ONLINE SOCIAL NETWORK ANALYSIS

FAIRNESS IN RECOMMENDATION SYSTEM

REPORT

Rajesh Kumar Bandaru (A20446254)
Venkata Manikanta Monic Kamisetty (A20446683)

1. Introduction:

A Recommendation System filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a user and based on that, recommends items that users might like. There exist many recommender systems that use collaborative filtering or content-based filtering approach to build a recommendation list and provide recommendations. Recommendation Systems have become ubiquitous nowadays, but they are not yet optimal.

Each type of Recommendation System has its own downsides. Collaborative Filtering takes into consideration just the ratings the user gave but not the additional (side) features that can improve the quality of the model. In this project, we consider one such side features 'category' to build a recommendation system. This recommendation system reflects most if not all user's interests.

The goal of the project is to build a recommendation system that takes 'category' as a side feature to give recommendations reflecting the user's interests.

Dataset

The data used for this project is the yelp hotel dataset taken from uic.edu. The data is in database format which contains the review given to a hotel, the reviewer who gave the review, and hotel information.

The primary dataset to generate the recommendations is the 'review' table containing the rating to a hotel and the user who gave it.

The secondary dataset is the 'hotel' table to generate calibrated recommendations.

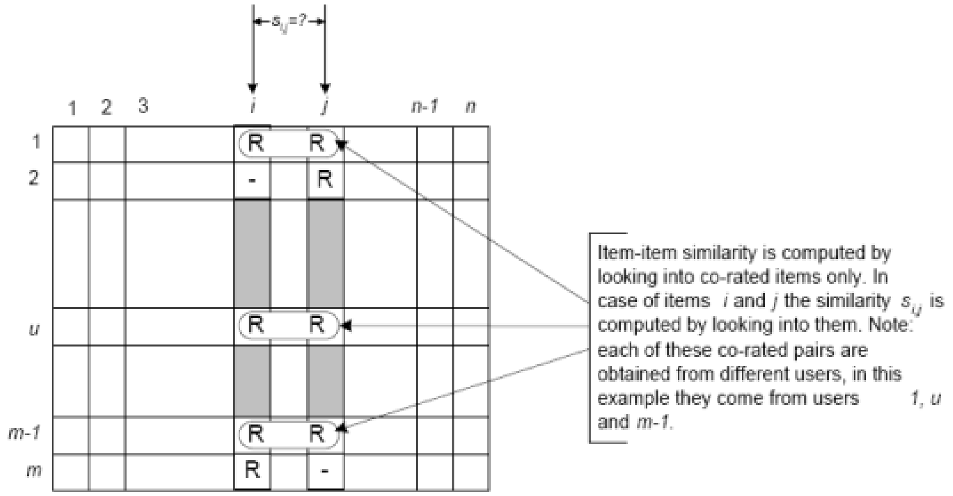
Algorithms and Techniques

For this project, we mainly focused on, Item-based Collaborative Filtering and KL-Divergence value.

Item-based Collaborative Filtering

Item-based collaborative filtering is a model-based algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset.

The similarity values between items are measured by observing **all the users who have rated both the items.** As shown in the diagram below, the similarity between the two items is dependent upon the ratings given to the items by users who rated both items.



Item-item similarity is computed by looking into co-rated items only. In case of items *i* and *j* the similarity $s_{i,j}$ is computed by looking into them. Note: each of these co-rated pairs are obtained from different users, in this example they come from users *1, u* and *m-1*.

There are different mathematical formulae that can be used to calculate the similarity of the items. The one used here is the Cosine-based similarity also known as the vector-based similarity.

Cosine-based Similarity

Cosine-based Similarity takes two items and their ratings as vectors and defines the similarity between them as the angle between these two vectors. The mathematical formula is as follows

$$\cos(\Theta) = \frac{A.B}{|A|\,.\,|B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sqrt{\sum_{i=1}^{n} B_i^2}}}$$

Calibrated Recommendations

A recommendation that actually reflects most if not all of the user's interest is considered as **Calibrated Recommendation.**

When a user has watched 70% action, 20% rom-com, and 10% thriller movies in the past. For a personalized experience, the list of recommended movies should consist of 70% action, 20% rom-com, and 10% thriller movies since all the user’s diverse set of interests are to be covered. Calibrated Recommendations is a post-processing procedure to get similar distribution over categories. These distributions are used to calculate KL-Divergence.

