IDS 572 – TEXT MINING – YELP REVIEWS CASE STUDY

IDS 572 – Assignment 4 on Restaurant Reviews Yelp Data

Abstract

Performing sentiment analysis for YELP reviews data

Submitted By:
Archana Singh - 668528470
Nikita Bawane - 661069000
Ritu Gangwal – 670646774

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

Solution:

Introduction -Yelp is one of the most popular sites for the users to rate several local businesses. It is a platform which acts as the bridge between the customers and the businesses. It helps users to rate the listings organized by the businesses. The ratings usually vary on a scale of 1 to 5 with 1 indicating a bad experience while 5 indicating an awesome experience of the user with the business. This setup leads the way to enormous amount of data generation from the Yelp site. In this assignment, we aim to inspect this data from Yelp by converting the unstructured review text from the customer, into structured data. The structured data would help us explore and build models for predicting the rating based on the review from a customer. In a world of data accompanied with uncertainty, such models are of great essence as customer reviews are a significant source of 'Voice of Customers', thus, providing the authentic insights of the customers. These insights in response help businesses make better decisions for the future.

Description of the Dataset

- The original Yelp dataset is available in the json files and comprises of five files namely business, review, user, check in and tip.
- Amongst the five files listed above, the primarily used files for predictive tasks are business and review data files.
- The review data files contain the information about reviews of different business from the user under the attributes like review id, review text, cool, funny, useful, business id, etc.The business data files hold the information specific to the business under the attributes like business id, name, attributes, categories, etc.
- For this assignment, we are using a sample of the original dataset (over 4 million review by over a million users for 144K businesses). This sample has been preprocessed to consider reviews from restaurants only.
- The sample dataset contains 26 columns and ~45k rows wherein each observation (row) is a review of a business by a user.
- The table below gives a brief description of the major attributes that define the data.
- In the next section, we will focus on acquiring the important attributes by exploring the data. These attributes will essentially be used for text mining and sentiment analysis in the subsequent parts of the assignment.

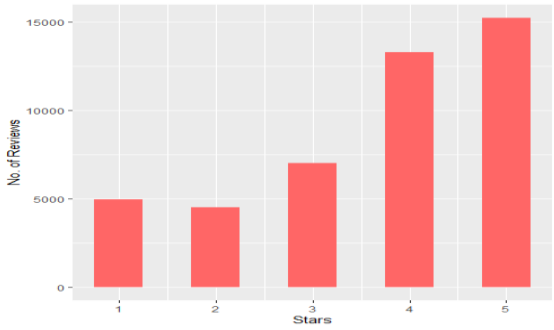
Exploratory Analysis

We first begin with exploring the data as this would give us the appropriate context of the data which is highly crucial before we can perform modelling. Thus, we have performed descriptive analysis on the distribution of reviews across different ratings, state and postal codes. Moreover, we have also analyzed the comments on reviews left by other users. This analysis will help us analyze the use of different comments across star ratings from 1 through 5. As yelp users begin with rating the restaurants they visited, we will commence our exploration with the review distribution across ratings.

Attributes	Description
Business ID	Unique id for a particular business
Cool	Represent the comments on review left by other users
Funny	
Useful	
Review ID	Unique id for a particular review
Stars	Rating of the business by the customer (No. of stars are from 1 through 5)
Text	Text of the review given by the user
User ID	Unique id of the user writing a review or rating a business
Address	Address of the business
Attributes	Various attributes of the business (free WiFi, Parking, etc)
Categories	Category to which a business belongs to (restaurants, beauty and salon, etc)
Name	Name of the business (Name of the restaurant for our sample)
Postal Code	Postal Code of the user giving the review
State	State of the user giving the review

a. Review Distribution across Star Ratings

To comprehend the distribution of reviews across star ratings from 1 through 5, we counted the number of reviews across each rating. The graph and table below give us a better understanding of the distribution.



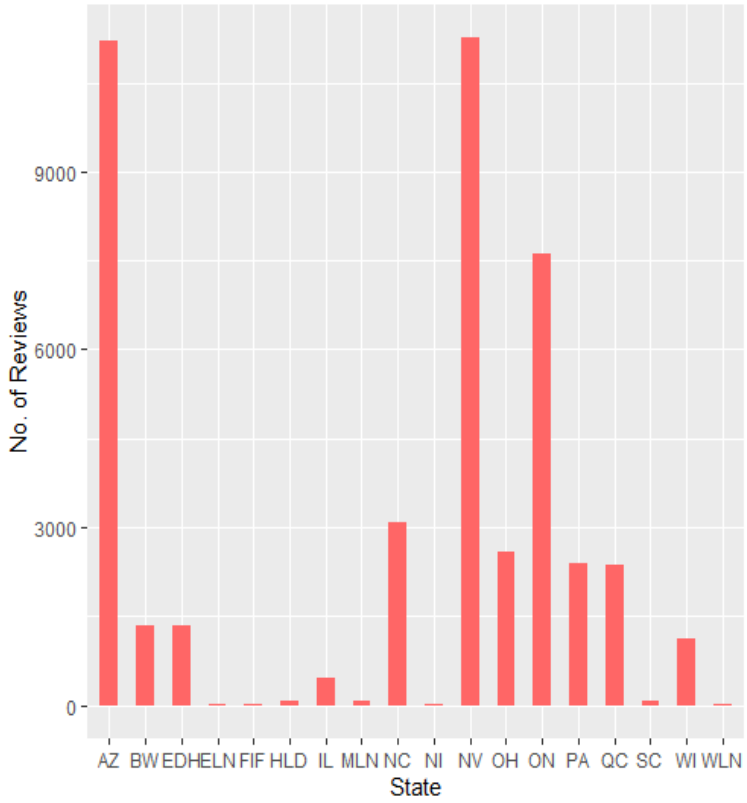
Stars	No of Reviews
1	4953
2	4516
3	6999
4	13306
5	15226

Average Star Rating
3.65

- The graph shows a plot of star ratings vs the number of reviews.
- As we can discern from the above graph, the number of reviews is increasing with the star ratings from 1 through 5.
- Majority of the reviews are inclining towards higher star ratings i.e. star rating of 4 and 5 account for approximately 63.5% of the reviews while the others together account for 36.5% of the reviews thus, making it a very small proportion as compared to 4 and 5 star rating.
- A higher proportion of reviews for star ratings of 4 and 5 signals the customer that more people are choosing a restaurant over others, which in response helps in attracting more customers towards the restaurant and this progression eventually steers the number of reviews even higher for the 4 and 5 star ratings.
- Furthermore, the average star rating for this dataset is **3.65.** We have used this rating as a threshold to obtain a 'positive' or 'negative' label for the documents.
- Hence, the documents having a star rating above 3.65 will be considered positive whereas the ones below 3.65 rating will be considered negative.

b. Review Distribution across States

To comprehend the distribution of reviews across various states, we counted the number of reviews for each state and plotted them. The graph and table below will help us to get a better sense of the distribution.

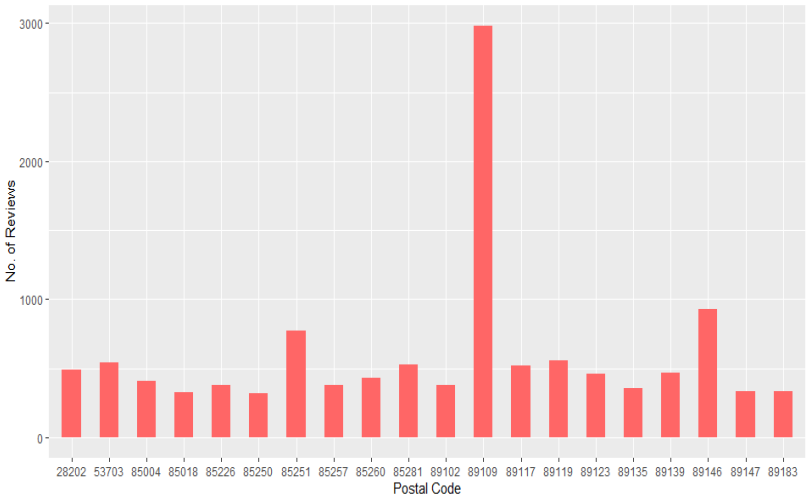


Sr. No.	State	No of Reviews
1	NV	11258
2	AZ	11207
3	ON	7614
4	NC	3083
5	OH	2587
6	PA	2380
7	QC	2375
8	BW	1348
9	EDH	1345
10	WI	1109
11	IL	457
12	HLD	63
13	SC	61
14	MLN	60
15	FIF	19
16	NI	14
17	WLN	14
18	ELN	6

- The graph shows a plot of states vs the number of reviews per state.
- It can be inferred from the graph that some states are outside the United States and thus, do not have well recognized abbreviations.
- The list of the states that belong to the United States and come into sight while exploring this dataset are Nevada, Arizona, North Carolina, Ohio, Pennsylvania, Wisconsin, Illinois and South Carolina.
- It is noticeable from the graph that the two peaks are at NV (Nevada) and AZ (Arizona) with 11258 and 11207 reviews, respectively. Thus, reviews from NV and AZ together account for 49.9% of the total number of reviews.
- The rationale behind the people of AZ and NV giving higher number of reviews could be the presence of substantial number of epicures.
- In addition to this, it could also be the presence of a set of specific restaurants that serve superior quality of food which fascinates huge crowd who turn out to give reviews on Yelp to express their experience and to aid the other epicures.
- While the peaks were observed at NV and AZ, a valley was observed at ELN with just 6 reviews in total.
- The reason behind the people of ELN not giving many reviews could be their lack of interest to use Yelp or their reluctance to go out and explore restaurants.

c. Review Distribution across Postal Codes

To apprehend the distribution of reviews across several postal codes, we counted the number of reviews for each postal code and plotted them. The total number of postal codes in our dataset are over 2000. Thus, we have taken into consideration the top 20 postal codes only. The graph and the table below will help us to get a finer apprehension of the distribution.



Sr. No.	Postal Code	No of Reviews
1	89109	2982
2	89146	930
3	85251	770
4	89119	556
5	53703	539
6	85281	528
7	89117	521
8	28202	488
9	89139	470
10	89123	462
11	85260	428
12	85004	409
13	85257	382
14	85226	378
15	89102	375
16	89135	358
17	89183	336
18	89147	335
19	85018	330
20	85250	316

- The graph shows a plot of postal codes vs number of reviews per postal code.
- It is an extension of the state vs number of reviews graph in the previous section as it helps us to explore postal codes associated with the different areas in a state.
- The highest number of reviews are under the postal code 89109 and 89146 which fall in Las Vegas, Nevada. This observation is very obvious and predictive beforehand because Nevada is the state with highest number of reviews.
- The postal code 89109 alone forms 6.63% of the total number of reviews.
- Moreover, while exploring the postal codes, we observed that not all the postal codes are 5-digit numbers.
- Thus, we have filtered out the postal codes that are not five-digit numbers.
- The total number of postal codes in the Yelp dataset are 2742 and after removing the non-five-digit postal codes, we move ahead with 512 postal codes for the subsequent parts of the assignment.

d. Review Comment distribution across Star Ratings

- Review comments are the comments left on a review by other users on Yelp.
- These comments indicate what other users think of the review in consideration. They may consider it good or bad and then can depict their view in the form of three review comments namely ‘Cool’, ‘Funny’ and ‘Useful’.
- Thus, the quality of a review can be assessed with the help of associated review comment.
- These comments could be associated with both positive and negative reviews.
- Thus, all the comments are distributed across 1 through 5-star ratings.
- The table below illustrates the exact distribution of comments across different ratings.

