Twits: perusing GNIP Twitter data

By Dodge Coates

Lots and lots of Tweets

Lots and lots of Tweets

- ~2 terabytes in size, uncompressed

Lots of data

Lots of data

- ~5,500 json files

Lots of data

- ~5,500 json files
- ~2 terabytes in size, uncompressed

Lots of data

- ~5,500 json files
- ~2 terabytes in size, uncompressed'
- This was reduced to ~250 gigabytes of csv

Lots of data

- ~5,500 json files
- ~2 terabytes in size, uncompressed
- This was reduced to ~250 gigabytes of csv
- No schema available for the dataset

Gnip Enterprise Access to Twitter Data

- Provide 1/10 of the twitter firehose
- Accessed for roughly 2 ½ weeks, between late January and mid February, 2012

Data Sample


```
[
    "id_str":"163372337134182401",
    "in_reply_to_status_id":null,
    "created_at":"Sat Jan 28 21:26:15 +0000 2012",
    . . .
    "user":
        {
            "id_str":"44153313",
            "favourites_count":25,
            "contributors_enabled":false,
            . . .
            "lang":"en",
            "utc_offset":-21600,
            "profile_sidebar_border_color":"BDDCAD",
            "followers_count":349,
            "url":"http://kingsofkauffman.com/",
            "profile_image_url":"http://a1.twimg.com/profile_images/1442806914/My_Photo__normal.jpg"
        },
    "retweet_count":3,
    "favorited":false,
    "id":163372337134182401,
    "entities":
        {"hashtags":[
            {
                "text":"Mizzou",
                . . .
        },
        "retweeted_status":
            {
                "id_str":"163369042332229632",
                "place":null,"geo":null,
                "text":"Funny line from #Mizzou's Kim English about DGB: \"We were 100 percent focused on toda
                "in_reply_to_status_id_str":null,
                "coordinates":null,
                "user":
                {
                    "id_str":"168902884",
                    "favourites_count":0,
                    "profile_use_background_image":true,
                    "screen_name":"tpkcstar",
                    . . .
                    "favorited":false,
                    "id":163369042332229632,
                }
            }
        }
        }
```

Goals

Goals

1. Exploring the data

Goals

1. Exploring the data

2. Developing an interesting prediction

Goals

1. Exploring the data
2. Developing an interesting prediction
3. Get a feel for working with a large dataset

The Data

- A lot to process!

Storage

- SQL

Storage

- SQL
- Distributed file systems (Hadoop)
  - Hardware is too expensive
  - API is too time-consuming

Storage

- SQL
- Distributed file systems (Hadoop)
  - Hardware is too expensive
  - API is too time-consuming
- Settled on sampling

Preprocessing

Preprocessing

- Takes about 20-30 hours on my desktop, depending on the complexity of the preprocessing

Preprocessing

- Takes about 20-30 hours on my desktop, depending on the complexity of the preprocessing
- Produces ~5,500 csv files

Preprocessing

- Takes about 20-30 hours on my desktop, depending on the complexity of the preprocessing
- Produces ~5,500 csv files
-

Sampling

- A sampling script is used to to uniformly sample the thousands of files

Sampling

- A sampling script is used to to uniformly sample the thousands of files
- Because each file represents a 3 minute portion of twitter activity, there is high variance between file sizes

Sampling

- A sampling script is used to to uniformly sample the thousands of files
- Because each file represents a 3 minute portion of twitter activity, there is high variance between file sizes
- Therefore sampling must be done carefully to not distort time series data

Inspecting Data Features

- Complete json object contains ~130 attributes
- Components:
  - User data
  - Tweet data
  - Original retweet data (a copy of the outer user/tweet data)
- Reduced to 38 features in the processed data