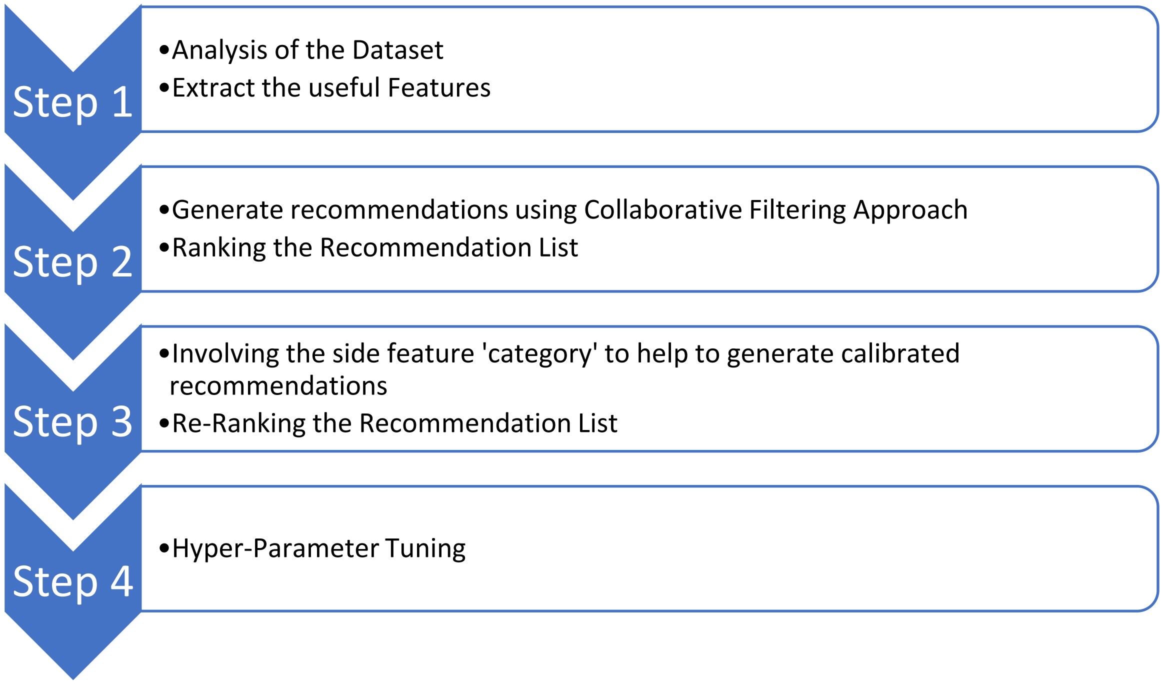
Kullback - Leibler Divergence

KL-Divergence or Relative Entropy is a measure of how one probability is different from the second.

The KL-Divergence value is used to re-rank the recommended list given by any of the filtering techniques.

2. **Experiment**

Project Flow



Step 1
- Analysis of the Dataset
- Extract the useful Features

Step 2
- Generate recommendations using Collaborative Filtering Approach
- Ranking the Recommendation List

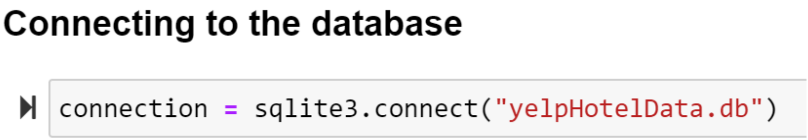
Step 3
- Involving the side feature 'category' to help to generate calibrated recommendations
- Re-Ranking the Recommendation List

Step 4
- Hyper-Parameter Tuning

- Step 1:
  The yelp hotel database is downloaded from the given link. A connection is made to the database from the workspace. A cursor object is created to fetch the tables.

Analysis

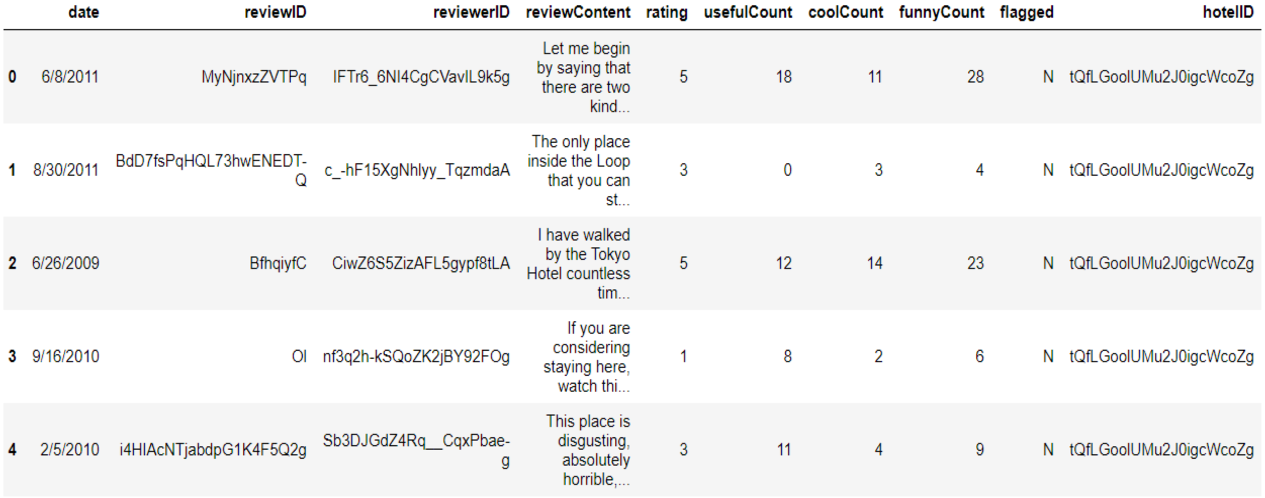
As the data needed for the project is in database format, the ‘sqlite3’ package is used to make a connection to it, and tables are converted into a data frame for further analysis.



