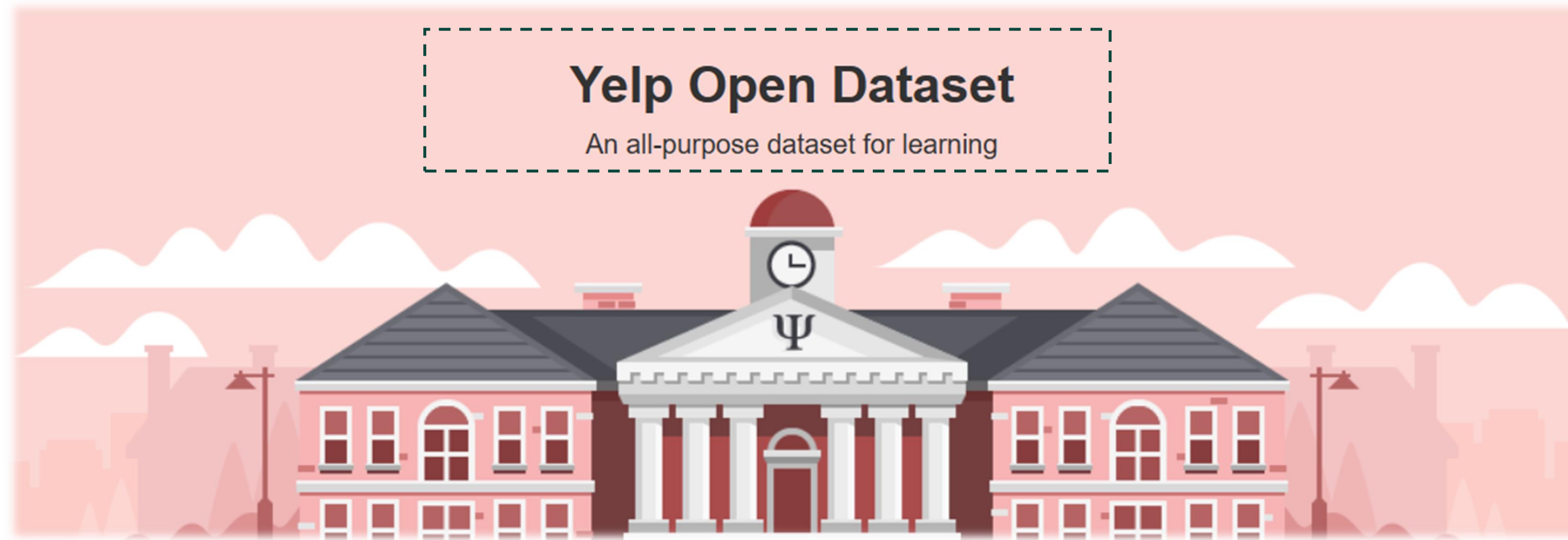


Project Mid-BDM 1024 - Data Technology Solutions



GROUP B

Adriana Marcela Penaranda Baron

Carlos Rey Pinto

Eduardo Roberto Williams Cascante

Haldo Jose Somoza Solis

Ishika Sukhija

Luis Alejandro Gutierrez Hayek

Marzieh Mohammadi Kokaneh

Nilesh Khurana

YELP Overview



- Yelp is a community-driven platform that connects people with great local businesses.
- Yelp facilitates effortless discovery, connection, and transactions between consumers and businesses in a wide array of categories, including requesting service quotes and booking restaurant tables.

