USER-CUSTOMIZED RESTAURANT RECOMMENDATION USING NLP ON YELP DATASET

Spring 2020: CSE6240: Web search and Data Mining

Somdut Roy

Devanshee Shah

Vitaly Marin

Introduction

- **Problem Description:**
  - Given a user with a favorite restaurant name, we aim at building a recommender system that provides a list of relevant restaurants in the city along with a possible rating he/she is likely to give those restaurants.
  - This breaks our problem into two distinct parts
    - A recommender system (to suggest those restaurants to a user)
    - A rating predicting system (to predict ratings for those suggested restaurants)

- **Problem Importance:**
  - This provides a user, a list of restaurants only based on the information about favorite restaurant.
  - Along with that, our model is capable of predicting the stars that the user is going to give each of those recommended restaurants. Here lies the uniqueness in our problem.

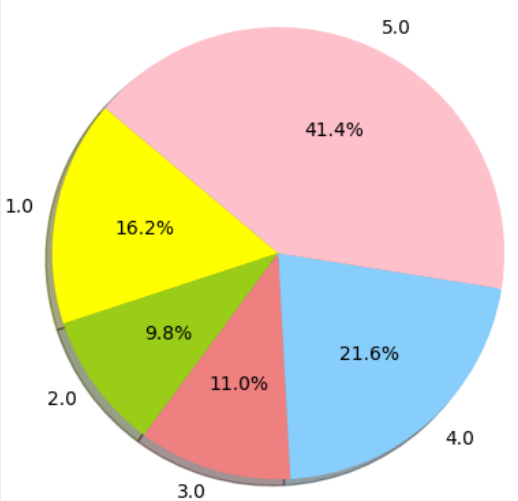
Data Discovery

Dataset Properties

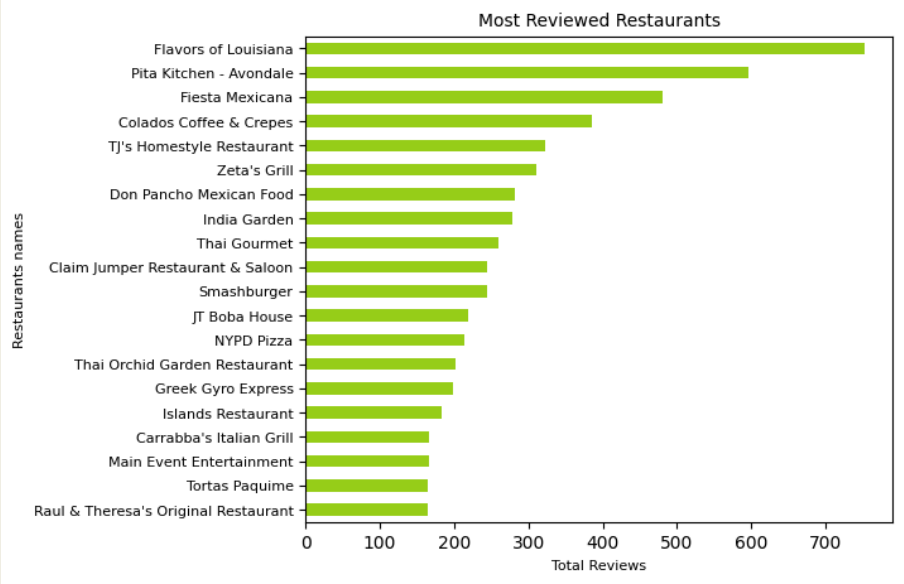
- Yelp Dataset for Business Reviews provided by Kaggle was used for our study after extracting only “restaurant” business of city Avondale, AZ from JSON file of around 6GBs. Final dataset of 7MBs consists of around 10k-12k data points (reviews).
- Dataset Attributes include Review Id, User Id, Restaurant Id, Restaurant Name, Stars, Text Review and Date. Data range from 2005 till 2018 with 8031 unique users and 163 unique restaurants.
- Text Reviews are used for sentiment analysis. Ratings (1 to 5 stars) used to find the cosine similarities among restaurants.

