

WeRateDogs - Twitter Data

Introduction

Real-world data rarely comes clean. Using Python and its libraries, we will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling.

The dataset that we wrangled (and analyzed and visualized) was the tweet archive of Twitter user [@dog_rates](#), also known as [WeRateDogs](#). WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "[they're good dogs Brent](#)." WeRateDogs has over 4 million followers and has received international media coverage.

Key Points that were specified by Udacity:

Key points to keep in mind when data wrangling for this project:

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
 - Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills
-

in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.

- Cleaning includes merging individual pieces of data according to the rules of **tidy data**.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This **unique rating system** is a big part of the popularity of WeRateDogs.
- You do *not* need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

Wrangling efforts:

i) Gathering Data

- The WeRateDogs Twitter archive, was given by course instructor. The file manually downloaded by clicking the following link:
`twitter_archive_enhanced.csv`
- The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) was present in each tweet according to a neural network. The file (`image_predictions.tsv`) was hosted on Udacity's servers and was downloaded programmatically using the **Requests** library and the following URL:
`https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv`
- Each tweet's retweet count and favorite ("like") count at minimum, and any additional data we find interesting. The `tweet_json.txt` was

directly downloaded from udacity as there were some issues using twitter API.

Assessing Data for this Project

After gathering each of the above pieces of data, assessed them visually and programmatically for quality and tidiness issues. The assessed document have least eight (8) quality issues and two (2) tidiness issues in our `wrangle_act.ipynb` Jupyter Notebook.

Dataset Issues found in archive dataset :

i.QUALITY :

1. Missing values in [in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id , retweeted_status_user_id, retweeted_status_timestamp, expanded_urls] columns
2. Rating_numerator and rating_denominator had some inconsistent values in the numerator and denominator.
3. tweet id 835246439529840640 had a rating of denominator = 0
4. in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id must be integers instead of float (They had id's similar to tweet_id)
5. timestamp and retweeted_status_timestamp were object but they should be datetime.
6. The columns doggo, floofer, pupper, puppo had missing values has None instead of NaN

7. Dogs name such as - 'a', 'actually', 'all', 'an', 'by', 'getting', 'his', 'incredibly', 'infuriating', 'just', 'life', 'light', 'mad', 'my', 'not', 'officially', 'old', 'one', 'quite', 'space', 'such', 'the', 'this', 'unacceptable', 'very' were not likely.

ii.Tidiness :

1. Dog stages were found in multiple columns, They should be passed in under a single column. That reduced the dimensionality of the dataframe

Dataset Issues found in tweet dataset :

i.QUALITY :

1. Missing values in [geo, coordinates, place, contributors, possibly_sensitive, possibly_sensitive_appealable, retweeted_status, quoted_status_id, quoted_status_id_str, quoted_status, in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id , retweeted_status_user_id, extended_entities] columns

2. We needed to remove retweet and replay

ii.Tidiness :

1. User column had data in dictionaries and have several unrequired data stored, we needed followers_count separately so we could access them easily.

2. Retweets and Favorites had to be joined to the archive data table, because all the tweets information was found there.

Dataset Issues found in ip (image-predictions) dataset :

i.QUALITY :

1. Only 2075 tweet_id have images.

ii.Tidiness :

1. All the prediction outputs from different algorithms have to be joined with archive and tweet.

Cleaning Data

Cleaning few of the issues you documented while assessing. Performed these cleaning in `wrangle_act.ipynb`.

- Creating a single column for dog stages and removing “doggo”, “floofer”, “pupper” and “puppo” columns.

Creating Dog Stages column

```
In [71]: archive_clean["stages"]=archive.apply(lambda x: x.doggo if x.doggo=="doggo" else ("floofer" if x.floofer=="floofer" else ("pupper"
```

```
In [81]: archive_clean.drop(["doggo", "floofer", "pupper", "puppo"],axis=1,inplace=True)
```

```
In [83]: archive_clean[archive_clean.stages!="None"]
```

Out[83]:

	tweet_id	timestamp	rating_numerator	rating_denominator	name	stages
9	890240255349198849	2017-07-26 15:59:51 +0000	14	10	Cassie	doggo
12	889665388333682689	2017-07-25 01:55:32 +0000	13	10	None	puppo
14	889531135344209921	2017-07-24 17:02:04 +0000	13	10	Stuart	puppo
29	886366144734445568	2017-07-15 23:25:31 +0000	12	10	Roscoe	pupper
43	884162670584377345	2017-07-09 21:29:42 +0000	12	10	Yogi	doggo
...
1995	672594978741354496	2015-12-04 01:55:13 +0000	9	10	Scott	pupper
2002	672481316919734272	2015-12-03 18:23:34 +0000	12	10	Jazz	pupper
2009	672254177670729728	2015-12-03 03:21:00 +0000	11	10	Rolf	pupper
2015	672205392827572224	2015-12-03 00:07:09 +0000	9	10	Opal	pupper
2017	672160042234327040	2015-12-02 21:06:56 +0000	8	10	Bubba	pupper

