Data Wrangling Report

Introduction

WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. We will use the data from this account for our project.

The purpose of this project is to practice data wrangling (a section of Udacity Data Analysis Nanodegree program).

Data wrangling process is divided into 3 main steps:

- **Gathering data**, download resources from deferent references.
- **Assessing data**, for quality and tidiness issues.
- **Cleaning data**, clean the quality and tidiness issues that identified in previous step.

The most tools, libraries and programming language used in this project are:

- Python
- Pandas Library
- Numpy Library
- Requests Library
- Tweepy Library
- Json Library
- Matplotlib Library
- Jupyter Notebook
- Twitter's API

Step 1: Gathering Data

Data was gathered from 3 different sources:

- **Twitter archive file:**
- The twitter_archive_enhanced.csv was provided by Udacity and downloaded manually.
- **The tweet image predictions**:
- This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library and URL information.
- **Twitter API & JSON:**
- Using the tweet IDs in the WeRateDogs Twitter archive.

Step 2: Assessing Data

Assess data visually and programmatically for quality and tidiness issues using pandas.

- **Visual Assessment**

  Once I opened twitter_archive_enhanced.csv I assessed the data as following:

  - Quality: html tags in source column of twitter archive. Like <a href=""http://twitter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>
  - Quality: None values instead NaNs valus.

- ## Programmatic Assessment

  - **Quality issues**

    - **Twitter Archive file (df_arch)**

- Timestamp and retweeted_status_timestamp are object type instead of datetime.
- Source is HTML format.
- There are records have more than one dog stage.
- doggo, floofer, pupper, and puppo have missing values with "None" instead of NaN
- There are 181 retweeted tweets
- There are many columns in this dataframe making it hard to read, and some will not be needed for analysis.

    - **Image Prediction (df_pred)**

- There are 66 images are duplicated.
- The tweet_id column should be dtype object instead of int64.

    - **Twitter API (df_api)**

- we need Just 3 columns id, retweet_count, favorite_count

- **Tidiness issues**

- doggo, floofer, pupper, and puppo are unique columns instead one column "dog_stage"
- Merge all the 3 dataframe to one dataframe based on tweet_id.

Step3: Cleaning Data

This part of the data wrangling was performed in three stages:
Define, Code and Test.
In the first I made a copy for all the dataframes.
These three steps were executed on each of the issues described in the assess section (this included ‘melting’ the dog stages into one column instead of four columns as originally presented in the twitter archive file).

- ### Storing Data

After I clean the data, now we have cleaned and structure data. I stored it into dataframe then I save as csv by panada’s to_csv() function and named as twitter_archive_master.