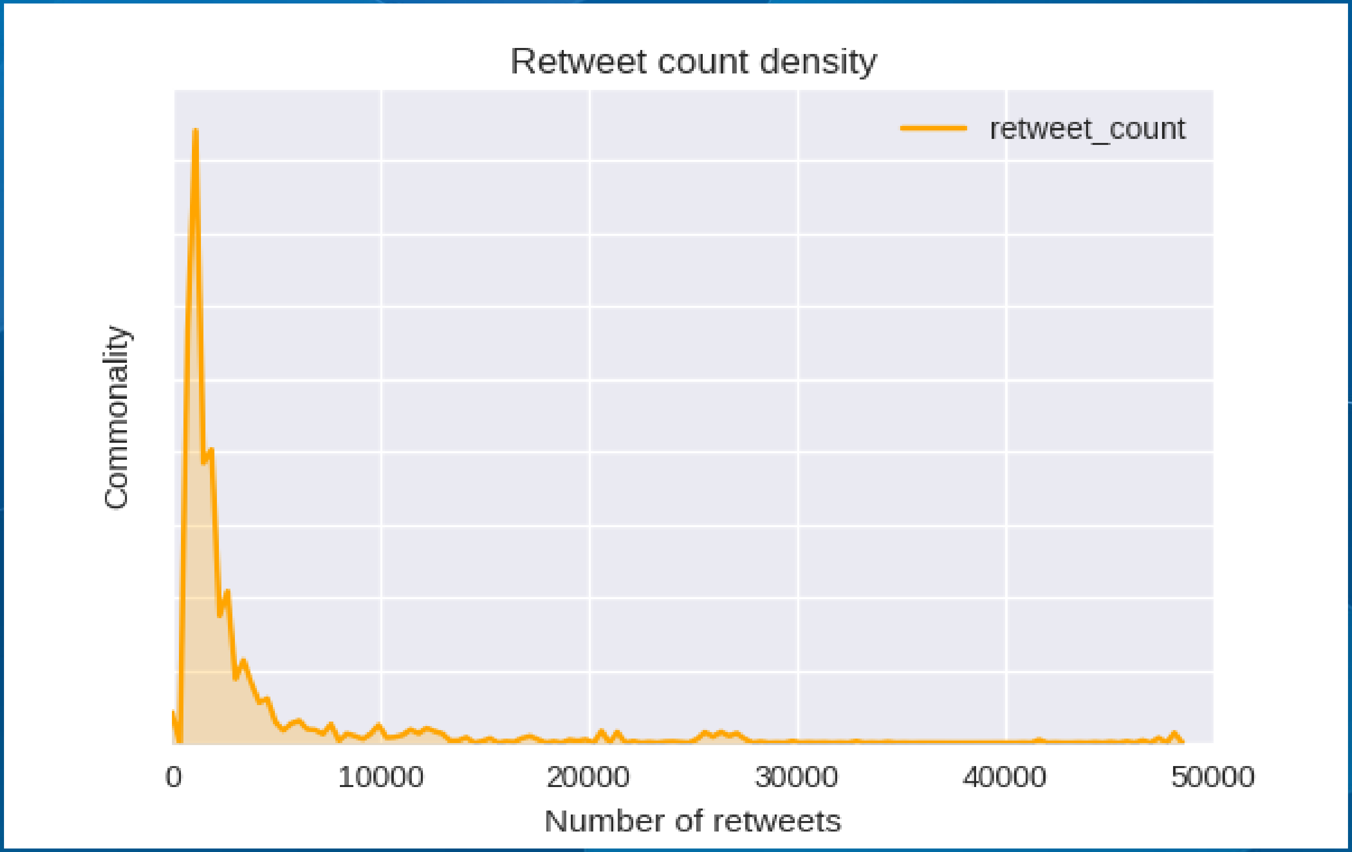
Developing a Model Target: what to look for?