Connecting to the database

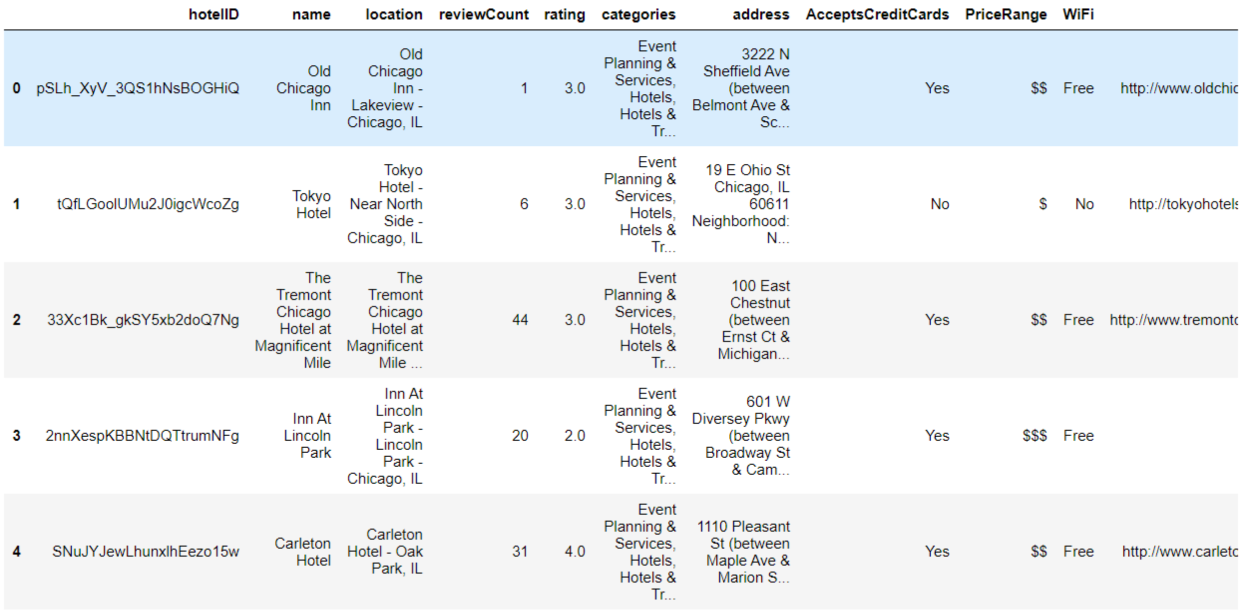
```
connection = sqlite3.connect("yelpHotelData.db")
```

Fig. Database Connection



	date	reviewID	reviewerID	reviewContent	rating	usefulCount	coolCount	funnyCount	flagged	hotelID
0	6/8/2011	MyNjnxzZVTPq	IFTr6_6NI4CgCVavIL9k5g	Let me begin by saying that there are two kind...	5	18	11	28	N	tQfLGoolUMu2J0igcWcoZg
1	8/30/2011	BdD7fsPqHQL73hwENEDT-Q	c_-hF15XgNhlyy_TqzmdaA	The only place inside the Loop that you can st...	3	0	3	4	N	tQfLGoolUMu2J0igcWcoZg
2	6/26/2009	BfhqiyfC	CiwZ6S5ZizAFL5gypf8tLA	I have walked by the Tokyo Hotel countless tim...	5	12	14	23	N	tQfLGoolUMu2J0igcWcoZg
3	9/16/2010	OI	nf3q2h-kSQoZK2jBY92FOg	If you are considering staying here, watch thi...	1	8	2	6	N	tQfLGoolUMu2J0igcWcoZg
4	2/5/2010	i4HIAcNTjabdpG1K4F5Q2g	Sb3DJGdZ4Rq__CqxPbae-g	This place is disgusting, absolutely horrible,...	3	11	4	9	N	tQfLGoolUMu2J0igcWcoZg

