CS 109B: Midterm Exam 1

Feb 23, 2017

Nikhila Ravi

The set of questions below address the task of predicting the merit of a restaurant on Yelp. Each restaurant is described by a set of business attributes, and is accompanied by a set of text reviews from customers. For the purpose of the problems below, the average rating (originally on a scale from 0 to 5) was converted into a binary variable depending on whether the average was above 3.5, in which case it is considered "good" (labeled 1), or below 3.5 in which case it is considered "bad" (labeled 0). The overall goal is to predict these binary ratings from the information provided.

The data are split into a training and test set and are in the files `dataset_1_train.txt` and `dataset_1_test.txt` respectively. The first column contains the rating for the restaurant (0 or 1), columns 2-21 contain the business attributes, and columns 22-121 contain text features extracted from the customer reviews for the restaurant. The details about the business attributes are provided in the file `dataset_1_description.txt`.

We use the bag-of-words encoding to generate the text features, where the set of reviews for a restaurant are represented by a vector of word counts. More specifically, we construct a dictionary of 100 frequent words in the customer reviews, and include 100 text features for each restaurant: the i-th feature contains the number of times the dictionary word i occurs in customer reviews for the restaurant. For example, a text feature 'fantastic' with value 18 for a restaurant indicates that the word 'fantastic' was used a total of 18 times in customer reviews for that restaurant.

Load the data:

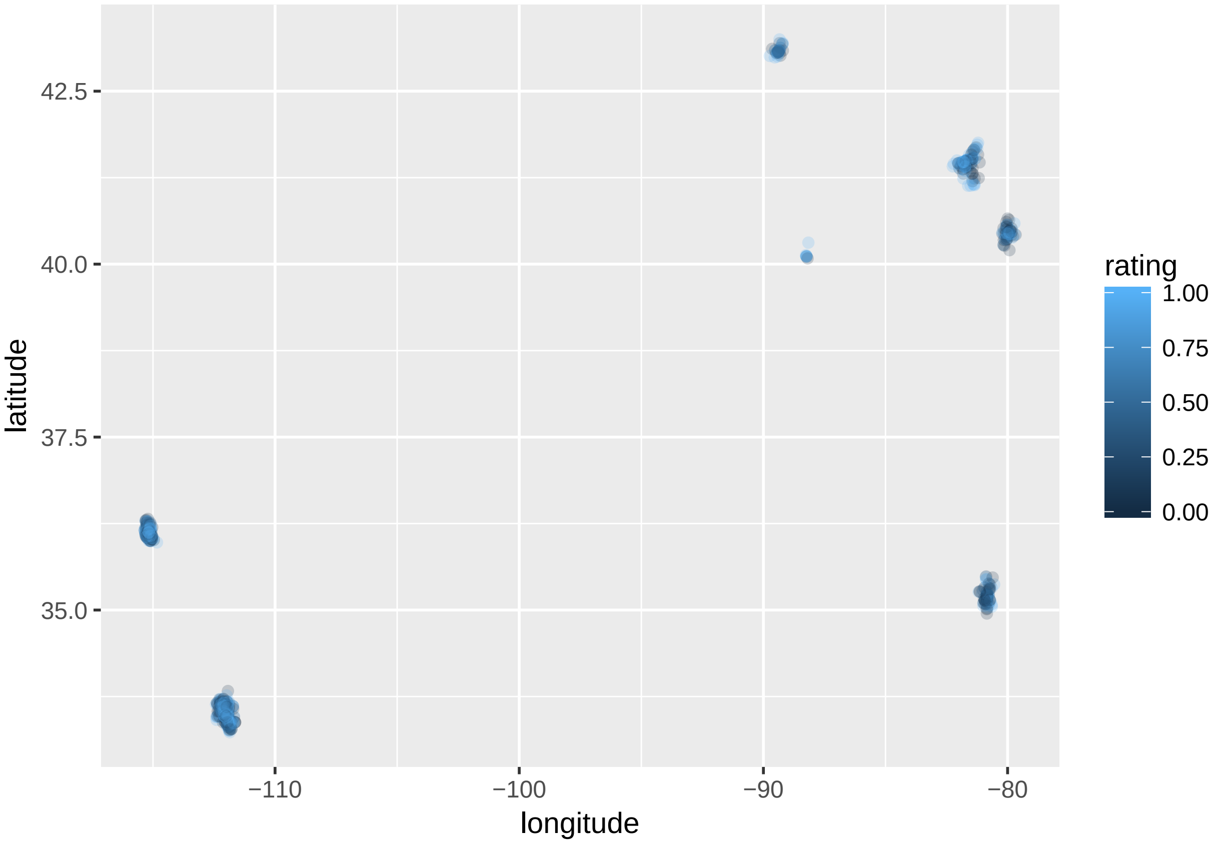
```
##
##   0   1
## 301 369
```

Problem 1 [20 points]

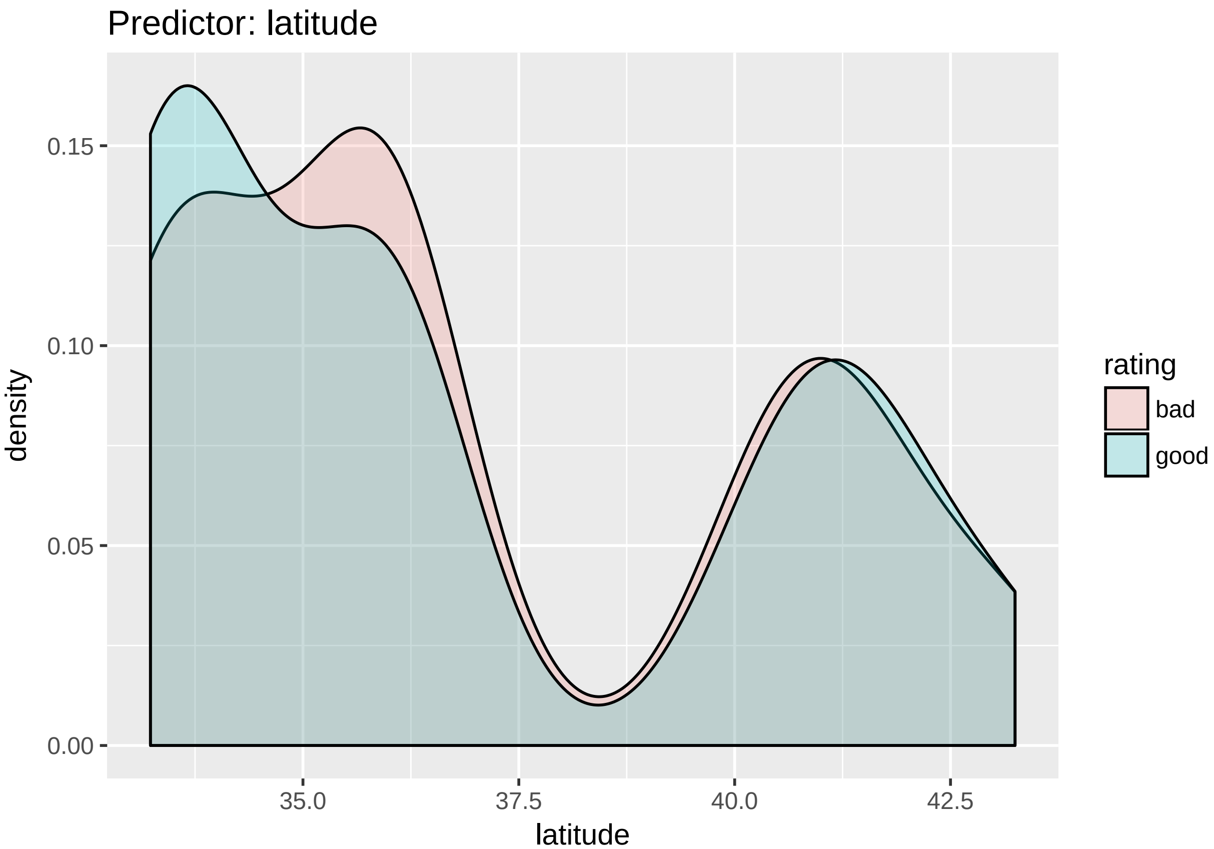
Does the location of a restaurant relate to its rating? Construct a compelling visualization to address this question, and write a brief (under 300 words) summary that clearly explains your analysis and conclusions to someone without a data science background.

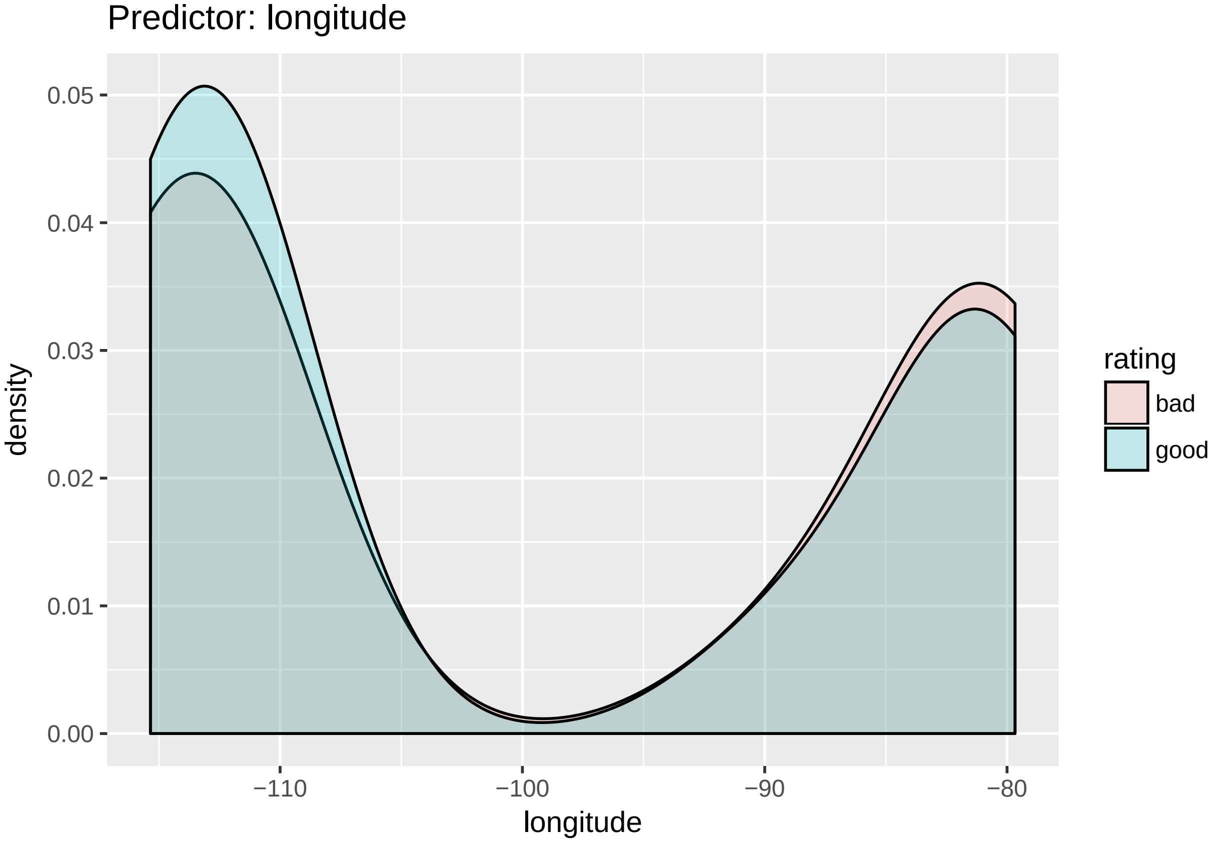
In order to determine if location is a contributing factor to the rating a number of different visualisations can be used on the training data. The important predictors to consider are the geo coordinates (latitude and longitude), postal code (which encodes both lat and long) as well as the city and state.

