Assignment 3 – Analyzing text in Yelp reviews - Text mining, Sentiment Analysis

IDS 572

Adrian Blamires

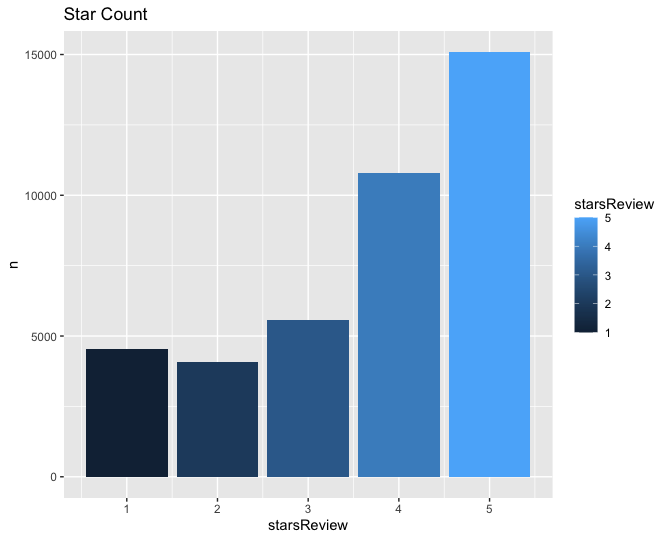
Jinrong Qiu

Michael Gannon

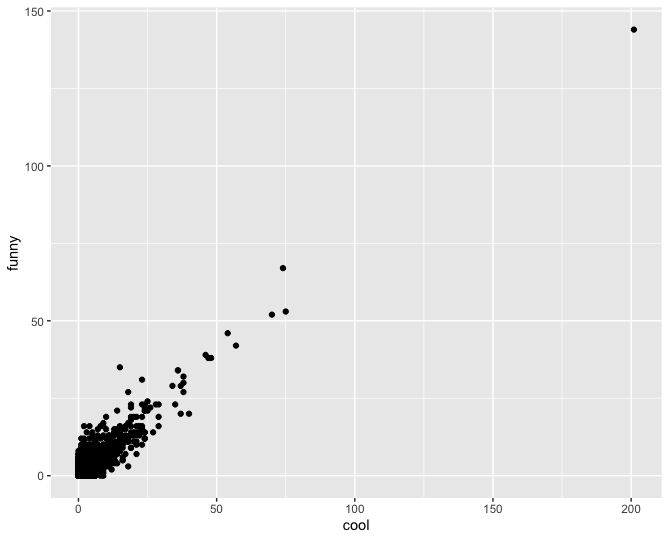
(a) Explore the data. (i) How are star ratings distributed? How will you use the star ratings to obtain a label indicating ‘positive’ or ‘negative’ – explain using the data, graphs, etc.? Do star ratings have any relation to ‘funny’, ‘cool’, ‘useful’? Is this what you expected? (ii) How does star ratings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

Looking at the data the star ratings are not evenly distributed. The graph shows that there are about one third the amount of 1 and 2 star ratings as there are 4 and 5 stars. This ratio may be important when determining the threshold for the models. This is also a non-normal distribution. It can be assumed that anything rated 4 or 5 would be considered positive and anything rated 1 or 2 would be considered negative. 3 ratings would be neutral with no positive or negative connotation.

