Data Wrangling Project Report

Quality Issuess:

After investigating the data in “twitter-archive-enhanced.csv” the following assessment issues encountered:

```
tweet_id                       0
in_reply_to_status_id       2278
in_reply_to_user_id         2278
timestamp                      0
source                         0
text                           0
retweeted_status_id         2175
retweeted_status_user_id    2175
retweeted_status_timestamp  2175
expanded_urls                 59
rating_numerator               0
rating_denominator             0
name                           0
doggo                          0
floofer                        0
pupper                         0
puppo                          0
dtype: int64
```

1. the following columns have NULL values: a. in_reply_to_status_id has NULL values
   b. in_reply_to_user_id has NULL values
   c. retweeted_status_id has NULL values
   d. retweeted_status_user_id has NULL values
   e. retweeted_status_timestamp has NULL values
   f. expanded_urls has NULL values
2. Name column has invalid names(e.g a , an and None)
3. "doggo" "floofer" "pupper" "puppo" columns have "None" value
4. Timestamp is string.
5. The url of the tweet's source is written instead of the source itself.(e.g.Twitter for iPhone)
6. 1455 rows have rating_numerator with invalid data(greater than 10)
7. 20 rows have rating_denominator with invalid data(greater than 10)
8. one tweet has no denominator value

Tidiness Issuess:

Using data.info() the following observations encountered:

1. Dog type is represented in 4 columns.(e.g doggo, floofer, pupper and puppo).

- **Define**: write a function that merges the dogs type into a new single column.

```
def get_dog_type(row):

  if row['doggo']!='None':

    return 'doggo'

  elif row['floofer']!='None':

    return 'floofer'

  elif row['pupper']!='None':

    return 'pupper'

  elif row['puppo']!='None':

    return 'puppo'

  else :

    return 'None'
```

- **Code**:apply the function on the dataframe using dataframe.apply()

```
clean_data['Type']=clean_data.apply(get_dog_type,axis=1)
```

- **Test** :Dispaly clean_data['Type']

```
clean_data['Type']
```

2. The results of the prediction algorithms is represented in 9 columns.
   Define: Remove the inappropriate columns.
   Code: Apply the drop on the dataframe using dataframe.drop()
   Test :Dispaly clean_data.columns
   ```
   clean_data.drop(['p1_dog', 'p1_conf', 'p2_dog', 'p2_conf', 'p3_dog',
   'p3_conf'],axis=1,inplace=True)
   clean_data.columns
   ```

Quality Issuess:

1: Handling NULL values

- Find the total number of rows in the data set.
  clean_data.tweet_id.values.shape[0]
  2356
- Find the total number of null values in the dataset.
  clean_data.isnull().sum()

```
tweet_id                         0
in_reply_to_status_id         2278
in_reply_to_user_id           2278
timestamp                        0
source                           0
text                             0
retweeted_status_id           2175
retweeted_status_user_id      2175
retweeted_status_timestamp    2175
expanded_urls                   59
rating_numerator                 0
rating_denominator               0
name                             0
jpg_url                        281
img_num                        281
p1                             281
p1_conf                        281
p1_dog                         281
p2                             281
p2_conf                        281
p2_dog                         281
p3                             281
p3_conf                        281
p3_dog                         281
favorite_count                1179
retweet_count                 1179
Type                             0
```

The output from the isnull() function we observe that :

- 2278 tweets have no in_reply
- 2175 tweets have no retweets
- 281 tweets images have to probability
- 1179 tweets have no favorite_count or retweet_count

Code:

clean_data.drop(['in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id','retweeted_status_user_id','retweeted_status_timestamp','expanded_urls'],axis=1,inplace=True)

Test: check the columns names:

4. **Timestamp is string.**

Define: convert the Time stamp into date time data type

Code: use the pandas.to_datetime(series, format='%Y%m%d', errors='ignore')

```
clean_data['timestamp']=clean_data['timestamp'].apply(pd.to_datetime)
```

Test :Dispaly the dtype of the column

```
clean_data.info()
```

5. **The url of the tweet's source is written instead of the source itself.(e.g.Twitter for iPhone):**

Define: replace the source of url with the source of the tweet

Code: develope a function to replace the source of url with the source of the tweet

```
def extract_tweet_source(row):

   if(row=='<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>'):

      return 'Twitter for iPhone'

   if(row=='<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>'):

      return 'Vine - Make a Scene'

   if(row=='<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>'):

      return 'Twitter Web Client'

   if(row=='<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>'):

      return 'TweetDeck'

   else:

      return''

clean_data['tweet_source']=clean_data['source'].apply(extract_tweet_source)
```

Test :Dispaly the new column values

```
clean_data['tweet_source'].value_counts()
```

6. **1455 rows have rating_numerator with invalid data(greater than 10)**

    Define: Modify the numerator values to 10

    Code: Develope a function to replace the value of numerator to 10

    ```
    clean_data['num'] = clean_data['rating_numerator'].apply(lambda x: 10 if x >10 else x)
    ```

    Test :Dispaly the numerator column values

    ```
    clean_data['num'].value_counts()
    ```

7. **Issue 7:20 rows have rating_denominator with invalid data(greater than 10)**

    Define: Modify the rating_denominator values to 10

    Code: Develope a function to replace the value of numerator to 10

    ```
    clean_data['denominator'] = clean_data['rating_denominator'].apply(lambda x: 10 if x >10 else x)
    ```

    Test :Dispaly the numerator column values

    ```
    clean_data['denominator'].value_counts()
    ```

8. **Issue 8:one tweet has no denominator value**

    Define: Modify the rating_denominator values to 10

    Code:Modify it manually

    ```
    clean_data.loc[313,'rating_denominator']=10
    ```

    Test :Dispaly the row

    ```
    clean_data[clean_data['tweet_id']==835246439529840640]
    ```