Scalable Analytics Team 7 Final Project Report

Team Members: Annie Gaver, Shreyashi Mukhopadhyay, Leticia Davordzi, Hedieh Basiri, Mohammad Khasawneh

Introduction

Group 7 has designed a tool that can recommend restaurants to yelp users based on how they have rated other restaurants in the past. Our approach to building this tool was to utilize key skills learned in MSA 8050. In this report, we will describe how we cleaned data using Google Cloud, explored ratings content with Sentiment Analysis, conducted a Market Basket Analysis with RDDs, and completed further investigation using scalable Machine Learning pipelines.

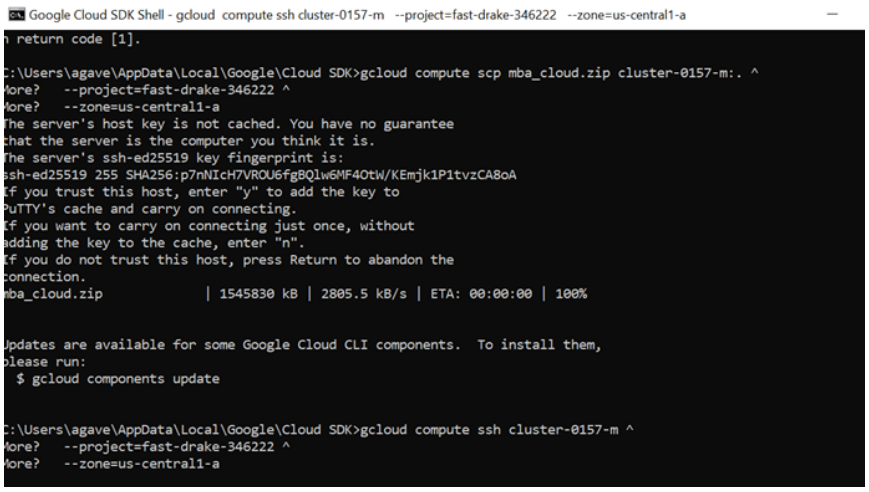
Cleaning Data on Google Cloud

Cleaning the data consisted of filtering the “yelp_review.csv” to return only Restaurant reviews from the city of Toronto. Originally, our team wanted to analyze restaurant reviews only from Atlanta, but no Atlanta data was found in our exploration. Because of this, we settled on Toronto because it had the highest count of restaurant reviews by city. Then we joined the “yelp_business.csv” on the unique business ID to return all of the relevant restaurant information for the reviews. Lastly, we dropped any columns with NA values to return a cleaned dataframe.

This dataframe in itself was able to be cleaned locally and went on to be used for our Machine Learning Pipelines. However, our Market Basket Analysis (described later in the report) required further cleaning and an output file, which we were not able to do on our local machines alone.

The additional cleaning for the Market Basket Analysis involved filtering the reviews by only 3, 4, and 5 star ratings, aggregating the highly rated restaurant names into a list by user, dropping all other columns but the list of restaurant names, and removing all special characters. Once complete, the pyspark program returned lists of highly rated restaurants by user.

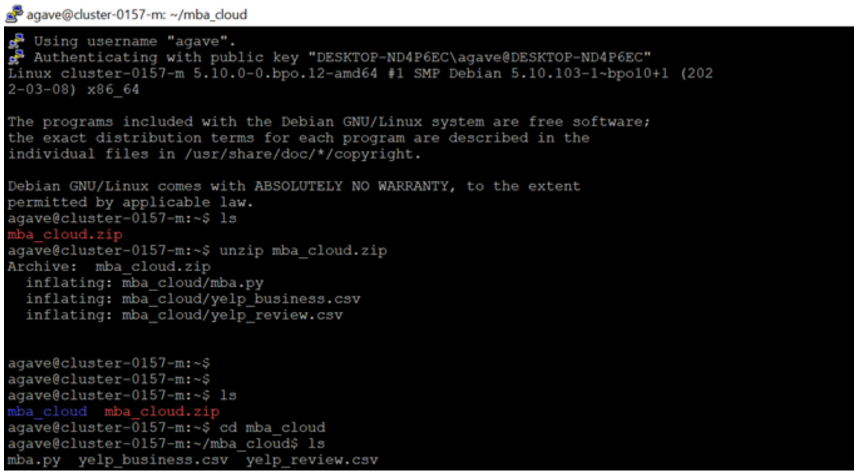
In order to run the program in the cloud, we zipped the two csv files and the python file into a single folder. Below is a screenshot of us uploading the zip file to the cluster and then connecting locally with the SSH key.



Google Cloud SDK Shell - gcloud compute ssh cluster-0157-m --project=fast-drake-346222 --zone=us-central1-a

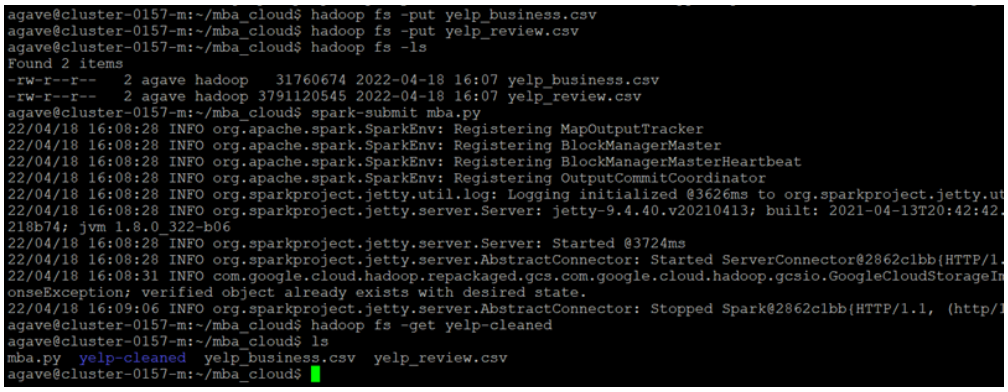
```
n return code [1].

C:\Users\agave\AppData\Local\Google\Cloud SDK>gcloud compute scp mba_cloud.zip cluster-0157-m:. ^
More?   --project=fast-drake-346222 ^
More?   --zone=us-central1-a
The server's host key is not cached. You have no guarantee
that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:p7nNIcH7VROU6fgBQlw6MF4OtW/KEmjk1P1tvzCA8oA
If you trust this host, enter "y" to add the key to
PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
mba_cloud.zip             | 1545830 kB | 2805.5 kB/s | ETA: 00:00:00 | 100%


Updates are available for some Google Cloud CLI components.  To install them,
please run:
  $ gcloud components update


C:\Users\agave\AppData\Local\Google\Cloud SDK>gcloud compute ssh cluster-0157-m ^
More?   --project=fast-drake-346222 ^
More?   --zone=us-central1-a
```

Below you will see the command line code to run the program. This involved linux commands such as unzipping the file, changing the directory, and pushing the csv files to the hadoop fileshare. Without pushing over the files, the program would not run and return the desired cleaned files. Once the program was run and cleaning was complete, we retrieved the output files from the hadoop fileshare. We then zipped the cleaned files and locally pulled the new zip file back from the cluster.



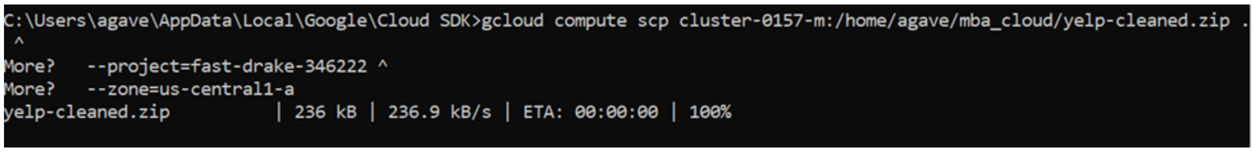
agave@cluster-0157-m: ~/mba_cloud

```
  Using username "agave".
  Authenticating with public key "DESKTOP-ND4P6EC\agave@DESKTOP-ND4P6EC"
Linux cluster-0157-m 5.10.0-0.bpo.12-amd64 #1 SMP Debian 5.10.103-1~bpo10+1 (202
2-03-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
agave@cluster-0157-m:~$ ls
mba_cloud.zip
agave@cluster-0157-m:~$ unzip mba_cloud.zip
Archive:  mba_cloud.zip
  inflating: mba_cloud/mba.py
  inflating: mba_cloud/yelp_business.csv
  inflating: mba_cloud/yelp_review.csv


agave@cluster-0157-m:~$
agave@cluster-0157-m:~$
agave@cluster-0157-m:~$ ls
mba_cloud  mba_cloud.zip
agave@cluster-0157-m:~$ cd mba_cloud
agave@cluster-0157-m:~/mba_cloud$ ls
mba.py  yelp_business.csv  yelp_review.csv
```



```
agave@cluster-0157-m:~/mba_cloud$ hadoop fs -put yelp_business.csv
agave@cluster-0157-m:~/mba_cloud$ hadoop fs -put yelp_review.csv
agave@cluster-0157-m:~/mba_cloud$ hadoop fs -ls
Found 2 items
-rw-r--r--   2 agave hadoop   31760674 2022-04-18 16:07 yelp_business.csv
-rw-r--r--   2 agave hadoop 3791120545 2022-04-18 16:07 yelp_review.csv
agave@cluster-0157-m:~/mba_cloud$ spark-submit mba.py
22/04/18 16:08:28 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/04/18 16:08:28 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/04/18 16:08:28 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/04/18 16:08:28 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
22/04/18 16:08:28 INFO org.sparkproject.jetty.util.log: Logging initialized @3626ms to org.sparkproject.jetty.ut
22/04/18 16:08:28 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.
218b74; jvm 1.8.0_322-b06
22/04/18 16:08:28 INFO org.sparkproject.jetty.server.Server: Started @3724ms
22/04/18 16:08:28 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@2862c1bb{HTTP/1.
22/04/18 16:08:31 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageIm
onseException; verified object already exists with desired state.
22/04/18 16:09:06 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@2862c1bb{HTTP/1.1, (http/1
agave@cluster-0157-m:~/mba_cloud$ hadoop fs -get yelp-cleaned
agave@cluster-0157-m:~/mba_cloud$ ls
mba.py  yelp-cleaned  yelp_business.csv  yelp_review.csv
agave@cluster-0157-m:~/mba_cloud$
```

Locally getting the cleaned file back from the cluster.



```
C:\Users\agave\AppData\Local\Google\Cloud SDK>gcloud compute scp cluster-0157-m:/home/agave/mba_cloud/yelp-cleaned.zip .
 ^
More?   --project=fast-drake-346222 ^
More?   --zone=us-central1-a
yelp-cleaned.zip         | 236 kB | 236.9 kB/s | ETA: 00:00:00 | 100%
```

Sentiment Analysis:

Model used with the TF-IDF method:

SVM with Stochastic gradient descent

In our model we applied an attempt to construct a code using the TF-IDF method to statistically show the importance of a word in a document and in a collection of documents. The primary need for this method is to find the importance of words under the reviews variable in each restaurant in Toronto within each rating level (1-5 stars), ultimately, this allows business owners to see their current performance from customer's standpoint and identify existing errors and improve products and services.

The TF-IDF method works in a way that the more frequent the word is within a document, the more important it is. On the other hand, the more that same word is being repeated among the documents, the less important it gets. The model assigns weights on each word to assist in the evaluation process.

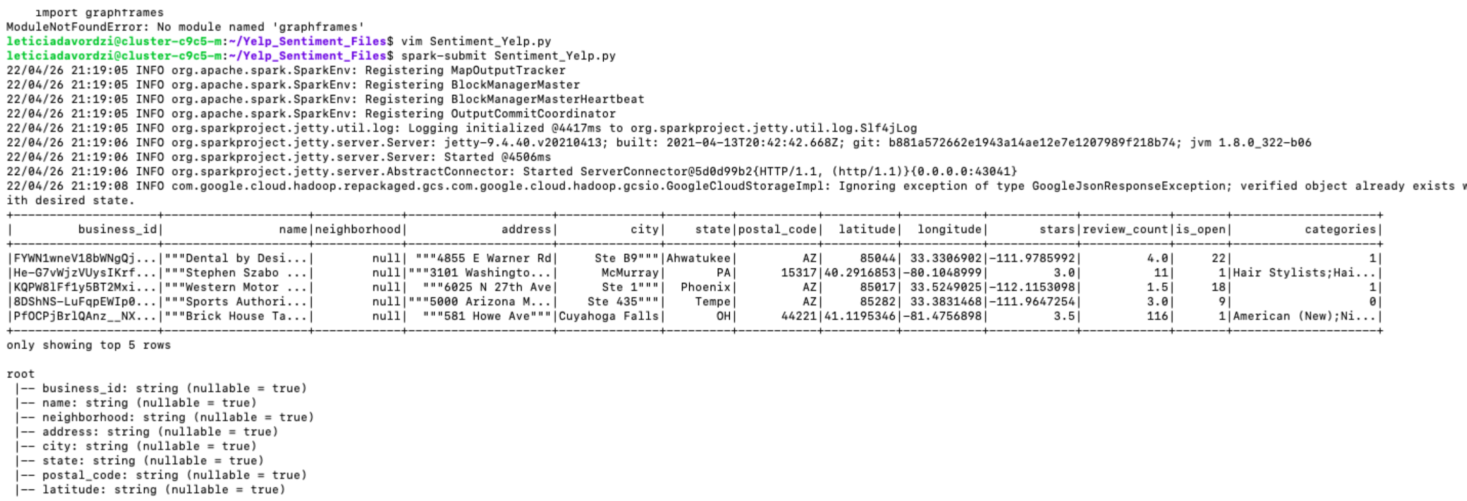
Next, we created a machine learning pipeline that consists of series of codes that first breaks natural language text into chunks using tokenizer, a term-frequency (TF) code to obtain certain weights for the words within a document, inverse-document frequency (IDF) that checks for the frequency of words within all the document and finally, a Support Vector Machine model with Stochastic gradient descent that is used to calculate or predict the probability/weights of the highest frequency occurring words.