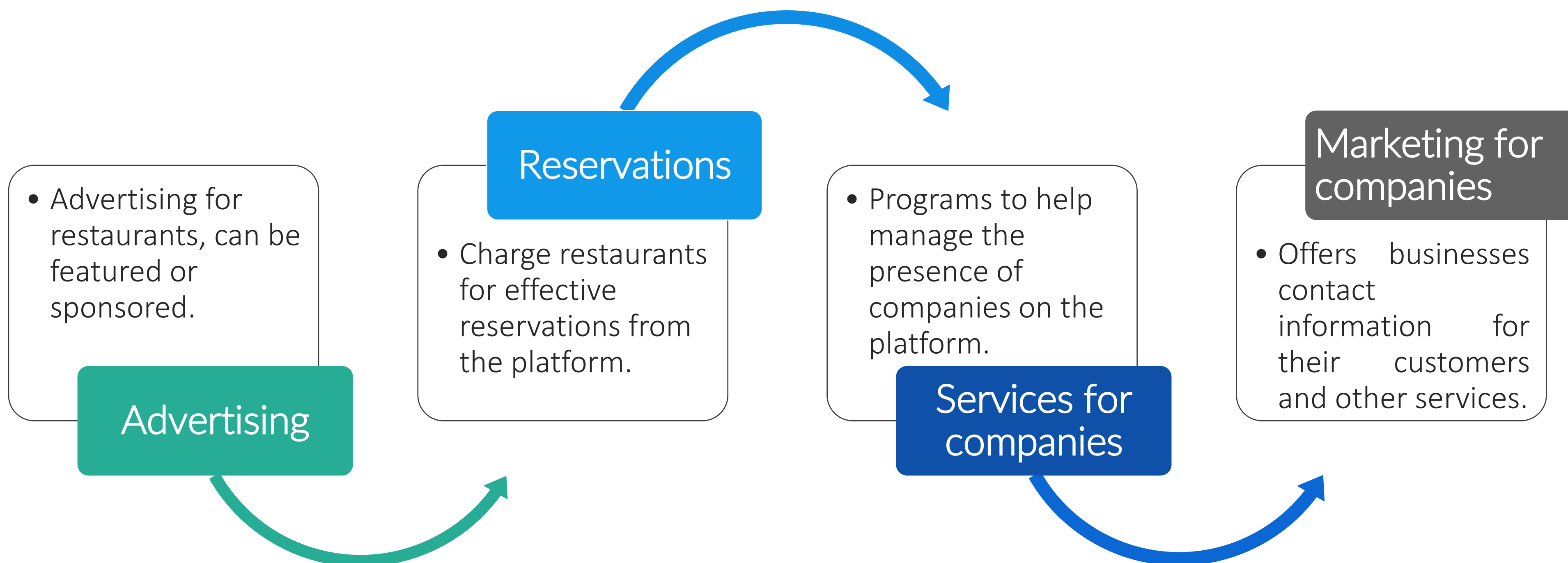
\$312M

Q1 2023 Net Revenue

\$1M

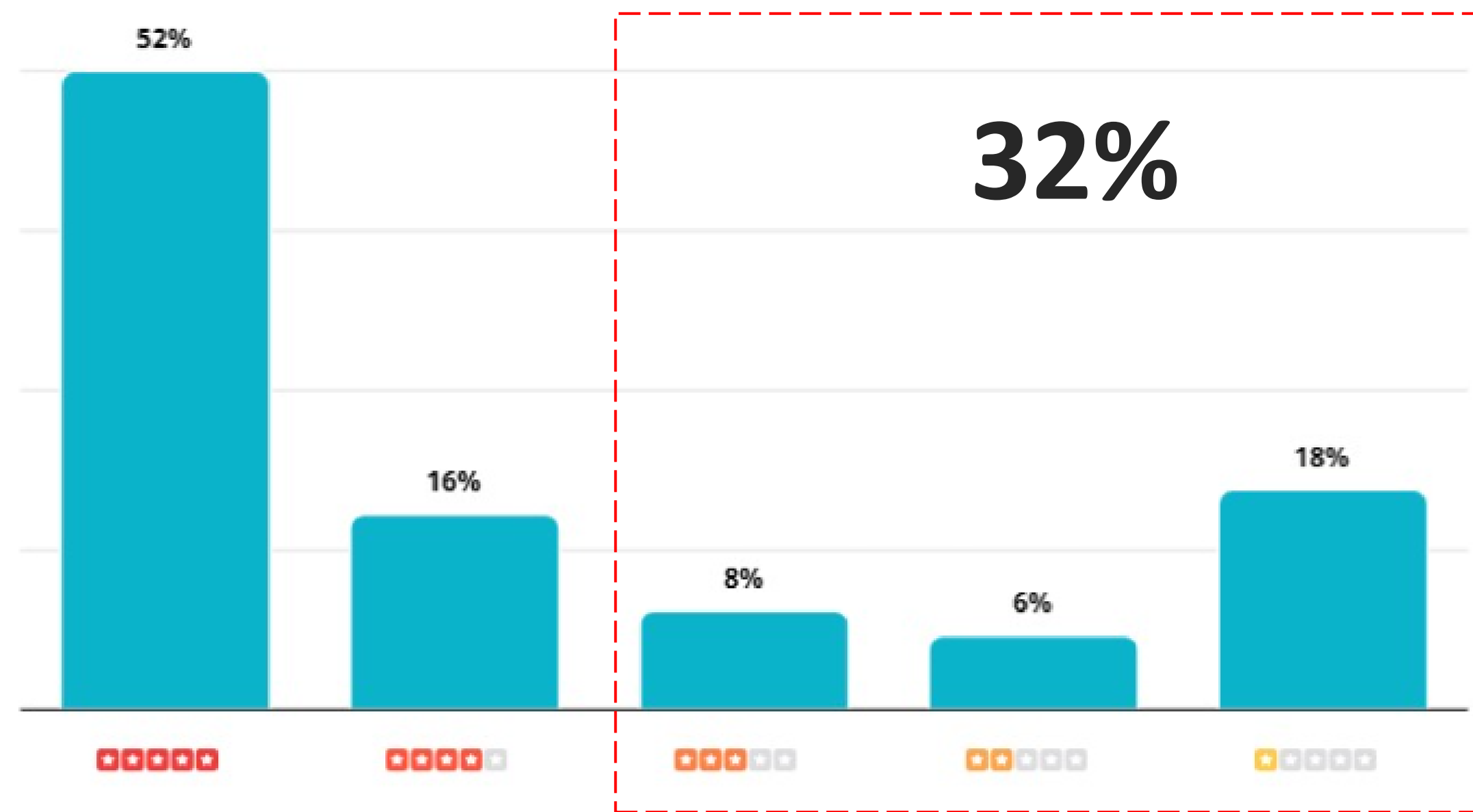
Q1 2023 Net Loss

YELP main sources of income



Opportunity the project is addressing

Distribution of Review Star Ratings



(Yeld 2023). Distribution of Review star ratings.
<https://www.yelp-press.com/company/fast-facts/default.aspx>

