Restaurant Recommendation and Restaurant Improvement using Yelp Dataset

Capstone Project – Final Report

SUBMITTED BY

Shobin Joyakin

Sonia Mawandia

Nishita Ravindra

Vignesh Athavan

Ganapathi Subramanian Nagarajan

Project Mentor: Mukesh Rao

PGP – BDML Chennai 2017 – 18

Acknowledgement

We wish to place on record our deep appreciation for the guidance and help provided to us by our Mentor **Mr. Mukesh Rao**, Bangalore. Mr. Mukesh Rao helped us narrow down on the choice of the project as well as the scope and focus area of the project. He gave us valuable feedback at every stage to enhance the process and the outputs.

We would also like to place on record our appreciation for the guidance provided by **Dr. Narayana Darapaneni** for giving us valuable feedback and being a source of inspiration in helping us to work on this project.

We would also like to place on record our appreciation for the help provided by Great Lakes PMO team (**Pijush Chatterjee and Mohita Ajmera**) for giving us all the necessary support which helped us greatly in completing the project on time.

Contents

1. INTRODUCTION

The Objective of this capstone project is to help customers choose suitable restaurants in a city and restaurant owners to improve their services based on the data made available in Yelp website.

For users we wanted to provide personalized restaurant recommendations based on a search criterion (Attributes and Cuisines in the restaurant) selected in the application by taking restaurant ratings and reviews into consideration. To enable the search engine, we considered the past ratings of the users on the resultant list of restaurants. In situations where we are unable to provide a specific restaurant to a user, based on the selected criteria, the users could select new restaurants based on similar restaurants the user has already enjoyed. In case the user is new, then he could select the most popular restaurants available in the city.

For Restaurant Owners whose ratings are below 3, we show them why the users are rating their restaurants poorly and also what are the improvements points the owners can act upon, based on the review text provided by the users for that restaurant.

1.1 Problem Statement

Apply data science tools and techniques on the Yelp Dataset that is available

a. For restaurant/ business owners to improve their services and business.
b. For users to choose a best restaurant from the available choices.

Solution:

a. Provide actionable items to restaurants based on the negative reviews given to their services.
b. Recommend restaurants to customers based on their eating preferences and other information such as previous ratings and feedback for restaurants.

1.2 Dataset

The dataset provided for the Capstone Project is part of the Yelp Dataset Challenge and the specific dataset used in this capstone corresponds to Round 11 of their challenge and can be accessed from the following link for download: http://www.yelp.com/dataset_challenge.

The dataset is stored in 5 files of JSON format, where each file is composed of a single object type (a one-json-object per line). The respective data files provide information about:

1. Businesses and their attributes (business.json)
2. Check-in times of customers at given businesses store (checkin.json)
3. Reviews submitted by customers about the businesses (reviews.json)
4. Tips on the businesses (tips.json)
5. Users of the businesses (users.json)

For the capstone project we are using **business**, **reviews** and **users** data.

The data provided was exclusively in JSON format so we stored the data in MongoDB and Python code was used to convert the data into CSV format.

1.3 Findings and Implication

The data made available in YELP was about 6 GB which included businesses other than Restaurants and from various geographical locations. To provide our solutions more precisely, we filtered the business to only those open restaurants in top 5 cities in the state of Arizona that were reviewed between 2013 and 2017 (5 years' time period). This solution could be expanded to any city available in the yelp dataset by proper preprocessing of data.

- The final CSV file that was used for text analytics and recommendation is of size 200 MB and contains 2,73,000 reviews for 7365 restaurants
- Users mostly provide positive reviews
- For low rated restaurants most of the reviews mention about quality of food served and services rendered.

2. OVERVIEW OF FINAL SOLUTION

2.1 Features and Preprocessing

For our objective we have used the following data files and features

Business	
Field	Description
Business Id	Encrypted Restaurant Id
Name	Name of the Restaurant
Address	Address of the Restaurant
City	City name
State	State name
Postal Code	Location postal code
Stars	Restaurant Star Ratings
Review count	Number of Reviews
Is Open	Flag to represent open restaurants
Attributes	Attributes of Restaurant
Categories	Categories of Cuisines

Users	
Field	Description
User Id	Encrypted User Id
Review Count	Number of Reviews

Review	
Field	Description
Business Id	Encrypted Restaurant Id
User Id	Encrypted User Id
Stars	Star Ratings
Text	Review Text

2.2 Methodology

Data preprocessing:

Following steps were identified to prepare the data for analytics

- Connect and load the JSON data files to a NOSQL db like MongoDB.
- Filter the open Restaurants in the state of Arizona that were reviewed between 2013 and 2017.
- Consolidate the restaurant details along with reviews given by each user in JSON format.

Analytical Approach:

The Analytical Approach will involve the following (not necessarily in the order) activities:

- Cleaning, Refining and Transformation of data.
- Study each of the variables by exploring the data and finding key attributes for the model building

For User Recommendation

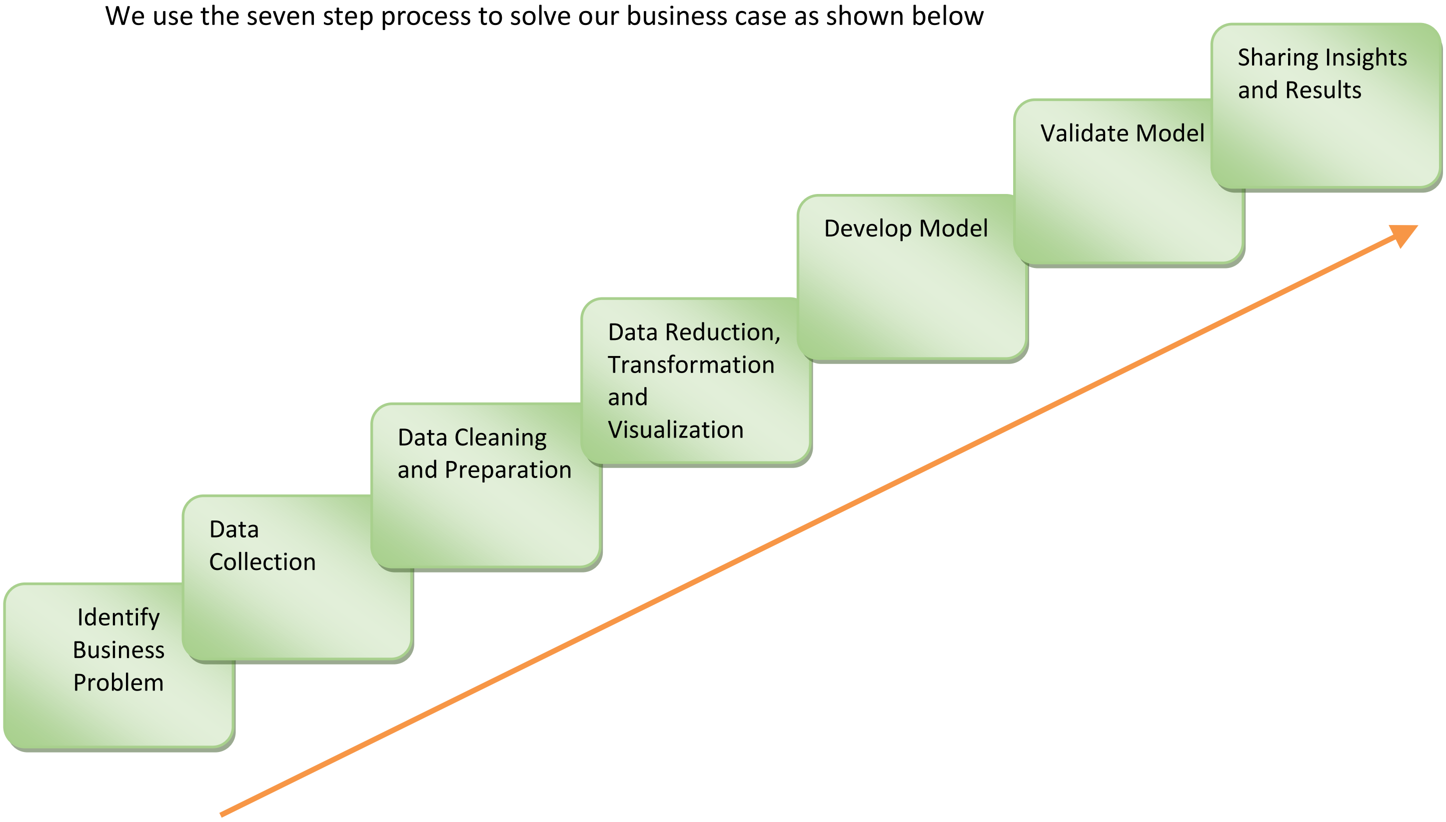
- Identifying top 5 cities which have high number of restaurants to build city based models.
- Identify the density of the User-Restaurant matrix for each city.
- Increase the density of the matrix by handling outliers.
- Build popularity based model based on **calculated** weighted rating of restaurants for each city
- Build content based search engine based on user preference
- Build Collaborative Recommendation using "**Surprise**" Recommendation library along with famous algorithms like SVD, KNN.
- Split the data into Train and Test.
- Build the model for the 5 cities.
- Model evaluation using K-fold validation.
- Recommend restaurants using algorithm with best performance.

For Restaurant Improvements

- Text Preprocessing on all the review text
- Build word cloud for different price range restaurants and compare between low and high rated restaurants.
- Identifying restaurants that are low rated from the 5 selected cities.
- Collect all the negative reviews of those restaurants.
- Identify and display top 5 most negative tokenized bigram for each restaurant from their reviews.
- Identify distinct topics using Latent Dirichlet Allocation also called as LDA for those reviews.
- Suggest areas of improvements for each restaurant based on the LDA model.

2.3 Process Flow Chart

We use the seven step process to solve our business case as shown below



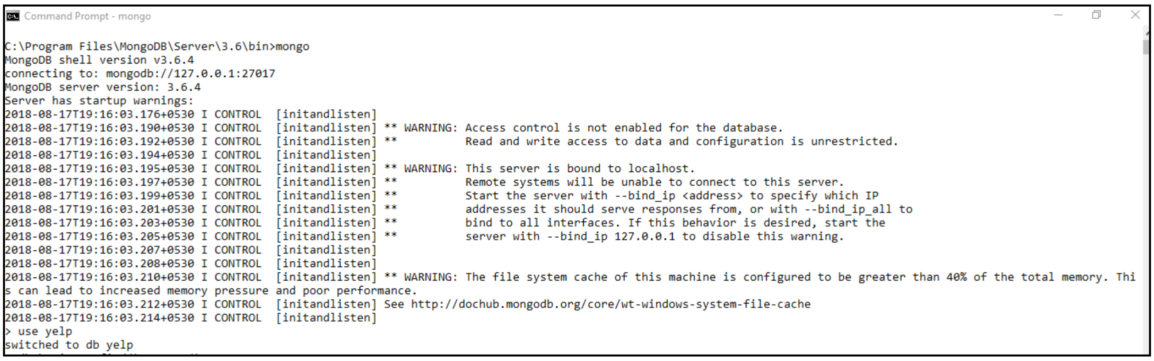
3. STEP BY STEP APPROACH TO SOLUTION

3.1 Data Preparation and Transformation

The Current JSON data available from the YELP website is very huge up to 6 GB in the form of 5 different JSON files. We will be creating a recommendation model using a sample of all open restaurants in the State of Arizona in USA that where reviewed between 2013 and 2017. By this we can restrict the data and create much lesser sparse dataset for our Analysis. This is achieved by filtering the JSON data in MongoDB. Since the User dataset is huge, we are analyzing only those users who reviewed the open restaurants in Arizona between 2013 and 2017. The created model can be scaled for other restaurants in USA which use similar features upon reengineering of model and features.

3.1.1 Setup DB connection in Mongo and data import

Create a new DB called ‘yelp’ using ‘MongoDB compass community’ or Mongo command prompt and import the business, user and review JSON files under its respective collections.



The collections would look as shown below:

a) Review Collection:

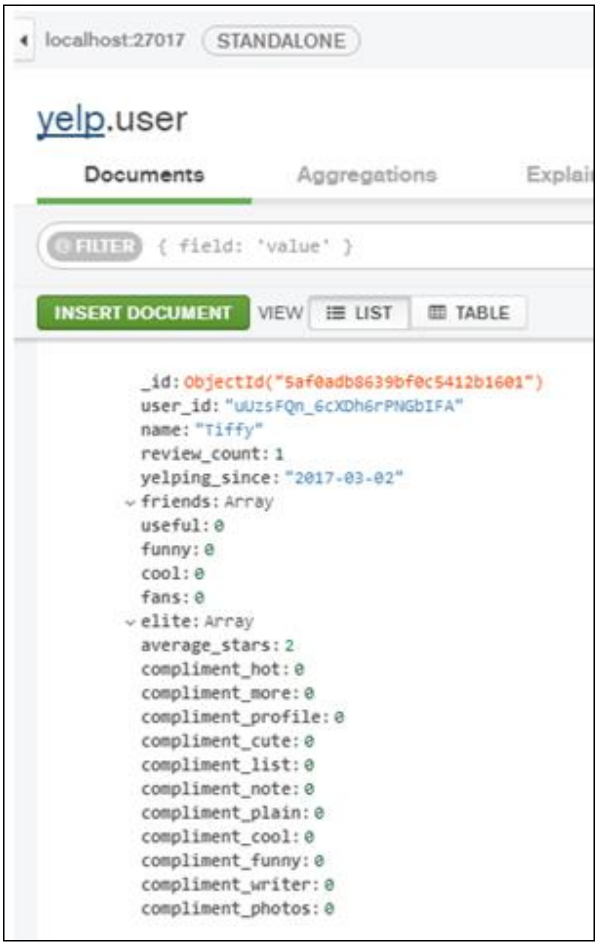
There are overall 5,261,669 reviews captured in the DB.

b) Business Collection:

There are overall 174,567 businesses captured in the DB.

c) User Collection:

Currently we have 1,326,101 user's information in the DB.



3.1.2 Data filtering and pre-processing using MongoDB

We will filter the restaurants that are currently opened in a state as we will not be able to provide any improvement or recommendation on closed restaurants. In order to get more relevant and recent information from the review text, we will filter the reviews that were given in the last 5 years. This would help us provide solutions that are more close to the current scenario. Since each state in the United States of America has its own flavor of food and unique usage of restaurants by people. We would create the solution of the State of Arizona as it has a variety of restaurants.

