Act Report

Intro:

In this project I gathered, assessed and analyzed sets of data from Udacity and Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. Software used include: Python(pandas, NumPy, requests, tweepy, json), Google Docs.

Files provided ahead include: 'twitter-archive-enhanced-2.csv', image-predictions.tsv, tweet-json.txt. The archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. 'image-predictions' is a file contains prediction result of whether the object in the picture is a dog or not as well as dog breed. It was powered by a neural network that can classify breeds of dogs and ran through every image in the WeRateDogs Twitter. 'tweet-json' was provided as additional file that contains all information in every tweet.

Insight:

We can have an overall insight about statistical results in the following graph. On average, those tweets was retweeted 2912 times, was 'liked' 8440 times, the average quotient, although giving the fact 'those are good dogs' and expect quotients would be way above 1, there are few were rated under 1, make up the average quotient round 1. This also make sense consider 50% quotient is 1.1 or less.

	tweet_id	retweet_count	favorite_count	retweeted_status_user_id	rating_numerator	rating_denominator	quotient
count	1.140000e+03	1140.000000	1140.000000	4.000000e+01	1140.000000	1140.000000	1140.000000
mean	7.386192e+17	2911.939474	8440.280702	4.311139e+09	11.352211	10.481579	1.082615
std	6.733339e+16	4750.252658	11430.204431	5.896539e+08	8.014061	6.949499	0.192088
min	6.660293e+17	16.000000	0.000000	4.196984e+09	1.000000	2.000000	0.200000
25%	6.766059e+17	620.500000	1662.000000	4.196984e+09	10.000000	10.000000	1.000000
50%	7.140304e+17	1440.500000	3946.500000	4.196984e+09	11.000000	10.000000	1.100000
75%	7.909661e+17	3443.250000	10555.750000	4.196984e+09	12.000000	10.000000	1.200000
max	8.918152e+17	56625.000000	107015.000000	7.832140e+09	165.000000	150.000000	3.428571

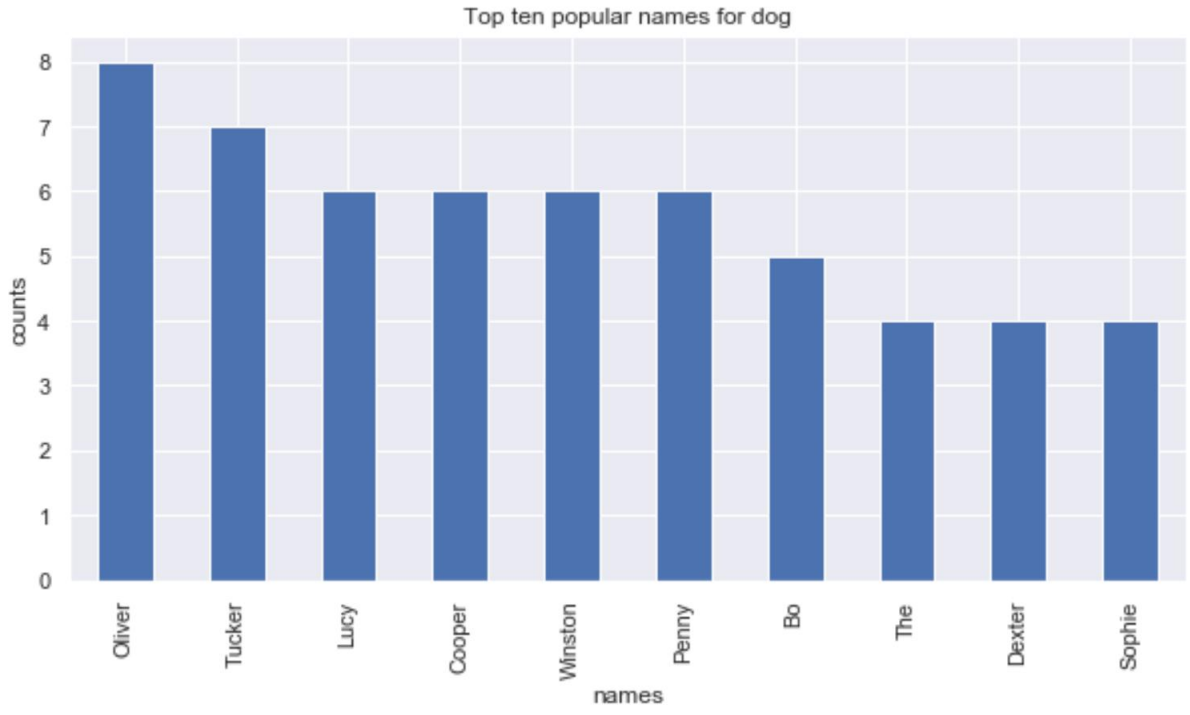
I made a quotient column during wrangling process and attempt to use as one of key metrics, but as I dived deeper, I found out data quality issues:

```
pd.set_option('max_colwidth', 800)
# inspect ones has the least quotient
df.query('quotient == 0.2')
```

