ISSN 2278-3091

Volume 10, No.3, May - June 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse261032021.pdf

https://doi.org/10.30534/ijatcse/2021/261032021

Restaurant Recommendation System using Machine Learning

Ketan Mahajan[1], Varsha Joshi[2], Mohini Khedkar[3], Jacky Galani[4], Mayuri Kulkarni[5]
[1]SVKM Institute Of Technology, India, Ketan664464@gmail.com
[2] SVKM Institute Of Technology, India, Varshasharadjoshi@gmail.com
[3] SVKM Institute Of Technology, India, mohinikhedkar2@gmail.com
[4] SVKM Institute Of Technology, India, jackiegalani40@gmail.com
[5] SVKM Institute Of Technology, India, mayuridkulkarni@gmail.com

ABSTRACT

Nowadays a big challenge when going out to a new restaurant or cafe, people usually use websites or applications to look up nearby places and then choose one based on an average rating. But most of the time the average rating isn't enough to predict the quality or hygiene of the restaurant. Different people have different perspectives and priorities when evaluating a restaurant. Many online businesses now have implemented personalized recommendation systems which basically try to identify user preferences and then provide relevant products to enhance the users experience . In turn, users will be able to enjoy exploring what they might like with convenience and ease because of the recommendation results. Finding an ideal restaurant can be a struggle because the mainstream recommender apps have not yet adopted the personalized recommender approach. So we took up this challenge and we aim to build the prototype of a personalized recommender system that incorporates metadata which is basically the information provided by interactions of customers and restaurants online(reviews), which gives a pretty good idea of customers satisfaction and taste as well as features of the restaurant. This type of approach enhances user experience of finding a restaurant that suits their taste better. This paper has used a package called lightfm(the library of python for implementing popular recommendation algorithms) and the dataset from yelp. There are different methods of filtering the data, here we have used Hybrid filtering which is a combination of Content-based filtering (CBF) and Collaborative Filtering (CF). Since the results from Hybrid filtering are far more closer to accuracy than CBF or CF respectively. Then hybrid filtering gives results in the form of personalized recommendations for users after training and testing of the data

Key words : Restaurant recommendation system, Content-based filtering, Collaborative Filtering and Hybrid filtering

1. INTRODUCTION

Going out to a new restaurant is a big challenge faced by the people as nowadays there are a lot of restaurants and choosing the one which has good taste according to the needs of the person can be a nightmare. The opinions and feelings of the public about a restaurant's taste and hygiene greatly influences the user's opinion about the restaurant. Suppose a product has some copious negative reviews, it affects the user's opinion and trust regarding that product in a negative way. While exploring the available offers for a certain product, it is appreciated by the user having the possibility to access items that are generated by the Recommended System as it saves time as well as the money but the recommendation system should only consist of the products which suits the user's preference the most. In this paper we have implemented a recommendation system using hybrid filtering which is the combination of Content-based filtering (CBF) and collaborative filtering (CF). CBF models recommend based on the user's past behaviors and not from other users' data. If there is lack of enough information, the CBF will not be able to discriminate the items properly. It will not perform upto the standards. On the other hand, CF looks upon the user's interactions and it tries to recommend items that were similar to those items. In Cf, data sparsity problems occur because interactions of many users are insufficient. However, there are limitations to both methods. To avoid these, we have tried to use a hybrid approach that uses a combination of both methods to give sufficient results and we have also compared the results running the CF model with our Hybrid Model to see which performs the best**.**

2. RELATED WORK

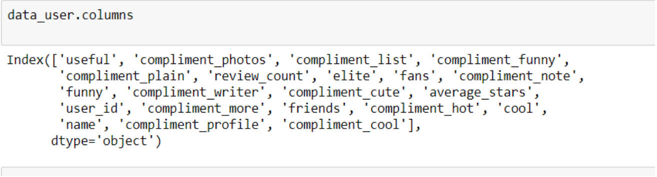
The available recommendation system utilizes techniques across various fields, such as machine learning, data mining, database, statistics, similarity testing, etc. It generates predictions of user satisfaction and/or recommends an item to a user. Generally, the recommendation system is

implemented using three main conceptual approaches.
Collaborative filtering,
Content-based filtering,
Hybrid approach.

