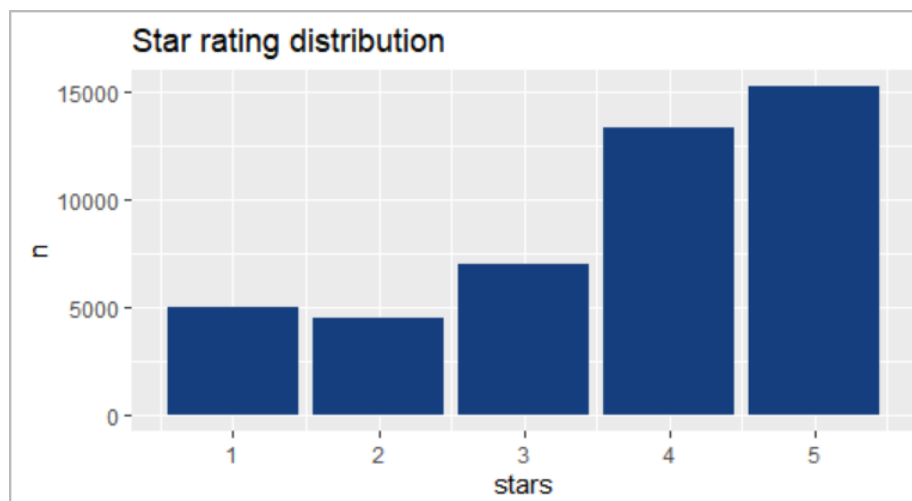


## Text mining, Sentiment Analysis

*Explore the data.*

**Distribution of Star Ratings:**

stars	n
1	4953
2	4516
3	6999
4	13306
5	15226



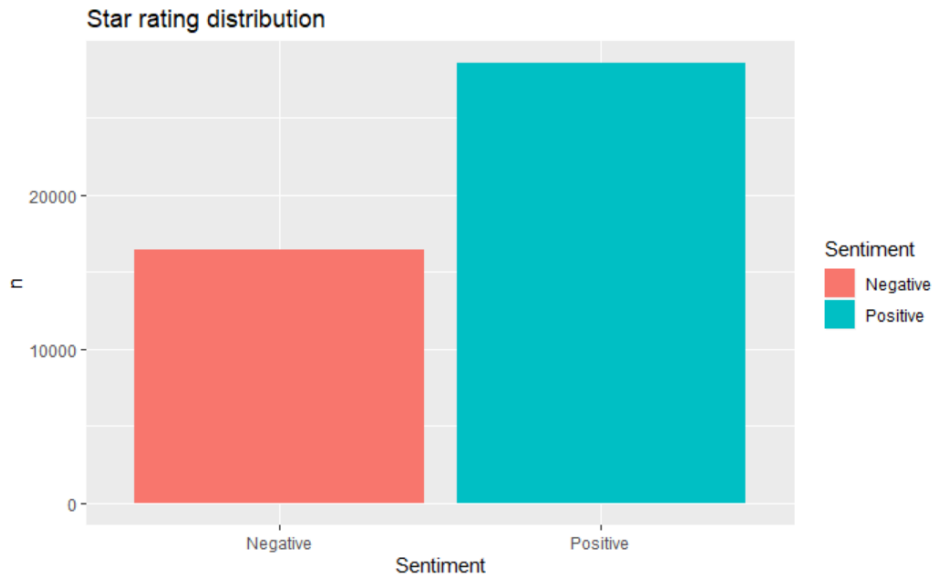
Highest rating is 5 which can be indicated as “Very Good”. Similarly 4 indicates “Good” and so on. Rating 1 indicates “Not Satisfied”. From the above plot, we see that the number of reviews are maximum for star rating 4 and 5. Number of ratings recorded is least for star rating 2.

**Assigning a label to the star ratings to indicate a positive or negative review as follows:**

- Star-ratings 4,5 = positive;
- Star-ratings 1,2 = negative
- Star-rating 3 is dropped since it is a neutral review which doesn't add much to the model prediction.

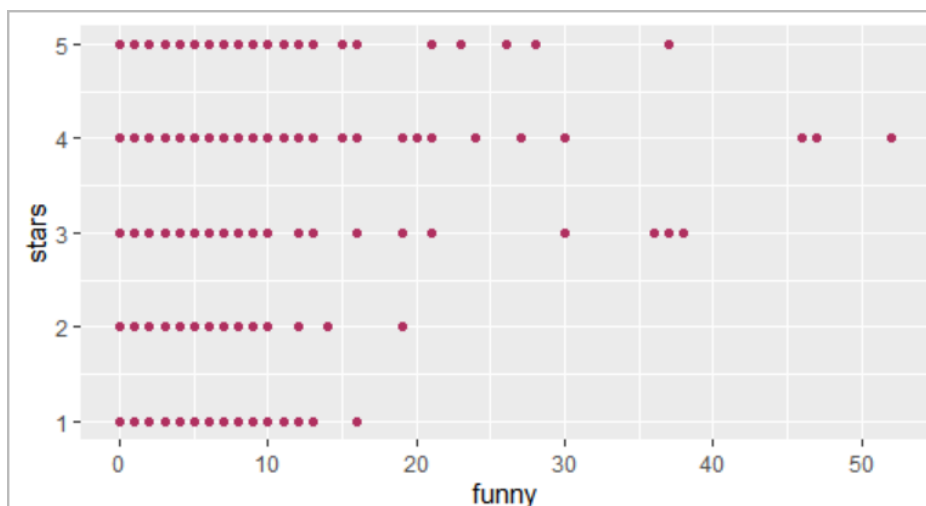
The distribution of positive and negative sentiments in the dataset:

Sentiment <chr>	n <int>
Negative	16468
Positive	28532



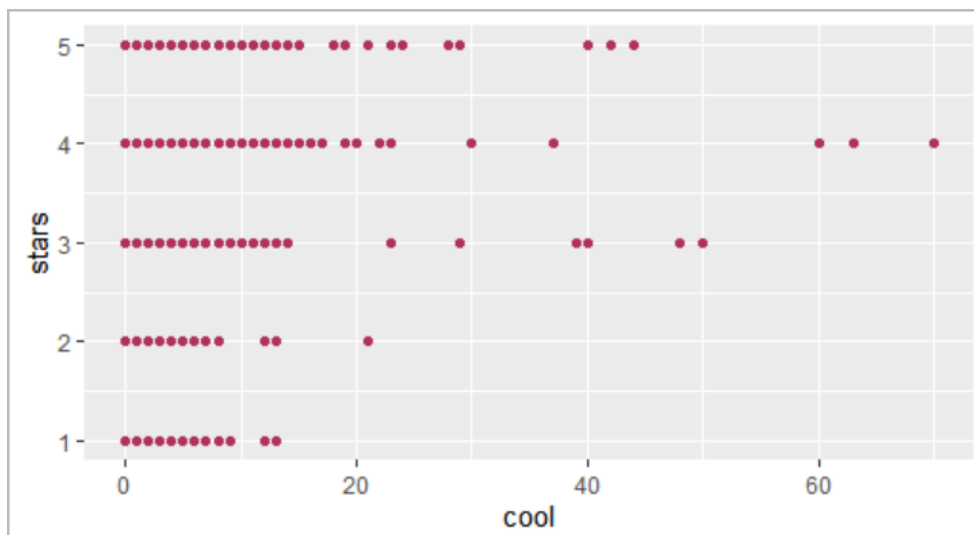
We use `rbind` to assign the sentiment values/moods to words which are not present in the dictionaries, but present in the data (reviews). Example: shrimp, tastes, etc