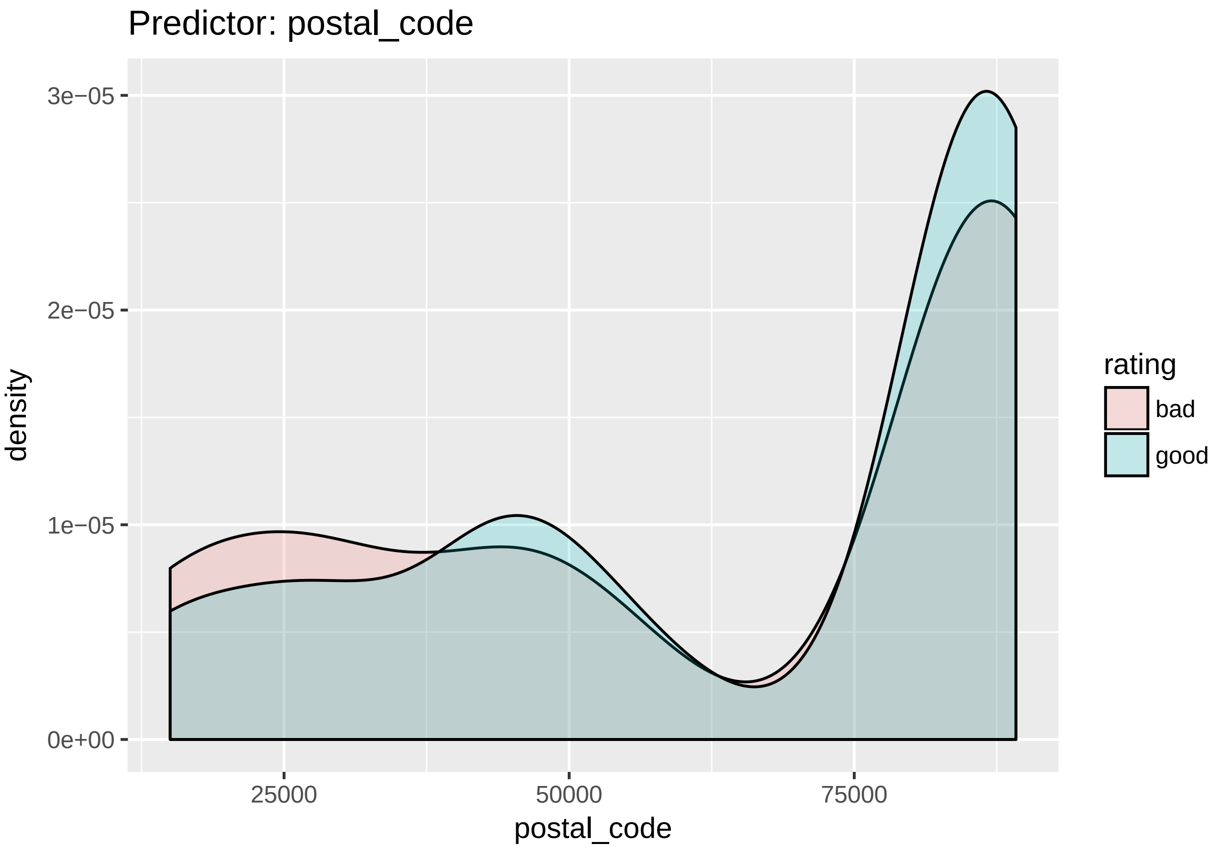
Firstly, the restaurants can be plotted on a map color coded by the rating.

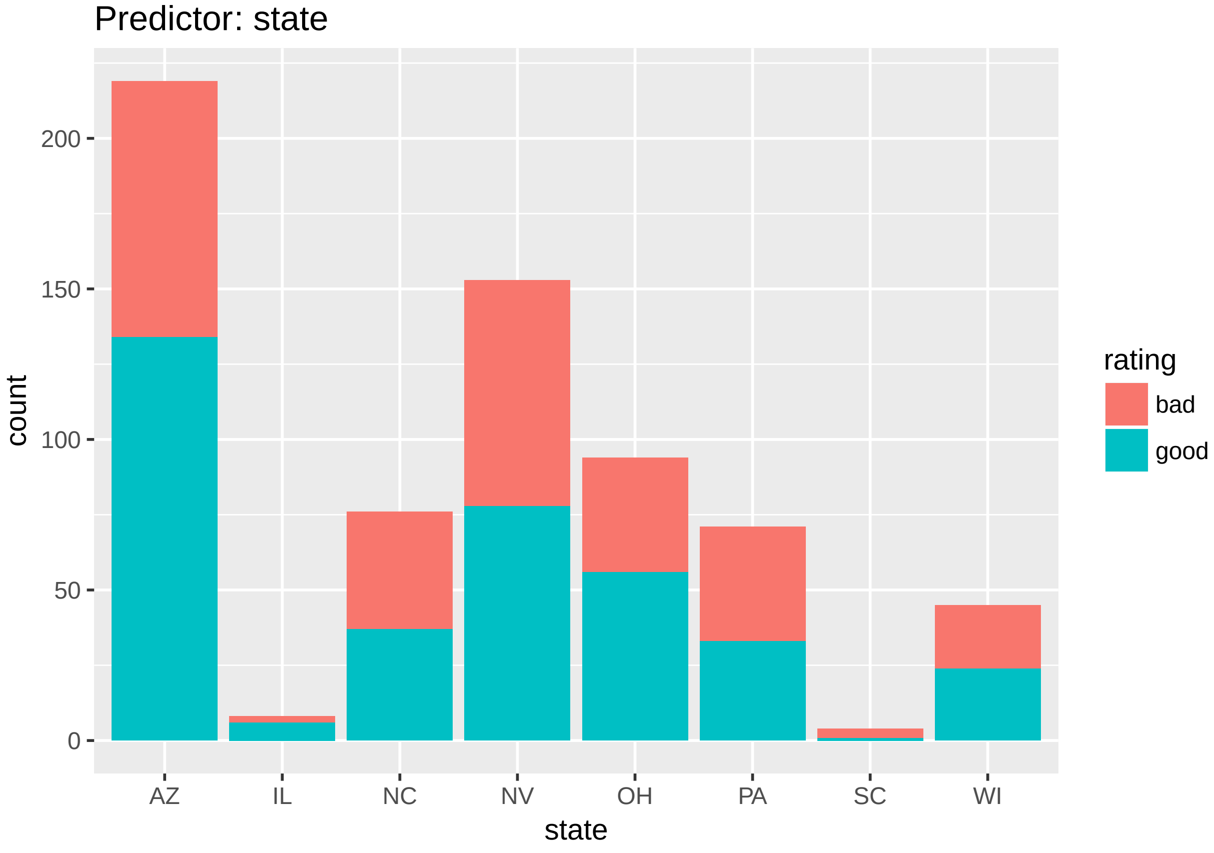


From the geoplot, it is clear that the resturants in the training set are located in a few specific regions, however the relationship between location and rating is not immediately apparent. A histogram of the latitude and longitudes as well as the postcodes color coded by the rating can be used to explore the geolocation and rating relationship further.


Predictor: latitude
density
0.15
0.10
0.05
0.00
35.0
37.5
40.0
42.5
latitude
rating
bad
good


Predictor: longitude
density
0.05
0.04
0.03
0.02
0.01
0.00
−110
−100
−90
−80
longitude
rating
bad
good

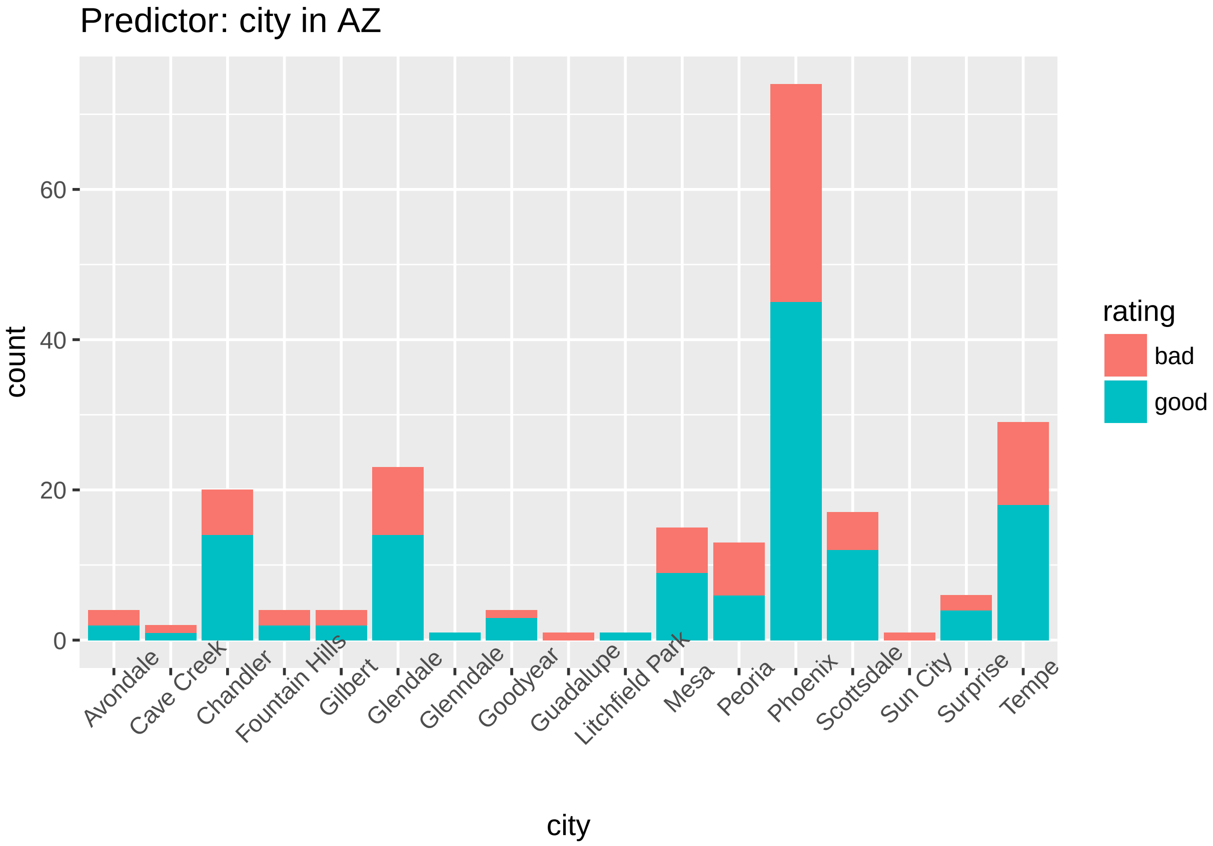

Predictor: postal_code

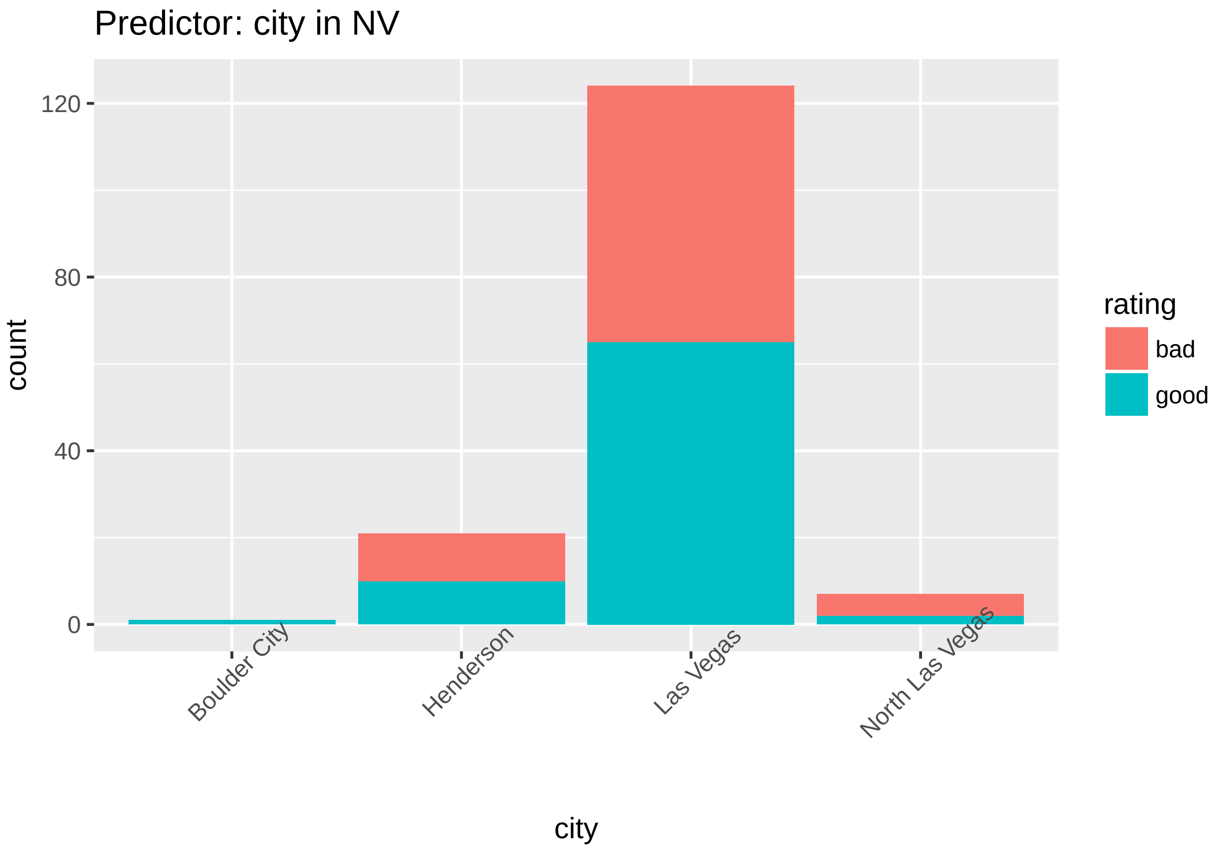

Predictor: state

As observed before, the latitude and longitude values are clustered into a few regions. This is also clear from the distribution of postcodes (which encode both the latitude and longitude). On average, there are approximately equal numbers of good and bad reviews in each region. However in two specific latitude regions, either good reviews dominate or bad reviews dominate: restaurants in the latitude range 34.5-37 have higher number of bad ratings compared to restaurants in the latitude range 33-34.5 which have a larger number of good ratings. In the longitude range -115 - -105, there are a larger number of good restaurants than bad ones.

