Restaurant Recommender System using Yelp Dataset

Atishay Jain
Department of Computer Science
UC San Diego
A59002361
atjain@ucsd.edu

Purva Kothari
Department of Computer Science
UC San Diego
A59001637
pukothar@ucsd.edu

Shivaank Agarwal
Department of Computer Science
UC San Diego
A59008799
s2agarwal@ucsd.edu

December 1, 2021

Abstract

Online businesses nowadays offer personalized recommendations to users, identifying user's preferences and providing them with relevant recommendations to improve their experience. Therefore, users would be able to enjoy the convenience of exploring what they might like with ease, often being surprised by the accuracy of the recommendations. The mainstream restaurant recommendation apps, however, have so far not incorporated personalized restaurant recommenders. Newcomers, and sometimes even locals, who are seeking new and exciting dining possibilities, have a difficult time finding an ideal restaurant. Inspired by such a challenge, we plan to develop a prototype personalized restaurant recommendation system that takes into account the interactions (visits) between customers and restaurants in different cities all over the world to get a better understanding of user's preferences. Yelp, the popular review website, has generated an abundance of information regarding modern consumer preferences and personalities. Taking advantage of the vast number of reviews, ratings, and general information provided by the community about businesses, we make recommendations for users by utilizing their personal preferences and those of similar users. We evaluate the performance of a variety of baselines built on popularities, review counts, user-user similarity and also evaluate the performance of our own models, namely Matrix Factorization (MF) and Bayesian Personalized rankings (BPR). From our results in Section 5, we conclude that BPR provides the best results in terms of recommending a restaurant to a user in a particular city.

1 DATASET

1.1 Description

Yelp dataset [1] contains information about users, businesses, and reviews of users for businesses. The dataset contains information about 8,635,403 reviews, 21,89,457 users, 160,585 businesses, 2,00,000 photos related to businesses all captured in 80 cities worldwide. Additionally, the dataset contains information of "tips" written by users for a business and user "checkin" information of businesses.

User data includes information about each user's account on Yelp, such as the number of reviews given by the user, user's joining date on Yelp, average star rating given by the user, user's friends, compliments received by user, and votes on user's reviews.

Business data includes information about location, star rating, review count, business categories, number of reviews, hours of operation and other info such as take-out availability, parking and open/closed.

Review data includes the ID of the user who wrote the review, the ID of the business for which the review was written, star rating, date, text, and other attributes such as useful, funny and cool votes.

1.2 Preprocessing

Our dataset consists of businesses of various types - restaurants, convenience stores, gyms etc. In order to extract only those businesses which represent restaurants, we created a list of category labels which indicate that

the business is a restaurant. These labels included food cuisines, food items/delicacies, restaurant types or simply restaurants/food. We filtered out only those businesses which had any one of these labels in the category information.

For evaluating our model effectively, we pre-process the data created above in such a way that we only consider the user, restaurant and review data in cases when the number of restaurants visited by the user in a city is greater than a threshold (set to 50). Also, since our task as described later is to recommend top-k restaurants (where k=3,5 in our experiments), we need enough negative samples. Hence, we take only those (user, city) pairs in which the user has not visited at least a certain number of restaurants in that city (set to 10).

After the pre-processing step, our data consists of information about 44,162 restaurants and 6,603 users in 36 cities. Fig. 1 depicts the heat map indicating the concentration of restaurants in different regions present in the filtered data.

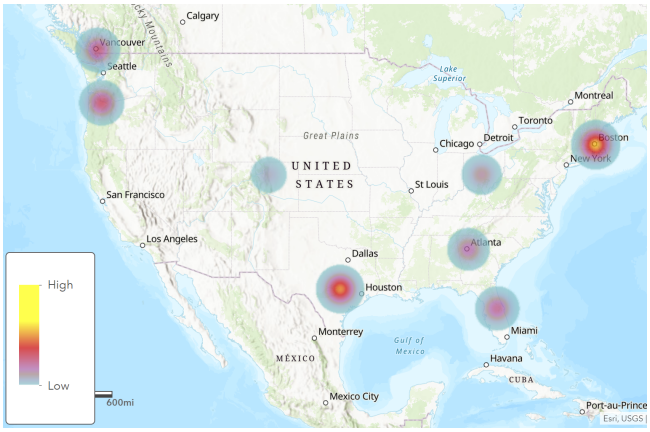

Figure 1: Heat map of restaurants in the filtered data

1.3 Feature Analysis

Although the dataset has visual features, we have not made use of them in our final models. In the case of visual features, we found the photos associated with a restaurant to belong to five different categories – "drink", "food", "outside", "interior" and "menu". Also, not every business had a photo to describe it. In the case of a few restaurants, there were many photos associated with the same restaurant which made it hard to define how a restaurant should be modeled using photos from different categories. Some of the data was observed to not be useful at all for instance "menu" photos without using OCR can be irrelevant in recommending restaurants to a user.

