Rating Prediction of Yelp Reviews

Shaodong Wang, Wenjin Li, Jia Liu, Jiayin Wang
University of Wisconsin-Madison – Department of Statistics

Outline

- Data Cleaning
- Primary Analysis & Visualization
- Text Cleaning
- Feature Extraction
  - TF-IDF
  - Word2Vec
  - CountVectorizer
- Model Selection

Data Cleaning

- Longitude& Latitude

```
new_train %>% filter(longitude > 20, longitude < 120) %>%
              select(city, longitude, latitude) %>%
              group_by(longitude, latitude, city) %>%
              summarize(count = n())
```

longtitude	latitude	city	count
115.0858	36.01932	henderson	45

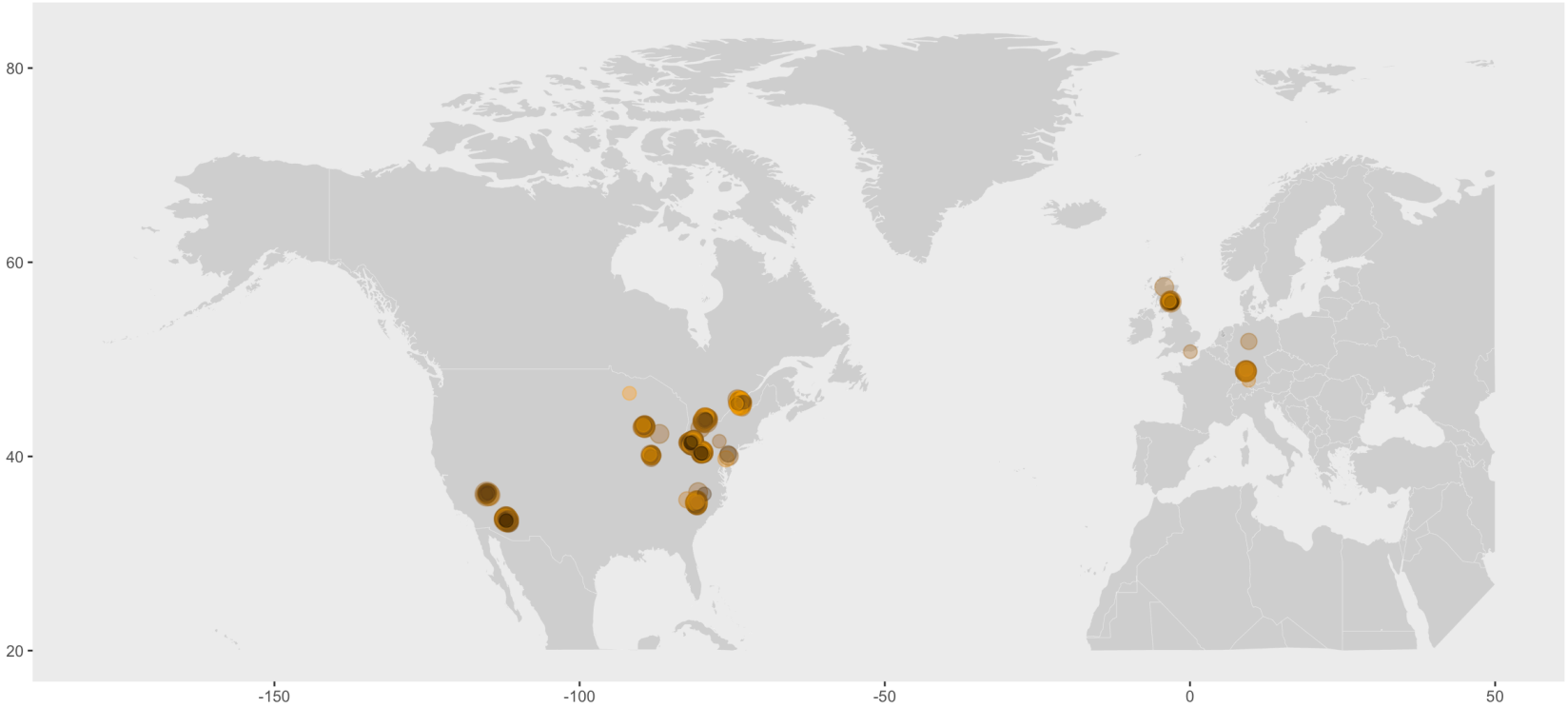
longtitude	latitude	city	count
-115.0858	36.01932	henderson	45