# Reviews	Features	# Users	# Restaurants
12662	Review Id, User Id, Restaurant Id, Restaurant Name, Stars, Text Review and Date	8031	163

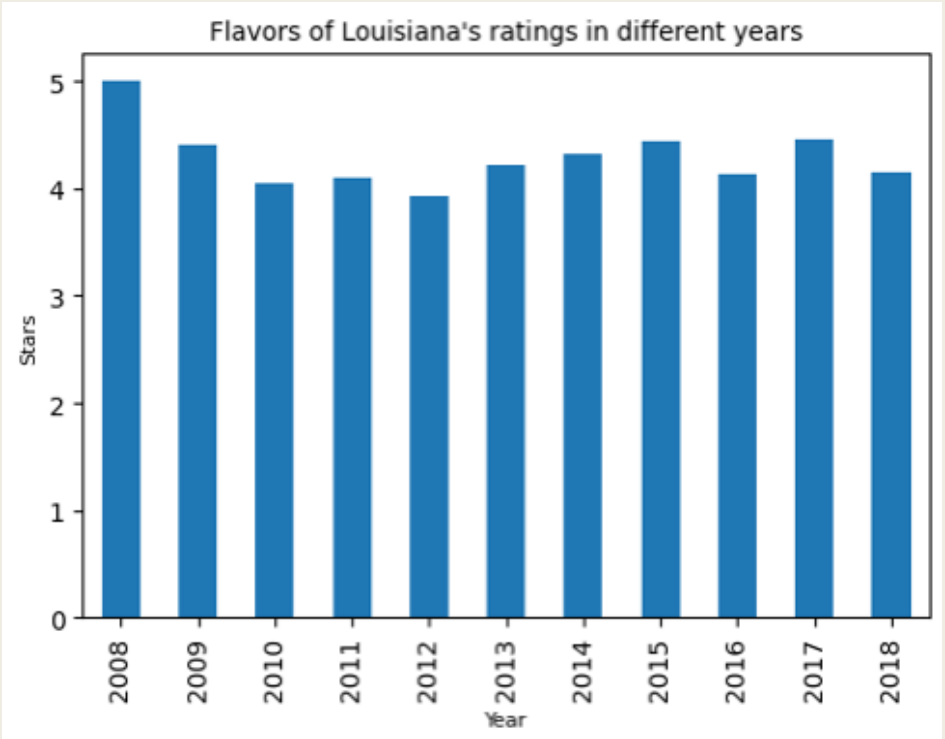
Dataset Highlights



- Star distribution for the restaurant is not uniform. More than 60% restaurants are rated very high. (4 and 5 stars).



