Big Data Project REPORT

CS-GY-6513

Analyzing Reviews on Yelp Restaurants Dataset

Team Members:

Srividhya Ravichandran (sr5962)
Viswanath Nagarajan (vn2065)
Priyanka Shelar (ps4497)
Anand Pitale (avp5522)

Acknowledgement

We are highly indebted to Prof. Juan Rodriguez for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We learnt various Big Data technologies and their implementations while developing this project. We will certainly remember this experience as we move forward with similar projects in the industry.

Table of Contents

1. Introduction

Yelp is an application to provide the platform for customers to write reviews and provide a star-rating. Research indicates that a one-star increase led to 59% increase in revenue of independent restaurants. Therefore, we see great potential in the Yelp dataset as a valuable insights repository.

The main purpose of our project is

- to conduct thorough analysis on different cuisine types of restaurants and the reviews they have received on Yelp, to figure out whether the customers like the food or not.

- Building a sentiment analysis model which can take user reviews and classify as a positive or negative review based on the feature set.

- Recommendation modeling for user

2. Dataset Used

The main dataset which we have used is fetched using Yelp Fusion API.
The Yelp dataset is a subset of our businesses, reviews, and user data for use inpersonal, educational, and academic purposes available as JSON data points.

A subset of this dataset is also available for reference:
https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset

We have used two specific streams of data,Data
about Businesses:

address	attributes	business_id	categories	city	hours	is_open	latitu
935 Race St	{null, null, u'no...	MTSW4McQd7CbVtyjq...	Restaurants, Food...	Philadelphia	{7:0-21:0, 7:0-20...	1	39.9555
8025 Mackenzie Rd	{null, null, u'fu...	k0hlBqXX-Bt0vf1op...	Pubs, Restaurants...	Affton	null	0	38.5651
2312 Dickerson Pike	{null, null, u'no...	bBDDEgkFA1Otx9Lfe...	Ice Cream & Froze...	Nashville	{6:0-16:0, 0:0-0:...	1	36.2081
	{null, null, 'non...	eEOYSgkmpB90uNA7l...	Vietnamese, Food,...	Tampa Bay	{11:0-14:0, 11:0-...	1	27.9552
8901 US 31 S	{null, null, 'non...	il_Ro8jwPlHresjw9...	American (Traditi...	Indianapolis	{6:0-22:0, 6:0-22...	1	39.6371332
2575 E Bay Dr	{null, null, u'no...	0bPLkL0QhhPO5kt1_...	Food, Delis, Ital...	Largo	{10:0-20:0, 10:0-...	0	27.91611
205 Race St	{null, null, 'ful...	MUTTqe8uqyMdBl186...	Sushi Bars, Resta...	Philadelphia	{13:30-23:0, null...	1	39.953
1224 South St	{null, null, u'no...	ROeacJQwBeh05Rqg7...	Korean, Restaurants	Philadelphia	{11:30-20:30, 11:...	1	39.943
6625 E 82nd St	{null, null, null...	kfNv-JZpuN6TVNSO6...	Steakhouses, Asia...	Indianapolis	{11:0-21:0, 11:0-...	1	39.90432031
5505 S Virginia St	{null, null, 'ful...	9OG5YkX1g2GReZM0A...	Restaurants, Italian	Reno	{11:0-21:0, 11:0-...	1	39.47611
215 1st Ave S	{null, null, u'fu...	tMkwHmWFUEXrC9Zdu...	Restaurants, Japa...	Nashville	{16:0-23:0, null,...	0	36.1598
767 S 9th St	{null, null, u'fu...	QdN72BWoyFypdGJhh...	Cocktail Bars, Ba...	Philadelphia	{12:0-2:0, 16:0-0...	0	39.93982457
4105 Main St	{null, null, u'no...	Mjboz24M9NlBeiOJK...	Pizza, Restaurant...	Philadelphia	{17:0-0:30, null,...	0	40.0224
10 Rittenhouse Pl	{null, null, u'no...	kV_Q1oqis8Qli8dUo...	Pizza, Restaurants	Ardmore	{11:0-1:0, 11:0-0...	1	40.0067
901 N Delaware Ave	{null, null, null...	aPNXGTDkf-4bjhyMB...	Eatertainment, Ar...	Philadelphia	{16:0-19:0, 0:0-0...	1	39.9625
116 N Pottstown Pike	{null, null, u'fu...	2xVsWBNFwZOxIOdd9...	Restaurants, Burgers	Exton	null	0	40.029
312 Piasa St	{null, null, u'fu...	ljxNT9p0y7YMPx0fc...	Restaurants, Spec...	Alton	{16:0-22:0, 0:0-0...	1	38.896
1625 W Valencia R...	{null, null, 'bee...	wghnIlMb_i5U46HMB...	Restaurants, Chinese	Tucson	{11:0-21:0, 11:0-...	0	32.1323
2031 Broadway	{null, null, u'be...	lk9IwjZXqUMqqOhM7...	Coffee & Tea, Res...	Nashville	{7:0-17:0, 7:0-17...	0	36.1483
10440 N Dale Mabr...	{null, null, u'fu...	uI9XODGY_2_ieTE6x...	Restaurants, Amer...	Tampa	{11:30-22:0, 11:3...	0	28.04620281

only showing top 20 rows

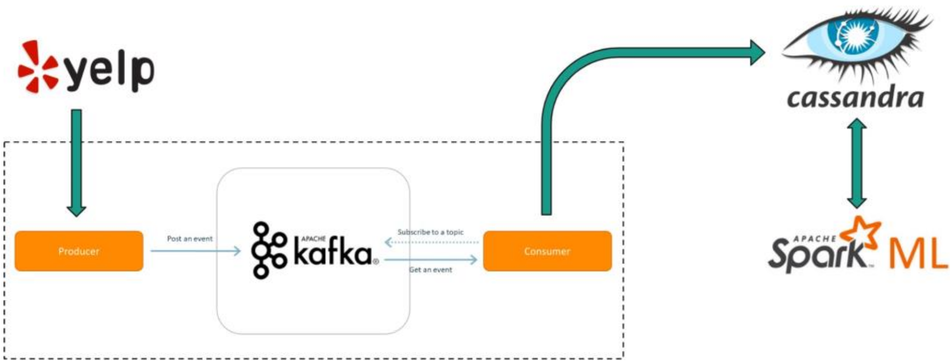
Data about Reviews:

```
root
 |-- business_id: string (nullable = true)
 |-- cool: long (nullable = true)
 |-- date: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- review_id: string (nullable = true)
 |-- stars: double (nullable = true)
 |-- text: string (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
```