Content-based filtering generates a prediction from attributes of an item that the user prefers. Collaborative filtering is an algorithm to generate a prediction using similarity of users' taste and preference. There are some critical problems in both concepts, such as limited content analysis, overspecialization, new user's problem, and sparsity problem. Limited content analysis is alimitation of content-based filtering that can only draw a conclusion from features that are explicitly associated with users. Over specialization is an overfitting problem where the recommendation system provides a high accuracy only on test data, but low accuracy on real data. New user problems arise from a lag in data to provide a new user with an accurate prediction. Several recommendation systems use machine learning techniques to reduce the impact of those problems. A hybrid approach combines the two concepts together authorized licensed use limited to. The hybrid recommendation system utilizes various approaches, such as combining separate recommenders, adding content-based characteristics to collaborative models, using multi-criteria, etc.

3. DATASET

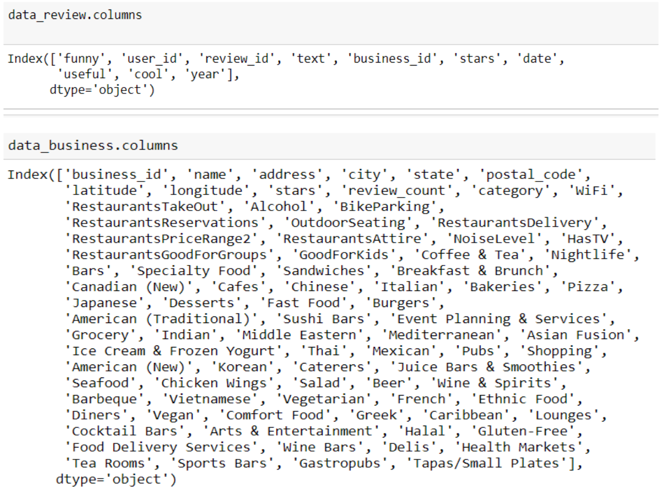
The dataset used for this project is taken from Yelp. In that we have three files that we converted to .csv viz users.csv, reviews,csv, Business.csv. Given below are the column names from each of the dataset.



```
data_user.columns

Index(['useful', 'compliment_photos', 'compliment_list', 'compliment_funny',
       'compliment_plain', 'review_count', 'elite', 'fans', 'compliment_note',
       'funny', 'compliment_writer', 'compliment_cute', 'average_stars',
       'user_id', 'compliment_more', 'friends', 'compliment_hot', 'cool',
       'name', 'compliment_profile', 'compliment_cool'],
      dtype='object')
```

Figure 1: User.csv



```
data_review.columns

Index(['funny', 'user_id', 'review_id', 'text', 'business_id', 'stars', 'date',
       'useful', 'cool', 'year'],
      dtype='object')
```

```
data_business.columns

Index(['business_id', 'name', 'address', 'city', 'state', 'postal_code',
       'latitude', 'longitude', 'stars', 'review_count', 'category', 'WiFi',
       'RestaurantsTakeOut', 'Alcohol', 'BikeParking',
       'RestaurantsReservations', 'OutdoorSeating', 'RestaurantsDelivery',
       'RestaurantsPriceRange2', 'RestaurantsAttire', 'NoiseLevel', 'HasTV',
       'RestaurantsGoodForGroups', 'GoodForKids', 'Coffee & Tea', 'Nightlife',
       'Bars', 'Specialty Food', 'Sandwiches', 'Breakfast & Brunch',
       'Canadian (New)', 'Cafes', 'Chinese', 'Italian', 'Bakeries', 'Pizza',
       'Japanese', 'Desserts', 'Fast Food', 'Burgers',
       'American (Traditional)', 'Sushi Bars', 'Event Planning & Services',
       'Grocery', 'Indian', 'Middle Eastern', 'Mediterranean', 'Asian Fusion',
       'Ice Cream & Frozen Yogurt', 'Thai', 'Mexican', 'Pubs', 'Shopping',
       'American (New)', 'Korean', 'Caterers', 'Juice Bars & Smoothies',
       'Seafood', 'Chicken Wings', 'Salad', 'Beer', 'Wine & Spirits',
       'Barbeque', 'Vietnamese', 'Vegetarian', 'French', 'Ethnic Food',
       'Diners', 'Vegan', 'Comfort Food', 'Greek', 'Caribbean', 'Lounges',
       'Cocktail Bars', 'Arts & Entertainment', 'Halal', 'Gluten-Free',
       'Food Delivery Services', 'Wine Bars', 'Delis', 'Health Markets',
       'Tea Rooms', 'Sports Bars', 'Gastropubs', 'Tapas/Small Plates'],
      dtype='object')
```