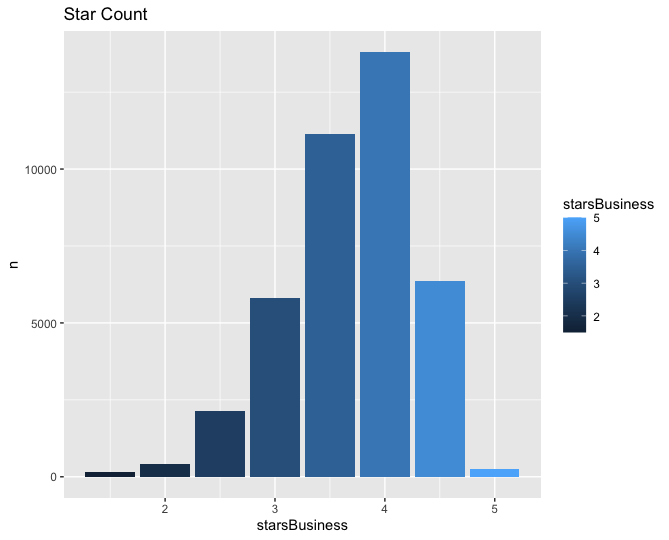
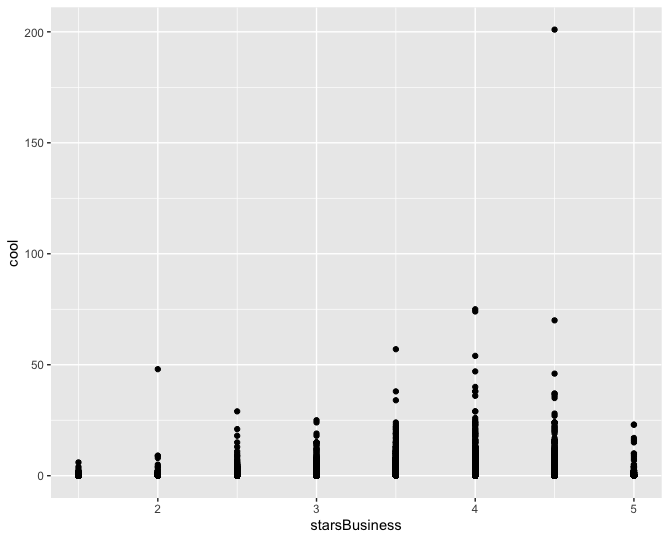
| starsReview | n |
| --- | --- |
| 1 | 4553 |
| 2 | 4094 |
| 3 | 5561 |
| 4 | 10795 |
| 5 | 15084 |



The plot for funny vs cool shows a linear relationship between the two labels. Most of the restaurants that are marked funny have a similar amount of cool ratings. Majority of these restaurants have very few of either.



When looking at business stars there appears to be a correlation between cool and more stars. The distribution of the business stars appears much more normalized than the reviews by users. Majority of the business stars are at 4 and there are very few 1’s or 5’s.



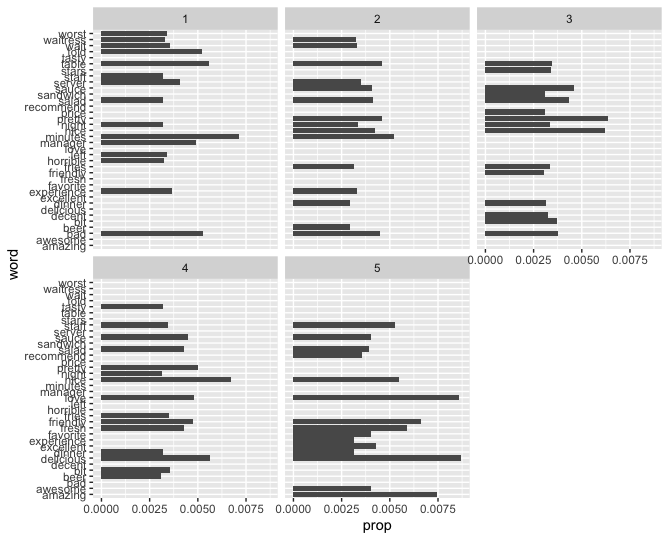
(b) What are some words indicative of positive and negative sentiment? (One approach is to determine the average star rating for a word based on star ratings of documents where the word occurs). Do these ‘positive’ and ‘negative’ words make sense in the context of user reviews being considered? (For 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).

After tokenizing the reviews column it was found that there are 43,993 distinct words. By removing the stop words, words containing digits, and any word that appears less than 10 times the total number of distinct words is reduced to 9167. Looking into sentiment it is clear that some words are associated with positive or negative reviews. For example a distribution of the word love and the number of reviews based on star rating can be seen below. There is a much larger proportion of people using the word love for 5 star reviews than there are for 1 or 2 star reviews. On the other hand a word like “appalling” only appears in one and two star reviews.

| starsReview | word | n | prop |
| --- | --- | --- | --- |
| 5 | love | 3848 | 0.0086 |
| 4 | love | 2008 | 0.00481 |
| 3 | love | 642 | 0.0027 |
| 2 | love | 380 | 0.0022 |
| 1 | love | 267 | 0.00148 |

| starsReview | word | n | prop |
| --- | --- | --- | --- |
| 1 | appalling | 10 | 0.0000554 |
| 2 | appalling | 2 | 0.0000116 |

Looking at the most commonly occurring words based on star distribution a pattern of positive and negative words can be pulled out. One thing to note on the first chart is that some words have no positive or negative sentiment at all and are just common words in a review. Looking at words that appear in all star ratings the following words can be removed from a sentiment check: ​​'food', 'time', 'restaurant', 'service', 'chicken', 'menu', 'eat', 'pizza', ‘bar’, ‘burger’, ‘cheese’, ‘lunch’, ‘meal’, ‘people’. The results of the word distribution with and without these neutral words can be seen below.



