RESTAURANT RECOMMENDATION SYSTEM

SUBMITTED BY:
RIYA GARG - MT22058
THANMAYEE MATHA - MT22084
KOMALJYOT KAUR - MT22105
ABHINAV GARG - MT22002
SAKSHAM NAUTIYAL - MT22061

Updated Problem Statement:

Design and implement a restaurant recommendation system by considering the user's past dining history and ratings, as well as the features of restaurants, such as location, cuisine, price range, ambiance, and ratings.
The system uses hybrid filtering which combines the results of collaborative filtering to identify similar users based on their past dining history and ratings, recommend restaurants that the user might like and content based filtering techniques to recommend restaurants that match the user's preferences based on the features of the restaurant such as location, cuisine, price range, ambiance, and ratings to provide personalized restaurant recommendations to users. The system also provides a user-friendly interface that allows users to easily search for and discover new restaurants based on their preferences and recommendations.
The system is developed based on the user reviews in Yelp Dataset which consists of various information about the restaurants including their menus, ratings, and reviews, as well as user preferences, including their cuisine preferences, price range, and other factors.

Literature Review:

By using the users' current geographical position, Md. Ahsan Habib et al. present a novel location, preference, and time based restaurant recommendation system. The
method evaluates user check-in accounts to investigate his visiting habits, meal preferences, and popular eateries. Four main factors are used to calculate recommendation scores are User preference score, Restaurants' separation , the hour of the day , the popularity ratings of the eatery[1].

In order to predict the customer satisfaction rating, Nanthaphat Koetphrom et al. present a method based on real data (connected to customer/restaurant features) and similarity in consumer preferences. The three filtering strategies being suggested are collaborative, content-based, and hybrid. The results show that the collaborative filtering approach uses cluster-based methodology to regulate information among its peers.[2]

The purpose of the study, according to Khushbu Jalan et al., is to recommend inn names to the explorer based on their interests and inclinations, using the feedback from other explorers and the rating as an incentive to increase prediction accuracy. The setting-aware cross-breed methodology is employed where CF method aggregates wistful research and provides personalized inn ideas. The use of a setting-based procedure then improves the proposal's results even more.[3]

The adaptive climate is used in the café recommender framework developed by Jun Zeng et al. The framework eventually gives suggestion outcomes based on the model after first building an inclination model for customers based on client/café area nuances and café visits. The contextual study also showed that the BMCS and BWCS-based café recommender system could effectively use the client's tendency.[4]

Ling Li et al. suggest three changes to the conventional UCF algorithm. The UCF algorithms' accuracy was quite poor because there were numerous factors that could affect a user's preference for a restaurant. Last but not least, actual private information of registered internet users is being used to gauge the similarity connected to user features.The results make it abundantly evident that the ACFmodified algorithm improves the accuracy of similarity computation and provides the user with a highly accurate restaurant suggestion. [5]

The study by Michelle Renee D. Ching et al., helps the restaurants to improve their customer satisfaction through analysing the text reviews of the customers by oversiving the business aspect of the reviews. This was done with the use of Aspect Based Sentiment Anaylsis(ABSA) endpoint of the AYLEIN Text Analysis API with which we can perform opinion mining.Time series forecasting using linear regression for one year with the use of the Weka machine learning workbench will be performed and conduct a linear regression with the one-year predicted data to understand its pattern to extract valuable information that will help in recommendation of business strategies.[6]

Boya Yu et al., proposed a recommendation system which uses the support vector machine(SVM) model to decipher the sentiment tendency of each review from word frequency.Word scores generated from the SVM models are further processed into a polarity index indicating the significance of each word for special types of restaurant.This method includes collecting keywords from different cuisines, our model can also be used for automatically generating ratings for tips (short reviews that are not accompanied with ratings) on Yelp by assigning weights to tips using the sentiment score of words and hence gives more reasonable overall ratings for restaurants.[7]

A multilingual recommender system based on sentiment analysis to help Algerian users decide on products, restaurants, movies and other services using online product reviews is prposed by Amel ZIANI et al., combines bothrecommendation system and sentiment analysis in order to generate the most accurate recommendations for users. This system detects the opinions polarity score using the semisupervised SVM.The results analysis evaluation provides interesting findings on the impact of integrating sentiment analysis into a recommendation technique based on collaborative filtering with very high precision and 100% recall.[8]

