ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

ASSIGNMENT 3

Analyzing text in Yelp reviews – Text mining, Sentiment Analyses

1. Explore the data.

(a) How are star ratings distributed? How will you use the star ratings to obtain a label indicating 'positive' or 'negative' – explain using the data, summaries, graphs, etc.? Do star ratings have any relation to 'funny', 'cool', 'useful'? Is this what you expected? (b) How does star ratings for reviews relate to the star-rating given in the dataset for businesses (attribute 'businessStars')? (Can one be calculated from the other?)

Ans:

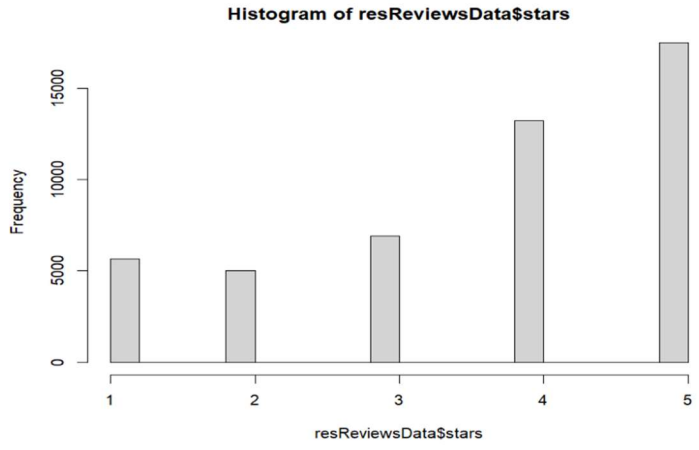
Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations.

Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral. Here in the yelp review dataset, we would be considering the reviews for restaurants. The business ID corresponds to restaurants. Each business has multiple reviews and there is a review ID for each review. Each review has a star rating from 1-5. In the dataset when we group the reviews by stars, the number of positive reviews is more than the negative reviews as seen in the distribution below (Table 1.1). The positive reviews accounts to 17461 whereas the highest negative reviews accounts to 6916.

stars <int>	n <int>
1	5665
2	4995
3	6916
4	13209
5	17461

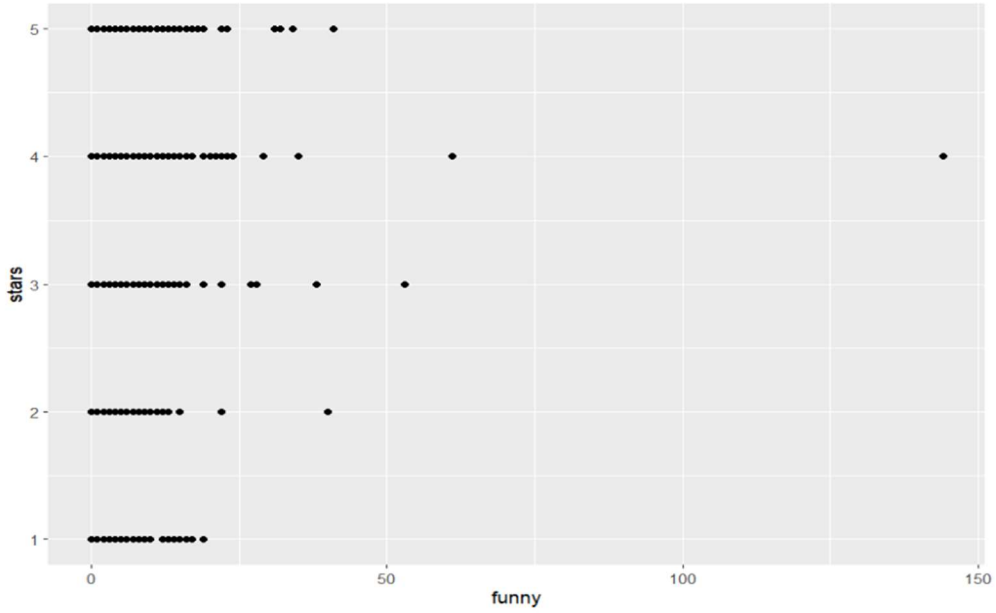
Table 1.1: Star ratings with counts of words

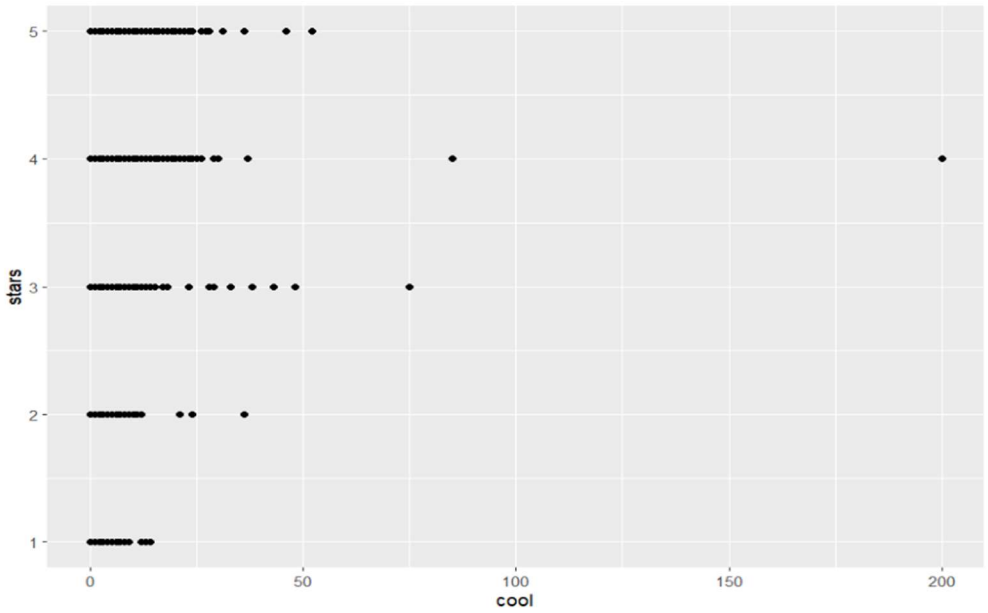
The graph below shows the distribution of the star ratings. It is evident that there are more positive ratings as the bars of the higher ratings increases.



ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

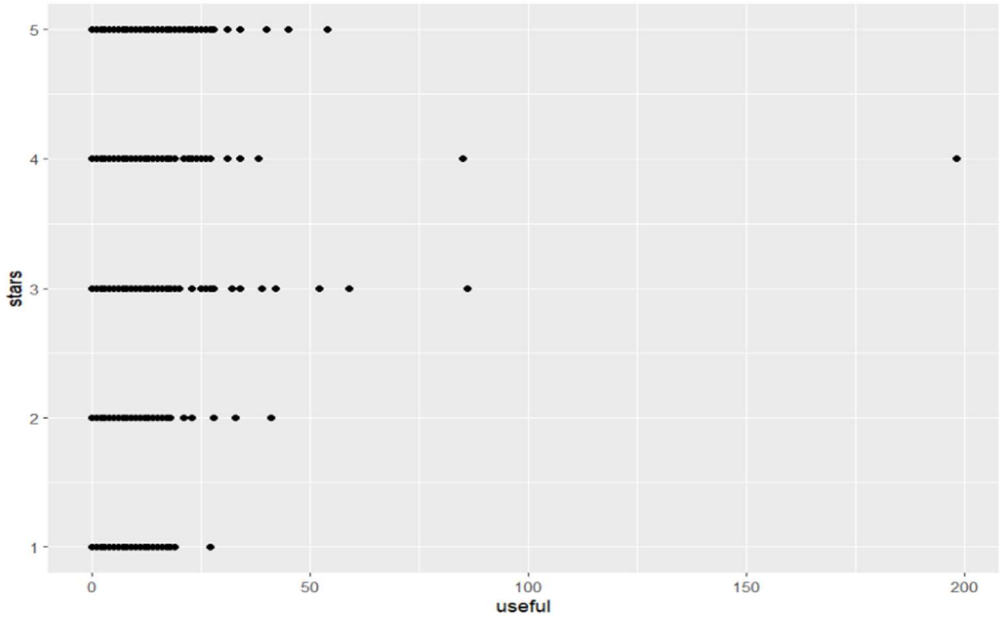
It can be seen from the below graph that as the star rating increases from 1 to 5 the number of “funny” reviews also increases. Star rating = 4 has the maximum number of “funny” reviews. But “funny” reviews are also associated with star rating =1.





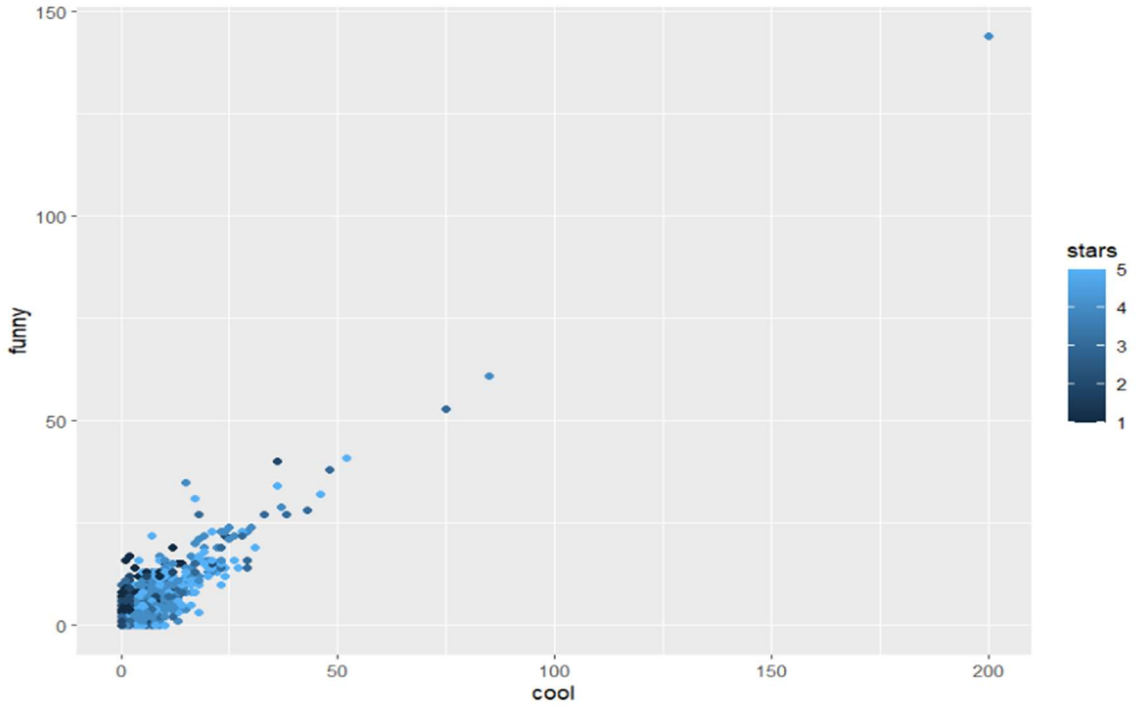
From the above graph as the star rating increases from 1 to 5 the number of cool reviews changes. Star rating = 4 has the maximum number of “cool” reviews. While star rating = 1 has the least “cool” reviews.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

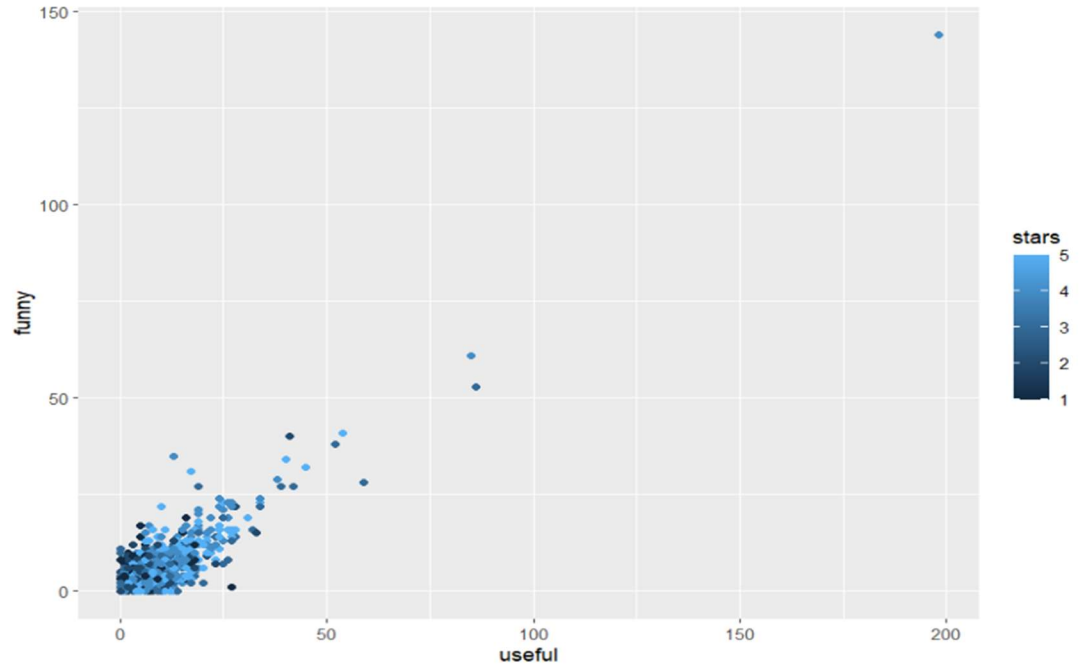


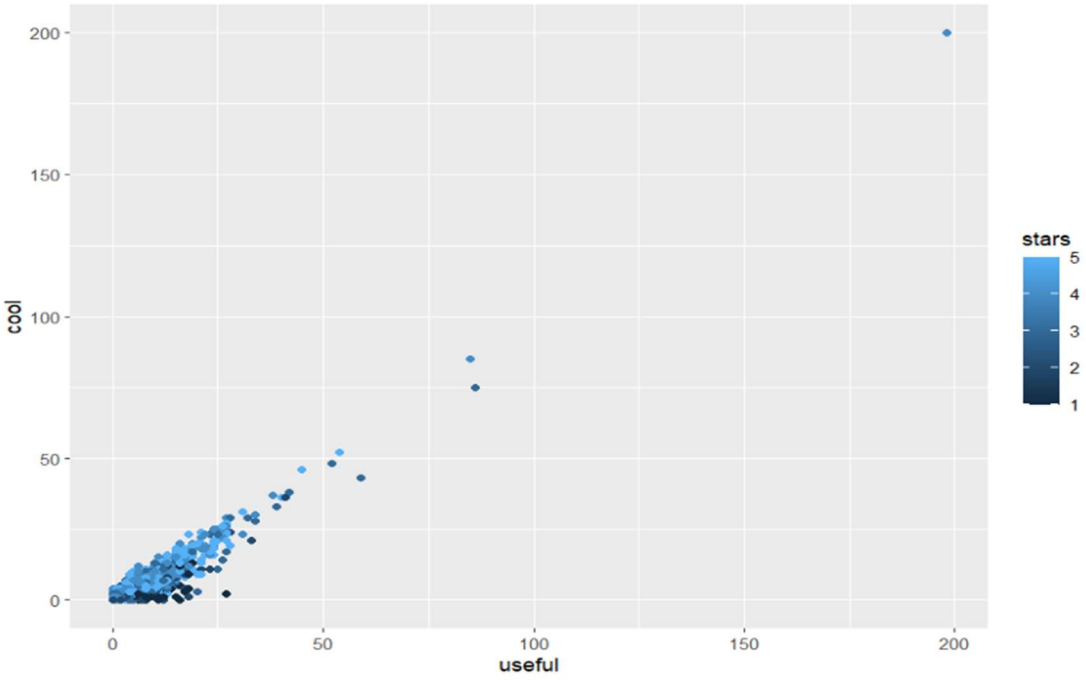
Star ratings 3, 4, 5 have the maximum number of “Useful” reviews from the above graph. Star rating 1 has very few reviews with the term “useful”.

There are clear relationships between “Funny” vs “Cool”, “Funny” vs “Useful” and “Cool” vs “Useful” which can be seen from the graphs below, they are all linearly correlated with each other, and the maximum can be found of 5 star ratings.



ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)





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

Ans:

There are multiple reviews for 1 business/restaurant and each review has a star rating associated to it. So, if we average the star rating of the reviews for a particular business, we get the starsBusiness column. This is evident from the below table 1.3 round(avgStars,1) value.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

business_id <chr>	review_id <chr>	stars <int>	starsBusiness <dbl>
DAMQ2VtdwZsVed8BC3Z9rA	nCe9ydT6SSUG2HwbOwnVTA	5	4.0
DAMQ2VtdwZsVed8BC3Z9rA	zMW4nH0DoyQTPQ4CJqpZoQ	5	4.0
DAMQ2VtdwZsVed8BC3Z9rA	YFFNp6QX4F206IOlwjlqqQ	4	4.0
DAMQ2VtdwZsVed8BC3Z9rA	NUHX_7M08-Agnz2iz1pxvw	4	4.0
-Y9woV5m-KaTzu9VpirYyg	8SYCrtHtNioIL0_UeGb1xw	5	3.5
-Y9woV5m-KaTzu9VpirYyg	-fjXwcKBk37iF_fFNayCZg	4	3.5
-Y9woV5m-KaTzu9VpirYyg	p1X-3vxkjpuQ6cxv-ccQTw	5	3.5
-Y9woV5m-KaTzu9VpirYyg	w6kp2NZTgGFKBUL2GaK06g	1	3.5
-Y9woV5m-KaTzu9VpirYyg	Y_hpI3MiODb4YzTj__btZg	5	3.5
-Y9woV5m-KaTzu9VpirYyg	9pb-FmDNplw2aKC1gHm-kQ	3	3.5
-Y9woV5m-KaTzu9VpirYyg	drDnE_2wTj9Q3oqL-E14bg	2	3.5
-Y9woV5m-KaTzu9VpirYyg	pyPCe8441YsXXb7QI80O2w	2	3.5
-Y9woV5m-KaTzu9VpirYyg	pMKlgCQlQT7UD7EP_aQ3eA	4	3.5
-Y9woV5m-KaTzu9VpirYyg	M5kQEqj9geWF3fTIc0Wc_g	3	3.5
-Y9woV5m-KaTzu9VpirYyg	aB4LhTFNCEPgQ0gjhcg49g	4	3.5
-Y9woV5m-KaTzu9VpirYyg	wONjPsZ_8qVC6cu2Ikb-0w	4	3.5
-Y9woV5m-KaTzu9VpirYyg	Pv6Q-pnDfGuZV6YA-ZLV0Q	4	3.5
-Y9woV5m-KaTzu9VpirYyg	Yi7X9QyJMjnAzZ88B8h5QQ	3	3.5
-Y9woV5m-KaTzu9VpirYyg	u15s-Z-hR580BJ8Fw2VV9w	4	3.5
-Y9woV5m-KaTzu9VpirYyg	pcP0acfHpWdE62an_VnjmQ	4	3.5
-Y9woV5m-KaTzu9VpirYyg	RmcDSw5Rr1pwMVAc9Eq-nA	4	3.5
-Y9woV5m-KaTzu9VpirYyg	WAbcwtq6HdYGAgXSoBC1Lg	3	3.5
-Y9woV5m-KaTzu9VpirYyg	n1DdcwE5klX-jRx-ZPZ7xQ	3	3.5
-Y9woV5m-KaTzu9VpirYyg	qpvhoX55qoUN4O31d15Xgg	5	3.5
-Y9woV5m-KaTzu9VpirYyg	9213S_JIngrscqXFmleydw	5	3.5
-Y9woV5m-KaTzu9VpirYyg	rQkD-N8ie8z2y28up-HrlQ	2	3.5
-Y9woV5m-KaTzu9VpirYyg	iNbUp4el78LgqaGpuK9EKA	4	3.5
-Y9woV5m-KaTzu9VpirYyg	Y9kV6l3kJ1EbO7dwupp7Jg	3	3.5
-Y9woV5m-KaTzu9VpirYyg	sVn2vcQio8o7YoYvY20Hpw	4	3.5
-Y9woV5m-KaTzu9VpirYyg	NikUgoCh6u3Do2yLdSP1uQ	5	3.5
-Y9woV5m-KaTzu9VpirYyg	vY564VpjLC-fDvgrpMJFKg	4	3.5
-Y9woV5m-KaTzu9VpirYyg	x19-Gtp7COy5yxhmJFCsNA	3	3.5

97-128 of 48,246 rows Previous 1

Table 1.2: Average star rating per review

business_id <chr>	starsBusiness <dbl>	round(avgStars, 1) <dbl>
--FBCX-N37CMYDfs790Bnw	4.0	3.9
-865Ps6xb3h1LP67JcQ3mA	3.0	3.1
-9yC5SmYxH8BLg4bML8VQg	4.0	4.0
-guxo51AuUa_J8RuJ70EWg	3.5	3.5
-ITj6Pu8Gdw8MmLf0X8EKQ	4.0	3.8
-K_qCSp7q61BkkjP_FSsEQ	3.0	3.2
-K3kqmykKlhlB4arCsLHOw	3.0	2.8
-IJtyCOTVInWusU9YF120A	3.5	3.4
-OEIW0dO96-492qa_luxaw	4.0	4.1
-pGO9M4JLQJHiaCOrqCgbQ	3.0	2.8
-sjCxkxv6xU5rEVLFybAuA	3.5	3.5
-tcJmqzfaeEnpFMAeIB7bA	3.5	3.7
-Ut87cwCFsO3444Rd11p0Q	3.5	3.5
-Y9woV5m-KaTzu9VpirYyg	3.5	3.5
-YfDhpfSesaoEPWWJvIKQg	3.5	3.5
-YGePLsJ2pYccR3oaeCSAw	2.5	2.7
__zA29wBC0LleSxMzNHpwQ	4.0	4.0
_83jnaQVtDXqydea1y92fQ	4.0	3.8
_DOjM_VCK-ojgjoY6esOdg	3.5	3.5
_j750dvyAFCkd0OzPOV09Q	3.0	2.9
_qA2HPqgLi8c-M4wRyml0g	4.5	4.4
_Qund-w4Ol9StNm8P-rk1w	4.5	4.3
_uMRu3765QYo0Qg1z1LKqw	4.5	4.4
01Ov9eDxKRY5k6ImMdiWLQ	3.0	3.1
0B_aEPeZjrXs6T7vS3arrA	4.0	3.8
0fyxLMASE5bfYAgW6MYM_g	2.5	2.5
0k9yCNjHyXaniwK5pfe1iA	3.5	3.4
0u8sXOFXBlscojMipiumeA	2.5	2.7
0ZZlqTdEb7yUa13QZLKuGQ	4.0	4.0
13-n6Cm0AK0QK28iatttWQ	4.5	4.4

Table 1.3: Round off average star rating

Here if see both the tables, starbusiness is the round off version of avgStars this is done to ease out the analysis.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

2) 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).

Ans: In order to perform the positive and negative sentiment analysis, we have used the bag of words approach of text mining which involves steps like tokenize, data transformation and filtering, stemming /lemmatization.

The libraries used to perform the analysis below are tidytext, SnowballC, and textsystem. The first step tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords.

We have tokenized the text of the reviews from the column named 'text' and kept only the review ID and stars columns giving us a data set of 5323888 x 3. We have further removed the stop words. After removing the stop words, we get 47229 distinct words. So, we can see a difference of around 700 words being eliminated after removing the stop words.

Further we have counted the total occurrences of different words and sorted them based on maximum occurrence which is shown in Table 2.1. We have also removed the words which do not occur in at least 10 reviews as these could be considered rare along with the terms with digits in them. We are now left with 10121 distinct tokens.

	word	n
1	food	39253
2	service	19011
3	time	14976
4	restaurant	10725
5	chicken	10581
6	nice	9349
7	menu	9329
8	love	8668
9	delicious	8510
10	sauce	7363
11	bar	7343
12	friendly	7278
13	cheese	7202
14	pretty	6954
15	lunch	6945
16	eat	6829
17	fresh	6724
18	staff	6723
19	salad	6673
20	pizza	5949
21	people	5904
22	meal	5806
23	amazing	5733
24	table	5614

Table 2.1: Words with their frequency of occurrence

The next step we have done is stemming/lemmatization. Stemming and Lemmatization are text normalization techniques that are used to prepare text, words, and documents for further processing.

Stemming is the process of getting reduced forms/root word of the actual word. Stemming chops off letters from the end until the root/base word is reached. It produces word stems which are not actual words.

review_id <chr>	stars <int>	word <chr>	word_stem <chr>
wZcut40vGTumt2Q2_wAmwQ	1	yesterday	yesterdai
wZcut40vGTumt2Q2_wAmwQ	1	buying	bui
wZcut40vGTumt2Q2_wAmwQ	1	deal	deal
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	kids	kid
wZcut40vGTumt2Q2_wAmwQ	1	found	found
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	worth	worth
wZcut40vGTumt2Q2_wAmwQ	1	buying	bui
wZcut40vGTumt2Q2_wAmwQ	1	people	peopl
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	valid	valid
wZcut40vGTumt2Q2_wAmwQ	1	kids	kid
wZcut40vGTumt2Q2_wAmwQ	1	eat	eat
wZcut40vGTumt2Q2_wAmwQ	1	cents	cent
wZcut40vGTumt2Q2_wAmwQ	1	paying	pai
wZcut40vGTumt2Q2_wAmwQ	1	adult	adult
wZcut40vGTumt2Q2_wAmwQ	1	mention	mention
wZcut40vGTumt2Q2_wAmwQ	1	food	food
wZcut40vGTumt2Q2_wAmwQ	1	priced	price
wZcut40vGTumt2Q2_wAmwQ	1	quality	qualiti
wZcut40vGTumt2Q2_wAmwQ	1	eat	eat
wZcut40vGTumt2Q2_wAmwQ	1	fish	fish
wZcut40vGTumt2Q2_wAmwQ	1	fridays	fridai
wZcut40vGTumt2Q2_wAmwQ	1	roughly	roughli
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	waitresses	waitress
wZcut40vGTumt2Q2_wAmwQ	1	rude	rude
wZcut40vGTumt2Q2_wAmwQ	1	told	told
wZcut40vGTumt2Q2_wAmwQ	1	waiting	wait
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon

1-32 of 1,758,877 rows

Table 2.2: Words obtained after Stemming

In contrast to stemming, lemmatization keep the context of the word intact. They produce lemma of the words which are meaningful. Lemmatization is more informative than stemming as it provides meaningful words which can be further used with sentiment dictionaries for analysis. Therefore, we have used lemmatization in our analysis.

review_id <chr>	stars <int>	word <chr>	word_lemma <chr>
wZcut40vGTumt2Q2_wAmwQ	1	yesterday	yesterday
wZcut40vGTumt2Q2_wAmwQ	1	buying	buy
wZcut40vGTumt2Q2_wAmwQ	1	deal	deal
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	kids	kid
wZcut40vGTumt2Q2_wAmwQ	1	found	find
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	worth	worth
wZcut40vGTumt2Q2_wAmwQ	1	buying	buy
wZcut40vGTumt2Q2_wAmwQ	1	people	people
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	valid	valid
wZcut40vGTumt2Q2_wAmwQ	1	kids	kid
wZcut40vGTumt2Q2_wAmwQ	1	eat	eat
wZcut40vGTumt2Q2_wAmwQ	1	cents	cent
wZcut40vGTumt2Q2_wAmwQ	1	paying	pay
wZcut40vGTumt2Q2_wAmwQ	1	adult	adult
wZcut40vGTumt2Q2_wAmwQ	1	mention	mention
wZcut40vGTumt2Q2_wAmwQ	1	food	food
wZcut40vGTumt2Q2_wAmwQ	1	priced	price
wZcut40vGTumt2Q2_wAmwQ	1	quality	quality
wZcut40vGTumt2Q2_wAmwQ	1	eat	eat
wZcut40vGTumt2Q2_wAmwQ	1	fish	fish
wZcut40vGTumt2Q2_wAmwQ	1	fridays	friday
wZcut40vGTumt2Q2_wAmwQ	1	roughly	roughly
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon
wZcut40vGTumt2Q2_wAmwQ	1	waitresses	waitress
wZcut40vGTumt2Q2_wAmwQ	1	rude	rude
wZcut40vGTumt2Q2_wAmwQ	1	told	tell
wZcut40vGTumt2Q2_wAmwQ	1	waiting	wait
wZcut40vGTumt2Q2_wAmwQ	1	groupon	groupon

1-32 of 1,758,877 rows

Table 2.3: Words obtained after Lemmatization

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

We have used the bind_tf_idf function to obtain the term frequency ie the number of times a term occurs in a document, inverse document frequency ie inverse of the number of words in a document and the tf_idf which is the term frequency inverse document frequency of the words in the document (Table2.4).

review_id <chr>	stars <int>	word <chr>	n <int>	tf <dbl>	idf <dbl>	tf_idf <dbl>
--8lBjSMyKeiV1hkxBebsw	2	convenient	1	0.100000000	5.0282223	0.502822235
--8lBjSMyKeiV1hkxBebsw	2	disappoint	1	0.100000000	2.6654595	0.266545949
--8lBjSMyKeiV1hkxBebsw	2	fishy	1	0.100000000	6.0654657	0.606546569
--8lBjSMyKeiV1hkxBebsw	2	live	1	0.100000000	3.1467302	0.314673017
--8lBjSMyKeiV1hkxBebsw	2	price	1	0.100000000	1.8318769	0.183187692
--8lBjSMyKeiV1hkxBebsw	2	smell	1	0.100000000	4.2993293	0.429932933
--8lBjSMyKeiV1hkxBebsw	2	sorely	1	0.100000000	7.1730466	0.717304665
--8lBjSMyKeiV1hkxBebsw	2	sushi	2	0.200000000	3.1671888	0.633437750
--8lBjSMyKeiV1hkxBebsw	2	taste	1	0.100000000	1.9793393	0.197933933
--AJNR61XcHGY5nVUY2GBg	3	food	2	0.142857143	0.6381585	0.091165497
--AJNR61XcHGY5nVUY2GBg	3	hostess	1	0.071428571	4.0063180	0.286165569
--AJNR61XcHGY5nVUY2GBg	3	price	2	0.142857143	1.8318769	0.261696702
--AJNR61XcHGY5nVUY2GBg	3	special	1	0.071428571	2.5258012	0.180414371
--AJNR61XcHGY5nVUY2GBg	3	speed	1	0.071428571	6.1300042	0.437857444
--AJNR61XcHGY5nVUY2GBg	3	stand	1	0.071428571	3.7781755	0.269869682
--AJNR61XcHGY5nVUY2GBg	3	star	3	0.214285714	2.3714654	0.508171166
--AJNR61XcHGY5nVUY2GBg	3	talk	1	0.071428571	3.6130761	0.258076863
--AJNR61XcHGY5nVUY2GBg	3	taste	1	0.071428571	1.9793393	0.141381381
--AJNR61XcHGY5nVUY2GBg	3	waitress	1	0.071428571	2.8678865	0.204849033
--bRqe3_4P1G3eWOKeBYpQ	3	acceptable	1	0.034482759	5.8713097	0.202458954
--bRqe3_4P1G3eWOKeBYpQ	3	bad	1	0.034482759	2.2344978	0.077051649
--bRqe3_4P1G3eWOKeBYpQ	3	bbq	1	0.034482759	3.3639846	0.115999470
--bRqe3_4P1G3eWOKeBYpQ	3	business	1	0.034482759	3.3888570	0.116857138
--bRqe3_4P1G3eWOKeBYpQ	3	chance	1	0.034482759	4.1638914	0.143582460
--bRqe3_4P1G3eWOKeBYpQ	3	close	1	0.034482759	2.8135697	0.097019643
--bRqe3_4P1G3eWOKeBYpQ	3	decent	1	0.034482759	3.0498433	0.105167009
--bRqe3_4P1G3eWOKeBYpQ	3	dish	1	0.034482759	2.4941740	0.086005999
--bRqe3_4P1G3eWOKeBYpQ	3	drink	1	0.034482759	2.1010875	0.072451292
--bRqe3_4P1G3eWOKeBYpQ	3	food	1	0.034482759	0.6381585	0.022005465
--bRqe3_4P1G3eWOKeBYpQ	3	hour	1	0.034482759	2.6604067	0.091738163
--bRqe3_4P1G3eWOKeBYpQ	3	late	2	0.068965517	3.8188842	0.263371325
--bRqe3_4P1G3eWOKeBYpQ	3	lot	1	0.034482759	2.4283496	0.083736192

1-32 of 1,458,091 rows

Previous 1 2 3 4 5 6 ... 32 Next

Table 2.4: tf, idf and tf-idf calculations

The table 2.5 shows the words by star ratings of reviews. The word “Food” occurs most of the times and mostly in 5-star reviews. Similarly, the word ‘service’ appears in 5-star reviews along with the word ‘love’, ‘time’.

stars <int>	word <chr>	n <int>
5	food	8881
4	food	6492
5	service	5607
5	love	4545
5	time	4392
4	service	4149
5	delicious	3960
3	food	3873
4	time	3751
5	friendly	3322
5	amaze	3245
1	food	3188
5	restaurant	3107
2	food	3050
5	eat	2994
4	love	2811
4	nice	2632
4	eat	2563
5	fresh	2563
5	staff	2533
5	nice	2506
4	delicious	2474
4	price	2474
3	service	2444
4	restaurant	2438
5	price	2387
4	menu	2379
5	menu	2337
5	chicken	2302
4	friendly	2247
5	recommend	2177
5	favorite	2131

Table 2.5: Words by star ratings of review

Let's look at the word "love" for example, it occurs in all the star ratings from 1 to 5. Though maximum frequency is observed to be in star rating 5 with a proportion of 0.01 but the word "love" also occurs in negative or lower star rating that might be because of word formations like "don't love", "No love" etc.

stars <int>	word <chr>	n <int>	prop <dbl>
5	love	4545	0.010083843
4	love	2811	0.006643898
3	love	943	0.003989356
2	love	523	0.003003193
1	love	360	0.002071966

Table 2.6: Proportion of word "LOVE" for different star ratings

Below are the plots showing the most used words by star rating and top 20 words by star ratings.

