Data Wrangling Report

Yasmina A. Kamal
February 2021

Table of Contents

Data Wrangling 1

Data Assessment 1

- Quality Issues 1
- Tidiness Issues 2

Data Cleaning 2

Data Store 3

Analyze and Visualize 3

Data Wrangling

In this step, collecting data. For this project, there were three main sources for the data to deal with:

- Twitter_archive_enhanced.csv file, this file was downloaded manually from the workspace to my working directory and then imported into our working environment using Pandas function “pd.read_csv”.

- Image_prediction.tsv is the second file that has been hosted on a webpage and downloaded from its relevant URL, using the Requests library get function and pd.read_csv panda’s function. This file encompassed image predictions for the dogs’ breeds obtained through a neural network on the most of the tweets in the archive file.

- The final dataset was gathered from twitter REST API via the Tweepy library by querying the API to obtain extra information pertinent to the tweets’ ids in the first file, e.g. retweets count and favorite count aspects.

Data Assessment

In this step, I investigate the imported datasets both visually and programmatically for the quality and tidiness issues.

- The visual assessment done on excel and Jupyter Notebook and programmatic assessment is conducted in also Jupyter Notebook

Quality Issues

- Twitter_archive table
  - Missing values in name column and invalid names less than 2 characters.
  - By comparing the number of rows in image_prediction and twitter_archive tables, we found that there are many tweets in twitter_archive table has no image. This rows should be dropped.
  - NaN values in 'expanded_urls' column, it represnt tweets with no image, should be dropped.
  - Some tweets are actually retweets and replies not original tweets that have to be deleted.
  - Some columns have represntations of null values as 'None' not 'NaN'
  - ‘retweeted_stauts_timestamp' and 'timestamp' should be datetime not object.
  - Deal with rating_numerator and rating_denominator to make sure it extracted in right way from the text
- Image_prediction table
  - Create 1 column for image prediction and 1 column for confidence level
  - Drop rewteets and replies from the table
- Api_df table

- Keep original tweets only

Tidiness Issues

- values are column names ('doggo','floofer','pupper','puppo') in `twitter_archive` table
- Merge `twittwer_archive` with `api_df` tables
- columns headers are values, not variable names in `image_prediction` table

Data Cleaning

In this step, searching and find solutions for the mentioned issues:

1. 'retweeted_status_timestamp' and 'timestamp' should be datatime not object in twitter_archive table
   - 'retweeted_stauts_timestamp' and 'timestamp' should be datetime not object.
   - we should convert data type of each column from object to datetime

2. Some columns have represntations of null values as 'None' not 'NaN' in twitter_archive table
   - convert 'None' values with "" as empty string, in the columnns 'doggo','floofer', 'pupper'and 'puppo' in twitter_archive table

3. values are column names ('doggo','floofer','pupper','puppo') in twitter_archive table

   - concatenate the columns in one column 'dog_breed'
   - drop the old columns
   - Replace the empty string to np.nan
   - if the value of 'dog_breed' is combined two type, make it readable

4. Nan values in 'expanded_urls' column, it represnt tweets with no image, should be dropped in twitter_archive table

   - drop any row with 'NaN' value in column 'expanded_urls' with dropna()

5. ome tweets are actually retweets and replies not original tweets that have to be deleted in twitter_archive table

   - drop any row has value (not 'NaN') in column 'retweeted_status_id' because it is retweet not original tweet
   - drop any row has value (not 'NaN') in column 'in_reply_to_status_id' because it is reply not original tweet
   - drop retweets and replies columns as we don't need them anymore

6. drop retweets and replies in image_prediction_clean table

- drop and 'tweet_id' that matches 'tweet_id' of replies or retweets

7. By comparing the number of rows in image_prediction and twitter_archive tables, we found that there are many tweets in twitter_archive table has no image. This rows should be dropped.

   - check 'tweet_id' in 'image_prediction' table and 'twitter_archive' table, then drop rows in the 'twitter_archive' that their 'tweet_id' not in 'image_prediction' table

8. Missing values in name column and invalid names less than 2 characters in twitter_archive table

   - correct names which have value 'a'

   - replace 'None' values with 'NaN' with np.nan

9. Keep original tweets only in api_clean table

   - delete any row its 'tweet_id' not found in 'tweet_id' column in 'archive_clean' table

10. Deal with rating_numerator and rating_denominator to make sure it extracted in right way from the text in twitter_archive table

    - slice the records to investigate the right value of denominators that below or above 10
    - slice the records to investigate the right value of numerator that below 6 and above 15

11. columns headers are values, not variable names in image_prediction table

    - create 1 column for image-prediction and 1 column for confidence

12. Merging archive_clean with api_clean

    - merging tables with merge() function

Data Store

In this step, save all cleaned dataframes into two files, twitter_archive and api_df are combined and saved in one file named ‘twitter_archive_master.csv’. And the another table image_prediction was saved in file named ‘image_prediction_cleaned.csv’.

Analyze and Visualize

In this step, all insights and possible visualizations are represented and will explained in more details in ‘act_report.pdf’ file.