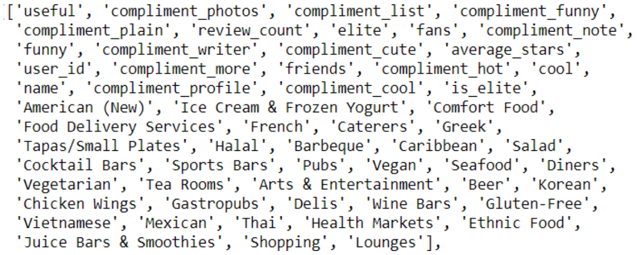
Figure 2: Reviews.csv

For more details, please check out the dataset. [2]

4. DATA PRE-PROCESSING

Now for building a recommendation system, we need an interaction matrix between users and items, metadata associated with customers that indicates their taste preference and metadata of restaurants that summarizes their characteristics.

There is business of all categories from over 100 cities in the business dataset. We decided to filter out the dataset only for one city because considering different cities means the items (restaurants) have less interactions with each other, So we selected Toronto which has 10,093 restaurants. Then we explored the restaurant attributes that would potentially be useful for recommendations. Then we explored the attributes of restaurants that can be useful for recommendations. We picked three attributes : 1. The rating of restaurants, 2. Review count of the restaurant, 3.Restaurant categories as item features since there are many features that have missing values. Usually Yelp has assigned an average of 10 categories/tags for each item(restaurant), and in total, across all restaurants, 436 tags exist such as breakfast, brunch, seafood, vegetarian, bars and so on. We selected upto 58 tags with highest popularities since including some tags that only appear a few times out of more than 10k restaurants would add more noise in the recommender.[3]



```
['useful', 'compliment_photos', 'compliment_list', 'compliment_funny',
 'compliment_plain', 'review_count', 'elite', 'fans', 'compliment_note',
 'funny', 'compliment_writer', 'compliment_cute', 'average_stars',
 'user_id', 'compliment_more', 'friends', 'compliment_hot', 'cool',
 'name', 'compliment_profile', 'compliment_cool', 'is_elite',
 'American (New)', 'Ice Cream & Frozen Yogurt', 'Comfort Food',
 'Food Delivery Services', 'French', 'Caterers', 'Greek',
 'Tapas/Small Plates', 'Halal', 'Barbeque', 'Caribbean', 'Salad',
 'Cocktail Bars', 'Sports Bars', 'Pubs', 'Vegan', 'Seafood', 'Diners',
 'Vegetarian', 'Tea Rooms', 'Arts & Entertainment', 'Beer', 'Korean',
 'Chicken Wings', 'Gastropubs', 'Delis', 'Wine Bars', 'Gluten-Free',
 'Vietnamese', 'Mexican', 'Thai', 'Health Markets', 'Ethnic Food',
 'Juice Bars & Smoothies', 'Shopping', 'Lounges'],
```

Figure 3: Data pre-processing

Then we calculated the term frequency-inverse document frequency (TF-IDF) values for each tag which would be used as weights in model fitting later.

$$\text{TF (i)} = \frac{\text{number of tag 'i' appear in each restaurant's tag list}}{\text{number of total tags of each restaurant}} = \frac{1}{\text{number of total tags of each restaurant}}$$

$$\text{IDF (i)} = \frac{\text{number of restaurants}}{\text{number of restaurant with tag 'i'}}$$

$$\text{TF - IDF (i)} = \text{TF (i)} \times \log[\text{ IDF (i) }]$$

Figure 4: TF-IDF

In the review dataset, there was some possibility that one user would rate one restaurant many times the history. To solve that we used a very recent review as it was the reflection of the latest preference of the user.

