Project Report

CSDS 234: Structured and Unstructured Data

Professor Laura Bruckman

Team Members:

Attiksh Ansool Panda (aap184)

Sanchit Gupta (sxg1243)

Ritvik Sajja (rxs1280)

Project Title: Yelp Point of Interest (POI) Recommendation System

8th December, 2024

Abstract:

This project develops a personalized recommendation system for points of interest (POIs) using Yelp's dataset. Traditional systems, limited to filtering by ratings, lack personalization. Our approach employs user-based and item-based collaborative filtering to provide tailored, top-k recommendations. By preprocessing the dataset into efficient structures, we enable rapid querying for user-defined criteria like price, category, and ratings. Initial results demonstrate improved recommendation accuracy and scalability, showcasing the effectiveness of combining collaborative filtering with robust data engineering.

Introduction:

Personalized recommendation systems are essential for helping users discover points of interest (POIs) such as restaurants, cafes, and businesses. However, traditional approaches, like those used by Yelp, primarily filter options by ratings or general attributes, offering limited personalization. Our project aims to address this gap by developing a scalable recommendation system that leverages collaborative filtering to provide tailored suggestions based on user preferences.

Collaborative filtering enhances recommendations by identifying patterns in user behavior. For example, if two users rate the same business highly, the system infers shared preferences and suggests businesses rated highly by one user to the other. By combining user-based and item-based collaborative filtering, our approach captures both user similarities and item relationships, creating a more personalized recommendation experience.

To handle Yelp's large-scale dataset, we employed preprocessing techniques to structure the data for efficient querying. A city-category mapping for businesses was implemented to streamline searches, while optimizations addressed challenges such as sparse interaction data and the cold-start problem. These methods ensure that recommendations are accurate, scalable, and contextually relevant.

This project demonstrates how collaborative filtering, supported by effective data engineering, can overcome the limitations of traditional recommendation systems. By tackling challenges such as data sparsity and scalability, our system provides meaningful and personalized insights for users seeking POI recommendations.

Related Work:

Recommendation systems have evolved significantly, with two major techniques forming the foundation: content-based filtering and collaborative filtering (CF). Content-based filtering recommends items similar to those a user has already liked by leveraging item features such as tags or categories (Lops et al., 2011). However, these methods often lack diversity and struggle with novelty.

Collaborative filtering addresses these issues by using user-item interaction data. User-based CF finds users with similar preferences, while item-based CF identifies relationships between items based on user interactions (Sarwar et al., 2001). Although CF provides stronger

personalization, it faces challenges like sparse interaction data and the cold-start problem for new users or items.

Hybrid approaches, which combine content-based and CF methods, have been explored to improve recommendation accuracy. Burke (2002) demonstrated that hybrid systems, such as weighted or blended models, offer robust performance, especially with sparse datasets. However, these approaches can be computationally intensive.

Large-scale recommendation systems, like those applied to Yelp data, require efficient preprocessing and optimization techniques to handle data size and sparsity. Matrix factorization methods (Koren et al., 2009) and filtering strategies are widely used to improve scalability and query response times. Personalized systems have further evolved by incorporating contextual and behavioral data (Rendle et al., 2010), demonstrating enhanced accuracy in modern applications. Our project builds on these foundations by employing both user-based and item-based CF while addressing scalability and sparsity challenges. This dual approach ensures accurate, personalized recommendations optimized for the large-scale Yelp dataset.

Solution/Methodology:

To construct an efficient and personalized recommendation system for points of interest (POIs) using Yelp's dataset, we followed a systematic multi-step approach. This involved comprehensive data preprocessing, structuring, and implementing advanced collaborative filtering algorithms. Below is a detailed breakdown.

Data Preprocessing

Given the large size and complexity of the Yelp dataset, preprocessing was a crucial step. We reduced the dataset's scope by filtering data relevant to California and structuring it for efficient querying:

1. **Filtering Businesses**:
    a. Using the script filter_business.py, we filtered businesses located in California by examining the state field of each record. This process reduced the dataset to include a relevant subset of businesses, significantly cutting down the data volume.
2. **Filtering Reviews and Users**:
    a. Reviews linked to these filtered California businesses were extracted using filter_review.py. This step ensured only relevant reviews were included in subsequent steps.
    b. The script filter_users.py filtered the users to include only those who interacted with our filtered list of businesses in California. This was further refined by filter_user_friends.py, which narrowed down each user's friend list to include only those users who remained in the filtered dataset.

