Text Sentiment Analysis with R

INSY 7130 Data Mining Project

Dan O'Leary

Presented 12/3/20

Introduction

- Animal Crossing: New Horizons
  - Released March ‘20, Nintendo Switch
  - 22m units sold in less than 6 months
  - Critics 90% 🥰, users 55% 🤬
- Analyze User Reviews
  - Interpret text
  - Supervised learning
  - Binary classification – *good* or *bad*
  - Understand critic / user discrepancy?

This work is based on Julia Silge’s blog post, juliasilge.com/blog/animal-crossing/, and other references provided on slide #14.

Data, Interpretation

2,999 observations, 4 features; Scraped from MetaCritic

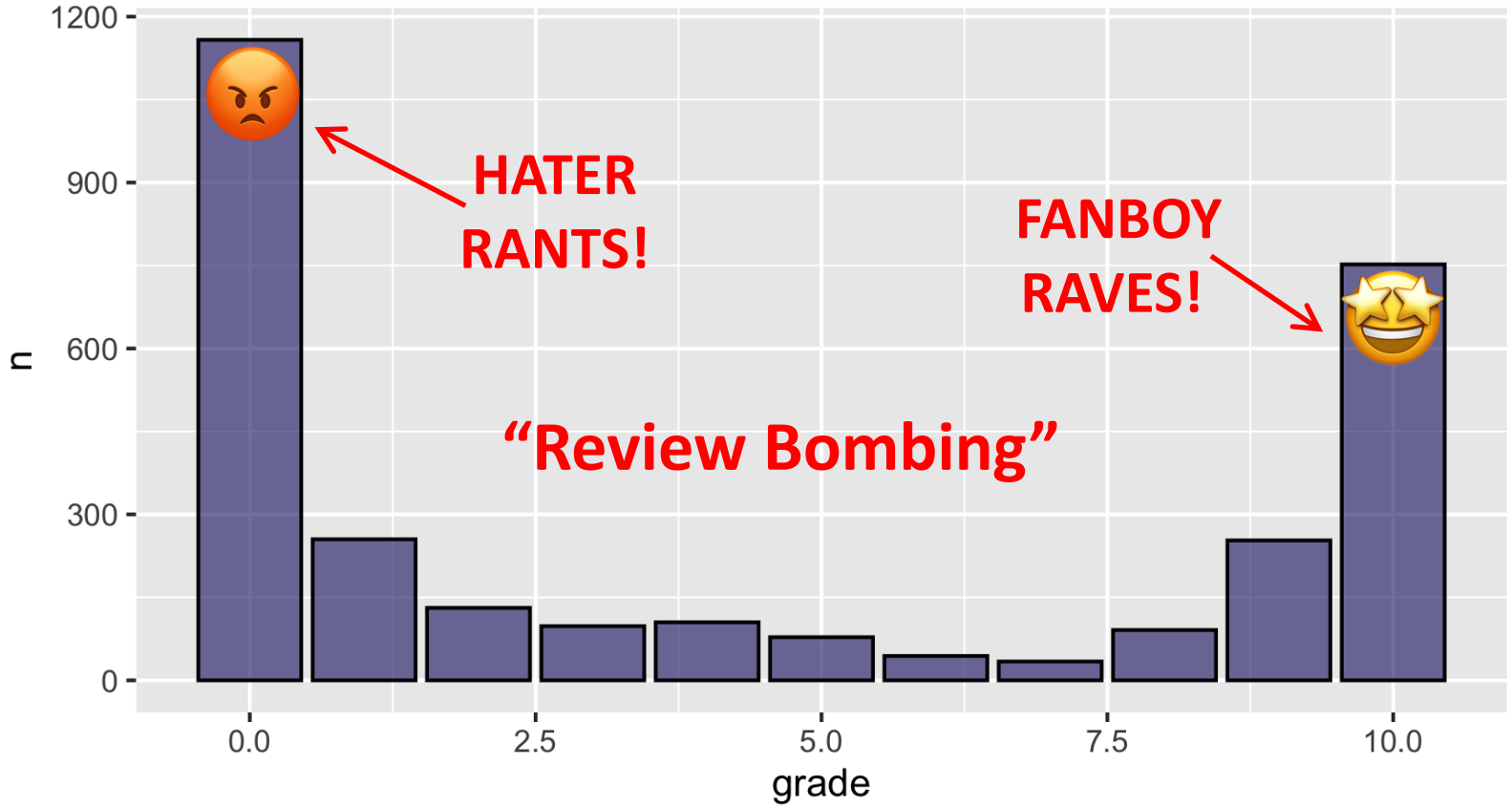
- Grade (int) – score, [0, 10]
- Text (str) – written review text
- User Name – unused
- Date – unused

```
## Rows: 2,999
## Columns: 4
## $ grade     <dbl> 4, 5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0...
## $ user_name <chr> "mds27272", "lolo2178", "Roachant", "Houndf", "Profe...
## $ text      <chr> "My gf started playing before me. No option to creat...
## $ date      <date> 2020-03-20, 2020-03-20, 2020-03-20, 2020-03-20, 202...
```