### Distribution of word “Funny” in the different star rated reviews:



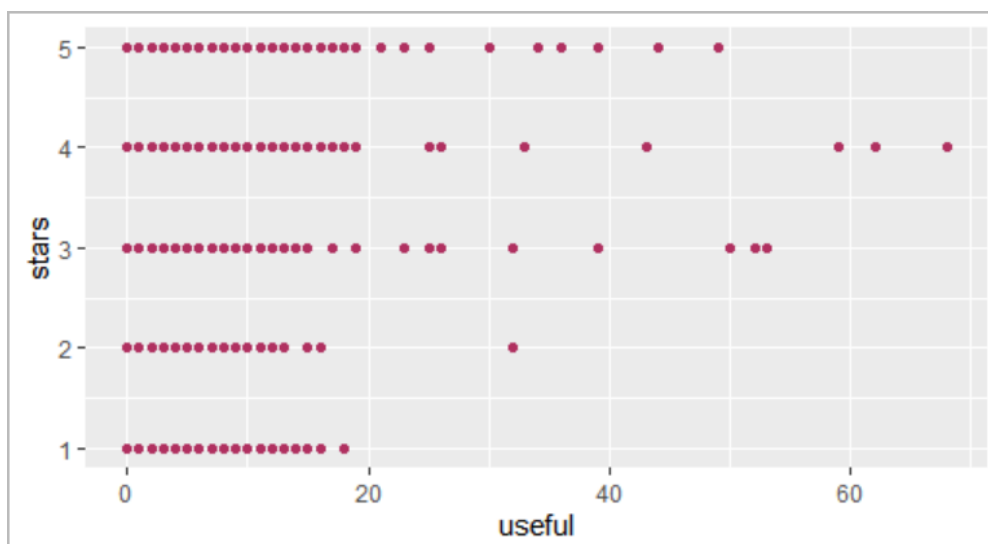
From the above distribution we could say that occurrence of “Funny” is higher in ratings such as 3,4 and 5, which implies positive sentiment to the word ‘Funny’.

### Distribution of word “Cool” in the different star rated reviews:



Occurance of “Cool” is also higher in ratings such as 3,4 and 5 which is expected and implies a positive sentiment.

#### Distribution of word “Useful” in the different star rated reviews:



Useful is slightly higher in star-ratings of 4 and 5 compared to 1,2 and 3. We see a few outliers, however on the whole, we see an approximately equal distribution among all the star-ratings.

#### Additional Data exploration:

To check the top 10 cities that provide ratings:

state <fctr>	n <int>
NV	11258
AZ	11207
ON	7614
NC	3083
OH	2587
PA	2380
QC	2375
BW	1348
EDH	1345
WI	1109

We observe the states Nevada, Arizona and Ontario, Canada are responsible for the highest number of reviews.

**To check the top 10 postal codes that provide ratings:**

postal_code <fctr>	n <int>
89109	2982
89146	930
85251	770
89119	556
53703	539
85281	528
89117	521
28202	488
89139	470
89123	462

Noting that this has been done only for 5-digit postal codes. The highest number of ratings is from postal code 89109 which is Las Vegas Valley , Nevada, second highest being from Clark County, Las Vegas, Nevada and so on. Hence using this we get to analyse in depth observation from each country of a city.

The data was cleansed by the method of tokenizing, removing stopwords, most occurring words, removing rare words such as those not present in at least 10 reviews and words starting with or including numbers.

Total number of stopwords which were removed from our data is 701.

We use word frequency to list the most frequently occurring words or concepts in a given text. Below are the top 10 words which occur frequently in the document.

	word	n
1	food	23928
2	service	11755
3	time	8910
4	chicken	7331
5	restaurant	6852
6	nice	5680
7	pizza	5498
8	delicious	5287
9	love	5243
10	menu	5073

From the above table, we observe that more frequently occurring words like food, service, time, chicken etc. do not have any sentiments associated with it. So these words can be eliminated and we are left with 11,58,457 observations.

We can also find out the rare words which are present in less than 10 reviews and eliminate them. In our dataset, there are 58,651 words which could be eliminated as they appear very rarely in the document. After removing all the rare words, we are left with 10,49,664 observations.

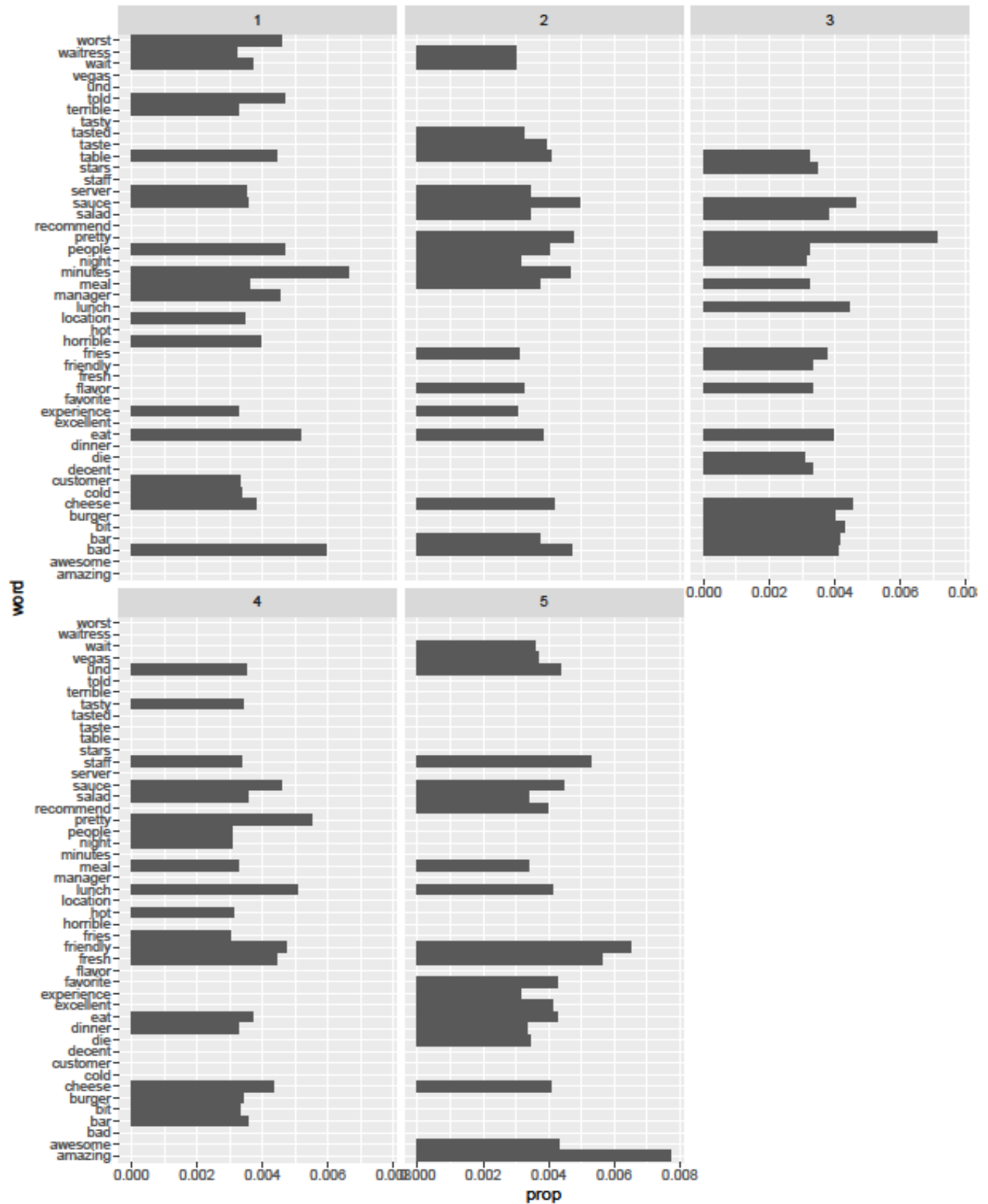
We could also remove the words that start with numbers as most likely they are not associated with any sentiment. After removing those words, we are left with 10,20,918 observations.