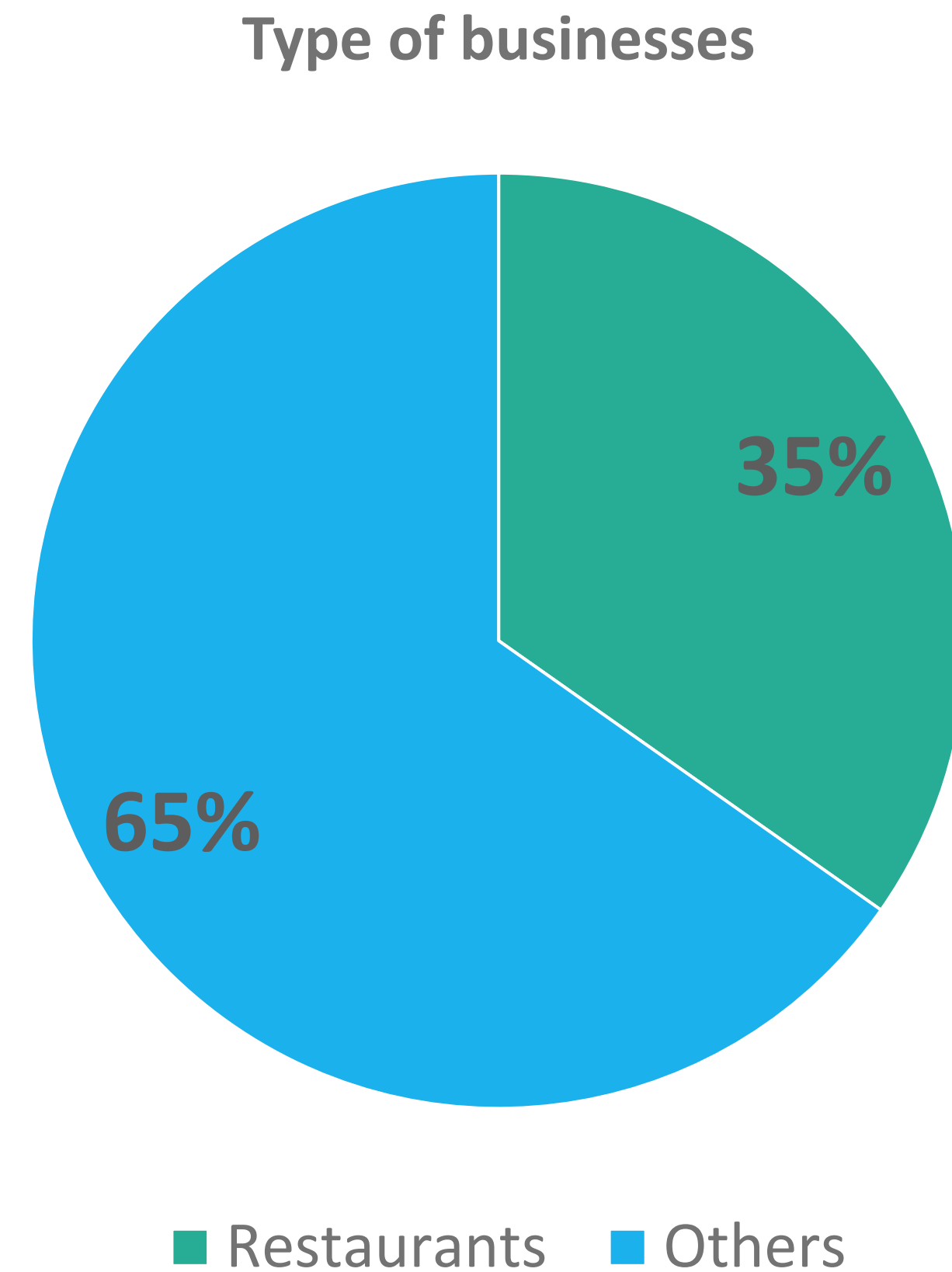
32% of businesses on the Yelp platform have a review start rating under 3, which affects their online reputation.



YELP should be concerned with improving the business rating in order to demonstrate the success of its platform and attract additional customers looking for reputation management solutions. Increasing customer ratings benefits Yelp and businesses, resulting in better revenues.

Pilot

4



Restaurants represent **35%** of the dataset, which is a significant percentage of the categories Yelp manages on its Platform



This data analytics project will offer Yelp the opportunity to focus on helping struggling businesses improve their online reputation by analysing customer feedback, identifying recurring issues, and recommending actionable strategies for enhancement.

Why Yelp should consider us as the best solution?

01

Cloud implementation

Our analytical products could offer the opportunity to be executed in the cloud, which gives Yelp analytics in a short time!



04

Competitive pricing

Our solution implemented in the CLOUD offers "Pay-as-you-go," which means you pay only for what you use, so they won't need to invest in infrastructure, which makes us competitive!



02

Integrates with Yelp's platform

We have the capacity to extract the data from their platform and add it to different analytical tools for faster and better results.



03

Uses the latest technology tools

We use great technology for this Pilot such as Python, PostgreSQL, Cloud from Azure Services, and for the next step we will use Apache Hadoop technologies.



