Wrangling Report

1. Overview

Project goal: wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "*Wow!*"-worthy analyses and visualizations.

WeRateDogs™ @dog_rates · Apr 16
Say hello to Alice. I'm told she enjoys car rides and smells good. 12/10 would give her everything she could ever want

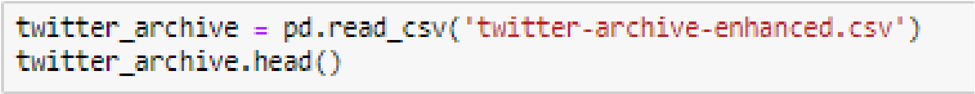
WeRateDogs™ @dog_rates · Apr 16
A photographer took pictures before and after he told his bunny he's a good boy. Here are the results. 13/10

2. Tool Used

- **Python** as a programming language
- **Jupyter Notebook** as the EDA

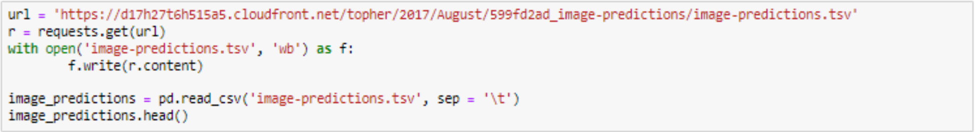
3. Gathering Data

3.1. Directly download the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)



```
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')
twitter_archive.head()
```

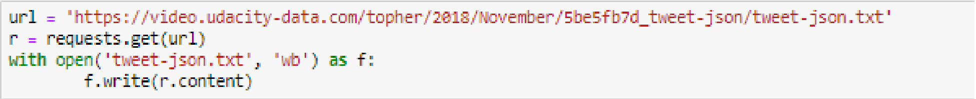
3.2. Directly download the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)



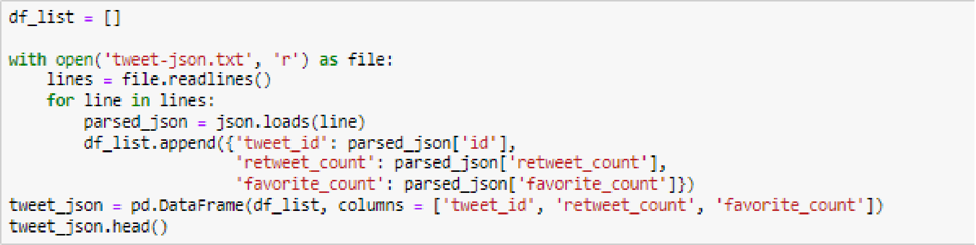
```
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
r = requests.get(url)
with open('image-predictions.tsv', 'wb') as f:
        f.write(r.content)

image_predictions = pd.read_csv('image-predictions.tsv', sep = '\t')
image_predictions.head()
```

3.3. Use the Tweepy library to query additional data via the Twitter API (tweet_json.txt)



```
url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-json.txt'
r = requests.get(url)
with open('tweet-json.txt', 'wb') as f:
        f.write(r.content)
```



```
df_list = []

with open('tweet-json.txt', 'r') as file:
    lines = file.readlines()
    for line in lines:
        parsed_json = json.loads(line)
        df_list.append({'tweet_id': parsed_json['id'],
                        'retweet_count': parsed_json['retweet_count'],
                        'favorite_count': parsed_json['favorite_count']})
tweet_json = pd.DataFrame(df_list, columns = ['tweet_id', 'retweet_count', 'favorite_count'])
tweet_json.head()
```

4. Assessing Data

4.1. Twitter Archive quality

- Data Type: "timestamp" column should be in datetime type
- Outliers: there are some values greater than 10 in "rating_numerator" and some values are not equal to 10 in the "rating_denominator"
- Unreasonable value: the 4 column "doggo", "floofer", "pupper" and "puppo" should have binary value like 0 and 1 to be the one-hot encoded column
- We should remove retweets and replies records to meet the expectation
- Remove redundant columns: source and expanded_urls

4.2. Image Prediction quality

- Duplicate Images: 1 tweet should represent 1 image but we have some tweet that represent same image
- We should keep only the most confident prediction for the dog type
- Some entries are not dog name likes kimono, window_screen
- We should merge this table to be part of the tweet archive table

4.3. Tweet Json quality

- We have actually no quality issues
- This table should be merged to be part of the tweet archive table

4.4. Summary

4.4.1. Quality Summaries:

1. Data Type: "timestamp" column should be in datetime type
2. There are some values greater than 10 in "rating_numerator"
3. There are some values are not equal to 10 in the "rating_denominator"
4. We should remove retweets and replies records to meet the expectation
5. Remove redundant columns: source and expanded_urls
6. Duplicate Images: 1 tweet should represent 1 image, but we have some tweet that represent same image
7. We should keep only the most confident prediction for the dog type
8. Some entries are not dog name likes kimono, window_screen

4.4.2. Tidiness Summaries:

1. Unreasonable value: the 4 column "doggo", "floofer", "pupper" and "puppo" should have binary value like 0 and 1 to be the one-hot encoded column

2. We should merge the image prediction and tweet json to be part of the tweet archive to extract information easily

5. Cleaning Data

5.1. Treating Quality Issue

- *Issue #1:* *"timestamp" column should be datetime*

  Define: remove +0000 and change to datetime

- *Issue #2:* *Outliers: there are some values are not equal to 10 in the "rating_denominator"*

  Define:

  - Filter rating_denominator not equal to 10 and replace them to 10 for all.
  - If the numerator of the record that containing rating_denominator not equal to 10 is exceed the denominator, normalize them to 10
  - Else If the numerator is less the denominator, normalize them exactly (for example 4/20 → 2/10 → numerator = 2)

- *Issue #3:* *Outliers: there are some values are greater than 10 in the "rating_numerator"*

  Define:

  - Filter rating_numerator greater than 10 and replace them to 10 for all

- *Issue #4:* *We should remove retweets and replies records to meet the expectation*

  Define:

  - Only keep the rows where retweeted_status_id column is NaN
  - Only keep the rows where in_reply_to_status_id column is NaN
  - Drop all the column related to retweets and replies

- *Issue #5:* *Remove redundant columns: source and expanded_urls*

Define:

- Drop the source and expanded_urls columns

- *Issue #6:* *Duplicate Images: 1 tweet should represent 1 image but we have some tweet that represent same image*

  Define:

  - Use drop_duplicates function to drop duplicates of the jpg_url

- *Issue #7:* *We should keep only the most confident prediction for the dog breeds*

  Define:

  - if p1_dog = True then store the p1_confident into the confident and store the correspond dog_breed_1 to the new column dog breed
  - else if p2_dog = True then store the p2_confident into the confident and store the correspond dog_breed_2 to the new column dog breed
  - else if p3_dog = True then store the p3_confident into the confident and store the correspond dog_breed_3 to the new column dog breed
  - else fill 0 for confident and "Unknown" for dog_breed

5.2. Treating Tidiness Issue

- *Issue #1:* *Unreasonable value: the 4 columns "doggo", "floofer", "pupper" and "puppo" should have binary value like 0 and 1 to be the one-hot encoded columns.*

  Define

  - column_list = "doggo", "floofer", "pupper" and "puppo" Loop to these columns then:
    - if value = "None" then replace to 0
    - else replace to 1

- *Issue #2:* *We should merge the image prediction and tweet json to be part of the tweet archive to extract information easily*

  Define:

  - Using merge function to left join the image prediction and tweet json to be part of the tweet archive