IDS 572 Assignment 3

Analyzing text in Yelp reviews - Text mining, Sentiment analyses

Submitted by –

Dhananjay Singh (668437546)

Srinanda Kurapati (663244158)

Sunny Patel (676645654)

Introduction -

Text is typically considered to be unstructured data. It is a natural language in which one cannot quite use the typical techniques of machine learning. In this assignment, we are going to understand/predict opinions of customers based on their reviews on Yelp. We will be using Sentiment Analysis to classify reviews as positive or negative.

Sentiment Analysis is the process of analysing a piece of text to understand the sentiment behind it, by giving scores/categorizing components of a sentence. This is used in large scale enterprises/markets to know what the public thinks of their service or product. Reading tens of thousands of reviews is hard, hence we need to work on building a model that does this for us.

This works on primarily decomposing a sentence into words, phrases and its parts of speech and assigning a score to each part and then getting an aggregate score. Unfortunately, not every sentence can be discreetly identified as positive or negative. There are a lot of grey areas here and the primary reason is context. Hence, we have to be careful we also preserve the context of the sentences. We are going to use 3 libraries - NRC, AFINN and Bing

(a) Explore the data.
(i) How are star ratings distributed? How will you use the star ratings to obtain a label indicating 'positive' or 'negative' – explain using the data, graphs, etc.?
Do star ratings have any relation to 'funny', 'cool', 'useful'? Is this what you expected?

Number of reviews by star-ratings

A tibble: 5 × 2 Groups: starsReview [5]

starsReview <dbl>	n <int>
1	4553
2	4094
3	5561
4	10795
5	15084

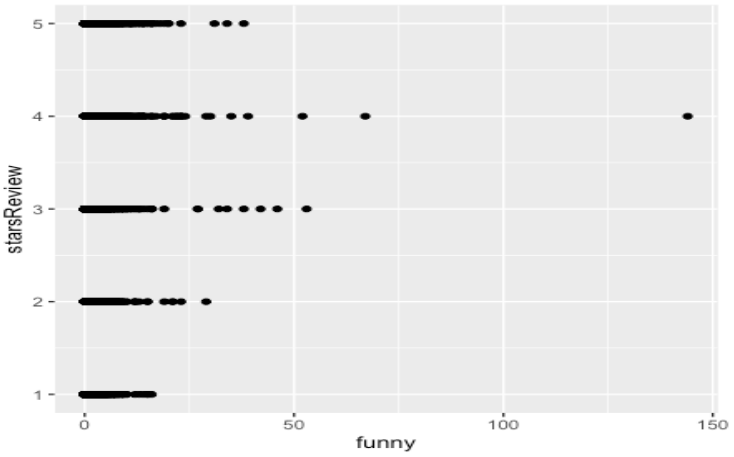
5 rows

Star-ratings are unequally distributed with star-ratings 4, and 5 having the greatest number of reviews. While star-ratings 1, 2, and 3 have fewer reviews as compared to star-ratings 4, and 5. We will label the reviews having star-ratings 4, and 5 as "Positive", and the reviews having star-ratings 1, and 2 as "Negative". We have discarded reviews with star-rating 3, considering it a neutral review.

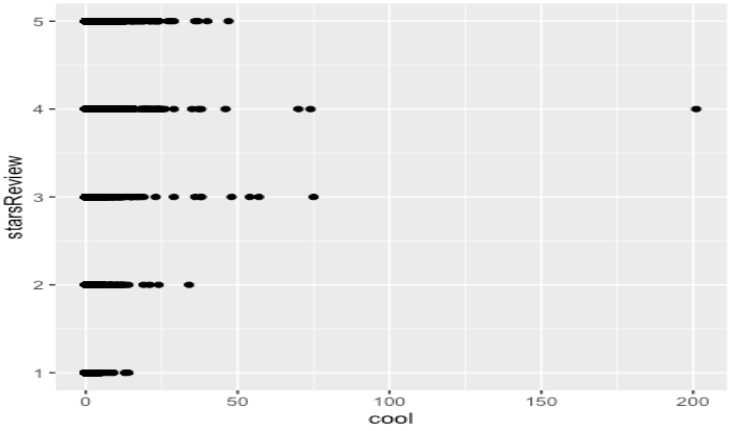
We have plotted specific words Vs. star-ratings to check for any relation. The words we considered are funny, useful, and cool. Word "funny" is mostly present in star-rating 4, and least frequency in star-rating 1. Word "cool" is has higher frequency in star-ratings 3, 4, and 5, and has very low count for star-ratings 1, and 2. Word "useful" has a very similar distribution for all the star-ratings.

Plot of star-rating Vs. specific words

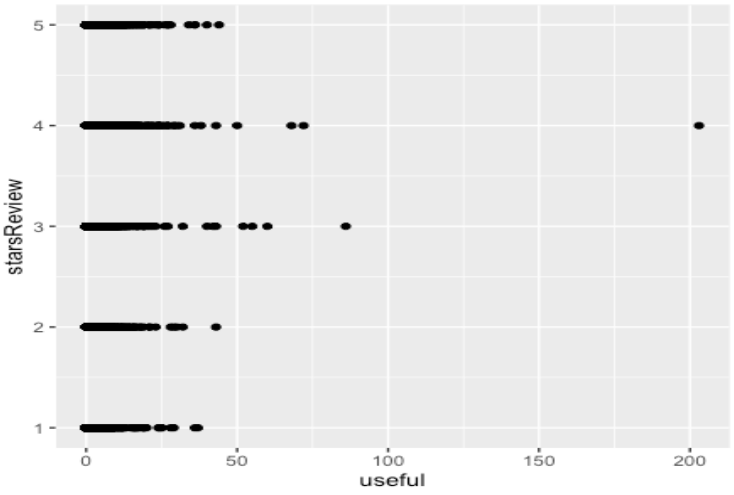
- Star-rating Vs. funny



- Star-rating Vs. cool



- Star-rating Vs. useful



Count of reviews in each State –

	state	n
1	AZ	16670
2	IL	339
3	NC	3483
4	NV	11560
5	OH	3390
6	PA	3080
7	SC	190
8	WI	1375

Total occurrences of different words, & sort by most frequent –

word <chr>	n <int>
food	32113
service	15844
time	12521
chicken	9411
restaurant	8833
nice	7907
menu	7574
love	7145
delicious	7090
bar	6202

Least frequent terms are as follows –

	word	n
1	13.00	10
2	14.00	10
3	15.99	10
4	2007	10
5	3.25	10
6	32oz	10
7	4.95	10
8	8am	10
9	98	10

Most frequent terms are as follows –

	word	n
1	food	32113
2	service	15844
3	time	12521
4	chicken	9411
5	restaurant	8833
6	nice	7907
7	menu	7574
8	love	7145
9	delicious	7090
10	bar	6202

(a)

(ii) How does star ratings for reviews relate to the star-rating given in the dataset for businesse (attribute 'businessStars')? (Can one be calculated from the other?)

We have selected and viewed the "starsReview" and "starsBusiness" and filtered based on different "business_id" to check for any relation between the two.

Example – starsReview and starsBusiness for restaurant with business_id - 4uiijOUDzc-DeIb2XcKW_A

	starsReview	business_id	starsBusiness	n
	All	4uiijOUDzc-DeIb2Xc	All	All
1	1	4uiijOUDzc-DeIb2XcKW_A	2.5	5
2	2	4uiijOUDzc-DeIb2XcKW_A	2.5	15
3	3	4uiijOUDzc-DeIb2XcKW_A	2.5	7
4	4	4uiijOUDzc-DeIb2XcKW_A	2.5	8
5	5	4uiijOUDzc-DeIb2XcKW_A	2.5	2

Example – starsReview and starsBusiness for restaurant with business_id - __zA29wBG0LleSxMzNHpwQ

	starsReview	business_id	starsBusiness	n
	All	\29wBG0LleSxMzNHpw	All	All
1	1	__zA29wBG0LleSxMzNHpwQ	4	6
2	2	__zA29wBG0LleSxMzNHpwQ	4	5
3	3	__zA29wBG0LleSxMzNHpwQ	4	7
4	4	__zA29wBG0LleSxMzNHpwQ	4	21
5	5	__zA29wBG0LleSxMzNHpwQ	4	38

Example – starsReview and starsBusiness for restaurant with business_id - h4U3h1RbgLvHl0fKSSUhPA

	starsReview	business_id	starsBusiness	n
	All	h4U3h1RbgLvHl0fKS	All	All
1	1	h4U3h1RbgLvHl0fKSSUhPA	1.5	30
2	2	h4U3h1RbgLvHl0fKSSUhPA	1.5	3
3	3	h4U3h1RbgLvHl0fKSSUhPA	1.5	1
4	4	h4U3h1RbgLvHl0fKSSUhPA	1.5	2
5	5	h4U3h1RbgLvHl0fKSSUhPA	1.5	2

Example – starsReview and starsBusiness for restaurant with business_id - 16d3BlncEyCTzb0GxXrBXQ

Filter

	starsReview	business_id	starsBusiness	n
	All	l3BlncEyCTzb0GxXrB	All	All
1	1	16d3BlncEyCTzb0GxXrBXQ	5	2
2	3	16d3BlncEyCTzb0GxXrBXQ	5	2
3	4	16d3BlncEyCTzb0GxXrBXQ	5	7
4	5	16d3BlncEyCTzb0GxXrBXQ	5	144

Average star rating Vs. Business star rating

starsBusiness <dbl>	avstarsReview <dbl>
1.5	1.490909
2.0	2.054321
2.5	2.607696
3.0	3.033333
3.5	3.511762
4.0	3.981303
4.5	4.460390
5.0	4.891386

8 rows

Overall, if we were to take a summary of all the business stars and their aggregate ratings, we can see that the business star of a restaurant is the mean of the star ratings given by its customers.

