RESTAURANT RECOMMENDATION

Megala Jeyapal(5481906) | Yogalakshmi Saravanan(5497555)
megalajeyapal@Knights.ucf.edu | yogalakshmi@Knights.ucf.edu

Abstract

Restaurants are now an essential aspect of a significant section of the populace. Recommendations for restaurants based on user preferences is something that is highly sought after by consumers. With the rapidly expanding food industry, there are innumerable choices to pick from and a recommendation system for this use case can be utilized to a large extent by the end users. Therefore a recommendation system is proposed utilizing machine learning on the yelp business dataset. This report illustrates a detailed walkthrough of the implementation, the exploratory findings, results and the potential issues or future scope associated with this project.

1. Introduction:

1.1. Background:

The restaurant industry has become a rapidly evolving domain, with innumerable choices being offered to customers. As society shifts from home cooked meals to a more luxurious lifestyle, and more restaurants are being utilized. From the customer point of view the variety of options can be overwhelming and a tedious task. Customers are always seeking the fastest possible options. Instead of having to painstakingly look up restaurants that suit their tastes, a simple, quick and accurate system to predict their likings will go a long way. Instead of having to ask other people for recommendations, we can make use of the reviews already provided by customers along with other information, combine these inputs with the users' preferences and easily locate suggestions using machine learning.

A customer will likely have multiple factors they consider before looking at a restaurant, including the reviews of the restaurant, the price range, ambience, location etc. These opinions or considerations form each customer's unique preferences. Based on these preferences, from reviews of other customers, and from previous history a recommendation system can be constructed either through collaborative or content-based filtering. Through this recommendation system we can enhance the user experience and provide a novel feature for food-based applications.

1.2. Business Understanding and Objective:

The food business is changing through the growth of restaurant technology and online ordering. In one study, 56% of 1,000 Americans responded that they typically ate out in a restaurant at least 2 to 3 times per week. In this same study, 10% of participants said that they ate out 4 to 6 times a week, and 6% of people stated that they ate out for at least one meal every single day. 90% of visitors research a restaurant online before going and 33% of diners refuse to eat at a restaurant with less than four stars. Reviews are dire for reaching future consumers. Around 92% of diners choose to read the reviews for a restaurant they plan to attend. For many people, if these reviews fall below a certain expectation, they write the entire location off and move on to finding somewhere else. While the growth rate of the food industry as a whole tends to grow by a steady 3.7%, the growth of online ordering has demonstrated an annual growth rate of 15% to 20%. With these enormous options of restaurants, finding individualized and personalized restaurants to the customers based on their interest is a challenge.
In our project, we are exploring various methods in building an efficient Restaurant Recommendation system using Yelp Data Set. Our main objective is to recommend similar restaurants to Yelp users based on factors like the user's own interest and similar user's interest, past purchase history, product ratings, etc with the help of restaurant recommender system

For more information : [45+ Must-Know US Restaurant Industry Statistics [2022]: How Many Restaurants Are In The US – Zippia]

1.3. Recommender System:

Recommendation system predicts the user ratings that the user may give to a product, those predictions are ranked and used to give recommendations for the users. Recommendation systems are classified into three methods, 1. Content based methods, 2. Collaborative based methods, 3. Hybrid Methods

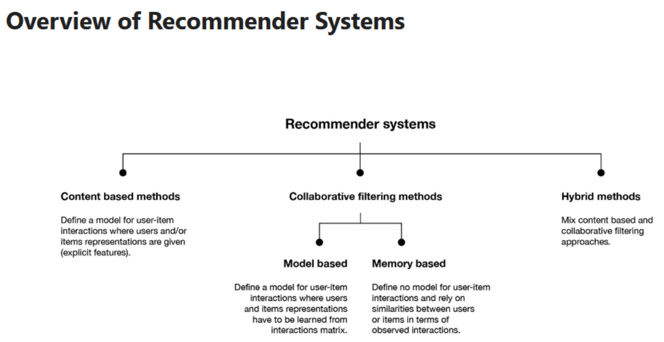

Overview of Recommender Systems
Recommender systems
Content based methods
Define a model for user-item interactions where users and/or items representations are given (explicit features).
Collaborative filtering methods
Model based
Define a model for user-item interactions where users and items representations have to be learned from interactions matrix.
Memory based
Define no model for user-item interactions and rely on similarities between users or items in terms of observed interactions.
Hybrid methods
Mix content based and collaborative filtering approaches.

Fig 1. Overview of Recommendation system

Collaborative filtering - recommends products based on similarity measures of like-minded people or items
Content based filtering - recommends products that have similar features to the products previously utilized by the user
Hybrid based filtering - combines both collaborative and content-based filtering to tackle the dynamic and ever-changing customer preferences.
Recommendation systems have economic benefits to businesses that offer the products or services and hence can play a vital role in bringing in a customer base. In our project, we construct both Content based and Collaborative based machine learning models.

1.4. Data Collection:

Yelp Dataset:

Yelp is the popular and comprehensive open-source dataset which is currently the most in-depth recording of restaurant's data. Yelp dataset contains a staggering amount of over 6.9 million restaurant reviews from 2 million users, for around 1.5 million different businesses and restaurants. Over 1.2 million business attributes like hours, parking, availability, and ambience.
Datasets we used for our recommender system are review.json(6.33 GB), user.json(3.27 GB), business.json(152.9 MB).

2. Components of Recommender system:

2.1. Pipeline:

Stage 1: Data Loading and Preprocessing – Creating new Database to loading the data files JSON format and converting to csv. Creating a workable dataset and then handling null values and datatypes. Applying basic preprocessing followed by feature engineering to cater to the use case.
Stage 2: EDA and Feature Selection - Exploring the relationships between the features in the datasets and selecting the features with the highest impact towards the use case.
Stage 3: Model implementation – Merging multiple datasets, Finding suitable models with the help of evaluations metrics, Train-Test data splitting, Parameter Tuning and Model fitting.
Stage 4: Experimental - Formulating possible future scope and expansion of this project.