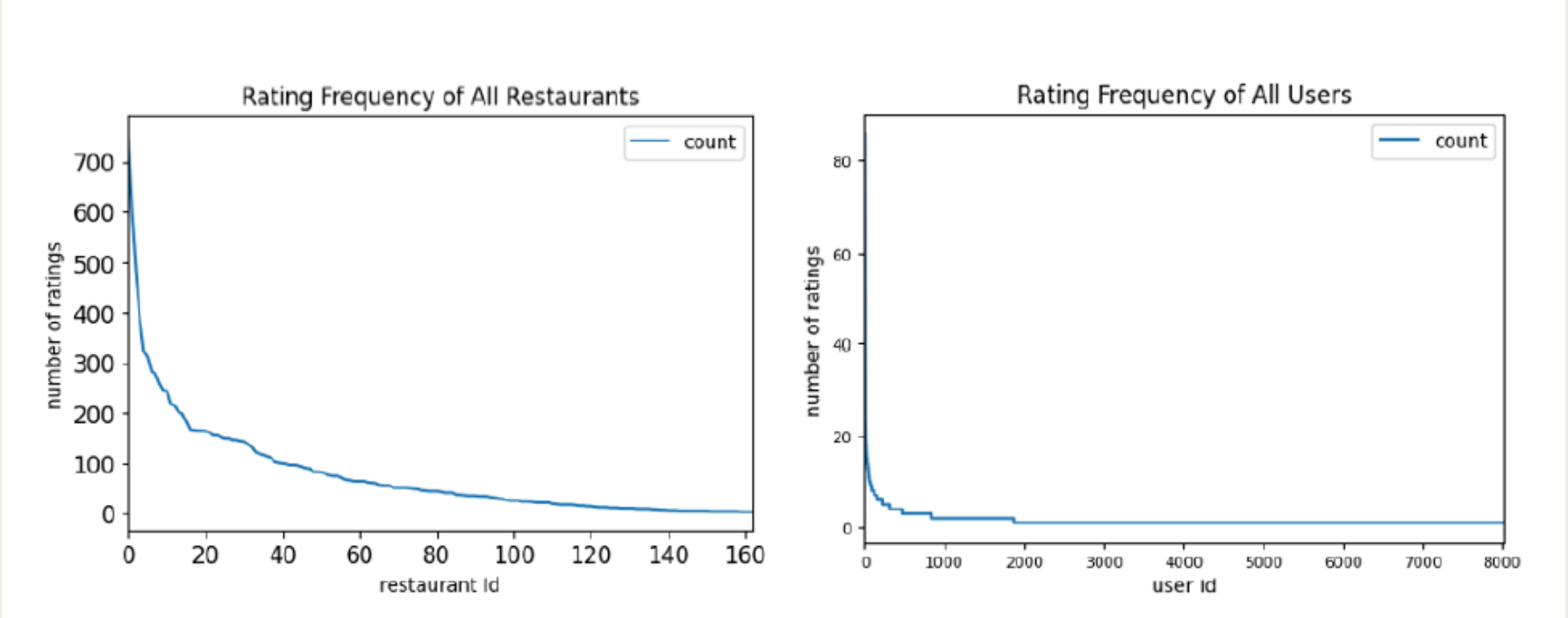
- ‘Flavors of Louisiana’ is to most reviewed restaurant in Avondale with more than 700 reviews.



- ‘Flavors of Louisiana’ consistently gets more than 4 over all the years.