retweet_count

- The current retweet number for a given tweet

retweet_count

- The current retweet number for a given tweet
  - Not including zero retweets



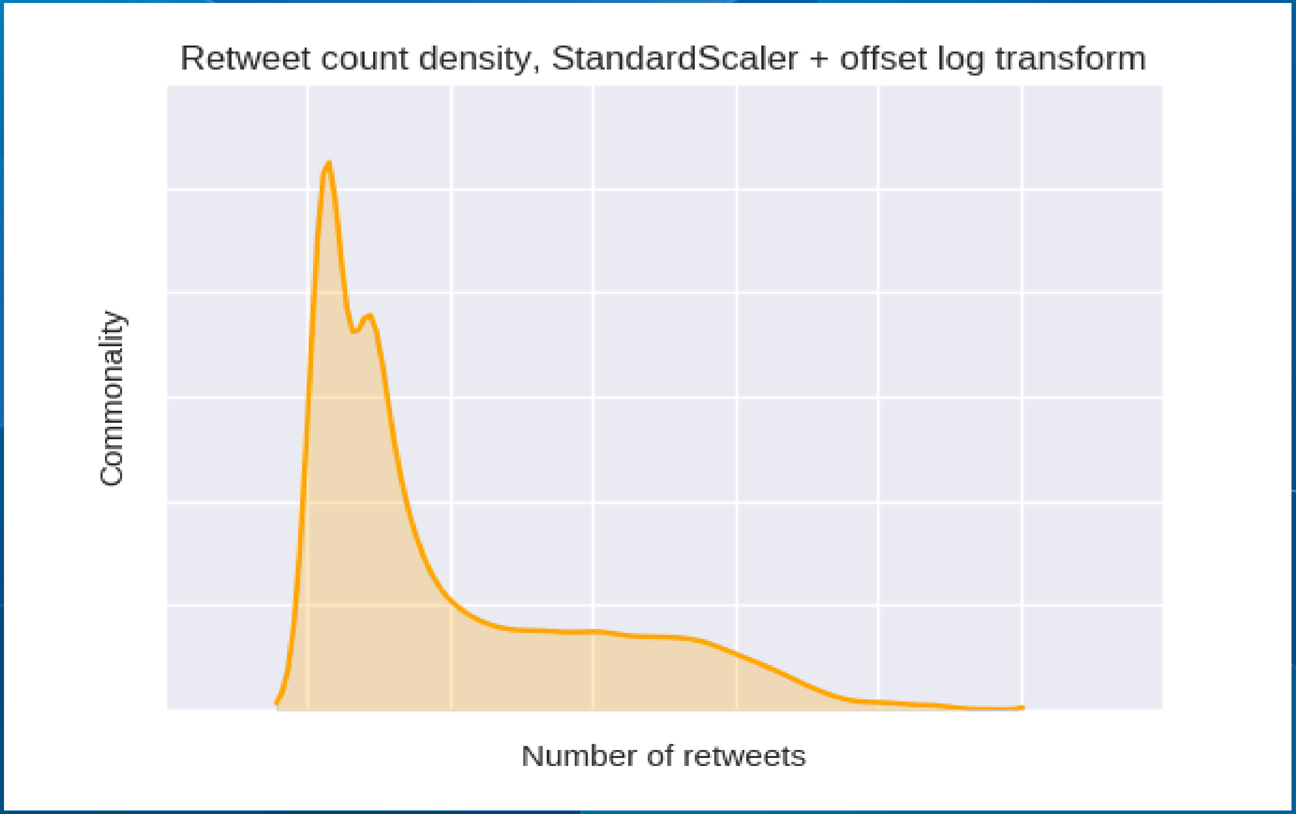
Transforming retweet_count

Transforming retweet_count

- sklearn.preprocessing.StandardScaler applied to a log transform (offset by 1)