Below steps in Mongo were used to filter the data as part of our data filtering and consolidation.

a) Identify only the restaurants from the business json and save as 'restaurant'. The filter condition identified 54,618 restaurants out of 174,567 businesses.

```
Select Command Prompt - mongo
>
> db.restaurant.save( db.business.find({categories:"Restaurants"} ).toArray() )
```

b) Identify only the open restaurants and save as ‘open restaurant’. Out of 54,618 restaurants only 40,394 restaurants are currently open.

```
Command Prompt - mongo
> db.open_restaurant.save( db.restaurant.find({is_open: 1}).toArray())
```

c) Identify the businesses that were reviewed from 2013 and 2017 and save as ‘review_2013_17’. Out of 5,261,669 reviews in the db, 4,243,868 reviews were given in the last 5 years.

```
Command Prompt - mongo
>
> db.review_2013_17.save(db.review.find({date: {$in : [/^2013-/, /^2014-/, /^2015-/, /^2016-/, /^2017-/]}}).toArray())
```

d) Filter the reviews only for the open restaurants and save in a new collection ‘open_restaurant_rev_2013_17’. Out of 4,243,868 reviews, 2,305,049 reviews were for open restaurants in the last 5 years.

```
Command Prompt - mongo
> var open_restaurant_unique = db.open_restaurant.distinct( "business_id" )
> db.open_restaurant_rev_2013_17.save(db.review_2013_17.find({"business_id" : {"$in" : open_restaurant_unique}}).toArray() )
```

e) Merge the reviews of all open restaurants into a new collection ‘final_restaurant_data’. This would contain the reviews of 40,394 open restaurants merged into a single collection along with the restaurant details.

```
Select Command Prompt - mongo
> db.final_restaurant_data.save(db.open_restaurant.aggregate([
... {
... $lookup:
... {
... from : "open_restaurant_rev_2013_17",
... localField: "business_id",
... foreignField: "business_id",
... as : "review_details"
... }
... },
... {
... $project:
... {
... "is_open": 0,
... "review_details": { "_id" : 0, "business_id" : 0 }
... }
... }]).toArray() )
```

f) Filter the open restaurants in the state of Arizona into 'AZ_restaurant_review_data' which includes all open restaurants in Arizona that were reviewed between 2013 and 2017. There are 7367 open restaurants in Arizona that were reviewed between 2013 and 2017 from the overall count of 40,394.

```
> db.AZ_restaurant_review_data.save(db.final_restaurant_data.find({"state" : {"$eq" : "AZ"}}).toArray() )
```

This collection will be further used in the data preparation for the Natural Language Processing (NLP) of review text for recommending improvements to low performing restaurants (**3a_NLP_DataPrep.ipynb**) and also for the recommendation engine for users (**4_Recommendation.ipynb**).

g) Create another collection AZ_restaurant_data from the AZ_restaurant_review_data collection after stripping the reviews.

```
> db.AZ_restaurant_data.update({state:"AZ"}, { $unset : {review_details : 1} },{ multi: true });
```

This collection will be further used in Data preparation step by flattening all the restaurant details like attribute and categories in **'1_Restaurant cleanup.ipynb'** file.

3.1.3 Data pre-processing using Python

The AZ_restaurant_data.json file created in the preprocessing step contains the restaurant attributes and cuisine categories in a nested format which is not compatible for analysis and processing in python. The attributes section contains the services and facilities provided by the restaurant and the cuisine categories mentions all the cuisine varieties offered by the restaurant. Upon flattening of the attributes and categories for a restaurant, all the individual facilities and cuisines would become a column of the dataset.

Upon further analysis of the new columns formed from the categories after flattening, it was found that there are 255 categories that are not relevant to restaurant cuisines. The below list of categories refer to the list of categories that were removed from the dataset.

category_& Probates	category_Dance Clubs	category_International Grocery	category_Public Services & Government
category_Acai Bowls	category_Dance Schools	category_Internet Cafes	category_Race Tracks
category_Accessories	category_Dance Studios	category_Investing	category_Real Estate
category_Active Life	category_Day Spas	category_Jazz & Blues	category_Real Estate Services
category_Adult Entertainment	category_Dentists	category_Jewelry	category_Resorts
category_Advertising	category_Department Stores	category_Karaoke	category_Restaurant Supplies

category_Air Duct Cleaning	category_Discount Store	category_Kids Activities	category_Roofing
category_Airport Lounges	category_Distilleries	category_Kitchen & Bath	category_Rotisserie Chicken
category_Airports	category_Dive Bars	category_Lakes	category_RV Parks
category_Amusement Parks	category_DJs	category_Landmarks & Historical Buildings	category_RV Repair
category_Antiques	category_Doctors	category_Laser Tag	category_Screen Printing
category_Apartments	category_Drugstores	category_Lawyers	category_Screen Printing/T-Shirt Printing
category_Appliances	category_Education	category_Leisure Centers	category_Seafood Markets
category_Appliances & Repair	category_Electronics	category_Libraries	category_Security Systems
category_Art Galleries	category_Empanadas	category_Life Coach	category_Session Photography
category_Art Schools	category_Estate Liquidation	category_Local Services	category_Shared Office Spaces
category_Arts & Crafts	category_Estate Planning Law	category_Magicians	category_Shaved Ice
category_Auto Customization	category_Ethnic Grocery	category_Mags	category_Shaved Snow
category_Auto Detailing	category_Event Photography	category_Marketing	category_Shoe Stores
category_Auto Repair	category_Farmers Market	category_Masonry/Concrete	category_Shopping
category_Auto Upholstery	category_Farms	category_Massage	category_Shopping Centers
category_Automotive	category_Fashion	category_Massage Therapy	category_Smokehouse
category_Bankruptcy Law	category_Financial Services	category_Medical Spas	category_Social Clubs
category_Barbers	category_Fire Protection Services	category_Medical Transportation	category_Souvenir Shops
category_Beauty & Spas	category_Fitness & Instruction	category_Men's Clothing	category_Speakeasies
category_Bed & Breakfast	category_Flooring	category_Mini Golf	category_Specialty Schools
category_Beverage Store	category_Florists	category_Motorcycle Repair	category_Sporting Goods
category_Bike Repair/Maintenance	category_Flowers & Gifts	category_Museums	category_Sports Clubs
category_Bikes	category_Food Tours	category_Music & Video	category_Sports Wear
category_Bingo Halls	category_Foundation Repair	category_Musicians	category_Stadiums & Arenas
category_Boating	category_Fruits & Veggies	category_Nail Salons	category_Supernatural Readings
category_Books	category_Furniture Reupholstery	category_Nail Technicians	category_Swimming Pools
category_Bookstores	category_Furniture Stores	category_Nutritionists	category_Tattoo
category_Botanical Gardens	category_Gas Stations	category_Office Cleaning	category_Tax Law
category_Bowling	category_Gay Bars	category_Olive Oil	category_Tax Services
category_Building Supplies	category_Gelato	category_Organic Stores	category_Teeth Whitening
category_Business Consulting	category_General Dentistry	category_Orthodontists	category_Tires
category_Butcher	category_Gift Shops	category_Outlet Stores	category_Tobacco Shops
category_Candy Stores	category_Go Karts	category_Party & Event Planning	category_Tours
category_Car Wash	category_Golf	category_Party Equipment	category_Towing

		Rentals	
category_Cardiologists	category_Golf Equipment	category_Party Supplies	category_Trainers
category_Champagne Bars	category_Golf Equipment Shops	category_Pawn Shops	category_Transportation
category_Check Cashing/Pay-day Loans	category_Golf Lessons	category_Pediatricians	category_Trusts
category_Cheese Shops	category_Graphic Design	category_Performing Arts	category_Unofficial Yelp Events
category_Christmas Trees	category_Grilling Equipment	category_Personal Chefs	category_Used Bookstore
category_Cinema	category_Guest Houses	category_Personal Shopping	category_Vacation Rentals
category_Climbing	category_Gyms	category_Pet Adoption	category_Vape Shops
category_Clothing Rental	category_Hair Salons	category_Pet Services	category_Venues & Event Spaces
category_Club Crawl	category_Hair Stylists	category_Pets	category_Veterinarians
category_Comedy Clubs	category_Handyman	category_Pharmacy	category_Virtual Reality Centers
category_Community Service/Non-Profit	category_Health & Medical	category_Photographers	category_Vitamins & Supplements
category_Contractors	category_Health Markets	category_Piano Bars	category_Water Stores
category_Convenience Stores	category_Health Retreats	category_Piercing	category_Web Design
category_Cooking Classes	category_Home & Garden	category_Pita	category_Wedding Chapels
category_Cooking Schools	category_Home Cleaning	category_Playgrounds	category_Wedding Planning
category_Cosmetic Dentists	category_Home Health Care	category_Plumbing	category_Whiskey Bars
category_Cosmetic Surgeons	category_Home Inspectors	category_Poke	category_Wholesale Stores
category_Cosmetics & Beauty Supply	category_Home Services	category_Pool & Billiards	category_Wholesalers
category_Country Clubs	category_Horse Racing	category_Pool & Hot Tub Service	category_Wills
category_Country Dance Halls	category_Horseback Riding	category_Pool Halls	category_Wineries
category_Couriers & Delivery Services	category_Hospitals	category_Post Offices	category_Women's Clothing
category_Cultural Center	category_Hot Tub & Pool	category_Preschools	category_Yoga
category_Currency Exchange	category_Hotels & Travel	category_Professional Services	category_Zoos
category_Damage Restoration	category_Interior Design	category_Property Management	

Steps for data cleansing

- Read the flattened json file in the form of CSV to Python.
- Final dataframe has 7367 rows and 276 columns.
- Identify the number of null values for each of the columns in the dataframe.

- Below mentioned columns are either not relevant to the restaurant

neighborhood	HairSpecializesIn_curly	BYOBCorkage
AcceptsInsurance	HairSpecializesIn_extensions	Caters
BestNights_friday	HairSpecializesIn_kids	GoodForDancing
BestNights_monday	HairSpecializesIn_perms	HairSpecializesIn_africanamerican
BestNights_saturday	HairSpecializesIn_straightperms	HairSpecializesIn_asian
BestNights_sunday	Music_background_music	HairSpecializesIn_coloring
BestNights_thursday	Music_dj	hours_Monday
BestNights_tuesday	Music_jukebox	hours_Tuesday
BestNights_wednesday	Music_karaoke	hours_Wednesday
BusinessAcceptsBitcoin	Music_live	hours_Thursday
ByAppointmentOnly	Music_video	hours_Friday
BYOB	Open24Hours	hours_Saturday
RestaurantsCounterService	CoatCheck	hours_Sunday
category_Restaurants	Corkage	

- The cleaned up restaurant data now has total of 7367 restaurants and 235 features.

  For the following features, NaN values are handled as shown below

Column Name	Column Values Before Handling NaN	Column Values After Handling NaN	Final Values	Description
AgesAllowed	[nan, 'allages', '21plus']	['allages', '21plus']	[0, 1]	21plus : 1, allages : 0
Alcohol	['none', 'full_bar', nan, 'beer_and_wine']	['none', 'full_bar', 'beer_and_wine']	[0, 2, 1]	none: 0, beer and wine: 1, Full bar: 2
Music_no_music	[nan, False]	[nan, False]	[0, 1]	False: 1, True: 0
NoiseLevel	['loud', 'average', nan, 'quiet', 'very_loud']	['loud', 'average', 'quiet', 'very_loud']	[2, 1, 0, 3]	quiet: 0, average: 1, loud: 2, very_loud:3
RestaurantsAttire	['casual', nan, 'dressy', 'formal']	['casual', 'dressy' ,'formal']	[0, 1, 2]	casual:0, dressy:1,formal: 2
RestaurantsPriceRange2	[1., 2., nan, 3., 4.]	[1., 2., 0, 3., 4.]	[1, 2, 0, 3, 4]	0, 1, 2, 3, 4
Smoking	[nan, 'no', 'outdoor', 'yes']	['no', 'outdoor', 'yes']	[0, 1, 2]	no:0, outdoor:1, yes:2
WiFi	['free', 'paid', nan, 'no']	['free', 'paid', 'no']	[1, 2, 0]	no:0, free:1, paid:2

- All the category column values are converted to 0's and 1's, after handing their missing values as 0's
- Missing values in address are not handled since correct information is not available.
- There are 9 restaurants that do not postal code. These values are determined from Tableau after plotting the map and taking the closest value to the missing zip code value or determining the actual zip code value.
- The table provided below gives the details of the business id's with the updated postal code information obtained from Tableau.

Business id	Postal code
8N5A5jW8sTG7ozGmZ_uj0Q	85004
o-cqIwS2nWJGWA6nNPQ2_w	85210
3LWTG3f52gdVZILl42FCAg	85374
DscqZ5DUZSWTggbv6N_j3Q	85003
Rut5I04WZ2Hm2GEyDwk8YA	85034
BxRsg5YTYnxXEhoGcabydg	85331
R0gfUYIiC5MlLxGjRcYifw	85206
YH9jx2vlDBD47qTBoCvkoA	85003
wFJFBZDMcV0hBMYOx88IYQ	85003
tY_Hmm0rT4sRDL-geEYQmg	85027
DmyS9b7ykIOo7XwYt5I9wg	85034
NOUYmgX2Hl2BlhqsWCUAyg	85339

- As part of data cleaning, there were restaurants with same name but with extra symbols, wrong city details based on the postal code provided in the data and also the business having wrong latitude and longitude values during plotting of charts, maps and graphs in Tableau. Such discrepancies are handled and tabled below.
  - The business id's mentioned below are having wrong city details based on their postal code values. They are updated with correct value as their city, based on the graphs from Tableau.

Business_id	Updated city
ivYI5FZ7ULlfaqsc-wIVSQ	Phoenix
j8j0IV_eemJpFuRRvdNobQ	Phoenix
pgpnvPd3mWxI0O9WYUeLnA	Glendale
svU7GceMV2PsJe6MLSJWIA	Mesa
mhjSXQR9LvL10SSHH6aHlg	Phoenix
0Zu6KcjFJjBpCOjiOa2HvQ	Scottsdale
JvHv4HDT7sU7rMYyDzJuMQ	Scottsdale
Y3aGrZR8QBlj8C2l7oEvDg	Scottsdale
boFrQQWEXyNe79kQ_DOjYA	Sun City West

- The restaurant name is updated for the below mentioned business id's as their names are same but they differ based on apostrophes and additional special characters.

Business_id	Old restaurant name	Updated restaurant name
mhjSXQR9LvL10SSHH6aHlg	M'OlÃ©	Mole
kkEqZmVvVkgmCaOqE13mDg	#1 Sushi	1 Sushi
ysaGG0hm7-ug-IuRd-8_ew	#1 Fried Rice	1 Fried Rice
4z-QW_f3RwCAxHB5fd58TA	#1Brothers Pizza	1 Brother's Pizza

- The following business id's are dropped from our dataset, due to lack of valid data.