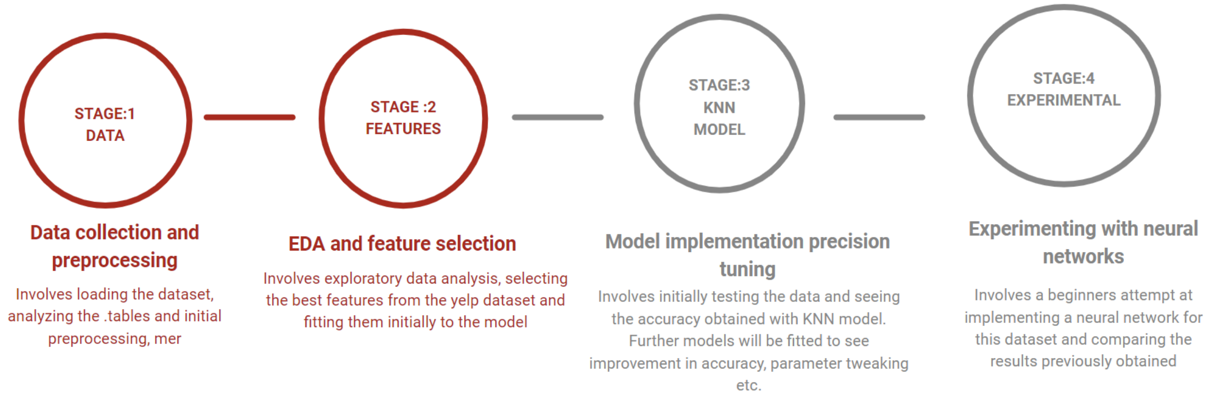
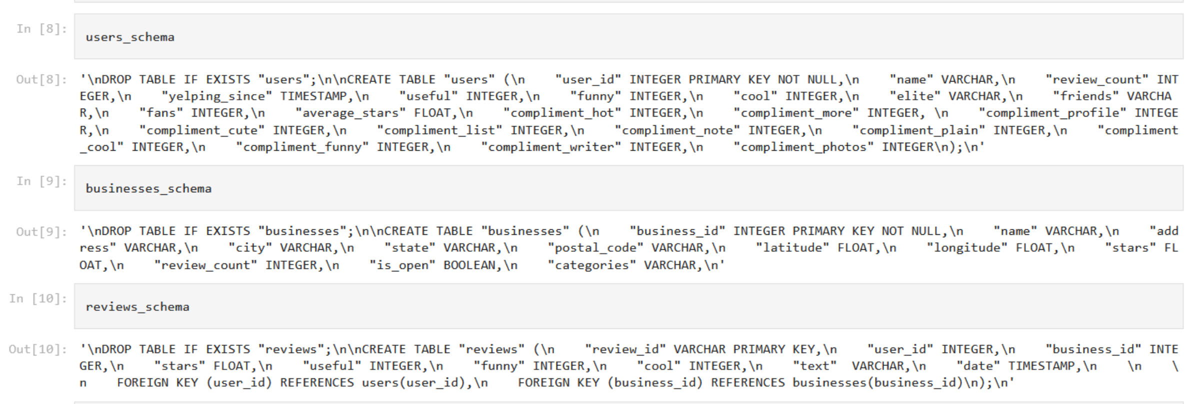

STAGE:1
DATA
Data collection and preprocessing
Involves loading the dataset, analyzing the .tables and initial preprocessing, mer
STAGE :2
FEATURES
EDA and feature selection
Involves exploratory data analysis, selecting the best features from the yelp dataset and fitting them initially to the model
STAGE:3
KNN
MODEL
Model implementation precision tuning
Involves initially testing the data and seeing the accuracy obtained with KNN model. Further models will be fitted to see improvement in accuracy, parameter tweaking etc.
STAGE:4
EXPERIMENTAL
Experimenting with neural networks
Involves a beginners attempt at implementing a neural network for this dataset and comparing the results previously obtained

Fig 2. Machine Learning Pipeline

2.2. Data Loading:

Yelp dataset size is too big to handle. So, we created a new database for Restaurant Recommendation dataset using SQLite3. Python provide Python SQLite3 interface, which is standardized DBI API 2.0 compliant interface for working with SQLite database. We opened database connection with sqlite3 for quick data loading. Figure 3 shows the table schema for User, Business and Review tables to be created in the database.



```
In [8]:  users_schema

Out[8]: '\nDROP TABLE IF EXISTS "users";\n\nCREATE TABLE "users" (\n    "user_id" INTEGER PRIMARY KEY NOT NULL,\n    "name" VARCHAR,\n    "review_count" INT
        EGER,\n    "yelping_since" TIMESTAMP,\n    "useful" INTEGER,\n    "funny" INTEGER,\n    "cool" INTEGER,\n    "elite" VARCHAR,\n    "friends" VARCHA
        R,\n    "fans" INTEGER,\n    "average_stars" FLOAT,\n    "compliment_hot" INTEGER,\n    "compliment_more" INTEGER, \n    "compliment_profile" INTEGE
        R,\n    "compliment_cute" INTEGER,\n    "compliment_list" INTEGER,\n    "compliment_note" INTEGER,\n    "compliment_plain" INTEGER,\n    "compliment
        _cool" INTEGER,\n    "compliment_funny" INTEGER,\n    "compliment_writer" INTEGER,\n    "compliment_photos" INTEGER\n);\n'

In [9]:  businesses_schema

Out[9]: '\nDROP TABLE IF EXISTS "businesses";\n\nCREATE TABLE "businesses" (\n    "business_id" INTEGER PRIMARY KEY NOT NULL,\n    "name" VARCHAR,\n    "add
        ress" VARCHAR,\n    "city" VARCHAR,\n    "state" VARCHAR,\n    "postal_code" VARCHAR,\n    "latitude" FLOAT,\n    "longitude" FLOAT,\n    "stars" FL
        OAT,\n    "review_count" INTEGER,\n    "is_open" BOOLEAN,\n    "categories" VARCHAR,\n'

In [10]: reviews_schema

Out[10]: '\nDROP TABLE IF EXISTS "reviews";\n\nCREATE TABLE "reviews" (\n    "review_id" VARCHAR PRIMARY KEY,\n    "user_id" INTEGER,\n    "business_id" INTE
         GER,\n    "stars" FLOAT,\n    "useful" INTEGER,\n    "funny" INTEGER,\n    "cool" INTEGER,\n    "text" VARCHAR,\n    "date" TIMESTAMP,\n    \n    \
         n    FOREIGN KEY (user_id) REFERENCES users(user_id),\n    FOREIGN KEY (business_id) REFERENCES businesses(business_id)\n);\n'
```

