Data Wrangling – We Rate Dogs

Data Gathering

Data for the project comes from three sources:

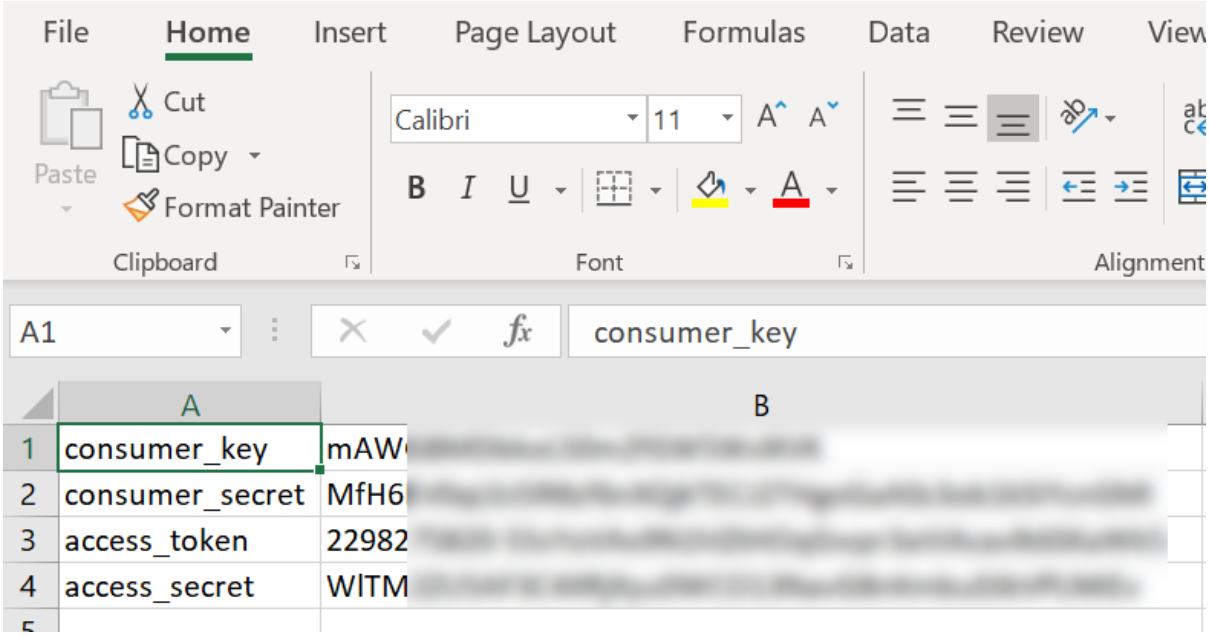
The first dataset – twitter-archive – is downloaded manually and is relatively easy to import. This was done by clicking on the hyperlink provided to me. Once downloaded, I used pandas.read_csv to load and read its data.

The next dataset is downloaded programmatically using the Requests package. The package takes a URL and saves the response to a variable which is then written/saved to a file. Again, data is then loaded into a dataframe using pandas library.

The final dataset was more complicated to download. The connection to the twitter API was done using tweepy. The response was written to data/tweet_json.txt using the json library. If data/tweet_json.txt didn't exist, the code will fetch tweet data. This data will take sometime to download. Fortunately, tqdm library is used to keep track of downloading progress. If data/tweet_json.txt already existed then this step is skipped.

Again, pandas library is used to load data into dataframe.

To use the Twitter API, please use your own keys.



The following files and folder will be produced during data gathering and wrangling.

Name	Status	Date modified	Type	Size
.ipynb_checkpoints		14/11/2018 23:32	File folder	
wrangled		08/12/2018 10:22	File folder	
api-details A		23/10/2018 04:14	Microsoft Excel Co...	1 KB
expanded_urls_exist B		29/11/2018 15:59	Text Document	140 KB
failed_ids.pkl C		08/12/2018 10:21	PKL File	1 KB
image-predictions.tsv D		08/12/2018 10:21	TSV File	328 KB
predictions_jpg_url_exist E		29/11/2018 10:44	Text Document	101 KB
tweet_json F		08/12/2018 10:21	Microsoft Excel Co...	78 KB
tweet_json G		05/11/2018 06:40	Text Document	10,380 KB
twitter-archive-enhanced H		23/10/2018 07:47	Microsoft Excel Co...	895 KB

This is where I will store cleaned data

- Folder **wrangled:** this is where I will store master clean data.
- **A**: This is where Twitter API will be stored. Please your own APIs.
- **B**: storing expanded_urls that can be opened or not.
- **C**: storing twitter_ids that can't be fetched.
- **D**: programmatically downloaded file.
- **E**: storing predictions'jpg_url that can opened or not.
- **F**: tweet_json excel file for visual assessment.
- **G**: json data from twitter API. It is a large file.
- **H**: manually downloaded file.