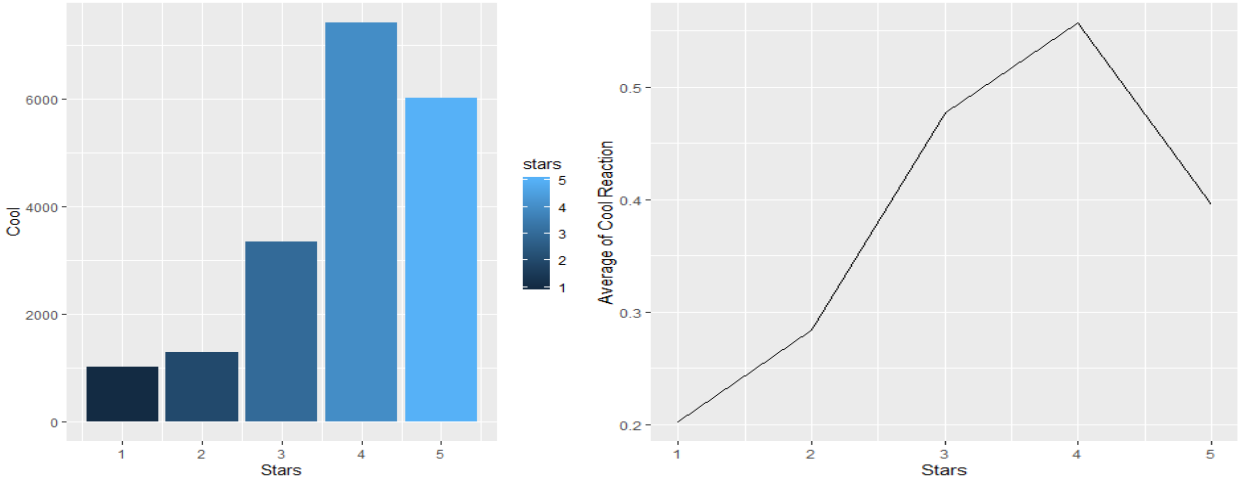
Reaction	Star Rating	Reactions	Total Number of Reactions
Cool	1	657	10335
	2	797	
	3	1731	
	4	3736	
	5	3414	
Funny	1	1135	7497
	2	968	
	3	1309	
	4	2164	
	5	1921	
Useful	1	2017	17557
	2	2012	
	3	2887	
	4	5376	
	5	5265	

- As can be seen from the above table, the maximum count from all the reactions is for 'Useful'.

- All the review comments have a general increasing trend across star ratings from 1 through 5.

- We move ahead with plotting the above data to get a better understanding of the relation between the review comments and star ratings.

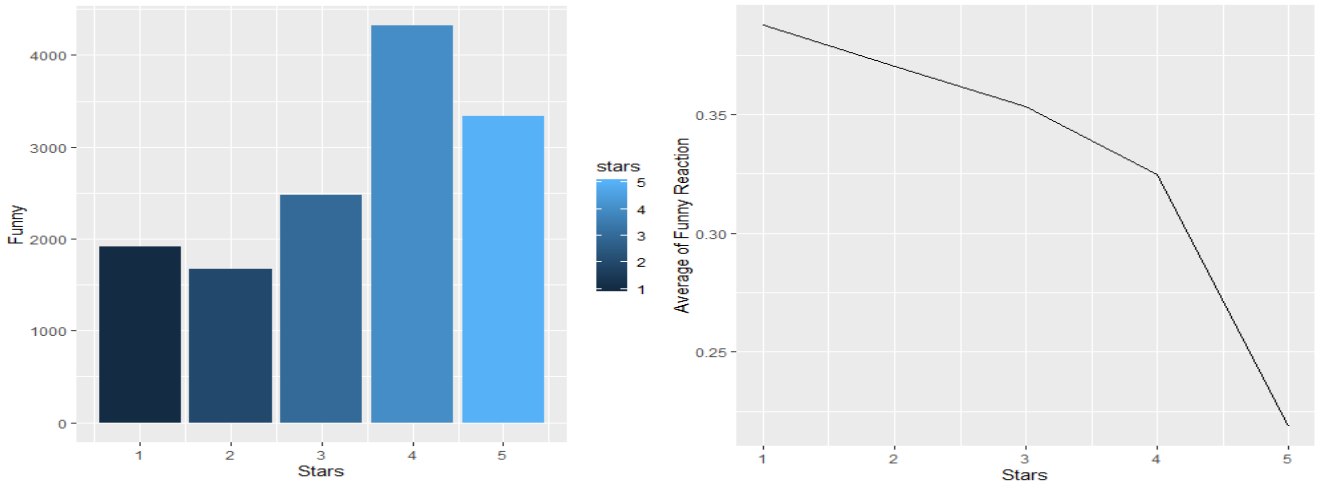
Review Comment: 'Cool'



- Plot 1 shows the distribution of 'Cool' comment across star ratings while Plot 2 shows the distribution of average of 'Cool' comment across star ratings.
- In general sense, we would associate a cool comment with a positive review.
- The visualization of our dataset also validates this general presumption.
- Though there is a small drop in review comment count from 4-star to 5-star rating, it can be intelligibly inferred from the two plots that the number of cool review comments increase with the increase in star ratings.
- Thus, when any other user drops a 'Cool' comment on an existing review, it would generally indicate a positive review with a higher rating, to which the user agrees.

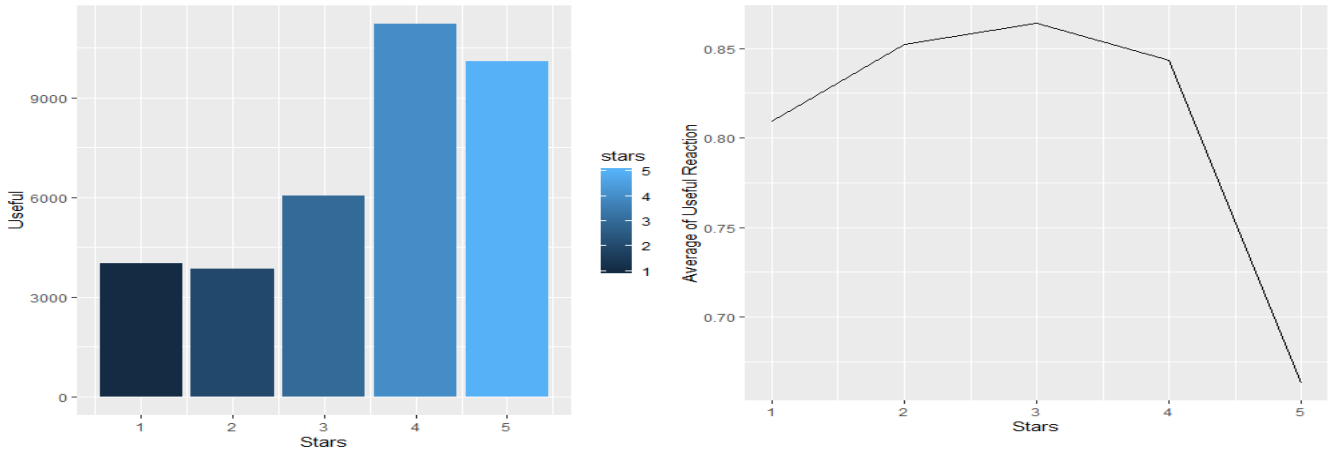
Review Comment: 'Funny'

- Plot 1 shows the distribution of 'Funny' comment across star ratings while Plot 2 shows the distribution of average of 'Funny' comment across star ratings.



- In general sense, we would associate a funny comment with a negative review.
- The visualization of our dataset also validates this general presumption.
- It can be intelligibly inferred from the second plot that the number of funny review comments decrease with the increase in star ratings.
- Thus, when any other user drops a 'Funny' comment on an existing review, it would generally indicate a negative review with a lower rating.

Review Comment: Useful



- Plot 1 shows the distribution of 'Useful' comment across star ratings while Plot 2 shows the distribution of average of 'Useful' comment across star ratings.
- In general sense, useful review comment can be associated with both positive and negative reviews. A user can mark any type of review as useful.
- The visualization of our dataset also validates this general presumption.
- It can be inferred from the second plot that the number of useful review comments do not follow any trend with respect to the star ratings.
- Thus, when any other user drops a 'Useful' comment on an existing review, it could indicate any review which the user finds useful.

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?
(For this, since we wish 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).

Solution:

After all the data exploration part, in order to get some words indicative of positive and negative sentiment, we performed the below data cleaning steps and analyzed those positive and negative words that make sense in the context of user reviews:

General data cleaning steps:

a. **Tokenization** - Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. In the process of tokenization, some characters like punctuation marks are discarded. Hence here all the reviews which are basically big sentences are broken down into words/tokens.

The "rrData" i.e. the initial data loaded was tokenized to "rrTokens" with dimension of 3248086 rows and 26 variables. Each row corresponds to a single token/word.

The total no. of distinct tokens/ words = 68204 i.e. our initial dataset has these many distinct words.

The image shown is a part of our dataset after tokenization where every row corresponds to a word/token.

	review_id	stars	word
	<chr>	<dbl>	<chr>
1	9OP3RlRPhSXrIb-yiJbSjA	3	good
2	9OP3RlRPhSXrIb-yiJbSjA	3	for
3	9OP3RlRPhSXrIb-yiJbSjA	3	chinese
4	9OP3RlRPhSXrIb-yiJbSjA	3	fast
5	9OP3RlRPhSXrIb-yiJbSjA	3	food
6	9OP3RlRPhSXrIb-yiJbSjA	3	it's

b. **Non- Alphabetic characters** – After tokenizing, we have removed all the non-alphabetic characters like numbers, time, etc. as shown in the image from our dataset/ list of tokens.

This left us with 49683 distinct tokens.

	word	n
1	00am	10
2	1.75	10
3	11.99	10
4	245	10
5	40th	10
6	53	10
7	57	10
8	6.75	10
9	6oz	10
10	70s	10

c. **Removing stop words** - Words such as articles and some verbs are usually considered stop words because they do not help us to find the context or the true meaning of a sentence. These are words that can be removed without any negative consequences to the final model that you are training. Hence after removing all the stop words, we are left with 49037 distinct tokens in our data set.

d. **Stemming/Lemmatization** –

- *Stemming* is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words. We have used Snowball algorithm to perform stemming on our data set.

- *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

We have decided to move forward with lemmatization on our dataset and converted all words into their basic form i.e. lemma.

