**Title: Building a Restaurant Recommendation System based on Yelp Dataset**
https://github.com/priyasjsu/Data-Mining

**Iqra Bismi**
**Priya Khandelwal**
**Saniya Lande**
**Shilpa Shivarudraiah**

**Department of Applied Data Science, San Jose State University**
**Data 240: Data Mining**
**Professor: Shayan Shams**
**Date: May 18, 2023**

## Motivation

Review data serves as an asset in enhancing restaurant recommendation systems. By leveraging the wealth of information contained in reviews, we can gain insights into users' preferences, needs, and experiences. This enables us to deliver personalized recommendations that align with individual tastes. The inclusion of review data along with business data helps address limitations such as data sparsity, by providing a broader understanding of user preferences and enhancing recommendation accuracy. Additionally, review data plays a crucial role in mitigating the cold start problem, allowing for effective recommendations even for new or lesser-known restaurants. Overall, utilizing review data along with business data empowers recommendation systems to offer more tailored and relevant suggestions, ultimately enhancing the dining experience for users.

## Background

Yelp is one of the most popular platforms that allow users to share their experience, rating and reviews for a wide range of business such as restaurant, shopping, food, travel etc. This project aims to build recommendation systems for the restaurant industry as it is one of the most popular businesses on Yelp. With vast amounts of information it is often overwhelming for users to manually search through various reviews and ratings to find the most suitable restaurant as per their preferences. Hence, to address this challenge, recommendation systems built on user's rating, review and restaurant's features can assist individuals to make well informed decisions. These systems are built by leveraging machine learning, NLP and statistical techniques to analyze user's historical data, and their preferences to make more personalized recommendations.

## Literature review

Krishnaraj, S.S. et al. in his paper implemented a content-based filtering approach to recommend top similar restaurants. They considered average rating and average price to calculate the weighted score to estimate the relevance of a restaurant to the user's preference. However, in our proposed content-based filtering method, we extend this approach by taking into consideration the sentiments expressed by the user's reviews as well in calculating the weighted score.

Moghadam et al. proposed an exponential similarity as a measure to compute the similarity between the users and the items which helps solving the data sparsity issues. However, this approach is not widely used in recommendation systems. For our proposed memory based recommendation system, we try to solve the data sparsity issue by considering only the business and the users having sufficient number of reviews.

Yao et al. proposed a model which combines the strength of both the aspect-based opinions as well as collaborative filtering techniques for building recommendation systems. Their approach is computationally intensive when dealing with large datasets. In our approach, we implement latent factor collaborative filtering with regularization wherein the cold-start problems can be minimized to certain extent.

Xiuze Zhoe et al. proposed a model called Rating Latent Dirichlet Allocation which takes into consideration the ratings information while implementing collaborative filtering on the LDA framework. In our proposed hybrid approach, we built an integrated model which uses techniques like LDA topic modeling for extracting the latent topics from the reviews given by the users.

## Methodology

### Data Collection

The dataset is collected from Yelp Open Dataset, and we have considered the business and review dataset for our project. The data can be downloaded from the link: https://www.yelp.com/dataset. The business dataset provides information about various businesses, including restaurants. It contains details such as business IDs, names, addresses, cuisine, restaurant attributes etc. This dataset serves as a foundation for understanding the characteristics and features of different restaurants.
The review dataset contains customer reviews for businesses, including restaurants. It includes review text, user IDs, business IDs, star ratings, etc.These reviews offer valuable insights into the experiences and opinions of customers.

### Data-Preprocessing

The business data was filtered to consider only the restaurant businesses. The restaurants which were closed were dropped. The number of restaurants was highest for the state of Pennsylvania. Hence, we have built a restaurant recommendation system for the state of PA. The category attribute consisted of the information about the cuisine type, the ambience, etc. for the restaurant. In-order to consider these features while building the recommendation system, we exploded the category attribute and stored these features as different columns.

For the review data, review text was converted to lowercase and extra special characters and punctuation were removed. Also,review dataset for each restaurant was combined into a single text corpus.In addition to this, NLTK tools were leveraged to remove stopwords so that preprocessed review can be used for modeling and analysis.