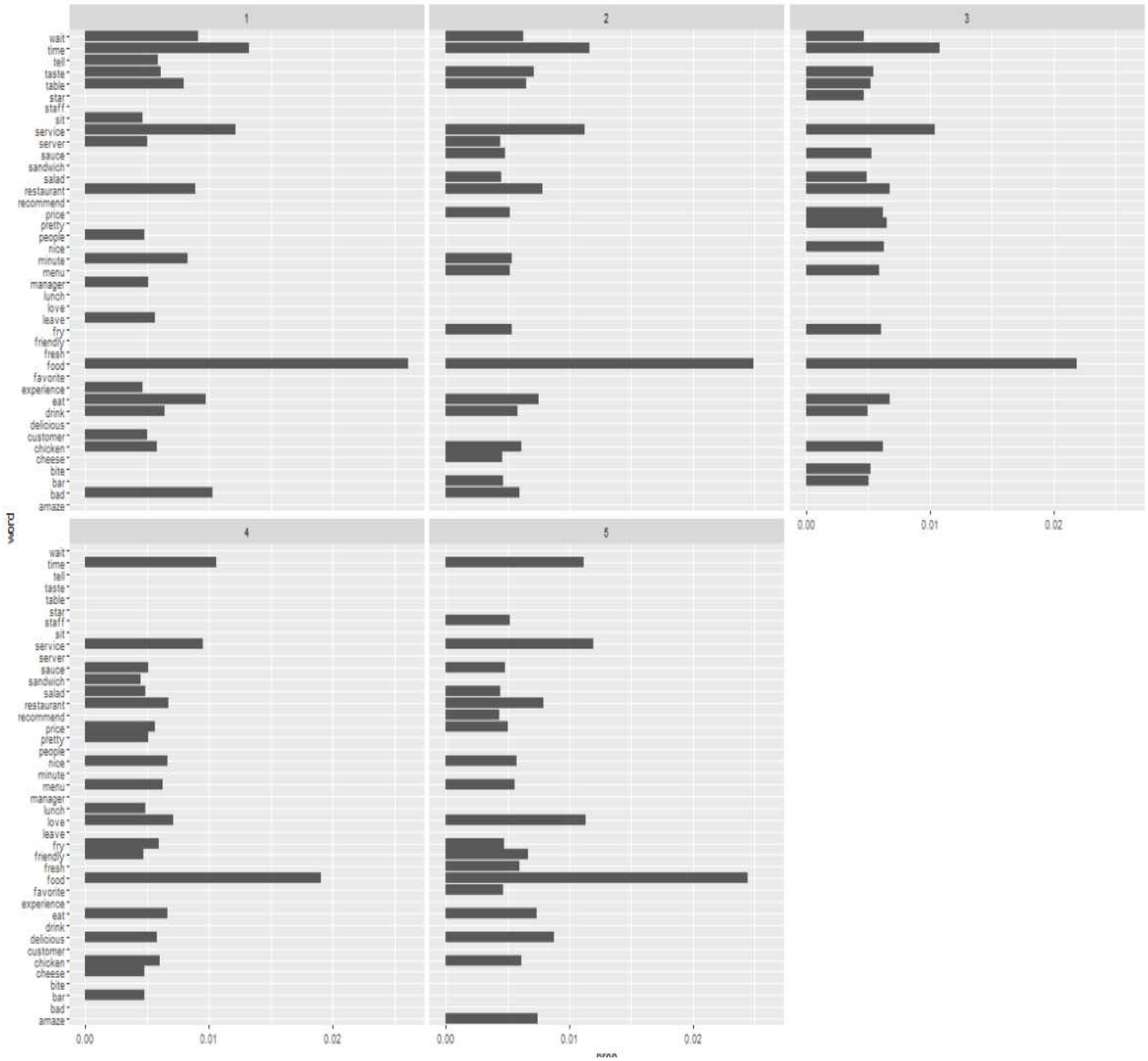


Table 2.7: Top 20 words by star ratings

From the above figure, we that some words like “FOOD”, “TIME”, “RESTAURANT”, “SERVICE” are commonly used words, and these don’t tell us more about the sentiment behind the reviews. Therefore, we are removing these common words from our analysis.

After removing we get the plot like shown in table 2.8. In this plot, let’s look at the word “BAD”, it is evident that word ‘BAD’ indicates a negative word as it has maximum reviews with star rating 1 and with star rating 5 there are no reviews for this word. So, by default this word can be considered as a negative sentiment. Words like worst, wait, manager, bad appear in lower star ratings

Similarly considering the word “FRIENDLY”, we can see that this word has no reviews for star rating 1, while it has some reviews for star ratings 2, 3 and 4. It is clearly evident that with star rating 5 the word “FRIENDLY” has highest reviews. So, this word can be considered as a positive sentiment. Words like love, nice, pretty, friendly, fresh, delicious, awesome and amazing appear in higher star ratings.

Table 2.8: Top 20 words by star ratings after removing common words

To find the words which are associated with lower or higher star ratings in general, we have calculated the average star rating associated with each word. Below table shows the top 20 words with highest star ratings.

word <chr>	totWS <dbl>
chicken	0.07194397
delicious	0.07829971
drink	0.06003276
eat	0.09810047
food	0.26242642
fresh	0.06001220
friendly	0.07411651
fry	0.06295419
love	0.09704123
menu	0.07715634
nice	0.08245675
pretty	0.05859133
price	0.08107960
restaurant	0.09572898
sauce	0.05959733
service	0.16495921
staff	0.06368366
taste	0.06855400
time	0.13914349
wait	0.06280929

Table 2.9: Top 20 words with highest star ratings

Below table shows the top 20 words with lower star ratings. In the context of restaurant reviews, the words displayed here makes sense in terms of the star ratings for higher and lower.

word <chr>	totWS <dbl>
acknowledgment	6.225568e-05
boyardee	6.583774e-05
cuervo	6.730754e-05
defensive	6.450758e-05
disgrace	6.500967e-05
drawer	6.332122e-05
filter	6.478161e-05
filth	5.988297e-05
hagar	5.178526e-05
inexcusable	6.106932e-05
inspector	6.359526e-05
johnny's	6.597738e-05
livid	6.611177e-05
rethink	6.703351e-05
robbery	6.583774e-05
scold	6.207532e-05
tong	6.074516e-05
uncaring	6.813561e-05
violent	6.588371e-05
violently	6.846919e-05

Table 2.10: Top 20 words with lowest star ratings

3) We will consider three dictionaries, available through the tidytext package – (i) the NRC dictionary of terms denoting different sentiments, (ii) the extended sentiment lexicon developed by Prof Bing Liu, and (iii) 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.

(a) How many matching terms (i.e. terms in your data which match the dictionary terms) are there for each of the dictionaries?

Ans:

We have performed sentiment analysis using the 3 sentiment dictionaries available with textdata package. The three sentiment dictionaries are as follows:
1) BING
2) NRC
3) AFINN

BING DICTIONARY: The Bing dictionary categorizes words into positive and negative categories.

NRC DICTIONARY: The NRC dictionary categorizes words into positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

AFINN DICTIONARY: The AFINN dictionary assigns words with a score from -5 to 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

Here we have used the function get_sentiments() which allows us to get specific sentiment lexicons with the appropriate measures for each one.

sentiment <chr>	n <int>
negative	92151
positive	180307
NA	1185633

3 rows

Table 3.1: Bing dictionary

The above table for BING dictionary only provides sentiments for those words that are present in the dictionary (positive or negative or NA). Positive sentiment has 180307 terms while negative has 92151 number of terms in the data which match the dictionary terms.

sentiment <chr>	n <int>
anger	46328
anticipation	115812
disgust	36401
fear	44960
joy	151191
negative	110231
positive	280267
sadness	44232
surprise	52342
trust	153427
NA	1001397

11 rows

Table 3.2: NRC dictionary

The table 3.2 for NRC dictionary gives the number of matching terms for each sentiment from the dictionary. It gives different sentiment words for e.g word “disappoint” is given the sentiment anger/disgust/negative/sadness. Here total tokens from our dataset have only 1,458,067 rows, but after

joining the words from the NRC dictionary, the data set has 2,036,570 rows. This indicates that NRC dictionary gives multiple sentiments for a single word.

value <dbl>	n <int>
-5	73
-4	1398
-3	13598
-2	26949
-1	23508
1	31139
2	62470
3	44937
4	11473
5	1060
NA	1241486

11 rows

Table 3.3: AFINN dictionary listings of all -5 to +5 sentiments with word counts

The table 3.3 for AFINN dictionary gives a -5 to 5 sentiment rating to all the words from the dataset and above displayed is the count of words for each sentiment. The -1 to -5 ratings are considered negative and 1 to 5 are considered positive. Majority of the sentiments belong to value 2 which is closer to a positive sentiment.

(b) What is the overlap in matching terms between the different dictionaries? Based on this, do you think any of the three dictionaries will be better at picking up sentiment information from you text of reviews?

Ans:

Considering for Bing dictionary, we have done an inner join to retain those words which match the sentiment dictionary. The positive words counts to 180307 while negative words counts to 92151(table 3.4 and table 3.5).

review_id <chr>	stars <int>	word <chr>	n <int>	tf <dbl>	idf <dbl>	tf_idf <dbl>	sentiment <chr>
--8lBjSMyKeiV1hkxBebsw	2	convenient	1	0.100000000	5.028222	0.502822235	positive
--8lBjSMyKeiV1hkxBebsw	2	disappoint	1	0.100000000	2.665459	0.266545949	negative
--8lBjSMyKeiV1hkxBebsw	2	smell	1	0.100000000	4.299329	0.429932933	negative
--8lBjSMyKeiV1hkxBebsw	2	sorely	1	0.100000000	7.173047	0.717304665	negative
--bRqe3_4P1G3eWOKeBYpQ	3	bad	1	0.034482759	2.234498	0.077051649	negative
--bRqe3_4P1G3eWOKeBYpQ	3	decent	1	0.034482759	3.049843	0.105167009	positive
--bRqe3_4P1G3eWOKeBYpQ	3	perfectly	1	0.034482759	3.519934	0.121377049	positive
--bRqe3_4P1G3eWOKeBYpQ	3	pretty	1	0.034482759	2.166926	0.074721587	positive
--bRqe3_4P1G3eWOKeBYpQ	3	suspect	1	0.034482759	6.219616	0.214469530	negative
--CKvYCwEJLH7bsNWR2H2A	5	crisp	1	0.041666667	4.618547	0.192439446	positive
--CKvYCwEJLH7bsNWR2H2A	5	delicious	1	0.041666667	1.877978	0.078249075	positive
--CKvYCwEJLH7bsNWR2H2A	5	excellent	1	0.041666667	2.625162	0.109381753	positive
--CKvYCwEJLH7bsNWR2H2A	5	fresh	2	0.083333333	2.137851	0.178154216	positive
--CKvYCwEJLH7bsNWR2H2A	5	friendly	1	0.041666667	1.932015	0.080500620	positive
--CKvYCwEJLH7bsNWR2H2A	5	nice	1	0.041666667	1.821060	0.075877518	positive
--F3lDF1PBm7KPEE-br0Tg	2	reasonable	1	0.055555556	3.610006	0.200555902	positive
--F3lDF1PBm7KPEE-br0Tg	2	slow	1	0.055555556	3.386403	0.188133501	negative
--omDbUZ7UIk-QeCoilkNQ	2	bland	2	0.012738854	3.620018	0.046114877	negative
--omDbUZ7UIk-QeCoilkNQ	2	disappoint	2	0.012738854	2.665459	0.033954898	negative
--omDbUZ7UIk-QeCoilkNQ	2	dissatisfy	1	0.006369427	7.200446	0.045862711	negative
--omDbUZ7UIk-QeCoilkNQ	2	distinctive	1	0.006369427	7.839526	0.049933284	positive
--omDbUZ7UIk-QeCoilkNQ	2	enjoyable	1	0.006369427	4.963882	0.031617080	positive
--omDbUZ7UIk-QeCoilkNQ	2	enjoyment	1	0.006369427	7.257604	0.046226777	positive
--omDbUZ7UIk-QeCoilkNQ	2	fail	1	0.006369427	5.208015	0.033172073	negative
--omDbUZ7UIk-QeCoilkNQ	2	favor	1	0.006369427	5.432106	0.034599404	positive
--omDbUZ7UIk-QeCoilkNQ	2	generously	1	0.006369427	6.852139	0.043644197	positive
--omDbUZ7UIk-QeCoilkNQ	2	lack	3	0.019108280	3.624673	0.069261261	negative
--omDbUZ7UIk-QeCoilkNQ	2	needless	1	0.006369427	5.204235	0.033147992	negative
--omDbUZ7UIk-QeCoilkNQ	2	pleasant	1	0.006369427	4.043445	0.025754428	positive
--omDbUZ7UIk-QeCoilkNQ	2	smash	1	0.006369427	6.377245	0.040619397	negative
--omDbUZ7UIk-QeCoilkNQ	2	sour	1	0.006369427	4.264817	0.027164441	negative
--omDbUZ7UIk-QeCoilkNQ	2	stainless	1	0.006369427	7.839526	0.049933284	positive

1-32 of 272,458 rows Previous 1 2 3 4 5 6 ...

Table 3.4: Words which match the sentiment dictionary

sentiment <chr>	n <int>
negative	92151
positive	180307

2 rows

Table 3.5: Counts of positive and negative sentiment words

Below table 3.5 shows the words which accounts to positive and negative sentiments with their count of appearances.

	word	sentiment	totOcc
1	bad	negative	6352
2	disappoint	negative	3584
3	hard	negative	2227
4	cold	negative	2151
5	cheap	negative	1925
6	slow	negative	1791
7	wrong	negative	1774
8	bland	negative	1476
9	miss	negative	1459
10	lack	negative	1387
11	smoke	negative	1192
12	expensive	negative	1188
13	horrible	negative	1141
14	terrible	negative	1097
15	pricey	negative	1003
16	greasy	negative	980
17	rude	negative	966
18	issue	negative	918
19	complaint	negative	906
20	loud	negative	842

	word	sentiment	totOcc
1	love	positive	11768
2	nice	positive	9615
3	delicious	positive	8510
4	friendly	positive	7358
5	pretty	positive	6954
6	fresh	positive	6887
7	amaze	positive	5847
8	enjoy	positive	5074
9	recommend	positive	4777
10	favorite	positive	4523
11	hot	positive	4332
12	excellent	positive	4036
13	sweet	positive	3898
14	top	positive	3884
15	happy	positive	3834
16	awesome	positive	3830
17	worth	positive	3397
18	super	positive	3156
19	clean	positive	3109
20	fast	positive	2938

Table 3.6: Positive and Negative sentiments with total counts

Further we have, grouped and segregated out the top 25 positive and negative sentiment words and they are as follows:

word <chr>	sentiment <chr>	totOcc <int>
love	positive	11768
nice	positive	9615
delicious	positive	8510
friendly	positive	7358
pretty	positive	6954
fresh	positive	6887
amaze	positive	5847
enjoy	positive	5074
recommend	positive	4777
favorite	positive	4523
hot	positive	4332
excellent	positive	4036
sweet	positive	3898
top	positive	3884
happy	positive	3834
awesome	positive	3830
worth	positive	3397
super	positive	3156
clean	positive	3109
fast	positive	2938
perfect	positive	2864
decent	positive	2544
free	positive	2328
fantastic	positive	1975
fun	positive	1925

