Report on the wrangled Data

Introduction

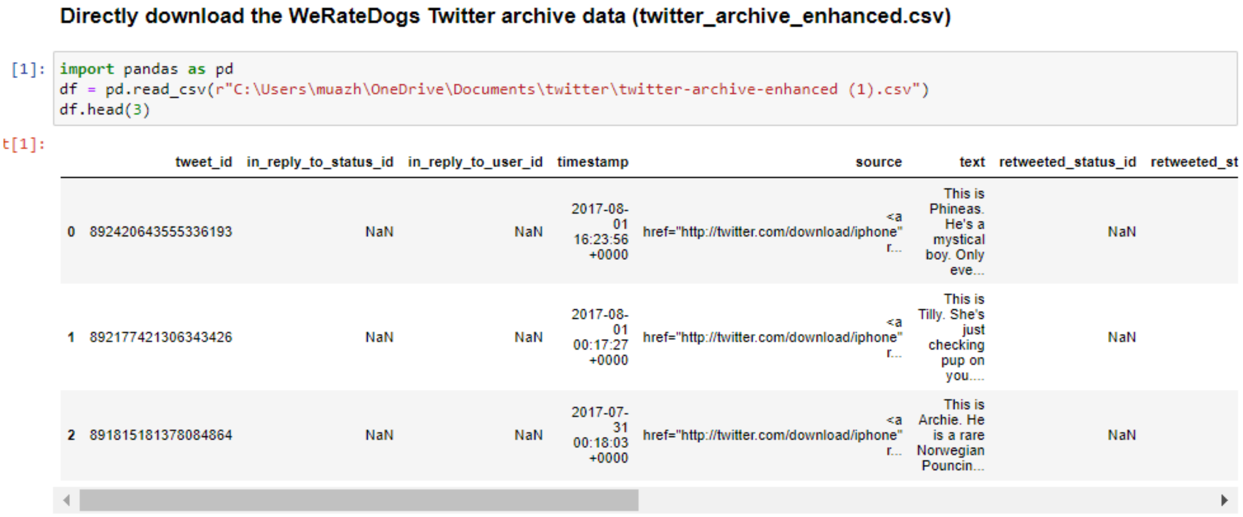
The objective of this project is to wrangle, clean, and generate insight from Twitter data knowns as WeRateDogs which their focus is based on rating dogs and categorizing them. To complete what is being taught in class and to have a better understanding of the wrangling and cleaning lesson in the Nanodegree program, this project was given out as an assignment. Data was collated from different sources csv, tsv, and twitter itself. Wrangling was done with the use of python libraries, and finally visualization was obtained.

What Is Data Wrangling?

To have a better insight from a data as an analyst, wrangling must take place, which is one of the basic steps, in fact the most compulsory step when provided a data as an analyst in a company, school or anywhere else before visualization could occur. A lot of data is untidy and dirty to have a clear view and to meaningful information this process (wrangling and cleaning) must occur. Wrangling can then be defined as a process of cleaning and unifying messy and complex data sets for easy access and analysis.

Gathering Data

As reported earlier in the introduction aspect of this report, data were sourced from three different location. The first one was provided in a CSV file which was given by Udacity and was read using the function pd.read_csv



