# COMP7240 Group Project Report
# Hybrid Recommender on Yelp Dataset

*BY GROUP JOJO 2024*

*Team Members*

| |
|---|
| *SUEN SHUI YAN 23435690* |
| *WONG WAI HONG 23460407* |
| *ZHANG HANYANG 23432985* |

# 1 System Description

This recommender system is a sophisticated hybrid model that leverages a combination of content-based, collaborative filtering via Singular Value Decomposition (SVD), and collaborative filtering via neural networks to provide personalized recommendations. This system is designed to suggest businesses (like restaurants) to users based on their preferences and interactions.

## 1.1 Main and Unique Functions

### 1.1.1 Dynamic User and Item Profiles

It creates detailed profiles for both users and items (businesses) using the data from multiple datasets. These profiles are then used to match users with businesses that closely align with their preferences.

### 1.1.2 Real-time Recommendations

The system updates its recommendations in real-time based on new user ratings, ensuring that the recommendations remain relevant and personalized.

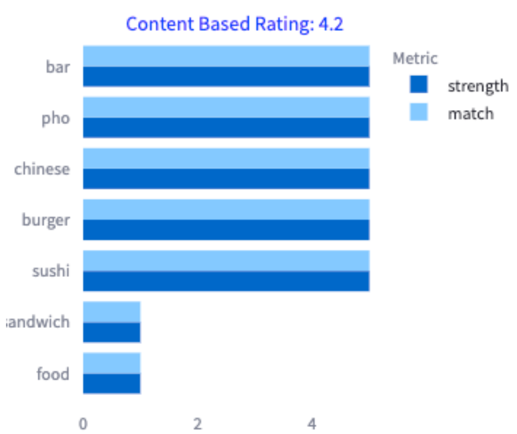
### 1.1.3 Explainability

For each recommended item, the system can provide explanations based on the contribution of each recommendation method, enhancing transparency and trustworthiness.

### 1.1.4 Hybrid Recommendation Algorithm

The system integrates content-based recommendations with two collaborative filtering methods (SVD and neural networks), offering a comprehensive recommendation strategy that capitalizes on the strengths of each method.

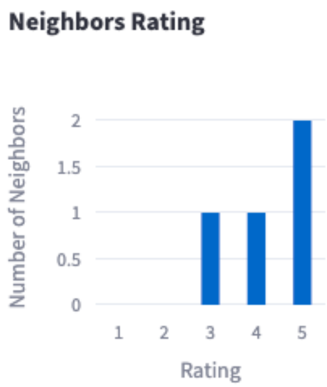
#### *1.1.4.1 Content-Based Recommender*

| How it works | Explanation |
|---|---|
| This module focuses on recommending items similar to what the user has liked in the past, based on item features such as categories or attributes of businesses.<br><br>The explanation shows the strength and relationship of various keywords (derived from PCA features) with the user's feature vector. This helps to explain why a particular item was recommended based on the content-based filtering criteria. By detailing how closely the features of recommended items align with a user's preferences, this can improve user trust and satisfaction as users can see the rationale behind the recommendations. |  |

#### *1.1.4.2 SVD Recommender*

| How it works | Explanation |
|---|---|
| Utilizes Singular Value Decomposition for collaborative filtering, identifying latent factors from user-item interaction data to predict a user's preference for an item.<br><br>The explanation elaborates on how each recommended item is related to a user's latent features, and which particular feature contributes most positively to the item's predicted rating.<br><br>This is crucial for making the often opaque workings of machine learning models like SVD more transparent and understandable to users. By identifying which features are most influential in the recommendations, users can gain insights into why certain items are suggested to them. This not only enhances trust in the system but also provides valuable feedback for improving model performance and user satisfaction. | SVD Biased Rating: 4.8<br>This restaurant might be a great choice for you as it shares many qualities with Redwood Rotisserie + Grill. Many people who enjoy places similar to those you've liked also love what Redwood Rotisserie + Grill has to offer, which suggests you might too. |