25 rows

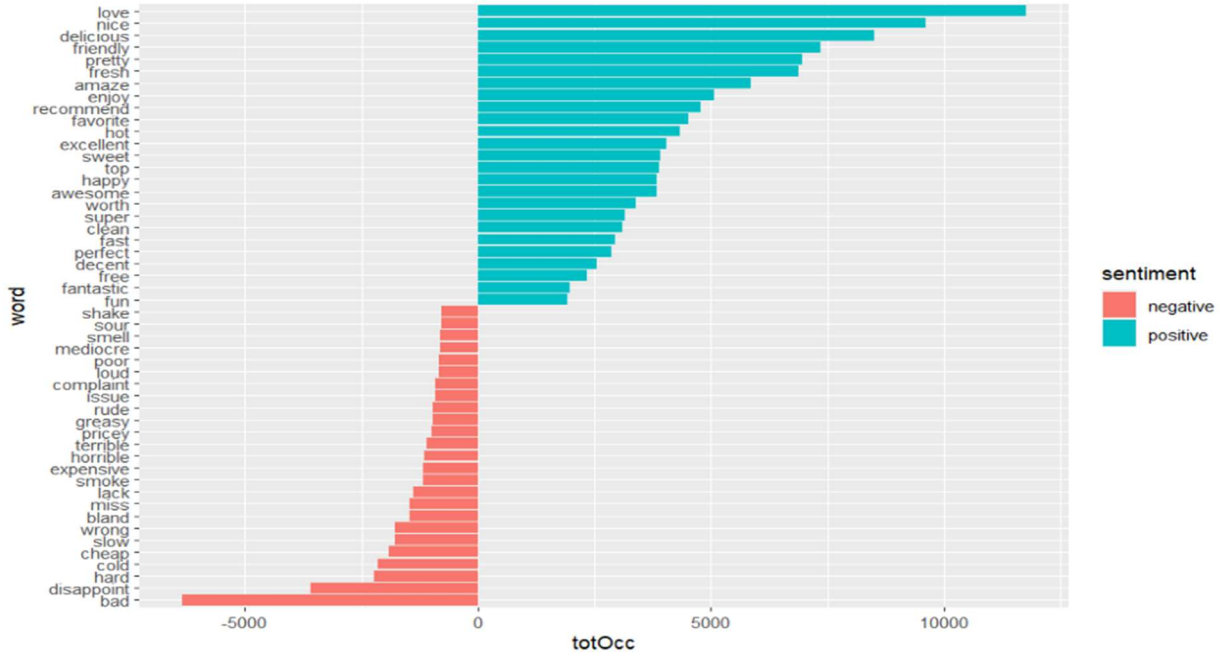
Table 3.7: Top 25 words with positive sentiments

word <chr>	sentiment <chr>	totOcc <int>
bad	negative	-6352
disappoint	negative	-3584
hard	negative	-2227
cold	negative	-2151
cheap	negative	-1925
slow	negative	-1791
wrong	negative	-1774
bland	negative	-1476
miss	negative	-1459
lack	negative	-1387
smoke	negative	-1192
expensive	negative	-1188
horrible	negative	-1141
terrible	negative	-1097
pricey	negative	-1003
greasy	negative	-980
rude	negative	-966
issue	negative	-918
complaint	negative	-906
loud	negative	-842
poor	negative	-837
mediocre	negative	-823
smell	negative	-802
sour	negative	-798
shake	negative	-797

25 rows

Table 3.8: Top 25 words with negative sentiments

Below plot shows the positive and negative words with their frequency of appearances. Words like “LOVE”, “NICE”, “FRIENDLY”, “EXCELLENT”, “PERFECT”, “FRESH”, “ENJOY” can be mapped to the sentiment of the review being positive while words like “RUDE”, “GREASY”, “TERRIBLE”, “HORRIBLE”, “DISAPPOINT” etc. can be linked to the review’s negative emotions. But some cases like word “HOT” as defined by the Bing dictionary is positive but might not necessarily be a positive sentiment can be a negative sentiment too.



Considering for NRC dictionary, the number of words coming under different sentiments categories like anger, anticipation, disgust, joy, fear are given below in the table. It can be seen that the maximum count of words come under the sentiment category “positive” with 792 while the least count is for sentiment category “surprise” with 195.

sentiment <chr>	count <int>	sumn <int>
anger	260	52545
anticipation	312	137463
disgust	234	40640
fear	274	52568
joy	293	186630
negative	670	125827
positive	792	334294
sadness	257	49390
surprise	195	58606
trust	404	184606

1-10 of 10 rows

Table 3.9: NRC dictionary sentiments categories with count of words

Below tables 3.10 and 3.11 represent the words categorized with the sentiments according to NRC dictionary and the total count of their appearances in the reviews. anger reflects a poor review but, yelp itself is an anger word and that is the name of the company. While word “MONEY”, “HIT” are not necessarily anger or negative. Anticipation is not necessarily positive in restaurant reviews but majorly could be a positive sentiment. Disgust sentiment could be negative words.

word <chr>	sentiment <chr>	totOcc <int>
bad	anger	6352
hot	anger	4332
disappoint	anger	3584
yelp	anger	1693
buffet	anger	1456
money	anger	1432
hit	anger	1233
horrible	anger	1141
terrible	anger	1097
excite	anger	1088
time	anticipation	19668
wait	anticipation	8669
friendly	anticipation	7358
pretty	anticipation	6954
star	anticipation	5613
enjoy	anticipation	5074
sweet	anticipation	3898
top	anticipation	3884
happy	anticipation	3834
expect	anticipation	3249
bad	disgust	6352
disappoint	disgust	3584
finally	disgust	1962
horrible	disgust	1141
terrible	disgust	1097
treat	disgust	1077
greasy	disgust	980
smell	disgust	802
sour	disgust	798
hate	disgust	754
chicken	fear	10602
bad	fear	6352

Table 3.10: Words categorized with the sentiments according to NRC

word <chr>	sentiment <chr>	totOcc <int>
surprise	fear	1828
yelp	fear	1693
watch	fear	1567
change	fear	1536
horrible	fear	1141
terrible	fear	1097
excite	fear	1088
treat	fear	1077
food	joy	39717
love	joy	11768
delicious	joy	8510
friendly	joy	7358
pretty	joy	6954
star	joy	5613
beer	joy	5515
friend	joy	5156
enjoy	joy	5074
special	joy	4665
wait	negative	8669
bad	negative	6352
bite	negative	6326
serve	negative	5625
leave	negative	4402
disappoint	negative	3584
cold	negative	2151
cheap	negative	1925
wrong	negative	1774
yelp	negative	1693
food	positive	39717
eat	positive	12890
love	positive	11768
delicious	positive	8510

Table 3.11: Words categorized with the sentiments according to NRC

Suppose if consider {anger, disgust, fear sadness, negative} to denote 'bad' reviews, and {positive, joy, anticipation, trust} to denote 'good' reviews. Below table represents the words with sentiment categories,

count and their good/bad scores. Table 3.12 represents top 20 words with negative values for negative words and table 3.13 represents top 20 words with positive values which supports the positive words.

word <chr>	sentiment <chr>	totOcc <int>	goodBad <dbl>
bad	disgust	6352	-6352
disappoint	disgust	3584	-3584
finally	disgust	1962	-1962
chicken	fear	10602	-10602
bad	fear	6352	-6352
surprise	fear	1828	-1828
yelp	fear	1693	-1693
wait	negative	8669	-8669
bad	negative	6352	-6352
bite	negative	6326	-6326
serve	negative	5625	-5625
leave	negative	4402	-4402
disappoint	negative	3584	-3584
cold	negative	2151	-2151
cheap	negative	1925	-1925
wrong	negative	1774	-1774
yelp	negative	1693	-1693
bad	sadness	6352	-6352
leave	sadness	4402	-4402
disappoint	sadness	3584	-3584

Table 3.12: Top 20 words with negative value

Words like “CHICKEN”, “BITE”, “SURPRISE” comes as negative although it could be related to a positive sentiment as well in the restaurant context. Similarly, words like “TIME” and “WAIT” could be related to a negative sentiment as well.

word <chr>	sentiment <chr>	totOcc <int>	goodBad <dbl>
time	anticipation	19668	19668
wait	anticipation	8669	8669
friendly	anticipation	7358	7358
pretty	anticipation	6954	6954
star	anticipation	5613	5613
food	joy	39717	39717
love	joy	11768	11768
delicious	joy	8510	8510
friendly	joy	7358	7358
pretty	joy	6954	6954
star	joy	5613	5613
food	positive	39717	39717
eat	positive	12890	12890
love	positive	11768	11768
delicious	positive	8510	8510
friendly	positive	7358	7358
pretty	positive	6954	6954
star	positive	5613	5613
food	trust	39717	39717
friendly	trust	7358	7358
pretty	trust	6954	6954
serve	trust	5625	5625
star	trust	5613	5613

Table 3.13: Top 20 words with positive values

Considering for AFINN dictionary, here in table 3.14, the avglen is nothing but the average number of words and avgSenti is the average sentiment scores according to the star ratings. The average sentiment scores for lower star ratings is low and increases with the star ratings.

stars <int>	avgLen <dbl>	avgSenti <dbl>
1	4.895860	-2.4356785
2	5.135049	0.8966397
3	4.939015	3.7594565
4	4.836806	6.4339908
5	4.389218	7.2823600

Table 3.14: Average sentiment scores for star ratings

The overlap of words between the Bing and the NRC dictionary words are as shown below:

word
<chr>
worth
complain
enjoy
wrong
bomb
friendly
delicious
overdo
hot
cold
distract
terribly
loyalty
top
pretty
impress
garbage
classic
amaze
waste
wonderful
hate
love
fun
festive
headache
excellent

Even when we look at the set of matching words from the Bing dictionary, it has a greater number of words as compared to the NRC dictionary. Afinn dictionary provides sentiment ratings to all words, so it also takes in a wider range of words into consideration for the sentiment analysis. However, since Bing has a clear segregation of words into positive and negative sentiments, we think it would be better at predicting the sentiment of the reviews for the restaurants.

4. Consider using the dictionary based positive and negative terms to predict sentiment (positive or negative based on star rating) of a review. 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?

Ans: For this analysis, we used all the three dictionaries to predict the sentiments of the reviews.

1. <u>Analysis using Bing Dictionary</u>:

Using this dictionary, we categorized all the words in a review into positive and negative sentiments. We calculated the total number of positive words (posSum) and negative words (negSum) per review. Then we calculated the proportion of positive and negative sentiments (posProp and negProp) per review and the difference of this proportion gave us the sentiment score (sentiScore) for each review. (see table 4.1 below)

review_id	stars	nwords	posSum	negSum	posProp	negProp	sentiScore	hiLo	pred_hiLo
--8lBjSMyKeiV1hkxBebsw	2	4	1	3	0.2500	0.75000	-0.50000	0	-1
--bRqe3_4P1G3eWOKeBYpQ	3	5	3	2	0.6000	0.40000	0.20000	0	1
--CKvYCwEJLH7bsNWR2H2A	5	6	6	0	1.0000	0.00000	1.00000	1	1
--F3lDF1PBm7KPEE-br0Tg	2	2	1	1	0.5000	0.50000	0.00000	0	-1
--omDbUZ7Uik-QeCoilkNQ	2	16	8	8	0.5000	0.50000	0.00000	0	-1
--QZ-v3Vq3KygmOptcLbZw	5	4	2	2	0.5000	0.50000	0.00000	1	-1
--Y9SInhiIlCwZ3UzrWBGw	1	3	0	3	0.0000	1.00000	-1.00000	-1	-1
-_2FX2Gnc50BL1sYS192ug	3	6	4	2	0.6667	0.33333	0.33333	0	1
-_8LIdfXYtq_-0LhwcgY6g	1	1	0	1	0.0000	1.00000	-1.00000	-1	-1
-_D1YBB_4zLyH7op2JXtKQ	2	3	0	3	0.0000	1.00000	-1.00000	0	-1

Table 4.1: Bing Dictionary analysis

stars	avgPos	avgNeg	avgSentiSc
1	0.3076	0.6924	-0.3848
2	0.4474	0.5526	-0.1052
3	0.6085	0.3915	0.2171
4	0.7485	0.2515	0.4970
5	0.8287	0.1713	0.6573

Table 4.2: Star ratings with sentiment scores for Bing dictionary

From Table 4.2, we can see that a higher star rating is associated with higher average positive score and lower average negative scores. Similarly, in case of lower star ratings, they have lower average positive score and higher average negative score. Since sentiment scores are the difference between positive and negative scores, similar relation is found in sentiment scores as well. Lower the ratings, lower is the sentiment score and vice versa.

For predictions, we have considered only reviews with 1 star ratings as negative (-1) and 5 star ratings as positive(1) and rest all values as 0. Therefore, we see only -1, 0 and 1 in hiLo column of the table 4.1. The 0 values are not considered in the predictions. The sentiment score (sentiScore) column is used for predicting against the hiLo column. If a sentiment is a greater than 0, it is considered positive otherwise negative. We get the below confusion matrix from this prediction.

	Predicted	
Actual	-1	1
-1	4660	820
1	1773	15237

Accuracy = **88.4%**

2. <u>Analysis using NRC Dictionary</u>:

The NRC dictionary segregates the data into a set of positive and negative sentiment words like joy, anticipation, trust, surprise, anger, disgust, fear, sadness, positive and negative. For our analysis, we have considered the words like positive, joy, anticipation, trust and surprise as positive sentiment words and the rest like negative, anger, disgust, fear and sadness as negative sentiment words. Again, based on these, we calculated the sum of positive (possum) and negative(negSum) words per review. Then based on their

proportions (posProp and negProp), calculated their difference as the sentiment score(sentiScore) for each review. (see table 4.3 below for these values)

review_id <chr>	stars <int>	nwords <int>	posSum <int>	negSum <int>	posProp <dbl>	negProp <dbl>	sentiScore <dbl>	hiLo <dbl>	pred_hiLo <dbl>
--8lBjSMyKeiV1hkxBebsw	2	10	1	9	0.1000	0.90000	-0.80000	0	-1
--AJNR61XcHGY5nVUY2GBg	3	10	10	0	1.0000	0.00000	1.00000	0	1
--bRqe3_4P1G3eWOKeBYpQ	3	31	21	10	0.6774	0.32258	0.35484	0	1
--CKvYCwEJLH7bsNWR2H2A	5	17	16	1	0.9412	0.05882	0.88235	1	1
--F3lDF1PBm7KPEE-br0Tg	2	7	5	2	0.7143	0.28571	0.42857	0	1
--ggBQTFxNl2UPVV_CQS0g	5	2	2	0	1.0000	0.00000	1.00000	1	1
--omDbUZ7Ulk-QeCoilkNQ	2	73	48	25	0.6575	0.34247	0.31507	0	1
--QZ-v3Vq3KygmOptcLbZw	5	18	16	2	0.8889	0.11111	0.77778	1	1
--vElx2x5RGTgyCU9kr8VA	4	7	6	1	0.8571	0.14286	0.71429	0	1
--Y9SInhillCwZ3UzrWBGw	1	16	8	8	0.5000	0.50000	0.00000	-1	-1

Table 4.3: NRC Dictionary analysis

stars <int>	avgPos <dbl>	avgNeg <dbl>	avgSentiSc <dbl>
1	0.5435	0.4565	0.08699
2	0.6394	0.3606	0.27890
3	0.7218	0.2782	0.44355
4	0.7892	0.2108	0.57846
5	0.8306	0.1694	0.66111

Table 4.4: Star ratings with sentiment scores for NRC dictionary

Like the Bing dictionary, from table 4.4, we see that average positive score is higher for higher rating and lower for lower ratings and average negative score is higher for lower ratings and lower for higher ratings. The average sentiment score also increases with star ratings.

For predictions, we have considered only reviews with 1-star ratings as negative (-1) and 5 star ratings as positive (1) and rest all values as 0. Therefore, we see only -1, 0 and 1 in hiLo column of the table 4.3. The 0 values are not considered in the predictions. The sentiment score (sentiScore) column is used for predicting against the hiLo column. If a sentiment is a greater than 0, it is considered positive otherwise negative. We get the below confusion matrix from this prediction.

	Predicted	
Actual	-1	1
-1	2372	3269
1	853	16461

Accuracy = **82.03%**

As compared to NRC dictionary, the Bing dictionary has better performance as their accuracy is greater than NRC dictionary. Also, since Bing dictionary has only two categories its better to analyze than NRC dictionary which has a generalized set of categories which are not context specific. For example, NRC has some words like yelp, surprise, hope, cream which are categorized into anger and surprise categories which are incorrect when considering restaurant reviews as the context.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

3. Analysis using AFINN Dictionary:

AFINN dictionary gives out a positivity score from -5 to +5 for each word in each review. It assigns 1 to 5 values to positive set of words and -5 to -1 values to negative set of words. For our analysis, we used the sum of this positivity score for all words in a review as the sentiment score (sentiSum) for each review. (see table 4.6 below)

review_id <chr>	stars <int>	nwords <int>	sentiSum <dbl>
--8lBjSMyKeiV1hkxBebsw	2	1	-2
--bRqe3_4P1G3eWOKeBYpQ	3	5	2
--CKvYCwEJLH7bsNWR2H2A	5	4	9
--omDbUZ7UIk-QeCoilkNQ	2	8	0
--QZ-v3Vq3KygmOptcLbZw	5	6	4
--vElx2x5RGTgyCU9kr8VA	4	1	1
--Y9SInhiIlCwZ3UzrWBCw	1	2	-7
-_2FX2Gnc50BL1sYS192ug	3	4	-2
-_D1YBB_4zLyH7op2JXtKQ	2	1	-2
-_hDCUEOl0LyNYLoHAE8PQ	1	5	2

Table 4.6: AFINN Dictionary analysis

stars <int>	avgLen <dbl>	avgSenti <dbl>
1	4.896	-2.4360
2	5.135	0.8966
3	4.939	3.7595
4	4.837	6.4340
5	4.389	7.2822

Table 4.7: Star ratings with sentiment scores for AFINN dictionary

From table 4.7, we can see that on an aggregate level, the average sentiment score of the AFINN dictionary do reflect the sentiments for a restaurant review as the average sentiment score increases with the star ratings. The average length is just the average number of words in set of reviews for that star rating.