We calculate the proportion of words by:

**Proportion of a word (A) = no:of occurrences of A/ sum of number of occurrence of all words.**

Now, we move to analyze words by star ratings

Proportion of top 20 words in each star rating is shown in below plot:



Based on the above plot, we can say that for star rating 5, most frequent words are amazing, friendly, fresh. Similarly,

For star rating 4, most frequent words are - pretty, lunch, friendly. fresh.






For star rating 3, most frequent words are pretty, sauce.

For star rating 2, most frequent words are bad, sauce.

For star rating 1, most frequent words are minutes,bad.

Now, to observe words related to higher or lower ratings by calculating average star rating by each word. This is done by summing the star ratings associated with each word and observing the proportions of these words in each star rating. totWS is the average star rating associated with each word.

Below, we observe the top 20 words related to highest and the lowest star ratings

20 words related to highest star ratings			20 words related to lowest star ratings						
		word				word		totWS	
1		amazing			1	arguing		0.0001226257	
2		bar			2	blech		0.0001251450	
3		burger			3	bullshit		0.0001178461	
4		cheese			4	coffe		0.0001219128	
5		die			5	disgust		0.0001087735	
6		dinner			6	disrespectful		0.0001088625	
7		eat			7	dolmas		0.0001264276	
8		fresh			8	mask		0.0001202078	
9		friendly			9	neven		0.0001195511	
10		fries			10	nnwas		0.0001233835	
11		lunch			11	patronizing		0.0001178664	
12		meal			12	recieved		0.0001218238	
13		night			13	rubio's		0.0001261933	
14		people			14	santi		0.0001146341	
15		pretty			15	tipping		0.0001242744	
16		salad			16	understands		0.0001259794	
17		sauce			17	unedible		0.0001161614	
18		staff			18	unwilling		0.0001111352	
19		und			19	useless		0.0001227631	
20		wait			20	vehicle		0.0001259794	

Further, we calculated the mean of average star rating of each word, i.e 0.0017353. All the words below this mean value are considered to be negative sentiment and more than this mean value are considered to be positive sentiment. For example, friendly, pretty, amazing, fresh makes sense to be associated with positive sentiment. Similarly, for negative sentiment, words like disgust, disrespectful, useless, bulshit etc. makes sense.

Below tables show few words associated with negative or positive sentiment.

▲	word	totWS	Sentiment	▲	word	totWS	Sentiment
1	sauce	0.06809084	Positive	1	disgust	0.0001087735	Negative
2	friendly	0.06697523	Positive	2	disrespectful	0.0001088625	Negative
3	pretty	0.06643491	Positive	3	unwilling	0.0001111352	Negative
4	cheese	0.06347256	Positive	4	santi	0.0001146341	Negative
5	lunch	0.06243453	Positive	5	unedible	0.0001161614	Negative
6	eat	0.06067223	Positive	6	bullshit	0.0001178461	Negative
7	fresh	0.05989732	Positive	7	patronizing	0.0001178664	Negative
8	staff	0.05696736	Positive	8	neven	0.0001195511	Negative
9	amazing	0.05627503	Positive	9	mask	0.0001202078	Negative
10	salad	0.05240525	Positive	10	recieved	0.0001218238	Negative
11	und	0.05190426	Positive	11	coffe	0.0001219128	Negative
12	bar	0.05125994	Positive	12	arguing	0.0001226257	Negative
13	meal	0.05081960	Positive	13	useless	0.0001227631	Negative
14	people	0.04968037	Positive	14	nnwas	0.0001233835	Negative
15	wait	0.04895535	Positive	15	tipping	0.0001242744	Negative

We analyse the sentiment of extracted words from the reviews with the help of the three dictionaries:

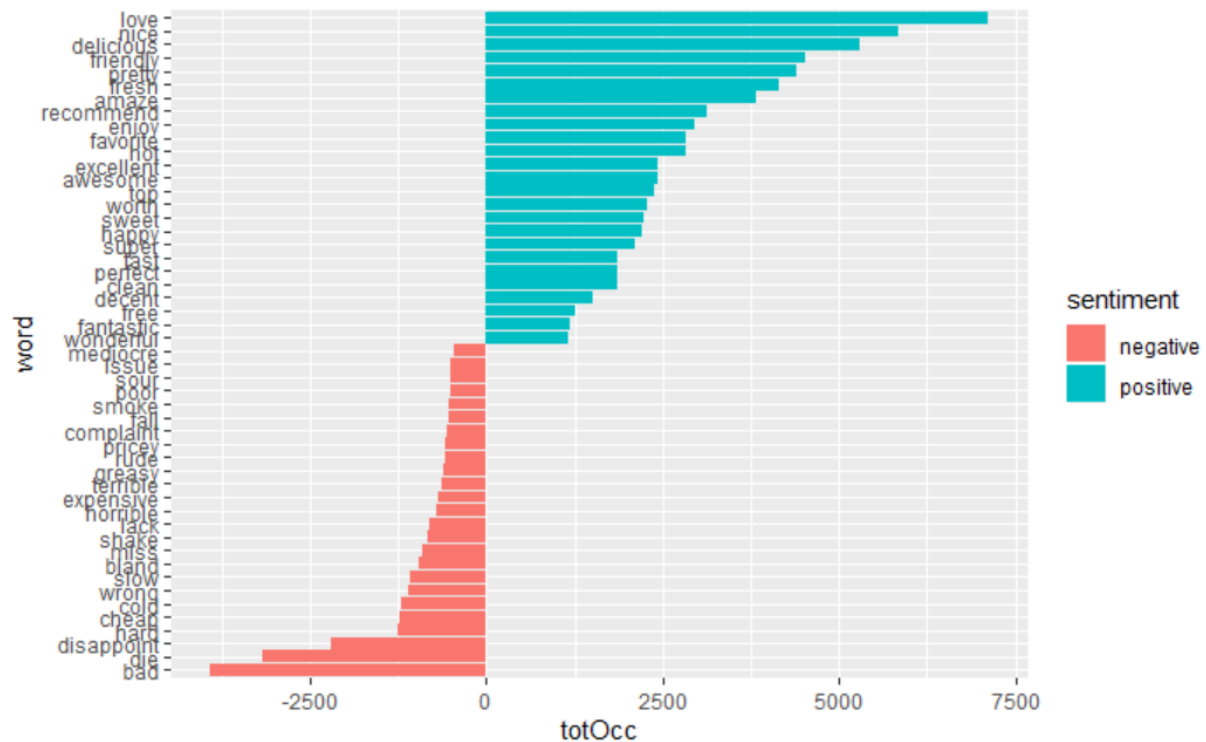
### 1)Prof Bing Liu Dictionary:

This dictionary assigns only 2 sentiments to words “positive” and “negative”

For this dictionary, out of 6964 unique words, only 935 words had a sentiment match with 437 unique words having a positive sentiment and 498 unique words having a negative sentiment.