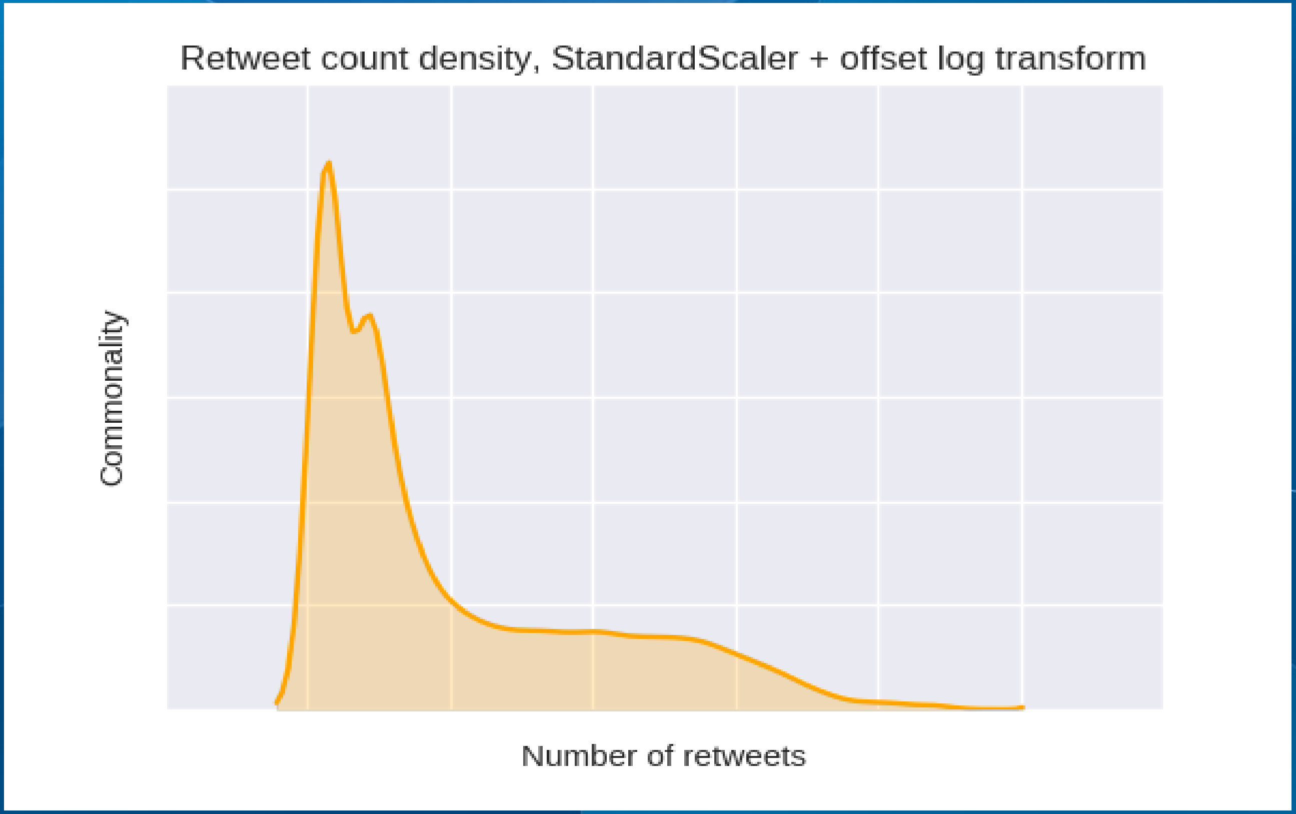
Transforming retweet_count

- sklearn.preprocessing.StandardScaler applied to a log transform (offset by 1)



Transforming retweet_count

- sklearn.preprocessing.StandardScaler applied to a log transform (offset by 1)



Transforming retweet_count

Doesn't work.

retweet_count is a bad target

- Very highly correlated with number of followers

retweet_count is bad

- Very highly correlated with number of followers
- Very easy to predict

retweet_count is bad

- Very highly correlated with number of followers
- Very easy to predict
- Not very interesting

Tweetability

Tweetability

- retweet_count / followers_count

Target: Tweetability

- retweet_count / followers_count
- More interesting idea for getting at the "goodness" of a tweet

Target: Tweetability

- retweet_count / followers_count
- More interesting idea for getting at the “goodness” of a tweet
- Foolish to presume that tweetability corresponds with “goodness”?

Target: Tweetability

- retweet_count / followers_count
- More interesting idea for getting at the "goodness" of a tweet
- Foolish to presume that tweetability corresponds with "goodness"?
  - Probably.