Business_id
WGQpjj6eA6mRBxp4XST6Ig
82bfom6b5BQlnnd4m3wwew

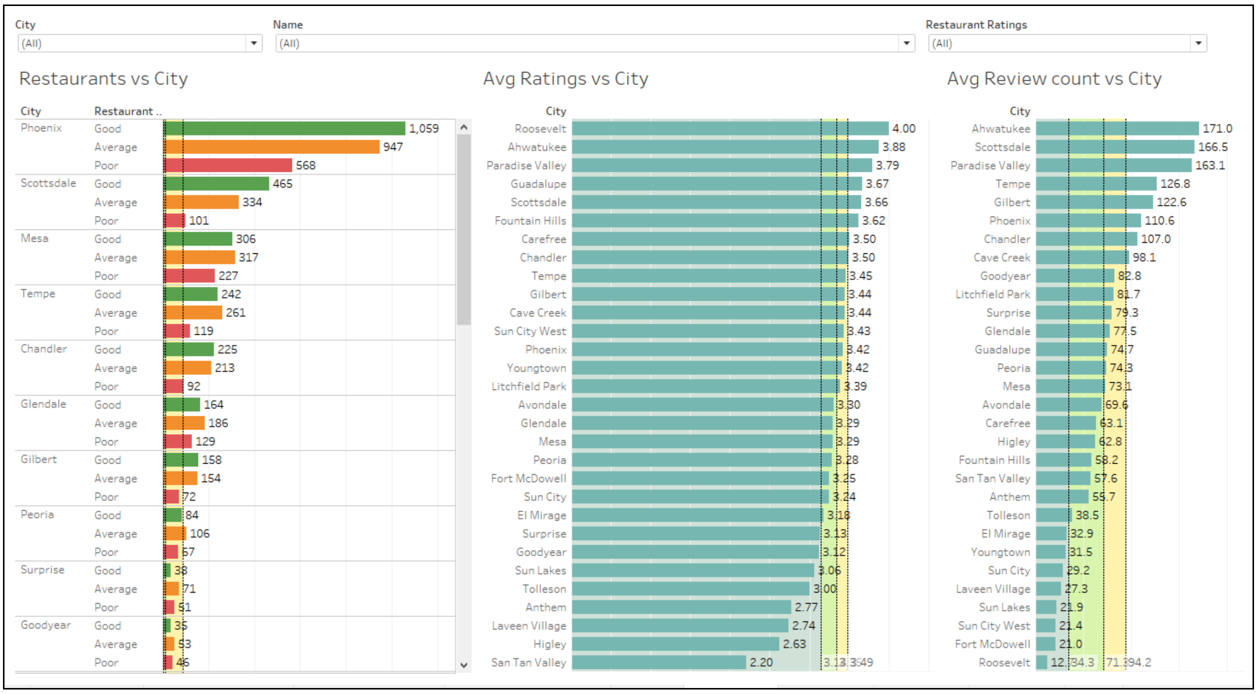
- Business id 'gZGsReG0VeX4uKViHTB9EQ' is having incorrect latitude and longitude value based on its pin code. It has been updated with '33.5959024' and '-112.1531041' as its latitude and longitude values respectively.

- We have converted all columns other than 'business_id', 'name', 'address', 'city', 'state', 'postal_code', 'latitude', 'longitude', 'stars' and 'review_count' to their label encoded values .

- The final cleaned up data is exported as ‘CSV’ with the file named ‘AZ_Restaurant_Final_Clean_Data’ for EDA using Tableau and Recommendation.

3.2 Exploratory Data Analysis

The below analysis shows how each city in Arizona is performing based on ratings and the number of reviews given.



Following are the inferences made.

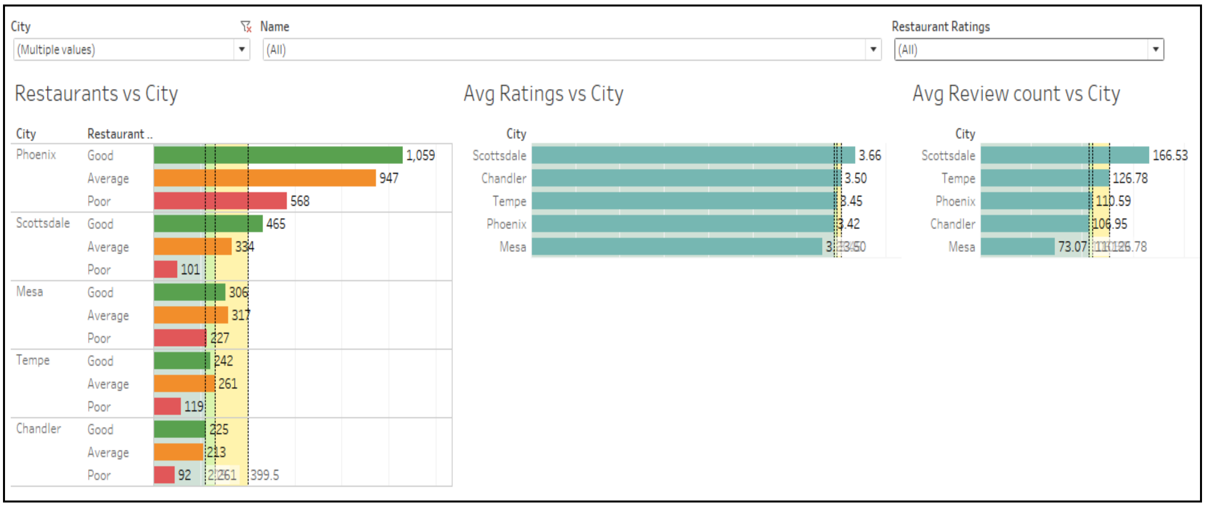
- There is a big variation in the restaurant count through-out the state of Arizona with most number of restaurants in **Phoenix, Scottsdale, Mesa, Tempe, Chandler**. Hence these cities alone would be considered for further analysis and processing.
- There is wide variation in the Good vs poor rated restaurants in Phoenix. While this gap gets reduced as the number of restaurant count decreases. There are few cities like ‘Surprise’ and ‘Goodyear’ with equal or less number of good rated restaurants when compared to poor rated restaurants.
- The average city ratings vary from 4 to 2.2. This average rating needs to be compared with the overall review counts for that city. The average review count has a large variation as well; hence we wouldn’t be able to completely rely on top average rating of the city. Example: Average rating in Roosevelt is 4.0 but the number of review given for the restaurants in that city is only 12. **Hence we**

need to take a weighted rating when identifying top performing city. They would apply to identify top performing restaurants in a city as well.

- Based on the spread of the rating, we understand that recommendation of good restaurants in few cities could be a challenge as the number of good reviews is very limited. However we have options to suggest improvement to the restaurant owners to increase their performance.

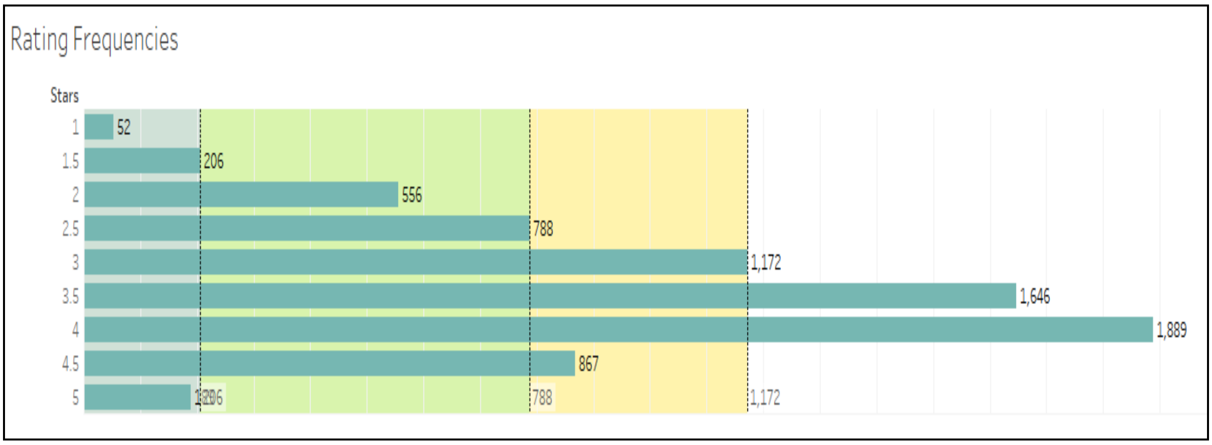
1) **Word cloud** based on number of restaurants in the city denotes top cities with most restaurants.

2) Analysis of **restaurants in the top 5 cities** Phoenix, Scottsdale, Mesa, Tempe, Chandler



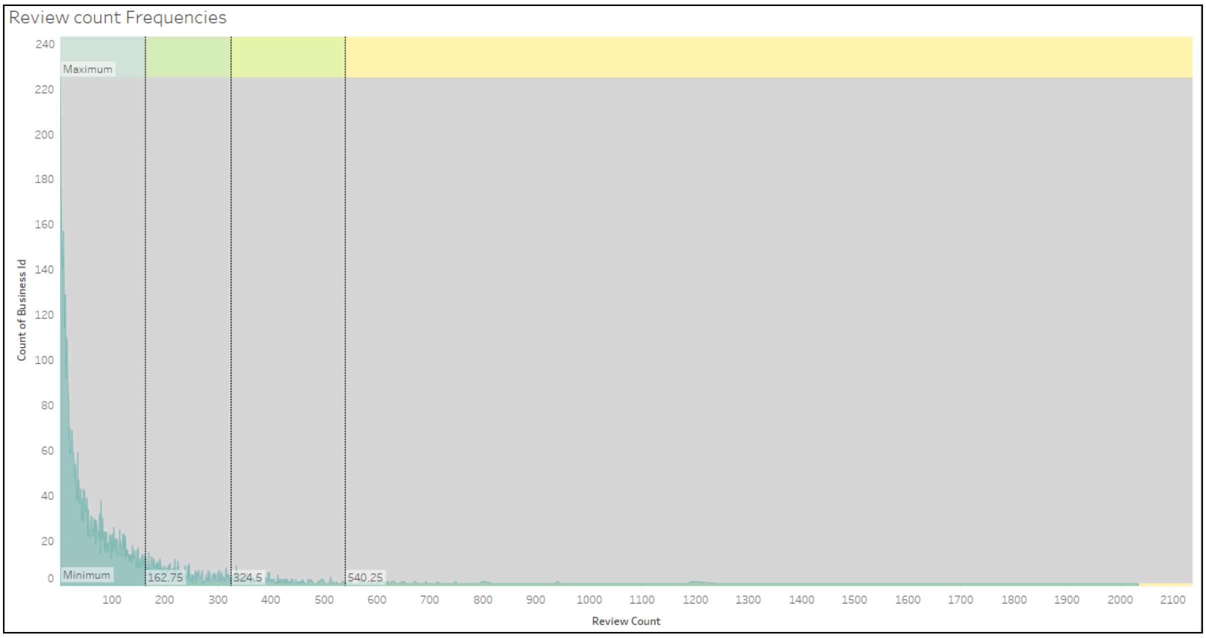
Overall average rating of the restaurants is very close; however the overall review count has big variation.

3) The **restaurant rating frequency** across all the restaurants are shown below



Most restaurants in the city of Arizona seem to be performing well with a rating of 3.5 and 4. More than 50 % of the restaurants have a rating between 3 and 4.5.

4) Analysis of **Review count** over all the cities in Arizona



The graph shows a highly skewed review count spread through the state of Arizona, with 225 restaurants given the least review count of 3 and 1 restaurant with a review count of whopping 2035.

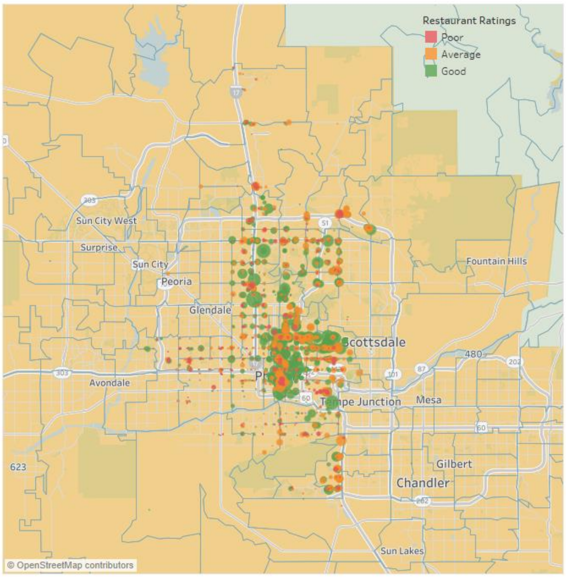
5) Below charts shows the spread of Good, Average and poor rated restaurants in the top 5 cities in Arizona. This will help us identify geographical area where poor and good rated restaurants are present. This can give us some kind of hint on the socio-economic background of that particular area.

Phoenix:

City: Phoenix | Name: (All) | Restaurant..: (All)

Restaurant Detail Analysis

Name	City	Restaurant Ratin..	Review Count	
1 Sushi	Phoenix	Good	55	1
1000 Degrees Neapolitan ..	Phoenix	Good	165	1
1130 The Restaurant	Phoenix	Average	204	1
12th Fairway Bar & Grill	Phoenix	Poor	22	1
24th Street Pizza & Gyros	Phoenix	Average	60	1
2601 on Central	Phoenix	Good	25	1
27th Ave Bar n Grill	Phoenix	Average	27	1
32 Shea	Phoenix	Good	597	1
3on Smith Cafe	Phoenix	Good	158	1
40th Street Cafe	Phoenix	Good	140	1
5 & Diner	Phoenix	Average	249	1
5 Star Wings	Phoenix	Average	5	1
56th St Deli & Market	Phoenix	Good	16	1
5th Avenue Cafe	Phoenix	Good	202	1
613 Grill	Phoenix	Average	16	1
68 Degrees Sabor	Phoenix	Poor	4	1
AMC Dine-in Theatres Esp..	Phoenix	Average	651	1
ATL Wings	Phoenix	Good	177	1
AZ Kabob House	Phoenix	Good	231	1
AZTECA Bakeries & Resta..	Phoenix	Good	62	1
Abacus Inn	Phoenix	Average	117	1
Abyssinia Restaurant and ..	Phoenix	Good	126	1
Adela's Italian	Phoenix	Good	222	1
Adobo Dragon	Phoenix	Good	82	1
Ah-So Sushi & Steak	Phoenix	Average	222	1
Ahipoki - CityScape	Phoenix	Good	14	1
Ahipoki Bowl	Phoenix	Good	95	2
Ahwatukee Country Club	Phoenix	Poor	18	1
Aiello's East Coast Italian	Phoenix	Good	130	1
Aioli Gourmet Burgers	Phoenix	Good	135	1

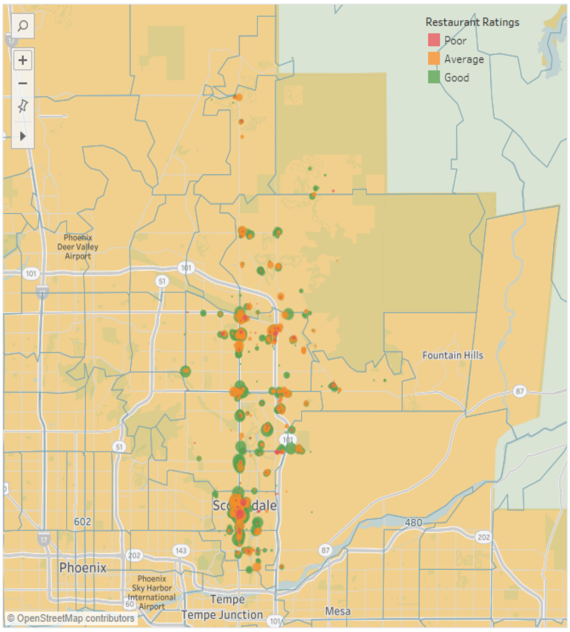


Scottsdale:

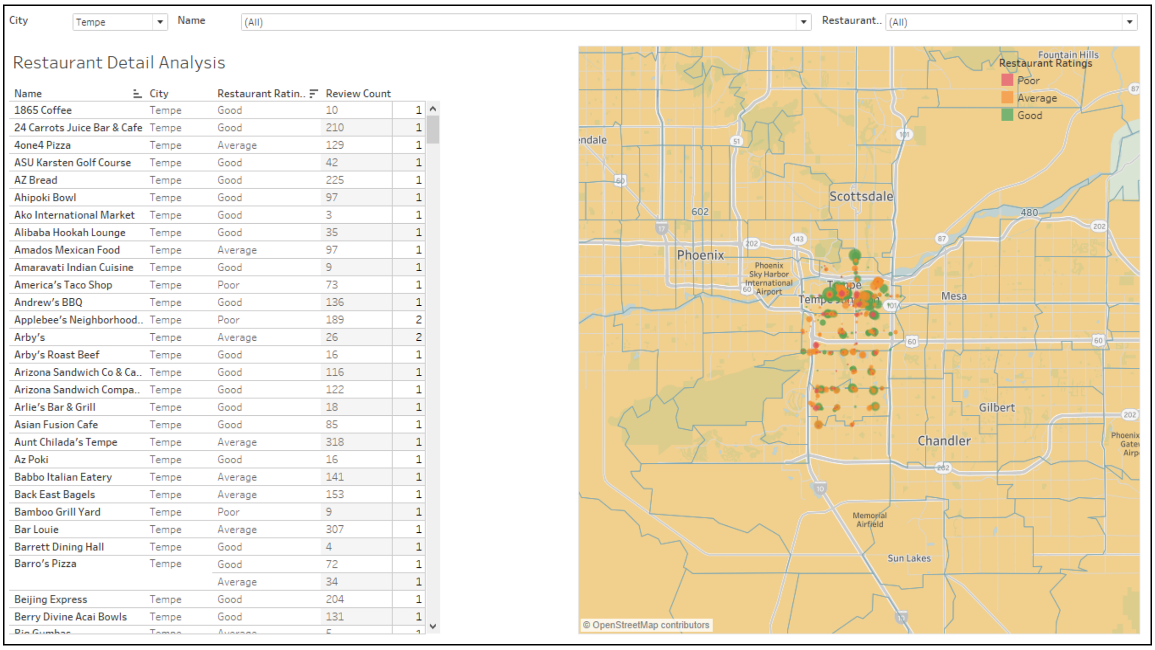
City: Scottsdale | Name: (All) | Restaurant..: (All)

Restaurant Detail Analysis

Name	City	Restaurant Ratin..	Review Count	
18 Degrees Neighborhood..	Scottsdale	Average	77	1
3rd Avenue Grill	Scottsdale	Average	3	1
5 & Diner	Scottsdale	Average	114	1
6 Degrees Urban Kitchen	Scottsdale	Good	56	1
92nd Street Cafe	Scottsdale	Average	124	1
ATL Wings	Scottsdale	Good	4	1
AZ88	Scottsdale	Good	625	1
Adaptive Force Performin..	Scottsdale	Good	13	1
Agave Southwest Bar and..	Scottsdale	Average	13	1
Ahipoki Bowl	Scottsdale	Good	539	1
Ajo Al's	Scottsdale	Average	169	1
Ajo Al's Mexican Cafe	Scottsdale	Average	97	1
Akita Sushi	Scottsdale	Average	381	1
Al Hamra	Scottsdale	Average	106	1
All American Modern Spor..	Scottsdale	Average	106	1
Alo Cafe	Scottsdale	Good	378	1
Altitude Coffee Lab	Scottsdale	Good	231	1
Alto Ristorante e Bar	Scottsdale	Good	123	1
America's Taco Shop	Scottsdale	Average	283	2
American Legion Post 44	Scottsdale	Good	6	1
Andaz Scottsdale Resort /..	Scottsdale	Good	48	1
Andreoli Italian Grocer	Scottsdale	Good	322	1
Arby's	Scottsdale	Poor	16	1
		Average	23	2
Arcadia Farms Cafe	Scottsdale	Good	580	1
Arcadia Farms Marketplace	Scottsdale	Good	7	1
Arena Sports Grill	Scottsdale	Average	104	1
Arizona BBQ Shack	Scottsdale	Good	109	1
Arizona Bread Company	Scottsdale	Average	65	1
Arizona Room Patio Grill	Scottsdale	Poor	12	1

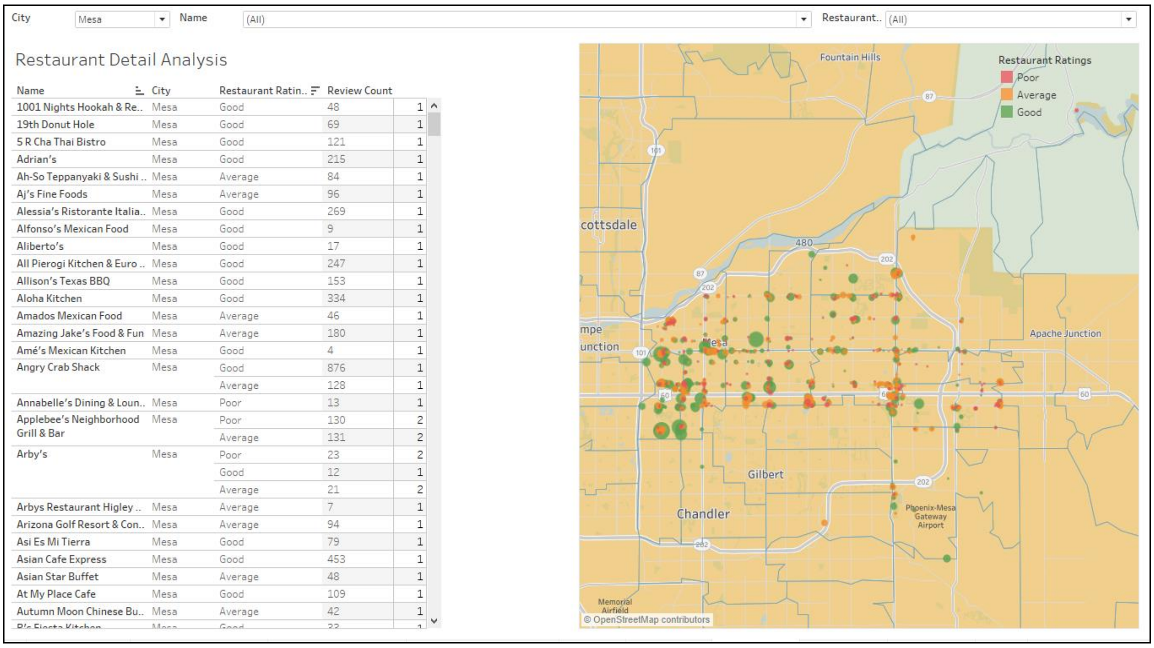


Tempe:



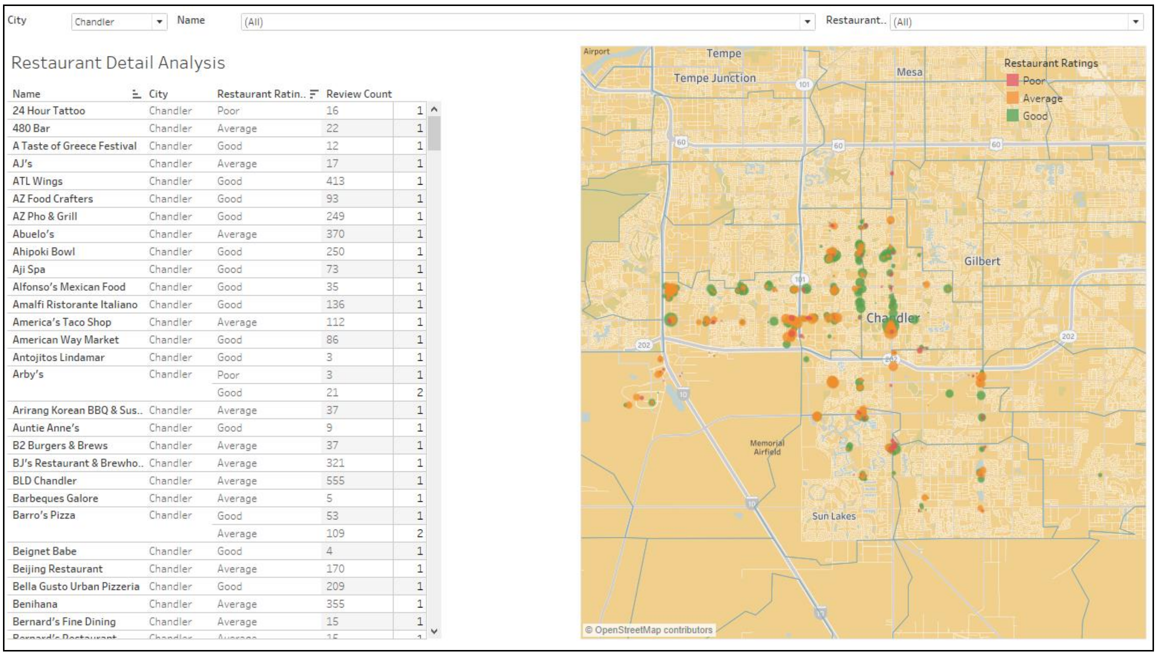
Name	City	Restaurant Ratin..	Review Count	
1865 Coffee	Tempe	Good	10	1
24 Carrots Juice Bar & Cafe	Tempe	Good	210	1
4one4 Pizza	Tempe	Average	129	1
ASU Karsten Golf Course	Tempe	Good	42	1
AZ Bread	Tempe	Good	225	1
Ahipoki Bowl	Tempe	Good	97	1
Ako International Market	Tempe	Good	3	1
Alibaba Hookah Lounge	Tempe	Good	35	1
Amados Mexican Food	Tempe	Average	97	1
Amaravati Indian Cuisine	Tempe	Good	9	1
America's Taco Shop	Tempe	Poor	73	1
Andrew's BBQ	Tempe	Good	136	1
Applebee's Neighborhood..	Tempe	Poor	189	2
Arby's	Tempe	Average	26	2
Arby's Roast Beef	Tempe	Good	16	1
Arizona Sandwich Co & Ca..	Tempe	Good	116	1
Arizona Sandwich Compa..	Tempe	Good	122	1
Arlie's Bar & Grill	Tempe	Good	18	1
Asian Fusion Cafe	Tempe	Good	85	1
Aunt Chilada's Tempe	Tempe	Average	318	1
Az Poki	Tempe	Good	16	1
Babbo Italian Eatery	Tempe	Average	141	1
Back East Bagels	Tempe	Average	153	1
Bamboo Grill Yard	Tempe	Poor	9	1
Bar Louie	Tempe	Average	307	1
Barrett Dining Hall	Tempe	Good	4	1
Barro's Pizza	Tempe	Good	72	1
		Average	34	1
Beijing Express	Tempe	Good	204	1
Berry Divine Acai Bowls	Tempe	Good	131	1

Mesa:



Name	City	Restaurant Ratin..	Review Count	
1001 Nights Hookah & Re..	Mesa	Good	48	1
19th Donut Hole	Mesa	Good	69	1
5 R Cha Thai Bistro	Mesa	Good	121	1
Adrian's	Mesa	Good	215	1
Ah-So Teppanyaki & Sushi ..	Mesa	Average	84	1
Aj's Fine Foods	Mesa	Average	96	1
Alessia's Ristorante Italia..	Mesa	Good	269	1
Alfonso's Mexican Food	Mesa	Good	9	1
Aliberto's	Mesa	Good	17	1
All Pierogi Kitchen & Euro ..	Mesa	Good	247	1
Allison's Texas BBQ	Mesa	Good	153	1
Aloha Kitchen	Mesa	Good	334	1
Amados Mexican Food	Mesa	Average	46	1
Amazing Jake's Food & Fun	Mesa	Average	180	1
Amé's Mexican Kitchen	Mesa	Good	4	1
Angry Crab Shack	Mesa	Good	876	1
		Average	128	1
Annabelle's Dining & Loun..	Mesa	Poor	13	1
Applebee's Neighborhood Grill & Bar	Mesa	Poor	130	2
		Average	131	2
Arby's	Mesa	Poor	23	2
		Good	12	1
		Average	21	2
Arbys Restaurant Higley ..	Mesa	Average	7	1
Arizona Golf Resort & Con..	Mesa	Average	94	1
Asi Es Mi Tierra	Mesa	Good	79	1
Asian Cafe Express	Mesa	Good	453	1
Asian Star Buffet	Mesa	Average	48	1
At My Place Cafe	Mesa	Good	109	1
Autumn Moon Chinese Bu..	Mesa	Average	42	1

Chandler:



Name	City	Restaurant Ratin..	Review Count	
24 Hour Tattoo	Chandler	Poor	16	1
480 Bar	Chandler	Average	22	1
A Taste of Greece Festival	Chandler	Good	12	1
AJ's	Chandler	Average	17	1
ATL Wings	Chandler	Good	413	1
AZ Food Crafters	Chandler	Good	93	1
AZ Pho & Grill	Chandler	Good	249	1
Abuelo's	Chandler	Average	370	1
Ahipoki Bowl	Chandler	Good	250	1
Aji Spa	Chandler	Good	73	1
Alfonso's Mexican Food	Chandler	Good	35	1
Amalfi Ristorante Italiano	Chandler	Good	136	1
America's Taco Shop	Chandler	Average	112	1
American Way Market	Chandler	Good	86	1
Antojitos Lindamar	Chandler	Good	3	1
Arby's	Chandler	Poor	3	1
		Good	21	2
Arirang Korean BBQ & Sus..	Chandler	Average	37	1
Auntie Anne's	Chandler	Good	9	1
B2 Burgers & Brews	Chandler	Average	37	1
BJ's Restaurant & Brewho..	Chandler	Average	321	1
BLD Chandler	Chandler	Average	555	1
Barbeques Galore	Chandler	Average	5	1
Barro's Pizza	Chandler	Good	53	1
		Average	109	2
Beignet Babe	Chandler	Good	4	1
Beijing Restaurant	Chandler	Average	170	1
Bella Gusto Urban Pizzeria	Chandler	Good	209	1
Benihana	Chandler	Average	355	1
Bernard's Fine Dining	Chandler	Average	15	1

3.3 Restaurant recommendation for users

As per the exploratory data analysis performed on the restaurant data, there is wide variation in the performance of restaurants in each city. Going by the analysis, we would create individual recommendation models for each city. This could be avoided in production with one model on a larger geographical location if we have large number of restaurants in all the cities and we could create clusters of cities based on the tier levels. i.e. All Tier 1 cities could have 1 model and Tier 3 or 4 could have another model as the restaurant types / offerings and people usage patterns would be different in different Tiers.

Below steps were followed to prepare the data for recommendation for the capstone project:

- Connect to Mongodb to read AZ_restaurant_review_data.json that contains all the user review information of open restaurants in Arizona.
- Read the flatten Restaurant file AZ_Restaurant_Final_Clean_Data.csv to get all attributes and category information for each restaurant.
- Merge the 2 files and create an individual dataset for the top 5 cities with the following key features of the Restaurant and flattened users review for each restaurant.