We found interaction data – reviews given by a user for a particular restaurant – to be the most helpful in recommending a restaurant based on her/his previous interactions. We also explored the usage of user star ratings as a helpful predictor in recommending the restaurants the user is likely to rate the highest. As described later, we classified the data into positive class and negative class using the following definitions of positive and negative interaction - Positive interaction means that the user visited a particular restaurant and has given a rating while Negative interaction means that the user has not visited a restaurant. Using these definitions, for every (user, city) pair in our pre-processed data, we categorized the restaurants into the corresponding positive and negative classes.

Fig. 2 shows the dominant category for restaurants found in the cities around Boston region. We explored these category features in some of our models.

While dealing with Yelp dataset, we have to keep in mind the enormity of the data and feasibility to include all features for predictions. Hence, understandably, there are features we didn't use while predicting our recommendations, most importantly, parts of dataset that we were not able to accommodate were "checkin", "tip" and "photo" files provided by Yelp. Of course, as part of future work, we would like to explore these features. Moving on, the parts of data that we used also had several features that we didn't exploit.

In "User" data, firstly, we tried incorporating "friends" attribute in context to similar users for recommendations, but we discovered that friends of a particular user have explored the same cities as our current user and so for a user visiting a new city, this attribute doesn't help in recommendation as studied extensively in [8]. If we take average over user and their immediate friends city span, it accounts to 8.63 cities per user, but if we remove current user from this average, it becomes almost null, which indicates that friends and users data overlap considerably, hence we found it irrelevant to proceed further with this feature. Other user data such as user's review count and his fans and compliments received were found to be more useful when taking into consideration sentiment analysis on yelp dataset and not our current problem statement which is recommendation based on user restaurant interactions in cities.

For "Review" data provided by Yelp, again, except the textual information and temporal factors of a review we tried utilizing all other parameters. In fact, "Review" data was the primary source of interactions between user and businesses. Similarly, in "Business" data, we used city as the only spatial factor in consideration (Fig. 3) , ignoring other features like address, state or geographical locations as shown in figures above because it would not make sense either to categorize too deeply based

on postal codes, or sparsely based on states. We used a location classifier as cities since it is easier to work with and also informative enough to train our models. We also made use of business ratings and review counts in one of our baselines determining how they affect recommendation to users.

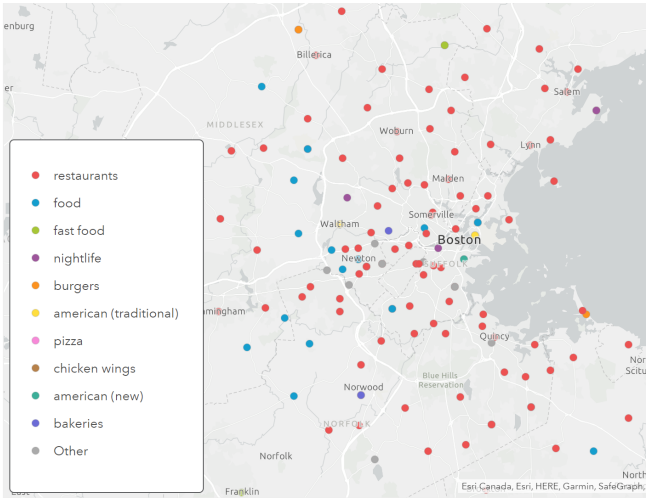

Figure 2: Depiction of the dominant category of restaurant in cities around Boston area

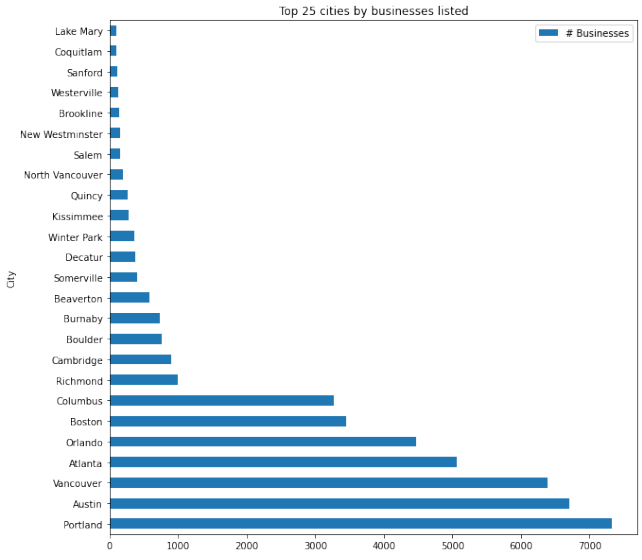

Figure 3: No. of businesses per city (Top 25)