business_id	cool	date	funny	review_id	stars	text	useful	user_id
aQgBYZMH7yrkQVXDP...	0	2020-12-01 13:45:24	0	MgT7fYUteUYahmuJ6...	5.0	As a local New Or...	1	kvHFjFFU0bxwDPsvK...
MfWGs8TvGI0vSvGsz...	2	2020-07-29 21:10:01	0	joNamNgy54u6O_KVi...	5.0	This has been my ...	3	dKXg01uJ_o68KxhiK...
qesrSLuacbxAcLz0P...	0	2019-04-21 02:39:56	0	RgocR6DgEFF_IoafU...	5.0	One of the best p...	0	XPoD-yO1K8RjB_KAp...
XHf-UmAKwYGKH35tY...	0	2021-08-22 05:25:29	0	BYtC6dp5F7dkbuQhp...	1.0	Very disappointed...	0	yKfKGUSmtzA27LGe4...
VPEZwyog5oF82gzd9...	1	2014-02-20 21:12:11	2	WKw076mG1Z67B90mW...	1.0	The atmosphere is...	3	A0ZCDfL1l-CQrJzsK...
BNMnyrNZNQNxQ7Qz1...	0	2021-12-03 17:52:46	0	heOB911doAAMEnOY_...	5.0	The staff was ver...	0	-xw55MHWGjWq_7yAK...
mhj9KWNqKnEAb55-9...	0	2016-02-20 03:07:07	0	EbWiIl9yUitzgqP1j...	5.0	This place is ama...	0	xNHpxKhIh1DSI-V2U...
ILUd8DXxX21FOgUmE...	1	2014-12-28 06:46:46	0	wLY6nNdyXdxnTpVbl...	4.0	can't wait to get...	0	bfHHPfcT2wyJ5tkVG...
GXqzvoBm9iEZ_R6Ub...	0	2014-12-07 21:14:12	0	iU6I1iLKsGTPZZQ3X...	4.0	I recommend a ste...	0	OgR_OXLx2ZlPoBdZ7...

We can see that using about two sets of data, we can present a complete picture of howreviews are provided by different users.

3. Architecture



The above is the architecture that we have used for our project. As can be seen in the image, we have taken data from the Yelp Database using Yelp Fusion API.

We use kafka as a messaging queue since we are handling huge amounts of data and we can not load all the data in one go. In Kafka architecture, we use Producer to call the Yelp Api which then fetches a batch of records and pushes it to the Kafka queue. From this queue, data is being picked up by the Consumer and loaded into Cassandra Database.

We have used Cassandra for two specific reasons. First, we are getting the data from the Yelp Api in json format, and thus we wanted to store them in a NoSql database. Secondly, Cassandra provides a simple CQL (query language) like SQL to query data

sorted in the database. This makes it a lot easier to query data as compared to

databases like MongoDb where for fetching data, you will need to learn a separate language.The Kafka messaging queue is also helpful in order to refresh the database over a certain period of time.

After we have stored data in the database, we can now use it to get useful information out of it. For this we are using Spark along with its ML libraries. Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming clusters with implicit data parallelism and fault tolerance.

4. Implementation

4.1 Preprocessing

Categories :

Looking at different categories by selecting unique from categories column.

```
us_restaurants.select('categories').show(truncate=False)
```

```
+------------------------------------------------------------------------------------------------------------------------------------------------------+
|categories                                                                                                                                            |
+------------------------------------------------------------------------------------------------------------------------------------------------------+
|Restaurants, Food, Bubble Tea, Coffee & Tea, Bakeries                                                                                                 |
|Pubs, Restaurants, Italian, Bars, American (Traditional), Nightlife, Greek                                                                            |
|Ice Cream & Frozen Yogurt, Fast Food, Burgers, Restaurants, Food                                                                                      |
|Vietnamese, Food, Restaurants, Food Trucks                                                                                                            |
|American (Traditional), Restaurants, Diners, Breakfast & Brunch                                                                                       |
|Food, Delis, Italian, Bakeries, Restaurants                                                                                                           |
|Sushi Bars, Restaurants, Japanese                                                                                                                     |
|Korean, Restaurants                                                                                                                                   |
|Steakhouses, Asian Fusion, Restaurants                                                                                                                |
|Restaurants, Italian                                                                                                                                  |
|Restaurants, Japanese, Seafood                                                                                                                        |
|Cocktail Bars, Bars, Italian, Nightlife, Restaurants                                                                                                  |
|Pizza, Restaurants, Salad, Soup                                                                                                                       |
|Pizza, Restaurants                                                                                                                                    |
|Eatertainment, Arts & Entertainment, Brewpubs, American (Traditional), Bakeries, Breweries, Food, Restaurants                                         |
|Restaurants, Burgers                                                                                                                                  |
|Restaurants, Specialty Food, Steakhouses, Food, Italian, Pizza, Pasta Shops                                                                           |
|Restaurants, Chinese                                                                                                                                  |
|Coffee & Tea, Restaurants, Wine Bars, Bars, Nightlife, American (Traditional), Event Planning & Services, Food, Caterers, Breakfast & Brunch, Cafes, Diners|
|Restaurants, American (New), Italian                                                                                                                  |
+------------------------------------------------------------------------------------------------------------------------------------------------------+
only showing top 20 rows
```

Column:

```
'business_id_duplicated', 'cool', 'date', 'funny', 'review_id', 'stars',
'text', 'useful', 'user_id', 'address', 'attributes', 'business_id',
'categories', 'city', 'hours', 'is_open', 'latitude', 'longitude',
'name', 'postal_code', 'review_count', 'avg_stars', 'state', 'category',
'review_length', 'labels'],
```

These are the columns which we will explore using functions available in Spark.

Null Check and Imputation:

```
#Filtering and dropping all rows which do not belong to any 1 of the selected category
filtered_restaurants=filtered_restaurants.na.drop(subset=["category"])
filtered_restaurants.show()
filtered_restaurants.count()
```

```
+--------------------+--------------------+--------------------+--------------------+-------------+-----------
|             address|          attributes|         business_id|          categories|         city|
+--------------------+--------------------+--------------------+--------------------+-------------+-----------
|    8025 Mackenzie Rd|{null, null, u'fu...|k0hlBqXX-Bt0vf1op...|Pubs, Restaurants...|       Affton|
|                    |{null, null, 'non...|eEOYSgkmpB90uNA7l...|Vietnamese, Food,...|    Tampa Bay|{11:0-14:0,
|         8901 US 31 S|{null, null, 'non...|il_Ro8jwPlHresjw9...|American (Traditi...| Indianapolis|{6:0-22:0,
|          2575 E Bay Dr|{null, null, u'no...|0bPLkL0QhhPO5kt1_...|Food, Delis, Ital...|        Largo|{10:0-20:0,
|         205 Race St|{null, null, 'ful...|MUTTqe8uqyMdBl186...|Sushi Bars, Resta...| Philadelphia|{13:30-23:0
|       1224 South St|{null, null, u'no...|ROeacJQwBeh05Rqg7...| Korean, Restaurants| Philadelphia|{11:30-20:3
|   5505 S Virginia St|{null, null, 'ful...|9OG5YkX1g2GReZM0A...|Restaurants, Italian|         Reno|{11:0-21:0,
|       215 1st Ave S|{null, null, u'fu...|tMkwHmWFUEXrC9Zdu...|Restaurants, Japa...|    Nashville|{16:0-23:0,
|         767 S 9th St|{null, null, u'fu...|QdN72BWoyFypdGJhh...|Cocktail Bars, Ba...| Philadelphia|{12:0-2:0,
|   901 N Delaware Ave|{null, null, null...|aPNXGTDkf-4bjhyMB...|Eatertainment, Ar...| Philadelphia|{16:0-19:0,
|        312 Piasa St|{null, null, u'fu...|ljxNT9p0y7YMPx0fc...|Restaurants, Spec...|        Alton|{16:0-22:0,
|1625 W Valencia R...|{null, null, 'bee...|wghnIlMb_i5U46HMB...|Restaurants, Chinese|       Tucson|{11:0-21:0,
|       2031 Broadway|{null, null, u'be...|lk9IwjZXqUMqqOhM7...|Coffee & Tea, Res...|    Nashville|{7:0-17:0,
|10440 N Dale Mabr...|{null, null, u'fu...|uI9XODGY_2_ieTE6x...|Restaurants, Amer...|        Tampa|{11:30-22:0
|      6880 E 82nd St|{null, null, 'ful...|seKihQKpGGnCeLuEL...|Sports Bars, Amer...| Indianapolis|{11:0-19:0,
|        117 E Gay St|{null, null, u'fu...|qfWJmJ0g96eM_fWma...|Seafood, Restaura...| West Chester|
|820 N Black Horse...|{null, null, u'fu...|8rb-3VYXE37IZix4y...|American (Traditi...| Williamstown|
|3173 Cypress Ridg...|{null, null, u'fu...|pJfh3Ct8iL58NZa8t...|Burgers, Sports B...|Wesley Chapel|{11:30-23:3
|       1931 Park Ave|{null, null, u'fu...|1MeIwdbTnZOBFCKOr...|American (New), R...|  Saint Louis|{16:0-22:0,
|3322 Old Capitol Trl|{null, null, u'no...|Dy91wdWkwtI_qgjAI...|Mexican, Restaurants|   Wilmington|
+--------------------+--------------------+--------------------+--------------------+-------------+-----------
only showing top 20 rows
```

