**Assignment 3**

Analyzing text in Yelp reviews - Text mining, Sentiment analysis

**Report By:**

Avanish Singh, Alique Zohran, Alex Kwan

**(a) Explore the data.**

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

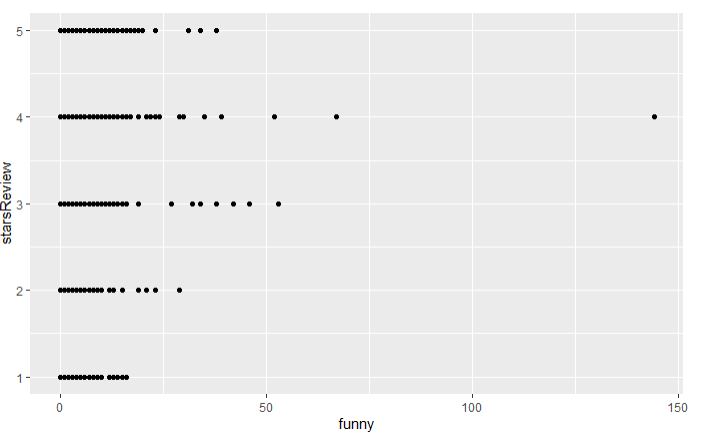
Total number of reviews with respect to star ratings are shown in the table below.



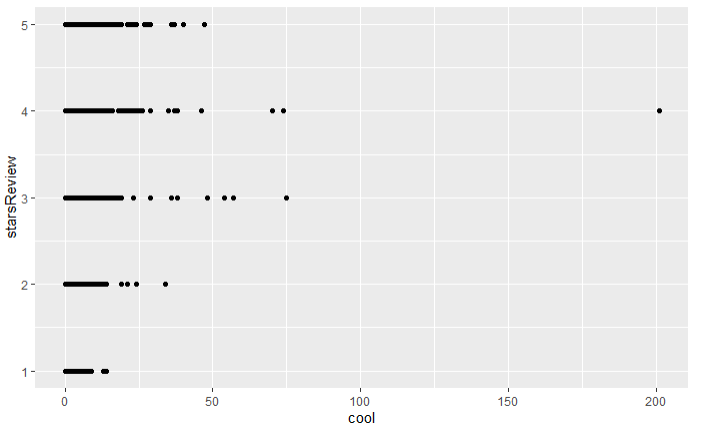
The star ratings will be used on a scale of 1-5(1-low rating of restaurants and 5- for highest rating)

More reviews are positive as seen from above that restaurants with 4,5 ratings are mostly review positive with total numbers are 10795,15084 respectively. While star ratings with 1 are negative ratings with total numbers are 4553 which are much lesser than 4,5 ratings.

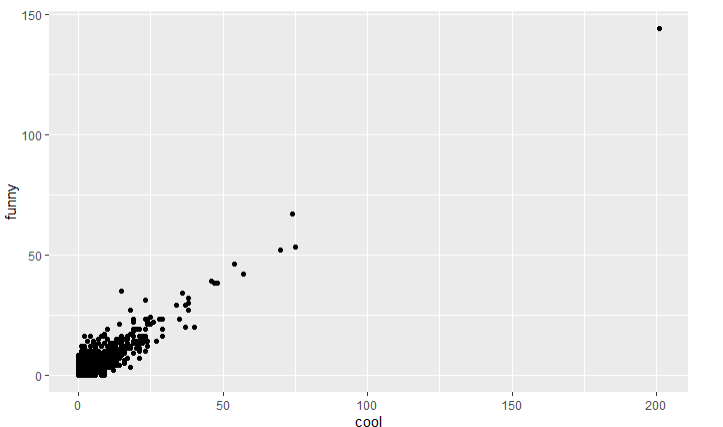
Relation of star Review with funny is shown below