2 PREDICTIVE TASK

2.1 Problem Statement

While deciding on a restaurant in a new city, we are often confused by the many restaurants to choose from. To tackle this problem to some extent, we propose using the Yelp dataset to build a restaurant recommendation system. Given the user and the city, our system recommends top restaurants to the user in the chosen city, based on the user and restaurant features. Unlike previous approaches that focus on a particular city, our algorithm caters to all the users and cities worldwide present in the Yelp dataset.

2.2 Evaluation Metrics

Each row of our dataset consists of a user, a city, a list of restaurants the user has visited in that city, and a list of restaurants the user has not visited in that city. The data is split into train, test, and validation sets where for each user-city combination, a random 80% of the restaurants he has visited is placed in the test set, 10% into validation, and 10% into test sets. This is done so that each user and city are present in all three sets, but none of the restaurants in these sets overlap, i.e., a user-restaurant pair will only be present in one of the three sets. 2 evaluation metrics were used:

2.2.1 Precision@k for k = 3,5

For each city user pair, top k restaurants are recommended by our algorithm and the evaluation is based on the number of restaurants the user has visited out of the ones we recommended. For example, a score of 0.2 would indicate that each of the k restaurants has a 20% percent probability of the user visiting it.

2.2.2 Top k score for k = 3,5

For each city user pair, top k restaurants are recommended by our algorithm and the evaluation is based on whether the user has visited one of these restaurants.For example, A score of 0.7 would indicate that there is a 70% probability that the top k results contain a restaurant that the user would visit.

Note that we were testing only the unseen user-restaurant pairs in both the metrics; hence our algorithm was not biased towards a particular user-city pair. Therefore the results on the test set would be equally representative of the results we would expect when recommending a restaurant to a user in a city he has never visited before.

2.3 Baselines

To get an insight of how personalized recommendation models perform better using user interactions and by how much, we user several baselines as described below:

2.3.1 Popularity baseline

A common but usually hard to beat approach is using a popularity model. This model simply recommends the most popular restaurants to the user in a particular city. So, using our training data interactions, we store the most visited restaurants in a particular city in a set along with its no. of visits and sort them in descending order to get most popular restaurants. When we encounter a test set pair of (user,city), we simply take the city parameter and recommend restaurants based on our stored restaurant popularity data for that city. We took this baseline because it is highly probable that a new user in a different city or even a local user will most likely visit the most popular restaurants in the city based on word of mouth or online reviews. So, we had to compare our model with the basic instinct of a new user. After implementing this baseline, we discovered that the popularity model, as expected performed with almost half the accuracy to the models involving user interactions.

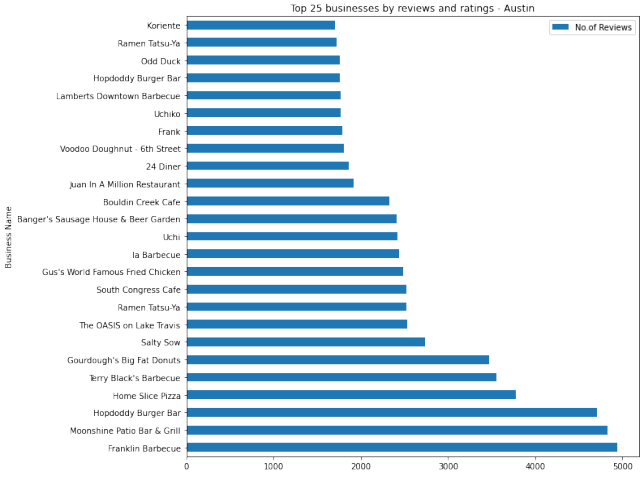

Figure 4: Top 25 restaurants in Austin based on reviews/ratings

2.3.2 Rating/Review Baseline

Another approach to compare our model with some feature of business data was using business ratings and no of reviews as feature to identify the restaurants that are most likely to be visited. We can consider that these features involve a bit of user knowledge as well as no. of reviews are mostly calculated based on user interactions and star rating can be deemed to be the average of all ratings provided by users' past history at a particular restaurant. Likewise in popularity model, here, we sort the restaurants based on (number of reviews, star rating) and provide recommendations to users. An example of how we sort restaurants based on review/ratings is shown in (Fig. 4) This model also performed poorly in comparison to our model because we don't know the origin of the number of reviews and rating of a particular restaurant. It is possible that the reviews are from particular business' history rather than the current data in consideration.

