Predicting Fake News on Facebook

Marissa Graham

CS 401R Final Project

Motivation

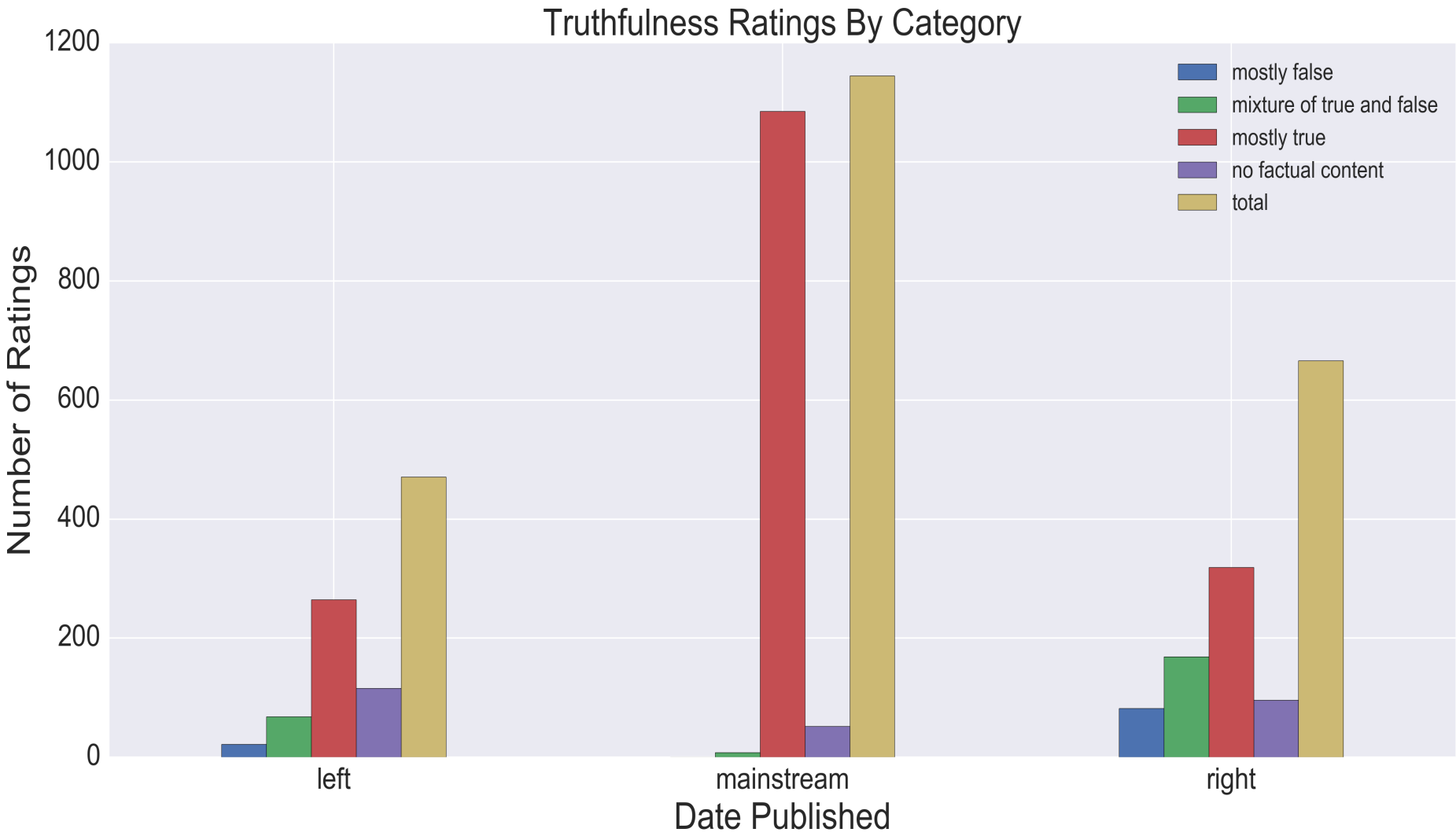
- Fake news has become more prevalent in U.S. politics.
- Democracy can't work unless citizens have accurate information.

