	Imposters2	

Topic: Code Similarity Detection

Description:
Code similarity detection is used in plagiarism check, defect detection, etc. In this project, we are interested in utilizing code similarity detection in a particular use case: Detecting plagiarism in school projects.
We plan to use source code similarity detection tools, binary code similarity detection tools and text diff tools to detect plagiarism in the 3 individual coding assignments in COMS4156 and evaluate the performances of these tools.

Plan:

Test data:
With the help of Shrish, we were able to find real school coding assignment data for COMS 4156 in the following github link: https://github.com/jxm033f/4156-PublicAssignment/network/members. Although not all students actually pushed commits to their public repositories, we were able to find some with meaningful commits, and believe those would be enough for the purpose of this project.

We will be using these students' submissions as our test data. One issue to resolve is that students were given a code skeleton and were asked to implement the same features, such as endpoints. Therefore ~~the code skeleton's~~ such effects should be eliminated when evaluating the similarity across different code segments. Another issue is that the assignments are generally large in size and we need to decide on which specific sections to focus on when detecting similarities and how to merge them to form a final similarity score.

To address both of these issues, we will first be removing the skeleton code and extract only the sections coded by students, and merge them into a single file for each student's work. Then, to obtain the *baseline* similarity score expected among non-plagiarized submissions, we will compare our own submissions, after accounting for the fact that we did not collaborate and received full credit, to obtain a similarity score that we can reasonably expect from other students' submissions. Other possible bias may still remain, such as the prescribed API names, though we will leave it there for the time being, to assess the initial behavior of code similarity tools upon encountering the same names. A possible follow-up is to examine if the effectiveness remains the same if we change the variable names that are uniform across the students' submissions.

Experiment:
Some processing is needed to transform the github submissions into input files that are executable for similarity detection tools.
Similarity labels for each code pair need to be generated manually. Luckily, both students in this team have taken COMS 4156 in Fall and have some knowledge of the expected behaviour of the assignments.

We plan to run 3 sets of similarity detection on the test data: one for source code similarity detection tools only, one for binary code similarity detection tools only and one for source code tools + binary code tools combined.

Our current concern is that source code detection tools will always outperform binary code detection tools and combining both tools will always just output the results of source code tools. Therefore a preliminary test is needed to compare the results of these 2 types of tools on a smaller test set.

For the preliminary test, we already have data for binary code detection tools from Claire's midterm paper. The work can be simplified by repurposing this experiment with the same test inputs but with source code detection tools.

Due to the limitations of computing resources and time (and ability), we plan to select 2 tools for source code detection tools and 2 tools for binary code detection tools (SAFE, Gemini). Because neither students studied source code detection tools before, we are currently in the process of deciding which source code detection tools to use.

We plan to compare the results with the manual label for accuracy and analyze the incorrect pairs for each tool and suggest the best approach in detecting code plagiarisms in school.

Relation to midterm paper:
This project is related to Claire's midterm paper in the way that Claire's paper is also performing plagiarism detection using current binary code similarity detection tools on school projects. The differences include:

Test data: Claire's paper used leetcode answers to approximate school assignments while this project will use real school coding assignment data

Tools used: Claire's paper only compared binary code similarity detection tools but this paper will also evaluate source code similarity tools

Jun: This project is unrelated to my midterm paper; however, I am particularly drawn to this topic as I am interested in assessing the effectiveness of recent code similarity detection tools, and in a more applicable sense, whether it can effectively check for plagiarism or not for advanced

school assignments such as COMS4156 individual projects. Through this project, I hope to learn which code similarity detection approach is more effective than others, and drawbacks of this technology that could be improved.