Location-Based Restaurant Recommendation System

Food is one of the most basic needs for any human in their daily routine and the reason behind choosing this area to work on our project was to advance the already existing system to improve the response that is generated nowadays and make it more efficient for the user to have a better experience from the Information that is already available based on the different kind of the ratings provided by the customer and how the people can get the best recommendation based on the choices they have in terms of food preference, ambiance and depending moreover on the location of the restaurant from its source to destination based on the current search made by the users. Moving on further we will see the process of our project on different resources that we have used and delivering the best result based on the dataset available to us with the help of the professor allowing us to move further with the suitable data

Data Collection

Initially, we had been planning to work on the Yelp data that is available on open platforms on the internet. Still, we ended up having trouble accessing that information since maybe the systems were not compatible to handle the amount and size of information. So we decided to try all the ways possible to try and use that data based on the needs of the project and we found similar data with a small size from another source which was data world with a relevant size supporting the system and with relevant data which we desired to work.
The data size was 445.3 MB and had four files with 90 columns in the form of CSV files including multiple different types like strings, numerical values, decimal, and integer values.

Data Cleaning

Data Cleaning was one of the most tedious processes for us and it almost took us two weeks to resolve the Data set as it has multiple missing values. Since we started working on data by using apache spark to move further with the project we found that there are multiple missing values even though the data was not as big as Yelp. We have been using three files from the four files which had a piece of information about the restaurant, its reviews, and the user guide. We have found several missing values and have dropped a few of the columns in order to have a quick response from the Big Data information that was available and make it more user-friendly with a good interface.

Data Extraction

Data Extraction was about taking and considering the columns from the table which were needed and made the process of searching for a restaurant better. Data frames were created to look for the type of information available, the number of rows in the table, and the values that are needed to be replaced in order to have the proper code and the correct analysis while running the query.

Data Interpretation

Data Interpretation is so interesting to know about the analysis that we got from data after creating the clusters and its visualization is shown in the figures below. It is easily visible that there are few places that have more restaurants in comparison to other states since the data is filtered but it helps us with the best response when we search for Philadelphia.

