Wrangle Report :
WeRateDogs Twitter Archive

Gathering Data:

---

1. Manually download the *twitter archive enchanced csv* file from Udacity. I create the dataframe named *twit_arch*.
2. Query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Read this file and create a dataframe with tweet ID, retweet count, and favorite count named *df_cleanjson*.
3. Systemically download *Image_prediction* file and saved the dataframe named *df_image.*

Assessing Data & Cleaning Data:

I assess those three dataframe visually and programmatically for quality and tidiness issues.

1. **twit_arch dataframe**

   <u>Quality</u>

- Timestamp should be changed from object datatype into datetime datatype
- Removed the row that in_reply_to_status_id != NaN because that row just retweet the original content.
- Due to text column, there are some rating_numerator should be float such as 75 should be 9.75. I will change rating_numerator datatype from int into float.
- Some rating_numerator and rating_denominator are invalids (i.e. date, 7/11, etc.). I will extract the second xx/xx pattern to rating_numerator and rating_denominator.
- There are too various rating_denominator. I will create the score out of ten column by using rating_numerator divided by rating_denominator and multiply by ten
- Removed duplicated expanded_urls data
- Cleansing the data in source column to be more readable

- There are some incorrect name such as 'a', 'an', 'the', etc. I will use string patterns like named * , and name is * to find the real dog name. (The pattern This is*, and , and Meet * is already used to detect name in this dataframe). If the text does not contain dog names, I will change invalid names into ‘None’.
- Remove the column will not be used for analyzing : in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, rating_numerator, rating_denominator

Tidiness

- Merge all dog stages (doggo, floofer, pupper, puppo) into one column called ‘dog_stages’ and make a ‘multiple’ stage value for those tweet_id which contains more than 1 stage.

2. **df_cleanjson dataframe**

Quality

- Rename id into tweet_id

Tidiness

- Merge df_arch_copy and df_cleanjson_copy and create a new dataframe named *combine*

3. **df_image dataframe**

Quality

- Delete the duplicate jpg_url
- Because I will use only p1 for analysis, I will remove unnecessary columns: img_num,p1_conf,p1_dog,p2, p2_conf, p2_dog, p3, p3_conf ,p3_dog
- Make all dog breeds prediction of p1 into capitalize name
- Rename p1 column to predict_dogbreed

Tidiness

- Merge combine and df_image_copy table and create a new dataframe named *full_combine*

Storing Data:

After merge all three tables name full_combine, I save it into *twitter_archive_master.csv* and I will use it for data analysis and visualization.