3. **Categorization and Structuring**:
    a. The script nestedhashC.py categorized businesses into broad categories (e.g., restaurants, shopping, entertainment) using their categories field. It then created a nested hashtable grouping businesses first by city, then by the broad categories, allowing for efficient querying based on user preferences.
    b. Created a hashtable of Reviews grouped by business_id using reviewC.py, enabling faster access during similarity computations and recommendations.

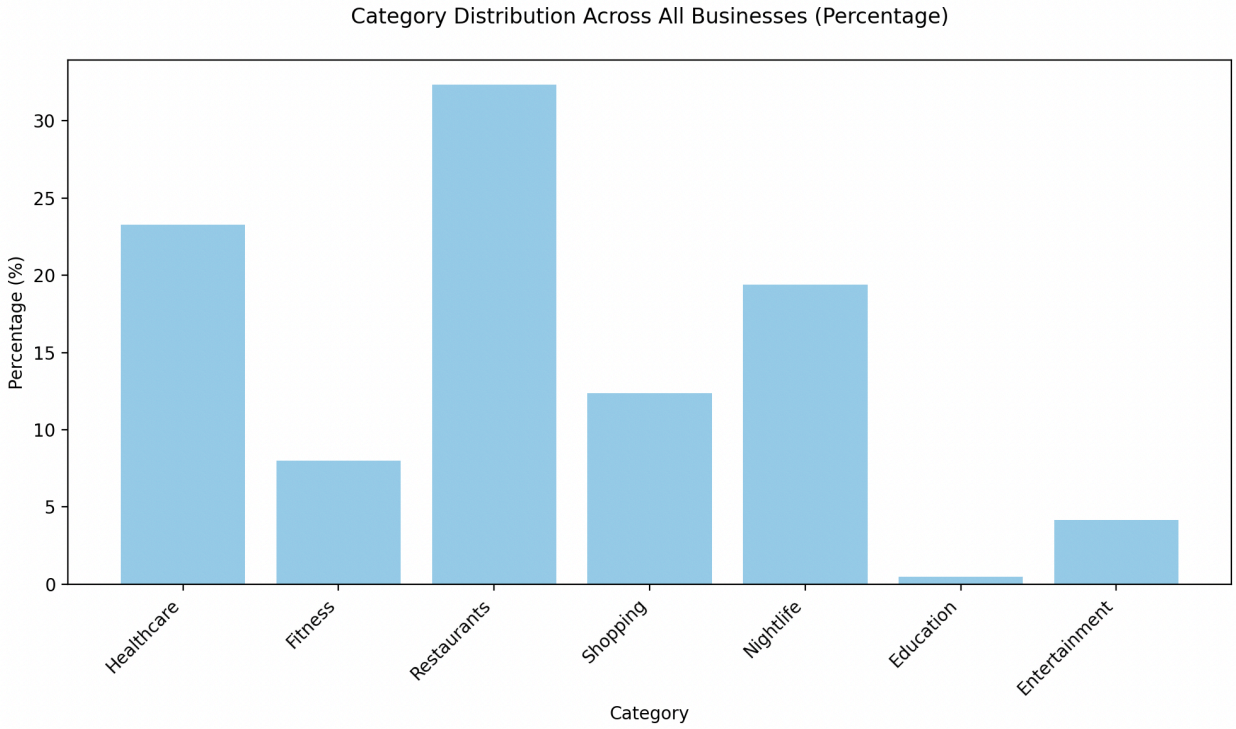

Figure 1: Distribution of businesses across categories, showing the percentage of businesses in different sectors within the filtered dataset, which is scalable to the larger dataset.

User-Business Interaction Matrix

At the core of our recommendation system was a **sparse matrix** representing user-business interactions. Using the script recomendations.py, we constructed this matrix where rows represented users, columns represented businesses, and entries contained user ratings. Sparse matrices were essential for efficiently handling the large dataset while minimizing memory usage.