(b) What are some words indicative of positive and negative sentiment? (One approach is to determine the average star rating for a word based on star ratings of documents where the word occurs). Do these 'positive' and 'negative' words make sense in the context of user reviews being considered? (For this, since we'd like to get a general sense of positive/negative terms, you may like to consider a pruned set of terms -- say, those which occur in a certain minimum and maximum number of documents).

Words by star rating of reviews –

	starsReview	word	n
1	5	food	10827
2	4	food	7861
3	5	service	5369
4	3	food	4952
5	1	food	4339
6	2	food	4134
7	4	service	3955
8	5	delicious	3897
9	5	love	3848
10	5	time	3823

	starsReview	word	n
1	1	aaron	1
2	1	abdul	1
3	1	abound	1
4	1	abundant	1
5	1	accommodations	1
6	1	accomplish	1
7	1	accustomed	1
8	1	ace	1
9	1	achieve	1
10	1	active	1

To indicate words as having positive or negative sentiment, we have found out the star rating associated with each word along with the frequency of occurrence, and proportion of that word for all star ratings. Word "love" has positive connotation, and this is proven as its count is the highest in star ratings 4, and 5. Word "bad" has the highest count in 1 star rating reviews, and least in 5 star rating reviews.

Proportion of 'love' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	love	3848	0.008600363
4	love	2008	0.004814678
3	love	642	0.002701577
2	love	380	0.002199583
1	love	267	0.001479839

Proportion of 'bad' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
1	bad	948	0.005254261
3	bad	890	0.003745177
2	bad	773	0.004474415
4	bad	681	0.001632866
5	bad	536	0.001197971

Some words like “service”, “food”, “time”, “restaurant”, and “chicken” have a high count for high star rating reviews. However, these words do not indicate a positive sentiment and are very much neutral words. We found out that these words have the highest count in our tokenized set, which helped us use a threshold value of 8833 to remove these common words.

Proportion of 'service' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	service	5369	0.011999830
4	service	3955	0.009483093
3	service	2411	0.010145641
1	service	2131	0.011811002
2	service	1978	0.011449410

Proportion of 'food' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	food	10827	0.02419858
4	food	7861	0.01884870
3	food	4952	0.02083833
1	food	4339	0.02404877
2	food	4134	0.02392915

Proportion of 'time' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	time	3823	0.008544487
4	time	3333	0.007991694
3	time	1944	0.008180475
1	time	1843	0.010214771
2	time	1578	0.009134059

Proportion of 'restaurant' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	restaurant	2922	0.006530733
4	restaurant	2210	0.005299023
3	restaurant	1304	0.005487315
1	restaurant	1279	0.007088818
2	restaurant	1118	0.006471405

Proportion of 'chicken' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

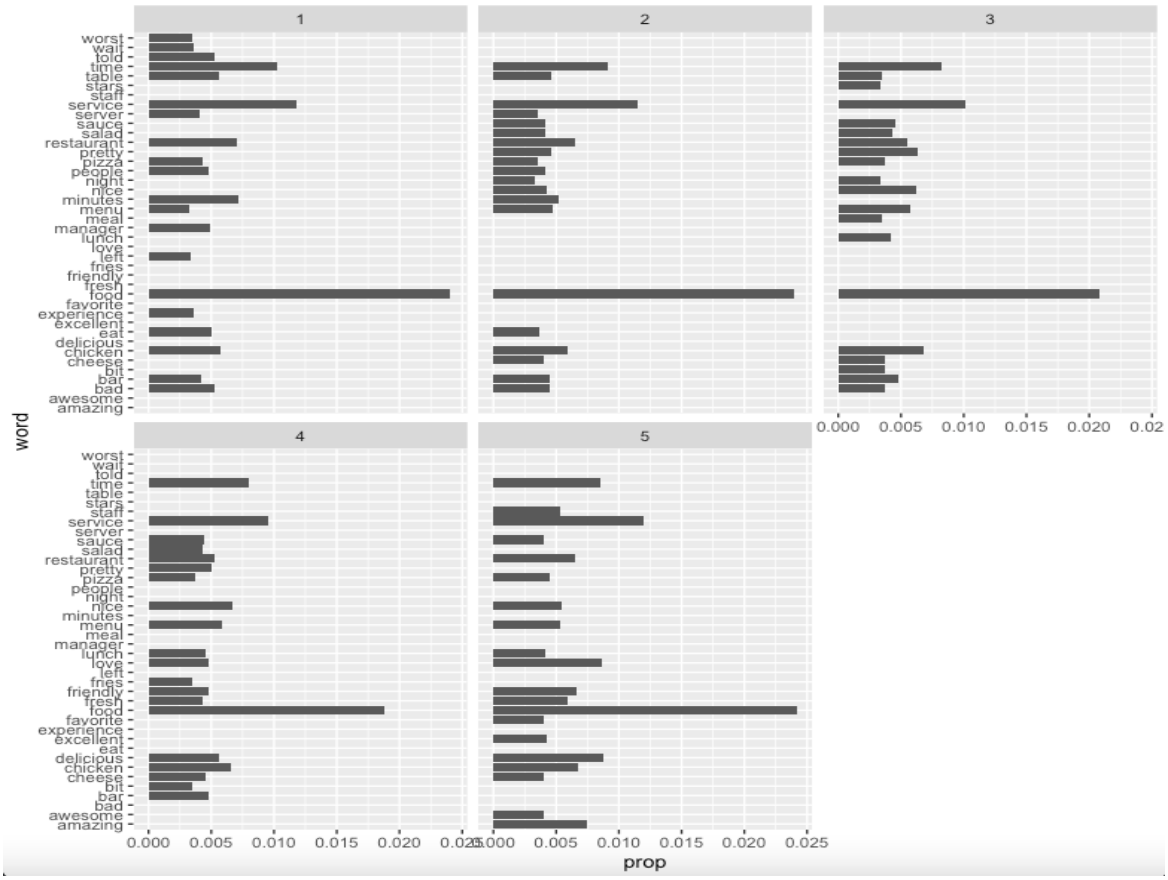
starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	chicken	3020	0.006749765
4	chicken	2729	0.006543454
3	chicken	1608	0.006766566
1	chicken	1030	0.005708743
2	chicken	1024	0.005927298

#check the proportion of 'manager' among reviews with 1,2,..5 stars

A tibble: 5 × 4 Groups: starsReview [5]

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
1	manager	885	0.0049050852
2	manager	330	0.0019101644
5	manager	311	0.0006950917
4	manager	188	0.0004507766
3	manager	184	0.0007742837

Plot of top 20 words by star ratings



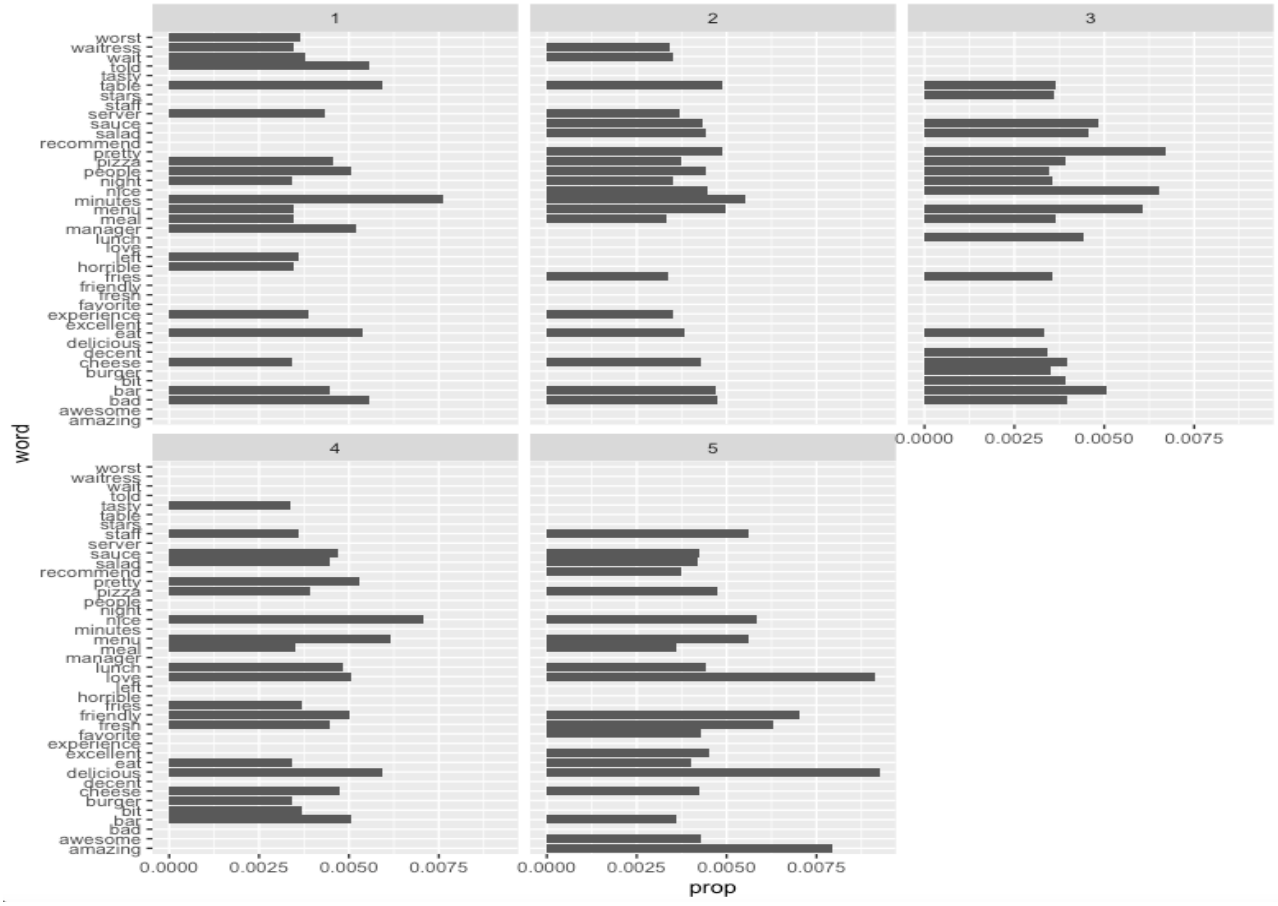
Top 20 words associated with highest star rating