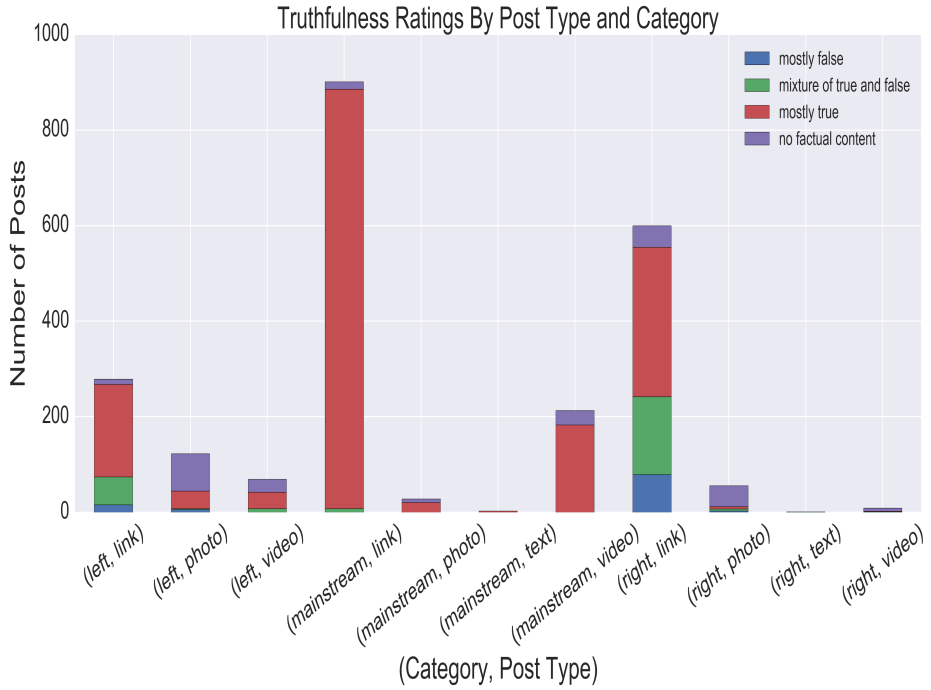
Dataset

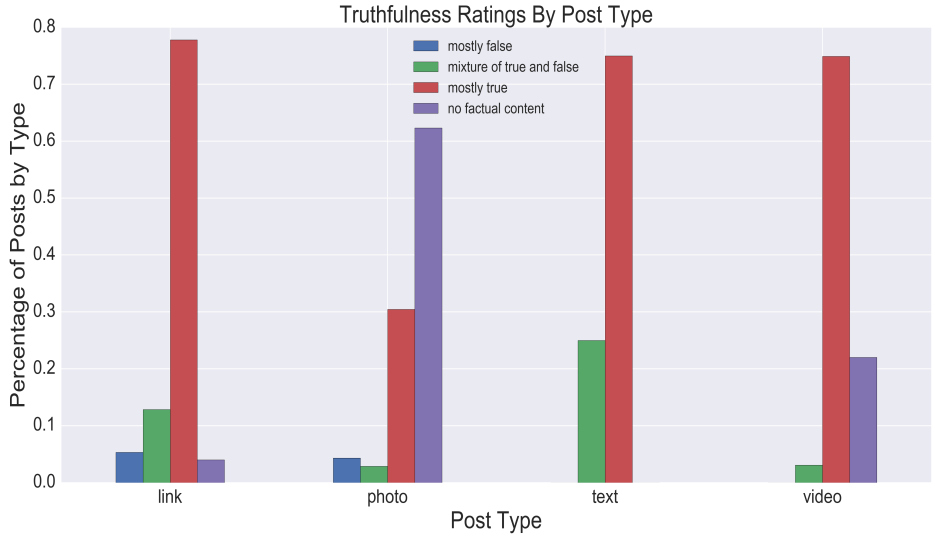
Being able to predict/flag fake news with some amount of accuracy using machine learning tools is helpful.

