Dorm Match: The Solution to Finding a College Roommate

Richie Fleming

5/6/2024

Table of Contents:

1.) Introduction:

My experience as a college freshman was often great but sometimes overwhelming. While it was exciting to meet new people and explore new things, it was tough to deal with being paired with a roommate I was not compatible with. For me, my room was a place to retreat, gather my thoughts, and find peace and quiet. However, this was not always possible due to my roommate situation.

I chose to have a random roommate my freshman year because it was a hassle to find and request a compatible roommate. This process typically involves navigating through social media, specifically Facebook groups. Currently, there is no school-endorsed, easy way to find a compatible roommate, leaving students to either take their chances with a random assignment or go through a cumbersome process on their own.

The goal of this project is to address this issue by employing various methods, including machine learning, linear algebra, and problem-solving techniques. By making roommate matching as easy and accessible as possible for all prospective freshmen at a university, I aim to improve the overall freshman experience and help students find compatible roommates with ease.

2.) Objectives:

The implementation of my solution to the problem proposed above is an app that will behave similar to a dating app. Users will be able to swipe through a carousel of other freshmen that are also looking for roommates.

The interface is meant to be easy to use and engaging for the user. Users will be able to connect their Snapchat account, using the snapkit library, in order to use their bitmoji for their profile picture. User hobbies will also be listed on their profile with the similar ones highlighted.

3.) Technology Stack:

React Native with Typescript (Expo): React Native Expo is used because I am on windows and this provides the most versatility when coding for IOS and Android, without needing a MacOS device.

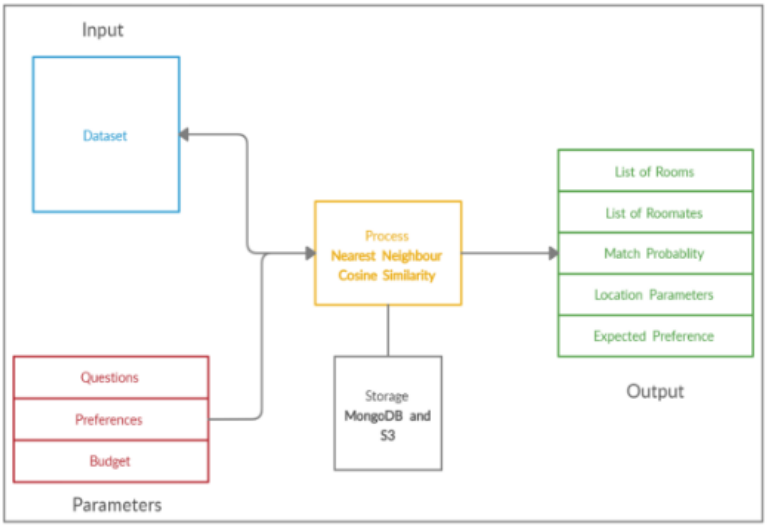
Python with Flask: Python is used because of its easy syntax and access to a variety of libraries including Flask. Flask is used for API communication. SQLalchemy is a tool in Flask that is used for retrieving data and other database methods.

4.) Backend / Methods (Mathematics)

The backend will be supported by Python and Flask, but at its core it will be powered by K nearest neighbors, K-means clustering, and finally cosine similarity.

- K means clustering will be used based on statistics gathered from students such as GPA, extracurriculars, home state, etc. These statistics will place students into clusters.
- From these clusters, euclidean distance will be used to find the K nearest neighbors of our user. These K users will be added to our card stack. (Note: since we don't gather a dependent variable, like roommate satisfaction % we just use euclidian distance from KNN, not the classification aspect.). Data will be standardized and manipulated accordingly.
- Cosine similarity will then be computed for each of the 5 users in the stack compared to the current user. The cosine similarity will be computed based on a survey of 10 questions based on user habits (Bed time, nights spent out a week…). Each question will be multiple choice and be given a score 1-3, making the data standard for each question.

Finally, there will be a tab in the application that shows the users that have liked your profile. This built in feature will guide the user into using the Gale-Shapley algorithm (assuming there is a like limit). This means that each user will like their first options in their card stack, however, there will be other users that they may not have initially liked that they could potentially match with (Similar to Hinge).



(Reference 1 Below)

Above is a rough structure of the database architecture. MongoDB replaced with Flask and some of the outputs from the diagram won't be included.

Initial app.py: https://github.com/richiefleming10/AppProject/tree/main/MyNewProject/Backend

5.) Frontend

The frontend of this application is designed using React Native and Expo. React Native is used because of the vast amount of design and animation libraries available that make our interface enjoyable and intuitive to use. It features three main tabs that enable users to interact with the app's core functionalities: the card stack, the direct message tab, and the likes tab.

1.) Card Stack Tab:

- This tab displays the card stack. Similar to a deck of cards.
- K amount of users are loaded at a time. Once a stack of users reaches the end, another stack is loaded.
- Swipe Feature: Users can swipe right to like a user and left to pass.

Demo of Swipe: RPReplay_Final1715378202.MP4

2.) Direct Message Tab

- Overview: This tab allows users to directly message other users they have been matched with.
- Messaging Interface: It features a chat interface where users can send and receive messages in real-time, fostering communication between potential roommates.

3.) Likes Tab

- Overview: This tab displays a list of users who have liked the current user's profile.
- Interaction: Users can see who is interested in them and decide whether to like them back, creating mutual matches.
- Profile View: Clicking on a user in the likes tab brings up their profile for more detailed viewing, helping users make informed decisions.

GitHub Repo: https://github.com/richiefleming10/AppProject/tree/main

6.) References:

(Gale-Shapley)

https://en.wikipedia.org/wiki/Gale%E2%80%93Shapley_algorithm

(Backend Methods) - M. Tech, Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science & Technology, Chennai, India

https://deliverypdf.ssrn.com/delivery.php?ID=413074031111068083003091119070027106002054027061023062027093008090013031030101072124042100017007024022045011097083085030022099039010011061018089103109089090089001067050087064064122122101089012121019094076065080007080107001092098084066120109025082114116&EXT=pdf&INDEX=TRUE

(Snapkit)

https://devportal.snap.com/manage/apps/bc443695-17b1-4751-adbe-861b77b1b386/version/0

(Cosine Similarity)
https://www.geeksforgeeks.org/predict-tinder-matches-with-machine-learning/

(GS / Stable Marriage Problem)
https://www.geeksforgeeks.org/stable-marriage-problem/#