Report: Data Wrangling

This report describes my data wrangling efforts for WeRateDogs' Twitter data.

Gathering Data

For this project, the gathering data phase involved three datasets:

1. `archive` table: contains the information WeRateDogs sent Udacity and I downloaded directly.
2. `fav_rt` table: contains the information I gathered from Tweepy. Namely, tweet ID, favorite count and retweet count.
3. `image_prediction` table: WeRateDog's tweets were processed through a neural network that can identify dog breed. This dataset was downloaded programmatically.

Assessing Data

After gathering data from multiple methods, I moved on to assessing and cleaning data.

During the visual assessment, I checked every dataset for possible quality and tidiness issues. During the programmatic assessment, I used specific methods and functions to reveal something about the data's quality and tidiness, such as `.info()`, `notna()`, `.head()`, `.tail()`, `.sample()`, among others.

I ended up with the following list of issues:

Compiled list of issues

Quality:

`archive` **table:**

1. `source` column includes the HTML tag
2. Erroneus datatypes (`timestamp`, `in_reply_to_status_id`, `in_reply_to_user_id`, `retweeted_status_user_id`, `retweeted_status_id`, `retweeted_status_timestamp`)
3. `name` column has 109 rows with invalid, lower case words
4. `name` column has 705 rows in which a dog name wasn't listed but instead of a null value, there is a None string as an entry instead
5. In the `text` column, because of the string quotation marks, the url link breaks if one tries to click on it
6. `doggo`, `pupper`, `puppo`, `floofer`: columns have rows in which a dog stage wasn't listed but instead of a null value, there is a None string as an entry instead
7. `doggo`, `pupper`, `puppo`, `floofer`: erroneous data type, string instead of bool
8. Retweets should be dropped

`image_prediction` **table:**

1. `p1`, `p2`, `p3`: dog breeds are separated by '_' instead of spaces ' '
2. `p1`, `p2`, `p3`: some dog breed are not capitalized

3. The neural network apparently identified that some images are not from dogs

Tidiness:

1. The dataset could be tidily represented with the columns `favorite_count` and `retweet_count` in the `archive` table
2. Four variables in one column in the `archive` table (`dog_stage` , i.e., doggo, floofer, pupper, and puppo)
3. The dataset could be tidily represented joining the `image_prediction` table into the `archive` table to create one master dataset

Cleaning Data

> I started cleaning from the first two tidiness issues.

Tidiness Issue #1

I used `df.merge` to merge the tables `archive` and `fav_rt` .

Tidiness Issue #2

I created a function to extract the dog stage from each of the four dog stage columns (i.e., `doggo` , `puppo` , `pupper` , and `floofer`). Then, I use `df.apply` to apply the function and create the `dog_stage` column.

> Note: I couldn't make `pd.melt` work, since there are many rows without any classification for dog stage at all. Dropping them would have meant losing a huge chunk of the dataset that is perfectly valid for other analises. Instead, I found help at this discussion from the Udacity forum.
>
> At this point, I moved on to the quality issues. I could have acted on tidiness issue #3 at this point, but I would have ended up with a huge dataset before I could safely drop any columns.

Quality Issue #1:

I used a regular expression to leave just the text for the HTML tag in the `source` column.

Quality Issue #2:

I dealt with various columns that were in erroneous datatypes using `pd.datetime` and `pd.astype` .

Quality Issue #3:

I found the rows in which invalid, lowercase words were used in the `name` column. Then, I changed those to pd.NA.

Quality Issue #4:

First, I found the rows in which a dog name wasn't listed but instead of a null value, there is a `None` string as an entry instead. Then, I changed those to pd.NA.

Quality Issue #5:

For every entry in the `text` column, I added a trailing space. This way, if you click on the link, it won't assume the quotation mark belongs to the url and it'll open just fine.

Quality Issues #6 and #7:

While dealing with tidiness issue #2, these columns were repurposed as a single `dog_stage` column and the null values were already converted to `pd.NA`. Therefore, both of these issues were already addressed.

Quality Issue #8:

First, I used `pd.index` to get the indices for the rows representing retweets. Then, I dropped those rows. Finally, since those columns weren't useful anymore, I dropped them.

Quality Issues #9 and #10:

I used `str.replace` and `str.title` to replace the underscores ('_') with spaces (' ') and converted the first character of each word to uppercase, respectively.

Quality Issue #11:

First, I used `pd.index` to get the indices for the rows in which none of the predictions (`p1` , `p2` , `p3`) seemed to come from dogs. Then, I dropped those rows.

> At this point, the list of issues was exhausted and I went back to tidiness issue #3.

Tidiness Issue #3:

I used `df.merge` to merge the tables `archive_copy` and `image_prediction_copy` .