Wrangling WeLoveDogs Twitter Data

Overview

The project objective is to gather, assess, clean, store and provide analysis & conclusions on a dataset containing tweets from the WeLoveDogs Twitter account prior to August 1, 2017. Initial tables have been collected and need to be modified, merged and expanded upon in order to arrive at a final clean dataset.

I. Gathering the Data

Data was gathered from three sources:

1. The twitter-archive-enhanced.csv file. This file was read into the Jupyter Notebook from the working directory via the 'read_csv' function. contains a table of raw data (such as tweet id, timestamp, and text content) combined with the data derived from it (such as rating and mention of a dog stage).
2. The image-predictions.tsv file. This file was gathered from a website using the requests library and read in using the 'read_table' function. It contains predictions regarding the breed for each tweet, with the tweet id as the primary key.
3. The Twitter API. The API was called to gather the JSON for the relevant tweets and dump it in a .txt file. Then, a select few JSON fields relevant to the desired analysis were loaded into the Notebook.

The three tables collected were merged together using Pandas 'merge' function in order to get a single view of the data. While this could have been done at the end of the process or not at all (i.e. if the predictions were to be kept in a separate table than the tweets), joining the tables was deemed more efficient for the assessment and cleaning sub-processes.

II. Assessing the Data

First, certain issues which had been identified in the project requirements were validated. Then, the data was assessed programmatically and visually. Overall, 8 quality issues and 3 tidyness issues were raised.

Data Quality Issues

1. There are null values in some columns
2. Some dog names are inaccurate
3. Some observations are replies and tweets
4. The tweet_id field isn't the correct data type
5. The timestamp field isn't the correct data type

6. The format of dog breed is inconsistent
7. There are URLs present in the text field
8. The expanded_urls contains duplicate entries in the same observation

Data Tidyness Issues

1. There are multiple dog stages columns
2. There are multiple prediction columns
3. The same link is present in the text field as in the expanded_urls field (same issue as data quality #7)

III. Cleaning the Data

The 11 issues mentioned above were cleaned following the *define, code, test* approach.

	Define	**Code**	**Test**
Null values	Replace null values in the dataframe with 'None'	Use the df.fillna method	Use the df.info() method
Dog names	Replace observations with false dog names with 'None'	Use a combination of the ‘replace’ method and regex	Use the str.contains method
Retweets & replies	Filter out replies and retweets from the dataframe	Use indexing to isolate replies and retweets and drop the corresponding rows	Look at the dataframe shape
`tweet_id` column data type	Change the tweet_id column data type from integer to string	Use the astype method passing the str argument	Compare the type in the column of the new dataframe vs the original dataframe
`timestamp` column data type	Change the timestamp column data type from string to datetime	Use the astype method passing the datetime argument	Compare the type in the column of the new dataframe vs the original dataframe
Format of dog breed	Format the dog breeds such that the first letter is capitalized and there are no underscores.	Define a function to capitalize and replace underscores by spaces	Look at the head of impacted columns
URLs in the `text` column	Keep only the string to the left of the URL in the text column	Split the string on the ‘https//’ string and keeping only the first	Sample 5 random rows to see if a link is present

		element	
Duplicate URLs in `expanded_urls` column	Remove the instances of a duplicate URL in the expanded_urls column	Split the string on every comma and keeping only the first element	Compare duplicate observations from the initial dataframe with the new dataframe
Multiple dog stage columns	Create a single column containing the stage for each dog (if available)	Use the melt and drop methods	Compare old dataframe dog stages with those in the new column and looking at columns present
Prediction columns	The prediction columns will be cleaned such that the only columns remaining will be: ● prediction_breed: the breed of the prediction that has the highest confidence level and that is also a dog type ● prediction_confidence: the confidence level of the prediction in prediction_breed	● Combine a for loop iterating through the rows with if statements to create and merge a clean dataframe ● Drop superfluous columns	Compare new columns with initial columns
Link in `text` and `expanded_urls` columns	**Note:** this issue was resolved in **IV.g - URLs in text Column**.		

IV. Storing the Data

Finally the data was stored in a .csv file as well as a SQLlite database.