Data Visualization

Figure 1: Viewing the top 20 restaurants with their ratings

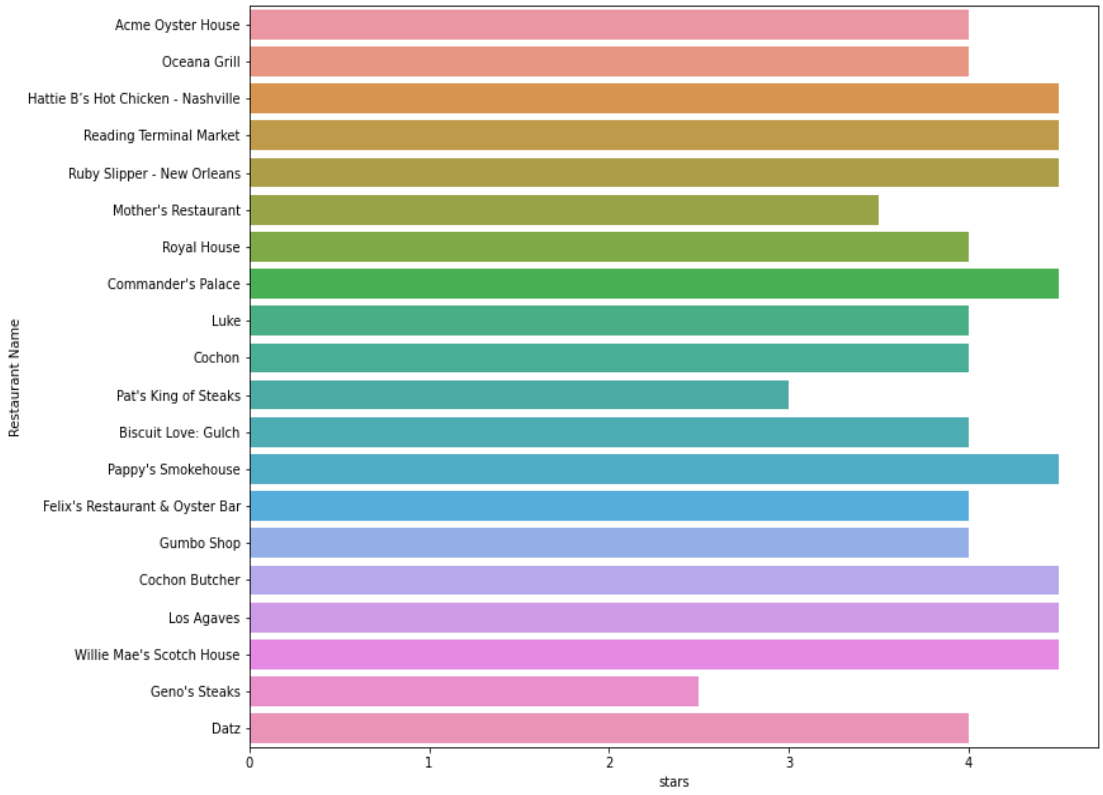


Figure 2: Viewing all the restaurants in our dataset in their spatial distribution

