Report for the Wrangle and Analyze Data project at Udacity (https://www.udacity.com/)

For requirements see the link (https://review.udacity.com/#!/rubrics/1136/view)

Introductory steps:

1. The necessary libraries `pandas` , `numpy` , `requests` , `tweepy` , `json` and `matplotlib.pyplot` were imported
2. The 'magic' command `%matplotlib inline` is used.

Data Gathering

1. The file `twitter_archive_enhanced.csv` was downloaded from the link (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv) that was provided by Udacity.
2. The `image_predictions.tsv` file was downloaded programmatically using the Requests (https://2.python-requests.org//en/master/) library.
3. The `image_predictions_url` and `image-predictions.tsv` were provided by Udacity.
4. The methods and commands that were used to handle the files: `requests.get()` , `response.status_code` `with open() as` statement (https://docs.python.org/3/reference/compound_stmts.html#the-with-statement), the `OAuthHandler` from `tweepy` , the `default_timer` from `timeit` (https://docs.python.org/3/library/timeit.html), `read_csv()` , `with` , `try` and `except` .
5. The obtained `tweet_json.txt` file was read line by line using `with` , `readline()` , and the `for` loop.

Assess Data

For requirements see the link (https://review.udacity.com/#!/rubrics/1136/view).

The steps that were done: the Data Frames obtained from the files `twitter-archive-enhanced.csv` , `image_predictions.tsv` and `tweet_json.txt` were inspected using the `.head()` , `.info()` , `.value_counts()` , `.duplicated()` , `.islower()` .

Quality issues that were detected

in twitter-archive-enhanced: Only original ratings (no retweets) that have images are needed, however we have

1. replies: 78 ,
2. retweets: 181 ,
3. entries without images = entries without urls = 2356 entries - 2297 expanded_urls = 59.
4. The columns *in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id retweeted_status_user_id retweeted_status_timestamp* that are not needed.
5. The *timestamp* is object, while it should be date and time.

- The issue pointed by a referee: some ratings are wrongly extracted.

1. There are only four sources of images: *Twitter for iPhone*, *Vine - Make a Scene*, *Twitter Web Client*, and *TweetDeck*, which is not seen, because the full url is given.
2. Lower case names are not names.

in img_predictions

1. The p1, p2, and p3 column names are unclear.
2. The 543 images may be not dogs.
3. The underscore in the breed names is unnecessary.
4. The 619 breed names are lower case, but 1532-692 = 840 are upper case.

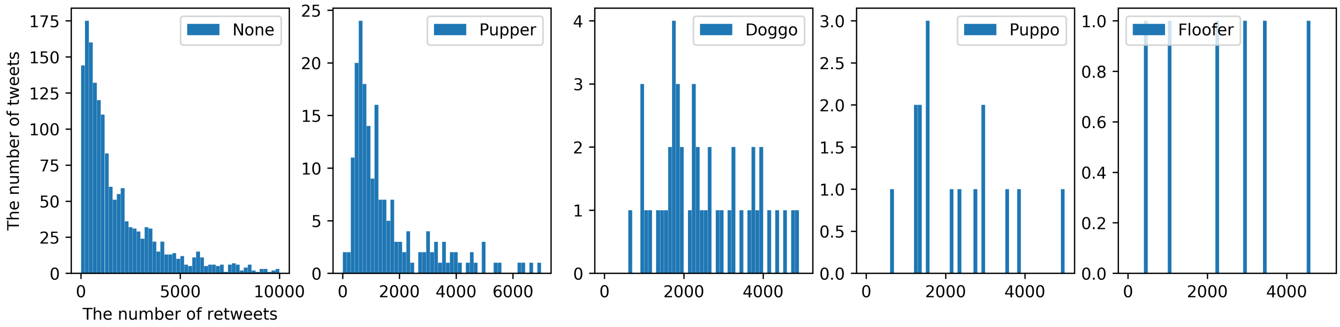
Tidiness issues

1. All the tables can be merged into the new one on 'tweet_id' index
2. The dog "stage" (i.e. doggo, floofer, pupper, and puppo) is a variable. The 'stage' should be one column.
3. The issue pointed by a referee: multiple dog stages.