Below is a graph representing the highest occuring words with respective sentiments matched as per the Bing Liu dictionary:





## 2) NRC dictionary

The NRC dictionary has eight basic emotions and 2 sentiments as follows:

**Emotions:** anger, anticipation, disgust, fear, joy, sadness, surprise, trust.

**Sentiments:** Positive and negative

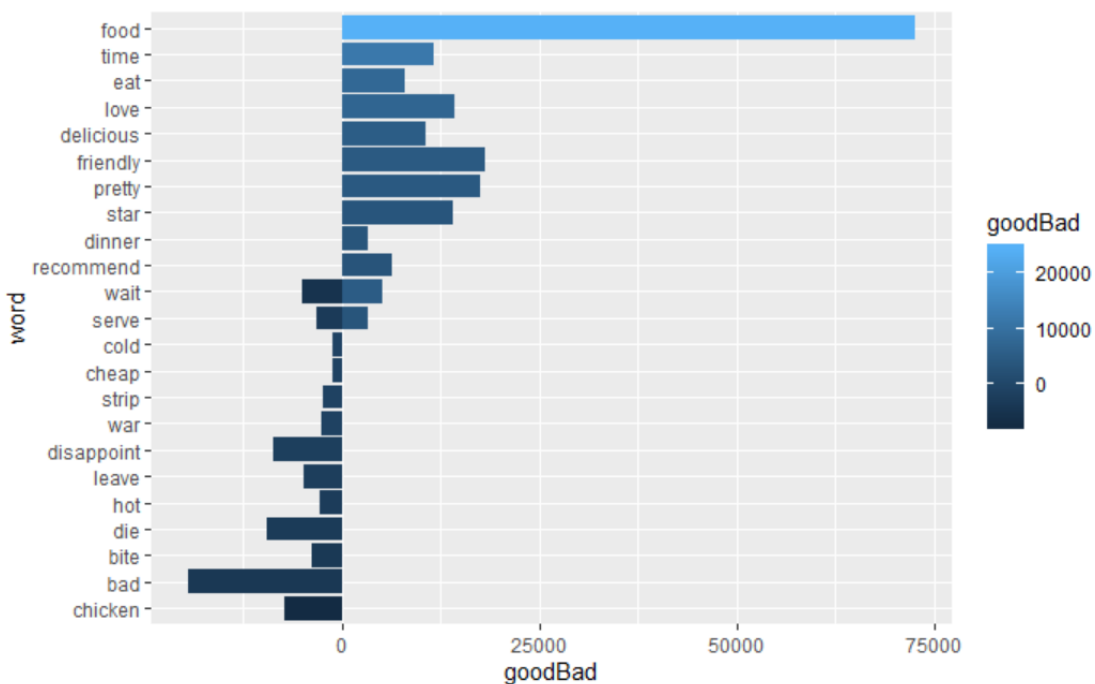
With this dictionary, we had a total of 2831 unique words assigned with a sentiment or an emotion out of a total of 6964 unique words from the document. The Details are as follows:

Sentiment	Count of Unique Words assigned to this sentiment
positive	636
negative	499

Emotion	Count of Unique Words
---------	-----------------------

	assigned to this emotion
anger	186
disgust	164
fear	201
sadness	182.
joy	237
anticipation	249
trust	332
surprise	145

Suppose we consider 'anger, disgust, fear sadness, negative' to denote 'bad' reviews, and 'positive, joy, anticipation, trust' to denote 'good' reviews, then the following plot shows the top words representing good and bad reviews:

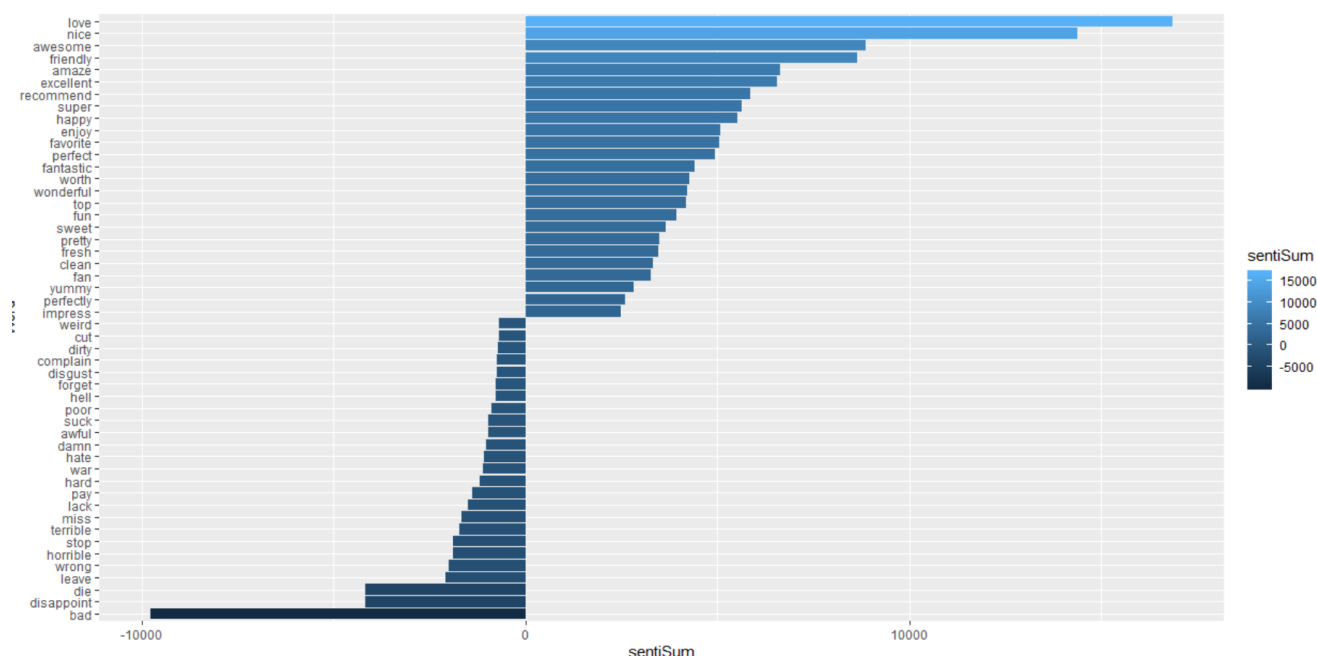


### 3) AFINN Dictionary:

As per this dictionary, each word would be associated with a positivity score from -5 to +5.

With this dictionary, we had only 518 unique words assigned with a score out of a total of 6964 unique words from the document.

For each unique word, we create a numeric value SentiSum which is the product of its associated score and the frequency of its occurrence. The below graph shows the top words with positive and negative SentiSum scores.



The above analysis gives overall sentiment by the words, now we would further analyse sentiment by reviews and to check how the associated sentiment would relate with the review's star ratings.

## 1) Bing Liu Dictionary:

To analyse sentiment by review as per Bing Liu dictionary , we count the number of total positive words (posSum) and negative words (negSum) in each review. Further, using posSum and negSum , we obtain the proportion of the positive words and negative words in each review as follows:

$$\text{posProp} = \text{posSum} / \text{nwords}$$

$$\text{negProp} = \text{negSum} / \text{nwords}$$

where, nwords = total number words identified with a sentiment as per Bing Liu dictionary in each review.

Finally, we compute the SentiScore as  $\text{posProp} - \text{negProp}$  for each review. We compare this SentiScore to the star-rating of each review. The below table shows the average positive score, average negative score and average SentiScore for each star ratings:

stars <int>	avgPos <dbl>	avgNeg <dbl>	avgSentiSc <dbl>
1	0.2925157	0.7074843	-0.4149685
2	0.4464773	0.5535227	-0.1070453
3	0.5964308	0.4035692	0.1928615
4	0.7373578	0.2626422	0.4747157
5	0.8109949	0.1890051	0.6219898

As observed from the above table, the star rating 1 has an average SentiScore of -0.414 and the SentiScore gradually increases with the star-rating.

## 2) NRC Dictionary

Since the NRC dictionary has 8 emotions and 2 sentiments, we would map the sentiments and the emotions of each unique word in the document to only 2 sentiments 'positive' and 'negative' as follows:

The emotions 'joy, anticipation, trust, surprise' and sentiment 'positive' would be mapped to a single sentiment 'positive'.

Similarly, the emotions, 'anger, disgust, fear, sadness' and sentiment 'negative' would be mapped to a single sentiment 'negative'.