For predictions, we have considered only the 1 and 5-star rating reviews. 1 star rating is considered negative with a value of -1 and 5-star ratings as positive with a value of 1 for our analysis. We are predicting these -1 and 1 star ratings using the sentiment score as 1 for sentiment scores above 0 and -1 for all the sentiment scores below 0. We get the below predictions

	Predicted	
Actual	-1	1
-1	3988	1399
1	1164	15532

Accuracy = **88.39%**

AFINN and Bing dictionaries have almost similar accuracy for the predictions. So we could say that they do capture to some extent the star ratings but using prediction models for this purpose could give us better results.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

5. 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, random forest (use ranger for faster run-times). Report on performance of the models you develop. Compare performance with that in part 4 above. Explain your findings (and is this what you expected).

(a)How do you evaluate performance? Which performance measures do you use, why?

(b)Which types of models does your team choose to develop, and why? Do you use term frequency, tfidf, or other measures, and why?

(c) 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. What is the size of the document-term matrix? Should you use stemming or lemmatization when using the dictionaries? Why?

(d) 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 or lemmatization here, and why?

Ans: For our analysis, we created the Naïve Bayes, Lasso logistic regression and random forest models to predict the review sentiments. We used the ROC curves and confusion matrix to evaluate the performance of the different models. Since we want to predict the positive and negative sentiment of the reviews, we are building the classification models and the AUC of an ROC curve is a very useful metric to evaluate or validate our classification model because it is threshold invariant. Using the confusion matrix we can calculate the accuracy of the models predictions.

Naïve Bayes is the simplest and quickest classification algorithm for text analytics. It is commonly used with large chunk of data. Random forest model has low computational cost, high predictive accuracy and they are constructed based on bootstrap sample from the data set. Lasso regression model is typically used where the dataset has multicollinearity and we want to do variable selection. The variable selection that lasso regularization performs can tell you which words are important for your analysis and this makes it a great fit for such classification problems.

Term frequency(tf) is how frequently a word occurs in a document. These words may or may not be important or may even be important for one document but not the other. It's difficult to find the importance of a word just based on term frequency. The inverse document frequency(idf) decreases the weights of commonly used words and increases the weights for rare words in a document. This when multiplied with the term frequency gives us the tf-idf measure. Tf-idf gives us the importance of a word in a document in a collection of documents. The document in our case is the reviews. The tf-idf is the used as the cell value while converting the dataset into the document term matrix.

First, we created the 3 models using each of the 3 dictionaries thereby using only the terms matching any or all of the sentiment dictionaries. Using these models, we are trying to predict the star ratings column

which we have modified to values -1 and 1. 3-star ratings have been removed and anything 2 or below is considered -1 and above 2 as 1. -1 indicating negative sentiment and 1 indicating positive sentiment.

1. Bing Dictionary:

The size of the document term matrix after applying the Bing dictionary and removing the 3 star rating reviews: 40231 x 1228

When we see the distribution of the ratings, we see there are more positive reviews then negative reviews.

hiLo <dbl>	n <int>
-1	10337
1	29894

We have taken a sample size of 20115 x 1228 and divided that into training and test dataset while building our model to reduce the runtime.

- **Random Forest Model:** We created the random forest model using the ranger function to predict the hiLo (-1 or 1) ratings from the set of words in the review.

  Variable importance from the random forest model:

	x
bad	1.423e-02
delicious	1.250e-02
horrible	1.133e-02
love	7.316e-03
terrible	7.036e-03
amaze	6.888e-03
awesome	5.707e-03
friendly	5.695e-03
bland	5.206e-03
disappoint	4.808e-03
rude	4.633e-03
favorite	4.242e-03
excellent	4.084e-03
disgust	4.067e-03
awful	3.754e-03
mediocre	3.645e-03
fresh	3.256e-03
poor	3.212e-03
perfect	3.145e-03
overpriced	2.889e-03

  If we look at the above set of words which are given higher importance according to the random forest model, they do belong to the restaurant review context very well.

  Confusion Matrix for training data:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	Predicted	
Actual	-1	1
-1	2411	231
1	99	7316

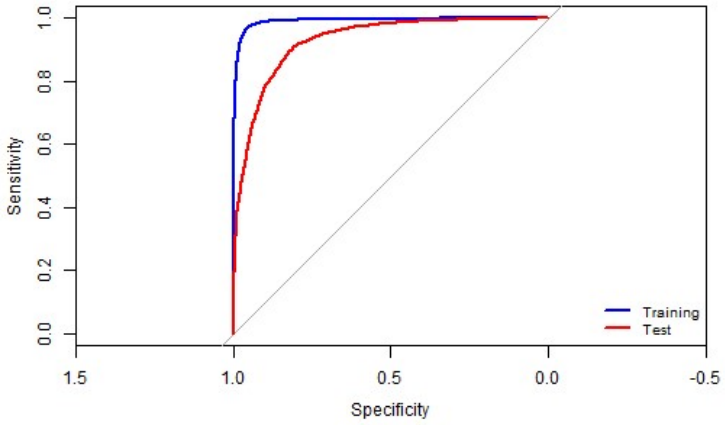
Accuracy = 96.71%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	1764	766
1	347	7181

Accuracy = 88.93%

ROC Curve:



AUC value for training data: 0.992
AUC value for test data: 0.928

By looking at the above performance measures, we can say that the text in the reviews are indicative of their star ratings and hence can be used to discern the star ratings.

- **Naïve Bayes Model:** We used the naïve bayes model to predict the hiLo(-1 or 1) ratings with the set of words in the reviews. We have used the original full dataset for this model since it has the capacity to run large chunks of data.

Confusion Matrix for training data:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

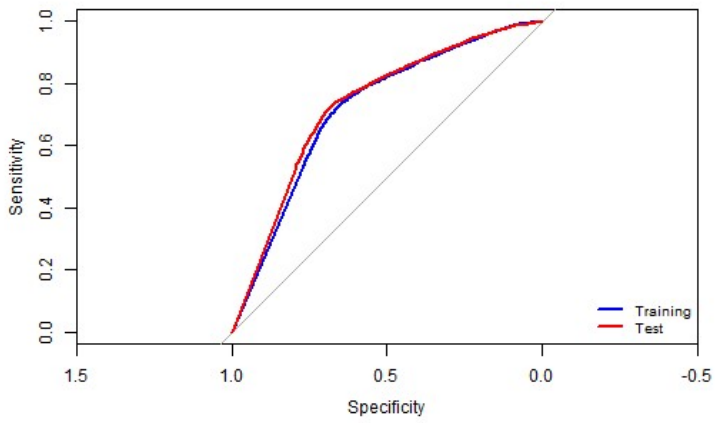
	Predicted	
Actual	-1	1
-1	3771	1423
1	5588	9333

Accuracy = 65.14%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	3846	1297
1	5619	9354

Accuracy = 65.61%
ROC Curve:

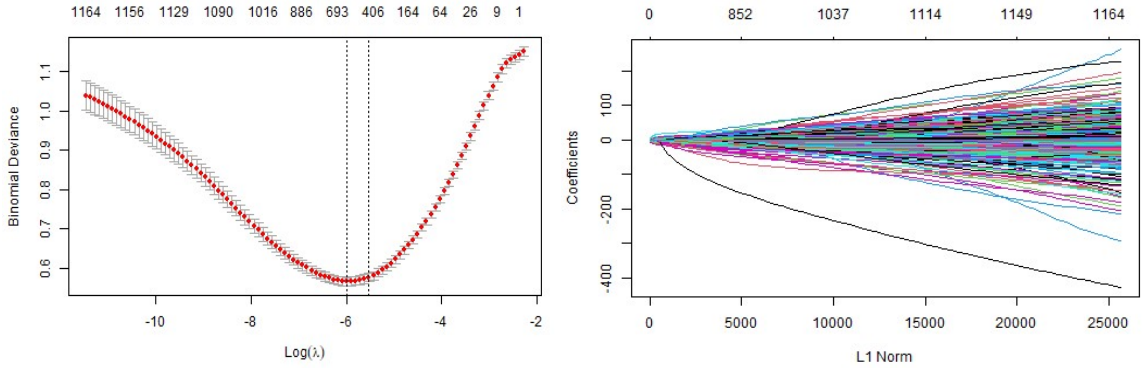


AUC value for training data: 0.7174
AUC value for test data: 0.7308

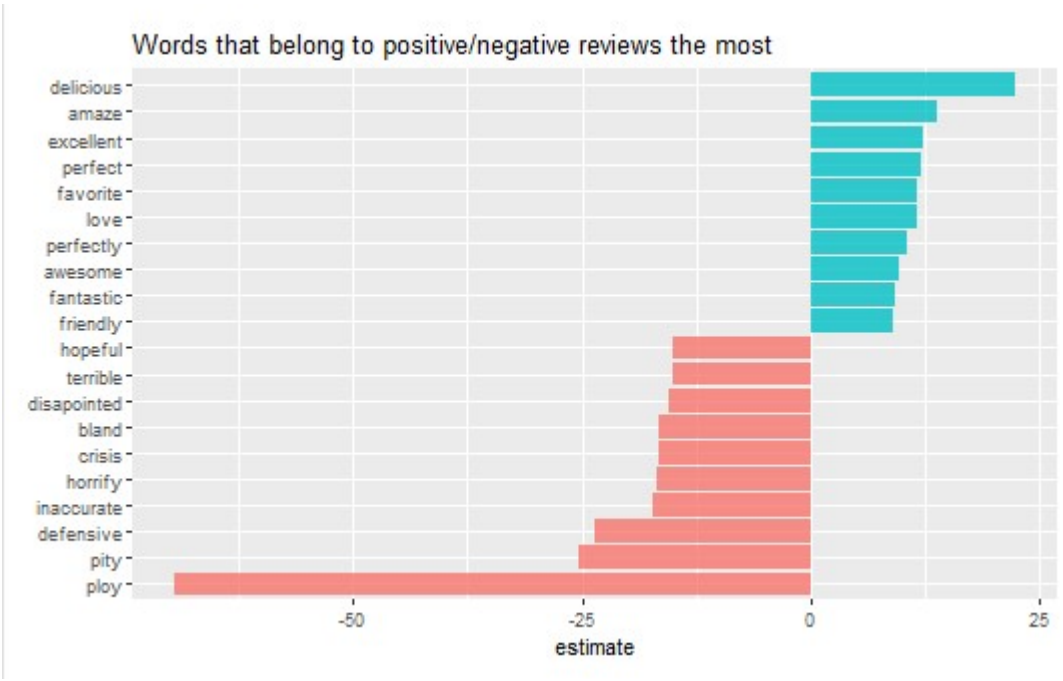
The naïve bayes model is faster but has lower prediction accuracy then random forest.

- **Lasso Logistic Regression Model:** We used the cv.glmnet to fit the lasso regression model. We used lasso model since the dataset may have multicollinearity and also to perform variable selection.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)



The right graph provides us the minimum and maximum lambda values and the left graph provides is the coefficient profile plot of the coefficients paths for the fitted model.



The above distribution from the lasso regression model clearly tells us the distribution of the positive sentiment words and the negative sentiment words related to their respective reviews.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	1861	184
1	780	7232

Accuracy = 90.41%

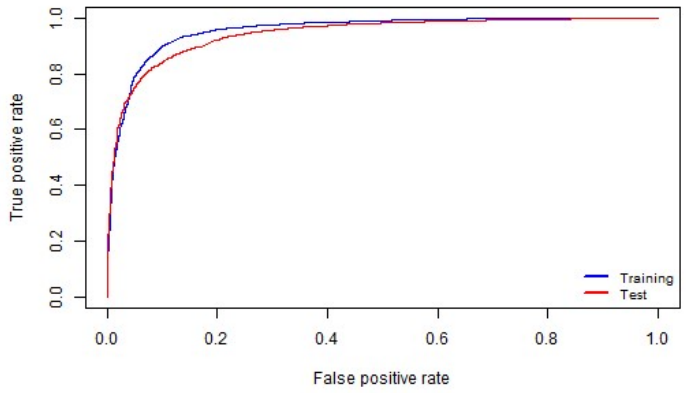
ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	1649	246
1	904	7259

Accuracy = 88.57%

ROC Curve:



AUC value for training data: 0.9568161
AUC value for test data: 0.9436305

The performance of lasso logistic regression and random forest is approximately the same for the Bing dictionary.

1. AFINN Dictionary:

The size of the document term matrix after applying the AFINN dictionary and removing the 3 star rating reviews: 39359 x 655

When we see the distribution of the ratings, we see there are more positive reviews than negative reviews.

hiLo <dbl>	n <int>
-1	10089
1	29270

We have taken a sample size of 39359 x 655 and divided that into 50% training and 50% test dataset while building our model.

- **Random Forest Model:** We created the random forest model using the ranger function to predict the hiLo (-1 or 1) ratings from the set of words in the review.

  Variable importance from the random forest model:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	x
bad	1.393035e-02
horrible	9.643950e-03
terrible	9.058298e-03
love	8.715400e-03
amaze	7.305726e-03
favorite	6.783193e-03
friendly	6.163213e-03
awesome	5.981519e-03
excellent	5.233175e-03
awful	4.901969e-03
poor	4.387381e-03
disgust	4.048166e-03
perfect	3.910443e-03
disappoint	3.737122e-03
leave	3.616034e-03
fresh	3.281481e-03
pay	2.960034e-03
lack	2.845509e-03
waste	2.696125e-03
fantastic	2.655416e-03
gross	2.423381e-03
wonderful	1.962430e-03
happy	1.867363e-03
suck	1.832757e-03
enjoy	1.783815e-03
perfectly	1.693988e-03
dirty	1.674137e-03
yummy	1.600905e-03
sad	1.546305e-03

If we look at the above set of words which are given higher importance according to the random forest model, they do relate to the restaurant review context very well.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	4314	681
1	213	14471

Accuracy = 95.45%

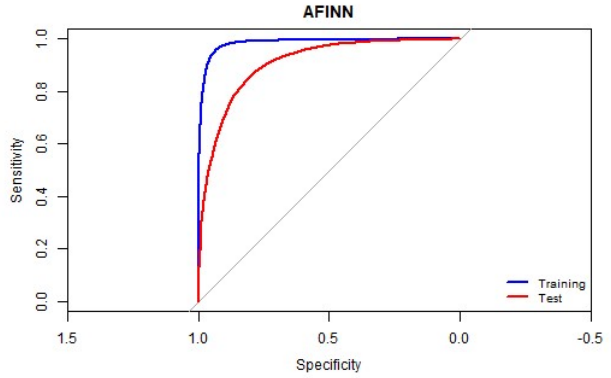
ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	3254	1840
1	822	13764

Accuracy = 86.47%

ROC Curve:



AUC value for training data: 0.9855
AUC value for test data: 0.9062

By looking at the above performance measures, we can say that the text in the reviews is indicative of their star ratings and hence can be used to discern the star ratings.