	tweet_id	retweet_count	favorite_count	breed	text	retweeted_status_user_id	rating_numerator	rating_denominator	quotient	n
537	722974582966214656	1764	4493	great_dane	Happy 4/20 from the squad! 13/10 for all	NaN	4.0	20.0	0.2	

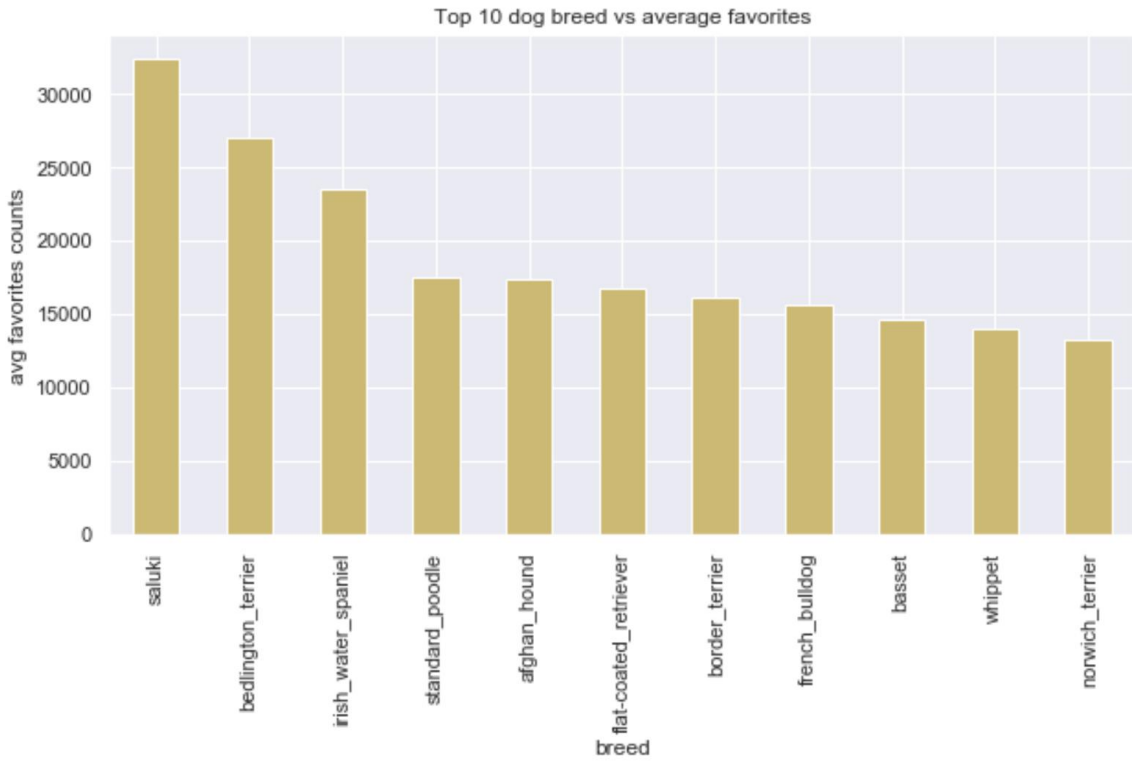
Consider tweet id 722974582966214656 as an example. The text is :"*Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a.*" the correct extraction would be 13/10, but regex will capture both 4/20 and 13/10. We can choose to extract the second group manually, but in other text the correct match might be the first group, there is no good way to extract match correctly except going through one by one manually, which is not practical. Therefor, I decided not to take quotient in as a metric.