Duplicates Check:

```
#checking if there are duplicates
filtered_restaurants.groupby(['business_id']).count().where('count > 1').sort('count', ascending=False).show()
```

```
+-----------+-----+
|business_id|count|
+-----------+-----+
+-----------+-----+
```

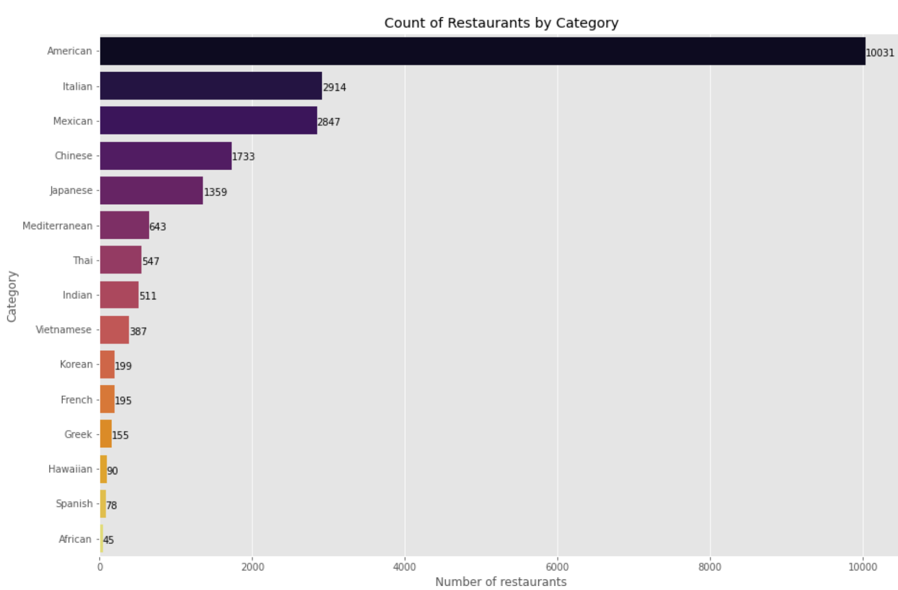
4.2 Merging the 2 data sets:

We performed an inner join on the 2 data sets based on Business ID as the primary and foreign key.

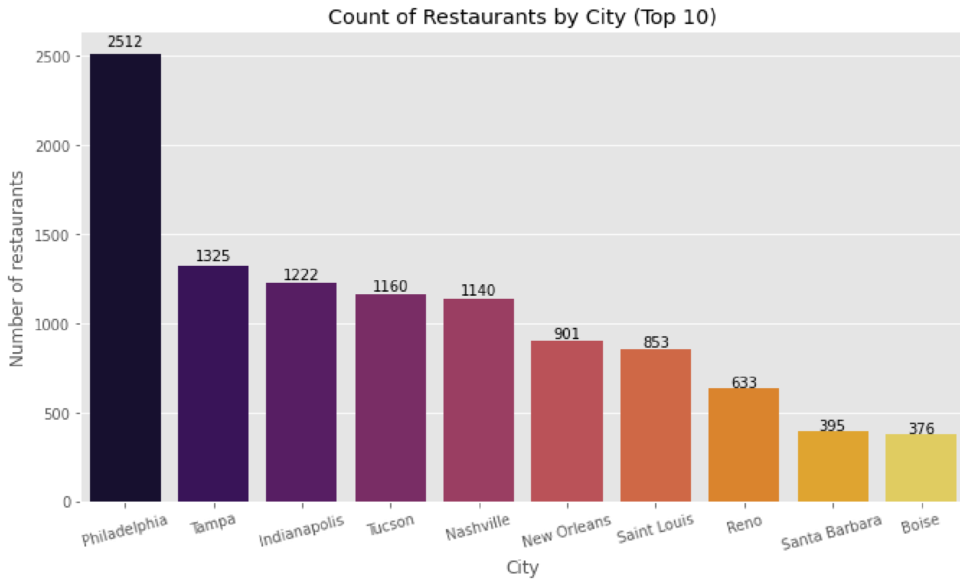
```
# Joining reviews and filtered_restaurants to create a new merged dataframe

# Renaming stars column in filtered_restaurants to avg_stars before join
filtered_restaurants = filtered_restaurants.withColumnRenamed("stars","avg_stars")

restaurant_reviews = reviews.join(filtered_restaurants,reviews.business_id == filtered_restaurants.business_id,"inner")
restaurant_reviews.printSchema()
```

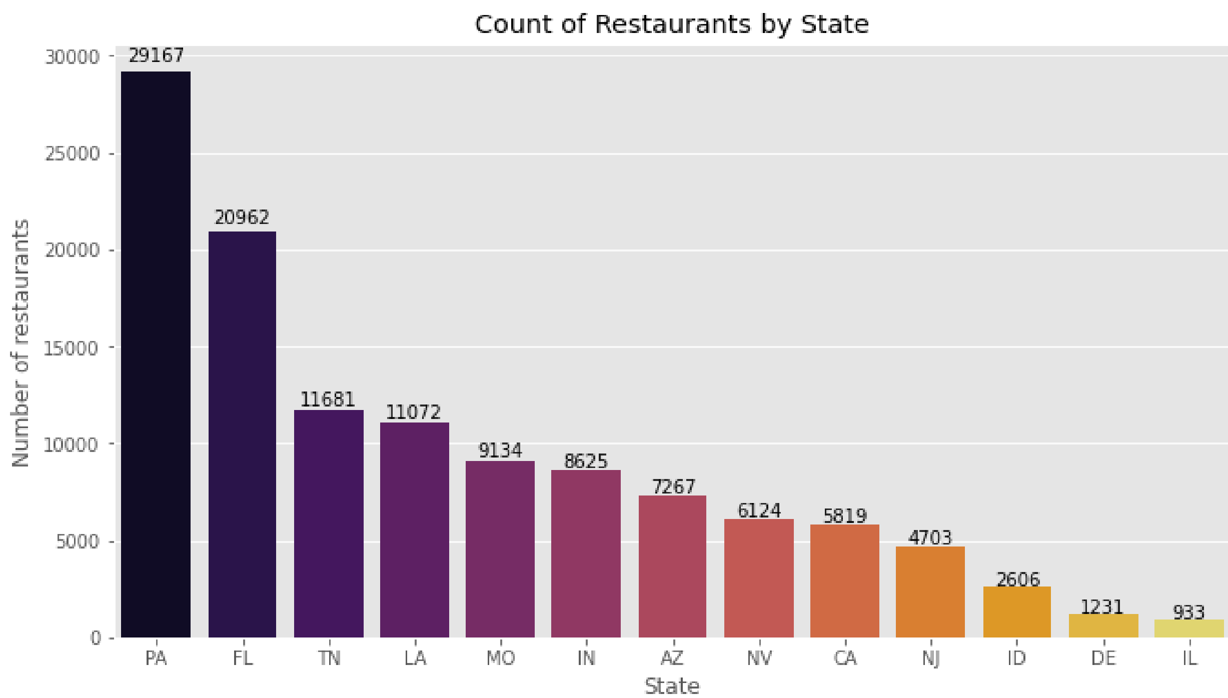
```
root
 |-- business_id: string (nullable = true)
 |-- cool: long (nullable = true)
 |-- date: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- review_id: string (nullable = true)
 |-- stars: double (nullable = true)
 |-- text: string (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
 |-- address: string (nullable = true)
 |-- attributes: struct (nullable = true)
 |    |-- AcceptsInsurance: string (nullable = true)
```

