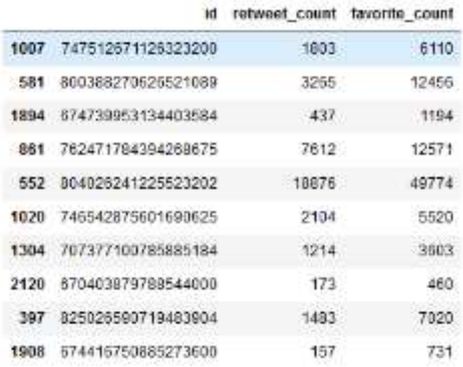
Documentation for Data Wrangling Steps

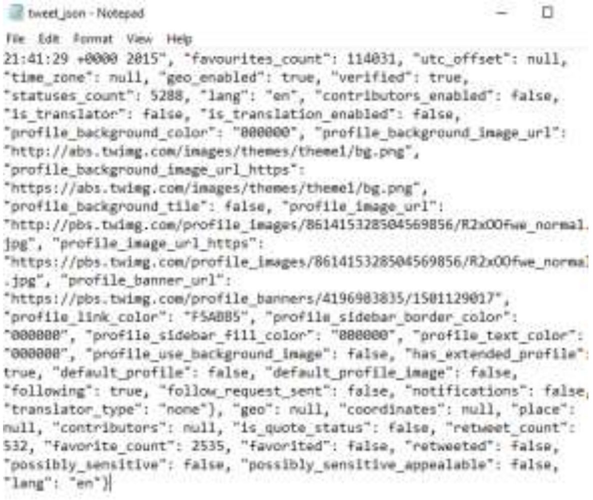
Data wrangling can be broken into three phase : gathering, assesing, cleaning. This document is about the report regarding the process.

1. Gathering Phase
   Gathering phases is required to obtain the data required to data analysis. There are three documents in this project, twitter_archive_enhanced.csv, image_predictions.tsv, tweet_json.txt. , Twitter_archive_enhanced.csv is given from the project and can be downloaded manually. Image_predictions.tsv is in tsv format is downloaded using python requests modul. Tweet_json.txt is in json format and required twitter api to gather the data. In addition, to acquire the data required, information regarding relevant tweets_id is provided using twitter_archive_enhanced.csv data. Then, the file is imported to python using Pandas library. Twitter_archive_enhanced.csv is already open to get the tweets_id, image_predictions.tsv is opened using “tab” as separator for the data, Tweet_json.txt is opened using read_json and set the lines parameter to True to read file line by line. Only tweets_id, retweet, and favorite count is used. The other columns in tweet_json.txt is dropped.
2. Asseing Phase
   Assesing phase is done in two steps, visually and programmatically. Visually is done by seeing the data visually. Example of visual assesing is done in Figure 2 and Figure 2.



	id	retweet_count	favorite_count
1007	747512671126323200	1803	6110
581	800388270626521089	3255	12456
1894	674739953134403584	437	1194
861	762471784394268675	7612	12571
552	804026241225523202	18876	49774
1020	746542875601690625	2104	5520
1304	707377100785885184	1214	3603
2120	670403879788544000	173	460
397	825026590719483904	1483	7020
1908	674416750885273600	157	731

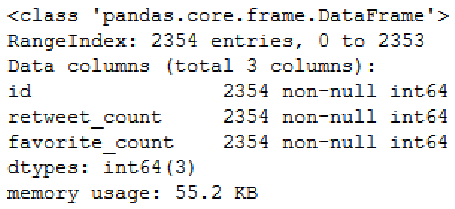
Figure 1 visual asseing tweets table



```
21:41:29 +0000 2015", "favourites_count": 114031, "utc_offset": null,
"time_zone": null, "geo_enabled": true, "verified": true,
"statuses_count": 5288, "lang": "en", "contributors_enabled": false,
"is_translator": false, "is_translation_enabled": false,
"profile_background_color": "000000", "profile_background_image_url":
"http://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_image_url_https":
"https://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_tile": false, "profile_image_url":
"http://pbs.twimg.com/profile_images/861415328504569856/R2xOOfwe_normal.
jpg", "profile_image_url_https":
"https://pbs.twimg.com/profile_images/861415328504569856/R2xOOfwe_normal
.jpg", "profile_banner_url":
"https://pbs.twimg.com/profile_banners/4196983835/1501129017",
"profile_link_color": "F5ABB5", "profile_sidebar_border_color":
"000000", "profile_sidebar_fill_color": "000000", "profile_text_color":
"000000", "profile_use_background_image": false, "has_extended_profile":
true, "default_profile": false, "default_profile_image": false,
"following": true, "follow_request_sent": false, "notifications": false,
"translator_type": "none"}, "geo": null, "coordinates": null, "place":
null, "contributors": null, "is_quote_status": false, "retweet_count":
532, "favorite_count": 2535, "favorited": false, "retweeted": false,
"possibly_sensitive": false, "possibly_sensitive_appealable": false,
"lang": "en"}
```