### Model 1: Content-Based Filtering

This technique is used to provide personalized recommendations to users based on the characteristics of the restaurants and the preferences of the users. Firstly, the system receives a business name from the user as input.Then it retrieves the business data, which contains information about various restaurants, including categories and attributes. Using cosine similarity, the system calculates the similarity between the input business and other businesses based on their categories and attributes. The top 5 businesses with the highest cosine similarity scores are selected.The system then accesses the review data, which contains reviews for different businesses. Among the top 5 businesses from the previous step, the system identifies the top 5 most useful reviews for each business.From the selected useful reviews, the system determines the number of positive, negative, and neutral reviews associated with each business.The system then computes a weighted score for each business using a formula that combines the similarity score, the business's star rating, review count, and the number of positive reviews. The formula calculates the weighted score as follows: weighted_score = 0.5 x Similarity_Score + 0.3 x (stars x review_count) + 0.2 x (number of positive/5).The top 5 businesses are sorted based on their weighted scores and the system presents the top 5 recommended restaurants to the user.

### Model 2: Memory-Based Collaborative Filtering

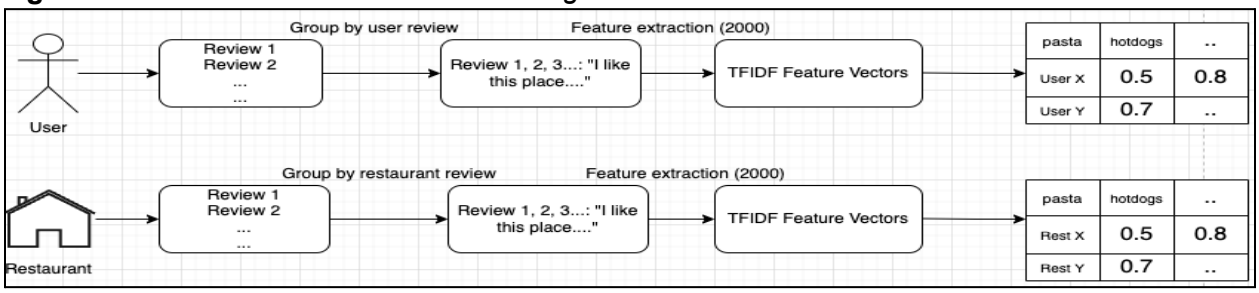
This technique takes into consideration the preference of similar users (based on ratings) to make recommendations (User-Based Filtering). The data is initially filtered to exclude the users and businesses with review count less than a minimum set threshold (set to 10) for ensuring reliable recommendations. The user-restaurant matrix is created which captures the interactions between the user and the restaurants in terms of the ratings provided by the users.

The similarity between a given user for which recommendations will be provided and other users is then calculated using cosine similarity. This helps in identifying the similar users for the given user. The training data is then filtered to consider only the businesses which were rated by these similar users. This helps to extract the information relevant to the user's preference for making reliable recommendations. This data is then grouped by business_ID and the mean rating is calculated for these businesses and the top N restaurants with highest mean ratings are recommended to the given user.

**Model 3: Latent Factor Collaborative Filtering**

Here, we have concentrated on reviews given by users using Latent Factor Collaborative Filtering on. Using this model we have concentrated only on reviews text and rating to train the model and get the recommendation. The huge matrix of user and item ratings can be divided into two smaller user-feature and item-feature matrices. For instance, if user X likes pasta but dislikes hotdogs and restaurant P serves excellent hotdogs, we may multiply the matrices using the dot product function to get the ratings. Below figure illustrates the data flow of reviews of users and restaurants. After grouping the reviews and applying the TFIDF feature to extract the feature we have restricted the count of features for now to 2000. Also, Avoid inconsistencies in data Model used regularization that prevent model from overfitting.

**Figure 1:***Data Flow of Collaborative Filtering*




**Model 4: Hybrid Model (Knowledge Based + LDA + SVD)**

Hybrid model combines the strength of content and collaborative filtering to provide more diverse and accurate recommendations. In the first stage, users are allowed to give the restaurant's location and cuisines as per their preference. This filters the data based on the user's input using FuzzyWuzzy technique i.e. if the user has entered wrong information for city and cuisine, the model will be able to fetch correct name regarding preferred city or cuisine as per the similarity score between query and data i.e. as per the similarity score between query and data.In the second stage, Positive and negative sentiment scores for the preprocessed restaurant reviews were calculated using Sentiment Analyser from NLTK tool. Also, LDA was applied to assign topic weights to each review based on the probability of the review belonging to a particular topic. Weighted score was calculated based on review count, sentiment score and LDA ( Latent Dirichlet Allocation) model topic weights. Below is the formula for the weighted score.