Main Steps

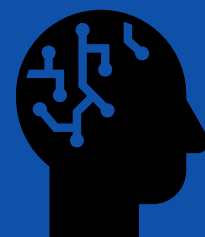
START

1 Dataset evaluation matrix

2 Design the PostgreSQL database (yelp Dataset Dictionary and Model Diagram)

3 Design a python program to transform the information from .json to PostgreSQL database

6 Issues when importing the dataset, but We finally overcame them!



5 Import the information (.json documents) to the Azure database (about 8.9M of rows migrated in 6 tables)

4 Create the PostgreSQL database and tables with relationships in the Cloud (Azure)

7 Dataset cleaning process

8 Dataset analysis

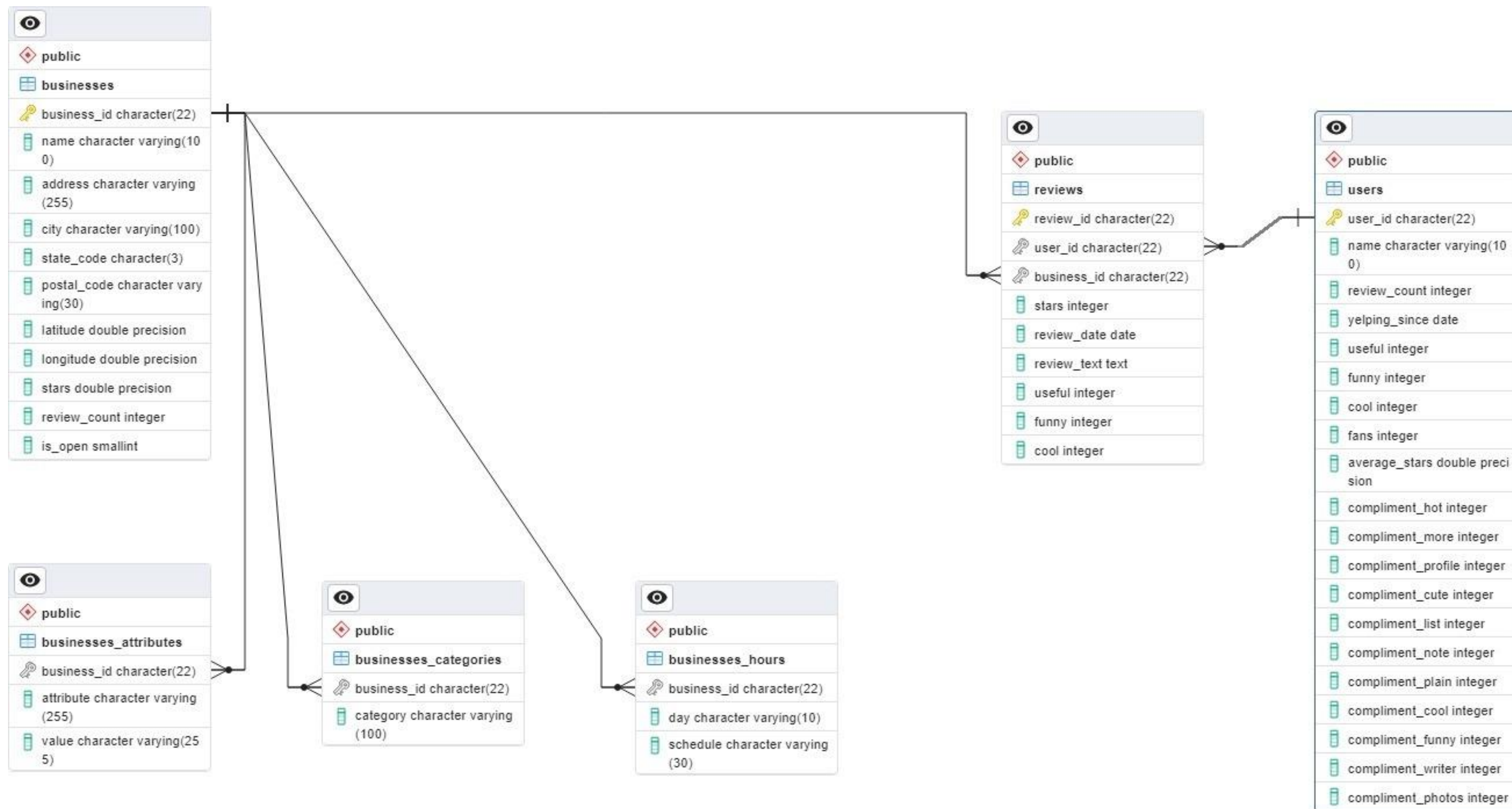
9 Insights for yelp!!



END

Data Model Diagram

Entities (11) → Relationships → Decision Tree Analysis → **ER Diagram** (6 entities)