This observation is confirmed from the distribution of ratings by postcode. There are two regions of postcodes which have a larger number of good ratings than bad ratings and one region which has a larger number of bad ratings.

Considering the plot of ratings by state, it is clear that in some states there are a much larger number of reviews than in other states and hence it is not fair to directly compare the number of good/bad restaurants. In particular IL and SC have very few restaurant ratings. A more interesting analysis can be done for states which have a large number of restaurants with rating for example AZ and NV.





Looking at the breakdown of ratings by state, the captials appear to have the largest number of recorded ratings. In Arizona, Phoenix stands out as a place with a large number of good restaurants and overall most of the cities appear to have more good than bad ratings. In Nevada, Las Vegas has the largest number of ratings but the distribution between good and bad is 50:50 so the location is less informative of the likelihood of a restaurant being good.

Overall, the location of a restaurant is only tenuously related to its rating. While in most of the locations in the training set there are approximately equal numbers of good and bad ratings, in a few specific locations, there are slightly more restaurants of one class than the other. This analysis suggests that predicting the rating of a restaurant based on its location alone would likely not be accurate.

Problem 2 [35 points]

This problem is concerned with predicting a restaurant's rating based on text features. We'll consider the Multinomial-Dirichlet Bayesian model to describe the distribution of text features and finally to predict the binary ratings.

Probability model: Let $(y_1^g, y_2^g, \dots, y_{100}^g)$ denote the total counts of the 100 dictionary words across the reviews for "good" restaurants, and $(y_1^b, y_2^b, \dots, y_{100}^b)$ denote the total counts of the 100 dictionary words across the reviews for "bad" restaurants. We assume the following *multinomial* likelihood model:

$$p(y_1^g, y_2^g, \dots, y_{100}^g \,|\, \theta_1^g, \dots, \theta_{100}^g) \,\propto\, (\theta_1^g)^{y_1^g} (\theta_2^g)^{y_2^g} \dots (\theta_{100}^g)^{y_{100}^g}$$

$$p(y_1^b, y_2^b, \dots, y_{100}^b \,|\, \theta_1^b, \dots, \theta_{100}^b) \,\propto\, (\theta_1^b)^{y_1^b} (\theta_2^b)^{y_2^b} \dots (\theta_{100}^b)^{y_{100}^b}.$$

The model parameters $(\theta_1^g, \dots, \theta_{100}^g)$ and $(\theta_1^b, \dots, \theta_{100}^b)$ are assumed to follow a *Dirichlet* prior distribution with parameter α. That is

$$p(\theta_1^g, \dots, \theta_{100}^g) \,\propto\, (\theta_1^g)^{\alpha} \dots (\theta_{100}^g)^{\alpha}$$

$$p(\theta^b_1, \ldots, \theta^b_{100}) \propto (\theta^b_1)^\alpha \ldots (\theta^b_{100})^\alpha.$$

Hence we can interpret, for example, θ^g_5 as the probability the word "perfect" is observed once in a review of "good" restaurants. For the purposes of this problem, set $\alpha = 2$.

(a) Describe briefly in words why the posterior distribution formed from a Dirichlet prior distribution and a multinomial likelihood is a Dirichlet posterior distribution? What are the parameters for the Dirichlet posterior distribution? [5 points]

The Dirichlet distribution is a conjugate prior for a multinomial likelihood. This means that a posterior distribution formed from a Dirichlet prior and Multinomial likelihood will also follow a Dirichlet distribution (with different parameters).

The Multinomial distribution represents the probability of observing each possible outcome c_i exactly X_i times in a sequence of n yes/no trials. It is parameterised by a vector of occurence counts 'N'. The Dirichlet distribution is a density over n positive numbers parameterised by a vector α - the dirichlet parameters α can be regarded as "pseudo-counts" from "pseudo-data". Therefore the Dirichlet-Multinomial posterior is a Dirichlet distribution with parameters $N + \alpha$.

(b) From a Monte Carlo simulation of the Dirichlet posterior distribution for "good" restaurants, what is the posterior mean probability that the word "chocolate" is used? From a Monte Carlo simulation of the Dirichlet posterior distribution for bad restaurants, what is the posterior mean probability that the word "chocolate" is used? **Hint**: use the `rdirichlet` function in the `MCMCpack` library. [15 points]

```r
#----Helper function to calculate posterior mean probabilities ----#
posterior_mean_A_B = function(alpha = 1, yA = NULL, n.sim = NULL) {
    # number of features
    K = length(yA)
    alpha0 = rep(alpha, K)
    # posterior parameter values post_thetaA =
    # MCmultinomdirichlet(yA, alpha0, mc = n.sim) # good
    # restaurants class
    post_thetaA = rdirichlet(n.sim, alpha + yA)
    ER_A = apply(post_thetaA, 2, mean)
    return(ER_A)
}
```

First calculate the the total word counts $(y^A_1, y^A_2, \ldots, y^A_K)$ from all good restaurants in the training set and the similar counts $(y^B_1, y^B_2, \ldots, y^B_K)$ for the bad restaurants.

```r
# the total word counts from all good restaurants in the
# training set
yA = as.numeric(apply(restaurants_train[restaurants_train$rating ==
    1, 22:121], 2, sum))
# the total word counts from all bad restaurants in the
# training set
yB = as.numeric(apply(restaurants_train[restaurants_train$rating ==
    0, 22:121], 2, sum))
```

Now the posterior mean probabilities for each word in each class can be calculated (using an alpha of 2)

```r
ER_A = posterior_mean_A_B(alpha = 2, yA, 2000)
ER_B = posterior_mean_A_B(alpha = 2, yB, 2000)
```

The probability of the word chocolate in good/bad rated restaurants can be found.

```r
words = names(restaurants_train[22:121])
# find index of word chocolate:
chocolate_index = which(words %in% c("chocolate"))
```

```r
cat("good restaurant probability of occurence of chocolate",
    ER_A[chocolate_index])
```

```
## good restaurant probability of occurence of chocolate 0.01723074
```

```r
cat("\nbad restaurant probability of occurence of chocolate",
    ER_B[chocolate_index])
```

```
##
## bad restaurant probability of occurence of chocolate 0.005538789
```

For a good restaurant, the probability of the word 'chocolate' occurring is approximately 1.72% compared to 0.5% for a bad restaurant.

The 50 words with the highest posterior mean probability for each class can be examined:

```r
good.max = which(ER_A >= sort(ER_A, decreasing = T)[50], arr.ind = TRUE)
bad.max = which(ER_B >= sort(ER_B, decreasing = T)[50], arr.ind = TRUE)

cat("Top words in good reviews\n")
```

```
## Top words in good reviews
```

```r
print(words[good.max])
```

```
##  [1] "wine"       "chocolate"  "perfect"    "sweet"      "bread"
##  [6] "breakfast"  "happy"      "loved"      "selection"  "spot"
## [11] "dessert"    "fantastic"  "super"      "bbq"        "crust"
## [16] "dish"       "asked"      "wonderful"  "waitress"   "enjoyed"
## [21] "local"      "fried"      "cool"       "shop"       "bacon"
## [26] "cream"      "absolutely" "spicy"      "hour"       "yelp"
## [31] "enjoy"      "prices"     "italian"    "feel"       "friends"
## [36] "tea"        "pork"       "wings"      "ribs"       "places"
## [41] "pasta"      "friend"     "cooked"     "outside"    "different"
## [46] "free"       "told"       "options"    "town"       "huge"
```

```r
cat("\nTop words in bad reviews\n")
```

```
##
## Top words in bad reviews
```

```r
print(words[bad.max])
```

```
##  [1] "sum"        "dim"        "perfect"    "sweet"      "chinese"
##  [6] "bread"      "breakfast"  "happy"      "loved"      "selection"
## [11] "manager"    "spot"       "super"      "taco"       "dish"
## [16] "asked"      "rice"       "waitress"   "enjoyed"    "worst"
## [21] "fried"      "cool"       "horrible"   "cream"      "salsa"
## [26] "spicy"      "hour"       "pittsburgh" "enjoy"      "prices"
## [31] "feel"       "friends"    "primanti"   "rude"       "pork"
## [36] "wings"      "places"     "chips"      "terrible"   "mexican"
## [41] "friend"     "cooked"     "outside"    "different"  "free"
## [46] "told"       "options"    "town"       "delivery"   "huge"
```

For good restaurants, chocolate is the second most probable word after wine. Other top words include things like 'perfect', 'loved' and 'fantastic' all indicative of a good review.

For bad restaurants, the top 50 words feature negative words like 'words', 'horrible' 'rude', 'terrible', indicative of a poor review.

(c) For the restaurants in the test data set, estimate the probability based on the results of the Dirichlet-Multinomial model that each is good versus bad. You may want to apply the function `posterior_pA` provided below (in `midterm-1.Rmd`). Create a visual summary relating the estimated probabilities and the actual binary ratings in the test data. [15 points]

The output of the $posterior_pA$ function is the posterior probability $P(A\,|\,data)$ that the restaurant is 'good'.

One can use the above function to infer the rating for a given test restaurant by predicting rating 1 if $p_i = p(A\,|\,y_i, data) > 0.5$ and rating 0 otherwise.