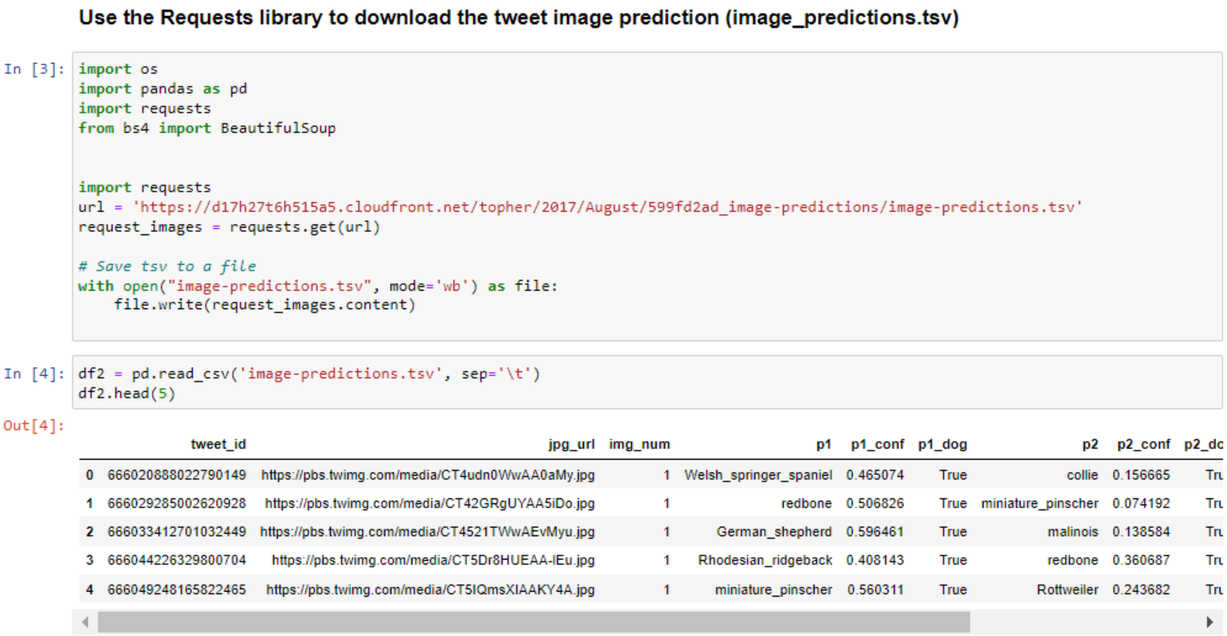
Directly download the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)

```
[1]: import pandas as pd
     df = pd.read_csv(r"C:\Users\muazh\OneDrive\Documents\twitter\twitter-archive-enhanced (1).csv")
     df.head(3)
```

t[1]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_st
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	NaN	
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	NaN	

The second one which is the image_prediction.tsv file which was hosted on Udacity's server and was downloaded using the python requests library. The image below shows how the data was sourced.



Use the Requests library to download the tweet image prediction (image_predictions.tsv)

