#Gamergate

Twitter User Similarity

Chelsea Waida
Minying Lu

Background

GamerGate was a Twitter hashtag that appeared in August 2014 when several women in the video game industry were targeted with accusations and threats.
To examine how significant events affect social circles through social media, and whether users are influenced by the opinions and sentiment of others in their social circles, we examine a set of tweets scraped over 72 hours in the midst of this incident.

vaderSentiment Analysis

- Created specifically as a tool for sentiment analysis of social media text
- Lexicon takes into account emoticons, slang, and common idioms to be more accurate for our purposes than a more traditional lexicon
- Sentiment analysis tool provides a score of both sentiment and intensity
- Compound score is our polarity measure

```
This is the worst day ever
{'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249}

Time to get writing! :)
{'neg': 0.0, 'neu': 0.549, 'pos': 0.451, 'compound': 0.5093}
```

Popularity, Polarity Correlation

popularity - total number of times a tweet has been interacted with by users, defined as retweets plus favorites

We predicted that polarity of a tweet, the number of mentions in a tweet, and whether it contained a URL would be highly correlated its popularity, however this was not accurate.

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
Intercept	-3.624e-12	3.29e-14	-110.263	0.000	-3.69e-12 -3.56e-12
Polarity	1.963e-11	4.44e-14	441.795	0.000	1.95e-11 1.97e-11
numMentions	-3.979e-13	1.91e-14	-20.791	0.000	-4.35e-13 -3.6e-13
Has_url	1.448e-12	4.1e-14	35.336	0.000	1.37e-12 1.53e-12

The results from our linear regression show that these correlations, although statistically significant, are not the strong positive correlations we expected.
In fact, the number of mentions a tweet has is actually slightly negatively correlated with its popularity.

Clustering

We wanted to find out the correlation of a user's opinion with the opinion of the popular account the user follows and the trending opinion in the user's social circle. To do this, we decided to cluster users in our dataset depending on the tweets they posted. We used keyword extraction to define the features of the tweet. We then reduced the dimension and use Kmeans to perform clustering.

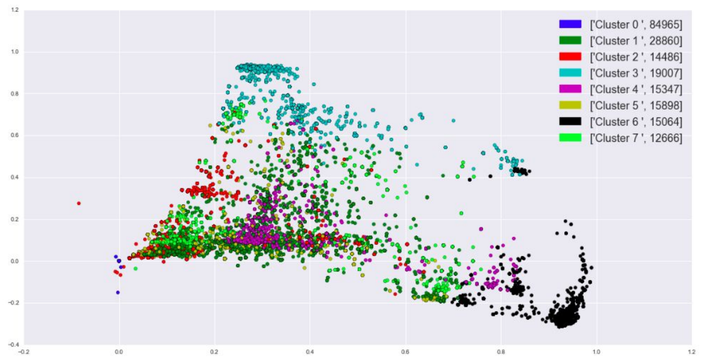
```
stop_words = list(vectorizer.get_stop_words())
stop_words.append('like')
stop_words.append('just')
stop_words.append('tweets')
stop_words.append('twitter')
stop_words.append("gamers")
stop_words.append("gamer")
stop_words.append("games")
stop_words.append("game")
stop_words.append("gaming")
stop_words.append("month")
stop_words.append("hours")
stop_words.append("20")
stop_words.append("want")
stop_words.append("don")
```

To extract keywords, we used TfidfVectorizer. During the process, we found that the default stopword list `'english'` is not very effective in terms of fulfilling our purposes.
Therefore, we added our own stop words on top of the `'english'` stop word list.
As a result, we get the following keyword features. We can see that there are some pretty good keywords such as `'bullying'` and `'threats'`.

```
n_samples: 206293, n_features: 40
[u'adobe', u'anti', u'bad', u'bullying', u'chriswarcraft', u'chsommers',
u'clickhole', u'ethics', u'feliciaday', u'femfreq', u'gawker', u'gg', u'g
roup', u'harassment', u'hate', u'intel', u'journalism', u'media', u'men',
u'movement', u'need', u'nero', u'people', u'piece', u'playdangerously', u
'remember', u'right', u'sargon_of_akkad', u'say', u'shit', u'stop', u'sup
port', u'theralphretort', u'think', u'threats', u'trying', u'video', u'wh
ite', u'woman', u'women']
```