The reason why we have used the Support Vector machines model with the stochastic gradient descent is because the Support vector machine model can better classify the non-linear boundaries and SGD - stochastic gradient descent is a simple variant of classical gradient descent where the stochasticity comes from employing a random subset of the measurements (mini-batch) to compute the gradient at each descent. It also has implicit regularization effects, making it suited for highly non-convex loss functions, such as those entailed in training deep networks for classification.

We have implemented the following preprocessing steps and ran the code Sentiment-yelp.py in the cluster to produce weights for the top words:

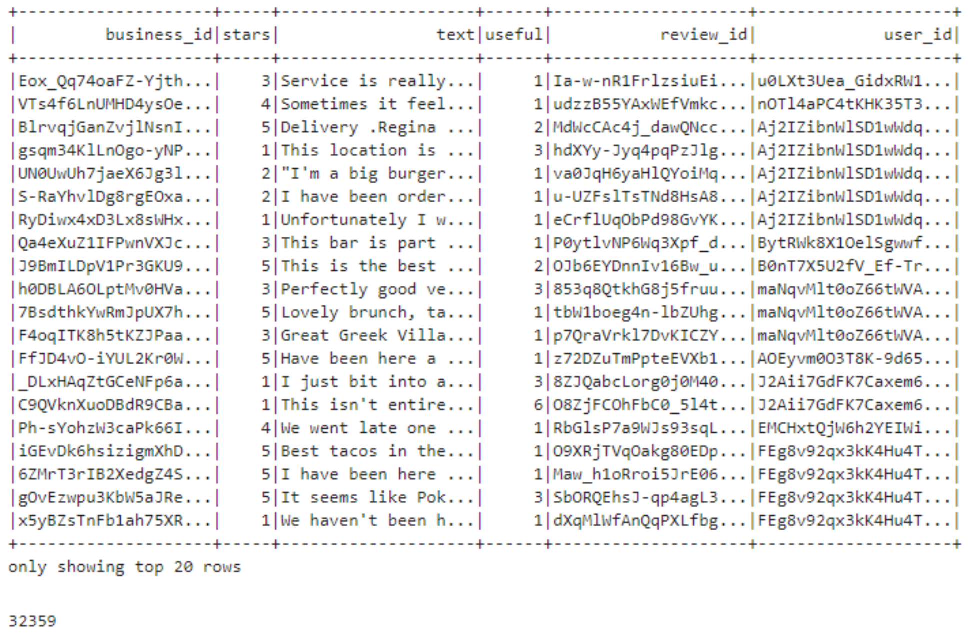
Notes to the Instructor/Grader: I tried running the file Sentiment_Yelp.py on the cluster but it errored out half way through the code. So I re-ran the code on my local machine and got rest of the screen shots of the output from the i-python notebook running on my local machine. Please let me know if you would like to see the ipython notebook used for generating the pyspark code and I would be happy to email you my code at the earliest.
Thank you, Shreyashi Mukhopadhyay.

# Running the code on the cluster:



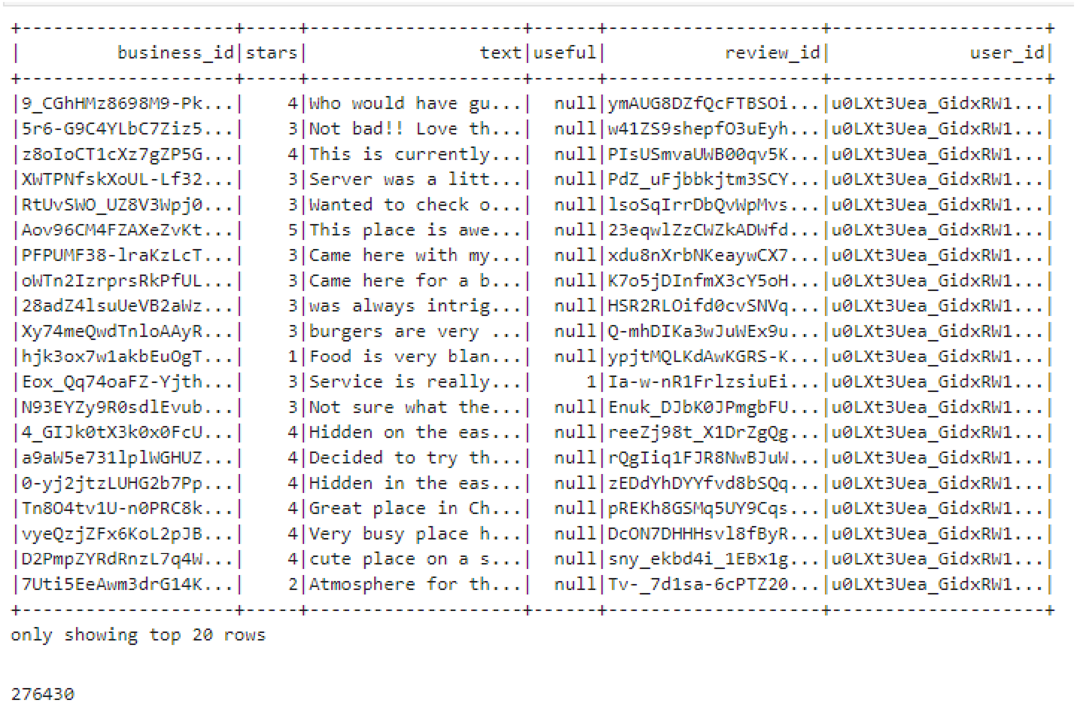
```
    import graphframes
ModuleNotFoundError: No module named 'graphframes'
leticiadavordzi@cluster-c9c5-m:~/Yelp_Sentiment_Files$ vim Sentiment_Yelp.py
leticiadavordzi@cluster-c9c5-m:~/Yelp_Sentiment_Files$ spark-submit Sentiment_Yelp.py
22/04/26 21:19:05 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/04/26 21:19:05 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/04/26 21:19:05 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/04/26 21:19:05 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
22/04/26 21:19:05 INFO org.sparkproject.jetty.util.log: Logging initialized @4417ms to org.sparkproject.jetty.util.log.Slf4jLog
22/04/26 21:19:06 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_322-b06
22/04/26 21:19:06 INFO org.sparkproject.jetty.server.Server: Started @4506ms
22/04/26 21:19:06 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@5d0d99b2{HTTP/1.1, (http/1.1)}{0.0.0.0:43041}
22/04/26 21:19:08 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists w
ith desired state.
+--------------------+--------------------+------------+--------------------+--------------+---------+-----------+----------+-----------+------------+------------+-------+--------------------+
|         business_id|                name|neighborhood|             address|          city|    state|postal_code|  latitude|  longitude|       stars|review_count|is_open|          categories|
+--------------------+--------------------+------------+--------------------+--------------+---------+-----------+----------+-----------+------------+------------+-------+--------------------+
|FYWN1wneV18bWNgQj...|"""Dental by Desi...|        null| """4855 E Warner Rd|     Ste B9"""|Ahwatukee|         AZ|     85044| 33.3306902|-111.9785992|         4.0|     22|                   1|
|He-G7vWjzVUysIKrf...|"""Stephen Szabo ...|        null|"""3101 Washingto...|      McMurray|       PA|      15317|40.2916853|-80.1048999|         3.0|          11|      1|Hair Stylists;Hai...|
|KQPW8lFf1y5BT2Mxi...|"""Western Motor ...|        null|  """6025 N 27th Ave|      Ste 1"""|  Phoenix|         AZ|     85017| 33.5249025|-112.1153098|         1.5|     18|                   1|
|8DShNS-LuFqpEWIp0...|"""Sports Authori...|        null|"""5000 Arizona M...|    Ste 435"""|    Tempe|         AZ|     85282| 33.3831468|-111.9647254|         3.0|      9|                   0|
|PfOCPjBrlQAnz__NX...|"""Brick House Ta...|        null|  """581 Howe Ave"""|Cuyahoga Falls|       OH|      44221|41.1195346|-81.4756898|         3.5|         116|      1|American (New);Ni...|
+--------------------+--------------------+------------+--------------------+--------------+---------+-----------+----------+-----------+------------+------------+-------+--------------------+
only showing top 5 rows

root
 |-- business_id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- neighborhood: string (nullable = true)
 |-- address: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- postal_code: string (nullable = true)
 |-- latitude: string (nullable = true)
```

1. Filter the cleaned data on   categories = restaurant, city = Toronto, stars >0 , Useful >0 and 32359 records are returned.



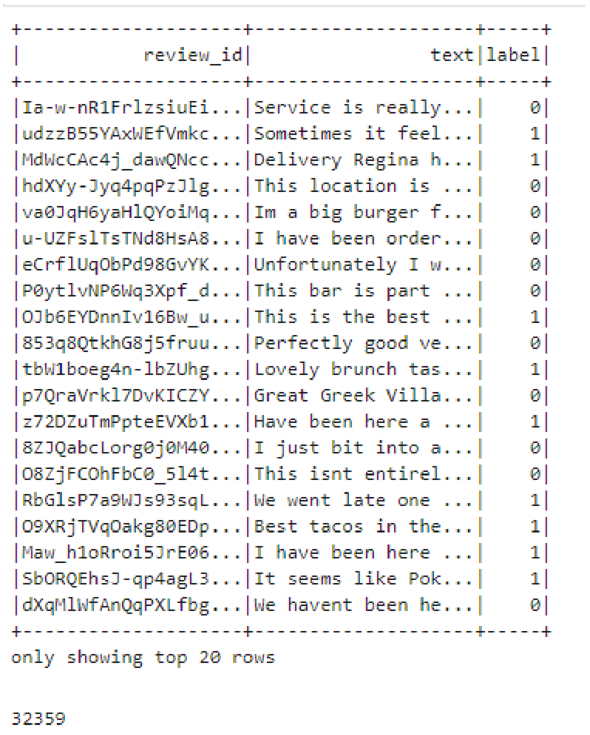
```
+--------------------+-----+--------------------+------+--------------------+--------------------+
|         business_id|stars|                text|useful|           review_id|             user_id|
+--------------------+-----+--------------------+------+--------------------+--------------------+
|Eox_Qq74oaFZ-Yjth...|    3|Service is really...|     1|Ia-w-nR1FrlzsiuEi...|u0LXt3Uea_GidxRW1...|
|VTs4f6LnUMHD4ysOe...|    4|Sometimes it feel...|     1|udzzB55YAxWEfVmkc...|nOTl4aPC4tKHK35T3...|
|BlrvqjGanZvjlNsnI...|    5|Delivery .Regina ...|     2|MdWcCAc4j_dawQNcc...|Aj2IZibnWlSD1wWdq...|
|gsqm34KlLnOgo-yNP...|    1|This location is ...|     3|hdXYy-Jyq4pqPzJlg...|Aj2IZibnWlSD1wWdq...|
|UN0UwUh7jaeX6Jg3l...|    2|"I'm a big burger...|     1|va0JqH6yaHlQYoiMq...|Aj2IZibnWlSD1wWdq...|
|S-RaYhvlDg8rgEOxa...|    2|I have been order...|     1|u-UZFslTsTNd8HsA8...|Aj2IZibnWlSD1wWdq...|
|RyDiwx4xD3Lx8sWHx...|    1|Unfortunately I w...|     1|eCrflUqObPd98GvYK...|Aj2IZibnWlSD1wWdq...|
|Qa4eXuZ1IFPwnVXJc...|    3|This bar is part ...|     1|P0ytlvNP6Wq3Xpf_d...|BytRWk8X1OelSgwwf...|
|J9BmILDpV1Pr3GKU9...|    5|This is the best ...|     2|OJb6EYDnnIv16Bw_u...|B0nT7X5U2fV_Ef-Tr...|
|h0DBLA6OLptMv0HVa...|    3|Perfectly good ve...|     3|853q8QtkhG8j5fruu...|maNqvMlt0oZ66tWVA...|
|7BsdthkYwRmJpUX7h...|    5|Lovely brunch, ta...|     1|tbW1boeg4n-lbZUhg...|maNqvMlt0oZ66tWVA...|
|F4oqITK8h5tKZJPaa...|    3|Great Greek Villa...|     1|p7QraVrkl7DvKICZY...|maNqvMlt0oZ66tWVA...|
|FfJD4vO-iYUL2Kr0W...|    5|Have been here a ...|     1|z72DZuTmPpteEVXb1...|AOEyvm0O3T8K-9d65...|
|_DLxHAqZtGCeNFp6a...|    1|I just bit into a...|     3|8ZJQabcLorg0j0M40...|J2Aii7GdFK7Caxem6...|
|C9QVknXuoDBdR9CBa...|    1|This isn't entire...|     6|O8ZjFCOhFbC0_5l4t...|J2Aii7GdFK7Caxem6...|
|Ph-sYohzW3caPk66I...|    4|We went late one ...|     1|RbGlsP7a9WJs93sqL...|EMCHxtQjW6h2YEIWi...|
|iGEvDk6hsizigmXhD...|    5|Best tacos in the...|     1|O9XRjTVqOakg80EDp...|FEg8v92qx3kK4Hu4T...|
|6ZMrT3rIB2XedgZ4S...|    5|I have been here ...|     1|Maw_h1oRroi5JrE06...|FEg8v92qx3kK4Hu4T...|
|gOvEzwpu3KbW5aJRe...|    5|It seems like Pok...|     3|SbORQEhsJ-qp4agL3...|FEg8v92qx3kK4Hu4T...|
|x5yBZsTnFb1ah75XR...|    1|We haven't been h...|     1|dXqMlWfAnQqPXLfbg...|FEg8v92qx3kK4Hu4T...|
+--------------------+-----+--------------------+------+--------------------+--------------------+
only showing top 20 rows

32359
```