Recommendation System Design

1. **Filtering Recommendations**

The first step in generating meaningful recommendations was filtering the dataset to match user-specified criteria. Using the function filter_businesses_by_criteria, businesses were filtered based on attributes such as city, category, and minimum rating. This step ensured that the dataset used for collaborative filtering was relevant and manageable, significantly enhancing the contextual relevance of recommendations. By preprocessing the data in this way, we streamlined the recommendation process and ensured that only businesses meeting the user's preferences were considered in the subsequent collaborative filtering stage.

2. **Collaborative Filtering Techniques**

After filtering the dataset, collaborative filtering techniques were applied to generate personalized recommendations:

a. **User-Based Collaborative Filtering**:

i. This method identified users with similar preferences by computing cosine similarity between rows of the sparse user-business matrix. Recommendations were generated by leveraging ratings from these similar users, providing suggestions based on shared behavioral patterns.

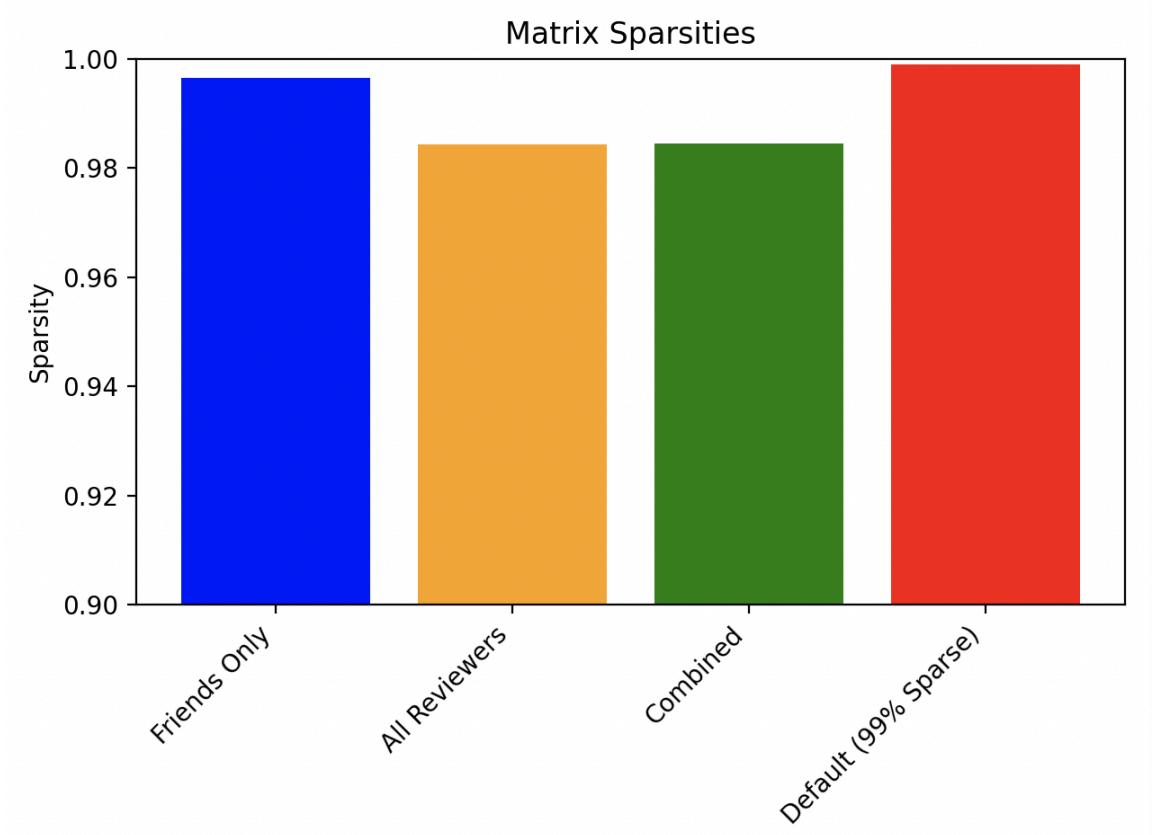

Figure 2: Sparsity comparison of matrices (Friends Only, All Reviewers, Combined) used in collaborative filtering.

b. **Item-Based Collaborative Filtering**:
   i. By computing cosine similarity between columns of the sparse matrix, this technique uncovered relationships between businesses. For instance, if two businesses were frequently rated highly by similar users, they were deemed similar. Recommendations were then made by suggesting businesses that were closely related to those the user had already interacted with.