Data Cleaning

- Longitude& Latitude
- City (Converting text to lower case)
- Stars
- Length (Length of the review)
- Nword (# of words in the review)
- Nupper (# of emphatic words like 'NOT' and 'IT')
- Ques (# of question marks in the review)
- Excla (# of exclamation marks in the review)

Primary Analysis

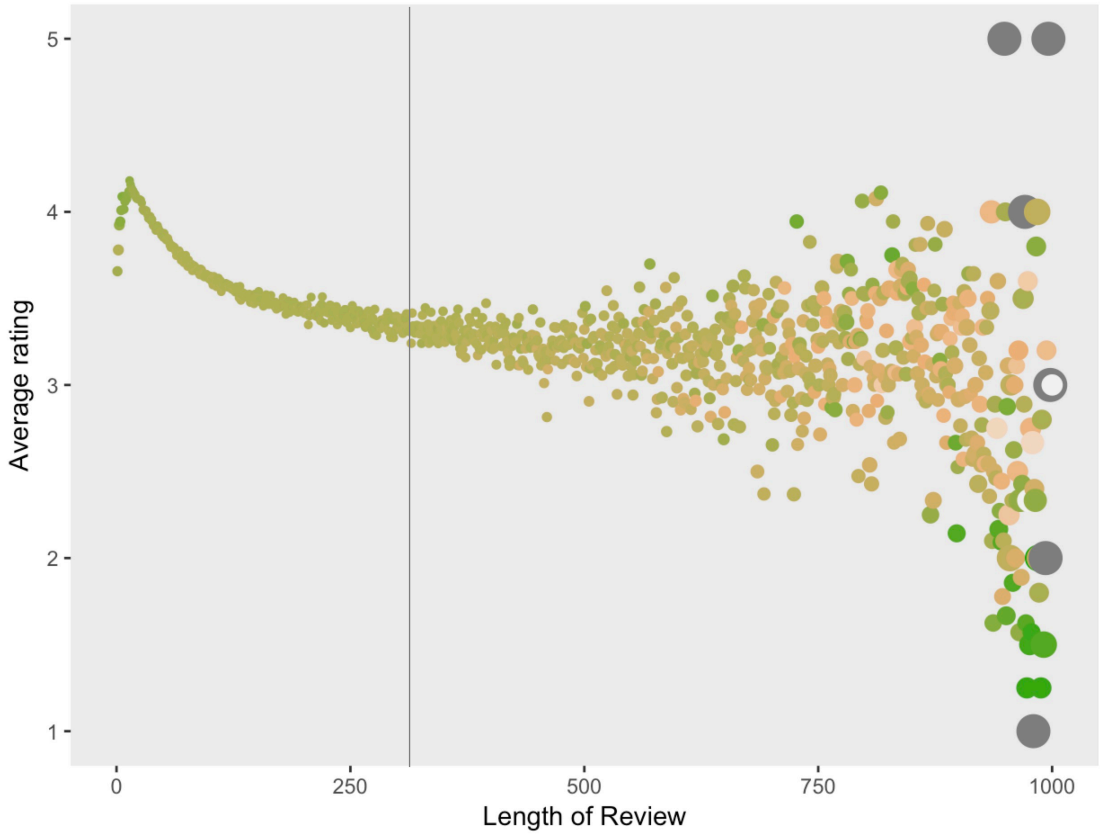
Average Yelp Rating(1-5) by Region



Color shows average stars. Size shows # of reviews.