The dirichlet parameter $alpha$ is set to 2.

```
test_labels <- restaurants_test[, 1]
test_features <- restaurants_test[, 22:121]

n.test = nrow(restaurants_test)
dirichlet.probs = rep(NA, n.test)

for (i in 1:n.test) {
    y_til = as.numeric(as.character(test_features[i, ]))
    dirichlet.probs[i] = posterior_pA(alpha = 2, yA = yA, yB = yB,
        y_til = y_til)
}

# predictions
dirichlet.preds <- ifelse(dirichlet.probs > 0.5, 1, 0)
# confusion matrix
table(test_labels, dirichlet.probs > 0.5)
```

```
##
## test_labels FALSE TRUE
##           0   111   35
##           1    34  148
```

```
print(classError(dirichlet.preds, test_labels)$errorRate)
```

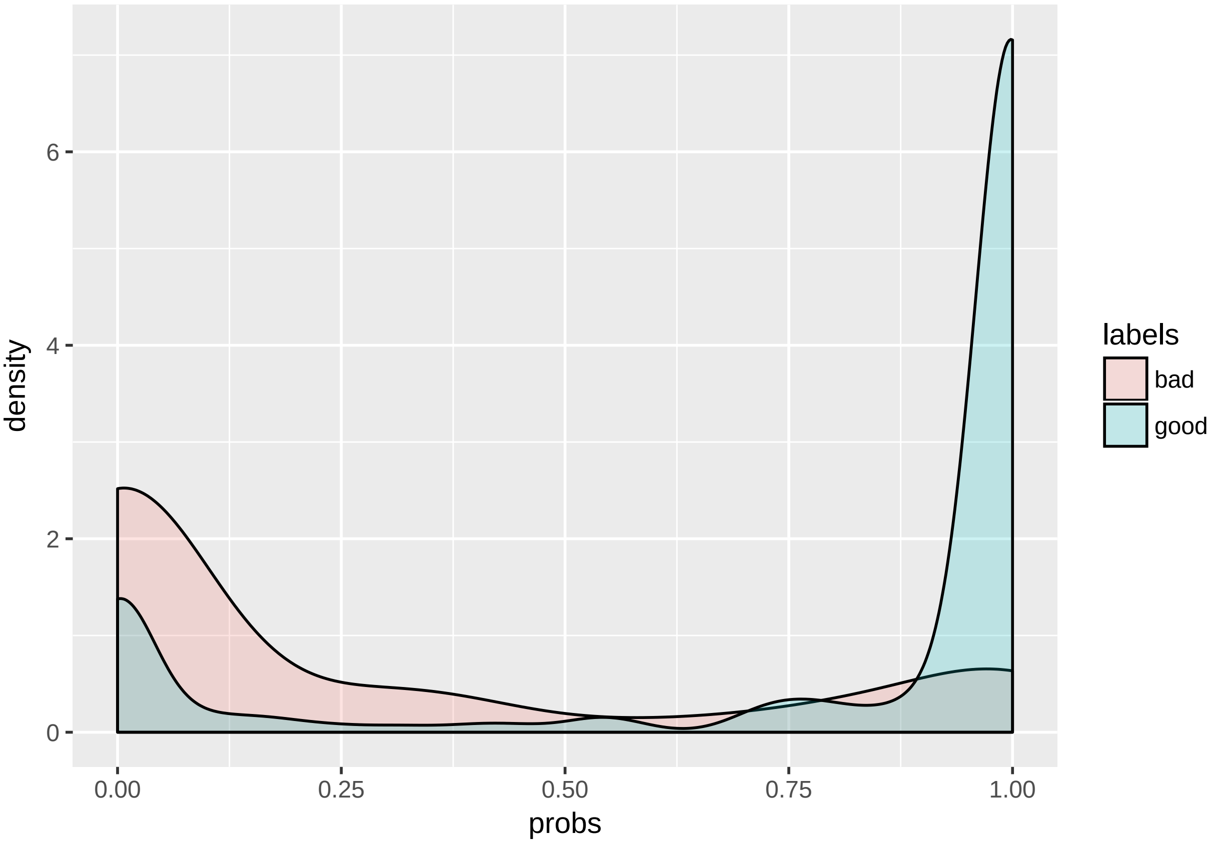
```
## [1] 0.2103659
```

```
dirichlet_preds_labels <- data.frame(probs = dirichlet.probs,
    labels = ifelse(test_labels == 1, "good", "bad"))
```

The confusion matrix shows that a large number of 'bad' restaurants are misclassified as good restaurants. This results in a class error rate of 21% on the test set.

A histogram of the probabilities calcualted from the $posterior_pA$ function can be plotted color coded by their acutal class label.

```
ggplot(dirichlet_preds_labels, aes(probs, fill = labels)) + geom_density(alpha = 0.2)
```



The classification boundary is at probability = 0.5. The histogram shows that while a large number of the good restaurants lie on the right hand side of the 0.5 boundary there are still some good restaurants which lie to the left. A similar observation can be made for bad restaurants and in this case, a large proportion of the predicted posterior probabilities lie on the wrong side of the classification bounary.

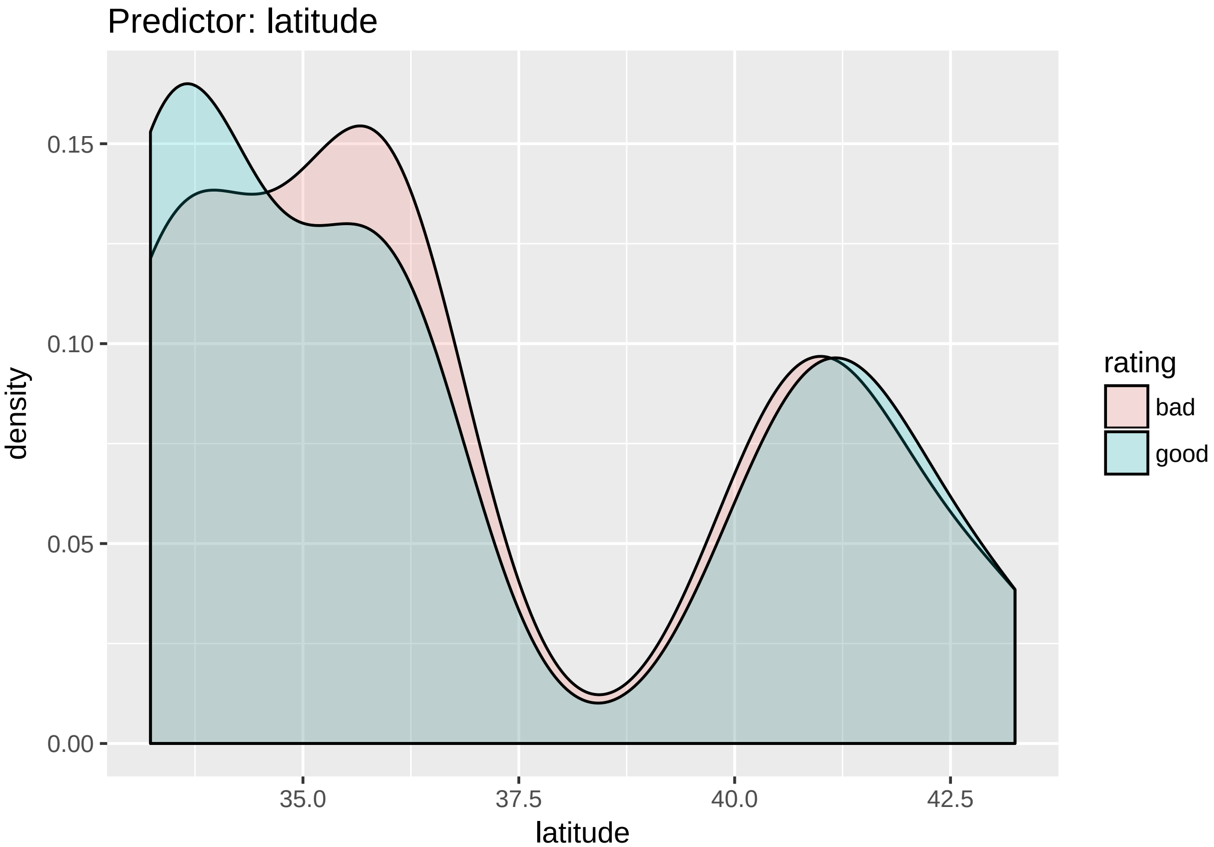
Problem 3 [45 points]

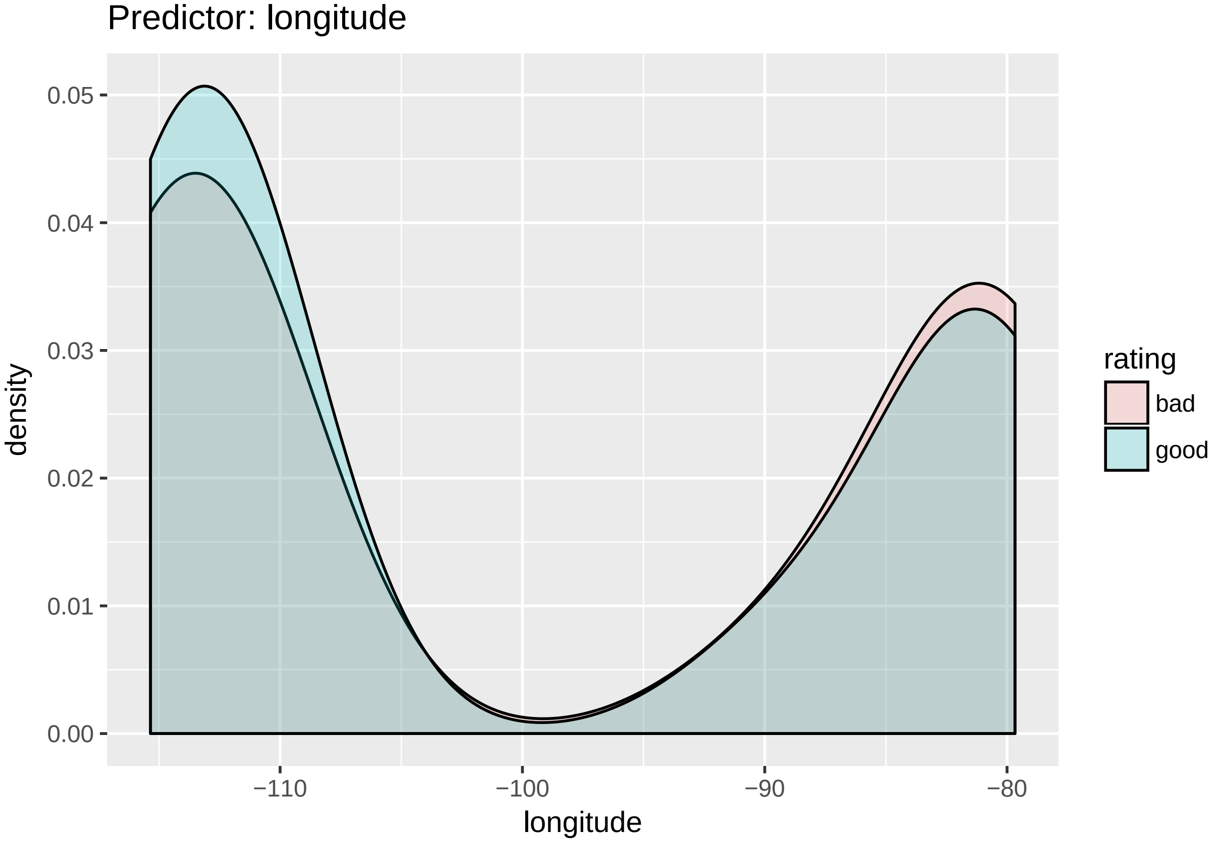
This problem is concerned with modeling a restaurant's rating on factors other than word occurrences.

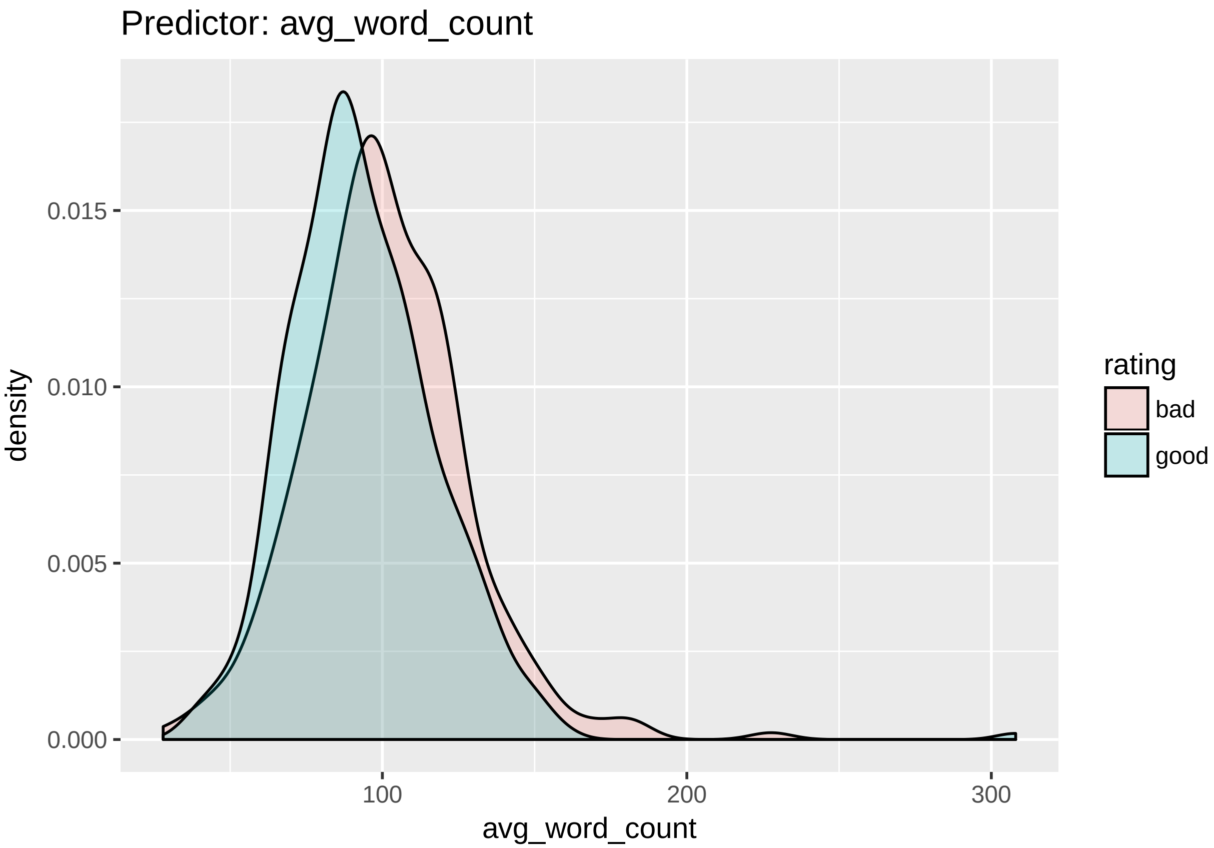
(a) Construct a model for the probability a restaurant is rated "good" as a function of latitude and longitude, average word count, and business attributes. Include quantitative predictor variables as smoothed terms as you see appropriate. You may use default tuning parameter. Summarize the results of the model. Does evidence exist that the smoothed terms are significantly non-linear? Produce visual displays to support your conclusions. [20 points]