Dataset used:

- Sponsored by Buzzfeed
- 2284 Facebook posts from nine different news outlet pages (three left-wing, three right-wing, three mainstream) between 9/19/2016 and 9/27/2016, classified by accuracy level
- Relevant features:
  - `Category` (mainstream, right, or left)
  - `Page`
  - `Post Type` (video, link, text, photo)
  - `Rating` (mostly false, mostly true, mixture of true and false, no factual content)


Truthfulness Ratings By Category
Number of Ratings
1200
1000
800
600
400
200
0
left
mainstream
right
Date Published
mostly false
mixture of true and false
mostly true
no factual content
total


Truthfulness Ratings By Post Type and Category
Number of Posts
1000
800
600
400
200
0
mostly false
mixture of true and false
mostly true
no factual content
(left, link)
(left, photo)
(left, video)
(mainstream, link)
(mainstream, photo)
(mainstream, text)
(mainstream, video)
(right, link)
(right, photo)
(right, text)
(right, video)
(Category, Post Type)


Truthfulness Ratings By Post Type
Percentage of Posts by Type
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
mostly false
mixture of true and false
mostly true
no factual content
link
photo
text
video
Post Type

Approach

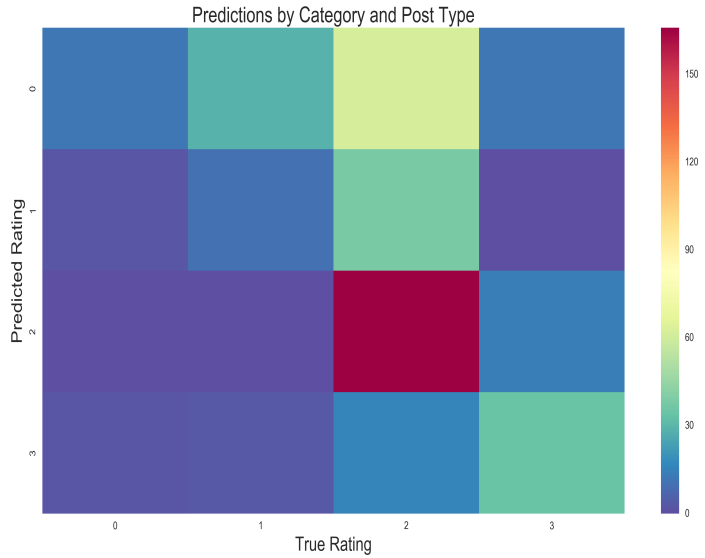
- Since our data is a bunch of independent features with categorical outputs, we choose a Naive Bayes-based method.
- After making visualizations of the distributions of ratings across each feature of the data, `Category` or `Page` and `Post Type` are the only ones that are categorical, independent, generally available immediately after a post is published, and seem to have an effect on the ratings distribution.

Approach (Outline)

- Read in data and replace categorical data with ints for ease of use, partition into test and training data
- For each relevant feature, get the array `A` such that `A[i,j]` gives the probability of the `j`-th possibility of the feature given the `i`-th possibility for the rating. So $P(a_j = v_j|C_i)$, where a_j is the feature, C_i are the classes for the ratings, and v is the test point.
- $P(C_i) = \Pi_j P(A_j = v_j|C_i)$
- Take the argmax to get the prediction (which C_i has the highest probability?)

Initial Results

- Small dataset with low predictive value, results are not good.
- Error rates around 40% (varies depending on test split), even though marking everything as true gives you 30% since about 70% of posts are "mostly true".
- We need to reevaluate goals and success measurement.



Flagging Methods

- Instead of trying to predict the exact truthfulness rating of every post, we attempt to flag potentially false posts for further investigation.
- Measure success with sensitivity (true positives, or how many of the false posts did we catch) and specificity (true negatives, or how many of the not-false posts did we mark as not-false)
- Use the probabilities C_i obtained from the Naive Bayes model
- Goal: High sensitivity, reasonable specificity

Results

- Use Bayes' theorem to estimate probability that a post is false, given that we flagged it.
- Before: 4.6% of posts rated "mostly false"
- After: about 10% of flagged posts rated "mostly false", with no false posts left behind

Method	**Sensitivity**	**Specificity**
Report false always (positive control)	100%	0%
Report false never (negative control)	0%	100%
Report false if "mostly false" has highest probability	64.28%	74.87%
Report false if "mostly false" or "mixture of true and false" has highest probability	92.85%	61.6%
Report false if "mostly false" is highest or second highest probability	100.00%	54.92%