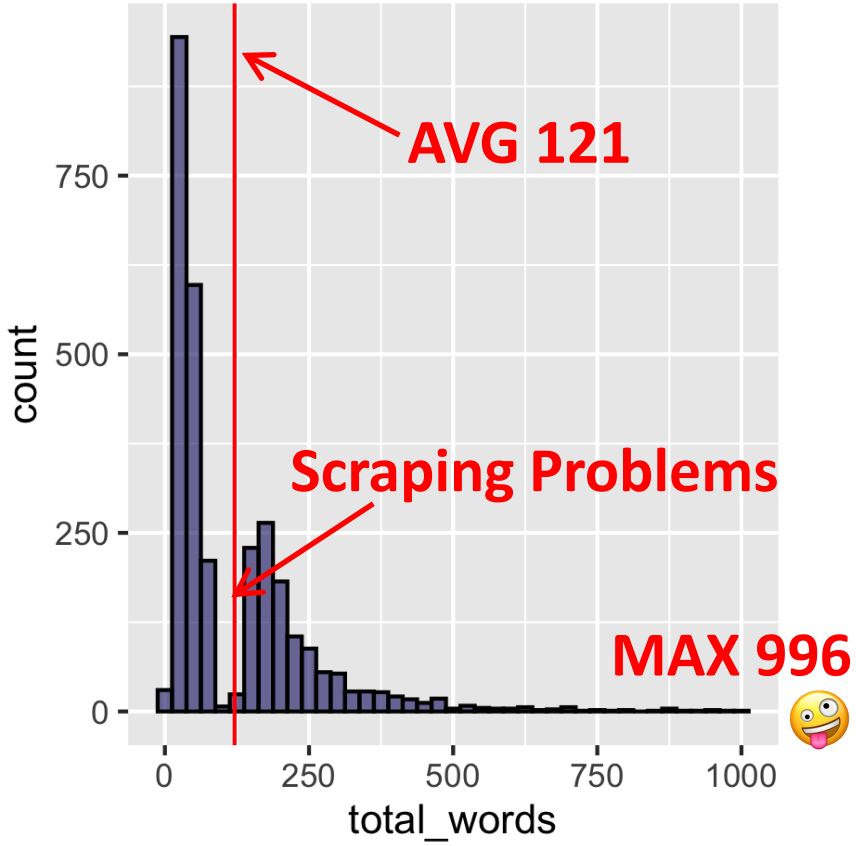
Data, Exploration

“they only allow one island per console…”
“one island per switch is insane…”
“one island per console. greedy and unfair…”

