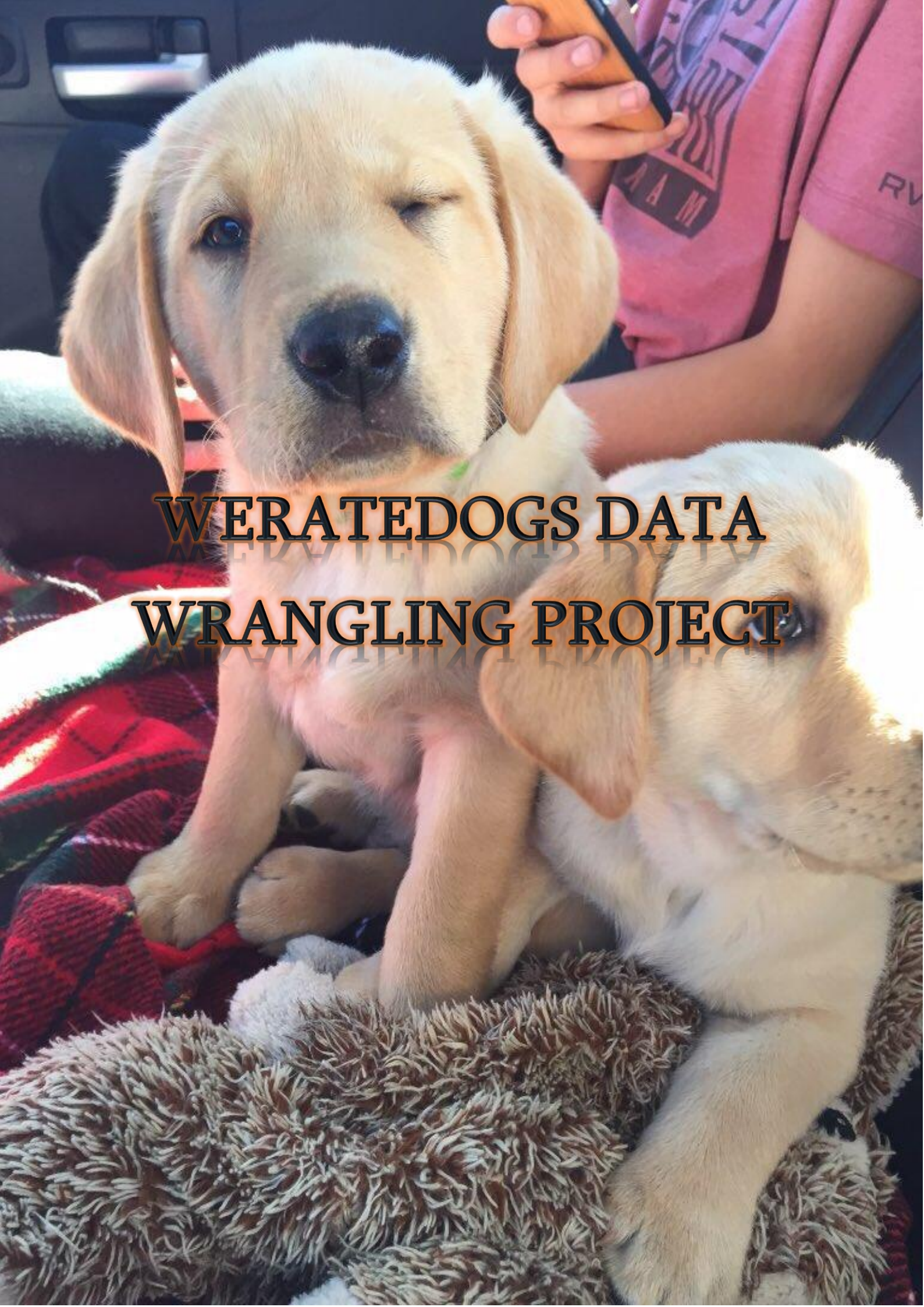
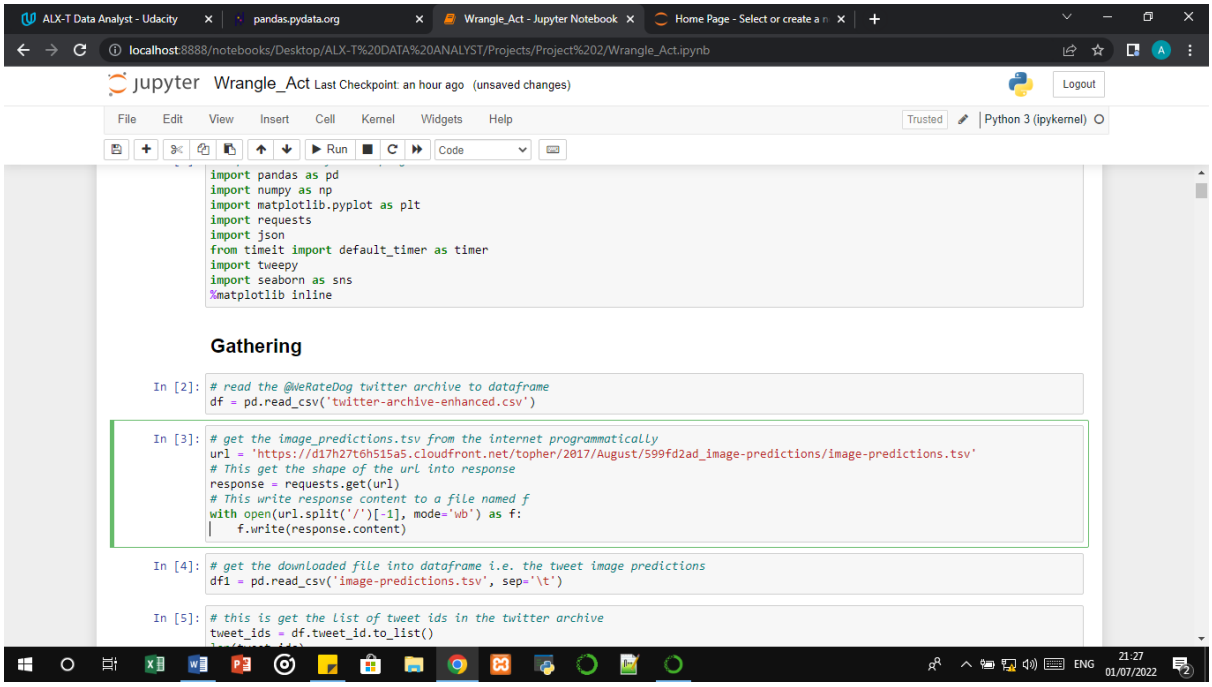