There were some cases where users give high ratings for each restaurant , in such cases we subtract ratings from mean ratings and classify the result as positive for 1 value and negative for -1 value and 0 for non-rated restaurants . In this research paper instead of predicting user rating for each item(restaurant) we focus on ranking which restaurant user liked and disliked in proper order , as all this will lead to high variance as time passes. So for the data cleansing step as the final step, we selected users characteristic . We chose 4 not sparse attribute, viz. The total written reviews , number of useful reviews, if the user is elite/active in Yelp and the list of liked restaurants of the user.[3]

- Evaluation Technique (AUC) :-

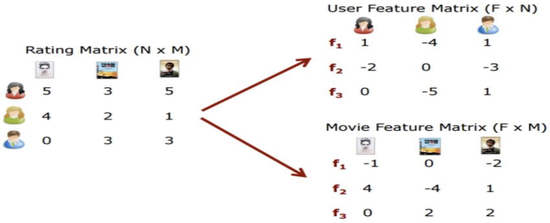
As per our case, the general idea is to figure out the customer's preference which is practical and more important than the prediction of rating for each restaurant. So, to evaluate recommenders, we arrived at the conclusion of using AUC(Area Under the Curve) over the more traditional technique used for measurement of the performance of explicit recommendation system which is the Root Mean Square Error. AUC has a metric to support decisions that checks only whether the item is preferred or not preferred by the user.

5. METHODOLOGY

I. So basically the algorithm is to recommend restaurants(items) that are high in ratings or are popular regardless of the feedback from users or item features. This is a useful method for new customers, in which case limited information of user/item is available to us. These types of scenarios are also called cold start scenarios. The sorting of restaurants was done by putting the number of reviews and ratings in descending order. And at random, for all customers, treated the top k items as a list of recommendations for the implementation of our model in Yelp. But while implementing this model the results infer that there is a 50% chance for a random user (AUC value turned out close to 0.5) to like the recommended restaurant. That concludes that this basic model unsurprisingly performs poorly.

II. Now, Lightfm package is used since it incorporates a matrix factorization model. Matrix factorization decomposes a matrix in two or more matrices such that when multiplying those matrices you get the original matrix.In the recommendation system, the typical starting point is a matrix of interaction/rating between users and items and matrix factorization algorithm. It will decompose this matrix into aitem and user feature matrix which is also known as embeddings. These embeddings have the same number of rows that are called latent vector dimensions but the number of columns is different depending on size of items or usersThe latent embeddings could secure the features about attributes of users and items, which also represents their taste. Let's take an example, users who like South Indian Restaurants would have similar embeddings with users who like North Indian Restaurants but won't resemble the embeddings of an Italian food in the vector space. Since the embeddings are estimated for every featur e. And the embeddings across all features sums up the representations for items and users.

III. Example of interaction matrix - user-movie ratings for movie recommender(refer figure given below),



	5	3	5
	4	2	1
	0	3	3

f_1	1	-4	1
f_2	-2	0	-3
f_3	0	-5	1

f_1	-1	0	-2
f_2	4	-4	1
f_3	0	2	2

Figure 5: Example

IV. For various structures, there are 4 loss functions that are available in the LightFM model. They are listed as follows: Logistics loss, BPR (Bayesian Personalized Ranking pairwise law), WRAP (Weighted approximate-Rank pairwise loss) and k-OS WRAP, which are specified accurately in the light FM package.

• Logistic loss: It is functional when both the negative (-1) and the positive (+1) interactions are available.