380 rows × 6 columns

• Removing Reply Tweets

```
In [228]: tweet[tweet.in_reply_to_status_id.notna()==True]
```

	id	full_text	in_reply_to_status_id	user	retweet_count	favorite_count
29	886267009285017600	@NonWhiteHat @MayhewMayhem omg hello tanner yo...	8.862664e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	4	117
54	881633300179243008	@roushfenway These are good dogs but 17/10 is ...	8.816070e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	7	129
63	879674319842796034	@RealKentMurphy 14/10 confirmed	8.795538e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	10	315
112	870726314365509632	@ComplicitOwl @ShopWeRateDogs >10/10 is res...	8.707262e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	3	121
147	863427515083354112	@Jack_Septic_Eye I'd need a few more pics to p...	8.634256e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	105	2363
...
2036	671550332464455680	After 22 minutes of careful deliberation this ...	6.715449e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	229	967
2147	669684865554620416	After countless hours of research and hundreds...	6.693544e+17	{'id': 4196983835, 'id_str': '4196983835', 'na...	99	551
...

```
In [229]: tweet.drop(tweet[tweet.in_reply_to_status_id.notna()==True].index,inplace=True)
```

```
In [230]: tweet[tweet.in_reply_to_status_id.notna()==True]
```

	id	full_text	in_reply_to_status_id	user	retweet_count	favorite_count
...

• Removing Retweets

```
tweet[tweet.retweeted_status.notna()==True]
```

	created_at	id	id_str	full_text	truncated	display_text_range	entities	extended_entities
31	2017-07-15 02:45:48+00:00	886054160059072513	886054160059072512	RT @Athletics: 12/10 #BATP https://t.co/WxwJmv...	False	[0, 50]	{'hashtags': [{'text': 'BATP', 'indices': [21, ...	NaN
35	2017-07-13 01:35:06+00:00	885311592912609280	885311592912609280	RT @dog_rates: This is Lilly. She just paralle...	False	[0, 133]	{'hashtags': [], 'symbols': [], 'user_mentions...}	{'media': [{'id': 830583314243268608, 'id_str': ...
67	2017-06-26 00:13:58+00:00	879130579576475649	879130579576475648	RT @dog_rates: This is Emmy. She was adopted t...	False	[0, 140]	{'hashtags': [], 'symbols': [], 'user_mentions...}	NaN
72	2017-06-24 00:09:53+00:00	878404777348136964	878404777348136960	RT @dog_rates: Meet Shadow. In an attempt to r...	False	[0, 140]	{'hashtags': [], 'symbols': [], 'user_mentions...}	NaN
73	2017-06-23 18:17:33+00:00	878316110768087041	878316110768087040	RT @dog_rates: Meet Terrance. He's being yelle...	False	[0, 140]	{'hashtags': [], 'symbols': [], 'user_mentions...}	NaN
...
...	RT @dog_rates: This	{'hashtags': []	...

```
tweet.drop(tweet[tweet.retweeted_status.notna()==True].index,inplace=True)
```

```
tweet[tweet.retweeted_status.notna()==True]
```

	created_at	id	id_str	full_text	truncated	display_text_range	entities	extended_entities	source	in_reply_to_status_id	...	favorite_count	favorited	retweeted
...

0 rows x 31 columns

- Removal of other unnecessary columns from the datasets.
- Timestamp Datatype

Changing timestamp object to datetime

```
In [170]: archive_clean["timestamp"] = pd.to_datetime(archive_clean.timestamp)
```

- Changing unacceptable names

Changing unacceptable names to "unknown"

```
145]: cr_name=["None", 'a', 'actually', 'all', 'an', 'by', 'getting', 'his', 'incredibly', 'infuriating', 'just', 'life', 'light', 'mad',
archive_clean.name=archive_clean.name.apply(lambda x: "Unknown" if (x in cr_name) else x)

171]: archive_clean.query("name in ['a', 'actually', 'all', 'an', 'by', 'getting', 'his', 'incredibly', 'infuriating', 'just', 'life', '

171]:
tweet_id  timestamp  rating_numerator  rating_denominator  name  stages
```

- Fixing Non-Integer rating_numerator issue

Fixing Non-Integer rating_numerator Issue

```
In [185]: pattern = "(\d+\.\d+/\d+)"
merge1.full_text.str.extract(pattern, expand = True)[0].dropna()

Out[185]: 39      13.5/10
499      9.75/10
549     11.27/10
1359    11.26/10
Name: 0, dtype: object

In [186]: num = merge1.full_text.str.extract(pattern, expand = True)[0].dropna().str.split('/', n=1, expand=True)[0]

In [187]: num_index = num.index
num_values = num.values.astype("float64")

In [188]: merge1.rating_numerator = merge1.rating_numerator.astype("float64")
merge1.rating_denominator = merge1.rating_denominator.astype("float64")
merge1.loc[num_index, "rating_numerator"] = num_values
merge1.loc[num_index].rating_numerator

Out[188]: 39      13.50
499      9.75
549     11.27
1359    11.26
Name: rating_numerator, dtype: float64
```

The result dataset is a high quality and tidy pandas DataFrame that is used for Analyzing and Visualizing the data.