No. of distinct words/tokens after lemmatization = 42180

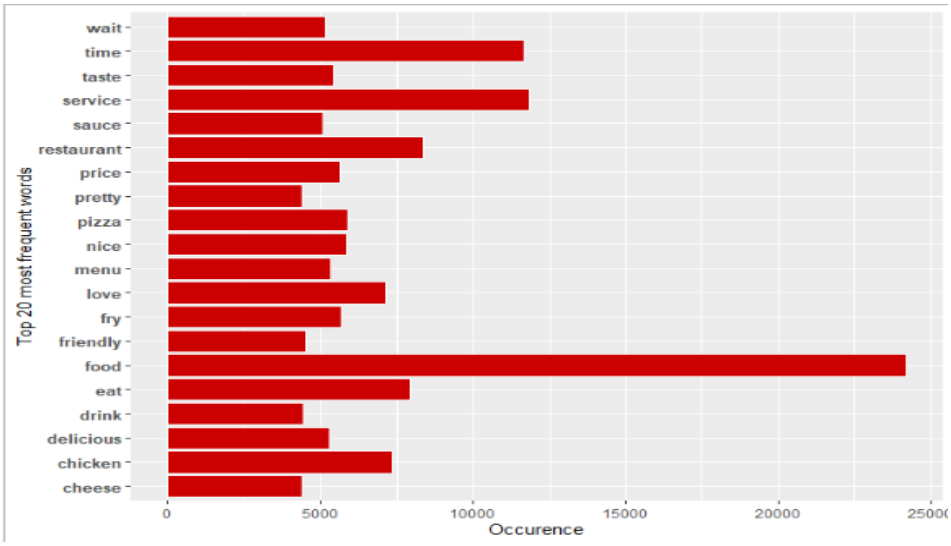
On the right is the example of part of our dataset after applying stemming and lemmatization.

	review_id	stars	word	word_stem	word_lemma
14	E3iln0KTrRM2kJpY5Pt7DA	3	american	american	american
15	E3iln0KTrRM2kJpY5Pt7DA	3	express	express	express
16	E3iln0KTrRM2kJpY5Pt7DA	3	holiday	holidai	holiday
17	E3iln0KTrRM2kJpY5Pt7DA	3	gift	gift	gift
18	E3iln0KTrRM2kJpY5Pt7DA	3	cards	card	card
19	z342HNuMNZmuI2fy4gOlhQ	3	shared	share	share
20	z342HNuMNZmuI2fy4gOlhQ	3	chicken	chicken	chicken
21	z342HNuMNZmuI2fy4gOlhQ	3	waffles	waffl	waffle
22	z342HNuMNZmuI2fy4gOlhQ	3	sunday	sundai	sunday
23	z342HNuMNZmuI2fy4gOlhQ	3	afternoon	afternoon	afternoon

e. **Removing words with length < 3 or > 15** – We have then removed all the words with character length less than 3 characters or more than 15 characters as they do not significantly contribute in final data modelling.
After all these general data cleaning, we are left with 40634 tokens/words.

Data Specific cleaning steps:

a. **Word Occurrence/frequency –** We have analyzed the occurrence of each word and found the top 20 frequent words. Below is the graph representing those words. Word "food" is the most frequent word.



	word	n
1	food	24194
2	service	11823
3	time	11679
4	restaurant	8368
5	eat	7947
6	chicken	7353
7	love	7122
8	pizza	5887
9	nice	5832
10	fry	5660
11	price	5631
12	taste	5408
13	menu	5316
14	delicious	5287
15	wait	5157
16	sauce	5088
17	friendly	4525
18	drink	4455
19	pretty	4389
20	cheese	4387

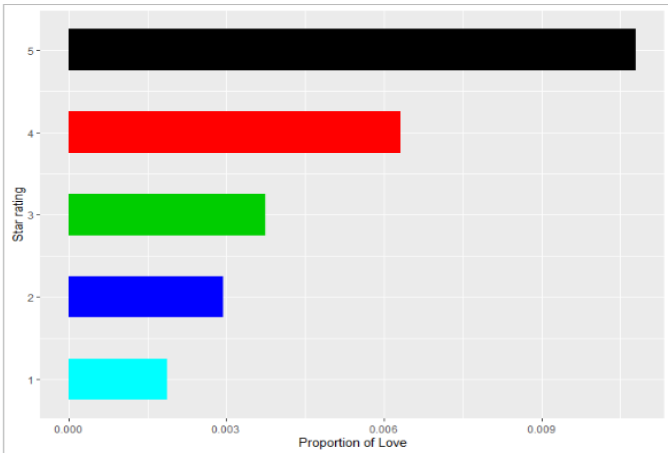
b. **Rare words** – We have then analyzed rare words in our data set i.e. words that occur less than 10 times in total. Below are some of the rare words in our data set. These rare words were removed and we had 6859 distinct tokens left.

word <chr>	n <int>
aaron	9
acidity	9
active	9
adobada	9
adresse	9
amanda	9
amaretto	9
amd	9
anxiously	9
apiece	9

c. **Proportion of words by star rating** – After removing the rare words, we have seen the proportion of each word according to their star rating. Below are the top 20 words from each star rating with highest proportion. Food, service, time are some of the words with highest proportion.

We have then studied each word with high proportions and seen their variation across ratings.

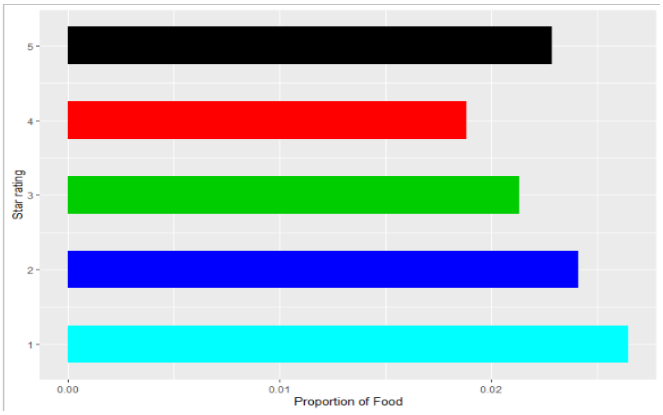
Token/Word “Love” -



	stars	word	n	prop
1	5	love	3778	0.010796352
2	4	love	2089	0.006311350
3	3	love	674	0.003741950
4	2	love	353	0.002931017
5	1	love	228	0.001878321

As we observe that the proportion of word “Love” decreases with rating. Its highest for rating 5 review. This is expected as it relates to the positive sentiment of people. Hence, this word plays an important role in deciding the rating of a particular review.

Token/ Word “Food”-



	stars	word	n	prop
1	5	food	7999	0.02285866
2	4	food	6234	0.01883435
3	3	food	3840	0.02131912
4	2	food	2906	0.02412900
5	1	food	3215	0.02648597

On contrary, the food occurs in almost same proportion in each rating as can be seen from the bar plot. Hence, one can conclude that the presence of this particular word does not contribute in deciding the rating of any review.

After studying all these highly occurring words, we found that presence of words like 'food', 'time', 'service', 'eat', 'restaurant', 'chicken, 'pizza', 'price', 'sauce', 'fry', 'menu' and ‘taste' do not make any contribution in deciding the star rating of any review and hence, deleted these tokens from our dataset.

Now, in order to get the sense of positive and negative words, we have calculated the average star rating of each word as totWS = proportion of word * star rating. The greater this value, the more positive sentiment of the word and vice versa.

Positive words

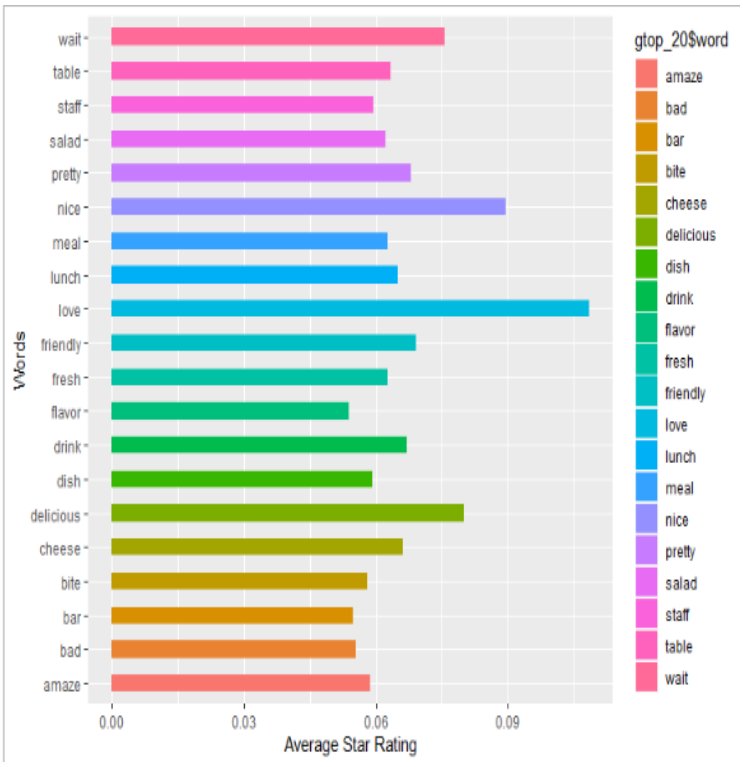
	word	totWS
1	love	0.10840466
2	nice	0.08943606
3	delicious	0.08011405
4	wait	0.07551276
5	friendly	0.06921075
6	pretty	0.06808314
7	drink	0.06693979
8	cheese	0.06621853
9	lunch	0.06488563
10	table	0.06330855

Negative words

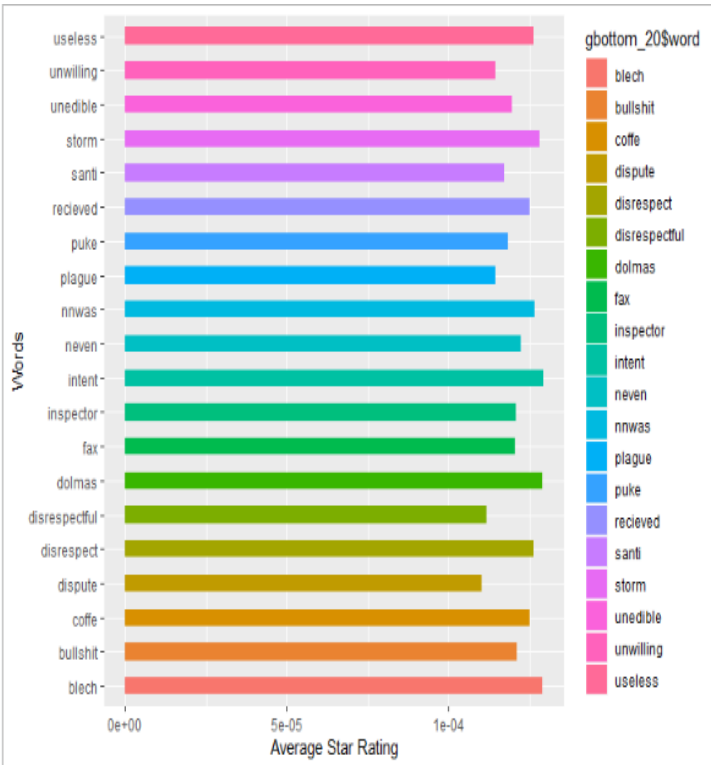
	word	totWS
1	intent	0.0001291291
2	dolmas	0.0001289387
3	blech	0.0001288676
4	storm	0.0001279447
5	nnwas	0.0001264846
6	disrespect	0.0001261290
7	useless	0.0001260768
8	coffe	0.0001250388
9	recieved	0.0001248864
10	neven	0.0001223503

Graphical representation of words with high average star rating – **Positive Words** and low average star rating – **Negative Words:**

Positive Sentiment



Negative Sentiment



Intuitively, we as customers would also associate similar words with good and bad experiences. So, the words we generated do make sense when compared with how reviews with low rating and reviews with high rating are structured.

d. **Calculation of term frequency-inverse document frequency (TFIDF) -** TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document (tf), and the inverse document frequency (idf) of the word across a set of documents.

