Classification

Justin Hardy & Benji Frenkel

Description

Linear Models, in the context of Classification, aim to separate observations into two separate regions so that outputs can be classified in a binary manner. For us to begin this assignment, it's important to understand the strengths and weaknesses of Linear Models for Classification.

There are a number of Generalized Linear Models (GLMs) which can help us model our data using classification, which we will explore in this assignment. Particularly, we'll be exploring a data set through the use of the Logistic Regression and Naïve Bayes Models.

Like Linear Regression, Logistic Regression focuses on predicting for a single target variable, but differs in that it must target a qualitative value. It's very inexpensive and keeps the classes linearly separable, but lacks the flexibility required for capturing non-linear decision boundaries.

Naïve Bayes, on the other hand, has the same goals as Linear Regression, but functions far differently. It will make the naïve assumption that every predictor is independent of one another, allowing for easy implementation and interpretability, at the cost of performance… generally.

Both linear models will be used on a data set we selected off of the internet. The data set consists of data related to a bank's campaigns to get clients to subscribe a term deposit.

Modeling

Data Set Setup

Starting out, we'll load our data set into R.

```r
# data set input
BankMarketing <- read.csv("bank-additional-full.csv")
```

We'll then create a factor for various qualitative values.

```r
# data set cleanup
BankMarketing$y <- factor(BankMarketing$y)
BankMarketing$poutcome <- factor(BankMarketing$poutcome)
BankMarketing$contact <- factor(BankMarketing$contact)
BankMarketing$housing <- factor(BankMarketing$housing)
BankMarketing$loan <- factor(BankMarketing$loan)
BankMarketing$default <- factor(BankMarketing$default)
BankMarketing$marital <- factor(BankMarketing$marital)
BankMarketing$education <- factor(BankMarketing$education)
BankMarketing$day_of_week <- factor(BankMarketing$day_of_week)
BankMarketing$month <- factor(BankMarketing$month)
BankMarketing$previously_contacted <- as.factor(ifelse(BankMarketing$pdays==999, "no", "yes"))
```

Diving Into Train / Test

Diving the data into train/test…

```
# train/test division
i <- sample(1:nrow(BankMarketing), nrow(BankMarketing)*0.8, replace=FALSE)
train <- BankMarketing[i,]
test <- BankMarketing[-i,]
```

Data Exploration / Graphing

We'll be exploring the data within our train data set, which makes up 80% of the shopper intentions data set. The following are various details/statistics about the data set itself:

Rows / Columns Info:

```
## 'data.frame':    32950 obs. of  22 variables:
##  $ age                 : int  32 31 43 26 37 35 33 56 33 43 ...
##  $ job                 : chr  "technician" "admin." "services" "admin." ...
##  $ marital             : Factor w/ 4 levels "divorced","married",..: 2 2 2 3 2 2 2 2 2 2 ...
##  $ education           : Factor w/ 8 levels "basic.4y","basic.6y",..: 7 7 4 7 4 2 7 4 6 8 ...
##  $ default             : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ housing             : Factor w/ 3 levels "no","unknown",..: 1 3 3 1 3 3 3 1 1 1 ...
##  $ loan                : Factor w/ 3 levels "no","unknown",..: 1 3 1 1 1 1 1 1 1 1 ...
##  $ contact             : Factor w/ 2 levels "cellular","telephone": 1 1 1 2 2 2 1 1 1 2 ...
##  $ month               : Factor w/ 10 levels "apr","aug","dec",..: 2 2 7 4 7 5 7 2 2 7 ...
##  $ day_of_week         : Factor w/ 5 levels "fri","mon","thu",..: 3 3 3 2 2 3 2 4 3 1 ...
##  $ duration            : int  12 12 197 127 149 64 740 173 80 512 ...
##  $ campaign            : int  8 5 1 7 1 6 1 1 3 6 ...
##  $ pdays               : int  999 999 999 999 999 999 3 999 999 999 ...
##  $ previous            : int  0 0 0 0 0 0 2 0 0 0 ...
##  $ poutcome            : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 3 2 2 2 ...
##  $ emp.var.rate        : num  1.4 1.4 -1.8 1.4 1.1 1.4 -1.8 1.4 1.4 1.1 ...
##  $ cons.price.idx      : num  93.4 93.4 92.9 93.9 94 ...
##  $ cons.conf.idx       : num  -36.1 -36.1 -46.2 -42.7 -36.4 -41.8 -40 -36.1 -36.1 -36.4 ...
##  $ euribor3m           : num  4.96 4.96 1.33 4.96 4.86 ...
##  $ nr.employed         : num  5228 5228 5099 5228 5191 ...
##  $ y                   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 2 1 1 1 ...
##  $ previously_contacted: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 2 1 1 1 ...
```