2. Tokenize the data and add a tokenized column to the tokenized_review dataframe.



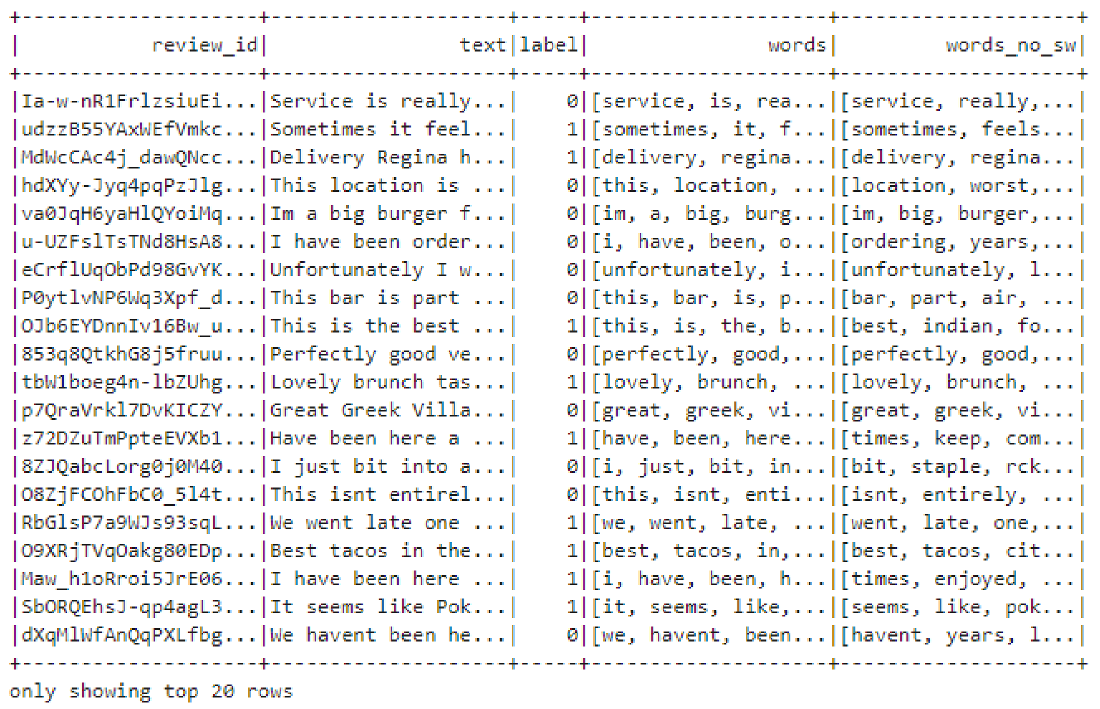
```
+--------------------+-----+--------------------+------+--------------------+--------------------+
|         business_id|stars|                text|useful|           review_id|             user_id|
+--------------------+-----+--------------------+------+--------------------+--------------------+
|9_CGhHMz8698M9-Pk...|    4|Who would have gu...|  null|ymAUG8DZfQcFTBSOi...|u0LXt3Uea_GidxRW1...|
|5r6-G9C4YLbC7Ziz5...|    3|Not bad!! Love th...|  null|w41ZS9shepfO3uEyh...|u0LXt3Uea_GidxRW1...|
|z8oIoCT1cXz7gZP5G...|    4|This is currently...|  null|PIsUSmvaUWB00qv5K...|u0LXt3Uea_GidxRW1...|
|XWTPNfskXoUL-Lf32...|    3|Server was a litt...|  null|PdZ_uFjbbkjtm3SCY...|u0LXt3Uea_GidxRW1...|
|RtUvSWO_UZ8V3Wpj0...|    3|Wanted to check o...|  null|lsoSqIrrDbQvWpMvs...|u0LXt3Uea_GidxRW1...|
|Aov96CM4FZAXeZvKt...|    5|This place is awe...|  null|23eqwlZzCWZkADWfd...|u0LXt3Uea_GidxRW1...|
|PFPUMF38-lraKzLcT...|    3|Came here with my...|  null|xdu8nXrbNKeaywCX7...|u0LXt3Uea_GidxRW1...|
|oWTn2IzrprsRkPfUL...|    3|Came here for a b...|  null|K7o5jDInfmX3cY5oH...|u0LXt3Uea_GidxRW1...|
|28adZ4lsuUeVB2aWz...|    3|was always intrig...|  null|HSR2RLOifd0cvSNVq...|u0LXt3Uea_GidxRW1...|
|Xy74meQwdTnloAAyR...|    3|burgers are very ...|  null|Q-mhDIKa3wJuWEx9u...|u0LXt3Uea_GidxRW1...|
|hjk3ox7w1akbEuOgT...|    1|Food is very blan...|  null|ypjtMQLKdAwKGRS-K...|u0LXt3Uea_GidxRW1...|
|Eox_Qq74oaFZ-Yjth...|    3|Service is really...|     1|Ia-w-nR1FrlzsiuEi...|u0LXt3Uea_GidxRW1...|
|N93EYZy9R0sdlEvub...|    3|Not sure what the...|  null|Enuk_DJbK0JPmgbFU...|u0LXt3Uea_GidxRW1...|
|4_GIJk0tX3k0x0FcU...|    4|Hidden on the eas...|  null|reeZj98t_X1DrZgQg...|u0LXt3Uea_GidxRW1...|
|a9aW5e731lplWGHUZ...|    4|Decided to try th...|  null|rQgIiq1FJR8NwBJuW...|u0LXt3Uea_GidxRW1...|
|0-yj2jtzLUHG2b7Pp...|    4|Hidden in the eas...|  null|zEDdYhDYYfvd8bSQq...|u0LXt3Uea_GidxRW1...|
|Tn8O4tv1U-n0PRC8k...|    4|Great place in Ch...|  null|pREKh8GSMq5UY9Cqs...|u0LXt3Uea_GidxRW1...|
|vyeQzjZFx6KoL2pJB...|    4|Very busy place h...|  null|DcON7DHHHsvl8fByR...|u0LXt3Uea_GidxRW1...|
|D2PmpZYRdRnzL7q4W...|    4|cute place on a s...|  null|sny_ekbd4i_1EBx1g...|u0LXt3Uea_GidxRW1...|
|7Uti5EeAwm3drG14K...|    2|Atmosphere for th...|  null|Tv-_7d1sa-6cPTZ20...|u0LXt3Uea_GidxRW1...|
+--------------------+-----+--------------------+------+--------------------+--------------------+
only showing top 20 rows

276430
```

3. Define a function that uses the Regular Expression (re module) to remove punctuations and numbers from the text column using the udf function.



```
+--------------------+--------------------+-----+
|           review_id|                text|label|
+--------------------+--------------------+-----+
|Ia-w-nR1FrlzsiuEi...|Service is really...|    0|
|udzzB55YAxWEfVmkc...|Sometimes it feel...|    1|
|MdWcCAc4j_dawQNcc...|Delivery Regina h...|    1|
|hdXYy-Jyq4pqPzJlg...|This location is ...|    0|
|va0JqH6yaHlQYoiMq...|Im a big burger f...|    0|
|u-UZFslTsTNd8HsA8...|I have been order...|    0|
|eCrflUqObPd98GvYK...|Unfortunately I w...|    0|
|P0ytlvNP6Wq3Xpf_d...|This bar is part ...|    0|
|OJb6EYDnnIv16Bw_u...|This is the best ...|    1|
|853q8QtkhG8j5fruu...|Perfectly good ve...|    0|
|tbW1boeg4n-lbZUhg...|Lovely brunch tas...|    1|
|p7QraVrkl7DvKICZY...|Great Greek Villa...|    0|
|z72DZuTmPpteEVXb1...|Have been here a ...|    1|
|8ZJQabcLorg0j0M40...|I just bit into a...|    0|
|O8ZjFCOhFbC0_5l4t...|This isnt entirel...|    0|
|RbGlsP7a9WJs93sqL...|We went late one ...|    1|
|O9XRjTVqOakg80EDp...|Best tacos in the...|    1|
|Maw_h1oRroi5JrE06...|I have been here ...|    1|
|SbORQEhsJ-qp4agL3...|It seems like Pok...|    1|
|dXqMlWfAnQqPXLfbg...|We havent been he...|    0|
+--------------------+--------------------+-----+
only showing top 20 rows

32359
```

4. Apply stop words and perform tokenization on the text column data.



```
+--------------------+--------------------+-----+--------------------+--------------------+
|           review_id|                text|label|               words|         words_no_sw|
+--------------------+--------------------+-----+--------------------+--------------------+
|Ia-w-nR1FrlzsiuEi...|Service is really...|    0|[service, is, rea...|[service, really,...|
|udzzB55YAxWEfVmkc...|Sometimes it feel...|    1|[sometimes, it, f...|[sometimes, feels...|
|MdWcCAc4j_dawQNcc...|Delivery Regina h...|    1|[delivery, regina...|[delivery, regina...|
|hdXYy-Jyq4pqPzJlg...|This location is ...|    0|[this, location, ...|[location, worst,...|
|va0JqH6yaHlQYoiMq...|Im a big burger f...|    0|[im, a, big, burg...|[im, big, burger,...|
|u-UZFslTsTNd8HsA8...|I have been order...|    0|[i, have, been, o...|[ordering, years,...|
|eCrflUqObPd98GvYK...|Unfortunately I w...|    0|[unfortunately, i...|[unfortunately, l...|
|P0ytlvNP6Wq3Xpf_d...|This bar is part ...|    0|[this, bar, is, p...|[bar, part, air, ...|
|OJb6EYDnnIv16Bw_u...|This is the best ...|    1|[this, is, the, b...|[best, indian, fo...|
|853q8QtkhG8j5fruu...|Perfectly good ve...|    0|[perfectly, good,...|[perfectly, good,...|
|tbW1boeg4n-lbZUhg...|Lovely brunch tas...|    1|[lovely, brunch, ...|[lovely, brunch, ...|
|p7QraVrkl7DvKICZY...|Great Greek Villa...|    0|[great, greek, vi...|[great, greek, vi...|
|z72DZuTmPpteEVXb1...|Have been here a ...|    1|[have, been, here...|[times, keep, com...|
|8ZJQabcLorg0j0M40...|I just bit into a...|    0|[i, just, bit, in...|[bit, staple, rck...|
|O8ZjFCOhFbC0_514t...|This isnt entirel...|    0|[this, isnt, enti...|[isnt, entirely, ...|
|RbGlsP7a9WJs93sqL...|We went late one ...|    1|[we, went, late, ...|[went, late, one,...|
|O9XRjTVqOakg80EDp...|Best tacos in the...|    1|[best, tacos, in,...|[best, tacos, cit...|
|Maw_h1oRroi5JrE06...|I have been here ...|    1|[i, have, been, h...|[times, enjoyed, ...|
|SbORQEhsJ-qp4agL3...|It seems like Pok...|    1|[it, seems, like,...|[seems, like, pok...|
|dXqMlWfAnQqPXLfbg...|We havent been he...|    0|[we, havent, been...|[havent, years, l...|
+--------------------+--------------------+-----+--------------------+--------------------+
only showing top 20 rows
```

5. Add a trigram column to the tokenized_review dataframe and preview the top 50 trigrams

```
['the service is', 'we will be', 'bit of a', 'the quality of', 'it was a', 'will not be', 'food was great', 'and the staff', 'to eat here', 'great selection of', 'wife and i', 'a bunch of', 'you pay for', 'in your mouth', 'had a great', 'the pizza was', 'is very friendly', 'go back again', 'to try this', 'are looking for', 'i have been', 'this place out', 'here for the', 'the service was', 'at this place', 'i like the', 'place in the', 'it is a', 'i come here', 'food is amazing', 'was very good', 'went there for', 'came here on', 'this place a', 'i will be', 'the taste of', 'i thought it', 'give this place', 'by far the', 'in the neighbourhood', 'have to say', 'highly recommend this', 'i ended up', 'was the best', 'on top of', 'addition to the', 'friendly and the', 'any of the', 'the area and', 'i would go']
```

6. Trigrams preprocessing:

# Perform tokenization and remove stop words again on the trigrams

# Use Count vectorizer and TF-IDF

```
+--------------------+-----+--------------------+--------------------+--------------------+--------------------+
|                text|label|               words|         words_no_sw|                  tf|               tfidf|
+--------------------+-----+--------------------+--------------------+--------------------+--------------------+
|service is really...|    0|[service, is, rea...|[service, really,...|(44846,[0,1,4,7,3...|(44846,[0,1,4,7,3...|
|sometimes it feel...|    1|[sometimes, it, f...|[sometimes, feels...|(44846,[2,7,17,18...|(44846,[2,7,17,18...|
|delivery regina h...|    1|[delivery, regina...|[delivery, regina...|(44846,[0,5,9,12,...|(44846,[0,5,9,12,...|
|this location is ...|    0|[this, location, ...|[location, worst,...|(44846,[0,1,6,7,8...|(44846,[0,1,6,7,8...|
|im a big burger f...|    0|[im, a, big, burg...|[im, big, burger,...|(44846,[0,2,6,7,9...|(44846,[0,2,6,7,9...|
+--------------------+-----+--------------------+--------------------+--------------------+--------------------+
only showing top 5 rows
```

7. Replace Unigrams in the Text Corresponding to the Selected Trigrams

8. Run the same pipeline of Tokenize --> CountVectorizer (BagOfWords) --> TF-IDF to the New Text.

9. Run a SVM with SGD Model on the Transformed and Vectorized Data.

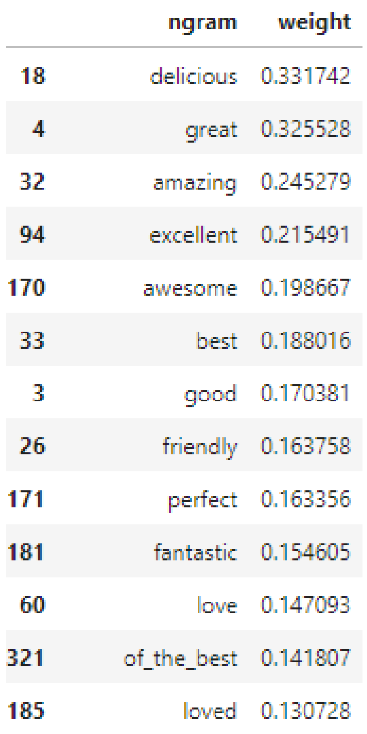
Model Performance:

```
F1 score: 0.8781
Area under ROC: 0.8687
Area under PR: 0.8662
```

