Capstone 3: Troll Tweet Identifier

Elizabeth Rogers

Troll Tweets

- Troll accounts following 2016 election suspended by Twitter.
- Linked to Russian Internet Agency
- NBC News reproduced over 200,000 tweets and account info on troll users
- The dataset they released could be invaluable as a case study on how these accounts operate and the sort of tweets they produce to misinform and cause disarray.

Twitter and Harvard election tweet dataset API

- Following the 2016 election, a Harvard study used the Twitter API to pull the tweet ids of election-related tweets made in the run up to the election.
- There are millions of tweet id saved in the dataset, though many of the tweets' accounts have been suspended or deleted since their compilation.
- These ids can be used to pull the same account and tweet information that was stored in the reconstructed troll tweet dataset and used to run comparisons.

Tweet Collection Method

- Harvard election tweet dataset through Twitter API
- All retrievable tweets from first 5,000 ids within 6 election filter datasets
- Future study should better randomize this API call to spread out dates more

Data Cleaning

- Troll tweets : 200,000 +
- Normal tweets from the API: 13,000+
- After selecting down to English language users and tweets within the timeline: 50,000+ troll tweets
- The troll tweet dataset, being reconstructed, was also incomplete and flawed in places

Data Cleaning: Fixing the Troll Tweet Dataset

- Duplicate tweet ids, sometimes with a different user attached
- NAs from numeric categories filled with the median value from the combined dataset
- Missing source addresses filled with most common source address logged for that user
- Missing or incomplete data in mentions and url categories recreated based on tweet text data

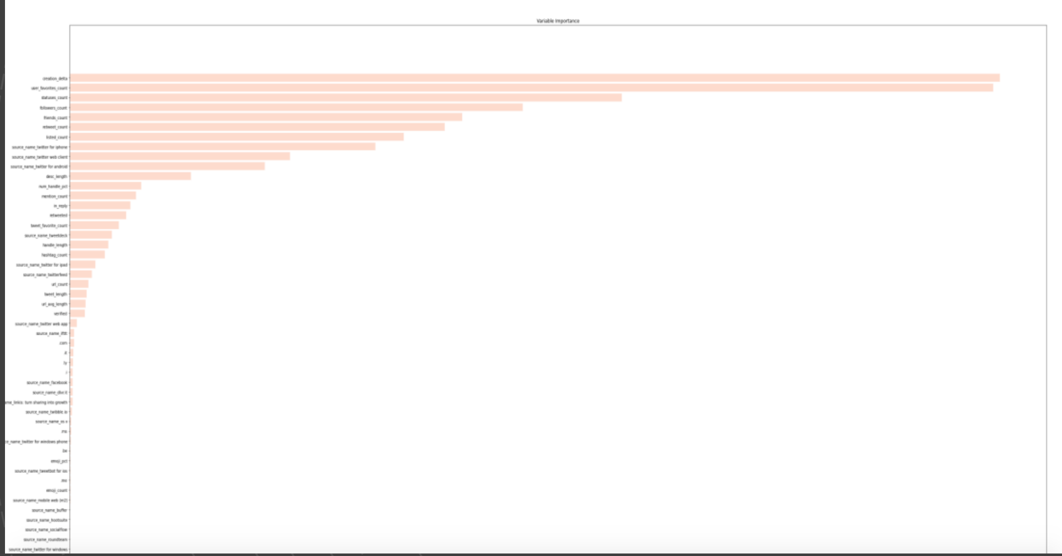
Feature Engineering

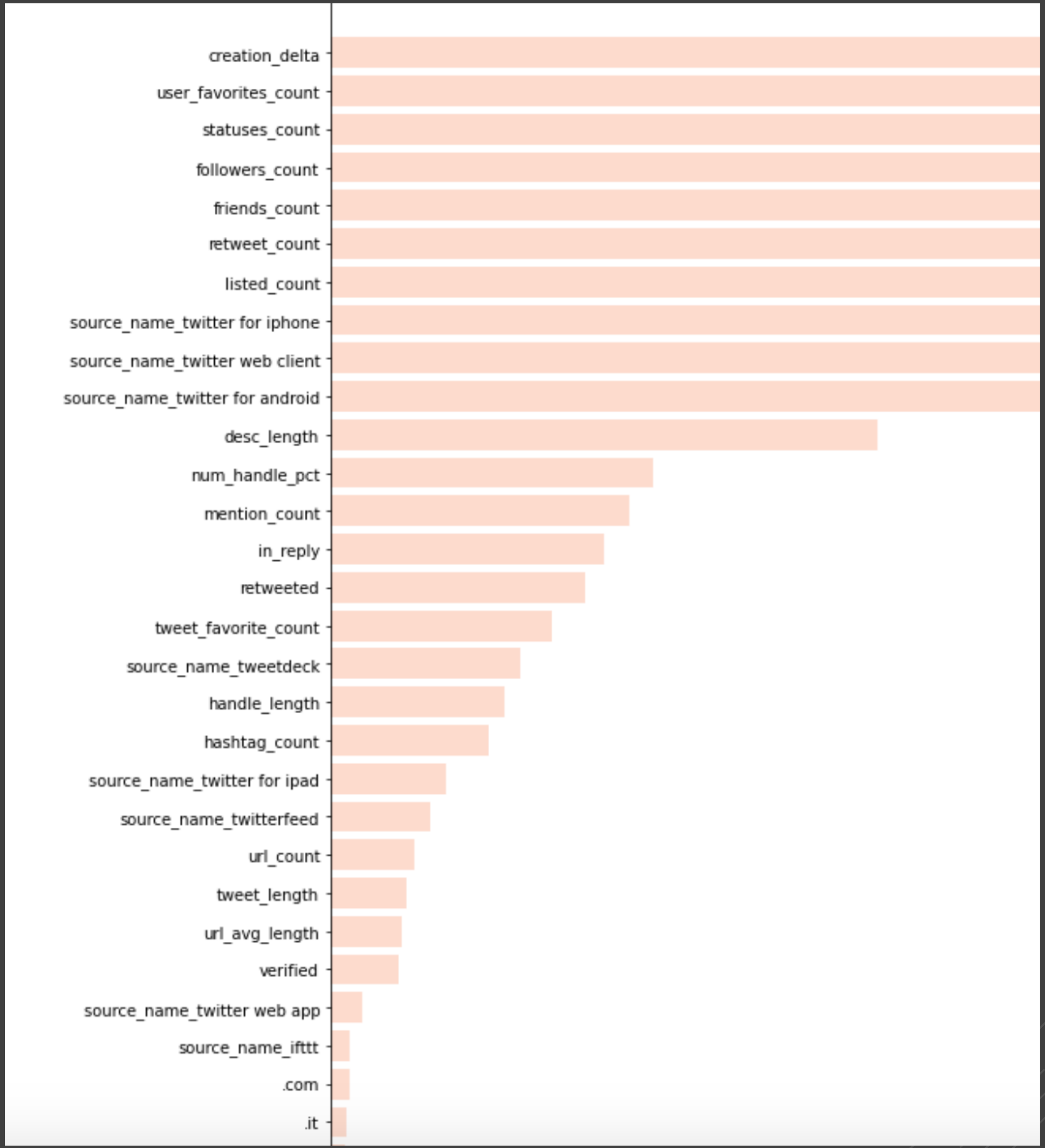
- Creation_delta
- Emoji_count
- Emoji_pct
- Tweet_length
- Desc_length
- Mention_count
- Hashtag_count
- Url_count
- Handle_length
- Num_handle_pct
- Url_avg_length
- Desc_vecs
- Text_vecs

EDA: CAVEAT

- Imbalanced dataset!
- Around 5x more troll tweets than normal tweets in the study
- Additionally and more importantly, only 261 unique users within the troll tweet dataset at the end of data cleaning and isolating down to the same timeline and language.
- This is being compared to 10253 unique users in the election dataset.
- The differences shown are thus exaggerated, though the diversity extant in users on Twitter is clear in comparison to troll user accounts.
- Future studies will need to control for this imbalance.