Fig. review table

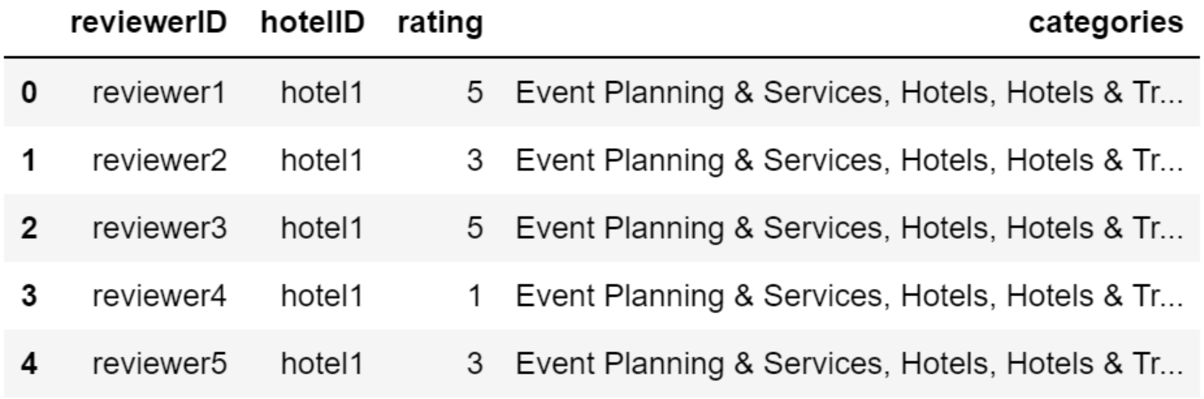


	hotelID	name	location	reviewCount	rating	categories	address	AcceptsCreditCards	PriceRange	WiFi	
0	pSLh_XyV_3QS1hNsBOGHiQ	Old Chicago Inn	Old Chicago Inn - Lakeview - Chicago, IL	1	3.0	Event Planning & Services, Hotels, Hotels & Tr...	3222 N Sheffield Ave (between Belmont Ave & Sc...	Yes	$$	Free	http://www.oldchic
1	tQfLGoolUMu2J0igcWcoZg	Tokyo Hotel	Tokyo Hotel - Near North Side - Chicago, IL	6	3.0	Event Planning & Services, Hotels, Hotels & Tr...	19 E Ohio St Chicago, IL 60611 Neighborhood: N...	No	$	No	http://tokyohotels
2	33Xc1Bk_gkSY5xb2doQ7Ng	The Tremont Chicago Hotel at Magnificent Mile	The Tremont Chicago Hotel at Magnificent Mile ...	44	3.0	Event Planning & Services, Hotels, Hotels & Tr...	100 East Chestnut (between Ernst Ct & Michigan...	Yes	$$	Free	http://www.tremontc
3	2nnXespKBBNtDQTtrumNFg	Inn At Lincoln Park	Inn At Lincoln Park - Lincoln Park - Chicago, IL	20	2.0	Event Planning & Services, Hotels, Hotels & Tr...	601 W Diversey Pkwy (between Broadway St & Cam...	Yes	$$$	Free	
4	SNuJYJewLhunxlhEezo15w	Carleton Hotel	Carleton Hotel - Oak Park, IL	31	4.0	Event Planning & Services, Hotels, Hotels & Tr...	1110 Pleasant St (between Maple Ave & Marion S...	Yes	$$	Free	http://www.carletc

Fig. hotel table

As hotel IDs and reviewer IDs are randomly generated strings, they are converted into simple terms.

Not every column is necessary for building a recommendation system, useful features are to be extracted from these tables. For **Collaborative Filtering**, 'hotelID', 'reviewerID', and 'rating' are required. For making **Calibrated Recommendations**, 'categories' are required. The Final features table is as follows



	reviewerID	hotelID	rating	categories
0	reviewer1	hotel1	5	Event Planning & Services, Hotels, Hotels & Tr...
1	reviewer2	hotel1	3	Event Planning & Services, Hotels, Hotels & Tr...
2	reviewer3	hotel1	5	Event Planning & Services, Hotels, Hotels & Tr...
3	reviewer4	hotel1	1	Event Planning & Services, Hotels, Hotels & Tr...
4	reviewer5	hotel1	3	Event Planning & Services, Hotels, Hotels & Tr...

Fig. Features

- Step 2

  An item-matrix is generated using the 'hotelID', 'reviewerID', and 'rating' from the 'review' table. The missing values i.e., the hotel's reviewer didn't rate are filled using the mean of their rated values. A cosine similarity matrix is constructed using the updated item matrix.

Ranking

For each reviewer, hotels only he/she visited/rated are taken. The similarity of those hotels with every other hotel is extracted from the cosine similarity matrix. The top 2 similarity scores (hotels) for each hotel the reviewer rated are taken. The final ranking is done by sorting this list.

- Step 3
  The side feature 'category' is involved in the process now to generate calibrated recommendations. The distributions over category for the user's history and user's recommended list are necessary to calculate the **KL-divergence**.

