Wrangling Twitter Data: @WeRateDogs®

In this blog I will be briefly documenting the wrangling process I carried out on the tweet data extracted from the account called WeRateDogs®.

Process 1: Inputting Files as Dataframes

The following files were used to create Dataframes:

- twitter_archive_enhanced.csv (provided)
- tweet_json.json (extracted from twitter API)
- image_predictions.tsv (provided)

Process 2: Additional Twitter Extracted using Twitter API

Now additional twitter data was scraped using the Twitter API. The code for which can be found in the IPython Notebook. The twitter data extracted was saved as a json and read into Pandas and stored as a DataFrame called df_twitter_api.

Process 3: Setting up Dataframes for Wrangling

A copy of the three dataframes were created called:

- df_twitter_archive_clean
- df_image_clean
- df_twitter_api_clean

The data wrangling process will take place in these newly created Dataframes.

Process 4: Data Wrangling – Twitter Archive Data

The following data wrangling steps were taken to clean the twitter archive data:

1. Duplicates due to Retweets were removed.
2. Incorrect Denominator Values in the ratings were fixed
3. Incorrect Numerator Values in the ratings were fixed
4. Missing Values in the following columns were fixed:
    i. doggo
    ii. floofer
    iii. pupper
    iv. puppo
5. Duplicate labelling of Dog Stages was fixed
6. Each observation was placed in rows for the dog stages information
7. Incorrect Datatype for the following columns was fixed:
    i. timestamp
    ii. dog_stage
8. Following unnecessary Columns were removed:
    i. 'in_reply_to_status_id'
    ii. 'in_reply_to_user_id'
    iii. 'retweeted_status_id'
    iv. 'retweeted_status_user_id'
    v. 'retweeted_status_timestamp'

vi. 'expanded_urls'
vii. 'rating_numerator'
viii. 'doggo'
ix. 'floofer'
x. 'pupper'
xi. 'puppo'
xii. 'source'

Process 5: Data Wrangling – Additional Twitter Data

The following data wrangling steps were taken to clean the additional twitter data gathered using twitter API:

1. Duplicates due to Retweets were removed.
2. Missing Data was evaluated
3. Following unnecessary columns were removed:
    i. 'possibly_sensitive'
    ii. 'possibly_sensitive_appealable'
    iii. 'lang'
    iv. 'user-screen_name'
    v. 'quoted_status_id'
    vi. 'quoted_status_id_str'
    vii. 'quoted_status_permalink'
    viii. 'quoted_status'
    ix. 'is_quote_status'
    x. 'retweeted_status'
    xi. 'id_str'
4. Following columns were removed:
    i. ‘favorited’
    ii. ‘retweeted’
5. Incorrect Datatype for the following column was fixed:
    i. 'created_at'
6. Following column was renamed:
    i. 'id' to 'tweet_id'

Process 6: Data Wrangling – Image Predictions Data

After thoroughly evaluating the DataFrame called df_image_clean which has the image prediction data for the twitter data, it can be concluded that it needs no data wrangling.

Process 7: Data Wrangling – Merging of Dataframes

The dataframes will be merged in the following order:

1. 'tweet_id' is a column found in all three DataFrames, hence the datatype for this column is converted to object in all three DataFrames.
2. The df_twitter_archive_clean DataFrame was merged with df_twitter_api_clean DataFrame and was stored in df_final DataFrame. Inner Join was used as it keeps only those entries which match tweet id.
3. The df_final DataFrame was merged with df_image_clean DataFrame and was stored in df_final DataFrame. Inner Join was used as it keeps only those entries which match tweet id.

Process 8: Data Wrangling – df_final DataFrame

Final Cleaning of df_final DataFrame will be taking place.

1. The following duplicate columns were removed from df_final DataFrame:
    i. 'full_text'
    ii. 'created_at'
2. Removal of the observations(rows) from df_final DataFrame based on non-dog breeds found in the following columns, using drop function:
    i. 'p1_dog'
    ii. 'p2_dog'
    iii. 'p3_dog'
3. The following columns were removed from df_final DataFrame:
    i. 'p1_dog'
    ii. 'p2_dog'
    iii. 'p3_dog'
4. The following columns were formatted, converting the figures into percentage values:
    i. 'p1_conf'
    ii. 'p2_conf'
    iii. 'p3_conf'
5. Replace the ambiguous data in the name column of the df_final dataframe with the appropriate replacement using replace function.