EDA: Feature Importance

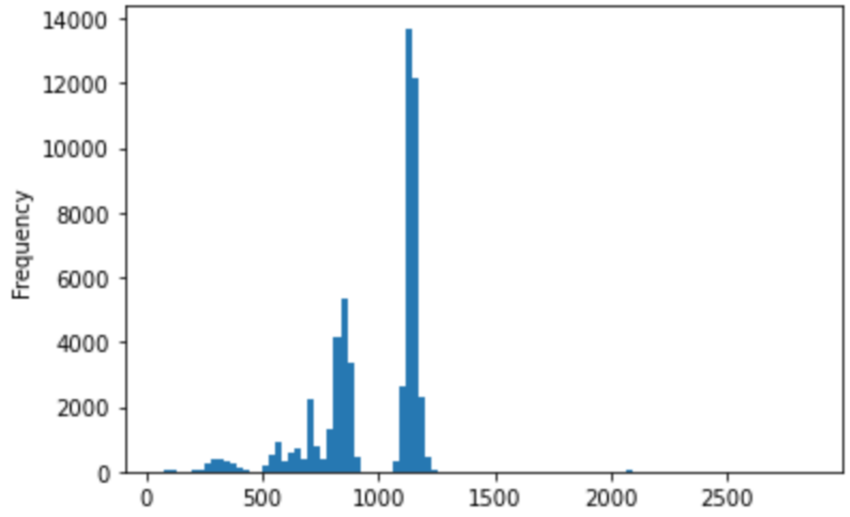




EDA: #1 The creation delta

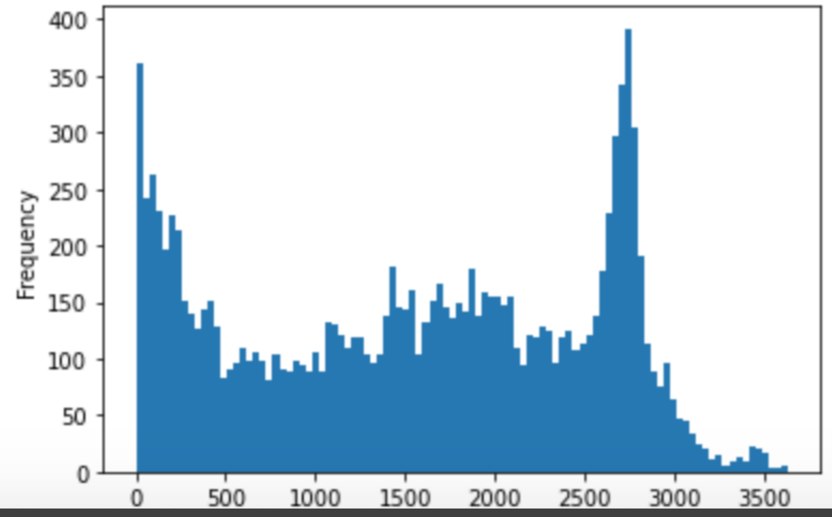
```
troll_tweets['creation_delta'].plot.hist(bins=100)
```

`<AxesSubplot:ylabel='Frequency'>`




```
normal_tweets['creation_delta'].plot.hist(bins=100)
```

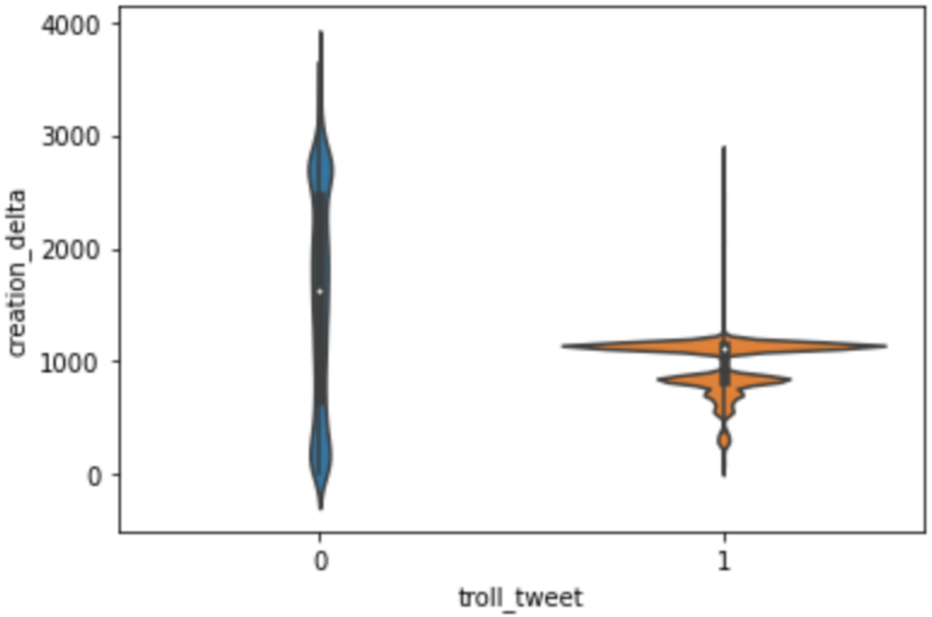
`<AxesSubplot:ylabel='Frequency'>`



EDA: #1 The creation delta

```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='creation_delta')
```

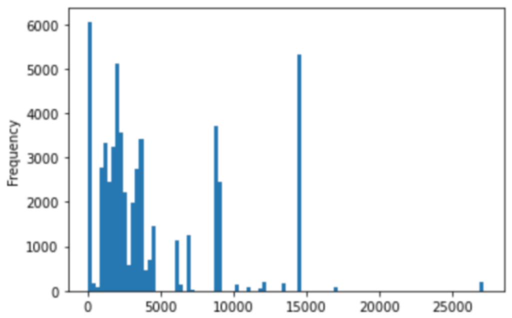
```
<AxesSubplot:xlabel='troll_tweet', ylabel='creation_delta'>
```



EDA: #2
User favorites count

```
troll_tweets['user_favorites_count'].plot.hist(bins=100)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
troll_tweets['user_favorites_count'].mean()
```

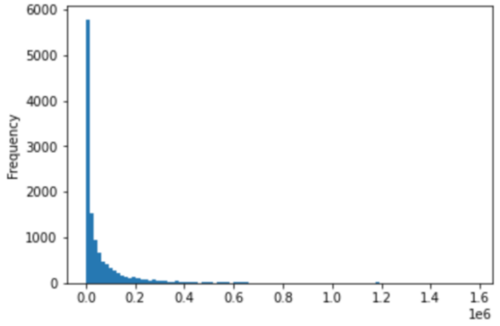
```
4442.91922025738
```

```
troll_tweets['user_favorites_count'].max()
```

```
27181
```

```
normal_tweets['user_favorites_count'].plot.hist(bins=100)
#WOW these numbers are extremely different
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['user_favorites_count'].mean()
```

```
65586.74155749044
```

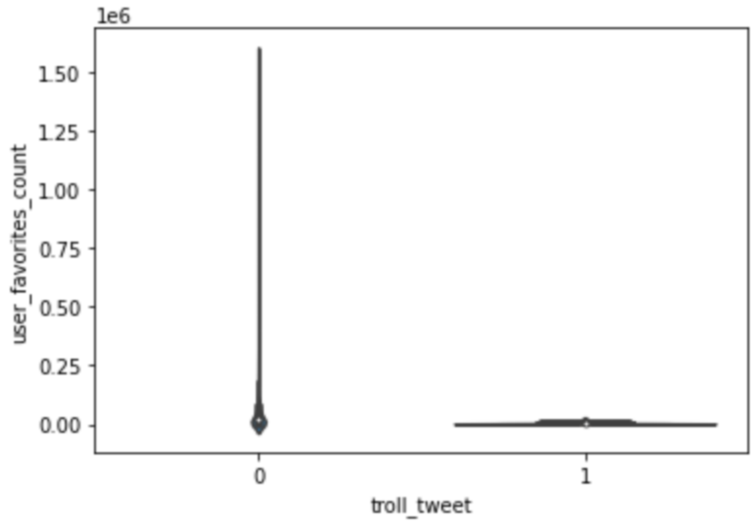
```
normal_tweets['user_favorites_count'].max()
```

```
1569811
```

EDA: #2
User favorites count

```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='user_favorites_count')
```

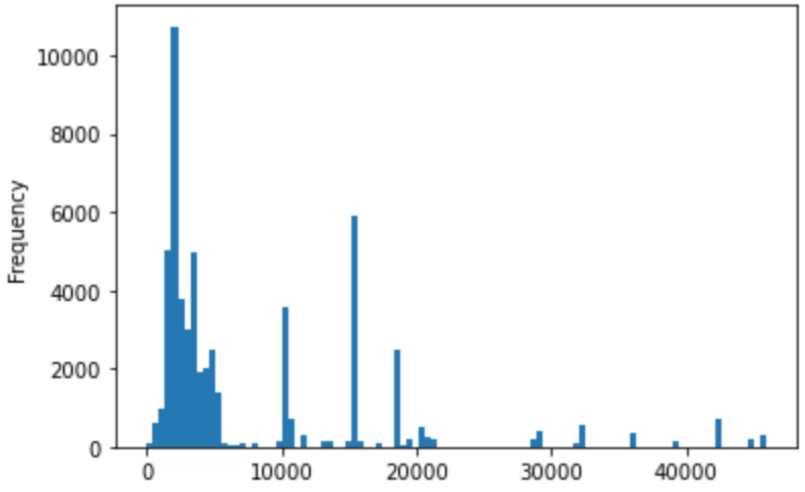
```
<AxesSubplot:xlabel='troll_tweet', ylabel='user_favorites_count'>
```



