Sarcasm detection

Andrew vanderWilden

8/24/2021

This analysis uses a kaggle dataset containing headlines from both The Onion and the Huffington Post. The goal of the analysis is to try to predict if the article is a sarcastic (Onion) or real (Huffington Post) story using only the headline. We perform a lasso regression after tokenization and tfidf transformations to determine words most associated with each category.

```
df <- tibble(stream_in(file('Sarcasm_Headlines_Dataset_v2.json')))
```

```
##  Found 500 records... Found 1000 records... Found 1500 records... Found 2000 records... Found 2500 re
```

```
df <- df %>%
  mutate(is_sarcastic = factor(case_when(is_sarcastic == 1~'Sarcasm',
                                         TRUE~'Not_Sarcasm')),
         is_sarcastic = relevel(is_sarcastic, ref = 'Sarcasm'))
```

```
library(tidymodels)
tidymodels_prefer()
```

```
set.seed(2917)

df_split <- initial_split(df, strata = is_sarcastic)
df_train <- training(df_split)
df_test <- testing(df_split)

df_folds <- vfold_cv(df_train, strata = is_sarcastic)
```

Lasso Model

```
library(textrecipes)

sarcasm_rec <- recipe(is_sarcastic ~ ., data = df_train) %>%
  update_role(article_link, new_role = 'link') %>%
  step_tokenize(headline) %>%
  step_stopwords(headline) %>%
  step_tokenfilter(headline, max_tokens = 1000) %>%
  step_tfidf(headline) %>%
  step_normalize(all_predictors())

prep(sarcasm_rec)
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##       link          1
##    outcome          1
##  predictor          1
##
## Training data contained 21463 data points and no missing data.
##
## Operations:
##
## Tokenization for headline [trained]
## Stop word removal for headline [trained]
## Text filtering for headline [trained]
## Term frequency-inverse document frequency with headline [trained]
## Centering and scaling for tfidf_headline_1, ... [trained]
```

```
lasso_spec <- logistic_reg(penalty = tune(), mixture = 1) %>%
  set_engine('glmnet') %>%
  set_mode('classification')
```

```
lasso_wf <- workflow(sarcasm_rec, lasso_spec)
```