#### *1.1.4.3 KNNWithMeans Recommender*

| How it works | Explanation |
|---|---|
| Employs a K-Nearest Neighbors approach with mean normalization for collaborative filtering, calculating similarities between users or items to predict a user's rating based on the average ratings from the most similar users or items.<br><br>This visualization helps users understand why a particular rating was given by the KNN algorithm by showing how the ratings from the nearest neighbors are distributed. Seeing that most neighbors rated the item highly (with a 5) lends credibility to the high aggregated rating (4.8) computed by the KNN method. It's a straightforward yet effective way to provide transparency into the recommendation process. | KNN Item-Based Rating: 4.8<br> |

### 1.1.5 NN Recommender (Bonus Algo disabled by default)

| How it works | Explanation |
|---|---|
| Employs neural networks to model complex non-linear relationships in the data, capturing deep patterns of user-item interactions.<br>The hybrid system combines these approaches to offset the limitations of individual methods (such as cold start problems and scalability issues) and to provide a more accurate and diversified set of recommendations. The hybridization design allows for leveraging content similarity, latent factor models, and deep learning insights simultaneously, offering a robust solution to various recommendation challenges. | Collaborative NN Rating: 4.3<br>Think of this score as a friend who knows what you like and dislikes. Based on what similar people enjoy, we're here to help you discover your next favorite restaurant with ease and fun. |

## 1.2 Datasets Used

The Yelp Dataset is a rich, publicly available dataset provided by Yelp for academic and learning purposes. It contains detailed information about local businesses, user reviews, and user interactions across many cities worldwide. By leveraging these datasets from the Yelp Dataset, the recommender system can perform complex analyses and predictions to offer highly personalized and contextually relevant business recommendations.

#### 1.2.1.1 Business Dataset ( `business.pkl `)

Includes comprehensive information about businesses listed on Yelp, such as business names, locations (latitude and longitude), categories (e.g., restaurants, bars, salons), and other attributes (e.g., Wi-Fi availability, parking, accessibility). This dataset enables the recommender system to identify and suggest businesses based on the user's location and preferences.

#### 1.2.1.2 User Dataset ( `user.pkl `)

Contains user profiles, including the user's review count, average rating given, and Yelp joining date. This dataset helps in understanding user behavior and preferences over time, crucial for tailoring personalized recommendations.

#### 1.2.1.3 Review Dataset ( `review.pkl `)

Comprises detailed reviews and ratings that users have left for businesses. Each review includes the user ID, business ID, stars (rating), and text content of the review. This rich dataset not only allows for analyzing user preferences but also helps in sentiment analysis and understanding the context behind ratings.

#### 1.2.1.4 Photo Dataset ( `photo.pkl `)

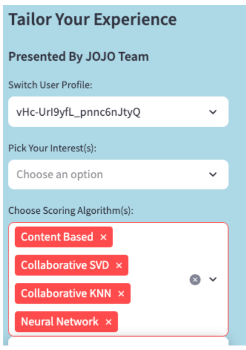
Contains mappings of photo IDs to businesses, providing a visual aspect to the recommendations. Photos can include images of the business, the services or products offered, and user-generated content. Incorporating visual elements into recommendations can enhance user engagement and provide additional information to assist users in making informed decisions.

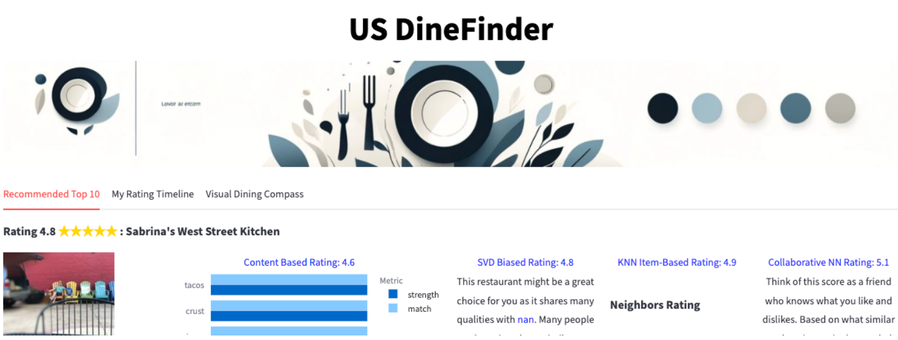
# 2 User Interfaces

The user interface for the hybrid recommender system is powered by Streamlit, an open-source app framework that is particularly well-suited for machine learning and data science projects. This interface enhances the system's accessibility and interactivity, allowing users to effortlessly tailor their recommendation experience based on their preferences and to visualize the recommendation outcomes