The distribution of categories c for each hotel h is as follows,

$$p(c|h)$$

Distribution over categories of user's history

$$p(c|u) = \sum_{i \,€\, R_H} p(c|h)$$

Distribution over categories of user's recommended list

$$q(c|u) = \sum_{i \,€\, R_P} p(c|h)$$

Using the above probabilities, KL - divergence value is calculated. The formula for KL – divergence is given by

$$C_{KL} = \sum_{c} p(c|u)\,.\,log\frac{p(c|u)}{q^{*}(c|u)}$$

If the denominator $q(c|u)$ tends to be zero, KL – divergence formula becomes invalid. To avoid this, $q(c|u)$ is replaced by $q^{*}(c|u)$.

$$q^{*}(c|u) = (1 - \alpha)\,.\,q(c|u) + \alpha\,.\,p(c|u)$$

To determine the re-ranked recommendation list,

$$I^* = \arg\max\,(1-\lambda)\,.\;s(I) - \;\lambda\,.\;C_{KL}$$

- Step 4
  The alpha α value from revamped $q(c|u)$ distribution and lambda λ value from the re-ranked recommended list I^* are the hyperparameters for calibrated recommendations.

A set of α and λ values are given to the respective functions to calculate the KL-divergence of the calibrated recommended list. The best values are selected which gives the lowest KL-divergence.

Problems

1. When reading the review table. A UTF-8 encoding error was raised for the *review content* column in the '**review**' table.

This problem is solved using a lambda function, decoding the columns by ignoring the errors.

2. How to rank the suggestions.

The problem is solved using 'Ranking' mentioned in the Experiment section.

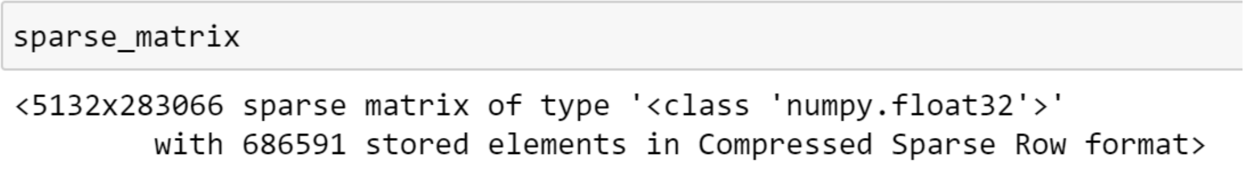
3. Because of the size of the data, multiple memory errors are raised.

This problem is solved by creating a sparse matrix for the item matrix. The sparse matrix is given to the bpr module to rank the recommendations.

3. Experiment and Results

After the data analysis, an item matrix is created for Collaborative Filtering. Due to memory errors, it is created in the sparse matrix format.

The sparse matrix for item matrix is as follows,

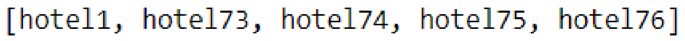


```
sparse_matrix
```

```
<5132x283066 sparse matrix of type '<class 'numpy.float32'>'
	with 686591 stored elements in Compressed Sparse Row format>
```

Fig. Sparse Matrix

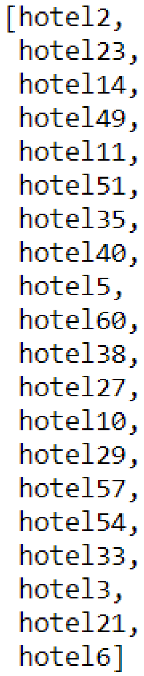
Here the analysis is done for reviewer 1, The hotels visited by reviewer 1 are as follows,



```
[hotel1, hotel73, hotel74, hotel75, hotel76]
```

Fig. Hotel visited by reviewer 1

Using the sparse matrix, recommendations are generated using the item-based Collaborative Filtering technique. The recommended hotels by Collaborative Filtering are

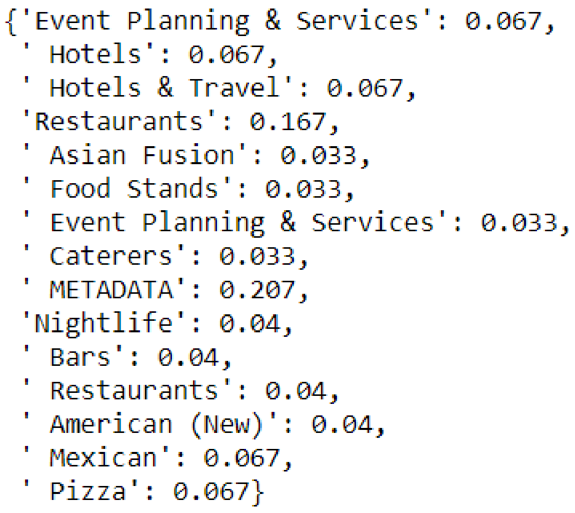


```
[hotel2,
 hotel23,
 hotel14,
 hotel49,
 hotel11,
 hotel51,
 hotel35,
 hotel40,
 hotel5,
 hotel60,
 hotel38,
 hotel27,
 hotel10,
 hotel29,
 hotel57,
 hotel54,
 hotel33,
 hotel3,
 hotel21,
 hotel6]
```

Fig. Hotels recommended to reviewer 1 by Collaborative Filtering

For Calibrated recommendations, the weights to which each hotel belongs is necessary to calculate KL-divergence.