Fig 3. User, Business & Review Tables schema

2.2.1 Business data set:

Columns in business dataset: “business id”, “name”, “address”, “city”,“state”, “postal code”, “latitude”, “longitude”,“stars”, “review count”, “is open”, “attributes”,“categories”, “hours”. Figure 4 summarizes the Top restaurants in the yelp dataset with highest number of good reviews(ratings > 4) from users, “Blues City Deli” is the restaurant with higher number of user reviews. Figure 5 summarizes the restaurants with higher number of bad reviews(ratings<3) from the user, “Geno’s Steaks” is the least rated restaurant .

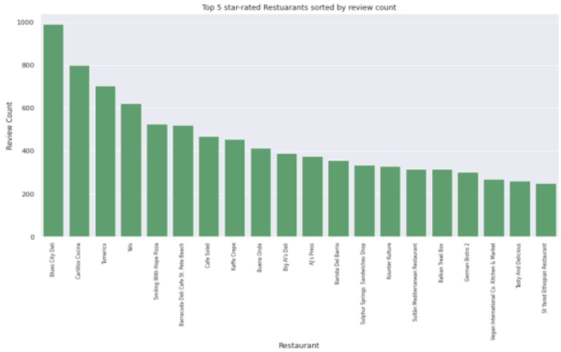


Fig4. Top Restaurants based on higher review count

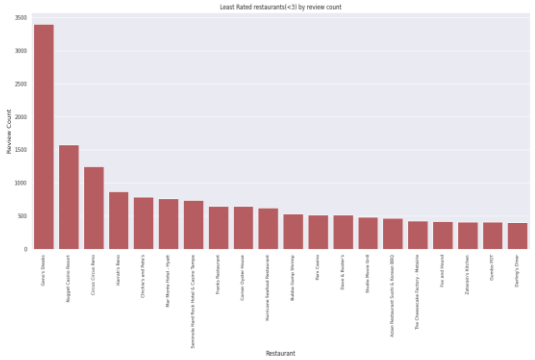


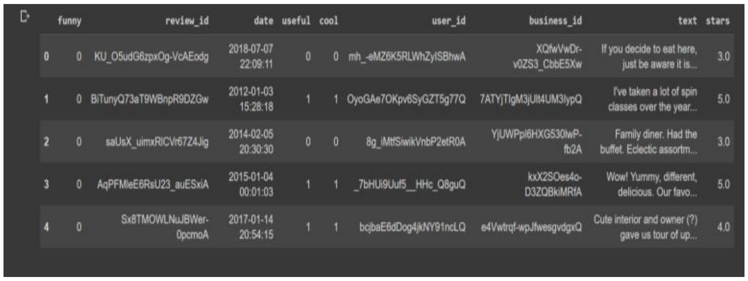
Fig 5. Restaurants with least review count

2.2.2 Review dataset:

Columns in Review dataset: “review id”, ”user id”, “business id”, “stars”, “useful”, “funny”, “cool”, “text” and ”date” columns. ‘stars’ and ‘text’ are the primary features in the Review dataset, provided by the users. ”user id” and “business id”, are common features between the tables, used for merging the datasets.

EDA is performed more in depth to find the weight of each feature and the unnecessary features are dropped or modified. The text feature is left untouched in this section and further analysis on that will be performed in later sections

Figure 6 shows the review data table after loading and Figure 7 summarizes the ‘useful’ column from the review table to show the number of useful reviews over a period of 2004 to 2018. Figure 8shows average rating vs restaurants, from the graph we can understand that FIVE star and ONE-star reviews are flagged as most useful.



	funny	review_id	date	useful	cool	user_id	business_id	text	stars
0	0	KU_O5udG6zpxOg-VcAEodg	2018-07-07 22:09:11	0	0	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	If you decide to eat here, just be aware it is...	3.0
1	0	BiTunyQ73aT9WBnpR9DZGw	2012-01-03 15:28:18	1	1	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	I've taken a lot of spin classes over the year...	5.0
2	0	saUsX_uimxRlCVr67Z4Jig	2014-02-05 20:30:30	0	0	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	Family diner. Had the buffet. Eclectic assortm...	3.0
3	0	AqPFMleE6RsU23_auESxiA	2015-01-04 00:01:03	1	1	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	Wow! Yummy, different, delicious. Our favo...	5.0
4	0	Sx8TMOWLNuJBWer-OpcmoA	2017-01-14 20:54:15	1	1	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	Cute interior and owner (?) gave us tour of up...	4.0

Fig. 6 Review dataset after loading

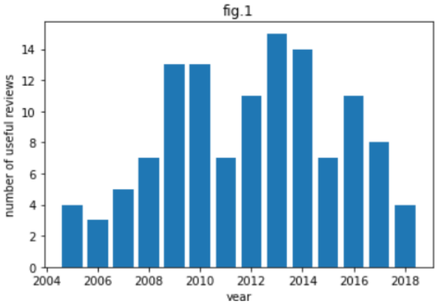


Fig. 7 Number of good reviews over years

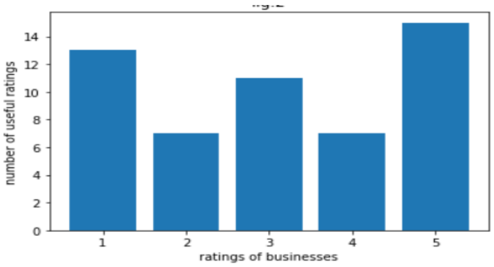


Fig 8 FIVE star and ONE star reviews are flagged as most useful

2.2.3 User dataset:

Columns in User dataset: 'user_id', 'name', 'review_count','yelping_since', 'useful','funny','cool', 'elite', 'friends', 'fans', 'average_stars', 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', ‘compliment_list','compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny','compliment_writer','compliment_photos'.