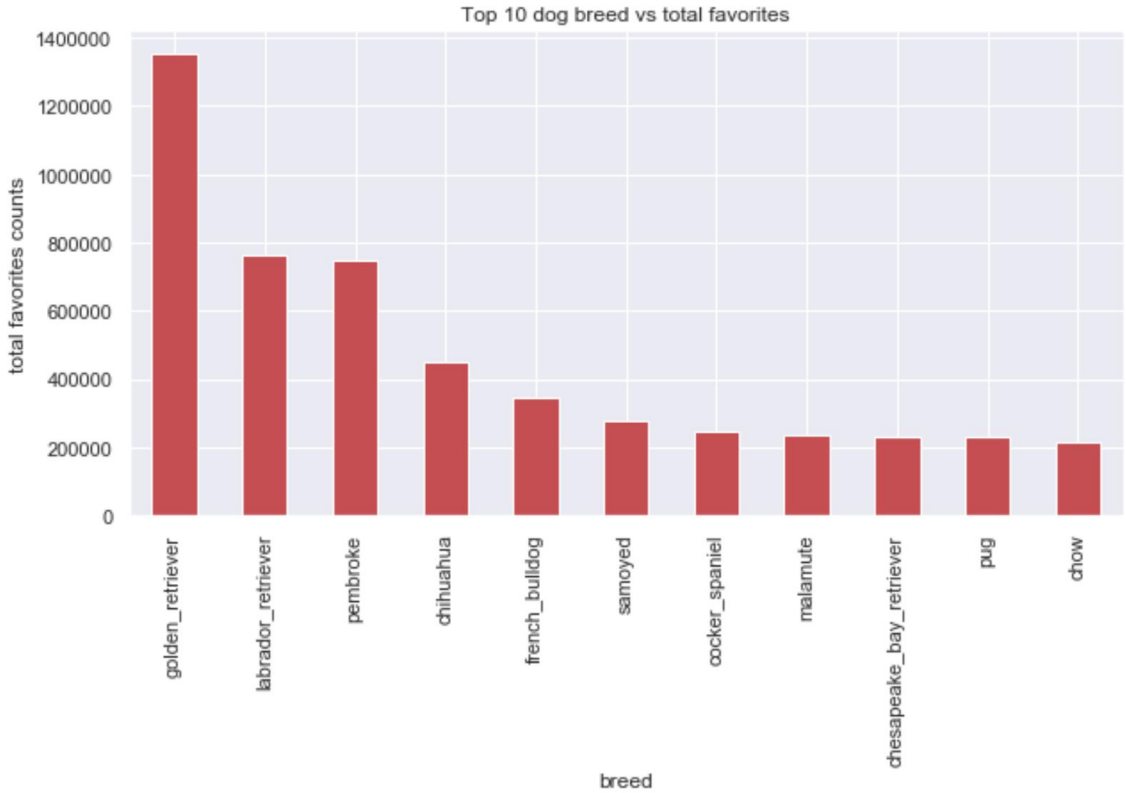
Some interesting insight about dog names:



Here we can see the difference of 'top 10 most favorite dog breeds' when we use different

metrics. On average, every saluki dog post earned 32444 'favorites' as the most popular dog breed, while golden retriever seems to be the most popular dog breed when we compare the total ‘favorites’ earned over time.





Those two graphics leads me to a hypothesis: there is a potential correlationship between retweet_count, favorite_count and possibly different dog stages as well. I decided to build a linear regression model to test out the hypothesis. There were some more wangling needed, steps were:

1. Make mask for records with valid dog stage.
2. Investigate ones with multiple stages.
3. Split records with two dog stage to two dataframes, each with one dog stage
4. Append two dataframes together.
5. Drop records with multiple stage from dfstage then append df1 to dfstage.
6. Set ‘intercept’= 1 and choose a ‘baseline’: ‘pupper’.
7. Fit model and interpret result.

	coef	std err	t	P>|t|	[0.025	0.975]
intercept	3243.9797	783.089	4.143	0.000	1699.417	4788.542
retweet_count	1.5904	0.111	14.279	0.000	1.371	1.810
puppo	5111.4323	2043.147	2.502	0.013	1081.536	9141.328
doggo	1548.7552	1622.554	0.955	0.341	-1651.565	4749.075
floofer	904.3662	3848.761	0.235	0.814	-6686.917	8495.650

From the model we can tell retweet_count and favorite_count has a very strong positive correlationship. Every additional retweet is likely to result more than 1 'favorite'. With everything else being equal, we expect dogs at 'puppo' stage receive 5111 more 'favorites' than dogs at 'pupper' stage(baseline), 'doggo' receive 1549 more 'favorites' than 'pupper', 'floofer' receive 904 more 'favorites' than 'pupper'

Limitation: sample size is not large (197 counts), leads to inaccurate prediction. On the other hand, p-value and scatter plot indicate different dog stage don't have strong correlation with number of 'favorite'.