	word	totWS
1	bar	0.06381670
2	cheese	0.06053668
3	chicken	0.09778568
4	delicious	0.07588496
5	eat	0.05381260
6	food	0.33080974
7	fresh	0.05969134
8	friendly	0.06604759
9	love	0.07624426
10	lunch	0.05973625
11	menu	0.07974868
12	nice	0.08387143
13	pizza	0.05984146
14	pretty	0.06053339
15	restaurant	0.09034333
16	salad	0.06126506
17	sauce	0.06254664
18	service	0.16307827
19	staff	0.05765902
20	time	0.12771353

Top 20 words associated with lowest star rating

	word	totWS
1	aloud	7.926930e-05
2	appalling	7.857820e-05
3	bathing	7.073849e-05
4	bullshit	7.114014e-05
5	canceled	7.708425e-05
6	cockroaches	7.352755e-05
7	disrespected	7.303573e-05
8	enemy	6.250641e-05
9	glue	7.559031e-05
10	hagar	6.250641e-05
11	hires	7.218757e-05
12	kerry	7.478701e-05
13	praying	7.765307e-05
14	puke	7.457498e-05
15	reasoning	7.870050e-05
16	responds	7.522077e-05
17	riders	7.883554e-05
18	scam	6.709161e-05
19	shameful	7.154178e-05
20	violations	7.962564e-05

Words like time, service, food, restaurant, and chicken are associated with highest star ratings but are neutral in sense. We have used a threshold value to remove these common words.
Plot of proportion of word occurrences by star ratings after removing common words -



Top 20 words associated with highest star rating after removing common words

	word	totWS1
1	amazing	0.05637144
2	bar	0.06742154
3	cheese	0.06397699
4	delicious	0.08025833
5	dinner	0.04907584
6	eat	0.05690917
7	fresh	0.06311515
8	friendly	0.06983530
9	love	0.08066498
10	lunch	0.06311650
11	meal	0.05302132
12	menu	0.08426760
13	nice	0.08859558
14	night	0.04922673
15	people	0.05278307
16	pizza	0.06327508
17	pretty	0.06389033
18	salad	0.06474757
19	sauce	0.06609337
20	staff	0.06099183

Top 20 words associated with lowest star rating after removing common words

	word	totWS1
1	accused	8.430374e-05
2	aloud	8.389184e-05
3	appalling	8.344250e-05
4	bathing	7.502750e-05
5	bullshit	7.543940e-05
6	canceled	8.174047e-05
7	cockroaches	7.805032e-05
8	disrespected	7.755332e-05
9	enemy	6.631109e-05
10	glue	8.003844e-05
11	hagar	6.631109e-05
12	hires	7.647253e-05
13	kerry	7.921465e-05
14	praying	8.202656e-05
15	puke	7.908345e-05
16	reasoning	8.305969e-05
17	responds	7.970469e-05
18	riders	8.340180e-05
19	scam	7.125225e-05
20	shameful	7.585129e-05

(c) We will consider three dictionaries, available through the tidytext package – the NRC dictionary of terms denoting different sentiments, the extended sentiment lexicon developed by Prof Bing Liu, and the AFINN dictionary which includes words commonly used in user-generated content in the web. The first provides lists of words denoting different sentiment (for eg., positive, negative, joy, fear, anticipation, ...), the second specifies lists of positive and negative words, while the third gives a list of words with each word being associated with a positivity score from -5 to +5.

How many matching terms are there for each of the dictionaries?

Consider using the dictionary based positive and negative terms to predict sentiment (positive or negative based on star rating) of a movie. One approach for this is: using each dictionary, obtain an aggregated positiveScore and a negativeScore for each review; for the AFINN dictionary, an aggregate positivity score can be obtained for each review. Describe how you obtain predictions based on aggregated scores. Are you able to predict review sentiment based on these aggregated scores, and how do they perform? Does any dictionary perform better?

So far, we have done some basic data exploration and we have analysed the positive/negative nature of words by means of stars assigned to each review. We need to rather find the opinion behind each review and using stars for doing so is unreliable. Hence, we perform sentiment analysis. In this process, not only do we find the sentiment behind each word, but we also see the contribution of each word to its sentiment.

We use 3 data dictionaries NRC, Bing, AFINN as they help us evaluate sentiment using different metrics for each individual word. We must note that the overall sentiment of the review will be the sum of sentiment values/scores of the tokens that comprise it.

NRC dictionary - It uses the preexisting words and their sentiments to categorize new words in a binary format into categories of positive, negative, anger, sadness, joy etc.

AFINN dictionary - Consists of a list of words rated from -5 (negative) to 5 (positive) and uses this to compare and rate new words similarly

Bing dictionary - Consists of words that are classified as positive and negative, uses this to classify new words accordingly

The total number of words that exist in each dictionary is:
Bing Dictionary – Total number of terms = 6787
NRC Dictionary – Total number of terms = 13875
AFINN Dictionary – Total number of terms = 2477

Number of terms matching in all three dictionaries = 2154

We compare our tokens with these individual dictionaries by extracting the words using the get_sentiment() function and then performing an inner join on it.

On running the dictionaries, we have found out that -

Dictionary	Number of common occurrences with our Tokenized set
Bing	250,594
NRC	1,011,984
AFINN	201,398

This count is high for our NRC library because for one review there could be multiple instances of the same word for different values of the sentiment. For example the word friendly has sentiments like anticipation, joy, positive, and trust.

We see that we have maximum matching tokens with the NRC dictionary

To make predictions of reviews we must first find out the aggregated sentiment scores.

For each of our dictionaries we have used inner join to match the sentiment dictionary with our words. Next, we find out the total positive/negative sentiment words per review. Finally, we calculate the aggregated sentiment scores based on the proportion of positive and negative words as follows –

sentiScore=Proportion of positive words-Proportion of negative words

We now create a variable “hiLo” based on the star ratings of the reviews. For star ratings 1, and 2 we assign value -1 to this variable. We assign value 1 to “hiLo” for star ratings 4, and 5. We assign value 0 to this variable for star rating 3 which we later discard.

We then create a variable “pred_hilo” based on the value of average “sentiScore”. If “sentiScore” is greater than average score for star rating 3 then assign “pred_hiLo” as 1, otherwise as -1.

Now create a confusion matrix for actual “hiLo” values Vs. predicted “pred_hiLo” values to calculate accuracy for all three dictionaries.

- Bing Dictionary

Most positive words in reviews

	word	sentiment	totOcc
1	love	positive	9693
2	nice	positive	8118
3	delicious	positive	7090
4	friendly	positive	6254
5	fresh	positive	5736
6	pretty	positive	5671
7	amaze	positive	5024
8	enjoy	positive	4352
9	recommend	positive	3940
10	favorite	positive	3643

Most negative words in reviews

	word	sentiment	totOcc
1	bad	negative	-5124
2	disappoint	negative	-2940
3	cold	negative	-1926
4	hard	negative	-1908
5	cheap	negative	-1491
6	wrong	negative	-1460
7	slow	negative	-1428
8	miss	negative	-1197
9	bland	negative	-1184
10	lack	negative	-1181