User dataset provides statistics of each user. From these statistics it is possible to obtain how influential a reviewer is. Extensive analysis has been performed to see the trends in these features. Columns like number of reviews, number of useful reviews, how funny, cool or impactful they were, number of fans each reviewer has etc.

For example fig.9 shows the number of useful reviews from the users, here Daniel and Jane gave more useful reviews, amongst five users selected.

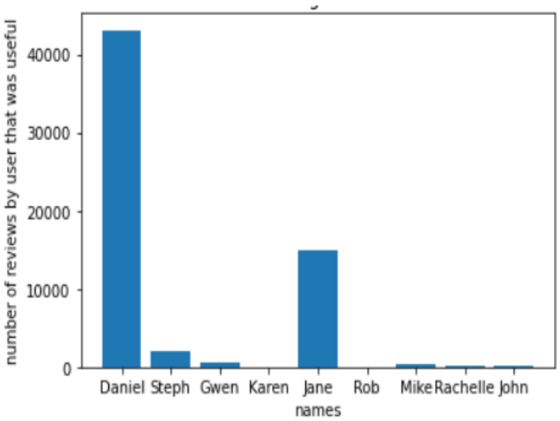
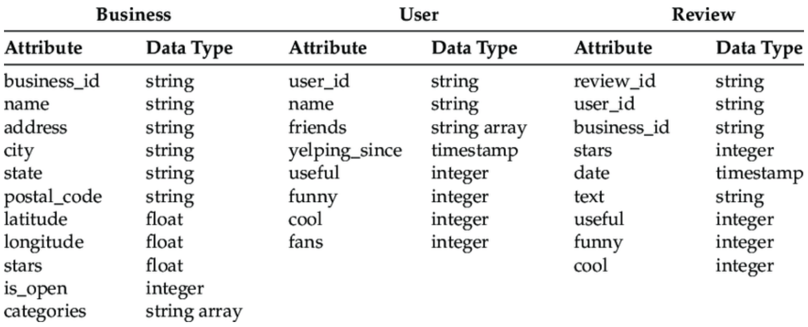


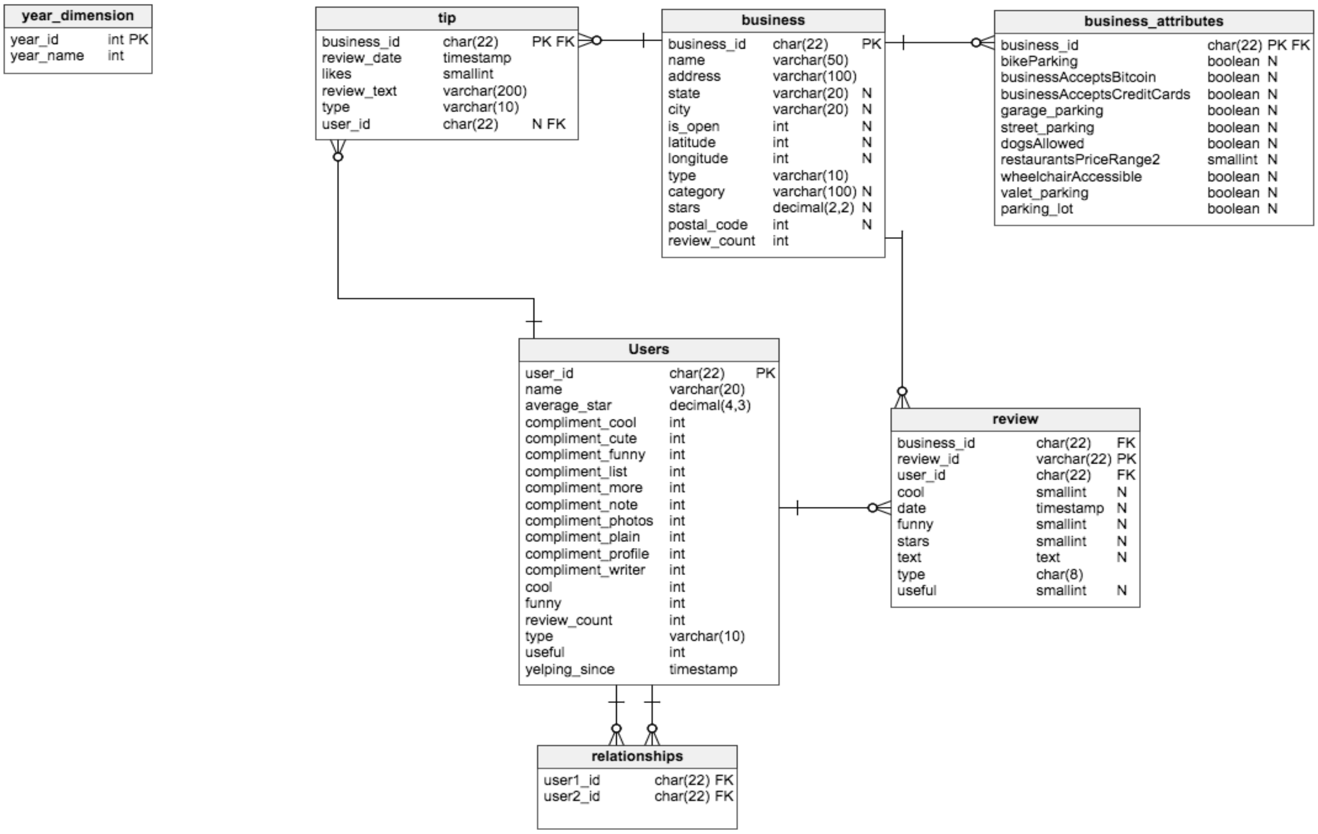
Fig 9 number of useful reviews

Figure 10 and 11 shows the in depth description of the features present in the three data tables and their relationship in the database.



