**Technical Report**

ETL -  Yelp Restaurant Reviews in Las Vegas Metropolitan area vs Restaurant  Inspection Grade

**Prepared by**: Tom VanEyck, Maria Rotich, and Jemi Assefa

**Data Origin** :

Yelp Reviews CSV

Kaggle: <https://www.kaggle.com/jagangupta/what-s-in-a-review-yelp-ratings-eda/data> ,

Restaurant Inspection Data:

DataWorld: <https://data.world/lasvegasnevada/restaurant-inspections>

**4/15/2020**

**Project Objectives**

The report will lay out the steps taken to extract, import, clean, transform and analyze data. The objective of the project is to identify relationships between yelp restaurant reviews in Las Vegas Metropolitan area and Restaurant inspection grades from the Southern Nevada Health District Restaurant Inspections for the year 2017.

**Import**

Once we found CSV data sets, we created a jupyter file, the imported dependencies such as pandas as pd, numpy as np and from sqlalchemy import create\_engine in order to get all our libraries needed. We imported CSV files for both sets of data into jupyter and read in the data frame for further inspection. The first step after import was to only view columns that were going to be relevant and rename some of the columns that could potentially become an issue due to reserved names in SQL.

**Clean up and Transformation of data:**

We knew right from the start that we did not have clean sets of data or even a simple joinable field. The closest we had to a common column was the “name of business” and “street address”. We understood and accepted that this fact was going to keep us from using all the available rows because the way a business was listed in a yelp data set may not match the exact way it is listed in the restaurant data set. For this reason, we needed to create a new column that could be used when joining data later.

We also realized the restaurant data listed the same restaurant name several times for each inspection done within the same year, listed the same restaurant name but different address for different locations, as well as different restaurant name but same address due to the location being in the same strip mall. Because of this, simply joining on name would not work, and using street address would not work either.

Our solution was to join the restaurant name column and street address to create a composite key. We called it name\_key.

When examining the yelp data and comparing it to the layout of restaurant data, we realized there were many differences in structure such as “” around each business address and each business name. This will present a challenge in merge of columns and for that reason we stripped the “” from both columns. The address columns in both data sets also had commas (which explained the use of quotes in these comma delimited files), which would be ok if it was consistent, but since there were rows with commas and rows without, we decided to substitute a space for commas in both columns. Another issue was how the business names were listed. Some of the business names were listed in all caps, some in all small letters and some with first letter capitalized. To resolve this issue, we set both business name and street address columns to lowercase. In the next step we created a dataframe that filtered the yelp data to only show the state of Nevada.

We removed null and empty values. We tried to remove “-” from the zip code columns for consistency purposes but we were not able to figure out why the code was not working, and decided it ultimately will not create an issue or us. Once data was cleaned, we were able to merge the business name and street address columns to create the name\_key column.

Once both files had name\_key columns, we created a database connection to PgAdmin4 using sqlalchemy create\_engine. Before importing data into sql tables, we created a new database and table schemas in PgAdmin4. We decided to create the table in PgAdmin4 instead of having sqlalchemy automatically create it for us in order to have more control over the structure. We ran into the most issues in this section, such as connection issues and errors importing data into tables. The connection issues were resolved by installing psycopg2. The errors importing data came mostly from mismatch names and incorrect data types.

After importing data into PgAdmin4, we needed to filter categories data in a yelp table to only reflect “restaurants” and create new tables with that data set before joining our tables. We also still have the problem of repeated business names due to multiple inspections in the same year within the restaurant inspection data. The best way to resolve this issue, and not complicate our finding, was to simply filter inspection dates for each restaurant by the most recent inspection date.

**Results**

Description of the results from the project, amalgamated as appropriate, include graphs / photos and statistical analysis.

**Conclusion**

Conclusions should include why the results and findings are important, what they mean in **farm** terms, their utility as it may vary in different regions or environments.

 The conclusion of the report indicates why the preceding information is important. It briefly revisits the main points of the report to gesture towards future implications of the research. While the conclusion does not introduce new topics or ideas, it does consider the broader implications of the research question, discuss possibilities for future research, and restate the significance of the research findings.