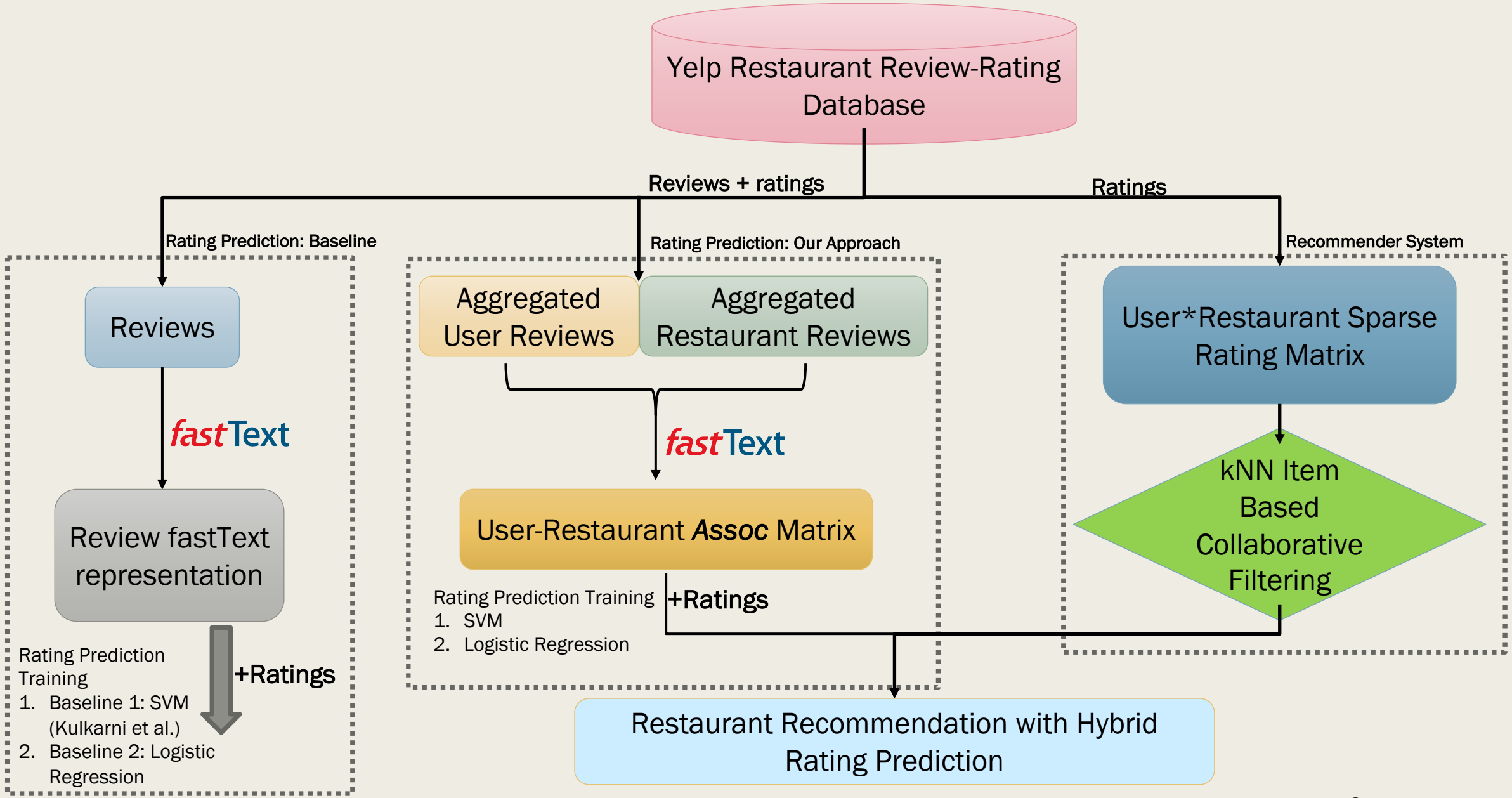
Dataset Insights

- Distribution of Rating Frequency of all the restaurants and frequency of ratings provided by users, both follow long tail property.
- That means, very few restaurants are rated very frequently - which are called the popular restaurants. Similarly, very less users are interested in rating the restaurants.



