Wrangle Report

This document describes the steps taken to wrangle the #WeRateDogs Dataset. The Project is part of Udacity Data Analyst Nanodegree.

By Gwiza Bonhomme Maryse

11.02.2019

Data Wrangling Steps:

1. Collect the data
2. Assess the data
3. Clean the data

1. Collecting the Data

For this project we worked with data from 3 differents sources.

a. The dataset `twitter_archive_enhanced.csv` that we manually downloaded from the Udacity. We used pandas read_csv to import the dataset in jupyter notebook. In the file `wrangle_act.ipynb` as requested in the project details. I named the dataframe containing this data df_archive.

b. We also programmatically downloaded the file containing image and breed predictions. The file was downloaded and saved in a file called `image_predictions.tsv.` The file was imported in a pandas dataframe as image_df.

c. The last piece of data came from twitter API. we used the python library Tweepy as recommended in the project details. I first created an account on twitter developer platform. I then created an app. And I could finally get a `consumer_key, consumer_secret, access_token and access_secret` that I used for authentication so that I could download data from twitter API. I made a query to twitter api for data based on the list of tweet_id that I collected from `twitter_archive_enhanced.csv` tweet_id column. 19 tweet_ids failed out of 2356 entries. The data was stored in a file named `tweet_json.txt.` The data collected was in json format. And then stored in an array api_data = [] . using pd.dataframe converted to a pandas dataframe that we named api_data_df. This new dataframe had 32 columns, but we only needed 2 as explained in the project details. Those columns are favorite and retweet counts. This new diminished dataframe name was twitter_api_df.

2. Assessing the data

Now we are done collecting our data, we are going to visually and programatically assess our 3 pandas dataframes. The goal of the assessment is to find issues in our data. As specified in the project details, we had to select 8 quality issues and 2 tidiness issues in our data.
We are going to assess the 3 previously collected dataframes:

a. archive_df

   I first visually assessed this dataframe. I could notice a lot of column with None value. But also NA. I then programmatically assessed the dataframe with the command archive_df.info() command. The command indicated that most columns like the following had most values marked as non null. This is misleading because it might lead to think that the dataset is more full that it's actually is.

b. image_df

   With this dataset I also did a visual assessment. Visually, The dataset looked "cleaner " compared to archive_df. I could see that the dataset hold a column named jpg_url with an image url for each tweet. Another important information was the breed prediction. In the project motivation, we could see that the first prediction could be trusted with a 95% confidence. That is a very good confidence level so I considered those breed prediction in the analysis.

c. Twitter_api_df

   For this dataset in order to visually conduct the assessment, I used the command twitter_api_df.sample(500). I runned the command several times so that the values could shift. This is the smallest dataset that we have. We checked for duplicates. Null values. And we used pandas info() to conduct programmatic assessment.

In addition to the steps described above, I also used pandas shape() to compare the 3 datasets Sizes. I took a screenshots that is below. It was somehow expected that all 3 datasets wouldn't hold the exact same of tweets. This is a tidiness issue. And the cleaned dataset to be used for analysis should hold the same tweets.

```
In [31]: image_df.shape
Out[31]: (2075, 12)

In [32]: twitter_api_df.shape
Out[32]: (2337, 3)

In [33]: df_archive.shape
Out[33]: (2356, 17)
```

Conclusions on Assessment:

Data Quality Issues:

1. In the twitter archive file, (from the csv) I doubt that 745 dogs's name is None. I would tend to consider this as missing value. this is a quality issue and should be replaced by NA for missing value.

2. The columns doggo , floofer, pupper and puppo which are dogs stage have the value None (python equivalent of NA). which is an issue because when doing programmatic assessment with pandas, its shows no null values

3. in_reply_to_status_id with 78 non-null float64 has too many missing values, it would be hard to do any analysis with so many missing values.

4. in_reply_to_user_id with only 78 non-null float64 has too many missing values, it would be hard to do any analysis with this column, to drop

5. retweeted_status_user_id has too many missing values 181 non-null float64

6. retweeted_status_timestamp has too many missing values 181 non-null object

7. retweeted_status_timestamp with only 181 non-null object has too many missing values to be useful. The column will be dropped.

8. 19 tweet_id failed while taking data from the twitter api.

9. I also doubt that 55 dogs are really called "a". This might be a typo.

Tidiness Issues:

1. rename the name in the columns for image predictions. from p1 to breed_prediction_1

2. merge the archive dataset and api dataset based on tweet_ids

3. move the image and breed_prediction columns from image_df to new_df

3. Cleaning the data

Each time, I will address the issue and then test that the solutions I applied actually worked.

Fixing Quality Issues:

To avoid unnecessary lengthy explanations, when possible issues identified in the previous sections are combined in the cleaning phase below:

1. **In the df_archive file, in the column name, replace value None by NA. In this way, while performing programmatic assessment, we won't find that the most common name in the dataset is "None". This None might have represented python None.**
2. **The columns doggo , floofer, pupper and puppo which are dogs stage have the value None.I am going to replace None with NA. I used pandas replace(). Pandas info() was used to verify the value in each of the columns.**
3. **Deleting the ids that were present in the `twitter_archive_enhanced.csv` but not in twitter API. It is not necessary to keep those ids for the analysis. I created a new dataframe called df_archive_new. In that dataframe I stored tweet_ids from `twitter_archive_enhanced.csv` which were successfully downloaded from twitter API. To test the solution I subtracted the counts from df_archive and api_df. As expected the result was 0. Which proves that the solution was successful. Now both datasets hold the same tweets.**
4. **In df_archive, 55 dog's name were "a", which seems very unlikely. And seemed more like a typo. So I replaced a by NA for missing value. I tested the solution by making a query to find all names "a" and of course the query came back empty. Which is what we wanted**

Fixing Tidiness Issues:

1. **In image_df, the column name p1 is not saying much. To make that dataset more tidy and readable, we renamed the column breed_prediction_1. We tested this by using the following codes: image_df.columns.tolist(). The name breed_prediction_1 appeared as one of the column. Good.**
2. **merge the archive dataset and twitter_api_df dataset based on common tweet_ids. I created new_df.**
3. **move the image and breed_prediction columns from image_df to new_df. The new dataframe created is called final_df**

I concluded the wrangling efforts by saving final_df to `twitter_archive_master.csv` as in the project details. The analysis and insights will be described in a different document, `act_report.pdf.`

Thank you