Below are some of the words depicting their tf-idf value:

review_id <chr>	stars <dbl>	word <chr>	n <int>	tf <dbl>	idf <dbl>	tf_idf <dbl>
--9qM_dRW4rrKTWO_SX_qQ	1	buffet	1	0.125	4.075446	0.5094308
--9qM_dRW4rrKTWO_SX_qQ	1	copper	1	0.125	6.862220	0.8577775
--9qM_dRW4rrKTWO_SX_qQ	1	kettle	1	0.125	7.373045	0.9216307
--9qM_dRW4rrKTWO_SX_qQ	1	soo	1	0.125	6.048120	0.7560150
--9qM_dRW4rrKTWO_SX_qQ	1	star	1	0.125	2.464953	0.3081191
--9qM_dRW4rrKTWO_SX_qQ	1	suck	1	0.125	4.636824	0.5796030

6 rows

Hence, after all the data exploration and data cleaning with all the calculations, total no. of distinct tokens or words in our dataset = 6847 tokens.

3. We will consider three dictionaries, available through the tidytext package – the NRC dictionary of terms denoting different sentiments, the extended sentiment lexicon developed by Prof Bing Liu, and the AFINN dictionary which includes words commonly used in user-generated content in the web.
How many matching terms are there for each of the dictionaries?
Consider using the dictionary based positive and negative terms to predict sentiment (positive or negative based on star rating) of a movie. One approach for this is: using each dictionary, obtain an aggregated positive Score and a negative Score for each review; for the AFINN dictionary, an aggregate positivity score can be obtained for each review. Are you able to predict review sentiment based on these aggregated scores, and how do they perform? Does any dictionary perform better?

Solution:

Sentiment analysis is one of the most common text classification method that analyzes an incoming text and tells whether the underlying sentiment is positive, negative, or neutral. One of the most regular way to analyze the sentiment of a text is to consider the text as a combination of its individual words. We can then approximate the overall sentiment content of the text by examining the sentiment content of all the words in the text (tokens in the document). There are a variety of dictionaries available for evaluating the sentiment attached to a word. The tidytext package in R provides access to several sentiment lexicons. We have used the three general lexicons namely afinn, bing and nrc. The table below gives us an overview of all the three dictionaries that we will be using in our assignment.

Dictionary	Total No. of Words	Total No. of Distinct Words	Description of Sentiments in the Dictionary
Bing	6786	6783	This dictionary categorizes words into two sentiments namely Positive' and 'Negative'
Nrc	13901	6468	This dictionary categorizes words into ten sentiments namely 'Positive', 'Negative', 'Anger', 'Anticipation', 'Disgust', 'Fear', 'Joy', 'Sadness', 'Surprise' and 'Trust'
Afinn	2477	2477	This dictionary assigns words with a value that runs from -5 to +5 with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

The approach to discover the sentiment score of the reviews from the customers, is alike for all the dictionaries. The below steps give a prompt overview of the steps we have followed in order to compute the sentiment score.

- We have performed an inner join of our Yelp data with all the dictionaries individually to acquire a dataset that has a sentiment associated with every word of a review.
- The inner join helps us to retain only those words from the Yelp data whose corresponding sentiment is present in the dictionary thus generating a dataset that has no NA values in the sentiment column.
- We then enumerate the occurrence of each word in the dataset that was derived in the previous step.
- Going forward, we change the sign of the total occurrence of a word if the sentiment associated with the word is negative whereas for words with a positive sentiment, we do not make any changes in the sign of the total occurrence of a word.
- We then observe the most positive and least negative words based on the total occurrence.

- Advancing further, we group the dataset based on Review Id and Star Rating to summarize the count of positive and negative words for a review, respectively.
- We then quantify the proportion of positive and negative words by using the below mentioned formulas.

$$\text{Proportion of Positive words in a Review} = \frac{\text{Total number of positive words in the review}}{\text{Total number of words in the review}}$$

$$\text{Proportion of Negative words in a Review} = \frac{\text{Total number of negative words in the review}}{\text{Total number of words in the review}}$$

- The values calculated in the previous step are now used to determine the Sentiment Score for each review. We use the below mentioned formula to calculate the score.

$$\textbf{Sentiment Score} = \text{Proportion of Positive words in a Review} - \text{Proportion of Negative words in a Review}$$

- Now that we have the sentiment score for all the reviews, we can examine how accurate the sentiment score is by comparing it with the star rating of a review.
- We have filtered out all the three-star ratings to exclude the neutral rating and include the extreme ones only.
- We then created two new columns namely 'hiLo' and 'pred_hiLo' to create a confusion matrix.
- 'hiLo' takes value 1 when the star rating is greater than or equal to 4 and value -1 when the star rating is less than or equal to 2. 'pred_hiLo' takes value 1 when the sentiment score is greater than 0 and value -1 when the sentiment score is equal to or less than 0.
- Going forward, we create a confusion matrix where the star rating is the actual value while the sentiment score is the predicted value.
- We compute the accuracy using the confusion matrix. Accuracy is used to compare the different dictionaries and pick the one which is most precise for our dataset.

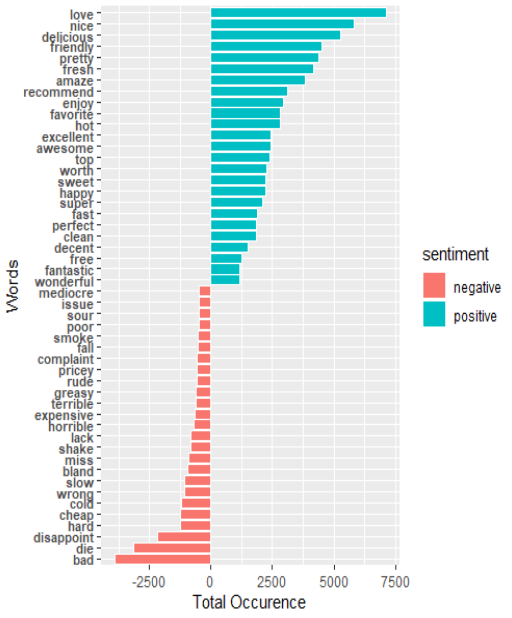
Bing Dictionary

The Bing dictionary classifies words into two sentiments. We use this dictionary to get the sentiments of words in our dataset. After performing the inner join, we observe that exactly 1000 words from our dataset get an associated sentiment from the Bing dictionary. The distribution of our dataset under these sentiments is summarized in the below table.

Sentiment	Word Count	Word Total Occurrence
negative	543	61452
positive	457	122184

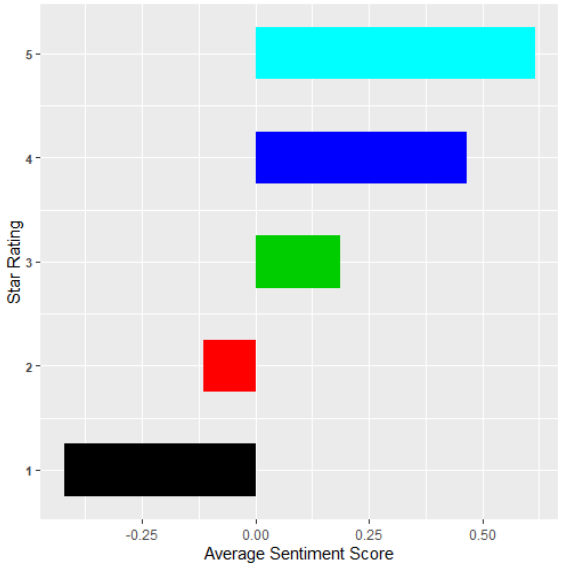
As mentioned in the previous section, we can contemplate the recurring positive and negative words by arranging the words in descending order of total occurrence. After having done this arrangement, the top and bottom 25 words give us a list of most frequently occurring positive and negative words, respectively. As can be seen in the below plot and tables, the words with negative sentiments have negative total occurrence and the ones with positive sentiments have a positive total occurrence. We can easily infer from the below plot that words such as 'love', 'nice', 'delicious', 'friendly' etc. must have been used by

customers who loved the restaurant and in response to their satisfaction level, gave a high rating to the restaurant on Yelp whereas words like 'bad', 'disappoint', 'wrong', 'slow' etc. must have been used by the customers who did not like the restaurant and ended up giving a low rating to it on Yelp.



Sr. No.	Word	Sentiment	Total Occurrence
1	love	positive	7122
2	nice	positive	5832
3	delicious	positive	5287
4	friendly	positive	4525
5	pretty	positive	4389
6	fresh	positive	4156
7	amaze	positive	3823
8	recommend	positive	3130
9	enjoy	positive	2954
10	favorite	positive	2830
11	hot	positive	2823
12	excellent	positive	2431
13	awesome	positive	2423
14	top	positive	2379
15	worth	positive	2279
16	sweet	positive	2226
17	happy	positive	2217
18	super	positive	2102
19	fast	positive	1867
20	clean	positive	1849
21	perfect	positive	1849
22	decent	positive	1505
23	free	positive	1265
24	fantastic	positive	1184
25	wonderful	positive	1171

Sr. No.	Word	Sentiment	Total Occurrence
1	bad	negative	-3915
2	die	negative	-3175
3	disappoint	negative	-2199
4	hard	negative	-1257
5	cheap	negative	-1237
6	cold	negative	-1211
7	wrong	negative	-1094
8	slow	negative	-1076
9	bland	negative	-955
10	miss	negative	-918
11	shake	negative	-821
12	lack	negative	-809
13	horrible	negative	-695
14	expensive	negative	-674
15	terrible	negative	-632
16	greasy	negative	-598
17	rude	negative	-583
18	pricey	negative	-575
19	complaint	negative	-568
20	fall	negative	-532
21	smoke	negative	-528
22	poor	negative	-503
23	sour	negative	-502
24	issue	negative	-496
25	mediocre	negative	-468



- After having analyzed the most frequently occurring positive and negative words, we will now explore if there is any relationship between the star rating and the computed sentiment score.
- It can be clearly inferred from the graph that with the increase in the rating given by a customer, the average sentiment score is also increasing.
- As sentiment score is calculated by removing the proportion of negative words from the proportion of positive words in a review, a positive sentiment score for a given rating would indicate that it has more positive words than negative.
- Similarly, a negative sentiment score for a given rating would indicate that it has more negative words than positive words.

The plot for Bing dictionary suggests that star rating 1 and 2 have higher number of negative words than positive words while star rating 3, 4 and 5 have higher number of positive words than negative words.

Stars	Average Positive Proportion of Words	Average Negative Proportion of Words	Average Sentiment Score of Reviews
1	0.288886661	0.711113339	-0.422226678
2	0.44243472	0.55756528	-0.11513056
3	0.593246731	0.406753269	0.186493462
4	0.732900557	0.267099443	0.465801114
5	0.80787608	0.19212392	0.61575216

We proceed further to create the confusion matrix of star ratings (actual) vs the sentiment scores(predicted). This confusion matrix is created by keeping into consideration only the extreme star ratings. Thus, we have excluded star rating 3 as it tends to be a neutral rating. The accuracy of the confusion matrix would assist us to inspect how proficient the Bing dictionary is in computing the sentiment score of the reviews in the Yelp dataset.

Bing Dictionary		Prediction	
		-1	1
Actual	-1	5157	1526
	1	3608	17205

The accuracy of our dataset with respect to Bing dictionary is **81.33%**. As accuracy is defined as the percentage of correctly classified instances, we can conclude that 81.33% of the reviews were correctly classified by the Bing dictionary.

Nrc dictionary

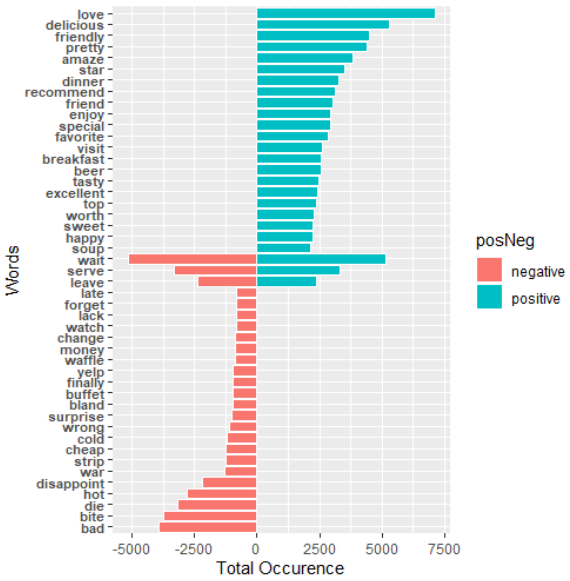
Sentiment	Word Count	Word Total Occurrence
anger	208	30950
anticipation	272	69740
disgust	177	24437
fear	217	27753
joy	255	88011
negative	556	79154
positive	683	166460
sadness	201	32313
surprise	163	33882
trust	357	86778

- The Nrc dictionary classifies words into ten sentiments.
- After performing the inner join, we observe that exactly 1428 words from our dataset get an associated sentiment from the Nrc dictionary.
- The distribution of our dataset under these sentiments is summarized in the table.
- It can be deduced from the table that sentiments like 'anticipation', 'joy', 'negative', 'positive' and 'trust' have been assigned to a larger proportion of the words while the other sentiments like 'anger', 'disgust', 'fear', 'sadness' and 'surprise' have been assigned to lesser number of words.

After having explored the sentiment distribution in Nrc dictionary, let us move ahead and look at the frequently occurring positive and negative words. As the Nrc dictionary does not directly allocate words into binary classifications, we have analyzed and classified the ten sentiments into positive and negative sentiments.

Assigned Sentiment	Original Sentiment
Positive	Anticipation, Joy, Positive, Surprise, Trust
Negative	Anger, Disgust, Fear, Negative, Sadness

We can now contemplate the recurring positive and negative words by arranging the words in descending order of total occurrence. After having done this arrangement, the top and bottom 25 words give us a list of most frequently occurring positive and negative words, respectively. We can easily infer from the below plot that words such as 'love', 'delicious', 'friendly', 'pretty' etc. must have been used by customers who loved the restaurant and in response to their satisfaction level and happiness, gave a high rating to the restaurant on Yelp whereas words like 'bad', 'disappoint', 'cold', 'bland' etc. must have been used by the customers who did not like the restaurant and ended up giving a low rating to it on Yelp.



Sr. No.	Word	Total Occurrence	posNeg Sentiment
1	wait	5157	positive
2	friendly	4525	positive
3	pretty	4389	positive
4	star	3493	positive
5	enjoy	2954	positive
6	top	2379	positive
7	sweet	2226	positive
8	happy	2217	positive
9	love	7122	positive
10	delicious	5287	positive
11	friend	3029	positive
12	special	2929	positive
13	favorite	2830	positive
14	beer	2578	positive
15	excellent	2431	positive
16	dinner	3280	positive
17	recommend	3130	positive
18	visit	2627	positive
19	breakfast	2578	positive
20	tasty	2491	positive
21	worth	2279	positive
22	soup	2155	positive
23	amaze	3823	positive
24	leave	2392	positive
25	serve	3316	positive

Sr. No.	Word	Total Occurrence	posNeg Sentiment
1	bad	-3915	negative
2	hot	-2823	negative
3	disappoint	-2199	negative
4	buffet	-954	negative
5	yelp	-945	negative
6	waffle	-868	negative
7	money	-862	negative
8	finally	-947	negative
9	die	-3175	negative
10	war	-1273	negative
11	surprise	-1012	negative
12	change	-855	negative
13	watch	-846	negative
14	wait	-5157	negative
15	bite	-3765	negative
16	serve	-3316	negative
17	leave	-2392	negative
18	strip	-1251	negative
19	cheap	-1237	negative
20	cold	-1211	negative
21	wrong	-1094	negative
22	bland	-955	negative
23	lack	-809	negative
24	forget	-804	negative
25	late	-803	negative

Word	Original Sentiment	Assigned Sentiment
wait	anticipation	positive
	negative	negative
serve	negative	negative
	trust	positive
leave	negative	negative
	sadness	negative
	surprise	positive

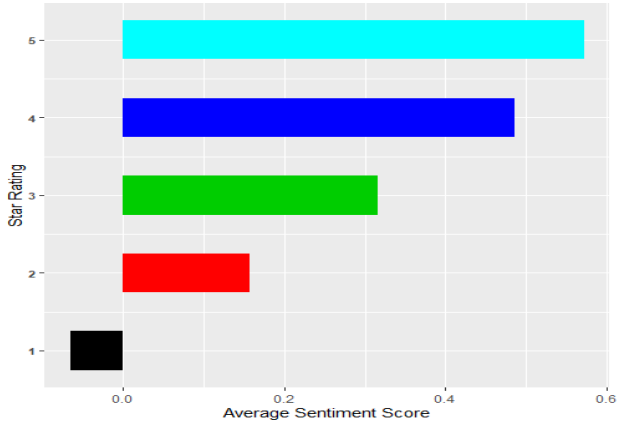
- It can be inferred from the plot and the table that words like 'wait', 'serve' and 'leave' have been assigned to multiple original sentiments.
- Some of these original sentiments have been classified as positive while the others are classified as negative.
- Hence, the bar graph for these words have extensions both towards the positive and negative sides.

After having analyzed the most frequently occurring positive and negative words, we will now explore if there is any relationship between the star rating and the computed sentiment score. In general sense, we would expect the sentiment score to increase with an increase in the star rating. The below table shows the average positive and negative proportion of words along with the average sentiment score for star ratings from 1 through 5.

It can be analyzed from the below table that the average positive proportion of words is increasing with the increase in star rating, but the average negative proportion of words is decreasing with the increase in star rating.

Stars	Average Positive Proportion of Words	Average Negative Proportion of Words	Average Sentiment Score of Reviews
1	0.466926045	0.533073955	-0.06614791
2	0.578120728	0.421879272	0.156241455
3	0.657879564	0.342120436	0.315759127
4	0.742693239	0.257306761	0.485386477
5	0.785972128	0.214027872	0.571944255

- It can be clearly inferred from the graph that with the increase in the rating given by a customer, the average sentiment score is also increasing.
- As sentiment score is calculated by removing the proportion of negative words from the proportion of positive words in a review, a positive sentiment score for a given rating would indicate that it has more positive words than negative



- Similarly, a negative sentiment score for a given rating would indicate that it has more negative words than positive words.
- The plot for Nrc dictionary suggests that star rating 1 has higher number of negative words than positive words.
- Star ratings 2, 3, 4 and 5 have higher number of positive words than negative words.

We proceed further to create the confusion matrix of star ratings (actual) vs the sentiment scores(predicted). This confusion matrix is created by keeping into consideration only the extreme star ratings. Thus, we have excluded star rating 3 as it tends to be a neutral rating. The accuracy of the confusion matrix would assist us to inspect how proficient the Nrc dictionary is in computing the sentiment score of the reviews in the Yelp dataset.

Nrc Dictionary		Predicted	
		-1	1
Actual	-1	3237	3655
	1	2708	18443

The accuracy of our dataset with respect to Nrc dictionary is **77.31%**. As accuracy is defined as the percentage of correctly classified instances, we can conclude that 77.31% of the reviews were correctly classified by the Nrc dictionary.

Afinn Dictionary

Value	Word Count	Word Total Occurrence
-5	1	20
-4	9	948
-3	47	11983
-2	152	17725
-1	71	14795
1	70	21300
2	131	40990
3	60	31417
4	13	7761
5	2	696

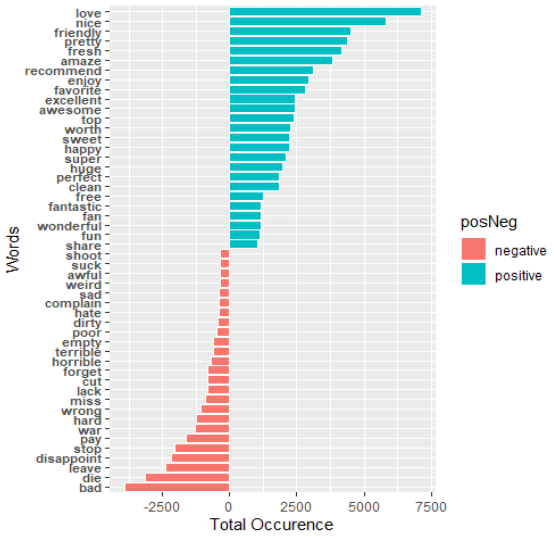
- The Afinn dictionary assigns each word with a value ranging -5 to +5 with negative values indicating negative sentiment while positive values indicating positive sentiment.
- After performing the inner join, we observe that exactly 556 words from our dataset get an associated sentiment from the Afinn dictionary.
- The distribution of our dataset under these sentiments is summarized in the table. None of the words from our dataset is assigned to value 0.

It can be deduced from the table that values like -3, -2, -1, 1, 2 and 3 have been assigned to larger number of words while values like -5, -4, 4 and 5 have been assigned to relatively lesser number of words.

After having explored the sentiment distribution in Afinn dictionary, let us move ahead and look at the frequently occurring positive and negative words. As the Afinn dictionary does not directly allocate words into binary classifications, we have analyzed and classified the values into positive and negative sentiments.

Assigned Sentiment	Original Value
Positive	5, 4, 3, 2, 1
Negative	-5, -4, -3, -2, -1

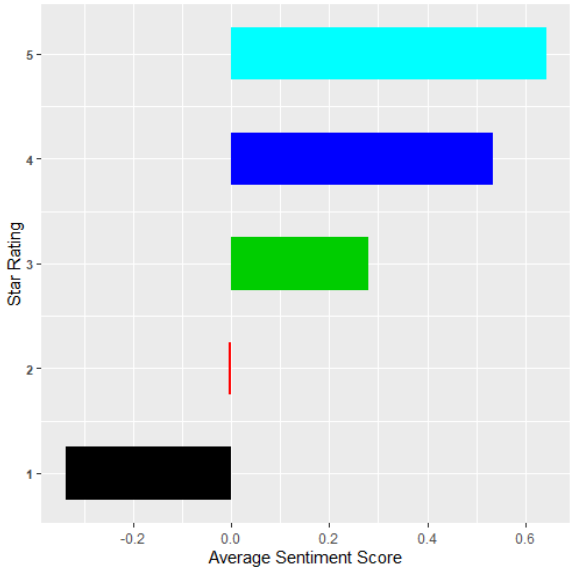
We can now contemplate the recurring positive and negative words by arranging the words in descending order of total occurrence. After having done this arrangement, the top and bottom 25 words give us a list of most frequently occurring positive and negative words, respectively. We can easily infer from the below plot that words such as 'love', 'nice', 'friendly', 'pretty' etc. must have been used by customers who loved the restaurant and in response to their satisfaction level and happiness, gave a high rating to the restaurant on Yelp whereas words like 'bad', 'disappoint', 'lack', 'horrible' etc. must have been used by the customers who did not like the restaurant and ended up giving a low rating to it on Yelp.