Target Penalty

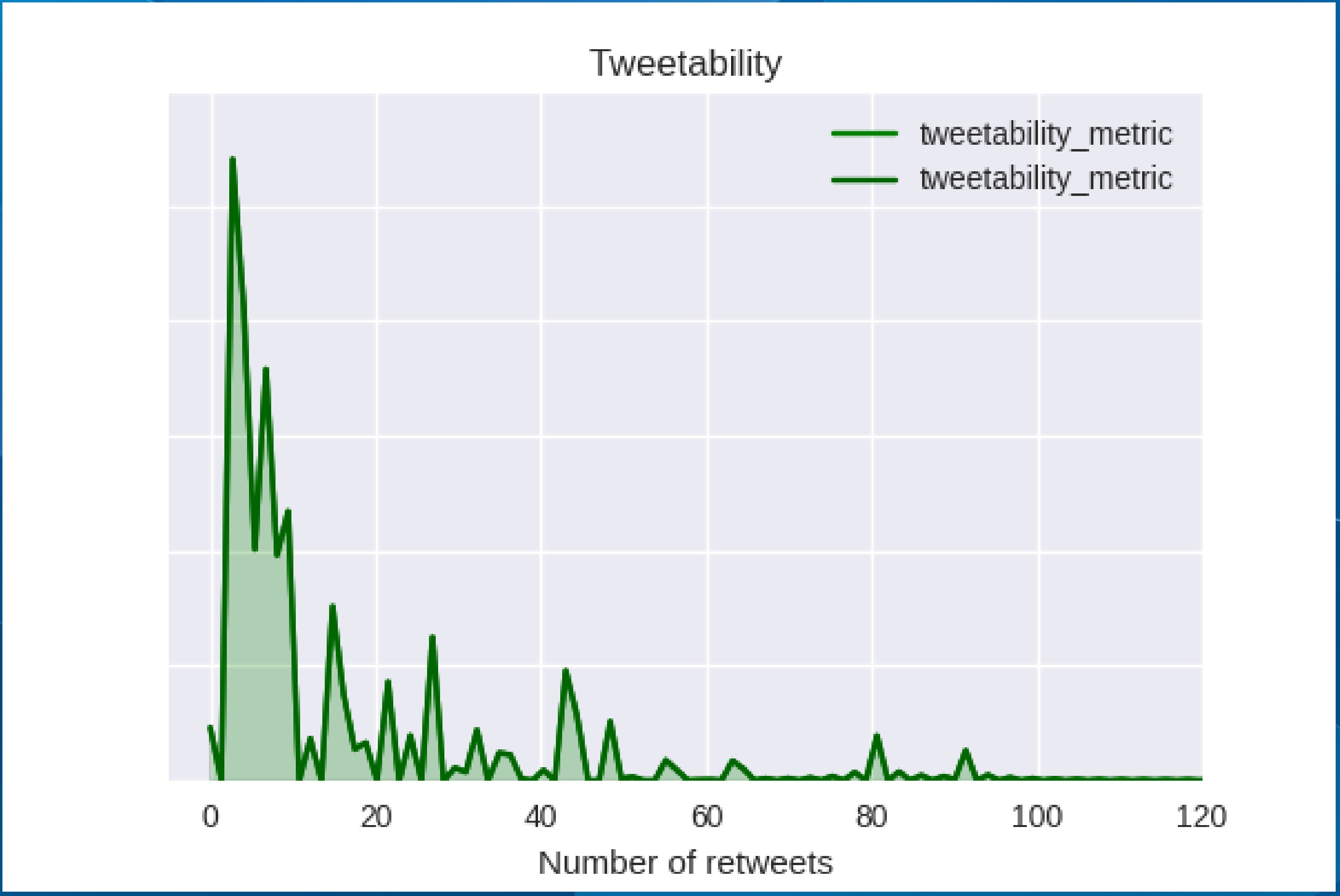
- Add a constant penalty
  - Low retweet counts should be disproportionately rewarded for high retweet/followers ratio
  - Goal is not to maximize predictability

Tweetability

- Add a constant penalty
  - Low retweet counts should be disproportionately rewarded for high retweet/followers ratio
  - Goal is not to maximize predictability
- Add a penalty for diminishing returns

