# Udacity – Data Wrangling Project

Data Analyst Nano Degree

hind baaqeel

11/28/20

# Data Wrangling Report

## Introduction

This Project demonstrate the data wrangling process using twitter data on archive called WeRateDogs. In this archive, users are given a dog image and they rate it with their comments. It is worth noting that the setup of the rating system requires that the rating denominator is always higher than 10. In the project, I have used this dataset to practice data wrangling steps through gathering, accessing and cleaning the data inside the archive.

## Methodology

Data Gathering: The first step of data wrangling process it to gather the required data.

- WeRateDogs Twitter archive --> this dataset includes rating related information

  → this dataset is available from Udacity

- Dog Image Prediction --> this dataset includes the prediction of dog images

  → this dataset was downloaded programmatically using python codes

- Twitter Data--> this data include retweet counts and likes and others.

→ this dataset was gathered using twitter API by setting up twitter developer account and acquiring the required access credentials.

Data Assessment:

The second step was to assess the dataset both manually and programmatically to find issues that need to be cleaned. These issues were divided into quality issues and tidiness issues. The quality issue in more concern with content of the data and the tidiness issues was more focused on the structure of the dataset.

This step resulted in finding the following issues:

- **Quality Issues**

twitter_archieve

1- error in datatype (timestamp) --> object instead of datetime
2- according to the rating system, the denominator should be always 10. but it appears that we have other values such as 3, 2 .. etc
3- some values in the numerator is less than the denomnator (10) which is uncorrect according the rating system. [uncleaned]
4- null values in the columns(name, doggo, floofer, pupper, puppo) are not shown because it is written as None
5- There is an html header in the source column
6- False Dogs names like: a, an, the ... ----> written in lower case

image predict Table:

1- There is duplicates in images URL ---> repeated images are not useful to us

Twitter data Table:

1- missing data (2331), according to Udacity file it should be 2356
2- repeated column (text) --> since that they should be merged
3- id column is this table is not consistent with tweet_id in twitter_archieve ---> they should be the same to be merged into one table

- **Tidiness Issues**

Twitter_archieve Table:

1- dog types are separated in different columns(doggo, floofer, pupper, puppo)
2- Columns with no use (in_reply_to_status_id, in_reply_to_user_id)

image predict Table:*

1- Unclear Column names(p1,p2,p3)
2- dog image prediction can be summarized into two columns (dog prediction type, confidence level) instead of the 9 given columns related to prediction

Twitter data Table:

1- The table twitter data and table twitter_archieve contains info about a single tweet. It can be merged into single table. Also, it can be merged with image_predict to combine it with a prediction of a breed

Data Cleaning:

The third step was to start cleaning the data. I did this using the following steps:

- Change the datatype from object to datetime
- Reassign all values in rating denominator to be 10
- Change columns names(p1,p2,p3) to ---> first_pred, second_pred, third_pred
- Change the string None in these columns to null values using replace method and np.nan
- Find the missing data from the twitter data table so it can match the records in twitter archive. This can be done by downloading the complete json file provided by Udacity and append the missing record to our table.
- twitter archive and twitter merged into one table.
- duplicated images with the same URL should be dropped since they are of no use to us
- html tags need to be removed from the source column in full_archive table. This step will be done using regular expression that strips out all html tags from a string.
- Remove one of the two duplicated columns from merging twitter_data and twitter_archive --> text and full_text.
- Drop unwanted columns from full_archive table ('in_reply_to_status_id','in_reply_to_user_id','retweeted')
- We have a tidiness issue which is that 4 columns (doggo,floofer,pupper,puppo) represent single feature which is stage. first, we will create a new column called stage which contain a concatenated string of all of these columns. Then we will replace the nan value with empty string to extract the stage name only. Then we will check if there rows with multiple stages and inspect them separately using row index
- We have lots of incorrect dogs names such as a, an, the.., they are all in lower case letters. we suggest to replace it with null values so that we can know that the real dog name is missing.
- Merging image predtion table with full archive using pd.merge on tweet id as the key

- We need to reduce the number of predictions columns into two columns(prediction, confidence level), where prediction will give results only if the image was predicted to be dog

The final cleaned full_archieve set had the following features:

| FEATURE NO. | NAME | DATA TYPE |
|---|---|---|
| **0** | tweet_id | String |
| **1** | timestamp | Datetime |
| **2** | source | String |
| **3** | text | String |
| **4** | retweeted_status_id | Float |
| **5** | retweeted_status_user_id | Float |
| **6** | retweeted_status_timestamp | Datetime |
| **7** | expanded_urls | String |
| **8** | rating_numerator | Integer |
| **9** | rating_denominator | Integer |
| **10** | name | String |
| **11** | retweet_count | Integer |
| **12** | favorite_count | Integer |
| **13** | stage | String |
| **14** | jpg_url | String |
| **15** | img_num | Integer |
| **16** | Prediction | String |
| **17** | Confidence | Float |

## Challenges:

While working with this project, the first challenge that we had is that we needed to work with three separated datasets. Each of them has their own set of features so we needed to understand the relationship between them to combine them into one table. The second challenge that we have faced that we found some quality issues but we couldn't clean them. For example, when we found out that some of the rating numerator was actually less than 10 which is not acceptable according to the rules of the rating system. We think the reason for this mistake is that the users are not aware of the rating rule or it is may be a data entry mistake. But, we're not really sure about the cause of this mistake.

## Conclusion:

In this project, we practiced data wrangling techniques through three main stages: data gathering, assessing and cleaning. In this project, we used python as data assessment tool instead of inspecting it manually through sheet application such as excel. The reason for doing this is that python provide better tools that summarize all of your data in few lines of codes. Furthermore, using python is better when dealing with large number of data as explained in Udacity DAND data wrangling course. Also, the strength of using python comes especially from the open source libraries that made it easy to gather, assess and clean data programmatically.

Resources:

https://www.datacamp.com/community/tutorials/wordcloud-python

https://stackoverflow.com/questions/3398852/using-python-remove-html-tags-formatting-from-a-string

https://stackoverflow.com/questions/8209568/how-do-i-draw-a-grid-onto-a-plot-in-python

https://seaborn.pydata.org/tutorial/color_palettes.html

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html