Tune parameters

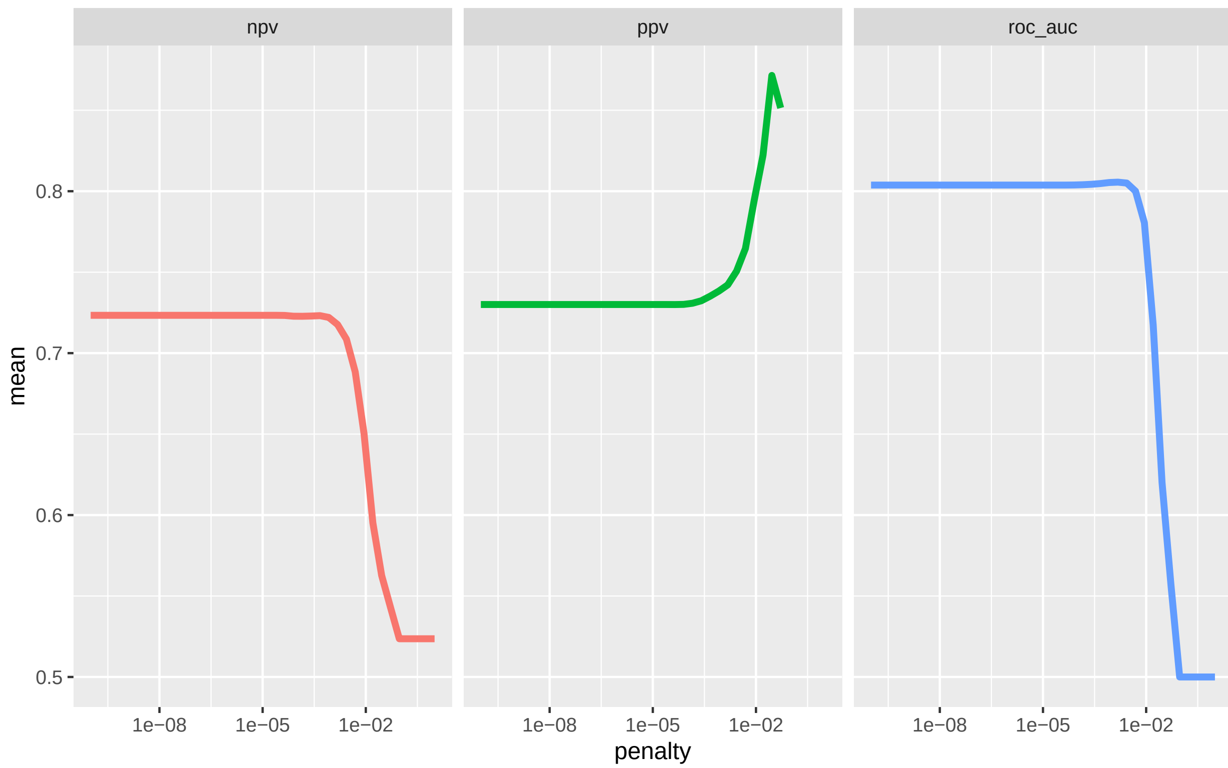
```
set.seed(3891)
lasso_grid <- grid_regular(penalty(), levels = 40)
```

```
cl <- parallel::makePSOCKcluster(3)
doParallel::registerDoParallel(cl)

set.seed(181)

lasso_res <- tune_grid(
  lasso_wf,
  resamples = df_folds,
  grid = lasso_grid,
  metrics = metric_set(roc_auc, npv, ppv)
)

collect_metrics(lasso_res)
```

```
## # A tibble: 120 x 7
##     penalty .metric .estimator  mean     n std_err .config
##       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
##   1 1    e-10 npv     binary     0.723    10 0.00323 Preprocessor1_Model01
##   2 1    e-10 ppv     binary     0.730    10 0.00345 Preprocessor1_Model01
##   3 1    e-10 roc_auc binary     0.804    10 0.00254 Preprocessor1_Model01
```

```
##   4 1.80e-10 npv     binary     0.723    10 0.00323 Preprocessor1_Model02
##   5 1.80e-10 ppv     binary     0.730    10 0.00345 Preprocessor1_Model02
##   6 1.80e-10 roc_auc binary     0.804    10 0.00254 Preprocessor1_Model02
##   7 3.26e-10 npv     binary     0.723    10 0.00323 Preprocessor1_Model03
##   8 3.26e-10 ppv     binary     0.730    10 0.00345 Preprocessor1_Model03
##   9 3.26e-10 roc_auc binary     0.804    10 0.00254 Preprocessor1_Model03
##  10 5.88e-10 npv     binary     0.723    10 0.00323 Preprocessor1_Model04
## # ... with 110 more rows
```

```
lasso_res %>%
  collect_metrics() %>%
  ggplot(aes(penalty, mean, color = .metric)) +
  geom_line(size = 1.5, show.legend = FALSE) +
  facet_wrap(~.metric) +
  scale_x_log10()
```