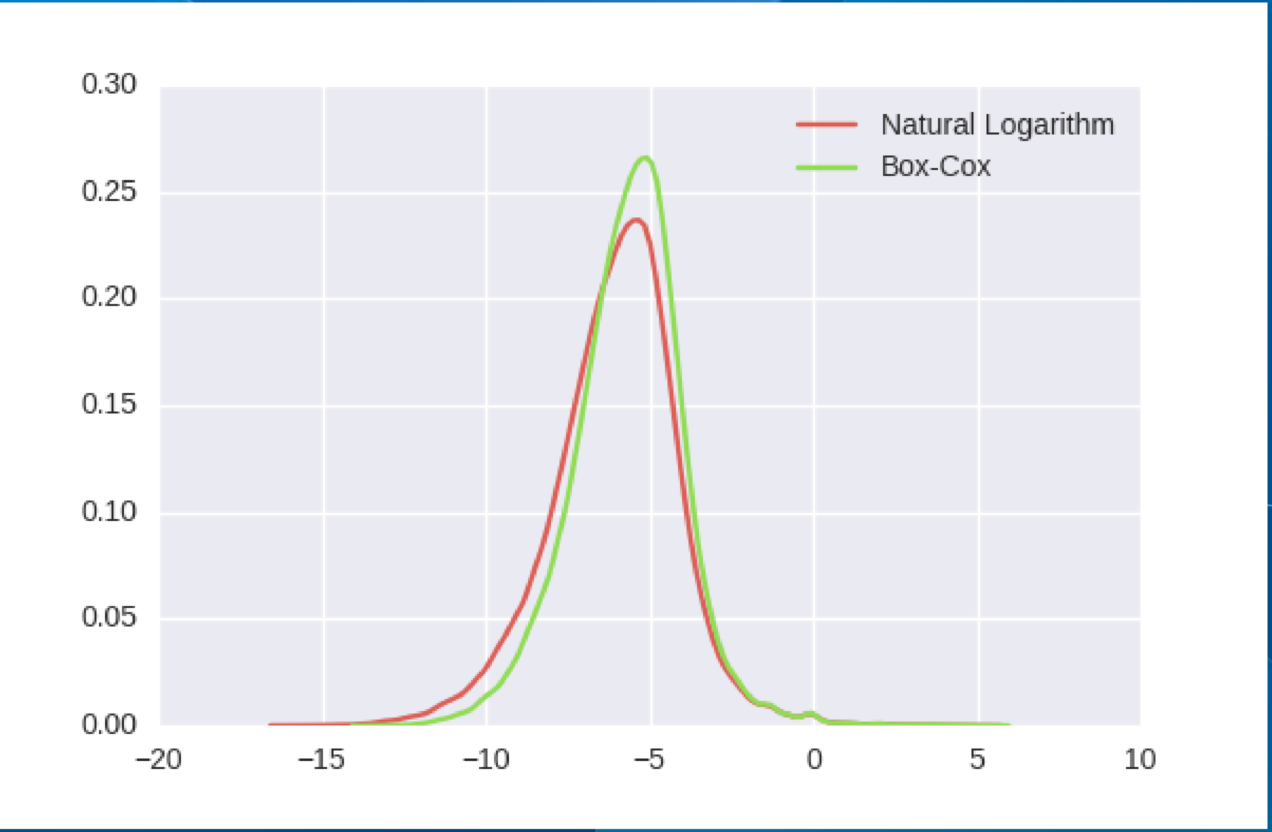
$$\frac{retweetcount + \log(followerscount)}{followerscount + C / retweetcount}$$

