Wrangle Report #weratedogs

Introduction

This report describes the wrangling process that led to the act_report. The main data source is twitter-archive-enhanced.csv file which is a comma separated file created by Weratedogs for Udacity. Weratedogs created this file by downloading their twitter archive containing basic tweet data for all 5000+ of their tweets as they stood on August 1, 2017. The data wrangling process consists of

- Data gathering
- Assessing data
- cleaning data Naturally this is an iterative process.

Data gathering

Data was gathered from three places as described below.

- WeRateDogs Twitter archive. This archive contains basic tweet data for all 5000+ of their tweets.
- Twitter api. We use the Twitter api to extract some extra data not present in the archive.
- Tweet image prediction. Udacity used a neural network able to classify breeds of dogs to classify the images from weratedogs archive.

Gathering data from the Weratedogs Twitter archive

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. This archive was downloaded manually from the following link twitter-archive-enhanced.csv. This file was read into a dataframe (df_archive) using panda's read_csv function.

A description of what the column names stand for can be found here.

Tweet image predictions

The tweet image predictions, what breed of dog is present in each tweet according to a neural network is saved in a tab separated file (image_predictions.tsv). This file is hosted on Udacity's servers at the following url: image_predictions.tsv. This file was downloaded programmatically using the Requests library. Finally the file was read into a dataframe (df_image_predictions) using pandas read_csv function.

Twitter API

Extra information was received from the Twitter API using the tweepy library. In order to use this library I had to install it first on my system. For this I used conda.

```
conda install -c conda-forge tweepy
```

To retrieve data from the Twitter API I needed the Tweet id's. This list was retrieved from the weratedogs archive and from the tweet image prediction. Naturally duplicates were removed. The results were saved into another dataframe. Finally the data was saved to the 'tweet_json.txt' file using panda's to_csv function.

A description of what the column names stand for can be found here.

Assessing data

The first step in assessing the data was a visual inspection. This was done by simply displaying the three dataframes ,go through the data visually, and writing down what was not correct.

The second step was assessing the dataframes programmatically. This was done by using the following Pandas functions.

- DataFrame.info -- returns a concise summary of a DataFrame.
- Series.nunique -- returns number of unique elements in the object.
- Series.value_counts -- returns object containing counts of unique values.

The issues found are listed below.

Tidiness issues (structural issues)

- Too much information in df_image_predictions dataframe, (tweet_id and jpg_url is what we need)
- Various stages of dogs (doggo, floofer, pupper, puppo) are in four columns.
- All dataframes should be merged into one.

Dirtiness issues (quality)

- wrong datatypes for several columns
- Tweets without images
- Dataset contains retweets
- Incorrect dog names
- Sources difficult to read
- Some tweet_ids have the same jpg_url
- Some pictures are not of dogs
- Some pictures are flagged as not a dog but are in fact of a dog.
- Fractions in the text are mistaken for rates.

Cleaning data

The following steps were taken to clean the data.
1. In the 'source' column I removed the html tags using a regular expression. The type of the column was changed into a category.
2. The 'doggo', 'floofer', 'pupper', and 'puppo' columns were merged into a 'dog_stage' column.
3. The three dataframes were merged into one.
4. Only original tweets were kept. That means that rows where retweeted_status_user_id and in_reply_to_user_id are not null wee removed.
5. After this all columns related to retweeting and replying were removed.
6. Records without images were removed.
7. I only wanted to keep predictions with the greatest confidence so all columns related to p2 and p3 were removed.
8. Multiple columns were changed to the correct datatype.
9. ratings that were not really ratings were fixed
10. Tweets without rating were deleted
11. All ratings were converted to be out of 10
12. Invalid dog names were fixed as far as possible.

Note: many tweets did not show a dog name.

Finally the cleaned data was stored in 'twitter_archive_cleaned_and_shiny.csv'