Queries and Insights

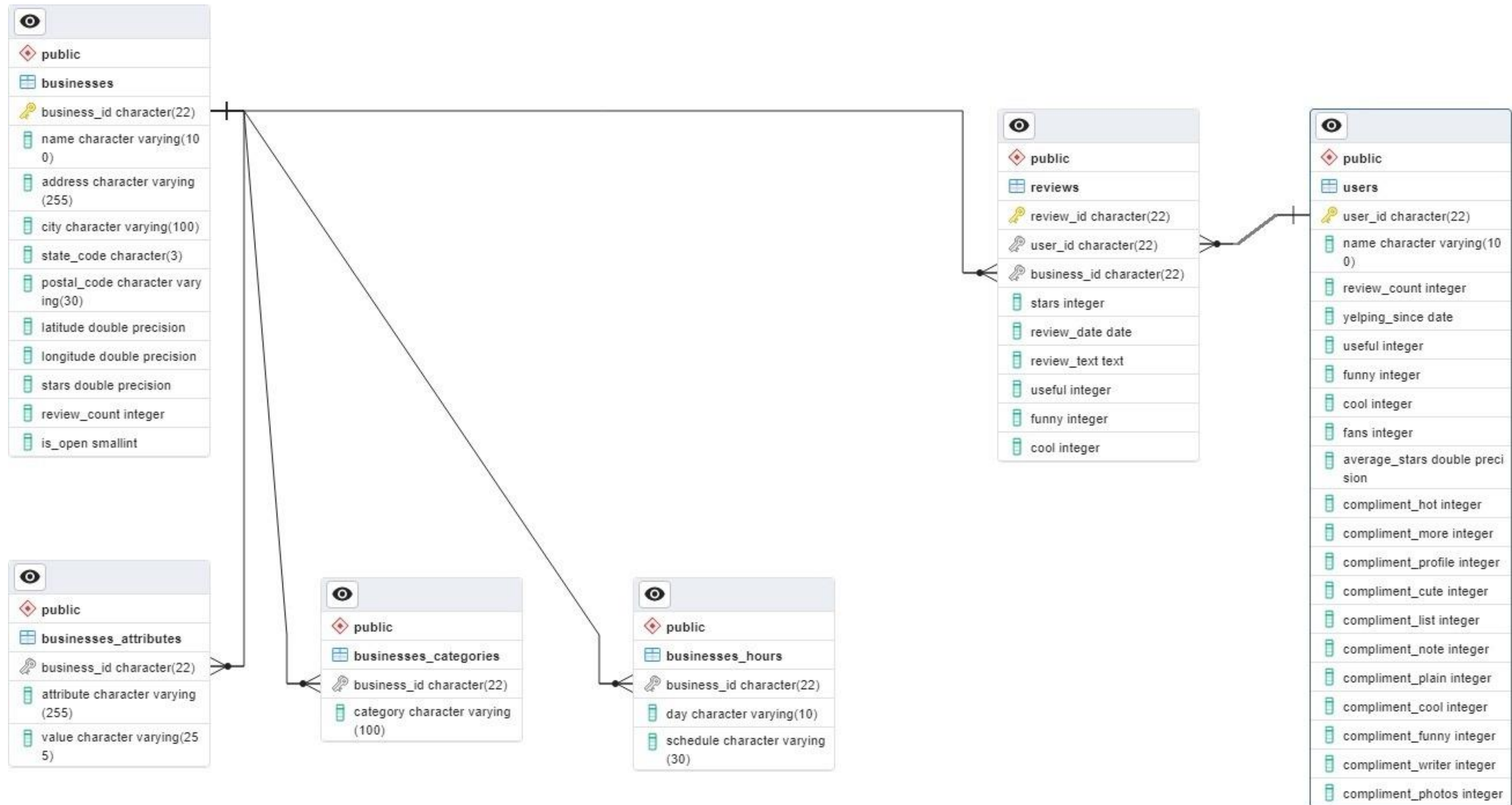
N°	Query	Output	Conclusions
1	Find the restaurants with the highest stars	1553 restaurants with a rating of 5 stars	Identify these restaurants' best practises and encourage businesses with low ratings to replicate them and enhance their reputation.
2	Identify businesses with the worst ratings (≤ 2) and analyze reviews to identify recurring issues	56,079 restaurant reviews were found with a star rating equal to or less than a score of 2 for 765 restaurants.	This output facilitates our product to focus on this 765 restaurants to identify patterns and specific areas where businesses can improve. This may include aspects such as the quality of the products or services, customer service, cleanliness, waiting time, among others.
3	Identify the number of reviews related to the issues "customer service", "clean" or "waiting time"	19,667 of 56,079 restaurant reviews that represents 35%	This result support us in identifying the major issues that users are concerned about, resulting in a negative review and rating, and Yelp needs to work on them.

Queries and Insights

N°	Query	Output	Conclusions
4	Identify the restaurants where comment issues "customer service" and has more than 10 reviews	58 restaurants	These restaurants are part of the priority of this analysis to identify what is making users give them a bad rating, and the recommendation is that Yelp prioritise executing the improvement strategies on them. The solution may include setting quality guidelines for Yelp-listed restaurants, providing additional training for customer service staff, and encouraging improved cleaning and customer service standards at partner establishments.
5	Identify the restaurants where comment issues "cleanliness" and has more than 10 reviews	45 restaurants	
6	Identify the restaurants where comment issues "waits" and has more than 10 reviews	296 restaurants	
7	Find the most active users and reward them with special promotions to build their loyalty	The query shows the first 100 most active users	After implementing the improvements in its services, YELP will be able to recommend the most active users visit these restaurants and rate them in order to improve their score and start a loyalty programme where these same people attract more people.

Data Model Diagram

Entities (11) → Relationships → Decision Tree Analysis → **ER Diagram** (6 entities)



Spark jobs

N°	Spark job	Output
1	Identify what categories are over and under the general average rate	<div>1.qty_categories.csv</div> <div>2.avg_categories.csv</div>
2	Standard variances among other the categories	<div>3.standard_deviation.csv</div>
3	Relation between cities and categories in terms of high and low ratings or extremes	<div>5.cities_categories.csv</div>