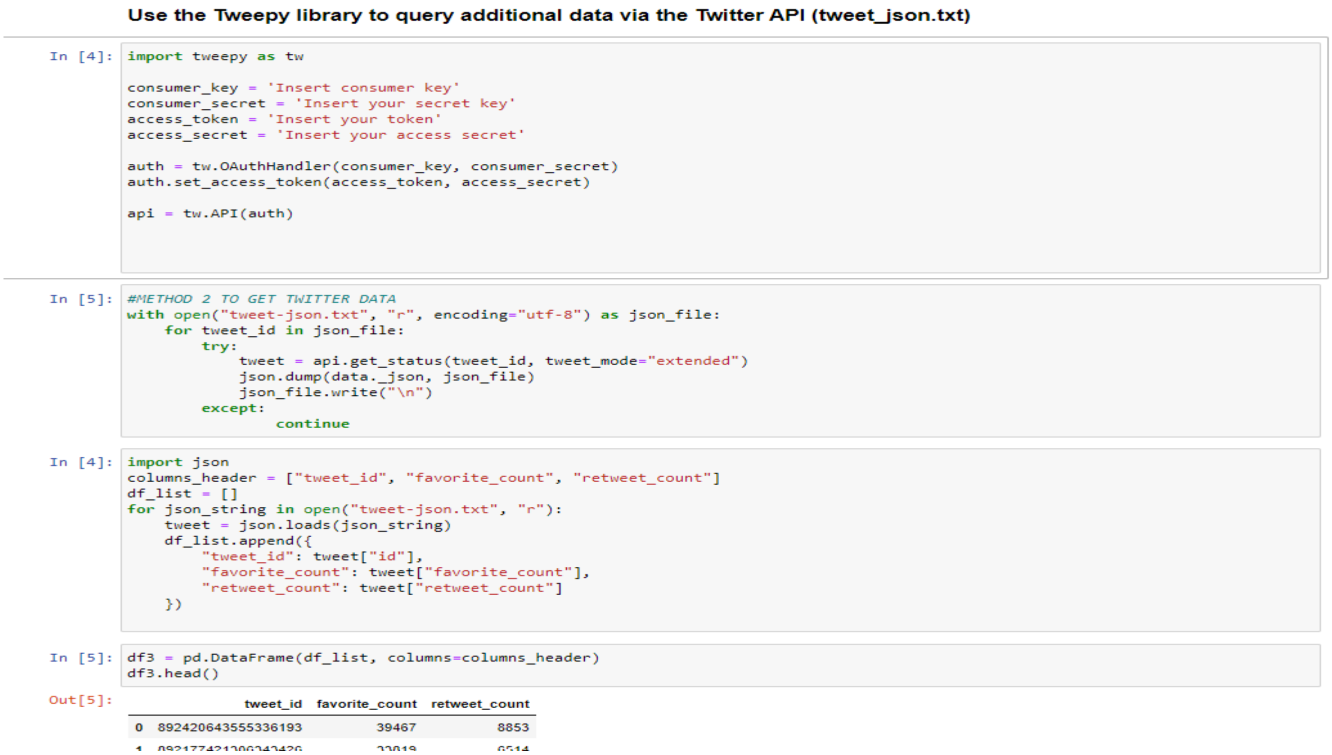
```
In [3]: import os
        import pandas as pd
        import requests
        from bs4 import BeautifulSoup


        import requests
        url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
        request_images = requests.get(url)

        # Save tsv to a file
        with open("image-predictions.tsv", mode='wb') as file:
            file.write(request_images.content)
```

```
In [4]: df2 = pd.read_csv('image-predictions.tsv', sep='\t')
        df2.head(5)
```

Out[4]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dc
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	Tru
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	Tru
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	Tru
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	Tru
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	Tru

The third data was obtained from Twitter with the aid of Twitter API and Tweet JSON file. This data was generated to serve as an additional data to the previous ones. During the process of obtaining the data new columns that was needed was generated which are favorite_count and retweet_count querying.



Use the Tweepy library to query additional data via the Twitter API (tweet_json.txt)

```
In [4]: import tweepy as tw

        consumer_key = 'Insert consumer key'
        consumer_secret = 'Insert your secret key'
        access_token = 'Insert your token'
        access_secret = 'Insert your access secret'

        auth = tw.OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_secret)

        api = tw.API(auth)
```

```
In [5]: #METHOD 2 TO GET TWITTER DATA
        with open("tweet-json.txt", "r", encoding="utf-8") as json_file:
            for tweet_id in json_file:
                try:
                    tweet = api.get_status(tweet_id, tweet_mode="extended")
                    json.dump(data._json, json_file)
                    json_file.write("\n")
                except:
                        continue
```

```
In [4]: import json
        columns_header = ["tweet_id", "favorite_count", "retweet_count"]
        df_list = []
        for json_string in open("tweet-json.txt", "r"):
            tweet = json.loads(json_string)
            df_list.append({
                "tweet_id": tweet["id"],
                "favorite_count": tweet["favorite_count"],
                "retweet_count": tweet["retweet_count"]
            })
```

```
In [5]: df3 = pd.DataFrame(df_list, columns=columns_header)
        df3.head()
```

Out[5]:

	tweet_id	favorite_count	retweet_count
0	892420643555336193	39467	8853
1	892177421306343426	33019	6514