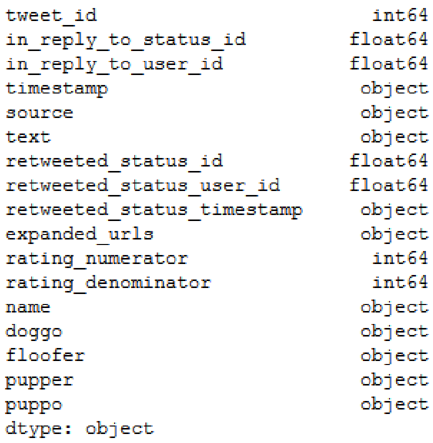
Figure 2 visual assesing tweet_json

Programmatical assesing is done using code to show the data characteristics. Programmatical is done using info module or query some data that is interesting. Example of programmatical assesment is in Figure 3 and Figure 4.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
id                 2354 non-null int64
retweet_count      2354 non-null int64
favorite_count     2354 non-null int64
dtypes: int64(3)
memory usage: 55.2 KB
```

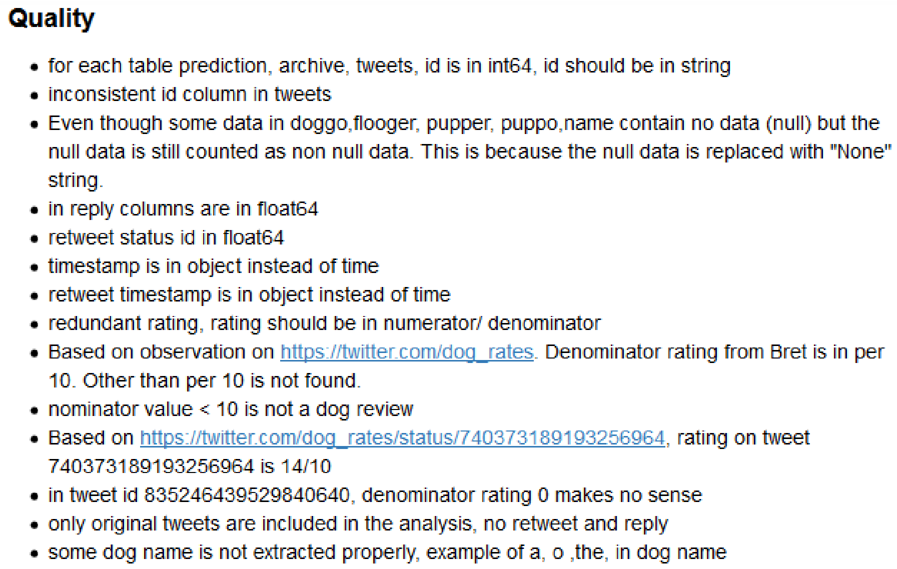
Figure 3 tweet.info