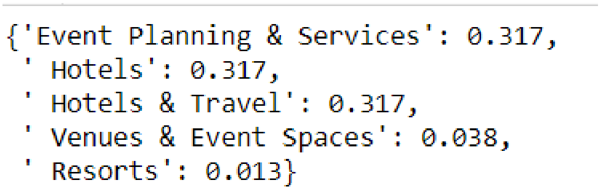
The weights for the visited hotel categories are as follows,



```
{'Event Planning & Services': 0.067,
 ' Hotels': 0.067,
 ' Hotels & Travel': 0.067,
 'Restaurants': 0.167,
 ' Asian Fusion': 0.033,
 ' Food Stands': 0.033,
 ' Event Planning & Services': 0.033,
 ' Caterers': 0.033,
 ' METADATA': 0.207,
 'Nightlife': 0.04,
 ' Bars': 0.04,
 ' Restaurants': 0.04,
 ' American (New)': 0.04,
 ' Mexican': 0.067,
 ' Pizza': 0.067}
```

Fig. Visited hotels weights

The weights for the recommended hotel categories are as follows,

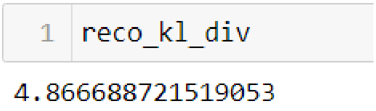


```
{'Event Planning & Services': 0.317,
 ' Hotels': 0.317,
 ' Hotels & Travel': 0.317,
 ' Venues & Event Spaces': 0.038,
 ' Resorts': 0.013}
```

Fig. Recommended hotels weights

Clearly, the weights of the recommended hotels are deviating from the visited hotels. This means the proportions of visited and recommended hotels are not the same.

The KL-divergence for these recommendations is,

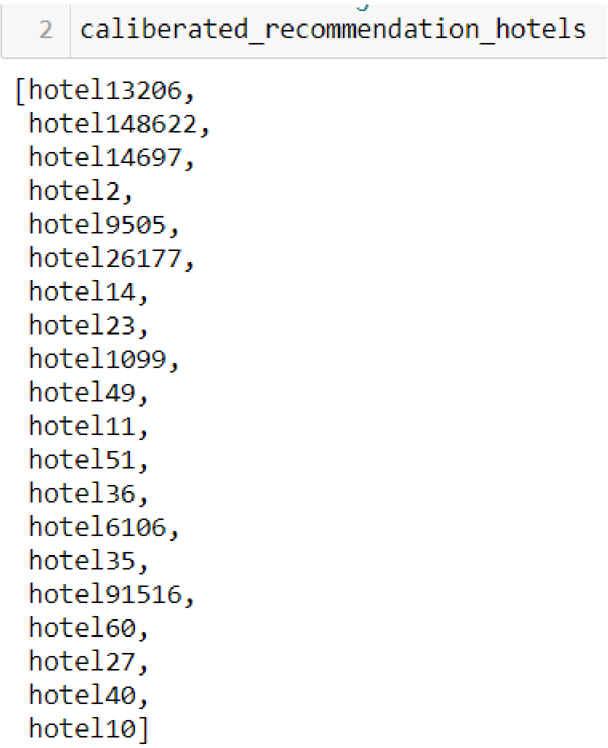


```
reco_kl_div
```

```
4.866688721519053
```

Fig. KL-divergence for Collaborative Filtering recommendations

Calibrated Recommendations are generated by making use of KL-Divergence above,

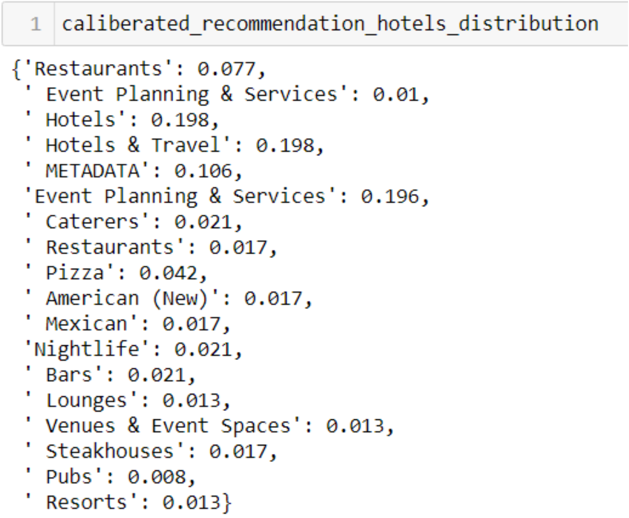


```
caliberated_recommendation_hotels
```

```
[hotel13206,
 hotel148622,
 hotel14697,
 hotel2,
 hotel9505,
 hotel26177,
 hotel14,
 hotel23,
 hotel1099,
 hotel49,
 hotel11,
 hotel51,
 hotel36,
 hotel6106,
 hotel35,
 hotel91516,
 hotel60,
 hotel27,
 hotel40,
 hotel10]
```