- **Naïve Bayes Model:** We used the naïve bayes model to predict the hiLo(-1 or 1) ratings with the set of words in the reviews. We have used the original full dataset for this model since it has the capacity to run large chunks of data.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	3223	1772
1	3408	11276

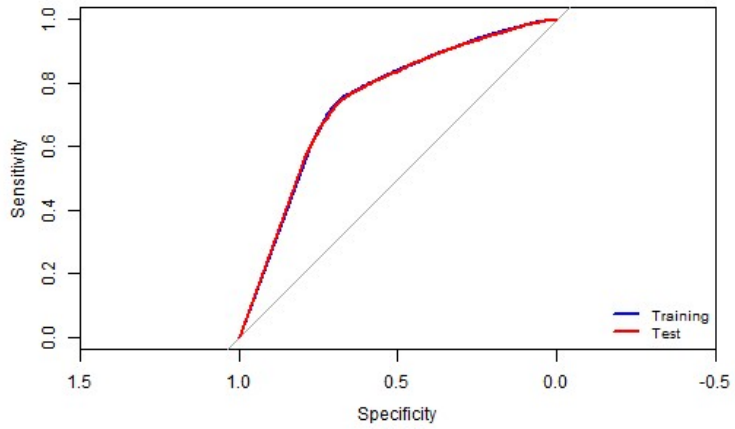
Accuracy = 73.67%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	3308	1786

1	3440	11146

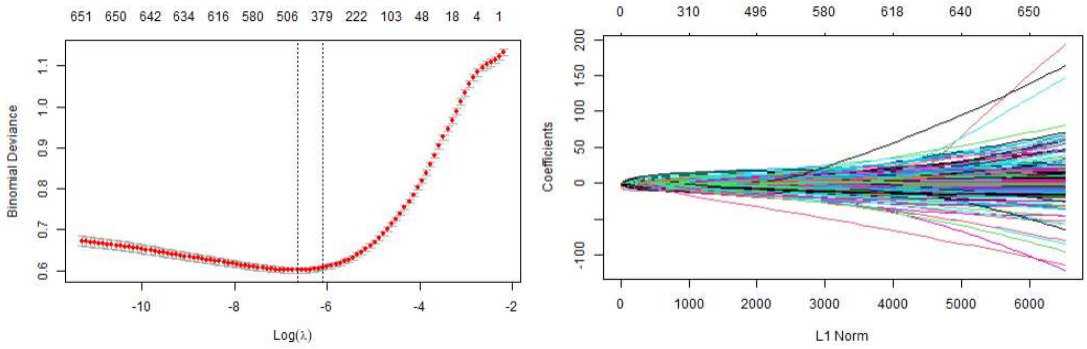
Accuracy = 73.44%
ROC Curve:



AUC value for training data: 0.7422
AUC value for test data: 0.7414

The naïve bayes model has lower prediction accuracy than random forest model even in case of AFINN dictionary.

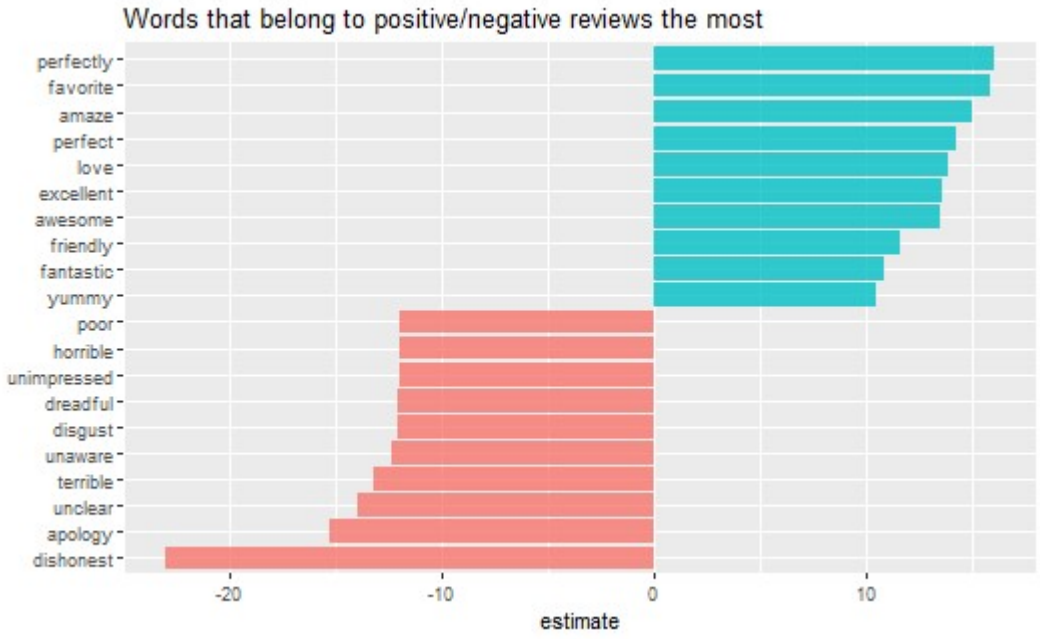
- **<u>Lasso Logistic Regression Model:</u>** We used the cv.glmnet to fit the lasso regression model. We used lasso model since the dataset may have multicollinearity and also to perform variable selection.



The right graph provides us the minimum and maximum lambda values and the left graph provides is the coefficient profile plot of the coefficient's paths for the fitted model.

Words that belong to positive/negative sentiments as per the lasso regression model:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)



The above distribution from the lasso regression model clearly tells us the distribution of the positive sentiment words and the negative sentiment words related to their respective reviews.

Confusion Matrix for training data:

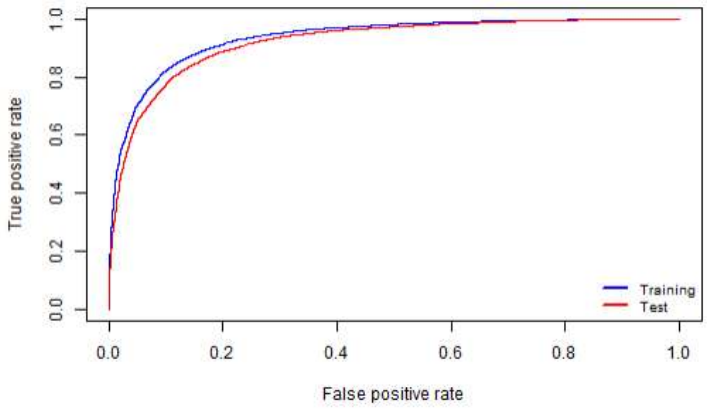
	Predicted	
Actual	-1	1
-1	3234	530
1	1761	14154

Accuracy = 88.36%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	3106	589
1	1988	13997

Accuracy = 86.91%
ROC Curve:



AUC value for training data: 0.9368899
AUC value for test data: 0.9217028

Even in case of AFINN dictionary, the lasso logistic regression model has similar prediction accuracy as random forest model.

2. NRC Dictionary:

The size of the document term matrix after applying the Bing dictionary and removing the 3 star rating reviews: 47921 x 1677

When we see the distribution of the ratings, we see there are more positive reviews then negative reviews.

hiLo <dbl>	n <int>
-1	10611
1	30441

We have taken a sample size of 10263 x 1677 and divided that into training and test dataset while building our model to reduce the runtime.

- **Random Forest Model:** We created the random forest model using the ranger function to predict the hiLo (-1 or 1) ratings from the set of words in the review.

  Variable importance from the random forest model:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	x
bad	7.611445e-03
delicious	4.012685e-03
horrible	3.714714e-03
terrible	2.608298e-03
bland	2.430602e-03
amaze	2.269610e-03
love	2.198168e-03
disgust	1.668126e-03
awful	1.660839e-03
friendly	1.658126e-03
excellent	1.524179e-03
favorite	1.430329e-03
mediocre	1.169157e-03
leave	1.076992e-03
pay	1.057597e-03
customer	1.024597e-03
cold	1.012797e-03
disappoint	9.917199e-04
money	9.682087e-04
perfect	9.005169e-04
dirty	8.891219e-04
waste	8.740475e-04
overpriced	7.290631e-04
suck	6.969228e-04
tasteless	6.258413e-04
management	6.052345e-04

If we look at the above set of words which are given higher importance according to the random forest model, they do belong to the restaurant review context very well.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	502	2180
1	0	7581

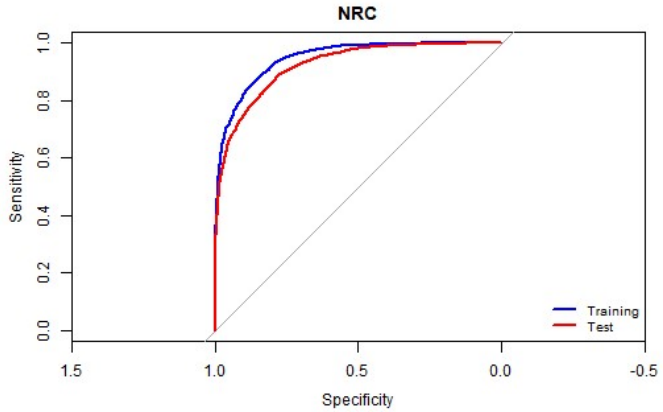
Accuracy = 78.75%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1

-1	419	2231
1	6	7607

Accuracy = 78.20%
ROC Curve:



AUC value for training data: 0.9469
AUC value for test data: 0.9221

By looking at the above performance measures, we can say that the text in the reviews are indicative of their star ratings and hence can be used to discern the star ratings. This model on NRC has the lowest accuracy as compared to the other two dictionaries.

- **Naïve Bayes Model:** We used the naïve bayes model to predict the hiLo(-1 or 1) ratings with the set of words in the reviews. We have used the original full dataset for this model since it has the capacity to run large chunks of data.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	4088	1244
1	7790	7404

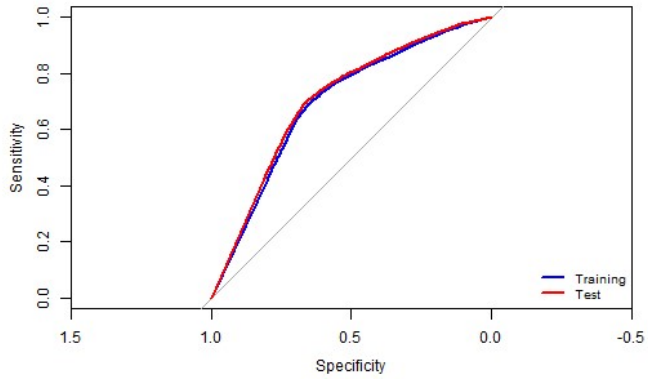
Accuracy = 55.98%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	4116	1163
1	7835	7412

Accuracy = 56.16%

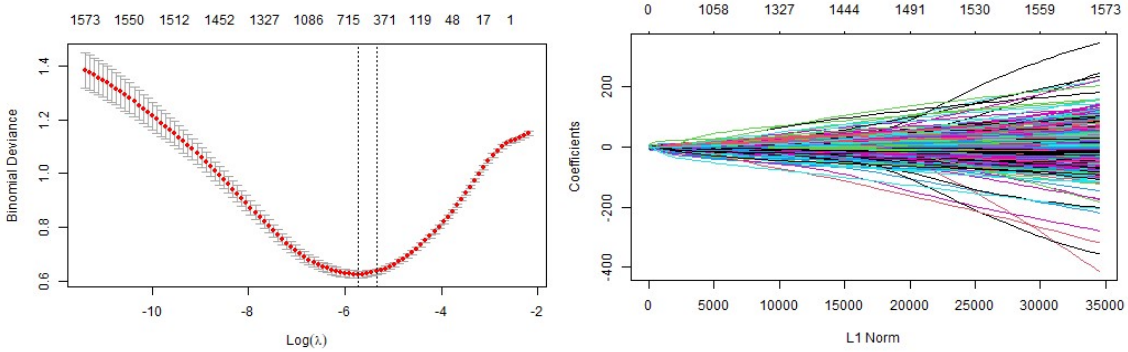
ROC Curve:



AUC value for training data: 0.6928
AUC value for test data: 0.705

The naive bayes model has lower accuracy as compared to random forest model even in case of NRC dictionary.

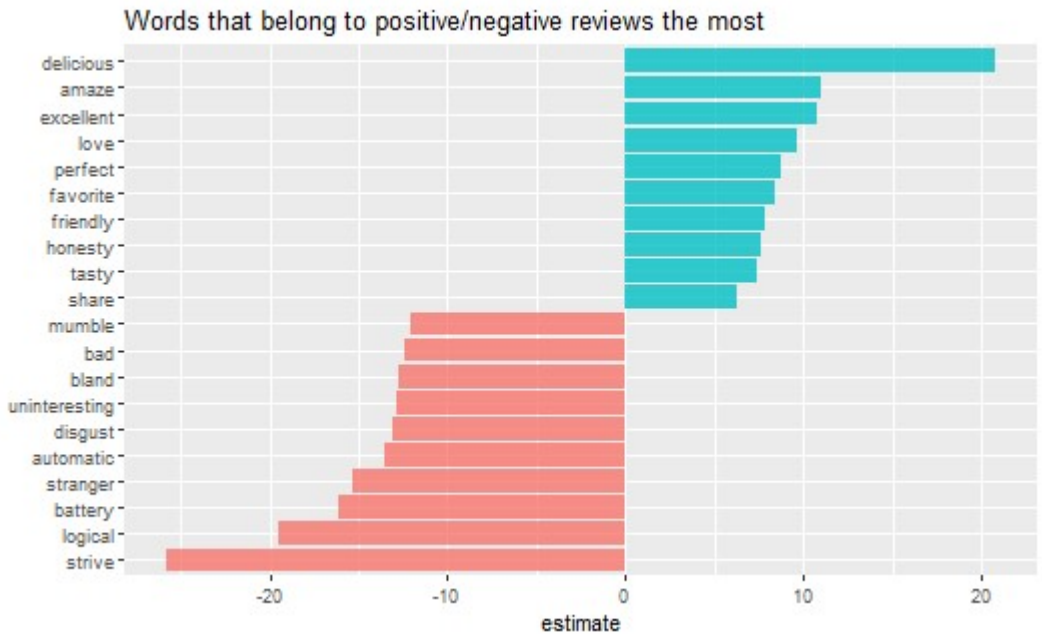
- **<u>Lasso Logistic Regression Model:</u>** We used the cv.glmnet to fit the lasso regression model. We used lasso model since the dataset may have multicollinearity and also to perform variable selection.



The right graph provides us the minimum and maximum lambda values and the left graph provides is the coefficient profile plot of the coefficient's paths for the fitted model.

Words that belong to positive/negative sentiments as per the lasso regression model:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)



The above distribution from the lasso regression model clearly tells us the distribution of the positive sentiment words and the negative sentiment words related to their respective reviews.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	1772	196
1	910	7385

Accuracy = 89.22%

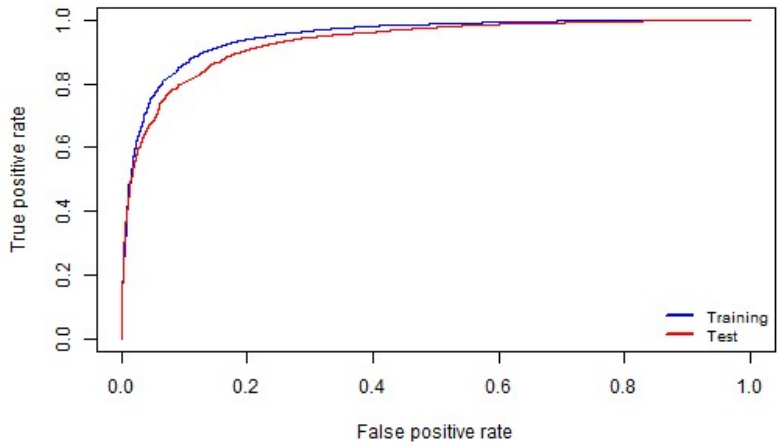
Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	1609	299
1	1041	7314

Accuracy = 86.94%

ROC Curve:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)



AUC value for training data: 0.9498417
AUC value for test data: 0.9324948

The lasso logistic regression model has better performance than random forest and naive bayes model in case of NRC dictionary.

Below is the summary of the performance measures of all the 3 dictionaries with all the 3 models.

	Random Forest Model	**Naïve Bayes Model**	**Lasso Regression Model**
Bing Dictionary	*AUC:* **0.928** *Accuracy:* **88.93%**	*AUC:* **0.7308** *Accuracy:* **65.61%**	*AUC:* **0.9436305** *Accuracy:* **88.57%**
AFINN Dictionary	*AUC:* **0.9062** *Accuracy:* **86.47%**	*AUC:* **0.7414** *Accuracy:* **73.44%**	*AUC:* **0.9217028** *Accuracy:* **86.91%**
NRC Dictionary	*AUC:* **0.9221** *Accuracy:* **78.20%**	*AUC:* **0.705** *Accuracy:* **56.16%**	*AUC:* **0.9324948** *Accuracy:* **86.94%**

The **random forest and lasso regression** models have better performance as compared to naïve bayes model. Comparing the 3 dictionaries, the **Bing** dictionary has an overall better predictive accuracy than others. Also, from our previous predictive analysis from the question4, we see that Bing dictionary had better accuracy which is also the case here.

When comparing from the analysis from the previous question, the random forest and lasso regression models here gave approximately the same accuracy for the Bing dictionary as predicted from the previous analysis.