***Weighted Score = [ Topic Weights* (Positive Score - Negative Score) ] * Review Count***

The restaurants were sorted based on their weighted scores and top 50 restaurant names were selected for collaborative filtering using SVD++ model. In the third stage, User-item Collaborative filtering was done to capture personalized recommendations using SVD++ model which captures both explicit as well as implicit feedback.

## Experiments and Results

### Model 1: Content Based Filtering

Below is the figure of top recommendations obtained from content-based modeling for the user input, Restaurant Name: Starbucks.

```
Enter a business name to get recommendations or press 'q' to exit: Starbucks
100%|██████████████████████████████████████████████████████████████████████████| 25/25 [13:39<00:00, 32.77s/i
t]
{'ReAnimator Coffee': 126.62, 'Baltic Bakery': 16.82, 'Fine Wine & Good Spirits': 9.06, 'Snacks By The Pound': 8.98,
'Liberty City Roasters': 11.499999999999998}
['ReAnimator Coffee', 'Baltic Bakery', 'Liberty City Roasters', 'Fine Wine & Good Spirits', 'Snacks By The Pound']
[0, 1, 4, 2, 3]
Recommended Businesses are

Business Name:  ReAnimator Coffee
Stars:  4.0
Review Count:  105
Combined Features:  Food
Weightage Scores:  126.62
Business Name:  Baltic Bakery
Stars:  4.5
Review Count:  12
Combined Features:  Food
Weightage Scores:  16.82
Business Name:  Liberty City Roasters
Stars:  4.5
Review Count:  8
Combined Features:  Food
Weightage Scores:  11.499999999999998
Business Name:  Fine Wine & Good Spirits
Stars:  3.5
Review Count:  8
Combined Features:  Food
Weightage Scores:  9.06
Business Name:  Snacks By The Pound
Stars:  3.5
Review Count:  8
Combined Features:  Food
Weightage Scores:  8.98
Please enter the number of recommendations which you find relevant: 3
Precision is:  0.6
Enter a business name to get recommendations or press 'q' to exit: q
Mean Average Precision is:  0.6
```

A precision-based evaluation metric is employed for content based restaurant recommendations, using user feedback to determine the number of relevant restaurants in each recommendation providing a way to assess the performance of the content-based model without ground truth data.

### Model 2: Memory-based Collaborative Filtering:

Figure below shows the top 5 restaurants recommended to the given user with id: **_7bHUi9Uuf5__HHc_Q8guQ**

```
Recommended businesses for user _7bHUi9Uuf5__HHc_Q8guQ:
- The Sweet Life Bakeshop
- John's Roast Pork
- Terakawa Ramen
- Zahav
- Barbuzzo
- ICI Macarons & Cafe
- Morimoto
- Talula's Garden
- Reading Terminal Market
- Cafe La Maude
```

For evaluating the user-based collaborative filtering recommendation system, Root Mean Square Error (RMSE) was calculated. The recommendation system was trained on the training dataset (80:20 split ratio), whereas the test set was used to predict the ratings that a user would

give to a restaurant. The predicted and the actual rating for the restaurant was then recorded for the cases where the rating obtained was not None. Finally, RMSE was then computed by taking the square root of the mean difference between the actual and the predicted ratings. The number of similar users (k) is a hyper-parameter (set to k=10) which determines the size of the neighborhood/similar users considered for making the recommendations.The RMSE value obtained when k = 10 is 1.06.