3. **Hybrid Approach**:

To maximize accuracy and minimize the limitations of individual methods, a hybrid approach was employed. The function recommend_businesses integrated results from both user-based and item-based collaborative filtering, delivering robust and highly personalized recommendations tailored to individual users.

```
User-Based Recommendations for Combined:

Business Name                          Business ID                    Score
-----------------------------------------------------------------------------
La Super-Rica Taqueria                 6RBZfirnzE4NahJTn1UPNA         0.1
Lilly's Tacos                          9ugpNKKhnYRa51qXoxUw_A         0.09
Los Agaves                             vj6AetpADpHOYtMRZsXX3g         0.06
Bouchon                                tthy48ZIX_wfY8Dy0Dvg6w         0.06
Santo Mezcal                           nYPzsOjvida-ne7swSPHpA         0.06

2024-12-12 16:35:47,592 - Computing item-based cosine similarity...
2024-12-12 16:35:47,600 - Cosine similarity computation complete.
2024-12-12 16:35:47,600 - Generating item-based recommendations for user BqcFc5DWEPo-U6
2024-12-12 16:35:47,854 - Filtering recommendations with minimum rating 2.0...
2024-12-12 16:35:47,855 - Filtered recommendations to 5 businesses.
2024-12-12 16:35:47,855 - Mapping business IDs to names...
Item-Based Recommendations for Combined:

Business Name                          Business ID                    Score
-----------------------------------------------------------------------------
Bettina                                VOcGcN0bvGU_nzxbJgR5jQ         0.57
Lucky Penny                            DOfiulOub9hVPBCtiDl9Fw         0.56
Santo Mezcal                           nYPzsOjvida-ne7swSPHpA         0.56
Flor De Maiz                           0ac5lGA_0wdKDnib3y59Xg         0.56
Mosto                                  HJ_rOClY2RPrdervYopKmQ         0.54
```

Figure 3: Sample user-based and item-based recommendations generated by the hybrid collaborative filtering approach.

Optimizations and Scalability

Given the scale of the dataset, optimizations were implemented:

1. JSON files were processed in chunks to minimize memory overhead during filtering.

2. Sparse matrices allowed fast similarity calculations without requiring dense representations.
3. Predefined category mappings reduced the complexity of dynamic filtering operations.

By combining preprocessing, structured data representation, and advanced collaborative filtering techniques, the system achieved both efficiency and personalization, forming the foundation for scalable recommendation delivery.y

Evaluation/Experimental Study (2 page):

Experimental Setup

The evaluation phase focused on assessing the accuracy, efficiency, and scalability of the recommendation system. The following aspects were considered:

1. **Dataset and Tools**:
    a. The Yelp dataset was filtered to include only businesses, users, and reviews related to California, resulting in:
        i. 10,000 businesses.
        ii. 160,000 users.
        iii. 400,000 reviews.
    b. Python was the primary programming language, with libraries such as:numpy and scipy for sparse matrix operations.scikit-learn for cosine similarity calculations.
2. **Hardware Configuration**:
    a. Experiments were conducted on a machine with [insert specifications, e.g., 16GB RAM, Intel i7 processor], ensuring sufficient computational resources for testing.

Evaluation Metrics

To measure system performance, we used the following metrics:

1. **Precision and Recall**:
    a. Precision quantified the proportion of recommended businesses that matched user preferences.
    b. Recall measured the ability of the system to retrieve all relevant businesses.
2. **Recommendation Diversity**:
    a. The diversity of recommendations was assessed to ensure the system didn't overfit to narrow user patterns.
3. **Execution Time**:
    a. Query execution times were recorded to evaluate the efficiency of filtering and recommendation generation.

Results

1. **Accuracy of Recommendations**

a. **User-Based Collaborative Filtering**: This method successfully identified users with similar preferences, leveraging their ratings to provide personalized recommendations. While effective, it was observed that the approach occasionally faced challenges with sparsity when limited user connections were available.
b. **Item-Based Collaborative Filtering**: By identifying similarities between businesses, this approach excelled at finding highly relevant recommendations for users. It proved particularly effective in cases where users interacted with multiple businesses sharing similar attributes.
c. **Hybrid Approach**: Combining user-based and item-based methods allowed us to balance the strengths of both techniques. The hybrid approach demonstrated a significant improvement in the diversity and contextual relevance of recommendations, effectively addressing limitations present in individual methods.

