DATA WRANGLE REPORT

Table of Contents

1. INTRODUCTION 3
2. DATA WRANGLING 3
2.1 Data Gathering 3
2.2 Assessing Data 4
2.3 Cleaning Data 5
3. CONCLUSION 8
- END OF DOCUMENT - 9

1. INTRODUCTION

A Dataset was provided for analysis. The dataset is a tweet archive for the twitter user @**dog_rates** also known as **WeRateDogs**. It is a twitter account that rates people's dog with a funny comment about the dog. This report aims at describing the wrangling effort used in gathering and cleaning this data from 3 different sources before the actual analysis and visualization are carried out.

2. DATA WRANGLING

The steps and procedures performed in other to have a clean and well-organized data will be broken down into three groups namely: Gathering, Assessing and Cleaning.

****Prerequisites****

All packages and Libraries such as numpy, request, pandas, matplotlib, json, seaborn, tweepy etc have been installed and imported prior to beginning the wrangling process.

2.1 Data Gathering

Data was gathered from three different sources. The data are listed below in the order of gathering:

1. **twitter_archive_enhanced.csv**: This was done by manual download. The file was sent by WeRateDogs twitter user via Udacity to be used for the purpose of this project. After download, file was then loaded into a pandas dataframe named **twitter_archive**.
2. **image-predictions.tsv**: This file contains the tweet image predictions ran through a neural network. It is hosted on Udacity server. The file was downloaded programmatically using the **Request** library from the URL provided by Udacity. This was downloaded and loaded into the pandas dataframe named **image_predictions**
3. **tweet_json.txt**: This text file contains the output gotten when tweet ID in the twitter_archive_enhanced is used in querying the twitter API to get additional information such as retweet count and favourite(like) count. This was achieved by setting up a developer account in order have access to query twitter API, after which you get each

JSON's data per tweet on a new line via tweepy library which are then stored in this text file. The text file is now read line by line and loaded into a pandas dataframe named **twitter_api**

2.2 Assessing Data

All loaded dataframes to be used are now assessed. This was done in two ways:

- **Visually**: The data was opened using an external application such as excel and assessed.
- **Programmatically**: The data were assessed using some pandas method such as info, value_counts(), isduplicate() etc.

The table below shows summary of observations which includes both quality and tidiness done during assessment on the three dataframes, at least 8 quality and 2 tidiness issues were identified.

Quality Issues

DataFrame	Visual Assessment	Programmatic Assessment
twitter_archive	Missing values in the 5 Columns (retweeted_status_timestamp, retweeted_status_user_id, retweeted_status_id,in_reply_to_status_id and in_reply_to_user_id)	Incomplete and null values for 6 columns namely (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id,retweeted_status_user_id, retweeted_status_timestamp, expanded_urls).
	Some tweets containing retweets are part of the dataframe, original ratings should be used as per specification.	Original ratings that have images in the expanded url column should also be used as per specification.
		Incorrect data type for the timestamp column.
		Incorrect data type for the tweetid column.
		Incorrect name of dog as "a", "o","the", "actually" etc in the name column
image_prediction	Lesser data at 2074 than the twitter_archive_enhanced file which is 2356	Incorrect data type for the tweet_id column

	Inconsistent decimal places in the prediction confidence columns	
	Inconsistent alphabet casing used in the **p1**, **p2** and **p3** columns.	
	Non-descriptive header	
twitter_api	Lesser data at 2325 than the twitter_archive_enhanced file which is 2356 but more than the image prediction file at 2074	Incorrect data type for the tweet_id column

Tidness Issues

DataFrame	Visual Assessment
twitter_archive	Various stages (*doggo*, *floofer*, *pupper* and *floofer*) of the dog can be classified under one column
	The Source column has some HTML before the actual source
image_prediction	Three variables in six columns (**p1,p1_conf,p1_dog, p2,p2_conf,p2_dog,p3,p3_conf,p3_dog**)
All DataFrame	Data in the 3 dataframes are all related to the same object Dog and its ratings but separated.

2.3 Cleaning Data

This is the stage whereby the quality and tidy issue observed during assessment are worked on to create a clean data that will be used to make meaningful insights and analysis using the define-code-test framework. Before a data is cleaned, it is good practice to always create and use a copy of the original dataset for cleaning purposes. A copy of the three dataframes were created namely: **twitter_copy**, **predictions_copy** and **api_copy**.