Sample of First Five Rows:

```
##       age        job marital         education default housing loan   contact
## 23643  32 technician married university.degree      no      no   no  cellular
## 23691  31     admin. married university.degree      no     yes  yes  cellular
## 31586  43   services married       high.school      no     yes   no  cellular
## 17664  26     admin.  single university.degree      no      no   no telephone
## 5593   37   services married       high.school      no     yes   no telephone
##       month day_of_week duration campaign pdays previous    poutcome
## 23643   aug         thu       12        8   999        0 nonexistent
## 23691   aug         thu       12        5   999        0 nonexistent
```

```
## 31586   may         thu      197        1   999        0 nonexistent
## 17664   jul         mon      127        7   999        0 nonexistent
## 5593    may         mon      149        1   999        0 nonexistent
##       emp.var.rate cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 23643          1.4         93.444         -36.1     4.962      5228.1 no
## 23691          1.4         93.444         -36.1     4.962      5228.1 no
## 31586         -1.8         92.893         -46.2     1.327      5099.1 no
## 17664          1.4         93.918         -42.7     4.962      5228.1 no
## 5593           1.1         93.994         -36.4     4.857      5191.0 no
##       previously_contacted
## 23643                   no
## 23691                   no
## 31586                   no
## 17664                   no
## 5593                    no
```

Sample of Last Five Rows:

```
##       age          job  marital           education default housing loan
## 30920  36 entrepreneur  married         high.school      no     yes   no
## 9773   35   technician   single         high.school      no      no   no
## 22388  30   technician divorced   university.degree      no     yes   no
## 2676   46      unknown  married   university.degree      no      no   no
## 1915   50     services  married professional.course unknown     yes   no
##         contact month day_of_week duration campaign pdays previous    poutcome
## 30920  cellular   may         tue      283        1   999        1     failure
## 9773  telephone   jun         mon       94        4   999        0 nonexistent
## 22388  cellular   aug         fri      152        3   999        0 nonexistent
## 2676  telephone   may         wed       93        4   999        0 nonexistent
## 1915  telephone   may         fri      211        2   999        0 nonexistent
##       emp.var.rate cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 30920         -1.8         92.893         -46.2     1.344      5099.1 no
## 9773           1.4         94.465         -41.8     4.961      5228.1 no
## 22388          1.4         93.444         -36.1     4.964      5228.1 no
## 2676           1.1         93.994         -36.4     4.859      5191.0 no
## 1915           1.1         93.994         -36.4     4.855      5191.0 no
##       previously_contacted
## 30920                   no
## 9773                    no
## 22388                   no
## 2676                    no
## 1915                    no
```

NA Count:

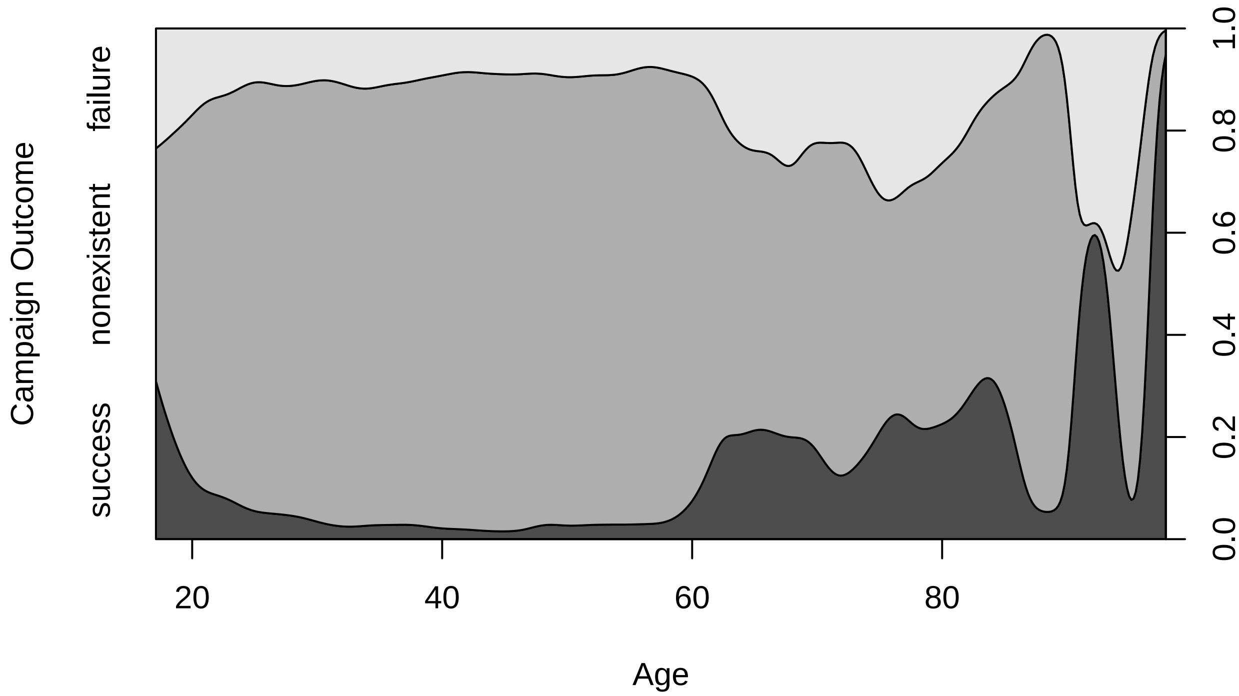
```
## [1] "Number of NAs: 0"
```

General Summary:

```
##       age            job                 marital                    education
##  Min.   :17.00   Length:32950       divorced: 3683   university.degree   :9739
##  1st Qu.:32.00   Class :character   married :19959   high.school         :7606
```

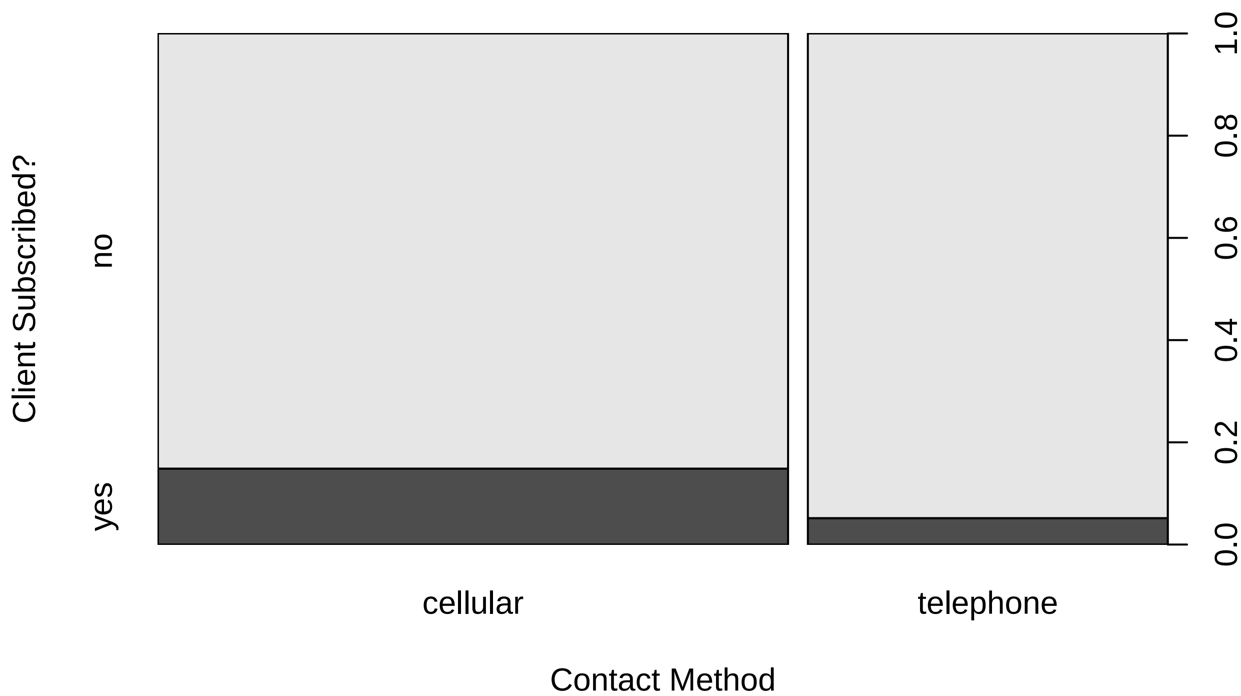
```
##  Median :38.00   Mode  :character   single  : 9245   basic.9y           :4821
##  Mean   :40.04                      unknown :   63   professional.course:4184
##  3rd Qu.:47.00                                       basic.4y           :3389
##  Max.   :98.00                                       basic.6y           :1826
##                                                      (Other)            :1385
##     default         housing           loan           contact
##  no     :26083   no     :14890   no     :27121   cellular :20970
##  unknown: 6864   unknown:  803   unknown:  803   telephone:11980
##  yes    :    3   yes    :17257   yes    : 5026
##
##
##
##
##      month       day_of_week    duration        campaign          pdays
##  may    :11015   fri:6265    Min.   :   0.0   Min.   : 1.000   Min.   :  0.0
##  jul    : 5715   mon:6759    1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.:999.0
##  aug    : 4998   thu:6920    Median : 180.0   Median : 2.000   Median :999.0
##  jun    : 4220   tue:6527    Mean   : 258.4   Mean   : 2.564   Mean   :961.8
##  nov    : 3267   wed:6479    3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0
##  apr    : 2091               Max.   :4918.0   Max.   :56.000   Max.   :999.0
##  (Other): 1644
##     previous            poutcome      emp.var.rate      cons.price.idx
##  Min.   :0.0000   failure    : 3396   Min.   :-3.40000   Min.   :92.20
##  1st Qu.:0.0000   nonexistent:28429   1st Qu.:-1.80000   1st Qu.:93.08
##  Median :0.0000   success    : 1125   Median : 1.10000   Median :93.44
##  Mean   :0.1729                       Mean   : 0.08011   Mean   :93.57
##  3rd Qu.:0.0000                       3rd Qu.: 1.40000   3rd Qu.:93.99
##  Max.   :6.0000                       Max.   : 1.40000   Max.   :94.77
##
##  cons.conf.idx      euribor3m      nr.employed     y
##  Min.   :-50.80   Min.   :0.634   Min.   :4964   no :29215
##  1st Qu.:-42.70   1st Qu.:1.344   1st Qu.:5099   yes: 3735
##  Median :-41.80   Median :4.857   Median :5191
##  Mean   :-40.49   Mean   :3.620   Mean   :5167
##  3rd Qu.:-36.40   3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :-26.90   Max.   :5.045   Max.   :5228
##
##  previously_contacted
##  no :31716
##  yes: 1234
##
##
##
##
##
```

Note that he “y” column dictates whether or not the client subscribed a term deposit. Also, that the client’s age may not actually have much consistent influence on whether or not the client subscribes.



Campaign Outcome
failure
nonexistent
success
1.0
0.8
0.6
0.4
0.2
0.0
20
40
60
80
Age

Graphs:



Logistic Regression Model

We'll proceed now by making a logistic regression model, where we use the y column as our target, and various other columns as our predictors. The "y" column is our target as, in this hypothetical scenario, the bank wants to predict what clients their campaigns are getting to subscribe for a term deposit.

We'll then generate a summary of the model, so we can see the residuals and what R thinks about the correlation between the predictors with the target.

```
# logistic regression model
glm <- glm(y~poutcome+duration+contact+previously_contacted+emp.var.rate+cons.price.idx+cons.conf.idx,

# summary
summary(glm)
```

```
##
## Call:
## glm(formula = y ~ poutcome + duration + contact + previously_contacted +
##     emp.var.rate + cons.price.idx + cons.conf.idx, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7555  -0.3332  -0.1981  -0.1491   3.3257
##
```

```
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)    
## (Intercept)            -1.347e+02  5.224e+00 -25.782  < 2e-16 ***
## poutcomenonexistent      6.098e-01  6.985e-02   8.731  < 2e-16 ***
## poutcomesuccess          9.952e-01  2.279e-01   4.366 1.26e-05 ***
## duration                 4.438e-03  7.948e-05  55.841  < 2e-16 ***
## contacttelephone        -1.057e+00  6.716e-02 -15.733  < 2e-16 ***
## previously_contactedyes  9.572e-01  2.264e-01   4.228 2.36e-05 ***
## emp.var.rate            -9.656e-01  2.236e-02 -43.181  < 2e-16 ***
## cons.price.idx           1.422e+00  5.621e-02  25.292  < 2e-16 ***
## cons.conf.idx            6.539e-02  4.127e-03  15.843  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23294  on 32949  degrees of freedom
## Residual deviance: 14333  on 32941  degrees of freedom
## AIC: 14351
##
## Number of Fisher Scoring iterations: 6
```

Looking through the summary, we can observe that R believes each of our chosen predictors to be effective predictors for the model, as each are getting a triple '*' significant code. We can also see that our null deviance and residual deviance are fairly high, which is rather concerning. However, this may be in relation to the data on the model. What is important to note that is a good sign, is that the residual deviance is significantly lower than that of the null deviance. This is specifically something we want to see, as the larger the difference is between the two, the better. But most importantly, keeping the residual deviance lower than the null deviance is very necessary. As not having either or can be a clear sign that the model doesn't explain the data very well.