of Reviews by Grade



of Reviews by Word Count



Data, Prep

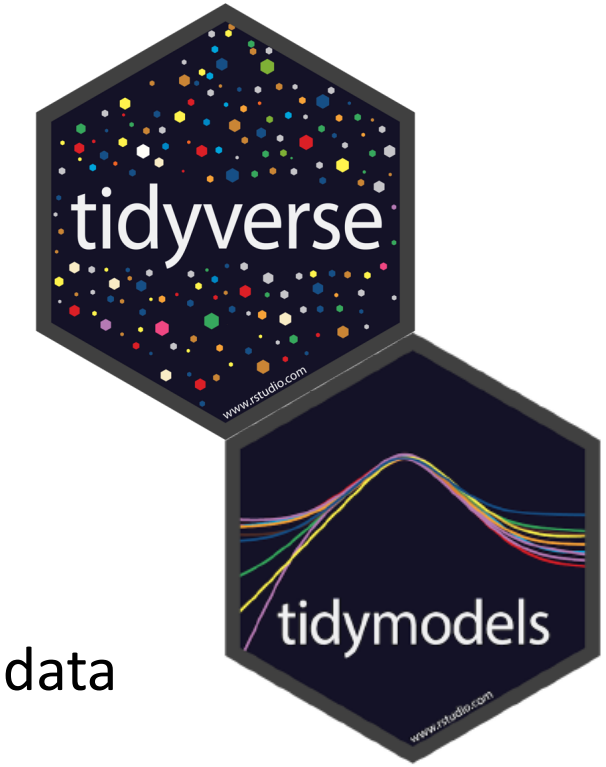
- 75/25 Train, Test Split
  - Stratified by response (rating)
- Light initial cleanup
- Tokenization process
  - 1 text str → 500 TFIDF values

rating	count	percent	source
bad	1369	60.84	train
good	881	39.16	train
bad	456	60.88	test
good	293	39.12	test

```
review_rec <- recipe(rating ~ text, data = review_train) %>%
  step_tokenize(text) %>%
  step_stopwords(text) %>%
  step_tokenfilter(text, max_tokens = 500) %>%
  step_tfidf(text) %>%
  step_normalize(all_predictors())
```

Model Building

The tidyverse way…



- Four Different Classifiers, Increasing Capacity
  - Null Model (baseline)
    - Uses majority class for all predictions
  - Naïve Bayes, SVM, Logistic Regression
    - Commonly used for text – sparse, high-dimensionality data
- For Each Model:
  - Define workflow = recipe + specification
  - Fit training data with 10-fold cross-validation, grid of tuning parameters
  - Choose optimal parameters based on CV metrics
  - Fit all training data with resulting model and validate against test data
  - Assess results

SVM Example

```
svm_spec ← svm_rbf(rbf_sigma = tune()) %>%
  set_mode("classification") %>%
  set_engine("liquidSVM")

svm_wf ← workflow() %>%
  add_recipe(review_rec) %>%
  add_model(svm_spec)

svm_grid ← grid_regular(rbf_sigma(), levels = 10)

set.seed(2020)
svm_grid ← tune_grid(
  svm_wf,
  resamples = review_folds,
  grid = svm_grid,
  metrics = metric_set(accuracy, sensitivity, specificity),
  control = control_resamples(save_pred = TRUE)
)
```

Model Spec

Workflow

Tune Grid

Fit Training

Results, Null and Naïve Bayes

Null Model

- Train Accuracy 60.8%
- Test Accuracy 60.9%
- ≈ Equal due to stratification
- Both equal to % bad
- Test Confusion Matrix:

	Truth	
Prediction	bad	good
bad	456	293
good	0	0

Naïve Bayes

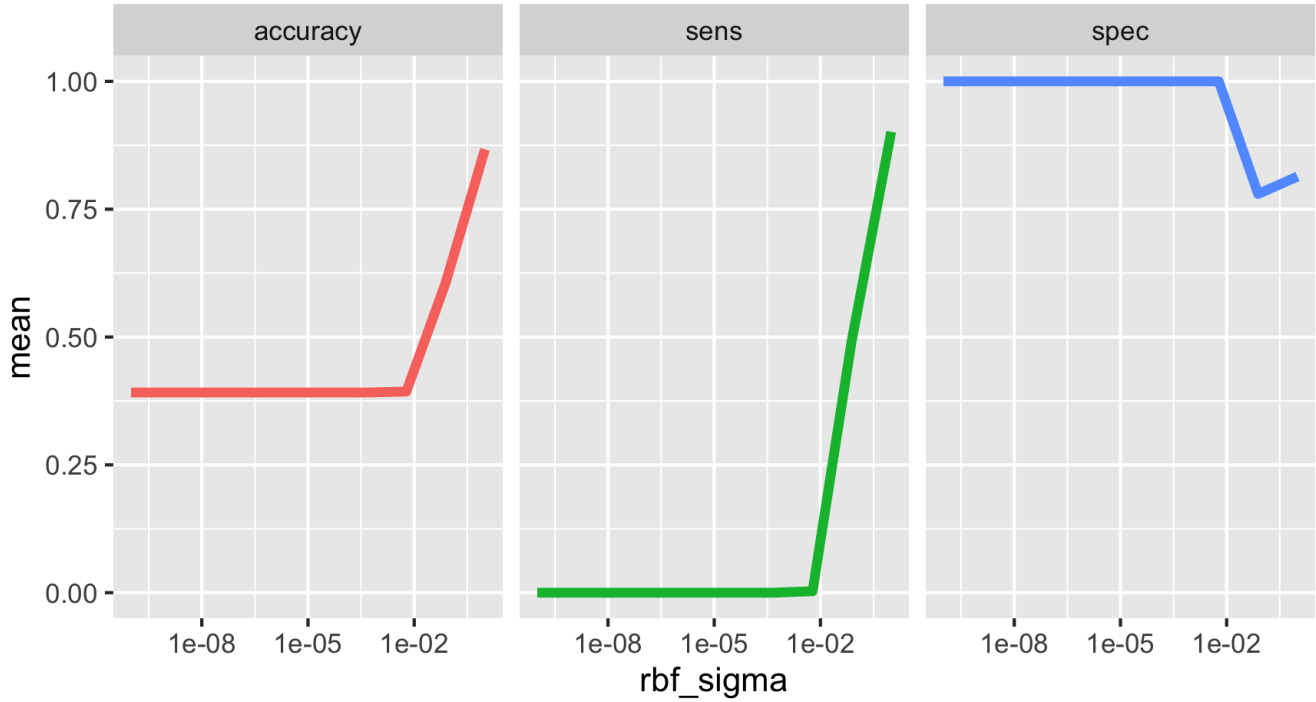
- Train Accuracy 74.0%
- Test Accuracy 74.1%
- No hyperparameter tuning
- No evidence of overfitting
- Test Confusion Matrix:

	Truth	
Prediction	bad	good
bad	418	156
good	38	137

Results, Support Vector Machine (RBF)

- Train Accuracy 86.7%
- Test Accuracy 87.4%
- 10 levels of σ (complexity)
- Assessed acc, sens, spec
- Test Confusion Matrix:

Prediction	Truth bad	Truth good
bad	409	47
good	47	246

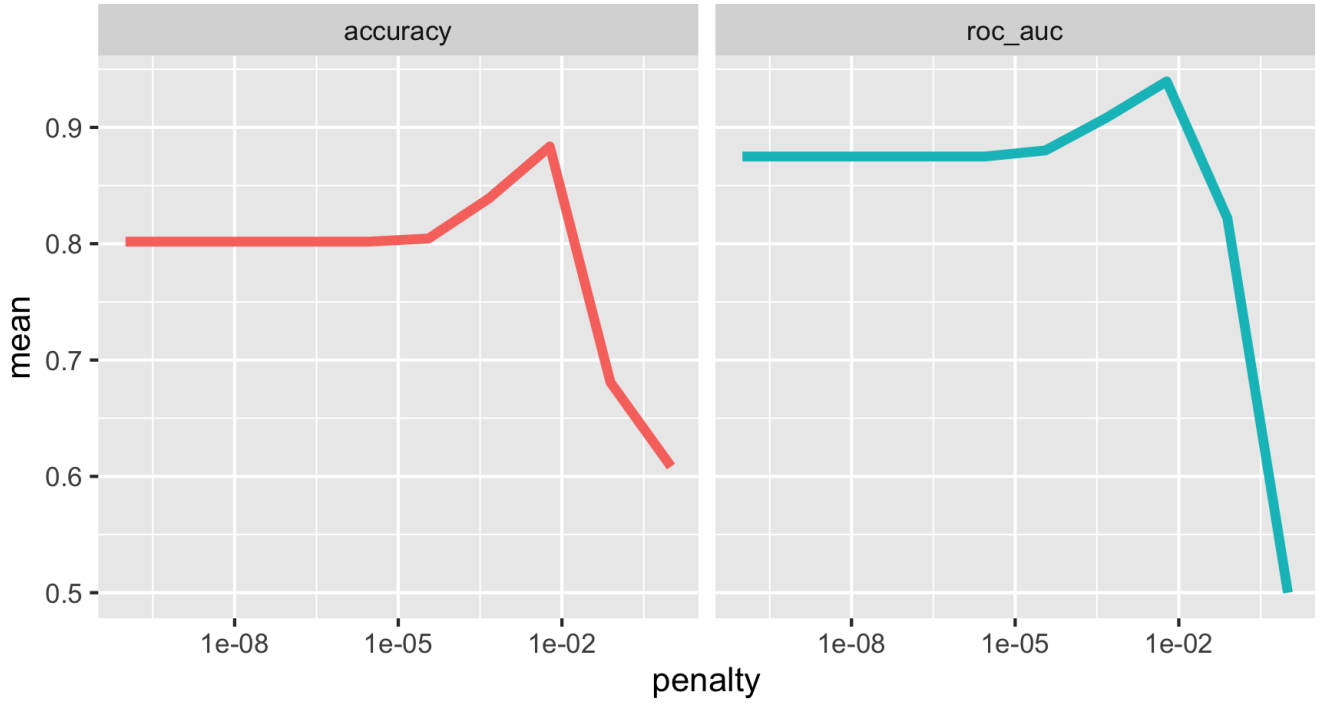


	rbf_sigma	.metric	.estimator	mean	n	std_err	.config
	<dbl>	<chr>	<chr>	<dbl>	<int>	<dbl>	<chr>
1	1	accuracy	binary	0.867	10	0.00650	Preprocessor1_Model10
2	0.0774	accuracy	binary	0.606	10	0.0136	Preprocessor1_Model09
3	0.00599	accuracy	binary	0.393	10	0.000786	Preprocessor1_Model08

Results, Logistic Regression (Lasso)

- Train Accuracy 88.4%
- Test Accuracy 87.9%
- 10 levels of *penalty* (L1)
- Assessed acc, roc_auc
- Test Confusion Matrix:

Prediction	Truth bad	good
bad	417	52
good	39	241

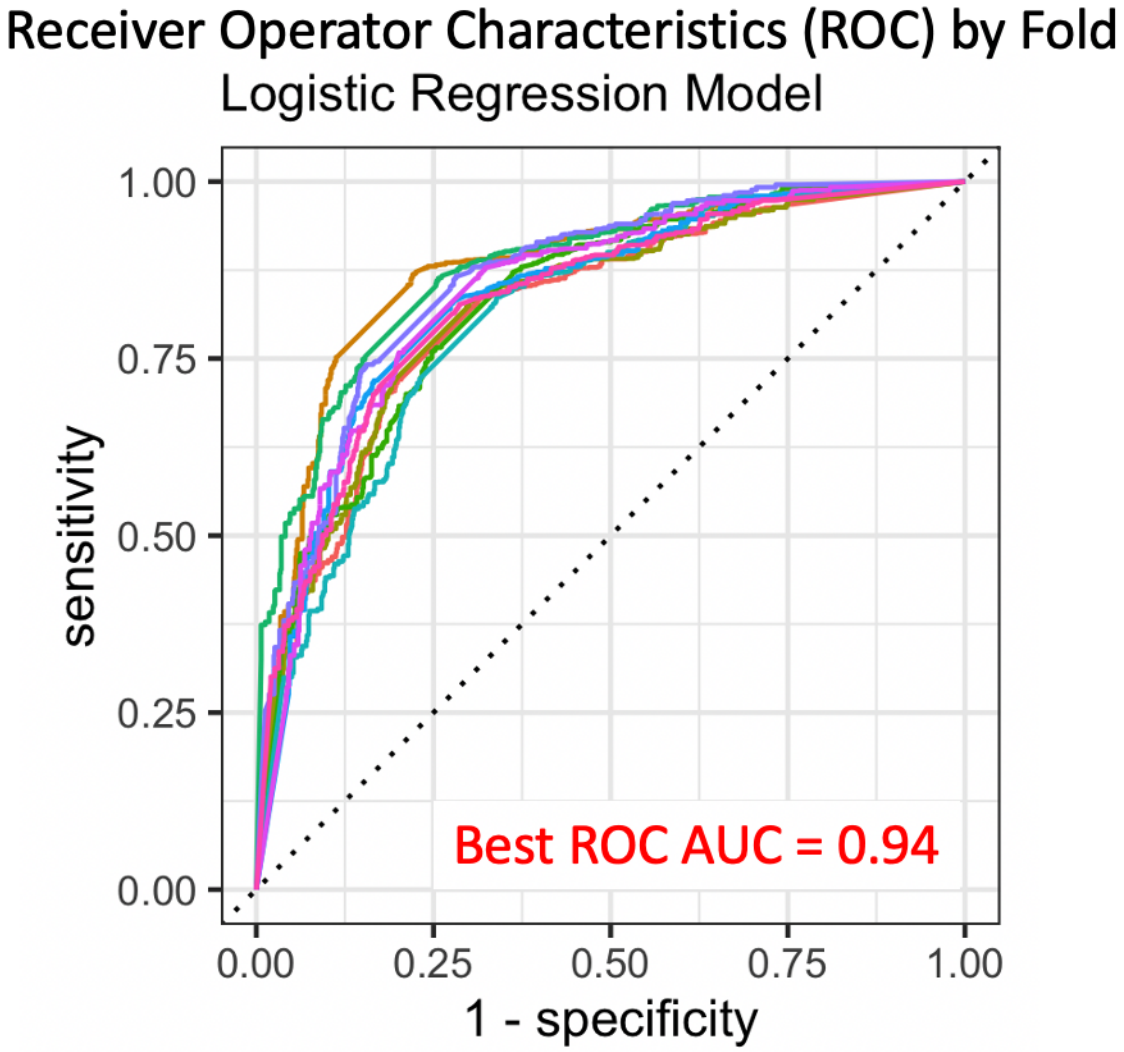


	penalty	.metric	.estimator	mean	n	std_err	.config
	<dbl>	<chr>	<chr>	<dbl>	<int>	<dbl>	<chr>
1	0.00599	accuracy	binary	0.884	10	0.00534	Preprocessor1_Model08
2	0.000464	accuracy	binary	0.839	10	0.00800	Preprocessor1_Model07
3	0.0000359	accuracy	binary	0.804	10	0.00931	Preprocessor1_Model06

Results, Summary

	Model	TrnAcc	TstAcc
1	Null	60.80	60.90
2	NB	74.00	74.10
3	SVM	86.71	87.40
4	LR	88.36	87.90

- Logistic Regression, by a hair
- “Hair” can have big impact
- Good generalization overall
- No evidence of overfitting
- Concerns about SVM results