10. Extract the top 10 Negative weighted words:

	ngram	weight
274	worst	-0.211198
278	bland	-0.203700
189	ok	-0.186743
248	terrible	-0.183859
82	bad	-0.182215
395	overpriced	-0.177729
431	mediocre	-0.172336
311	rude	-0.169969
226	average	-0.167883
436	disappointing	-0.166319
122	nothing	-0.161750
400	horrible	-0.155763

11. Extract the top 10 Positive words

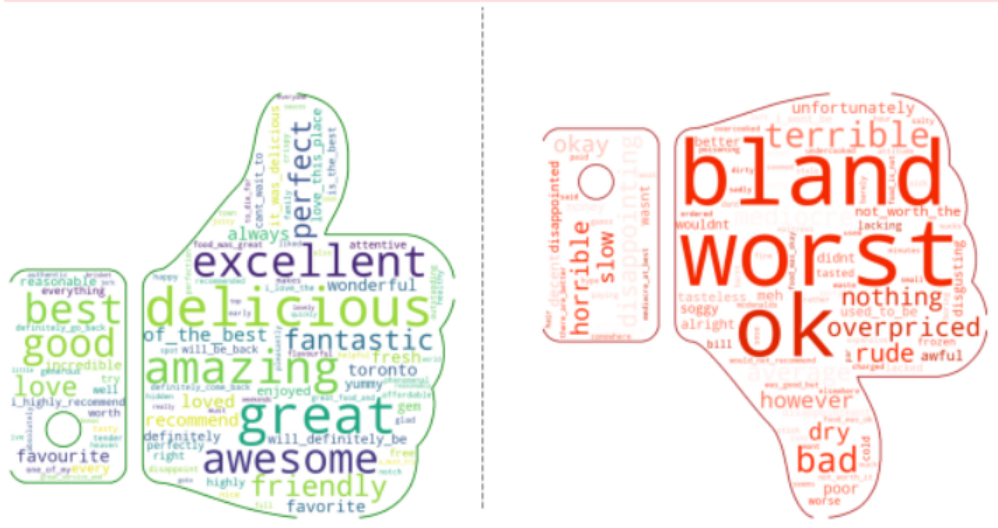


	ngram	weight
18	delicious	0.331742
4	great	0.325528
32	amazing	0.245279
94	excellent	0.215491
170	awesome	0.198667
33	best	0.188016
3	good	0.170381
26	friendly	0.163758
171	perfect	0.163356
181	fantastic	0.154605
60	love	0.147093
321	of_the_best	0.141807
185	loved	0.130728

12. Plot a word cloud using the above positive and negative weighted words.

Generate word cloud

```
# Create a WordCloud for better visualization
# Read in the masks to be used when plotting the word clouds

!pip install random
import random

pos_mask = np.array(Image.open("thumbspos.png"))
neg_mask = np.array(Image.open("thumbsdown.png"))


# Generate the word cloud for the positive reviews
d1 = {}
for a, x in pos.values:
    d1[a] = x


wordcloud = WordCloud(width=1600, height=800, max_words=100, background_color="white",
                      mask=pos_mask, contour_width=3, contour_color='green') \
                      .generate_from_frequencies(frequencies=d1)


# Generate the word cloud for the negative reviews
d2 = {}
for a, x in pos.values:
    d2[a] = -x


def red_color_func(word, font_size, position, orientation, random_state=None, **kwargs):
    return "hsl(10, 100%%, %d%%)" % random.randint(40, 100)


wordcloud2 = WordCloud(width=1600, height=800, max_words=100, background_color="white",
                       mask=neg_mask, contour_width=3, contour_color='firebrick') \
                      .generate_from_frequencies(frequencies=d2) \
                      .recolor(color_func = red_color_func)


# Plot the wordclouds side by side
fig = plt.figure(figsize=(20,16))
plt.subplot(1,2,1)
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")

plt.subplot(1,2,2)
plt.imshow(wordcloud2, interpolation="bilinear")
plt.axis("off")

line = plt.Line2D((.5,.5),(.3,.8), color="grey", linewidth=2, linestyle = '--')
fig.add_artist(line)

plt.show()
```

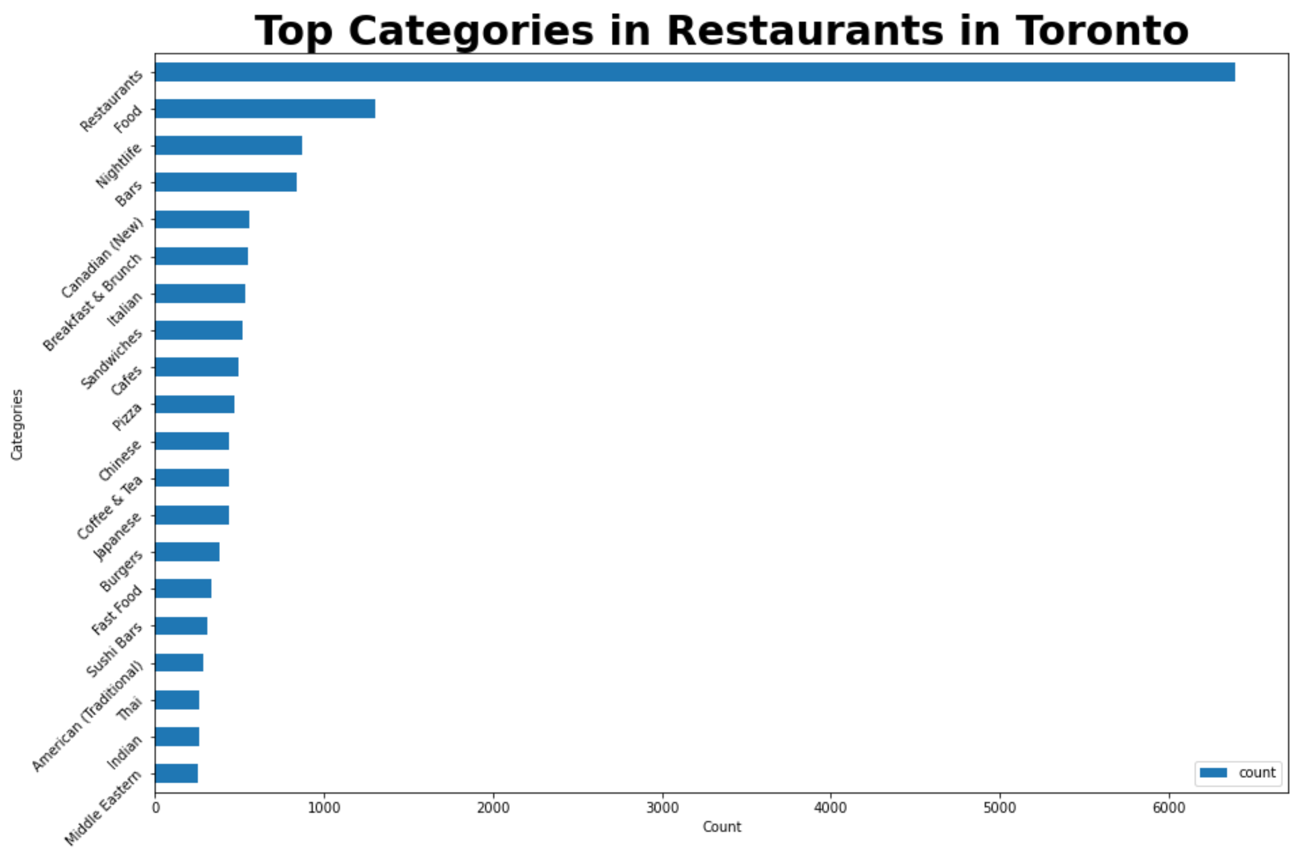


Visualizing the top Restaurant categories in Toronto: (jupyter notebook)

```python
# Split categories to each distinct category
from pyspark.sql.functions import explode, split

business_id_categories = df_business.withColumn("categories", explode(split('categories', ";")))
business_id_categories.show(5)
```

```
+--------------------+--------------------+--------------------+--------------------+-------+-----+----------+-----------+-----+------------+-----------+
|         business_id|                name|        neighborhood|             address|   city|state|  latitude|  longitude|stars|review_count| categories|
+--------------------+--------------------+--------------------+--------------------+-------+-----+----------+-----------+-----+------------+-----------+
|109JfMeQ6ynYs5MCJ...|"""Alize Catering"""|Yonge and Eglinton|  """2459 Yonge St"""|Toronto|   ON|43.7113993|-79.3993388|  3.0|          12|    Italian|
|109JfMeQ6ynYs5MCJ...|"""Alize Catering"""|Yonge and Eglinton|  """2459 Yonge St"""|Toronto|   ON|43.7113993|-79.3993388|  3.0|          12|     French|
|109JfMeQ6ynYs5MCJ...|"""Alize Catering"""|Yonge and Eglinton|  """2459 Yonge St"""|Toronto|   ON|43.7113993|-79.3993388|  3.0|          12|Restaurants|
|1K4qrnfyzKzGgJPBE...|"""Chula Taberna ...|        Leslieville|"""1058 Gerrard S...|Toronto|   ON|43.6692562|-79.3359022|  3.5|          39|  Tiki Bars|
|1K4qrnfyzKzGgJPBE...|"""Chula Taberna ...|        Leslieville|"""1058 Gerrard S...|Toronto|   ON|43.6692562|-79.3359022|  3.5|          39|  Nightlife|
+--------------------+--------------------+--------------------+--------------------+-------+-----+----------+-----------+-----+------------+-----------+
only showing top 5 rows
```

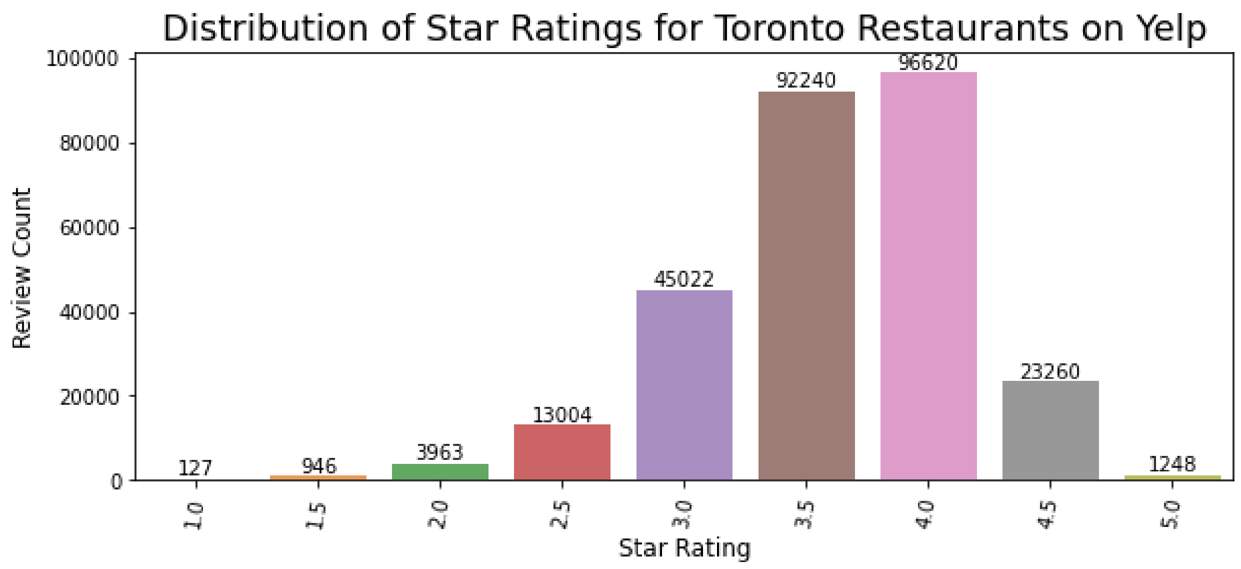
```python
top_category = business_id_categories.groupby("categories").count().orderBy('count', ascending=False).limit(20).toPandas()
top_category = top_category.set_index('categories','count')
top_category = top_category.sort_values(by='count', ascending=True)
top_category
```



Visualizing the star ratings across restaurants in Toronto:

```
# Extract all ratings from the Review dataset and group them by the star ratings
stars_distr = df_review.groupBy('stars').agg(func.count('review_id') \
                    .alias('count')).sort('stars').toPandas()

#stars_distr = stars_distr.iloc[24:29, ]
stars_distr
```

	stars	count
0	1.0	127
1	1.5	946
2	2.0	3963
3	2.5	13004
4	3.0	45022
5	3.5	92240
6	4.0	96620
7	4.5	23260
8	5.0	1248



Conclusion from the visualizations: There are more than 6391 restaurants in Toronto which indicates that people do spend a lot of money dining out and have very strong preferences with respect to the kind of experience that is important to them. The consumer sentiment is also more positive than negative in the city of Toronto since we have more number of stars for the ratings from 3.0 and above but not a lot of 5 star reviews so that can be a place of improvement recommended for the restaurants.

