Documenting Wrangling Efforts for Project: Wrangle and Analyze Data

Initial Steps

Once I reviewed the requirements for this project the first thing that I did was to create my Twitter development account. After exchanging some emails I was finally given access to the API.

I also downloaded the file **twitter-archive-enhanced.csv** to my local drive, then uploaded to the Jupyter Notebook.

Gathering

Gathering twitter-archive-enhanced.csv to a dataframe

This was very simple since I already had the file in the notebook, I just had to read it using pandas read_csv method for which I have a lot of experience accumulated during the past lessons. I saved it to a dataframe called “csv_df”

Gathering image_predictions.tsv to a dataframe

For this one I needed to read a website and then transfer the data to a .tsv file. I only did this process in one quiz so I had to do read the docs for **requests** and **open()** in order to understand how to complete the transfer. I also checked the encoding just to make sure it was utf-8.

The only issue I had was that the first time I forgot to add the separator value to read_csv and I ended up with all the columns values being joined in the same column. I saved it to a dataframe called “predictions_df”

Gathering tweet_json.txt to a dataframe

Once I got access to my developer account for twitter, I created a local notebook to document the file creation process. First, I created a loop to update a dictionary with all the json data from each tweet, then after the loop ended I would dump the json dictionary to a file. For some reason this only gave me 32 rows of data, so to avoid investing more time, I decided to do the file append process inside the loop, below is the main file creation code (the file will be read into a dataframe called “json_df”).

```
for i in csv_df['tweet_id']:
    try:
        tweet = api.get_status(i, tweet_mode='extended')
        with open('tweet_json.txt', mode='a') as file:
            json.dump(tweet._json,file)
            file.write('\n')
        if count % 100 == 0:
            print("Progress: " + str(count) + " of " + str(completed))
        count += 1
    except:
        pass
```

Assessing

For this section I did some research on how to improve dataframe visualization in Jupyter Notebook. I did not like that when viewing cells with long values the notebook would wrap the text in multiple nested rows. I also did not like truncated data since it may be hiding some issue in the data.

So, for the first issue of word wrapping, after many attempts with pandas options, I decided to use jupyter's built-in magic command %%html to add CSS styling to both headers and values of the dataframe. I also added borders.

Additionally, to display full values for each cell, I changed pandas options to increase the limits on column width and number of characters per cell.

Finally, just as an extra adjustment for my taste, I also added the CSS properties to left align everything.

Assessing csv_df

I did some research of how twitter works since this was the first time I was using it. I wanted to understand what Retweets are and how replies work.

From the visual and programmatically analysis I found the following quality and tidiness issues:

- Source column is not necessary for this analysis
- 78 rows are replies and 181 are retweets
- There are 220 tweets with no photo uploaded since they used a video instead
- Type of dog is in separate columns (doggo, floofer, pupper and puppo)
- Reply and retweet info not necessary after cleaning replies and retweets
- Some numerators and denominators are too low or too high to be correct (0, 1776, etc)
- Some values in name column are not names

Assessing predictions_df

Findings:

- For these columns I only need one column with the dog prediction that has the highest confidence.

img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog

Assessing json_df

Findings:

- Most of the columns do not have interesting information for this analysis

Cleaning

For the cleaning part of this project I looked for ways to keep the final dataframe as simple as possible, so, after creating copies of each of the 3 dataframes, I performed the following steps:

- Quality csv_df- Some values in name column are not names
    - Replace None and the most common values (>1 ocurrence) misintrepreted as names ('a','the','an','one' ,'just', 'getting') as NaN
- Quality csv_df - Denominators different than 10 (0 for example)¶
    - Remove 23 records with denominators different than 10 (most of them are because they are rating groups of dogs)
- Quality csv_df - Some numerators are too high to be correct (1776 for example)
    - Remove 28 records with numerators higher than 14 (14 is the valid current highest record)
- Quality csv_df - There are 220 tweets with no photo uploaded since they used a video instead
    - Remove 220 records without photos
- Quality csv_df - 78 records are replies
    - Remove 78 records that are replies
- Quality csv_df - 181 records are retweets¶
    - Remove 181 records that are retweets
- Quality csv_df - Some records have more than one stage of dog
    - Remove 14 records that have more than one stage of dog (doggo, pupper, etc) since they are not 100% defined.
- Quality predictions_df - Images are of different size
    - Add ":thumb" to the end of each image link jpg_url to ensure they are of the same size
- Tidiness csv_df - Stage of dog is in separate columns (doggo, floofer, pupper and puppo)¶
    - Add one column that contains the stage of the dog
- Tidiness csv_df - Removing unnecesary columns:
    - Remove columns no longer needed (in_reply_to_status_id, in_reply_to_user_id, source, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls,doggo, floofer, pupper, puppo)
- Tidiness predictions_df - For these columns I only need one column with the dog prediction that has the highest confidence: img_num p1 p1_conf p1_dog p2 p2_conf p2_dog p3 p3_conf p3_dog
    - Create new column with best dog prediction and if none are dogs set the value to "Not a dog!"
- Tidiness json_df - Most of the columns do not have interesting information for this analysis
    - Keep only columns favorite_count, id and retweet_count
- Quality twitter_archive_master_df - Some tweets do not include dogs¶
    - Remove Tweets were the predicted image was not of a dog