Assessing

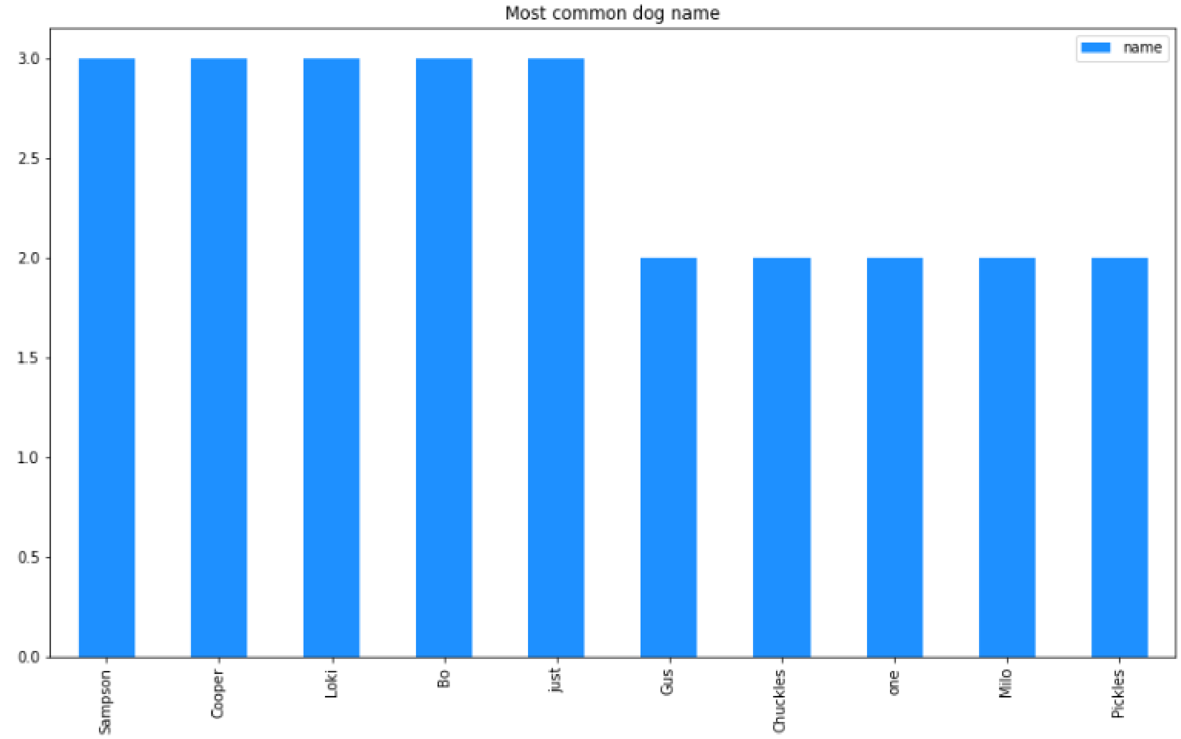
The data were assessed using two methods

1. Data quality issues: This is to check for missing values, duplicates, and incorrect values in the data frame.
2. Tidiness issues: This are structural issues in the data frame.

Both were done using the visual assessment which was done using spreadsheet and notebook while on the other hand the programmatic assessment was done using some specific pandas' functions to have an overview of the data structure. Immediately after assessing the data and it has been cleaned it was then declared ready for the descriptive analysis. The three data set were merge into a single data using the panda's function". merge ()" and the saved as a csv file. Meaningful information was generated during the descriptive analysis phase.