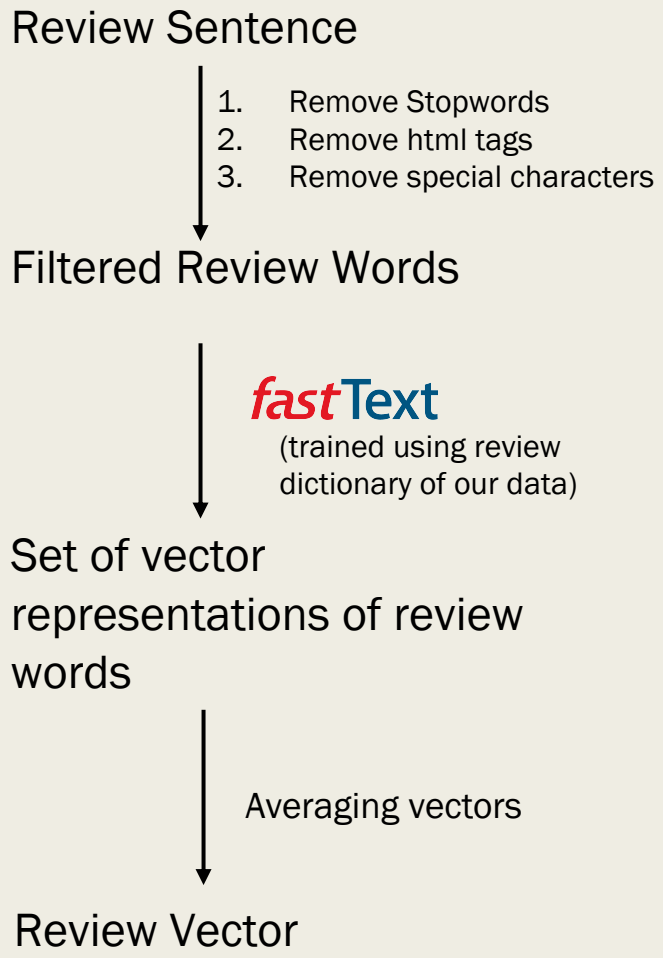
Methodology

Methodology: Overall Architecture

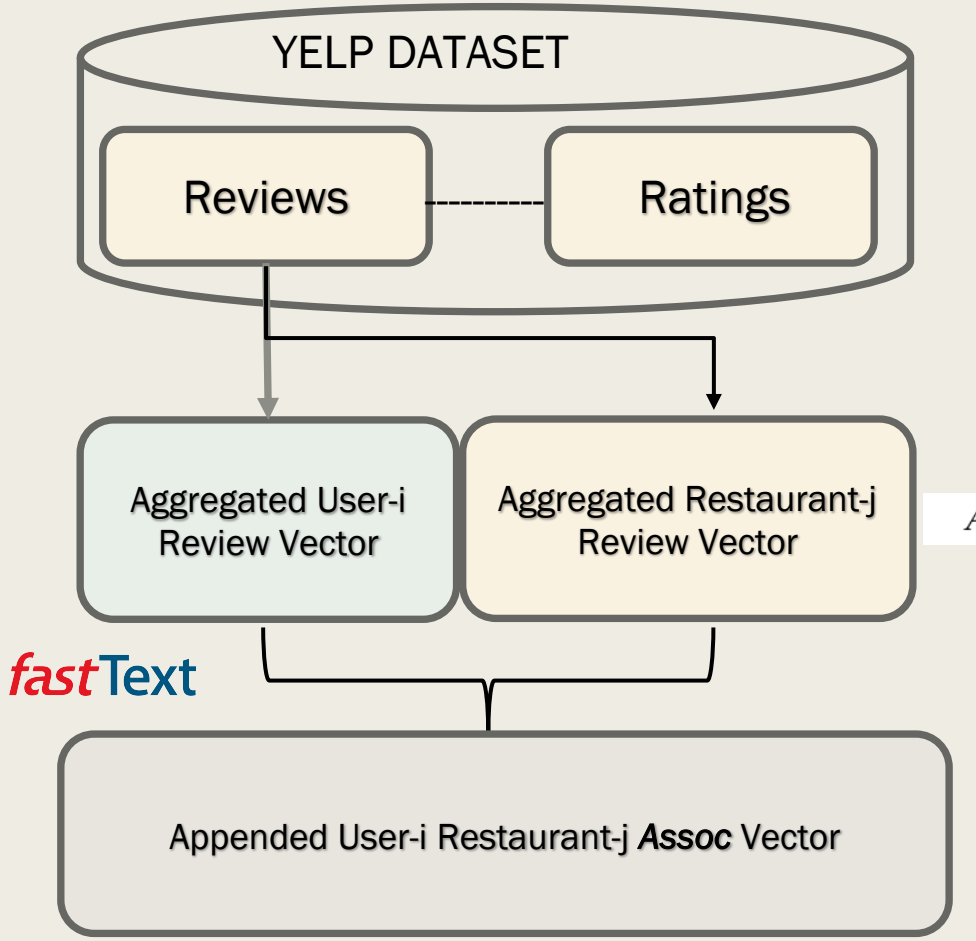


Methodology: Baselines

- Our baseline is based on a paper by Kulkarni et al., where they use review text to predict the ratings.
- While we use their machine learning techniques to create our baselines, the paper does not specify the method for word embedding. This gives us liberty to explore options for word embedding techniques.
- In assignment 2 of this course, we learned to use ***word2vec***. So that was an option. However, going by a blog by Anon (2016), using ***fastText*** suits best for a moderately small dataset like ours.
- We create embedding for reviews using the method shown here.