Fig. Calibrated Recommendations for reviewer 1

The category weights for the calibrated recommendations are as follows,

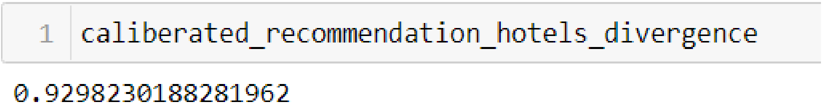


```
caliberated_recommendation_hotels_distribution
```

```
{'Restaurants': 0.077,
 ' Event Planning & Services': 0.01,
 ' Hotels': 0.198,
 ' Hotels & Travel': 0.198,
 ' METADATA': 0.106,
 'Event Planning & Services': 0.196,
 ' Caterers': 0.021,
 ' Restaurants': 0.017,
 ' Pizza': 0.042,
 ' American (New)': 0.017,
 ' Mexican': 0.017,
 'Nightlife': 0.021,
 ' Bars': 0.021,
 ' Lounges': 0.013,
 ' Venues & Event Spaces': 0.013,
 ' Steakhouses': 0.017,
 ' Pubs': 0.008,
 ' Resorts': 0.013}
```

Fig. Calibrated Recommendations weights for reviewer 1

The Kl-Divergence value for the Calibrated Recommendations is,

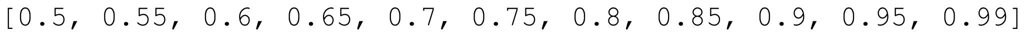


```
caliberated_recommendation_hotels_divergence
```

```
0.9298230188281962
```

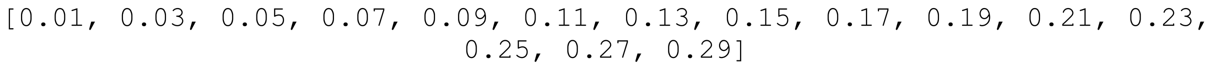
Fig. KL-Divergence for Calibrated Recommendations

As mentioned in the experiment section, α and λ are the hyperparameters for calibrated recommendations. Hyperparameter tuning is done a set of α and λ values.



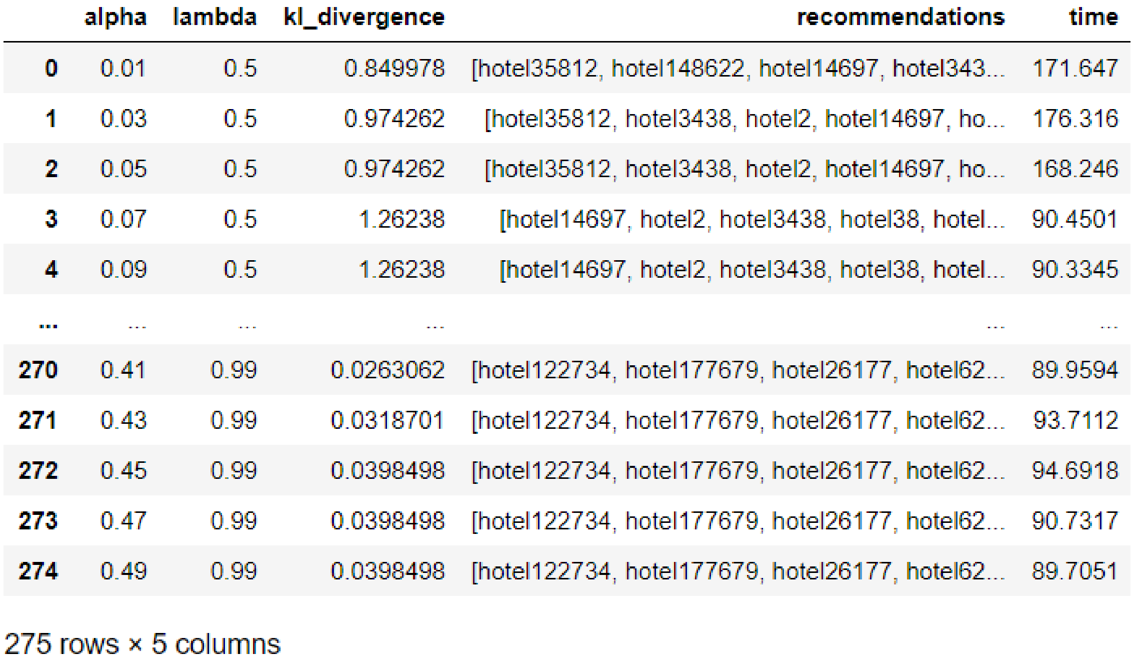
```
[0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.99]
```

Lambda values list



```
[0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23,
0.25, 0.27, 0.29]
```