Data Assessment and Data Cleaning

I will not be discussing data assessment and data cleaning at length here. For that, please refer to wrangle_act.ipynb.

Instead I will tabulate issues I encountered during data assessment. This is done below:

Table	**#**		**Issue**
df_archive_clean			
	1		in_reply_to_status_id has only 78 non-null float
	2		in_reply_to_user_id has only 78 non-null float
	3		retweeted_status_id has only 181 non-null float
	4		retweeted_status_user_id has only 181 non-null
	5		retweeted_status_timestamp has only 181 non-null
	6	a	NaN is represented by "None" string in doggo,floofer,pupper,and puppo
		b	There many "None objects" but the count is still 2,356

	7	a	There is rating numerator and denominator in text column
		b	There is a 'disjoint' max in both rating numerator(1776) and denominator(170)
	8	a	Name of dog generally is in the first sentence.
		b	First character of name of dog is a capital.
		c	There is None in name column
		d	Name is missing even though there is a name in text column.
		e	There are compound names in text column that are not captured in name column
		f	There is a dog type in text column
	9		There are redundant double data urls in `expanded_urls' column
	10		There is a shortened url in text column.
	11		Can we just use df_predictions's tweet_id? Is it a subset of df_archive's tweet_id?
	15		Can the jpg_url link be opened?
	16		Can expanded_urls link be opened?
	17		Remove any tweet ids in the `df_archive_clean` table that aren't in the `df_predictions_clean` table
	25		short_url == expanded_url ???
	19		tweet_is is an int64. Change this to string object.
	20		timestamp is a string. Change this to datetime object.
	21		change dog_type to categorical
	24		retweet_count and favorite_count is not the same size as tweet_id
df_predictions_clean			
	18		Remove non-shared ids from df_predictions_clean table.
	22		tweet_is is an int64. Change this to string object.
	23		change prediction_order to categorical
Tidiness			
	12		Redundant columns: p1,p2,p3. p1_conf,p2_conf,p3_conf. p1_dog,p2_dog,p3_dog.
	13		Can we just merge df_reponse to df_archive?
	14		Multiple columns contain the same type of data: doggo, floofer, pupper, puppo

Saving and Loading Clean Data

After you wrangle data, save it to data/wrangled folder as shown below:

6. Save Cleaned Data

```
df_archive_clean.to_csv('data/wrangled/twitter_archive_master_index_false',index=False)
df_predictions_clean.to_csv('data/wrangled/predictions_master_index_false',index=False)
```

ıts > nanodegree > DAND > 4_Data_Wrangling > Project >

Name	Status	Date modified	Type	Size
predictions_master_index_false		08/12/2018 10:22	File	589 KB
twitter_archive_master_index_false		08/12/2018 10:22	File	666 KB

And when you need to use this data for analysis and visualization, load them into pandas dataframe. Don't forget to do data conversion. This helps in reducing memory used if data is extremely large. The screenshot is shown below:

A. Loading Data

```
df_archive_c = pd.read_csv('data/wrangled/twitter_archive_master_index_false')
df_predictions_c = pd.read_csv('data/wrangled/predictions_master_index_false')
```

```
df_archive_c.tweet_id = df_archive_c.tweet_id.astype('str')
df_archive_c.timestamp = pd.to_datetime(df_archive_c.timestamp)
df_archive_c.dog_type = df_archive_c.dog_type.astype('category')
```

```
df_archive_c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1967 entries, 0 to 1966
Data columns (total 12 columns):
tweet_id              1967 non-null object
timestamp             1967 non-null datetime64[ns]
source                1967 non-null object
text                  1967 non-null object
expanded_urls         1967 non-null object
rating_numerator      1967 non-null float64
rating_denominator    1967 non-null float64
name                  1363 non-null object
dog_type              321 non-null category
retweet_count         1967 non-null float64
favorite_count        1967 non-null float64
short_url             1967 non-null object
dtypes: category(1), datetime64[ns](1), float64(4), object(6)
memory usage: 171.2+ KB
```

```
df_predictions_c.tweet_id = df_predictions_c.tweet_id.astype(str)
df_predictions_c.prediction_order = df_predictions_c.prediction_order.astype("category")
```

```
df_predictions_c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5901 entries, 0 to 5900
Data columns (total 7 columns):
tweet_id            5901 non-null object
jpg_url             5901 non-null object
img_num             5901 non-null int64
prediction_order    5901 non-null category
prediction          5901 non-null object
confidence          5901 non-null float64
dog                 5901 non-null bool
dtypes: bool(1), category(1), float64(1), int64(1), object(3)
memory usage: 242.2+ KB
```