wrangle_report

Wrangling for this project included the steps of gathering, assessing, cleaning, and storing the data. This was done so that we could later analyze and visualize this data to provide insights into WeRateDogs' twitter history.

To gather the data, I first read the twitter archive CSV that had been provided to me into a pandas DataFrame. Then, I used the requests library to download the tweet image predictions TSV and read them into another pandas DataFrame. Finally, I used the tweet ids from the provided archive and the Twitter API to download every available tweet's JSON to a text file, which I then read into a new pandas DataFrame.

To assess the data, I used a variety of both visual and programmatic methods to look for both quality and tidiness issues in the data. This involved many different python queries, both before and even during the cleaning stage. Some of the visual assessment was done in Excel for easier scrolling.

Cleaning the data was by far the most complicated step of this project. The data I was provided had attempted to extract the dog's names from the tweets where provided. I created my own method of name extraction with a regex search to eliminate false positives and do a reasonable job of extracting most of the names correctly. Posts formatted unlike any other, and those with multiple named dogs are exceptions.

I used a similar method to extract the dog stage from almost every tweet where it was provided.

To extract the numerator and denominator of dog ratings I also used a regex search that looked for the last occurrence of something formatted like a rating. It also allowed for

numerators with decimal values. This led to a very high success rate for extraction of these values. I also had to make these values numeric again after my extraction.

Removing the unwanted retweets was as simple as making the DataFrame only those tweets without retweet data.

I joined the Prediction and Tweets DataFrames on their tweet id columns to create a single unified DataFrame, this also removed tweets without an image.

The tweet_id was represented in two different columns, so I removed the unwanted column and I also made some non-issue visual changes such as column names.

The tweet_id was stored numerically, so I changed it to be stored properly as a string using the astype function.

Finally, storing the data was done with the pandas to_csv function to store the DataFrame as a CSV.