Primary Analysis

Average Yelp Rating(1-5) by Length of Reviews

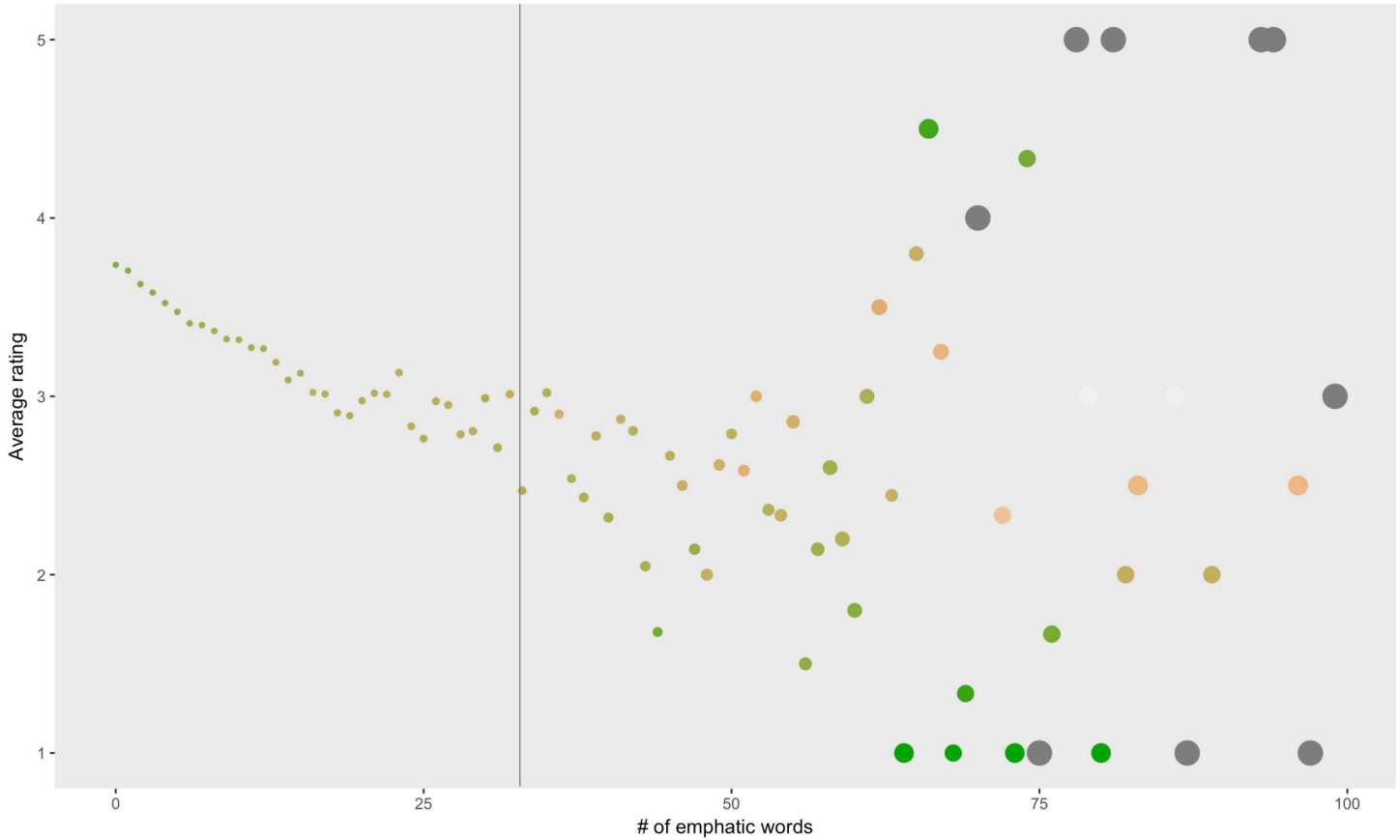


Color shows variances. Size shows (1/# of reviews).

- Decreases before 300
- No obvious trend after 300
- # of reviews decreases
- Rating variance increases

