Data Wrangling Report

By Abdelrahman Mohamed Ashraf

As an assignment for the Udacity Data Analyst Nanodegree; This report illustrates the main steps involved in the data wrangling of Twitter account "WeRateDogs".

Data Gathering

In this step, collecting the data takes place. There were three main resources for data in this project:

1. Twitter_archive_enhanced.csv file, this file was delivered on the project details page and downloaded manually then imported to the workspace using Pandas' "pd.read_csv" function.
2. Image_prediction.tsv is the second file, it was hosted on a webpage and downloaded from its url using the requests library in Python and then imported to the workspace using Pandas' "pd.read_csv" function as well.
3. The final dataset was gathered from the Twitter API via the Tweepy library in Python by querying the API to get extra data for each tweet in the first file using its IDs.

Data Assessment

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

1. The visual assessment was done through the Excel application to read the datasets and assess them visually and then the programmatic assessment was done through some Pandas' in the Jupyter Notebook.
2. Missing data was addressed first and then the messy structures that caused tidiness issues were addressed to enhance or facilitate dealing with the quality issues that fall in the 3 categories of Accuracy, Validity and Consistency.
3. Some of the cleaning efforts were guided or specified by the project motivation; for example, one of the key points in the project motivation was that the data does not include retweets or replies or tweets that had no images.

Table	#	Issue (Quality Issues)	Solution
Archive	1	Data types: timestamps are object type.	Type conversion from string to datetime data type.
	2	Tweet ids has an integer data type.	Type conversion to string.
	3	There were retweets and replies in the dataset.	Removing retweets and replies by indexing for the columns that had null values in the retweets and replies.
	4	Expanded_url column had null values.	Those rows were dropped as they did not have images in them.
	5	Had tweets that have no images so they need to be dropped.	Those rows were removed by comparing and slicing with the image_predictions dataset.
	6	In the newly created dog_stage column, separate the values that have two stages for example: doggopuppo with a '-' to be doggo-puppo.	Separated the two values by putting a hyphen between them using string slicing in Pandas.
	7	The rating denominator column sometimes had values larger than 10 which is not right.	Normalized all the values in the rating denominator by assigning them the value 10.
Api_df	8	Data types: timestamps are object type.	Type conversion from string to datetime data type.
		Issue (Tidiness Issues)	
Archive_df	1	The dataset had columns related to replies and retweets that needed to be dropped.	The columns were dropped using Pandas' drop function.
	2	The dataset had values represented as variables spanning the 4 dog_stage columns, and also they had a wrong	This is actually both a quality and tidiness issue. First the wrong representation of null values was dealt with as a quality issues by using the Numpy np.nan function to

		representation of null values as “None”.	assign NaN values to the values that were None. And then a single column named dog_stage was created to have this value.
Image_predictions_df	3	The dataset had values represented as variables in the p1,p2,p3 columns and their respective dog,conf columns	Rename the columns and then use Pandas' wide_to_long function to reshape the dataset.
Archive_df & Api_df	4	Retweet count and favorite count should be part of the archive_df.	Created a subset with the tweet_id, and both its favorite and retweet count from the api_df. Then left merged the archive_df with dataset on the values of the tweet_id column.

Output

Three tables:

1. Archive Table 1971 records, this is the main dataset.
2. Predictions Table 6225 records.
3. API table 2354 records, this has now use but is stored for future reference.