WRANGLE REPORT

1. INTRODUCTION

Real-world data rarely comes clean. Using Python and its libraries, I will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling.

The dataset that I will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. WeRateDogs has over 4 million followers and has received international media coverage.

Project Details

i. Gathering data
ii. Assessing data
iii. Cleaning data

2. GATHERING DATA

A. The WeRateDogs Twitter archive

The twitter_archive_enhanced.csv file should be downloaded manually.

B. The tweet image predictions

i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (image_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and a given URL.

C. Twitter API & JSON

Each tweet's retweet count and favorite ("like") count at minimum, and any additional data that is interesting. Using the tweet IDs in the WeRateDogs Twitter archive, 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. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

3. ASSESSING DATA

After gathering each of the above pieces of data, assess them visually by printing the three dataframes separated, and programmatically by using different methods (e.g. info, describe, value_counts, duplicated).

Quality Issues

Issues with content: Completeness, Validity, Accuracy, Consistency.

Twitter_archive_data

- Convert the data-type of the 'timestamp' column to a DateTime type.
- Some missing values are represented as None instead of NaN, hence they are not counted as missing values.
- Keep only original tweets, remove retweeted tweets.
- Drop duplicated rows in the 'expanded_urls' column.
- The 'rating_numerator' and 'rating_denominator' columns should be of type float.
- The 'rating_numerator' should be correctly extracted.
- The 'name' column has invalid names, they can be easily identified as they all start with lower case letters.
- Drop Unneeded columns.

Image_prediction_data

- Drop duplicated rows in the 'jpg_url' column.
- Rename the columns img_num, p1, p2, p3, p1_conf, p2_conf, p3_conf, p1_dog, p2_dog and p3_dog.

Tweet_json_data

- Rename the 'id' column to 'tweet_id'.
- Keep only original tweets, remove retweets from the rating.
- Drop Unneeded columns.

Tidiness

Issues with structure: Untidy data.

- In Twitter_archive_data, the doggo, floofer, pupper and puppo columns will be merged in one column called 'dog_stage'
- Merge all the dataframes in one dataframe.

4. CLEANING DATA

The issues that was found during the assessment process were cleaned and tested using the technique: Define, code and test, and the methods (e.g. copy, drop, replace, rename,

value_counts, info, drop_duplicates, isnull, fillna, merge), after that the three dataframes were merged and stored in a CSV file named twitter_archive_master.csv.