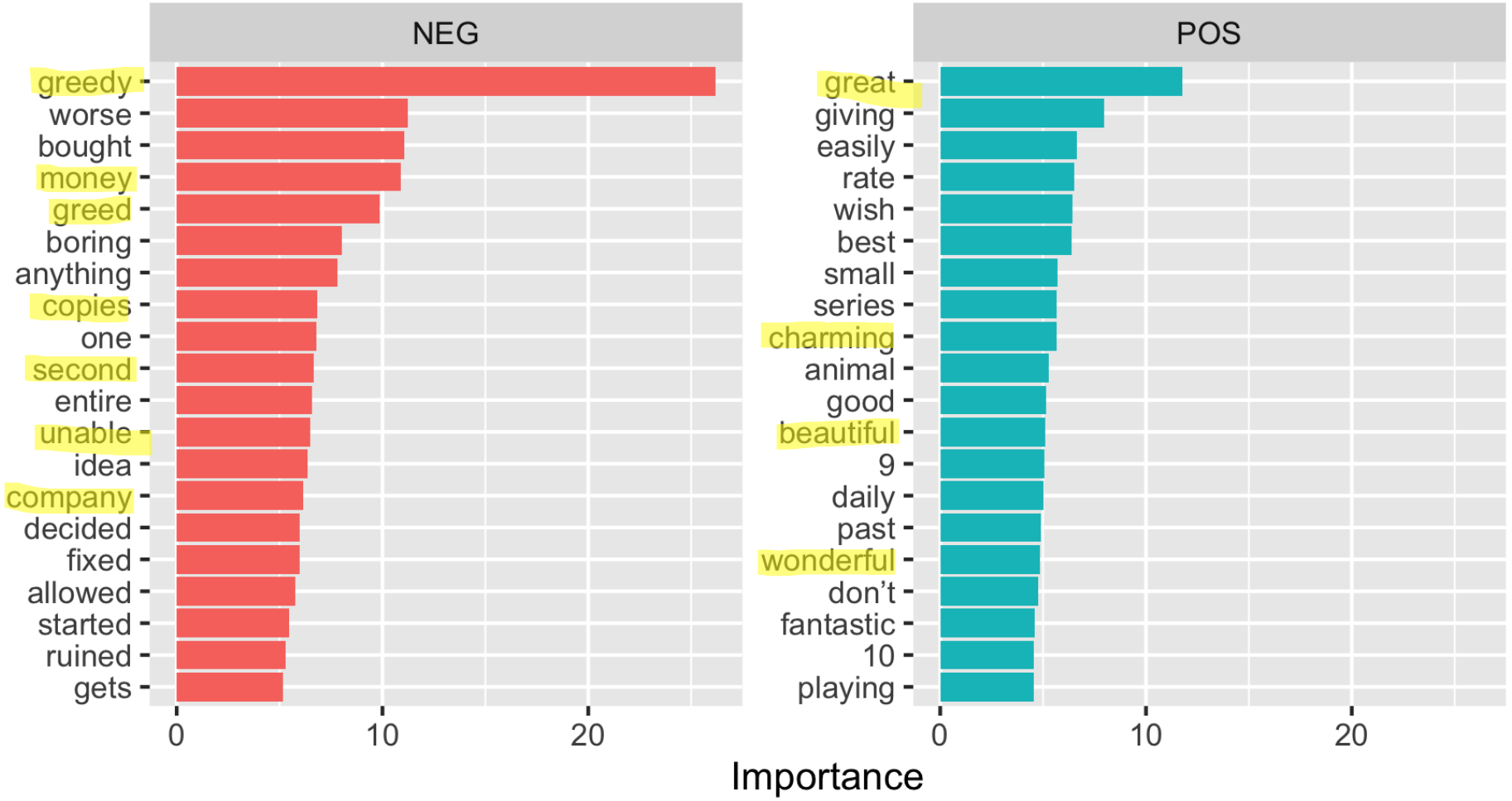


Conclusion

- Only scratched the surface of Natural Language Processing
  - First papers appeared in 2004, over 7,000 since
  - Valuable, many applications
- tidyverse / tidymodels provide welcome structure to the process
- Future work – many possible improvements, e.g.:
  - Multi-class (e.g. good, ok, bad)
  - Improve input data, eliminate scraping errors
  - Experiment with different stop word lists, tokenizing approaches (n-grams)
  - Use clustering and/or association rules to improve understanding of trends
  - Try neural net / deep neural net models, known to be effective

Bonus, Variable Importance

Credit: Julia Silge



References

Mäntylä, Mika V., et al. "The Evolution of Sentiment Analysis—A Review of Research Topics, Venues, and Top Cited Papers." *Computer Science Review*, vol. 27, Feb. 2018, pp. 16–32, doi:10.1016/j.cosrev.2017.10.002.

Silge, Julia. *Supervised Machine Learning Case Studies in R! · A Free Interactive Course.* https://supervised-ml-course.netlify.app/. Accessed 21 Oct. 2020.

---. *Supervised Machine Learning for Text Analysis in R.* https://smltar.com/. Accessed 21 Oct. 2020.

---. *Tidy Modeling with R.* https://www.tmwr.org/. Accessed 21 Oct. 2020.

Silge, Julia, and Emil Hvitfeldt. *Predictive Modeling with Text Using Tidy Data Principles.* https://emilhvitfeldt.github.io/useR2020-text-modeling-tutorial/. Accessed 2 Dec. 2020.