EDA: #3
Statuses count

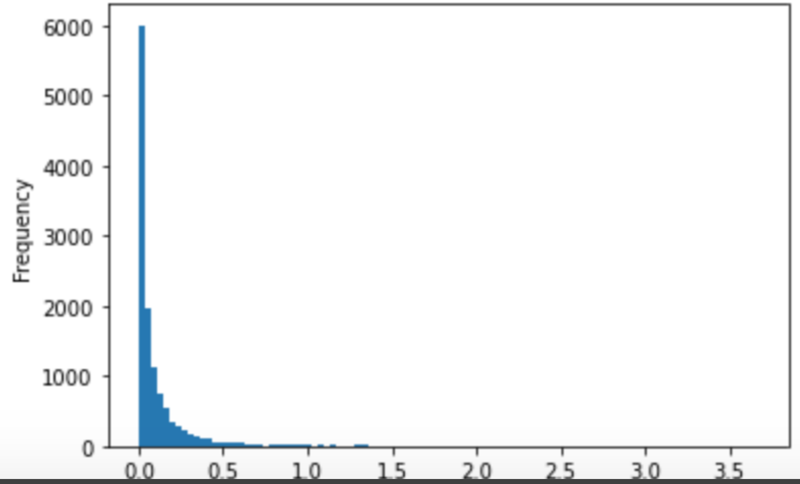
```
troll_tweets['statuses_count'].plot.hist(bins=100)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['statuses_count'].plot.hist(bins=100)
```

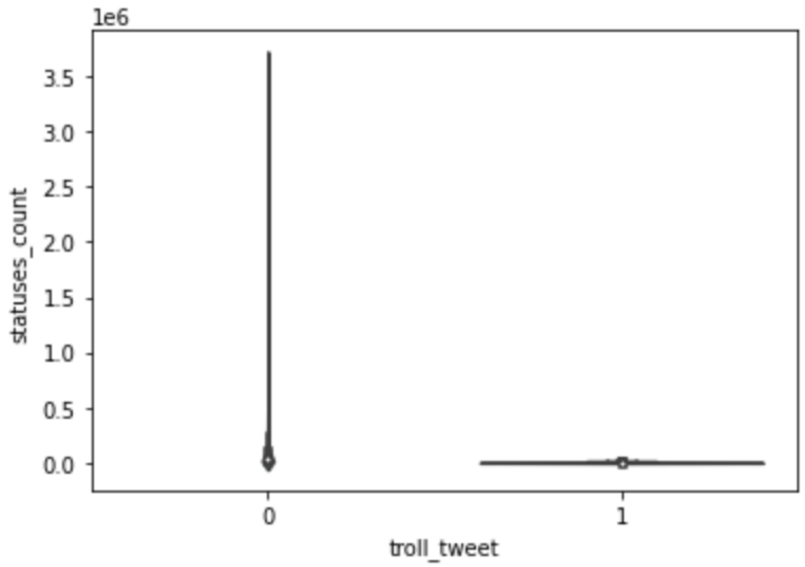
```
<AxesSubplot:ylabel='Frequency'>
```



EDA: #3
Statuses count

```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='statuses_count')
```

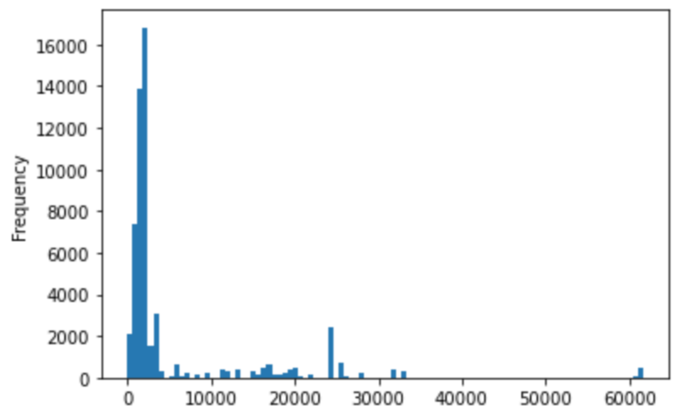
```
<AxesSubplot:xlabel='troll_tweet', ylabel='statuses_count'>
```



EDA: #4
Followers count

```
troll_tweets['followers_count'].plot.hist(bins=100)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
troll_tweets['followers_count'].mean()
```

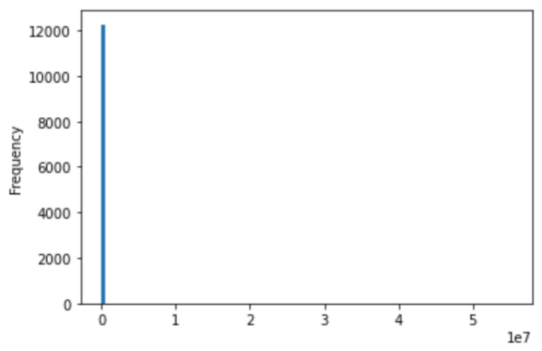
```
5412.386106535865
```

```
troll_tweets['followers_count'].max()
```

```
61609
```

```
normal_tweets['followers_count'].plot.hist(bins=100)
#It looks like most normal accounts have few followers, though normal
#amounts of followers
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['followers_count'].mean()
```

```
17380.99853527545
```

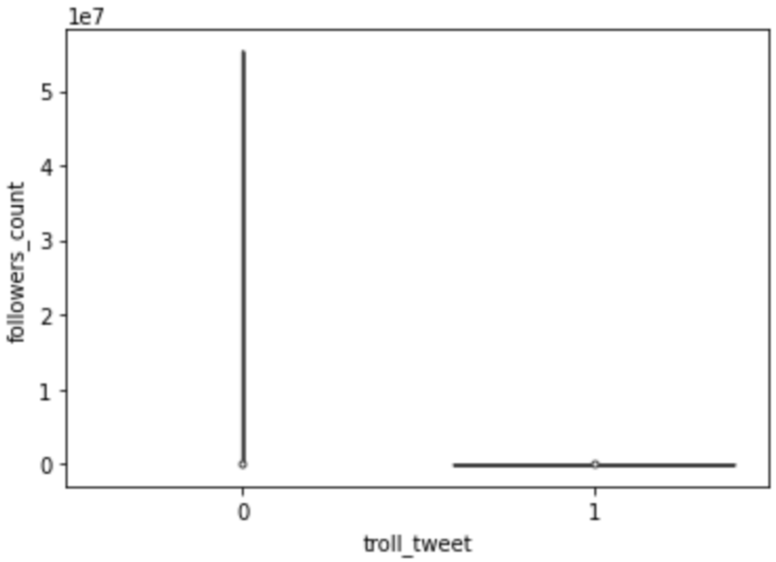
```
normal_tweets['followers_count'].max() #Whoa I wonder who that was.
```

```
55245748
```

EDA: #4 Followers count

```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='followers_count')
```

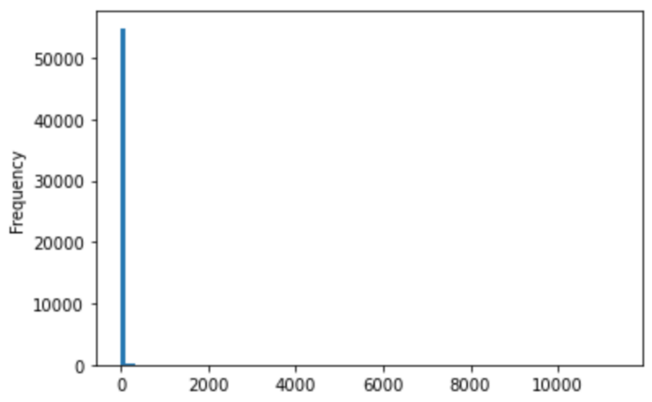
```
<AxesSubplot:xlabel='troll_tweet', ylabel='followers_count'>
```



EDA: #6 Retweet count

```
troll_tweets['retweet_count'].plot.hist(bins=100)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
troll_tweets['retweet_count'].mean()
```

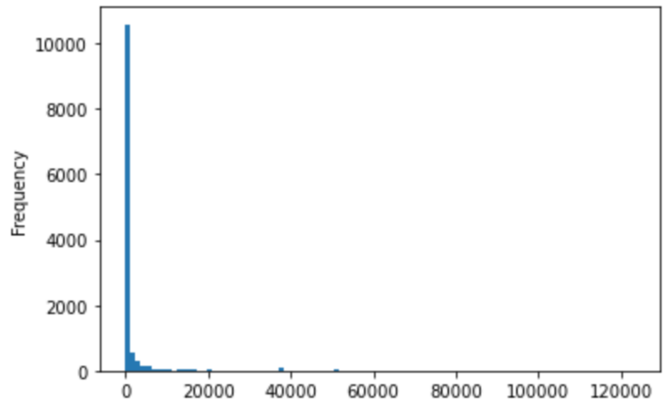
```
3.6639939184419625
```

```
troll_tweets['retweet_count'].max()
```

```
11363
```

```
normal_tweets['retweet_count'].plot.hist(bins=100)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['retweet_count'].mean()
```

```
1435.8314753031166
```

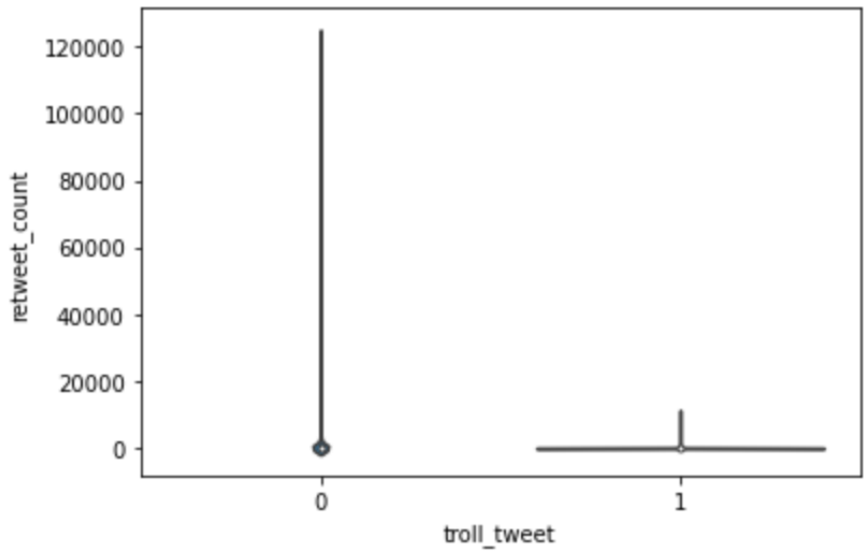
```
normal_tweets['retweet_count'].max()
```

```
123215
```

EDA: #6
Retweet count

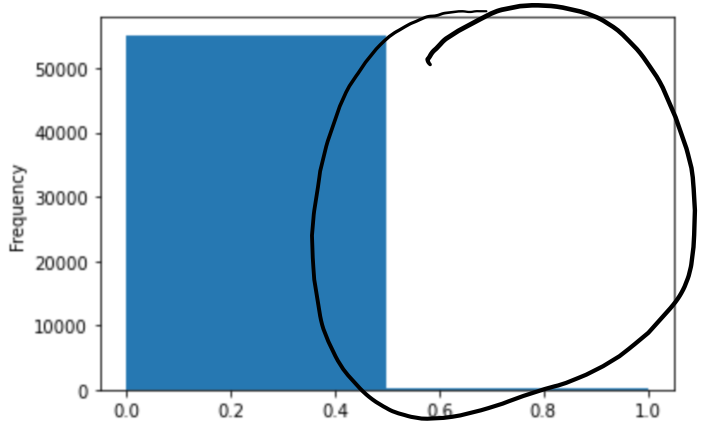
```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='retweet_count')
```

```
<AxesSubplot:xlabel='troll_tweet', ylabel='retweet_count'>
```

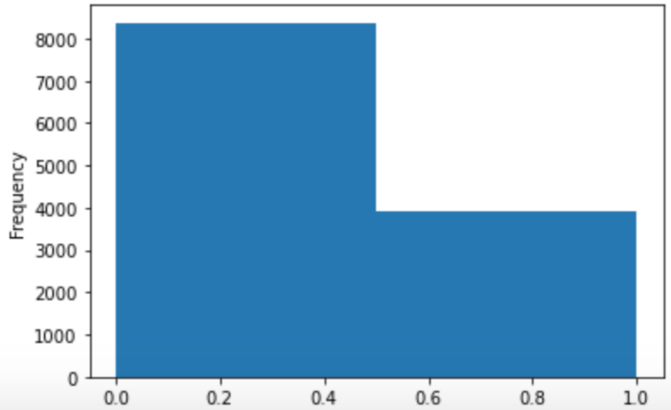


EDA: #8
Source name Twitter for iphone

```
troll_tweets['source_name_twitter for iphone'].plot.hist(bins=2)
<AxesSubplot:ylabel='Frequency'>
```



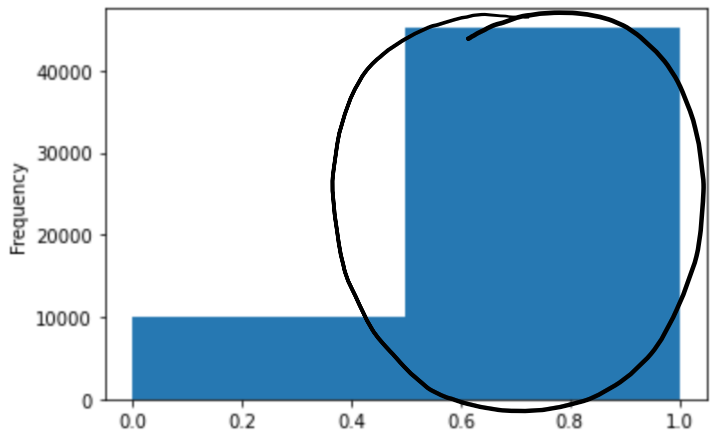

```
normal_tweets['source_name_twitter for iphone'].plot.hist(bins=2)
<AxesSubplot:ylabel='Frequency'>
```



EDA: #9
Source name Twitter for web client

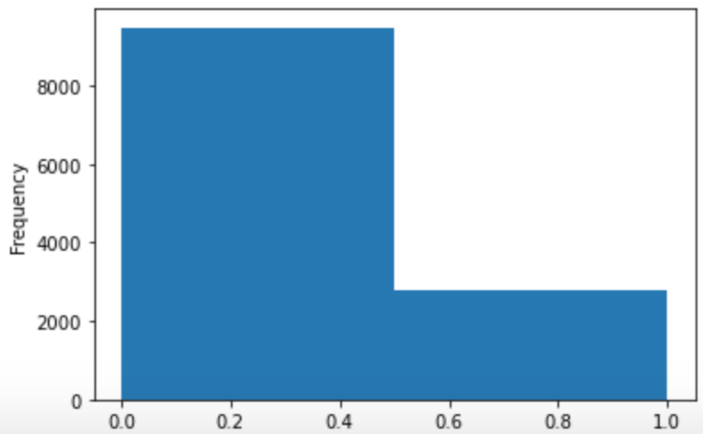
```
troll_tweets['source_name_twitter web client'].plot.hist(bins=2)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['source_name_twitter web client'].plot.hist(bins=2)
```

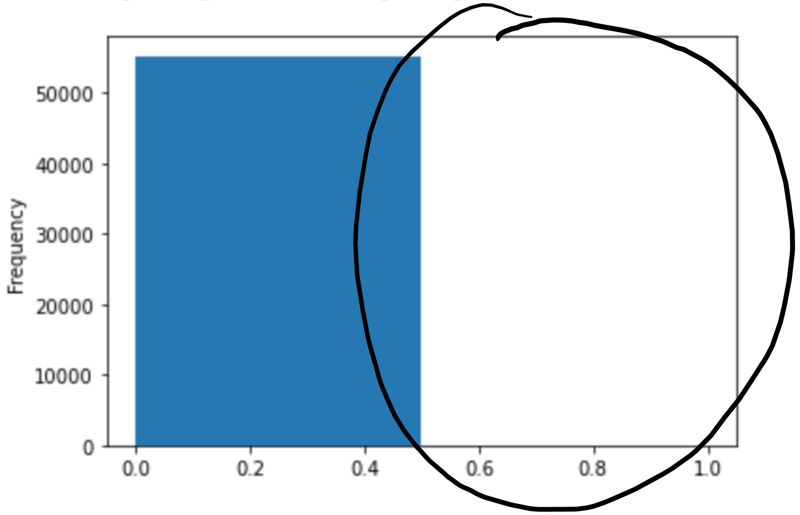
```
<AxesSubplot:ylabel='Frequency'>
```



EDA: #10 Source name Twitter for android

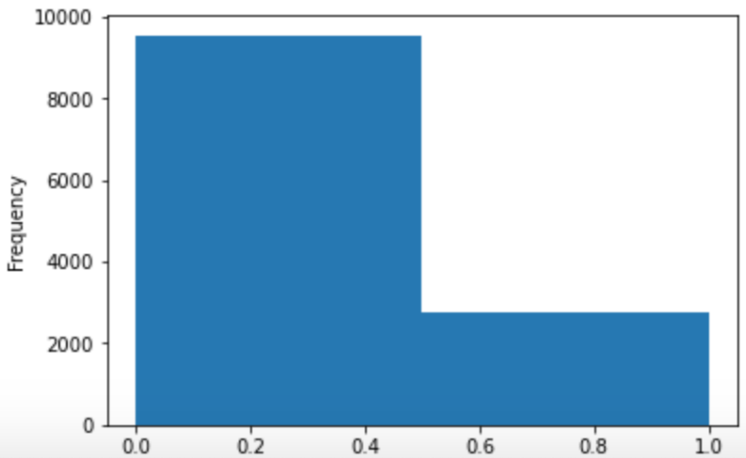
```
troll_tweets['source_name_twitter for android'].plot.hist(bins=2)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['source_name_twitter for android'].plot.hist(bins=2)
```

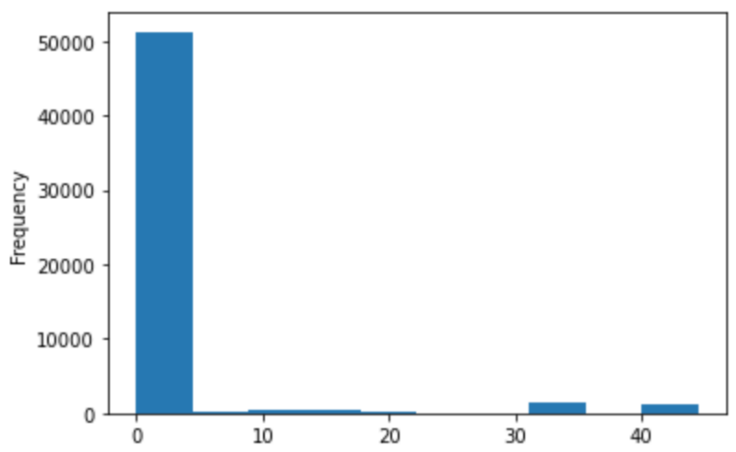
```
<AxesSubplot:ylabel='Frequency'>
```



EDA: #12
Num handle percent

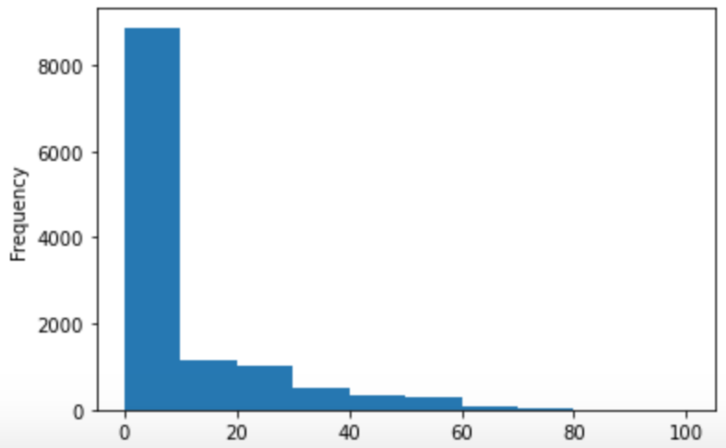
```
troll_tweets['num_handle_pct'].plot.hist(bins=10)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['num_handle_pct'].plot.hist(bins=10)
```

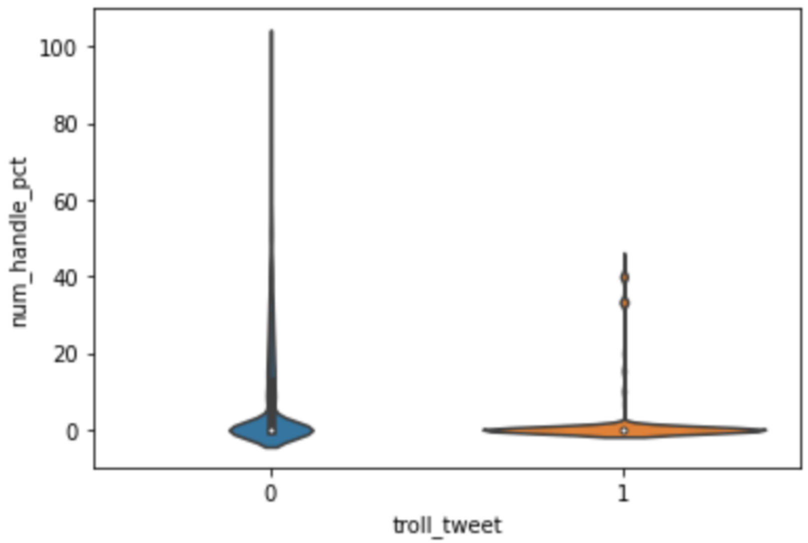
```
<AxesSubplot:ylabel='Frequency'>
```



EDA: #12 Num handle percent

```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='num_handle_pct')
```

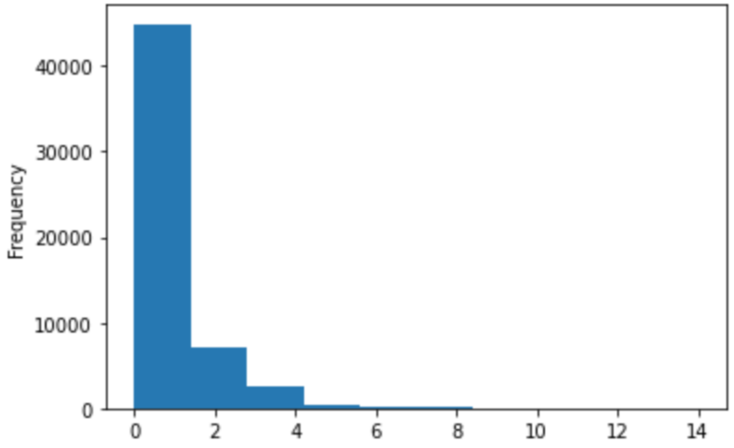
```
<AxesSubplot:xlabel='troll_tweet', ylabel='num_handle_pct'>
```



EDA: #13
Mention count

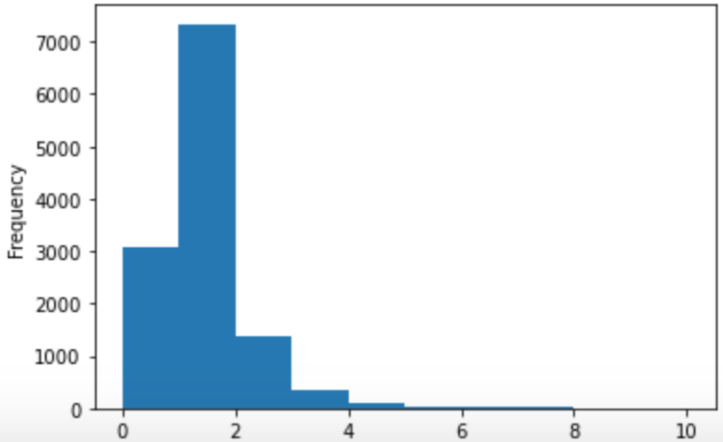
```
troll_tweets['mention_count'].plot.hist(bins=10)
```

```
<AxesSubplot:ylabel='Frequency'>
```




```
normal_tweets['mention_count'].plot.hist(bins=10)
```

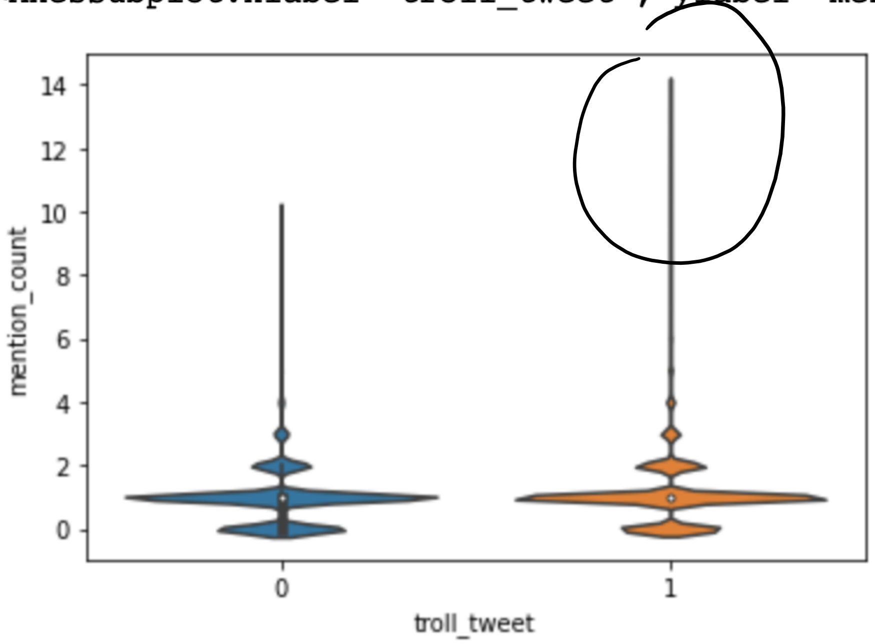
```
<AxesSubplot:ylabel='Frequency'>
```



EDA: #13
Mention count

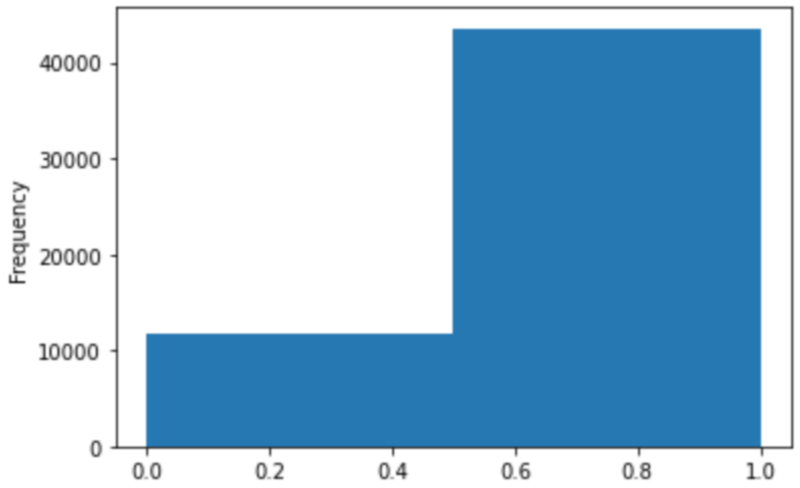
```
sns.violinplot(data=final_tweet_df, x='troll_tweet',y='mention_count')
```

```
<AxesSubplot:xlabel='troll_tweet', ylabel='mention_count'>
```



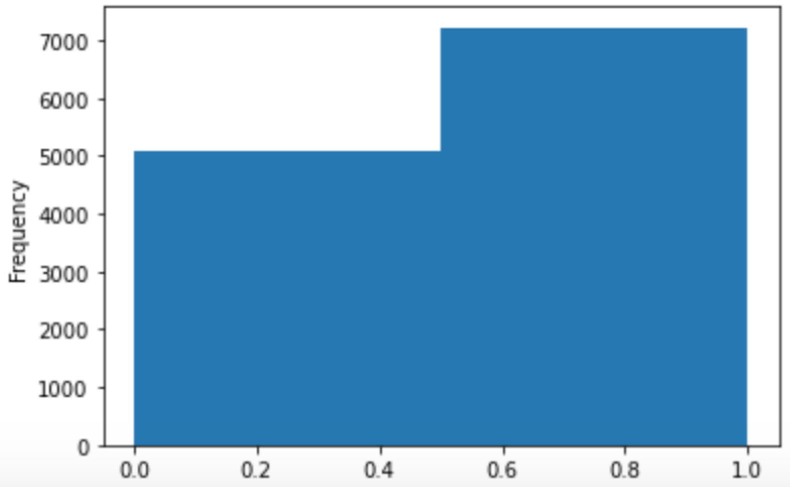
EDA: #15 Retweeted

<AxesSubplot:ylabel='Frequency'>

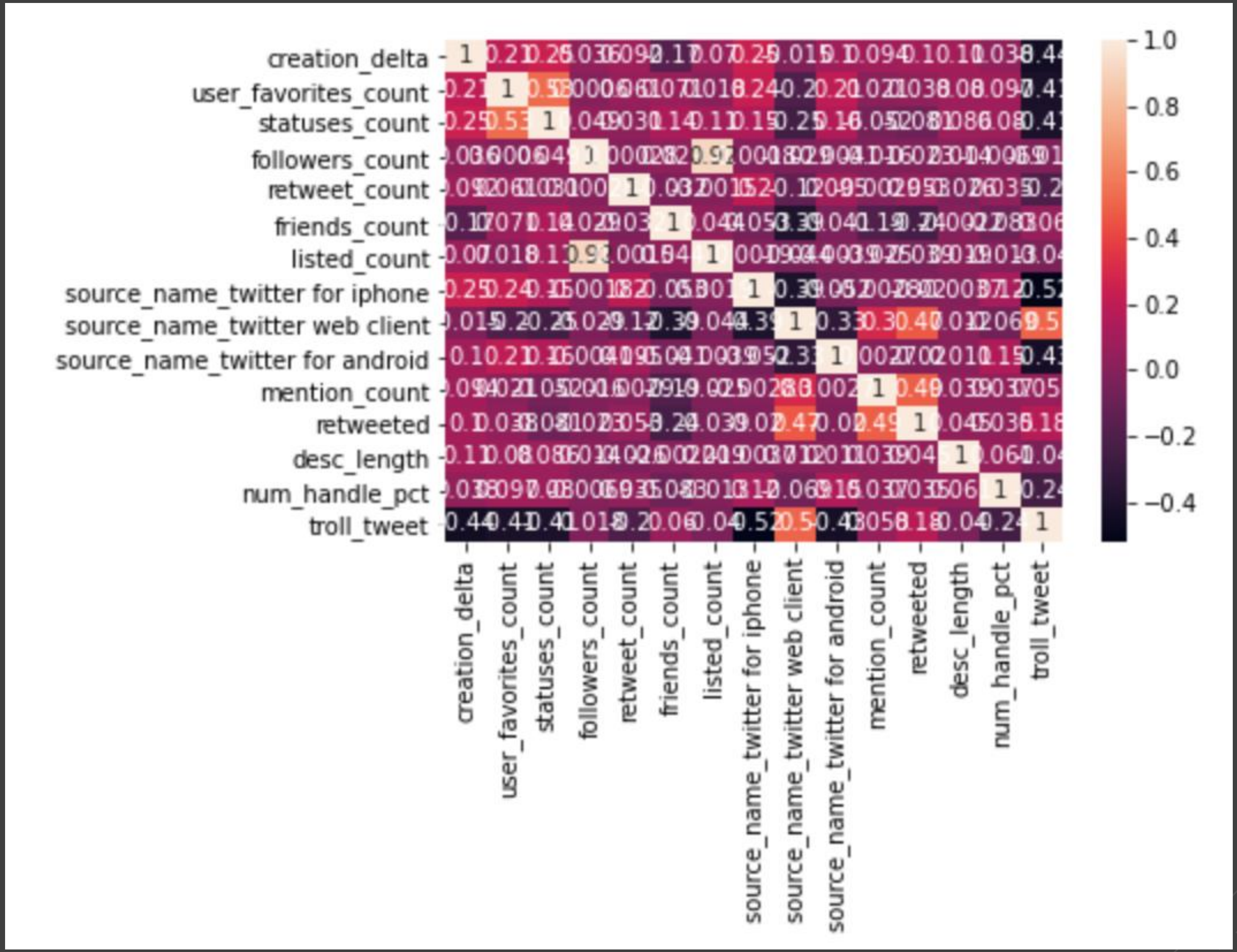



```
normal_tweets['retweeted'].plot.hist(bins=2)
```

<AxesSubplot:ylabel='Frequency'>



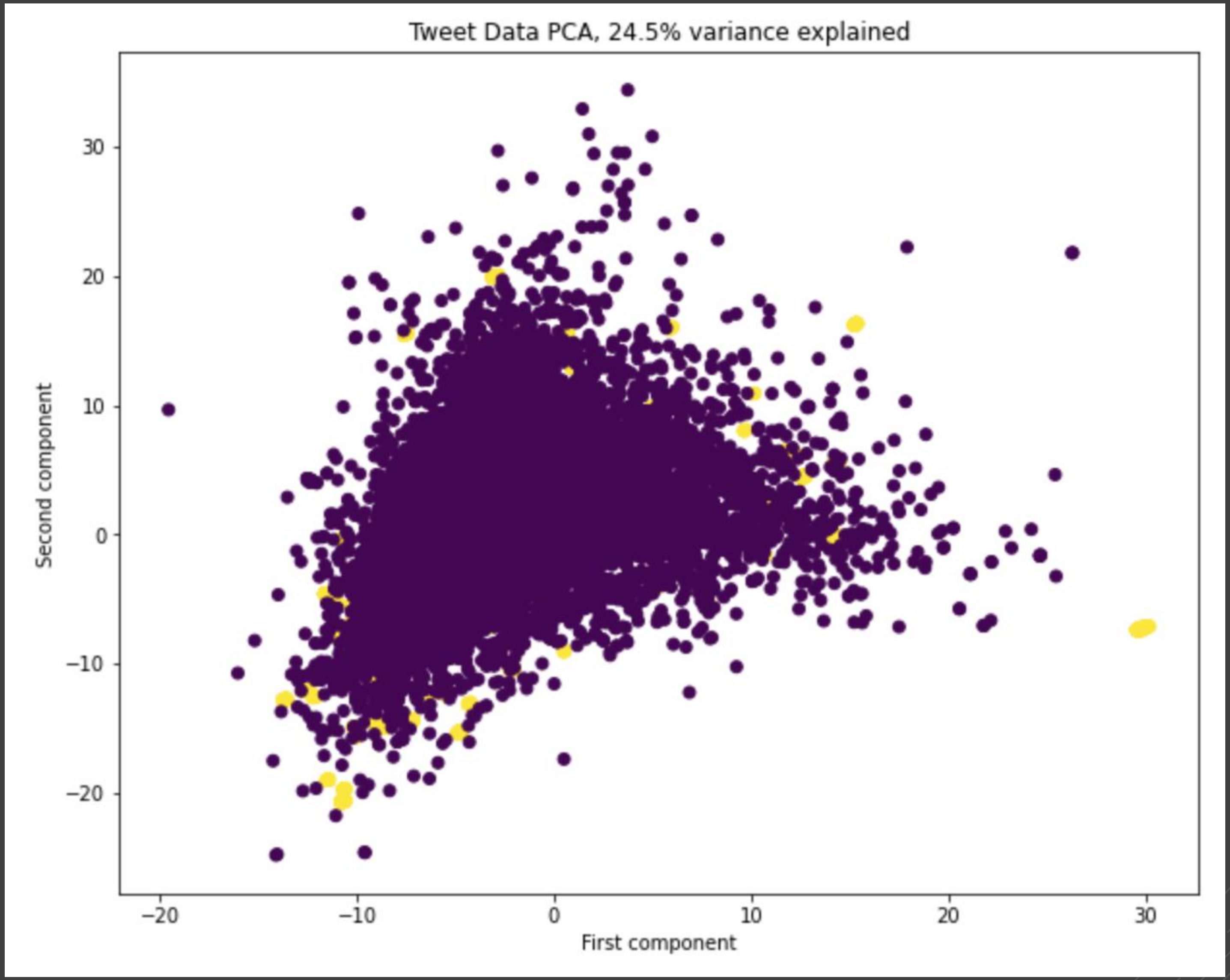
EDA: Heatmap


creation_delta
user_favorites_count
statuses_count
followers_count
retweet_count
friends_count
listed_count
source_name_twitter for iphone
source_name_twitter web client
source_name_twitter for android
mention_count
retweeted
desc_length
num_handle_pct
troll_tweet
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

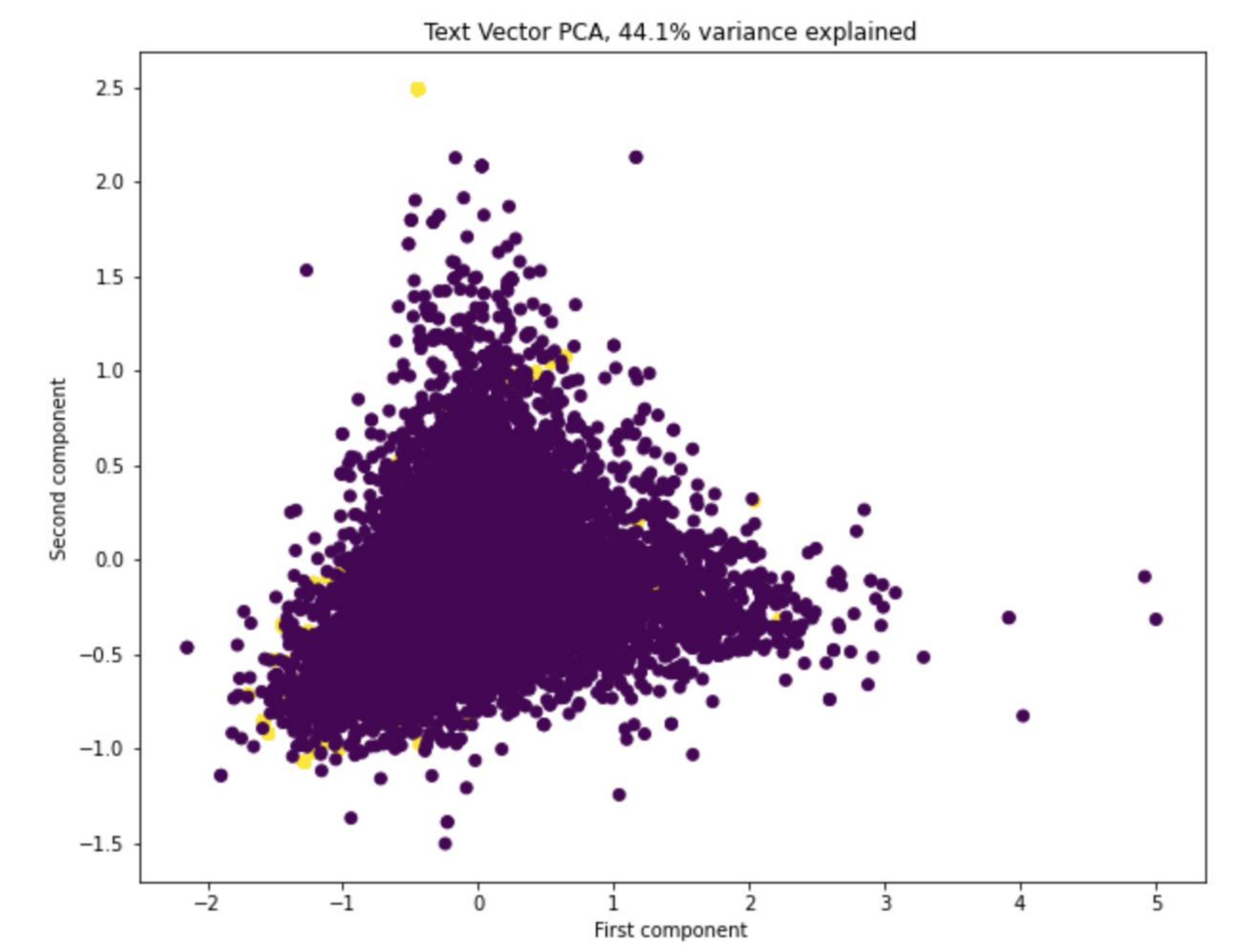
Models

- PCA on the full dataset
- PCA on just the text vectors
- PCA on just the description vectors
- PCA on just the text vectors, with retweeted tweets removed
- These did not prove helpful in their current state