(c) We will consider three dictionaries, available through the tidytext package – the NRC dictionary of terms denoting different sentiments, the extended sentiment lexicon developed by Prof Bing Liu, and the AFINN dictionary which includes words commonly used in user-generated content in the web. The first provides lists of words denoting different sentiment (for eg., positive, negative, joy, fear, anticipation, …), the second specifies lists of positive and negative words, while the third gives a list of words with each word being associated with a positivity score from -5 to +5. How many matching terms are there for each of the dictionaries? Consider using the dictionary based positive and negative terms to predict sentiment (positive or negative based on star rating) of a movie. One approach for this is: using each dictionary, obtain an aggregated positiveScore and a negativeScore for each review; for the AFINN dictionary, an aggregate positivity score can be obtained for each review. Describe how you obtain predictions based on aggregated scores. Are you able to predict review sentiment based on these aggregated scores, and how do they perform? Does any dictionary perform better?

The table below summarizes the number of words matched for each dictionary. The NRC had the most matches with 13,875 followed by Bing, with half as many matches as NRC. Finally AFFIn had approximately a third of the matches as Bing and thus about one sixth of the matches of NRC.

Bing - 6786

NRC - 13875

AFFIN - 2477

The table below summarizes the average number of positive words, negative words, and an average sentiment score for each of the reviews. You can see that the average number of positive scores has a direct relationship with the overall star rating, and the average number of negative words has an inverse relationship with the star rating. Overall average sentiment is negative for ratings with reviews of 1 or 2 and positive for those with stars three and above. Its important to note that the average sentiment score is more than double for star rating of 4 and 5 when compared to a rating of 3. This is important because it shows that there may be a point of differentiation between a star rating of 3 vs. higher than 3.

starsReview avgPos avgNeg avgSentiSc

1 1 0.311 0.689 -0.378

2 2 0.448 0.552 -0.103

3 3 0.610 0.390 0.221

4 4 0.755 0.245 0.510

5 5 0.832 0.168 0.665

The confusion matrix for the NRC dictionary is printed below. Overall the accuracy was 79% with a recall of 81% and a precision of 94%.

NRC

predicted

actual -1 1

-1 2831 5766

1 1433 24234

The confusion matrix for Bing is printed below. Overall the accuracy was 84% with a recall of 92% and a precision of 86%.

Bing

predicted

actual -1 1

-1 6362 2001

1 3441 21793

The confusion matrix for the Affin model is printed below. Overall the accuracy was 84% with a recall of 88% and a precision of 91%.

AFFIN

predicted

actual -1 1

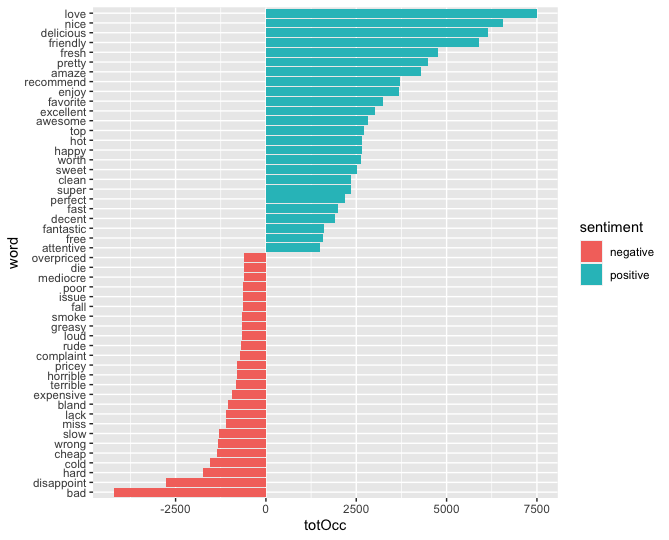
-1 5117 3081

1 2132 22572

Overall it seems as though the Bing or the Affin may be the best approach in terms of predicting word sentiment. Teh Bing model may have a better performance of picking true positive, while

the Afiin will have a better method to minimize false negative.