Primary Analysis

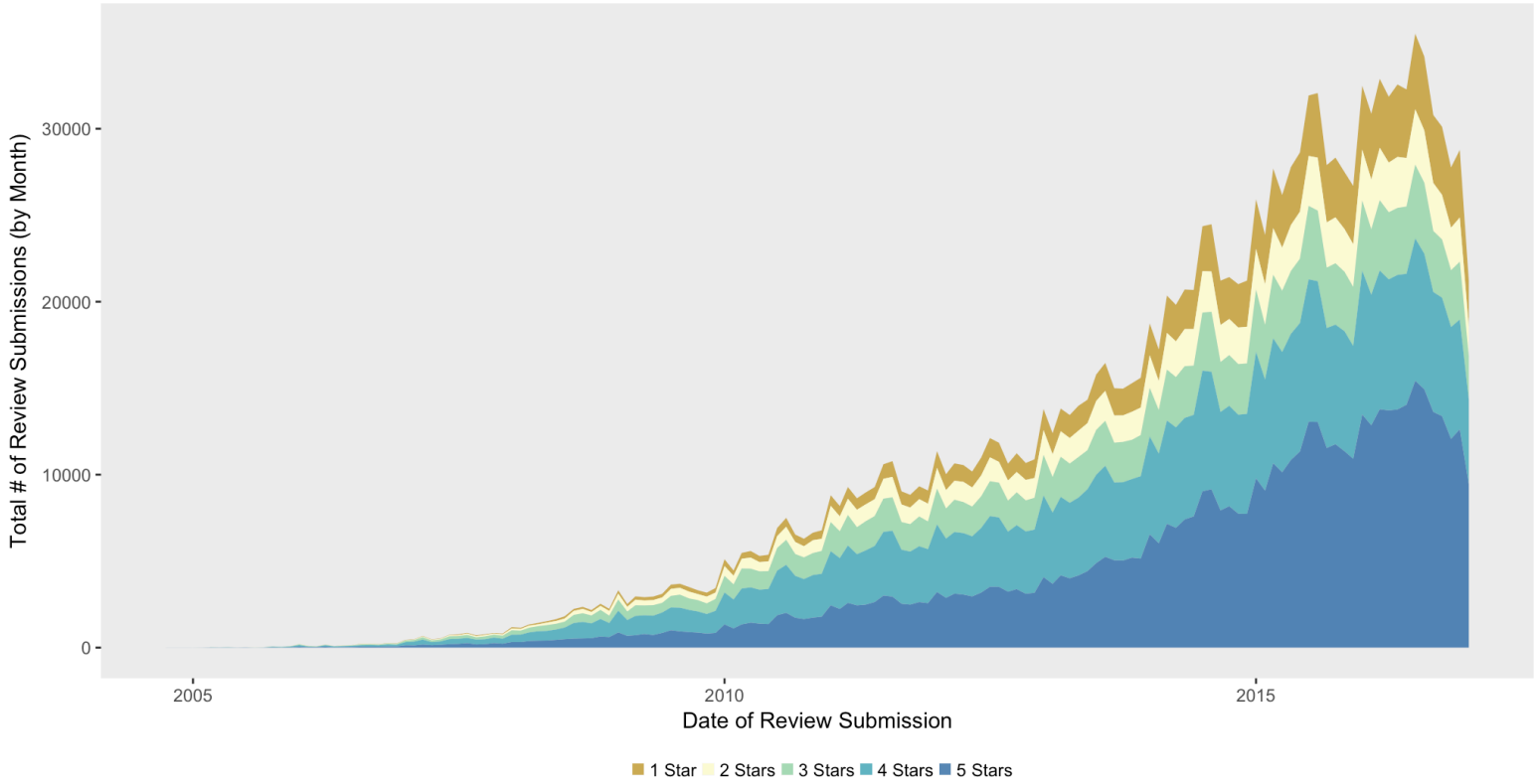
Average Yelp Rating(1-5) by # of Emphatic Words in the Reviews



Color shows variances. Size shows (1/# of reviews).