Figure 3: Viewing the restaurants in Philadelphia in a spatial distribution

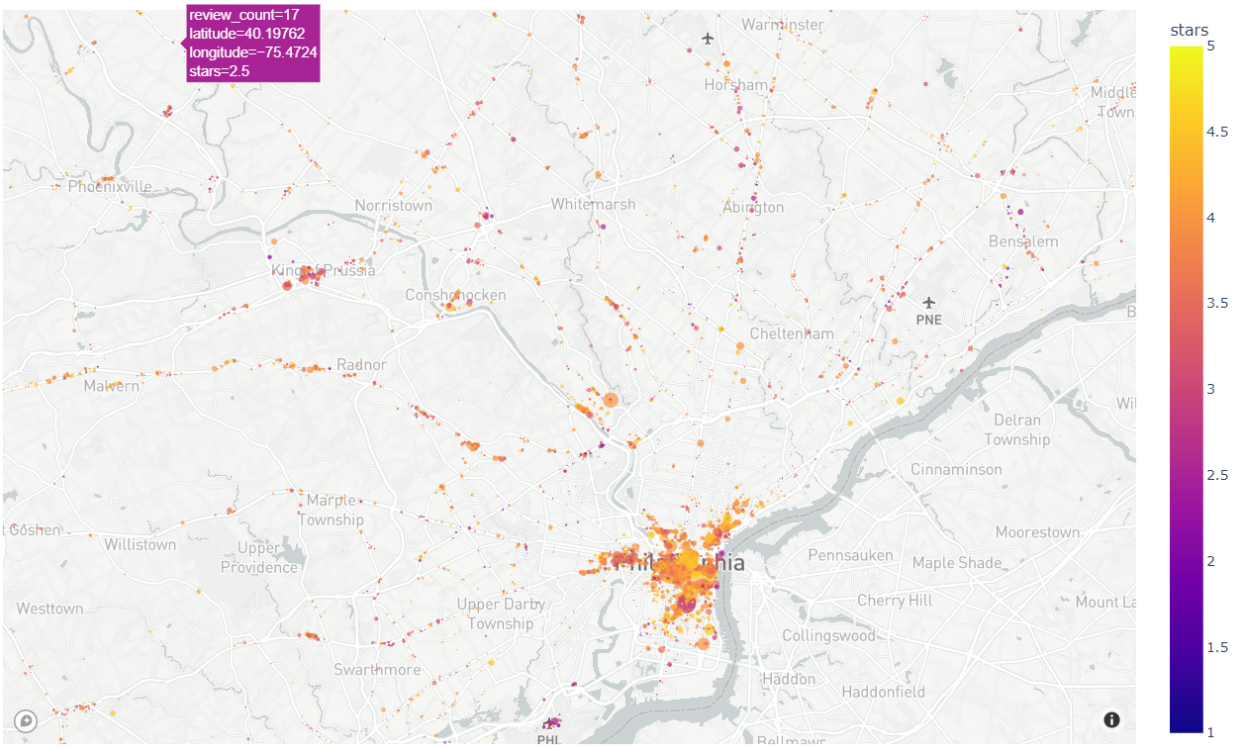


Figure 4: Determining the optimal number of clusters using the Elbow method

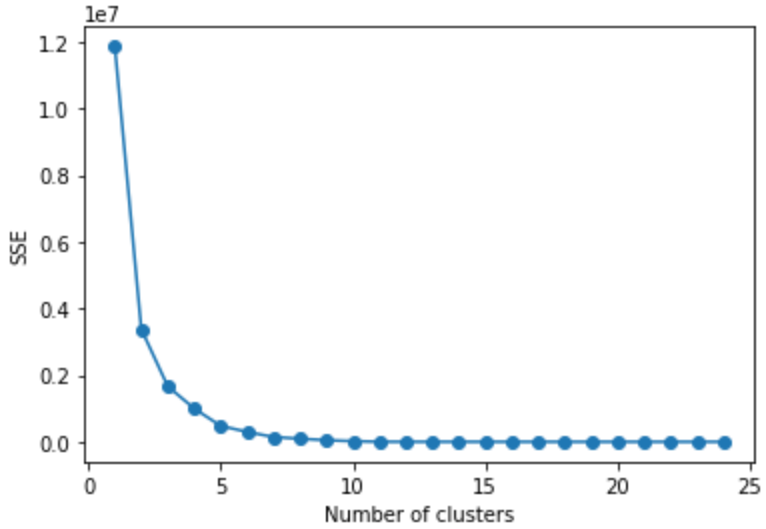


Figure 5:Visualizing the clusters on a map to see their spatial distribution

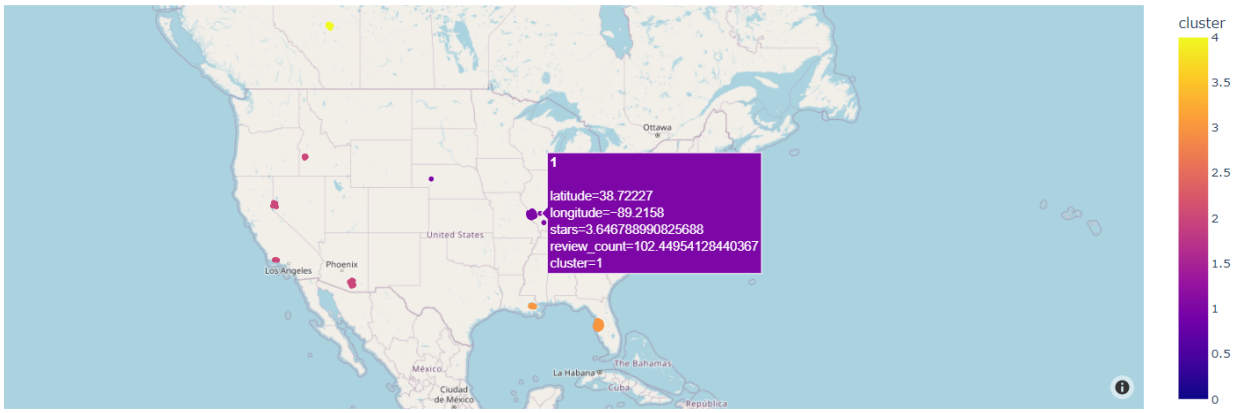
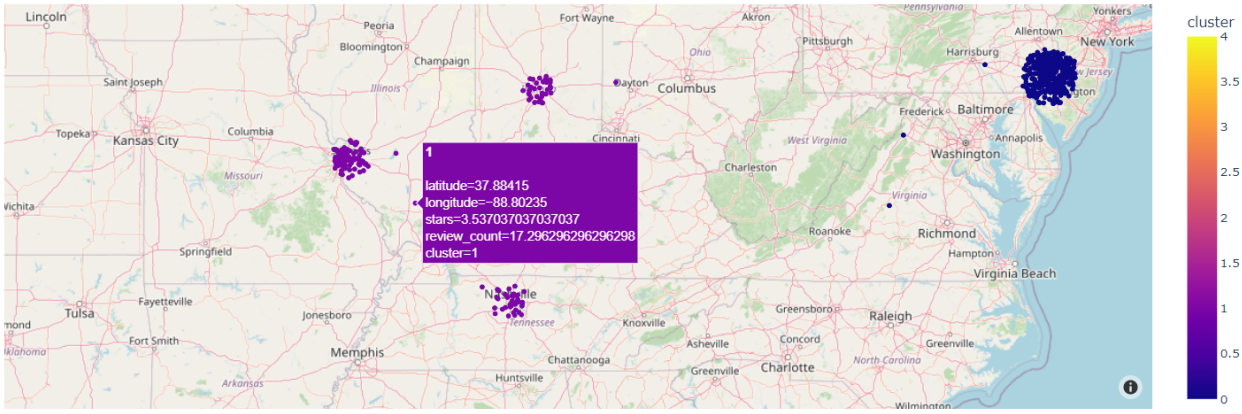