Business		User		Review	
Attribute	Data Type	Attribute	Data Type	Attribute	Data Type
business_id	string	user_id	string	review_id	string
name	string	name	string	user_id	string
address	string	friends	string array	business_id	string
city	string	yelping_since	timestamp	stars	integer
state	string	useful	integer	date	timestamp
postal_code	string	funny	integer	text	string
latitude	float	cool	integer	useful	integer
longitude	float	fans	integer	funny	integer
stars	float			cool	integer
is_open	integer				
categories	string array				

Fig. 10 Description of the features and datatypes in business user and review csv files



year_dimension

year_id	int PK
year_name	int

tip

business_id	char(22)	PK FK
review_date	timestamp	
likes	smallint	
review_text	varchar(200)	
type	varchar(10)	
user_id	char(22)	N FK

business

business_id	char(22)	PK
name	varchar(50)	
address	varchar(100)	
state	varchar(20)	N
city	varchar(20)	N
is_open	int	N
latitude	int	N
longitude	int	N
type	varchar(10)	
category	varchar(100)	N
stars	decimal(2,2)	N
postal_code	int	N
review_count	int	

business_attributes

business_id	char(22)	PK FK
bikeParking	boolean	N
businessAcceptsBitcoin	boolean	N
businessAcceptsCreditCards	boolean	N
garage_parking	boolean	N
street_parking	boolean	N
dogsAllowed	boolean	N
restaurantsPriceRange2	smallint	N
wheelchairAccessible	boolean	N
valet_parking	boolean	N
parking_lot	boolean	N

Users

user_id	char(22)	PK
name	varchar(20)	
average_star	decimal(4,3)	
compliment_cool	int	
compliment_cute	int	
compliment_funny	int	
compliment_list	int	
compliment_more	int	
compliment_note	int	
compliment_photos	int	
compliment_plain	int	
compliment_profile	int	
compliment_writer	int	
cool	int	
funny	int	
review_count	int	
type	varchar(10)	
useful	int	
yelping_since	timestamp	

review

business_id	char(22)	FK
review_id	varchar(22)	PK
user_id	char(22)	FK
cool	smallint	N
date	timestamp	N
funny	smallint	N
stars	smallint	N
text	text	N
type	char(8)	
useful	smallint	N

relationships

user1_id	char(22) FK
user2_id	char(22) FK

Fig.11 Data Schema of the data set used to feed the models with their specific features

3. Data Preprocessing and Feature Engineering:

3.1 Business data:

Yelp Business dataset has a lot of business information other than Restaurants, so we filtered out ‘categories’ fields with Restaurant. For filtering out restaurants alone,we created a new column name ‘RESTAURANT and marked ‘True’ when the ‘categories’ column contains ‘restaurant’ value.

For initial data analysis we focused only on restaurants. Since the original dataset is so large, we used limited columns “business id”, “name”, “stars”, “review count”, “attributes”,“categories”, from the business dataset to use in this project.

“Attributes” column contains more detailed information about the restaurants e.g. Parking information, Wifi, Accepts Credit cards, Delivery information etc. So we splitted Attributes column to 39 columns of features of Boolean variables and plotted for better understanding of data pattern. Figure 10 proves that restaurants with wifi connection have better ratings than the other. Figure 11 proves that restaurants with business parking have better ratings.

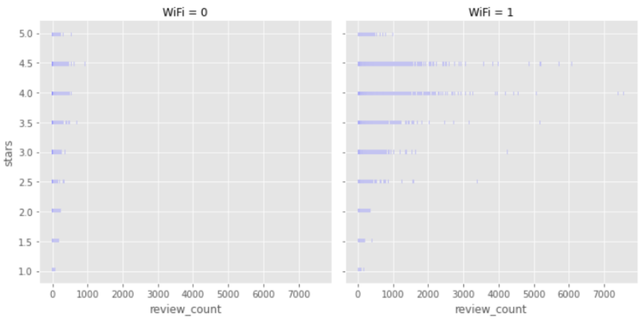


Fig 10 Restaurants with WIFI have higher review count

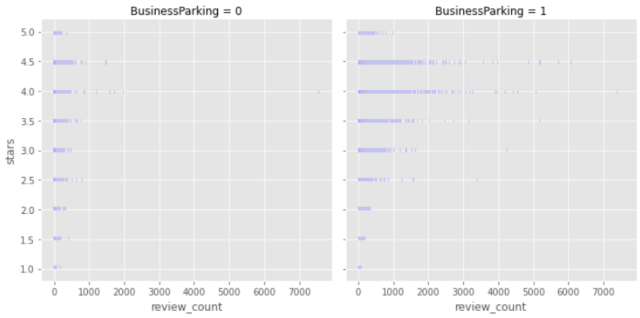
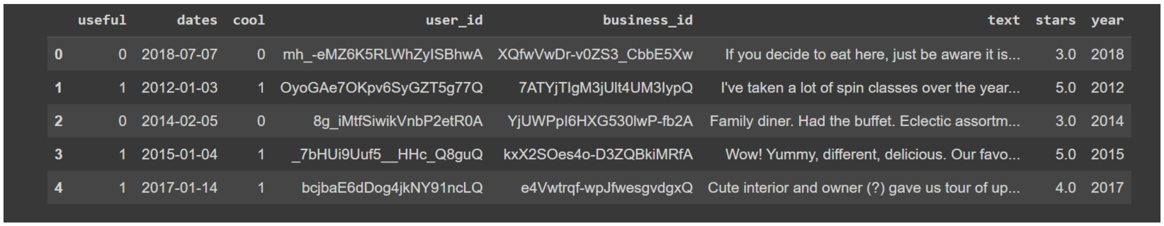


Fig 11 Restaurants with BusinessParking have higher review count

3.2 Review data:

'funny' and 'review_id' columns are irrelevant to our analysis so dropped. Created new column name 'year' retrieved from 'dates' columns and dropped 'dates' column with timestamp as it is not necessary for further analysis. There are no NULL values. Figure 12 shows the format of the Review dataset after preprocessing.



	useful	dates	cool	user_id	business_id	text	stars	year
0	0	2018-07-07	0	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	If you decide to eat here, just be aware it is...	3.0	2018
1	1	2012-01-03	1	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	I've taken a lot of spin classes over the year...	5.0	2012
2	0	2014-02-05	0	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	Family diner. Had the buffet. Eclectic assortm...	3.0	2014
3	1	2015-01-04	1	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	Wow! Yummy, different, delicious. Our favo...	5.0	2015
4	1	2017-01-14	1	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	Cute interior and owner (?) gave us tour of up...	4.0	2017

Fig. 12 review dataset after initial preprocessing

3.3 User data:

There are no null values in the User dataset except the 'elite' feature. So the 'elite' column was dropped as it is not useful for our analysis.

