Udacity Data Analyst Nanodegree

DATA WRANGLING PROJECT

OKONEYO ETIENEOBONG PETER

INTRODUCTION

In this project, I gathered data from various sources, assessed it, and cleaned it. The data was curated from the twitter's account WeRateDogs twitter archive which was provided to me by Udacity. The we rate dogs twitter account is popular account known for rating dogs. The data from the WeRateDogs archive was given to us in a csv file, it contains data that has been extracted from the twitter archive like ratings, name, and dog stages. The data from the twitter archive is not clean and neither does it contain all that is needed for proper analysis, therefore I queried the twitter API for more information like retweet count and favorite count. Querying the twitter API was done using some access and consumer tokens which can only be done with elevated access to a twitter developer account which I applied for and got. Finally, data was gotten which predicted the breed of dogs using the image of the dogs in the tweets and the confidence level for each dog.

THE DATA WRANGLING PROCESS

Data wrangling involves three processes, they are gathering, accessing and cleaning

1. **Gathering data:**
   As said earlier, data was gathered from three sources, they are:

- The we rate dogs twitter archive, which was a csv file given to us to download manually, and was read into a data frame called *df_twitter*
- A data set for image predictions which was predicted using neural networks given by Udacity to download programmatically, using the requests library in python. It was a TSV file which was read into a data frame called *df_image*
- The last data set was to be queried from the twitter API using the tweepy library in python and twitter developers access into a json format in a file called tweet_json.txt which was read into a data frame called *df_api*.

2. **Accessing data:**
   This was the longest stage of the data wrangling process. I viewed my data sets individually using both visual and programmatic methods. Visual in the sense that I looked through all of my data to discover issues in my data, and programmatic in the sense that I used libraries and python methods and functions to access my data. Although I could not thoroughly document every issue in my data to be cleaned but later in the analysis step further issues arose when analyzing so I had to iterate through the data wrangling process and clean it again as data wrangling steps are not just straightforward, they are also iterative.

Some of the issues I found and documented are:

Quality issues

Twitter archive dataset

- Irrelevant columns = (*in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp,expanded_url*) and rows with non-Nan values in these columns.
- Incorrect datatype for timestamp and tweet_id.
- The sources of tweets in source column are inside html anchor tags.
- From the text column we can see tweets that are not for dogs, they are other animals.
- Errors in the ratings
- Unreasonable names like 'a', 'an', 'the' and others in the name column
- Missing values in the columns doggo, floofer, pupper, puppo have "None" instead of nan

Image Predictions dataset

- Incorrect datatype for tweet_id
- 66 duplicate image URLs in the "*jpg_url"* column

Twitter API dataset

- Incorrect datatype for tweet_id.

Tidiness issues

- One variable (dog stages) in four columns: doggo, floofer, pupper, puppo
- Two variables (breed and confidence) in 6 columns: p1_conf, p2_conf, p3_conf and p1,p2,p3
- Three datasets instead of one master dataset

3. Cleaning data:

In this step of the data wrangling process, the issues that were documented after accessing were cleaned. This was done in the pattern of define, code and test. The cleaning method was defined, the code was written and run and the output was tested to see if the code worked and the changes produced. A copy of the each data frame was created for the cleaning process, all the processes were done on this copy

These were the cleaning procedure for the data analysis process:

- The rows containing replies and retweets were dropped and the irrelevant columns (*refer to accessing issue1)* were also dropped.
- The data type for timestamp and tweet_id was changed to datetime and string respectively
- The sources of tweets were extracted using regex methods
- Drop all rows where "only rate dogs" appears in the text column
- Re extract the ratings numerators and denominators from the text column, Convert the ratings extracted to over 10 and turn all denominators to the value of 10
- Change all unreasonable names like a, an, the and names that begin with lowercase to "None"
- Replace missing values as "None" to Nan in the doggo, floofer, pupper, puppo columns
- Drop all rows with duplicate URLs

- Merge the doggo, floofer, pupper and puppo columns into one column called dog stages
- Using the “select”: function in the numpy library, merge the p1,p2,p3,p1_dog,p2_dog,p3_dog,p1_conf,p2_conf,p3_conf columns into two columns called breed and confidence.
- Merge all three data sets into one master data frame.

Finally, the data was stored into a csv file called 'twitter_archive_master.csv' using the “*to_csv”* function