The below table shows the issues observed and what was done to clean them.

DataFrame	Assessment	Cleaning
twitter_copy	Missing values in the 5 Columns (retweeted_status_timestamp, retweeted_status_user_id, retweeted_status_id,in_reply_to_status_id and in_reply_to_user_id)	Remove columns with missing values
	Some tweets containing retweets are part of the dataframe, original ratings should be used as per specification.	Remove rows that have retweet and replies
	Incomplete and null values for 6 columns namely(in_reply_to_status_id, in_reply_to_user_id,retweeted_status_id,retweeted_status_user_id, retweeted_status_timestamp, expanded_urls).	Remove columns with missing values , however, we won't be removing expanded_urls column but focus only on those with 0 value
	Original ratings that have images in the expanded url column should also be used as per specification.	Drop rows having no images in the expanded_urls column
	Incorrect data type for the timestamp column.	Fix the incorrect data types of column timestamp from object to date time.
	Incorrect data type for the tweetid column.	Fix the incorrect data type of column tweetid from integer to string/object
	Incorrect name of dog as "a", "o", "the", "actually" etc in the name column	Replace all lower-case names of dogs with None.

	Various stages (*doggo*, *floofer*, *pupper* and *floofer*) of the dog can be classified under one column	Create a new column `dog_stages` and put the various dog stages (*doggo*, *floofer*, *pupper* and *floofer*) column under it.
	The Source column has some HTML before the actual source	Create a new column `tweet_source` and extract the actual tweet source from the URL in the `source` column
prediction_copy	Lesser data at 2074 than the twitter_archive_enhanced file which is 2356	Fixed by merging all 3 dataframes
	Inconsistent alphabet casing used in the **p1**, **p2** and **p3** columns.	Convert values in the columns `breed_1`, `breed_2`,`breed_3` of the `predictions_copy` table to lower case
	Non-descriptive column header	Replace the column headers with a more descriptive header
	Incorrect data type for the tweet_id column	Fix the incorrect data type of column tweetid from integer to string/object
	Three variables in six columns (p1,p1_conf,p1_dog, p2,p2_conf,p2_dog,p3,p3_conf,p3_dog)	Bundle the 6 columns and create 4 columns out of it using the panda 'wide_to_long' function
	Inconsistent decimal places in the prediction confidence columns	Round up all the values in `conf_1`, `conf_2` and `conf_3` to 3 decimal places

api_copy	Lesser data at 2325 than the twitter_archive_enhanced file which is 2356 but more than the image prediction file at 2074	Fixed by merging all 3 dataframes
	Incorrect data type for the tweet_id column	Fix the incorrect data type of column tweetid from integer to string/object
All DataFrame	Data in the 3 dataframes are all related to the same object Dog and its ratings but separated.	The 3 dataframes should be merged on the common column tweet_id

3. CONCLUSION

A new cleaned data emerged after assessing and re-assessing for more observations to be cleaned which was then tested and confirmed to be fine using pandas' method. The file was stored as "**twitter_archive_master**" which would be analysed with insights derived from the data. The below image displays a sample of the final output:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5886 entries, 0 to 5885
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            5886 non-null   object
 1   timestamp           5886 non-null   datetime64[ns, UTC]
 2   text                5886 non-null   object
 3   expanded_urls       5886 non-null   object
 4   rating_numerator    5886 non-null   int64
 5   rating_denominator  5886 non-null   int64
 6   name                5886 non-null   object
 7   dog_stage           5886 non-null   object
 8   tweet_source        5886 non-null   object
 9   jpg_url             5886 non-null   object
 10  img_num             5886 non-null   int64
 11  pred                5886 non-null   int64
 12  breed               5886 non-null   object
 13  conf                5886 non-null   float64
 14  outcome             5886 non-null   bool
 15  retweet_count       5886 non-null   int64
 16  favorite_count      5886 non-null   int64
dtypes: bool(1), datetime64[ns, UTC](1), float64(1), int64(6), object(8)
memory usage: 787.5+ KB
```

- END OF DOCUMENT -