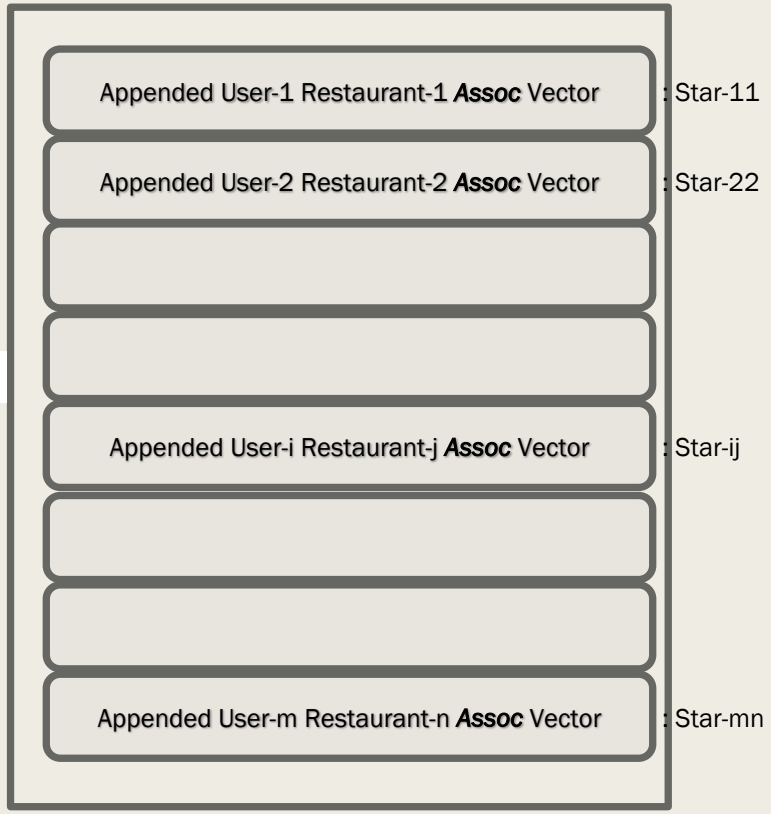


Methodology: Our Approach of *Assoc* Matrix Creation



$Assoc(U, R) = [V_U V_R]$

Assoc matrix from ***Assoc*** vector



Process of creating Assoc Vector for User I and Restaurant j

Assoc Matrix

Methodology: KNN –Item based Collaborative Filtering

- Fuzzy method is used with fuzzy ratio as 40% to find the closest matched restaurant in our dataset to the given input query.
- We fit the sparse user-restaurant matrix into KNN model with k= 20 neighbors. KNN measures cosine similarity between restaurant vectors.
- We get top 10 nearest neighbors of the closest matched restaurant with the distances.
- Result distances further computed with user-restaurant assoc matrix to get hybrid ratings.

User- Restaurant Sparse Matrix for Collaborative Filtering

Users

Restaurants	1	2		n-2	n-1	n
1	0	3		4	4	3
2	1	2		4	3	2
m-2	2	4		3	2	0
m-1	2	4		3	2	0
m	1	2		4	5	3

Restaurants **m-2** and **m-1** are having very high cosine similarity based on the ratings to both the restaurants given by users **1**, **2**, **n-2** and **n-1**.