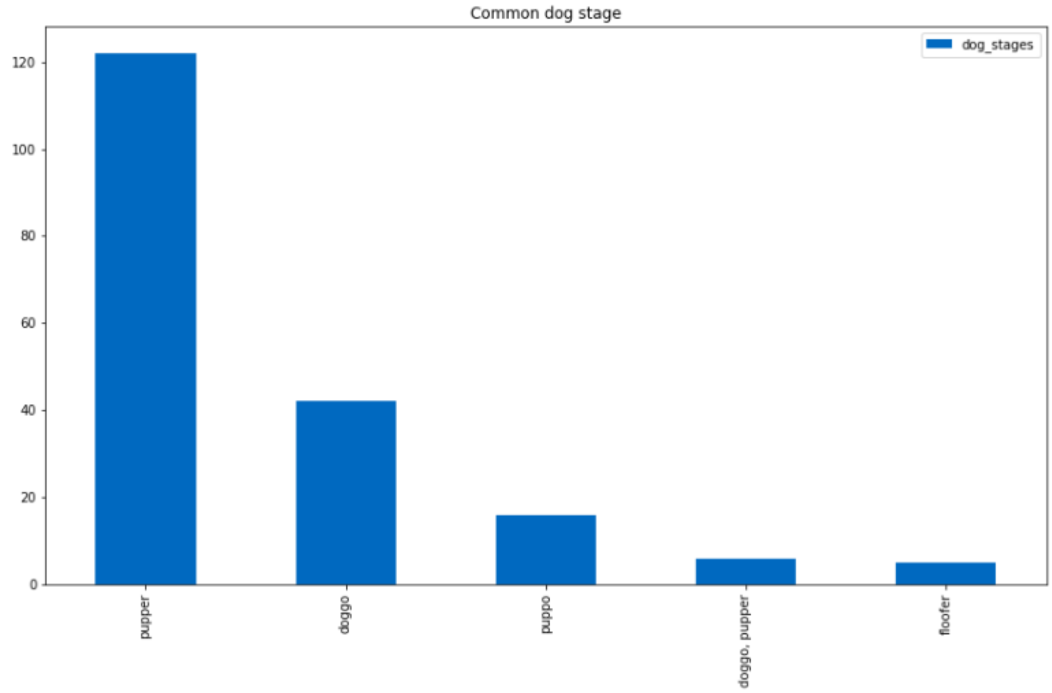
Descriptive Analysis and Visualization

After finishing with the cleaning of the data frame, insight and visualizations were generated to have an overview and a clearer picture of the datasets. Different columns were combined to have meaningful visualization.



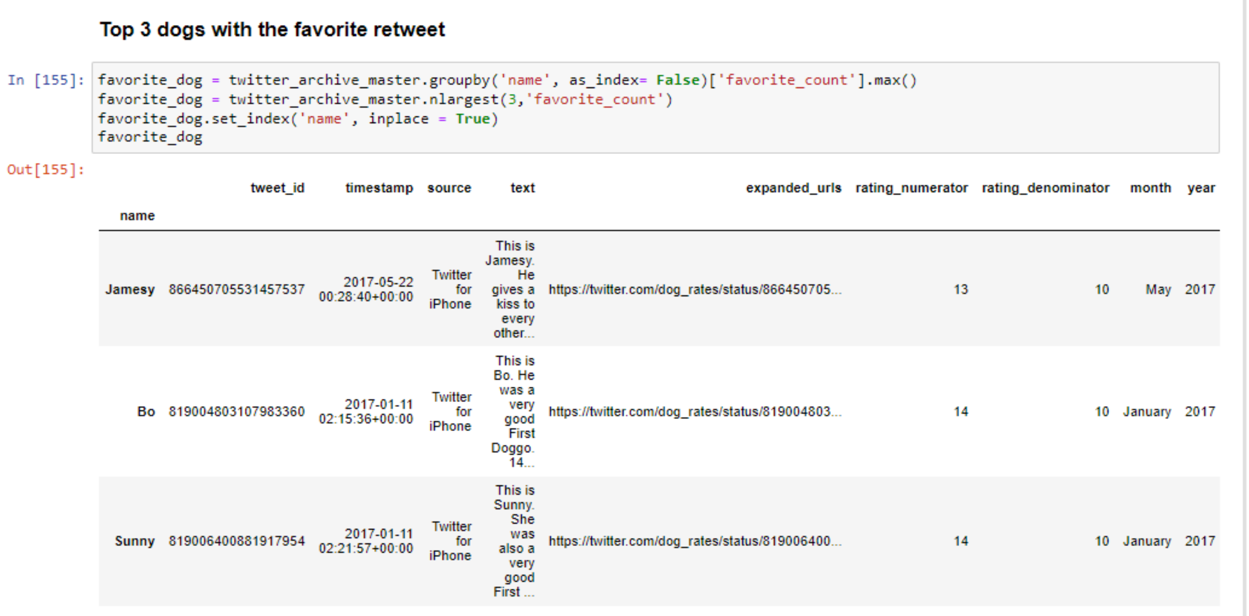
The chart shows the most common dog name which is called Sampson followed by Copper, and Loki