2.3.3 Jaccard User-User similarity Baseline

Since in our previous baselines, we have ignored the user data, here we take into consideration the user data and calculate similarity between users in each city based on the similar businesses visited by users to get a better understanding of what kind of restaurants each user likes and then provide recommendations based on similar users most visited restaurants and individual ratings of those restaurants. One problem encountered in this baseline was since the user data is city independent, finding similarity between users limiting to each city was a bad measurement since the user had sparse comparison data to other users for similarity measurement. So, because of above reason, this model was unable to perform better than even the popularity baseline. One approach to improve the similarity recommendation is tried below.

2.3.4 Similarity between Users + Popularity/Ratings Baseline

Using the basic similarity recommendation as described above, we included an additional parameter sighting that if similarity between two users is less than a particular threshold, we will prefer to user popularity and rating review data, sorted in that particular order above similarity. As expected, this baseline definitely performed better that the Jaccard similarity model, but still unable to beat ratings/review or even popularity baselines. This might be due the huge amount of user data at our disposal resulting in sparsity of user data similarity in a particular city, unable to provide a good similarity measurement for recommendations.

3 MODELS

3.1 Matrix Factorization

3.1.1 Model architecture

For recommending the top restaurants (not visited by the user) to the user, we predict the star rating that the user is likely to give to each restaurant. Based on the rating, we rank the restaurants and pick the top ranked restaurants as our selection for a particular user.
We design our feature vector to be one-hot encoded with the dimension equal to the sum of the number of users and the number of restaurants. For each user-restaurant interaction, we set the target variable to be the rating the user has given that particular restaurant. If a user has given more than one rating for a restaurant, we pick the highest value to be the target. If a user has not visited a restaurant, we assign the rating '0'. Each user and restaurant consists of a learned feature representation of k dimensions where k is a hyperparameter. Additionally, the model learns bias terms for every user and every item along with a global offset term. We consider two types of Latent Factor models in our case, as explained below:

3.1.1.1 Considering only user and restaurant interaction data We learn two sets of latent features, one for each user and one for each restaurant. Additionally, we learn the offset term and the bias terms for every user and every restaurant. Our aim is to minimize the following objective function:

$$F(u,i) = (\alpha + \beta_u + \beta_i + \gamma_u\gamma_i - \kappa_u, i) + \lambda_1(\sum |\beta_u|^2 + \sum |\beta_i|^2) + \lambda_2(\sum |\gamma_u|^2 + \sum |\gamma_i|^2) \tag{1}$$

Where α represents the global offset term,β_u represents the bias term for a user, β_i represents the bias term for a restaurant, γ_u represents the latent feature of the user, γ_i represents the latent feature of the restaurant, κ_u, i represents the rating given by user u to restaurant i (0 if not interacted), λ_1 is the regularization constant for the bias terms and λ_2 is the regularization constant for the latent factor features.

3.1.1.2 Including restaurant features In order to enrich our feature vector representation, we incorporated the category information present in the dataset for every restaurant. We one-hot encode these features for every restaurant and update our minimization objective function:

$$F(u,i) = (\alpha + \beta_u + \beta_i + \beta_c + \gamma_u\gamma_i + \gamma_i\gamma_c + \gamma_c\gamma_u - \kappa_u, i) + \lambda_1(\sum |\beta_u|^2 + \sum |\beta_i|^2 + \sum |\beta_c|^2) + \lambda_2(\sum |\gamma_u|^2 + \sum |\gamma_i|^2 + \sum |\gamma_c|^2) \tag{2}$$

Where α, β_u, β_i, γ_u,γ_i, λ_1 and λ_2 represent the same quantities in eq. 1 ,β_c represents the bias term corresponding to the category feature of the restaurant, γ_c represents the latent factor representation for category features of the restaurant.