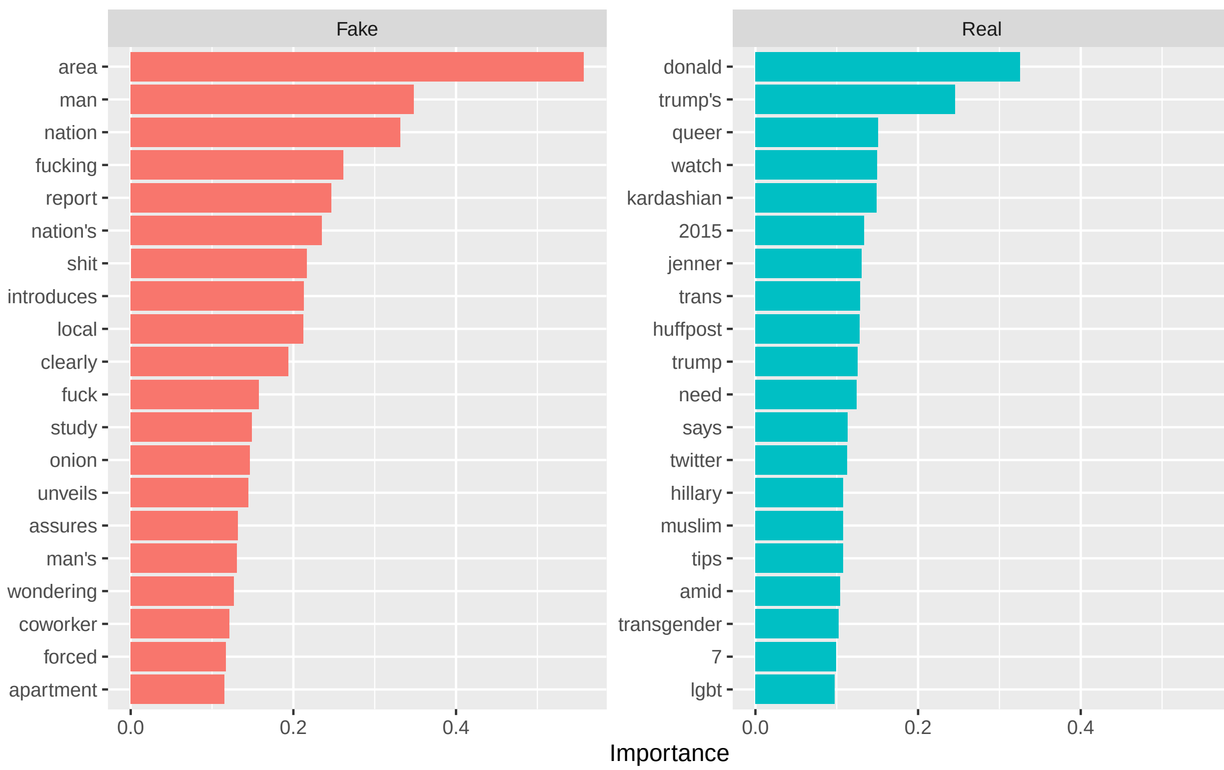



```
best_auc <- select_best(lasso_res, 'roc_auc')
final_lasso <- finalize_workflow(lasso_wf, best_auc)
```

```
train_full_fit <- final_lasso %>%
  fit(df_train)
```

```
train_full_fit %>%
  extract_fit_parsnip() %>%
  vip::vi(lambda = best_auc$penalty) %>%
  group_by(Sign) %>%
  top_n(20, wt = abs(Importance)) %>%
```

```
  ungroup() %>%
  mutate(
    Importance = abs(Importance),
    Variable = str_remove(Variable, "tfidf_headline_"),
    Variable = fct_reorder(Variable, Importance),
    Sign = if_else(Sign == 'POS', 'Real', 'Fake')
  ) %>%
  ggplot(aes(x = Importance, y = Variable, fill = Sign)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~Sign, scales = "free_y") +
  labs(y = NULL)
```



This plot shows lots of valuable information. The common phrase 'Area Man' is most associated with fake headlines. Additionally the use of swear words appear to only be associated with the Onion. This would be expected as the use of swears would almost never be allowed in a "straight news" organization. The words most associated with real stories offer hints as to were the most popular subjects to cover for the Huffington Post (Trump & lgbtq+ issues most notably). It would be interesting to re-run the analysis with news headlines from another publication i.e. NYT or WSJ to see how the results differed.