Counting Categories:



Distribution of Count of Restaurants per City and State:

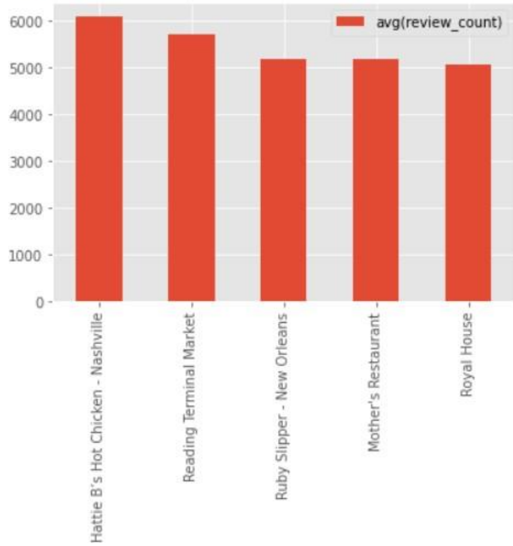




Average review count for restaurants with positive ratings:

```
restaurant_reviews.filter(restaurant_reviews.labels == "positive").groupby("name").mean().select("name","avg(review_c
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2ca35cb310>
```

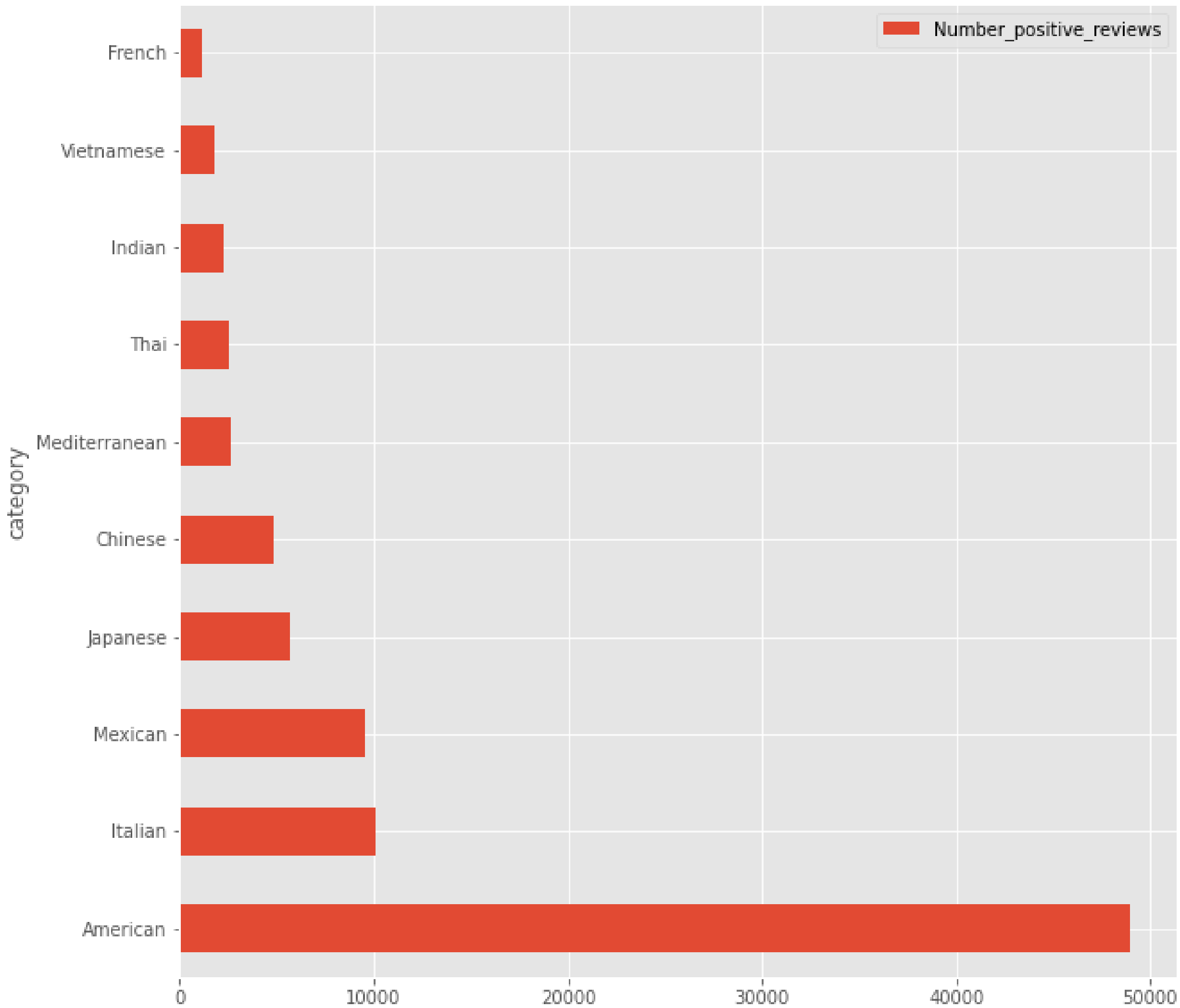


Average review count for restaurants with positive ratings per state:

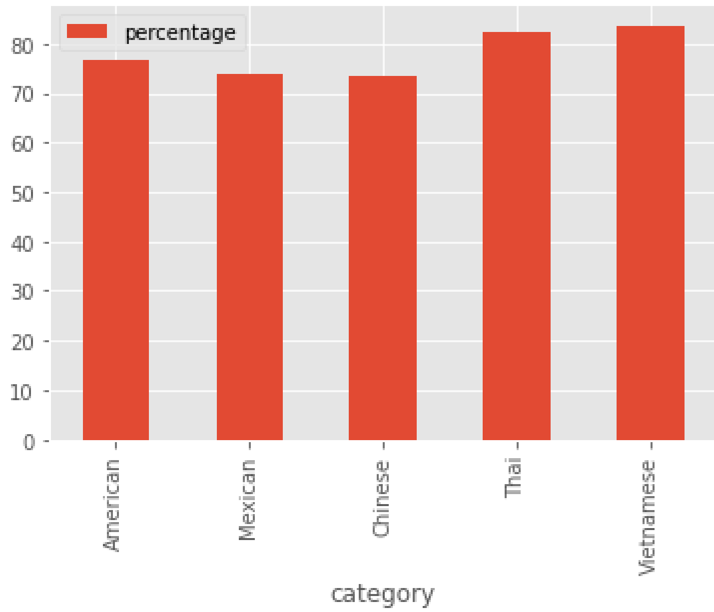
```
restaurant_reviews.filter(restaurant_reviews.labels == "positive").
```

	name	state	avg(review_count)
0	Hattie B's Hot Chicken - Nashville	TN	6093.0
1	Reading Terminal Market	PA	5721.0
2	Ruby Slipper - New Orleans	LA	5193.0
3	Mother's Restaurant	LA	5185.0
4	Royal House	LA	5070.0
...	...	...	...