3.1.2 Training and scalibility

We used FastFM library to build and evaluate our models. To fix our parameters, we explored different values of the hyperparameters λ_1, λ_2, k (or rank) and the number of iterations. After trying values for λ_1 and λ_2 - 0.01, 0.05, 0.1, 0.5, 1, 5, 10, best results were obtained when λ_1 was set to 5 and λ_2 was set to 1. The value of 200 for k gave the best results. The number of iterations was set to 250 (gradually increasing to this value as the set of parameters to explore reduced in size).
The number of experiments conducted were limited due to resource constraints. The results reported are based on an extensive number of experiments conducted on the resources available. Further fine-tuning of the hyperparameters can lead to better results.
The strength of this model is interpretability. The output of the model is a predicted user rating for a restaurant. This score is easy to interpret. There are many assumptions that were made while engineering our features and target values. When a user visits a restaurant multiple times, we picked that the highest rating given by the user as the target variable. This might not be true depiction of a user's preference towards the restaurant. For instance, may be the user liked the restaurant in the past but does not anymore because of many factors such as change of staff, change of pricing etc. Setting a rule for obtaining the right target value will never be able to generalize well in this scenario. Another drawback of this model is that it does not generalize well over spread out geographical regions. User behavior and restaurant characteristics which drive recommender systems might vary a lot over different geographical regions. There may be patterns such as users rating restaurants leniently in one region while not in the other. Restaurants themselves might be of different standards. For instance, Mexican restaurants on the east coast might not be as good as Mexican restaurants on the west coast. Hence, building rating based recommendation systems limited to a local

region is more suited than generalizing over different geographical regions.

3.2 Bayesian Personalized Ranking

3.2.1 Model architecture

As mentioned earlier, ratings may not be a good feature to work with due to several reasons such as multiple ratings of a user for the same restaurant, variation of ratings with respect to the dish ordered rather than several other factors such as restaurant ambiance, type of restaurant, etc. Therefore we treat our data as implicit where we only have positive instances of users visiting a restaurant labeled as 1 and the rest as 0. So rather than predicting a rating for a restaurant, we are more concerned with unseen instances having a lower score than positive instances. This motivates us to use the BPR model for recommending restaurants to a user.
The model is trained with instances of a user, a restaurant visited by the user, and a restaurant not visited by the user. The model aims to give a higher score to the pair {user, restaurant visited} than the pair {user, restaurant not visited}. Each user and restaurant consists of a learned feature of k dimensions where k is a hyperparameter. The score is a dot product between the user and the restaurant feature. The feature is learned for every user and restaurant through backpropagation on the training data. We consider two types of BPR models in our case, as explained below:

3.2.1.1 Considering only user and restaurant interaction data We learn two sets of latent features, one for each user and one for each restaurant. Our aim is to maximize the following function:

$$F(u,i) = (\gamma_u\gamma_i - \gamma_u\gamma_j) - \lambda(\sum|\gamma_u|^2 + \sum|\gamma_i|^2 + \sum|\gamma_j|^2) \tag{3}$$

Where γ_u represents the latent feature of the user, γ_i represents the latent feature of the restaurant visited by the user u, γ_j represents the latent feature of the restaurant not visited by the user u and λ is the regularization constant.

3.2.1.2 Including restaurant features The Yelp dataset also consists of keywords associated with restaurants such as "Mexican", "Burgers", "Bakeries", "Hotels", etc. We one-hot encode these features for every restaurant and include them in our loss function in the following way:

$$F(u,i) = (\gamma_u\gamma_i + \gamma'_u\gamma_{fi}) - (\gamma_u\gamma_j + \gamma'_u\gamma_{fj}) - \lambda(\sum|\gamma_u|^2 + \sum|\gamma'_u|^2 + \sum|\gamma_i|^2 + \sum|\gamma_j|^2) \tag{4}$$

Where γ_u,γ_i,γ_j, and λ represent the same quantities in eq. 3 , γ'_u represents the second set of features of the user u, γ_{fi} represents the one-hot encoded features of the restaurant visited by user u and γ_{fj} represents the one-hot encoded features of the restaurant not visited by user u.
Our main motivation for using the categorical features of the restaurants was to see the user compatibility with the type of restaurant. Hence a new term $\gamma' u$ is introduced for each user which captures this compatibility. Note that the terms γ_{fi} and γ_{fj} are fixed for each restaurant and hence are not included in the regularization terms and are also not updated via backpropagation.

3.2.2 Training and scalibility

For training purposes, PyTorch was used. The learning rate was decreased from 0.05 until the model loss stopped saturating at the early stages, and was subsequently set to 0.001. The regularization parameter (λ) was varied between 0.5, 0.1, 0.05, and 0.01. Higher values of λ resulted in the model not learning anything since, for high dimensionality latent features, the loss due to regularization constant would surpass the loss due to actual restaurant and user interactions. Hence $\lambda = 0.01$ was giving the best results. The dimension of the latent feature vector was set to 50.
Since our model deals with pairs of visited and not visited restaurants, one pass through all possible pairs would lead to an order greater than 10^{10} training samples. To avoid this issue, we randomly sampled one visited and one not visited restaurant for each user and city pair. This results in around 6890 training instances per epoch. For each epoch the random selection was varied. Since our model was seeing different instances for each epoch, the validation loss would only start decreasing after a certain number of epochs when the model has seen a sufficient number of instances. The epochs were varied between 100k and 200k depending on when the loss became saturated and stopped decreasing.
The strength of the BPR model is that it does not require explicit feedback for training. Hence if in the future the user visits a restaurant but does not review it, our model would be able to make use of this data which is not possible with models such as MF that recommend based on user ratings. One weakness of this model is that it requires the user to visit a few restaurants before recommending a new one (cold start). To avoid this problem