eduwil Term1 folder

CodeBlame124 lines (104 loc) · 5.15 KB

```
1  # import libraries
2  import sys
3  import pyspark as ps
4  from pyspark.sql      import SparkSession
5  from pyspark.sql.functions import *
6
7  print ("Importing pyspark libraries...OK")
8  print ()
9
10 # retrieve command line arguments and store them as variables
11 datadir  = sys.argv[1] # gs://dataproc-staging-us-central1-321442252608-e66zqwhf/yelp/data/
12 outfile  = sys.argv[2] # gs://dataproc-staging-us-central1-321442252608-e66zqwhf/yelp/results
13 print ("Retrieving command line arguments and store them as variables...OK")
14 print ()
15
16 # Defining spark/sql context
17 sqlContext = SparkSession.builder.getOrCreate()
18 print ("Defining spark/sql context...OK")
19 print ()
20
21 # loading csv files
22 df_reviews  = sqlContext.read.format('com.databricks.spark.csv').options(header = 'true', inferschema = 'true').load(datadir + 'reviews.csv')
23 df_categ_rest = sqlContext.read.format('com.databricks.spark.csv').options(header = 'true', inferschema = 'true').load(datadir + 'rest_categories.csv')
24 df_categ     = sqlContext.read.format('com.databricks.spark.csv').options(header = 'true', inferschema = 'true').load(datadir + 'categories.csv')
25 print ("Loading csv files...OK")
26 print ()
27
28 # Getting restaurant categories
29 lst_rest_cat = df_categ_rest.select("category").rdd.flatMap(lambda x: x).collect()
30 print ("Getting restaurant categories...OK")
31 print ()
32
33 # Filtering by restaurant categories
34 df_categ = df_categ.where(df_categ.cat_category.isin(lst_rest_cat))
35 print ("Filtering by restaurant categories...OK")
36 print ()
```


Map reduce jobs

Input: Spark Jobs **results** to identify what categories are over and under the general average rate.

01 Identify the keywords from reviews of the highest-rated restaurants

02 Identify the keywords from reviews of the lowest-rated restaurants

```

Eclipse-Workspace-for-Hadoop - WorldCount/src/WordCount.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer
WorldCount
  src
    (default package)
      WordCount.java
  JRE System Library [jre]
  Referenced Libraries
  Libraries
    hadoop-client-api-3.3.0.jar

*WordCount.java
1 import java.io.IOException;
12
13
14 public class WordCount {
15
16     public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
17
18         private final static IntWritable one = new IntWritable(1);
19         private Text word = new Text();
20
21     public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
22         String stringNormalized = value.toString().toUpperCase()
23             .replaceAll("[^A-Z0-9]", " ")
24             .replace(" ", " ").replace(" ", " ");
25         StringTokenizer itr = new StringTokenizer(stringNormalized);
26         while (itr.hasMoreTokens()) {
27             word.set(itr.nextToken());
28             context.write(word, one);
29         }
30     }
31 }
32
33     public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
34         private IntWritable result = new IntWritable();
35
36     public void reduce(Text key, Iterable<IntWritable> values, Context context)
37         throws IOException, InterruptedException {
38         int sum = 0;
39         for (IntWritable value: values) { sum += value.get(); }
40         result.set(sum);
41         context.write(key, result);
42     }
43 }
44
45     public static void main(String[] args) throws Exception {
46         Configuration conf = new Configuration();
47         Job job = Job.getInstance(conf, "word count");
48         job.setJarByClass(WordCount.class);
49         job.setMapperClass(TokenizerMapper.class);
50         job.setCombinerClass(IntSumReducer.class);
51         job.setReducerClass(IntSumReducer.class);
52         job.setOutputKeyClass(Text.class);
53         job.setOutputValueClass(IntWritable.class);
54         FileInputFormat.addInputPath(job, new Path(args[0]));
55         FileOutputFormat.setOutputPath(job, new Path(args[1]));
56         System.exit(job.waitForCompletion(true) ? 0 : 1);
57     }
58 }

```


Map reduce jobs

Google Cloud

hsomoza-test

Search (/) for resources, docs, products, and more

Dataproc

Jobs on Clusters

Clusters

Jobs

Workflows

Autoscaling policies

Serverless

Batches

Metastore Services

Metastore

Federation

Utilities

Component exchange

Workbench

Submit a job

Job ID *
job-ddc5f96c-WordCountInputLow

Region *
us-east1

Specifies the Cloud Dataproc regional service, which determines what clusters are available.

Cluster *
cluster-2702

Job type *
Hadoop

Main class or jar *
gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountInputLow/

The fully qualified name of a class in a provided or standard jar file, for example, com.example.wordcount, or a provided jar file to use the main class of that jar file

Jar files

Jar files are included in the CLASSPATH. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix.

Archive files

Archive files are extracted in the Spark working directory. Can be a GCS file with the gs:// prefix, an HDFS file on the cluster with the hdfs:// prefix, or a local file on the cluster with the file:// prefix. Supported file types: .jar, .tar, .tar.gz, .tgz, .zip.

Arguments
WordCount

gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountInputLow/

gs://dataproc-staging-us-east1-614478865812-rtflm40c/2023-08_Final-Project-BDM-1024/WordCountOutputLow/

Additional arguments to pass to the main class. Press Return after each argument.