CODE SNIPPETS AND EXPLANATION:

1.1 DATASET:

The data we used comes from the following link: http://www.yelp.com/dataset_challenge. This data has been made available by Yelp for the purpose of the Yelp Dataset Challenge. In particular, the challenge dataset contains the following data:

- 1.6M reviews and 500K tips by 366K users for 61K businesses
- 481K business attributes, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of 2.9M social edges.
- Aggregated check-ins over time for each of the 61K businesses

1.2 DATA PREPROCESSING

Data originally is in json format . So we first converted it into the csv format.
Firstly in business .json we extracted a business having restaurant category as this is only needed in our recommendation and converted it into yelp_business_final.csv
Similarly we convert other files like review.json to yelp_review_final.csv having common business_id present in business.json.
Then we similarly also converted tip.json,checkin.json,user.json to csv files respectively.

```
[6]  data_business = pd.read_json('/content/yelp_dataset/yelp_academic_dataset_business.json', lines=True)
     data_business.fillna('NA', inplace=True)
     print('Final Shape: ',data_business.shape)

     Final Shape:  (150346, 14)
```

we only need restaurants data in business so we can remove other data . But we can consider above left data for sentiment analysis later.

```
[7]  data_business = data_business[data_business['categories'].str.contains('Restaurants')]
     print('Final Shape: ',data_business.shape)

     Final Shape:  (52268, 14)
```

```
csv_name = "yelp_business_final.csv"
data_business.to_csv(csv_name, index=False)
```

+ Code

```
csv_name = "yelp_review_final.csv"
data_review.to_csv(csv_name, index=False)
```

```
df = pd.read_csv("/content/yelp_business_final.csv")
df.head()
```

```
[ ]  tip_data = pd.read_json("/content/yelp_dataset/yelp_academic_dataset_tip.json", lines=True)
```

```
[ ]  tip_data.to_csv("yelp_tipdata.csv",index=False)
```

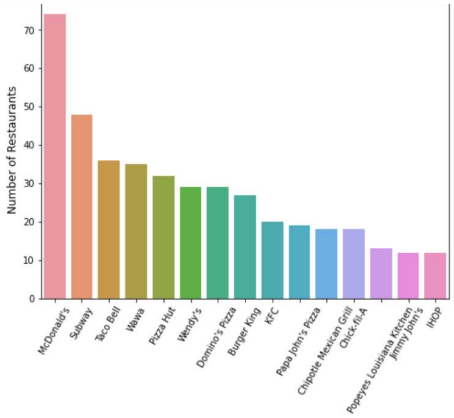
```
with open('/content/yelp_dataset/yelp_academic_dataset_user.json') as json_file:
    data = json_file.readlines()

    data = list(map(json.loads, data))
```

```
[ ]  user_data=pd.DataFrame(data)
```

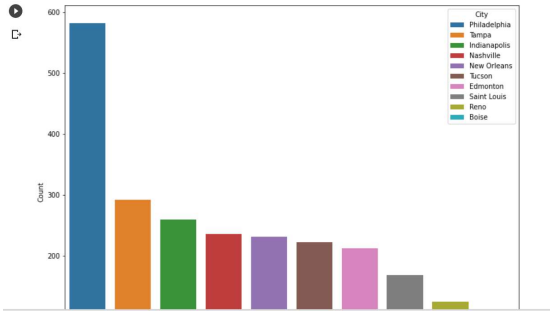
```
[ ]  user_df.to_csv("yelp_user.csv",index=False)
```

1.3 EXPLORATORY DATA ANALYSIS

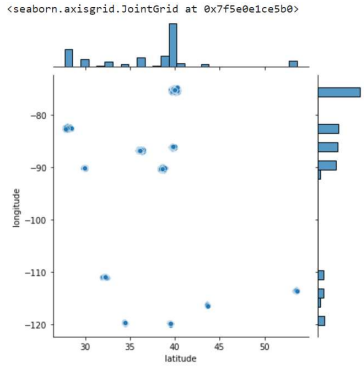
- Firstly we found 10 most popular restaurants in the dataset.