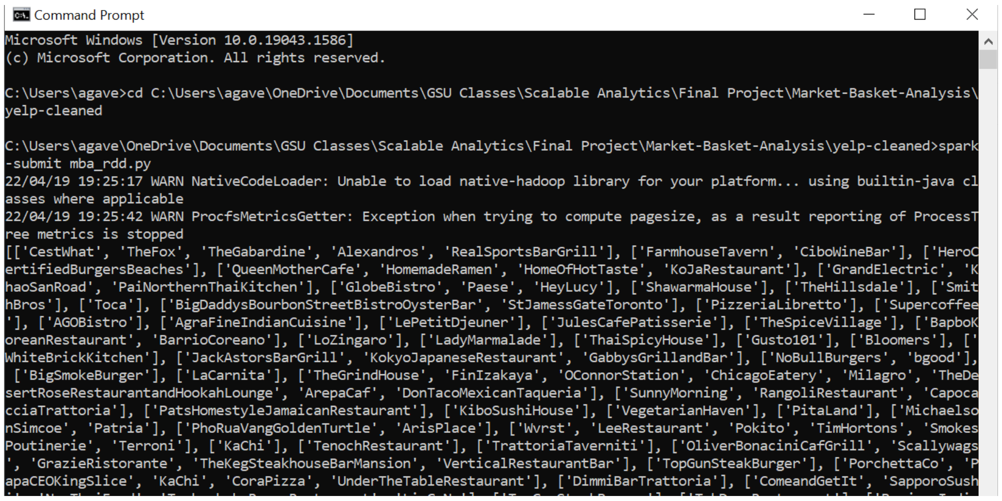
Market Basket Analysis with RDDs

The probability of a person liking a particular restaurant will be based on a bunch of different attributes. These attributes could be what they have said in past reviews, what types of restaurants they have rated highly in the past, or general socio demographic information. However, what happens if you don’t have this information?

If this is the case, you can use a method known as “Market Basket Analysis.” The goal of Market Basket Analysis is aimed at discovering which groups of products tend to be purchased together, or in this case, which groups of restaurants are rated highly by yelp users.

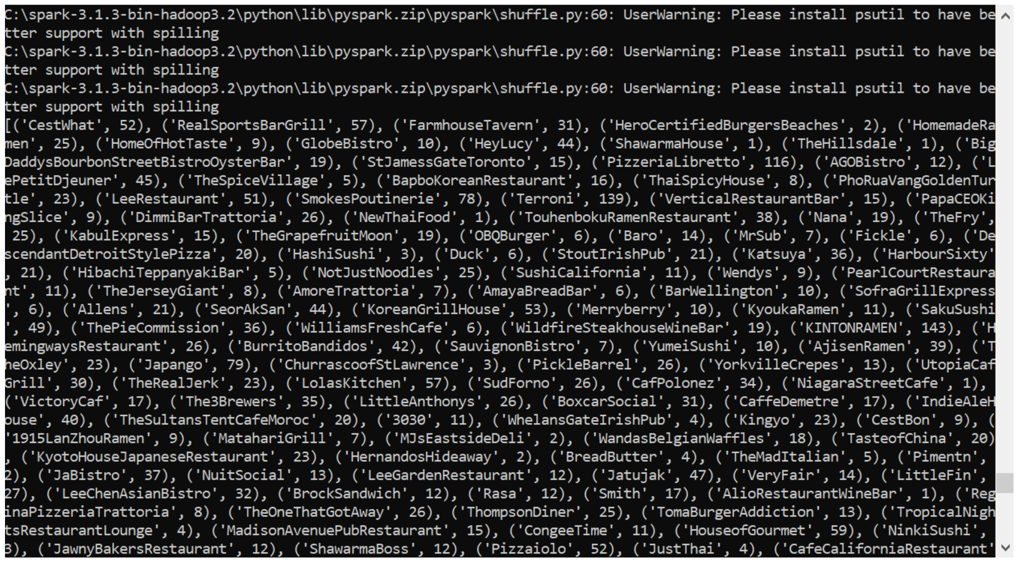
Because of how Market Basket Analysis looks at how many times a specific grouping has occurred, we thought this was a perfect chance to make use of RDD’s to implement the method.

Below is a screen shot of the results from running the python program on the cloud. The output returns lists of each highly rated restaurant by customer.



```
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\agave>cd C:\Users\agave\OneDrive\Documents\GSU Classes\Scalable Analytics\Final Project\Market-Basket-Analysis\
yelp-cleaned

C:\Users\agave\OneDrive\Documents\GSU Classes\Scalable Analytics\Final Project\Market-Basket-Analysis\yelp-cleaned>spark
-submit mba_rdd.py
22/04/19 19:25:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
22/04/19 19:25:42 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessT
ree metrics is stopped
[['CestWhat', 'TheFox', 'TheGabardine', 'Alexandros', 'RealSportsBarGrill'], ['FarmhouseTavern', 'CiboWineBar'], ['HeroC
ertifiedBurgersBeaches'], ['QueenMotherCafe', 'HomemadeRamen', 'HomeOfHotTaste', 'KoJaRestaurant'], ['GrandElectric', 'K
haoSanRoad', 'PaiNorthernThaiKitchen'], ['GlobeBistro', 'Paese', 'HeyLucy'], ['ShawarmaHouse'], ['TheHillsdale'], ['Smit
hBros'], ['Toca'], ['BigDaddysBourbonStreetBistroOysterBar', 'StJamessGateToronto'], ['PizzeriaLibretto'], ['Supercoffee
'], ['AGOBistro'], ['AgraFineIndianCuisine'], ['LePetitDjeuner'], ['JulesCafePatisserie'], ['TheSpiceVillage'], ['BapboK
oreanRestaurant', 'BarrioCoreano'], ['LoZingaro'], ['LadyMarmalade'], ['ThaiSpicyHouse'], ['Gusto101'], ['Bloomers'], ['
WhiteBrickKitchen'], ['JackAstorsBarGrill', 'KokyoJapaneseRestaurant', 'GabbysGrillandBar'], ['NoBullBurgers', 'bgood'],
 ['BigSmokeBurger'], ['LaCarnita'], ['TheGrindHouse', 'FinIzakaya', 'OConnorStation', 'ChicagoEatery', 'Milagro', 'TheDe
sertRoseRestaurantandHookahLounge', 'ArepaCaf', 'DonTacoMexicanTaqueria'], ['SunnyMorning', 'RangoliRestaurant', 'Capoca
cciaTrattoria'], ['PatsHomestyleJamaicanRestaurant'], ['KiboSushiHouse'], ['VegetarianHaven'], ['PitaLand'], ['Michaelso
nSimcoe', 'Patria'], ['PhoRuaVangGoldenTurtle', 'ArisPlace'], ['Wvrst', 'LeeRestaurant', 'Pokito', 'TimHortons', 'Smokes
Poutinerie', 'Terroni'], ['KaChi'], ['TenochRestaurant'], ['TrattoriaTaverniti'], ['OliverBonaciniCafGrill', 'Scallywags
', 'GrazieRistorante', 'TheKegSteakhouseBarMansion', 'VerticalRestaurantBar'], ['TopGunSteakBurger'], ['PorchettaCo', 'P
apaCEOKingSlice', 'KaChi', 'CoraPizza', 'UnderTheTableRestaurant'], ['DimmiBarTrattoria'], ['ComeandGetIt', 'SapporoSush
```

We can then use a flat map to flatten the lists, and count how many times each restaurant was reviewed. This single grouping is known as the first support RDD. Output is below.



```
C:\spark-3.1.3-bin-hadoop3.2\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning: Please install psutil to have be
tter support with spilling
C:\spark-3.1.3-bin-hadoop3.2\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning: Please install psutil to have be
tter support with spilling
C:\spark-3.1.3-bin-hadoop3.2\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning: Please install psutil to have be
tter support with spilling
[('CestWhat', 52), ('RealSportsBarGrill', 57), ('FarmhouseTavern', 31), ('HeroCertifiedBurgersBeaches', 2), ('HomemadeRa
men', 25), ('HomeOfHotTaste', 9), ('GlobeBistro', 10), ('HeyLucy', 44), ('ShawarmaHouse', 1), ('TheHillsdale', 1), ('Big
DaddysBourbonStreetBistroOysterBar', 19), ('StJamessGateToronto', 15), ('PizzeriaLibretto', 116), ('AGOBistro', 12), ('L
ePetitDjeuner', 45), ('TheSpiceVillage', 5), ('BapboKoreanRestaurant', 16), ('ThaiSpicyHouse', 8), ('PhoRuaVangGoldenTur
tle', 23), ('LeeRestaurant', 51), ('SmokesPoutinerie', 78), ('Terroni', 139), ('VerticalRestaurantBar', 15), ('PapaCEOKi
ngSlice', 9), ('DimmiBarTrattoria', 26), ('NewThaiFood', 1), ('TouhenbokuRamenRestaurant', 38), ('Nana', 19), ('TheFry',
 25), ('KabulExpress', 15), ('TheGrapefruitMoon', 19), ('OBQBurger', 6), ('Baro', 14), ('MrSub', 7), ('Fickle', 6), ('De
scendantDetroitStylePizza', 20), ('HashiSushi', 3), ('Duck', 6), ('StoutIrishPub', 21), ('Katsuya', 36), ('HarbourSixty'
, 21), ('HibachiTeppanyakiBar', 5), ('NotJustNoodles', 25), ('SushiCalifornia', 11), ('Wendys', 9), ('PearlCourtRestaura
nt', 11), ('TheJerseyGiant', 8), ('AmoreTrattoria', 7), ('AmayaBreadBar', 6), ('BarWellington', 10), ('SofraGrillExpress
', 6), ('Allens', 21), ('SeorAkSan', 44), ('KoreanGrillHouse', 53), ('Merryberry', 10), ('KyoukaRamen', 11), ('SakuSushi
', 49), ('ThePieCommission', 36), ('WilliamsFreshCafe', 6), ('WildfireSteakhouseWineBar', 19), ('KINTONRAMEN', 143), ('H
emingwaysRestaurant', 26), ('BurritoBandidos', 42), ('SauvignonBistro', 7), ('YumeiSushi', 10), ('AjisenRamen', 39), ('T
heOxley', 23), ('Japango', 79), ('ChurrascoofStLawrence', 3), ('PickleBarrel', 26), ('YorkvilleCrepes', 13), ('UtopiaCaf
Grill', 30), ('TheRealJerk', 23), ('LolasKitchen', 57), ('SudForno', 26), ('CafPolonez', 34), ('NiagaraStreetCafe', 1),
('VictoryCaf', 17), ('The3Brewers', 35), ('LittleAnthonys', 26), ('BoxcarSocial', 31), ('CaffeDemetre', 17), ('IndieAleH
ouse', 40), ('TheSultansTentCafeMoroc', 20), ('3030', 11), ('WhelansGateIrishPub', 4), ('Kingyo', 23), ('CestBon', 9), (
'1915LanZhouRamen', 9), ('MatahariGrill', 7), ('MJsEastsideDeli', 2), ('WandasBelgianWaffles', 18), ('TasteofChina', 20)
, ('KyotoHouseJapaneseRestaurant', 23), ('HernandosHideaway', 2), ('BreadButter', 4), ('TheMadItalian', 5), ('Pimentn',
2), ('JaBistro', 37), ('NuitSocial', 13), ('LeeGardenRestaurant', 12), ('Jatujak', 47), ('VeryFair', 14), ('LittleFin',
27), ('LeeChenAsianBistro', 32), ('BrockSandwich', 12), ('Rasa', 12), ('Smith', 17), ('AlioRestaurantWineBar', 1), ('Reg
inaPizzeriaTrattoria', 8), ('TheOneThatGotAway', 26), ('ThompsonDiner', 25), ('TomaBurgerAddiction', 13), ('TropicalNigh
tsRestaurantLounge', 4), ('MadisonAvenuePubRestaurant', 15), ('CongeeTime', 11), ('HouseofGourmet', 59), ('NinkiSushi',
3), ('JawnyBakersRestaurant', 12), ('ShawarmaBoss', 12), ('Pizzaiolo', 52), ('JustThai', 4), ('CafeCaliforniaRestaurant'
```

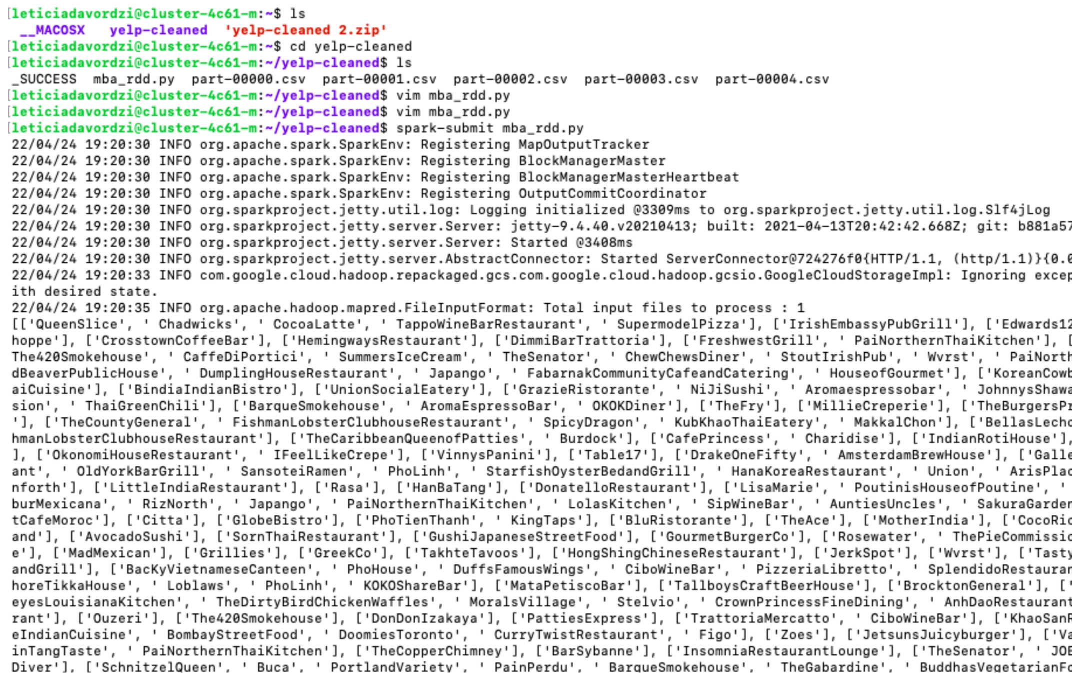
As you can see from support values, “RealSportsBarGrill” has more highly rated views than other restaurants. We can therefore assume that the probability of this restaurant being highly rated is larger than others, which is what is meant by “Support Value”. These support values are gotten by considering each item separately.