Review star ratings correspond to the positive/negative sentiment words –

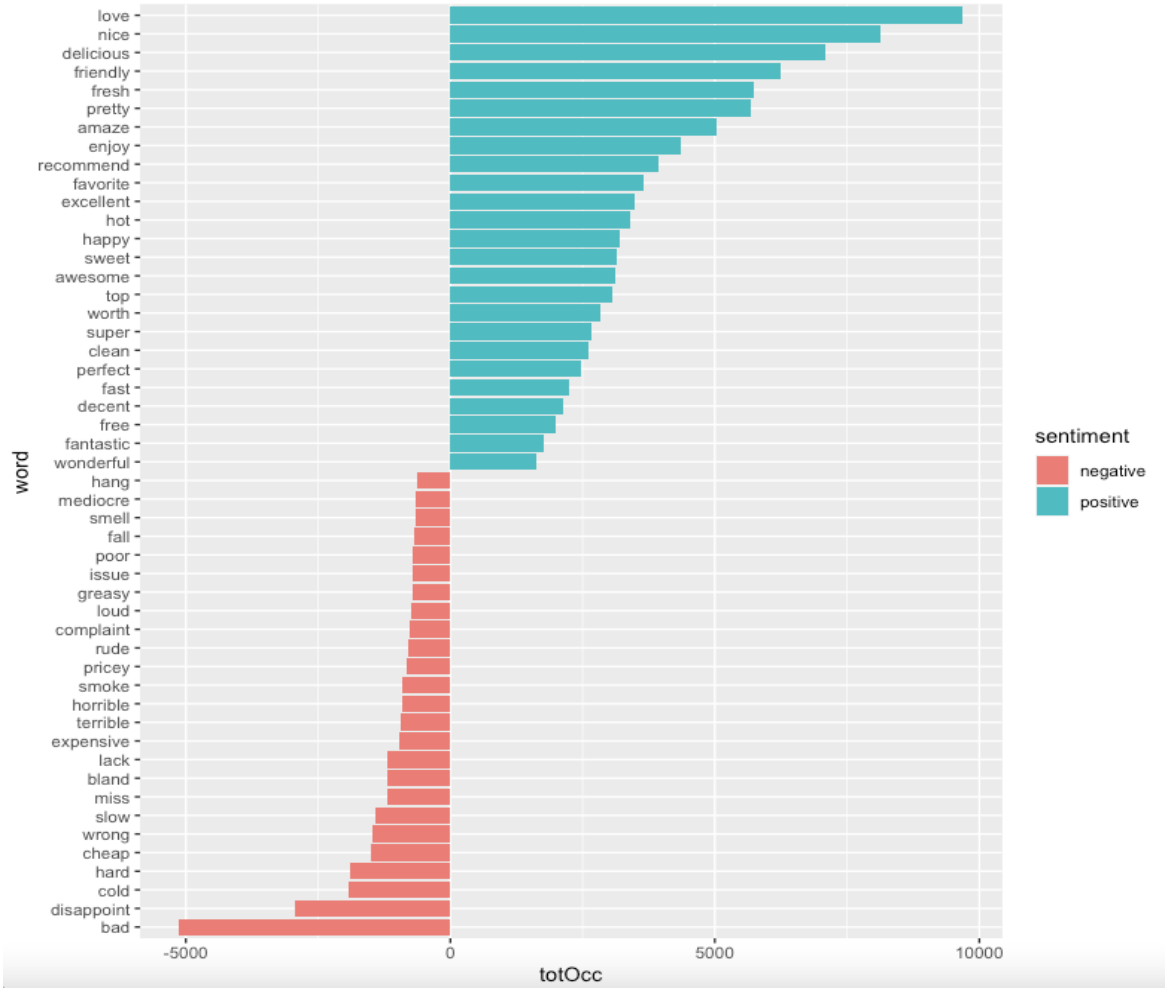
A tibble: 5 × 4

starsReview <dbl>	avgPos <dbl>	avgNeg <dbl>	avgSentiSc <dbl>
1	0.3107697	0.6892303	-0.3784607
2	0.4483663	0.5516337	-0.1032674
3	0.6104126	0.3895874	0.2208253
4	0.7550768	0.2449232	0.5101536
5	0.8322997	0.1677003	0.6645993

5 rows

Average sentiment score is greater for higher star ratings and vice-versa.

Top 25 most positive and most negative words in reviews –



Confusion matrix for Bing dictionary –

```
      predicted
actual    -1     1
    -1  6929  1434
    1   4706 20528
```

Accuracy = 81.72%

- NRC Dictionary

Count of words for the different sentiment categories –

A tibble: 10 × 3

sentiment <chr>	count <int>	sumn <int>
anger	232	42192
anticipation	303	101470
disgust	205	32561
fear	242	34050
joy	283	124231
negative	603	102199
positive	752	246186
sadness	223	40408
surprise	190	49242
trust	390	121196

1–10 of 10 rows

Top positive (good reviews) words -

	word	sentiment	totOcc	goodBad
1	wait	anticipation	7032	7032
2	friendly	anticipation	6254	6254
3	pretty	anticipation	5671	5671
4	star	anticipation	4708	4708
5	love	joy	9693	9693
6	delicious	joy	7090	7090
7	friendly	joy	6254	6254
8	pretty	joy	5671	5671
9	beer	joy	4894	4894
10	star	joy	4708	4708

Top negative (bad reviews) words -

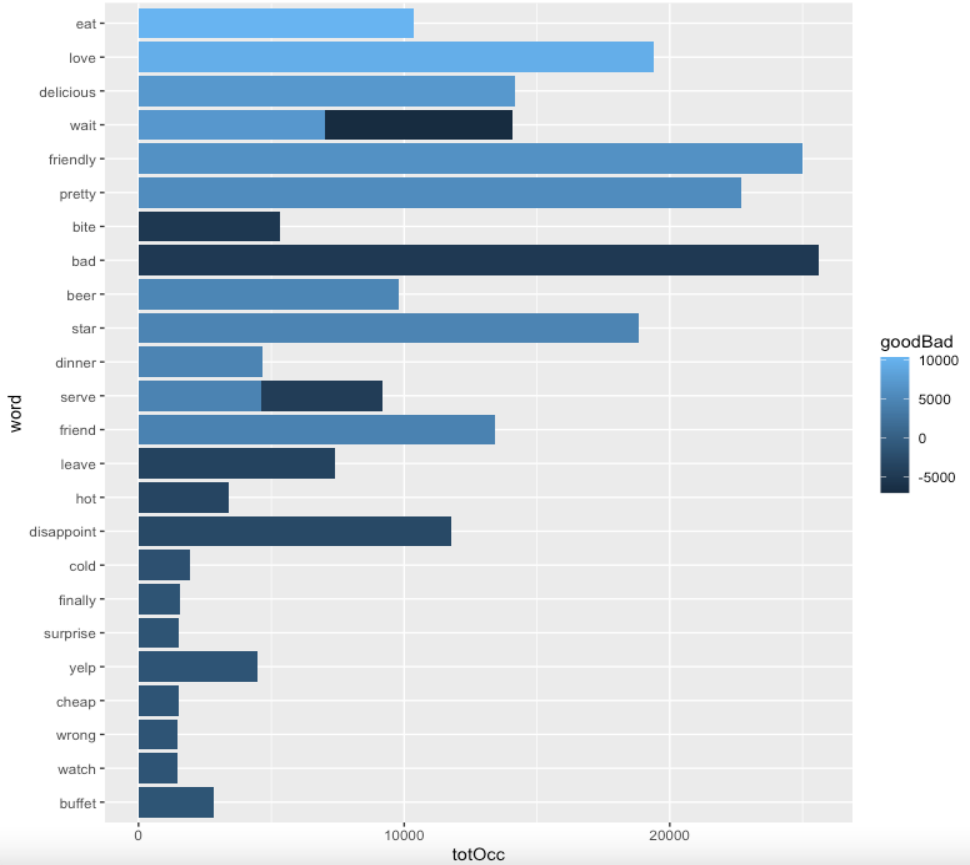
	word	sentiment	totOcc	goodBad
1	bad	anger	5124	-5124
2	hot	anger	3387	-3387
3	disappoint	anger	2940	-2940
4	yelp	anger	1492	-1492
5	bad	disgust	5124	-5124
6	disappoint	disgust	2940	-2940
7	finally	disgust	1576	-1576
8	bad	fear	5124	-5124
9	surprise	fear	1516	-1516
10	yelp	fear	1492	-1492

Confusion matrix for NRC dictionary –

```
        predicted
actual     -1     1
    -1   6696  1864
     1   7702 17758
```

Accuracy = 71.88%

Top 25 most positive and most negative words in reviews –



- AFINN Dictionary

Review star ratings correspond to the positive/negative sentiment words –

A tibble: 5 × 3

starsReview <dbl>	avgLen <dbl>	avgSenti <dbl>
1	4.994694	-2.3863899
2	5.051773	0.7765985
3	4.983273	3.7701958
4	4.870839	6.5102021
5	4.343821	7.2800636

5 rows

Confusion matrix for AFINN dictionary –

	predicted	
actual	-1	1
-1	5117	3081
1	2132	22572

Accuracy = 84.15%

Best Dictionary – Based on our analysis of three dictionaries (NRC, Extended Sentiment Lexicon, and AFINN), we got the highest prediction accuracy of 84.15% for AFINN dictionary.

(d) Develop models to predict review sentiment.
For this, split the data randomly into training and test sets. To make run times manageable, you may take a smaller sample of reviews (minimum should be 10,000).

One may seek a model built using only the terms matching any or all of the sentiment dictionaries, or by using a broader list of terms (the idea here being, maybe words other than only the dictionary terms can be useful). You should develop at least three different types of models (Naïve Bayes, and at least two others of your choice ....Lasso logistic regression (why Lasso?), xgb, svm, random forest (ranger).

(i) Develop models using only the sentiment dictionary terms – try the three different dictionaries;

how do the dictionaries compare in terms of predictive performance? Then with a combination of the three dictionaries, ie. combine all dictionary terms.
Do you use term frequency, tfidf, or other measures, and why? What is the size of the document-term matrix? Should you use stemming or lemmatization when using the dictionaries?

We are using the complete data set to develop our models. We have used Random Forest, Naïve Bayes, and SVM models on all three dictionaries. We have used term frequency- inverse document frequency (tfidf) values to create our Document Term Matrix.

Term Frequency – Count of the word that occurs in a document.

Inverse Document Frequency – This is log of total number of documents, divided by the number of documents that the word is present in.

tf-idf – Term Frequency * Inverse Document Frequency

Terms that occur multiple times in a document represent the document content/meaning. Terms that occur across many documents are not useful for differentiating between documents. The idea behind using tf-idf is to give more importance to those words that occur more frequently in one document and less frequently in other documents. These words are more useful in classifying the reviews as positive or negative.

