Wrangling Data Project

Wrangling data consists of three main steps: Gathering, Assessing and Cleaning. I will include what I did specifically in each stage of data wrangling process.

First: Gathering

I used the three sources presented by Udacity:

Twitter archive file: a given csv file that I read with the default read_csv method to read it as a dataframe.

Tweet Image prediction file: A tsv file that is first downloaded form a link using requests package and then reading the tsv file by using read_csv but with determining a parameter for the separator which is tab not a comma and by that this file is read as a dataframe.

And lastly reading from **Twitter API**: at first I didn't get the approval for the developer account, so I used the provided files of twitter_api.py and tweet-json.txt, but after a short time I was able to get the approval and then I started the real scraping using tweepy querying the API for each tweet's JSON data, and after that, all the data scraped from the API is then stored in a text file but with JSON structure.

We need to convert that txt file into a dataframe, so we will make a list with a dictionary inside for each entry from the JSON stored data, and after finishing the list we can convert that list into a dataframe as well as the previous two files.

Second: Assessing

We have two types of assessing, one is visually that is done by viewing the file in excel or google sheets, and programmatically using methods of pandas to show the content also pandas provide us with more methods that give us useful information about the data saved in the dataframe.

For the visual part, I viewed the twitter archive csv file in an excel sheet to see the consistency of the data, especially in names and rating denominator values.

For the programmatic part I used methods like head, describe, info and sample to get a better sense of the data we have.

From this step I determined what quality issues and tidiness issued that were found and also what I will work on.

In the quality issues I found:

twitter_archive:

- Erroneous datatypes in [tweet_ id - timestamp - retweeted_timestamp]
- no retweets keep the original only
- remove unnecessary columns [inreply x2 - retweeted x3]
- html tags in source, simplify source
- nulls in names and stages
- invalid denominator values

- some anti logic numbers in rating_numerator

images_predictions:

- Erroneous datatypes[tweet_id]
- duplicated URLs
- p1_dog, p2_dog, p3_dog if false, remove that row.
- columns names in image prediction

and for tidiness issues:

- divide text column in archive dataframe into two columns one for text and other for URL.
- merge arch and API tables together and then merge prediction with it.
- Make one columns for all prediction columns.
- Make one column for all dog stages(puppo, pupper, floofer, …)

I worked on the black points only.

Third: Cleaning

After I determined the issues, I started to clean the data, in my jupyter notebook I defined each assessment, coded it and then tested it.

The very first thing in cleaning process is to make a copy from all the dataframes we are working with and leaving the original data as it is.

After making the copies, we start the cleaning process, from removing unwanted data, removing duplicates, correcting datatypes and values.

The issues in gray I really couldn’t reach a satisfying result in them, I tried melting but this resulted in huge duplications, so I left it as it is and could work with that in insights in different ways.

Lastly:

The visualizations and insights part, I first questioned myself what are some things that are very interesting to be known and shown from this data?

And I found those in addition to many more but these are what I worked on:

- The most popular dog type. (insight + visualization)
- The most used source for tweets, retweets, favs. (insight + visualization)
- The most favorited dog image (insight + showing the image)
- The most popular name used (insight + visualization)
- The correlation between retweet_count and favorite_count (insight + visualization)
- The best prediction algorithm based on confidence. (insight + visualization)
- The most retweeted retweet (insight + opening the URL for that tweet)