Scott Barnett, Jim Comas, Eden Hailu
ETL Project Report - Breed Characteristics of NYC Licensed Dogs

Our ETL project centered around two datasets taken from Kaggle:

1. Cat and Dog Breed Parameters:
   https://www.kaggle.com/hocop1/cat-and-dog-breeds-parameters

2. NYC Dog Licensing Dataset:
   https://data.cityofnewyork.us/Health/NYC-Dog-Licensing-Dataset/nu7n-tubp

The first dataset provided a json file listing of a variety of trait ratings for over 200 dog breeds, as determined by dogtime.com. The second dataset provided a CSV file of over 340,000 records of licensed dogs in New York City. The goal of this project was to determine the most frequently-licensed dog breeds in New York City and cross-reference these figures with the traits of those breeds to determine, at a surface level, what patterns may live within the data.

Dog Ratings ETL

The Cat and Dog Breed Parameters ratings json data contained three indices: a root index, a “cat_breeds” index containing 50 listed breeds, and a “dog breeds” index containing 214 breeds. For the purpose of this project, we chose to focus on the dog_breeds index only.

We imported the json_normalize function from the pandas.io.json library to begin cleaning the data, then separated the delimited breed-trait pairs using the str.split() function. After dropping the resulting duplicates and reshaping the array to fit the number of breeds and traits, we were able to create a dataframe to visualize the data. From these 31 traits, seven were selected based on what we felt would be most relevant in determining breed preference in NYC. The list of breeds (operating as the index in this dataframe) was used as a filter for our other dataset using the isin() function, and the fully-transformed data was loaded into a local Postgres database.

NYC Dog data ETL

The NYC Dog Licensing dataset provided a CSV file that contained over 340,000 dog licensing records. Each record contained values for individual dogs’ breed name, given name, gender, and birth date. Each record also contained other information such as NYC borough and zip code for the owner.

We loaded the NYC Dog Licensing CSV into a Pandas dataframe. Noting that the boroughs column was filled with NaN, we removed it. We also filtered the dataset to only the breeds covered by the Dog Ratings set, which reduced the size of data to around 200,000 records. As we discussed the other columns in the dataset, we decided that all we really needed

was a count of dogs by breed. We dropped all columns except “BreedName” and “Zipcode”. We then grouped the data by “Breed Name” and performed a count operation. As a result, we renamed the “Zipcode” column “Amount,” with “BreedName” as the index. We then defined a SQL schema to represent this data and loaded the data into a local Postgres database.

SQL Schema, Load, and Join

We defined SQL schema for each table and created the tables in Postgres database. We loaded the tables through python pandas code in each ETL jupyter notebook. We created an SQL query to join the 2 tables by name of breed and pull the amount counts and breed characteristics data from their respective table.

Snippet of final joined table:

	BreedName character varying (50)	Amount integer	size integer	intelligence integer	energy integer	exercise_needs integer	friendly_overall integer	friendly_kids integer	friendly_strangers integer
1	Yorkshire Terrier	21921	1	3	5	4	3	2	2
2	Shih Tzu	19628	1	4	2	2	5	4	5
3	Chihuahua	15645	1	4	3	1	4	5	2
4	Maltese	11391	1	4	3	2	4	3	2
5	Labrador Retriever	11326	4	5	5	5	5	5	5
6	Pomeranian	6345	1	4	3	2	3	2	3
7	Havanese	5910	2	4	3	3	5	4	5
8	Beagle	5784	2	4	4	4	5	5	5
9	Jack Russell Terrier	5347	2	5	5	5	5	4	4
10	Golden Retriever	5184	3	5	5	5	5	5	5
11	German Shepherd Dog	5114	4	5	5	5	4	5	4
12	Cocker Spaniel	4359	2	4	4	3	4	3	3
13	French Bulldog	4258	2	3	3	2	4	4	4
14	Shiba Inu	4206	2	4	4	3	3	3	3
15	Cavalier King Charles Spaniel	3976	2	4	3	4	5	5	5
16	Pug	3945	2	2	3	3	4	4	4

Conclusions & Retrospective

While we did not do any extensive analysis of the data, it was fun to see the most and least licensed dogs in the NYC community. There one license issued in NYC over the time period of the data for a Karelian Bear Dog named Anthony. The breed name sounds like it comes out of Star Wars, but just look at [this puppy](https://www.akc.org/dog-breeds/karelian-bear-dog/) (https://www.akc.org/dog-breeds/karelian-bear-dog/). It would be interesting to test the breed data to see if particular characteristics drive dog popularity.

As a retrospective, here are some lessons learned.

Retrospective Lesson	**Description**
Integration Test	We had a good exercise with integration test. Need each module (e.g ETL notebook, sql schema) developer present to complete the integration test with quick fixes to get a working product. Difficult to do that separately.

Postgres schema with capital letters in field names	Postgres does not like SQL field names that are uppercase. It would lowercase them by default unless the SQL put quotes around the field name.
Team Interaction	These projects are not solo. We had a good discussion on the data we were using and how to transform and load that modified the project as we moved along
config.py	We leveraged a config.py to parametrize postgres login and password variables so our notebooks would not be hard coded with individual teammate login info. Allowed each of us to run the code in notebooks without changing it.

Project Repo & Files

Github Repository: https://github.com/JimCDabc/ETL_Project_2.git

File name	**Description**
dog_rating_ETL.ipynb	Jupyter notebook with extract, transform and load code for the dog rating json data for the Cats and Dog Traits dataset
NYC_Dog_ETL.ipynb	Jupyter notebook with extract, transform and load code for NYC Dog Licensing data
nyc_dog.sql	SQL code to create 2 tables with schema for dog_ratings and nyc_dog. SQL code to join the 2 tables for use in any additional analyses
data/NYC_Dog_Licensing_Dataset.csv	Source csv file for NYC Dog Licensing dataset
data/rating.json	Source json file for the Cats and Dog Characteristics dataset