**Model 3: Latent Factor Collaborative Filtering**

After Performing the data preprocessing and extracted the feature using TFIDF and trained the model for five hours. Hyper parameter used in this experiment: Epochs: 20, gamma: 0.001(learning rate), lambda: .02(regularization). The provided RMSE by model is 0.68, given below the input query and corresponding restaurant recommendation.

```
query = "i want to go for brunch place"
```

| | Restaurant Name | Category | Rating | Review |
|---|---|---|---|---|
| 0 | Cafe La Maude | Sandwiches | 4.5 | 1485 |
| 1 | Brunch Everyday | Restaurants | 4.5 | 198 |
| 2 | Green Eggs Café | Restaurants | 4.0 | 2679 |
| 3 | Five Guys | Food | 3.5 | 84 |
| 4 | Wm Mulherin's Sons | Restaurants | 4.5 | 610 |

**Model 4: Hybrid Model (Knowledge Based + LDA + SVD)**

Hyper-parameter tuning was done for LDA model to find the optimal number of topics based on coherence and perplexity score. In this project the optimal number of topics for review was 10 and the model was trained for 25 epochs. Also**,** hyper-parameter tuning was done for SVD++ model to increase the model's performance and accuracy. The model was trained for 100 epoch, regularization strength was set at 0.25 to reduce overfitting and improve generalization. Gradient descent optimisation algorithm was used to minimize the loss function ie. RMSE score.RMSE ( root mean squared error) of the model on test data was 0.14 which is quite less. After training, the model was used to predict ratings for a particular user and generate a list of top five recommended restaurants based on rating. Below figure shows the output obtained from the model.

**Query**

```
python trainedmodel.py chiese pihiladelpdia
Top Recommended Restaurants for User q14EAOU2U5dGbcoVqulTBw
--------------------
1.  Unit Su Vege

2.  Dump-N-Roll

3.  Chubby Cattle

4.  Wm Mulherin's Sons

5.  Saloon Restaurant
```

**Discussion**

Implemented four models using different approaches including content-based, memory-based, model (matrix factorization), and LDA, expecting different inputs and providing restaurant recommendations. The utilization of the different models allowed for a thorough exploration of all dimensions of the data, leading to the attainment of satisfactory results, thereby concluding the project successfully. This project offers users the accessibility to utilize a recommendation system with a wide range of inputs, enabling them to utilize the application in various aspects. While the hybrid model provided lower RMSE, we can say this model performed better than others but also we cannot compare the accuracy because all the models use a different set of attributes to train the model and input data and recommend the restaurants.

**Future Improvement**

Enhanced recommendation systems by integrating content-based filtering, query-based techniques on reviews, and hybrid approaches for more diverse applications and integrating all the models to implement a user interface to provide access to the end users and enhance the quality of restaurant recommendations. Implementing a mechanism to capture real-time user feedback and adapt the recommendation model accordingly will enhance the user experience. Experimenting with more advanced collaborative filtering techniques, such as deep learning-based models (e.g., Neural Collaborative Filtering), to capture complex user-item interactions is also a plan for the future.

## References


Choenyi, T., Tseyang, T., Choikyong, S., Tsering, P., & Gurme, T. (2021). A review on filtering techniques used in restaurant recommendation system. *Int. J. Comput. Sci. Mob. Comput*, *10*(4), 113-117.

Ahmed, T., Akhter, L., Talukder, F. R., Hasan-Al-Monsur, N., Rahman, H., & Sattar, A. (2021). Restaurant Recommendation System in Dhaka City using Machine Learning Approach. In 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART). https://doi.org/10.1109/smart52563.2021.9676197

Wu, Y., & Ester, M. (2015, February). Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In Proceedings of the eighth ACM international conference on web search and data mining (pp. 199-208)

https://towardsdatascience.com/how-to-build-a-restaurant-recommendation-system-using-latent-factor-collaborative-filtering-ffe08dd57dca

Chaohui Liu, Xianjin Kong, Xiang Li, Tongxin Zhang, "Collaborative Filtering Recommendation Algorithm Based on User Attributes and Item Score", Scientific Programming, vol. 2022, Article ID 4544152, 7 pages, 2022. https://doi.org/10.1155/2022/4544152

Zhou, X., & Wu, S. (2016). Rating LDA model for collaborative filtering. Knowledge Based Systems, 110, 135–143. https://doi.org/10.1016/j.knosys.2016.07.020

Moghadam, P. H., Heidari, V., Moeini, A., & Kamandi, A. (2019). An exponential similarity measure for collaborative filtering. *SN Applied Sciences*, *1*(10). https://doi.org/10.1007/s42452-019-1142-8