Twitter Data Wrangling Exercise

In this report, I have summarized my data wrangling work on the project that I have been working on recently. In this project, I have used a file containing previously pulled archived Tweets from @WeRateDogs twitter handle, used python's requests library to download a previously saved Image prediction file, and finally, I used Twitter's API to pull the tweet details like, Retweet counts and Favorite counts to analyze the data.

Exercise 1:

At First, I imported the Tweepy library and queries Twitter's API for JSON data for each tweet ID in the Archive. Then I used JSON library to read the data and loaded it into a Data Frame.

Then I Cleaned up Data frame a bit to make it useful. In this Dataset, I just updated the Tweet_ID data type to an Integer type.

Code

Read the json from the file and create a dataframe

```
all_data=[]
my_demo_list = []
for json_file in open('tweet_json.txt', encoding='utf-8'):
    all_data.append(json.loads(json_file))
```

```
for each_dictionary in all_data:
        tweet_id = each_dictionary['id']
        text = each_dictionary['full_text']
        favorite_count = each_dictionary['favorite_count']
        retweet_count = each_dictionary['retweet_count']
        created_at = each_dictionary['created_at']
        my_demo_list.append({'tweet_id': str(tweet_id),
                             'text': str(text),
                             'favorite_count': int(favorite_count),
                             'retweet_count': int(retweet_count),
                             'created_at': created_at,
                            })
        #print(my_demo_list)
        tweet_json = pd.DataFrame(my_demo_list, columns =
                                  ['tweet_id', 'text',
                                   'favorite_count', 'retweet_count',
                                   'created_at'])
```

```
tweet_json.head(2)
```

	tweet_id	text	favorite_count	retweet_count	created_at
0	666020888022790149	Here we have a Japanese Irish Setter. Lost eye...	2385	458	Sun Nov 15 22:32:08 +0000 2015
1	666029285002620928	This is a western brown Mitsubishi terrier. Up...	120	42	Sun Nov 15 23:05:30 +0000 2015

Clean

```
tweet_json_clean = tweet_json.copy()
```

Define

1. Change the tweet_id datatype to Int64

Code

```
#change ID format to join it with Twitter_data
tweet_json_clean.tweet_id = tweet_json_clean.tweet_id.astype(np.int64)
```

Test

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2059 entries, 0 to 2058
Data columns (total 5 columns):
tweet_id          2059 non-null int64
```

Exercise #2

Then I used the Pandas Library to read the Twitter-Archive-Enhanced file into a dataframe.

Then I cleaned up the Dataframe. Here is the list of updated I made to the twitter_data_clean dataframe.

1. Combined the 4 Dog stages fields into one and drop the 4 fields.

2. Dropped Retweets and Replies from the Dataframe.

3. Removed unwanted columns from the Dataframe.

4.Cleand-up Ratings. (Made sure that the Rating Denominator is a multiple of 10; if not see if a wrong rating was picked up and clean it up)

```
4a. Removed record if a rating is not available (1 record found)

4b. Picked the correct Ratings wherever available. (3 records fond. It was mostly the l
ast 4 characters in the tweet contains correct ratings)
```

5. The Archived dataset had not captured the ratings with a decimal point, I cleaned it up to make sure the correct ratin g was picked up.

6. When the Rating denominator was more than 10 (multiple dogs' rates in the same tweet), I applied a logic to average it out. For example, if the rating was 60/40, I made it 15/10. 13/10 stayed the same and so on.

Exercise #3

In this exercise, I use the Requests library to read the image-predictions.tsv file. This file contained the predicted Dog Breeds for each tweet.

Every Tweet in the dataset had three predictions, so, I had to clean up the dataset to remove the False/Wrong predictions, pick the most likely value and make the dataset easier to use by formatting it properly.

I made the following changes the Dataframe:

1. Transposed (Make the prediction into one column instead of multiple columns) the Predicted fields to make the data cleaner and easier to use.

2. Removed wrong/False prediction (e.g. in one instance "paper Towel" was predicted as a dog breed.)

3. Removed Duplicates.

4. Only kept the highest Prediction confidence record for each Image and dropped the rest of the records.

Exercise #4:

Finally, I merged all three Data frames into one and saved the data to a master file for further analysis.

Insights

At last, I used the cleaned data to find the Top Tweets based on the number of retweets and Favorites. I plotted the Top 20 Dog Breeds based on Average Rating and Also, Looked at the user behavior on the twitter @Dog_Rates handle to see how the popularity of the twitter handle is trending over time.