Udacity DAND – Wrangling & Analyzing Project: Wrangling Internal report

Introduction
This report captures wrangling work done as part of the Wrangling & Analyzing Project. The goal of the project was to gather, assess, and clean WeRateDogs (@dog_rates) twitter related data so that some insights could be drawn along with a visualization. Finally, was stored/saved, and two reports created.

Gathering
Three different dataset files with different file types were required for the work:

1. twitter_archive_enhanced.csv
2. image_predictions.tsv
3. tweet_json.txt

The csv-file was provided by Udacity as "on-hand" file; the tsv-file was another Udacity file, but it had to be accessed/downloaded programmatically using the Python Requests Library; and, the txt-file data had to be downloaded programmatically from via Twitter's API using the Tweepy access library. Once the files were obtained, Pandas dataframes were created to view/access/work with the data.

csv-file overview:

```
1 tweets.columns
```

```
Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
       'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
       'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
       'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo'],
      dtype='object')
```

tsv-file overview:

```
1 img_preds.columns
```

```
Index(['tweet_id', 'jpg_url', 'img_num', 'p1', 'p1_conf', 'p1_dog', 'p2',
       'p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog'],
      dtype='object')
```

txt-file overview (once created):

id_str	retweet_count	favorite_count

Assess
First off, a user-defined function (UDF) was created to take in a dataframe and automatically print out and display various summary information; this was done for each of the 3 dataframes. The UDF used the following dataframe methods/attributes: .shape, .head(), .tail(), .sample(), .info(), .duplicated().sum(), .describe(), .nunique().

Udacity DAND – Wrangling & Analyzing Project: Wrangling Internal report

For the csv (`tweets`) and tsv-based (`img_preds`) dataframes, closer looks were taken using various additional methods / approaches. For example, .value_counts() and slices were looked at to find erroneous data or peculiarities that might warrant further investigation. Regular Expression (regex) patterning matching was also used in some cases (e.g. str.contains()).

In general, many data quality and tidiness issues were found, and while some were identified visually, others were identified programmatically. Some examples of findings include: Nulls, extraneous columns, non-dog related tweets/rows, odd rating numerators and denominators, dtype 'int' requiring conversion to 'string', merging required (to join the tweet retweet and favorite counts), etc.).

Since the third dataframe (based on txt-file) was small and was carefully coded/constructed, little assessment was required here.

Clean

In general, the cleaning was performed using Pandas and Python functionality and built-ins, but also modules such as "re" (regex). To provide an overview without going into detail that is best viewed/understood with the notebook file (e.g. ipynb or html or pdf file), a table capturing some of the method/functions/etc. that were employed based on the situation, is shown below. Cleaning was performed on copies of the initial dataframes, not the originals.

General Quality/Tidiness Issue	Example methods/functions/approaches Used
id-values interpreted as `int`	pd.read_csv('xxxxx.txt', dtype={'id_str': str})
Extraneous/unnecessary column	df.drop(labels=[col1,col2], axis=1, inplace=True)
Find df elements (simple)	df.query()
Find df elements (difficult)	str.contains(), str.extract(), re.search()
Change/fix df elements	Loops, conditional statements, indexing (.loc or ,iloc)
Join dataframe	df.merge()
Append (add rows)	df.append()

Wrangling - Summary/Conclusion

Data from 3 sources were gathered, assessed, cleaned, saved, analyzed at a high level, and a visualization was produced.

The wrangling work required significant effort based on the lack of data quality, especially in the main "tweets" data (twitter-archive-enhanced.csv'); and more work could be done to improve the cleaned/final dataframe.

The "image predictions" data (from tsv) needed some cleaning to become useful as a tool to filter out erroneous rows in the main "tweets" data, and more refinement might be achievable here. Further, if the ML algo could be improved, it might be significantly easier to filter out the poor quality / erroneous data.