• BPR: Bayesian Personalized Ranking pairwise loss increases the prediction among a randomly chosen negative example along with a positive example. It is functional only when there are positive interactions available and enhances the ROC AUC is achieved.

• WRAP: Weighted Approximate-Rank pairwise loss maximizes the number of the positive examples by doing the sampling of the negative ones again and again until the rank violating is found to be one. It is useful when there aren't any negative interactions present. The positive interactions help in optimizing the top of the recommendation list to achieve the precision@k.

• k-OS WRAP : k-th order statistic loss modification of WRAP that uses the k-th positive examples for any available user as a ground for pairwise updates.

V. When comparing WRAP and BPR, generally, WRAP performs better than BPR. WRAP keeps negative sampling until there is a violation, that means if the rank is not violated, WRAP will take even more time to train. And as more iterations(epochs) are trained it becomes slower, because it tends to be problematic to find the violation. Therefore, setting a cut-off value for searching is necessary for training data WRAP loss. To know which loss gives the best result , we have to compare all 3 losses BPR,logistic and WRAP.

6. MODEL ESTABLISHMENT

For LightFM to achieve a Pure CF mode, it should fit the model in the interaction matrix. This gives the confirmation that only the interaction information is accessed by the model and no other metadata information. We splitted the data into train and test interactions in the ratio of 20:80. We ensured that the two sets were completely disjointed. After that, we initialized some random parameters to fit the model using different losses like logistic, BPR, WARP. As the parameters are randomly selected, it was required to do a hyperparameter search to get the best model. For that we used the scikit-optimizer package to look through a variety of values for various hyperparameters and judged them on the obtained average AUC score.

For Hybrid matrix factorization, user features and item features were fitted to the interaction matrix, with the dimensions 10093 x 10156(no. of items x no. of features). Lightfm creates unique features for every item (restaurant) therefore in the item feature matrix each row contains lightfm unique feature, ratings , sum of total number of reviews and tags (58). LightFm package implements a normalization process for each row to get each value in range of (0,1) , but it can create issues for features containing high values like review count . By considering this , we calculate log(feature with high value/max value of that feature) to normalize such features. Similarly we created a matrix of users and its feature (76367 x 76406). We also did regularization to maintain overfitting by adding item alpha and user alpha into the model .Then hyperparameter search to get the best model performed.

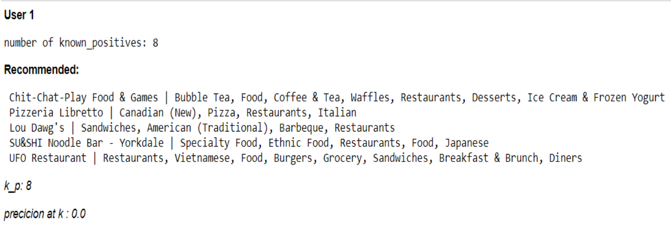
7. RESULT

I. AUC Result :-

In the figure given below, we get the result from the mean AUC score. Logistic loss did not perform well, WRAP outperforms both collaborative as well as in hybrid method, BPR looks fine for training data but performs badly for test data, it shows the problem of overfitting.

II. Demo :-

In the figure given below, restaurant recommendation for user 1 and user 17. The number of 'Known Positive' means the number of restaurant names that user is connected with, on that basis, it recommends 5 restaurants to each user.



User 1

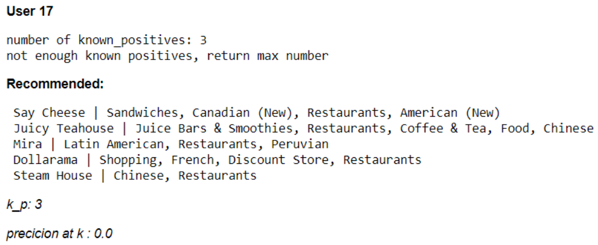
```
number of known_positives: 8
```

Recommended:

```
 Chit-Chat-Play Food & Games | Bubble Tea, Food, Coffee & Tea, Waffles, Restaurants, Desserts, Ice Cream & Frozen Yogurt
 Pizzeria Libretto | Canadian (New), Pizza, Restaurants, Italian
 Lou Dawg's | Sandwiches, American (Traditional), Barbeque, Restaurants
 SU&SHI Noodle Bar - Yorkdale | Specialty Food, Ethnic Food, Restaurants, Food, Japanese
 UFO Restaurant | Restaurants, Vietnamese, Food, Burgers, Grocery, Sandwiches, Breakfast & Brunch, Diners
```

k_p: 8

precicion at k : 0.0



User 17

```
number of known_positives: 3
not enough known positives, return max number
```