Most common dog stages



The image above depicts that Pupper was the common dog's name in the data set

Top 3 Dogs with the Favorite retweet



Top 3 dogs with the favorite retweet

In [155]:

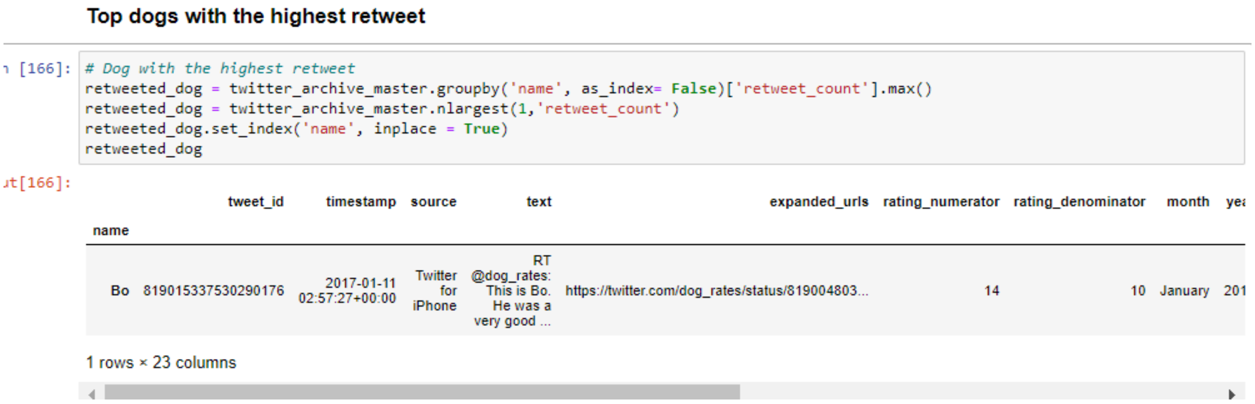
```
favorite_dog = twitter_archive_master.groupby('name', as_index= False)['favorite_count'].max()
favorite_dog = twitter_archive_master.nlargest(3,'favorite_count')
favorite_dog.set_index('name', inplace = True)
favorite_dog
```

Out[155]:

name	tweet_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	month	year
Jamesy	866450705531457537	2017-05-22 00:28:40+00:00	Twitter for iPhone	This is Jamesy. He gives a kiss to every other...	https://twitter.com/dog_rates/status/866450705...	13	10	May	2017
Bo	819004803107983360	2017-01-11 02:15:36+00:00	Twitter for iPhone	This is Bo. He was a very good First Doggo. 14...	https://twitter.com/dog_rates/status/819004803...	14	10	January	2017
Sunny	819006400881917954	2017-01-11 02:21:57+00:00	Twitter for iPhone	This is Sunny. She was also a very good First ...	https://twitter.com/dog_rates/status/819006400...	14	10	January	2017

The image above shows three dogs has the highest favorite count Jamesy, Bo, and Sunny

Top Dog with the highest retweet count



Top dogs with the highest retweet

```
ı [166]: # Dog with the highest retweet
retweeted_dog = twitter_archive_master.groupby('name', as_index= False)['retweet_count'].max()
retweeted_dog = twitter_archive_master.nlargest(1,'retweet_count')
retweeted_dog.set_index('name', inplace = True)
retweeted_dog
```

