Media bias

Michael Kuhn

Herman Schaumburg

May 9, 2020

Goals

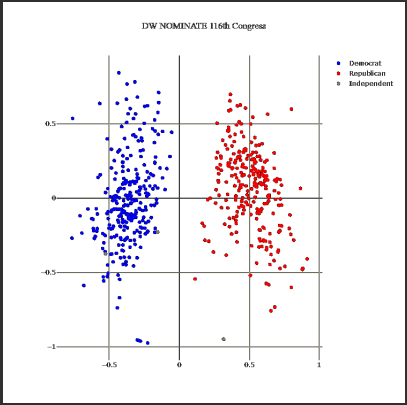
Primary: Find objective way to score bias in 36 selected news organizations.

Secondary: Estimate the political bias of a given tweet.

Objective Media Bias Rankings

www.mediabias.herokuapp.com

DW-NOMINATE scores



Methodology

1. Gathered tweets from Democratic lawmakers than Republicans (38,491 tweets vs 15,826 tweets) from Alex Litel github repository into SQL database.

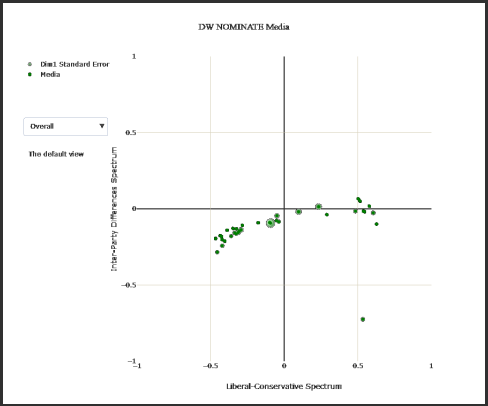
   https://github.com/alexlitel/congresstweets

2. DW-NOMINATE scores of congress people from voteview

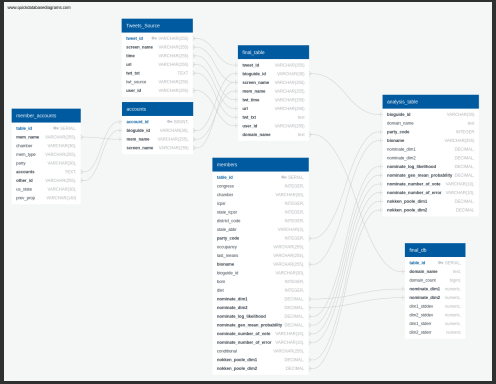
   https://voteview.com/data

3. Queried tweets of congresspeople for the media domains they tweet from and used the congressperson DW-NOMINATE score to score the media domain.

Media domain scores



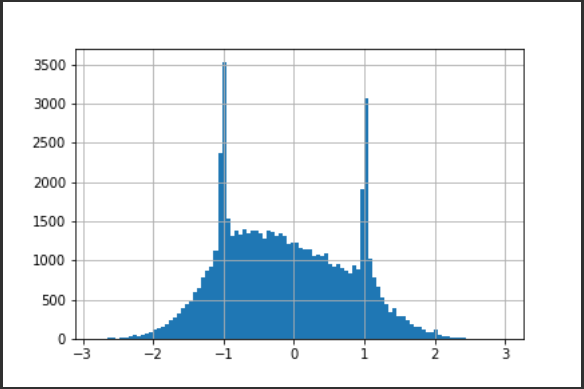
SQL Schema


www.quickdatasetsgenerator.com
tweets_source
tweet_id
screen_name
time
url
twt_txt
twt_source
user_id
accounts
account_id
blogpulse_id
mem_name
screen_name
member_accounts
table_id
mem_name
chamber
mem_type
party
assesments
other_id
cc_state
prev_prop
final_table
tweet_id
blogpulse_id
screen_name
mem_name
url
twt_time
url
twt_txt
user_id
domain_name
analysis_table
blogpulse_id
domain_name
party_code
blogname
nominate_dim1
nominate_dim2
nominate_log_likelihood
nominate_geo_mean_probability
nominate_number_of_vote
nominate_number_of_error
nominate_dim1_tstdev
nominate_dim2_tstdev
members
table_id
congress
chamber
type
state_icpsr
district_code
state_abbr
party_code
occupancy
last_means
blogname
blogpulse_id
born
died
nominate_dim1
nominate_dim2
nominate_log_likelihood
nominate_geo_mean_probability
nominate_number_of_vote
nominate_number_of_error
conditional
noklean_poole_dim1
noklean_poole_dim2
final_db
table_id
domain_name
domain_count
nominate_dim1
nominate_dim2
dim1_tstdev
dim2_tstdev
dim1_stderr
dim2_stderr

ML Model

1. Use nltk to remove stopwords from tweets.
2. Tweets are vectorized using several schemes.
3. Best performing scheme is selected for fine tuning.
4. End result is Stochastic Gradient Decent Model with X score.
5. Empirical CDF of errors in large test used to estimate probability of making an error.

Test for normality failed (couldn’t use prediction interval)



Empirical CDF

Heroku deployment features

- About page - interactive plots.
- ML models page - text box to predict party of tweeter.
- Search Media Scores Page - text box to search SQL data base for media domain.

SQL Query

```
select domain_name, domain_count, nominate_dim1, nominate_dim2, dim1_stddev,
dim2_stddev, round((dim1_stddev/sqrt(domain_count-1))::numeric,3) as dim1_stderr,
round((dim2_stddev/sqrt(domain_count-1))::numeric,3) as dim2_stderr
into final_db
from tenthousand_db
where domain_count>1
group by domain_name, domain_count, nominate_dim1, nominate_dim2, dim1_stddev, dim2_stddev
order by domain_count desc
```

jsonToCSV

```python
def jsonToCSV(json_path, csv_path):
    import json
    import csv
    merged_csv = []
    with open(json_path, encoding='utf-8') as ref:
        data = json.load(ref)
        headers = list(data[0].keys())
        csv_row = []
        for item in data:
            item_ls = []
            for col in headers:
                try:
                    item_ls.append(item[col])
                except:
                    item_ls.append(None)
            csv_row.append(item_ls)
        merged_csv += csv_row

    with open(csv_path, 'w', newline='', encoding='utf-8') as csvfile:
        spamwriter = csv.writer(csvfile)
        for row in merged_csv:
            spamwriter.writerow(row)
```