we could use some other heuristics such as popularity to recommend restaurants to a new user. The cold start problem is not the case with new restaurants since we are also making use of the keywords associated with the restaurant in our second BPR model. These keywords are specific to the restaurant and user interactions are not needed to generate these features. Hence when we see a new user we would need to recommend a restaurant based on popularity but when a new restaurant opens, our model could still recommend it to existing users.

4 LITERATURE

4.1 Yelp Dataset prior work

4.1.1 Feature Engineering

In [2], the scope of the dataset was reduced by only considering users/reviews/businesses in the city of Toronto. [2] also talks about location-based recommendation in which restaurants are grouped based on location information (latitude/longitude). The idea is to pick the cluster based on the location of the user and recommend the top K rated restaurants in that cluster to the user. The same work also mentions feature engineering using NLP. 'Super score' is computed by adding the product of Textblob's Polarity score and VADER compound score to the Yelp rating. Additionally, LDA (Latent Dirichlet Allocation) was used to build a topic per document model and words per topic model for content-based recommendations. [3] talks about interesting insights achieved using clustering analysis on the user data. It was found that users which were less popular rated restaurants more uniformly as compared to the others. In the same work, ratings (represented as classes) are predicted using vectorized features of the review text. This approach used methods such as Naive Bayes classification and SVM. Features used were Unigram + TF and Bigram + TF-IDF.
In our work, we focus on using the interaction between users and restaurants either through explicit feeback (MF) or implicit feedback (BPR). We additionally make use of the category information for the restaurants in our filtered dataset.

4.1.2 Content Based Recommendation

[2] mentions the use of Bag-of-words representation of reviews/category data (count-based vector representations) in which an input restaurant fed by user is compared with the rest of the restaurants to give top-K recommendations. [3] details an approach to model a user's profile by taking a weighted average of the restaurants she/he reviews where weights are the ratings. The user profile obtained is then used to fetch the top-K restaurant recommendations.
In our work, we combine the category data with the existing interaction data (implicit/explicit) to build models which perform better than those that do not use these features.

4.1.3 Collaborative Filtering based Recommendation

[3] explores user-user similarity to recommend a user the restaurant rated highest by the most similar user. Similarly, item-item similarity is studied to find the closest K unvisited restaurant matches to a query restaurant (highest rated restaurant by the user). In [4], ALS Matrix factorization method was used to predict the rating that a user would give to a restaurant he has not visited. While [3] explores using similarity-based approaches independently, we built one of our baselines using both similarity-based approach and popularity-based approach. Additionally, in addition to building a ranking based recommendation system as in [4], we incorporated category information to build a better performing model than one without category information.

4.1.4 Graph based methods

[5] models the problem as a weighted bipartite graph. Multiple approaches are applied such as Weighted Bipartitate Graph Projection, Clustered Weighted Bipartitate Graph Projection, Multi-Step Random Walk Weighted Bipartitate Graph Projection andCascaded Clustered Multi-step Weighted Bipartitate Graph Projection. The baseline is assumed to be bias-only latent factor model.

4.1.5 State-of-the-art methods

While [6] targets rating prediction by modeling the problem as a regression task using textual and non-textual features, [7] poses the problem as a a multi-class classification problem and uses only textual features. [6] uses techniques such as Decision Tree and Neural Networks, reporting the best accuracy achieved as 82.5%. [7] uses techniques such as Random Forest, Linear SVM and transformer-based deep learning models. XLNet, a transformer-based model is claimed to achieve an accuracy of 70.44%.
In our study, we recommend restaurants based on user-item restaurants which is in contrast to above techniques where the goal is to predict restaurant ratings relying on review text.