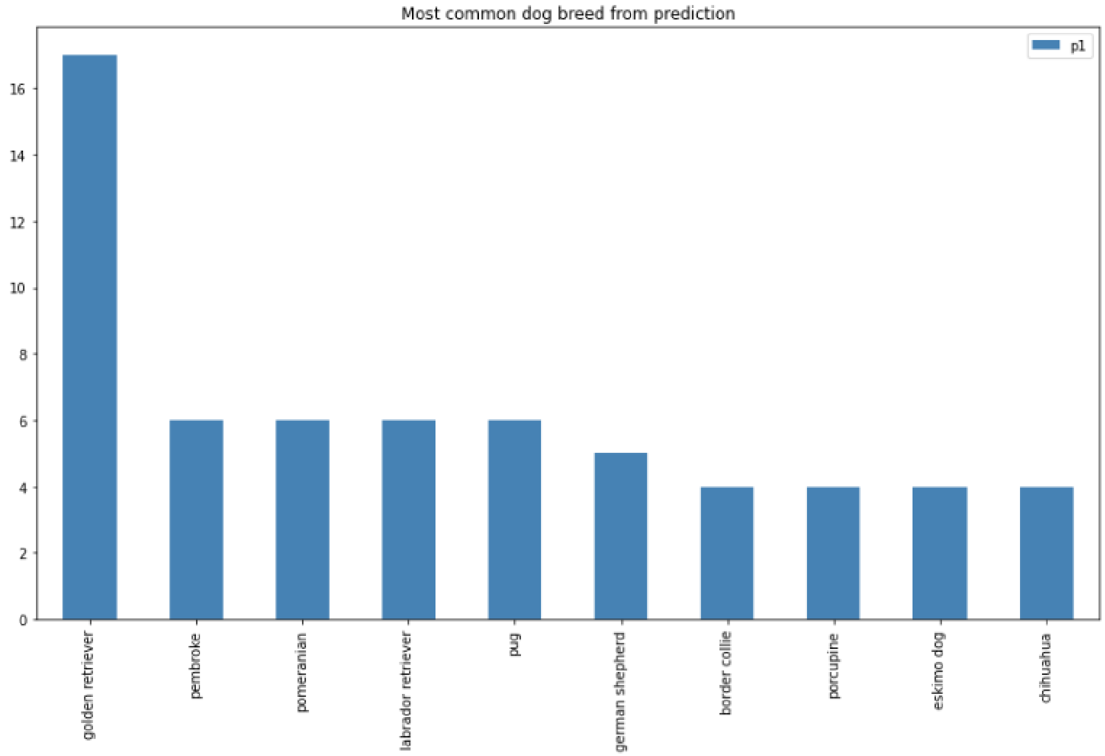
ıt[166]:

	tweet_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	month	yea
name									
Bo	819015337530290176	2017-01-11 02:57:27+00:00	Twitter for iPhone	RT @dog_rates: This is Bo. He was a very good ...	https://twitter.com/dog_rates/status/819004803...	14	10	January	201

1 rows × 23 columns

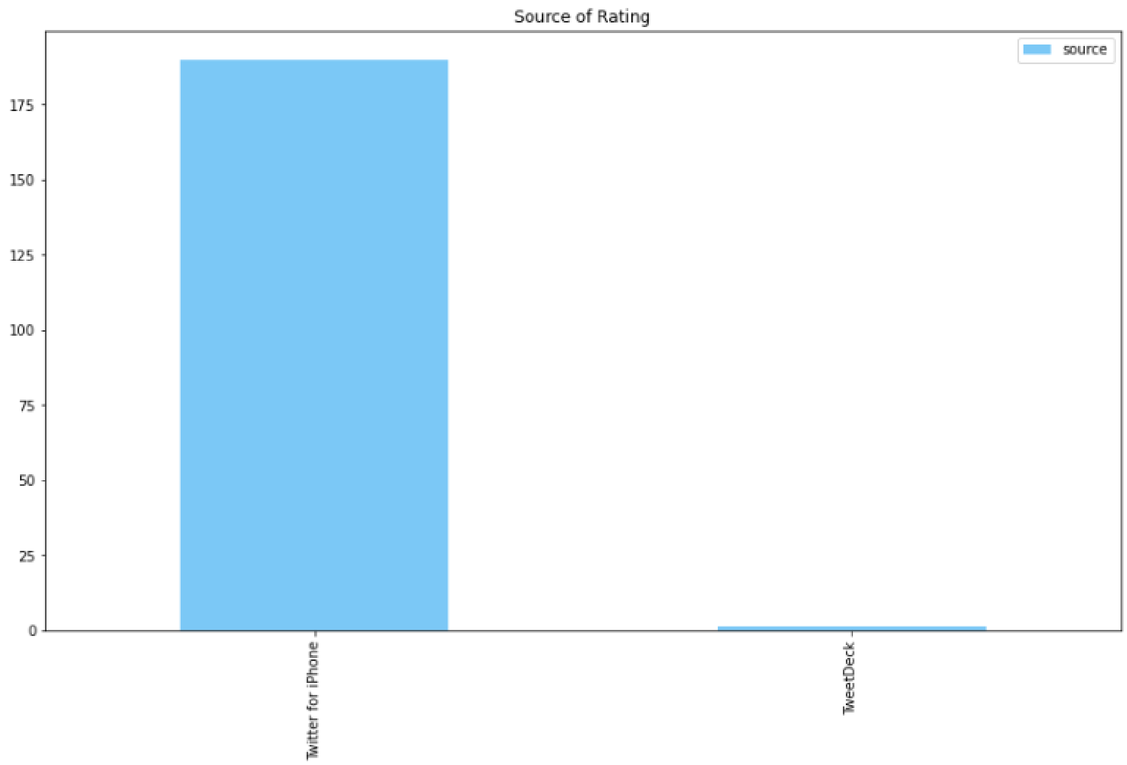
The chart above shows Bo has the highest retweet counts among other dogs