Recommended:

```
 Say Cheese | Sandwiches, Canadian (New), Restaurants, American (New)
 Juicy Teahouse | Juice Bars & Smoothies, Restaurants, Coffee & Tea, Food, Chinese
 Mira | Latin American, Restaurants, Peruvian
 Dollarama | Shopping, French, Discount Store, Restaurants
 Steam House | Chinese, Restaurants
```

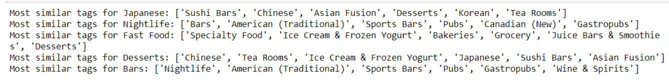
k_p: 3

precicion at k : 0.0

Figure 6: AUC Result.

III. Tag Similarity :-

LightFm package has its own feature it creates users and items features embedding that understand the similarity between tags. This makes this model highly efficient. Embeddings produced by LightFM encode important semantic information about features (tags).This is useful in the recommendation process to group the tags into categories . In fig tag shows its semantic similar tags.



```
Most similar tags for Japanese: ['Sushi Bars', 'Chinese', 'Asian Fusion', 'Desserts', 'Korean', 'Tea Rooms']
Most similar tags for Nightlife: ['Bars', 'American (Traditional)', 'Sports Bars', 'Pubs', 'Canadian (New)', 'Gastropubs']
Most similar tags for Fast Food: ['Specialty Food', 'Ice Cream & Frozen Yogurt', 'Bakeries', 'Grocery', 'Juice Bars & Smoothie
s', 'Desserts']
Most similar tags for Desserts: ['Chinese', 'Tea Rooms', 'Ice Cream & Frozen Yogurt', 'Japanese', 'Sushi Bars', 'Asian Fusion']
Most similar tags for Bars: ['Nightlife', 'American (Traditional)', 'Sports Bars', 'Pubs', 'Gastropubs', 'Wine & Spirits']
```

Figure 7: Tags.

8.CONCLUSION

In the proposed paper,it is a user preference restaurant recommendation system using yelp dataset and LightFM package. The goal of the paper is to give the best recommendation system by considering text-based review. Our study shows that hybrid filtering technique gives best performance as we compared performance of hybrid model with collaborative filtering model and different loss functions from LightFm package. Collaborative filtering uses just basic information between users and items. Hybrid filtering considers basic information as well as utilizes item & user metadata, which makes it perform better in the learning-to-rank setting and better for cold start problems.

REFERENCES

1. Mara-Renata Petrusel, Sergiu-George Limboi, **"A Restaurants Recommendation System: Improving Rating Predictions using Sentiment Analysis"**, 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing , SYNASC-2019.
2. https://www.kaggle.com/datafiniti/hotel-reviews
3. Asyush Singh, **Solving business usecases by recommender system using lightFM** of Towardsdatascience.com, 2018
4. Nanthaphat Koetphrom, Panachai Charusangvittaya, Daricha Sutivong, **"Comparing Filtering Techniques in Restaurant Recommendation System"**, Department

of Industrial Engineering Faculty of Engineering Chulalongkorn University Bangkok, Thailand, 2018..

5. R. M. Gomathi, P. Ajitha, G. Hari Satya Krishna, Harsha Pranay, **“Restaurant Recommendation System for User Preference and Services Based on Rating and Amenities”**, Second International Conference on Computational Intelligence in Data Science, ICCIDS-2019
6. P.Murugavel, Dr. M. Punithavalli, ***Improved Hybrid Clustering and Distance-based Technique for Outlier Removal***; IJCSE, 2011, Vol. 3.