We then need to consider how restaurants are rated highly together, which is an additional layer of conditional probability. With both of these probabilities, we are then able to calculate the probability that a grouping will occur using Bayes Theorem, which is where our recommendation can start to take place. Therefore, the next step is to get common counts of combinations of highly rated restaurants and then calculate the confidence values which essentially tell how likely a customer is to buy A after having purchased B.

Considering time and computational ability, the part-0004 cleaned data csv was used for basket analysis. The part-0004.csv was chosen because it has less data which we suspected will work better for the long loops needed in the code. A union of all 5 csvs had been considered previously but the data was too much to run on a local system (run for 80 hours without producing any output) or on google cloud (run for 9 hours and produced a broken pipe error).

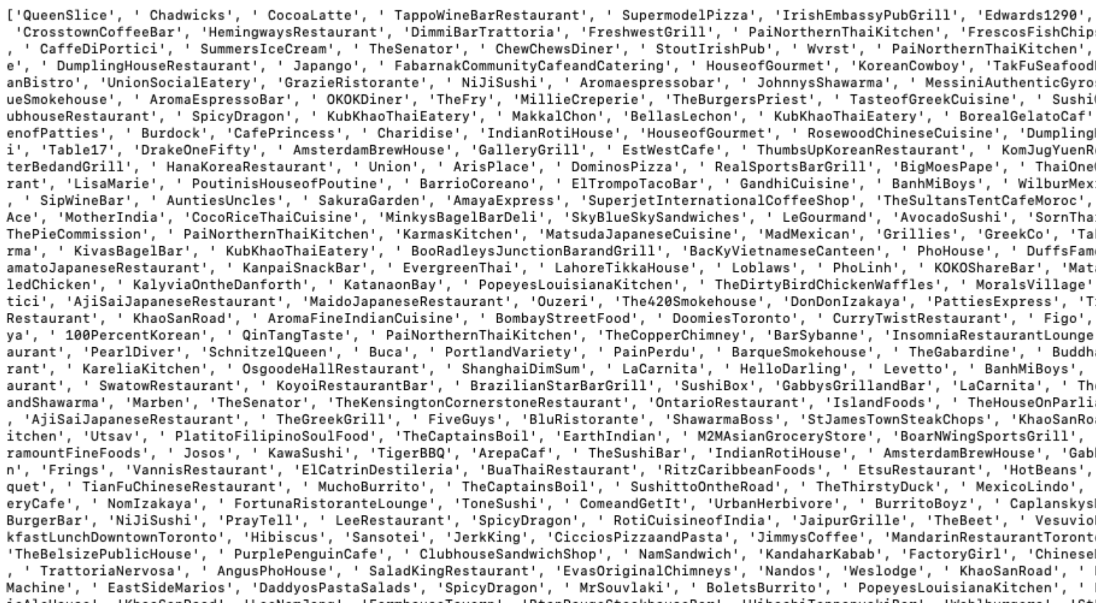
See below for the outputs when part-0004.csv was utilized:

1. The lists of restaurants reviewed by users.



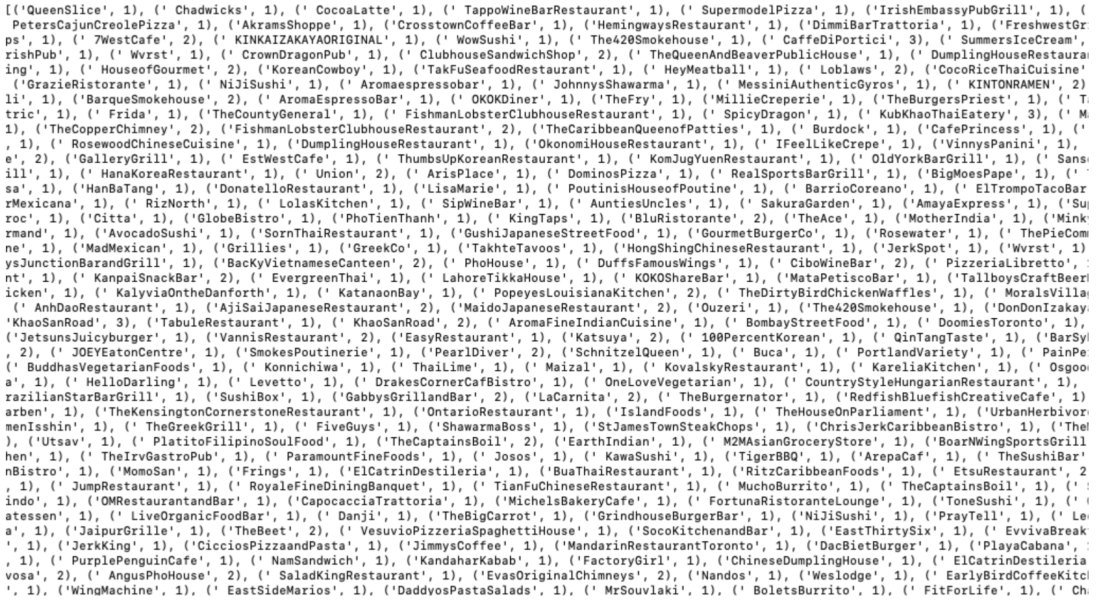
```
[leticiadavordzi@cluster-4c61-m:~$ ls
  __MACOSX   yelp-cleaned  'yelp-cleaned 2.zip'
[leticiadavordzi@cluster-4c61-m:~$ cd yelp-cleaned
[leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ ls
 _SUCCESS  mba_rdd.py  part-00000.csv  part-00001.csv  part-00002.csv  part-00003.csv  part-00004.csv
[leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ vim mba_rdd.py
[leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ vim mba_rdd.py
[leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ spark-submit mba_rdd.py
 22/04/24 19:20:30 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
 22/04/24 19:20:30 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
 22/04/24 19:20:30 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
 22/04/24 19:20:30 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
 22/04/24 19:20:30 INFO org.sparkproject.jetty.util.log: Logging initialized @3309ms to org.sparkproject.jetty.util.log.Slf4jLog
 22/04/24 19:20:30 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a57
 22/04/24 19:20:30 INFO org.sparkproject.jetty.server.Server: Started @3408ms
 22/04/24 19:20:30 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@724276f0{HTTP/1.1, (http/1.1)}{0.0
 22/04/24 19:20:33 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring excep
 ith desired state.
 22/04/24 19:20:35 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
 [['QueenSlice', ' Chadwicks', ' CocoaLatte', ' TappoWineBarRestaurant', ' SupermodelPizza'], ['IrishEmbassyPubGrill'], ['Edwards12
 hoppe'], ['CrosstownCoffeeBar'], ['HemingwaysRestaurant'], ['DimmiBarTrattoria'], ['FreshwestGrill', ' PaiNorthernThaiKitchen'], [
 The420Smokehouse', ' CaffeDiPortici', ' SummersIceCream', ' TheSenator', ' ChewChewsDiner', ' StoutIrishPub', ' Wvrst', ' PaiNorth
 dBeaverPublicHouse', ' DumplingHouseRestaurant', ' Japango', ' FabarnakCommunityCafeandCatering', ' HouseofGourmet'], ['KoreanCowb
 aiCuisine'], ['BindiaIndianBistro'], ['UnionSocialEatery'], ['GrazieRistorante', ' NiJiSushi', ' Aromaespressobar', ' JohnnysShawa
 sion', ' ThaiGreenChili'], ['BarqueSmokehouse', ' AromaEspressoBar', ' OKOKDiner'], ['TheFry'], ['MillieCreperie'], ['TheBurgersPr
 '], ['TheCountyGeneral', ' FishmanLobsterClubhouseRestaurant', ' SpicyDragon', ' KubKhaoThaiEatery', ' MakkalChon'], ['BellasLecho
 hmanLobsterClubhouseRestaurant'], ['TheCaribbeanQueenofPatties', ' Burdock'], ['CafePrincess', ' Charidise'], ['IndianRotiHouse'],
 ], ['OkonomiHouseRestaurant', ' IFeelLikeCrepe'], ['VinnysPanini'], ['Table17'], ['DrakeOneFifty', ' AmsterdamBrewHouse'], ['Galle
 ant', ' OldYorkBarGrill', ' SansoteiRamen', ' PhoLinh', ' StarfishOysterBedandGrill', ' HanaKoreaRestaurant', ' Union', ' ArisPlac
 nforth'], ['LittleIndiaRestaurant'], ['Rasa'], ['HanBaTang'], ['DonatelloRestaurant'], ['LisaMarie', ' PoutinisHouseofPoutine', '
 burMexicana', ' RizNorth', ' Japango', ' PaiNorthernThaiKitchen', ' LolasKitchen', ' SipWineBar', ' AuntiesUncles', ' SakuraGarden
 tCafeMoroc'], ['Citta'], ['GlobeBistro'], ['PhoTienThanh', ' KingTaps'], ['BluRistorante'], ['TheAce'], ['MotherIndia'], ['CocoRic
 and'], ['AvocadoSushi'], ['SornThaiRestaurant'], ['GushiJapaneseStreetFood'], ['GourmetBurgerCo'], ['Rosewater', ' ThePieCommissio
 e'], ['MadMexican'], ['Grillies'], ['GreekCo'], ['TakhteTavoos'], ['HongShingChineseRestaurant'], ['JerkSpot'], ['Wvrst'], ['Tasty
 andGrill'], ['BacKyVietnameseCanteen', ' PhoHouse', ' DuffsFamousWings', ' CiboWineBar', ' PizzeriaLibretto', ' SplendidoRestauran
 horeTikkaHouse', ' Loblaws', ' PhoLinh', ' KOKOShareBar'], ['MataPetiscoBar'], ['TallboysCraftBeerHouse'], ['BrocktonGeneral'], ['
 eyesLouisianaKitchen', ' TheDirtyBirdChickenWaffles', ' MoralsVillage', ' Stelvio', ' CrownPrincessFineDining', ' AnhDaoRestaurant
 rant'], ['Ouzeri'], ['The420Smokehouse'], ['DonDonIzakaya'], ['PattiesExpress'], ['TrattoriaMercatto', ' CiboWineBar'], ['KhaoSanR
 eIndianCuisine', ' BombayStreetFood', ' DoomiesToronto', ' CurryTwistRestaurant', ' Figo'], ['Zoes'], ['JetsunsJuicyburger'], ['Va
 inTangTaste', ' PaiNorthernThaiKitchen'], ['TheCopperChimney'], ['BarSybanne'], ['InsomniaRestaurantLounge'], ['TheSenator', ' JOE
 Diver'], ['SchnitzelQueen', ' Buca', ' PortlandVariety', ' PainPerdu', ' BarqueSmokehouse', ' TheGabardine', ' BuddhasVegetarianFo
```