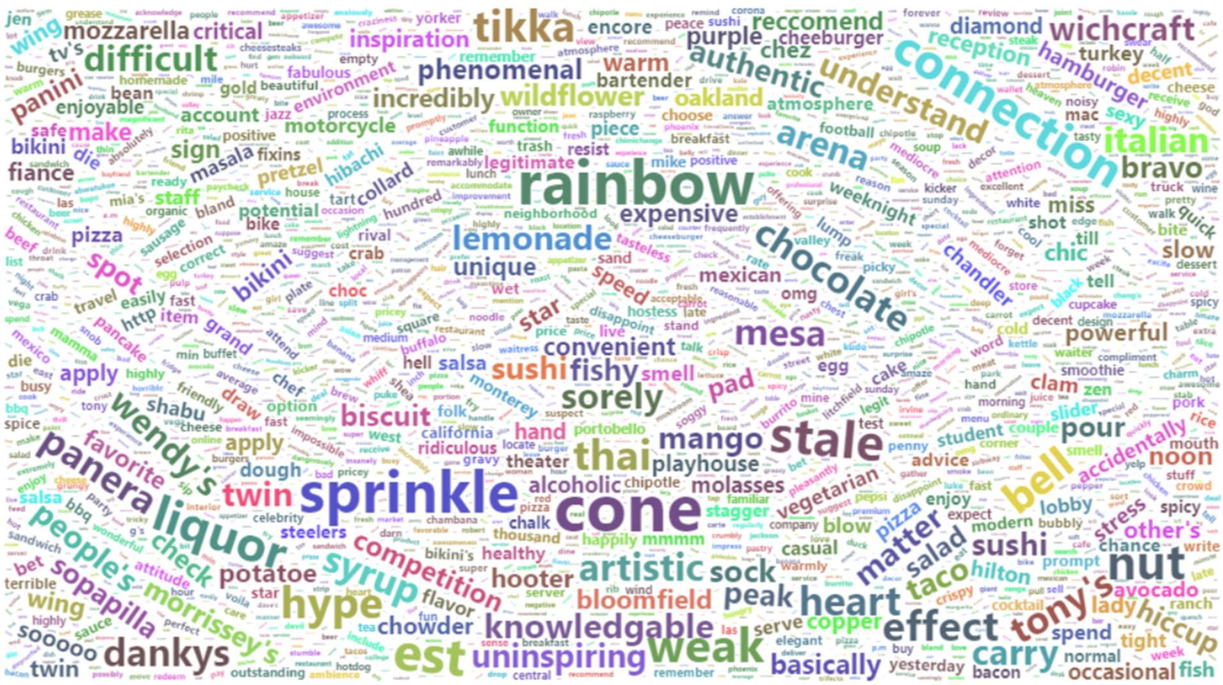
Combination of the three dictionaries:

We extracted the words from each of the three dictionaries (Bing, AFINN and NRC). Then joined them to get all the words from each of the dictionaries into a new combined dictionary. We joined our set of tokens from the original dataset with this combined dictionary terms to retain only the matching words. When we converted it into a document term matrix, we get a total of **48107 x 2264**. After filtering out the 3-star ratings, we get **41214 x 2264** document term matrix. They have below distribution of positive and negative reviews:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

hiLo <dbl>	n <int>
-1	10632
1	30582

Below is the word cloud of all the words from the 3 combinations of the dictionaries.



Each word has a root form or the stem. Stemming basically reduces a word to its root form. This may even be meaningless at times. Lemmatization groups also reduces the words to its root forms, but these root forms actually make sense unlike stemming. Lemmatization knows the context of the word before chopping the words. It is a dictionary-based approach, and it must be used when using the dictionaries. Stemming may reduce the words to meaningless or non-existent words which may not be found in our dictionaries used for sentiment analysis. Hence lemmatizations are preferred over stemming in this case.

We created a **random forest model** with this new set of combines sentiment dictionary. Again for quicker performance we have only considered a sample size of 20607 x 2264 which is divided equally into training and test data set.

Variable importance:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	x
bad	1.067474e-02
delicious	9.591158e-03
horrible	7.818387e-03
terrible	6.998335e-03
love	6.637178e-03
bland	6.004810e-03
amaze	5.995001e-03
rude	5.166590e-03
friendly	4.263021e-03
disgust	4.098046e-03
excellent	3.760040e-03
awesome	3.689808e-03
mediocre	3.643886e-03
favorite	3.326690e-03
leave	3.032104e-03
money	3.013741e-03
awful	2.969902e-03
pay	2.662806e-03
fresh	2.512808e-03
poor	2.486540e-03
cold	2.384400e-03
guess	2.352260e-03
perfect	2.337639e-03
gross	2.213702e-03
overpriced	2.050258e-03
suck	1.780188e-03
tasty	1.717887e-03
waste	1.674990e-03
disappoint	1.613696e-03

We can see that these words to belong to the restaurant review context and are given higher importance by the model.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	2517	124
1	26	7636

Accuracy = 98.54%

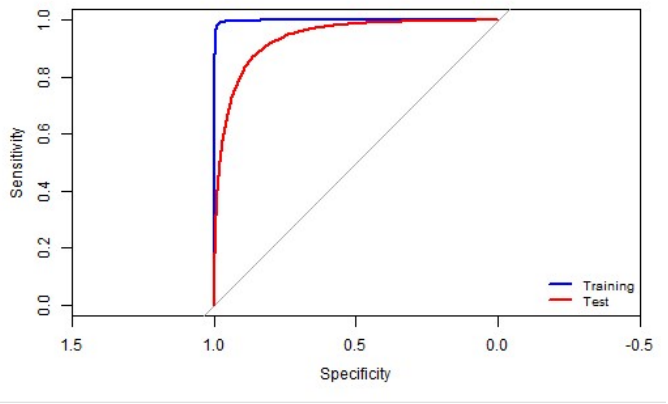
Confusion Matrix for test data:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	Predicted	
Actual	-1	1
-1	1744	877
1	248	7435

Accuracy = 89.08%

ROC Curve:



AUC value for training data: 0.9984
AUC value for test data: 0.9374

When compared to the performance of the individual dictionaries, the combined dictionary approach gives better performance with an **accuracy of 89.08% and AUC of 0.9374** as compared to the **average accuracy of 84.53% and AUC of 0.9187** for the random forest model for each of the dictionaries.

Using broader list of terms without using any dictionaries:

For this approach, we first found the count of each word occurring in multiple reviews and by this method we got a total of 7451 set of words from the reviews.
The top 20 words are:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

word <chr>	nr <int>
food	25484
service	16014
time	13626
eat	9677
restaurant	9360
love	9182
nice	7808
price	7724
menu	7388
delicious	7376
friendly	6988
chicken	6945
taste	6665
wait	6307
staff	6144
fry	6029
drink	5901
sauce	5715
fresh	5688
table	5574

The last 20 words are:

word <chr>	nr <int>
bla	6
cheeze	6
cho	6
chx	6
coordinator	6
deschutes	6
dougan	6
edens	6
frou	6
gammage	6
gaylord	6
ghee	6
gingerbread	6
jon	6
kudzu	6
lucas	6
marshall	6
maxwell	6
mcgrath	6
negroni	6
piero	6

Looking at these words some words don't make any sense regarding the sentiments for the reviews. Therefore, we will remove the words which occur in more than 90% of the reviews and less than 30 reviews. Now we get a total of 3868 words. Then we join these reduced set of words with our original tokens and convert the resultant dataset into a document term matrix by word. The document term matrix for this broader list is of size: **48237 x 3870**. After transforming the stars column and removing the 3-star rating reviews from the document term matrix, we get a document term matrix of size **41321 x 3870**.

We took a sample size of 20660 x 3868 and divided it into 50% training and 50% test data for building our random forest model to reduce the runtimes.

Variable importance from the random forest model:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

	x
terrible	9.565080e-03
bad	8.836706e-03
horrible	7.175696e-03
bland	5.938968e-03
minute	5.171852e-03
awful	4.555695e-03
rude	4.547839e-03
tell	4.514385e-03
mediocre	4.193449e-03
poor	4.123685e-03
amaze	4.006848e-03
disgust	3.155808e-03
spin-dry	2.850987e-03
excellent	2.779739e-03
awesome	2.667234e-03
overpriced	2.511237e-03
pay	2.386110e-03
cold	2.313500e-03
favorite	1.990080e-03
fresh	1.962778e-03

Looking at the variable importance, we do see that these words are closely related to the restaurant reviews context which are given higher importance by our model as expected.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	2584	26
1	1	7719

Accuracy = 99.7%

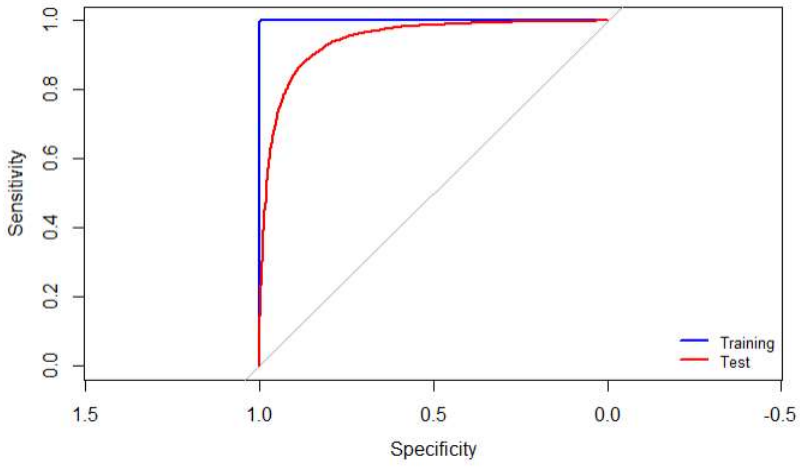
Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	1697	979
1	180	7474

Accuracy = 88.78%

ROC Curve:

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)



AUC value for training data: 1
AUC value for test data: 0.943

Let's look at the variable importance of the random forest model from Bing dictionary and this model:

	x
bad	1.423e-02
delicious	1.250e-02
horrible	1.133e-02
love	7.316e-03
terrible	7.036e-03
amaze	6.888e-03
awesome	5.707e-03
friendly	5.695e-03
bland	5.206e-03
disappoint	4.808e-03
rude	4.633e-03
favorite	4.242e-03
excellent	4.084e-03
disgust	4.067e-03
awful	3.754e-03
mediocre	3.645e-03
fresh	3.256e-03
poor	3.212e-03
perfect	3.145e-03
overpriced	2.889e-03

	x
terrible	9.565080e-03
bad	8.836706e-03
horrible	7.175696e-03
bland	5.938968e-03
minute	5.171852e-03
awful	4.555695e-03
rude	4.547839e-03
tell	4.514385e-03
mediocre	4.193449e-03
poor	4.123685e-03
amaze	4.006848e-03
disgust	3.155808e-03
spin-dry	2.850987e-03
excellent	2.779739e-03
awesome	2.667234e-03
overpriced	2.511237e-03
pay	2.386110e-03
cold	2.313500e-03
favorite	1.990080e-03
fresh	1.962778e-03

If we see their comparison most of the words are covered however words related to restaurants like cold, spin-dry are not included in the dictionary terms because they have a general set of words for sentiments. The general sentiment dictionaries are not applied to specific contexts and so may not include terms that are used in specific application contexts like in our case the restaurant reviews.

6. 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).
(a) 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.
(b) 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 6, 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 developing a separate dataframe with review_id, attribute, and then process this further to extract values for the different attributes.

Ans. People use various attributes for describing the restaurants found in the reviews like the ambience of the restaurant, whether they serve alcohol or not, whether they accept credit cards, whether they have wifi, whether it is good for a meal, good for kids, good for groups, if they provide take outs, if they have outdoor seating etc.

We tried to look at a few of them like Ambience, Good for which kind of Meal and Parking and extracted the values or descriptions given for each of these features for each restaurant review.

Ambience:

For Ambience, there are various descriptions used like casual, divey, trendy, classy, romantic, etc. which can be seen in the table 6.1. Out of these, there are 34634 reviews have ambience as casual, 5436 reviews don't specify the ambience at all and 1898 reviews talk about ambience being trendy for the different restaurants.

	amb	n
1	'casual'	34634
2	character(0)	5436
3	'trendy'	1898
4	'divey'	1526
5	'classy'	815
6	'intimate'	549
7	c(' 'classy'', ' 'upscale'')	516
8	'touristy'	449
9	'romantic'	323
10	c(' 'classy'', ' 'trendy'', ' 'upscale'')	310
11	c(' 'hipster'', ' 'trendy'')	255
12	'upscale'	244
13	c(' 'trendy'', ' 'casual'')	178
14	c(' 'touristy'', ' 'casual'')	166
15	c(' 'divey'', ' 'casual'')	147
16	c(' 'classy'', ' 'trendy'')	136
17	'hipster'	133
18	c(' 'romantic'', ' 'intimate'', ' 'classy'', ' 'casual'')	94
19	c(' 'intimate'', ' 'classy'', ' 'trendy'', ' 'casual'')	63
20	c(' 'hipster'', ' 'divey'', ' 'touristy'', ' 'casual'')	57
21	c(' 'romantic'', ' 'intimate'', ' 'casual'')	45
22	c(' 'romantic'', ' 'intimate'', ' 'classy'')	42
23	c(' 'romantic'', ' 'classy'')	37
24	c(' 'hipster'', ' 'casual'')	35
25	c(' 'romantic'', ' 'classy'', ' 'trendy'', ' 'upscale'')	35

Table 6.1: Distribution of the attribute Ambience in the reviews

For all the reviews that involves the ambience as being casual, the star rating distribution is as show below in Table 6.2. Overall, there a total of 35419 reviews which talk about the ambience as being casual and they provide a rating of 3.67 on an average and only 2048 reviews talk about the ambience being classy and they provide the average star rating of 3.85.

n() <int>	AvgStar <dbl>
35419	3.67108

n() <int>	AvgStar <dbl>
2048	3.853516

Table 6.2: Left: total number of 'casual' reviews and their average star ratings Right: total number of 'classy' reviews and their average star ratings

amb <chr>	n() <int>	AvgStar <dbl>
c(" 'hipster'", " 'divey'", " 'touristy'", " 'casual'")	57	4.649123
c(" 'trendy'", " 'casual'")	178	4.623596
c(" 'intimate'", " 'classy'", " 'trendy'", " 'casual'")	63	4.507937
c(" 'classy'", " 'trendy'", " 'upscale'")	310	3.925806
c(" 'classy'", " 'upscale'")	516	3.916667
'intimate'	549	3.903461
c(" 'romantic'", " 'intimate'", " 'classy'")	42	3.880952
'classy'	815	3.861350
'divey'	1526	3.817824
'romantic'	323	3.777090
c(" 'classy'", " 'trendy'")	136	3.764706
'trendy'	1898	3.734457
c(" 'romantic'", " 'intimate'", " 'casual'")	45	3.733333
'casual'	34634	3.669602
c(" 'hipster'", " 'trendy'")	255	3.666667
'hipster'	133	3.631579
character(0)	5436	3.580390
c(" 'romantic'", " 'intimate'", " 'classy'", " 'casual'")	94	3.468085
c(" 'hipster'", " 'casual'")	35	3.200000
c(" 'romantic'", " 'classy'")	37	3.135135
c(" 'divey'", " 'casual'")	147	3.108844
c(" 'romantic'", " 'classy'", " 'trendy'", " 'upscale'")	35	3.028571
c(" 'touristy'", " 'casual'")	166	3.000000
'upscale'	244	2.622951
'touristy'	449	2.385301

Table 6.3: Average star ratings as per the ambience mentioned in the reviews

amb <chr>	n_distinct(business_id) <int>	AvgStar <dbl>
'casual'	369	3.669602
'classy'	8	3.861350
'divey'	25	3.817824
'hipster'	2	3.631579
'intimate'	6	3.903461
'romantic'	2	3.777090
'touristy'	4	2.385301
'trendy'	14	3.734457
'upscale'	1	2.622951
c(" 'classy'", " 'trendy'")	1	3.764706
c(" 'classy'", " 'trendy'", " 'upscale'")	2	3.925806
c(" 'classy'", " 'upscale'")	3	3.916667
c(" 'divey'", " 'casual'")	2	3.108844
c(" 'hipster'", " 'casual'")	1	3.200000
c(" 'hipster'", " 'divey'", " 'touristy'", " 'casual'")	1	4.649123
c(" 'hipster'", " 'trendy'")	1	3.666667
c(" 'intimate'", " 'classy'", " 'trendy'", " 'casual'")	1	4.507937
c(" 'romantic'", " 'classy'")	1	3.135135
c(" 'romantic'", " 'classy'", " 'trendy'", " 'upscale'")	1	3.028571
c(" 'romantic'", " 'intimate'", " 'casual'")	1	3.733333
c(" 'romantic'", " 'intimate'", " 'classy'")	1	3.880952
c(" 'romantic'", " 'intimate'", " 'classy'", " 'casual'")	1	3.468085
c(" 'touristy'", " 'casual'")	2	3.000000
c(" 'trendy'", " 'casual'")	2	4.623596
character(0)	92	3.580390

Table 6.4: Count of restaurants based on the different ambiences and their star ratings

From table 6.3, we see that when people describe the place with multiple features of ambience, like casual, upscale, classy, touristy, hipster they seem to provide a higher star rating to the restaurant. Table 6.4 gives is us the distribution of all the restaurants based on the ambience specified in their reviews. For example, there are 369 restaurants with a casual ambience and only 8 which have a classy ambience in our overall dataset.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

Good For Meal:

There are various subcategories or sub-attributes for the attribute Good For Meal like the lunch, breakfast, dinner, dessert, late night and brunch. The table 6.5 provides us the distribution of each of these subcategories among all the reviews. There are 16091 reviews which talk about the restaurant being good for lunch and dinner and 2319 which do not talk about Good For meal attribute at all.