Sr. No.	Word	Value	Total Occurrence	posNeg Sentiment
1	pretty	1	4389	positive
2	fresh	1	4156	positive
3	huge	1	1955	positive
4	free	1	1265	positive
5	share	1	1047	positive
6	friendly	2	4525	positive
7	amaze	2	3823	positive
8	recommend	2	3130	positive
9	enjoy	2	2954	positive
10	favorite	2	2830	positive
11	top	2	2379	positive
12	worth	2	2279	positive
13	sweet	2	2226	positive
14	clean	2	1849	positive
15	love	3	7122	positive
16	nice	3	5832	positive
17	excellent	3	2431	positive
18	happy	3	2217	positive
19	super	3	2102	positive
20	perfect	3	1849	positive
21	fan	3	1176	positive
22	awesome	4	2423	positive
23	fantastic	4	1184	positive
24	wonderful	4	1171	positive
25	fun	4	1126	positive

Sr. No.	Word	Value	Total Occurrence	posNeg Sentiment
1	bad	-3	-3915	negative
2	die	-3	-3175	negative
3	horrible	-3	-695	negative
4	terrible	-3	-632	negative
5	hate	-3	-398	negative
6	awful	-3	-354	negative
7	suck	-3	-349	negative
8	disappoint	-2	-2199	negative
9	war	-2	-1273	negative
10	wrong	-2	-1094	negative
11	miss	-2	-918	negative
12	lack	-2	-809	negative
13	poor	-2	-503	negative
14	dirty	-2	-436	negative
15	complain	-2	-390	negative
16	sad	-2	-383	negative
17	weird	-2	-366	negative
18	leave	-1	-2392	negative
19	stop	-1	-2057	negative
20	pay	-1	-1617	negative
21	hard	-1	-1257	negative
22	cut	-1	-808	negative
23	forget	-1	-804	negative
24	empty	-1	-615	negative
25	shoot	-1	-346	negative

After having analyzed the most frequently occurring positive and negative words, we will now explore if there is any relationship between the star rating and the computed sentiment score. In general sense, we would expect the sentiment score to increase with an increase in the star rating.



- It can be clearly inferred from the graph that with the increase in the rating given by a customer, the average sentiment score is also increasing.
- As sentiment score is calculated by removing the proportion of negative words from the proportion of positive words in a review, a positive sentiment score for a given rating would indicate that it has more positive words than negative. Similarly, a negative sentiment score for a given rating would indicate that it has more negative words than positive.
- The plot for Afinn dictionary suggests that star rating 1 and 2 have higher number of negative words than positive words. Star ratings 3, 4 and 5 have higher number of positive words than negative words.

It can be analyzed from the table that the average positive proportion of words is increasing with the increase in star rating, but the average negative proportion of words is decreasing with the increase in star rating.

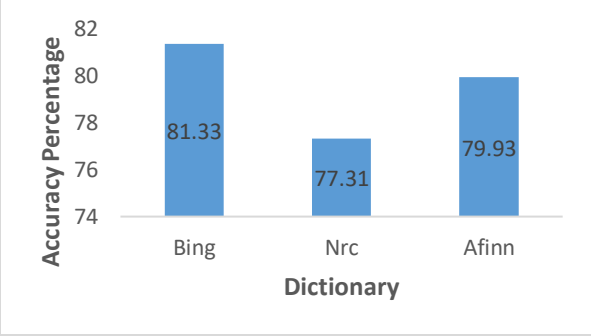
Stars	Average Positive Proportion of Words	Average Negative Proportion of Words	Average Sentiment Score of Reviews
1	0.330862074	0.669137926	-0.338275852
2	0.497420395	0.502579605	-0.005159209
3	0.640492682	0.359507318	0.280985364
4	0.767267775	0.232732225	0.53453555
5	0.821093463	0.178906537	0.642186926

We proceed further to create the confusion matrix of star ratings (actual) vs the sentiment scores(predicted). This confusion matrix is created by keeping into consideration only the extreme star ratings. Thus, we have excluded star rating 3 as it tends to be a neutral rating. The accuracy of the confusion matrix would assist us to inspect how proficient the Afinn dictionary is in computing the sentiment score of the reviews in the Yelp dataset.

Afinn Dictionary		Predicted	
		-1	1
Actual	-1	4592	1950
	1	3460	16950

The accuracy of our dataset with respect to Afinn dictionary is **79.93%**. As accuracy is defined as the percentage of correctly classified instances, we can conclude that 79.93% of the reviews were correctly classified by the Afinn dictionary.

Comparison of the Three Dictionaries



- The graph gives us a comparison between the three dictionaries in consideration.
- After performing inner join of the Yelp dataset with the Bing Liu dictionary, we found that exactly 1000 distinct words from our dataset were assigned with a sentiment from the dictionary.
- After performing inner join of the Yelp dataset with the Nrc dictionary, we found that exactly 1428 distinct words from our dataset were assigned with a sentiment from the dictionary.
- After performing inner join of the Yelp dataset with the Afinn dictionary, we found that exactly 556 distinct words from our dataset were assigned with a value from the dictionary.
- We observe that the Bing Liu dictionary performs the best amongst all others. It gives us the highest accuracy of 81.33%. The Afinn dictionary gives the next best result.
- Thus, to predict the sentiment associated with a review from a customer on Yelp, we would choose Bing Liu dictionary over others to perform the sentiment analysis.

4. Develop models to predict review sentiment.
You should develop at least three different types of models (Naïve Bayes, and at least two others of your choice - Lasso logistic regression (why Lasso?), xgb, svm, random forest)

(i). Develop models using only the sentiment dictionary terms – try the three different dictionaries; how do the dictionaries compare in terms of predictive performance for rating? Then with a combination of the three dictionaries, ie. combine all dictionary terms?
Do you use term frequency, tfidf, or other measures, and why? What is the size of the document term matrix? Should you use stemming when using the dictionaries?

Solution:

Predicting the sentiment of a review is the most crucial aim in this assignment. We performed sentiment analysis in the previous question and found out that Bing Liu dictionary works the best for our dataset. In this question, we plan to create models to predict the review sentiment. After having created multiple models on the matched dictionary terms, we will be at a stage to compare the results of review sentiment prediction in the previous question with the model-based review sentiment prediction in this question. We aim to create three models namely Random Forest, Naïve Bayes and Support Vector Machine on the matched dictionary terms. Before we can commence with the model creation, there are a few general steps that we follow for creating the dataset to run the models. The next section describes the preprocessing steps.

Common Methodology to Create Dataset Before Modeling

- In order to create models on the matched terms, we have used different dictionaries to get the sentiments of the words in the Yelp dataset. We have created matched datasets with Bing Liu dictionary, Nrc dictionary, Afinn dictionary and a combination of all the three dictionaries.
- We then inspected the matched datasets to examine if there are any duplicates. Removing duplicates is an essential step before we create the document term matrix because duplicates are considered as erroneous entries and hinder the accurate formation of the matrix.
- The most important step before creating any model is to generate the document term matrix. It gives us the frequency of each word (token) present in the review (document). We have created the document term matrix wherein the words from the matched dataset appear as column in the matrix along with two other columns viz. Review Id and Star Rating. The cell values are taken from the tfidf column.
- As we aim to consider only the positive and negative rating, we filtered out the star rating 3 as it tends to be a neutral rating.
- We then created a new column in the matrix called ‘hiLo’. It is a factor variable with values 1 and -1. 1 is assigned to hiLo when the star rating is greater than or equal to 4 and -1 is assigned to hiLo when the star rating is less than or equal to 2. Thus, this column partitions the words into binary categories.
- Before we can split the data into training and test dataset, we replaced all the NA’s with zeroes in the matrix. Now the document term matrix is in the ready to use form.
- We then split the matrix into training and test data with 50:50 proportion.

Extended Sentiment Lexicon - Prof Bing Liu Dictionary - Specifies lists of positive and negative words.

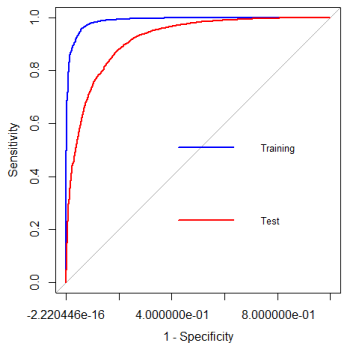
- Before creating models, we initiated the process to create the dataset that was supposed to be used for modeling.
- We performed inner join between the Yelp dataset and the Bing Liu dictionary to get the sentiment associated with every word.
- This matched dataset contained 1000 distinct words. This dataset was then used by us to create the document term matrix.
- The document term matrix had 32038 rows and 1002 columns (Review Id + Star Rating + 1000 words).
- After removing the 3-star rating, the document term matrix was left with 27496 rows and 1002 columns.
- The table gives the distribution of the words in the hiLo binary classifier target variable. The 6683 words with a hiLo value of -1 belong to the star rating of 1 or 2 while the 20813 words with a hiLo value of 1 belong to the star rating of 4 or 5.

hiLo value	Number of Words
-1	6683
1	20813

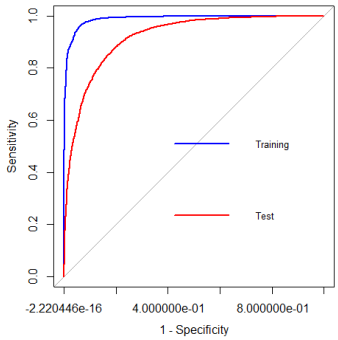
Random Forest Model:

The below table and graph summarize the input and output parameters for the multiple runs we made for the random forest model. We experimented with different combinations of ntree and mtry. The parameters that we have considered for performance evaluation are accuracy, precision, recall and receiver operating characteristics ROC.

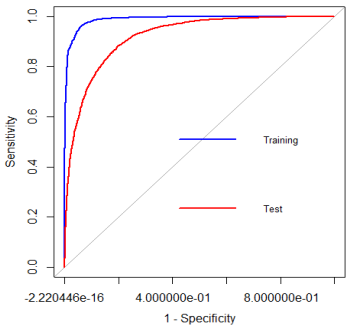
Bing Dictionary						Training Dataset			Test Dataset		
Model	Ntree	Mtry	Split Rule	OOB	Threshold	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Model 1	100	31	gini	0.09	0.64	0.95	0.98	0.96	0.87	0.92	0.90
					0.50	0.96	0.96	0.99	0.88	0.90	0.95
Model 2	500	31	gini	0.09	0.64	0.95	0.98	0.96	0.87	0.92	0.91
					0.50	0.96	0.96	0.99	0.88	0.90	0.95
Model 3	1000	31	gini	0.09	0.65	0.95	0.98	0.96	0.87	0.93	0.90
					0.50	0.96	0.96	0.99	0.88	0.90	0.95



ROC Curve for Model 1



ROC Curve for Model 2



ROC Curve for Model 3

NRC Dictionary - Provides lists of words denoting different sentiment (for eg., positive, negative, joy, fear, anticipation, etc.)

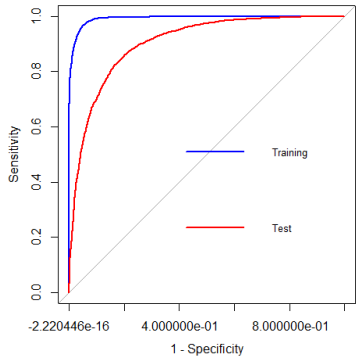
- We initiated the process to create the dataset that was supposed to be used for modeling.
- We performed inner join between the Yelp dataset and the Nrc dictionary to get the sentiment associated with every word.
- This matched dataset contained many duplicates as a single word could be associated to multiple sentiments in a Nrc dictionary.
- After removing the duplicates, the dataset had 1428 distinct words. This dataset was then used by us to create the document term matrix.
- The document term matrix had 32658 rows and 1430 columns (Review Id + Star Rating + 1428 words). After removing the 3-star rating, the document term matrix was left with 28043 rows and 1430 columns.
- The table gives the distribution of the words in the hiLo binary classifier target variable. The 6892 words with a hiLo value of -1 belong to the star rating of 1 or 2 while the 21151 words with a hiLo value of 1 belong to the star rating of 4 or 5.

hiLo value	Number of Words
-1	6892
1	21151

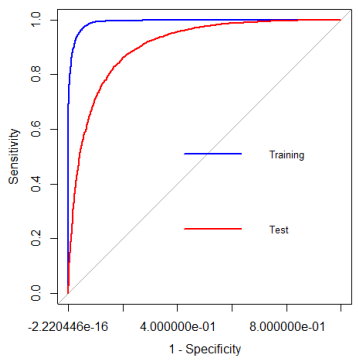
Random Forest Model:

The below table and graph summarize the input and output parameters for the multiple runs we made for the random forest model. We experimented with different combinations of ntree and mtry. The parameters that we have considered for performance evaluation are accuracy, precision, recall and receiver operating characteristics ROC.

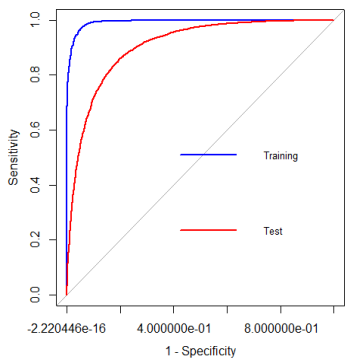
Nrc Dictionary						Training Dataset			Test Dataset		
Model	Ntree	Mtry	Split Rule	OOB	Threshold	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Model 1	100	37	gini	0.10	0.64	0.96	0.98	0.97	0.86	0.92	0.90
					0.50	0.97	0.96	0.99	0.87	0.88	0.95
Model 2	500	37	gini	0.10	0.65	0.96	0.98	0.97	0.86	0.92	0.90
					0.50	0.97	0.96	0.99	0.87	0.88	0.96
Model 3	1000	37	gini	0.10	0.63	0.97	0.98	0.97	0.86	0.91	0.91
					0.50	0.97	0.97	0.99	0.87	0.88	0.95



ROC Curve for Model 1



ROC Curve for Model 2



ROC Curve for Model 3

AFFIN Dictionary – Gives a list of words with each word being associated with a positivity score from -5 to +5.

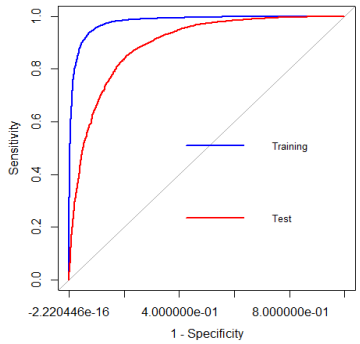
- Before creating models for AFINN, we initiated the process to create the dataset that was supposed to be used for modeling.
- We performed inner join between the Yelp dataset and the Afinn dictionary to get the sentiment associated with every word.
- The matched dataset had 556 distinct words. This dataset was then used by us to create the document term matrix.
- The document term matrix had 31346 rows and 558 columns (Review Id + Star Rating + 556 words).
- After removing the 3-star rating, the document term matrix was left with 26952 rows and 558 columns
- The table gives the distribution of the words in the hiLo binary classifier target variable. The 6542 words with a hiLo value of -1 belong to the star rating of 1 or 2 while the 20410 words with a hiLo value of 1 belong to the star rating of 4 or 5.

hiLo value	Number of words
-1	6542
1	20410

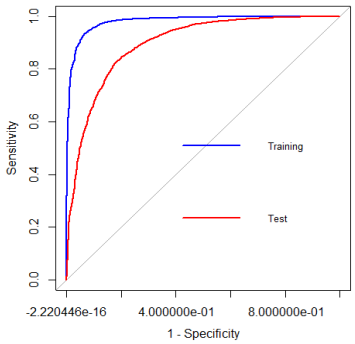
Random Forest Model:

The below table and graph summarize the input and output parameters for the multiple runs we made for the random forest model. We experimented with different combinations of ntree and mtry. The parameters that we have considered for performance evaluation are accuracy, precision, recall and receiver operating characteristics ROC.

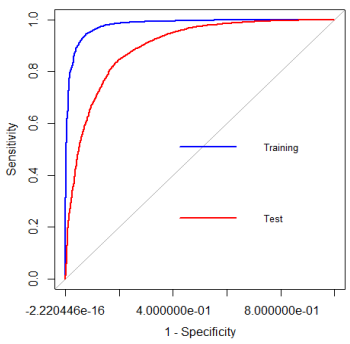
Afinn Dictionary						Training Dataset			Test Dataset		
Model	Ntree	Mtry	Split Rule	OOB	Threshold	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Model 1	100	23	gini	0.10	0.68	0.93	0.98	0.93	0.85	0.92	0.88
					0.50	0.94	0.95	0.98	0.87	0.88	0.95
Model 2	500	23	gini	0.10	0.69	0.93	0.98	0.94	0.85	0.92	0.87
					0.50	0.94	0.95	0.98	0.87	0.88	0.95
Model 3	1000	23	gini	0.10	0.67	0.94	0.97	0.94	0.85	0.92	0.89
					0.50	0.95	0.95	0.98	0.87	0.88	0.95



ROC Curve for Model 1



ROC Curve for Model 2



ROC Curve for Model 3

Combination of all three Dictionaries –

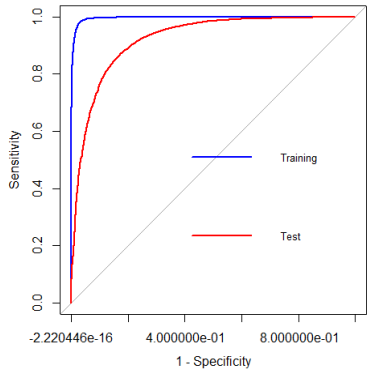
- Before creating models for combined dictionary, we initiated the process to create the dataset that was supposed to be used for modeling.
- We combined the matched word dataset of Bing Liu dictionary, Nrc dictionary and Afinn dictionary into single matched dataset.
- This matched dataset contained many duplicates as a single word could have multiple sentiments from different dictionaries.
- After removing the duplicates, the dataset had 1894 distinct words. This dataset was then used by us to create the document term matrix.
- The document term matrix had 33049 rows and 1896 columns (Review Id + Star Rating + 1894 words). After removing the 3-star rating, the document term matrix was left with 28377 rows and 1896 columns.
- The table gives the distribution of the words in the hiLo binary classifier target variable. The 6944 words with a hiLo value of -1 belong to the star rating of 1 or 2 while the 21433 words with a hiLo value of 1 belong to the star rating of 4 or 5.

hiLo value	Number of Words
-1	6944
1	21433

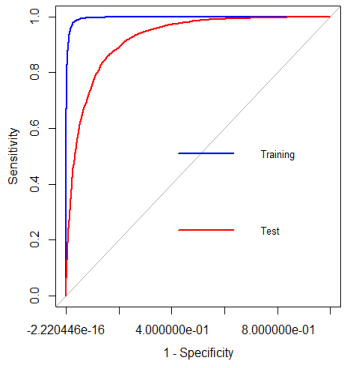
Random Forest Model:

The below table and graph summarize the input and output parameters for the multiple runs we made for the random forest model. We experimented with different combinations of ntree and mtry. The parameters that we have considered for performance evaluation are accuracy, precision, recall and receiver operating characteristics ROC.

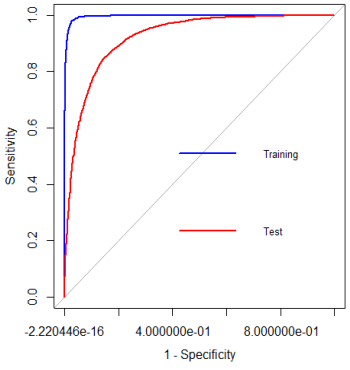
Combination of all Dictionaries						Training Dataset			Test Dataset		
Model	Ntree	Mtry	Split Rule	OOB	Threshold	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Model 1	100	43	gini	0.09	0.64	0.98	0.99	0.98	0.88	0.92	0.92
					0.50	0.98	0.98	1.00	0.88	0.89	0.97
Model 2	500	43	gini	0.09	0.65	0.98	0.99	0.98	0.88	0.93	0.91
					0.50	0.98	0.98	1.00	0.88	0.89	0.96
Model 3	1000	43	gini	0.09	0.64	0.98	0.99	0.98	0.88	0.92	0.92
					0.50	0.98	0.98	1.00	0.88	0.89	0.97



ROC Curve for Model 1



ROC Curve for Model 2



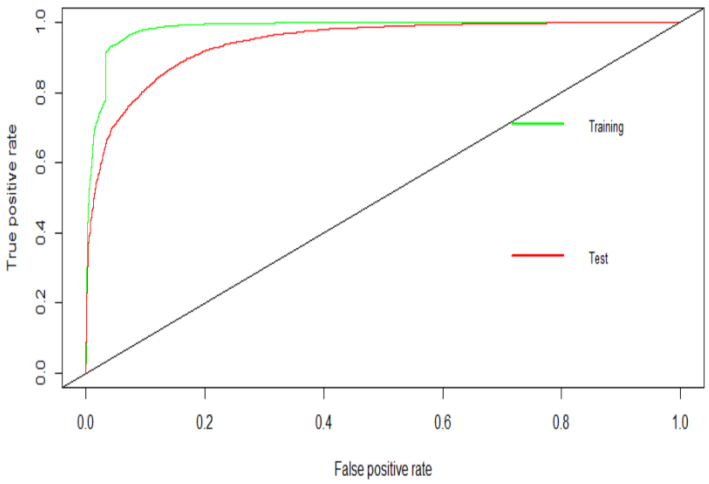
ROC Curve for Model 3

In a similar manner, we have created **Naïve Bayes and SVM** models on all 3 dictionaries and a combined dictionary and below are the results.
For Naïve Bayes, we have tried with and without Laplace smoothening and for SVM, we did radial kernels with different values of cost and gamma factors.

Dictionary used	Model	Parameters	Training Accuracy	Testing Accuracy	AUC Value
Bing Dictionary	Naïve Bayes	With Laplace Correction	52.80%	52.78%	71.49%
		Without Laplace Correction	51.80%	52.83%	71.26%
	Support Vector Machine (SVM) Radial	Cost = 1 , Gamma = 2	92.06%	88.52%	80.08%
		Cost = 10 , Gamma = 0.5	92.50%	89.01%	81.80%
		Cost = 10 , Gamma = 1	94.07%	88.67%	82.10%
NRC Dictionary	Naïve Bayes	With Laplace Correction	43.71%	44.04%	68.56%
		Without Laplace Correction	43.01%	44.04%	68.47%
	Support Vector Machine (SVM) Radial	Cost = 1 , Gamma = 2	92.52%	87.48%	79.67%
		Cost = 10 , Gamma = 0.5	93.08%	85.83%	80.60%
		Cost = 10 , Gamma = 1	95.37%	87.21%	80.18%
AFINN Dictionary	Naïve Bayes	With Laplace Correction	68.16%	68.61%	73.16%
		Without Laplace Correction	68.16%	68.61%	73.16%
	Support Vector Machine (SVM) Radial	Cost = 1 , Gamma = 2	88.83%	86.82%	76.47%
		Cost = 10 , Gamma = 0.5	89.33%	87.29%	78.40%
		Cost = 10 , Gamma = 1	90.56%	87.48%	79.01%
Combined Dictionary	Naïve Bayes	With Laplace Correction	48.46%	49.12%	70.06%
		Without Laplace Correction	48.34%	49.41%	70.08%
	Support Vector Machine (SVM) Radial	Cost = 1 , Gamma = 2	95.25%	88.99%	80.24%
		Cost = 10 , Gamma = 0.5	95.78%	89.58%	83.45%
		Cost = 10 , Gamma = 1	97.47%	89.27%	82.89%

Best Model among all 3 dictionaries and a combined one:

After all the modelling by Random forest, Naïve Bayes and SVM, we concluded that SVM radial as our best model. This has resulted for combined dictionary.