Size of our Document Term Matrix (Bing) = 33597 (reviews) * 1132 (variables)

Size of our Document Term Matrix (NRC) = 39494 (reviews) * 1560 (variables)

Size of our Document Term Matrix (AFINN) = 38163 (reviews) * 624 (variables)

Size of our Document Term Matrix (Combined Dictionaries) = 34344 (reviews) * 2106 (variables)

Size of our Document Term Matrix (Broader Terms) = 34520 (reviews) * 3540 (variables)

We have used lemmatization, instead of stemming when using the dictionaries for our models. Lemmatization replaces the words by their accurate lemma, keeping the context of the word's use intact. Stemming on the other hand reduces derived forms and inflections of words to a common base form.

Example – caring

Stemming – car

Lemmatize (without context) – care

Lemmatize (with context) - caring

The words we get after applying stemming would not match any dictionary terms and this would reduce the model's predictive power. Words we get after lemmatization would match the dictionary terms and help in classifying the sentiments.

- Random forest using Bing dictionary

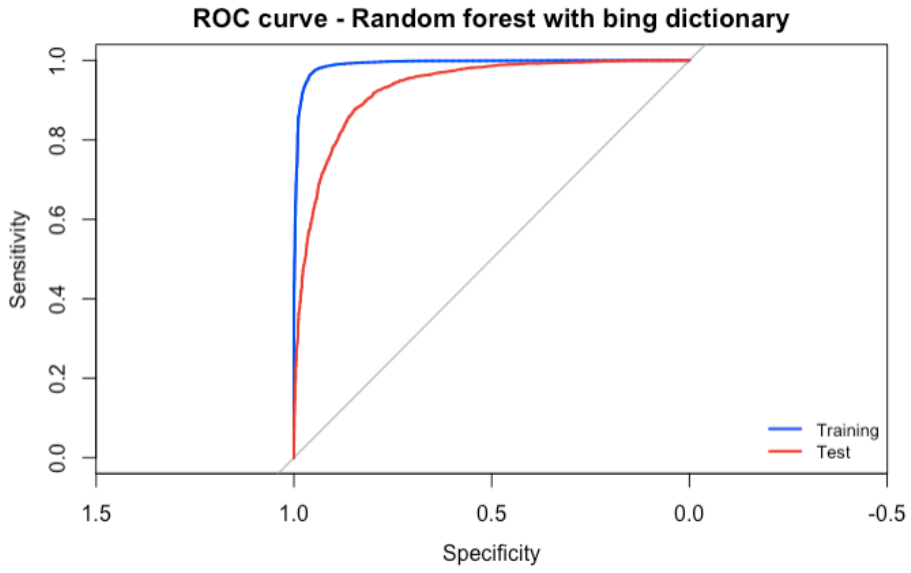
Training confusion matrix

actual	preds FALSE	TRUE
-1	5245	589
1	197	17486

Testing confusion matrix

actual	preds FALSE	TRUE
-1	1807	722
1	351	7200

ROC curve for random forest using Bing dictionary -



Model Details -

```
> rfModel1_bing
Ranger result

Call:
 ranger(dependent.variable.name = "hiLo", data = revDTM_sentiBing_trn %>%      select(-review_id), num.trees = 200,
 importance = "permutation",      probability = TRUE)

Type:                             Probability estimation
Number of trees:                  200
Sample size:                      23517
Number of independent variables:  1130
Mtry:                             33
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.0847654
```

Test Accuracy = 89.36335%

- Random forest using NRC dictionary

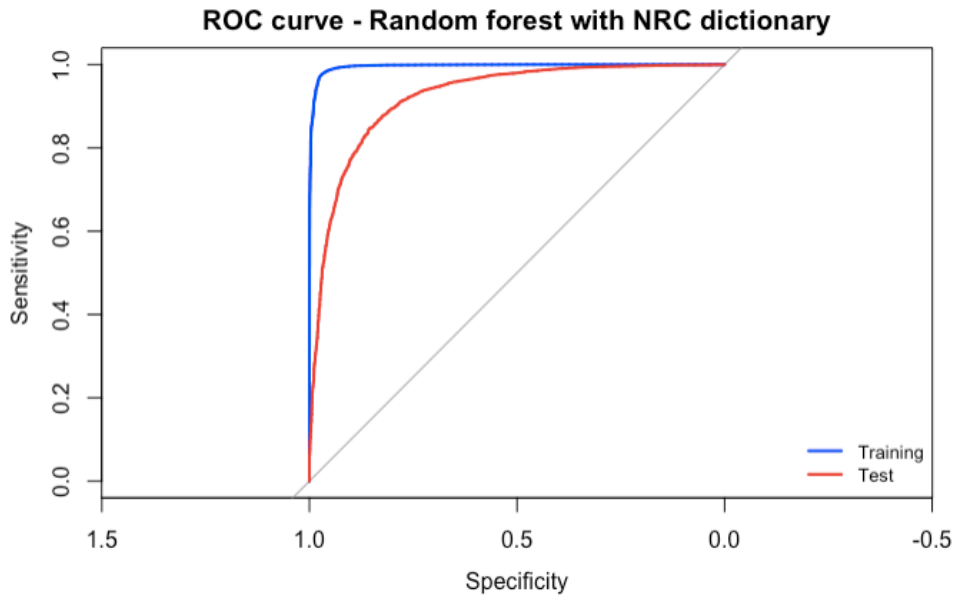
Training confusion matrix

```
      preds
actual FALSE  TRUE
    -1  5398   514
    1     92 17810
```

Testing confusion matrix

```
      preds
actual FALSE TRUE
    -1  1731  917
    1    318 7240
```

ROC curve for random forest using NRC dictionary -



Model Details -

```
> rfModel1_nrc
Ranger result

Call:
 ranger(dependent.variable.name = "hiLo", data = revDTM_sentiNRC_trn %>%      select(-review_id), num.trees = 200,
 importance = "permutation",      probability = TRUE)

Type:                             Probability estimation
Number of trees:                  200
Sample size:                      23814
Number of independent variables:  1558
Mtry:                             39
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.09045346
```

Test Accuracy = 87.89927%

- Random forest using AFINN dictionary

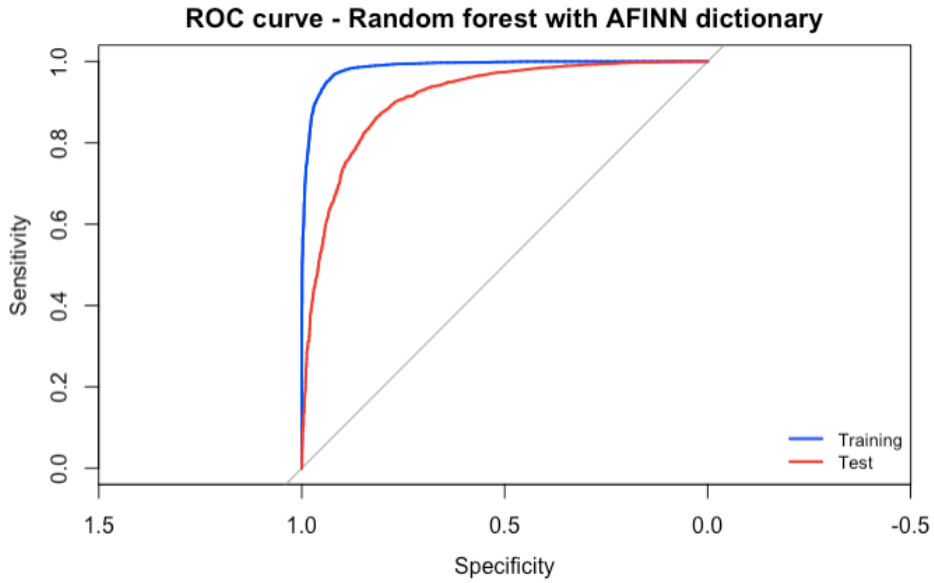
Training confusion matrix

actual	preds FALSE	TRUE
-1	4902	798
1	248	17083

Testing confusion matrix

actual	preds FALSE	TRUE
-1	1615	883
1	402	6971

ROC curve for random forest using AFINN dictionary -



Model Details -

```
> rfModel1_afinn
Ranger result

Call:
 ranger(dependent.variable.name = "hiLo", data = revDTM_sentiAFINN_trn %>%      select(-review_id), num.trees = 20
0, importance = "permutation",      probability = TRUE)

Type:                             Probability estimation
Number of trees:                  200
Sample size:                      23031
Number of independent variables:  622
Mtry:                             24
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.09191856
```

Test Accuracy = 86.98207%

- Naive Bayes model using Bing dictionary

Training confusion matrix

```
      predicted
actual FALSE  TRUE
    -1  4190  1644
    1   6291 11392
```

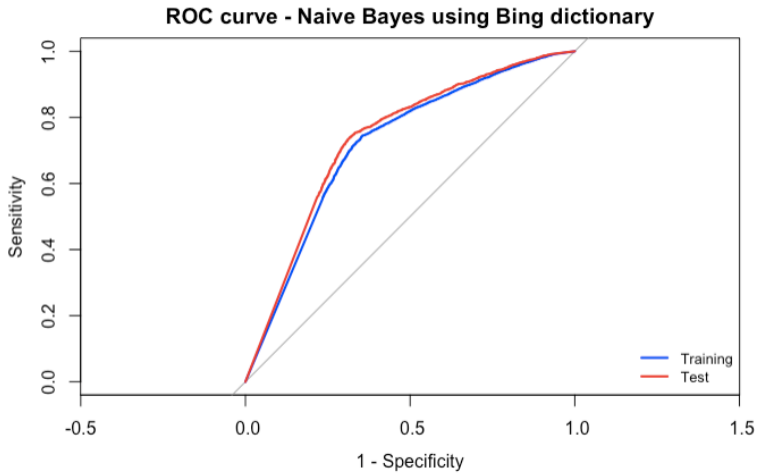
Testing confusion matrix

```
      predicted
actual FALSE TRUE
    -1  1864  665
    1   2624 4927
```

