Wrangling Report

Gbenga Thompson Awojinrin

In carrying out this project, I had to carry out wrangling steps to obtain 2 datasets, the twitter_archive_master.csv and image_archive_master.csv datasets. All the wrangling procedures observed in order to accomplish this were carried out using the Python programming language in a Jupyter notebook and are outlined below:

Gathering the data

- Downloading the twitter_archive_enhanced dataset manually
- Downloading the image_predictions dataset programmatically
- Putting checks in place to ensure that if the notebook is ever run in an environment without these 2 files, it downloads them programmatically.
- I then proceeded to setup my Twitter API object with the tweepy library, authenticating it with access keys gotten from the Twitter developer portal
- Using the .lookup_statuses method available to the API object, I query the Twitter API in batches, allowing me to get the 2000+ tweets without getting stopped by the rate limit. The query responses are saved as json files, allowing me to perpetually access them without having to query the Twitter API everytime I run the notebook.
- tweet_id, retweet_count and favorite_count are extracted from these json files, saved as a tweet_json.txt file, and then reloaded into the notebook as tweet_json, a pandas DataFrame

---

Data Assessment and Cleaning

The data was then inspected for tidiness and quality issues that would have made analysis difficult. Listed below are the issues that were discovered.
Using the Define-Code-Test framework, these issues were resolved respectively as tabulated below:

	Data Assessment Step	Data Cleaning Step
Tidiness	doggo, floofer, pupper and puppo columns of the archive_enhanced table all represent different values of a single variable, the dog stage	Melt the rows into one, making sure rows where all values are none is represented as None
Tidiness	The tweet_json variables should be part of the archive_enhanced table	Merge the tweet_json_clean dataframe with archive_enhanced_clean using pandas merge function
Quality	NA values in retweet_count and fav_count columns of the table after it is merged with the tweet_json table	Drop rows with NA values in these columns
Quality	The values in the source column are not properly formatted because they are still surrounded by html tags	Use string slicing to retrieve the text between the tags
Quality	expanded_url column contains links to Twitter and non-Twitter pages, e.g https://vine.co/v/iiLjKuYJpr	Replace the non-Twitter urls with the Twitter version by joining the https://twitter.com/dog_rates/status/ to the tweet_id
Quality	Some of the tweets in the archive_enhanced dataframe are not original tweets, but retweets of other tweets	Drop records that have a non-NAN value for retweeted_status_id

Quality	After resolving #3, retweeted_status_id, retweeted_user_id and retweeted_status_timestamp columns now contain nan values only	Drop the affected columns using the pandas drop function
Quality	Erroneous datatype for timestamp column	Convert the timestamp column to Timestamp object using the pandas.to_datetime function
Quality	Incorrect rating_numerator and rating_denominator values extracted from text in some records, e.g, value of 0 in rating_denominator column	Use new regex patterns to extract rating_numerator and rating_denominator values from the text column
Quality	Prediction values are completely lowercase for some while others are titlecase	Convert everything to lowercase for uniformity and consistency
Quality	Inconsistent records between the archive_enhanced and image_predictions tables	Drop records with tweet_ids that are not common in both tables

---

Storing the data

To ensure that work done up to this point was not lost, I saved the gathered, assessed and cleaned datasets to csv files named twitter_archive_master and image_archive_master

In [ ]: