Final – Juat Tan

1.What did you do and why is it interesting?

Trump's presidency has been largely polarizing. It should be the case that his approval ratings in polls fluctuate quite a bit, based off of the things he's said and done as president. Polling numbers for his approval and disapproval should change and should represent somewhat represent the true sentiments of the general public about his presidency. Trump has had a large presence on social media, especially Twitter. This has made it increasingly easy to find Tweets about him and his presidency from anybody. I want to take polling data, and see if I can match the public's sentiment about the Trump presidency by doing sentiment analysis on public Tweets which mention @realDonaldTrump. If polling data is taken as the true opinion about Trump, I'd like to see how closely different sentiment analysis models can get to reflecting/predicting approval from Tweets.

```
822594203058405376 2017-01-20 18:58:35 -0400 <mrkford1979> @realDonaldTrump God bless you Mr. President
822594201397379073 2017-01-20 18:58:35 -0400 <shakeraldafeery> @realDonaldTrump We congratulate you on the inauguration of the presidency of the USA.
822594198071373825 2017-01-20 18:58:34 -0400 <Bobbykeysradio> @realDonaldTrump YOU WORK FOR THE PEOPLE NOT HOME SHOPPING NETWORK  OR TWITTER THE PEOPLE  THE
```

```
1217959335747174401 2020-01-16 18:58:28 -0400 <SpartanMatt8497> @realDonaldTrump Ok Boomer.
1217959335281688582 2020-01-16 18:58:28 -0400 <TruckNuts3435> @JeffreyGuterman @realDonaldTrump You are one real pervert
```

Two screenshots of Tweets which show different sentiment towards @realDonaldTrump.

```
president,subgroup,modeldate,startdate,enddate,pollster,grade,samplesize,population,weight,influence,appr
Donald Trump,All polls,1/20/2021,1/20/2017,1/22/2017,Morning Consult,B/C,1992,rv,.6800286,0,46,37,45.6867
```

Example data from one poll spanning 3 days.

2.How do your model(s) work? (explain)

I used three different sentiment analysis models in three different natural language processing libraries. The first is from Natural Language ToolKit's SentimentIntensityAnalyzer, which uses the VADER algorithm. VADER (Valence Aware Dictionary for Sentiment Reasoning), takes values from a lexicon, which words map to. These values are a sentiment score, which are positive or negative values which represent how positive or negative the word is. Words which are not in the lexicon have a sentiment value of 0. All words in the sentence are added up and passed through a normalization function to output a range from 0 to 1, positive, neutral, or negative. The second model I used is from a library called Textblob. https://textblob.readthedocs.io/en/dev/index.html It's similar to NLTK's VADER sentiment analysis. Words are also looked up in a lexicon, and each word has a sentiment value as well. Negation words have modifiers which multiply the proceeding word's value by -0.5, inflecting its positivity or negativity score. Modifier words like, “very”, “super”, “quite”, multiply the following word by an intensity score which is contained within the lexicon to give the proceeding word greater weight. Sentiment scores for each word are clamped between the range 1, 1. The average of all scores is taken and presented as the sentence’s sentiment. The third model I used was from a library called Flair. https://github.com/flairNLP/flair. It contains a model pre-trained on IMDB movie reviews. The model itself is a LSTM neural network.

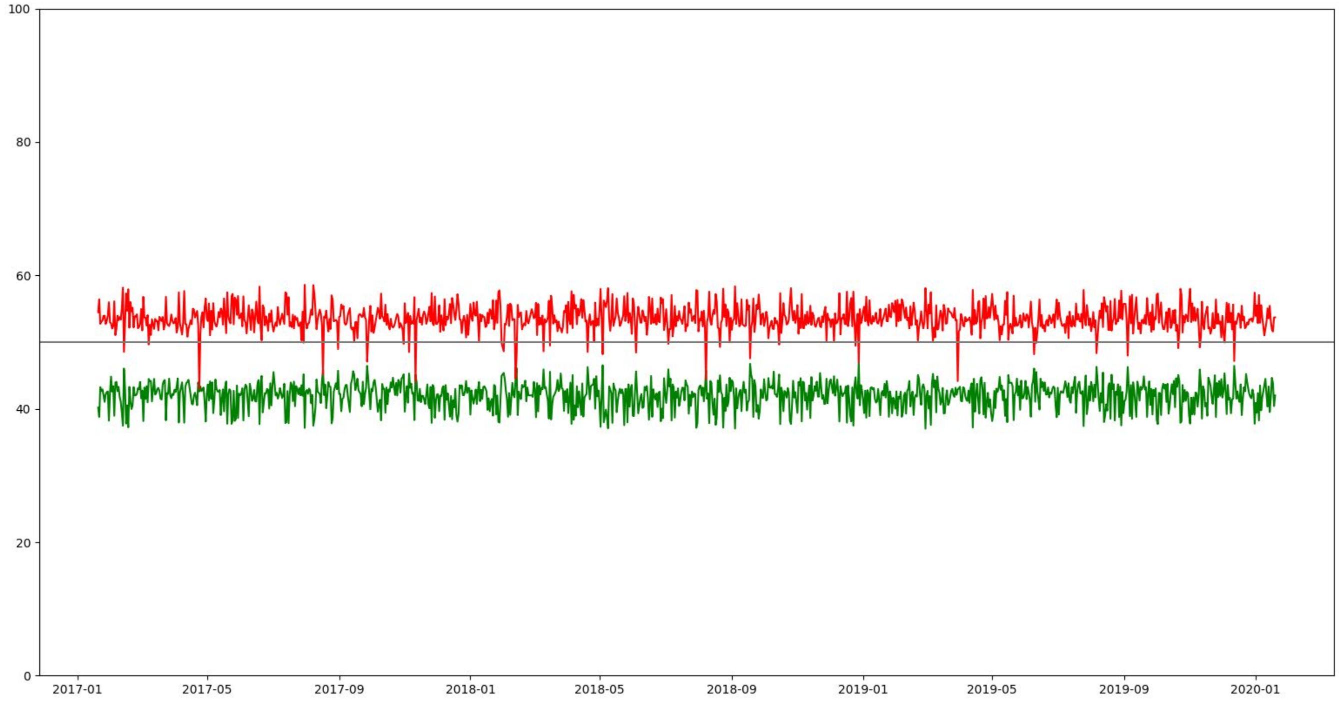
3.What are your results/conclusions?

The following are visualizations of

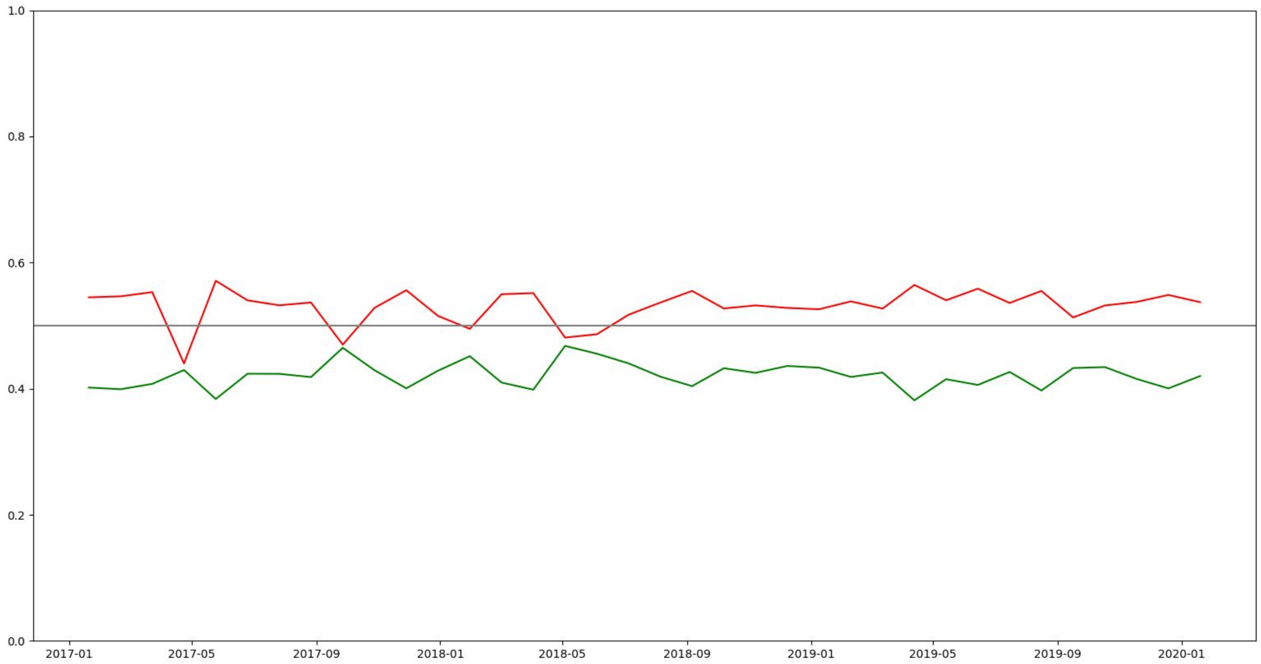
1. Trump approval ratings
2. Trump approval ratings, smoothed monthly with a spline function
3. Trump approval (approve – disapprove)
4. NLTK Sentiment of Tweets (green = positive, red = negative, blue = neutral)
5. Textblob Sentiment of Tweets
6. Flair Sentiment of Tweets
7. NLTK Sentiment of Tweets (left) & NLTK Sentiment of cleaned Tweets (right)
8. NLTK Sentiment of cleaned Tweets (difference approve - disapprove)
9. Textblob Sentiment of Tweets (left) & NLTK Sentiment of cleaned Tweets (right)
10. Textblob Sentiment of cleaned Tweets (difference approve - disapprove)
11. Flair Sentiment of Tweets (left) & NLTK Sentiment of cleaned Tweets (right)
12. Flair Sentiment of cleaned Tweets (difference approve – disapprove)
13. #3
14. #8 (left) & #8 vs. #3 (right)
15. #10 (left) & #10 vs. #3 (right)
16. #12 (left) & #12 vs. #3 (right)
17. #14 smoothed. Correlation coefficient in bold.
18. #15 smoothed. Correlation coefficient in bold.
19. #16 smoothed. Correlation coefficient in bold.

The sentiments for Tweets labeled by all three models are quite noisy and hard to visualize without smoothing. Both NLTK and Textblob sentiment analysis models labeled the majority of Tweets as neutral, which was somewhat helped by cleaning Twitter data of things like emojis, punctuation, stopwords, and @mentions. The Flair neural net model didn’t label any Tweets as neutral, and did the best following poll data. I condensed Tweet and poll sentiment into correlation coefficients to compare the three models, and the Flair neural net model outperformed the VADER and Textblob models by a large margin. The results were largely expected. Twitter data, even after cleaning, is not the best. It was a challenge to visualize and sum up findings. Calculating correlation coefficient between Twitter sentiment and poll data was the easiest way to sum up how accurately each model could be applied to Tweets to extrapolate approval. The neural net had more than double the correlation (0.26 vs. 0.13 and 0.12) between the two datasets than the two simpler models.

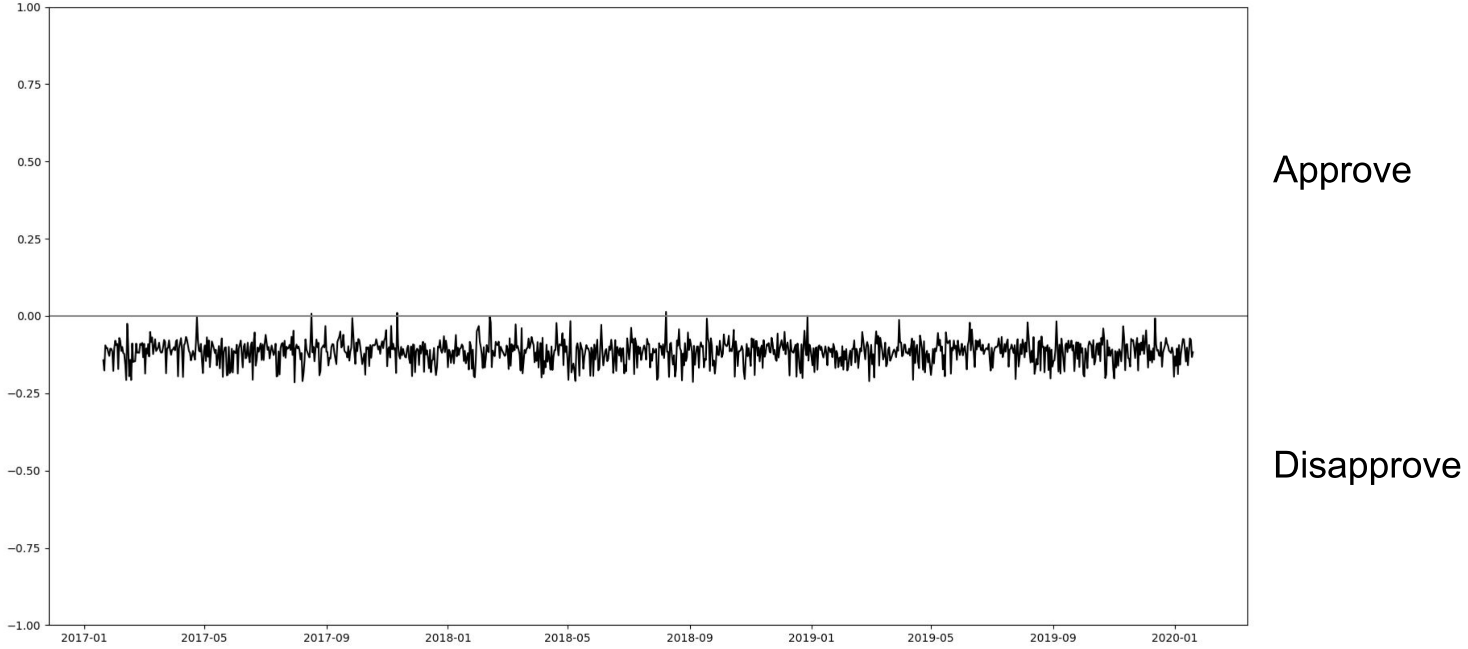
Trump Approval Ratings



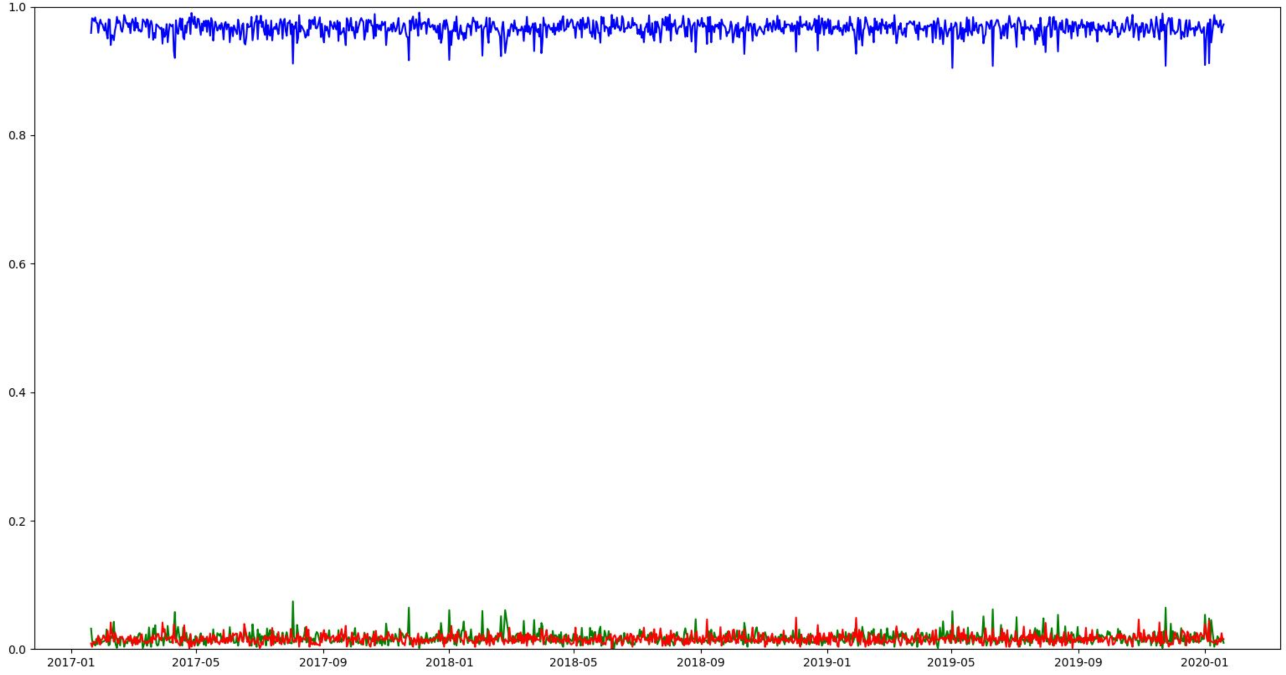
Trump Approval (Smoothed Monthly)



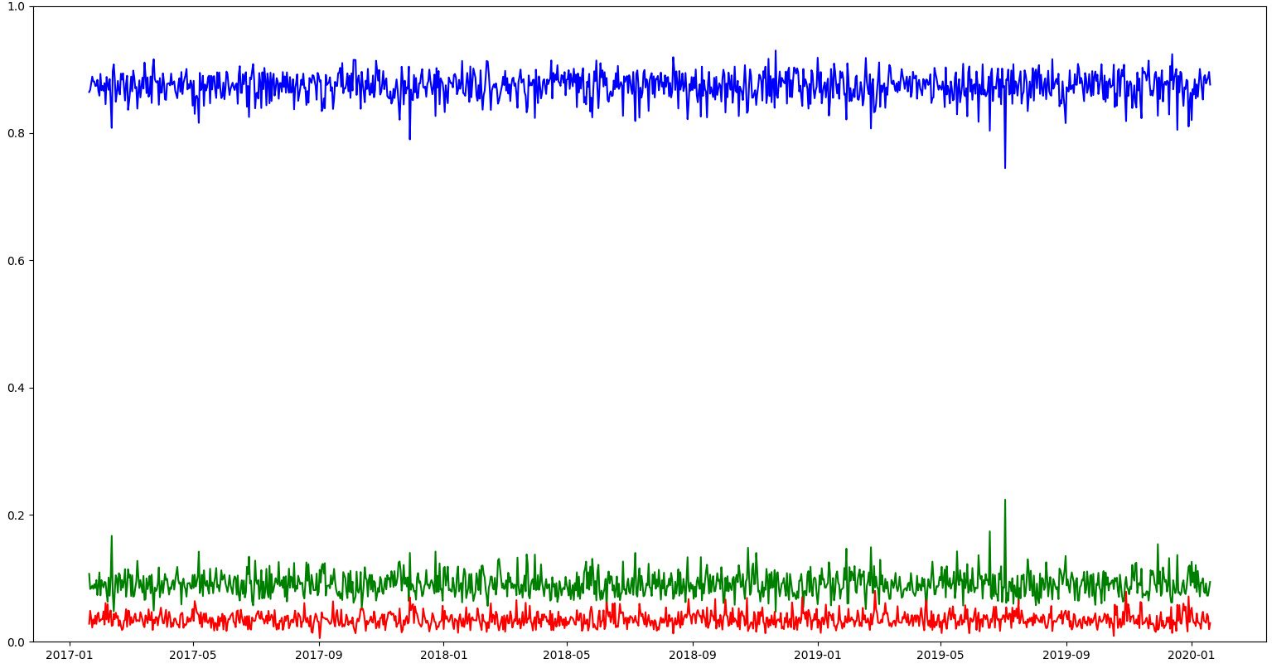
Trump Approval Difference (Approve - Disapprove)



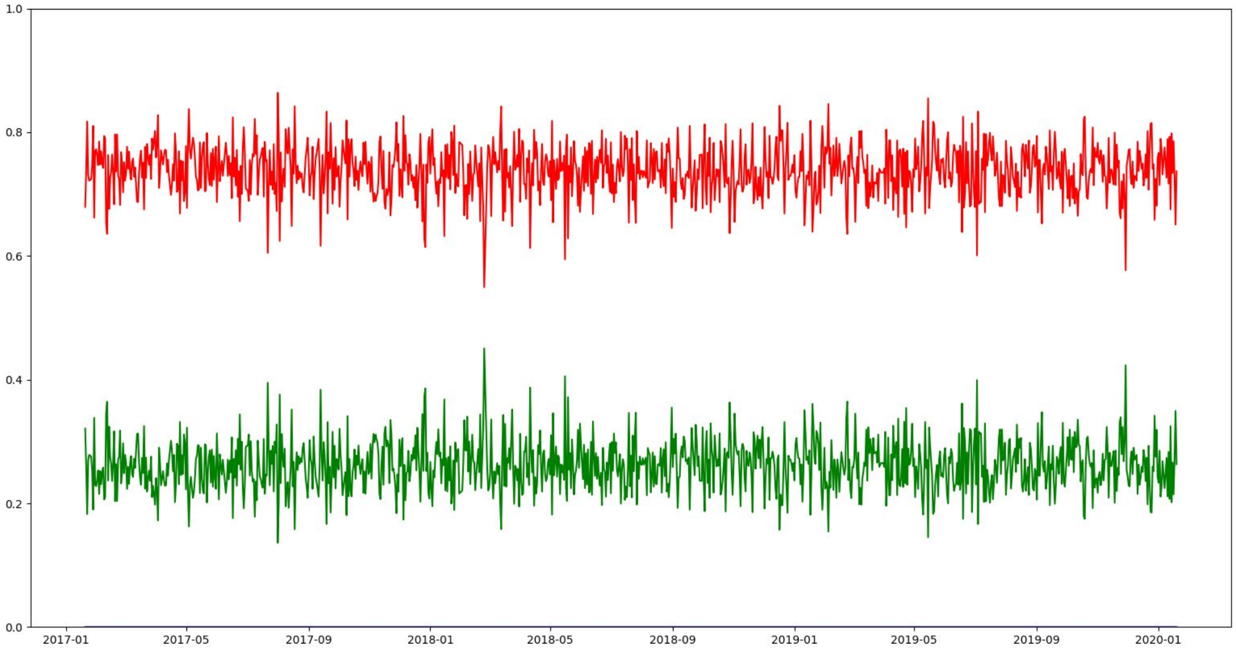
NLTK


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Textblob

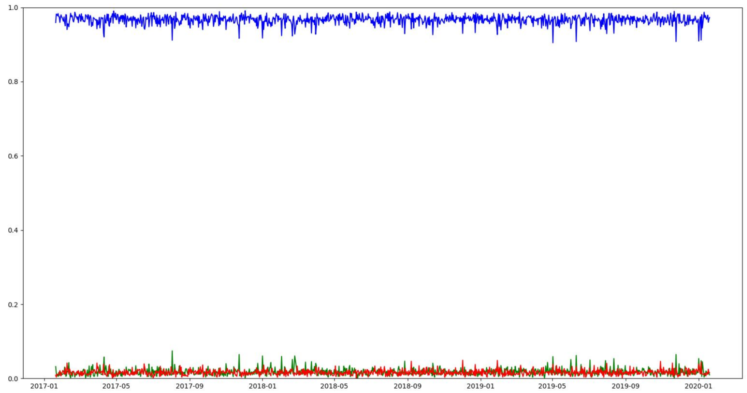

1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

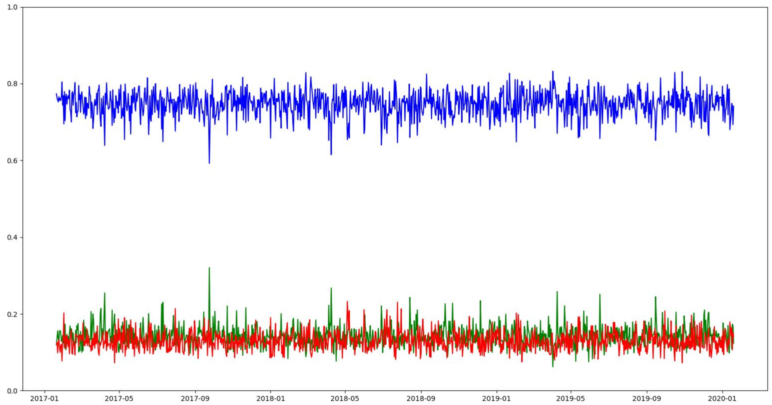
Flair



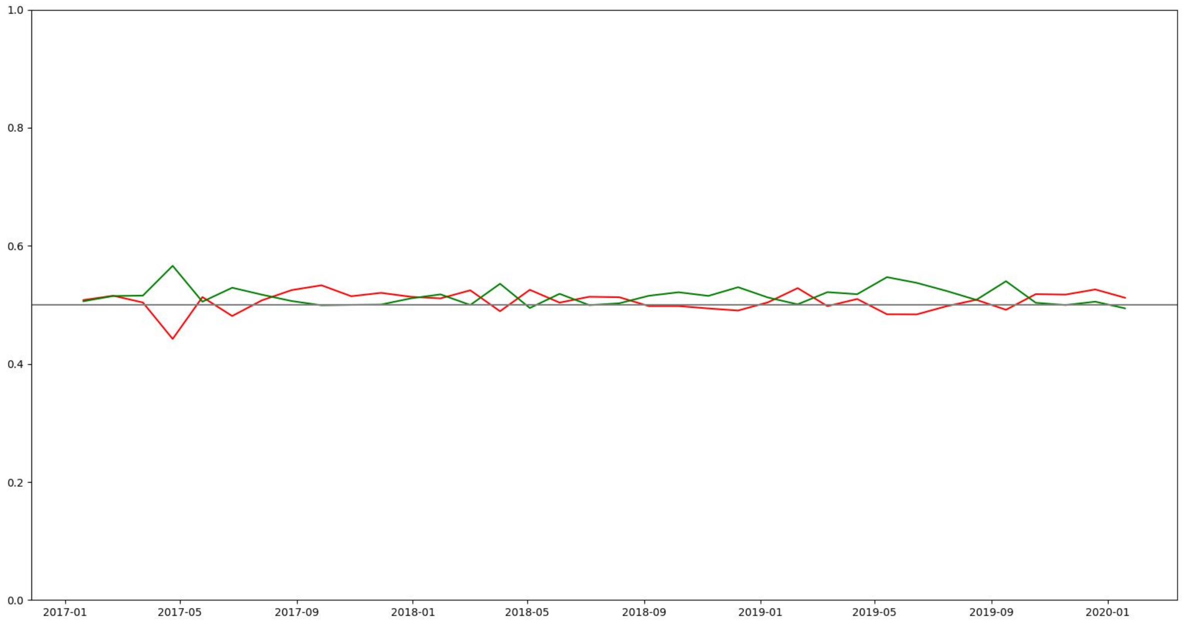
1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

