Wrangle and Analyze Data : WeRateDogs

Introduction

WeRateDogs is a Twitter account that scores people's pets and includes a funny comments about the dog. The denominator of these scores is generally invariably ten. But what about the denominators? It is almost usually larger than ten. 11/10, 12/10, 13/10, etc.

In this project, we used Tweepy to query Twitter's API for data included in the WeRateDogs Twitter archive. These statistics will include the number of retweets and favorites. We create a Twitter application before executing our API querying code, after which we construct an API object that collects Twitter data. Simply put, we will gather, assess, and clean data Twitter Data then act on it through analysis, visualization, and/or modelling. We executed this project with the following steps: Gathering data; Assessing data; Cleaning data; Storing data; Analyzing, and visualizing data and finally Reporting

Gathering data;

Data was obtained from three different sources.

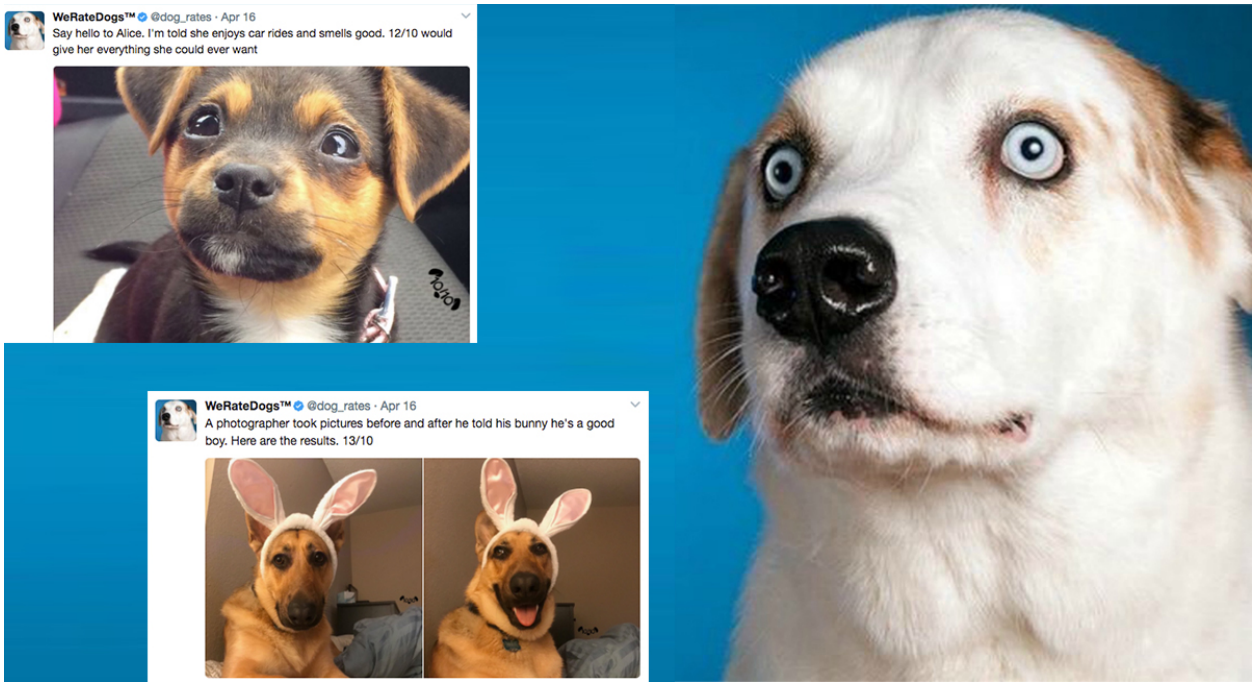
(1) The WeRateDogs Twitter archive

(2) The tweet image predictions

(3) Additional data from the Twitter API

(1) The WeRateDogs Twitter archive

WeRateDogs downloaded their Twitter archive and emailed it directly to Udacity so that we could use it for this project. This collection contains all 5000+ of their tweets' basic tweet information (tweet ID, timestamp, content, etc.) as of August 1, 2017. This file was manually downloaded as 'twitter archive enhanced.csv'. We read the data and loaded it into a pandas DataFrame.



(2) The tweet image predictions

According to a neural network, this file (image predictions.tsv) is contained in each tweet. It is hosted on Udacity's servers and may be downloaded programmatically by using the Requests library and this URL :https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

(3) Additional data from the Twitter API

We collected the number of retweets and favorites ("likes") for each tweet. Using the tweet IDs from the WeRateDogs Twitter archive, we next use Python's Tweepy package to query the Twitter API for each tweet's JSON data, and we save each tweet's whole set of JSON data in a file named tweet json.txt.

The JSON data for each tweet is provided on its own line. Then, line by line, we read this.txt file into a pandas DataFrame with the tweet ID, retweet count, and favorite count.

Assessing data

After gathering data we visually and programmatically examine them for quality and tidiness concerns. We discovered and documented certain quality and tidiness concerns in the

Quality

- Erroneous datatype assigned `timestamp` and `retweeted_status_timestamp`
- Missing values (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls, in_reply_to_status_id, in_reply_to_user_id)
- Incorrect dog names (e.g., not, a, one, the)
- "None" values in columns instead of null
- Found 181 retweets. We only want original ratings (no retweets) that have images
- * Irrelevant columns to be removed (`in_reply_to_status_id`, `in_reply_to_user_id`, `source`, `retweeted_status_id`, `retweeted_status_user_id`, `retweeted_status_timestamp`, and `expanded_urls`.)
- Inconsistent rating denominator at `rating_denominator` column. Rating should be 10. Found min 0 and max 170
- No seperate column for the most accurate dog breed prediction from the theree existing (p1, p2, and p3) predictions in the df_img_predictions (The tweet image predictions) dataset

Tidiness

- Observations in the `doggo`, `floofer`, `pupper`, and `puppo` columns of the `df_twitter_archive` dataset should be in **one** column which will be a categorical datatype
- retweet_count and favorite_count in the df_tweet (Additional data from the Twitter API) dataset should be part of the df_twitter_archive (The WeRateDogs Twitter archive) dataset.
- id column name not consistent with tweet_id as found in df_img_predictions and df_twitter_archive
- Nomalization is required to merge the tables to a master table for analysis