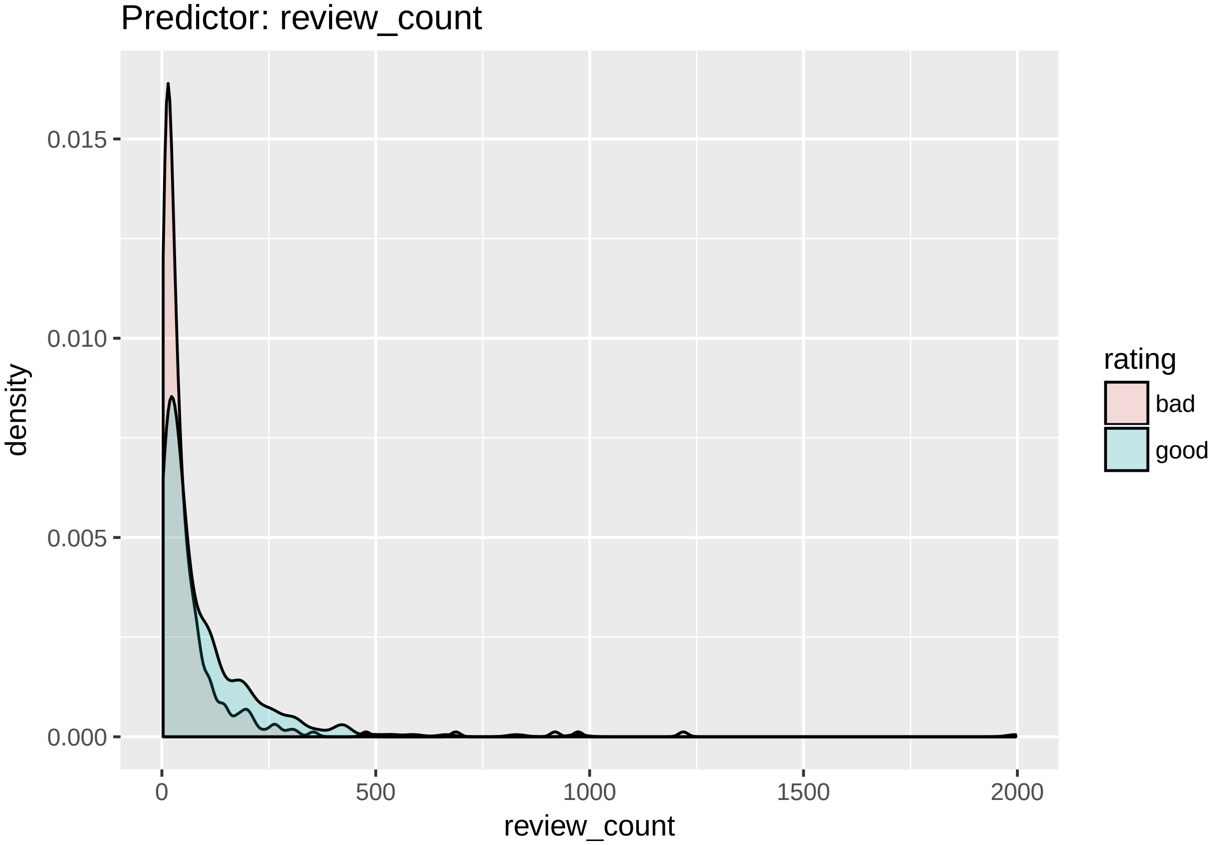
This problem suggests the use of a Generalised Additive model. The quanititate predictors can be modelled using smoothing splines and the qualitative predictors as factors.

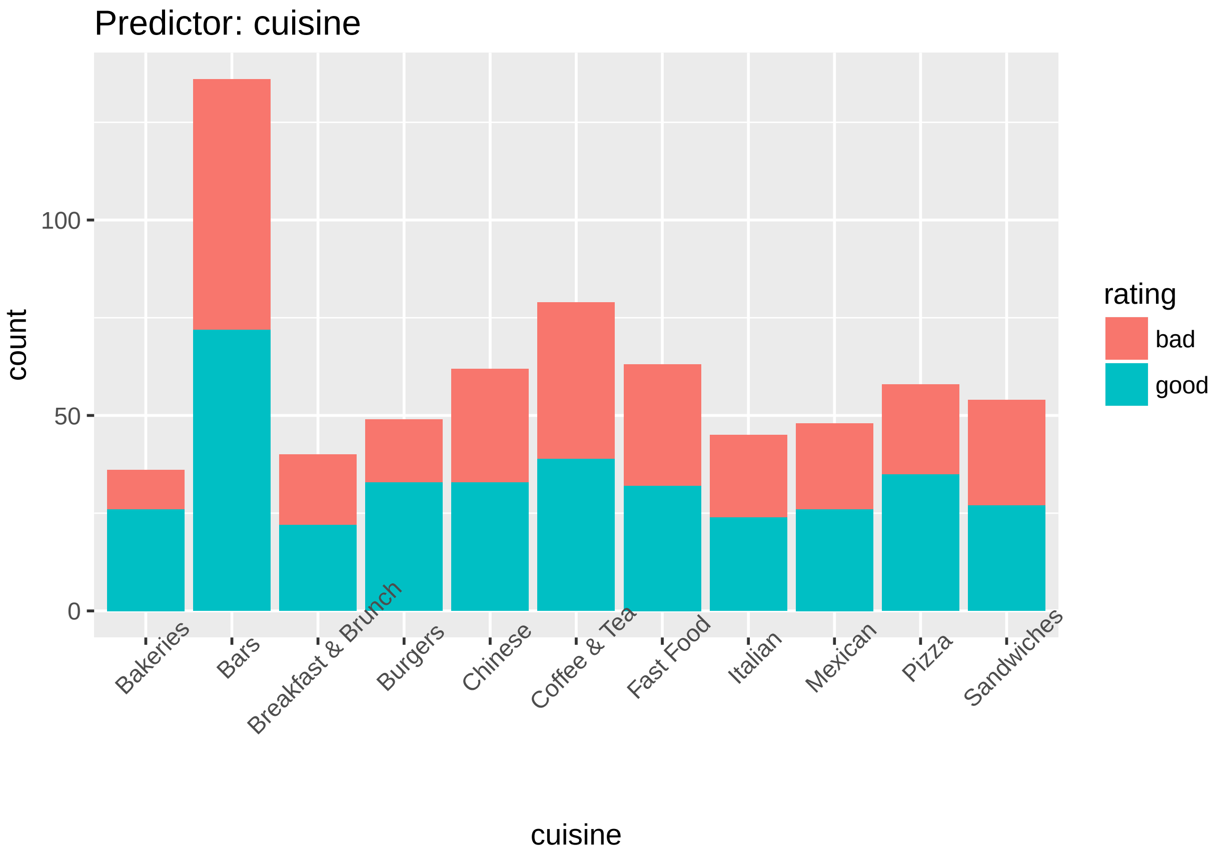
First plot the response as a function of each of the predictors to view individual relationships between predictors and the response.

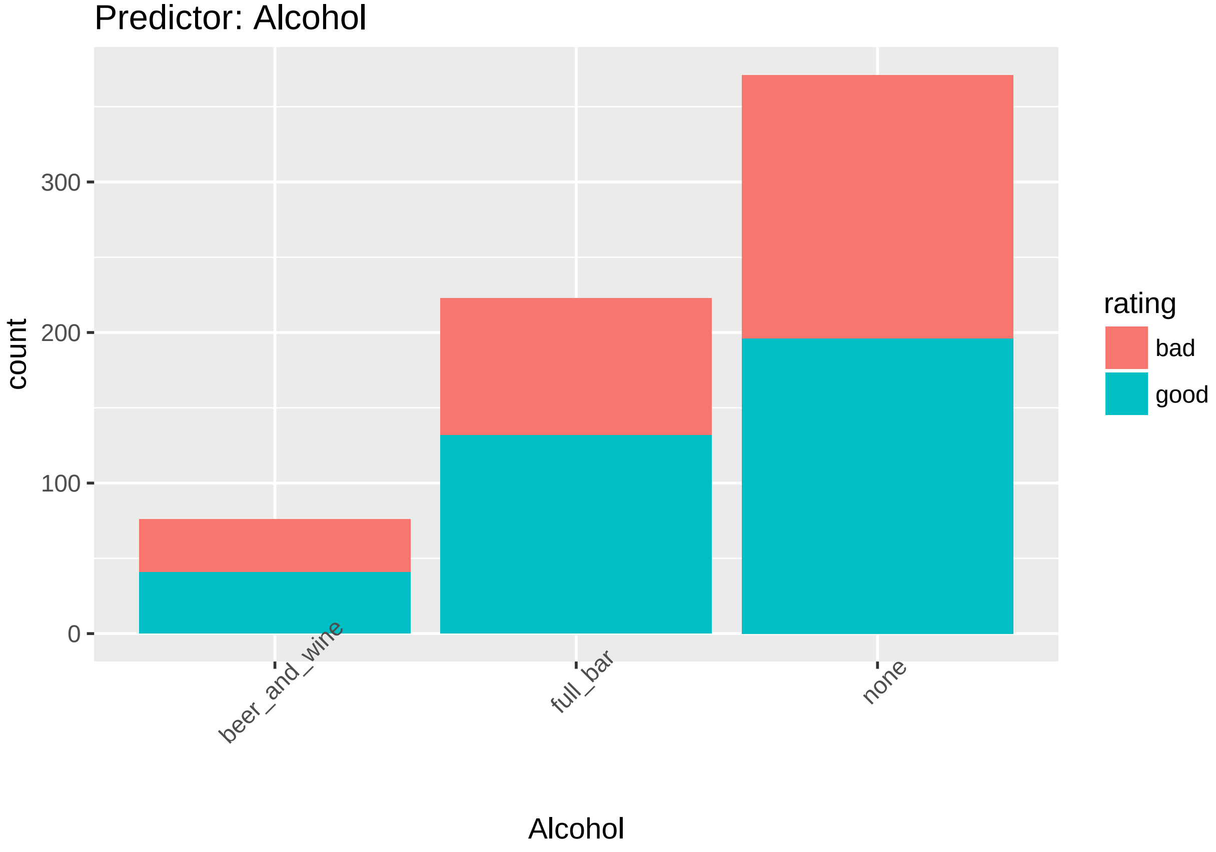

Predictor: latitude
density
0.15
0.10
0.05
0.00
35.0
37.5
40.0
42.5
latitude
rating
bad
good