Tranforming Tweetability



Transforming Tweetability

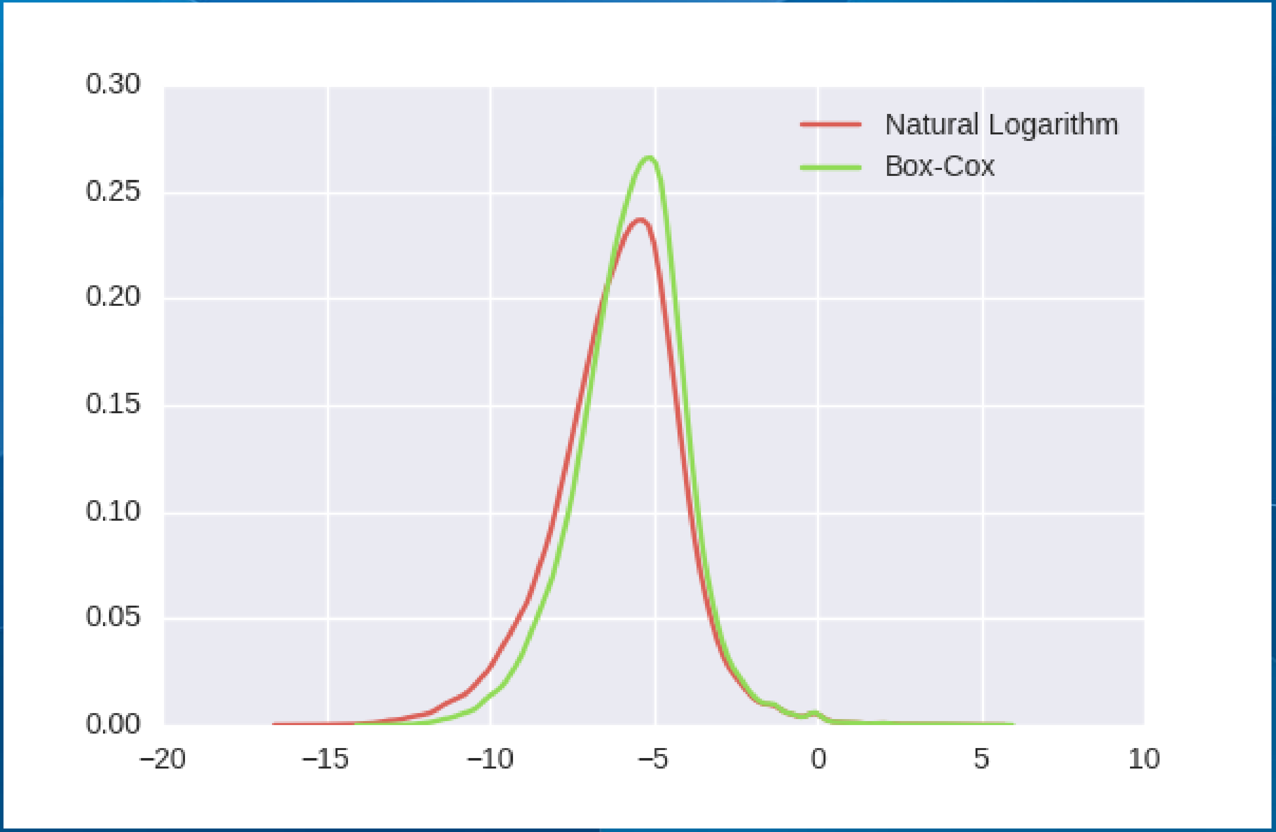
- After Transformations



- Minimized variance made partitioning better than k-means clustering for categorization

Transforming Tweetability

- After Transformations



Engineered features

- Hashtag count, user mention count
- sentiment polarity
- text diversity
- punctuation score
- word count
- hashtag popularity
- Etc.

Models

- RandomForest and simple decision trees quite well
- Both achieve about 70% accuracy, but high log loss
- Interestingly, the additional features past three or four core features (statuses count, favorites count, etc) barely improved accuracy (~1 point) and often harmed log loss.

Models (cont.)

- Voting classifier
  - Consists of Random forest, Naive Bayes, and Logistic regression estimators
  - Originally reduced log loss
  - Yet to run again on corrected data

Takeaways and thoughts

- Large datasets are brutal.
- Carefully examine foreign data, particularly variable json, for structure and message, not just content
  - The invested time is worth it.
- Feature engineering can be very unintuitive