Table 1: Model Accuracies (%)

Model	Top-3	Precision@3	Top-5	Precision@5
Popularity Baseline	18.07	6.47	29.61	6.86
Rating/Review Baseline	19.80	7.46	31.48	7.34
Jaccard user-user similarity Baseline	11.11	2.37	12.49	2.56
User Similarity + Popularity/Ratings Baseline	16.37	5.86	16.62	5.85
MF	26.50	9.34	36.74	10.07
MF + categorical	29.45	11.27	40.23	10.38
BPR	62.11	27.69	72.29	26.34
BPR + categorical	**66.07**	**32.32**	**78.55**	**29.26**

4.2 Similar Works

4.2.1 Similar Datasets

Restaurant recommendation is an existing domain in machine learning and there are several datasets that aim to provide interactions between users and restaurants. Like Yelp, Zomato is a famous app, that hosts tremendous data on restaurants spanning several countries and has been used extensively for studying prediction and recommendation models. Studying the EDA on Zomato dataset [9], we discovered that it has several similar features like Yelp restaurant review dataset whilst providing other interesting aspects about users like income range for those who want to find the value for money restaurants in various parts of the country for the cuisines. There is also a New York Restaurant review dataset [10], extensively scraped on city of New York, but contains similar features like Yelp. Tripadvisor is also a popular dataset used for obtaining interactions between users and businesses with individual ratings.

4.2.2 Other Restaurant Recommender Systems

In restaurant recommendation, handling cold start problems is very important branch, in [11], Point-of-interest recommendation method has been deployed using location similarity, which assumes that people may be interested in the places that are similar with the places that they have been to before. Using textual features of reviews [12] proposes to use NLP on data from TripAdvisor for examining and tagging all the previous user's comments (whether positive or negative) for every restaurant, based on a particular user the corresponding restaurants are fetched and the user comments are examined to identify the restaurant with the highest ranking. Using BPR for item recommendations has also been extensively studied in [13] indicating that for the task of personalized ranking BPR outperforms the standard learning techniques for MF and kNN.

4.2.3 Other Recommender Systems

Needless to say recommendation is not just limited to restaurants but has been used a s a medium to improve user experience in several domains. In [14] a new hybrid hotel recommendation system that has been developed by combining content-based and collaborative filtering approaches that recommends customer the hotel they need and save them from time loss. For improving movie recommendations, Koren [15] proposed an algorithm that called SVD++ during the competition of Netflix Prize competition and they won the Netflix Prize by outperform Netflix's own recommending algorithm by 10%. One thing to note in all above mentioned works is utilization of rating prediction by other users for suggesting new restaurants which is not an effective measure of will the user visit a certain place or not as we have shown from our models or as we can understand by the example of Netflix discontinuing star rating recommendations stating that there's a difference between what you rate highly and what you actually like watching. The key difference here is that the ratings that pop up were measure of likelihood that we like the recommendations, but instead to many users it represented how other users rated it. There was much confusion around this aspect which led to Netflix abandoning star rating system towards to a more general thumbs up/down system which we've used in our restaurant recommendations in that aspect.

5 RESULTS

5.1 Performance Tables

As stated in section 2.2, we evaluate the baselines, MF (Matrix Factorization), and BPR (Bayesian Personalized Ranking) with respect to 4 metrics on a common test set.

The results summarized in Table 1. We can see the BPR with categorical restaurant features outperforms all the other models in all 4 evaluation metrics.

5.2 Result Interpretation

A Top-3 accuracy of 0.67 indicates that the restaurant visited by the user in test set is present in our top 3 recommendations 67% of the time, while a Top-5 accuracy 0.78 indicates that the same is present in our top 5 recommendations 78% of the times. Precision@3 of 0.32 indicates that each of the restaurants in our top 3 prediction has a probability of 32% being visited by the user while Precision@5 of 0.29 indicates the same for each of our top 5 predictions. We can see the BPR with categorical restaurant features outperforms all the other models in all 4 evaluation metrics.

5.3 Feature Representations

In the baselines, as we can see from Table 1, the accuracies are the lowest of all models studied. This shows that naive ways of recommendations can only perform upto a certain level and we need to consider personal interactions and include information about businesses to be able to deliver better results. One more interesting result when deriving Jaccard similarity baseline accuracy was that finding similar users has minimal impact on recommendations due to the sparse nature of similar users in one particular city. So much so, that even after trying to improve the model post inculcating popularity and rating sort, it didn't even better the popularity result, thus showing that user-user similarity was a bad choice to derive recommendations. In case of ratings/review baseline and popularity baseline, we were able to see study city wise results but these models performed at almost half the accuracy compared to the BPR model. Taking the SVD model into consideration, as we can understand from our results that since it uses the quintessential technique of predicting the user rating and recommending thereafter, it is not as effective as BPR technique which doesn't rely on user rating prediction to give recommendations. Hence this model solidifies our problem statement of relying on user interactions for recommendations and signifies the use of BPR model as described in sections that follow.

