Documentation on the Wrangling process

Introduction

This document explains the steps taken to wrangle a number of datasets given as part of the WeRateDogs wrangling exercise. The 3 steps, Gathering, Assessment and Cleaning and discussed below.

Gathering

Most of the data was already gathered such as the twitter archive dataset. The image predictions dataset needed to be downloaded, additional information from twitter was obtained using the twitter API.

The twitter archive contained 2356 messages, 2075 of those tweets IDs were present in the image_prediction dataset. I used the tweet IDs from the twitter archive to obtain information on 2356 tweets which was saved to json_tweets.txt

Assessment stage

There were 2 ways to assess the data, visually and programmatically. Both methods were used and are discussed below.

Visual Assessment

A visual assessment just involved viewing a small sample of rows to see what the actual data looked like. This is what was discovered

1. The retweets and replies columns had a lot of nulls which could be removed once removing any retweets/replies.
2. The source column let as html tag
3. Dog stages all had separate columns

Programmatic Assessment

The programmatic assessment involved

- df.info() used to see datatypes, it was found the timestamp column was a string, rather than a datetime object
- Df.duplicated() for duplicates
- Tweet ID in json_tweets.txt file were loaded as string objects so needed to convert these to int64

One a list was compiled of all the issues I could move onto the cleaning phase of the wrangling process.

Cleaning

The cleaning process was broken down into many stages, each stage included a definition, the code to clean and a test to ensure the code was cleaned. Some of the main operations in the cleaning phase are outlined here.

Columns, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp would contain a value if the tweet was a retweet and in_reply_to_status_id, in_reply_to_user_id would contain a value if it was a reply.

Replies and retweets were deleted and these columns were then removed.

The timestamp column was converted from a string to a datetime object.

The source column was cleaned as it contained information in html tags, so a pattern was used to detect text inside the tags.

Dog stages needed to be collapsed into a single column, however some entries had multiple dog stages detected from the tweet text. 11 rows were found to contain multiple dog stages so these were cleaned manually by observing the text in the tweet

Image predictions contained guesses if the tweet was a dog or not, anything that wasn't a dog was removed from the table and only the prediction with the highest confidence was kept. This data was merged with the twitter archive dataset. The tweet data pulled via the API was also merged.

Dog stages also had a lot of null values, this was because the original algorithm only looked for pupper, doggo etc. There were many variations of this in the tweet text, when adding additional dog stage names more were found, such as puppa, floof etc.

A lot of dog names were also empty, due to the original was looking for such as "this is <dog_name>" would also pick up 'an' in the sentence "this is an <dog_breed>". I'd added search patterns like "named <dog_name>" and "name is <dog_Name>" and more dog names were found this way.

Finally, A number of tweets were for incorrect submissions for pictures that were not dogs. I looked for sentences such as

'please only send', 'we only rate dogs', 'not a dog', 'only send in dogs', 'guys'

And found 16 tweets that were not for dogs, these were removed.

This completed the cleaning process, the data was then saved to csv for analysis.