The below table shows the count of unique words after the above mapping process

sentiment <chr>	n <int>
negative	1232
positive	1599

Now we use the same process as the Bing Liu dictionary to get the average SentiScore for each review. The below table shows the average positive score, average negative score and average SentiScore for each star rating:

stars <int>	avgPos <dbl>	avgNeg <dbl>	avgSentiSc <dbl>
1	0.5352798	0.4647202	0.07055952
2	0.6294592	0.3705408	0.25891832
3	0.6951384	0.3048616	0.39027680
4	0.7645497	0.2354503	0.52909940
5	0.8044353	0.1955647	0.60887062

As observed from the above table, the star rating 1 has the lowest average SentiScore of 0.0705 and the SentiScore gradually increases with the star-rating.

## 3) AFINN dictionary

For the AFINN dictionary, we have a value associated with each word ranging from -5 to +5. Hence, for sentiment analysis by each review we would obtain the SentiScore as the sum of the values of all the words present in each review.

The below table shows the average SentiScore by each star rating:

stars <int>	avgLen <dbl>	avgSenti <dbl>
1	4.003408	-2.3476285
2	4.313061	0.7138584
3	4.314566	3.1422384
4	4.246564	5.5466212
5	3.963570	6.4570960

As expected, the average SentiScore increases from the lowest -2.347 corresponding to star-rating 1 to a high 6.457 corresponding to a star-rating 5.

#### **To Predict Review Sentiments Based on the Calculated Aggregate Scores:**

We assign a sentiment to each review in numeric form based on the star-rating as follows:

1. Reviews with 1 to 2 stars = -1 (indicating negative sentiment )
2. Reviews with 4 to 5 stars = +1 (indicating positive sentiment)
3. Reviews with 3 stars are filtered out.

The above is the “Actual Classification”

Using the calculated Senti Scores based on the dictionaries, we assign sentiment to each review in numeric form as follows:

1. Reviews with SentiScore > 0 = +1 (indicating positive sentiment)
2. Reviews with SentiScore < 0 = -1 (indicating negative sentiment)
3. Reviews with SentiScore = 0 are filtered out.

The above is the “Predicted Classification”

Observing the Confusion Matrix and Accuracy with each dictionary as follows:

Dictionary	Bing Liu Dictionary	NRC Dictionary	AFINN Dictionary																																																
Confusion Matrix	<table> <tr> <td></td><td colspan="3">predicted</td></tr> <tr> <td>actual</td><td>-1</td><td>1</td><td></td></tr> <tr> <td>-1</td><td>5099</td><td>1570</td><td></td></tr> <tr> <td>1</td><td>3534</td><td>17269</td><td></td></tr> </table>		predicted			actual	-1	1		-1	5099	1570		1	3534	17269		<table> <tr> <td></td><td colspan="3">predicted</td></tr> <tr> <td>actual</td><td>-1</td><td>1</td><td></td></tr> <tr> <td>-1</td><td>2500</td><td>4442</td><td></td></tr> <tr> <td>1</td><td>2063</td><td>19299</td><td></td></tr> </table>		predicted			actual	-1	1		-1	2500	4442		1	2063	19299		<table> <tr> <td></td><td colspan="3">predicted</td></tr> <tr> <td>actual</td><td>-1</td><td>1</td><td></td></tr> <tr> <td>-1</td><td>4203</td><td>2327</td><td></td></tr> <tr> <td>1</td><td>2535</td><td>17854</td><td></td></tr> </table>		predicted			actual	-1	1		-1	4203	2327		1	2535	17854	
	predicted																																																		
actual	-1	1																																																	
-1	5099	1570																																																	
1	3534	17269																																																	
	predicted																																																		
actual	-1	1																																																	
-1	2500	4442																																																	
1	2063	19299																																																	
	predicted																																																		
actual	-1	1																																																	
-1	4203	2327																																																	
1	2535	17854																																																	
Accuracy	0.814	0.77	0.819																																																

From the above, we see that Bing Liu and AFINN Dictionaries performed well with an accuracy of 81%.

### *Develop models to predict review sentiment.*

Develing models using only the sentiment dictionary terms using all the 3 dictionaries:

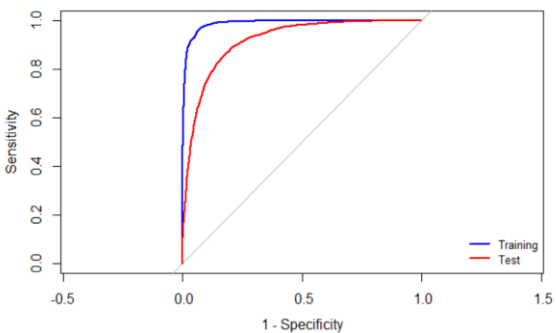
#### 1) Bing Liu Dictionary

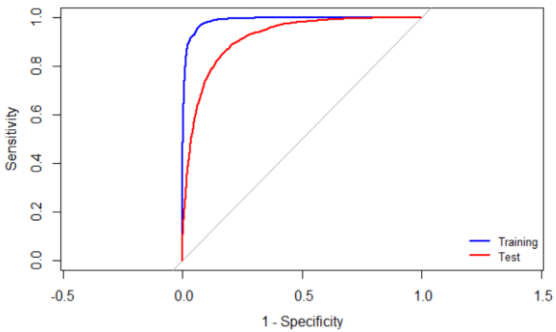
The final document matrix derived from the Bing Liu dictionary consists of 27472 entries after dropping star-rating 3 reviews from the dataset. The dataset was split in 50:50 ratio for the training and testing datasets. Hence both training and testing data each had 13736 entries with 937 variables.

##### a) Random Forest Model:

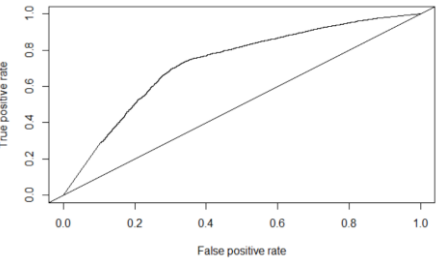
A random forest was run using the ranger package. The details of the random forest are as follows:

Type:	Probability estimation
Number of trees:	500
Sample size:	13736
Number of independent variables:	935
Mtry:	30
Target node size:	10
Variable importance mode:	permutation
Splitrule:	gini
OOB prediction error (Brier s.):	0.08985013

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
0.5	<p><b>Training Set:</b></p> <p>      preds</p> <p>actual FALSE TRUE</p> <p>-1 2883 426</p> <p>  1 135 10292</p> <p><b>Testing Set:</b></p> <p>      preds</p> <p>actual FALSE TRUE</p> <p>-1 2212 1148</p> <p>  1 551 9825</p>	<p><b>Training Accuracy: 95.91</b></p> <p><b>Testing Accuracy: 87.63</b></p>	 <p><b>AUC for training data: 0.9883</b> <b>AUC for testing data: 0.9161</b></p>

0.3	<b>Training Set:</b> preds actual FALSE TRUE -1 2157 1152 1 21 10406  <b>Testing Set:</b> preds actual FALSE TRUE -1 1543 1817 1 158 10218	<b>Training Accuracy: 91.46</b>  <b>Testing Accuracy: 85.62</b>	 <p>AUC for training data: 0.9883 AUC for testing data: 0.9161</p>
-----	--	---	--

## b) Naive Bayes

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 2952 617 1 5581 4546  <b>Testing Set:</b> preds actual FALSE TRUE -1 2847 713 1 5810 4366	<b>Training Accuracy: 0.545</b>  <b>Testing Accuracy: 0.52</b>	 <p>AUC for training data: 0.7035 AUC for testing data: 0.7282</p>

## c) SVM Model

The following models were run by using a radial kernel function and by varying the parameters:

Cost	Gamma	Training Confusion Matrix and Accuracy	Testing Confusion Matrix and Accuracy
------	-------	--	---------------------------------------

1	0.1	Training: <pre> predicted actual  -1    1       -1  963 2346        1   66 10361           </pre> <b>Training Accuracy: 0.824403</b>	Testing: <pre> predicted actual  -1    1       -1  937 2423        1   69 10307           </pre> <b>Testing Accuracy: 0.8185789</b>
5	0.5	Training: <pre> predicted actual  -1    1       -1 2301 1008        1  248 10179           </pre> <b>Training Accuracy: 0.908561</b>	Testing: <pre> predicted actual  -1    1       -1 2209 1151        1  346 10030           </pre> <b>Testing Accuracy: 0.8910163</b>
NIL	NIL	Training: <pre> predicted actual  -1    1       -1   0 3309        1   0 10427           </pre> <b>Training Accuracy: 0.759100</b>	Testing: <pre> predicted actual  -1    1       -1   0 3360        1   0 10376           </pre> <b>Testing Accuracy: 0.7553873</b>

From the above, we observe that a radial basis kernel with cost =5 and gamma = 0.5 provides a good model with 89% accuracy.