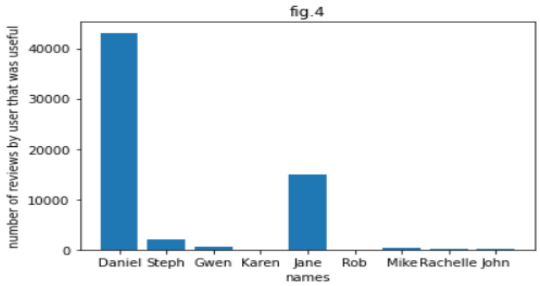


Fig 13 Daniel and Jane gave more useful reviews

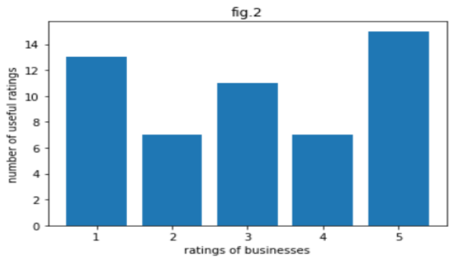
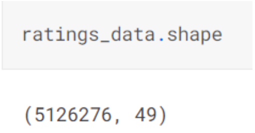


Fig 14 FIVE star and ONE star reviews are flagged as most useful

4.Exploratory Data Analysis:

4.1 Merging datasets:

'business_id' and 'stars' are common features for Review and Business datasets. So we used 'business_id' to merge the Business file with the Review file and created a new pandas dataframe called 'Ratings_data'. 'user_id' and 'Useful' are common features for Review and User datasets. So we used 'user_id' to merge Ratings_data and User data table. After merging all values to Ratings_data replace all missing values. In order to find the optimal data for training, filtered the merged dataset to only have users who gave more than 20 reviews, which results in data size as below:



```
ratings_data.shape
```

```
(5126276, 49)
```

Since the merged dataset is again larger than RAM size to handle, so filtered out 1000 users data.

4.2. Train-Test data splitting:

Merged dataset ‘Ratings_data’ split into two sets as Training data(75%) and Testing data(25%). Training data is used for Parameter tuning and model training. Finally tested the model predictions and performances with Testing data

5. Model Building and Performance Evaluation :

5.1 Technical Specifications :

Recommender system code is written in python programming language, and used standard python libraries such as

- numpy – To work with multi-dimensional arrays.
- Pandas – For data manipulation and analysis.
- scikit learn – With the scikit learn library, we can straight away use multiple machine learning algorithms.
- Surprise – Surprise library helps to build rating based recommender system
- SQLite3 – this library helps to create connections with sqlite databases for data loading.
- matplotlib and seaborn - These libraries are utilized for the data preprocessing visualization and EDA.

Online coding environments like kaggle kernels and google colab were used for our project development which offers GPU access for free. Using the available resources, we were able to execute the code but due to the massive dataset the process was slow. A system with better GPU and processing would be able to handle this dataset much better.

5.2 Evaluation Metrics :

We used Root Mean Squared Error(RMSE) and Mean Absolute Error(MAE) to evaluate our model. These two metrics help us to find how accurate our model predictions are and their deviations from actual values. RMSE and MAE both are average of error which are calculated as follows,

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

$$\text{MAE} = \frac{|(y_i - y_p)|}{n}$$

y_i = actual value
y_p= predicted value
n = number of observations/rows

Lower value of MAE and RMSE means that the model makes less errors and also the model learned latent features of restaurants and users.

Different Model performances evaluated with the value of RMSE and MAE. Model with least RMSE and MAE is selected for our recommendation system for more accurate predictions.

5.3 KNN inspired algorithms :

5.3.1. KNNBasic algorithm with Cosine Similarity based prediction:

For prediction problem, rating r = f(u,v), where u is the feature vector for Users and v is the feature vector Restaurants/items. we expect that the final rating will be higher if the type of restaurant matches with user’s interests, so we can say that the ratings prediction is a weighted combination of the user-restaurant “similarity” measure.

KNNBasic is a basic collaborative filtering algorithm. Prediction formulated as,

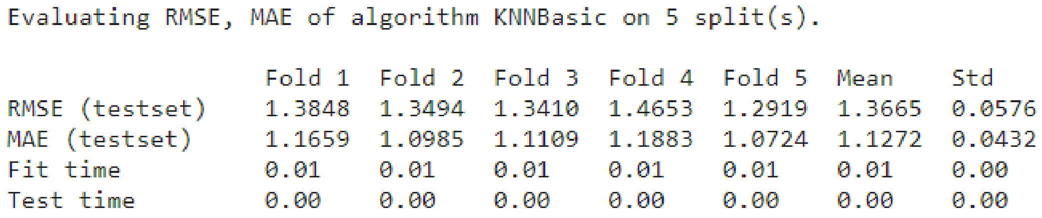
$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v)}$$

or

$$\hat{r}_{ui} = \frac{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j) \cdot r_{uj}}{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j)}$$

For more information : [k-NN inspired algorithms — Surprise 1 documentation]

The hyperparameter tuning performed for parameter K number of neighbours with cosine similarity measure and cross validated with 5 splits to evaluate RMSE and MAE. We get the best K=3 with best RMSE = 1.3505. After cross validation we get average RMSE = 1.3665 , average MAE = 1.1272



```
Evaluating RMSE, MAE of algorithm KNNBasic on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    1.3848  1.3494  1.3410  1.4653  1.2919  1.3665  0.0576  
MAE (testset)     1.1659  1.0985  1.1109  1.1883  1.0724  1.1272  0.0432  
Fit time          0.01    0.01    0.01    0.01    0.01    0.01    0.00    
Test time         0.00    0.00    0.00    0.00    0.00    0.00    0.00    
```

5.3.2. KNNWithMeans algorithm with Cosine Similarity based prediction:

KNNWithMeans collaborative algorithm takes mean ratings of user for predictions.
Prediction formulated as,