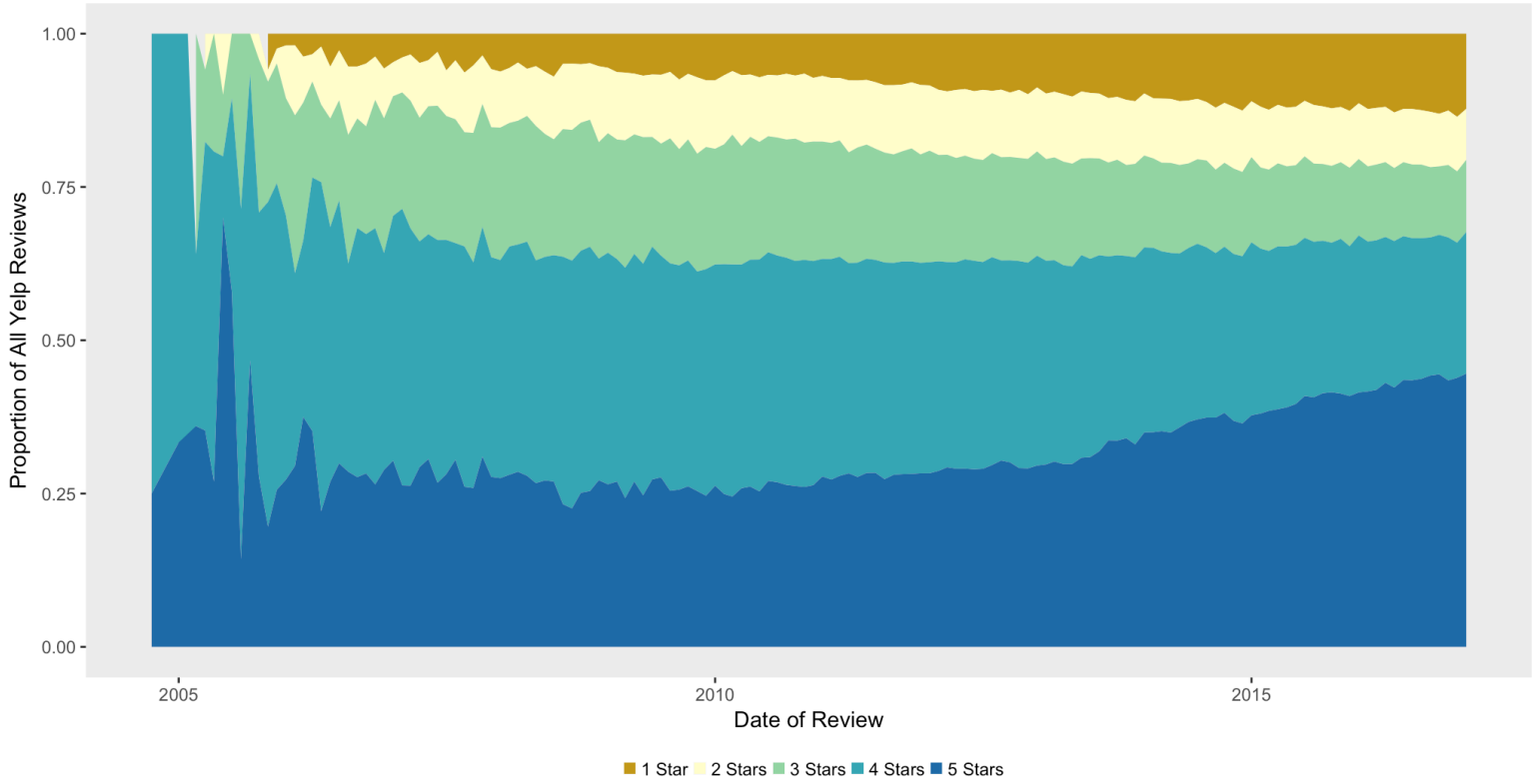
Primary Analysis

Yelp Rating Count over Time, 2005-2017



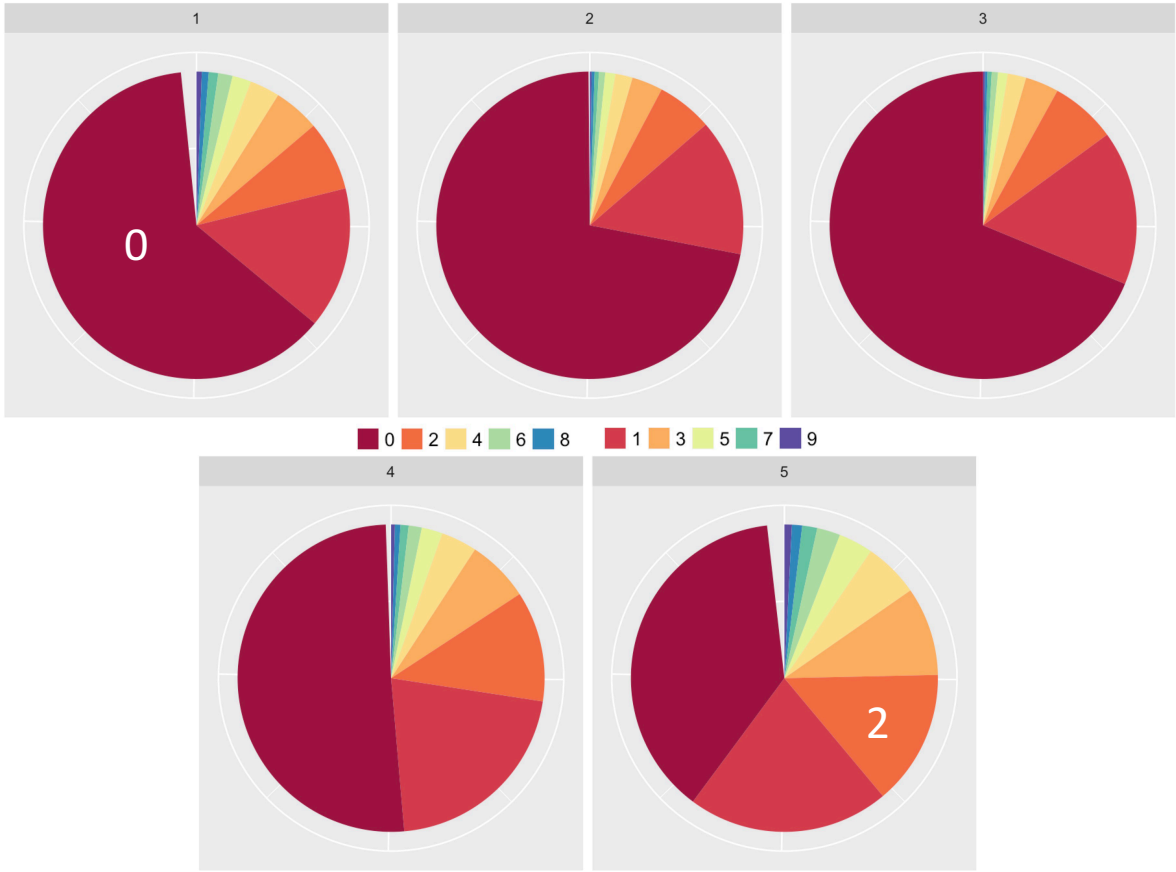
Primary Analysis

Yelp Rating Proportion over Time, 2005-2017



Primary Analysis

of Exclamation Marks Distribution by Stars, 1-5



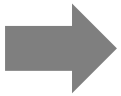
Text Cleaning

- Tokenizing text into bags of words
- Removing punctuation
- Stemming and lemmatizing
- Converting text to lower case
- Remove stop-words
- Translation

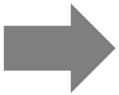
Text Cleaning

Negative postfix transformation

Wouldn't	*Shouldn't*	*Would*	*Should*
Couldn't	*Needn't*	*Could*	*Need*
Shan't	*Haven't*	*Shan*	*Have*
Aren't.	*Mustn't*	*Are*	*Must*



*"They **wouldn't** be very interesting, I'm afraid."*

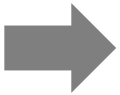


*"They **would not** be very interesting, I'm afraid."*

Text Cleaning

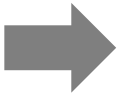
Tokenizing text into bags of words

"They would not be very interesting, I'm afraid."



['They', 'would', ”not", 'be', 'very', 'interesting', ',', 'I', "'m", 'afraid', '.']

*"The point of these examples is to _learn how basic text cleaning works_ on *very simple* data."*

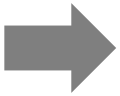


*['The', 'point', 'of', 'these', 'examples', 'is', 'to', '_learn', 'how', 'basic', 'text', 'cleaning', 'works_', 'on', '*very', 'simple*', 'data', '.']*

Text Cleaning

Removing punctuation

['The', 'point', 'of', 'these', 'examples', 'is', 'to', ***'_learn',*** *'how', 'basic', 'text', 'cleaning', 'works_', 'on****', '*very', 'simple*',*** *'data', '.']*



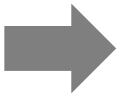
['The', 'point', 'of', 'these', 'examples', 'is', 'to', ***'learn',*** *'how', 'basic', 'text', 'cleaning', 'works', 'on****', 'very', 'simple',*** *'data']*

Text Cleaning

Cleaning text of stopwords

- Manually keep some of stopwords
- *‘not’, ‘he’, ‘she’, ‘very’, ‘again’, ‘most’, etc*.