5.4 Success of BPR

In the baselines and MF models, we make use of ratings to predict the preference of a user towards restaurants. As seen earlier ratings are not a good prediction feature due to many reasons. In BPR, we do not make use of the rating data but instead, use the fact that users would give a higher score to the restaurants visited than the ones not visited. Hence BPR is able to capture the preferences of a user towards restaurants by learning latent features for both the user and restaurants in a common space. As seen in the performance table, BPR is able to capture this user preference better than any of the other models.

5.5 Interpreting BPR model parameters

BPR with restaurant categorical features has two sets of parameters per user - Y_u ,Y'_u and two parameters per restaurant - Y_i , Y_{if} and it aims to maximize the function $Y_u Y_i + Y'_u Y_{if}$ if user u has visited the restaurant i, or minimize the same function if user u has not visited the restaurant i. Y_u and Y_i are the latent features of the user and restaurant respectively which indicate compatibility of the user u with the restaurant i. So if the dot product between Y_u and Y_i is large, it indicates high compatibility. On the other hand the hidden feature Y'_u indicates the compatibility of the user i with the categorical features of the restaurant Y_{if}. So a large dot product between Y'_u and Y_{if} indicates a high compatibility between the user and the categorical features of the restaurant. For example if $Y'u$ for a particular user consists of [0, 0.99, 0, 0.001, -1.5, 0] and the categorical features Y_{if} are ['burgers', 'mexican', 'fine dining', ' cafes', 'dessert', 'bakeries'], we could say that the user prefers mexican food but is not a fan of desserts. Note that only Y_u ,Y_i and $Y'u$ are learned through back-propagation, while Y_{if} is extracted from the dataset and hence remains constant for a given restaurant.

6 CONCLUSION

From above results, we can conclude that ratings is not the most effective way for providing personalized recommendations and our results from BPR model indicate that user item interactions and categorical features of restaurants are more successful in providing apt recommendations.

6.1 Comparision to prior work

Our experiments indicate that in addition to user interaction data, features such as categorical information help improve the accuracy of the model built using just interaction data. This observation is in line with the conclusions of prior work. For instance, [6] talks about the information gain achieved from features such as categories in addition to other features such as review count which

we covered in our baseline models but not in the other models.

7 FUTURE WORK

As a part of future extension, we want to to explore the possibility to use Natural Language Processing on "Review" textual data of Yelp dataset. Similarly, we can use Computer Vision techniques to extract meaningful features from restaurant food and related images on visual data from "Photos" provided in Yelp dataset. Motivated by prior work and our baseline models, we look to deeply explore non-textual cues such as review count and average star rating for restaurants. Additionally, we would like to apply Machine Learning/Deep Learning based approaches on textual/non-textual features for this task.

References

[1] Link: `https://www.yelp.com/dataset`

[2] Link: Comparitive study of different approaches to recommendation on Yelp Dataset

[3] Link: Feature engineering and analysis of recommendation approaches on Yelp Dataset

[4] Link: ALS based rating prediction on Yelp Dataset

[5] Sawant, Sumedh, and Gina Pai. "Yelp food recommendation system." (2013).Graph-based approaches for Yelp food recommendation system

[6] Y. Chen and F. Xia, "Restaurants' Rating Prediction Using Yelp Dataset," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), 2020, pp. 113-117, doi: 10.1109/AEECA49918.2020.9213704.

[7] Liu, Zefang. "Yelp Review Rating Prediction: Machine Learning and Deep Learning Models." arXiv preprint arXiv:2012.06690 (2020).

[8] Link : Relationship between Users' Friends and review Patterns Kaggle Data Analysis

[9] Link : Exploratory Data Analysis of Zomato Restaurant data

[10] Link : New York Restaurant review dataset

[11] J. Zeng, Y. Li, F. Li, J. Wen and S. Hirokawa, "A Point-of-Interest Recommendation Method Using Location Similarity," 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2017, pp. 436-440, doi: 10.1109/IIAI-AAI.2017.122.

[12] R. M. Gomathi, P. Ajitha, G. H. S. Krishna and I. H. Pranay, "Restaurant Recommendation System for User Preference and Services Based on Rating and Amenities," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862048.

[13] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." arXiv preprint arXiv:1205.2618 (2012).

[14] B. B. Türker, R. Tugay, İ. Kızıl and Ş. Öğüdücü, "Hotel Recommendation System Based on User Profiles and Collaborative Filtering," 2019 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 601-606, doi: 10.1109/UBMK.2019.8907093.

[15] Y. Koren. Factorization meets the neighborhood/ a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.