Figure 6: Zoomed-in version of spatial distribution



Output

```
tucson = get_recommendations('tucson')
df = pd.DataFrame(tucson)
df
```

```
<command-3931370796390141>:15: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

	Restaurant Name	city	latitude	longitude	stars	review_count	distance
0	Ciao Down	Tucson	32.222601	-110.974715	4.5	12	0.033054
1	Cafe A La Cart	Tucson	32.223630	-110.975401	4.5	364	0.098636
2	TallBoys	Tucson	32.224183	-110.974056	4.0	165	0.163222
3	LaCo	Tucson	32.224197	-110.974065	4.5	469	0.164258
4	Pie Bird Bakery and Cafe	Tucson	32.223954	-110.973406	4.0	77	0.180962
5	Arizona Bagel & Deli	Tucson	32.223954	-110.973406	3.5	13	0.180962
6	Wasted Grain	Tucson	32.221492	-110.975896	2.0	6	0.182878
7	Five To Oh! Coffee	Tucson	32.223481	-110.973005	4.5	6	0.185876
8	Ceres	Tucson	32.224543	-110.974505	4.5	53	0.188084
9	Baggin's Gourmet Sandwiches	Tucson	32.222520	-110.972339	3.0	32	0.239287

The recommendations are sorted by distance from the input city, then by the number of reviews, and finally by the average rating.

Lets verify some other cities as well.

```
nash = get_recommendations('Nashville')
df = pd.DataFrame(nash)
df
```

```
<command-3931370796390141>:15: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

	Restaurant Name	city	latitude	longitude	stars	review_count	distance
0	Acme Feed & Seed	Nashville	36.162016	-86.774451	4.0	1731	0.032114
1	National Underground	Nashville	36.161795	-86.774466	2.5	126	0.055622
2	Nashville Underground	Nashville	36.161771	-86.774496	3.0	266	0.058979
3	Hard Rock Cafe	Nashville	36.162405	-86.774979	3.0	312	0.062730
4	The Comedy Bar	Nashville	36.162405	-86.774979	4.5	6	0.062730
5	Hard Rock Cafe Sales	Nashville	36.162405	-86.774979	4.0	6	0.062730
6	Rock Bottom Restaurant & Brewery	Nashville	36.161750	-86.774800	3.0	764	0.073877
7	Big River Grille & Brewing Works	Nashville	36.161694	-86.774732	3.5	151	0.075587
8	Famous Nashville	Nashville	36.162290	-86.775283	3.5	109	0.088403
9	Wildhorse Saloon	Nashville	36.162766	-86.775178	3.5	559	0.095890

```
phil = get_recommendations('Philadelphia')
df = pd.DataFrame(phil)
df
```

```
<command-3931370796390141>:15: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

	Restaurant Name	city	latitude	longitude	stars	review_count	distance
0	Beneluxx	Philadelphia	39.952335	-75.163789	3.5	87	0.048682
1	Renaissance Sausage Truck	Philadelphia	39.952335	-75.163789	4.0	22	0.048682
2	59 Cafe	Philadelphia	39.952335	-75.163789	4.0	7	0.048682
3	Lale Gourmet Kebab	Philadelphia	39.952335	-75.163789	2.5	5	0.048682
4	Dilworth Park Café	Philadelphia	39.953245	-75.164426	3.0	10	0.096139
5	Rosa Blanca Cafe	Philadelphia	39.953186	-75.164531	3.5	19	0.099896
6	Dunkin'	Philadelphia	39.952588	-75.162245	3.0	22	0.110251
7	Delicias	Philadelphia	39.951818	-75.164107	3.5	61	0.112279
8	Pasto	Philadelphia	39.952809	-75.162199	4.5	5	0.113541
9	City Hall Coffee House	Philadelphia	39.951669	-75.163983	4.5	9	0.123572