2. List of all values from the previous lists but as one long list.



```
['QueenSlice', ' Chadwicks', ' CocoaLatte', ' TappoWineBarRestaurant', ' SupermodelPizza', 'IrishEmbassyPubGrill', 'Edwards1290',
 'CrosstownCoffeeBar', 'HemingwaysRestaurant', 'DimmiBarTrattoria', 'FreshwestGrill', ' PaiNorthernThaiKitchen', 'FrescosFishChip
, ' CaffeDiPortici', ' SummersIceCream', ' TheSenator', ' ChewChewsDiner', ' StoutIrishPub', ' Wvrst', ' PaiNorthernThaiKitchen',
e', ' DumplingHouseRestaurant', ' Japango', ' FabarnakCommunityCafeandCatering', ' HouseofGourmet', 'KoreanCowboy', 'TakFuSeafood
anBistro', 'UnionSocialEatery', 'GrazieRistorante', ' NiJiSushi', ' Aromaespressobar', ' JohnnysShawarma', ' MessiniAuthenticGyro
ueSmokehouse', ' AromaEspressoBar', ' OKOKDiner', 'TheFry', 'MillieCreperie', 'TheBurgersPriest', ' TasteofGreekCuisine', ' Sushi
ubhouseRestaurant', ' SpicyDragon', ' KubKhaoThaiEatery', ' MakkalChon', 'BellasLechon', ' KubKhaoThaiEatery', ' BorealGelatoCaf'
enofPatties', ' Burdock', 'CafePrincess', ' Charidise', 'IndianRotiHouse', 'HouseofGourmet', ' RosewoodChineseCuisine', 'Dumpling
i', 'Table17', 'DrakeOneFifty', ' AmsterdamBrewHouse', 'GalleryGrill', ' EstWestCafe', ' ThumbsUpKoreanRestaurant', ' KomJugYuenR
terBedandGrill', ' HanaKoreaRestaurant', ' Union', ' ArisPlace', ' DominosPizza', ' RealSportsBarGrill', 'BigMoesPape', ' ThaiOne
rant', 'LisaMarie', ' PoutinisHouseofPoutine', ' BarrioCoreano', ' ElTrompoTacoBar', ' GandhiCuisine', ' BanhMiBoys', ' WilburMex
, ' SipWineBar', ' AuntiesUncles', ' SakuraGarden', 'AmayaExpress', 'SuperjetInternationalCoffeeShop', 'TheSultansTentCafeMoroc',
Ace', 'MotherIndia', 'CocoRiceThaiCuisine', 'MinkysBagelBarDeli', 'SkyBlueSkySandwiches', ' LeGourmand', 'AvocadoSushi', 'SornTha
ThePieCommission', ' PaiNorthernThaiKitchen', 'KarmasKitchen', 'MatsudaJapaneseCuisine', 'MadMexican', 'Grillies', 'GreekCo', 'Ta
rma', ' KivasBagelBar', ' KubKhaoThaiEatery', ' BooRadleysJunctionBarandGrill', 'BacKyVietnameseCanteen', ' PhoHouse', ' DuffsFam
amatoJapaneseRestaurant', ' KanpaiSnackBar', ' EvergreenThai', ' LahoreTikkaHouse', ' Loblaws', ' PhoLinh', ' KOKOShareBar', 'Mat
ledChicken', ' KalyviaOntheDanforth', ' KatanaonBay', ' PopeyesLouisianaKitchen', ' TheDirtyBirdChickenWaffles', ' MoralsVillage'
tici', 'AjiSaiJapaneseRestaurant', 'MaidoJapaneseRestaurant', 'Ouzeri', 'The420Smokehouse', 'DonDonIzakaya', 'PattiesExpress', 'T
Restaurant', ' KhaoSanRoad', ' AromaFineIndianCuisine', ' BombayStreetFood', ' DoomiesToronto', ' CurryTwistRestaurant', ' Figo',
ya', ' 100PercentKorean', ' QinTangTaste', ' PaiNorthernThaiKitchen', 'TheCopperChimney', 'BarSybanne', 'InsomniaRestaurantLounge
aurant', 'PearlDiver', 'SchnitzelQueen', ' Buca', ' PortlandVariety', ' PainPerdu', ' BarqueSmokehouse', ' TheGabardine', ' Buddh
rant', ' KareliaKitchen', ' OsgoodeHallRestaurant', ' ShanghaiDimSum', ' LaCarnita', ' HelloDarling', ' Levetto', ' BanhMiBoys',
aurant', ' SwatowRestaurant', ' KoyoiRestaurantBar', ' BrazilianStarBarGrill', 'SushiBox', 'GabbysGrillandBar', 'LaCarnita', ' Th
andShawarma', 'Marben', 'TheSenator', 'TheKensingtonCornerstoneRestaurant', 'OntarioRestaurant', 'IslandFoods', ' TheHouseOnParli
, 'AjiSaiJapaneseRestaurant', ' TheGreekGrill', ' FiveGuys', 'BluRistorante', 'ShawarmaBoss', 'StJamesTownSteakChops', 'KhaoSanRo
itchen', 'Utsav', ' PlatitoFilipinoSoulFood', 'TheCaptainsBoil', 'EarthIndian', ' M2MAsianGroceryStore', 'BoarNWingSportsGrill',
ramountFineFoods', ' Josos', ' KawaSushi', 'TigerBBQ', 'ArepaCaf', ' TheSushiBar', 'IndianRotiHouse', ' AmsterdamBrewHouse', 'Gab
n', 'Frings', 'VannisRestaurant', 'ElCatrinDestileria', 'BuaThaiRestaurant', 'RitzCaribbeanFoods', ' EtsuRestaurant', 'HotBeans',
quet', ' TianFuChineseRestaurant', ' MuchoBurrito', ' TheCaptainsBoil', ' SushittoOntheRoad', ' TheThirstyDuck', ' MexicoLindo',
eryCafe', ' NomIzakaya', ' FortunaRistoranteLounge', 'ToneSushi', ' ComeandGetIt', 'UrbanHerbivore', ' BurritoBoyz', ' Caplansky
BurgerBar', 'NiJiSushi', 'PrayTell', ' LeeRestaurant', 'SpicyDragon', ' RotiCuisineofIndia', 'JaipurGrille', 'TheBeet', ' Vesuvio
kfastLunchDowntownToronto', 'Hibiscus', 'Sansotei', 'JerkKing', 'CicciosPizzaandPasta', 'JimmysCoffee', 'MandarinRestaurantToront
'TheBelsizePublicHouse', ' PurplePenguinCafe', ' ClubhouseSandwichShop', ' NamSandwich', 'KandaharKabab', 'FactoryGirl', 'Chinese
, ' TrattoriaNervosa', ' AngusPhoHouse', ' SaladKingRestaurant', 'EvasOriginalChimneys', 'Nandos', 'Weslodge', ' KhaoSanRoad', '
Machine', ' EastSideMarios', 'DaddyosPastaSalads', 'SpicyDragon', ' MrSouvlaki', ' BoletsBurrito', ' PopeyesLouisianaKitchen', '
```

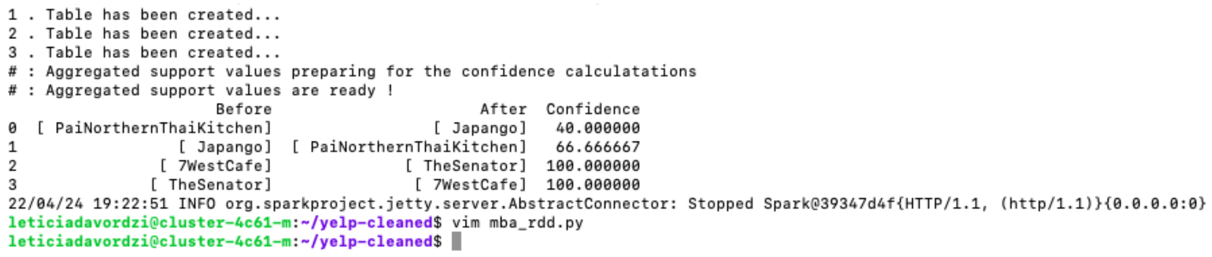
3. Adding one to all restaurant names to create a tuple.



```
[('QueenSlice', 1), (' Chadwicks', 1), (' CocoaLatte', 1), (' TappoWineBarRestaurant', 1), (' SupermodelPizza', 1), ('IrishEmbassyPubGrill', 1), (
 PetersCajunCreolePizza', 1), ('AkramsShoppe', 1), ('CrosstownCoffeeBar', 1), ('HemingwaysRestaurant', 1), ('DimmiBarTrattoria', 1), ('FreshwestGr
ps', 1), (' 7WestCafe', 2), (' KINKAIZAKAYAORIGINAL', 1), (' WowSushi', 1), (' The420Smokehouse', 1), (' CaffeDiPortici', 3), (' SummersIceCream',
rishPub', 1), (' Wvrst', 1), (' CrownDragonPub', 1), (' ClubhouseSandwichShop', 2), (' TheQueenAndBeaverPublicHouse', 1), (' DumplingHouseRestaura
ing', 1), (' HouseofGourmet', 2), ('KoreanCowboy', 1), ('TakFuSeafoodRestaurant', 1), (' HeyMeatball', 1), (' Loblaws', 2), ('CocoRiceThaiCuisine'
 ('GrazieRistorante', 1), (' NiJiSushi', 1), (' Aromaespressobar', 1), (' JohnnysShawarma', 1), (' MessiniAuthenticGyros', 1), (' KINTONRAMEN', 2)
li', 1), ('BarqueSmokehouse', 2), (' AromaEspressoBar', 1), (' OKOKDiner', 1), ('TheFry', 1), ('MillieCreperie', 1), ('TheBurgersPriest', 1), (' T
tric', 1), (' Frida', 1), ('TheCountyGeneral', 1), (' FishmanLobsterClubhouseRestaurant', 1), (' SpicyDragon', 1), (' KubKhaoThaiEatery', 3), (' M
1), ('TheCopperChimney', 2), ('FishmanLobsterClubhouseRestaurant', 2), ('TheCaribbeanQueenofPatties', 1), (' Burdock', 1), ('CafePrincess', 1), ('
, 1), (' RosewoodChineseCuisine', 1), ('DumplingHouseRestaurant', 1), ('OkonomiHouseRestaurant', 1), (' IFeelLikeCrepe', 1), ('VinnysPanini', 1),
e', 2), ('GalleryGrill', 1), (' EstWestCafe', 1), (' ThumbsUpKoreanRestaurant', 1), (' KomJugYuenRestaurant', 1), (' OldYorkBarGrill', 1), (' Sans
ill', 1), (' HanaKoreaRestaurant', 1), (' Union', 2), (' ArisPlace', 1), (' DominosPizza', 1), (' RealSportsBarGrill', 1), ('BigMoesPape', 1), ('
sa', 1), ('HanBaTang', 1), ('DonatelloRestaurant', 1), ('LisaMarie', 1), (' PoutinisHouseofPoutine', 1), (' BarrioCoreano', 1), (' ElTrompoTacoBar
rMexicana', 1), (' RizNorth', 1), (' LolasKitchen', 1), (' SipWineBar', 1), (' AuntiesUncles', 1), (' SakuraGarden', 1), ('AmayaExpress', 1), ('Su
roc', 1), ('Citta', 1), ('GlobeBistro', 1), ('PhoTienThanh', 1), (' KingTaps', 1), ('BluRistorante', 2), ('TheAce', 1), ('MotherIndia', 1), ('Mink
rmand', 1), ('AvocadoSushi', 1), ('SornThaiRestaurant', 1), ('GushiJapaneseStreetFood', 1), ('GourmetBurgerCo', 1), ('Rosewater', 1), (' ThePieCom
ne', 1), ('MadMexican', 1), ('Grillies', 1), ('GreekCo', 1), ('TakhteTavoos', 1), ('HongShingChineseRestaurant', 1), ('JerkSpot', 1), ('Wvrst', 1)
ysJunctionBarandGrill', 1), ('BacKyVietnameseCanteen', 2), (' PhoHouse', 1), (' DuffsFamousWings', 1), (' CiboWineBar', 2), (' PizzeriaLibretto',
nt', 1), (' KanpaiSnackBar', 2), (' EvergreenThai', 1), (' LahoreTikkaHouse', 1), (' KOKOShareBar', 1), ('MataPetiscoBar', 1), ('TallboysCraftBeer
icken', 1), (' KalyviaOntheDanforth', 1), (' KatanaonBay', 1), (' PopeyesLouisianaKitchen', 2), (' TheDirtyBirdChickenWaffles', 1), (' MoralsVilla
 (' AnhDaoRestaurant', 1), ('AjiSaiJapaneseRestaurant', 2), ('MaidoJapaneseRestaurant', 2), ('Ouzeri', 1), ('The420Smokehouse', 1), ('DonDonIzakay
'KhaoSanRoad', 3), ('TabuleRestaurant', 1), (' KhaoSanRoad', 2), (' AromaFineIndianCuisine', 1), (' BombayStreetFood', 1), (' DoomiesToronto', 1),
('JetsunsJuicyburger', 1), ('VannisRestaurant', 2), ('EasyRestaurant', 1), ('Katsuya', 2), (' 100PercentKorean', 1), (' QinTangTaste', 1), ('BarSy
, 2), (' JOEYEatonCentre', 1), ('SmokesPoutinerie', 1), ('PearlDiver', 2), ('SchnitzelQueen', 1), (' Buca', 1), (' PortlandVariety', 1), (' PainPe
(' BuddhasVegetarianFoods', 1), (' Konnichiwa', 1), (' ThaiLime', 1), (' Maizal', 1), (' KovalskyRestaurant', 1), (' KareliaKitchen', 1), (' Osgoo
a', 1), (' HelloDarling', 1), (' Levetto', 1), (' DrakesCornerCafBistro', 1), (' OneLoveVegetarian', 1), (' CountryStyleHungarianRestaurant', 1),
razilianStarBarGrill', 1), ('SushiBox', 1), ('GabbysGrillandBar', 2), ('LaCarnita', 2), (' TheBurgernator', 1), ('RedfishBluefishCreativeCafe', 1)
arben', 1), ('TheKensingtonCornerstoneRestaurant', 1), ('OntarioRestaurant', 1), ('IslandFoods', 1), (' TheHouseOnParliament', 1), ('UrbanHerbivor
menIsshin', 1), (' TheGreekGrill', 1), (' FiveGuys', 1), ('ShawarmaBoss', 1), ('StJamesTownSteakChops', 1), ('ChrisJerkCaribbeanBistro', 1), ('The
), ('Utsav', 1), (' PlatitoFilipinoSoulFood', 1), ('TheCaptainsBoil', 2), ('EarthIndian', 1), (' M2MAsianGroceryStore', 1), ('BoarNWingSportsGrill
hen', 1), (' TheIrvGastroPub', 1), (' ParamountFineFoods', 1), (' Josos', 1), (' KawaSushi', 1), ('TigerBBQ', 1), ('ArepaCaf', 1), (' TheSushiBar'
nBistro', 1), ('MomoSan', 1), ('Frings', 1), ('ElCatrinDestileria', 1), ('BuaThaiRestaurant', 1), ('RitzCaribbeanFoods', 1), (' EtsuRestaurant', 2
, 1), (' JumpRestaurant', 1), (' RoyaleFineDiningBanquet', 1), (' TianFuChineseRestaurant', 1), (' MuchoBurrito', 1), (' TheCaptainsBoil', 1), ('
indo', 1), ('OMRestaurantandBar', 1), ('CapocacciaTrattoria', 1), ('MichelsBakeryCafe', 1), (' FortunaRistoranteLounge', 1), ('ToneSushi', 1), ('
atessen', 1), (' LiveOrganicFoodBar', 1), (' Danji', 1), ('TheBigCarrot', 1), ('GrindhouseBurgerBar', 1), ('NiJiSushi', 1), ('PrayTell', 1), (' Le
a', 1), ('JaipurGrille', 1), ('TheBeet', 2), (' VesuvioPizzeriaSpaghettiHouse', 1), ('SocoKitchenandBar', 1), ('EastThirtySix', 1), (' EvvivaBreak
', 1), ('JerkKing', 1), ('CicciosPizzaandPasta', 1), ('JimmysCoffee', 1), ('MandarinRestaurantToronto', 1), ('DacBietBurger', 1), ('PlayaCabana',
, 1), (' PurplePenguinCafe', 1), (' NamSandwich', 1), ('KandaharKabab', 1), ('FactoryGirl', 1), ('ChineseDumplingHouse', 1), (' ElCatrinDestileria
vosa', 2), (' AngusPhoHouse', 2), (' SaladKingRestaurant', 1), ('EvasOriginalChimneys', 2), ('Nandos', 1), ('Weslodge', 1), (' EarlyBirdCoffeeKitc
', 1), ('WingMachine', 1), (' EastSideMarios', 1), ('DaddyosPastaSalads', 1), (' MrSouvlaki', 1), (' BoletsBurrito', 1), (' FitForLife', 1), (' Ch
```

4. The confidence values.

Here, we have only 4 pairs, but that can be attributed to the smaller data set used and the need to remove any combinations that only occurred once. We also notice that the first two pairs, although they have the same items, have different confidence values. This is because order matters in recommendations. Buying A and then B does not imply that a customer will buy A after buying B.



```
1 . Table has been created...
2 . Table has been created...
3 . Table has been created...
# : Aggregated support values preparing for the confidence calculatations
# : Aggregated support values are ready !
                      Before                          After  Confidence
0  [ PaiNorthernThaiKitchen]                   [ Japango]   40.000000
1                 [ Japango]  [ PaiNorthernThaiKitchen]   66.666667
2               [ 7WestCafe]               [ TheSenator]  100.000000
3              [ TheSenator]                [ 7WestCafe]  100.000000
22/04/24 19:22:51 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@39347d4f{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ vim mba_rdd.py
leticiadavordzi@cluster-4c61-m:~/yelp-cleaned$ 
```

Collaborative Filtering and ALS model:

The advantage of the recommender system:Recommendation systems can usually speed up searches, make it easier for users to access the content of interest, and bring surprises to users. Also recommendation systems can increase sales through very individual marketing and therefore user experience. So, we are going to implement the recommendation system based on content and collaborative filtering.