## 2) NRC dictionary

The final document matrix derived from the NRC dictionary consists of 28,304 entries and 1310 variables after dropping star-rating 3 reviews from the dataset. The count of +1 and -1 classifications were as follows:

hiLo <dbl>	n <int>
-1	6942
1	21362

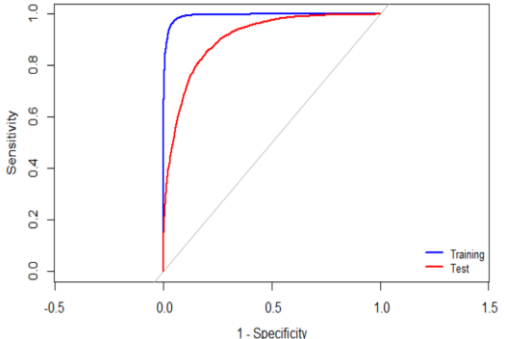
The dataset was split in 50:50 ratio for the training and testing datasets. Hence both training and testing data each had 14152 entries with 1310 variables.

### a) Random Forests:



A random forest was run using the ranger package. The details of the random forest are as follows:

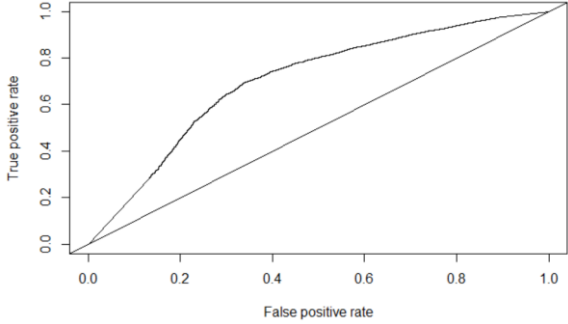
Type: Probability estimation  
 Number of trees: 500  
 Sample size: 14152  
 Number of independent variables: 1308  
 Mtry: 36  
 Target node size: 10  
 Variable importance mode: permutation  
 Splitrule: gini  
 OOB prediction error (Brier s.): 0.1009294

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 3094 395 1 57 10606  <b>Testing Set:</b> preds actual FALSE TRUE -1 1219 2234 1 96 10603	<b>Training Accuracy:96.80</b>  <b>Testing Accuracy:86.87</b>	 <p>AUC for training data: 0.993            AUC for testing data: 0.905</p>

## b) Naive Bayes

A threshold probability of 0.5 was used.

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
-----------------------	------------------	----------	-------------------

0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 2971 776 1 5444 4961  <b>Testing Set:</b> preds actual FALSE TRUE -1 2925 820 1 5922 4485	<b>Training Accuracy: 0.56</b>  <b>Testing Accuracy: 0.524</b>	 <p><b>AUC for training data:0.6736</b>  <b>AUC for testing data: 0.7027</b></p>
-----	---	--	--

### c) SVM Model

The following models were run by using a radial kernel function and by varying the parameters:

Cost	Gamma	Training Confusion Matrix and Accuracy	Testing Confusion Matrix and Accuracy
1	0.1	Training: predicted actual -1 1 -1 968 2529 1 68 10587  <b>Training Accuracy: 0.816492</b> <b>Area under the curve: 0.6352</b>	Testing: predicted actual -1 1 -1 913 2532 1 94 10613  <b>Testing Accuracy: 0.8144432</b> <b>Area under the curve: 0.6281</b>
5	0.5	Training: predicted actual -1 1 -1 2417 1080 1 227 10428  <b>Training Accuracy: 0.907645</b> <b>Area under the curve: 0.8349</b>	Testing: predicted actual -1 1 -1 2160 1285 1 435 10272  <b>Testing Accuracy: 0.8784624</b> <b>Area under the curve: 0.7932</b>

NIL	NIL	Training: predicted actual -1 1 -1 0 3497 1 0 10655  <b>Training Accuracy: 0.752897</b> <b>Area under the curve: 0.5</b>	Testing: predicted actual -1 1 -1 0 3445 1 0 10707  <b>Testing Accuracy: 0.7565715</b> <b>Area under the curve: 0.5</b>
-----	-----	---	--

From the above, we observe that a radial basis kernel with cost=5 and gamma=0.5 provides a good model with 87% accuracy.

### 3) AFINN Dictionary

The final document matrix derived from the AFFINN dictionary consists of 26,919 entries and 520 variables after dropping star-rating 3 reviews from the dataset. The count of +1 and -1 classifications were as follows:

hiLo <dbl>	n <int>
-1	6530
1	20389

#### a) Random Forest

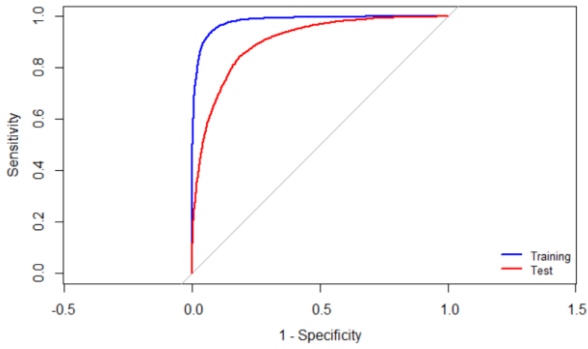
A random forest model was run using the ranger package. The details of the random forest are as follows:

```

Type:                                Probability estimation
Number of trees:                      500
Sample size:                          13460
Number of independent variables:      518
Mtry:                                 22
Target node size:                     10
Variable importance mode:             permutation
Splitrule:                            gini
OOB prediction error (Brier s.):      0.1020757

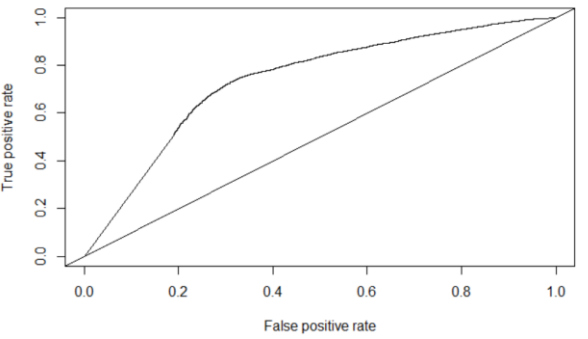
```

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
-----------------------	------------------	----------	-------------------

0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 2670 600 1 156 10034  <b>Testing Set:</b> preds actual FALSE TRUE -1 1294 1966 1 169 10030	<b>Training Accuracy:94.3833</b>  <b>Testing Accuracy:86.388</b>	 <p>Area under the curve Training: 0.9816 Area under the curve Testing : 0.9012</p>
-----	--	--	---

## b) Naive Bayes

A threshold probability of 0.5 was used.