Training AUC value – 0.7187

Testing AUC value – 0.7356

ROC curve for naïve bayes using Bing dictionary -



Test Accuracy = 67.37103%

- Naive Bayes model using NRC dictionary

Training confusion matrix

actual	predicted FALSE	predicted TRUE
-1	4446	1466
1	8771	9131

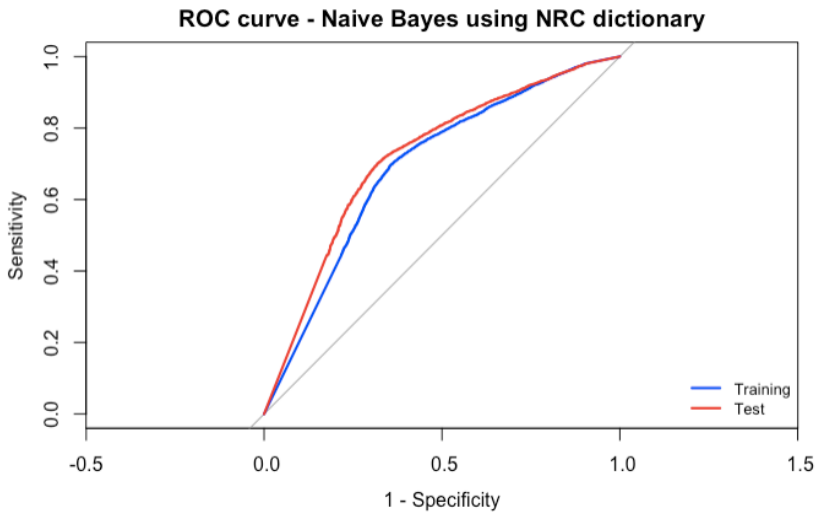
Testing confusion matrix

actual	predicted FALSE	predicted TRUE
-1	2098	550
1	3692	3866

Training AUC value – 0.6908

Testing AUC value – 0.7183

ROC curve for naïve bayes using NRC dictionary -



Test Accuracy = 58.43621%

- Naive Bayes model using AFINN dictionary

Training confusion matrix

```
      predicted
actual FALSE  TRUE
    -1  2889  2811
    1   2787 14544
```

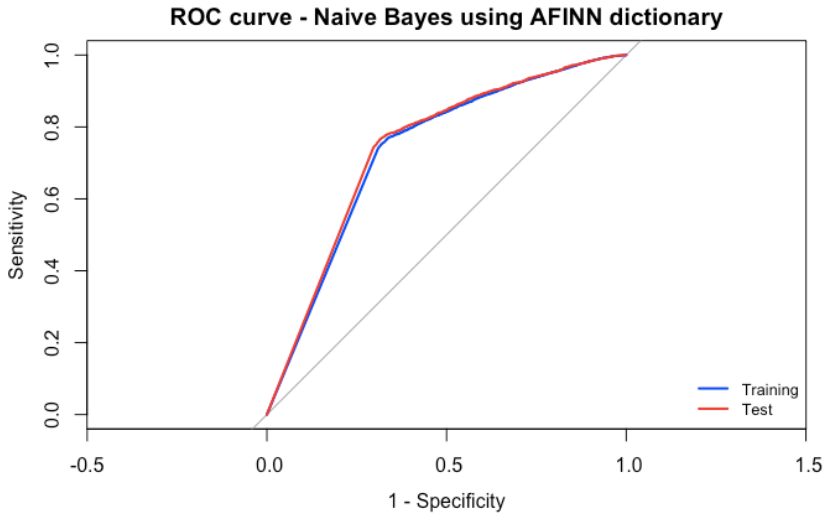
Testing confusion matrix

```
      predicted
actual FALSE TRUE
    -1  1248 1250
    1   1123 6250
```

Training AUC value – 0.7334

Testing AUC value – 0.7416

ROC curve for naïve bayes using AFINN dictionary -



Test Accuracy = 75.95988%

- SVM model using Bing dictionary

SVM model with kernel = ”radial”, and cost = 1 -

Training confusion matrix

```
      predicted
actual    -1     1
    -1     0  5834
    1      0 17683
```

Testing confusion matrix

```
      predicted
actual    -1    1
    -1     0 2529
    1      0 7551
```

SVM model with kernel = ”radial”, cost = 5, and gamma = 5 -

Training confusion matrix

```
      predicted
actual    -1     1
    -1  5253   581
    1    216 17467
```

Testing confusion matrix

```
      predicted
actual    -1     1
    -1 1800   729
    1   375 7176
```

SVM model with kernel = ”radial”, cost = 10, and gamma = 0.5 –

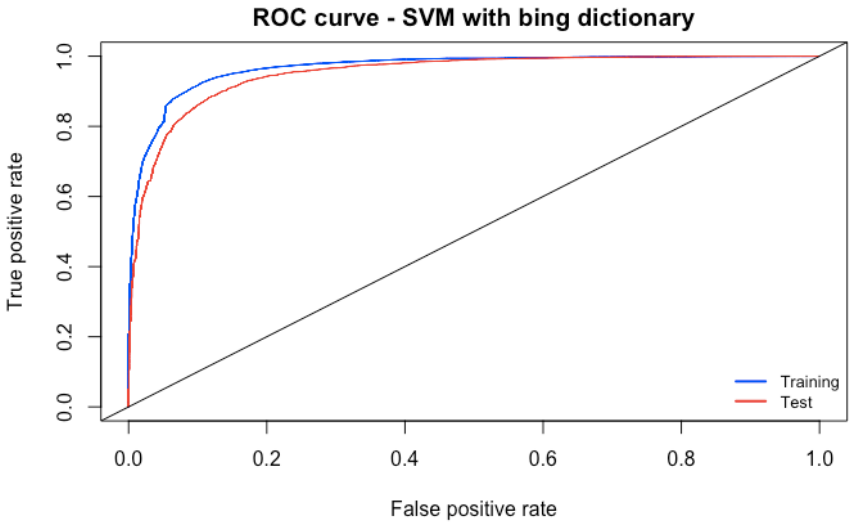
Training confusion matrix

```
      predicted
actual    -1     1
    -1  4583  1251
    1    537 17146
```

Testing confusion matrix

```
      predicted
actual    -1     1
    -1 1888   641
    1   323 7228
```

ROC curve for SVM using Bing dictionary -



Test Accuracy = 90.43651%

- SVM model using NRC dictionary

SVM model with kernel = ”radial”, cost = 10, and gamma = 0.5

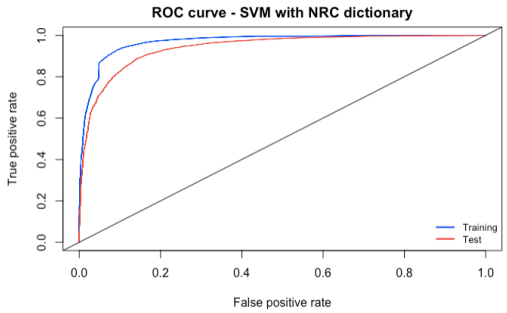
Training confusion matrix

```
       predicted
actual    -1     1
    -1  4712  1200
     1   453 17449
```

Testing confusion matrix

```
       predicted
actual    -1    1
    -1 1888  760
     1  357 7201
```

ROC curve for SVM using NRC dictionary -



Test Accuracy = 89.05546%

- SVM model using AFINN dictionary

SVM model with kernel = ”radial”, cost = 10, and gamma = 0.5

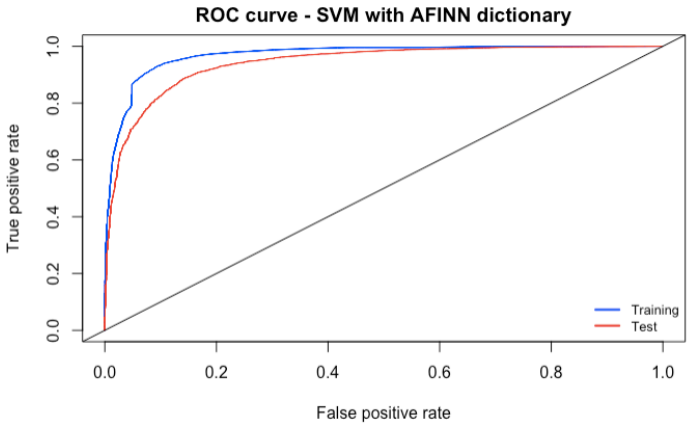
Training confusion matrix

actual	predicted -1	1
-1	4088	1612
1	654	16677

Testing confusion matrix

actual	predicted -1	1
-1	1632	866
1	353	7020

ROC curve for SVM using AFINN dictionary -



Test Accuracy = 87.65069%

Models using a combination of all three dictionaries –

- Random forest Model

Training set confusion matrix

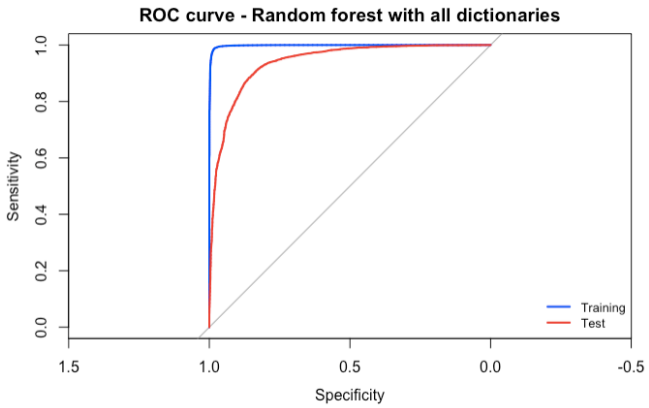
actual	preds FALSE	TRUE
-1	5705	318
1	67	17950