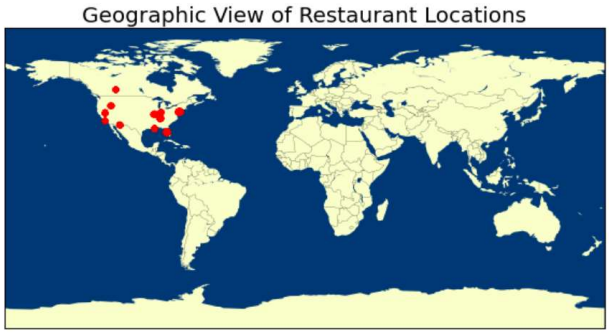
- We explored top 10 cities having maximum restaurants



- We see that locations of businesses are concentrated in clusters. These clusters must be big cities.



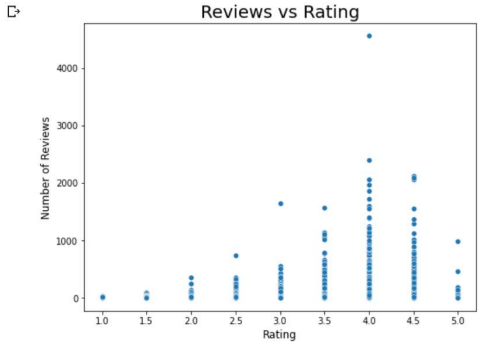
- We analyzed that our data has businesses from certain cities of U.S. and not all over U.S.



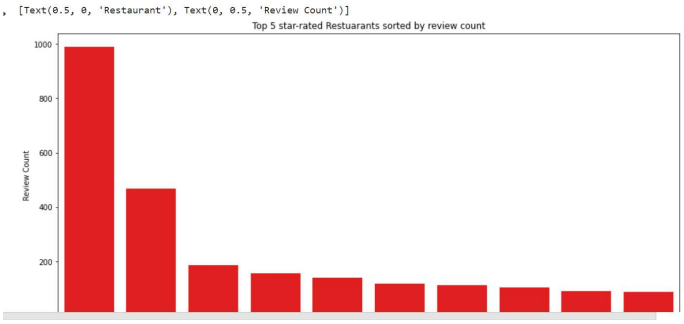
- Then we explored most reviewed food categories in business data

- Checked how rating and reviews are related to each other as these are important factors for restaurant recommendation.We can see that as the rating increases from 1.0 to 4.0, the number of reviews tends to

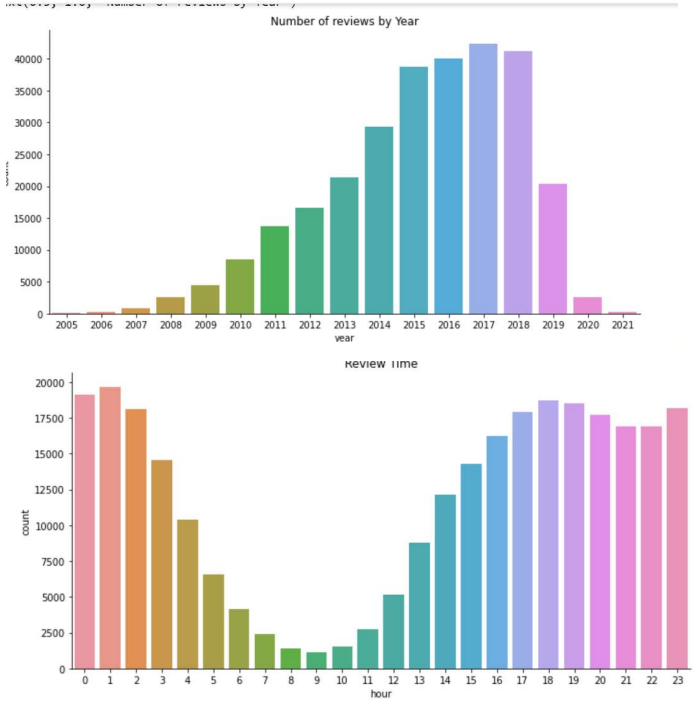
increases as well. However, as rating increases further, especially from 4.5 to 5.0, the number of review shrinks.