```
tweet_id                       int64
in_reply_to_status_id        float64
in_reply_to_user_id          float64
timestamp                     object
source                        object
text                          object
retweeted_status_id          float64
retweeted_status_user_id     float64
retweeted_status_timestamp    object
expanded_urls                 object
rating_numerator               int64
rating_denominator             int64
name                          object
doggo                         object
floofer                       object
pupper                        object
puppo                         object
dtype: object
```

Figure 4 archive.info

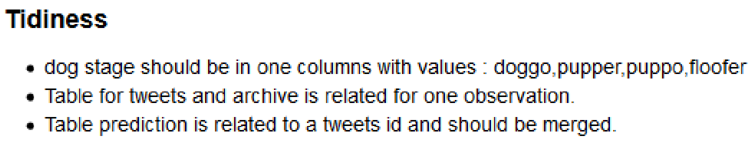
Identified data problem is documented is identified in this phase. Some problems identified in this phase is classified into two groups, quality and tidiness. Quality problem is in Figure 5. Tidiness problem is in Figure 6.



Quality

- for each table prediction, archive, tweets, id is in int64, id should be in string
- inconsistent id column in tweets
- Even though some data in doggo,flooger, pupper, puppo,name contain no data (null) but the null data is still counted as non null data. This is because the null data is replaced with "None" string.
- in reply columns are in float64
- retweet status id in float64
- timestamp is in object instead of time
- retweet timestamp is in object instead of time
- redundant rating, rating should be in numerator/ denominator
- Based on observation on https://twitter.com/dog_rates. Denominator rating from Bret is in per 10. Other than per 10 is not found.
- nominator value < 10 is not a dog review
- Based on https://twitter.com/dog_rates/status/740373189193256964, rating on tweet 740373189193256964 is 14/10
- in tweet id 835246439529840640, denominator rating 0 makes no sense
- only original tweets are included in the analysis, no retweet and reply
- some dog name is not extracted properly, example of a, o ,the, in dog name

Figure 5 Quality problem



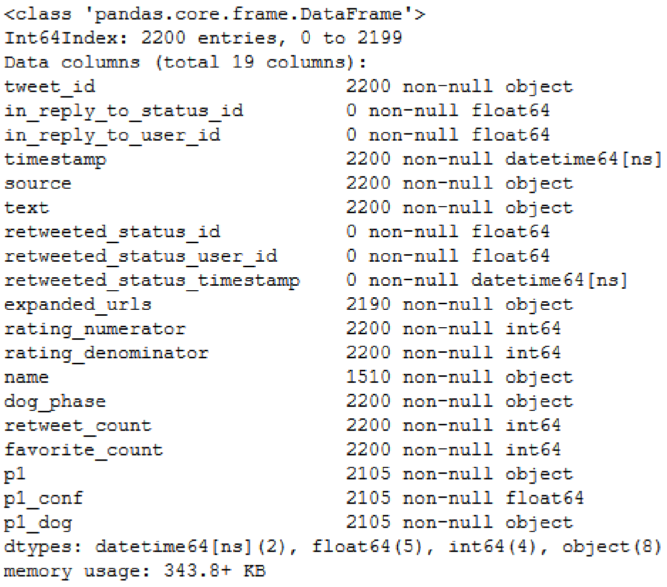
Tidiness

- dog stage should be in one columns with values : doggo,pupper,puppo,floofer
- Table for tweets and archive is related for one observation.
- Table prediction is related to a tweets id and should be merged.

Figure 6 Tidiness problem

Assesment is done to identified the data problems. This data problems are solved in the next phase, cleaning phase.

3. Cleaning Phase
   Cleaning phase is done to solve the problem in assesment phase. Cleaning is done in define-code-test framework. Define is defining the problem and the solution in human words. Code is implementing the solution in python code. Test is validating the code to solve the problems. Cleaning phase solved the quality and tidiness problems identified. The resulting cleaned data information is in Figure 7. Some columns that is not needed for analysis is dropped like model 2 and 3 in prediction, and more. Details regarding cleaning process is in the wrangle_act.ipynb code.



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2200 entries, 0 to 2199
Data columns (total 19 columns):
tweet_id                      2200 non-null object
in_reply_to_status_id         0 non-null float64
in_reply_to_user_id           0 non-null float64
timestamp                     2200 non-null datetime64[ns]
source                        2200 non-null object
text                          2200 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null datetime64[ns]
expanded_urls                 2190 non-null object
rating_numerator              2200 non-null int64
rating_denominator            2200 non-null int64
name                          1510 non-null object
dog_phase                     2200 non-null object
retweet_count                 2200 non-null int64
favorite_count                2200 non-null int64
p1                            2105 non-null object
p1_conf                       2105 non-null float64
p1_dog                        2105 non-null object
dtypes: datetime64[ns](2), float64(5), int64(4), object(8)
memory usage: 343.8+ KB
```

Figure 7 Cleaned data info