We have rejected Naïve Bayes model as it performs very poorly on our Yelp dataset. No combination has resulted in accuracy more than 60% except for Afinn dictionary where accuracy = 68%.

Also, we have rejected Random forest though its accuracy was good. This is because it results in overfitting by providing 100% accuracy on training for all different combinations of mtry and ntrees.

Do we use term frequency, tfidf, or other measures, and why?

For this question, we have used tfidf. Tf-idf works on the approach that high frequency words may not be able to provide much information gain thus indicating that rare words contribute more weighs to the model. We have used it over other weighing factors as it weighs down the frequent terms while scaling up the rare ones hence generating a fully comprehensive list of words in text mining.

Should we use stemming when using the dictionaries?

Stemming usually refers to a process that chops off the ends of words while lemmatization usually refers to doing things properly with the use of a vocabulary normally targeting to remove inflectional endings only and to return the dictionary form of a word. As stemming reduces the root words, we have not used stemming along with dictionaries as we might loose out on getting the sentiment of a word from the dictionary as the word will not be present in its root form in the Yelp data after performing stemming.

Hence, we should not use stemming as it will result in inaccurate results if we are using a dictionary.

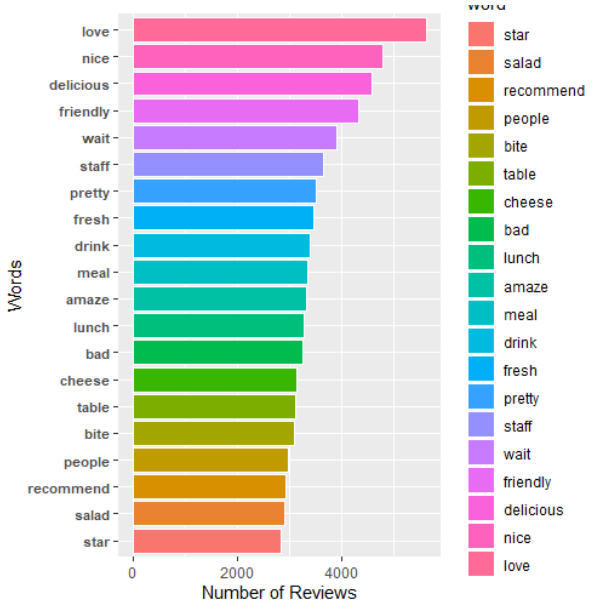
(ii) Develop models using a broader list of terms (i.e. not restricted to the dictionary terms only) – how do you obtain these terms? Will you use stemming here? Report on performance of the models. Compare performance with that in part (c) above. How do you evaluate performance? Which performance measures do you use, why?

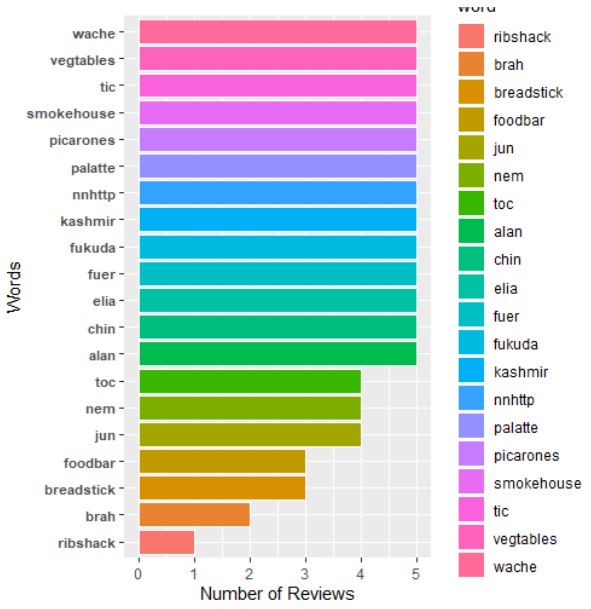
Solution:

Before we could commence with the model creation, there are a few general steps that we followed to obtain the dataset to run the model. The next section describes the preprocessing steps.

Methodology to Create Dataset Before Modeling

- We began with counting the number of reviews each word occurs in. We then plotted the most and least frequently occurring words in all the reviews together by arranging the words in descending order of the count for the number of reviews they occur in.





Words that occur in Max reviews **Words that occur in Min reviews**

- We then removed the words that occur in greater than 4500 reviews or lesser than 30 reviews. Before removing these words, we had 6847 words and after removing, we had 3236 distinct words left.
- We performed the left join of reduced words with the original dataset to fetch all the corresponding details of the reduced words dataset from the original dataset. This join helped us retain only the reduced words along with all the associated information. After performing this join, we had 777664 rows and 8 columns with 3236 distinct words.
- The most important step before creating any model is to generate the document term matrix. It gives us the frequency of each word (token) present in the review (document). We created the document term matrix wherein the words from the reduced words dataset appeared as column in the matrix along with two other columns viz. Review Id and Star Rating. The cell values were

taken from the tfidf column. The document term matrix contained 33436 rows and 3238 columns (Review Id + Star Rating + 3236 words)

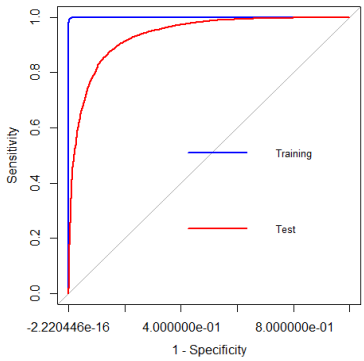
- As we aimed to consider only the positive and negative rating, we filtered out the star rating 3 as it tends to be a neutral rating. After removing the 3-star rating, we were left with 28726 rows and 3238 columns.
- We then created a new column in the matrix called 'hiLo'. It is a factor variable with values 1 and -1. 1 is assigned to hiLo when the star rating is greater than or equal to 4 and -1 is assigned to hiLo when the star rating is less than or equal to 2. Thus, this column partitions the words into binary categories.
- Before we could split the data into training and test dataset, we replaced all the NA's with zeroes in the matrix. Now the document term matrix is in the ready to use form.
- We then split the matrix into training and test data with 50:50 proportion.
- The table gives the distribution of the words in the hiLo binary classifier target variable. The 7031 words with a hiLo value of -1 belong to the star rating of 1 or 2 while the 21695 words with a hiLo value of 1 belong to the star rating of 4 or 5.

hiLo value	Number of Words
-1	7031
1	21695

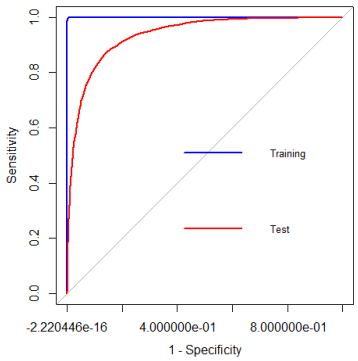
Random Forest on broader set of words:

The below table and graph summarize the input and output parameters for the multiple runs we made for the random forest model. We experimented with different combinations of ntree and mtry. The parameters that we have considered for performance evaluation are accuracy, precision, recall and receiver operating characteristics ROC.

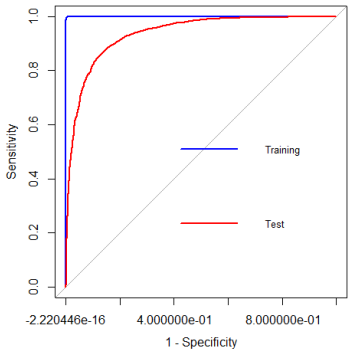
Broader set of Terms						Training Dataset			Test Dataset		
Model	Ntree	Mtry	Split Rule	OOB	Threshold	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Model 1	100	56	gini	0.09	0.62	1.00	1.00	0.99	0.89	0.93	0.92
					0.50	0.99	0.99	1.00	0.89	0.89	0.97
Model 2	500	56	gini	0.09	0.61	1.00	1.00	1.00	0.89	0.93	0.93
					0.50	0.99	0.99	1.00	0.89	0.89	0.96
Model 3	1000	56	gini	0.09	0.59	1.00	1.00	1.00	0.89	0.92	0.93
					0.50	1.00	0.99	1.00	0.89	0.89	0.97



ROC Curve for Model 1



ROC Curve for Model 2



ROC Curve for Model 3

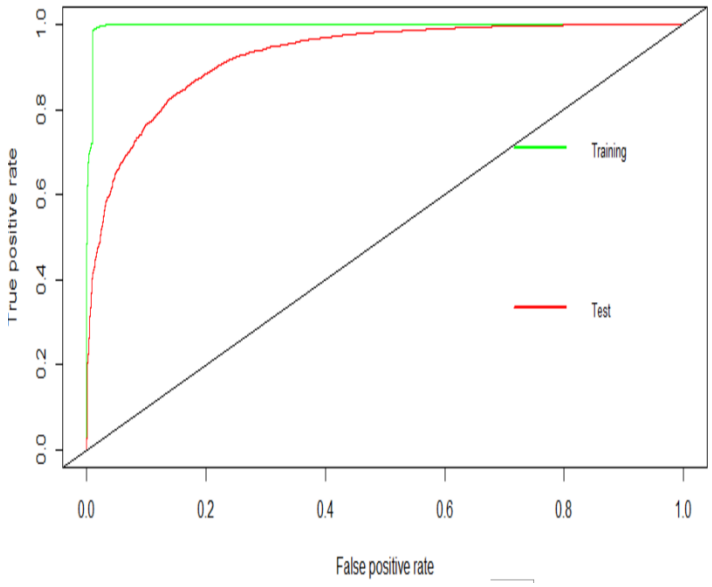
We have then performed all different models after random forest and tabulated the results as below:

Dictionary used	Model	Parameters	Training Accuracy	Testing Accuracy	AUC Value
Broader set of words	Naïve Bayes	With Laplace Correction	41.24%	42.66%	69.10%
		Without Laplace Correction	41.24%	42.66%	69.10%
	Support Vector Machine (SVM) Radial	Cost = 1 , Gamma = 2	99.12%	86.41%	73.65%
		Cost = 10 , Gamma = 0.5	99.71%	90.64%	85.75%
		Cost = 10 , Gamma = 1	100.00%	90.15%	83.76%
		Cost = 50 , Gamma = 1	100.00%	90.08%	83.63%

Will we use stemming here?

While creating models for broader set of terms, we wish to group words by their star rating and the number of reviews they occur in to categorize them as 1 or -1 in the hiLo column. But if we perform stemming, we will insensibly end up grouping the words by their word stem and will hence end up losing several words which will all convert into the same word stem. Thus, we are performing lemmatization instead of stemming for this question as well.

Overall best model:



On comparing the results from above three models for all the four dictionaries (Bing, Afinn, NRC and Combined) and Broader set of variables, we observe that the test accuracy values for Combined Dictionary and Broader Set of variables from models Random Forest and SVM are in the range of 87-89% (which is the highest). The training accuracies are as high as 96%. Since the test accuracy for SVM is the highest (89.58%) and AUC value is 83.45%, we take **SVM with kernel "Radial", Cost = 10 and Gamma = 0.5 values to be the best model.** ROC curve with AUC = 83.45% for the best model is shown on left.