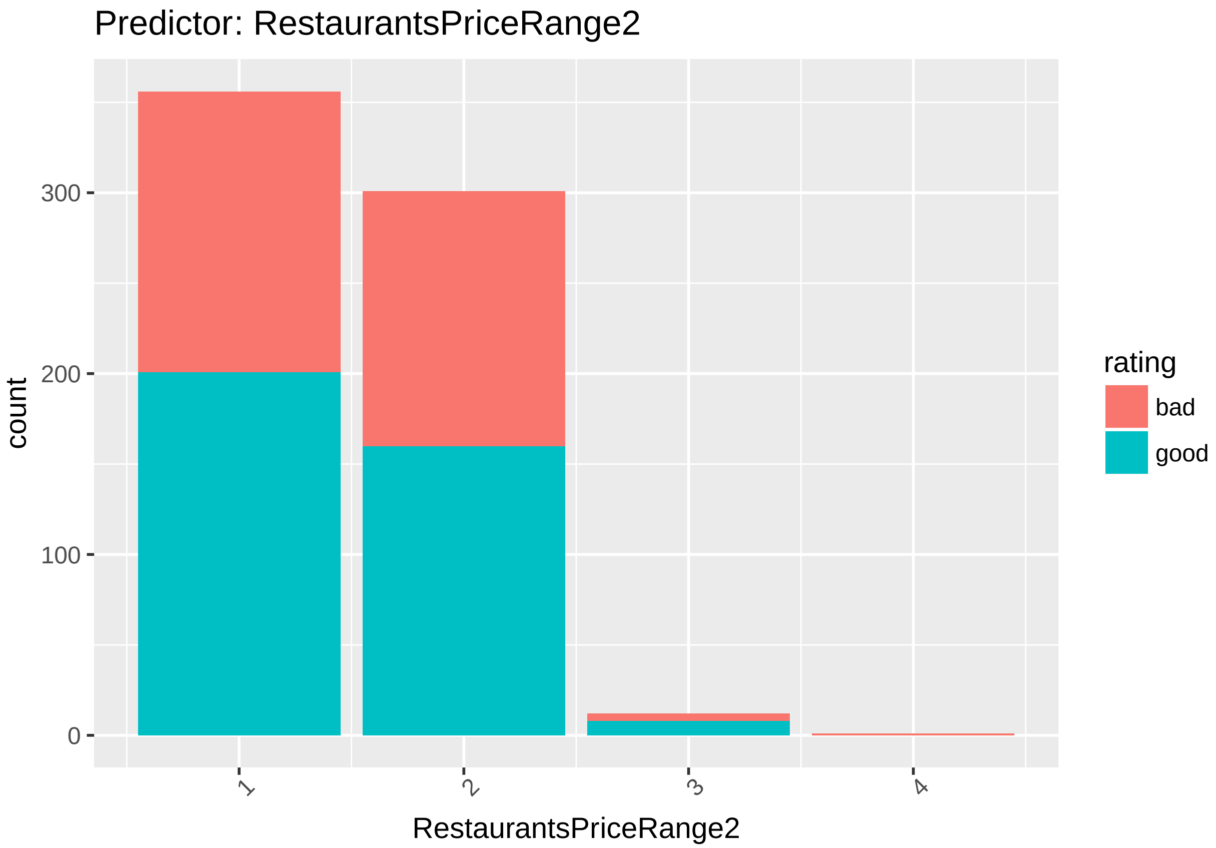

Predictor: longitude
density
0.05
0.04
0.03
0.02
0.01
0.00
−110
−100
−90
−80
longitude
rating
bad
good


Predictor: avg_word_count
density
0.015
0.010
0.005
0.000
100
200
300
avg_word_count
rating
bad
good

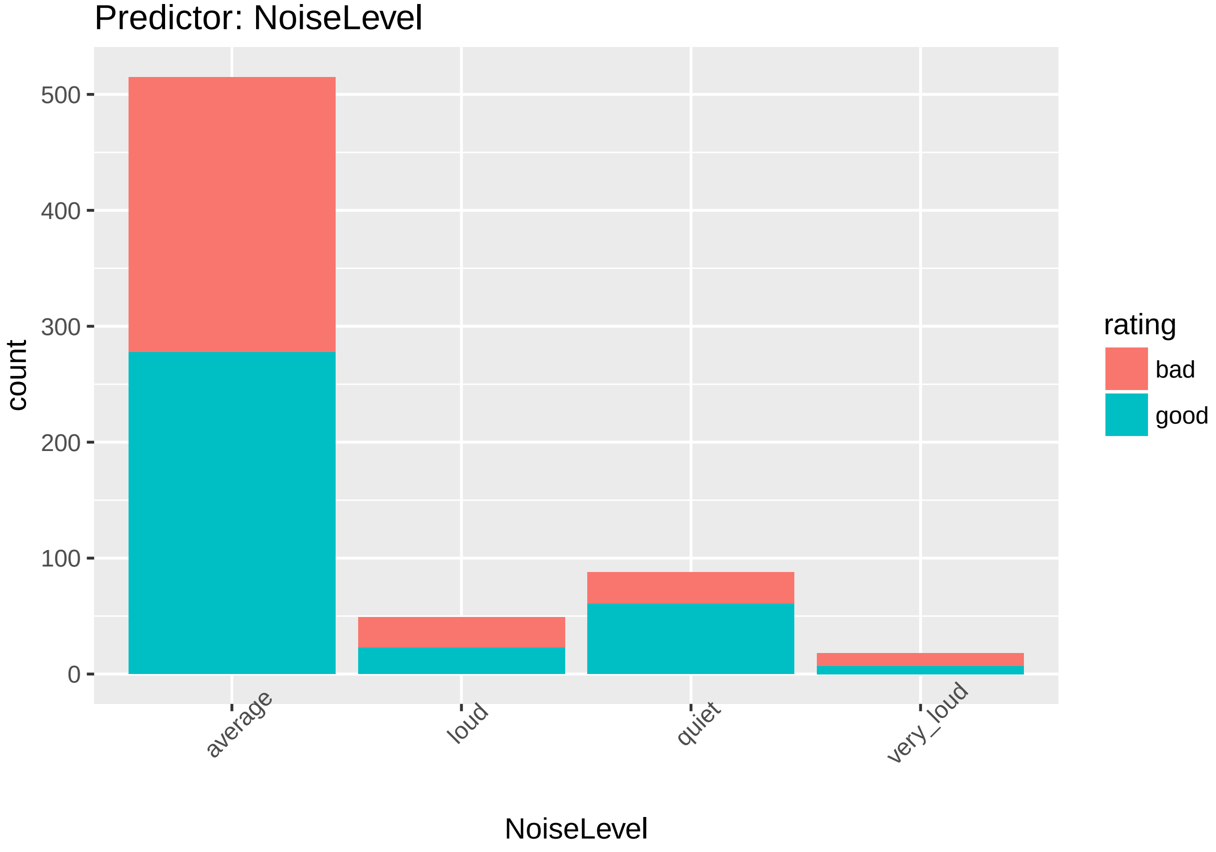

Predictor: review_count
density
0.015
0.010
0.005
0.000
0
500
1000
1500
2000
review_count
rating
bad
good


Predictor: cuisine
count
0
50
100
Bakeries
Bars
Breakfast & Brunch
Burgers
Chinese
Coffee & Tea
Fast Food
Italian
Mexican
Pizza
Sandwiches
cuisine
rating
bad
good

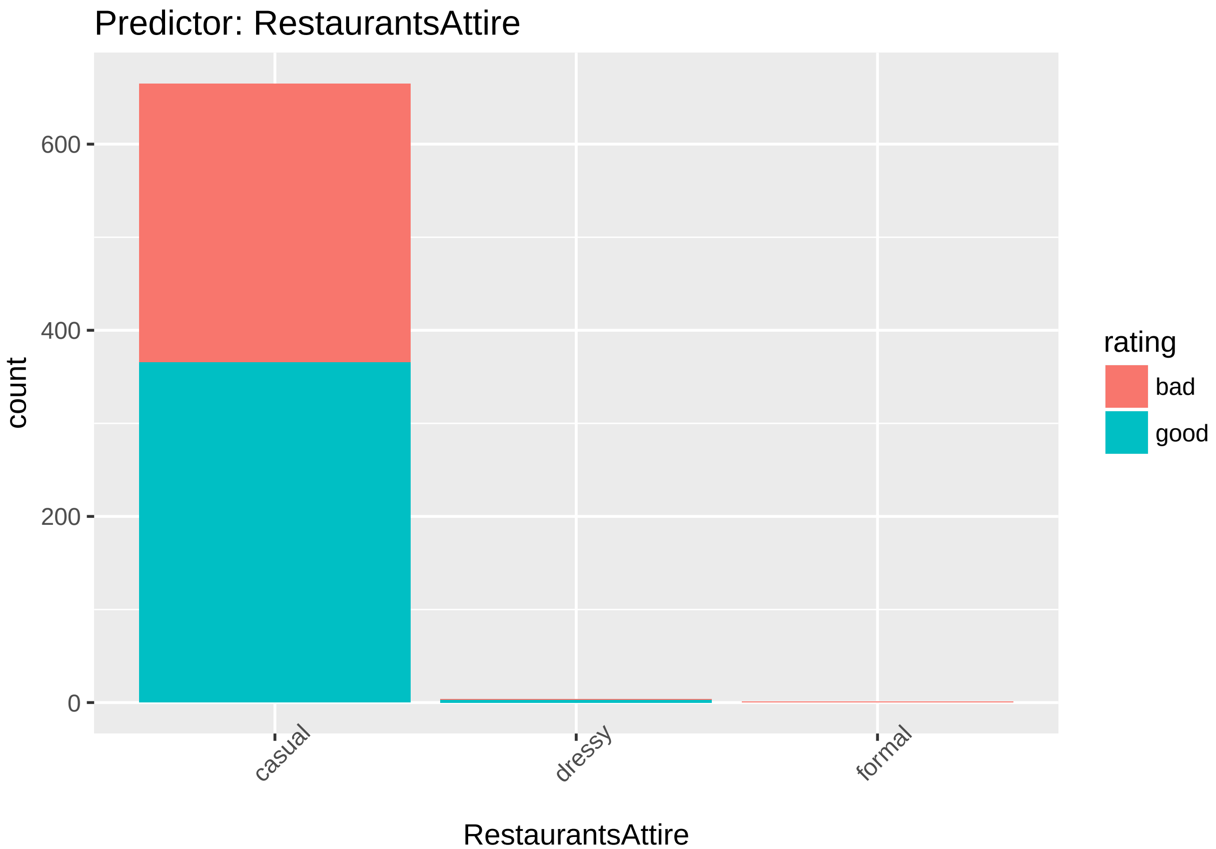

Predictor: Alcohol
count
0
100
200
300
beer_and_wine
full_bar
none
Alcohol
rating
bad
good

