Twitter Data Wrangling

1. Introduction

WeRateDogs is a twitter accounts with huge number of fans. The account main aim is to post photos of dogs and rate it from 10 and above out 10 because everyone loves dogs. So the data wrangling project consists of gathering, assessing and cleaning data from that account in order to be ready for analysis and reporting.

2. Gathering

The project requires gathering three different data formats from three different sources

2.1 Downloaded Data

The first data frame has been gathered easily form the downloaded file provided using pandas.read_csv () method in the form of csv format

2.2 URL

Using the URL provided the second data frame is downloaded using the OS library for file creation, requests library for downloading the URL and also the pandas library

2.3 API

After the twitter developer had been made the tweepy library has been used in order to connect to twitter's API using: tweepy.OAuthHandler (), set_access_token () and tweepy.API ()

Since they are a group of tweets with unique tweet_id a for loop has been created to get each tweet data

```
file_name_json = 'tweet_json.txt'
error = []
if not os.path.isfile(file_name_json):
    with open(file_name_json,'w') as file_json :
        for id_ in twitter_download['tweet_id']:
            try:
                status = api.get_status(id_ , tweet_mode = 'extended')
                content = status._json
                json.dump(content,file_json)
                file_json.write('\n')
            except:
                print('Error found for id {}'.format(id_))
                error.append(id_)
```

(As shown above json library has been used in order to download each tweet data in the file created)

3. Assessing

3.1 Visual Assessment

Visual assessment has been conducted for the three datasets using different methods: df.head () or checking the Excel file itself

3.2 Programmatic Assessment

For the programmatic assessment a variety of methods has been used since it's more helpful for detecting data problems: df.info() , df.describe() , df.sort_values() , df.duplicated().sum() , df.value_counts()

```
In [128]: twitter_download[twitter_download['rating_denominator'] < 10]
```

Out[128]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
313	835246439529840640	8.352460e+17	26259576.0	2017-02-24 21:54:03 +0000
516	810984652412424192	NaN	NaN	2016-12-19 23:06:23 +0000
2335	666287406224695296	NaN	NaN	2015-11-16 16:11:11 +0000

```
In [129]: twitter_download[twitter_download['rating_denominator'] > 10]
```

(Picture shows a method to detect incorrect values of 'rating_denominator' column)

3.3 Documentation

The final step in the assessment process is to document both quality and tidiness issues found in all data frames in order to be cleaned afterwards.

Quality Issues	Tidiness Issues
1. Unneeded columns in downloaded data frame	1. Dogga, floofer, pupper, puppo should be in one column called type
2. Timestamp data type should be date and found string	2. Predictions (Only dog photos) should be in one column for better analysis
3. Rating_denominator data entered wrongly	3. Json_df and twitter_download should be one data frame
4. Rating_numerator data entered wrongly	
5. Some records are retweets	
6. Inconsistent format in name column	
7. Some names are wrongly extracted like (A , The)	
8. Some data doesn't have images rows should be 2356 found 2075	
9. Inconsistent format for P1,P2,P3 should all be lower or upper case	
10. Non descriptive column names	
11. Some data are missing should be 2356 found 2331	
12. Some images aren't for dogs	
13. images that aren't dogs are still in image_clean data frame	
14. Inconsistent breed format should be all capitalized	

(Group of quality and tidiness issues found in all datasets)

4. Cleaning

4.1 Define

First before coding the cleaning steps are defined clearly in order to keep to track of the quality issues found

4.2 Code

All quality and tidiness issues are addressed one by one solving it programmatically and avoiding any manual cleaning to avoid mistakes and for time saving purposes

4.3 Test

All data frames has been tested to ensure the succession of the cleaning codes and to avoid any upcoming errors

Finally the whole process is iterative so the data frames has been reassessed again and cleaned if needed to reach the optimum level of data wrangling and facilitate data analysis and visualization processes afterwards.