Since K-Means doesn't perform very well with a large number of features, we decided to reduce the dimension using LSA/SVD. In addition, because SKLearn's Kmeans function expects a normalized matrix, we performed normalization of the result from our SVD step. After that we did a quick evaluation to determine the ideal number of clusters. In our case we have `n_clusters=8`.

Results



K-Means clustering results in many dense clusters that are close together, with a few more isolated clusters. Our ratio measure demonstrates that popular users often have a large percentage of their followers in the same cluster as they are in, helping us to conclude that there is greater similarity between a user and their followers than between random users. Therefore, users can be influenced by others in their social circles.

User ID	follower_count	Ratio
2421921523	1852	0.912533
1038152798	118	0.961039
2791121294	3014	0.901699
2421921523	1852	0.720627
2376881202	579	0.743590
2791121294	3014	0.707524
2377815434	216534	0.419380
2747504632	1601	0.669958
2421921523	1852	0.635770
2421921523	1852	0.739681
2791121294	3014	0.753641
830959110	1229	0.730198
2747504632	1601	0.675599
1052040667	1343	0.685484
2421921523	1852	0.659269
175378676	189642	0.583200
1874836183	1114	0.400000
302778669	170	0.574074
14172204	204	1.000000
246557633	197	0.571429
28959261	76	0.000000
2421921523	1852	0.710183
524901955	2975	0.441860
2781159212	236	0.691057

Above are users who own the 3 most popular tweets in each cluster. `Ratio` is the percentage of their followers who are in the same cluster.

Interesting Cases

```
id_group = tweets_df.groupby('User ID')
print id_group.size().sort_values(ascending=False)
User ID
2791121294    1918
2807060009    1171
2421921523     901
1132062200     730
```

```
id_group.get_group(2791121294)[['Retweets', "Text"]].sort_values
```

	Retweets	Text
46993	538	@CHSommers: Cathy Young concludes that is an...
176384	517	@Lemmingbot: Before @Sargon_of_Akkad distract...
26561	482	@CHSommers: Didn't the editorial director of ...
133075	434	@TheRalphRetort: Doing business with @gawker,...
109006	430	@CHSommers: New Pew study on online harassmen...
126127	427	@CHSommers: A defense of coming Monday from ...
143947	381	@Toshi_TNE: 40th in my series of people of . ...
108946	372	@MomsAgainstGam: Our interns asked actor @Ada...

Our user with the most tweets was also one of our popular users. We realized he was using Twitter to talk one on one with other users, tweets which were then receiving a lot of attention, explaining both his large number of tweets and his high popularity.

```
tweets_df[tweets_df['Labels'] == 2].sort_values('Popularity', ascending=False).drop(tweets_df.columns[[2,3,8]], axis=1)
```

	User ID	Retweets	Mentions	Text	Polarity	Popularity	Labels
193717	14172204	7709	[17264476]	@Spacekatgal: The police just came by. Husban...	0.2550	7709	2
22862	246557633	7709	[17264476]	@Spacekatgal: The police just came by. Husban...	0.8980	7709	2
181746	7578742	7709	[17264476]	@Spacekatgal: The police just came by. Husban...	1.0000	7709	2

Another interesting case we found is that the most popular users in cluster two are the ones that retweeted @Spacekatgal's tweet. When we take a deeper look, @Spacekatgal is one of the victims of this event, and a lot of users in cluster two are talking about her or retweeting her.

She is not in our popular user list since she tweeted before the data scrape.

References: [1] waxy.org/random/misc/gamergate_tweets.csv