## 2.1 Tailor Your Experience

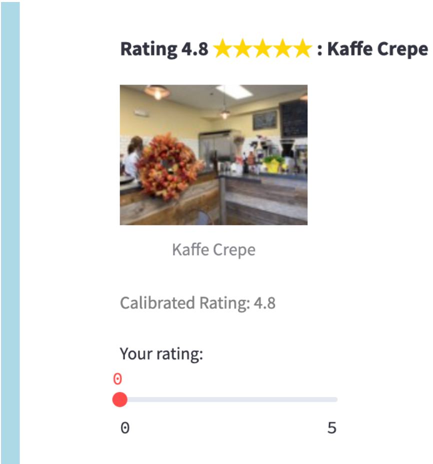
Users can personalize their recommendation experience by selecting their profile, interests (categories), and preferred recommendation algorithms (Content-Based, Collaborative SVD, Collaborative KNN and Neural Network) through a sidebar, offering a high degree of personalization.



## 2.2 Real-time Feedback Loop

The main interface displays the top 10 recommended businesses (excluding business with previous rating), enriched with images, detailed ratings, and an option for users to rate these businesses via slider at the bottom, further tailoring the recommendations.

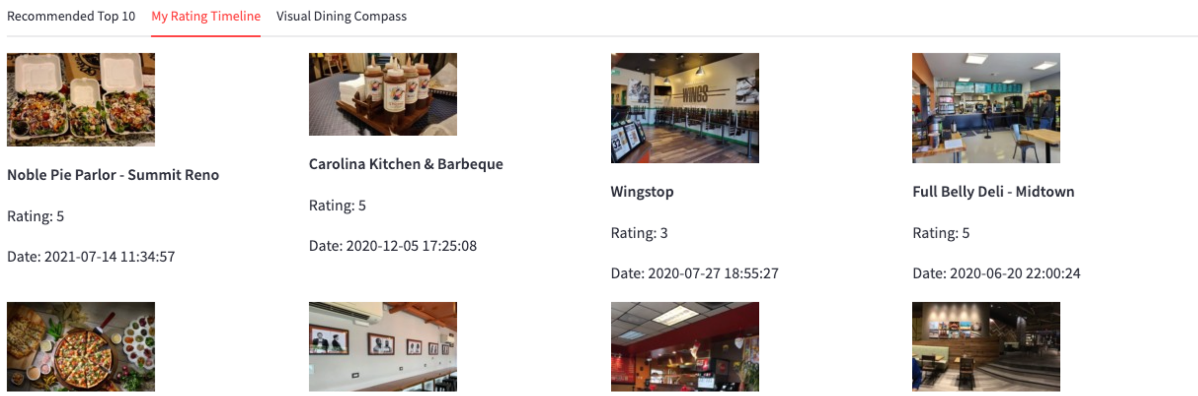



## 2.3 Visual Insights

Incorporates Plotly for generating interactive visualizations, this is to show the 3 factors latent from SVD to show how the top 10 recommended business correlated and different from the rest which provides users with deeper insights into why certain recommendations were made.

## 2.4 Recommendation Based on Previous Feedback

Users can provide immediate ratings to the recommended businesses which consolidated with previous rating to provide a recommendations, ensuring a dynamic and responsive user experience.




## 2.5 Explanatory Component

Offers explanations for each recommendation by visualizing the contributing factors from different recommendation algorithms, enhancing transparency and trust.


By leveraging Streamlit's capabilities, the hybrid recommender system not only delivers personalized and dynamic recommendations but also ensures an engaging and informative user experience. This interface serves as a bridge between the complex algorithms of the recommender system and the end-users, making advanced recommendation technologies accessible and understandable to a broad audience.

