p-Recommendation

AI-driven user recommendation system for a real world-mobile app

---

Project Manager: Landen Isacson, Pomona'27,
Contact: lcid2023@mymail.pomona.edu

Project Manager: Devansh Taliyan, Pomona'27
Contact: dtte2023@mymail.pomona.edu

Project Overview

Recommendation and matching algorithms play a pivotal role in enhancing user experiences across various platforms, including social media (Instagram, Snapchat), dating apps (Bumble, Hingle), video games (Fortnite), and professional networking sites (Linkedn). These algorithms utilize advanced data analysis techniques to personalize suggestions, match users with relevant content or individuals, and improve engagement by predicting user preferences. Our project explores this domain by focusing on developing a tailored recommendation algorithm to facilitate meaningful connections in real-world settings.

This algorithm will be implemented in the in-development mobile app Yaaro, which is already live with real users and active data. Yaaro is designed to make meeting people locally and forming in-person connections more accessible. By leveraging user data and behavioral patterns, the app aims to recommend nearby individuals with shared interests.

Through this project, team members will:

- *deepen their knowledge in database management, data preprocessing, and machine learning model construction while testing and refining algorithm performance.*
- *work on a live app with real-world impact, team members will not only enhance their technical skills but also gain insight into the challenges of deploying scalable, user-friendly systems.*

Tech Stack

- Front-end:
    - React Native mobile app development
    - TailwindCSS for beautiful UI
    - WebSocket for real-time features
- Back-end:
    - FastAPI (Python) backend
    - Sentiment analysis: Custom BERT model
    - Group matching: Constrained k-means clustering
    - Real-time processing: Redis
    - User data: PostgreSQL + Supabase

Dataset

We'll build our recommendation system by combining research data (UMich Psychology, EventBrite, ConvoKit) with real user interactions. Initial models will be trained on established datasets to understand group dynamics and event matching patterns.

The system will then be refined using our 200+ users' onboarding quiz responses, creating real-world interest clusters and testing our compatibility predictions. As users interact, we'll track messaging patterns, engagement levels, and event completion rates to continuously improve our matching algorithm. This data helps us understand which groups successfully meet up, which events lead to future interactions, and what conversation patterns indicate strong group chemistry.

Ethical Considerations
Our recommendation system actively works to prevent social bubbles while maintaining meaningful connections. We address potential biases in three key ways: 1) Regular auditing of our matching algorithms to ensure demographic fairness and prevent reinforcement of existing social patterns, 2) Transparent data usage and privacy controls that give users full ownership of their information, and 3) Diversity-aware group formation that balances compatibility with exposure to varied perspectives and backgrounds

Team Structure
- Target Team Size:
    - *4-6 students*
- Subteam Structure:
    - *AI/ML Team: Building recommendation and matching systems*
    - *Backend Team (2 members): API development and data infrastructure*
- Organizational & Collaboration Frameworks:
    - *Slack, Github, Google Docs, Supabase*

Timeline

Week	Tasks
Week 1-2	Setup & Planning - Group member introductions - Discuss project timeline - Assign roles and familiarize with tech stack
Week 3-4	Core Algorithm Development - Implement basic group matching using k-means clustering. - Set up the data pipeline for user responses. - Create a baseline recommendation system. - Begin PostgreSQL integration for user data.
Week 5-6	Feature Implementation - Develop NLP sentiment analysis using BERT. - Build group compatibility scoring system. - Implement event recommendation algorithm. - Create real-time WebSocket functionality.
Week 7-8	Integration & Testing - Connect ML algorithms with the React Native frontend. - Test with waitlist user data in Supabase. - Implement feedback collection system. - Begin A/B testing different approaches.
Week 9	Optimization - Address user test feedback and fine-tune the algorithm for better accuracy and fairness. - Audit the algorithm for potential biases - Optimize back-end performance for scalability and efficiency.
Week 10	Final Polish - Prepare a presentation showcasing project outcomes, key challenges, and solutions. - Document the codebase, algorithm design, and ethical considerations for future reference.

Project Member Requirements
- Minimum requirements:
    - Intro CS courses
    - Basic understanding of data science
    - Interest in ML/AI applications
- Helpful to know!
    - Machine-learning libraries
    - A fun attitude and eager to learn!!