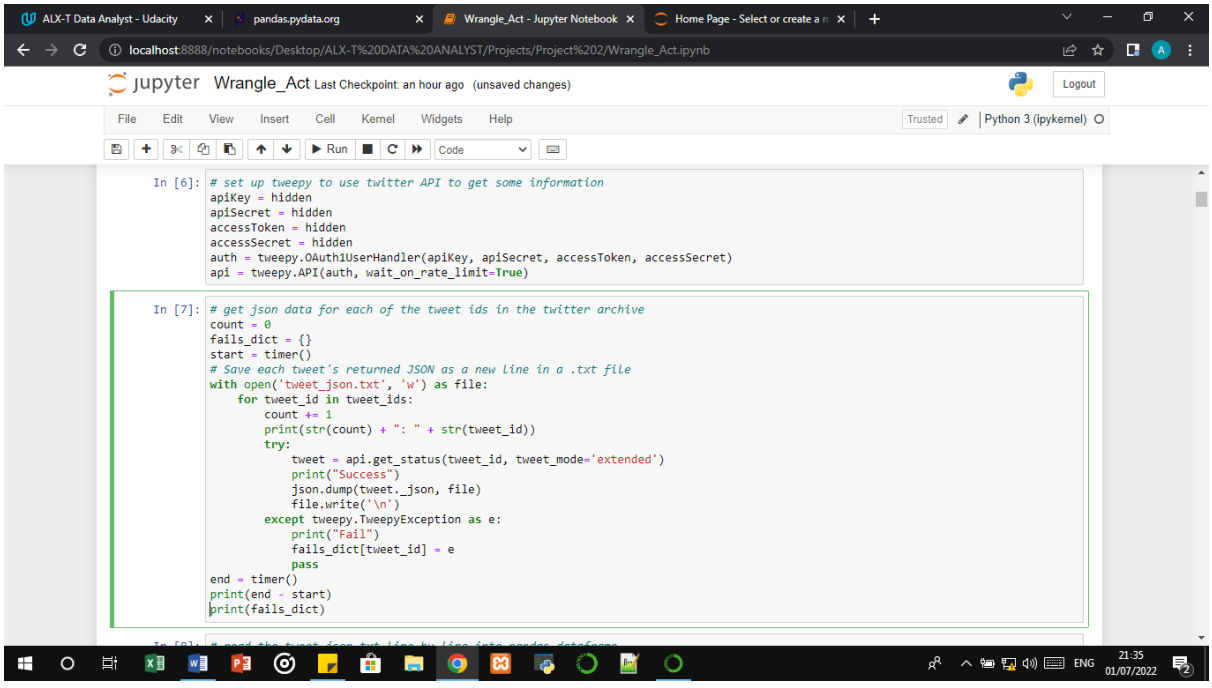