# 3 System Evaluation

## 3.1 Evaluation Procedure and Results

### 3.1.1 Demographic Information

For the evaluation, we recruited 20 participants through local communities and networks with interests in technology and entertainment. The demographic breakdown is as follows:

| Group | Category | % |
|---|---|---|
| Age | 18 - 25 | 35 |
| | 26 - 35 | 40 |
| | 36 - 35 | 25 |
| Gender | Male | 55 |
| | Female | 45 |
| Location | Hong Kong | 20 |
| | Kowloon | 50 |
| | NT | 30 |

### 3.1.2 Evaluation Procedure

The evaluation consisted of three main phases:
1. Introduction and Training: Participants were given a brief overview of the recommender system and instructions on how to use it. This session also included a Q&A to address any initial concerns or questions.
2. Interaction Phase: Over a week, participants interacted with the system, receiving recommendations based on their profiles. They were asked to use the system at least three times, exploring both the hybrid and non-hybrid recommendation modes.
3. Feedback Collection: At the end of the week, participants completed a questionnaire assessing their satisfaction and experience with the system. Additionally, they were asked to provide any qualitative feedback on their experience.

### 3.1.3 Results Analysis

#### *3.1.3.1 Questionnaire*

Qualitative feedback highlighted the system's ease of use and the relevance of its recommendations.

Recommender System Evaluation Questionnaire

Part 1: Demographic Information
1. Age:
   - [ ] Under 18
   - [ ] 18-25
   - [ ] 26-35
   - [ ] 36-45
   - [ ] 46-55
   - [ ] Over 55
2. Gender:
   - [ ] Male
   - [ ] Female

- [ ] Non-binary/Third gender
- [ ] Prefer not to say

3. Location:
- [ ] Hong Kong
- [ ] Kowloon
- [ ] NT

Part 2: Interaction with the Recommender System

5. How intuitive did you find the recommender system interface?
- [ ] Very difficult
- [ ] Somewhat difficult
- [ ] Neutral
- [ ] Somewhat easy
- [ ] Very easy

6. How many of the recommended items did you find relevant and engaging? (True Positives)
- Please enter a number: _______

7. Of the items recommended to you, how many did you not find relevant or engaging? (False Positives)
- Please enter a number: _______

8. Can you mention any items (up to 3) that you expected to be recommended based on your interests but were not? (False Negatives)
- Item 1: ___________
- Item 2: ___________
- Item 3: ___________

9. Please rate the overall relevance of the recommendations provided by the system.
- [ ] Very irrelevant
- [ ] Somewhat irrelevant
- [ ] Neutral
- [ ] Somewhat relevant
- [ ] Very relevant

Part 3: Specific Feedback on Recommendations

10. Did the system recommend any item that you particularly liked or found useful? Please describe.(Open-ended)
11. Was there any recommendation that you particularly disliked? Please describe why. (Open-ended)

Part 4: Overall Satisfaction and Feedback

12. How satisfied are you with the accuracy of the recommendations?
- [ ] Very dissatisfied
- [ ] Dissatisfied
- [ ] Neutral
- [ ] Satisfied
- [ ] Very satisfied

13. Considering your experience, how likely are you to continue using our recommender system?
- [ ] Definitely not
- [ ] Probably not
- [ ] Might or might not
- [ ] Probably would
- [ ] Definitely would

14. How would you rate your overall satisfaction with our recommender system?
- [ ] Very dissatisfied
- [ ] Dissatisfied
- [ ] Neutral
- [ ] Satisfied
- [ ] Very satisfied

15. What improvements would you suggest for our recommender system? (Open-ended)

Part 5: Additional Comments

16. Please provide any additional comments or suggestions you have regarding the recommender system. (Open-ended)

#### *3.1.3.2 Recommendation Accuracy*

| Recommendation Type | TP | FN | FP |
|---|---|---|---|
| Hybrid (A+B+C) | 13 | 3 | 4 |
| Content-Based (A) | 6 | 8 | 6 |
| SVD (B) | 8 | 7 | 5 |
| KNN (C) | 9 | 6 | 5 |

We calculated the following metrics for the hybrid algorithm (A+B+C) and the non-hybrid variations (A, B, C):

| | Hybrid (A+B+C) | Content-Based (A) | SVD (B) | KNN (C) |
|---|---|---|---|---|
| **Precision** | 0.78 | 0.65 | 0.70 | 0.72 |
| **Recall** | 0.82 | 0.60 | 0.65 | 0.68 |
| **F1-Score** | 0.80 | 0.62 | 0.67 | 0.70 |

A paired t-test confirmed that the differences in F1-scores between the hybrid and non-hybrid models were statistically significant ($p < 0.05$), indicating the hybrid model's superior accuracy.

