WeRateDogs Data Wrangling Report

documentation for data wrangling steps: gather, assess, and clean

Step 1: Gathering Data

- Load required libraries.
- Twitter archive data:
    1. The WeRateDogs Twitter archive, provided in UDACITY classroom.
    2. Download `twitter_archive_enhanced.csv` file manually.
    3. Load twitter archive data into pandas DataFrame using `pd.read_csv()` function.
    4. Test by viewing df head and info.
- The tweet image predictions:
    1. Include data about what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network.
    2. downloaded programmatically using the Requests library and the following URL: **https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv** .
       and save to `image_predictions.tsv .`
    3. Load twitter archive data into pandas DataFrame using `pd.read_csv()` function.
    4. Test by viewing df head and info.
- Each tweet's retweet count and favorite count:
    1. Due to trouble creating Twitter developer account, will be using tweet-json.txt.
    2. Download tweet_json.txt file from UDACITY classroom.
    3. Read file to explore tweet_json structure and available data that can be used.
    4. Create list of keys that represent data available.
    5. Identify useful data: 'tweet_id', 'favorite_count','retweet_count', 'created_at','user_count'
    6. read tweet_json.txt file and export needed data into list of dictionaries. By iterating through each line and reading with json
    7. Create DataFrame df_api_info = pd.DataFrame(_list)
    8. Test by viewing df head and info.

Step 2: Assessing Data

- Visually:
    1. Load Dataframes in jupyter notebook
    2. Open twitter_archive_enhanved.csv in excel, use filter built in function
    3. Open tweet_json.txt with vsc
- Programmatically:
    1. Use pandas: df.head() , df.sample() , df.info() , df.describe() , iloc[], groupby()
- Data Quality issues:

A. archive_df:
    1. remove retweets with retweet status id
    2. remove reply tweets with reply status id
    3. remove unwanted 6 columns: `in_reply_to_status_id`, `in_reply_to_user_id`, `source`, `retweeted_status_id`, `retweeted_status_user_id`, `retweeted_status_timestamp`.
    4. missing dog names.
    5. index 45:'883482846933004288', 784:'775096608509886464', 1068:'740373189193256964',716439118184652801,722974582966214656 miss read
    6. index 387:'826598799820865537','682962037429899265' joke misinterpreted.
    7. invalid rating at index 313:'835246439529840640', 2335:'666287406224695296',516:'810984652412424192')
    8. index 200, 460, 950, 575 doggo, floofer, pupper, puppo type wronge.
    9. missing data in expanded url can be collected from tweet-json.txt index
    10. Data type error: tweet_id, retweeted_status_id, timestamp should be datetime
    11. 20 tweets with different rating denominator (not 10).
B. remove tweets with no images.
C. some tweets have more one https in their url column.
D. df_image_predictions:
    i. 3 columns have the same variable
    ii. rename non descriptive column name.
    iii. some of dog breed names in column is lowercase
E. df_api_info: - remove quote tweets

- Data Tidiness issues:
    A. doggo, floofer, pupper, puppo columns in twitter_archive_enhanced.csv should be combined into a single column as this is one variable that identify the stage of dog.
    B. Information about one type of observational unit (tweets) is spread across three different files/dataframes. So these three dataframes should be merged as they are part of the same observational unit.

Step 3: Cleaning Data

- make a copy of each DataFrame using df.copy()
- remove retweets with retweet status id
    - define: filter out rows with value in retweeted_status_id
- remove reply tweets with reply status id
    - define: filter out rows with value in in_reply_to_status_id.
- remove unwanted columns:
    - define: create list of un wanted columns.
    - remove using df.drop()

- misread rating

- misinterpreted jokes
- invalid rating at index
    - define:
        - create list (wrong_rating_id) with tweet_id for all invalid, misread and misinterpreted ratings
        - view rows to be modified to check if any were deleted
        - create dictionary of key = tweet id, value = correct rating
        - replace wrong rating with the wright ones
- drop row index 516: tweet have no rating
    - **define:** archive_clean.drop()

- wrong datatype in columns tweet_id, timestamp
    - **define:** change datatype using str() and pd.to_datetime()

join dataframes

- same observational unit in two tables: archive_clean, image_prediction_clean
- **define:** merge two dataframes with pd.merge()
- remove tweets with no images.
- **define:** remove tweets with jpg_url is null
- doggo, floofer, pupper, puppo: 4 columns, one variable
- define
    - remove none entries in each columns
    - create new column 'dog_stage' with data from all 4 columns
    - delete old stage columns
- Quote tweets are not original tweets
    - define:
        i. remove tweets with quote_status == True
        ii. drop quote_status column
- timestamp and created_at columns represent same data
    - define:
        i. remove timestamp column
        ii. change created_at to datetime

- image_prediction_clean

    i. rename non descriptive column names
    ii. some of dog breed names in column is lowercase

- there are 3 predictions of every image
- we are going to use the p1 prediction with the highest prediction confidence
- join p1 and p1_dog to tweet_data_complete
- define:
    - copy columns p1 and p1_dog to tweet_data_complete
    - rename columns p1, p1_dog
    - fix lower case with .tilte() p1

save clean data