Wrangle Report

0. Introduction

In this report, we will explain with details what we've done through the wrangling phase only. We will report the wrangling of each dataset separately. So, let's get started

For each dataset we will report:

1. A brief description of the dataset
2. What data issues were found from our assessing
3. How we cleaned the data issues found in the assessing phase

Note: We have gathered data from three datasets which were:

1. twitter-archive-enhanced.csv
2. tweet-json.txt
3. image-prediction.tsv

---

1. Wrangling process

1.0. Reminder of the wrangling phases

In this project, we followed the three main steps of wrangling which were:

1. Gather data **(Note: This step was considered as a main data analysis phase in our project, not a step in data wrangling phase)**

2. Assess data: Finding all the possible data issues in the datasets

3. Clean data: Done through three steps: *Define Code Test*

1.1. Wrangling the twitter-archive-enhanced.csv dataset

1.1.0. Brief description of the dataset

This file is a CSV (Comma Separated Values) file, and I have read it through the read_csv function

This dataset represents the ratings and the kind of the dog that the tweet is talking about, along with other info about the tweet

1.1.1. Assessing the dataset

We have applied both programmatic and visual assessment, and this was the result:

Quality issues:

A- Completeness issues:

1. 2278 nulls in *in_reply_to_status_id* column
2. 2278 nulls in *in_reply_to_user_id* column
3. 2175 nulls in *retweeted_status_id* column
4. 2175 nulls in *retweeted_status_user_id* column
5. 2175 nulls in *retweeted_status_timestamp* column
6. 59 nulls in *expanded_urls* column
7. 745 nulls in *name* column
8. 2259 nulls in *doggo* column
9. 2346 nulls in *floofer* column
10. 2099 nulls in *pupper* column
11. 2326 nulls in *puppo* column

B- Uniqueness issues:

No issues were found

C- Validity issues:

1. *tweet_id* column is int64 whereas it should be object
2. *in_reply_to_status_id* is float64 whereas it should be object
3. *in_reply_to_user_id* is float64 whereas it should be object
4. *retweeted_status_id* is float64 whereas it should be object
5. *retweeted_status_user_id* is float64 whereas it should be object
6. *timestamp* column is object whereas it should be datetime64[ns]
7. *retweeted_status_timestamp* column is object whereas it should be datetime64[ns]

D- Consistency issues:

No issues were found

E- Accuracy issues:

1. The *rating_numerator* column had many ratings that were more than 1000 which created a very obvious outlier
2. The *rating_denominator* column had a value of 170, which is absolutely wrong

Tidiness issues:

The *doggo*, *floofer*, *puppo* and *pupper* columns all represent one variable which is Dog_type. And this violates the data tidiness rule that states that:

Each column represents a variable

1.1.2. Cleaning the dataset

Quality issues:

A- Completeness issues:

1. Use the fillna method to fill the nulls of *expanded_urls*, *name*, *doggo*, *floofer*, *puppo* & *pupper* columns with placeholder values

2. Drop the *in_reply_to_status_id*, *in_reply_to_user_id*, *retweeted_status_id*, *retweeted_status_user_id*, *retweeted_status_timestamp* columns

B- Validity issues:

1. Using astype method for the *tweet_id* column to make its dtype **object**

2. Using to_datetime function for the *timestamp* column

C- Accuracy issues:

1. Use the clip method to make any value more than 20 in the *rating_numerator* column to be equal to 20

2. Make a def function to make any number not equal to 10 to be equal to 10, then applied it on the *rating_denominator* column

Tidiness issues:

Make a def function that takes the kind of the dog if found in one of the columns, and if it wasn't found, then put 'Unknown' instead of the dog type, and then places this new data in a column called Dog_type and drops the *doggo*, *floofer*, *puppo* and *pupper* columns after it's finished.

1.2. Wrangling the image-prediction.tsv dataset

1.2.0. Brief description of the dataset

This is a TSV (Tab Separated Values) file, and I have read it through the read_csv function but added the sep='\t' argument.