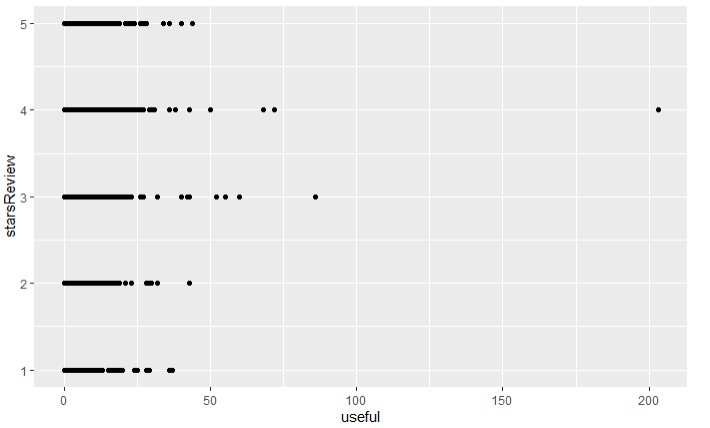
Star Review vs cool



Cool Vs Funny



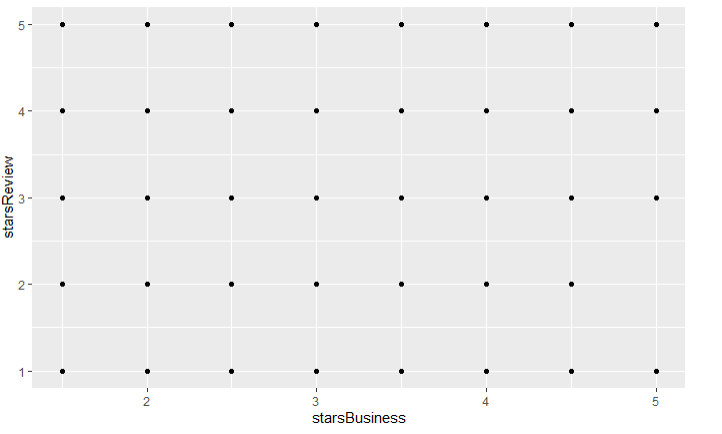
Star reviews vs useful



**(a)(ii)**

**How does star ratings for reviews relate to the star-rating given in the dataset for business (attribute ‘businessStars’)? (Can one be calculated from the other?)**

The relationships are shown in the graph below and we can calculate star reviews with help of businessStars and vice-versa as both are directly proportional.

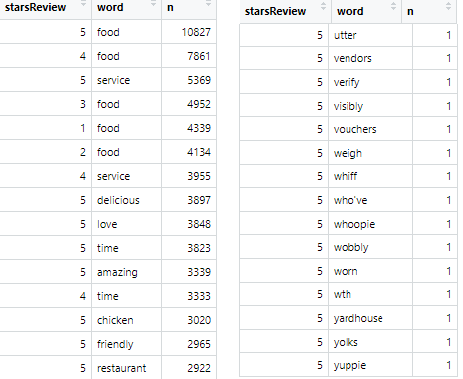


**(b)**

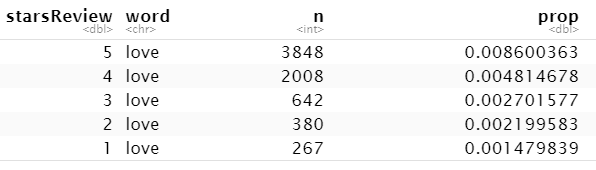
**What are some words indicative of positive and negative sentiment? (One approach is to determine the average star rating for a word based on star ratings of documents where the word occurs). Do these ‘positive’ and ‘negative’ words make sense in the context of user reviews being considered? (For this,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

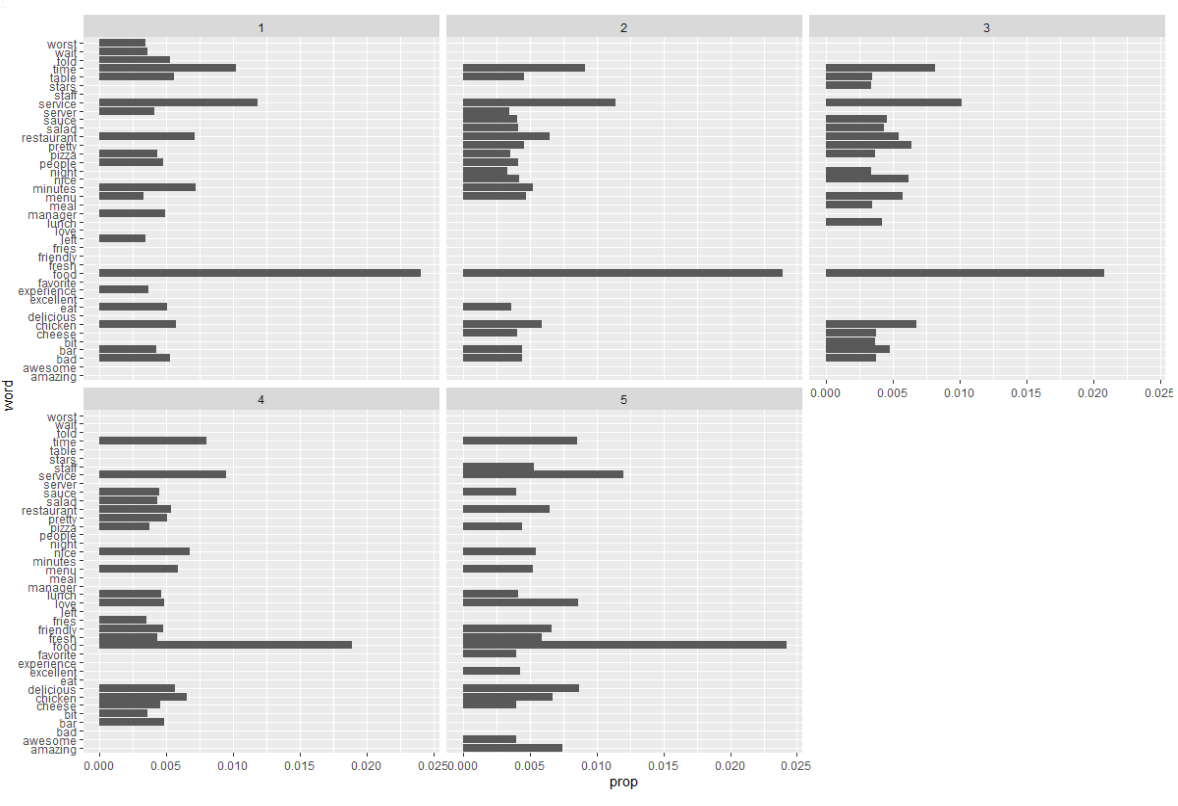
These are words which frequently appear with high star review ratings, for example word “food” appears frequently with star rating 5,4 but the food appears in a lesser star rating also. So with this we are not much clear about the words, in order to be sure we take proportional.

