WeRateDogs Data Analysis Project Report – Kamri Williams - April 2023

Data Gathering

This project contained 3 dataframes that are available:

- twitter-archive-enhanced.csv
- json_text
- image_predictions.tsv d

The twitter archive was downloaded directly from the course resources and read into my scripting environment using the read_cvs pandas command.

The json_text file was extracted from twitter.com using Tweepy API via Twitter development portal. The data was then read into a file called json_text.

The image prediction tsv file was downloaded using the request python library from the Udacity servers and read into my environment also using panda's 'read_cvs' with the tab delimiter parameter.

Accessing Data

Step 1: Performing a Visual Assessment of the data
- I used Excel to manually review the data and take any notes of data that will need to be cleaned later on.

Step 2: Perforing a Programmatic Assessment of the data
- I used the programmatic approach to find data quality errors that are not so easy to spot. The following functions were used in my exploration:
    - pd.info()
    - pd.head()
    - pd.describe()
    - pd.duplicated
    - pd.isin()

After preforming the 2 assessments I was able to choose 9 Quality issues and 2 Tidiness issues to work on in the cleaning stage:

Quality Issues

1. Erroneous Columns
2. Need to delete retweets.
3. Delete reply to tweets.
4. Erroneous dog names are present such as, 'by' or 'None'.
5. Abbreviating Source links
6. Filling in all the none type values with 'None' for more flexibility.
7. Some rating denominator values or more or less than 10.

8. There are tweets without images. They will need to be removed.
9. Choose the dog with the highest confidence rating.

Tidiness issues

1. Merge Dataframes and get rid of unneccesary columns
2. The dog stage columns (doggo,floofer,pupper,puppo) can be combined into one column.

Data Cleaning

1. I first copied our data sets so that I could have a backup handy in case I needed to start over.
2. I then decided to merge the 3 data sets together so that we could start off with all of our data together and represented. This allowed me to get the correct amount of columns for all dataframes and provide a cohesive dataframe.
3. I combined the 4 dog stages (doggo, floofer, pupper, puppo) into one column using a concatenation method and renaming the columns for a cleaner look. I also cleaned by removing the 4 separate column since they were not needed anymore.
4. Next I went down the listed of quality issues I listed earlier, using the following functions:
    a. .isduplcated() to remove the retweets
    b. .dropna() to drop any tweets that did not have pictures.
5. I then cleaned up the predictions to only show the strongest prediction, using the conf interval value as a determining factor between p1, p2, p3 conf columns.
6. I added a new column “confidence” and “breed” to display the breed with the high confidence interval value and the bread that was associated with that value.
7. Next I cleaned the source table by renamed the source url to something that would a little more readable. Creating a dictionary that holds the urls and source name as the key for a more formal look. The function returns the key that is association with the url.
8. I filled all of the ‘NAN’ values with ‘None’ so that I wouldn’t get any errors later on when analyzing the data.
9. I also changed the favorite and retweet count to whole numbers since that is how they should be represented.
10. I dropped the following irrevelevnt columns: ['p1', 'p1_dog', 'p1_conf','p2', 'p2_dog', 'p2_conf','p3', 'p3_dog', 'p3_conf','in_reply_to_status_id','in_reply_to_user_id', 'retweeted_status_id','retweeted_status_user_id','retweeted_status_timestamp','doggo','floofer','pupper','puppo']

Data Storing

Lastly I stored my merge master cleaned data set to a csv file “twitter-archived-clean.csv”.