Testing set confusion matrix

actual	preds FALSE	TRUE
-1	1766	817
1	279	7442

Test Accuracy = 89.36335%

ROC curve for random forest using all dictionaries -



- Naive Bayes model

Training set confusion matrix

actual	predicted FALSE	TRUE
-1	4493	1530
1	7906	10111

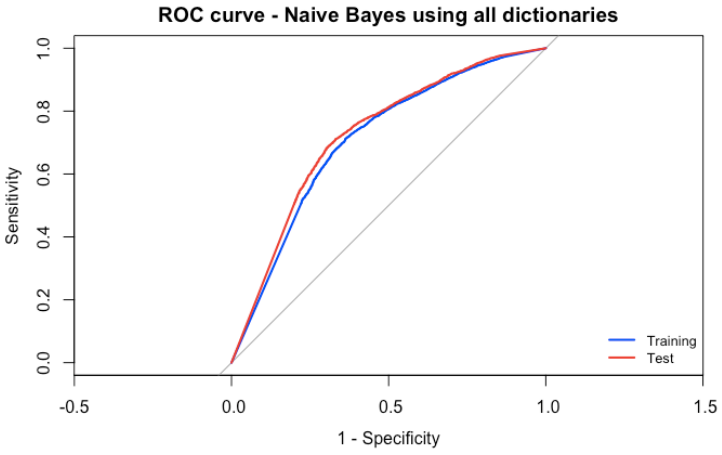
Testing set confusion matrix

actual	predicted FALSE	TRUE
-1	1999	584
1	3410	4311

Training auc – 0.708

Testing auc – 0.7246

ROC curve for naïve bayes using all dictionaries -



Test Accuracy = 61.23835%

- SVM Model

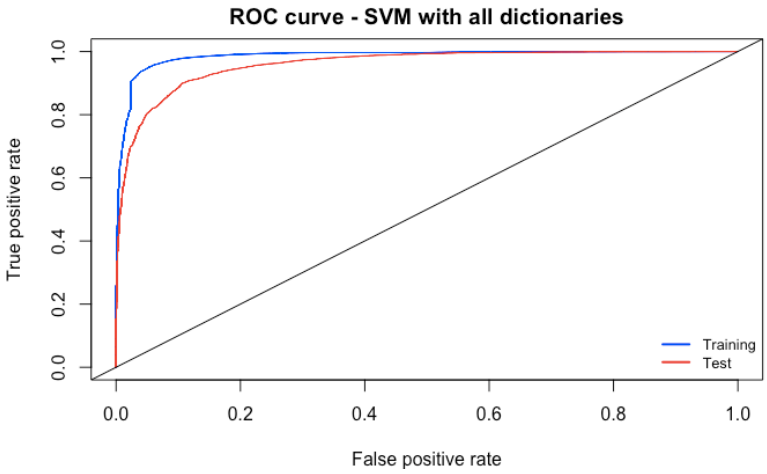
Training set confusion matrix

```
      predicted
actual    -1     1
    -1  5322   701
     1   355 17662
```

Testing set confusion matrix

```
      predicted
actual   -1    1
    -1 2021  562
     1  364 7357
```

ROC curve for SVM using all dictionaries -



Test Accuracy = 91.0132%

(d)

(ii) Develop models using a broader list of terms (i.e. not restricted to the dictionary terms only) – how do you obtain these terms? Will you use stemming here?

Report on performance of the models. Compare performance with that in part (c) above. How do you evaluate performance? Which performance measures do you use, why.

We have obtained a broader list of terms to generate our models and classify the reviews. We have used the lemmatized set of words to create the broader Document Term Matrix. Firstly, we have removed those words from our tokenized set which occur in more than 90% of reviews, and less than 30 reviews. Next, we do not match our data set's words with those of the words from dictionaries, and instead create the Document Term Matrix using the pivot wider function. We have used our lemmatized data set to create this broader Document Term Matrix. We do not use stemming here.

We have evaluated performance of our models using AUC values, ROC curves, confusion matrix, and accuracy.

Sentiment analysis is just another form of classification. In our assignment, we are classifying text based on the labels 'positive' and 'negative'. We have used these performance metrics as these are the best ways in which we evaluate classification models.

Count of reviews each word occurs in -

Top 10 words

word <chr>	nr <int>
eat	7871
love	7492
nice	6554
price	6552
delicious	6143
menu	6031
friendly	5902
taste	5398
wait	5163
staff	5105

Bottom 10 words

word <chr>	nr <int>
aji	6
angela	6
ant	6
carrie	6
deschutes	6
ghee	6
kerry	6
kudzu	6
massaman	6
meyer	6

- Random Forest Model on Broader Terms

Training confusion matrix

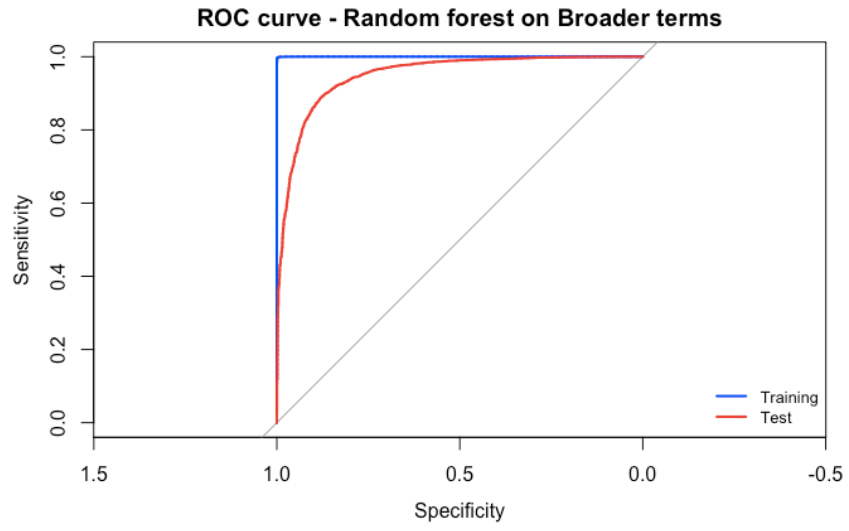
actual	preds FALSE	TRUE
-1	5970	78
1	8	18108

Testing confusion matrix

actual	preds FALSE	TRUE
-1	1807	790
1	229	7530

Test Accuracy = 90.16029%

ROC curve for random forest using broader terms -



- Naive Bayes Model on Broader Terms

Training confusion matrix

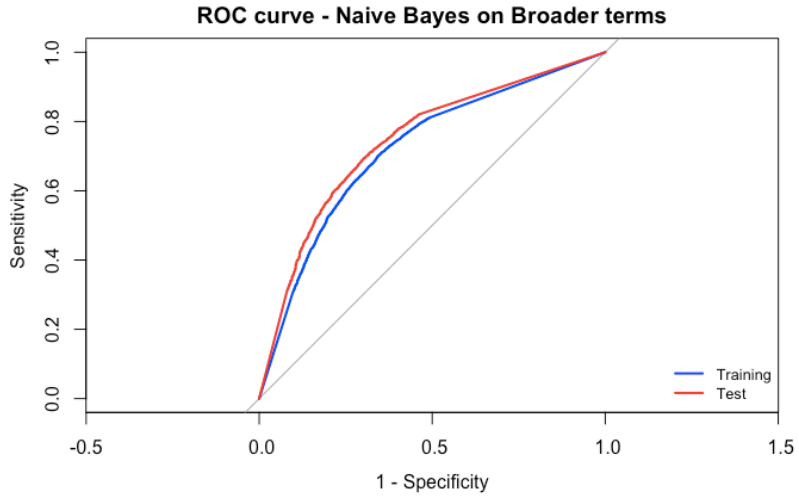
actual	predicted FALSE	TRUE
-1	5401	647
1	12150	5966

Testing confusion matrix

actual	predicted FALSE	TRUE
-1	2364	233
1	5168	2591

Test Accuracy = 47.84666%

ROC curve for naïve bayes using broader terms -



- SVM Model on Broader Terms

Training confusion matrix

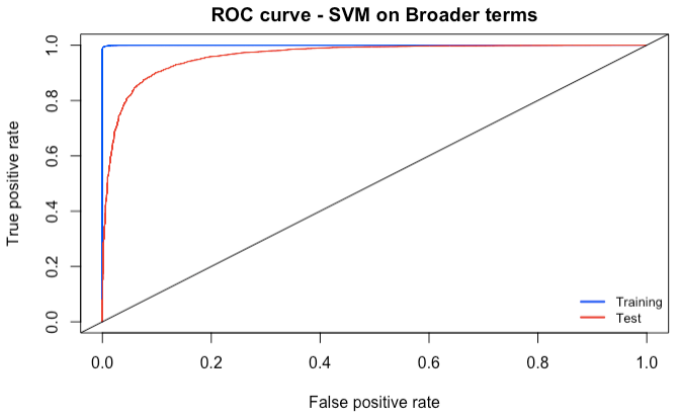
actual	predicted -1	1
-1	5982	66
1	34	18082

Testing confusion matrix

actual	predicted -1	1
-1	2105	492
1	347	7412

Test Accuracy = 91.89842%

ROC curve for SVM using broader terms -



Comparison of all the model's performance –

Models	Dictionaries	Training AUC	Test AUC	Test Accuracy
Random Forest	Bing	0.9915	0.9311	89.36%
Random Forest	NRC	0.996	0.9224	87.90%
Random Forest	AFINN	0.986	0.9094	86.98%
Naïve Bayes	Bing	0.7187	0.7356	67.37%
Naïve Bayes	NRC	0.6908	0.7183	58.44%
Naïve Bayes	AFINN	0.7334	0.7416	75.96%
SVM	Bing	0.9670708	0.9505338	90.44%
SVM	NRC	0.9693956	0.9408556	89.06%
SVM	AFINN	0.946533	0.9288142	87.65%
Random Forest	Combined	0.9983	0.9392	89.36%
Naïve Bayes	Combined	0.708	0.7246	61.24%
SVM	Combined	0.9856596	0.9594373	91.01%
Random Forest	Broader term	1	0.949	90.16%
Naïve Bayes	Broader term	0.7194	0.7433	47.85%
SVM	Broader term	0.9997441	0.9628084	91.90%

Best Model – SVM model on the Broader Document Term Matrix proved to be the best model with a test AUC value of 0.9628084, and test accuracy of 91.90%.