Business_id	GoodForMeal_dinner	BusinessParking_valet
Name	GoodForMeal_latenight	WiFi
Address	Alcohol	Category_Fast_Food
City	Ambience_casual	Category_Sandwiches
State	Ambience_classy	Category_Mexican
Postal_code	Ambience_romantic	Category_American_Traditional
Review_count	Ambience_trendy	Category_Nightlife
Review_details	Ambience_upscale	Category_Pizza
Restaurant_ratings	HasTV	Category_Bars
RestaurantsPriceRange2	BusinessParking_garage	Category_Burgers
GoodForMeal_breakfast	BusinessParking_lot	Category_Breakfast_Brunch
GoodForMeal_lunch	BusinessParking_street	Category_American_New

- In order to increase the performance of the recommendation systems to be more accurate and faster, the density of the restaurant and user rating matrix needs to be high. The density of the rating matrix is calculated in the below manner.

  Density of Matrix = Number of Restaurant Ratings *100 / (Unique count of users * Unique count of Restaurants)

- We will follow below steps to increase the density of the rating matrix for each city based on the data made available.
  - **Restaurant's Average Rating:** Since we are going to recommend only those restaurants that are rated good to the users, we would consider only the restaurants that are performing with average rating above 3.

  Note: A point to note here is that we would not be able to consider the previous experiences of a user on those poorly rated restaurants that were removed by the filter, and there could be a very rare chance of recommending a similar restaurant to the user.

  - **Restaurant's Total review:** Based on the analysis, the least review count on a restaurant is found to be 3. But when we are looking at the spread of review count, there are restaurants with more reviews. The rating of restaurants with more reviews could help us with much better recommendation that with those restaurants with lesser reviews.

    50^{th} percentile value for top 5 cities are:
    Phoenix – 240 reviews
    Scottsdale – 216 reviews
    Mesa – 122 reviews
    Tempe – 152 reviews
    Chandler – 140 reviews

  - **Users' Total review:** The users who often provide reviews are those who are very much impressed by the services or food served at a restaurant or those who are not. There are also users who are habitual at giving reviews for the restaurants they visit to help other users. Recommending restaurants to users who have not provided ratings to restaurants or who have provided very few ratings to restaurants is a challenge using normal item similarity based collaborative filter. Here we would not consider those **users who have**

given least number of reviews as we could recommend them restaurants based on **user similarity** or **other hybrid models**.

City	Original dataset Rating matrix density	Restaurant rating filter	Rating matrix density	Restaurant review count	Rating matrix density	User review	Rating matrix density
Phoenix	0.0888	> 3	0.1371	> 3	0.1408	> 2	0.4274
Scottsdale	0.2249	> 3	0.2879	> 3	0.2931	> 1	0.6121
Mesa	0.2215	> 3	0.3666	-	0.3666	> 1	0.7731
Tempe	0.2822	> 3	0.4166	-	0.4166	> 1	0.8903
Chandler	0.3548	> 3	0.4996	-	0.4996	-	0.4996

3.3.1 Popularity recommendation

The Simple Recommender offers **generalized recommendations** to every user **based on overall restaurant popularity.** This is a lightning fast and not so preferred approach. However, there is **no personalization** involved with this approach which is a major drawback. The **basic idea** behind this recommendation is, "those restaurants that are **more popular and highly rated will have a higher probability of being liked by most customers."**

This approach can work for below cases:

- Restaurant recommendation to a new user.
- Recommendation to users whose user level details are not known to group them to a known user base and recommend restaurants in a collaborative way.
- Existing user may want to try different restaurants other than his regular ones and may try the restaurants listed in this section.

We use the following weighted average formula (IMDB's Weighted Rating formula) to construct the popularity chart

$$Weighted\ Rating(WR) = \left(\frac{Rev\_cnt}{Rev\_cnt+n} . Res\_rating\right) + \left(\frac{n}{Rev\_cnt+n} . Avg\_rating\right)$$

Where,

'Rev_cnt' is the number of Review_counts for the Restaurant

'n' is the minimum count required to be listed in the recommendation

'Res_rating' is the overall rating of the Restaurant in Business dataframe

'Avg_rating' is the mean Rating across all the restaurants for the city

The Restaurants recommended in this section will be the same for any user for a given city

3.3.2 Content based search Engine

Filter based search engine or Content based search engine brings in customized restaurant recommendation based on the criteria selected. The preferences of a user could change without the knowledge of the recommender system. To allow more freedom of choices to users, we are building a content based search engine which would filter out only those restaurants that would satisfy the conditions keyed in by the user.

Once we have the list of filtered restaurants, we would take the weighted average of all the restaurants based on the review counts against each restaurant as we calculated for the popular based recommendation model. If the user has already visited any of those restaurants, we would consider the user's rating for that particular restaurant before picking the top 3 restaurants. By considering the user's past ratings on specific restaurants, we are making sure that if the user has had bad experience with a particular restaurant which has overall good rating, we will not be recommending the same restaurant again.

This kind of search engine could help users who are categorized under having **'cold-start problem'**.

3.3.3 Collaborative filter based recommendation

The collaborative filter based recommendation works based on the similarity measurement between users and/or items. There are predominantly two types of collaborative filter, user similarity based (2 similar users prefer same restaurant if one user has rated the restaurant high and other has not visited) or item similarity based (2 similar restaurants are recommended to a user, if has rated one restaurant high while he has not visited the other).

To enable this recommendation, we used the library called 'surprise' along with common techniques namely SVD (Singular Value Decomposition), KNN (K-Nearest Neighbor). The data was split in to trainset and test set by 5-fold cross validation technique and the error metrics are evaluated. Below are the steps followed for both the techniques.

SVD (Singular Value Decomposition) - Here the dimensions of the user-restaurant rating matrix is reduced for faster and evaluation. Using GridSearch, we find the hyper parameters and evaluate the SVD technique on the data using 5-fold cross validation. The error metrics used to test the results are RMSE (Root mean Square Error) and MAE(Mean absolute Error)

KNN (K-Nearest Neighbor) – Here we used different item similarity based distance calculation measures to identify the closest neighbor for a restaurant that a user has already rated high. Using distance measures like 'cosine similarity', 'Mean squared difference'(MSD), 'Pearson correlation coefficient', 'shrunk Pearson correlation coefficient' the closest neighbors were calculated for the trainset and validated on the test set using 5-fold cross validation.

Based on the error metrics SVD was chosen. The test set was created with all users and the restaurants not rated yet so as to identify the top 6 recommendation for each user. We would compare the top 6 recommendation against the Content based search results to avoid overlaps between restaurants. We could still have some overlaps between popularity based recommendation and the collaborative filter based model.

3.4 Improving Restaurant Ratings through Text Analytics

Key Steps:

1. Determining the most frequent words appearing in the positive and negative review for different price range. (**3b_NLP_Wordcloud.ipynb**)
2. Identifying the most negative 2-word combinations(bigrams) for each restaurant (**3c_NLP_Bigrams.ipynb**)
3. Identifying the topics involved for each single review (**3d_NLP_Topicmodeling.ipynb**)

3.4.1 Preprocessing of review text for Text Analytics

a) **Read the data:** The data is filtered and cleaned in the mongodb. Since the aim is to compare the text between high rated reviews and low rated reviews, the data is extracted with following rules:
   - High rated restaurants with average rating of 4 and 5.
   - Only reviews with rating 4 and 5 are taken for high rated restaurants.
   - Low rated restaurants with average rating of 1 and 2.

- Only reviews with rating 1 and 2 are taken for low rated restaurants.

b) **Separating low rated and high rated reviews**: The data separated into low rated reviews and high rated reviews into separate dataframes using the review rating.

c) **Identify the language:** A library called "langdetect" is used to detect the language text of each review.

d) **Filter the non-English reviews:** Since only English reviews texts are considered for further processing, non-English reviews were filtered. There were 246 non-English reviews that were filtered.

e) **Removing html tags:** Regular expressions were used to remove the html tags in the review text.

f) **Removing special and numeric characters:** Since special characters do not play any role in text analytics, they were removed. Additionally, numbers in review text cannot be interpreted by the text analytics algorithm, they have been filtered out.

g) **Removing extra spaces:** If there are additional spaces in review text, while calculating bigrams, white spaces are coming as additional tokens. Hence white space is removed using regular expression.

h) **Lemmatization:** WordNetLemmatizer from "NLTK.stem" library was used to lemmatize the words. Lemming was preferred over stemming as the lemmatized words were more meaningful words. Part of speech (POS) identification was performed to identify contextual meaning of the words and then the lemming was performed.

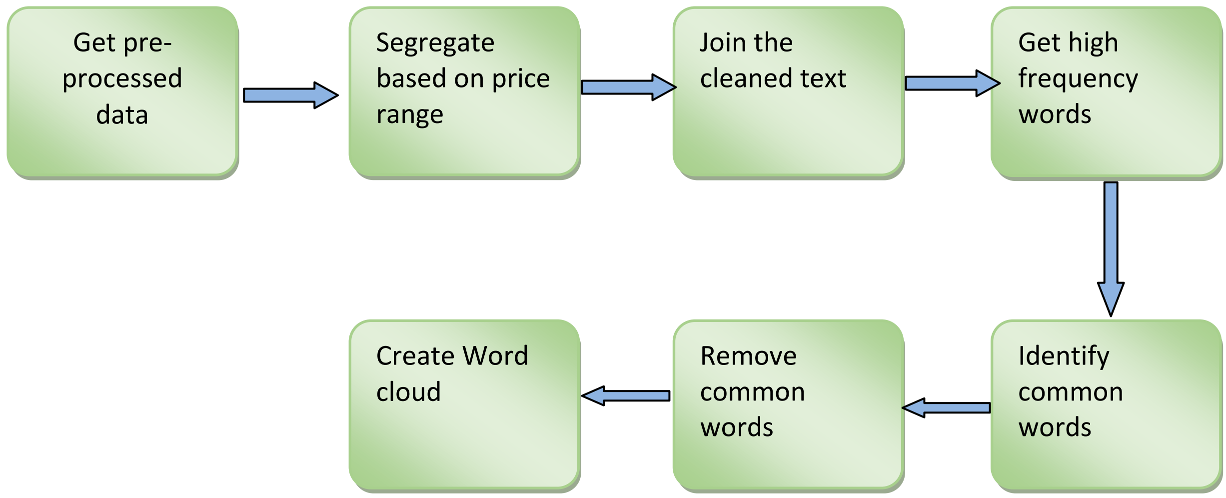
i) **Removing stop words:** "nltk.corpus.stopwords" was used to identify and filter out the stop words.

j) **Lower casing:** to normalize every data point, each of the text is brought to lower case.

Note:

1. No tokenization was performed in preprocessing as that's task specific.
2. **We have made extensive use of Dask libraries to improve the overall performance**.

3.4.2 Creating Word Cloud



a) **Input Data:** The data for word cloud is the preprocessed data. 2 dataframes were created, one belonging to the low rated reviews for low rated restaurant and other high rated reviews for high rated restaurants.

b) **Segregation based on price range:** The input data is segregated based on price-range 1-4. As a result, overall 8 separate dataframes are created.

c) **Joining the reviews:** Since the agenda is to identify unique words for each of the low-rated and high rated restaurant belonging to a specific price range, all the review text were joined to create a single text for each price range by restaurant ratings.

d) **Get high frequency words:** To create the word cloud, it's important the words that are unique and are occurring very frequently in the review text are taken. It has been identified that different price ranges have different frequencies of words.

1. Price range 1 with high rated review has 346553 unique words, and the word "place" is appearing 53477 times. At the same time reviews for restaurants belonging to price range 1 having low ratings has 13962 unique words and the word "order" is appearing 9964 times

	35	40	4	106	134	30	87	25	58	20	...	22676
word	place	food	good	great	get	i	go	order	time	love	...	crankiest
wordcount	53477	49130	40698	39306	32482	30821	29773	23834	22992	22902	...	1

2 rows × 34653 columns

	59	100	11	101	95	249	187	66	163	150	...	9823	9822	
word	order	get	go	food	time	i	place	say	one	s	...	informe	hola	ro
wordcount	9964	8288	7461	7264	5604	4559	4343	4242	4239	4030	...	1	1	

2 rows × 13962 columns

2. Price range 2, high rated reviews has 54574 unique words. The low rated reviews have 12624 unique words.

	244	82	1	16	179	187	103	199	284	163	...	33609	33608
word	food	great	place	good	go	get	i	service	time	come	...	obey	aladdin
wordcount	111951	107162	102518	91065	68397	65494	62203	61356	55307	54493	...	1	1

2 rows × 54574 columns

	107	23	60	114	143	73	67	149	29	215	...	8871	8870
word	food	order	get	go	come	time	place	service	say	us	...	unresponsive	devin
wordcount	5626	5258	5090	4598	3624	3315	3311	3203	2900	2701	...	1	1

2 rows × 12624 columns

3. Price range 3, high rated reviews has 18351 unique words. The low rated reviews have 1430 unique words. As can be seen, the word “order” with highest frequency is coming only 86 times in low rated reviews.

	142	20	21	202	79	231	3	363	151	439	...	12943	8379
word	food	great	service	place	go	good	restaurant	get	s	time	...	brat	rojo
wordcount	8840	8710	7038	6319	6260	6248	5284	4961	4796	4777	...	1	1

2 rows × 18351 columns

	16	2	72	13	53	24	66	58	33	75	...	787	788	792	79
word	order	get	food	go	give	say	back	chicken	time	ask	...	use	restroom	cat	wa
wordcount	86	68	66	50	46	46	41	41	40	36	...	1	1	1	

2 rows × 1430 columns

4. Price range 4, high rated reviews has 6634 unique words. The low rated reviews have 721 unique words.

	1	135	0	188	20	143	6	60	52	62	...	4154	4156
word	food	service	great	s	go	steak	good	place	restaurant	get	...	swap	startle
wordcount	1086	1018	1003	905	773	754	708	693	634	614	...	1	1

2 rows × 6634 columns

	24	236	73	293	327	277	41	266	74	333	...	415
word	food	get	employee	location	order	go	service	like	often	chick	...	woman
wordcount	30	29	21	19	16	15	15	15	14	14	...	1

2 rows × 721 columns

e) **Identify common words**: For the same price range the common words are identified between low rated and high rated reviews. Following is a small snippet of the common words:

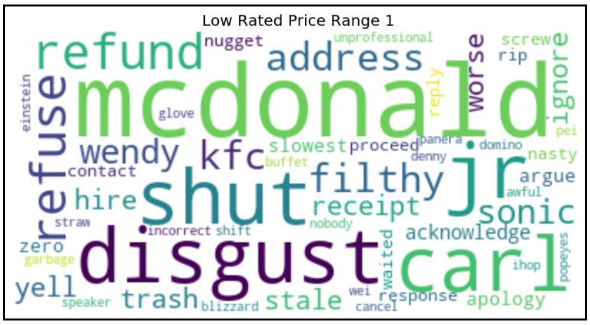
```
['hour', 'friday', 'sat', 'drive', 'thru', 'answer', 'look', 'forward', 'entire', 'franchise', 'go', 'business', 'i
t', 'll', 'fault', 'close', 'light', 'off', 'explain', 'you', 'please', 'thank', 'stop', 'location', 'back', 'burge
r', 'unfortunately', 'tell', 'longer', 'make', 'them', 'so', 'star', 'cheese', 'combo', 'receive', 'buck', 'change',
'up', 'mayo', 'fill', 'mess', 'together', 'order', 'super', 'pickle', 'give', 'three', 'needless', 'say', 'thing', 'f
ry', 'hot', 'normal', 'experience', 'much', 'this', 'particular', 'problem', 'old', 'employee', 'seem', 'like', 'don
t', 'care', 'save', 'crave', 'next', 'time', 'late', 'night', 'out', 'decide', 'get', 'food', 'here', 'coupon', 'mil
e', 'high', 'bacon', 'cheeseburger', 'menu', 'anymore', 'week', 'ago', 'new', 'item', 'replace', 'ask', 'extra', 'cri
spy', 'home', 'guess', 'what', 'almost', 'bun', 'want', 'would', 'think', 'that', 'also', 'onion', 'ring', 'never',
'again', 'away', 'cut', 'chicken', 'tender', 'husband', 'delicious', 'eat', 'meal', 'feel', 's', 'money', 'realize',
'fast', 'company', 'oh', 'maybe', 'top', 'every', 'one', 'know', 'not', 'worth', 'fine', 'issue', 'soda', 'ice', 'lit
erally', 'dirty', 'see', 'inside', 'complain', 'management', 'bring', 'come', 'no', 'place', 'first', 'door', 'two',
'din', 'area', 'yet', 'least', 'table', 'need', 'help', 'counter', 'leave', 'avoid', 'school', 'original', 'owner',
```

f) **Identify the unique words:** For low rated restaurants unique words are identified by filtering out the common words. Similarly, unique words are identified by filtering out the common words from the high rated reviews text.