Most common dog breeds



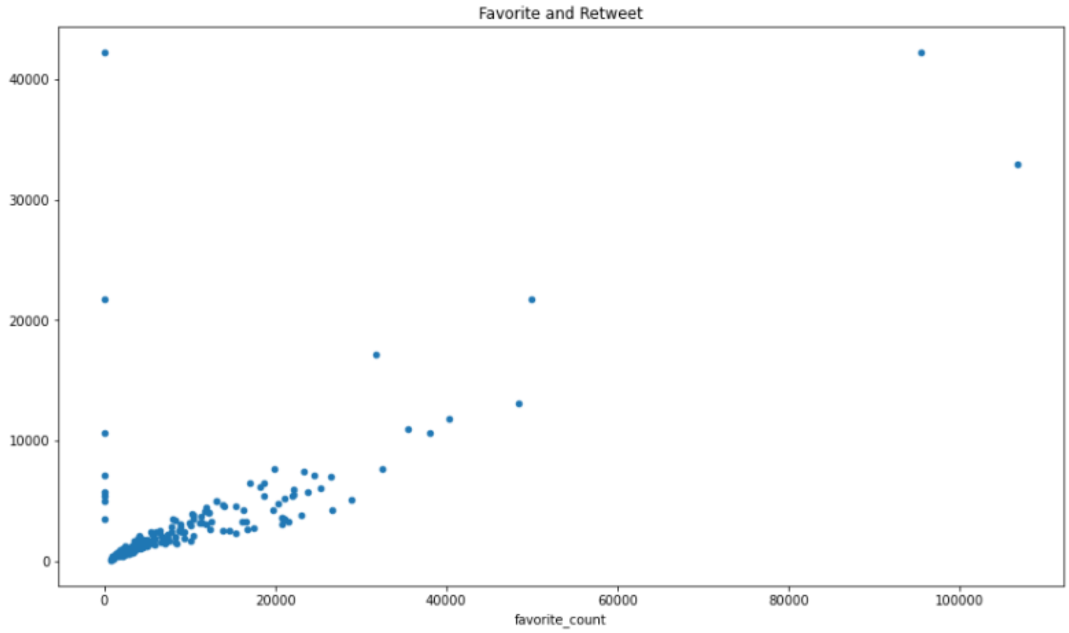
From the chart above golden retriever is the most common dog breed, followed by Pembroke, and Pomeranian.

Source where users are rating from



According to the chart, the most common source from users was iPhone.

Correlations between Favorite count and Retweet counts



A strong and positive correlation occurred between favorite counts and retweet counts