Max restarts per hour

Leave blank if you don't want to allow automatic restarts on job failure. [Learn more](#)

Properties

+ ADD PROPERTY

Labels

+ ADD LABEL

SUBMIT CANCEL

EQUIVALENT REST

Google Cloud

hsomoza-test

Search (/) for resources, docs, products, and more

Cloud Storage

Buckets

Monitoring

Settings

Bucket details

dataproc-staging-us-east1-614478865812-rtflm40c

Location

Storage class

Public access

Protection

us-east1 (South Carolina)

Standard

Subject to object ACLs

None

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

Buckets

dataproc-staging-us-east1-614478865812-rtflm40c

2023-08_Final-Project-BDM-1024

WordCountOutputLow

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

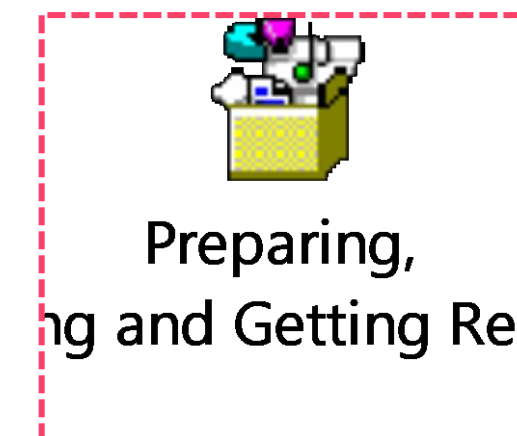
DOWNLOAD

Filter by name prefix only

Filter

Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created
<input type="checkbox"/>	_SUCCESS	0 B	application/octet-stream	Aug 16, 2023, 6:35:44 PM
<input type="checkbox"/>	part-r-00000	743.7 KB	application/octet-stream	Aug 16, 2023, 6:35:37 PM
<input type="checkbox"/>	part-r-00001	748.4 KB	application/octet-stream	Aug 16, 2023, 6:35:43 PM
<input type="checkbox"/>	part-r-00002	752.8 KB	application/octet-stream	Aug 16, 2023, 6:35:44 PM



Map reduce jobs

The screenshot shows an Excel spreadsheet with two columns: 'Column1' (words) and 'Column2' (word counts). The data is as follows:

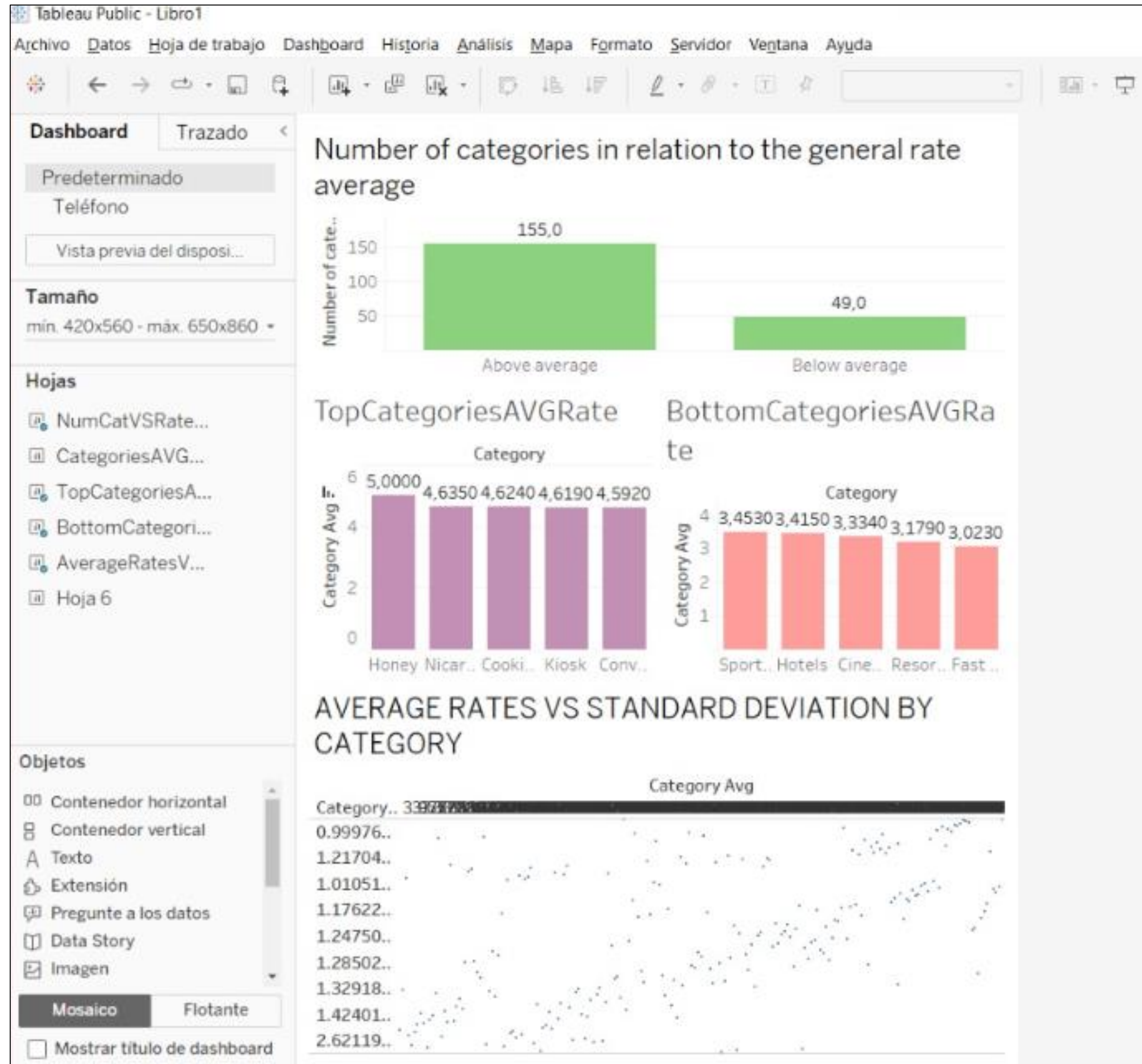
Column1	Column2
THE	14,979,940
AND	10,876,489
I	7,542,451
A	7,075,533
TO	6,147,001
WAS	6,128,502
IT	4,236,889
IS	3,697,106
OF	3,552,663
FOR	3,431,234
WE	3,188,934
IN	2,875,425
FOOD	2,616,651
THIS	2,430,202
MY	2,349,012
BUT	2,300,513
WITH	2,221,532
THEY	2,201,120
HAD	2,079,112
GOOD	1,952,639
ON	1,941,847
GREAT	1,938,764
THAT	1,932,030
WERE	1,908,635
T	1,884,233
YOU	1,856,532
NOT	1,756,340
PLACE	1,718,896

The 'Queries & Connections' pane on the right shows a query named '2023-08_Final-Project-BDM-1024_WordCountOutputLow' with 208,886 rows loaded.