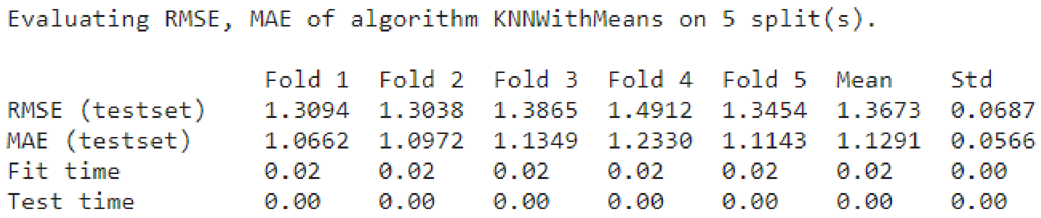
$$\hat{r}_{ui} = \mu_u + \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \mu_v)}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v)}$$

or

$$\hat{r}_{ui} = \mu_i + \frac{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - \mu_j)}{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j)}$$

For more information : [k-NN inspired algorithms — Surprise 1 documentation]

The hyperparameter tuning performed for parameter K number of numbers with similarity measure as Cosine and cross validated with 5 splits to evaluate RMSE and MAE. We get the best K=3 with best RMSE = 1.3218. After cross validation we get average RMSE = 1.3673 , average MAE = 1.1291.



```
Evaluating RMSE, MAE of algorithm KNNWithMeans on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    1.3094  1.3038  1.3865  1.4912  1.3454  1.3673  0.0687  
MAE (testset)     1.0662  1.0972  1.1349  1.2330  1.1143  1.1291  0.0566  
Fit time          0.02    0.02    0.02    0.02    0.02    0.02    0.00    
Test time         0.00    0.00    0.00    0.00    0.00    0.00    0.00    
```

5.3.3. KNNBaseline algorithm with Cosine Similarity based prediction:

KNNWithMeans collaborative algorithm takes baseline ratings of user for predictions.
Prediction formulated as,

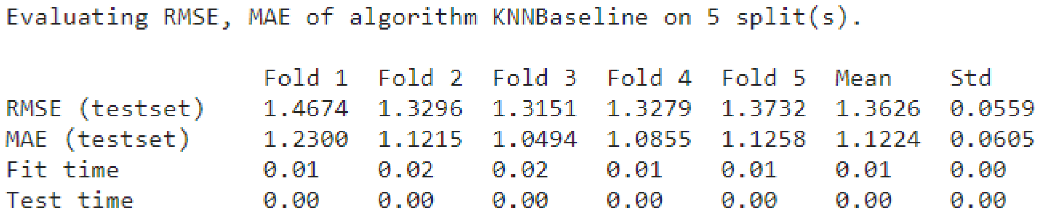
$$\hat{r}_{ui} = b_{ui} + \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - b_{vi})}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v)}$$

or

$$\hat{r}_{ui} = b_{ui} + \frac{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j) \cdot (r_{uj} - b_{uj})}{\sum\limits_{j \in N_u^k(i)} \text{sim}(i, j)}$$

For more information : [k-NN inspired algorithms — Surprise 1 documentation]

The hyperparameter tuning performed for parameter K number of numbers with similarity measure as Cosine and cross validated with 5 splits to evaluate RMSE and MAE. We get the best K=3 with best RMSE = 1.3221. After cross validation we get average RMSE = 1.3626 , average MAE = 1.1224.



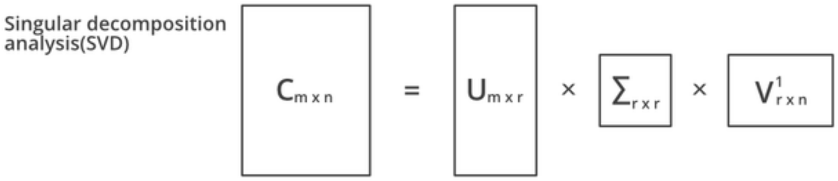
```
Evaluating RMSE, MAE of algorithm KNNBaseline on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    1.4674  1.3296  1.3151  1.3279  1.3732  1.3626  0.0559  
MAE (testset)     1.2300  1.1215  1.0494  1.0855  1.1258  1.1224  0.0605  
Fit time          0.01    0.02    0.02    0.01    0.01    0.01    0.00    
Test time         0.00    0.00    0.00    0.00    0.00    0.00    0.00    
```

5.4 . Sentiment analysis models:

SVD model:

The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. Singular Value Decomposition (SVD) is a widely used technique to decompose a matrix into several component matrices, exposing many of the useful and interesting properties of the original matrix. This model has been utilized to perform sentiment analysis on the review dataset to classify review text as either positive or negative.



Naïve Bayes:

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. The crux of the classifier is based on the Bayes theorem. This model has been implemented for sentiment analysis for accuracy comparison.

Bayes theorem:

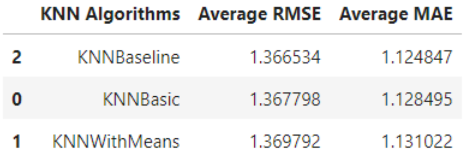
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

NLTK toolkit:

NLTK is a toolkit built for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. The stop words feature within this package has been used to extract the individual words from each review and then classified as positive or negative using the star reviews as basis

6. Experimental outcomes and summary:

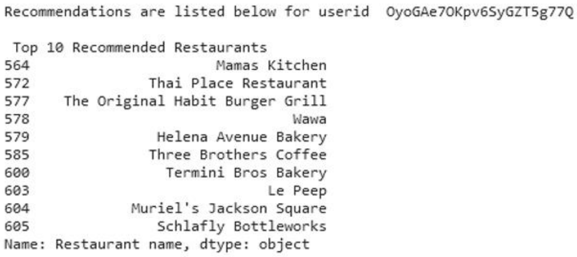
Table 1 shows average RMSE, and MAE given by KNNBasic, KNNWithMeans, and KNNBaseline algorithms. KNNBaseline algorithm gives least average RMSE and MAE (less average errors) so we chose the same as our model for recommender system.