*['The', 'point', 'of', 'these', 'examples', **'is', 'to',** 'learn', 'how', 'basic', 'text', 'cleaning', 'works', 'on', **'very',** 'simple', 'data']*

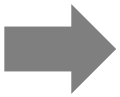


*['The', 'point', 'examples', 'learn', 'basic', 'text', 'cleaning', 'works', **'very',** ’simple', 'data']*

Text Cleaning

Stemming and lemmatizing

['The', 'point', 'examples',
'learn', 'basic', 'text',
'cleaning', 'works',
*'very', **'simple',** 'data']*

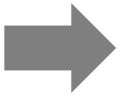


['The', 'point', 'exampl',
'learn', 'basic', 'text',
'clean', 'work', 'veri',
***'simpl',** 'data']*

Text Cleaning

Converting text to lower case

*[**'The',** 'point', 'examples',*
'learn', 'basic', 'text',
'cleaning', 'works',
'very', 'simple', 'data']

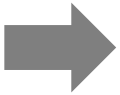


*[**'the',** 'point', 'exampl',*
'learn', 'basic', 'text',
'clean', 'work', 'veri',
'simpl', 'data']

Text Cleaning

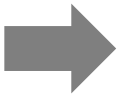
Translation

text =
‘これは素晴らしいですし、私はそれが大好き’



proc_text(text) = NA

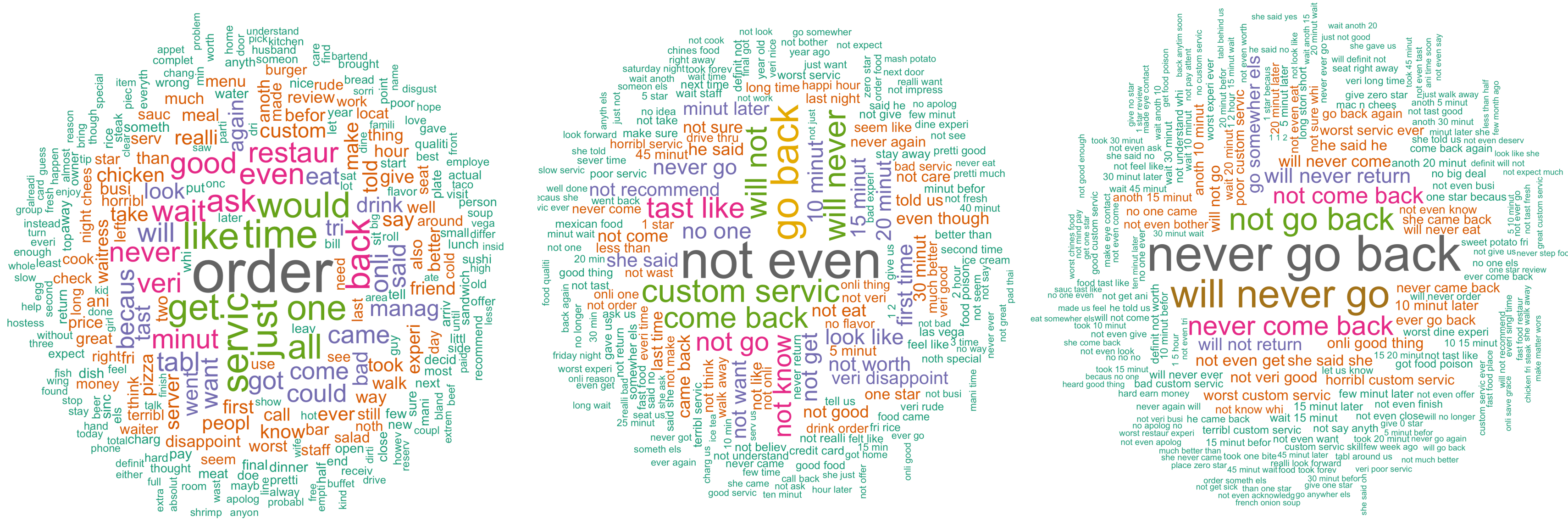
trans(text) =
’ This is wonderful and I love it’



proc_text(trans(text)) =
[‘wonderful’, ‘love’]

