Analyzing and Visualizing WeRateDogs

Introduction

This is a Data Wrangling project analyzing WeRateDogs. I had 3 sections: Gathering, Assessing, and Cleaning.

Gathering

We gathered data from 3 different sources in a Jupyter Notebook titled wrangling_act.ipynb.

I downloaded an archive of WeRateDogs tweets in csv format manually -- twitter_archive_enhanced.csv. The csv was provided by Udacity.

I also downloaded an image prediction file -- image_predictions.tsv, which was hosted by Udacity's server. This was downloaded programmatically using the Request library following a url: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

Lastly, I downloaded a text file manually -- tweet_json.txt. Because I did not want to create a twitter and was on a time crunch to finish, I downloaded the file given by Udacity. Once I downloaded, I read in the first line keys which became the columns. After getting the keys, I downloaded the rest of the text file as a json object. I skipped the multi-level keys to reduce the complications.

Assessing

After assessing the dataset programmatically and visually for quality and tidiness issues I came up with 10 different quality issues and 2 tidiness issues.

Here are the quality issues:

- Remove columns not being analyzed in the archive dataframe
- Remove columns not being analyzed in the image prediction dataframe
- Remove columns not being analyzed in the tweet json dataframe
- Separate the timestamp into 3 different columns (month, day, and year) in the archive dataframe
- Remove denominators that are not 10 in the archive dataframe
- Make reasonable numerators in the archive dataframe
- Remove unoriginal tweets in the archive dataframe
- Remove duplicate JPG in the image prediction dataframe
- Consolidate the 9 confusing columns (p1, p1_dogs, p1_pred….) into 2 columns representing the predicted dog and the confidence level from the image prediction
- Get the actual text size from the text range column

Here are the tidiness issues:

- Object columns in the archive dataframe (doggo, floofer, pupper, and puppo) are 1 column spread out in 4
- Make the 3 dataframe into 1 tidy master dataset.

Cleaning

After coming up with all the quality and tidy issues, I cleaned the 3 datasets programmatically and merged them into 1 master dataset.

Here are the quality issues:

- Created a list of all columns to be removed and dropped them.
- I extracted the 3 columns from the timestamp column using string slicing.
- I used the df.query function to only keep the rows with denominators 10 and under.
- I used value_counts() to find all the values and their counts. All the numerators over 14 were barely there and some were very high in the triple digits. As a result I used the df.query to keep the rows with numerators 14 and under.
- I subsetted the archive dataframe to only include rows where retweeted was NA or .retweeted_status_id.isna()
- I used drop_duplicated and keep='last' to remove the duplicate jpg.
- Using the create_prediction I converted the 9 columns into 2 columns (pred_dog and pred_conf). In this function, I took the first accurate prediction and took that dog and confidence. If neither of the 3 were accurate, I filled in NA.
- Using the .apply function I converted the text_range into text size by subtracting the lower from the upper part of the range.

Here are the tidiness issues:

- Using df.apply I converted the 4 different columns puppo, pupper, floofer and doggo into one column.
- I merged the 3 datasets. In the process I had to make sure the names were accounted for. I merged on id for tweet_df and tweet_id for the other 2.

Following all the cleaning, I stored the dataframe into twitter_archive_master.csv.