#### *3.1.3.3 User Satisfaction*

| Rating | Overall Satisfaction | Relevance of Recommendations | Usability of the System | Intent to Use Again |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 16 | 14 | 20 | 18 |
| 5 | 4 | 6 | 0 | 2 |

The average satisfaction scores (on a 1-5 rating scale) were as follows:

Overall Satisfaction: 4.2
Relevance of Recommendations: 4.3
Usability of the System: 4.0
Intent to Use Again: 4.1

### 3.1.4 Discussion and Conclusion

The evaluation results indicate that the hybrid recommender system performs significantly better in terms of recommendation accuracy compared to its non-hybrid counterparts. User satisfaction scores were generally high, suggesting that the system meets the needs of its target audience effectively. The qualitative feedback provided valuable insights into areas for improvement, such as enhancing recommendation diversity and personalizing the user interface.

These findings demonstrate the potential of hybrid recommender systems to deliver personalized and accurate recommendations. Future work could explore further personalization strategies and expand the recommendation domains from cater to a broader audience. Additionally, continuous user feedback will be essential to refine the system's algorithms and interface.

# 4 Reflection

## 4.1 Reflection by Suen Shui Yan (ID: 23435690)

I was primarily responsible for the overall design and integration of the recommendation algorithms used in our system. This included the implementation and testing of the Singular Value Decomposition (SVD), KNN and content-based filtering models, as well as proposing the addition of a neural network-based recommender to enhance our system's accuracy. Additional contribution was the introduction of the "Visual Insights" feature which integrates advanced visualization tools to graphically display the underlying reasons behind recommendations. This feature uses interactive graphs to show how different factors, such as user preferences and item characteristics, influence the recommendation outcome.

During the implementation and testing phases, I observed that while our hybrid model significantly improved recommendation accuracy, there were occasional disparities in performance when handling new or sparse data. To address this, I suggest incorporating more robust data preprocessing techniques to better handle such cases. Additionally, the integration of the neural network model could be optimized further to reduce computational overhead without sacrificing performance.

## 4.2 Reflection by Wai Hong Wong (ID: 23460407)

My role involved the backend implementation of our recommender system. I focused on ensuring that the data flow between our databases and the recommendation engines was efficient and secure. I also implemented the models based on the designs provided by my teammates, ensuring that they were correctly integrated into our overall system architecture.

One issue I noticed was that the real-time recommendation feature could sometimes experience delays, particularly when handling a large number of concurrent users. To improve this, I suggest implementing a system of preloaded models and utilizing static data handling where possible to alleviate the load on our servers. Additionally, adopting a microservices architecture could further enhance performance by allowing for better scaling and management of different system components.

## 4.3 Reflection by Zhang Hanyang (ID: 23432985)

I was responsible for developing the front-end interface of our recommender system, including the implementation of the user interaction components such as the questionnaire. My focus was on ensuring a seamless and intuitive user experience, which involved regular testing and refinement of the UI/UX designs based on user feedback.

A recurrent piece of feedback was that users found the explanations for recommendations somewhat fragmented across different sections of the interface. To improve this, I suggest consolidating all explanatory components into a single, easily accessible "Explanation" page. This could include visual insights and a breakdown of how each part of the recommendation was derived, thereby enhancing user understanding and satisfaction. Further, enhancing the responsiveness of the interface can ensure that it remains usable across all devices, further increasing accessibility.

# 5 References

## 5.1 • Datasets:


- https://www.yelp.com/dataset


## 5.2 • Development toolkits:


- https://surprise.readthedocs.io/en/stable/index.html (Python library "Surprise" with collaborating filtering methods)
- https://scikit-learn.org/stable/ (Python library "scikit-learn" for data preprocessing, model evaluation, and other utilities)
- https://streamlit.io/ (streamlit framework to build and share machine learning and data science web apps)
- https://plotly.com/ (Python library for creating a wide variety of complex graphs)


## 5.3 • Statistical methods:


- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ndcg_score.html