	GdFrMl	n
1	c(" 'lunch'", " 'dinner'")	16091
2	'lunch'	8933
3	'dinner'	7919
4	character(0)	2319
5	c(" 'lunch'", " 'breakfast'", " 'brunch'")	1612
6	c(" 'breakfast'", " 'brunch'")	1350
7	c(" 'latenight'", " 'lunch'", " 'dinner'")	1151
8	'breakfast'	989
9	c(" 'lunch'", " 'dinner'", " 'breakfast'")	791
10	'dessert'	784
11	c(" 'dessert'", " 'lunch'", " 'dinner'")	757
12	c(" 'dinner'", " 'brunch'")	732
13	c(" 'latenight'", " 'dinner'")	713
14	'brunch'	711
15	'latenight'	648
16	c(" 'lunch'", " 'breakfast'")	525
17	c(" 'dessert'", " 'breakfast'")	308
18	c(" 'lunch'", " 'brunch'")	278
19	c(" 'dessert'", " 'lunch'")	245
20	c(" 'latenight'", " 'lunch'")	150
21	c(" 'dinner'", " 'breakfast'", " 'brunch'")	142
22	c(" 'latenight'", " 'lunch'", " 'breakfast'")	141
23	c(" 'latenight'", " 'lunch'", " 'dinner'", " 'breakfast'")	135
24	c(" 'dessert'", " 'latenight'", " 'breakfast'", " 'brunch'")	127
25	c(" 'dessert'", " 'dinner'")	126
26	c(" 'dessert'", " 'lunch'", " 'breakfast'")	114
27	c(" 'lunch'", " 'dinner'", " 'brunch'")	108
28	c(" 'latenight'", " 'breakfast'", " 'brunch'")	69
29	c(" 'dessert'", " 'brunch'")	48
30	c(" 'latenight'", " 'lunch'", " 'breakfast'", " 'brunch'")	42
31	c(" 'latenight'", " 'breakfast'")	34
32	c(" 'lunch'", " 'dinner'", " 'breakfast'", " 'brunch'")	31

Table 6.5: Distribution of attribute Good for meal among all reviews

From table 6.6, we see that there are 31104 reviews which talk about the restaurants as a good place to have lunch and provide around 3.66 as an average star rating. Around 28696 reviews for restaurants talk about the restaurants as a good place to have dinner and provide a 3.68 as an average star rating.

n() <int>	AvgStar <dbl>
31104	3.669624

n() <int>	AvgStar <dbl>
28696	3.680234

Table 6.6: Left: total number of reviews with goodformeal as 'lunch' and their average star ratings Right: total number reviews with goodformeal as 'dinner' reviews and their average star ratings

GdFrMl <chr>	n() <int>	AvgStar <dbl>
c(" 'dessert'", " 'lunch'", " 'dinner'")	757	4.387054
c(" 'dessert'", " 'dinner'")	126	4.253968
c(" 'dessert'", " 'breakfast'")	308	4.237013
c(" 'lunch'", " 'breakfast'", " 'brunch'")	1612	4.029156
c(" 'lunch'", " 'dinner'", " 'breakfast'")	791	3.969659
c(" 'lunch'", " 'brunch'")	278	3.956835
'dessert'	784	3.950255
c(" 'lunch'", " 'dinner'", " 'breakfast'", " 'brunch'")	31	3.935484
c(" 'breakfast'", " 'brunch'")	1350	3.925185
c(" 'dessert'", " 'latenight'", " 'breakfast'", " 'brunch'")	127	3.913386
c(" 'lunch'", " 'breakfast'")	525	3.859048
c(" 'dessert'", " 'lunch'")	245	3.804082
'brunch'	711	3.756681
c(" 'lunch'", " 'dinner'")	16091	3.711267
'breakfast'	989	3.664307
'dinner'	7919	3.652103
c(" 'dinner'", " 'brunch'")	732	3.602459
c(" 'dessert'", " 'lunch'", " 'breakfast'")	114	3.561404
'lunch'	8933	3.510691
character(0)	2319	3.497197
'latenight'	648	3.453704
c(" 'latenight'", " 'lunch'")	150	3.446667
c(" 'dessert'", " 'brunch'")	48	3.375000
c(" 'latenight'", " 'lunch'", " 'dinner'")	1151	3.290182
c(" 'dinner'", " 'breakfast'", " 'brunch'")	142	3.197183
c(" 'latenight'", " 'lunch'", " 'dinner'", " 'breakfast'")	135	3.177778
c(" 'latenight'", " 'dinner'")	713	3.105189
c(" 'latenight'", " 'lunch'", " 'breakfast'", " 'brunch'")	42	3.047619
c(" 'lunch'", " 'dinner'", " 'brunch'")	108	3.046296
c(" 'latenight'", " 'breakfast'")	34	2.705882
c(" 'latenight'", " 'breakfast'", " 'brunch'")	69	2.608696
c(" 'latenight'", " 'lunch'", " 'breakfast'")	141	2.333333

Table 6.7: Average star ratings based on the values for Good For Meal attribute

GdFrMl <chr>	n <int>	AvgStar <dbl>
c(" 'lunch'", " 'dinner'")	146	3.711267
'lunch'	129	3.510691
'dinner'	81	3.652103
character(0)	43	3.497197
'breakfast'	17	3.664307
c(" 'breakfast'", " 'brunch'")	13	3.925185
c(" 'lunch'", " 'breakfast'", " 'brunch'")	13	4.029156
c(" 'latenight'", " 'lunch'", " 'dinner'")	12	3.290182
'dessert'	11	3.950255
'brunch'	9	3.756681
'latenight'	8	3.453704
c(" 'latenight'", " 'dinner'")	7	3.105189
c(" 'lunch'", " 'breakfast'")	6	3.859048
c(" 'lunch'", " 'dinner'", " 'breakfast'")	6	3.969659
c(" 'dessert'", " 'lunch'", " 'dinner'")	5	4.387054
c(" 'lunch'", " 'brunch'")	5	3.956835
c(" 'dessert'", " 'lunch'")	4	3.804082
c(" 'dinner'", " 'brunch'")	4	3.602459
c(" 'dinner'", " 'breakfast'", " 'brunch'")	3	3.197183
c(" 'latenight'", " 'lunch'")	3	3.446667
c(" 'latenight'", " 'lunch'", " 'dinner'", " 'breakfast'")	3	3.177778
c(" 'dessert'", " 'breakfast'")	2	4.237013
c(" 'dessert'", " 'latenight'", " 'breakfast'", " 'brunch'")	2	3.913386
c(" 'dessert'", " 'lunch'", " 'breakfast'")	2	3.561404
c(" 'latenight'", " 'lunch'", " 'breakfast'")	2	2.333333
c(" 'lunch'", " 'dinner'", " 'brunch'")	2	3.046296
c(" 'dessert'", " 'brunch'")	1	3.375000
c(" 'dessert'", " 'dinner'")	1	4.253968
c(" 'latenight'", " 'breakfast'")	1	2.705882
c(" 'latenight'", " 'breakfast'", " 'brunch'")	1	2.608696
c(" 'latenight'", " 'lunch'", " 'breakfast'", " 'brunch'")	1	3.047619
c(" 'lunch'", " 'dinner'", " 'breakfast'", " 'brunch'")	1	3.935484

Table 6.8: Count of restaurants based on the different good for meal values and their star ratings

From table 6.7, we see that when people describe the place to be good for lunch, dessert and dinner they tend to provide a higher star rating to the restaurant. Table 6.8 gives is us the distribution of all the restaurants based on the good for meal value specified in their reviews. For example, there are 146 different restaurants which are good for both lunch and dinner in our overall dataset.

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

Business Parking:

There are multiple subcategories for the attribute business parking like garage, street, validated, valet and lot. Table 6.9 shows us the distribution of these subcategories in the different restaurant reviews. There are around 28284 reviews which talk about restaurants providing lot parkings and around 4104 reviews don't talk about the parking at all.

	bsnsPrk	n
1	'lot'	28284
2	'street'	6611
3	character(0)	4104
4	'garage'	2869
5	c(" 'street'", " 'lot'")	1950
6	c(" 'garage'", " 'valet'")	918
7	'valet'	855
8	c(" 'street'", " 'valet'")	621
9	c(" 'garage'", " 'lot'", " 'valet'")	413
10	c(" 'garage'", " 'street'")	387
11	c(" 'lot'", " 'valet'")	235
12	c(" 'garage'", " 'street'", " 'validated'", " 'valet'")	234
13	c(" 'street'", " 'lot'", " 'valet'")	142
14	c(" 'garage'", " 'street'", " 'lot'")	138
15	c(" 'garage'", " 'lot'")	120
16	c(" 'garage'", " 'validated'")	88
17	c(" 'garage'", " 'street'", " 'validated'")	69
18	c(" 'garage'", " 'street'", " 'valet'")	46
19	'validated'	39

Table 6.9: Distribution of attribute business parking among all reviews

From table 6.10, we see that there are 31282 reviews which talk about the restaurants providing lot parking and these reviews have around 3.68 average star rating. Around 10198 reviews for restaurants talk about the restaurants providing street parking and these reviews have around 3.83 average star ratings.

n() <int>	AvgStar <dbl>
31282	3.684068

n() <int>	AvgStar <dbl>
10198	3.832418

Table 6.10: Left: total number of reviews with business parking as 'lot' and their average star ratings Right: total number reviews with business parking as 'street' reviews and their average star ratings

bsnsPrk <chr>	n() <int>	AvgStar <dbl>
c(" 'lot'", " 'valet'")	235	4.187234
c(" 'garage'", " 'street'", " 'valet'")	46	4.000000
c(" 'street'", " 'lot'")	1950	3.964103
'valet'	855	3.950877
c(" 'garage'", " 'street'", " 'validated'", " 'valet'")	234	3.914530
c(" 'garage'", " 'street'", " 'validated'")	69	3.884058
'street'	6611	3.825291
c(" 'garage'", " 'street'", " 'lot'")	138	3.746377
c(" 'street'", " 'valet'")	621	3.718196
'lot'	28284	3.670344
c(" 'garage'", " 'street'")	387	3.604651
character(0)	4104	3.567495
c(" 'garage'", " 'valet'")	918	3.474946
c(" 'garage'", " 'lot'")	120	3.358333
c(" 'street'", " 'lot'", " 'valet'")	142	3.345070
c(" 'garage'", " 'lot'", " 'valet'")	413	3.205811
'garage'	2869	3.143953
'validated'	39	2.897436
c(" 'garage'", " 'validated'")	88	2.602273

Table 6.11: Average star ratings based on the values for business parking attribute

bsnsPrk <chr>	n <int>	AvgStar <dbl>
'lot'	316	3.670344
character(0)	73	3.567495
'street'	65	3.825291
'garage'	30	3.143953
c(" 'street'", " 'lot'")	19	3.964103
c(" 'garage'", " 'valet'")	9	3.474946
'valet'	5	3.950877
c(" 'garage'", " 'lot'", " 'valet'")	4	3.205811
c(" 'garage'", " 'street'")	4	3.604651
c(" 'street'", " 'valet'")	4	3.718196
c(" 'lot'", " 'valet'")	3	4.187234
c(" 'garage'", " 'lot'")	2	3.358333
c(" 'garage'", " 'street'", " 'lot'")	2	3.746377
c(" 'garage'", " 'street'", " 'validated'", " 'valet'")	2	3.914530
c(" 'street'", " 'lot'", " 'valet'")	2	3.345070
'validated'	1	2.897436
c(" 'garage'", " 'street'", " 'valet'")	1	4.000000
c(" 'garage'", " 'street'", " 'validated'")	1	3.884058
c(" 'garage'", " 'validated'")	1	2.602273

Table 6.12: Count of restaurants based on the different business parking values and their star ratings

From table 6.11, we see that when customers get lot and valet parking in a restaurant, they tend to provide a higher star rating to the restaurant. Table 6.12 gives us the distribution of all the restaurants based on the business parking values specified in their reviews. For example, there are 316 different restaurants which provide lot parking in our overall dataset.

Music:

There are multiple sub features of the attribute Music like dj, background music, no music, karaoke, live, video and jukebox. Table 6.13 shows us the distribution of these sub features in the different restaurant reviews. There are around 444438 reviews which do not talk about music at all and around 1566 talk about background music. However, this distribution doesn't tell us about whether the reviews are positive or negative for these music features.

	music	n
1	character(0)	44438
2	'background_music'	1566
3	'live'	977
4	'jukebox'	552
5	'dj'	513
6	c(" 'background_music'", " 'live'")	76
7	c(" 'background_music'", " 'karaoke'", " 'live'")	1

Table 6.13: Distribution of attribute music among all reviews

From table 6.14, we see that there are 1643 reviews which talk about the restaurants background music and these reviews have around 3.37 average star rating. Around 1054 reviews for restaurants talk about the restaurants live music and these reviews have around 3.23 average star ratings.

n() <int>	AvgStar <dbl>	n() <int>	AvgStar <dbl>
1643	3.370663	1054	3.236243

Table 6.14: Left: total number of reviews with music as 'background_music' and their average star ratings
Right: total number reviews with music as 'live' reviews and their average star ratings

music <chr>	n() <int>	AvgStar <dbl>
'jukebox'	552	3.699275
character(0)	44438	3.681376
'dj'	513	3.485380
'background_music'	1566	3.411877
'live'	977	3.291709
c(" 'background_music'", " 'karaoke'", " 'live'")	1	3.000000
c(" 'background_music'", " 'live'")	76	2.526316

Table 6.15: Average star ratings based on the values for music attribute

music <chr>	n <int>	AvgStar <dbl>
character(0)	504	3.681376
'background_music'	15	3.411877
'live'	10	3.291709
'jukebox'	8	3.699275
'dj'	4	3.485380
c(" 'background_music'", " 'live'")	2	2.526316
c(" 'background_music'", " 'karaoke'", " 'live'")	1	3.000000

Table 6.16: Count of restaurants based on the different music values and their star ratings

From table 6.15, we see that when there is jukebox in a restaurant, people tend to provide a higher star rating to the restaurant. But most of the people don't talk about the music in the restaurants this could be either there are no music or people don't notice music that much. Table 6.16 gives us the distribution of all the restaurants based on the music values specified in their reviews. For example, most of the reviews don't talk about music of the restaurants in the reviews and only 15 restaurants have reviews about their background music.

Random Forest Model: For the model creation, we selected the random forest model since it had better performance in our previous analysis using the words of the reviews. In this model, we have used the various restaurant attributes like different features of ambience, good for meal, music, businessparking, BusinessAcceptsCreditCards , Caters , GoodForKids, OutdoorSeating, RestaurantsReservations,

ANISHA VIJAYAN (UIN: 662618335)
PREETHI SRINIVASAN (UIN: 663981973)

RestaurantsTableService, RestaurantsTakeOut, HappyHour, WheelchairAccessible, RestaurantsGoodForGroups, RestaurantsDelivery, HasTV.

Confusion Matrix for training data:

	Predicted	
Actual	-1	1
-1	215	504
1	120	1622

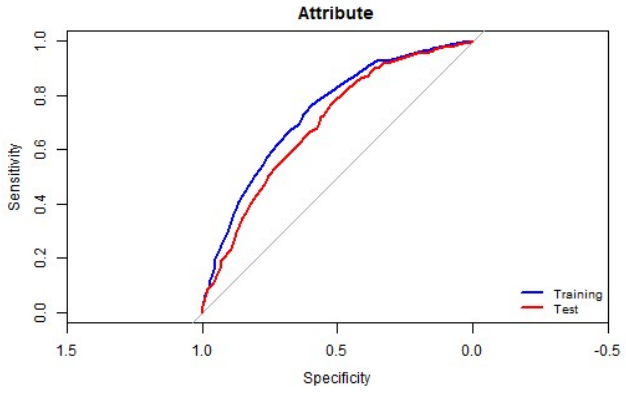
Accuracy = 74.64%

Confusion Matrix for test data:

	Predicted	
Actual	-1	1
-1	220	540
1	124	1578

Accuracy = 73.03%

ROC Curve:



AUC value for training data: 0.7381
AUC value for test data: 0.6948

As compared to the model with the words of the reviews, it does not have better prediction accuracy to the model with restaurant attributes as they provide a complete picture about the sentiment behind the reviews. The random forest model with the dictionaries had a prediction accuracy of **88.93%** and **0.928** AUC when we plotted the ROC curve for the unseen data. With this model we only get a prediction accuracy of **73.03%** and AUC of 0.6948 for unseen data.