This dataset is about the results of predicting of three machine learning models *p1* , *p2* and *p3* the type of the dog inside the picture of the tweet

1.2.1. Assessing the dataset

We have done both programmatic and visual assessment and this was the result:

Quality issues:

A- Completeness issues:

No issues were found

B- Uniqueness issues:

No issues were found

C- Validity issues:

The tweet_id column should be object not int64

D- Consistency issues:

No issues were found

E- Accuracy issues:

No issues were found

Tidiness issues:

No issues were found

1.2.2. Cleaning the dataset

A- Validity issues:

Use the astype method on the *tweet_id* column to make its dtype **object**

1.3. Wrangling the tweet-json.txt

1.3.0. Brief description of the dataset

This is a json file written in txt format, and I have read it through the read_json function and added the lines=True argument

This dataset is about the data of the tweet (length in char, full text, number of retweets, number of favourite, ...)

Note: The dataset kept returning a Unhashable type: List error, so I made a function to change the any unhashable element into a tuple which is hashable

1.3.1. Assessing the dataset

We have done both programmatic and visual assessment, and this was the result:

Quality issues:

A- Completeness issues: (Note: There are many columns that don't have any values inside them)

1. 281 nulls in *extended_entities* column
2. 2276 nulls in *in_reply_to_status_id* column
3. 2276 nulls in *in_reply_to_status_id_str* column
4. 2276 nulls in *in_reply_to_user_id* column
5. 2276 nulls in *in_reply_to_user_id_str* column
6. 2354 nulls in *geo* column
7. 2354 nulls in *coordinates* column
8. 2353 nulls in *place* column
9. 2354 nulls in *contributors* column
10. 143 nulls in *possibly_sensitive* column
11. 143 nulls in *possibly_sensitive_appealable* column
12. 2175 nulls in *retweeted_status* column
13. 2325 nulls in *quoted_status_id* column
14. 2325 nulls in *quoted_status_id_str* column
15. 2326 nulls in *quoted_status* column

B- Uniqueness issues:

No issues were found

C- Validity issues:

1. *id* & *id_str* columns should be object but they are int64
2. *in_reply_to_status_id* column should be object but it's float64
3. *in_reply_to_status_id_str* column should be object but it's float64
4. *in_reply_to_user_id* column should be object but it's float64
5. *in_reply_to_user_id_str* column should be object but it's float64
6. *quoted_status_id* column should be object but it's float64
7. *quoted_status_id_str* column should be object but it's float64

D- Consistency issues:

No issues were found

E- Accuracy issues:

No issues were found

Tidiness issues:

The *display_text_range* column has two values: The start character and the end character, which violates the tidiness rule that states that:

Each column represents a variable

1.3.2. Cleaning the dataset

Quality issues:

A- Completeness issues:

1. Use the fillna method to replace nulls with placeholder values in *extended_entities*, *possibly_sensitive* and *possibly_sensitive_appealable* columns
2. Drop any column that has more than 2000 nulls

B- Validity issues:

Use astype method to change *id* and *id_str* columns dtypes into objects

Tidiness issues:

Make a def function that extracts the two numbers of the tuple and get their difference, then place the difference in a new column called no_of_chars_of_the_tweet, then drop the *display_text_range* column

Note:

1. I have dropped the *user* and the *id_str* column as they are of no use to me in the analysis

2. I have renamed the *id* column to be tweet_id so that merging datasets can occur properly

---

2. Merging dataframes

After cleaning the dataframe, I have saved the cleaned version of the file as a CSV file, then I merged the cleaned versions on the *tweet_id* as a primary key and saved the new df as master_df.csv, then checked whether any problems occurred to the master_df.csv after the merge and found some problems

1. The column of full text of the tweet was duplicated

2. The *tweet_id* became int64

3. The *created_at* and *timestamp* columns became object

4. The full text of the retweet contains RT @ which should be removed

And we have solved these problems as follows:

1. We dropped the duplicate of the full text column

2. Use astype method to change *tweet_id* into **int64**

3. Use to_datetime function to change both *created_at* and *timestamp* columns into **object**

4. Make a new def function to extract the text of RT @user_name from the retweet, then delete it

---

Now that we have finished the wrangling phase completely, now we are ready to start the analysis on the master_df which will be reported in the act_report file