Results

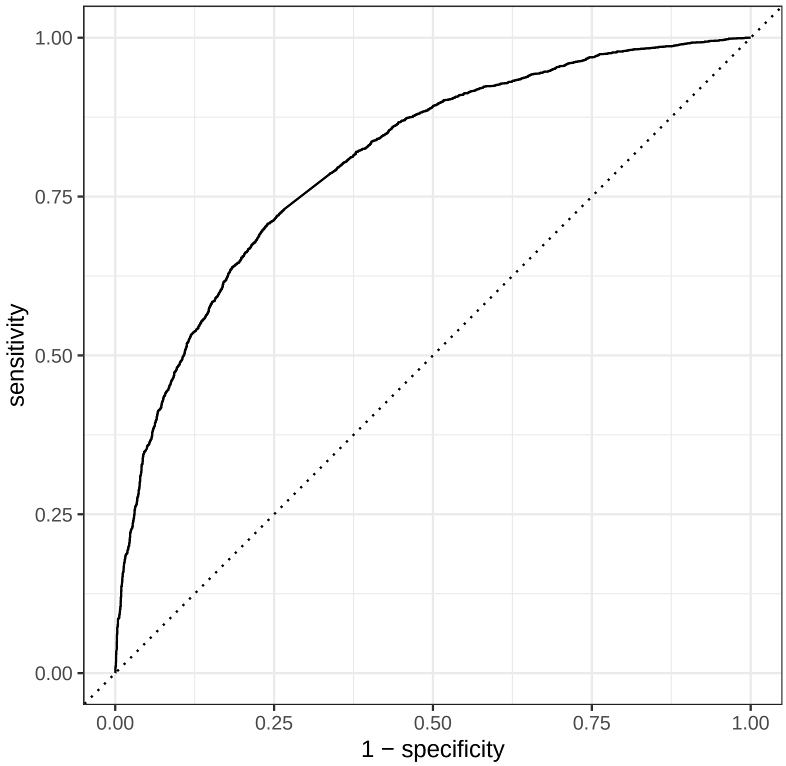
```
test_lasso <- last_fit(final_lasso, df_split)

collect_metrics(test_lasso)
```

```
## # A tibble: 2 x 4
```

```
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.732 Preprocessor1_Model1
## 2 roc_auc  binary         0.807 Preprocessor1_Model1
```

The model is able to accurately classify 73.1% of the headlines.

```
roc_res <- roc_curve(test_lasso %>% collect_predictions(), truth = is_sarcastic,`.pred_Sarcasm`)

autoplot(roc_res)
```