Alpha values list



	alpha	lambda	kl_divergence	recommendations	time
0	0.01	0.5	0.849978	[hotel35812, hotel148622, hotel14697, hotel343...	171.647
1	0.03	0.5	0.974262	[hotel35812, hotel3438, hotel2, hotel14697, ho...	176.316
2	0.05	0.5	0.974262	[hotel35812, hotel3438, hotel2, hotel14697, ho...	168.246
3	0.07	0.5	1.26238	[hotel14697, hotel2, hotel3438, hotel38, hotel...	90.4501
4	0.09	0.5	1.26238	[hotel14697, hotel2, hotel3438, hotel38, hotel...	90.3345
...	...	...	...	...	...
270	0.41	0.99	0.0263062	[hotel122734, hotel177679, hotel26177, hotel62...	89.9594
271	0.43	0.99	0.0318701	[hotel122734, hotel177679, hotel26177, hotel62...	93.7112
272	0.45	0.99	0.0398498	[hotel122734, hotel177679, hotel26177, hotel62...	94.6918
273	0.47	0.99	0.0398498	[hotel122734, hotel177679, hotel26177, hotel62...	90.7317
274	0.49	0.99	0.0398498	[hotel122734, hotel177679, hotel26177, hotel62...	89.7051

275 rows × 5 columns

Fig. Hyperparameter Tuning Results

The variation in KL-divergence value for α and λ values are

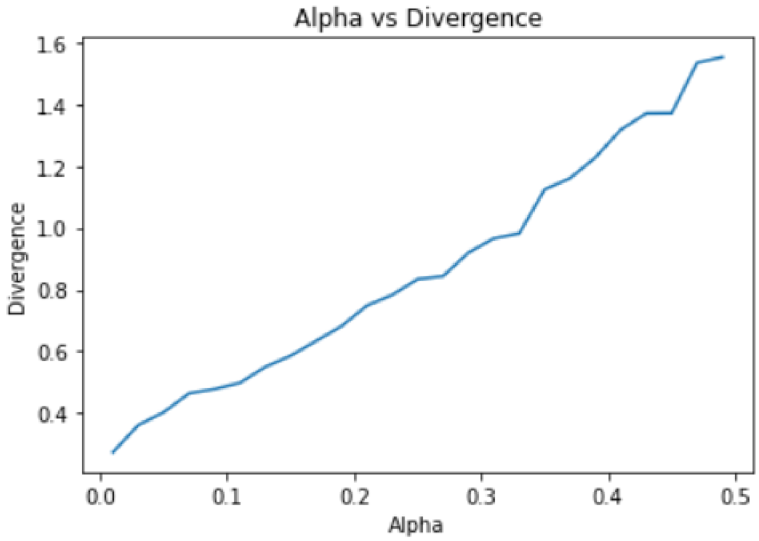


Fig. Alpha vs KL-Divergence

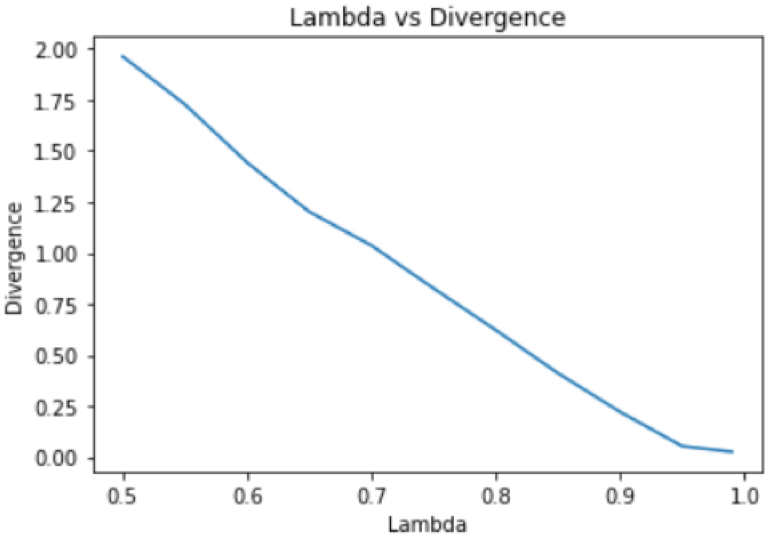


Fig. Lambda vs KL-Divergence

KL-Divergence is low for the lowest α value and highest λ value.

The KL-Divergence is 0.0085 for $\alpha = 0.01$ and $\lambda = 0.99$

4. Summary and Conclusion

Even though collaborative Filtering yields good results, it doesn’t cover all the interests of the user. It recommends the highest category of items the user visited/rated, which means it takes accuracy as its metric. But, Calibrated Recommendations takes all the interests of the user into consideration and gives results that don’t deviate from their interests. The KL-Divergence metric helps us in identifying the amount of deviation. We can conclude that the least KL-Divergence value is when the alpha value is lowest and the lambda value is highest.