WERATEDOGS DATA
WRANGLING PROJECT

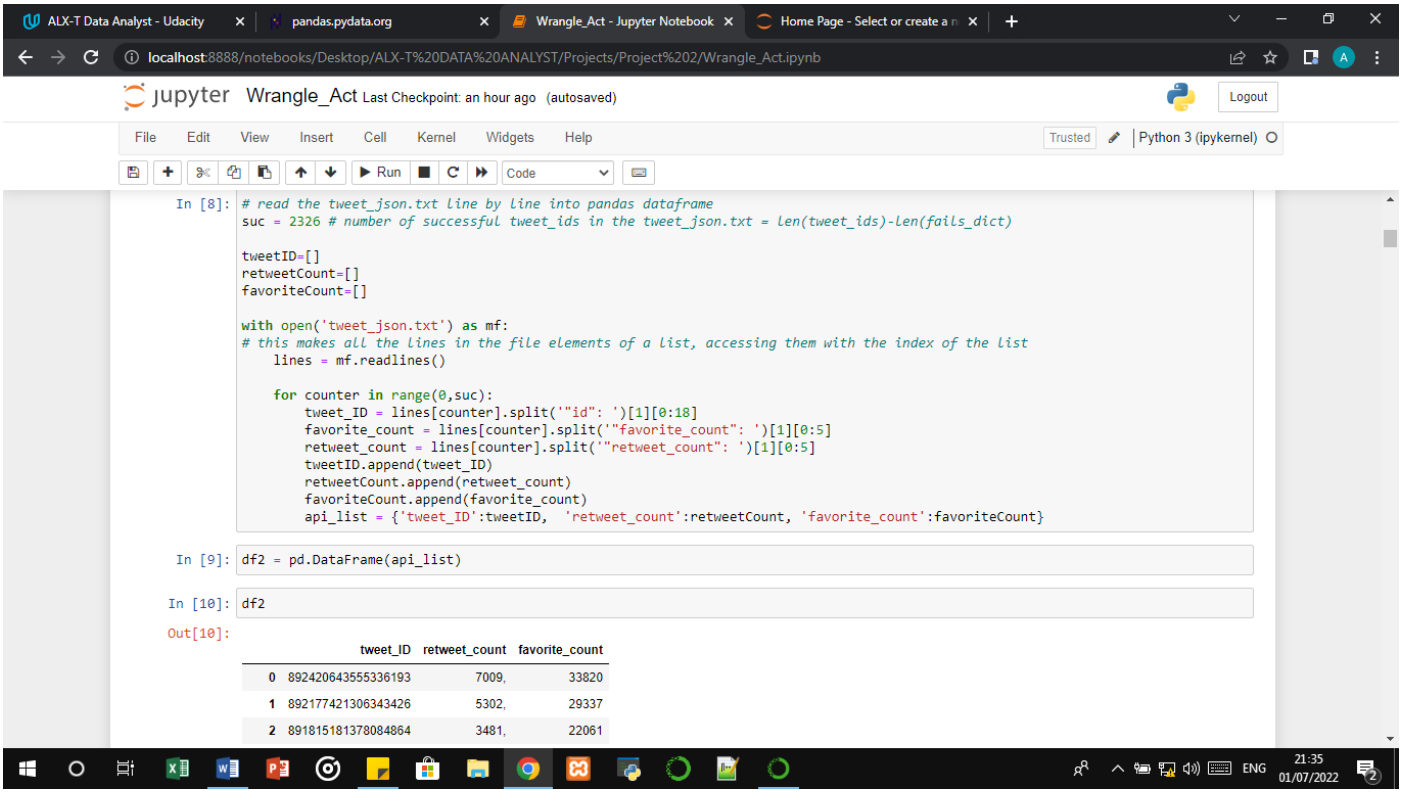
The dataset that I **wrangled**, analyzed and visualized is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent.