(e) Consider some of the attributes for restaurants – this is specified as a list of values for various attributes in the 'attributes' column. Extract different attributes (see note below).

(i) Consider a few interesting attributes and summarize how many restaurants there are by values of these attributes; examine if star ratings vary by these attributes.

Attribute - Ambience

- Count of amb='classy' for different star ratings

There are 17 restaurants with ambience 'classy'

	starsReview	n
1	1	182
2	2	179
3	3	284
4	4	423
5	5	586

- Count of amb='trendy' for different star ratings

There are 20 restaurants with ambience 'trendy'

	starsReview	n
1	1	216
2	2	251
3	3	301
4	4	505
5	5	714

- Count of amb='casual' for different star ratings

There are 313 restaurants with ambience 'casual'

	starsReview	n
1	1	3258
2	2	3002
3	3	4014
4	4	8081
5	5	11475

We can see that the reviews with ambience as 'classy', 'casual', and 'trendy' increase from star-ratings 1 to 5. This shows that this attribute is important as it influences the star-ratings.

Attribute – Outdoor Seating

Count of restaurants by values of attribute - Outdoor Seating

OutdoorSeating <chr>	n <int>
False	238
True	212
NA	6

Star Ratings Vs. Outdoor seating

OutdoorSeating <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	20172	39	27	13	10	11
True	19437	36	27	15	11	11
NA	355	40	25	13	8	13

Outdoor seating cannot be considered an important attribute because value of outdoor seating (true, false) is evenly distributed within different star-ratings.

Attribute - Alcohol

Count of restaurants on basis of alcohol

Alcohol <chr>	n <int>
beer_and_wine	71
full_bar	205
none	171
NA	9

Star-rating Vs. alcohol (full bar) –

	starsReview	n
1	1	2176
2	2	2173
3	3	2941
4	4	5141
5	5	6281

Star-rating Vs. alcohol (none) –

	starsReview	n
1	1	1478
2	2	1149
3	3	1588
4	4	3558
5	5	5825

Restaurants have got 5 star-rating reviews regardless of no alcohol serving present. This trend is same for those restaurants that do have alcohol. So, this attribute is not important as it has no influence on the star-rating of reviews.

Attribute – Restaurant table service

RestaurantsTableService <chr>	n <int>
False	134
True	307
NA	15

Star-rating Vs. Restaurant table service

RestaurantsTableService <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	10992	39	26	13	9	12
True	28058	37	27	14	11	11
NA	914	43	23	11	8	14

There is not any difference in the distribution of star-ratings based on restaurant table service being present or not. So, attribute 'Restaurant table service' is not important.

Attribute – Noise Level

Restaurant count based on values of Noise Level

NoiseLevel <chr>	n <int>
average	316
loud	31
quiet	79
very_loud	9
NA	21

Star rating Vs. Noise Level

NoiseLevel <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
average	30433	39	27	14	10	11
loud	2673	20	27	18	16	20
quiet	5170	41	25	13	10	11
very_loud	599	25	23	18	16	18
NA	1089	41	27	15	8	9

For noise levels 'loud', and 'very loud' the percent of 1 star ratings are very high as compared to reviews on restaurants with noise levels as 'average', and 'quiet'. This makes this attribute an interesting one as it influences the star ratings.

Attribute – Good for kids

Restaurant count based on values of Good for kids

GoodForKids <chr>	n <int>
False	95
True	352
NA	9

Star rating Vs. Good for kids

GoodForKids <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	8804	32	29	17	11	11
True	30621	39	27	13	10	11
NA	539	49	23	10	7	12

There is not any difference in the distribution of star-ratings based on 'good for kids' being true or false. So, attribute 'Good for kids' is not important.

Attribute – Good for groups

Restaurant count based on values of Good for groups

RestaurantsGoodForGroups <chr>	n <int>
False	36
True	411
NA	9

Star rating Vs. Good for groups

RestaurantsGoodForGroups <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	2871	41	25	13	10	12
True	36526	37	27	14	10	11
NA	567	47	23	11	8	11

There is not any difference in the distribution of star-ratings based on 'Good for groups' being true or false. So, attribute 'Good for groups' is not important.

Attribute – Price range

Restaurant count based on values of Price range

RestaurantsPriceRange2 <chr>	n <int>
1	173
2	259
3	19
4	5

Star rating Vs. Price range

RestaurantsPriceRange2 <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
1	12477	40	27	12	9	12
2	25183	37	27	14	11	11
3	1931	35	27	17	10	10
4	373	46	18	19	11	6

Restaurant with price range as 4 has the highest percent of 5 star ratings and lowest percent of 1 star ratings. This inference makes this attribute an interesting one.

Attribute – Wifi

Restaurant count based on values of Wifi

WiFi <chr>	n <int>
free	198
no	217
paid	5
NA	36

Star rating Vs. Wifi

WiFi <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
free	18189	39	26	14	10	11
no	19895	37	27	13	10	12
paid	288	18	24	21	18	19
NA	1592	29	33	16	10	12

Restaurants which have ‘paid’ Wifi have lower percent of 5 star ratings and higher percent of 1 star ratings as compared to restaurants which offer ‘free’ or ‘no’ Wifi. This makes attribute ‘Wifi’ and influential one.

Attribute – Happy hour

Restaurant count based on values of Happy hour

HappyHour <chr>	n <int>
False	10
True	77
NA	369

Star rating Vs. Happy hour

HappyHour <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	866	45	23	10	7	15
True	6762	31	29	16	12	12
NA	32336	39	27	13	10	11

There is not any difference in the distribution of star-ratings based on Happy hour being present or not. So, attribute ‘Happy hour’ is not important.

Attribute – Drive through

Restaurant count based on values of Drive through

DriveThru <chr>	n <int>
False	41
True	12
NA	403

Star rating Vs. Drive through

DriveThru <chr>	nratings <int>	n5star <dbl>	n4star <dbl>	n3star <dbl>	n2star <dbl>	n1star <dbl>
False	2679	30	28	16	13	13
True	820	36	22	11	12	20
NA	36465	38	27	14	10	11

Restaurants where drive through is present has higher percent of 1 star rating reviews. This potentially makes this attribute an interesting one.

(ii) For one of your models (choose your 'best' model from above), does prediction accuracy vary by certain restaurant attributes? You do not need to look into all attributes; choose a few which you think may be interesting, and examine these.

Note: for question (e), you will consider the values in the 'attribute' column. This has values of multiple attributes, separated by a '|'. Further, some of the values, like Ambience, carry a list of True/False values (like, for example, Ambience: {'romantic': False, 'intimate': False, 'classy': False, 'hipster': False, ...}. Care must be taken to extract values for different attributes. You can consider a separate dataframe with review_id, attribute, and then process this further to extract values for the different attributes.

The best model we got is SVM model on broader terms with **test AUC value of 0.9628084, and test accuracy of 91.90%.**

We had created a data frame by extracting the attributes column. We used left join to merge this data frame with the training and test data sets of broader terms.

Next, we added a new column for predicted hiLo values from the SVM model to our training and test datasets.

Then we checked how accuracy varies for different values of certain attributes.

Accuracy group by Attribute Noise Level –

NoiseLevel <chr>	AvgAcc <dbl>
average	0.9215391
loud	0.8945687
quiet	0.9227014
very_loud	0.8666667
NA	0.9079422

Our model is able to predict reviews with Noise Level 'average' , and 'quiet' more accurately than it can for Noise Level 'loud,', and 'very_loud'.

Accuracy group by Attribute Wifi –

WiFi <chr>	AvgAcc <dbl>
free	0.9231681
no	0.9152120
paid	0.9555556
NA	0.9110764

Models predicts more accurately for restaurants which have paid Wifi.

Accuracy group by Attribute DriveThru –

DriveThru <chr>	AvgAcc <dbl>
False	0.9049919
True	0.9085714
NA	0.9191423

Model has the same average accuracy for those reviews where the value of DriveThru is True or False.

Accuracy group by Attribute Price Range –

RestaurantsPriceRange2 <chr>	AvgAcc <dbl>
1	0.9134877
2	0.9207101
3	0.9538462
4	0.8965517
NA	0.9075938

Model accuracy is the highest for price range 3 restaurants and lowest for price range 4 restaurants.

Accuracy group by Attribute Upscale –

Upscale <chr>	AvgAcc <dbl>
False	0.9187162
True	0.9489796
NA	0.9075938

Our model predicts true positives more accurately for attribute upscale than it does true negatives.