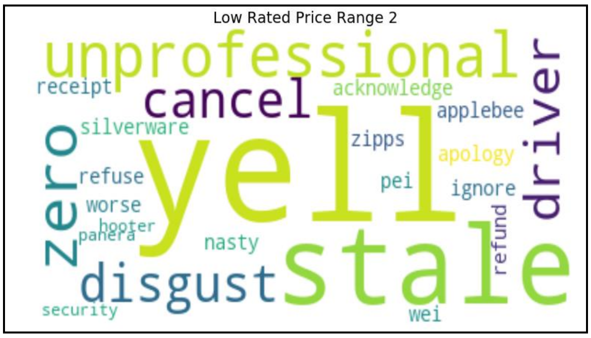
g) **Word Cloud:** Created 2 distinct word clouds for each price range for low rated restaurant with low rated reviews and high rated restaurant with high rated reviews.

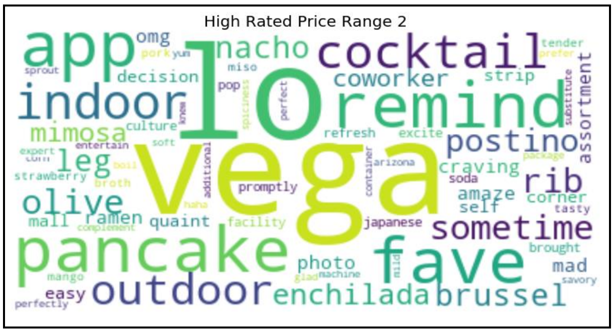
I. **Price range 1**:

From below word clouds it can be seen that high rated reviews frequently show the words like "remind" (old memories), "enchilada" (a dish). The low rates reviews are consistently talking about "McDonald", "KFC"(chain of restaurants), "disgust", "Filthy" etc.

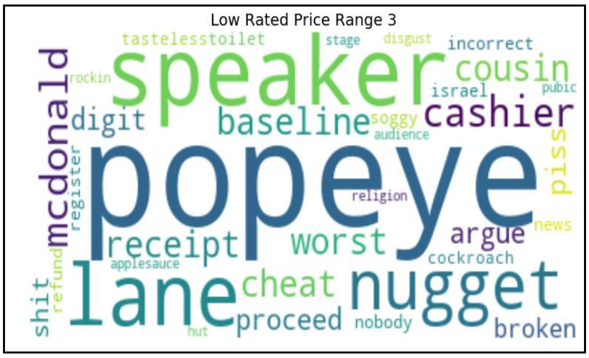


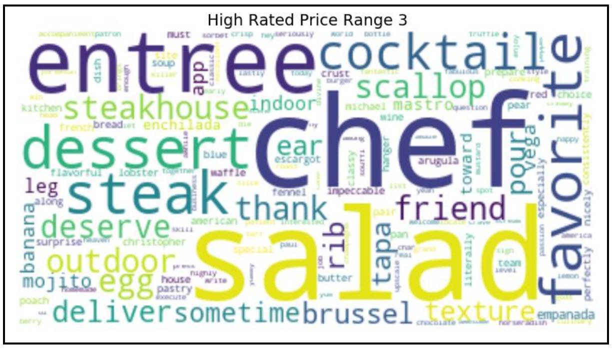
II. Price Range 2:



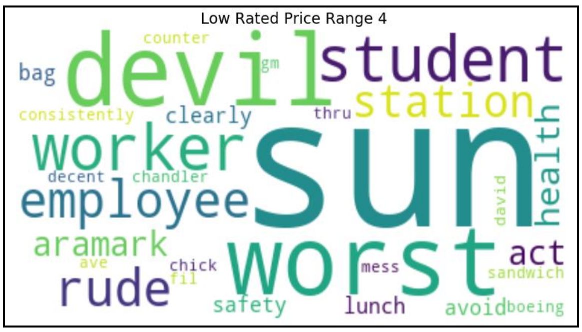


III. Price range 3:

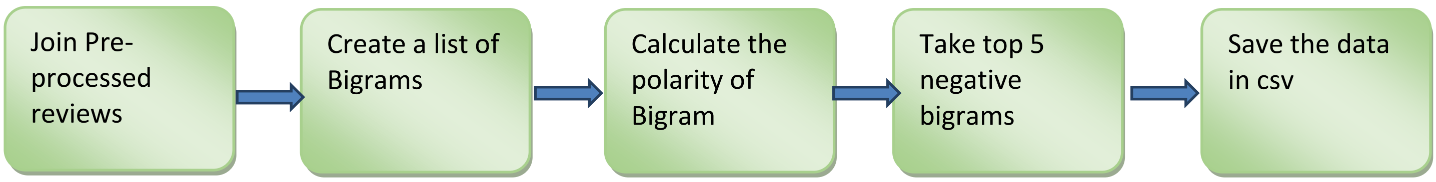




IV. Price Range 4:

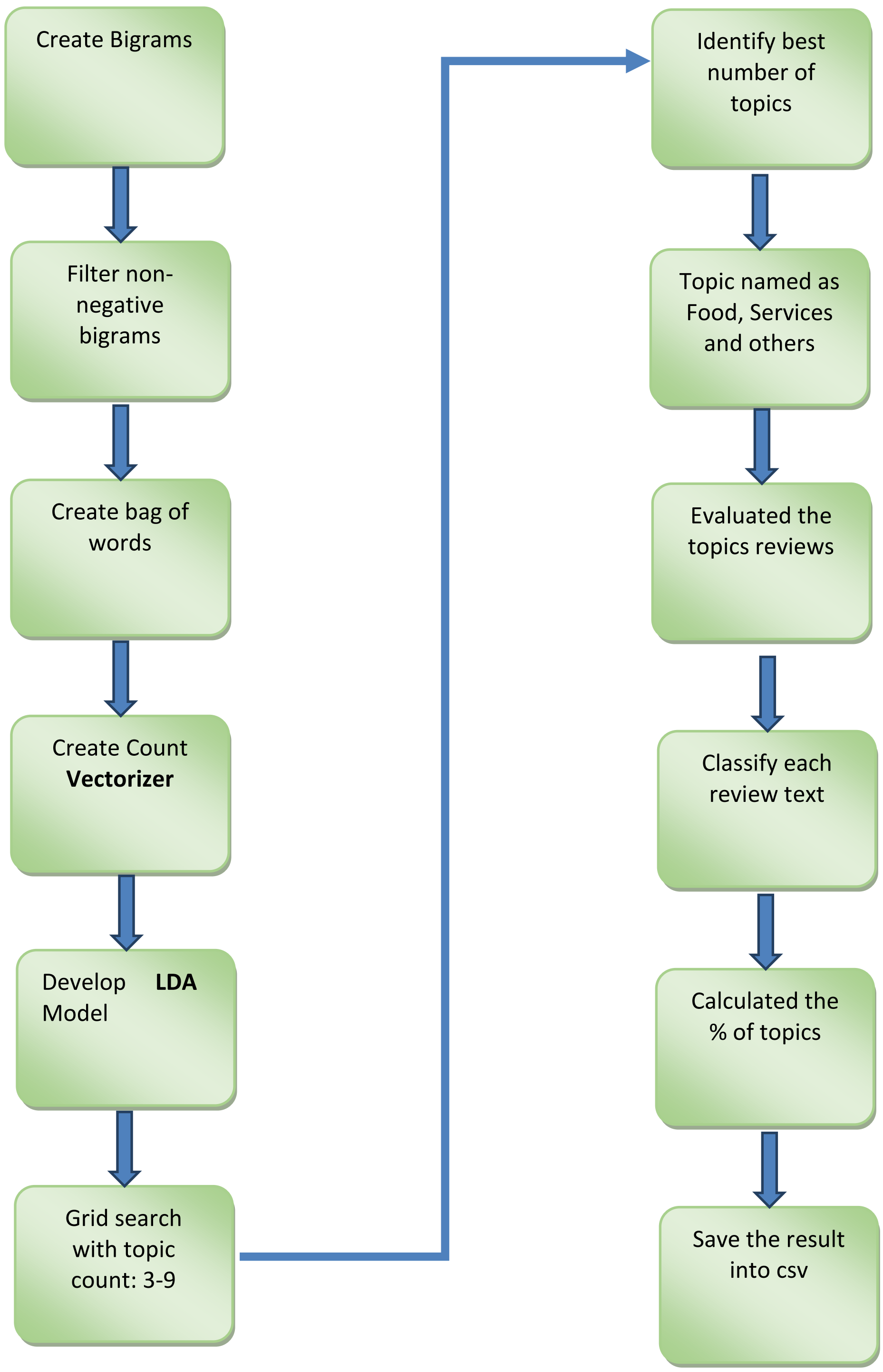


3.4.3 Creating negative Bigrams



- **Input data:** The cleaned and preprocessed data for each business and reviews for the business is taken for creating bigrams.
- **Join data:** All the reviews belonging to a specific business is joined.
- **Create bigrams:** The bigrams are the 2 consecutive texts appearing together. The text is tokenized, and a list of bigrams is created by using 2 consecutive 2 tokens.
- **Calculate Polarity:** The library called "Textblob" is used to calculate the polarity of each bigram. The polarity score ranges from -1 to 1. A negative polarity denotes that the bigram is a negative sentiment text while 0 polarity means a neutral text.
- **Take top 5 Negative:** The bigrams for each of the business is sorted in ascending order that is low to high. All negative bigrams will come at the top. Top 5 bigrams are selected for each business.
- **Save the data:** Saved the data with top 5 bigrams and 5 sample review text for each business in a csv file for the user interface application.

3.4.4 Topic Identification and evaluation:



a) **Input data:** The cleaned and preprocessed data for each business and reviews for the business is taken for Topic identification.

b) **Bigrams:** For each of the review text, bigrams are created by tokenizing the cleaned text and taking 2 consecutive words.

c) **Filtering out non-negative bigrams:** using "Textblob" library, polarity score for each of the bigrams is calculated. Any bigrams that is neutral or positively inclined is removed from the list of bigrams for each of the review text. This way it's ensured that only negative text is used to identify the improvement areas for the businesses.

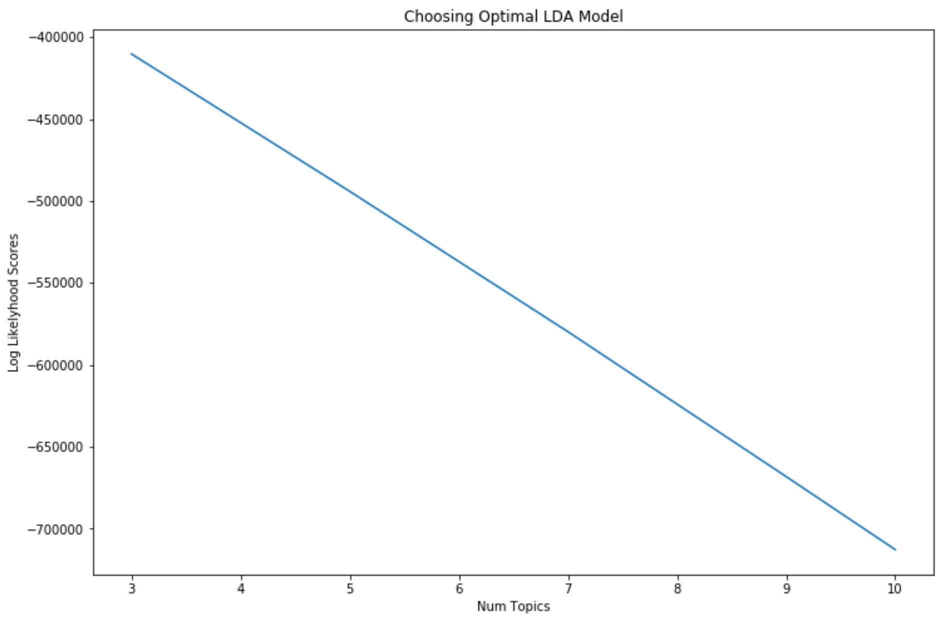
d) **Bag of words creation:** The negative bigrams are used to create the bag of words for the further processing.

e) **Count Vectorizer:** Created a count vectorizer from sklearn library for the reviews using the bag of words. Here the count vectorizer is used to ensure that the frequently occurring words are used in topic identification instead of TF-IDF which reduces the importance of frequently occurring words.

f) **Create LDA model:** Created an instance of the LatentDirichletAllocation() object from sklearn library with default parameters.

g) **Gridsearch:** The Latent Dirichlet Allocation Algorithm is used with Grid search with number of topics ranging from 3-9 to identify best number of topics.