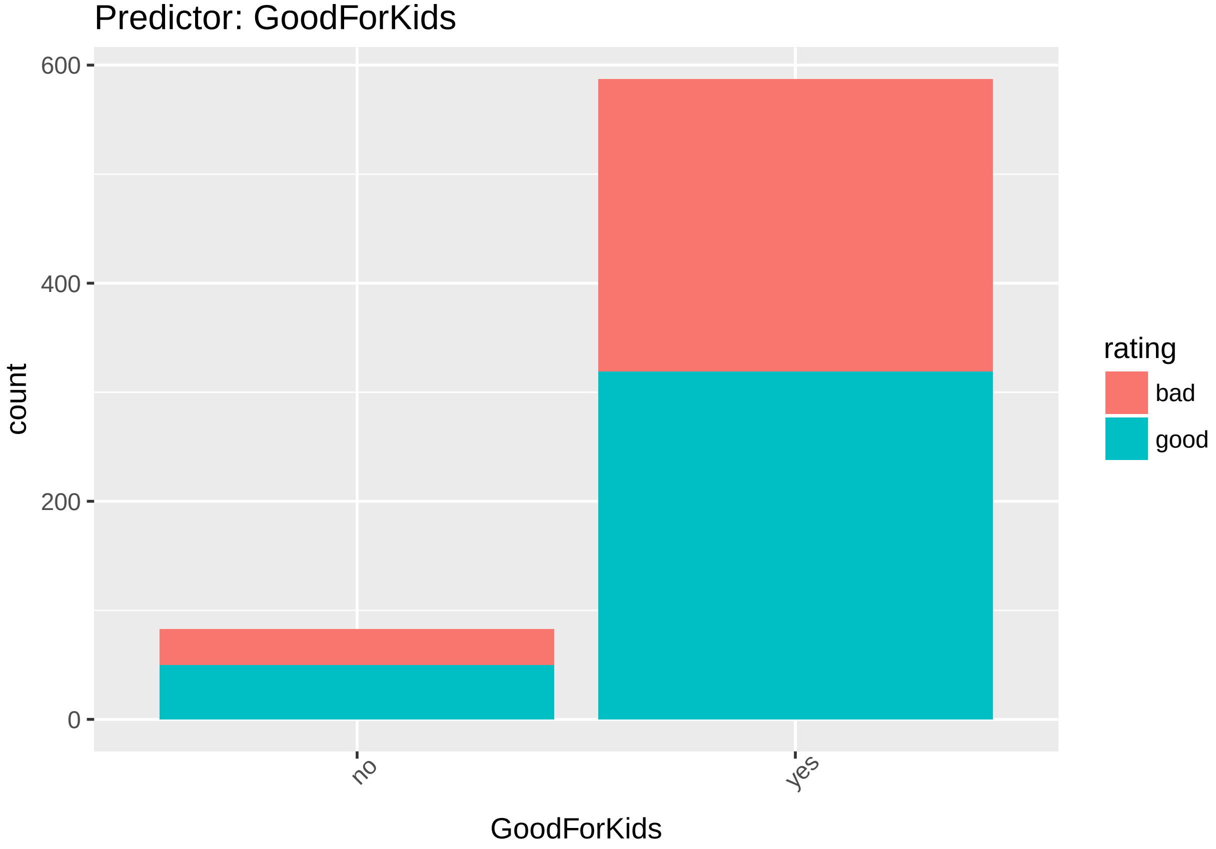


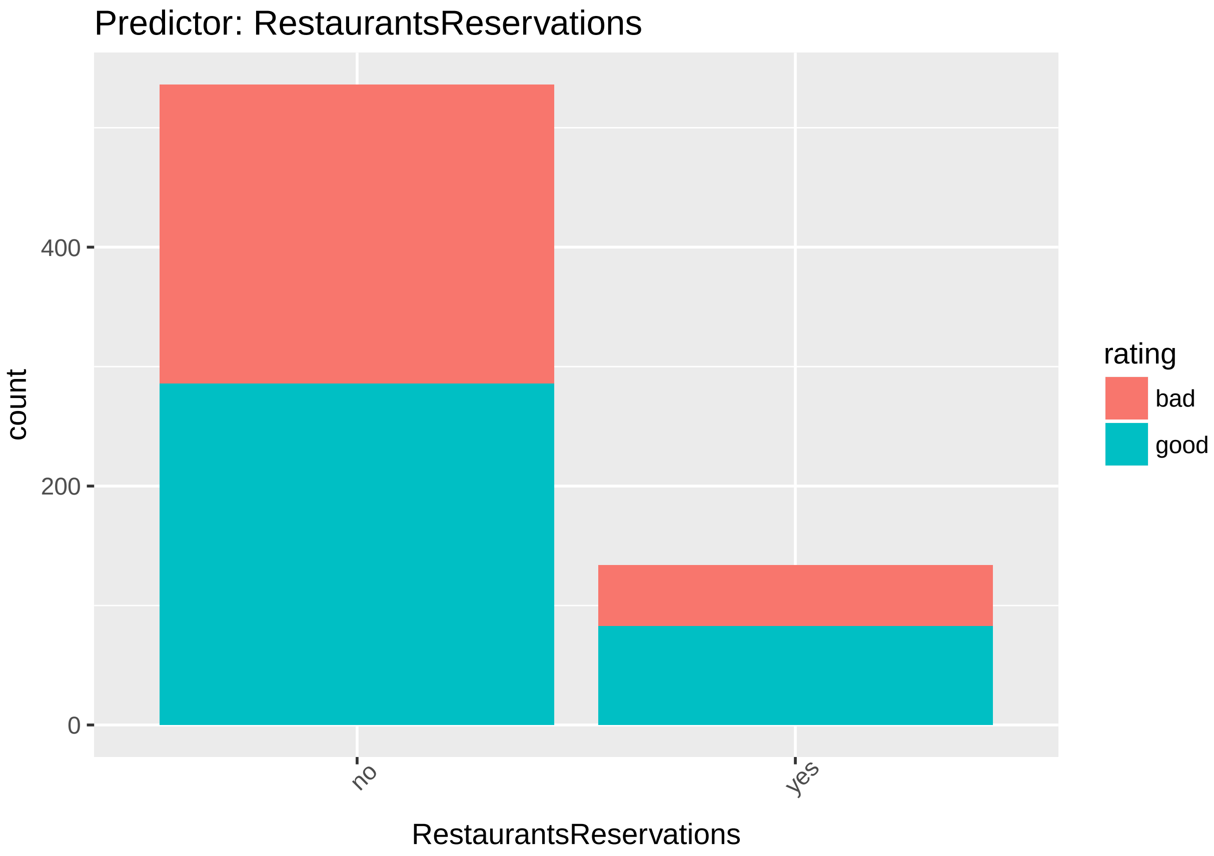
Predictor: RestaurantsPriceRange2



Predictor: NoiseLevel


Predictor: RestaurantsAttire
count
600
400
200
0
casual
dressy
formal
RestaurantsAttire
rating
bad
good


Predictor: GoodForKids
count
600
400
200
0
no
yes
GoodForKids
rating
bad
good



From visualisation of the distribution of each individual predictor with the response, several preliminary observations can be made. * (latitude/longitude) There appears to be a multi-modal distribution with the restaurants being clustered in a few regions. Overall, there are approximately equal numbers of good/bad reviews in each region but in two specific latitidue regions, either good reviews dominate or bad reviews dominate. Restaurants in the latitude range 34.5-37 have higher number of bad ratings compared to restaurants in the latitude range 33-34.5 which have a larger number of good ratings. * (avg_word_count) bad reviews tend to have slighty higher number of words compared to good reviews * (review_count) good restaurants tend to have a larger number of overall reviews compared to bad restaurants * (cuisine) The category of 'Bars' has the largest number of total reviews, however approx 50% of these are bad. Bakeries appear to have the smallest proportion of bad reviews compared to bars, fastfood and coffee/tea which have approx 50% good and 50% bad. * (restaurant price range) This predictor appears to be uninformative - there are approx 50% good and bad reviews for each price range (apart from category 3 which only makes up a very small proportion of the total number of reviews). * (noise level) This predictor also appears to be uninformative, as there are approximately 50% good and 50% bad reviews per category. * (attire, reservations) both uninformative predictors

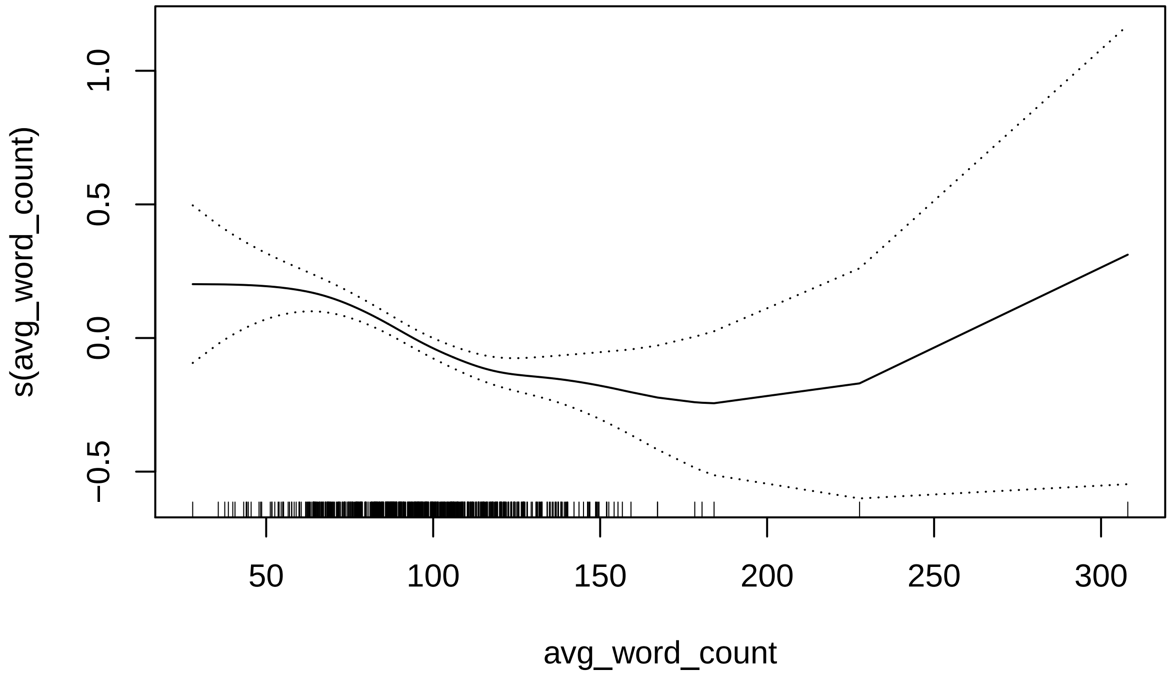
From these observations, it appears that the following predictors have some predictive relationship with the response: latitude, longitude, avg_word_count, review_count and potentially cuisine. Post code is likely to be correlated with latitude/longitude so it will not be included.

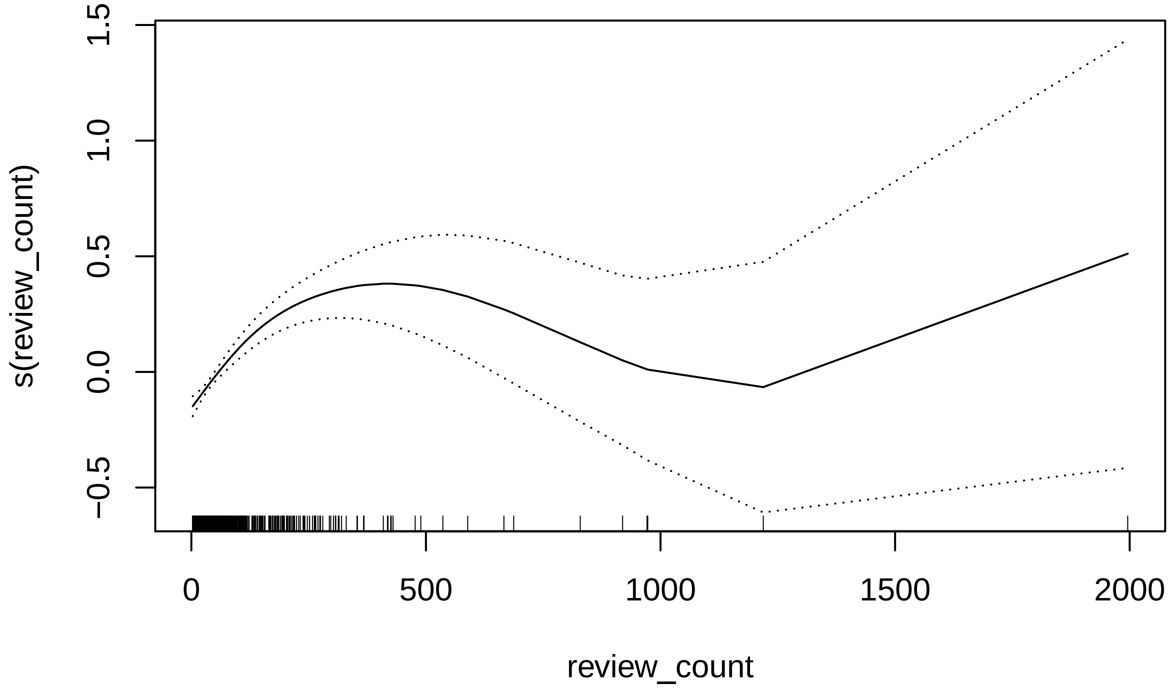
```
##
## Call: gam(formula = rating ~ s(avg_word_count) + s(review_count) +
##     s(latitude) + s(longitude) + cuisine, data = restaurants_train)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8678 -0.4571  0.1553  0.4150  0.8828
##
## (Dispersion Parameter for gaussian family taken to be 0.2232)
```