NLTK

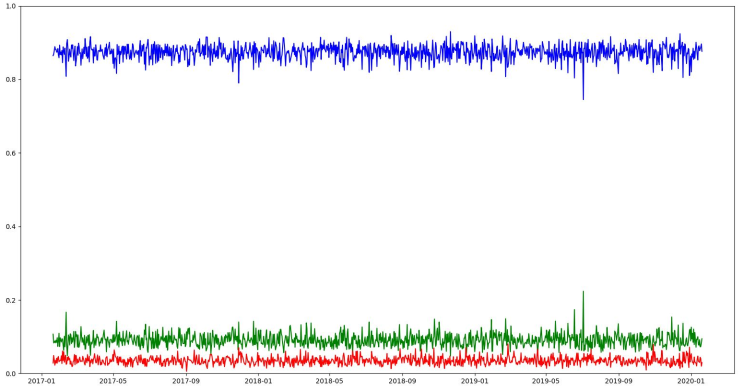

1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

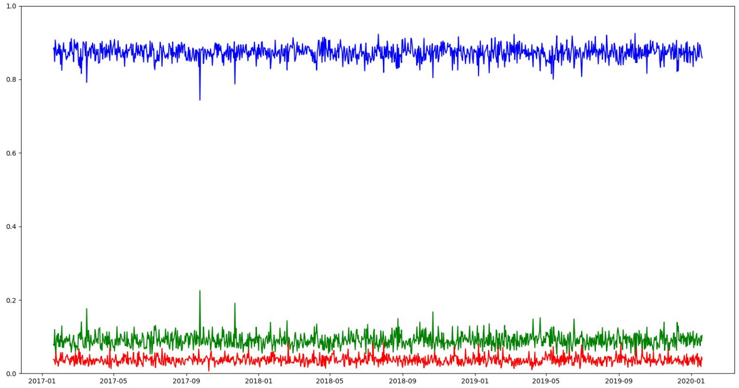
NLTK


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

TextBlob

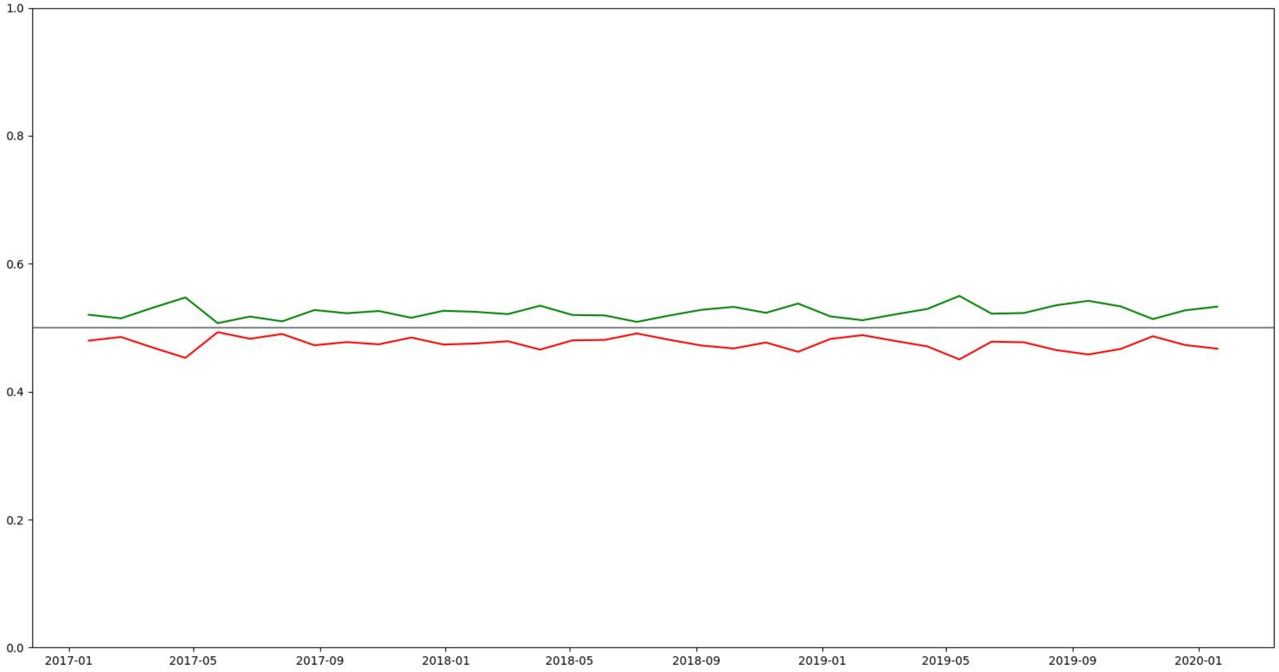


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

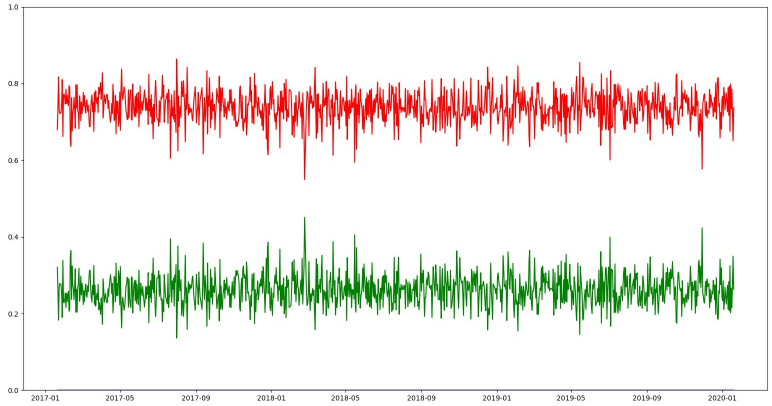


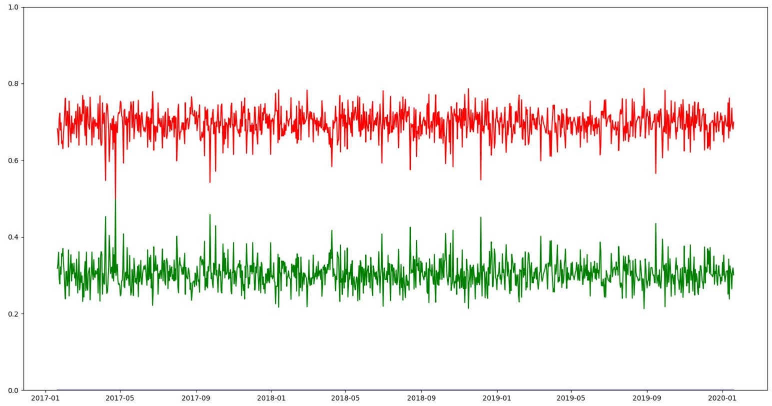
1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Textblob