**List of top words List of bottom words**

****

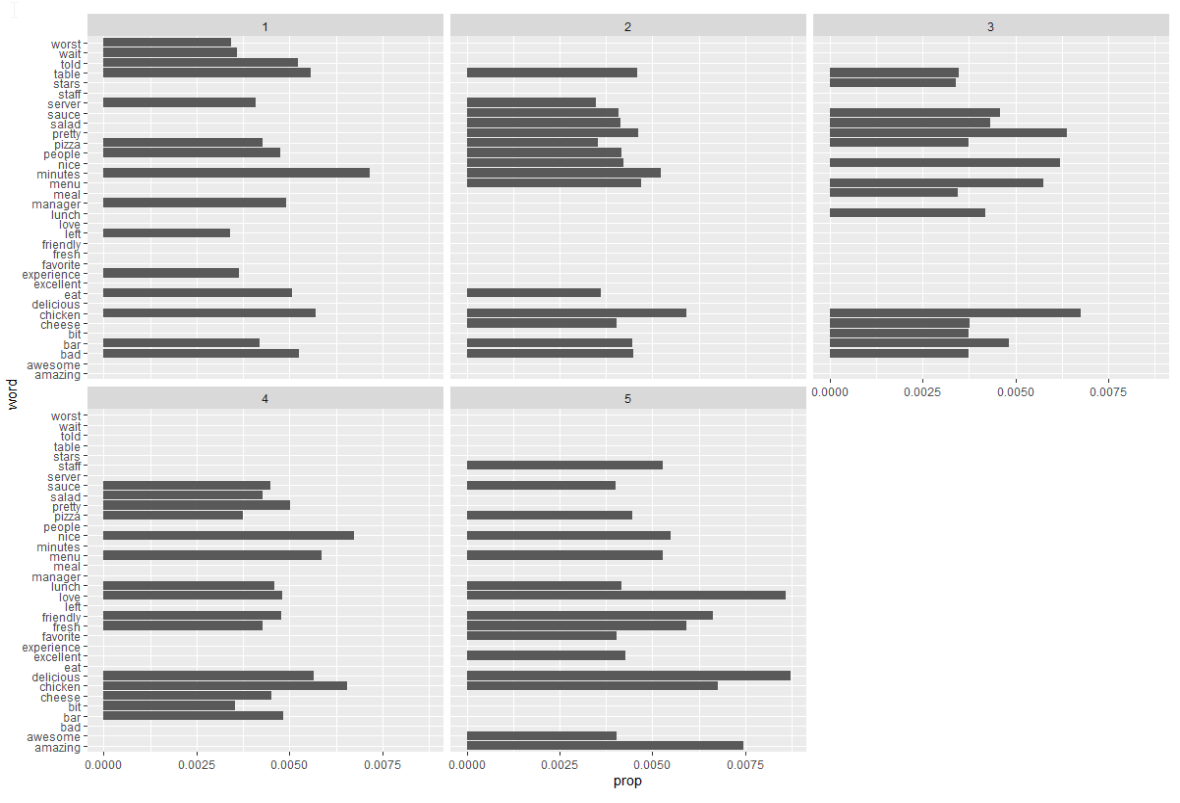
With the below table we can say that the word love is a positive word as its proportional in 5 star rating is higher than any other star ratings which means the number of times this word appears in 5 star rating is higher than any other rating. But there are some words which we cannot differentiate.

