Wrangle Report

Gathering Data

To gather data I pulled in data from 3 different data sources.

1. Link to a CSV
2. Link to TSV
3. Twitter API (tweepy)

I pulled in the CSV and TSV files using the request library paired with the IO library and was able to decode both using UTF-8.

The twitter API pulled in json format data which needed looping tweet by tweet to pull in the data to a pandas dataframe. I chose to pull in favorite count, retweet count, lang, source and followers count (ended up not using followers count). Because of tweepy rate limits I had to pass wait parameters to run the api successfully in one go.

Assess

I visually found some formatting issues with the CSV file. There were hyperlinks that did not have correct formatting and there appeared to be missing values in 5 fields in the CSV file (twitter-archive-enhanced.csv). I would end up removing these fields and removing the records that contained data in these fields.

Programmatically, I was able to dig deeper into some of the datatype formats. There were datetime issues that needed to be resolved. There were also duplicate issues that needed to be addressed. I have documented fairly well in my jupyter notebook and the rest of the assessment can be found there.

Clean

I was able to find more than the minimum cleaning requirements.

The code to perform the cleaning will be available in my Jupyter notebook. This document will outline the reasoning and thought process of why I chose to clean a particular issue.

Quality (1-5)

The fields “in_reply” and “retweeted_status” come to a total of five unique fields. These fields indicate if the tweet in question is a reply or a retweet. Because we only want to analyze top level tweets these fields should not contain any data. I did find 78 replies and 181 retweet records. I removed these records and then eventually removed the fields completely in a tidy step later on.

Quality 6

I found 55 dog names that contained just the letter “a”. I cleaned these records up by setting these names to None. While “a” can be a legitimate dog name, 55 occurrences appears to be too common and is likely a data issue.

Quality 7

The timestamp field was not in datetime format. In order to properly use this data in datetime order I needed to update the datatype to datetime. The datetime library was used.

Quality 8

There were a handful of instances where the “Expanded urls” fields were missing. I removed these records.

Tidy 1

The dog stages were broken up into multiple columns (doggo, floofler, pupper, puppo). To tidy these fields up I first checked if each tweet only contained one “dog stage “. After confirmation of unique statuses per each tweet, I proceeded to condense the four separate statuses into a single field named “dog_stage”. In the event that there was no dog stage prediction I set the dog stage to None (there were a high percentage of None).

Tidy 2

The score was broken out into two fields (numerator and denominator). I condensed the score into a percentage for reference. The numerator and denominator are still relevant but visually and analytically it will be easier to aggregate on a single score field.

Final DF

Because we are working with such a small dataset (less than 5000). I decided to merge all datafiles into a single master file. In the event that we continue and expand this analysis for future use it may make sense to keep the prediction data in one file and the tweet data (api and csv) in another.