- We also found the top 10 5 star restaurants sorted by review count



- Restaurants with TakeOut, AcceptCreditCard, GoodForKids, Reservation, GoodForGroups, BusinessParking, HasTV, Alcohol, BikeParking, Delivery,, Attire are more popular
- We saw correlations roughly at the centre of heatmap in business data
- We also analyzed the no of reviews in a year and the most used time to give reviews.



- We found that no reviews recorded are minimum in the morning.
- Overtime users become less harsh reviewers.

1.4 BASELINE RESULTS:

1.4.1 We build a recommender system on item similarity based collaborative filtering. It recommends restaurants like the given restaurant id. For this we have used a KNN based model.

- For this we first calculate the total rating count of restaurants by merging the review and business dataset and on the basis of stars .
- Then we find the popularity threshold.
- Then we find the cities where total rating count >=popularity threshold. Out of these cities we select the top 10 cities. We find the subset of the dataset of restaurants present in these cities.
- Then we find restaurant features and form a csr matrix.
- We use this matrix features to fit our model. We used the cosine metric in knn.
- At last we got recommendations for a restaurant on a priority basis.

```
[ ] Restaurant_rating = pd.merge(bus_df, review_df, on='business_id')
    restaurant_ratingCount = (restaurant_rating.
         groupby(by = ['name'])['stars_x'].
         count().
         reset_index().
         rename(columns = {'stars_x': 'totalRatingCount'})
         [['name', 'totalRatingCount']]
        )
    restaurant_ratingCount.head()
```

	name	totalRatingCount
0	'feine	101
1	1 Stop Pizza	4
2	101 Taiwanese Cuisine	715
3	10th Street Diner	30

```
#joining above two tables
rating_with_totalRatingCount = restaurant_rating.merge(restaurant_ratingCount, left_on = 'name', right_on = 'name', how = 'left')
rating_with_totalRatingCount.head()
```

	business_id	name	city	stars_x	review_count	categories	user_id	stars_y	totalRatingCount
0	MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	Philadelphia	4.0	80	Restaurants, Food, Bubble Tea, Coffee & Tea, B...	6_SpY41LIHZulaiDs5FMKA	4	49
1	MTSW4McQd7CbVtyjqoe9mw	St Honore Pastries	Philadelphia	4.0	80	Restaurants, Food, Bubble Tea, Coffee & Tea, B...	tCXElwhzekJEH6QJe3xs7Q	4	49

```
[ ] populatity_threshold = rating_with_totalRatingCount['totalRatingCount'].quantile(0.90)
```

```
[ ] rating_popular_rest = rating_with_totalRatingCount.query('totalRatingCount >= @populatity_threshold')
    rating_popular_rest.shape
```

```
(28984, 9)
```

```
[ ] rating_popular_rest['city'].value_counts()
```

```
New Orleans          13137
Philadelphia          5559
Nashville             3395
Santa Barbara         1457
Tucson                1351
```

```
[ ] us_city_user_rating = rating_popular_rest[rating_popular_rest['city'].str.contains("New Orleans|Philadelphia|Nashville|Santa Barbara|Tucson|Columbus|
```

```
[ ]
```

```
[ ] us_city_user_rating
```

```
[ ]
    restaurant_features_matrix = csr_matrix(restaurant_features.values)
```

```
[ ] restaurant_features_matrix
```

```
<24x25792 sparse matrix of type '<class 'numpy.float64'>'
        with 27565 stored elements in Compressed Sparse Row format>
```

OUTPUT:

```
Recommendations for Restaurant Honey's Sit-N-Eat on priority basis:

1: El Camino Real
2: Village Whiskey
3: Jones
4: HipCityVeg
5: Han Dynasty
6: McDonald's
7: Mr. B's Bistro
8: Surrey's Café & Juice Bar
9: Luke
10: Three Muses
```

1.4.2 **Model to recommend the rating of the user.**

In this the dataset contains the user id, buisnessid ,and stars from the final merged dataset .

Then we split the data into train ,test and validation dataset.

