Top 2020 Restaurant Performance Analysis

Team: Alekya Gadiraju, Alexander Gonzalez, Subhashree Mallick, Ahmed Mohamed

Extraction:

Our data was sourced from [Kaggle](https://www.kaggle.com/) where the restaurant is described by several variables in each row. We chose this dataset because it can be used to tell the story of what 2020 was like for restaurants, what was hot, what could be more popular soon, or what the difference is between large companies and smaller businesses. Both datasets that were used were formatted as CSV files.

Transformation:

Transformation refers to the cleansing and aggregation that may need to happen to data to prepare it for analysis. Architecturally speaking, there are two ways to approach ETL transformation:

The first type of transformation that we utilized was Multistage data transformation in which extracted data is moved to a staging area where transformations occur prior to loading the data into the warehouse. In this process, we filtered out datasets in order to select critical rows/columns. Specifically, we chose to include restaurant rank, name, sales, YOY sales, YOY units, franchising, and industry segment. In addition, we implemented format revisions in order to remove '%' from values. It should be noted we did not need to use any coding to drop any sort of null/duplicate values for our dataset, since each ranked restaurant gave a unique value for both datasets given. We did however check to see that there was no chance of any sort of duplicate/null values were included when checking the values given within the csv files.

Subsequently, we utilized In-warehouse data transformation in which data is extracted and loaded into the analytics warehouse (pgAdmin), and transformations are done there. Within pgAdmin, we created the two data tables, corrected data types for the sales, YOY sales, and YOY units from text to decimal format, and joined the independent datasets to create a table that can be used for future analysis.

Loading:

The final data table was created by loading the data to a relational production database (PostgreSQL) using rank as a joining characteristic and exhibits a direct comparison between top performing overall restaurants and smaller independent restaurants during 2020. In choosing 2020 restaurant datasets for transformation and aggregation, we wanted to be able to compare key attributes of top performing restaurants during the coronavirus pandemic which tended to generally be a period of struggle for the restaurant industry. The chosen columns can

be useful for future analysis on sales volume and restaurant success during 2020. By re-querying and representing data on franchising and industry segments as well, we would be able to also assess other details of restaurant performance in 2020.

Data Output

	Restaurant text	YOY_Sales numeric	Sales bigint	YOY_Units numeric	Restaurant text	YOY_Sales numeric	Sales bigint	YOY_Units numeric
1	Evergreens	130.5	24	116.7	McDonald's	4.9	40412	-0.5
2	Clean Juice	121.9	44	94.4	Starbucks	8.6	21380	3.0
3	Slapfish	81.0	21	90.9	Chick-fil-A	13.0	11320	5.0
4	Clean Eatz	79.7	25	58.6	Taco Bell	9.0	11293	2.7
5	Pokeworks	77.1	49	56.3	Burger King	2.7	10204	0.2
6	Playa Bowls	62.9	39	28.8	Subway	-2.0	10200	-4.0
7	The Simple Gre…	52.5	24	33.3	Wendy's	4.2	9762	0.7
8	Melt Shop	39.6	20	35.7	Dunkin'	5.0	9228	2.2
9	Creamistry	36.8	24	27.7	Domino's	6.9	7044	4.3
10	Joella's Hot Chi…	35.5	29	30.8	Panera Bread	4.0	5890	3.2
11	Eggs Up Grill	35.4	30	36.7	Pizza Hut	0.6	5558	-2.4
12	Dog Haus	34.5	39	42.9	Chipotle Mexic…	14.8	5509	5.3
13	Teriyaki Madne…	34.1	41	65.8	Sonic Drive-In	4.6	4687	-2.1
14	Bluestone Lane	33.0	48	37.1	KFC	2.5	4546	-0.2
15	Original ChopS…	32.5	21	20.0	Olive Garden	5.0	4287	1.3

Procedure:

We imported the CSV file into Python using read_csv() from pandas.

We created Pandas dataframe in Python and load data from CSV files into the Python code and then performed operations on it

From there we made an engine using the SQLAlchemy module and connected it to the databases in Python while using the Pandas module. We used PostgreSQL to connect them.

Next we used the “to_sql()” method of the dataframe object to interact with the database.

With the help of the “read_sql()” method ,we had more control over the data that we wanted to bring into the Python environment.The final result concluded with executing the table and data which had then been fetched into the Python environment.