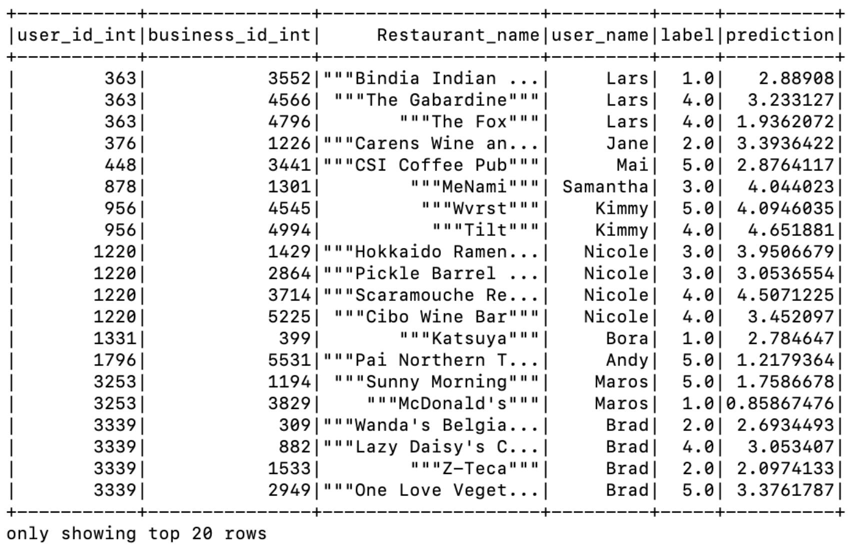
Content-based: In our system, content-based recommends new restaurants based on the similarity of a restaurant’s characteristics to a user's profile.

Collaborative filtering Recommendation system - To address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items simultaneously to provide recommendations. The idea of collaborative filtering is finding users in a community that share appreciation. If two users have the same or almost the same rated items in common, then they have similar taste. Such users are called neighborhoods. In that case, a user (A) gets recommendations for those restaurants that he/she hasn’t rated before, but was positively rated by a user (B) in his/her neighborhood.

Model -ALS - Alternating least squares (ALS) is the model we used to fit out data and find similarities. ALS is an interactive optimization process in which for every iteration, the model tries to arrive closer and closer to a factorized representation of our original data. For implicit data, the algorithm used is based on collaborative Filtering for implicit Datasets.

Using Stars for Recommendation of Toronto

Based on the above ALS model, we have the following predictions for restaurants in Toronto.



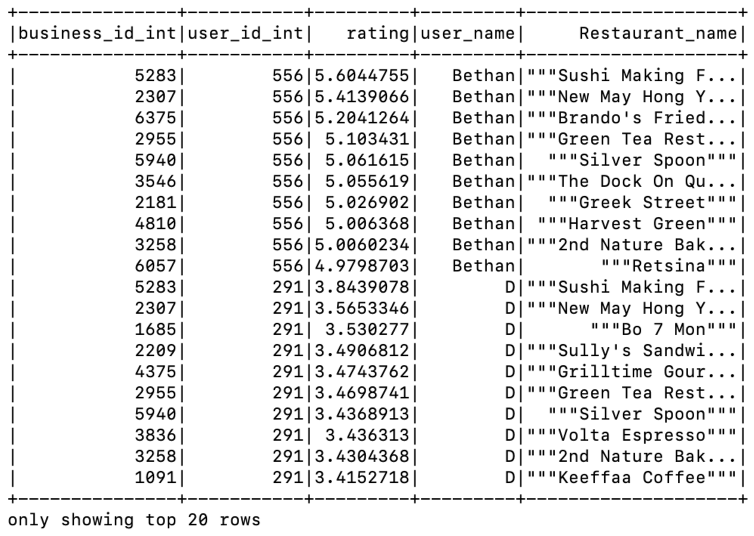
```
+-----------+---------------+--------------------+---------+-----+----------+
|user_id_int|business_id_int|     Restaurant_name|user_name|label|prediction|
+-----------+---------------+--------------------+---------+-----+----------+
|        363|           3552|"""Bindia Indian ...|     Lars|  1.0|   2.88908|
|        363|           4566| """The Gabardine"""|     Lars|  4.0|  3.233127|
|        363|           4796|       """The Fox"""|     Lars|  4.0| 1.9362072|
|        376|           1226|"""Carens Wine an...|     Jane|  2.0| 3.3936422|
|        448|           3441|"""CSI Coffee Pub"""|      Mai|  5.0| 2.8764117|
|        878|           1301|        """MeNami"""| Samantha|  3.0|  4.044023|
|        956|           4545|         """Wvrst"""|    Kimmy|  5.0| 4.0946035|
|        956|           4994|          """Tilt"""|    Kimmy|  4.0|  4.651881|
|       1220|           1429|"""Hokkaido Ramen...|   Nicole|  3.0| 3.9506679|
|       1220|           2864|"""Pickle Barrel ...|   Nicole|  3.0| 3.0536554|
|       1220|           3714|"""Scaramouche Re...|   Nicole|  4.0| 4.5071225|
|       1220|           5225| """Cibo Wine Bar"""|   Nicole|  4.0|  3.452097|
|       1331|            399|       """Katsuya"""|     Bora|  1.0|  2.784647|
|       1796|           5531|"""Pai Northern T...|     Andy|  5.0| 1.2179364|
|       3253|           1194| """Sunny Morning"""|    Maros|  5.0| 1.7586678|
|       3253|           3829|    """McDonald's"""|    Maros|  1.0|0.85867476|
|       3339|            309|"""Wanda's Belgia...|     Brad|  2.0| 2.6934493|
|       3339|            882|"""Lazy Daisy's C...|     Brad|  4.0|  3.053407|
|       3339|           1533|        """Z-Teca"""|     Brad|  2.0| 2.0974133|
|       3339|           2949|"""One Love Veget...|     Brad|  5.0| 3.3761787|
+-----------+---------------+--------------------+---------+-----+----------+
only showing top 20 rows
```

Model built using the default ALS parameters yields an average RMSE and r as:

```
Root-mean-square error = 1.2551424853217725

r2 = -0.09831557074995612
```

Visualize Recommendations

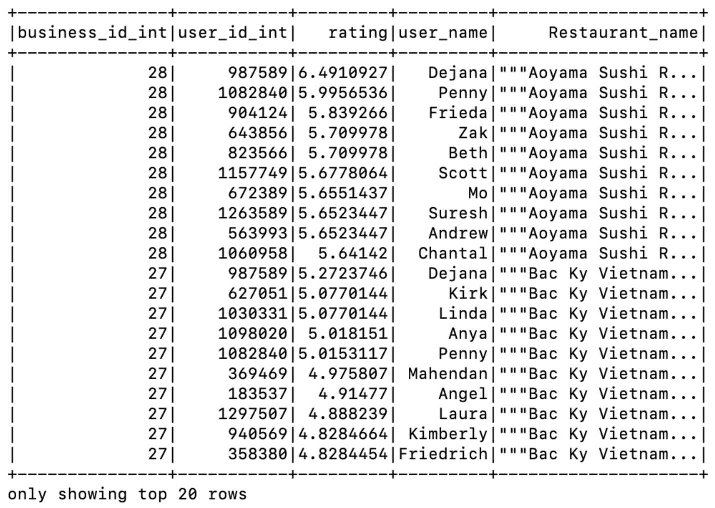
Using the designed ALS model, we can recommend 10 restaurants to each of top 10 users



```
+---------------+-----------+---------+---------+--------------------+
|business_id_int|user_id_int|   rating|user_name|     Restaurant_name|
+---------------+-----------+---------+---------+--------------------+
|           5283|        556|5.6044755|   Bethan|"""Sushi Making F...|
|           2307|        556|5.4139066|   Bethan|"""New May Hong Y...|
|           6375|        556|5.2041264|   Bethan|"""Brando's Fried...|
|           2955|        556| 5.103431|   Bethan|"""Green Tea Rest...|
|           5940|        556| 5.061615|   Bethan|  """Silver Spoon"""|
|           3546|        556| 5.055619|   Bethan|"""The Dock On Qu...|
|           2181|        556| 5.026902|   Bethan|  """Greek Street"""|
|           4810|        556| 5.006368|   Bethan| """Harvest Green"""|
|           3258|        556|5.0060234|   Bethan|"""2nd Nature Bak...|
|           6057|        556|4.9798703|   Bethan|       """Retsina"""|
|           5283|        291|3.8439078|        D|"""Sushi Making F...|
|           2307|        291|3.5653346|        D|"""New May Hong Y...|
|           1685|        291| 3.530277|        D|      """Bo 7 Mon"""|
|           2209|        291|3.4906812|        D|"""Sully's Sandwi...|
|           4375|        291|3.4743762|        D|"""Grilltime Gour...|
|           2955|        291|3.4698741|        D|"""Green Tea Rest...|
|           5940|        291|3.4368913|        D|  """Silver Spoon"""|
|           3836|        291| 3.436313|        D|"""Volta Espresso"""|
|           3258|        291|3.4304368|        D|"""2nd Nature Bak...|
|           1091|        291|3.4152718|        D|"""Keeffaa Coffee"""|
+---------------+-----------+---------+---------+--------------------+
only showing top 20 rows
```

(item-based collaborative Filtering).

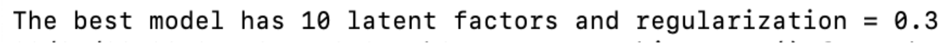
Or on the other hand, top 10 user recommendations for each of top 10 restaurants as (user-based collaborative Filtering):



```
+---------------+-----------+---------+---------+--------------------+
|business_id_int|user_id_int|   rating|user_name|     Restaurant_name|
+---------------+-----------+---------+---------+--------------------+
|             28|     987589|6.4910927|   Dejana|"""Aoyama Sushi R...|
|             28|    1082840|5.9956536|    Penny|"""Aoyama Sushi R...|
|             28|     904124| 5.839266|   Frieda|"""Aoyama Sushi R...|
|             28|     643856| 5.709978|      Zak|"""Aoyama Sushi R...|
|             28|     823566| 5.709978|     Beth|"""Aoyama Sushi R...|
|             28|    1157749|5.6778064|    Scott|"""Aoyama Sushi R...|
|             28|     672389|5.6551437|       Mo|"""Aoyama Sushi R...|
|             28|    1263589|5.6523447|   Suresh|"""Aoyama Sushi R...|
|             28|     563993|5.6523447|   Andrew|"""Aoyama Sushi R...|
|             28|    1060958|  5.64142|  Chantal|"""Aoyama Sushi R...|
|             27|     987589|5.2723746|   Dejana|"""Bac Ky Vietnam...|
|             27|     627051|5.0770144|     Kirk|"""Bac Ky Vietnam...|
|             27|    1030331|5.0770144|    Linda|"""Bac Ky Vietnam...|
|             27|    1098020| 5.018151|     Anya|"""Bac Ky Vietnam...|
|             27|    1082840|5.0153117|    Penny|"""Bac Ky Vietnam...|
|             27|     369469| 4.975807| Mahendan|"""Bac Ky Vietnam...|
|             27|     183537|  4.91477|    Angel|"""Bac Ky Vietnam...|
|             27|    1297507| 4.888239|    Laura|"""Bac Ky Vietnam...|
|             27|     940569|4.8284664| Kimberly|"""Bac Ky Vietnam...|
|             27|     358380|4.8284454|Friedrich|"""Bac Ky Vietnam...|
+---------------+-----------+---------+---------+--------------------+
only showing top 20 rows
```

Tuning ALS Parameters

For tuning the ALS model, we used parameters as maxIter = 10, regParams=[0.01, 0.3,0.8], ranks=[10,20] and the after running, the obtained result is



```
The best model has 10 latent factors and regularization = 0.3
```

Conclusions

During this project, we utilized a variety of skills taught in MSA 8050 including processing data on the cloud, RDD's, and ML pipelines. Also, we also got a real-life experience of the slowness of working with RDDs in comparison with a dataframe. However, we needed the RDDs' flexibility to build the Market Basket Analysis tool. A generous amount of data exploration was conducted through Sentiment Analysis, and ultimately, our two Recommendation Tools were built using techniques known as Market Basket Analysis and Alternating Least Squares.