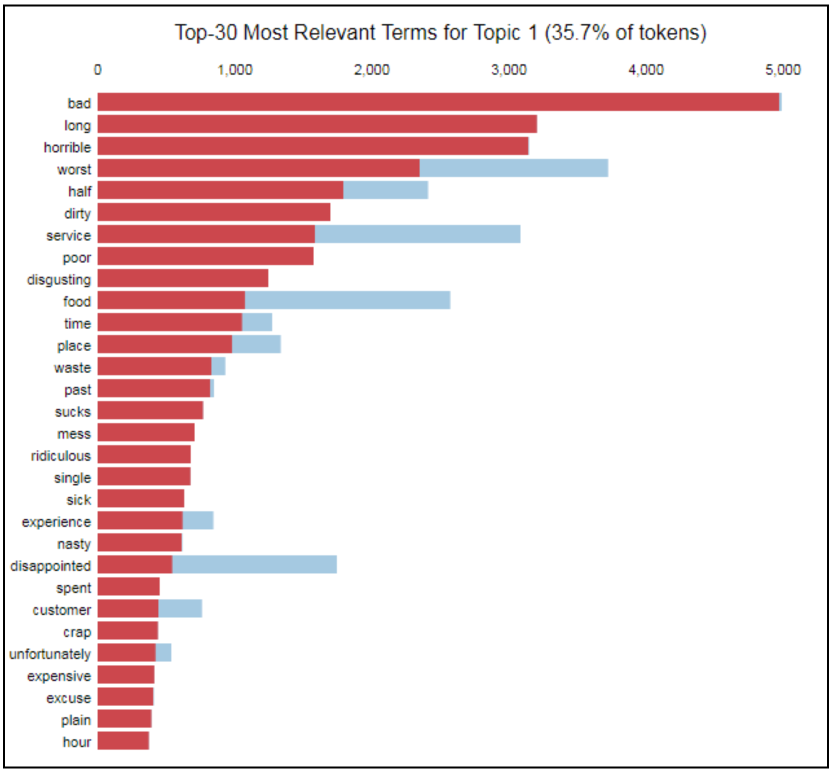
h) **Best number of topics:** A model with higher log-likelihoods a better model. The log likelihood scores with the number of topics is plotted to choose the best number of topics:



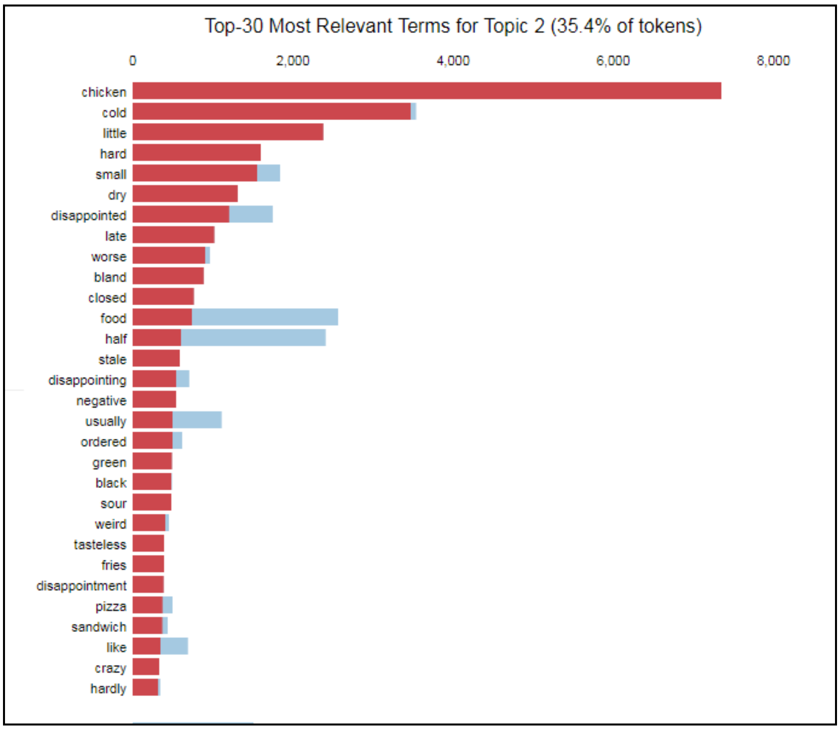
As can be seen from above, for number of topics as 3, the log likelihood score is maximum.

i) **Identifying the topics:** We have looked at the most occurring words for each of the topics. A sample of 15 frequent words for each topic can be seen here:

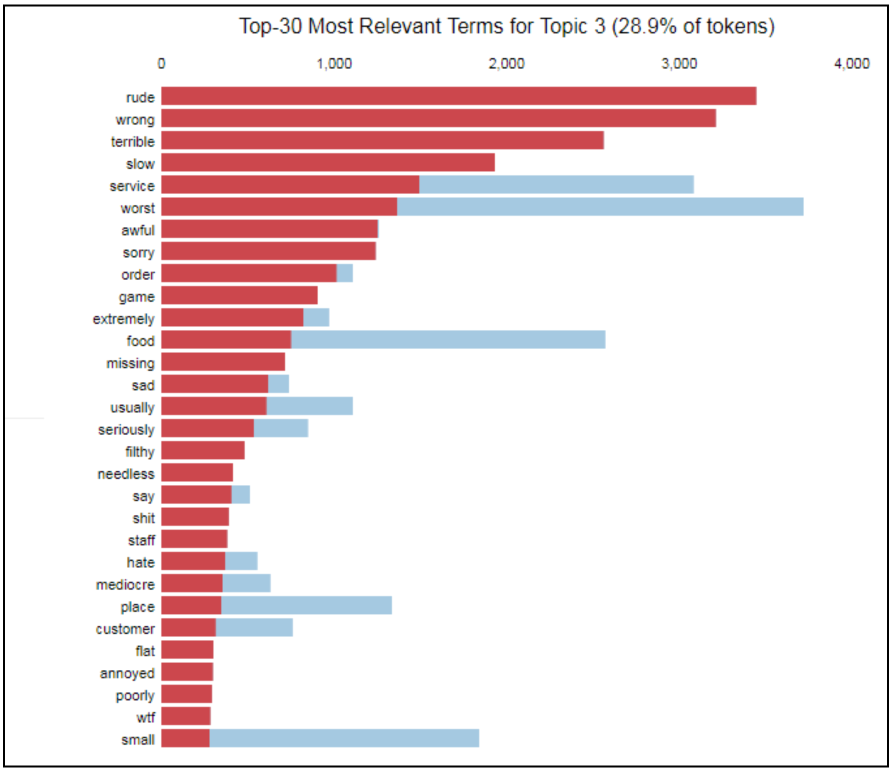
Topic 1:



Topic 2:



Topic 3:



From above its clear that the topic 2 is talking about "Food" and topic 3 is talking about "Service". Topic 1 is a mixed bag of words with food, service, ambience, waiting time, cost etc.

a) **Evaluated the LDA Model:** To evaluate the latent Dirichlet Allocation model we have manually labelled some of the reviews as food, service and others. With manually topic identified data we found the accuracy close to 69%.

b) **Labeling the reviews:** as we have identified the 3 topics to be food, service and others, we used the model to label each review. This way it's possible to calculate the number and percentage of reviews talking about the different areas of improvement.

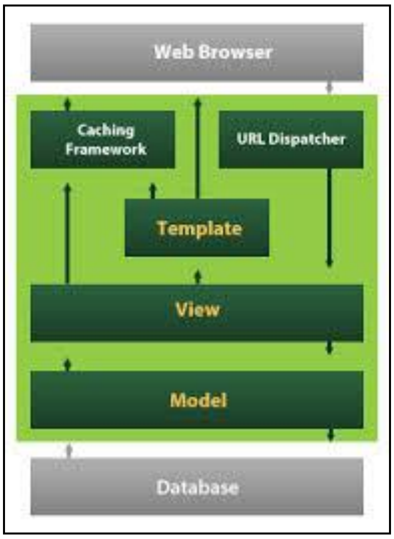
c) **Saving the data:** For each of the business the following data is saved

% of reviews having various topics is calculated.

It's saved in the csv file. The csv file is used to present the improvements areas in UI.

3.5 Integrating Model Output with HTML & Django

Traditional Django MVC architecture is followed in building the user interface. The technologies and their usage are as follows:-

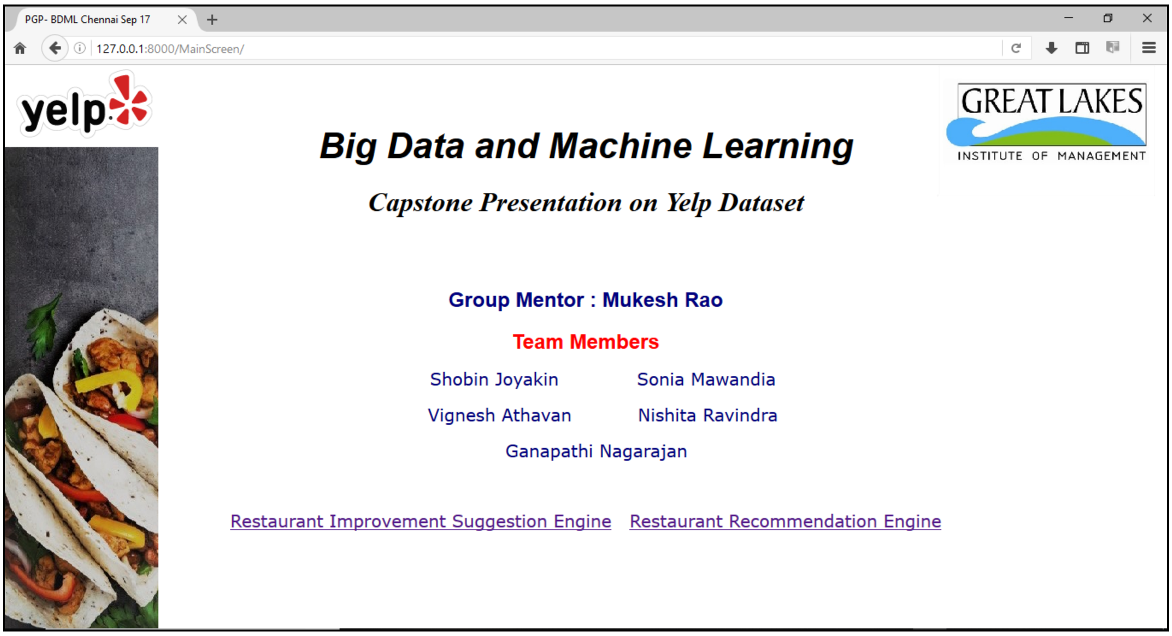


The following are the initial project setup files required to run and configure the application

- models.py – Classes and their member variables are declared here.
- settings.py – Configurations such as DB connections, third party frameworks, interceptors, urls, templates, Web Server Gateway Interface, etc are done here.
- views.py – Backed business logics and data processing are performed here
- urls.py - All the navigation webpages for the application is configured here
- wsgi.py - Django's primary deployment platform is WSGI, the Python standard for web servers and applications.

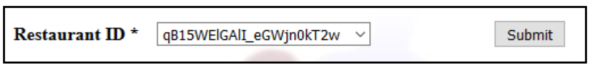
Once the project setup is done, the server is started by running the command **'python manage.py runserver'**. This invokes the developed html along with CSS, Javascript files that are placed in the templates folder when displaying the main page URL.

Main HTML page:

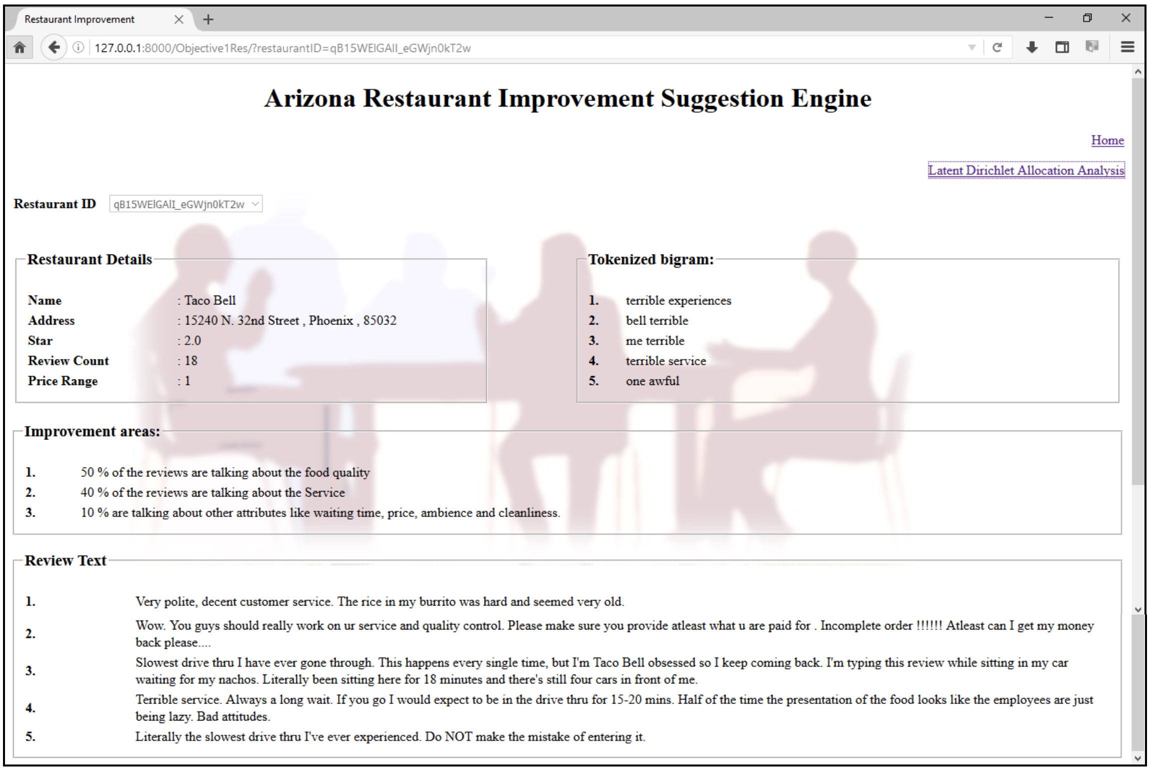


Objective 1 Request: (Restaurant Improvement Suggestion Engine)

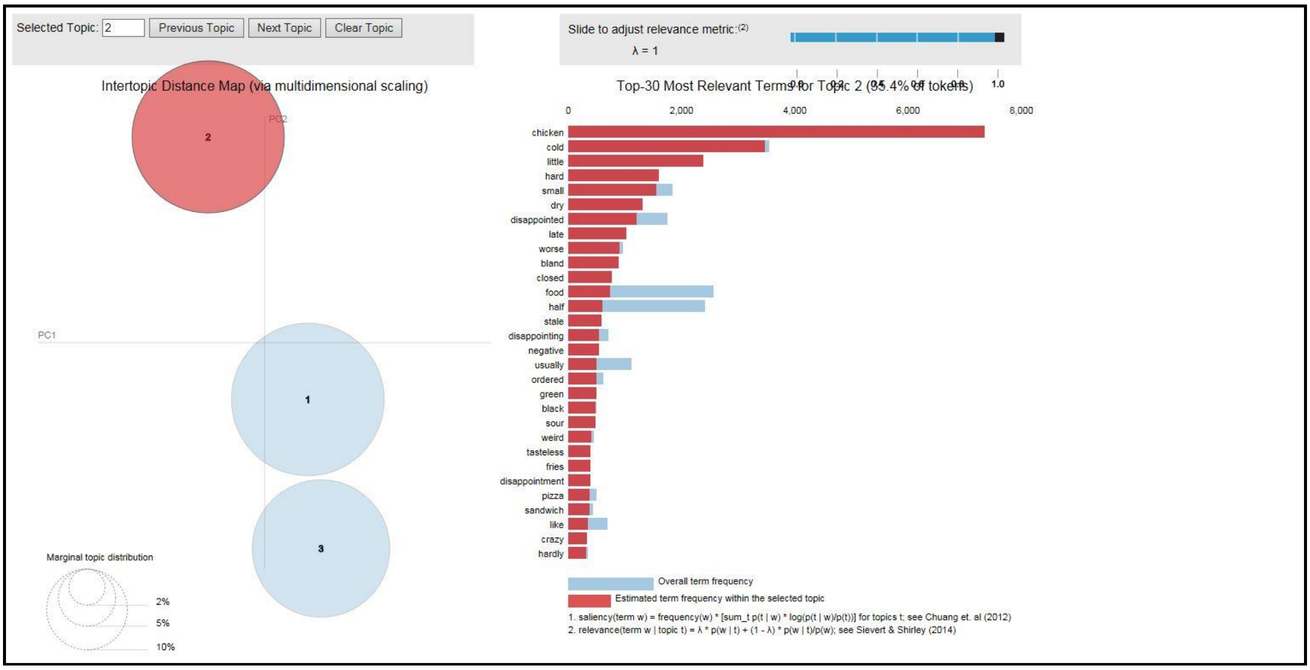
Select the Restaurant ID to display the improvement areas.