I start by importing all the necessary modules, i.e. pandas, numpy, matplotlib, requests, json, timer, tweepy, and seaborn. All these modules will be needed at one stage or the other later on in the project.

At the gathering stage, I have three data frames, the @WeRateDog twitter archive (twitter-archive-enhanced.csv) using pd.read_csv(), I programmatically downloaded image_prediction.tsv from the internet using python requests module which I later read into a data frame and lastly, I used tweepy to gather more data from twitter API which I saved in tweet_json.txt; it is from this I extracted tweet_id, favorite_count and retweet_id.







At the assessing stage, I did both visual and programmatic assessment. Some of the issues documented are:

Quality Issues

df2 (data extracted with tweepy)

- Some of the extracted data in the retweet_count and favourite_count contains some symbols which are not part of the desired data
- The data in retweet_count and favorite_count are meant to be int but they are read as string

df (twitter_archive data)

- Remove all the retweets in the dataset.
- In df (i.e. twitter_archive dataset), some of the columns have smaller data, less than 10% of the dataset rendering such columns unsuitable for analysis without regathering through for all tweet_ids such as in_reply_ columns and retweeted_status_columns
- Missing values in expanded_urls
- There are some ids in which the animal is not dog id=849776966551130114
- the timestamp in twitter_archive should be of the data type datetime and not string(object)
- Inconsistent numerator and denominator ratings (i.e. numerators and denominators having common multiples and denominators in 10s) at the ids 697463031882764288, 675853064436391936, 713900603437621249, 677716515794329600, 684225744407494656, 704054845121142784, 709198395643068416, 710658690886586372, 716439118184652801, 731156023742988288, 758467244762497024, 820690176645140481; (75 instead of 9.75) at 832215909146226688, 786709082849828864, 680494726643068929; (26 instead of 11.26), 778027034220126208 (27 instead of 11.27); for 810984652412424192, 24\7 means every time and not ratings; for 666287406224695296 (1/2 instead of 9/10);for 740373189193256964 (9/11 instead of 14/10); 682962037429899265 (7/11 instead of 10/10)
- Invalid dog names a, an. So, we can use text column to re-extract dog names for the affected dogs.

Tidiness Issues

- doggo, floofer, pupper and puppo are maturity stages in dog. Hence, the columns with these names in the dataframe twitter_archive (df) should be one column named *'dog_stages'*
- There should be one dataframe, that is merging the three datasets on the tweet ids.

At the cleaning stage, the best practices approach to cleaning was followed which is handling completeness issue (missing values) first, followed by the two tidiness issues, then other quality issues using the cleaning sequence, i.e. define, code and test. In order to perform the cleaning, I created a copy of each of the data frames using df.copy().

In df (i.e. twitter_archive dataset), some of the columns have smaller data, less than 10% of the dataset rendering such columns unsuitable for analysis without regathering through for all tweet_id such as in_reply_ columns and retweeted_status_columns, hence, I drop these columns.

Also, doggo, pupper, floofer, and puppo columns are values instead of column header, hence, I created a column named 'dog_stages' which indicated doggo, doggo/floofer, doggo/pupper, doggo/puppo, floofer, not specified, pupper, and puppo based on the data given in the dataset. The 'not specify' is specially curated for dogs in the dataset where the values for the four columns are none, while doggo are the one where doggo column is doggo, and others are None and ditto for floofer, pupper, and puppo. While doggo/floofer are for dogs whose doggo column is doggo and floofer column is floofer and others are None, and ditto for doggo/pupper and doggo/puppo. Then I drop the four columns – doggo, floofer, pupper, and puppo.

Still on tidiness, I merge twitter_archive (df) and the extracted data from twitter API using inner join. This make the data frames to be two, twitter related properties and image prediction data frames.

Other cleaning was carried out on the data frames using python and some of its libraries in a bit to clean it. After all this cleaning, I store the cleaned data on my local machine using another name for future purpose. Then carried out some analyses on the clean data.