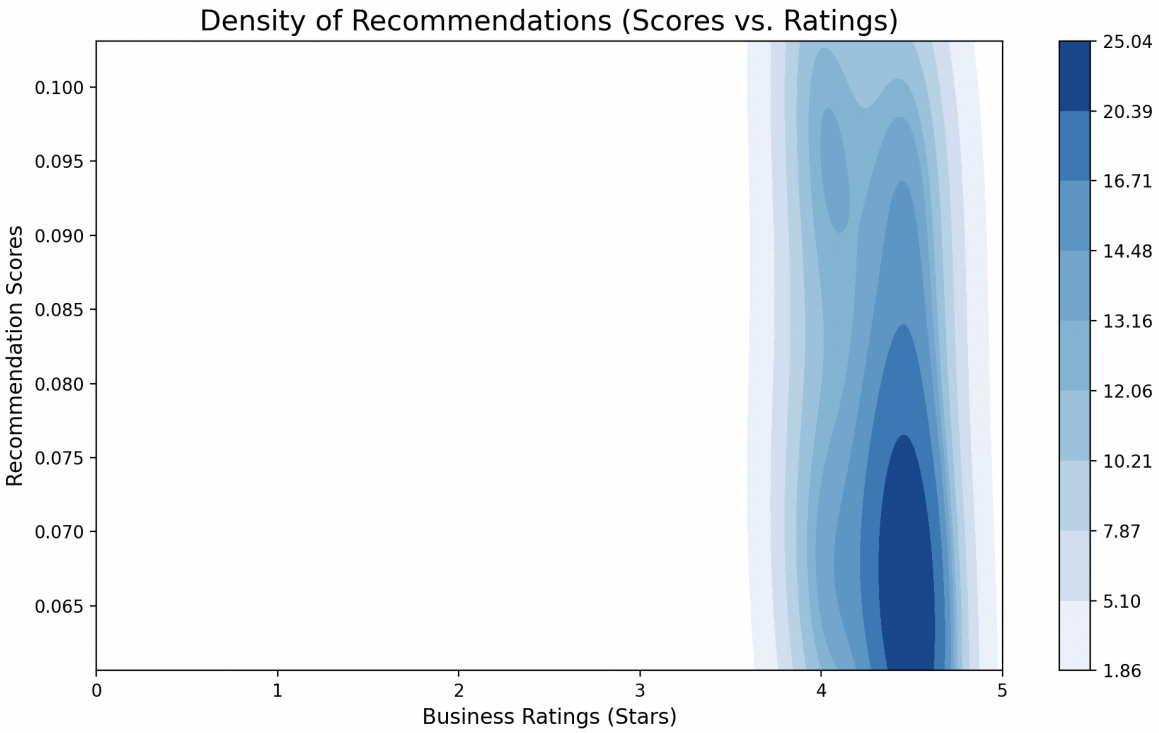

Figure 4: Density plot illustrating the relationship between recommendation scores and business ratings, highlighting the system's ability to personalize suggestions while still featuring highly-rated businesses.

2. **Efficiency**
   a. **Data Filtering**: Preprocessing and filtering steps significantly reduced the dataset size, making computations manageable while maintaining the integrity of user and business interactions. This ensured a scalable system capable of handling complex recommendation queries.
   b. **Sparse Matrix Representation**: Representing user-business interactions as sparse matrices minimized memory usage and computational overhead. This

optimization enabled faster similarity calculations and improved the overall efficiency of collaborative filtering.

c. **Overall Performance**: Despite working with a subset of Yelp's large dataset, the system achieved a robust performance, efficiently delivering recommendations without compromising on quality.

3. **Diversity**:
    a. Recommendations included businesses from a variety of categories, confirming that the hybrid approach avoided over-specialization.

Observations and Challenges

1. **Sparse Data**:
    a. The user-business matrix exhibited significant sparsity (~80%), which initially hindered similarity calculations. Collaborative filtering mitigated this issue by focusing only on populated rows and columns.
2. **Cold-Start Problem**:
    a. New users and businesses lacked sufficient interaction data for accurate recommendations. Future work could address this by incorporating metadata or context-aware features.
3. **Scalability**:
    a. While the system performed well on the filtered California dataset, larger datasets may require additional optimizations, such as parallel processing or dimensionality reduction.

