Udacity Project 3 : Wrangle and Analyze Data
Wrangle Report

Gathering Data :

I gathered data from 3 sources :

- The WeRateDog twitter archive file

Using pandas .read_csv() to twitter_archive_enhanced.csv that downloaded manually from udacity. (I saved it into the twitter_dog table.)

- The tweet image prediction

Using requests library from python to send requests to the server and get the response. If I get <Response [200]>, It means the request is successful and we got the image_predictions.tsv so I used pandas .read_csv() and set sep = '\t' to read tsv file. (I saved into the image_pred table.)

- The twitter API

I have some problems from creating a twitter account so I downloaded tweet_json.txt file from udacity manually but the file is a .txt file so I opened it then read each line and used json.loads() to transform data in each line to json. (I saved it into the twitter_api table.)

Assessing Data :

There are 12 Quality problems and 2 Tidiness problems

Quality - twitter_dog table

- tweeted and in_reply columns aren't necessary.
- Erroneous data types (tweet_id and timestamp columns)
- Null are represented as None in name columns.
- Some observations in the name column have incorrect names (such as one, not, a, 0, very, an etc.).
- Some dogs are not classified and represent in None (doggo, floofer, pupper and puppo columns).
- Some dogs are classified in 2 dog stages.
- Some rating fraction (both numerator and denominator) are incorrect such as rating denominator = 0.

Quality - image_pred table

- imag_num column isn't necessary.
- Erroneous data types (tweet_id)
- Duplicated row in jpg_url with different tweet_id
- Prediction sometimes lower sometimes capitalize

Tidiness

- Dog stage (doggo floofer pupper puppo and non type) should be in one column.
- twitter_dog, twitter_api and imag_pred tables should be merge together to main table on tweet_id column.

- twitter_df (master dataframe) should contain only breed types and the confident.

Quality - twitter_df

- Replace None in dog_stage column to NaN

Cleaning Data :

Quality - twitter_dog table

- First, I began with dropped tweeted and in_reply columns
- Then, used .astype('str') to tweet_id column and pd.to_datetime() to timestamp column.
- I used .replace('None',np.nan) to name column.
- I thought that I wouldn't do anything to incorrect names because they didn't carry necessary information.
- I changed 'None' in doggo floofer pupper puppo to np.nan by using .replace('None',np.nan) and created new column, None, to represented the data that doesn't classify to any stage and use 'None' as information in None column

source	text	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo	None
<a '/twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	NaN	NaN	NaN	NaN	None
<a '/twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	13	10	Tilly	NaN	NaN	NaN	NaN	None
<a '/twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	12	10	Archie	NaN	NaN	NaN	NaN	None
	This is								

- For dogs that classify into 2 dog stages, I will deal with it in the Tidiness section (using melt function and keep both).
- Then, I tried to fix incorrect data in rating_nominator and rating_denominator but I had no clue for fixing them except rating_denominator = 0 and I dropped it.

Quality - image_pred table

- I dropped the imag_num column because I thought that imag_num column doesn't carry necessary information.
- Like in twitter_dog table, I used .astype('str') to tweet_id column.
- I checked the data that have duplicated rows in jpg_url but difference in tweet_id by checked with another table that tweet_id are correct. If the tweet_id that has the same jpg_url are available in the other two tables, I will keep it and Yes, I will keep it.
- I used .str.lower() to p1, p2, p3 columns to make the word lower.

Tidiness

- I used melt function to twitter_dog table to get a new column, dog stage, that contains the dog stage type of each dog and None for dogs that do not classify.
- I merge 3 tables together on tweet_id and defined as twitter_df

Quality - image_pred table

- I used .replace('None',np.nan) to change dogs that do not classify to NaN.

Tidiness - twitter_df table

- Breed types is from the prediction that have top confident and it's a dog, confident should be top confident.