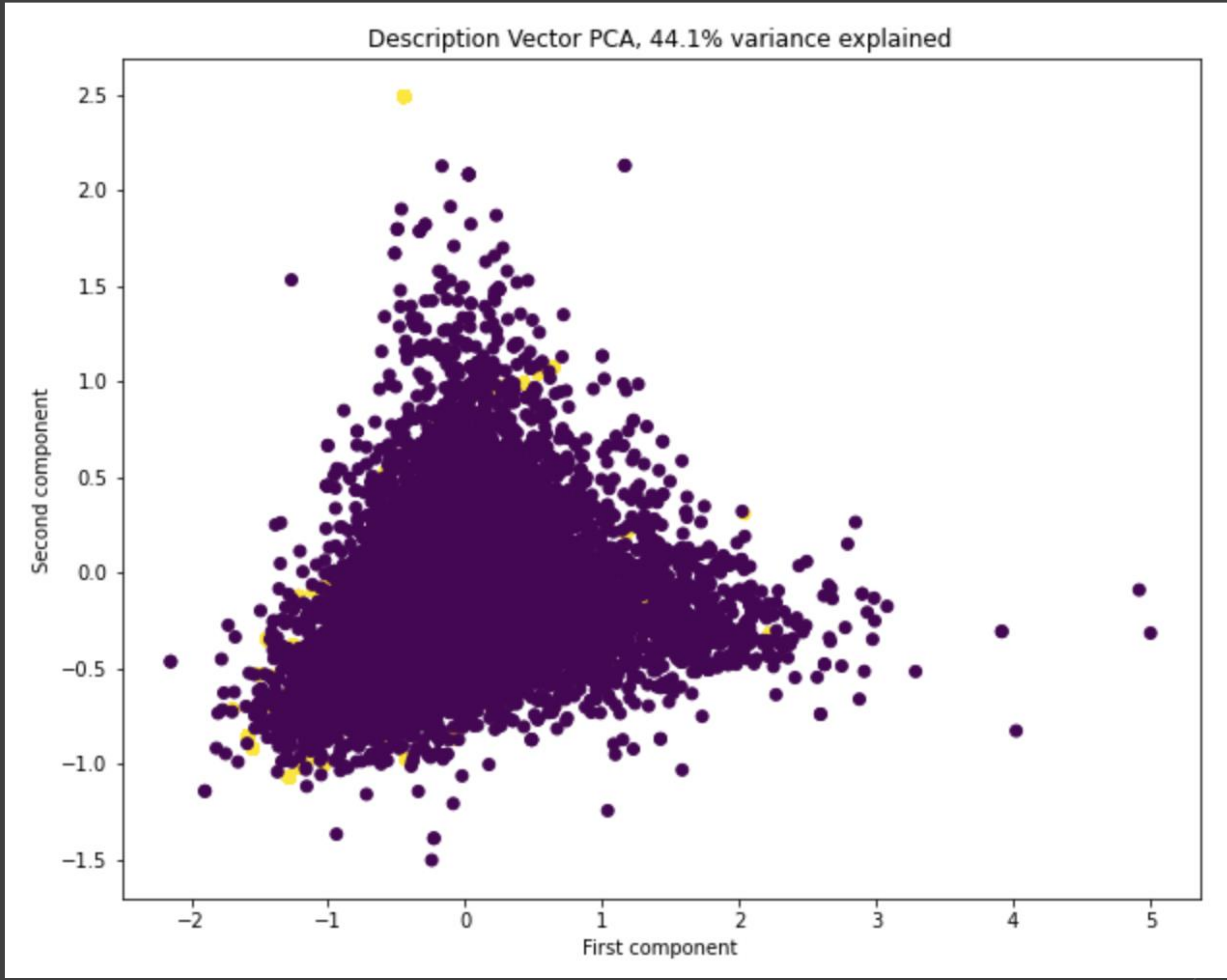
PCA on all data



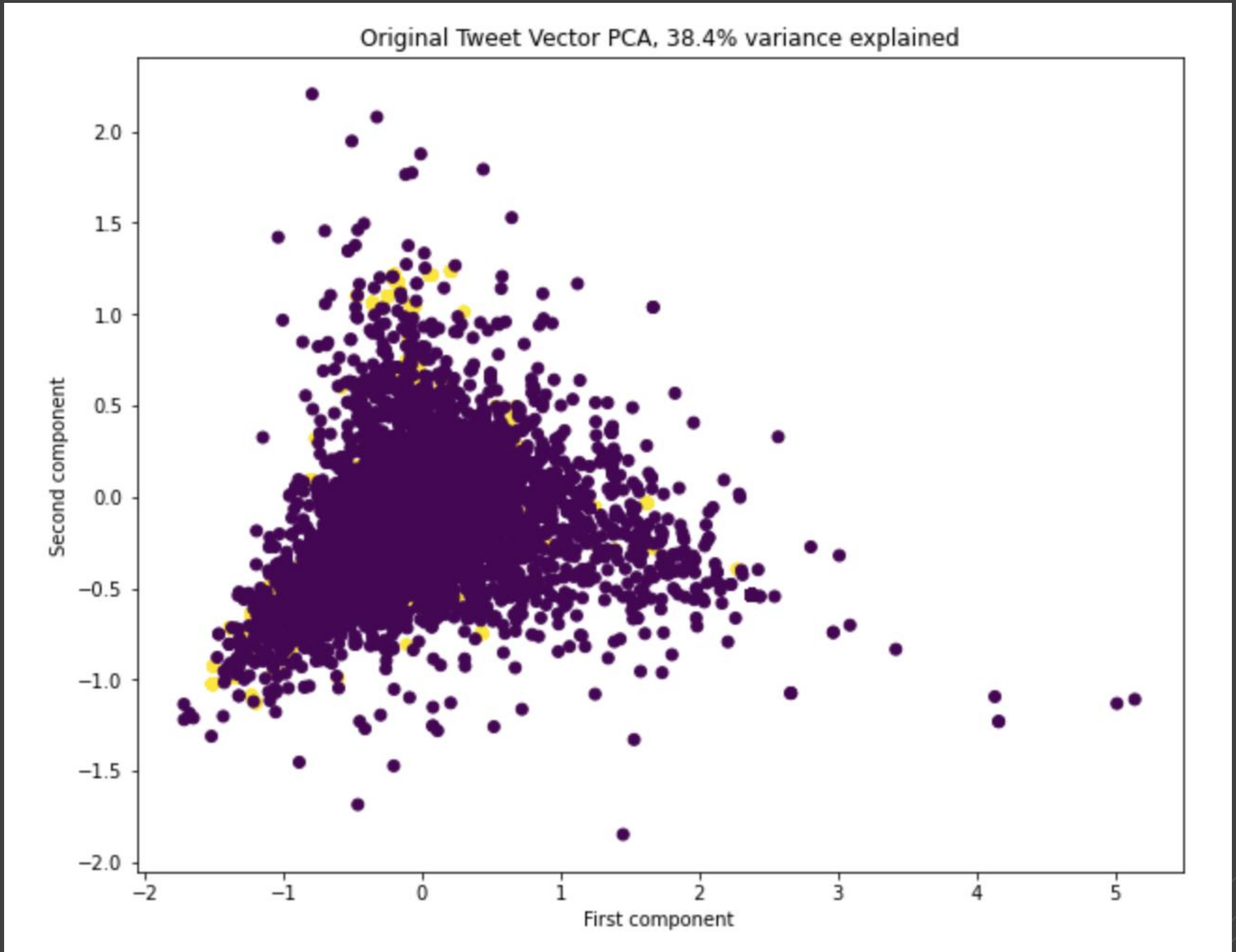
PCA on text vectors



PCA on description vectors



PCA on text vectors with retweets removed



Models

- Random Forest optimized by grid search
- Decision Trees with gini and entropy
- Gradient Boosting Classifier
- LGBM Model optimized by Bayesian Optimization

Model Metrics

	Accuracy:	Balanced Accuracy:	Precision Score:	Recall Score:
Random Forest (optimized by a grid search) - fluctuates	0.999	0.9989	0.999	0.999
Entropy Decision Tree	0.999	0.9984	0.999	0.999
Gini Decision Tree	0.998	0.996	0.998	0.999
Gradient Boosting Classifier	0.9800	0.9580	0.9829	0.9928
Light Gradient Boosting Model (LGBM - optimized by Bayesian Optimization) - fluctuates	0.999	0.9994	0.999	0.999

Conclusions

- Though imbalanced, the extreme variation in the normal tweet dataset when placed next to the more controlled boundaries of behavior within the troll tweet dataset indicates a direction for further studies into these general bounds.
- An extension of the study will need to control for the limited number of users in the troll dataset with a greater number of tweets coming from individual normal users.
- It would be interesting to compare data on other troll accounts linked to other periods of time to see whether the charted behavior and expected creation delta peaks remain consistent.

Resources:

- Twitter Deleted 200,000 Russian Troll Tweets: Read Them Here: https://www.nbcnews.com/tech/social-media/now-available-more-200-000-deleted-russian-troll-tweets-n844731
- Troll Tweet Dataset: https://www.kaggle.com/vikasg/russian-troll-tweets
- 2016 United States Presidential Election Tweet Ids: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/PDI7IN