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 2144 1126 1 2680 7510  <b>Testing Set:</b> preds actual FALSE TRUE -1 2195 1065 1 2603 7596	<b>Training Accuracy:71.72</b>  <b>Testing Accuracy:72.25</b>	 <p>AUC for training data:0.7235 AUC for testing data: 0.739</p>

## c) SVM Model

The following models were run by using a radial kernel function and by varying the parameters:

Cost	Gamma	Training Confusion Matrix and	TestingConfusion Matrix and
------	-------	-------------------------------	-----------------------------

		Accuracy	Accuracy
1	0.1	Training: predicted actual -1 1 -1 701 2569 1 63 10127  <b>Training Accuracy: 0.8044</b>  <b>Area under the curve: 0.6041</b>	Testing: predicted actual -1 1 -1 634 2626 1 77 10122  <b>Testing Accuracy: 0.799167</b>  <b>Area under the curve: 0.5935</b>
5	0.5	Training: predicted actual -1 1 -1 1958 1312 1 292 9898  <b>Training Accuracy: 0.88083</b>  <b>Area under the curve: 0.7851</b>	Testing: predicted actual -1 1 -1 1888 1372 1 389 9810  <b>Testing Accuracy: 0.86915</b>  <b>Area under the curve: 0.7705</b>
NIL	NIL	Training: predicted actual -1 1 -1 0 3270 1 0 10190  <b>Training Accuracy: 0.757057</b>  <b>Area under the curve: 0.5</b>	Testing: predicted actual -1 1 -1 0 3260 1 0 10199  <b>Testing Accuracy: 0.7577829</b>  <b>Area under the curve: 0.5</b>

From the above, we observe that a radial basis kernel with cost=5 and gamma=0.5 provides a good model with 86% accuracy.

#### 4) Combining all 3 Dictionaries: *(BING + NRC + AFINN)*

The final document matrix derived from the combining all the three dictionaries consists of 28,478 entries and 1749 variables after dropping star-rating 3 reviews from the dataset. The count of +1 and -1 classifications were as follows:

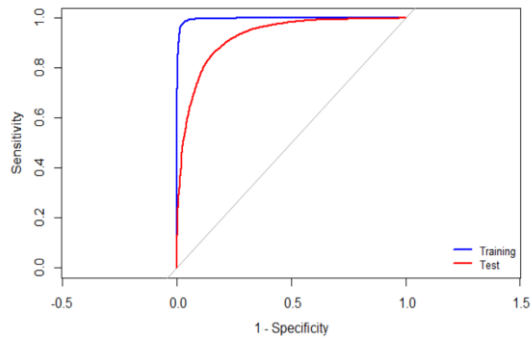
hiLo <dbl>	n <int>
-1	6968
1	21510

## a) Random Forests

A random forest was run using the ranger package. The details of the random forest are as follows:

```

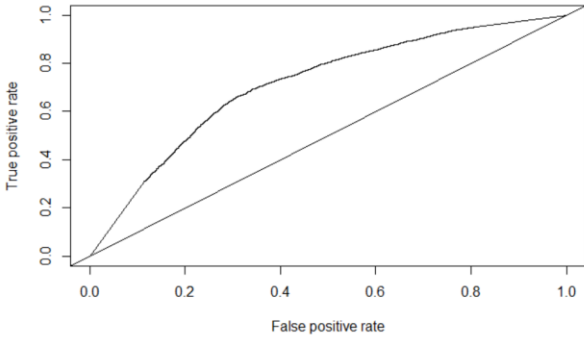
Type:                                Probability estimation
Number of trees:                      500
Sample size:                          14239
Number of independent variables:      1747
Mtry:                                 41
Target node size:                     10
Variable importance mode:              permutation
Splitrule:                            gini
OOB prediction error (Brier s.):      0.08994007
  
```

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
0.5	<p><b>Training Set:</b></p> <pre> preds actual FALSE TRUE -1 3208 275 1 42 10714           </pre> <p><b>Testing Set:</b></p> <pre> preds actual FALSE TRUE -1 2214 1271 1 396 10358           </pre>	<p><b>Training Accuracy:97.77</b></p> <p><b>Testing Accuracy:88.29</b></p>	 <p><b>AUC for training data: 0.9967</b>  <b>AUC for testing data: 0.9238</b></p>

## b) Naive Bayes

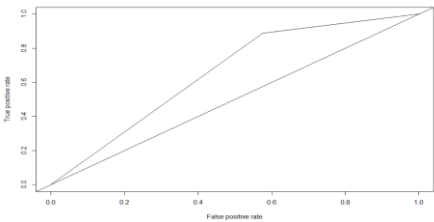
A threshold probability of 0.5 was used.

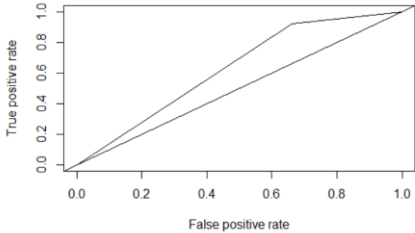
Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
-----------------------	------------------	----------	-------------------

0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 2785 590 1 4725 6142  <b>Testing Set:</b> preds actual FALSE TRUE -1 2893 671 1 5635 5040	<b>Training Accuracy: 0.626</b>  <b>Testing Accuracy: 0.557</b>	 <p>AUC for training data:0.6846 AUC for testing data: 0.7136</p>
-----	---	---	---

### c) SVM Model

The following models were run by using a radial kernel function and by varying the parameters:

Cost	Gamma	Training Confusion Matrix and Accuracy	Testing Confusion Matrix and Accuracy
10	0.5	<b>Training:</b> predicted actual -1 1 -1 2886 597 1 140 10616  <b>Training Accuracy: 0.94824</b>  <b>Area under the curve: 0.9078</b>  <b>ROC curve:</b> 	<b>Test:</b> predicted actual -1 1 -1 2489 996 1 481 10273  <b>Testing Accuracy: 0.8962</b>  <b>Area under the curve: 0.8347</b>

1	0.1	<b>Training:</b> predicted actual -1 1 -1 573 909 1 291 3515  <b>Training Accuracy: 0.773</b> <b>Area under the curve: 0.655</b>  <b>ROC curve</b> 	<b>Test:</b> predicted actual -1 1 -1 717 1401 1 432 5002  <b>Testing Accuracy: 0.7572</b>  <b>Area under the curve: 0.63</b>
---	-----	--	---

From the above, we observe that a radial basis kernel with cost=5 and gamma=0.5 provides a good model with 83.47% accuracy.

### **Term Frequency and tf-idf:**

We are using the term frequency to check the number of occurrences of any word in the document(i.e, review). The tf-idf values were used in our assignment since tf-idf value increases proportionally to the number of times a word appears in a review and is offset by the number of reviews in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. Hence, Tf-idf is used for term-weighting.

Because the term "food" is so common, term frequency will tend to incorrectly emphasize reviews which happen to use the word "food" more frequently, without giving enough weight to the more meaningful terms "good" and "unpleasant". The term "food" is not a good keyword to distinguish relevant and non-relevant reviews and terms, unlike the less-common words "good" and "unpleasant". Hence an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the reviews set and increases the weight of terms that occur rarely.

### **Document Term Matrix Size:**

3249 words which occur in > 90% reviews and less than 30 reviews.

The size of the Document Term Matrix is shown below:

**[1] 33443 3251**

and the total number of words considered for the models are **3249** after removing all the stopwords, rare words and based on term occurrence.



**Stemming** is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form.

Stemming is used when using the dictionaries because:

- 1) We use Inner join to get the matching words, reducing the words to their base form would find an exact match from the dictionary and helps in assigning the sentiment value easily since the dictionary maps a word to its lemma (stem).

### *Develop models using a broader list of terms.*

To Develop models using a broader list of terms, we

- We first obtain a broader list of terms by combining the words from the three dictionaries and proceed to use the resulting terms obtained by doing left-join.
- **Stemming:** The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Since we are considering the broader list of terms, for a better model predictability we have not implemented stemming. we compared the model predictions for both the cases, (i) with stemming, (ii) without stemming.
- **Document term matrix:** The dimension of the final document term matrix is 28733\* 3296 after removing star rating 3 from the dataset.