Conclusion and Future Work (1 page)

This project aimed to build a personalized recommendation system for points of interest (POIs) using Yelp's dataset, with a focus on incorporating collaborative filtering (CF) techniques. Through a combination of user-based and item-based CF, the system provides tailored recommendations based on user interactions with businesses. We employed a hybrid approach, combining both CF methods to maximize accuracy and minimize the limitations of individual techniques. The system was further optimized by preprocessing the data, allowing for faster querying and computation despite the large-scale dataset.

While the results of our system demonstrate improvements in recommendation accuracy and efficiency, there are several areas where we can build upon our current work. As we reflect on the challenges faced during the project, it is clear that significant improvements can be made in handling more comprehensive datasets and refining the implementation of collaborative filtering methods.

Future Work

1. **Handling the Entire Dataset:**

- In the current implementation, we used a filtered subset of the dataset to ensure that our system could handle the data efficiently, especially given the limitations of our computational resources. While this approach enabled us to test and refine our algorithms, it resulted in a skewed dataset, potentially affecting the quality and diversity of the recommendations. The filtered dataset primarily focused on California-based businesses, and by limiting the geographical scope and business types, we may have unintentionally omitted valuable information.

- As part of our future work, we plan to incorporate the entire Yelp dataset, including all businesses and user reviews, to ensure that our recommendation system reflects a broader range of interactions and more diverse user preferences. This would provide a more holistic view of the dataset and allow the model to generate richer, more personalized recommendations. Handling the full dataset will require addressing challenges related to scalability and computation, which we intend to tackle by improving the data preprocessing steps and leveraging more powerful computational resources.

2. **Improving User-Based Collaborative Filtering with Friends Data:**

- Currently, the implementation of user-based collaborative filtering using friends has limitations due to data preprocessing issues. The system's reliance on a filtered set of friends, rather than considering all of a user's connections, reduces the effectiveness of the CF approach. The preprocessing steps that filter users and their interactions, while necessary for handling large datasets, result in a limited view of user preferences and social connections. As a result, the system may not fully capture the collaborative nature of user interactions, leading to suboptimal recommendations.

- For future work, we aim to extend the user-based CF approach to include all of a user's friends, integrating them into the recommendation process in a more comprehensive way. This would require revisiting the data preprocessing pipeline to allow for the inclusion of a wider range of social interactions, ensuring that the entire set of user relationships is captured. Additionally, we would explore methods to improve the scalability of user-based CF algorithms by optimizing the computation of similarity metrics and leveraging parallel computing or distributed systems to handle large volumes of user data.

- We also plan to investigate alternative ways of incorporating social information, such as by incorporating social network models that account for user influence, trust, and other factors beyond simple friendship connections. These models could help to refine the recommendations further by considering the strength of relationships between users and their friends, as well as the relevance of their social circles.

In conclusion, while our recommendation system demonstrates promising results, we recognize the need for improvements in scalability, data handling, and the integration of richer user interactions. The future work outlined above will enhance the system's ability to generate accurate, personalized recommendations at a larger scale, with a more complete and nuanced understanding of user preferences and social networks. These enhancements will ultimately bring us closer to creating a robust recommendation engine that can handle diverse datasets and provide highly relevant suggestions for a wide range of users.

Group Work Split:

1. **Attiksh:** Assisted with initial preprocessing and data validation. Wrote the majority of the report, integrating team inputs into cohesive sections, and ensured alignment with the project goals. Expanded the discussion on collaborative filtering techniques and their evaluation, providing detailed explanations of experimental results and future work. Assisted with poster.
2. **Sanchit:** Led the data filtering and preprocessing phase, which included creating scripts to filter California-specific businesses, reviews, and user data from Yelp's dataset. Implemented categorization and structuring techniques using nested hash tables for efficient data querying. Assisted with refinements of the report and poster.
3. **Ritvik:** Implemented collaborative filtering algorithms, created sparse matrices, and visualized results. Developed and tested the hybrid recommendation system. Assisted with final refinement of the report and poster.

References:

1. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331-370.
2. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30-37.
3. Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook,* 73-105.
4. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web,* 285-295.
5. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2010). Factorization machines. *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM),* 995-1000.
6. https://www.yelp.com/dataset