Bing Distribution



NRC Distribution

| sentiment | count | sumn |
| --- | --- | --- |
| anger | 232 | 37202 |
| anticipation | 303 | 96457 |
| disgust | 205 | 29266 |
| fear | 242 | 36860 |
| joy | 283 | 126635 |
| negative | 603 | 89898 |
| positive | 752 | 233196 |
| sadness | 223 | 36290 |
| surprise | 190 | 43956 |
| trust | 390 | 127600 |

(d) Develop models to predict review sentiment

**Random Forest (rfModel1) - Bing**

**Training Data**

Training accuracy : 95.45%

preds

actual FALSE TRUE

-1 5255 595

1 201 17466

AUC: 0.9922

**Testing Data**

Testing accuracy: 86.37%

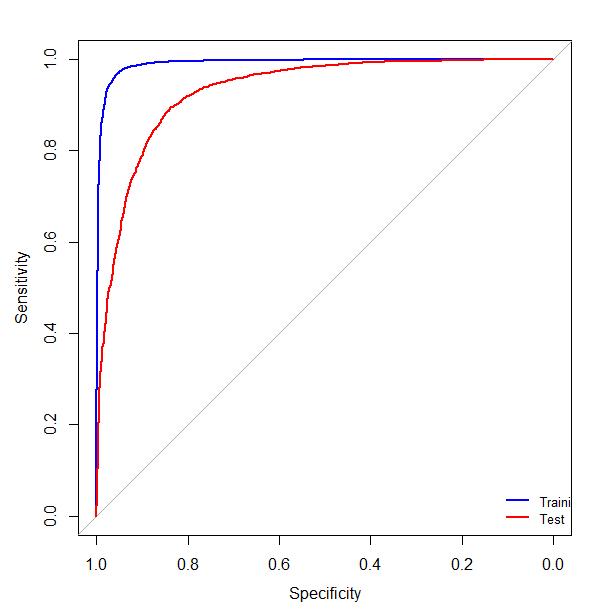
preds

actual FALSE TRUE

-1 1751 762

1 325 7242

AUC:0.9319



The random forest model showed that there may be some data leakage or multicollinearity occurring in the data set since the testing accuracy and AUC was lower than the training accuracy/auc. This could be a sign of potential overfit.

**SVM (svmM1) - Bing**

**Training Data**

Training accuracy: 89.80%

predicted

actual -1 1

-1 3883 1967

1 430 17237

AUC:0.8197

**Testing Data**

Testing accuracy: 89.087%

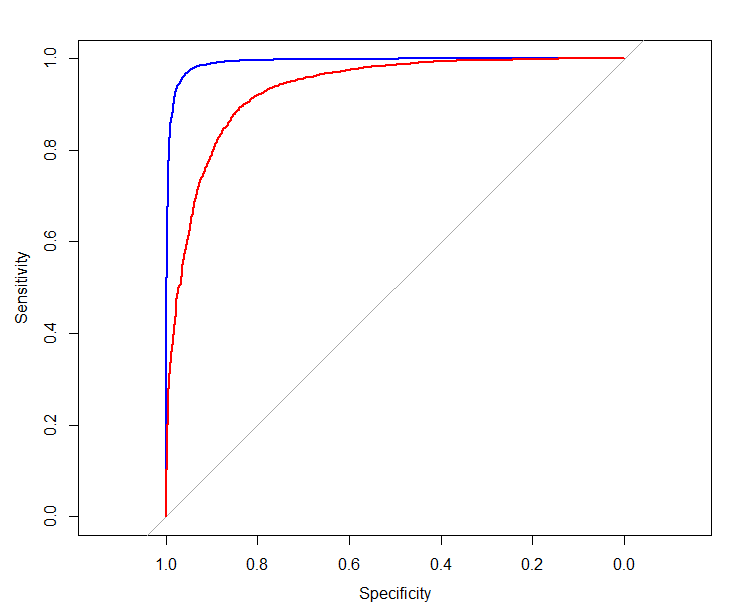
predicted

actual -1 1

-1 1615 898

1 202 7365

AUC:0.808



The SVM model showed consistent results between testing and training and still yielded accurate results as demonstrated by a high accuracy and auc.

**Naive Bayes - Bing**

**Training Data**

Training Accuracy: 71.6%

predicted

actual FALSE TRUE

-1 3879 1932

1 4745 12961

Area under the curve: 0.7244

**Testing Data**

Testing Accuracy: 71.7%

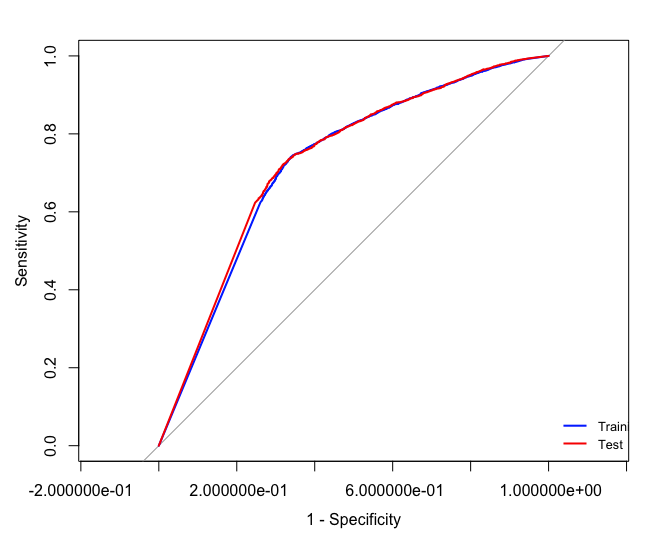
predicted

actual FALSE TRUE

-1 1701 851

1 1999 5529

Area under the curve: 0.7296



Performance for the Naive bayes model showed that performance was worse compared to the random forest/SVM models. This is demonstrated by the lower AUC, and there wouldn’t be points on the AUC curve that suggest the random forest would be appropriate (i.e, no opportunities for convex hull analysis where more than two models are combined).

**Random Forest (rfModel\_nrc) - NRC**

**Training Data**

Training accuracy : 92.24%

preds

actual FALSE TRUE

-1 4115 1850

AUC:0.9956

**Testing Data**

Testing accuracy: 83.77%

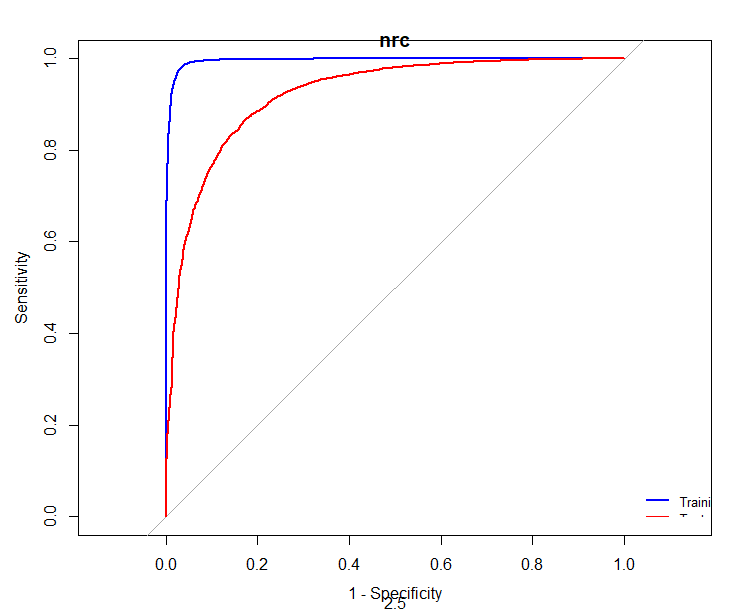
preds

actual FALSE TRUE

-1 1052 1580

1 88 7560

AUC:0.9216



The random forest model for NRC suggests that there may be some data leakage as the results are not as consistent between the test and training set (I.e, the accuracy and auc drop by almost 10% each).

**SVM (svmM1\_nrc) - NRC**

**Training Data**

Training accuracy : 92.75%

predicted

actual -1 1

-1 4697 1268

1 471 17548

AUC:0.8806

**Testing Data**

Testing accuracy: 88.92%

predicted

actual -1 1

-1 1847 785

1 353 729

AUC:0.8278

The SVM model using NRC shows that the testing set and training set are generating consistent results that are relatively high (88%).

**Naive Bayes - NRC**

**Training**

Training Accuracy: 66.2%

predicted

actual FALSE TRUE

-1 2412 3180

1 4153 11955

AUC: 0.6957

Testing Accuracy: 63.2%

predicted

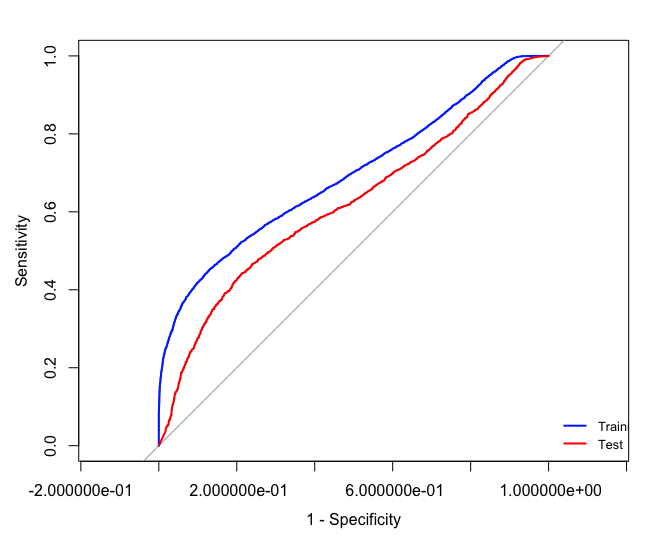
actual FALSE TRUE

-1 825 1478

1 1904 4986

AUC: 0.6204

The Naive Bayes model for NRC shows inferior performance compared to the other models.



**Random Forest (rfModel1\_affin) - Afinn**

**Training Data**

Training accuracy : 90.96%

preds

actual FALSE TRUE

-1 3726 2026

1 56 17223

AUC:0.98602

**Testing Data**

Testing accuracy: 85.28%

preds

actual FALSE TRUE

-1 1105 1341

1 112 7313

AUC:0.9161

The random forest model for Affin is consistent with the other data dictionaries where the training and testing differences suggest overfit.

**SVM (svmM1\_affin) - Affin**

**Training Data**

Training accuracy : 98.34%

predicted

actual -1 1

-1 4022 1730

1 724 16555

AUC:0.8287

**Testing Data**

Testing accuracy: 86.65%

predicted

actual -1 1

-1 1671 775

1 287 7138

AUC:0.8223

The SVM model using Affin suggests potential overfit/data leakage as demonstrated by the difference between accuracy for testing/training data.

**Naive Bayes - AFFIN**

**Training Data**

Training Accuracy: 61.3%

predicted

actual FALSE TRUE

-1 2937 2806

1 2803 14485

AUC: 0.7355

**Testing Data**

Testing Accuracy: 75.8%

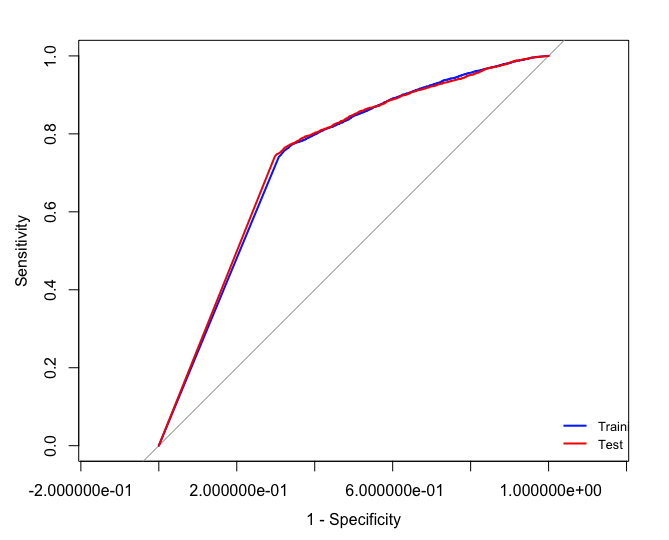
predicted

actual FALSE TRUE

-1 1302 1153

1 1238 6178

AUC: 0.7394



The naive bayes model has consistent results, although not as good as the other models.

Overall the Bing and NRC dictionaries appear to have the best results with the SVM models for each.

**Combined dictionaries**

**Training Data**

Training accuracy : 95.22%

predicted

actual -1 1

-1 5270 788

1 363 17673

AUC: 0.9249

**Testing data**

Testing accuracy: 91.91%

predicted

actual -1 1

-1 2008 555

1 365 7398

AUC: 0.8682

**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 documentterm matrix?**

* For Bing, NRC, Afinn, dictionaries, the SVM model out performed Naives and random forest model. The SVM model gives the test accuracy of 89.087%, 88.92%, and 86.65%.
* The best overall model is the Combined dictionaries with an accuracy of 91.91% for the testing data. This model performs even better than the SVM model in all three dictionaries.
* We use tf-idf, which stands for term frequency-inverse document frequency, to analyze the statistical value of a word’s importance to the document.

**Should you use stemming or lemmatization when using the dictionaries?**

* Yes, we should use stemming or lemmatization to condense the terms that fit within the library. It would reduce derived forms and transform words into their common base form.

**(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?**

We developed the Random Forest (Ranger) models using a broader list of terms with a broader set of terms. The broader list of terms are far more complicated than the dictionaries (Bing, NRC, Afinn) we used before. The broader terms allow our model to capture data that is not specifically within the individual three libraries. We obtain the broader list of terms by filtering out words that occurred in more than 90% of the reviews and less than 30% of the reviews. The broader terms come at greater cost as well. It takes a much longer time to run the model. The broader list model took the computer 30 minutes to run where the other dictionaries took about 8 minutes on average. We use stemming to reduce the number of words in the data set.

**Training Data**

Training accuracy: 95.418%

preds

actual FALSE TRUE

-1 4923 1107

1 0 18134

AUC:1

**Testing Data**

Testing accuracy: 83.57%

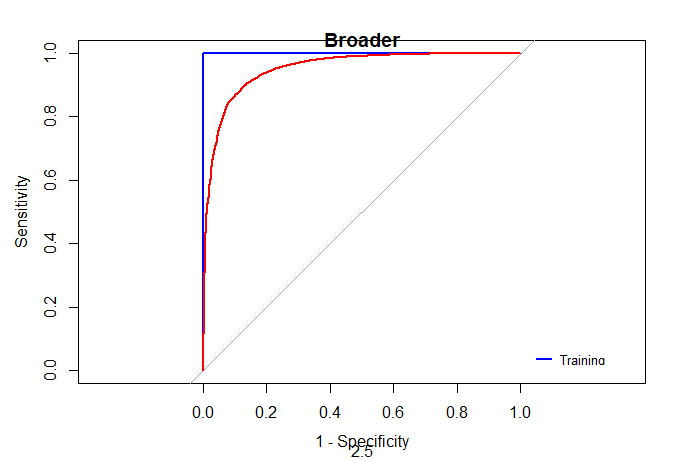
preds

actual FALSE TRUE

-1 951 1664

1 37 7704

AUC:0.9508



**Report on performance of the models. Compare performance with that in part (c) above. How do you evaluate performance?**

We evaluate the performance of different models by comparing their accuracy performance. Compared to part C, Random Forest and SVM models perform better. For the Bing dictionary, Random Forest model has 86.37% accuracy and SVM model has 89.087% accuracy, which is higher than 84% accuracy in part C. For the NRC dictionary, Random Forest model has 83.77% accuracy and SVM model has 88.92% accuracy, which is also out performed 79% accuracy in part C. For Afinn dictionary, Random Forest model has 85.28% accuracy and SVM model has 86.65% accuracy, which is higher than 84% in part C. However, the naive Bayes’ performance is lower compared to part C with 71% accuracy with Bing dictionary, 32.2% with NRC dictionary, and 75.8% with Afinn dictionary.

**Which performance measures do you use, why.**

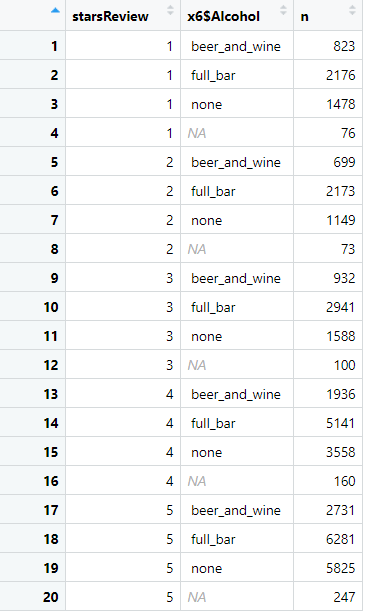
We would use the SVM combined dictionary model because it gives the test data highest accuracy performance of 91.91%. SVM was chosen because it has out performed the Random Forest model and the Naives Model in each of the three dictionaries (89.087% in Bing, 88.92% in NRC, and 86.65% in Afinn).

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

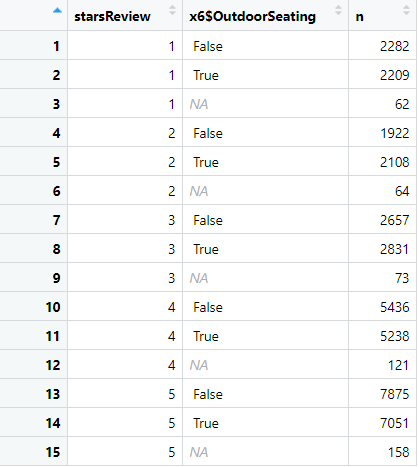
(i) Consider a few interesting attributes and summarize how many restaurants there are by values of these attributes; examine if star ratings vary by these attributes.

(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

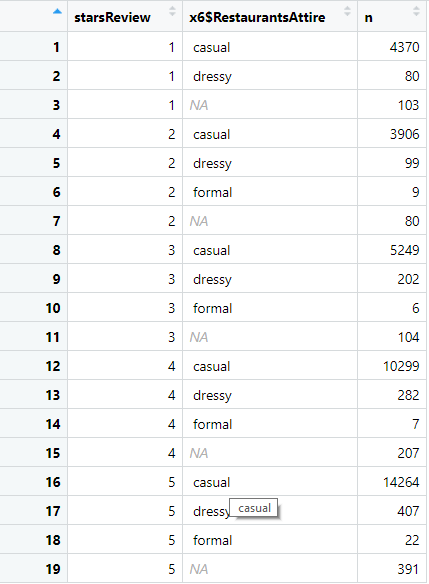
(e), you will consider the values in the ‘attribute’ column. This has values of multiple attributes, separated by a ‘|’. Further, some of the values, like Ambience, carry a list of True/False values (like, for example, Ambience: {'romantic': False, 'intimate': False, 'classy': False, 'hipster': False, …}. 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.



Restaurants that have either a full bar or serve beer and wine are more likely to have a higher stars rating. If you look at the percent of all responses for the alcohol attribute, the percent with either full bar/beer and wine is between 7 and 10% for star ratings 1 through 3. This jumps to 18 and 22% respectively for restaurants with a stars rating of 4 and 5.



At first glance of the attribute outdoor seating, there appears to be a direct relationship between teh percent of true responses and the star rating (higher stars is associated with a higher level of outdoor seating). However if you examine the percent of responses that are false, it appears that the proportion of true/false answers are pretty similar. Therefore this may not have as significant of an impact on predicting the overall star rating for a restaurant.



This offers a unique attribute to consider when building a model. There appears to be a direct relationship between casual attire and star ratings (star rating increase as the percent of reviews responding casual attire increases). Therefore this could offer an attribute that could help predict the overall star rating of a model. One thing to consider is the low response rate for formal attire. Only 2.7% of responses for formal attire were made for this attribute. Therefore this could be a factor that contributes to bias (i.e, customers that dine at formal restaurants may not be as likely to rate the restaurant, therefore the data set may be biased towards more casual restaurants). Furthermore the standards for customers at formal restaurants may be different than customers at casual restaurants, which could further be a potential explanation of bias.

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.

The best model was the SVM model. For the attributes, a model was run on the attributes referenced above. The initial thought was that alcohol and outdoor seating would have predictive capabilities for star rating because there was a direct correlation between the star ratings and percent of each attribute factor (with total reviews being the denominator). However the models didn’t show predictive capabilities with these attributes. At closer look, the relative proportion of responses at the star rating level (i.e, total responses for star level 1, level 2, etc.) was pretty consistent for each factor as the star levels increased. Therefore there wasn’t as much discrimination in the data to effectively and accurately predict the rating for the attributes identified.