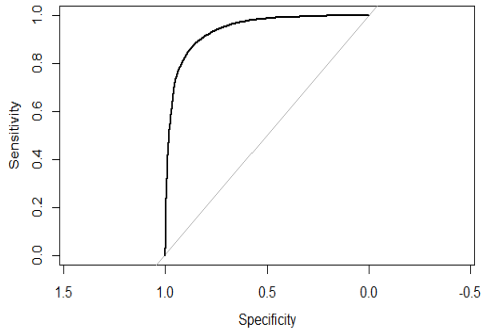
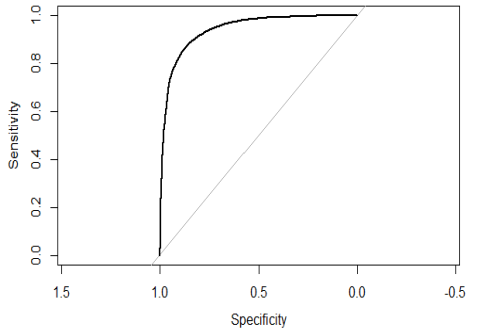
The count of +1 and -1 classifications were as follows:

	hiLo	n
1	-1	7034
2	1	21699

The training and the testing data were split into 50:50 ratios since we have a large number of reviews.

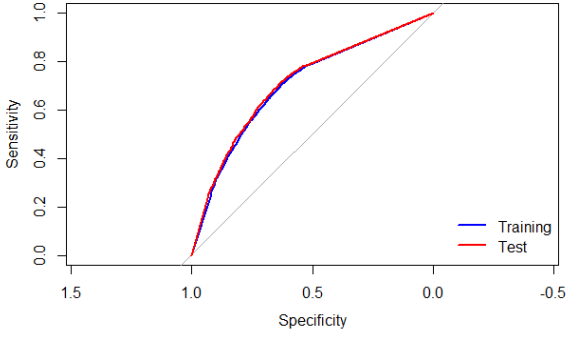
### **a)Random Forest:**

Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
-----------------------	------------------	----------	-------------------

0.5	<p><b>Training Set:</b>  preds  actual FALSE TRUE  -1 3448 56  1 4 10858</p> <p><b>Testing Set:</b>  preds  actual FALSE TRUE  -1 2243 1288  1 315 10520</p>	<p><b>Training Accuracy:99.58</b></p> <p><b>Testing Accuracy:88.84</b></p>	 <p><b>Area under the curve Testing : 0.9368</b></p>
0.8	<p><b>Training Set:</b>  preds  actual FALSE TRUE  -1 3504 0  1 749 10113</p> <p><b>Testing Set:</b>  preds  actual FALSE TRUE  -1 3271 260  1 2615 8220</p>	<p><b>Training Accuracy:94.79</b></p> <p><b>Testing Accuracy:79.99</b></p>	 <p><b>Area under the curve Testing : 0.9368</b></p>

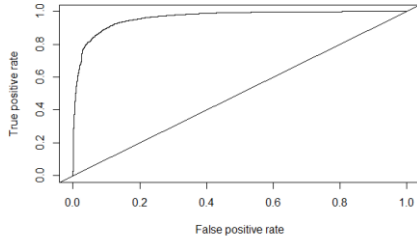
## b) Naive Bayes

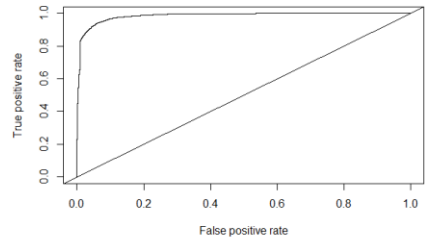
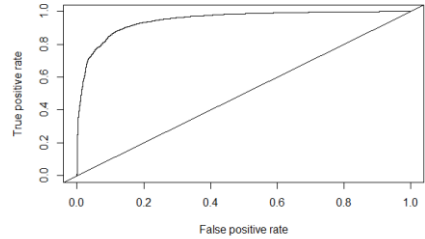
Threshold probability	Confusion Matrix	Accuracy	ROC Curve and AUC
-----------------------	------------------	----------	-------------------

0.5	<b>Training Set:</b> preds actual FALSE TRUE -1 4280 440 1 9175 6218  <b>Testing Set:</b> preds actual FALSE TRUE -1 1953 162 1 3752 2752	<b>Training Accuracy:0.52</b>  <b>Testing Accuracy:0.545</b>	 <p>AUC for training data:0.7041 AUC for testing data: 0.7111</p>
-----	---	--	---

### c) SVM Model:

The following models were run by using a radial kernel function and by varying the parameters:

Cost	Gamma	Training Confusion Matrix and Accuracy	Testing Confusion Matrix and Accuracy
1	0.1	Training: predicted actual -1 1 -1 2188 1266 1 109 10803  <b>Training Accuracy: 0.9042</b> <b>Area under the curve: 0.8117</b>	Testing: predicted actual -1 1 -1 2034 1547 1 147 10637  <b>Testing Accuracy: 0.882</b> <b>Area under the curve: 0.7772</b>  

5	0.5	<p>Training:</p> <p>predicted actual -1 1 -1 3386 68 1 18 10894</p> <p><b>Training Accuracy: 0.994</b> <b>Area under the curve: 0.9893</b></p>	<p>Testing:</p> <p>predicted actual -1 1 -1 2662 919 1 415 10369</p> <p><b>Testing Accuracy: 0.9017</b> <b>Area under the curve: 0.8524</b></p> 
NIL	NIL	<p>Training:</p> <p>predicted actual -1 1 -1 0 3454 1 0 10912</p> <p><b>Training Accuracy: 0.7596</b> <b>Area under the curve: 0.5</b></p>	<p>Testing:</p> <p>predicted actual -1 1 -1 0 3581 1 0 10784</p> <p><b>Testing Accuracy: 0.7507</b> <b>Area under the curve: 0.5</b></p> 

From the above, we observe that a radial basis kernel with cost=5 and gamma=0.5 provides a good model with 90.17% accuracy.

### Performance comparison:

We have compared model performance with part c for all three dictionaries here. We are not considering Naive Bayes model for our comparison as it didn't perform well with very low accuracy. The SVM and Random forest model has shown good performance for all three dictionaries.

Dictionary	Accuracy in part (c)	Model Accuracy -	Model Accuracy -
------------	----------------------	------------------	------------------

		SVM	Random Forest
Bing Liu Dictionary	0.814	0.89	0.87
NRC Dictionary	0.77	0.87	0.86
AFINN Dictionary	0.819	0.86	0.86

We see that the SVM model has performed best for Bing Liu Dictionary and NRC. The SVM and Random forest performed equally well for the AFINN dictionary.

For combined dictionaries and a broader set of variables, random forest performed well with accuracy 0.88 in both.

For a broader set of variables, the SVM model(accuracy=0.90) performed better with random forest(accuracy= 0.88).

**Performance:** The performance for the models is evaluated based on the (i) Accuracy, (ii) AUC value, (iii) ROC curve.

- **Accuracy:** We used Accuracy because it is the mean predictability, The accuracy of a model is given as the percentage of total correct predictions divided by the total number of instances. Since we are predicting the reviews as positive and negative, Accuracy measures can be used to classify future data tuples for which the class label is not known.
- **AUC:** area under the curve(we are considering tpr vs fpr), However we can consider the area for other measures like precision, recall etc. AUC gives a specific value which helps in determining the model's effectiveness on predicting the output classes.
- **ROC curve:** is the plot between tpr and fpr, through which we can visually check our model performance

## References:

<https://www.pluralsight.com/guides/evaluating-a-data-mining-model>