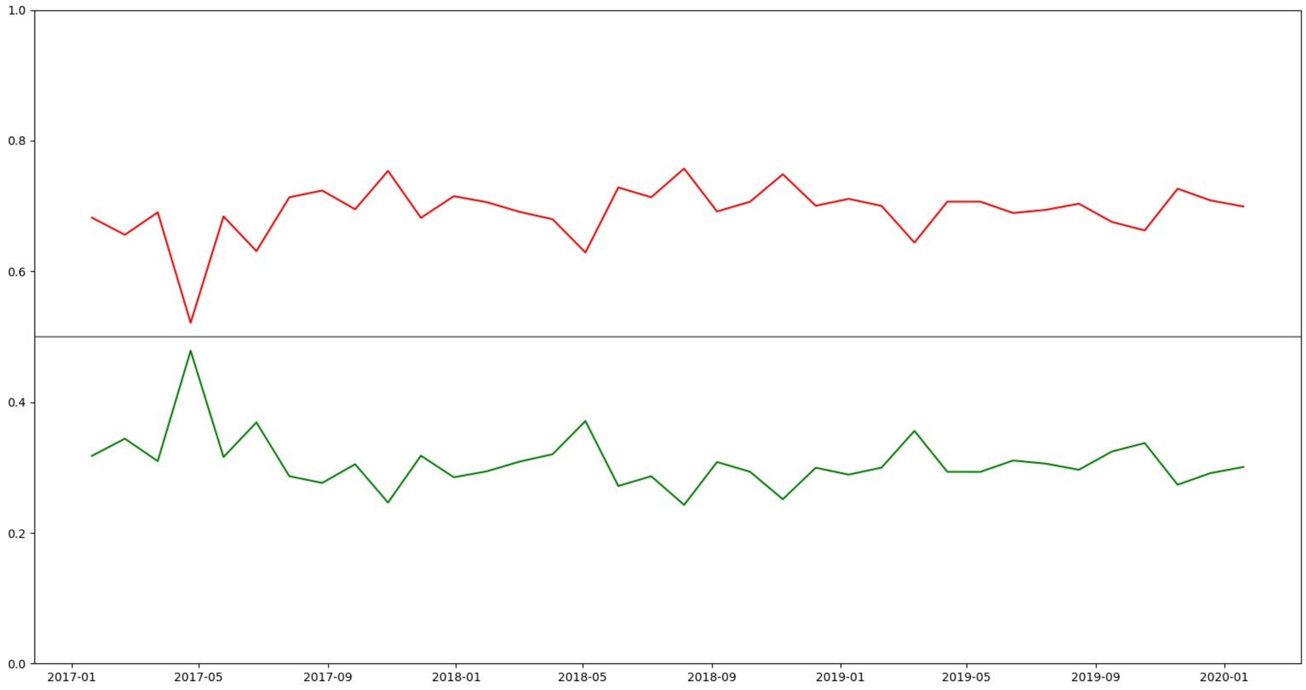

1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Flair


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

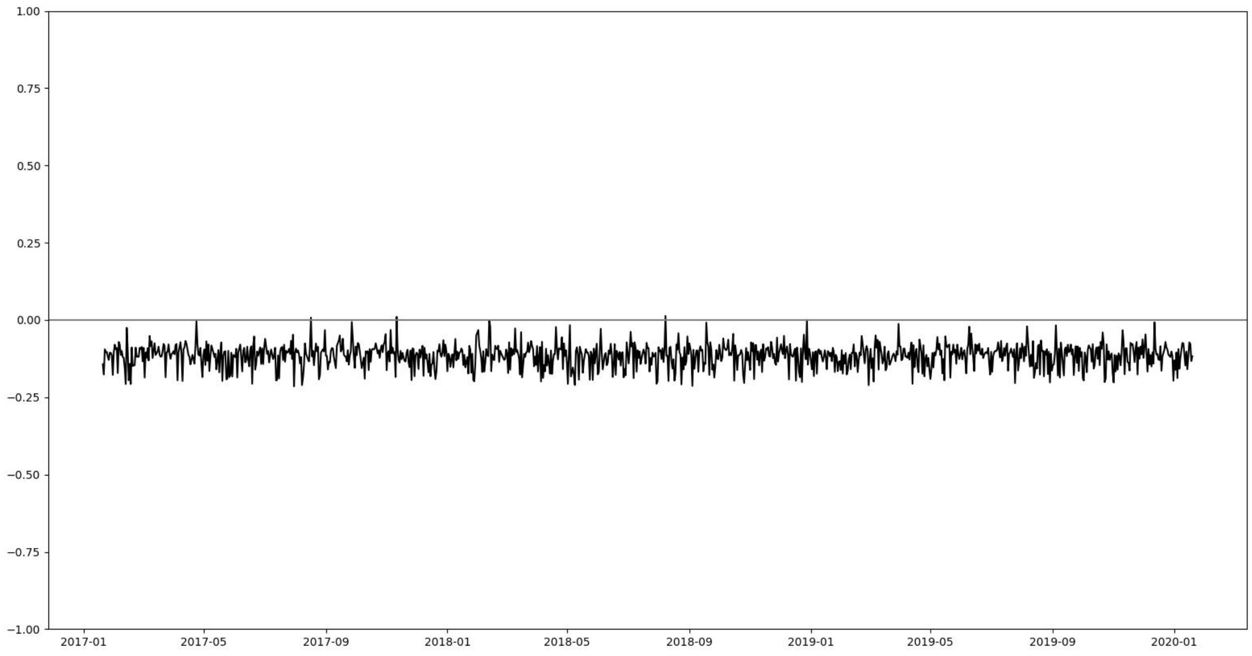

1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Flair

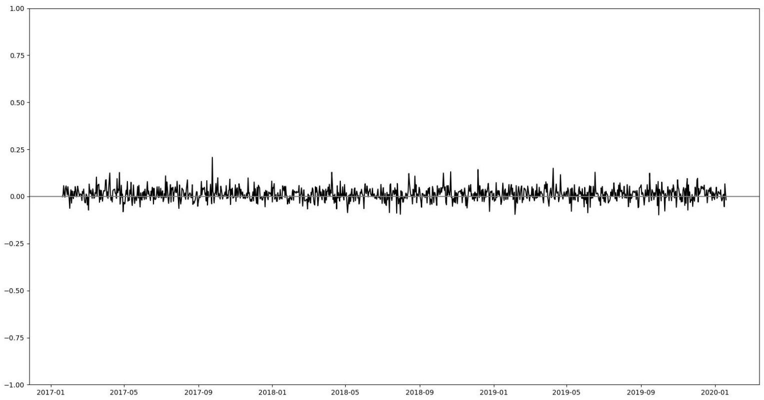


1.0
0.8
0.6
0.4
0.2
0.0
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

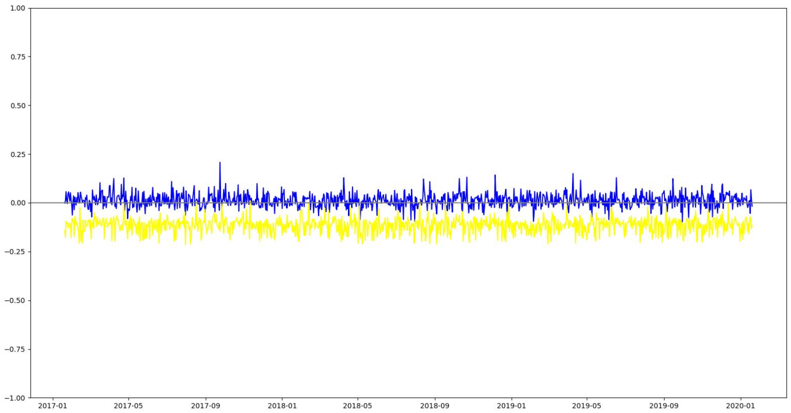
Trump Approval



NLTK

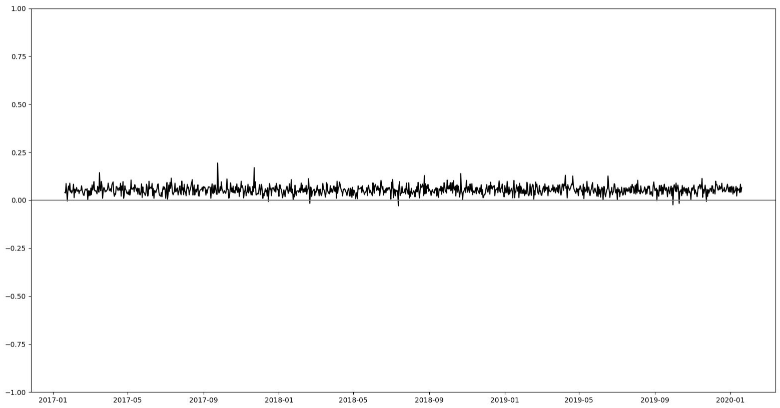


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

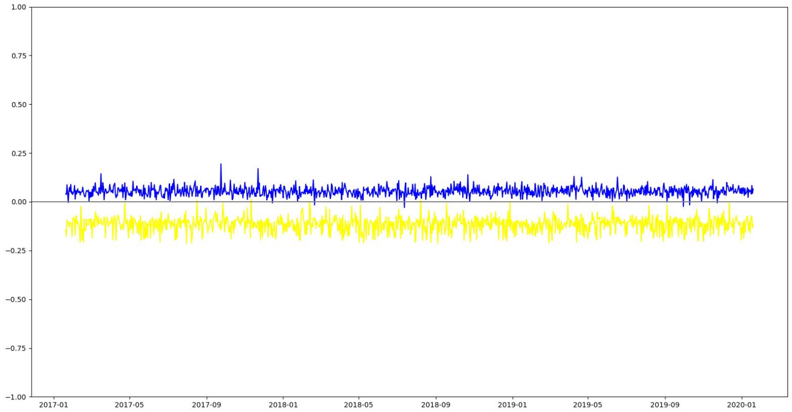


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Textblob

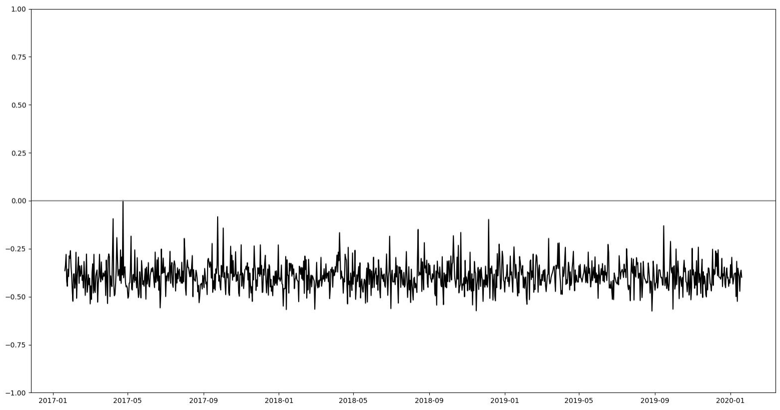


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

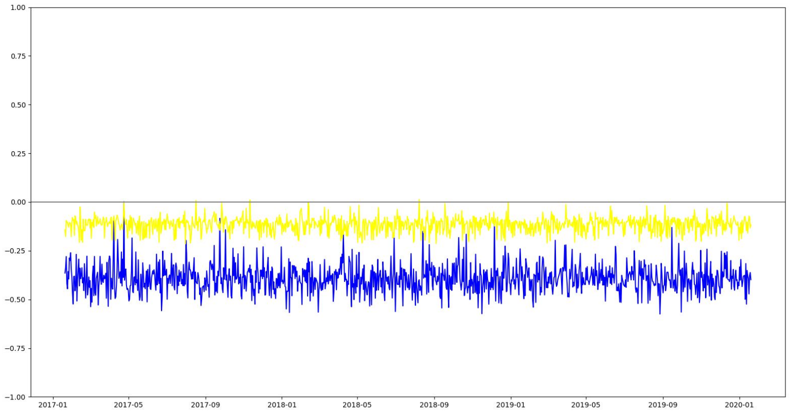


