Data Wrangling Report

The project is about data wrangling(gather, asses, clean) and analyzing the dataset archive of **WeRateDogs** twitter account (WeRateDogs rates people's dogs with a humorous comment)

Data Gathering

Overview of the Datasets

- Dataset1 : This archive consists of 2356 basic tweet data from November 2015 to August 2017.
- Dataset2 : Udacity created a new dataset which consists of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction.
- Dataset3 : Dataset created to get the count of retweets and likes on a tweet.

Gathering Sources

- Dataset1 : Using the link provided by Udacity, manually downloaded the WeRateDogs Twitter archive twitter_archive_enhanced.csv file and further imported this file into a dataframe
- Dataset2 : Programmatically downloaded the tweet image predictions file hosted on Udacity's servers using Python's Requests library and saved it locally to image_predictions.tsv file.
- Dataset3 : Gathered data from Twitter API using the tweet IDs from the Datasset1, accessed the tweet data and stored the entire set of JSON data in a txt file called tweet_json.txt file.

Data Assessing

This step includes assessing data based on visual and programmatic approaches. Assessment can be divided into two major parts :

- **Data Quality**(dirty data): Low quality data has content issues(missing data, incomplete, inaccurate, inconsistent data).
- **Data Tidiness**(messy data): Untidy data has structural issues. Few characteristics of tidy data are as below:
  - Each variable forms a column
  - Each observation forms a row
  - Each type of observational unit forms a table

Quality Issues

1. df_twitter_archive -> twitter_archive_enhanced.csv (Visual)
   - expanded URL is unnecessary as same information can be extracted from text
   - the name column contains wrong names like "None", "a", "the", "an"
2. df_image_pred -> image_predictions.tsv (Visual)
   - Redundant data in `p.._dog` column, should be melted
   - Values in p1, p2, p3 columns are not generalized, there is random use of -, _, lowercase and uppercase
3. df_twitter_archive -> twitter_archive_enhanced.csv (Programmatic)
   - tweet_id should be string

  - timestamp - columns should be datetime objects
  - Contains retweets
  - Low ratings are because of either no dog picture is there, or the picture is plagiarized as it already had been rated by the account
  - Some tweets have multiple patterns of rating format, like one of the tweet read “3 1/2 legged dog”, and was interpreted as rating 1/2, rather it should be 9/10(specified later in the same tweet)
  - Some photos contain more than one dogs, therefore they have high rating. These ratings can be generalized as per one dog

4. df_image_pred -> image_predictions.tsv (Programmatic)
   - tweet_id datatype should be string(object)
   - Only 2075 unique tweet_ids, less than df_twiiter_archive(2365)
5. df_tweet -> tweet_json.txt (Programmatic)
   - tweet_id datatype should be string

Tidiness Issues:

1. df_image_pred -> image_predictions.tsv
   - the prediction column should be melted into one column
2. df_tweet -> tweet_json.txt
   - `tweet_id`column from all three datasets should be merged
3. df_twitter_archive -> twitter_archive_enhanced.csv
   - `doggo`, `floofer`, `pupper` and `puppo` columns contain redundant information, these can be converted into a single column

Data Cleaning:

After assessing the data, we found some issues that need to be fixed. This section take care of that. Below are the steps followed to clean the data:

- Merge the tables
- Remove the replies and retweets, drop unnecessary columns [columns with which we are not concerned right now]
- Change the datatypes of the columns
- Clean the numerators/denominator rating - the ones with multiple occurrence of the pattern or misinterpreted
- Drop the expanded URL column
- Some denominator ratings are greater than 10. These denom/numerator ratings can be generalized as per one dog
- Remove the "None" out of the doggo, floofer, pupper and puppo column and merge them into one column
- Remove the wrong names of name column
- Reduce the prediction columns
- Clean the newly created column by generalizing the text pattern