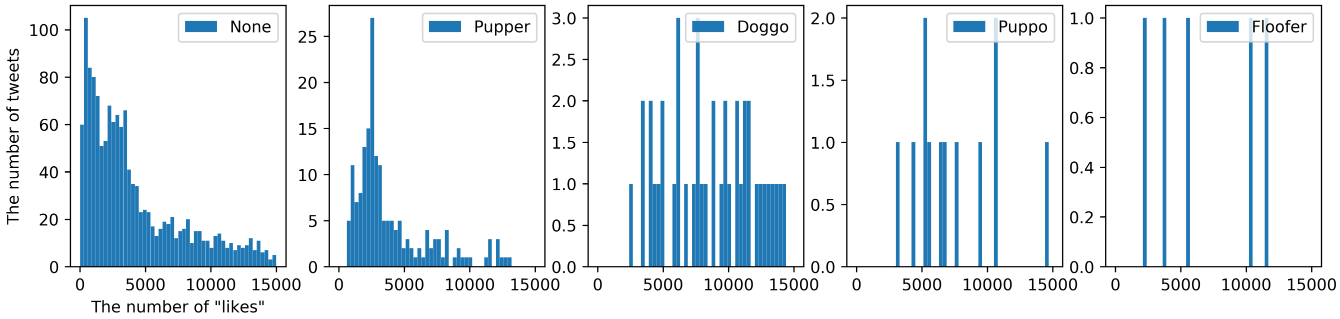
The findings:

1. The manual check of the figures shows that even with all three predictions as not dogs, there might be a dog in the picture, see, for example, the link (https://pbs.twimg.com/media/DBW35ZsVoAEWZUU.jpg). Therefore, in order to clean this issue, one needs a manual check, or a better prediction algorithm, which is beyond the scope of this project.
2. The `retweet count` and `favorite count` are correlated with each other with the correlation coefficient 0.93. However, they are practically independent from `rating numerator` and day of a week.
3. A small correlation is observed between the length of the description `text`, `p2_dog` and `p1_conf` with the `favorite_count` only. The corresponding correlation coefficient can be rounded to 0.1.
4. The most tweets are unclassified: `None` - 1661 tweets, `pupper` - 201, `doggo` - 62, `puppo` - 22, `floofer` - 7.
5. The `None` and `pupper` distributions are non-gaussian, with heavy right tale, while `doggo`, `puppo` and `floofer` distributions have too litle entries to define the shape, see figures below.
6. The favorite count distribution for `None` has two peaks. It may indicate that there is a group of unclasified images, which are much more popular than the rest.
7. The average number of `favorite_count` is several times larger than the `retweet_count` for every dog 'stage', see figures below.
8. The unclassified `None` tweets have similar popularity as `pupper`, while `doggo`, `puppo` and `floofer` are much popular in both the number of retweets and the number of favorite counts.

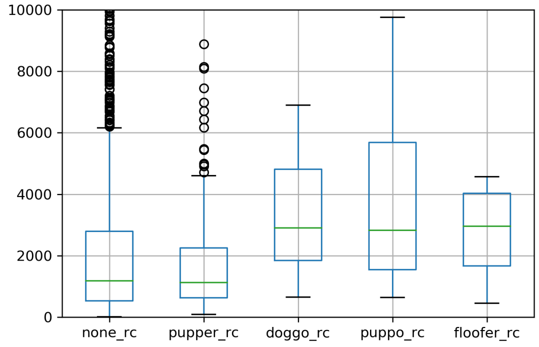
The number of tweets as the function of retweet count for different dog "stages".



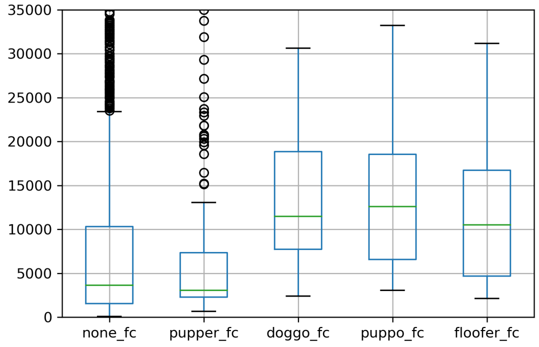
The number of tweets as the function of favorite count for different dog "stages".



The number of retweet counts for different dog "stages".



The number of favorite counts for different dog "stages".


none_fc
pupper_fc
doggo_fc
puppo_fc
floofer_fc