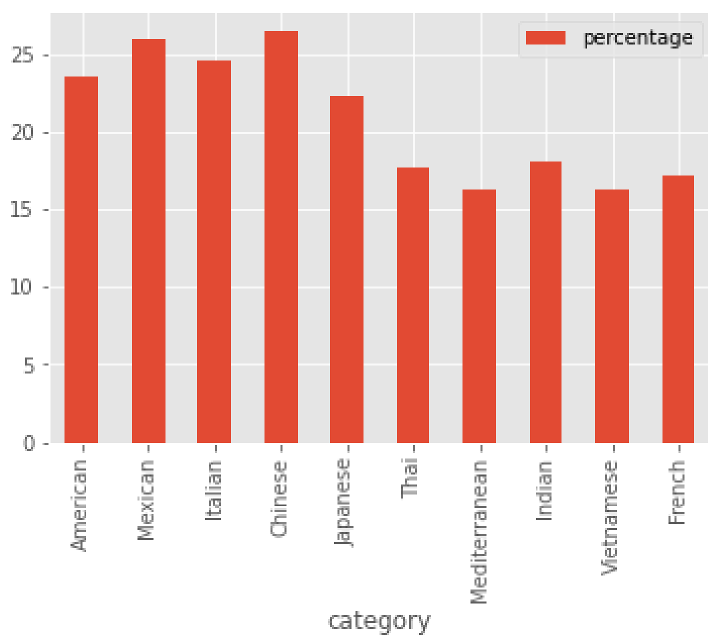
Maximum number of positive reviews per category:



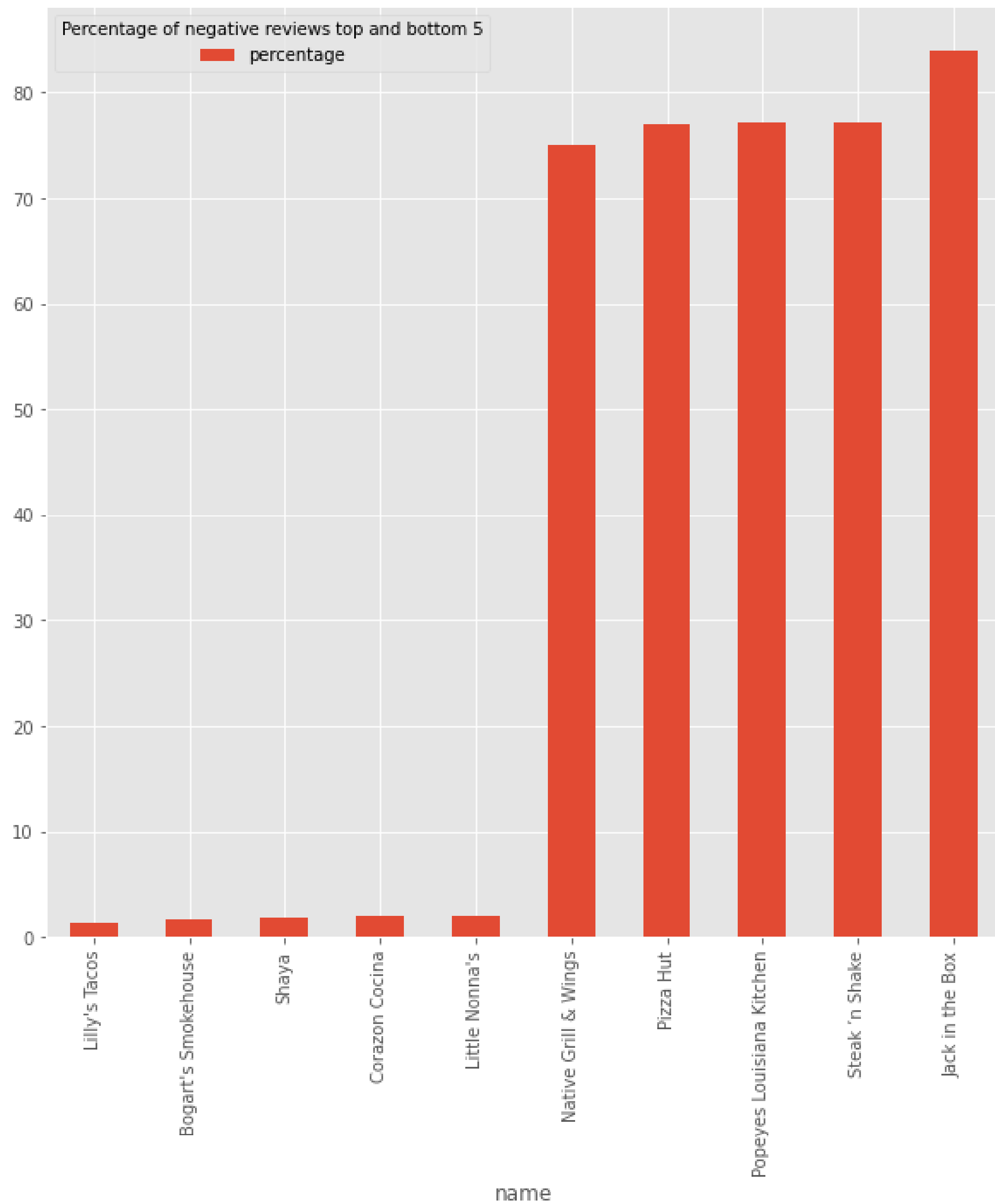
Percentage of positive reviews per category:



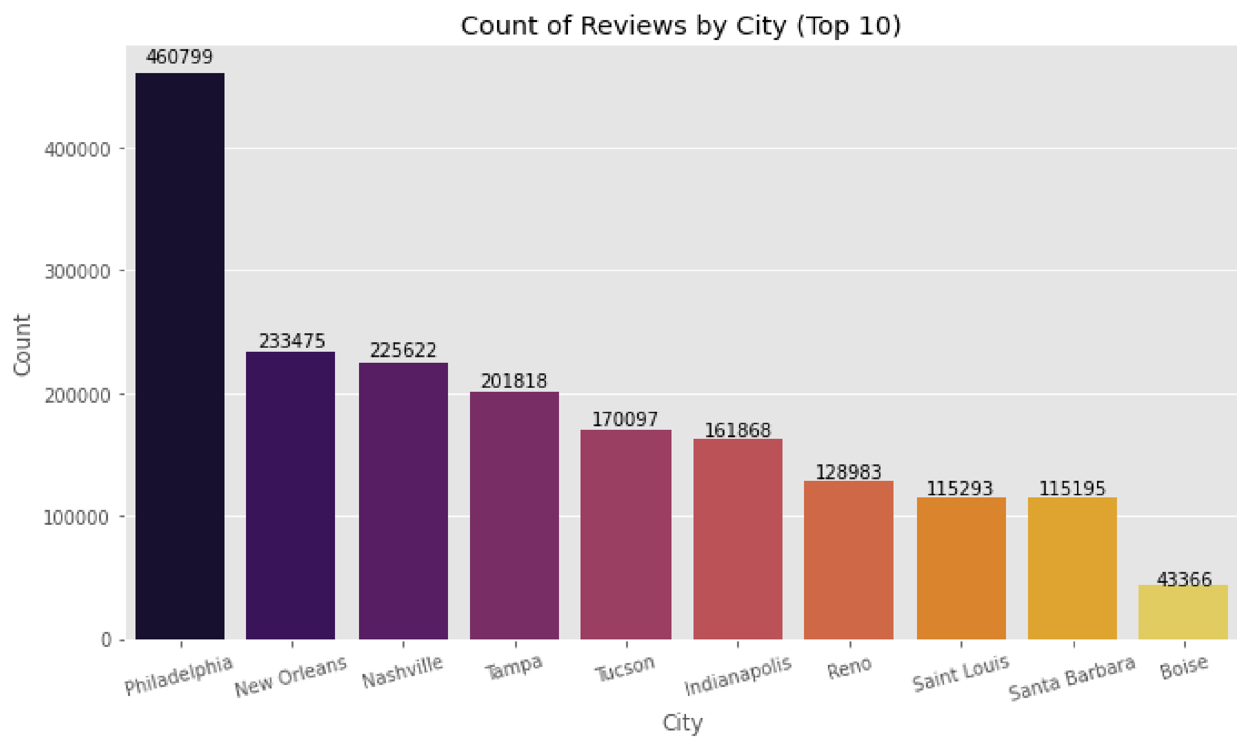
Percentage of negative reviews per category:



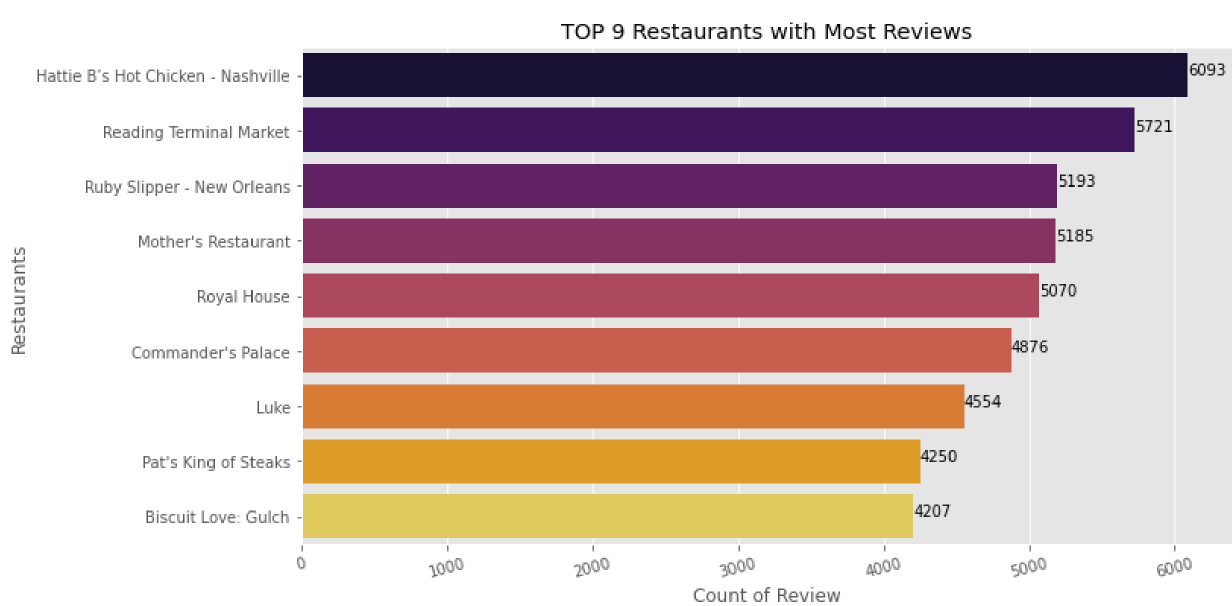
Percentage of negative reviews top and bottom 5:



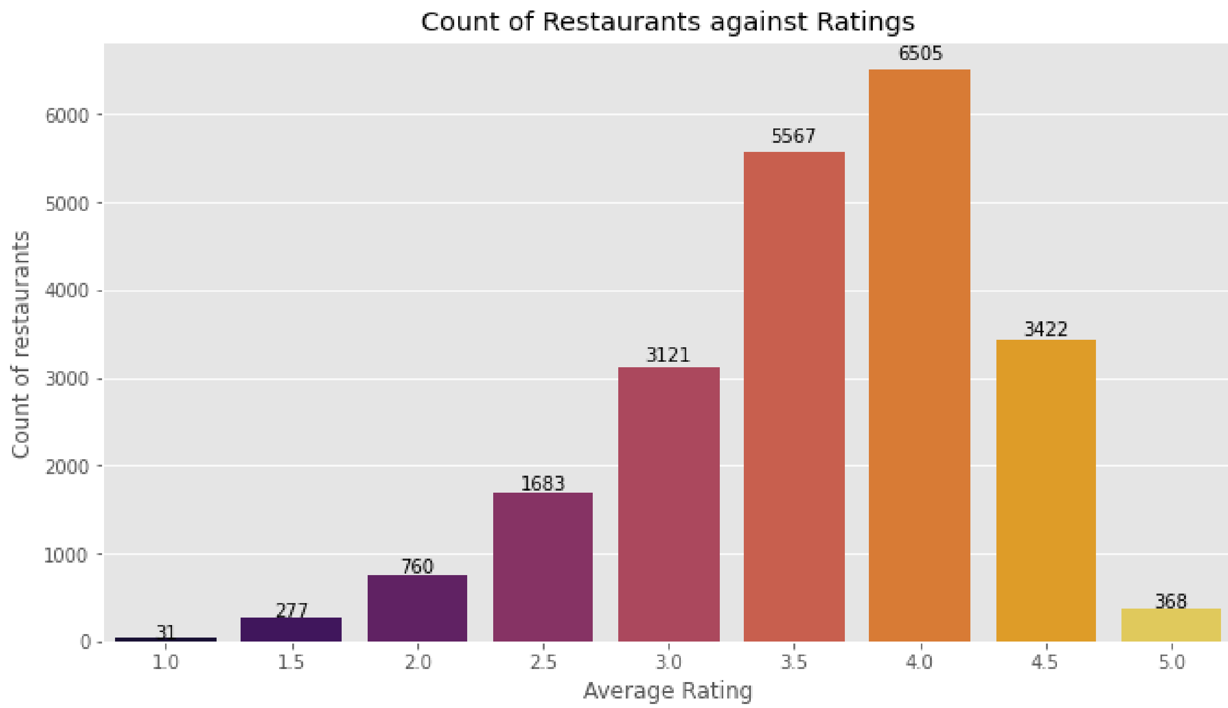
Top 10 cities with most Reviews:

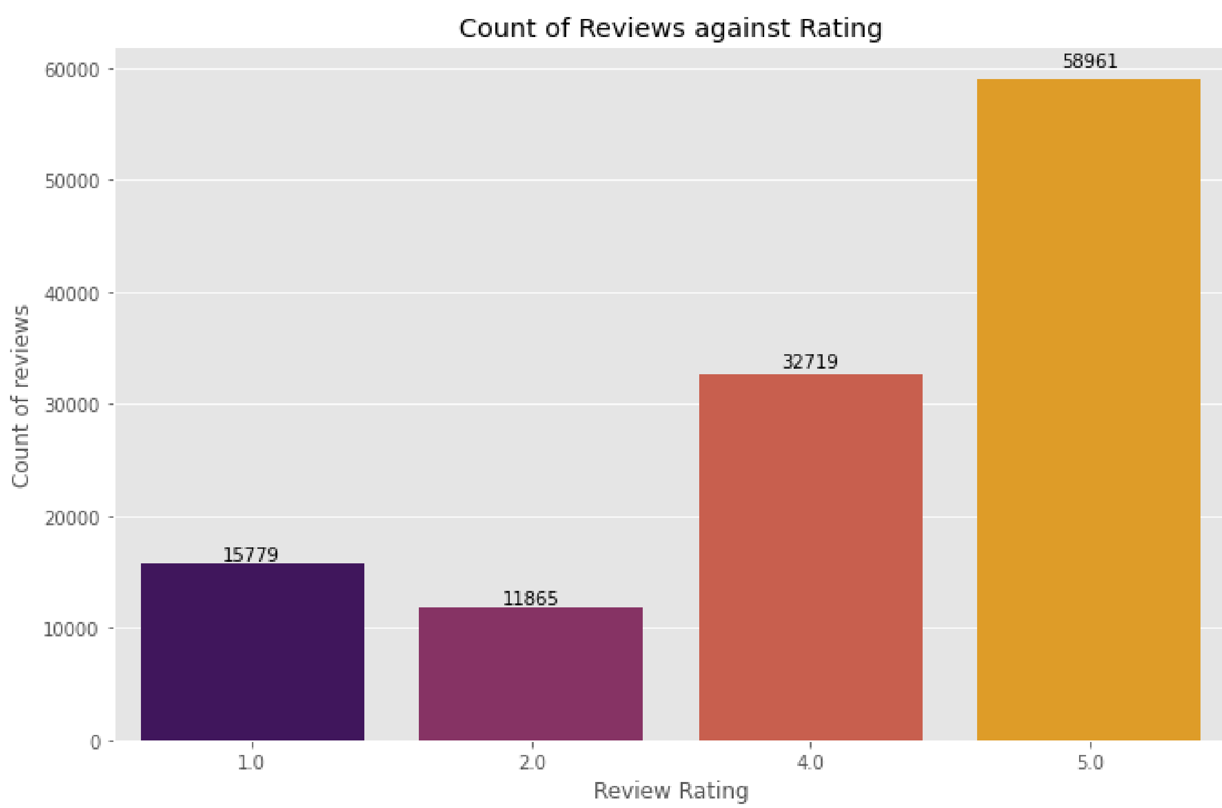


Top Restaurants with most reviews:



Distribution of ratings against count of restaurants:





4.2 NLP

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

The section below shows the distribution of labels across the data set.

```
restaurant_reviews.groupby('labels').count().show()
```

```
+--------+-----+
|  labels|count|
+--------+-----+
|positive|91680|
|negative|27644|
+--------+-----+
```

```
+--------------------+--------+
|                text|  labels|
+--------------------+--------+
|as a local new or...|positive|
|this has been my ...|positive|
|the atmosphere is...|negative|
|cant wait to get ...|positive|
|i am a vegetarian...|negative|
|i came here for m...|positive|
|quoted 10 minutes...|negative|
|lots of breweries...|positive|
|so me and my wife...|negative|
|best mexican food...|positive|
|i have been wanti...|positive|
|new owner have ta...|positive|
|quickly becoming ...|positive|
|i love this place...|positive|
|this is one of my...|positive|
|food was ok  noth...|negative|
|hanging out waiti...|positive|
|delicious my husb...|positive|
|delicious food my...|positive|
|we stopped in on ...|positive|
+--------------------+--------+
only showing top 20 rows
```

4.3 NLP : Preprocessing

- **Converting text to lowercase:**

  To reduce ambiguity and to be in the same format, All the reviews are converted to lower case letters. This is done by using the user defined functions available in spark.

```
def lower(text):
    return text.lower()

lower_udf =udf(lower,StringType())
```

- **Removing non Ascii characters, punctuations, stopwords**

  These are essential to be removed for the model to vectorize the words and the root words properly.

- **Fixing abbreviations**

  This is another important set as we humans tend to choose the shortest path in all aspects of life. It is essential for the model to understand the abbreviations used and the root words behind them.

- **Stemming**

  Stemming is the process of finding the root word from the corpus of words given.

- **Lemmatization**

  Lemmatization is finding the form of the related word in the dictionary

- Vectorizer (Count and TF-id)

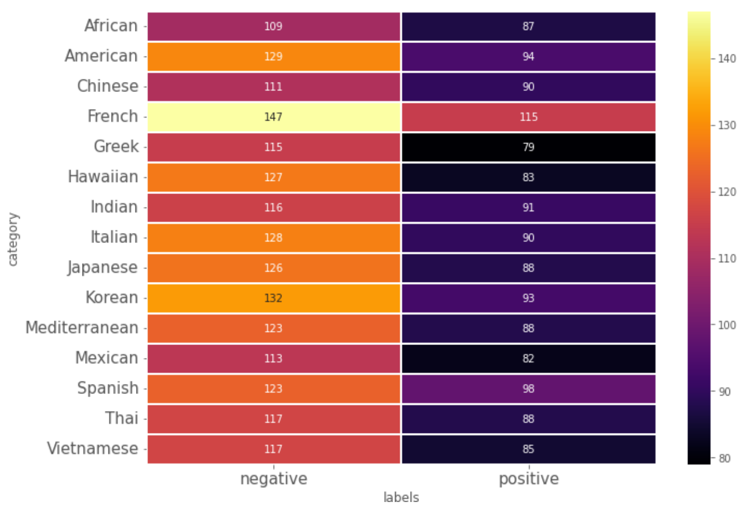
With Tfidftransformer we compute word counts using CountVectorizer and then compute the Inverse Document Frequency (IDF) values and only then compute the Tf-idf scores.

```
#Tokenizing the document
tokenizer = Tokenizer(inputCol="tag_and_remove_pos", outputCol="words")
wordsDataFrame = tokenizer.transform(df_pos_tagging)
for words_label in wordsDataFrame.select("words", "Target").take(3):
    print(words_label)

df_text = df.withColumn("text_lower",lower_udf(df["Text"])).select('text_lower','Target')
```

```
Row(words=['', 'local', 'new', 'orleanian', 'area', 'many', 'vietnamese', 'restaurants', 'choose', 'td', 'pho', 'house', 'moderneclectic', 'restaurant', 'combining', 'a
Row(words=['', 'go', 'banh', 'mi', 'spot', 'year', 'ride', 'harder', 'place', 'entering', 'youll', 'notice', 'woman', 'shop', 'keeps', 'kid', 'entertained', 'impressive
Row(words=['', 'pleasant', 'well', 'designed', 'hostess', 'friendly', 'meal', 'acceptable', 'soup', 'watery', 'couldnt', 'eat', 'sandwich', 'waitress', 'asked', 'food',
```

Feature Space after performing the preprocessing steps:

```
+--------------------+--------+------+--------------------+--------------------+--------------------+--------------------+
|                text|  labels|Target|          lower_text|       text_non_asci|        fixed_abbrev|    removed_features|
+--------------------+--------+------+--------------------+--------------------+--------------------+--------------------+
|as a local new or...|positive|     1|as a local new or...|as a local new or...|as a local new or...|as a local new or...|
|this has been my ...|positive|     1|this has been my ...|this has been my ...|this has been my ...|this has been my ...|
|the atmosphere is...|negative|     0|the atmosphere is...|the atmosphere is...|the atmosphere is...|the atmosphere is...|
|cant wait to get ...|positive|     1|cant wait to get ...|can not wait to g...|can not wait to g...|can not wait to g...|
|i am a vegetarian...|negative|     0|i am a vegetarian...|i am a vegetarian...|i am a vegetarian...|i am a vegetarian...|
+--------------------+--------+------+--------------------+--------------------+--------------------+--------------------+
only showing top 5 rows
```



4.4 NLP - Modeling:

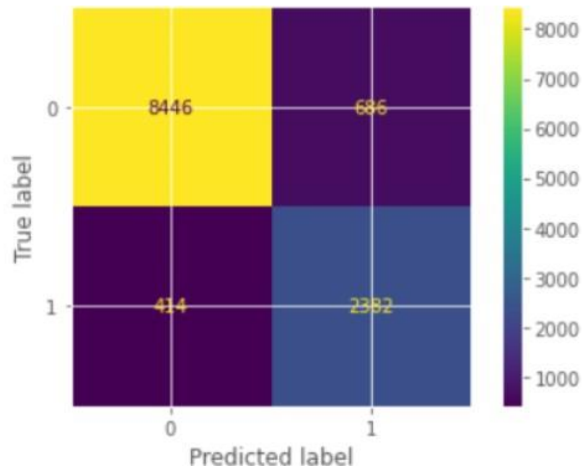
We have used Logistic Regression to model based on the text features. Logistic Regression is used to classify elements of a set into two groups (binary classification) by calculating the probability of each element of the set. It uses the sigmoid function to calculate the probabilities of each class between 0 and 1. If the probability of a class is greater than .5, it will be assigned class 1 (positive) else 0 (negative).

```
lr = LogisticRegression(maxIter=10, regParam=0.01)
pipeline = Pipeline(stages=[tokenizer, hashingTF, idf, lr])
# Training the model
model = pipeline.fit(training)
#Predicing Output
prediction = model.transform(test)
prediction.select("label", "prediction").show(10,False)
```

```
+-----+----------+
|label|prediction|
+-----+----------+
|0.0  |0.0       |
|1.0  |1.0       |
|1.0  |1.0       |
|1.0  |1.0       |
|1.0  |1.0       |
|0.0  |0.0       |
|1.0  |1.0       |
|1.0  |1.0       |
|1.0  |1.0       |
|1.0  |1.0       |
+-----+----------+
only showing top 10 rows
```

Results:

```
[[8446  686]
 [ 414 2382]]
0.9077800134138162
              precision    recall  f1-score   support

         0.0       0.95      0.92      0.94      9132
         1.0       0.78      0.85      0.81      2796

    accuracy                           0.91     11928
   macro avg       0.86      0.89      0.88     11928
weighted avg       0.91      0.91      0.91     11928
```



We have achieved a pretty good accuracy and a decent precision and recall for both positive and negative classes. Tree based models with boosting and bagging techniques can be further used to improve this metric for both the classes.

4.5 Recommendation Model

Since we have information user's reviews and which restaurants they visited, we can build a recommendation model which can suggest restaurants to customers where they can visit next time.

Specifically, we have used two separate methods to do this, content based filtering and collaborative filtering.

4.5.1 Content Based Filtering using K-Nearest Neighbors:

It is based on the features of the restaurants rather than the user features. The idea is if the user likes a restaurant, then he/she will like the other similar restaurants. KNN model has been used for recommendation in this approach. It takes similarities between two restaurants based on their features into consideration for recommendation. Euclidean dist was taken as selection criteria. Preprocessing: Following features were used: ['index', 'business_id', 'name', 'address', 'categories', 'attributes', 'stars', 'BusinessParking', 'Ambience', 'GoodForMeal', 'Dietary', 'Music']. For categorical data such as 'GoodForMeal', 'attributes' etc. we created one hot encoding.

Results for 'Adelita Taqueria & Restaurant' using Content Based Filtering:

```
Restaurant indices for restaurants that are
 similar to 'Adelita Taqueria & Restaurant'
```

	distance	index	name	stars
0	4.000000	2329	Los Taquitos de Puebla	4
1	4.123106	2312	Yummy Sushi	4
2	4.242641	888	The Flavor Spot	4
3	4.358899	2573	Maker artisan pizza	4
4	4.358899	1488	Mood Indian Restaurant	4

4.5.2. Collaborative Filtering using SVD

Collaborative filtering uses similar*ities* between users and restaurants simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend restaurants to user A based on the interests of a similar user B. We used SVD to generate recommendations based on the user's taste and likings. Pearson correlation coefficient was used as the selection criteria. In the preprocessing we created a user rating matrix where rows are user_ids and columns are the ratings given to a particular restaurant. We apply SVD to this matrix due to its sparsity. We created the correlation matrix from the above matrix. For any restaurant, we create a list of restaurants from the correlation matrix which have high correlation value with the given restaurant.

Results for Reading Terminal Market using collaborative filtering:

```
Restaurants similar to
 Reading Terminal Market are:
```

	corr_val	restaurant_name
0	0.999662	3J's Food Market
1	0.999722	@Ramen
2	0.921199	AmeriThai
3	0.999876	Bistro La Baia
4	0.963554	Café Soho

5. Conclusion

1. General analysis of reviews was performed wherein different aspects of restaurants and user statistics were analyzed. The analysis of user reviews data, restaurants touser analysis and text review analysis gives an account to the user to gauge the credibility, popularity, and reviews of restaurants.

2. The NLP classifier based on Logistic Regression is performing well with a decent precision and recall score for each class (negative and positive).

3. We are able to generate recommendations for a user using both the techniques. We also observe that the ratings provided also match.

6. Future Scope

1. Remove or set low preference to the reviews that are fake, using outlier detection.

2. Integrate our project with google maps API so that it can automatically generate recommendations based on the user's live location.

3. Use of deep learning models like LSTMs for modeling the text sentiment as it would be more precise.

4. Lastly, all the analytics and data can be viewed through the prism of a fancy UI where the platform can be easily used by anyone without any technical background

7. References

1. https://spark.apache.org/docs/latest/api/python/reference/index.html

2. https://www.yelp.com/developers/documentation/v3/get_started

3. https://matplotlib.org/

4. https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/

5. https://kavita-ganesan.com/tfidftransformer-tfidfvectorizer-usage-differences/#.YoQs7qjM_JPY

6. https://kafka-python.readthedocs.io/en/v0.9.5/usage.html

7. https://cassandra.apache.org/doc/latest/