Objective 1 response:



LDA_analysis



Objective 2: (Restaurant Recommendation Engine)

The Recommendation Engine will display recommended restaurants for a specific user in a city.

- Content Based search Engine: For a specific city and user preferences the recommendations are derived
- Item similarity based collaborative Recommendation: Based on Restaurants already visited by the user, new restaurants of similar kind are recommended.
- Popularity based Recommendation: Popular restaurants in a city/location are recommended.

Request 1:

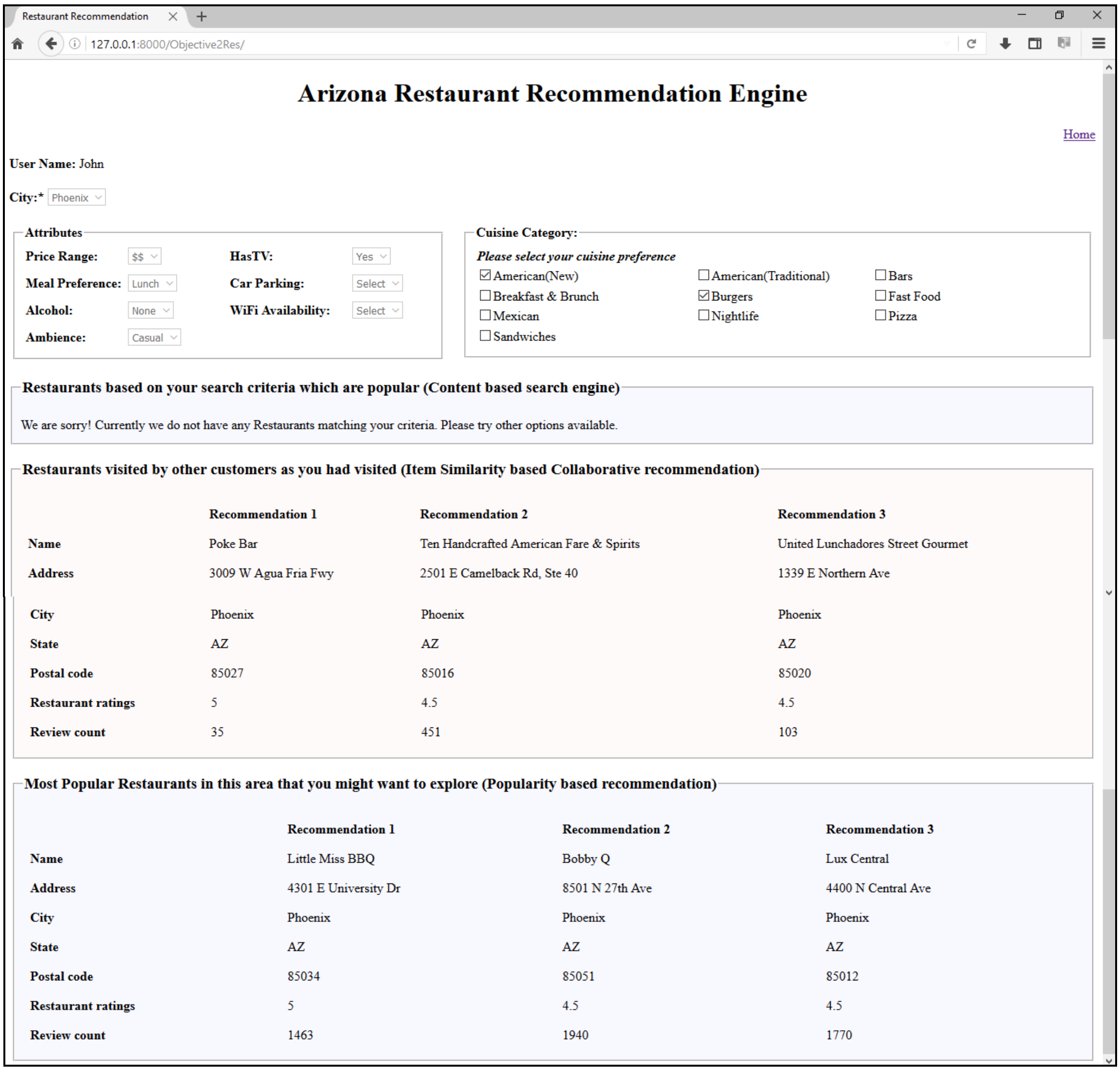
Shows the functioning of Collaborative and popularity based recommended filter

Request 2:

Shows the functioning of Content based search engine and popularity based recommended filter.

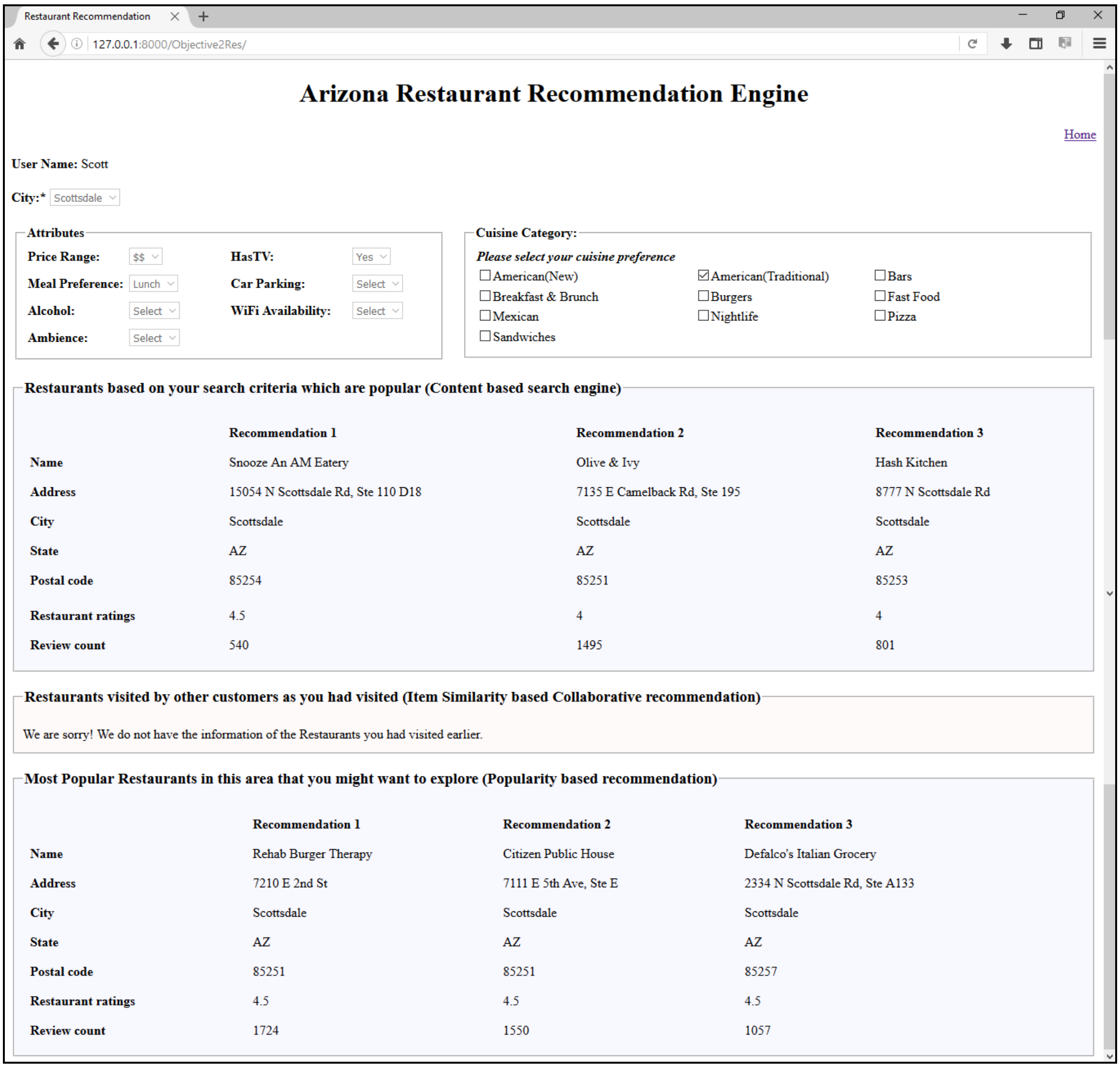
Objective 2- request 1:

Objective 2- response 1:



Objective 2- request 2:

Objective 2- response 2:



Restaurants based on your search criteria which are popular (Content based search engine)

	Recommendation 1	Recommendation 2	Recommendation 3
Name	Snooze An AM Eatery	Olive & Ivy	Hash Kitchen
Address	15054 N Scottsdale Rd, Ste 110 D18	7135 E Camelback Rd, Ste 195	8777 N Scottsdale Rd
City	Scottsdale	Scottsdale	Scottsdale
State	AZ	AZ	AZ
Postal code	85254	85251	85253
Restaurant ratings	4.5	4	4
Review count	540	1495	801

Restaurants visited by other customers as you had visited (Item Similarity based Collaborative recommendation)

We are sorry! We do not have the information of the Restaurants you had visited earlier.

Most Popular Restaurants in this area that you might want to explore (Popularity based recommendation)

	Recommendation 1	Recommendation 2	Recommendation 3
Name	Rehab Burger Therapy	Citizen Public House	Defalco's Italian Grocery
Address	7210 E 2nd St	7111 E 5th Ave, Ste E	2334 N Scottsdale Rd, Ste A133
City	Scottsdale	Scottsdale	Scottsdale
State	AZ	AZ	AZ
Postal code	85251	85251	85257
Restaurant ratings	4.5	4.5	4.5
Review count	1724	1550	1057

4. EVALUATION

4.1 Evaluation of Restaurant Recommendation to Users

In collaborative filtering, there are many metrics for evaluating recommender systems. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are among the most important ones. To calculate MAE/RMSE, predicted ratings are compared with their corresponding true ratings.

We used Python "Surprise" library for our collaborative filtering. This library supports several algorithms for collaborative filtering out of which we chose SVD and KNN.

GridsearchCV:

For SVD we used GridsearchCV to identify the best hyper parameters for the algorithm.

The following hyper parameters seem to bring out the best score.

{'n_factors': 140, 'n_epochs': 40, 'lr_all': 0.003, 'reg_all': 0.1}

	Fold1	Fold2	Fold3	Fold4	Fold5
RMSE	**1.16632661**	**1.1781625**	**1.19805187**	**1.19853795**	**1.20440856**
MAE	**0.93165396**	**0.94119723**	**0.94512725**	**0.95047884**	**0.95037895**

Best RMSE Score is **1.1947**

For KNN the following parameters seems to bring out the best RMSE score

sim_options = {'name': 'cosine', 'user_based': False }

	Fold1	Fold2	Fold3	Fold4	Fold5
RMSE	**1.41392439**	**1.3702721**	**1.3941455**	**1.37994864**	**1.37225716**
MAE	**1.00495541**	**0.97666753**	**1.00180278**	**0.99025588**	**0.97540987**

Out of the two algorithms we tried, SVD has the best score for RMSE.

RMSE percentage error = (1.1947 / 5) * 100 = **23.8 %**

4.2 Evaluation of Restaurant Improvements

Grid Search and LDA –LDA model was trained on the review text with the GridsearchCV which showed the best 'n' (topics) to be 3 because of the good "log likelihood" score. All the review text were classified into one of the three topics based on the highest probability score

Manual Evaluation - We randomly took some 110 review text and manually labeled them to be in one of the three topics (Food, Service or others). These reviews are then passed to the LDA model which classified them to be in one of the topics. **The Confusion matrix returned an accuracy score of 67.49%.**

5. LEARNING

The Following are the Key learnings that we learned while doing this capstone project

1. Connecting Jupyter Notebook to Mongo DB and retrieving the details for further processing.
2. Working on data stored as collections in MongoDB by applying filters, aggregate function, joins, etc.
3. Handling nested JSON data in python
4. Processing Dask library for faster processing.
5. Content based search engine using customized algorithm.
6. Surprise algorithm for item similarity recommendation
7. Hosting the pickled models and data into Django framework to interact with HTML UI

6. POSSIBLE ENHANCEMENTS

1. Creating user similarity based recommendation for users who were filtered out based on low review count.
2. In memory processing of recommendation engine using Spark on a larger data.
3. Creating clusters of cities for the entire country, based on different Tier level and creating individual models for each Tier based on data being available.
4. Add more Filter criteria to search Engine based on clean data being available.
5. Analyzing areas of improvement for high rated restaurants considering their low rated reviews.
6. Identify topics for Restaurants in other cities and states.

7. REFERENCES

Recommender Systems:

- **Surprise Library** - https://surprise.readthedocs.io/en/stable/getting_started.html
- **Book** – Chapter 9 (Recommender Systems) of "Mining Massive Data Sets"
- https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/
- https://datascienceplus.com/building-a-book-recommender-system-the-basics-knn-and-matrix-factorization/
- https://github.com/jalajthanaki/Movie_recommendation_engine/blob/master/Movie_recommendation_engine.ipynb
- **Paper** – "Cities and Restaurants" - Ahmet Oruc, Sergen Topcu, Denizkaan Araci in Machine Learning Project, Department of Computer Engineering Hacettepe University, Ankara, TURKEY

Text Analytics and Topic Modeling:

- **LDA**: https://www.youtube.com/watch?v=BuMu-bdoVrU
- **Dask**: https://towardsdatascience.com/how-i-learned-to-love-parallelized-applies-with-python-pandas-dask-and-numba-f06b0b367138
- **Dask**: https://www.analyticsvidhya.com/blog/2018/08/dask-big-datasets-machine_learning-python/
- **SKLearn LDA:** http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

Django Development:

- **Learner's Guide** - https://docs.djangoproject.com/en/2.1/
- https://github.com/django/django
- https://www.tutorialspoint.com/django/