****

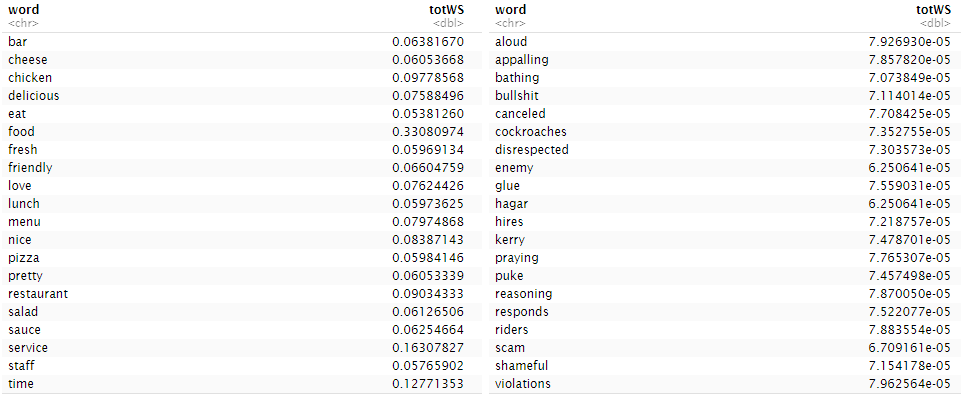
****

There are some words which appear more frequently in every rating so this word does not make any difference to determine the ratings, so we have to remove these words for eg, food,time, service etc.

In the below graph we see that words like amazing,awesome etc do not appear in small ratings only appear with higher ratings i.e 5. And some negative words like worst,wait etc appear in small rating Review i.e for 1 star rating.



top 20 words with highest star rating bottom 20 words with lowest star rating



Now it can easily differentiate the words with positive and negative meanings as seen on the left table. Words like friendly,love,nice,chicken sound positive and the words like aloud,bullshit,glue,puke sound negative words which are shown on the right side.

**(c)**

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

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

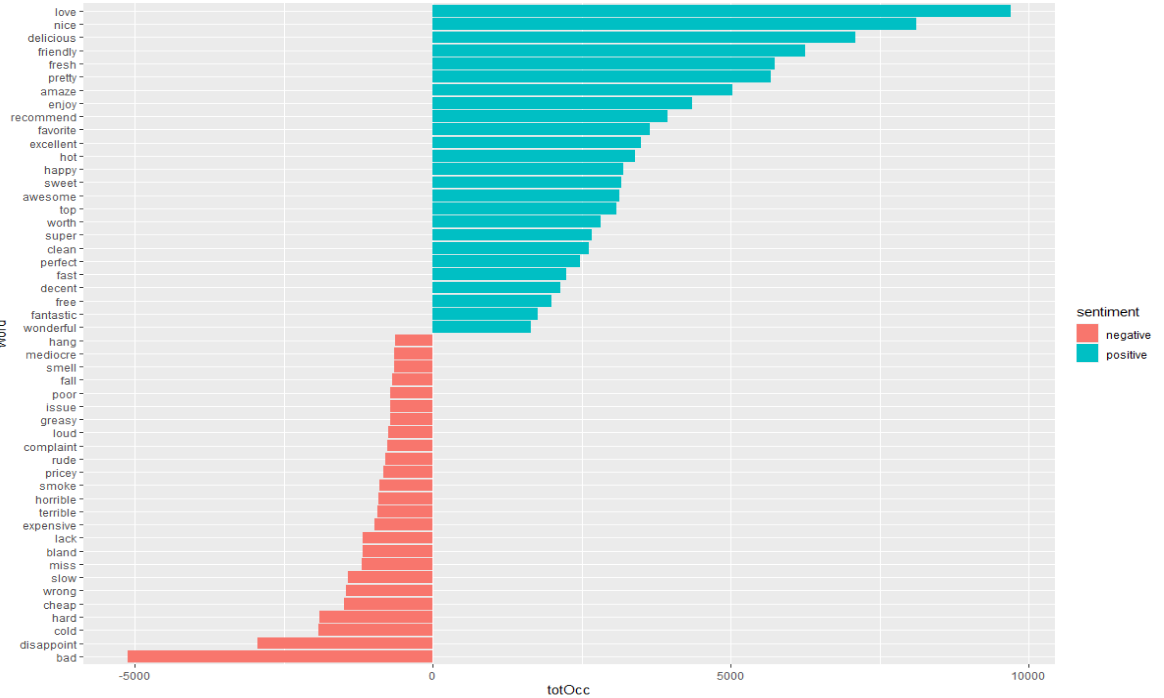
Using “bing” dictionary

Get the sentiment of the word review by ID, from below we can see that some of the word in sentiment column have N/A value so to remove that we can join the dictionary with inner\_join function

****

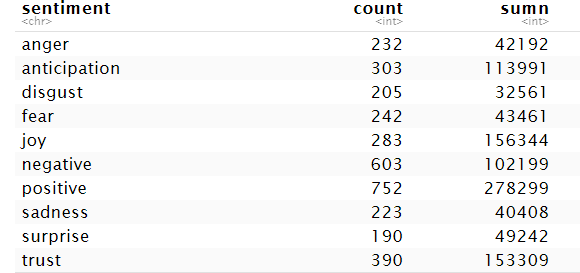
With inner\_join:



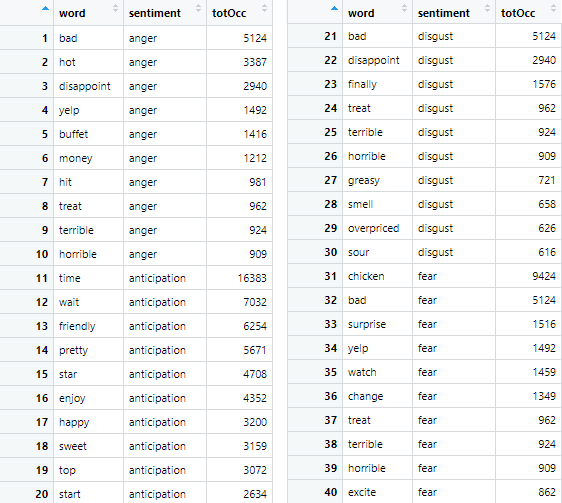


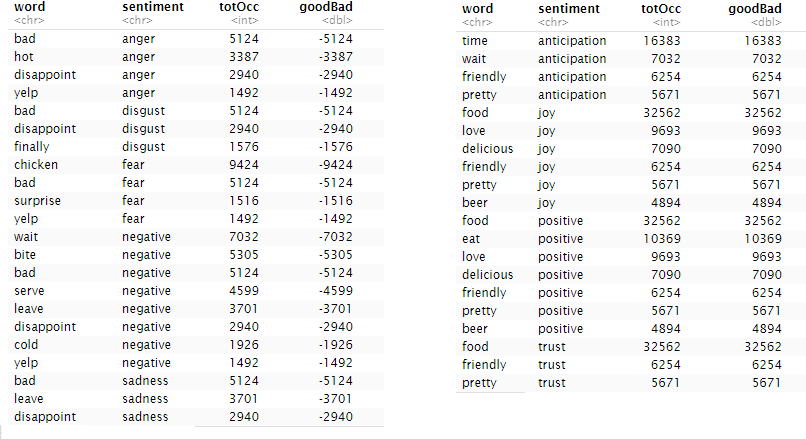
Using ‘nrc’ dictionary:

Number of words with different sentiment categories

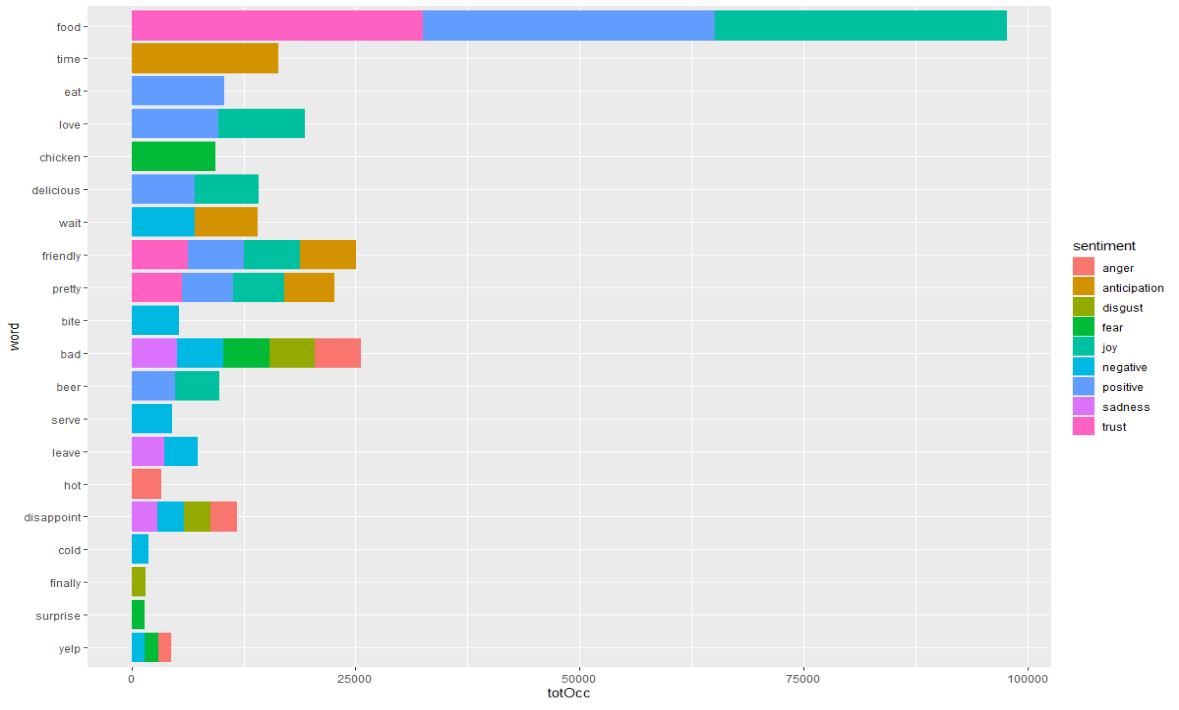


Top few words for different sentiment





Plot the above words:



With ‘afinn’ dictionary:



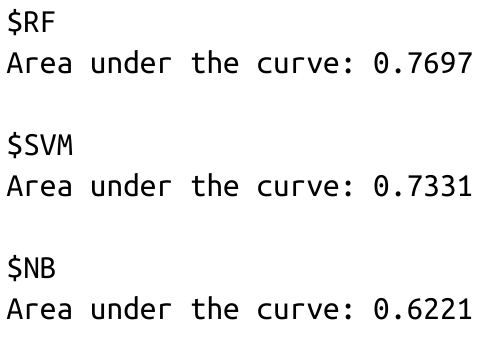
**(d)**

**Develop models to predict review sentiment. For this, split the data randomly into training and test sets. To make run times manageable, you may take a smaller sample of reviews (minimum should be 10,000). One may seek a model built using only the terms matching any or all of the sentiment dictionaries, or by using a broader list of terms (the idea here being, maybe words other than only the dictionary terms can be useful). You should develop at least three different types of models (Naïve Bayes, and at least two others of your choice ….Lasso logistic regression (why Lasso?), xgb, svm, random forest (ranger).**

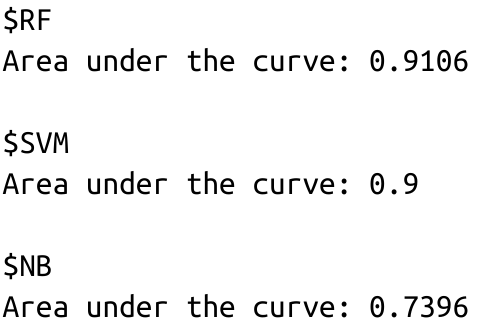
1. **Develop models using only the sentiment dictionary terms – try the three different dictionaries; How do the dictionaries compare in terms of predictive performance? Then with a combination of the three dictionaries, ie. combine all dictionary terms. Do you use term frequency, tfidf, or other measures, and why? What is the size of the document-term matrix? Should you use stemming or lemmatization when using the dictionaries?**
2. **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.**

Regardless of what model we use, we will use the tf-idf representation of terms because we want to weight less frequent terms more since they provide more information as to what a document may be about. We also use lemmatization since it provides extra information from which to separate sentiments. Since this is a classification task we will use AUC to judge the performance of the models with each of the dictionaries. The document-term matrices produced from the dictionaries tended to be fairly large, with AFINN at the smallest at 38,163 x 625, NRC in the middle at 39,780 x 883 and Bing at the largest with 38,984 x 1,132. The performance of the models were fairly similar between AFINN and Bing dictionaries, but was much worse with NRC.

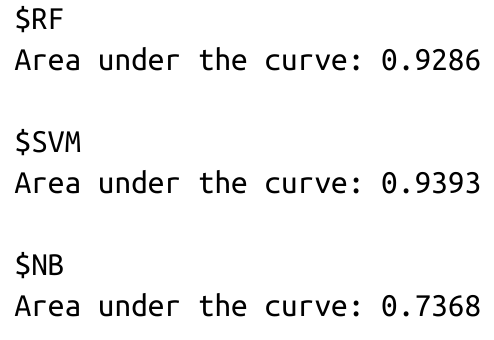
NRC performance:



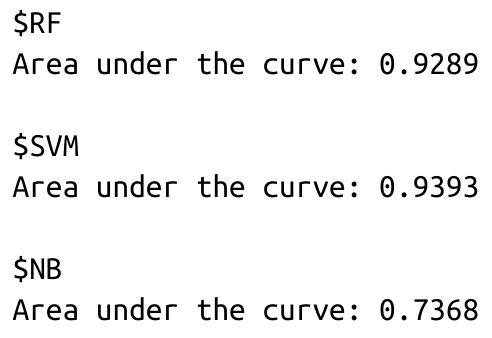
AFINN performance:



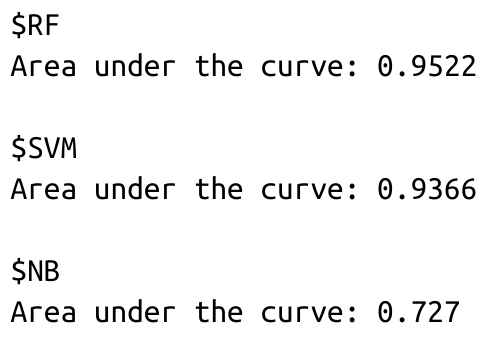
Bing performance:



The combination of dictionaries resulted in a DTM of size 39,963 x 1,723. Although the combination of dictionaries resulted in a larger DTM, it performed only slightly better than singular dictionaries as we can see below:



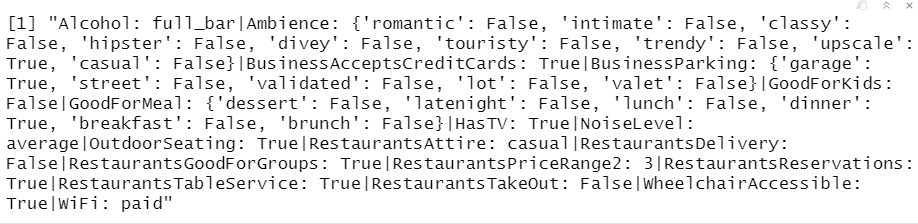
Both in terms of the narrower list of terms and the broader terms, the algorithms we trained performed better. For instance, the AUC from (c) using AFINN was only about 0.8354, while most of our algorithms performed above that here, with the exception of Naive Bayes which generally had AUC of about 0.1 less than that. These tokens were again lemmatized. Despite having a larger DTM the models did not perform that much better with the new DTM:



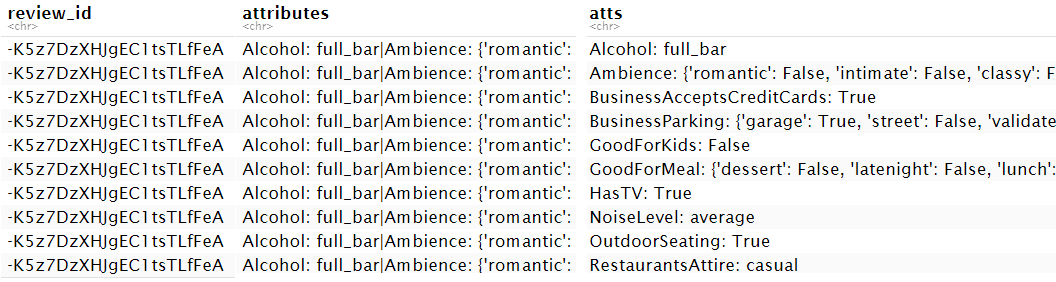
**(e)**

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

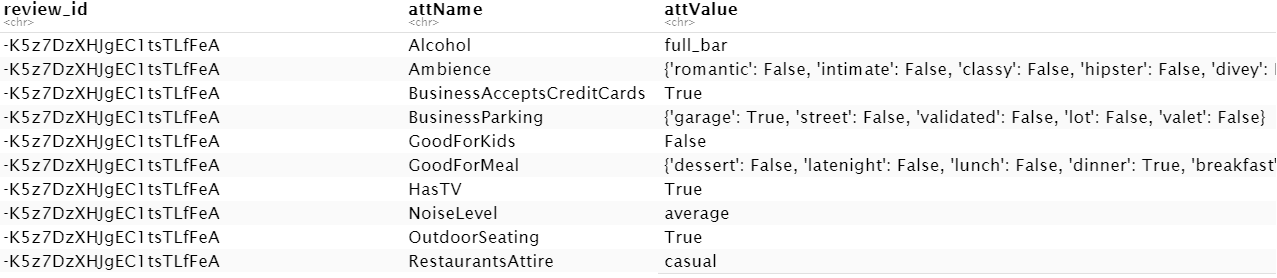
There are multiple attributes inside the attributes column so before using these attributes we have to extract them.



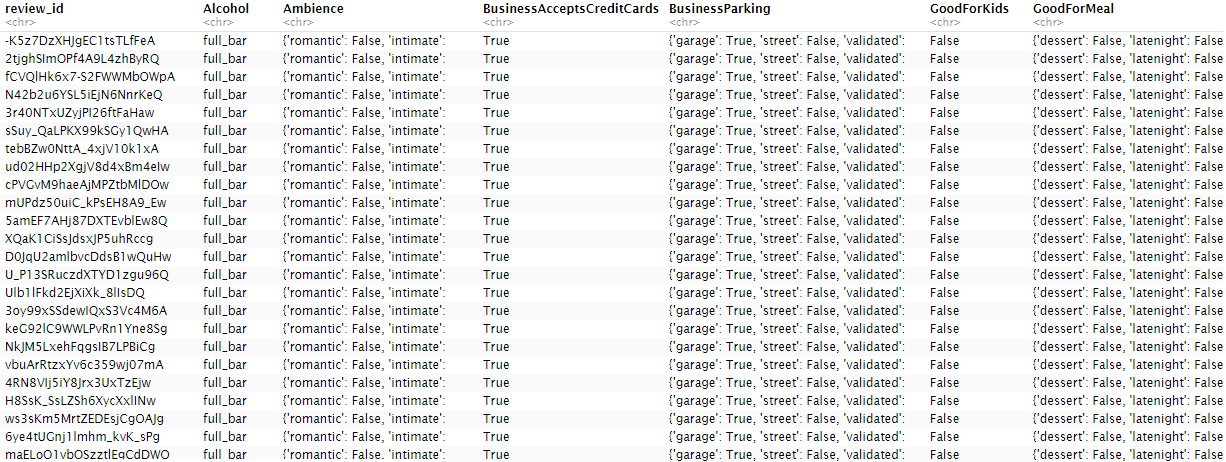
After splitting the attributes and store in other variable as shown below:



Now further split the attributes in column name “atts” and store in “attName” and “attValue” as shown below.