Methodology: Flow of Experiment

Item-based Collaborative Filtering	We use Item based Collaborative Filtering to get restaurant recommendation list with the distance metric based on a user's "favorite restaurant".
Creating input matrix using stored user and restaurant "review profile vector":	We already have the "review profile vector" for the user in question and the recommended restaurants. Appending the user vector to each of the restaurant vector gives us the input matrix.
Rating Prediction for Recommended Restaurants	We implement our best-performing ***Assoc***-trained model on the input matrix to predict ratings on all the recommended restaurants for the user.
Hybrid Rating for Restaurants	The output of Item-Based Collaborative Filtering comes with a distance metric. We multiply our predicted ratings with the distance metric to create a relevance-preference hybrid rating.
Sorting Restaurants on Hybrid Rating	Finally, we sort the recommended restaurants in the decreasing order of hybrid rating.

Methodology: Setup

Experimentation :

- We used ***Google Colab*** to run our experiments.
- For accuracy, we used F1 score as the metric.
- We use k-fold (k=5) cross validation to get the parameters corresponding to optimum accuracy.
  - Linear SVM - *regularization* hyperparameter ***C***.
  - Logistic Regression -"*inverse of regularization strength*" parameter ***C***.

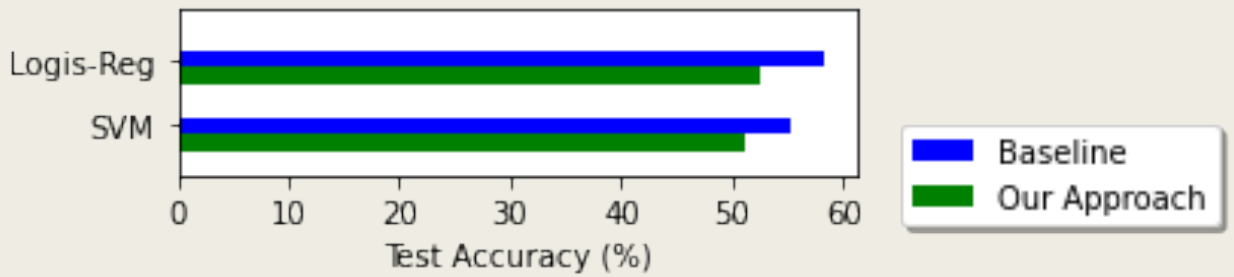
Evaluation :

- The success of our experiment was evaluated with the accuracy of predicted stars.

Results

Results: Rating Predictions

- Accuracy
  - Key difference between baselines and our approach
    - The baselines are using individual reviews to predict ratings
    - Our approach on the other hand uses the aggregated user and restaurant review profile to predict ratings
  - Cons of our approach
    - We lose some inherent domain knowledge when we use all their reviews in general instead of actual user-restaurant specific review. That causes a slight dip in accuracy compared to the baselines.
  - Pros of our approach
    - Our approach allows us to have predicted ratings for a user to a restaurant that he or she has not necessarily visited before. Since baselines needs actual reviews for the restaurant, it is unable to predict rating for a user that as not visited the place and put in a review
  - Unsurprisingly, the baselines do better in terms of accuracy.
  - Linear SVM and Logistic Regression gave similar accuracy ranging in **51 - 55 percent.**



Results: Rating Predictions (continued)

- **Computation time**
  - Linear SVM took significantly longer computation time than Logistic Regression
  - For 30 iterations, SVM baseline suggested by Kulkarni et al. was **3X slower**, compared to Logistic Regression on ***Assoc*** Matrix.
  - However, training ***Assoc*** matrix with Linear SVM took the longest (more than 10X of Logistic Regression baseline).