The baseline model has the average mean ratings of all the users.

```
rating_df = df2[['user_id','business_id','stars']].copy()
```

```
rating_df.head()
```

	user_id	business_id	stars
0	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	3
1	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3
2	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	5
3	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	4
4	eUta8W_HdHMXPzLBBZhL1A	04UD14gamNjLY0IDYVhHJg	1

```
rating_df.shape
```

```
(283029, 3)
```

```
#splitting data
from sklearn.model_selection import train_test_split
X_train, X, y_train, y = train_test_split(rating_df.drop('stars', axis=1), rating_df.stars, train_size=.8)
X_test, X_val, y_test, y_val = train_test_split(X, y, train_size=.5)
del X, y

print(f"Train Size: {round(X_train.shape[0]/rating_df.shape[0]*100)}%")
print("X train shape: ", X_train.shape)
print("y train shape: ", y_train.shape)

print(f"\nValidation Size: {round(X_val.shape[0]/rating_df.shape[0]*100)}%")
print("X val   shape: ", X_val.shape)
print("y val   shape: ", y_val.shape)

print(f"\nTest Size: {round(X_test.shape[0]/rating_df.shape[0]*100)}%")
print("X test  shape: ", X_test.shape)
print("y test  shape: ", y_test.shape)
```

Average Baseline Accuracy Average model always predict average of all the ratings.

```
#Baseline accuracy
from sklearn.metrics import mean_squared_error as mse
mean_rating = y_train.mean()

train_baseline = mse(y_train, [mean_rating]*y_train.shape[0])
val_baseline = mse(y_val, [mean_rating]*y_val.shape[0])
test_baseline = mse(y_test, [mean_rating]*y_test.shape[0])


print(f"""Baseline MSE using mean rating:\n
          Train Data: {train_baseline:.4f},
          Val   Data: {val_baseline:.4f},
          Test  Data: {test_baseline:.4f}""")
```

OUTPUT:

```
Baseline MSE using mean rating:

          Train Data: 1.8027,
          Val   Data: 1.8055,
          Test  Data: 1.8005
```

Further work:

Here we have done recommendation according to user rating using Collaborative Filtering. Later we use content-based filtering techniques to recommend restaurants that match the user's preferences based on the features of the restaurant so that the hybrid collaborative filtering gives better accuracy compared to traditional collaborative and content based filtering techniques. Later we also apply sentimental analysis on user reviews and provide the

accurate recommendations to the users.Finally there will be a GUI where users can easily search for and discover restaurants according to their preferences and recommendations.

REFERENCES:

[1] Ahsan Habib, Abdur Rakib, and Muhammad Abul Hasan, “Location, Time, and Preference Aware Restaurant Recommendation Method”, IEEE, 19th International Conference on Computer and Information Technology, pp. 315-319, 2016.

[2] Nanthaphat Koetphrom, Panachai Charusangvittaya, Daricha Sutivong, “Comparing Filtering Techniques in Restaurant Recommendation System”, IEEE, pp. 46-51, 2018

[3] Khushbu Jalan, Kiran Gawande, “Context-Aware Hotel Recommendation System based on Hybrid Approach to Mitigate ColdStart-Problem”, IEEE, Communication, Data Analytics and Soft Computing, pp. 2364-2369, 2017.

[4] Jun Zeng, Feng Li, Haiyang Liu, Junhao Wen, Sachio Hirokawa, “A Restaurant Recommender System Based on User Preference and Location in Mobile Environment”, 5th IIAI International Congress on Advanced Applied Informatics, IEEE, pp. 55-60, 2016

[5] Ling Li, Ya, Zhou, Han Xiong, Cailin Hu, Xiafei Wei, “Collaborative Filtering based on User Attributes and User Ratings for Restaurant Recommendation”, IEEE, pp. 2592-2596, 2017.
[6Ching, M.R.D., De Dios Bulos, R.: Improving restaurants’ business performance using yelp data sets through sentiment analysis. In: ACM International Conference Proceeding Series, no. 2013, pp. 62–67 (2019
[7]Yu, B., Zhou, J., Zhang, Y., & Cao, Y. (2017). Identifying restaurant features via sentiment analysis on yelp reviews. *arXiv preprint arXiv:1709.08698*.
[8] Amel ZIANI1, Nabiha AZIZI2, Didier SCHWAB3, Monther ALDWAIRI4, Nassira CHEKKAI5, Djamel ZENAKHRA2, Soraya CHERIGUENE, "Recommender System Through Sentiment Analysis"