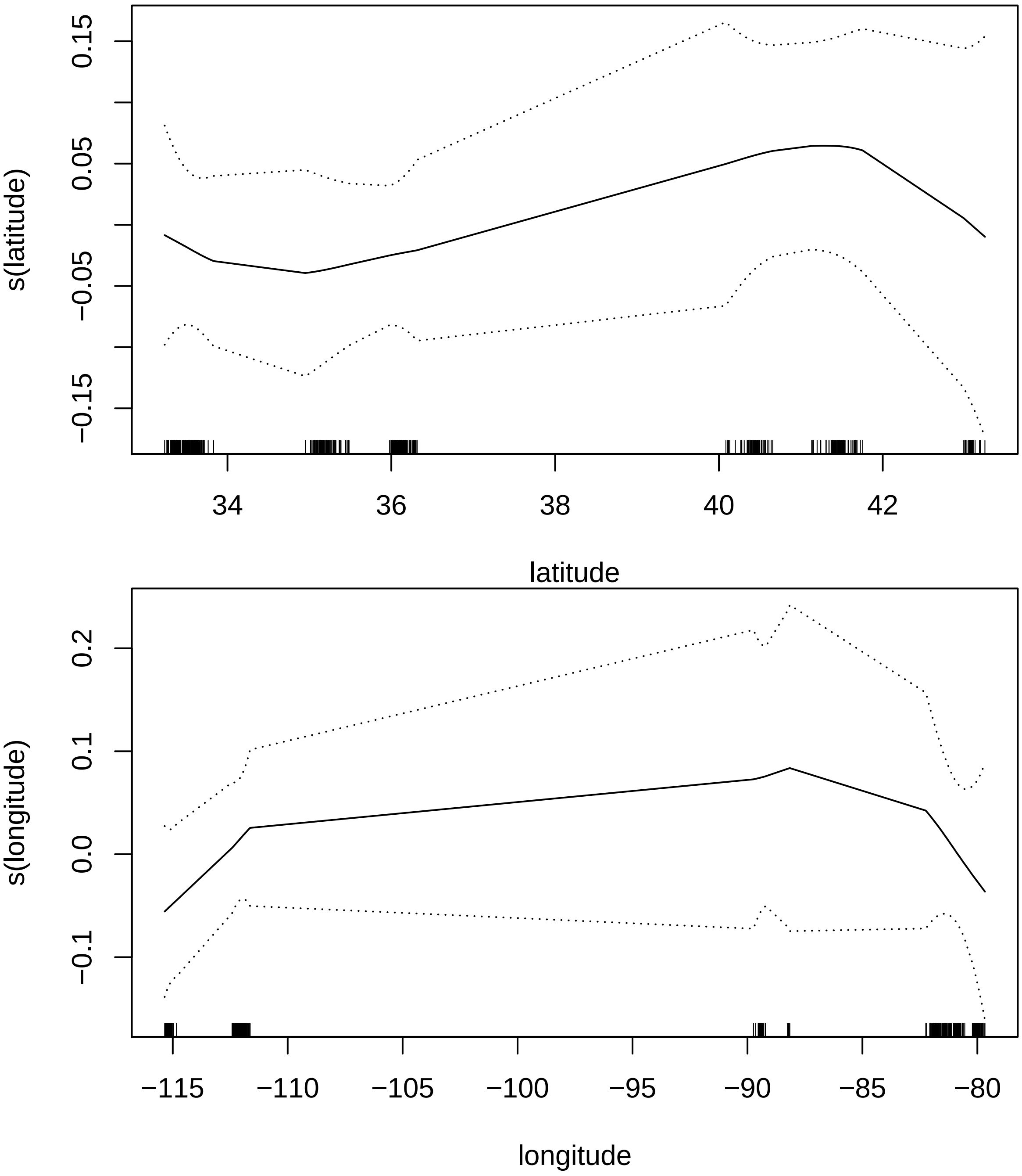
```
##
##     Null Deviance: 165.7746 on 669 degrees of freedom
## Residual Deviance: 143.5271 on 642.9994 degrees of freedom
## AIC: 925.0739
##
## Number of Local Scoring Iterations: 6
##
## Anova for Parametric Effects
##                    Df  Sum Sq Mean Sq F value    Pr(>F)
## s(avg_word_count)   1   4.031  4.0307 18.0574 2.461e-05 ***
## s(review_count)     1   5.481  5.4813 24.5561 9.241e-07 ***
## s(latitude)         1   0.727  0.7270  3.2570   0.07159 .
## s(longitude)        1   0.020  0.0202  0.0903   0.76390
## cuisine            10   2.058  0.2058  0.9221   0.51207
## Residuals         643 143.527  0.2232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df  Npar F     Pr(F)
## (Intercept)
## s(avg_word_count)       3  4.5643  0.003569 **
## s(review_count)         3 13.3681 1.792e-08 ***
## s(latitude)             3  0.6957  0.554900
## s(longitude)            3  1.2295  0.298042
## cuisine
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

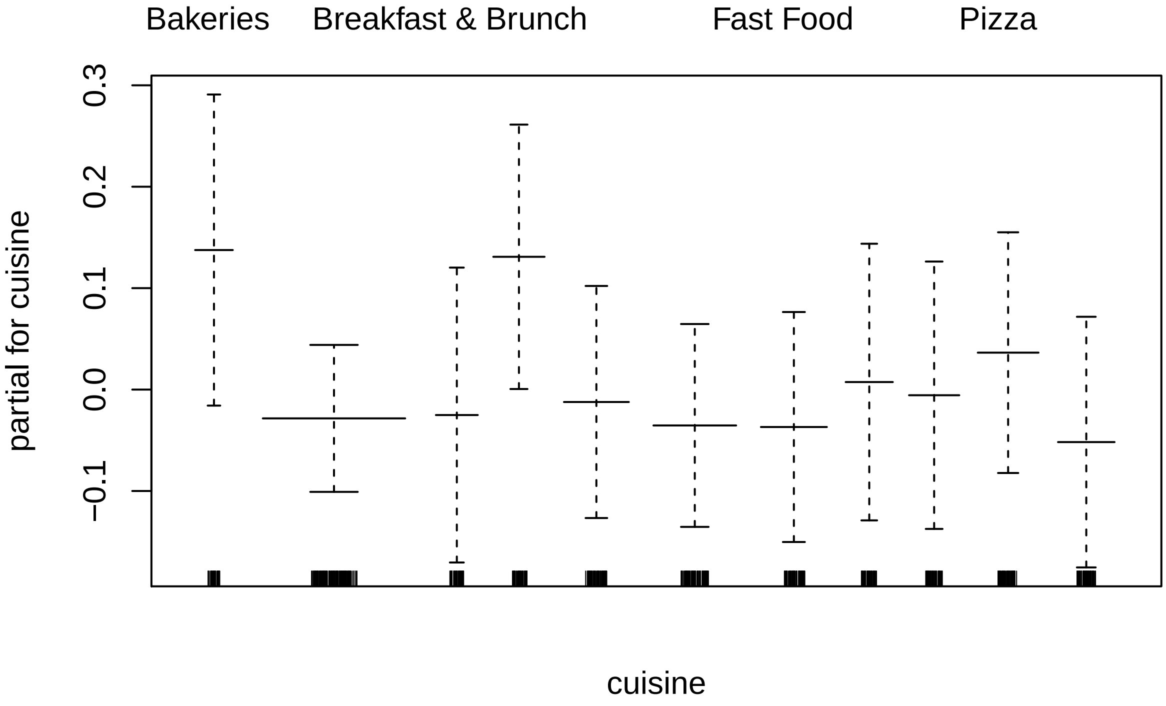
In the summary of the model, the chi squared test result of the smoothed predictors indicates whether the model term is significant. The smoothed terms for avg_word_count and review_count appear to be statistically significant. However the remaining predictors do not appear to be significant.

The smooth of the predictors can also be plotted to determine if the smoothed terms are significantly non linear.


s(avg_word_count)
1.0
0.5
0.0
−0.5
50
100
150
200
250
300
avg_word_count


s(review_count)
1.5
1.0
0.5
0.0
−0.5
0
500
1000
1500
2000
review_count


s(latitude)
0.15
0.05
−0.05
−0.15
34
36
38
40
42
latitude
s(longitude)
0.2
0.1
0.0
−0.1
−115
−110
−105
−100
−95
−90
−85
−80
longitude



The avg_word_count predictor appears to have a non linear relationship with the response (although for word counts > 150 there are very few data points and hence the the confidence intervals are large). The smoothed review_count predictor appears to be fairly linear for small numbers of review counts (approx in the range 0-200) but exhibits non linear effects for large values of review_count (and also has large confidence intervals in regions of the parameter space with few data points).

The latitude and longitude predictors show significant non linearity in their smoothing due to the values being concentrated in a few specific regions. There does not appear to be any clear trend in the smoothed values of these predictors.

This may be due to the fact that inidivudally, the latitude and longitude predictors provide limited information about a restaurant's rating but the combination of both might provide a better prediction.

(b) For your model in part (a), summarize the predictive ability by computing a misclassification rate. [10 points]

The predictions of the model on the test set can be calculated from which a misclassification rate can also be determined:

```r
gam.test.predictions <- predict(restaurants.gam, newdata = restaurants_test,
    type = "response")
gam.test.predictions <- ifelse(gam.test.predictions > 0.5, 1,
    0)
print(classError(gam.test.predictions, restaurants_test$rating)$errorRate)
```

```
## [1] 0.4054878
```

The misclassification rate is 40%, much higher than for the predictive model based on the word occurences in the reviews. This could be because the word occurences directly refer to customer opinions/satisfaction/experience, whereas other business attributes and geolocation may be more indirectly related.

(c) Consider a version of model (a) that does not include the `cuisine` predictor variable. Explain briefly how you would test in your model whether `cuisine` is an important predictor of the probability of a good restaurant rating. Perform the test, and explain your conclusions. [15 points]

A likelihood ratio test can be used to compare models built with and without the cuisine predictor. The test looks at two models, one of which is a subset of the other, and compares how many times more likely the data are under the more complex model than the simpler model.

In addition, the classification accuracy (or misclassification rate) can also be computed for the two models to determine if there is any improvment with the addition of the cuisine predictor.

First build the simpler model without the cuisine predictor:

```
restaurants.gam_nocuisine <- gam(rating ~ s(avg_word_count) +
    s(review_count) + s(latitude) + s(longitude), data = restaurants_train)
summary(restaurants.gam_nocuisine)
```

```
##
## Call: gam(formula = rating ~ s(avg_word_count) + s(review_count) +
##     s(latitude) + s(longitude), data = restaurants_train)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9193 -0.4651  0.1730  0.4168  0.8799
##
## (Dispersion Parameter for gaussian family taken to be 0.2229)
##
##     Null Deviance: 165.7746 on 669 degrees of freedom
## Residual Deviance: 145.5657 on 652.9994 degrees of freedom
## AIC: 914.5232
##
## Number of Local Scoring Iterations: 6
##
## Anova for Parametric Effects
##                    Df  Sum Sq Mean Sq F value    Pr(>F)
## s(avg_word_count)   1   4.107  4.1071 18.4242 2.036e-05 ***
## s(review_count)     1   5.527  5.5271 24.7941 8.174e-07 ***
## s(latitude)         1   0.758  0.7581  3.4008   0.06562 .
## s(longitude)        1   0.022  0.0217  0.0972   0.75532
## Residuals         653 145.566  0.2229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df  Npar F     Pr(F)
## (Intercept)
## s(avg_word_count)       3  4.6300  0.003258 **
## s(review_count)         3 13.2973 1.959e-08 ***
## s(latitude)             3  0.5988  0.615970
## s(longitude)            3  1.1091  0.344660
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(restaurants.gam_nocuisine, restaurants.gam, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: rating ~ s(avg_word_count) + s(review_count) + s(latitude) +
##     s(longitude)
## Model 2: rating ~ s(avg_word_count) + s(review_count) + s(latitude) +
##     s(longitude) + cuisine
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       653     145.57
## 2       643     143.53 10   2.0386   0.5196
```

```
gam.test.predictions <- predict(restaurants.gam_nocuisine, newdata = restaurants_test,
    type = "response")
gam.test.predictions <- ifelse(gam.test.predictions > 0.5, 1,
    0)
print(classError(gam.test.predictions, restaurants_test$rating)$errorRate)
```

```
## [1] 0.4085366
```

The result of the likelihood ratio test shows that the model with the inclusion of the cuisine parameter is not statistically significant compared to the model without the cuisine predictor. Comparing the classification accuracies of the two models, the error rates only differ by 0.3%. Therefore it can be concluded that the cuisine predictor is not an important predictor of the probability of a good restaurant rating.