Reporting: wragle_report

The dataset that I worked on 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

I imported the various tables needed for the project twitter_archived,image_predictions and tweet_json respectively.

upon inspecting the datasets both visually and programatically,i noticed that a bit of data wrangling is needed.these were the steps taken.

1.I converted timestamp in twit_arch_clean to datetime using the "to_datetime function"

twit_arch_clean.timestamp = pd.to_datetime(twit_arch_clean.timestamp)

.

2.row with outlier 144 in rating_numerator column in twit_arch_clean was removed using :

twit_arch_clean= twit_arch_clean[twit_arch_clean.rating_numerator != 144]

.

3.row with outlier 204 in rating_numerator column in twit_arch_clean was removed

4.row with outlier 960 in rating_numerator column in twit_arch_clean was removed

5.row with outlier 1776 in rating_numerator column in twit_arch_clean was removed

6.row with "Bookstore" in name column in twit_arch_clean was removed

7.rows with "none" in name column in twit_arch_clean was removed

8.row with "Actually" in name column in twit_arch_clean was removed

9.i checked for unique sources then proceeded to Extract the source by cleaning the "source" column in

twit_arch_clean table using regular expression ,returning unique values like 'Twitter for iPhone', 'Twitter Web Client', 'Vine - Make a Scene','TweetDeck as opposed to the original form of
<ahref="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone',

'< href="http://twitter.com" rel="nofollow">Twitter Web Client</a',

'< href="http://vine.co" rel="nofollow">Vine - Make a Scene</a',

'< href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a

code used was:

twit_arch_clean.source.unique() to check for unique values

twit_arch_clean['source'] = twit_arch_clean.source.str.extract('(?<=>)(.+?)(?=</a)', expand=True)
....using regex to

extract 'source

10 tweet_id column in twit_arch was renamed to id to enable join then

I Combined both twit_arch_clean and tweetj_clean to get one table named 'twit_main' after which

"source_y","in_reply_to_status_id","in_reply_to_user_id"

columns were dropped using:

twit_main=twit_main.drop(["source_y","in_reply_to_status_id","in_reply_to_user_id"], axis = 1)

11."source_x" column was renamed to "source" in twit_main using the code:

twit_main.rename(columns={"source_x":"source"},inplace=True)

12.final data was now stored as CSV file using code:

twit_main.to_csv ('twitter_archive_master.csv', index = False)