WRANGLING REPORT

In general, Data Wrangling is a process that is performed so that it helps in the process of analyzing data. Here in this project, we shall discuss about the data wrangling steps that are performed on the data.

1. GATHERING DATA:

First step here is gathering archive data for WeRateDogs twitter user. This is pretty simple as the archive data is available in a .csv file, hence we can use read_csv function to import the data.

Next, we need to gather tweet image predictions data which is a .tsv file available in a link. To extract this data, we need to import requests library and use get() function to extract data from the given url and store it in a file(tsv).The same read_csv function can be used to import the data.

Finally we need to get the retweet and favorite count for tweets in WeRateDogs twitter account. Here we need to use twitter API to extract the data. We have a library called tweepy in python to query the twitter API and extract the data. After creating a twitter developer account, we could use the tweet ids from the archive data and extract the JSON data for each tweet and store it in a text file. From the test file, we can get the required details (retweet & favorites count) from the JSON data for each tweet and finally store it in a Pandas dataframe.

2. ASSESSING DATA:

The second step in data wrangling process deals with analyzing the data for quality and tidiness issues. If any, we need to identify them and clean them later. Below are the data quality and tidiness issues found in the project.

- QUALITY ISSUES:
  1. The data type of ‘timestamp' variable is object (string) which needs to be changed
  2. We need to ignore retweeted tweets in the dataset since the analysis deals only with original tweets.

3. There seems to be null values in ‘expanded_urls’ column that needs to be fixed.
4. ‘Rating_denominator’ column has entries with values of 0 which doesn’t makes sense.
5. There seems to be a lot of different values in ‘Rating_denominator’ column which needs to be standardized so that it can be better for analysis purpose.
6. Rating_numerator needs to be cleaned, so that float values are extracted properly.
7. Names of columns needs to be meaningful in predictions_df dataframe.
8. Object (string) data in predictions_df needs to be capitalized for maintaining a regular structure in data.
9. The datatype of different columns should the reflective of the values present.

- TIDINESS ISSUES:
  1. There seems to be multiple columns which holds data about dog stage that needs to be made into a single column.
  2. We can merge/join data in all the 3 dataframes so that we can improve the tidiness in our data.

3. **CLEANING DATA:**

The analyzed quality and tidiness issues needs to be fixed in this step of cleaning the data.

- Changing the data type to datetime can be done by importing pandas and using to_datetime function.
- To ignore the retweeted tweets data, we can drop the rows for which the retweeted_status_id column contains any data (i.e. if this particular column stores data, then it’s for retweeted tweet.)
- It’s difficult to fix the null values in expanded_urls columns since this data was extracted from twitter archive. Either we can drop the columns having null values or drop the expanded_urls column itself since this column is not

being used in any analysis. I used the latter option in this project since it does not involve any data loss.

- Simply drop the rows having Rating_denominator value as 0 since we can’t rate anything out of 0
- Rating_denominator has multiple no. of values. While the majority of values are 10, there seems to be some other values too like 50, 80, and 20 etc... So if we plan to compare the rating for few tweets, the denominator value needs to be constant so that we can compare rating between them. We cannot compare two tweets with rating like 10/10 and 10/15.One way to fix this problem is by dropping all rows which don’t have a value of 10 as denominator since 2153 out of 2174 rows has a value of 10. Considering the majority portion of values belong to 10, we can drop other rows in the data frame. Other option would be to create a new column called rating and store the values in the column by dividing the numerator by denominator. By this process, we can standardize the ratings (both 10/10 and 20/20 will have same rating value of 1. If rating is greater than 1, then numerator value is greater than denominator). I chose the latter method here too since it does not involve any data loss.
- We can ignore the data after 21st August 2017 by comparing the values in timestamp column.
- Columns can be renamed for better understanding using the rename function on the data frame
- Capitalization of data can be done using str.capitalize function - Columns can be dropped from the data frame using drop function.
- Few values in rating_numerator column needs to be float values but that weren’t extracted properly. We can use regular expression to extract float values properly.

- For converting the data spread across four columns into a single column, create a new column called dog_rating and store the values in that column. Since the data in 4 columns are string, I appended all the 4 values into a single string and then used a function called standardize to store a single value in dog_rating column. This standardize function will store ‘None’ if there are no dog stages available in any of the four columns and if the columns contain any one of the dog stages, it will be stored in dog_rating.

- We can use merge function from pandas to merge two data frames.