	KNN Algorithms	Average RMSE	Average MAE
2	KNNBaseline	1.366534	1.124847
0	KNNBasic	1.367798	1.128495
1	KNNWithMeans	1.369792	1.131022

Table.1 Average errors

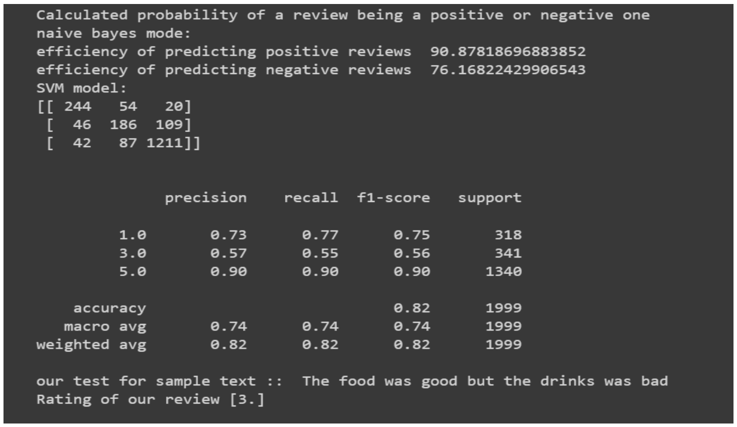
Trained our model with training data from merged dataset and made predictions as below,



```
Recommendations are listed below for userid  OyoGAe7OKpv6SyGZT5g77Q

 Top 10 Recommended Restaurants
564                       Mamas Kitchen
572               Thai Place Restaurant
577    The Original Habit Burger Grill
578                                Wawa
579                Helena Avenue Bakery
585               Three Brothers Coffee
600                  Termini Bros Bakery
603                             Le Peep
604             Muriel's Jackson Square
605                Schlafly Bottleworks
Name: Restaurant name, dtype: object
```

Sentiment analysis has been performed on the text column in the review dataset by splitting each word and classifying them as positive or negative and then inputting into both SVD and Naive Bayes models to predict each text as either positive or negative. fig.13 shows the accuracy for each model with SVD outperforming the other. This model can be incorporated to predict each review in the business dataset as a binary positive or negative.



```
Calculated probability of a review being a positive or negative one
naive bayes mode:
efficiency of predicting positive reviews  90.87818696883852
efficiency of predicting negative reviews  76.16822429906543
SVM model:
[[ 244   54   20]
 [  46  186  109]
 [  42   87 1211]]


              precision    recall  f1-score   support

         1.0       0.73      0.77      0.75       318
         3.0       0.57      0.55      0.56       341
         5.0       0.90      0.90      0.90      1340

    accuracy                           0.82      1999
   macro avg       0.74      0.74      0.74      1999
weighted avg       0.82      0.82      0.82      1999

our test for sample text ::  The food was good but the drinks was bad
Rating of our review [3.]
```

Fig,13 Results of sentiment analysis

7. Future scope:

The project holds multiple potential for future development. Some of the key areas that can be worked upon in the future are highlighted as below:

1. User dataset analysis has been performed and useful insights have been obtained from.
2. From the results it is clear that the dataset can be used to input the reviewer and the impact of their reviews. The dataset holds key information on how useful a reviewer's feedback was, the number of followers they have, and multiple other statistics. This information can be utilized for collaborative filtering.
3. Due to system restraints the computational speed of the system is on the lower end of the spectrum. Implementing this project with better system is a definitive way to improve the performance
4. A deep neural network with multiple mappings can be implemented to improve the metrics of this system.
5. An interactive and easy to use interface or front end can be developed to formulate this as a complete product for the end user.

8. Limitations :

We developed our recommender system with python using online platforms like Kaggle and Google colab. As the Data size is very huge and when the number of users in the platform rises, it automatically increases the difficulty to process the data and using traditional models takes huge time and training model becomes slow. Some big data frameworks like Spark helps parallelizing the computation the same way is not possible with python online platforms.

We tried to use SVD model for our recommender system, but SVD and Cosine Similarity based models complexity increase by n2 , training these models become increasingly inefficient as the number of users and restaurants increases. Since the model develops recommendation for users, it is very difficult to test if the predictions were correct or not.

9. Related works:

Restaurant Recommendation System using Machine Learning - Ketan Mahajan et al june 2021 is a recent work on restaurant recommendation systems that was helpful in understanding the approach towards restaurant recommendations systems. Here they have utilized Hybrid filtering which is a combination of Content-based filtering (CBF) and Collaborative Filtering (CF).

Yelp Restaurant Recommender System - Si Gao et al - was instrumental in approaching the yelp dataset and the preprocessing associated with it. They compare popular collaborative filtering (CF) and content-based (CB) methods that consider user-business interactions, restaurant attributes and text mining of user reviews.

Sentiment Analysis of Yelp‘s Ratings Based on Text Reviews - Yun Xu et al - was the paper for reference for sentiment analysis on the review dataset. The paper experiments with various feature selection and supervised learning algorithms to predict star ratings of the Yelp dataset using review text alone.

10. Conclusion:

The recommendation system so far built has performed well providing satisfactory results and accuracy. There remains scope for improvement in the areas highlighted in the previous section. A thorough exploratory analysis has been performed with a massive dataset like yelp and from this analysis was performed successfully. Databases were used successfully to handle the large dataset. Sentiment analysis has been performed for reviews and the datasets have been merged and fitted to the ML models successfully. Multiple models have been implemented at each stage for accuracy comparison and the results have been evaluated using the appropriate metrics.

11. Code links:

https://github.com/Megalajeyapal/Restaurant-Recommendation-system
https://github.com/genesis1899/restaurant-recommendation/blob/main/ml_project_checkpoint_2.ipynb
https://colab.research.google.com/drive/1PPzpRtoYFwDeiywtMj9YzrvopZQNEetW?usp=sharing

References:

https://www.yelp.com/dataset
https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/
Intro to Recommender Systems: Collaborative Filtering | Ethan Rosenthal
https://www.warse.org/IJATCSE/static/pdf/file/ijatcse261032021.pdf
https://cs229.stanford.edu/proj2014/Yun%20Xu,%20Xinhui%20Wu,%20Qinxia%20Wang,%20Sentiment%20Analysis
%
20of%20Yelp%27s%20Ratings%20Based%20on%20Text%20Reviews.pdf