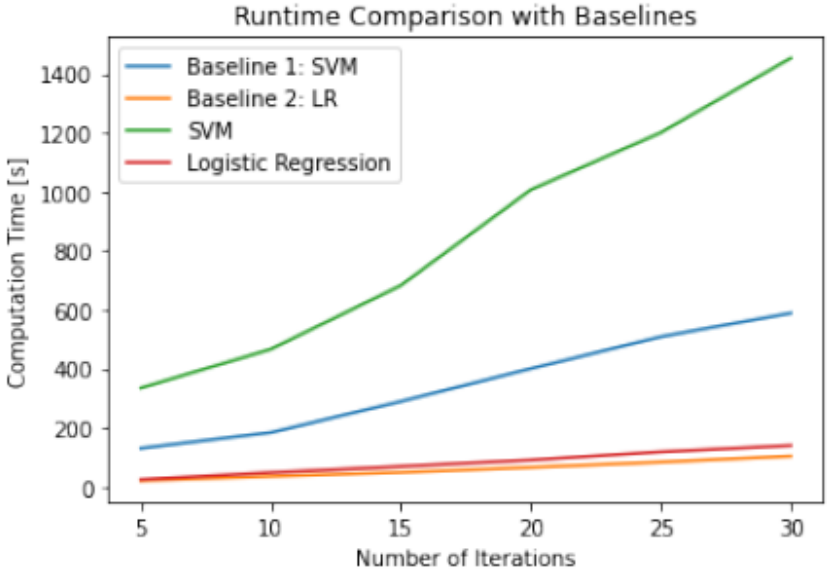


Figure 10: Run-time Comparison with Baselines

- **Verdict**
  - Based on our findings, Linear SVM and Logistic Regression give similar accuracy.
  - However, Logistic Regression takes significantly less time for same accuracy. Hence, Logistic Regression is the way to go.

Results: Recommendation with Hybrid Rating

For a user with id **'uFVAAe0JC81IPmxgT49Hcw'** who loves **Chipotle** would be recommended these restaurants:

Recommendations for Chipotle:

	Name	Distance[relevance]	Pred_Rating[fondness]	Hybrid_Rating
0	La Salsita Taco Shop	0.961311	5.0	4.806557
1	1 Brothers Pizza	0.961239	5.0	4.806196
2	Ono Hawaiian BBQ	0.961113	5.0	4.805566
3	Pei Wei	0.960317	5.0	4.801583
4	Tokyo Joe's	0.958209	5.0	4.791044
5	Raul & Theresa's Original Restaurant	0.957002	5.0	4.785008
6	NYPD Pizza	0.955957	5.0	4.779783
7	Ruby Tuesday	0.950314	5.0	4.751570
8	WaBa Grill	0.946270	5.0	4.731351
9	Culver's	0.937558	5.0	4.687792
10	Raising Cane's Chicken Fingers	0.935927	5.0	4.679637
11	Pita Kitchen - Avondale	0.935019	5.0	4.675095
12	Village Inn	0.932718	5.0	4.663588
13	Panera Bread	0.929837	5.0	4.649185
14	Red Robin Gourmet Burgers	0.908898	5.0	4.544491
15	Chick-fil-A	0.896419	5.0	4.482096
16	Yogurtland	0.834986	5.0	4.174930
17	Yogis Grill	0.957180	4.0	3.828721
18	Claim Jumper Restaurant & Saloon	0.951864	4.0	3.807456
19	Native Grill & Wings	0.951477	4.0	3.805909

Unsurprisingly, someone with 'Chipotle' as favorite restaurant has a Taco Place at the top.

Novelty and Future Work

- **Novelty of our approach**
  - Combining two worlds: The Novelty of our approach is to create the hybrid ratings by combining rating prediction for restaurants and Item based collaborative filtering to restaurant dataset together.

- **Limitations of our approach**
  - Explored one document-to-vector approach: We took the average of the word vectors in a text piece to represent the given text to create our training matrix. We could explore other operations, such as simple summation of word-vectors.
  - Used only one word embedding technique: Other word embedding techniques such as ***GloVe*** could have been explored.

- **Ongoing Work and Future Prospects**
  - Interactive GUI App: A GUI interface using Python's ***Tkinter*** library is in the works.
  - Extending Collaborative filtering to reviews: We could use the reviews for the item based collaborative filtering in the future work to benchmark it against our other two approaches.

Thank You