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

Flair

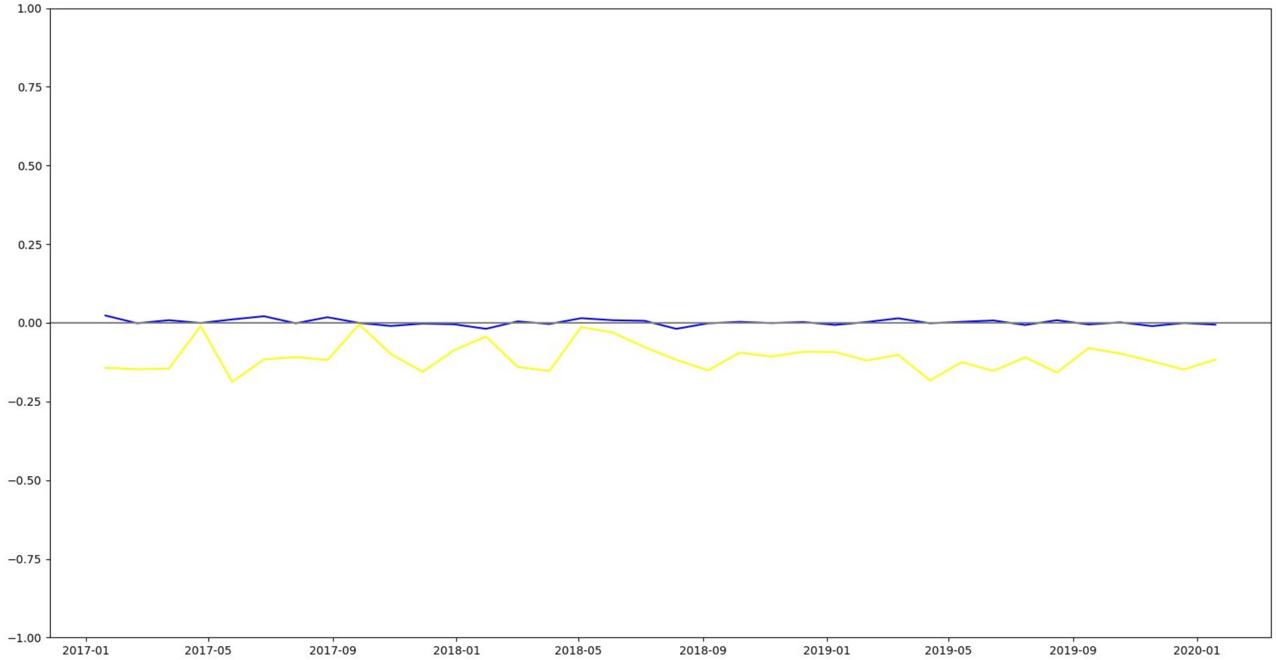


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

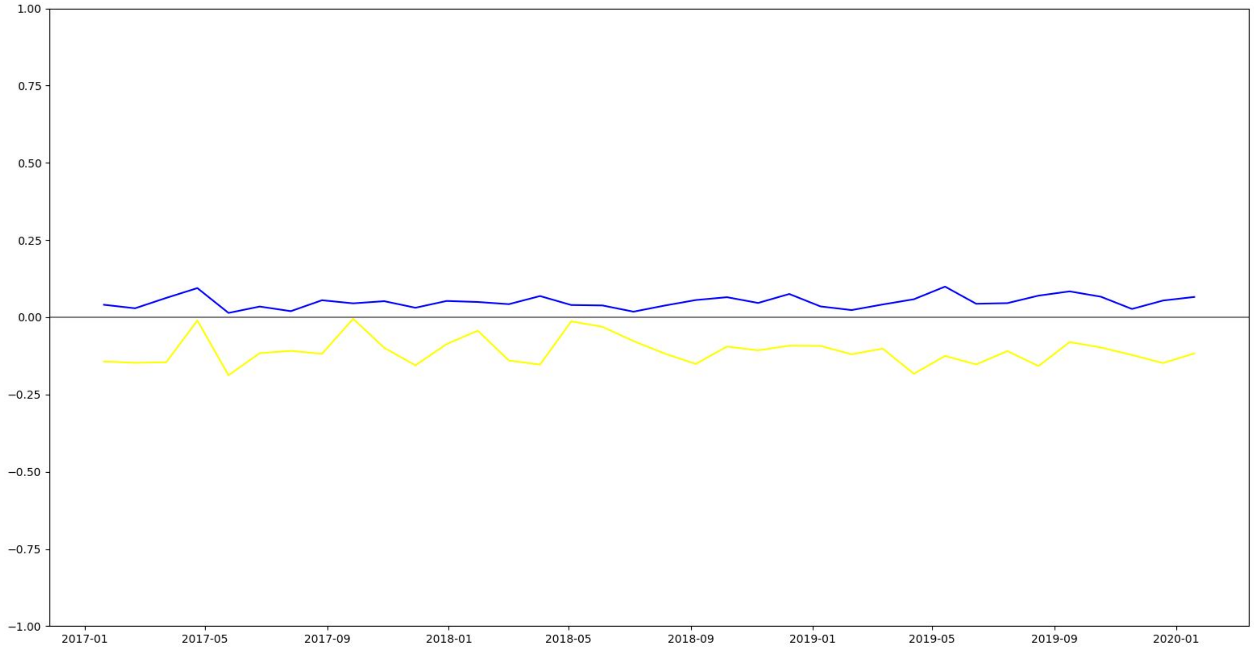


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01
2017-05
2017-09
2018-01
2018-05
2018-09
2019-01
2019-05
2019-09
2020-01

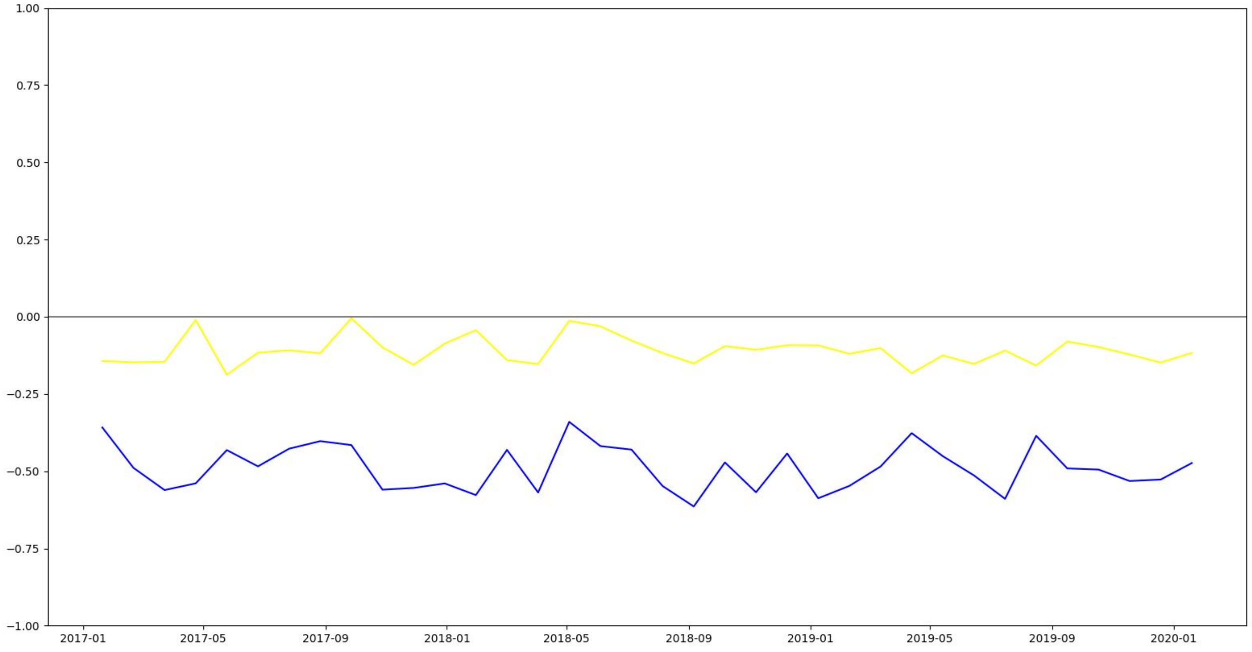
NLTK - 0.04775279 vs **0.11848797**



Textblob - 0.03088764 vs **0.13494615**



Flair - 0.06580662 vs **0.26252651**



4.Future experiments/work? (What would you want to do next, given more time?)

I'd like to find labeled data from Twitter and re-train the models used here on genre-specific data. The models here weren't trained on Twitter data, specifically, and could have improved performance if retrained and increased accuracy and better labeled which proportion of Tweets are positive and negative compared to poll data. I'd also like to get Tweets sorted by popularity, which mention @realDonaldTrump. That would probably be more indicative of more people's sentiment (if having more retweets and favorites), than a Tweet which has no interaction. More popular Tweets are also probably more readable, more grammatically correct, and may contain more words which convey sentiment more strongly.

5.Works Cited

1. https://github.com/flairNLP/flair
2. https://textblob.readthedocs.io/en/dev/index.html
3. https://www.nltk.org/api/nltk.sentiment.html
4. https://www.nltk.org/_modules/nltk/sentiment/vader.html
5. https://github.com/twintproject/twint
6. https://projects.fivethirtyeight.com/trump-approval-ratings/
7. https://data.world/fivethirtyeight/trump-approval-ratings