WeRateDogs Data Wrangling

Introduction

Real-world data rarely come clean. Using Python and its libraries, I gathered data from a variety of sources and in a variety of formats, assessed its quality and tidiness, then cleaned it. The dataset that I wrangled, analyzed, and visualized is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.

Data Gathering:

This is the first step of the project. I gathered the data from three different sources:

1- WeRateDogs Twitter archive data where I directly downloaded the CSV file from Udacity
2- Tweet image predictions where I programmatically downloaded it using the Requests library from a URL.
3- Additional Tweets Data where I collected using Twitter API.

Data Assessing

This is the second step of this project. I assessed the data visually and programmatically in Jupyter Notebook and noted the following issues:

Quality issues

1. There are images for retweets and replies.
2. 'timestamp' and 'tweet_id' columns have the wrong data types.
3. There are 66 duplicated images.
4. Dog categories 'doggo', 'floofer', 'pupper', 'puppo' have string data types that should be boolean. However, I will fix this issue with another solution to match tidiness issue #1.
5. Some rows have no dog category.
6. There is a row that has a zero 'rating_denominator'.
7. There are unrealistic ratings where the numerator is greater than the denominator.
8. Rating column needed.
9. Tweet Source has an HTML string that is not meaningful.
10. Fourteen records have two dog categories.
11. There are unnecessary columns 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp'.

Tidiness issues

1. Four variables in four columns in archive table 'doggo', 'floofer', 'pupper', 'puppo'.
2. All data frames should be in one data frame.

Cleaning Data

This is the third step of the project. First, I made a copy of each data frame that I created in 'Data Gathering; then, I cleaned all of the issues I documented while assessing.

Issue #1: Delete the retweets and replies in df_archive and their images in df_images.

a. I stored the tweet_id of the retweets and replies in an array.
b. I dropped the rows in df_archive using the IDs array.
c. I drop the rows that have the same tweet_id in the df_images using theIDs array.

Issue #2: Change the tweet_id data type from integer to string in df_archive and df_images, and timestamp from string to DateTime in df_archive.

a. I converted the 'tweet_id' column to string using the astype(str) function
b. I converted the 'timestamp' column to DateTime using to_datetime() function.

Issue #3: Delete duplicate images.

a. I dropped the duplicated images using the drop_duplicates() function.

Issue #4: Fix the row with zero 'rating_denominator' and the unrealistic ratings where the numerator is greater than the denominator and create 'rating' column

a. The row with the zero 'rating_denominator' was dropped in the earlier issues, so it was fixed.
b. I stored the indexes of the rows with the numerator is greater than the denominator in an IDs array.
c. I believe that the numerator should be equal to the denominator, so I replaced the numerator with the denominator using.
d. I created a 'rating' column and stored the result of numerator/denominator in it.

Issue #5: Replace the tweet source HTML string with the HTML content to be meaningful.

a. I replaced the HTML string with its content (meaningful strings) using replace() function.

Issue #6: Delete the rows that have two dog categories.

a. I stored indexes of these rows in an IDs array.
b. I dropped the rows using the IDs array and drop() function.

Issue #7: Delete unnecessary columns 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp'.

a. I dropped the unnecessary columns using .drop() function

Issue #8: Combine the 'doggo', 'floofer', 'pupper', and 'puppo' columns to a one-column 'dog_category' and impute the rows that have no dog categories with the 'Unknown' category.

a. I created the 'dog_category' column.
b. I filled the 'dog_category' column with the appropriate values.
c. I dropped the 'doggo', 'floofer', 'pupper', and 'puppo' columns using the drop() function
d. I imputed the rows that have no dog categories with the 'Unknown' category.

Issue #9: Join the df_archive and df_images to df_twitter.

a. I joined the three data frames using the merge() function.

Storing Data

This is the fourth step of the project. I store the cleaned master DataFrame in a CSV file with the main one named 'twitter_archive_master.csv'.

Analyzing and Visualizing Data

This is the fifth step of the project. I produced 6 insights and 4 visualizations.