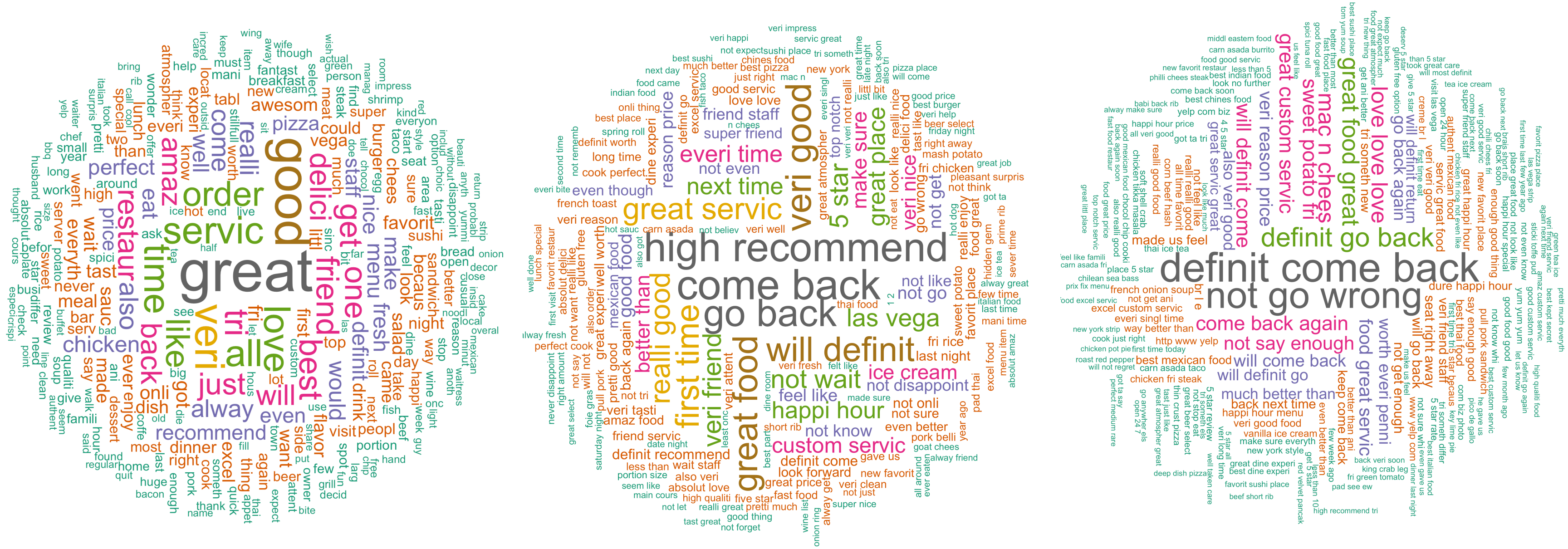
Feature Extraction

Unigram, Bigram and Trigram Word Clouds for Rating 1.



Feature Extraction

Unigram, Bigram and Trigram Word Clouds for Rating 5.



Feature Extraction

- TF-IDF

absolut disgust	always terribl	amaz	disappoint	anyth nice	busi	best lobster
0	0.0656	0	0	0.0656	0	0
0	0	0.1563	0	0	0	0
0	0	0	0	0	0.0423	0.0525
0.0353	0	0	0.3533	0	0.0285	0
0	0	0	0	0	0	0

- CountVectorizer
- Word2Vec ✘
- GloVe ✘

Feature Extraction

- TF-IDF
- CountVectorizer

absolut disgust	always terribl	amaz	disappoint	anyth nice	busi	best lobster
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	0	0	0	0	1	1
1	0	0	1	0	1	0
0	0	0	0	0	0	0

- Word2Vec ✘
- GloVe ✘

Model Selection

- Naïve Bayes
- SVM
- Random Forest
- k-Nearest Neighbors
- Long-Short Term Memory ✘
- Ridge ✘
- Lasso ✘
- Logistic ✘
- …

Model Selection

MSE of Different Regression Model with TF-IDF

Size of Train Set	SVM	Random Forest	k-NN
100	2.05	1.39	1.65
1,000	1.65	1.35	1.56
10,000	0.83	0.99	1.55
100,000	0.64	0.79	1.49
…	↓	↓	↓

Questions?