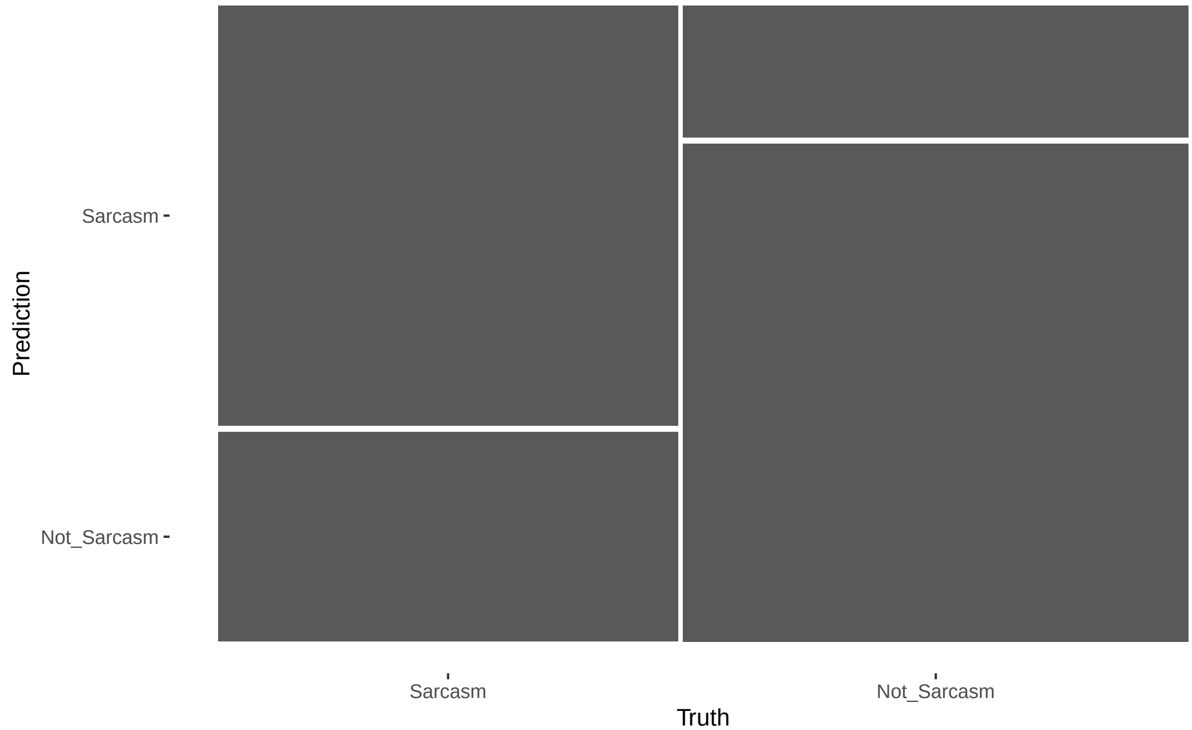
We can see the confusion matrix results below:

```
test_lasso %>%
  collect_predictions() %>%
  conf_mat(is_sarcastic, .pred_class)
```

```
##              Truth
## Prediction    Sarcasm Not_Sarcasm
##   Sarcasm        2274         784
##   Not_Sarcasm    1135        2963
```

And the same information presented visually:

```
test_lasso %>%
  collect_predictions() %>%
  conf_mat(is_sarcastic, .pred_class) %>%
  autoplot()
```




```r
z <- augment(train_full_fit, df_train) %>%
  select(-article_link)
```

We can also see a small sample of the headlines and the associated predictions:

```r
set.seed(1948)

knitr::kable(z %>%
  sample_n(10), format = 'latex', booktabs = TRUE) %>%
  kableExtra::kable_styling(latex_options = c('hold_position',
                                              'scale_down'))
```

is_sarcastic	headline	.pred_class	.pred_Sarcasm	.pred_Not_Sarcasm
Not_Sarcasm	how your morning and nighttime routines affect your health	Not_Sarcasm	0.2263203	0.7736797
Sarcasm	hospital gift shop figures it can soak 'em for 30 on the 'i'm thinking of you' teddy bear	Sarcasm	0.6485874	0.3514126
Not_Sarcasm	anna faris was dropping hints about trouble with chris pratt before split	Not_Sarcasm	0.0523475	0.9476525
Sarcasm	little butterball holding up ice cream line	Sarcasm	0.7256154	0.2743846
Not_Sarcasm	republicans are killing this regulation in order to save it	Not_Sarcasm	0.4054466	0.5945534
Not_Sarcasm	nationwide art project is making space for historic women in all 50 states	Not_Sarcasm	0.2213434	0.7786566
Sarcasm	hero dog fills out hospital paperwork	Not_Sarcasm	0.4519066	0.5480934
Sarcasm	afghanistan war veteran solemnly recalls seeing entire platoon killed by undiagnosed ptsd	Sarcasm	0.8913421	0.1086579
Not_Sarcasm	for a first-time marathoner, there's strength in numbers	Sarcasm	0.5863871	0.4136129
Sarcasm	jogging-suit shortage threatens nation's seniors	Sarcasm	0.9878893	0.0121107