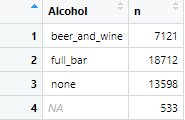
Further split the attribute value under column “attValue” to its individual values as shown below.



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

Some of the attributes which we summarize are Alcohol,Ambience,Noise Level.

Alcohol:

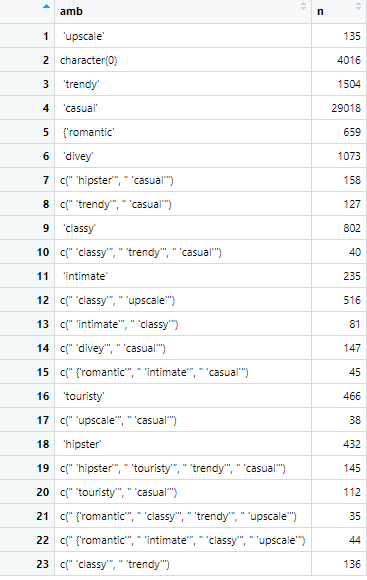


Ambience:

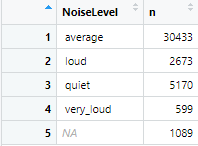
As ambience has many attributes which are listed below along with their total count.

Upscale is used for review 135 times.

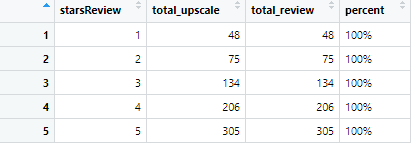
15-In ambience all three romantic,intimate and casual are true.



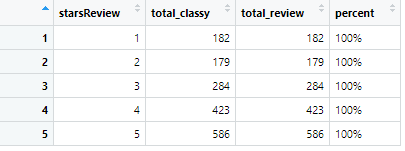
Noise Level:



Star rating vary by upscale shown below

****

Star ratings vary by classy shown below

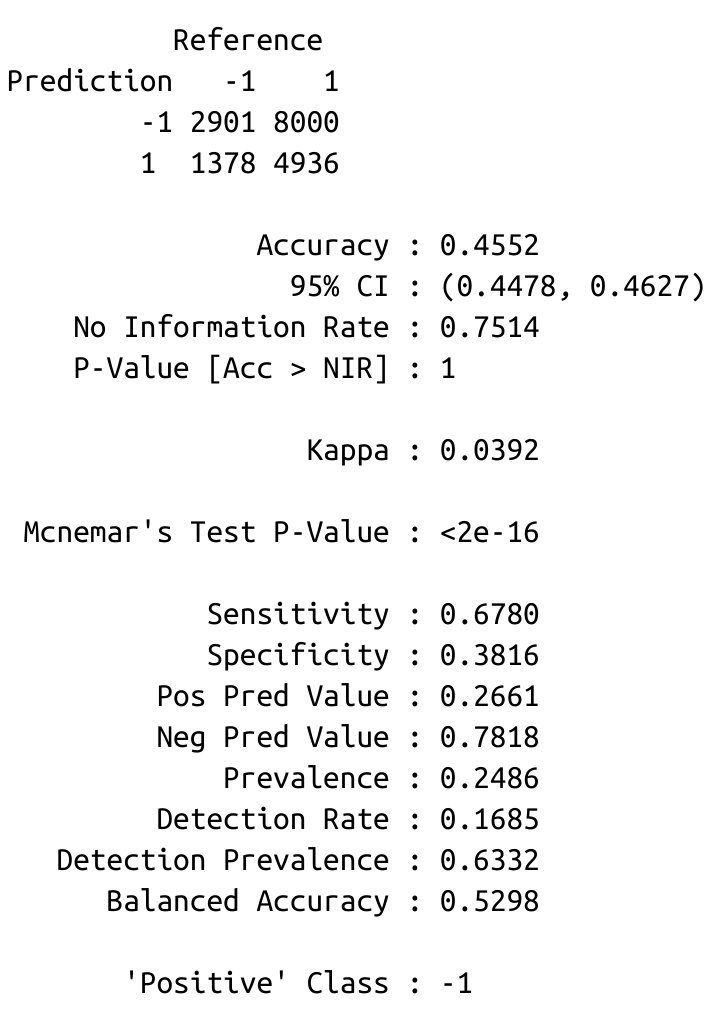
****

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

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

Yes, the accuracy does vary by certain attributes. For example when we compare how well a RF does with alcohol services versus restaurant noise we see that noise level is a better predictor of higher or lower ratings.

Alcohol model:



Noise level model:

