DATA MINING (IDS 572)

Assignment 3

Harika Kalyani Lakhinena - hlakhi2@uic.edu - 676218205

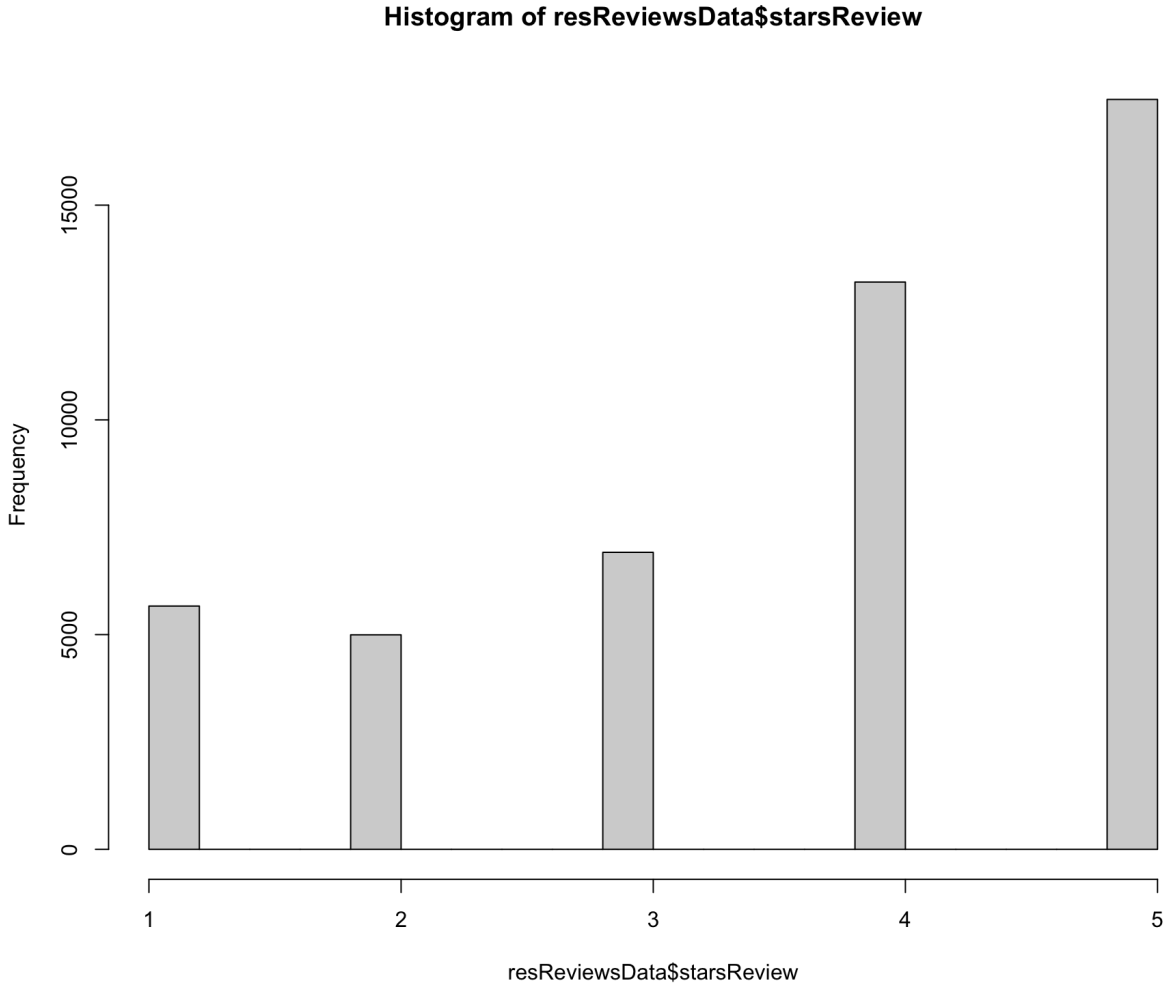
Roshan Dhanasiri - rdhana5@uic.edu - 676134844

Sathwik Reddy Maddi - smaddi9@uic.edu -

1. Explore the data.

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

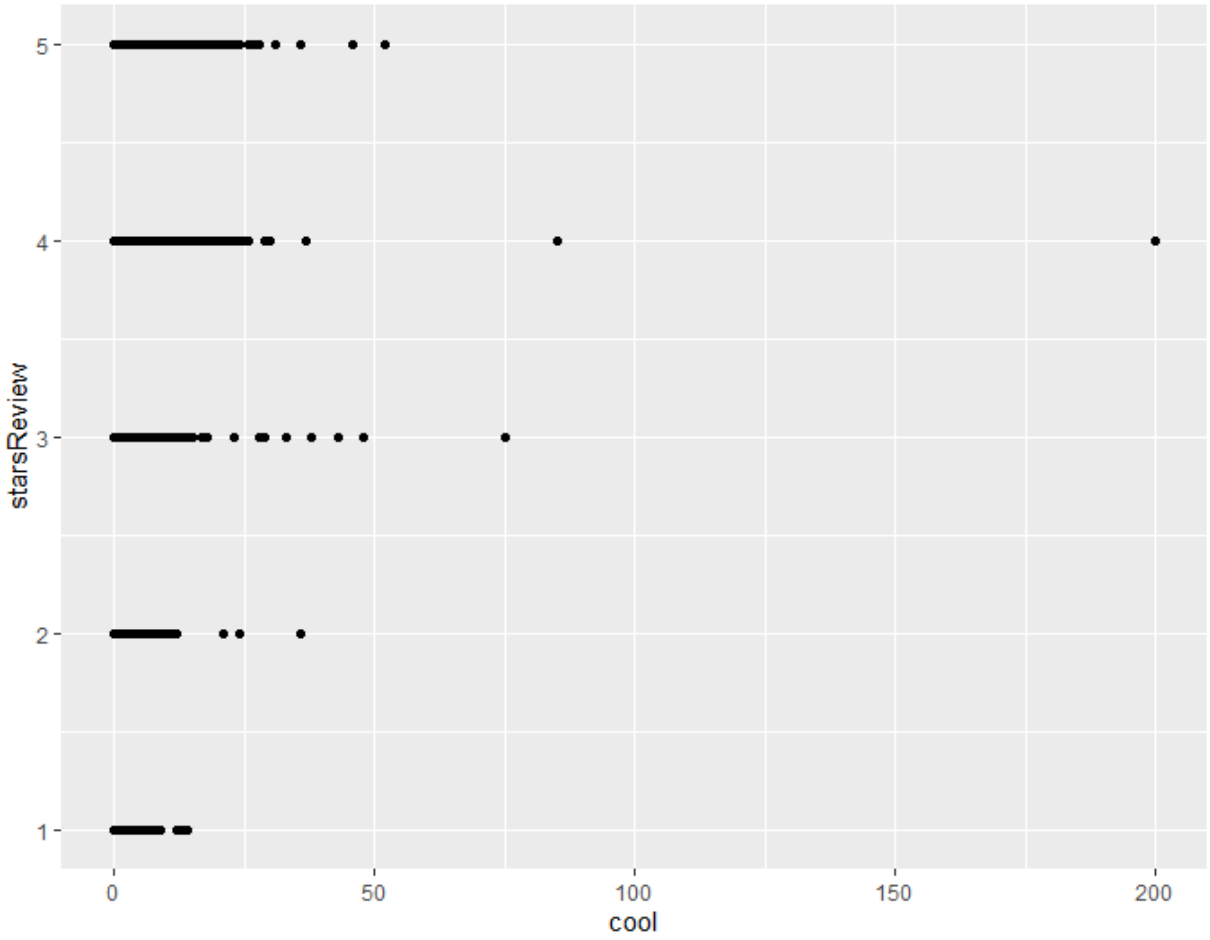
Ans. The star rating distribution is depicted in the diagram below. As seen, the majority of the restaurants have received a rating of 4 or above, with only a handful receiving a rating of 3 or worse.

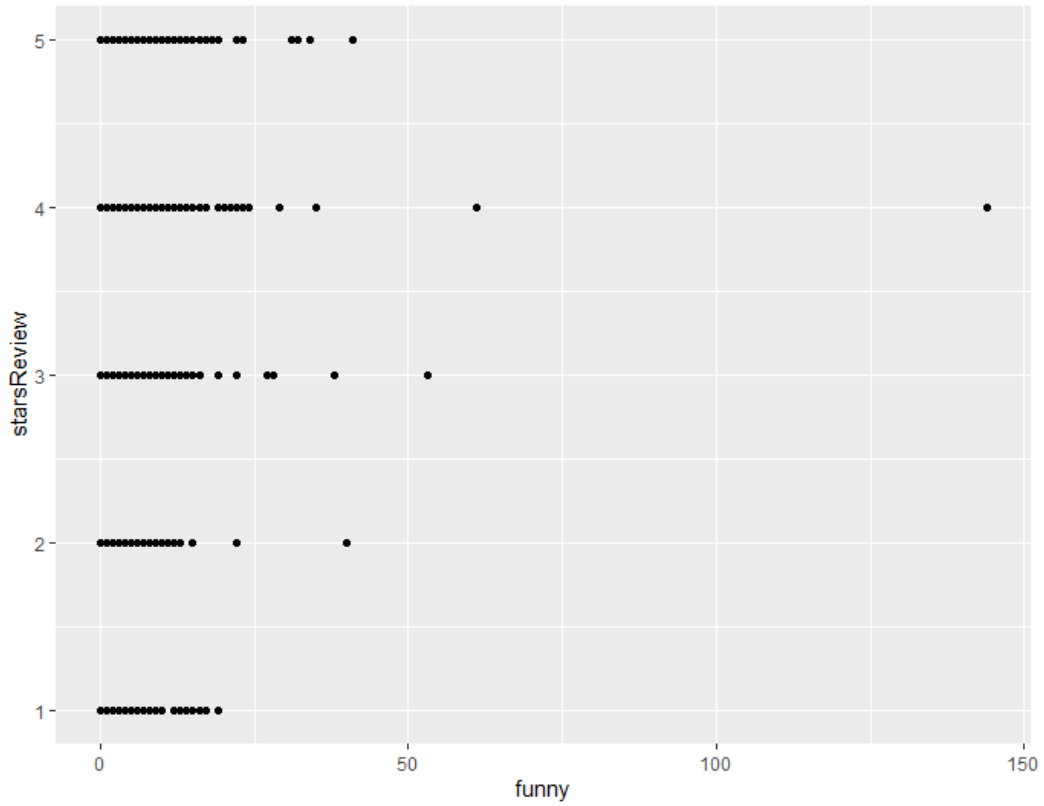


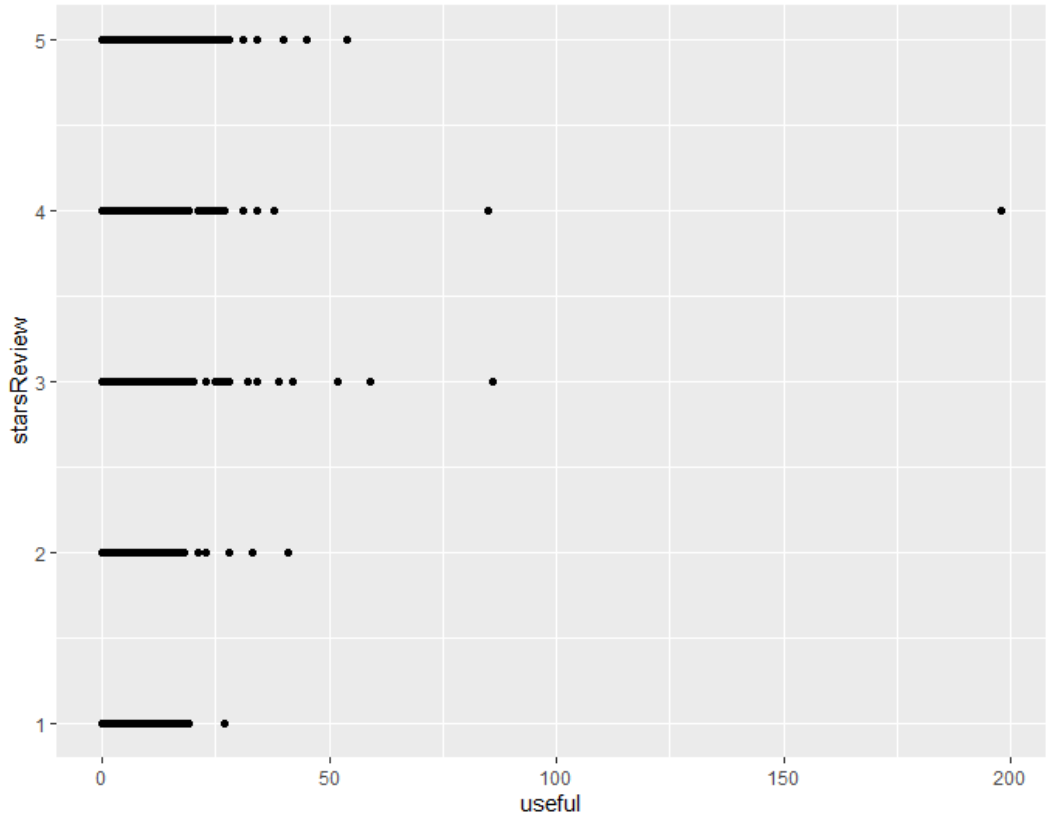
We used the histogram for average star ratings to define a threshold for producing a label indicating "positive" and "negative". 3.659 was chosen as the threshold since it generally separated the values into equal portions.

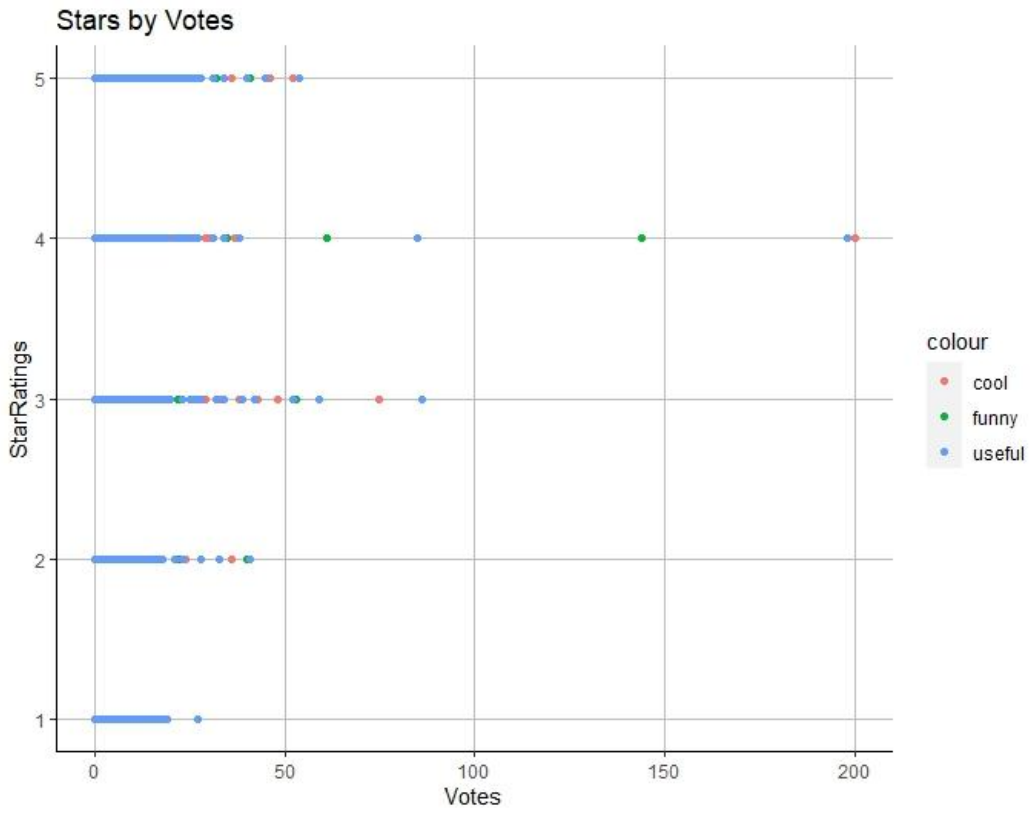
```
> mean(resReviewsData$starsReview)
[1] 3.659246
```

Relations of starsReview with Funny, Cool, Useful:

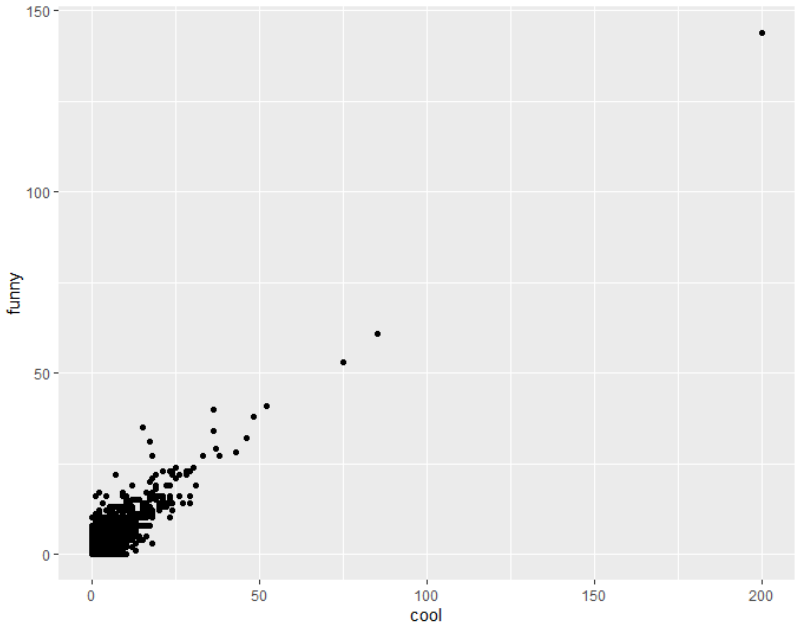



5
4
3
2
1
starsReview
0
50
100
150
funny


5
4
3
2
1
starsReview
0
50
100
150
200
useful

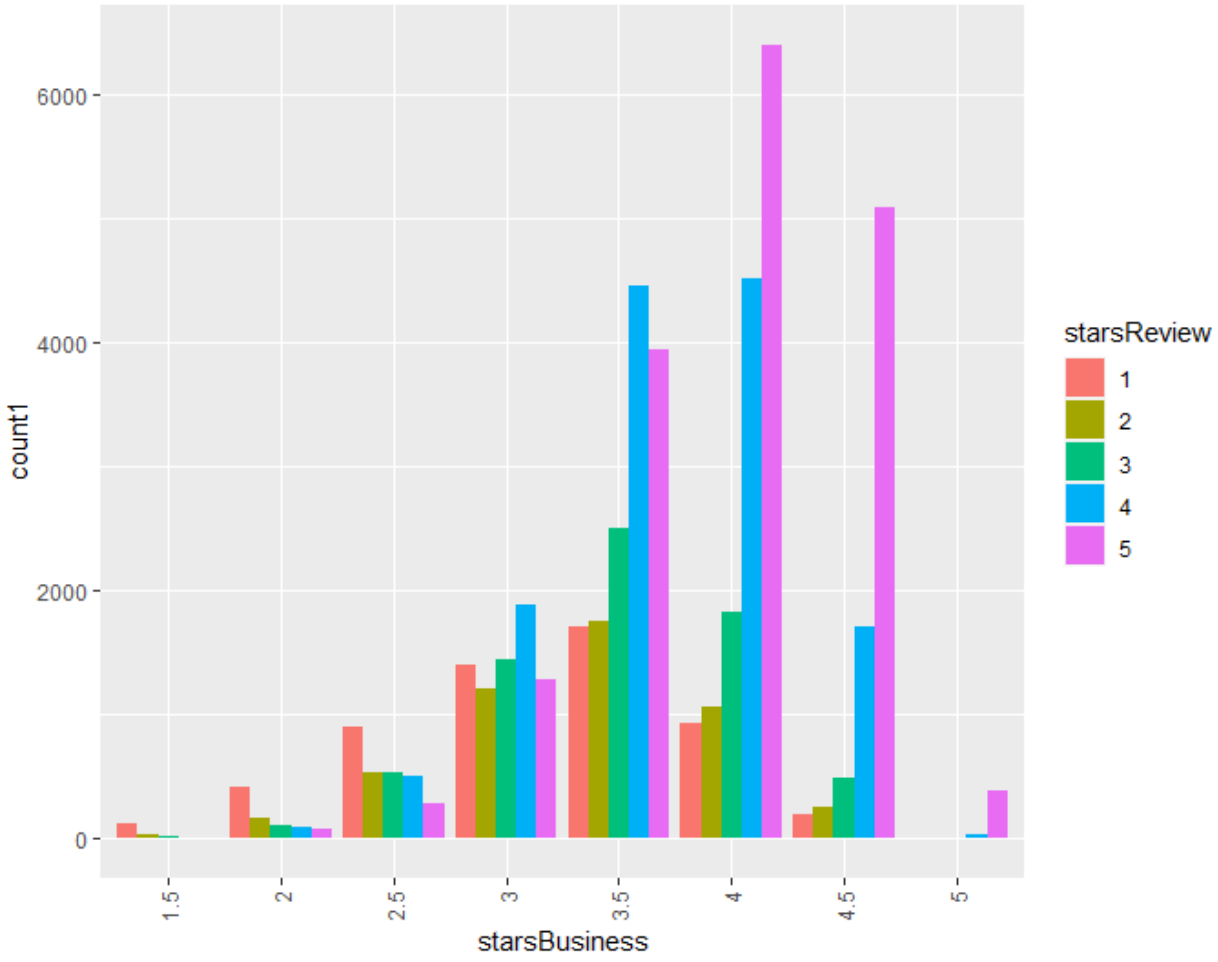


We expected higher scores for the terms funny, cool, and useful because these are all positive adjectives. The majority of evaluations using these terms are rated 3 or above, as predicted. A normal curve skewed towards higher ratings can also be seen.



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

Ans: The relationship between Business-stars and the restaurant's star rating is proportionate. The larger the number of business stars, the better the restaurant's rating. It's precisely what you'd anticipate. We can reliably predict that the total restaurant star rating will be within 0.1 of the business-stars rating if we know the business-stars rating.



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

We checked the average star rating for each word along with the number of times each word was mentioned in a positive / negative review, which we judged according to the ratings given. It can be observed that in most positive / negative reviews, the words 'food' and 'service' were mentioned more frequently. This can be attributed to the fact that most consumers judge a restaurant on the aspects of the quality and taste of food that they are served along with the kind of service that they are provided. This is why, in almost all the reviews from ones that are rated 1 to those that are given highly positive reviews, have included the words 'food', 'service' and 'waitress' or 'server'. This goes to show that these are two important factors that customers look out for when judging the overall quality of a restaurant. This is also personally relatable through the fact that when we personally visit a restaurant, regardless of all other aspects, if the food is enjoyed, it is more than probable that the restaurant will receive a good review. Also, the good service provided by the staff at the restaurant is also indicative of the fact that they hold a high regard for the experiences of customers regardless of how they perform in the kitchen, as they're two different aspects of their business.

It can be seen that the word 'love', which was quite frequently used in customer reviews was analyzed. Apparently, 'love' is highly correlated with positive reviews and emotions, as is indicated in the second table below. People like to talk more about the things they loved about an experience, which is why it was mentioned more in positive reviews compared to negative reviews, where they most likely would have mentioned what they would have rather loved from an experience, or spoken about what they loved about their experience but rather disliked.

#Check words by star rating of reviews

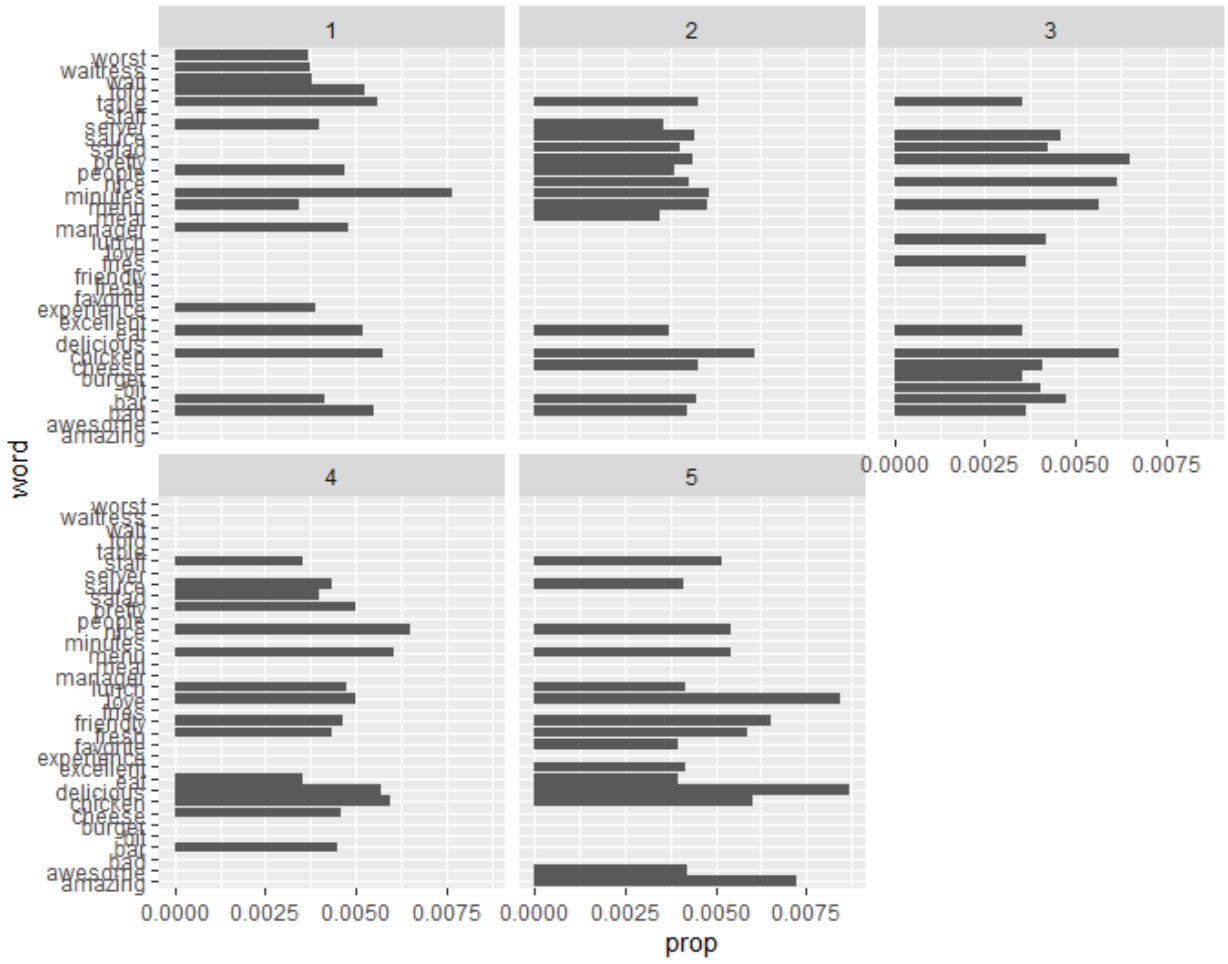
	starsReview	word	n
	<int>	<chr>	<int>
1	5	food	12619
2	4	food	9450
3	3	food	6229
4	5	service	6218
5	1	food	5633
6	2	food	5323
7	4	service	4777
8	5	delicious	4574
9	5	love	4441
10	5	time	4431

... with 46,965 more rows
> |

#check the proportion of 'love' among reviews with 1,2,..5 stars

```
# Groups:   starsReview [5]
  starsReview word      n    prop
        <int> <chr> <int>   <dbl>
1           5 love   4441 0.00844
2           4 love   2531 0.00500
3           3 love    892 0.00307
4           2 love    473 0.00218
5           1 love    331 0.00151
> |
```

But observing the overall number of reviews, it can be concluded that the words ‘awesome’, ‘amazing’, and ‘delicious’ can be associated with positive reviews whereas negative words such as ‘bad’, ‘worst’, and direct comments regarding the ‘waitress’ who served them were made. This can be used to strengthen our argument regarding the main points of experience a customer looks out for, with one being the kind of service provided (by the waiter or waitress).
plot without words like ‘food’, ‘time’,… which occurs across rating

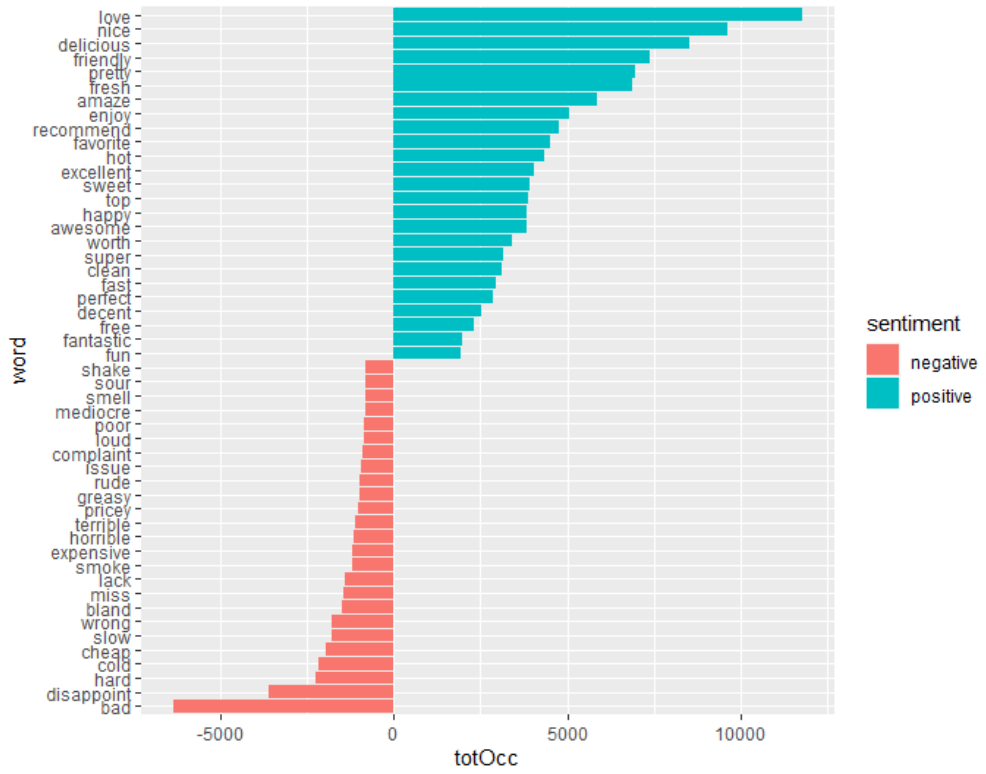


#Which words are associated with higher/lower star ratings in general ?
#Top 20

```
Selecting by totWS
# A tibble: 20 x 2
   word        totWS
   <chr>       <dbl>
 1 amazing    0.0523
 2 bar        0.0625
 3 cheese     0.0622
 4 chicken    0.0906
 5 delicious  0.0762
 6 eat        0.0571
 7 food       0.334
 8 fresh      0.0596
 9 friendly   0.0648
10 love       0.0773
11 lunch      0.0604
12 menu       0.0811
13 nice       0.0824
14 pretty     0.0611
15 restaurant 0.0904
16 salad      0.0578
17 sauce      0.0638
18 service    0.162
19 staff      0.0580
20 time       0.126
```

#Last 20

```
# A tibble: 20 x 2
   word                totWS
   <chr>               <dbl>
 1 acknowledgment 0.0000619
 2 appalling      0.0000642
 3 boyardee       0.0000654
 4 canceled       0.0000571
 5 choking        0.0000641
 6 defensive      0.0000641
 7 disgrace       0.0000646
 8 drawer         0.0000629
 9 filter         0.0000644
10 filth          0.0000595
11 hagar          0.0000515
12 inexcusable    0.0000607
13 inspector      0.0000632
14 roaches        0.0000619
15 robbery        0.0000654
16 scolding       0.0000617
17 spills         0.0000607
18 swallowed      0.0000631
19 tongs          0.0000604
20 violent        0.0000655
> |
```



From the above graph, it can be seen that the words ‘love’, ‘nice’, ‘delicious’, ‘friendly’ and ‘friendly’ are the words that are highly associated with positive emotions, mostly regarding the food and experience the customers have.

Where as, the words ‘bad’, ‘disappoint’, ‘hard’ and ‘cold’ are words that can also be attributed to the negative experience / food they would have had at the restaurant.

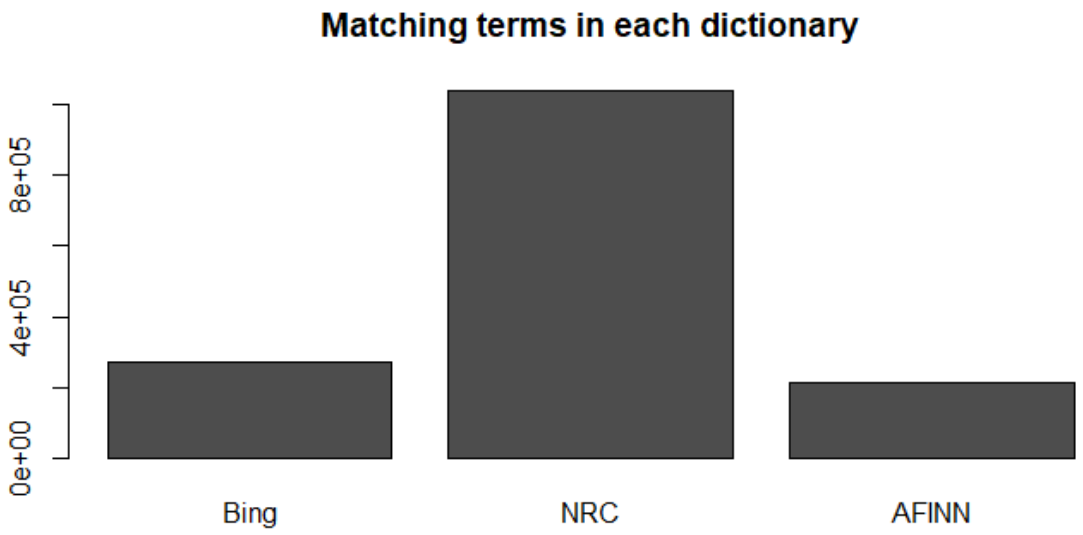
It is also apparent that the words correlated with positive experiences are more frequently mentioned in the restaurant’s reviews, which can help us conclude that on an average, a customer has more of a positive experience rather than a negative one.

Hence, words can indeed be classified as positive or negative and subsequent analysis will show that the positive/negative words used can be correlated with the experience a customer has in the restaurant.

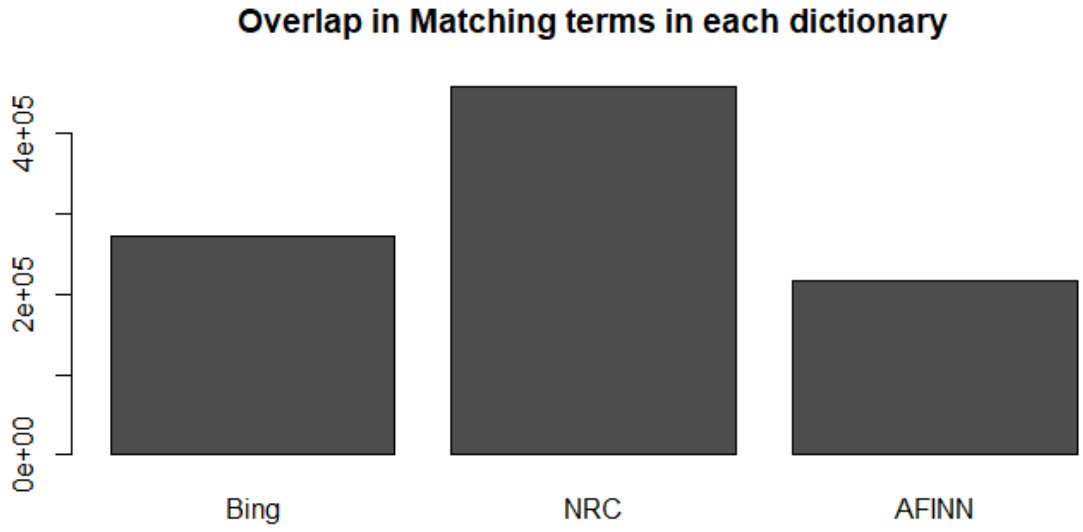
---

3. We will consider three dictionaries, available through the tidy text package –
(i) the NRC dictionary of terms denoting different sentiments,
(ii) the extended sentiment lexicon developed by Prof Bing Liu, and
(iii) the AFINN dictionary which includes words commonly used in user-generated content on the web.
The first provides lists of words denoting different sentiments (for eg., positive, negative, joy, fear, anticipation, …), the second specifies lists of positive and negative words, while the third gives a list of words with each word being associated with a positivity score from -5 to +5.
(a) How many matching terms (i.e. terms in your data that match the dictionary terms) are there for each of the dictionaries?

Ans:



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



NRC looks better at picking up sentiment information from text data than Bing and AFINN because it has the most number of matching terms.

---

4. Consider using the dictionary-based positive and negative terms to predict the 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?

Ans:

We used the following procedure to calculate ratings for each review based on dictionaries' output:
<u>Bing:</u>

In relation to a word, Bing gives "positive" and "negative" sentiment. We got a proportion of positive words and a proportion of negative words for each review. Then we subtracted the proportion of negative words from the proportion of positive words to get the final Sentiment score for the review. Then we assigned a positive sentiment value of 1 to positive reviews and a negative sentiment value of –1 to negative reviews.

NRC:

NRC comprises emotions like 'anger', 'anticipation', 'disgust', 'fear', 'joy', 'negative', 'positive', 'sadness', 'surprise', 'trust'.
These feelings were categorized as follows:
('anger', 'disgust', 'fear', 'sadness', 'negative') = "negative"
('positive', 'joy', 'anticipation', 'surprise', 'trust') = "positive"
We then used the same approach we did for Bing to get the sentiment score.

Afinn:

Afinn assigns a numerical value to each word according to its positive or negative sentiment. We aggregated values from the Afinn vocabulary to get a sentiment score for a review. Then we assigned a 1 to reviews with a positive score and a –1 to reviews with a negative score.

Bing Confusion Matrix and Accuracy

```
> bing_conMat
      predicted
actual    -1     1
    -1  7974  2389
    1   4326 25590
> bing_acc
[1] 0.8332878
```

NRC Confusion Matrix and Accuracy

```
> nrc_conMat
      predicted
actual    -1     1
    -1  4089  6525
    1   2167 28287
> nrc_acc
[1] 0.788351
```

AFFIN Confusion Matrix and Accuracy

```
> afinn_conMat
      predicted
actual    -1     1
    -1  6235  3855
    1   2574 26695
> afinn_acc
[1] 0.8366574
```

Afinn Dictionary provides slightly better accuracy than Bing Dictionary, while NRC Dictionary provides the lowest accuracy.

---

5. Develop models to predict review sentiment.
For this, split the data randomly into training and test sets. To make run times manageable, you may take a smaller sample of reviews (minimum should be 10,000).
One may seek a model built using only the terms matching any or all of the sentiment dictionaries, or by using a broader list of terms (the idea here being, maybe words other than only the dictionary terms can be useful). You should develop at least three different types of models (Naïve Bayes, and at least two others of your choice ….Lasso logistic regression (why Lasso?), xgb, random forest (use ranger for faster run-times).
Report on the performance of the models you develop. Compare performance with that in part 4 above.
Explain your findings (and is this what you expected).
(a)How do you evaluate performance? Which performance measures do you use, and why?
(b)Which types of models does your team choose to develop, and why?
Do you use term frequency, tfidf, or other measures, and why?
(c) Develop models using only the sentiment dictionary terms – try the three different dictionaries; how do dictionaries compare in terms of predictive performance? Then with a combination of the three dictionaries, ie. combine all dictionary terms.
What is the size of the document-term matrix?
Should you use stemming or lemmatization when using dictionaries? Why?

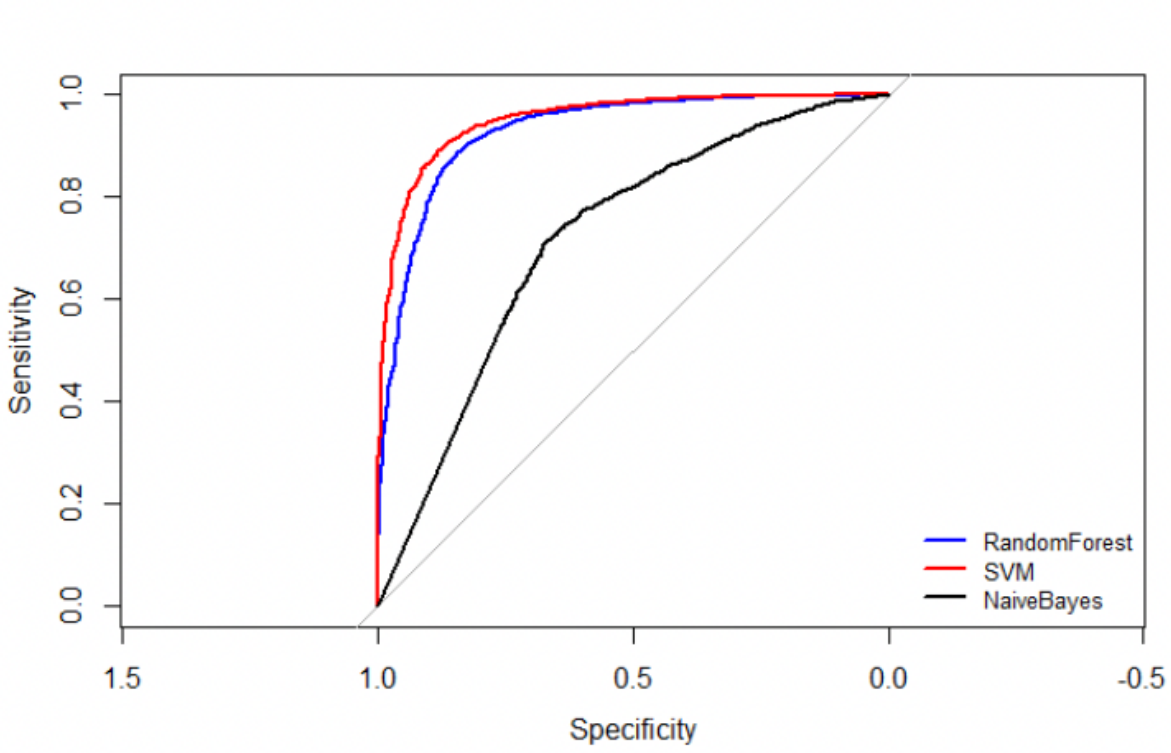
Ans:
We used Random Forest, SVM, and Naive Bayes to create a classification model. We used a portion of the original data to reduce run durations because SVM performance adjustments required significantly more processing power and RAM.

We chose TF-IDF because it assigns a value to a word's relative relevance inside a review and penalizes it if the word occurs in all reviews.

Because we need to match words with dictionary terms, lemmatization should be implemented. We won't be able to locate a match for that term in the dictionary since the final result of stemming may not be a word in the English dictionary, even if some version of that word is available in that dictionary.

Lasso regularization would have been helpful because it penalizes and makes coefficients of irrelevant characteristics 0. Because our feature vector is made up of words from a review, the size is very large. As a result, deleting unnecessary terms would help our model to better match the data.

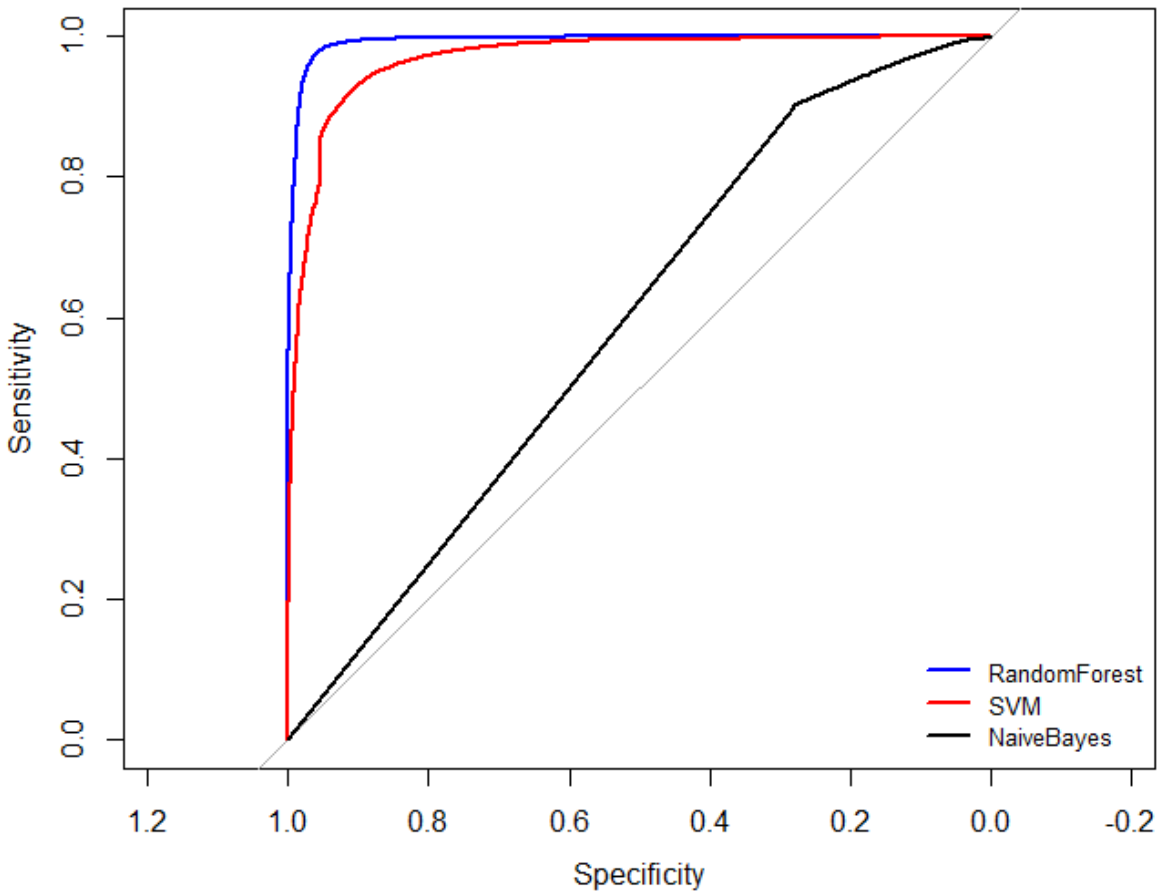
Bing Dictionary:



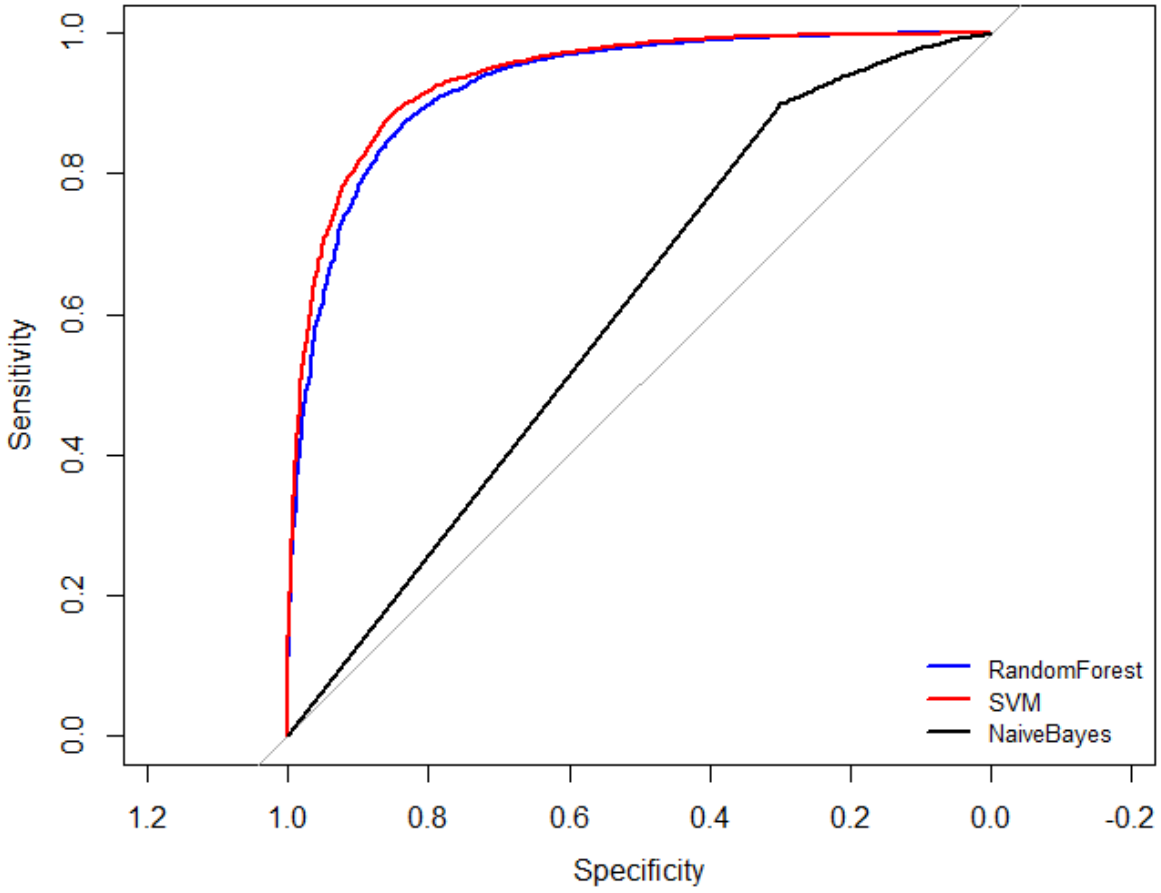
Bing ROC for Test Data

Size of Document-Term Matrix = [20,116 x 1,227]

NRC Dictionary:



NRC ROC for Training Data



NRC ROC for Test Data

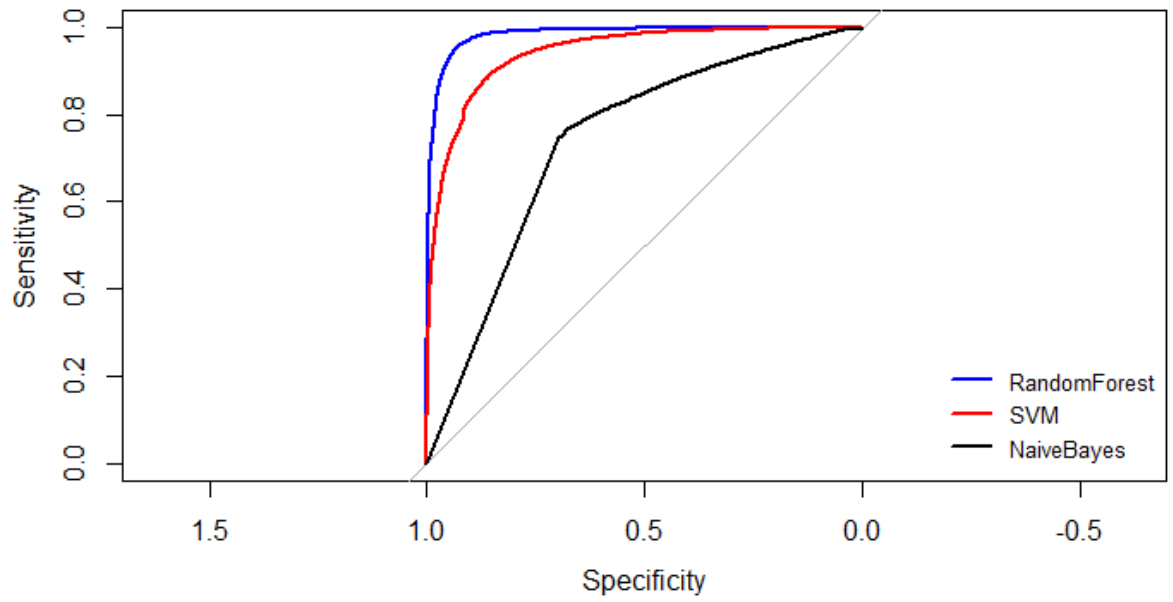
```
> ##Confusion matrix
> table(actual=revDTM_sentiNrc_trn$hiLo, preds=rfSentiNrc_predTrn[,2]>0.5)
      preds
actual FALSE  TRUE
    -1  7182   766
    1    150 22703
> table(actual=revDTM_sentiNrc_tst$hiLo, preds=rfSentiNrc_predTst[,2]>0.5)
      preds
actual FALSE TRUE
    -1  1737  929
    1    307 7294
```

Confusion Matrix Trn and Tst

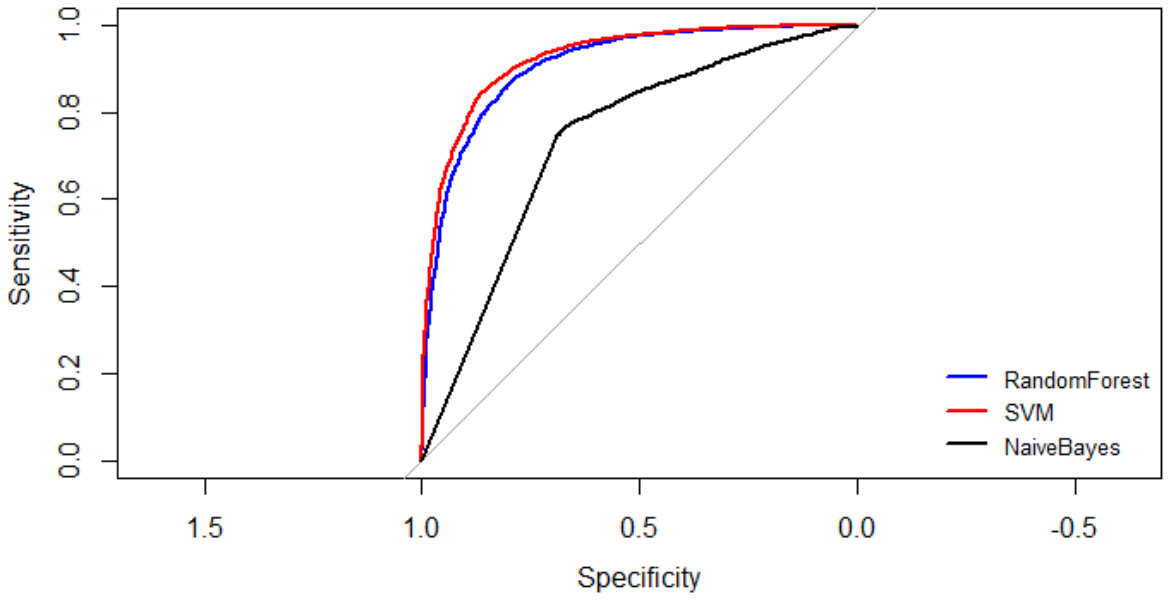
```
> sum(diag(rfNrc_table))/sum(rfNrc_table)
[1] 0.8796143
> sum(diag(svmNrc_table))/sum(svmNrc_table)
[1] 0.8876985
> sum(diag(nbNrc_table))/sum(nbNrc_table)
[1] 0.7439369
```

SUM

AFINN Dictionary:



AFFIN ROC For Training Data



AFFIN ROC For Test Data

```
> ##Confusion matrix
> table(actual=revDTM_sentiAfinn_trn$hiLo, preds=rfSentiAfinn_predTrn[,2]>0.5)
      preds
actual FALSE  TRUE
    -1  6510  1062
    1    325 21622
> table(actual=revDTM_sentiAfinn_tst$hiLo, preds=rfSentiAfinn_predTst[,2]>0.5)
      preds
actual FALSE TRUE
    -1  1667  851
    1    439 6883
```

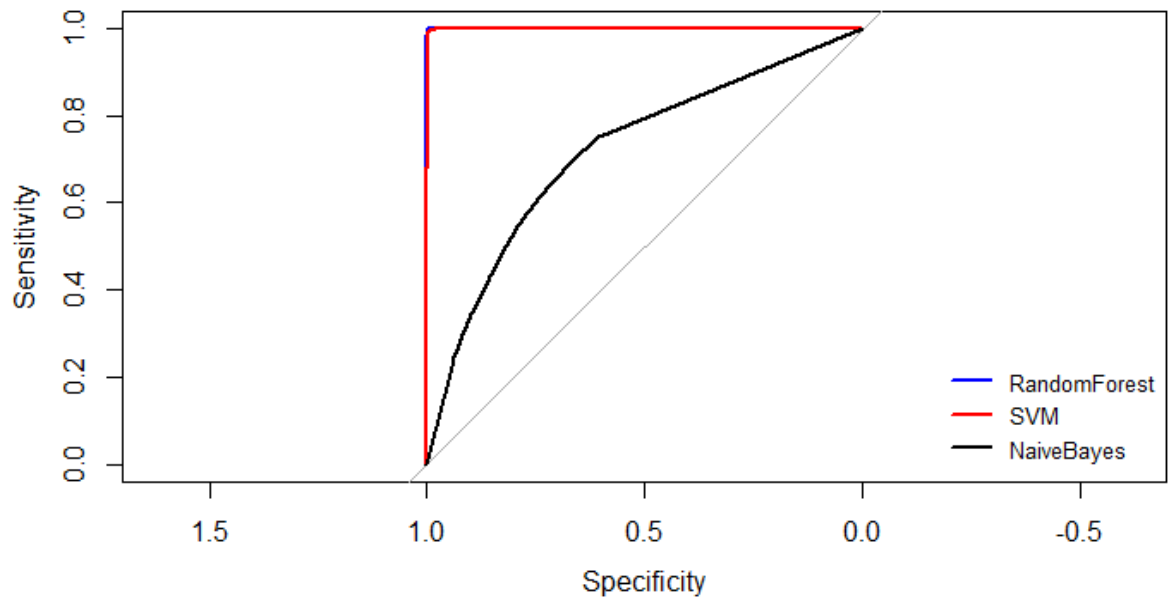
Confusion Matrix Trn and Tst

```
> sum(diag(rfAfinn_table))/sum(rfAfinn_table)
[1] 0.8689024
> sum(diag(svmAfinn_table))/sum(svmAfinn_table)
[1] 0.8778455
> sum(diag(nbAfinn_table))/sum(nbAfinn_table)
[1] 0.7568089
```

SUM

(d) Develop models using a broader list of terms (i.e. not restricted to the dictionary terms only) – How do you obtain these terms? Will you use stemming or lemmatization here, and why?

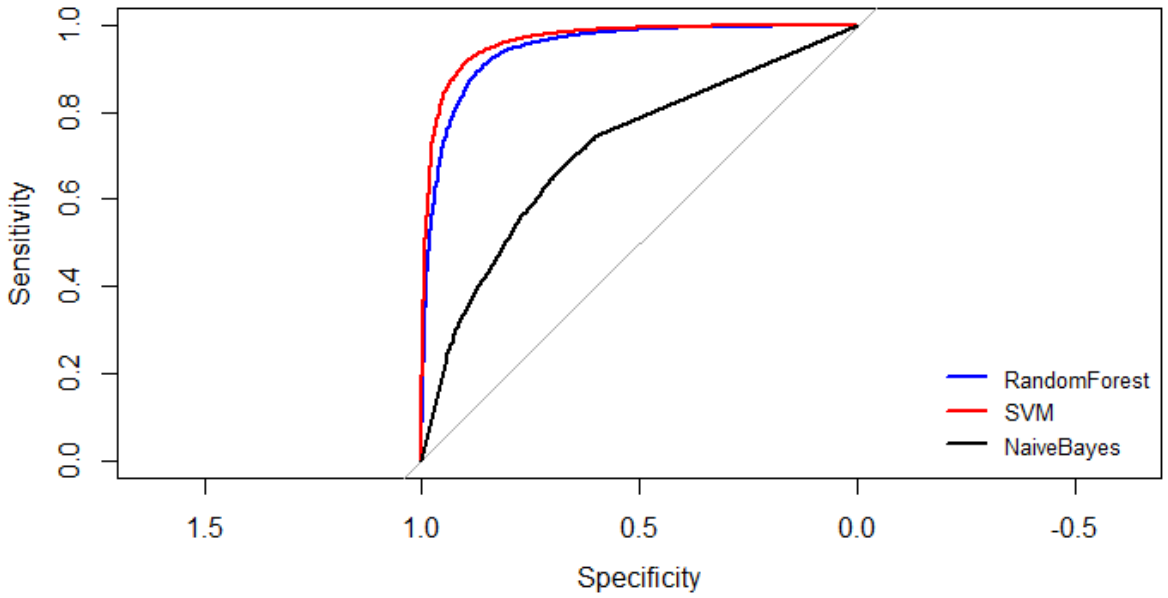
Because we aren't matching the term to a dictionary, stemming or lemmatization shouldn't be an issue. We've kept lemmatization going for the purpose of consistency and comparability. We further lowered the word count by including terms that appear in between 30% and 90% of all documents.



Broader list of terms ROC for Training Data

```
> # AUC Values for training data
> auc(as.numeric(revDTMBroad_trn$hiLo), rfSentiBroad_predTrn[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.9999
> auc(as.numeric(revDTMBroad_trn$hiLo), attr(svmSentiBroad_predTrn, "probabilities")[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.9993
> auc(as.numeric(revDTMBroad_trn$hiLo), nbSentiBroad_predTrn[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.7186
```

AUC for Training Data



Broader list of terms ROC for Test Data

```
> # AUC Values for test data
> auc(as.numeric(revDTMBroad_tst$hiLo), rfSentiBroad_predTst[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.9474
> auc(as.numeric(revDTMBroad_tst$hiLo), attr(svmSentiBroad_predTst, "probabilities")[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.9656
> auc(as.numeric(revDTMBroad_tst$hiLo), nbSentiBroad_predTst[,2])
Setting levels: control = 1, case = 2
Setting direction: controls < cases
Area under the curve: 0.7135
```

AUC for Testing Data

To assess the model's performance, we employed ROC curves and their accompanying AUC values. To compare the models, we got accuracy using a threshold of nearly 0.5 for each.

For all combinations, SVM fared marginally better than Random Forest in terms of AUC values. When we utilize broader terms, rather than employing a dictionary, the strategy appears to match the data well.

When compared to component c, the RF and SVM models appear to perform better in terms of accuracy.

---

6. Consider some of the attributes for restaurants – this is specified as a list of values for various

attributes in the ‘attributes’ column. Extract different attributes (see note below).
(a) Consider a few interesting attributes and summarize how many restaurants there are by values of these attributes; examine if star ratings vary by these attributes.

We analyzed restaurants based on different attributes such as ‘RestaurantsPriceRange2’, ‘Alcohol’ and ‘GoodMealFor.Dinner’. ’

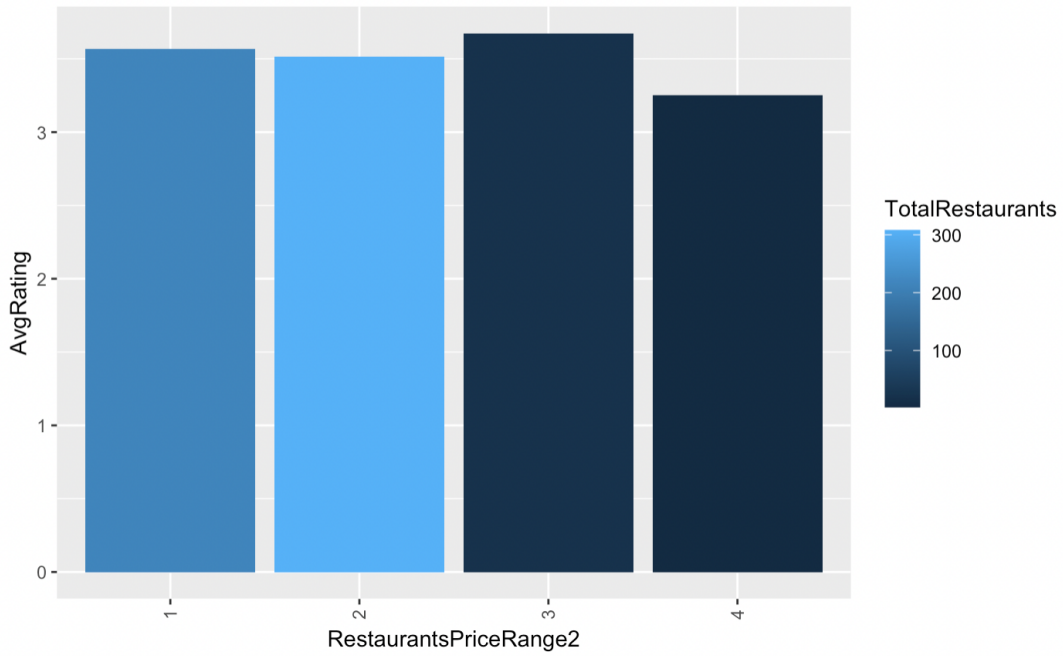
It can be seen that for ‘RestaurantsPriceRange2’, we have a total of 10,301 reviews given out of which 9513 of the total number of predictions we have are correct (either positive or negative) have been correct reviews.

Similarly, for the attributes ‘Alcohol’ and ‘GoodMealFor.Dinner’, 10,104 and 10,166 reviews have been given, out of which 9322 and 9379 have been correctly predicted.

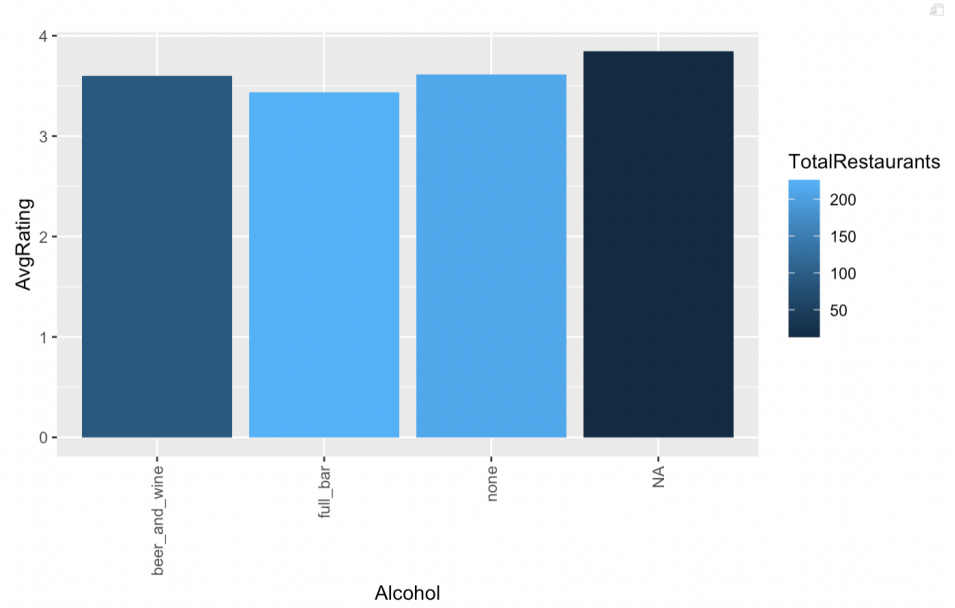
(b) For one of your models (choose your ‘best’ model from above), does prediction accuracy vary by certain restaurant attributes? You do not need to look into all attributes; choose a few which you think may be interesting, and examine these.
Note: for question 6, you will consider the values in the ‘attribute’ column. This has values of multiple attributes, separated by a ‘|’. Further, some of the values, like Ambience, carry a list of True/False values (like, for example, Ambience: {'romantic': False, 'intimate': False, 'classy': False, 'hipster': False, …}. Care must be taken to extract values for different attributes. You can consider developing a separate data frame with review_id, attribute, and then process this further to extract values for the different attributes.

Ans: business_id, starsBusiness and attribute_names were the columns and the attribute_val were the rows that were present in the dataframe that we had derived.

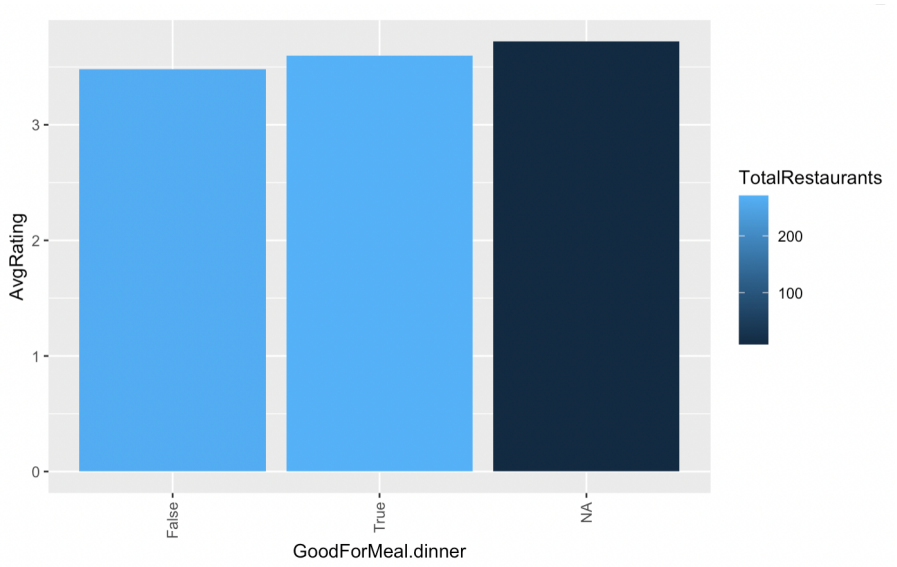
```
$ DriveThru                  <chr> NA, NA, NA, NA, NA, NA, NA, " True", NA, NA, " False", NA, NA, NA, " True", " True", NA, NA, NA, NA, NA, NA, ~
$ BYOBCorkage                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " yes_free", " no", NA, NA, NA, NA, NA, " yes_free", NA, ~
$ Corkage                    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, N~
$ BYOB                       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, NA, NA,~
$ BusinessAcceptsBitcoin     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, NA,~
$ CoatCheck                  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, NA,~
$ GoodForDancing             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " False", NA, NA, NA, NA,~
$ HappyHour                  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " True", NA, NA, NA, ~
$ Smoking                    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " outdoor", NA, NA, N~
$ DogsAllowed                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, " Tru~
$ DietaryRestrictions        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ ByAppointmentOnly          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ Open24Hours                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ Ambience.romantic          <chr> "False", "False", "False", "False", "False", "False", "False", "False", "True", "False", "False", "False", "F~
$ Ambience.intimate          <chr> "False", "False", "False", "False", "False", "False", "False", "False", "True", "False", "False", "False", "F~
$ Ambience.classy            <chr> "False", "False", "False", "False", "False", "False", "False", "False", "True", "False", "False", "False", "F~
$ Ambience.hipster           <chr> "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "~
$ Ambience.divey             <chr> "False", "True", "False", "False", "False", "True", "False", "False", "False", "False", "False", "False", "Fa~
$ Ambience.touristy          <chr> "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "~
$ Ambience.trendy            <chr> "False", "False", "False", "False", "False", "False", "False", "False", "False", "True", "False", "False", "F~
$ Ambience.upscale           <chr> "False", "False", "False", "False", "False", "False", "False", "False", "True", "False", "False", "False", "F~
$ Ambience.casual            <chr> "True", "False", "True", "True", "True", "True", "False", "False", "False", "False", "False", "True", "True",~
$ GoodForMeal.dessert        <chr> "False", "False", "False", "False", "False", "False", "False", "True", "True", "False", "False", "False", "Fa~
$ GoodForMeal.latenight      <chr> "False", "False", "False", "False", "False", "True", "False", "False", "False", "False", "False", "False", "F~
$ GoodForMeal.lunch          <chr> "True", "True", "False", "True", "True", "True", "False", "False", "False", "False", "True", "False", "True",~
$ GoodForMeal.dinner         <chr> "True", "True", "True", "False", "True", "True", "False", "False", "True", "True", "False", "True", "False", ~
$ GoodForMeal.breakfast      <chr> "False", "False", "False", "True", "False", "True", "False", "False", "False", "False", "False", "False", "Fa~
$ GoodForMeal.brunch         <chr> "False", "False", "False", "True", "False", "False", "True", "False", "False", "True", "False", "False", "Fal~
$ BusinessParking.garage     <chr> "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "~
$ BusinessParking.street     <chr> "False", "False", "False", "False", "True", "False", "False", "False", "False", "False", "False", "False", "F~
$ BusinessParking.validated  <chr> "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "False", "~
$ BusinessParking.lot        <chr> "True", "True", "True", "True", "False", "True", "True", "True", "False", "False", "False", "True", "False", ~
$ BusinessParking.valet      <chr> "False", "False", "False", "False", "False", "False", "False", "False", "True", "True", "False", "False", "Fa~
$ BestNights.monday          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "False", NA, NA, NA, ~
$ BestNights.tuesday         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "False", NA, NA, NA, ~
```



```
[1] " 2"
      preds
actual   -1    1
    -1 1446  313
    1   176 4560
[1] 0.9247113
[1] " 1"
      preds
actual   -1    1
    -1  582  167
    1   102 2385
[1] 0.9168727
[1] " 3"
      preds
actual  -1   1
    -1  79  17
    1    8 376
[1] 0.9479167
[1] " 4"
      preds
actual -1  1
    -1 37  8
    1   5 40
[1] 0.8555556
```



```
[1] " full_bar"
      preds
actual   -1    1
    -1 1156  249
    1   140 3242
[1] 0.9187382
[1] " none"
      preds
actual   -1    1
    -1  576  162
    1    94 2673
[1] 0.9269615
[1] " beer_and_wine"
      preds
actual   -1    1
    -1  388   83
    1    54 1287
[1] 0.9243929
```



```
[1] "False"
      preds
actual   -1    1
    -1  877  216
    1   105 2783
[1] 0.919367
[1] "True"
      preds
actual   -1    1
    -1 1243  280
    1   183 4479
[1] 0.9251415
```