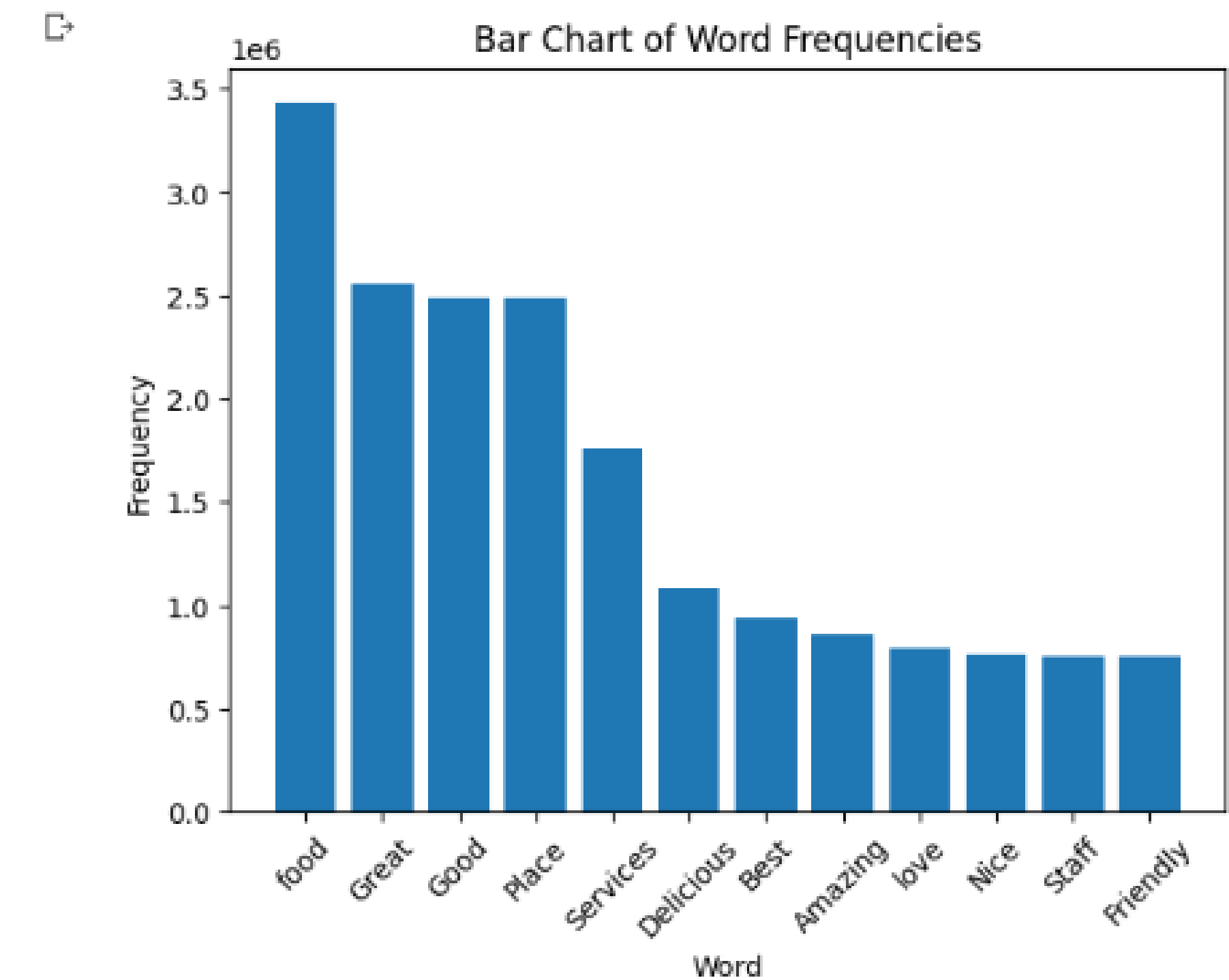
	A	B	C	D
1	Column1	Column2		
14	FOOD	3429798		
20	THEY	2685517		
21	GREAT	2552985		
22	GOOD	2484223		
27	PLACE	2372759		
35	SERVICE	1758807		
50	DELICIOUS	1087390		
56	BEST	943263		
61	AMAZING	857580		
69	LOVE	793568		
70	GOT	779726		
71	VE	768092		
72	NICE	767495		
73	BEEN	762117		
74	DEFINITELY	751551		
75	STAFF	751292		
76	FRIENDLY	751161		
77	CHICKEN	741094		
78	UP	736032		

Visualizations

Input: Spark Jobs results



Input: Map Reduce Jobs results



To conclude **our solution focus** on customer rating enhancement that creates a mutually beneficial environment, **improving Yelp's reputation and generating greater revenue opportunities for businesses and for Yelp as well!**