The AIC and Fishing Scoring iteration count don't seem to be as applicable in what we're trying to accomplish here. But the AIC does also seem considerably high, which may indicate that there exist many other models that will better explain the data. Just like the deviances - the smaller, the better.

Naïve Bayes Model

Next, we'll create a naïve bayes model for the data. We'll keep y as our target variable, but instead, use the poutcome, duration, contact, and previously_contacted variables as predictors. The main reason we're removing a lot of the predictors used in Logistic Regression is due Naive Bayes assuming the predictors are conditionally independent of one another. Therefore, variables that may not be consistent with this have been removed.

```
# naïve bayes model
nb <- naiveBayes(y~poutcome+duration+contact+previously_contacted, data=train)

#summary
nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
```

```
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        no       yes
## 0.8866464 0.1133536
##
## Conditional probabilities:
##      poutcome
## Y        failure nonexistent    success
##   no  0.10022249  0.88653089 0.01324662
##   yes 0.12530120  0.67710843 0.19759036
##
##      duration
## Y         [,1]     [,2]
##   no  220.7540 207.6058
##   yes 553.0289 405.9890
##
##      contact
## Y      cellular telephone
##   no  0.6111244 0.3888756
##   yes 0.8342704 0.1657296
##
##      previously_contacted
## Y             no        yes
##   no  0.98493924 0.01506076
##   yes 0.78741633 0.21258367
```

Observing the probabilities from the summary of our Naïve Bayes Model, we can see that our model indicates that we have roughly an 11% chance of getting a positive value for our target (the "yes" result). We can see how the Naïve Bayes Model is coming to this conclusion through its breakdown of each conditional probability. Generally, we want to avoid values that end up being comparable to that of a coin toss (even splits between each predictor value). Fortunately, this isn't the case with any of our values. Additionally, there is a pretty sparse difference in quantitative value within the duration's conditional mean values.

Model Predictions

```r
# glm predictions
probs_glm <- predict(glm, newdata=test, type="response")
pred_glm <- ifelse(probs_glm>0.5, 2, 1)
acc_glm <- mean(pred_glm==as.integer(test$y))
```

For Logistic Regression Model:

```
##
## pred_glm    1    2
##        1 7145  551
##        2  188  354
```

```
## Accuracy:  0.91029376062151
```

```r
# nb predictions
pred_nb <- predict(nb, newdata=test, type="class")
```

For Naïve Bayes Model:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  7043  505
##        yes  290  400
##
##                Accuracy : 0.9035
##                  95% CI : (0.8969, 0.9098)
##     No Information Rate : 0.8901
##     P-Value [Acc > NIR] : 4.331e-05
##
##                   Kappa : 0.4492
##
##  Mcnemar's Test P-Value : 3.204e-14
##
##             Sensitivity : 0.44199
##             Specificity : 0.96045
##          Pos Pred Value : 0.57971
##          Neg Pred Value : 0.93309
##              Prevalence : 0.10986
##          Detection Rate : 0.04856
##    Detection Prevalence : 0.08376
##       Balanced Accuracy : 0.70122
##
##        'Positive' Class : yes
##
```

Naïve Bayes Model vs Logistic Regression Model

As we can see from the accuracy of predictions on this data set, the Logistic Regression Model is getting a slightly higher accuracy over Naïve Bayes. To understand why this is the case with this data set, we need to understand the strengths & weaknesses of both Logistic Regression & Naïve Bayes.

Logistic Regression is strong in that it does well in separating classes when they are linearly separable, and gives a nice output in probabilities that can be analyzed conveniently. It's also incredibly inexpensive. However, it's weak due to it tending to underfit data, due to its lack of flexibility in making non-linear decisions.

Naïve Bayes is strong in that it works well this smaller data sets and has high interpretability. It's also great at handling high dimensions of data. However, it's weakness lies with the fact that it tends to get outperformed for large data sets by other classifiers. The algorithm is naïve in the assumptions it makes, as well. When the predictors are not independent, the algorithm will assume they are, impacting the algorithm's performance.

The data set we chose is rather large in size; triple that of the minimum we were asked to find online (10,000). Additionally, when we'd initially made the naïve bayes model, it used all the predictors that were used in the logistic regression model. Needless to say, the results were worse than the one we ended with, due to the algorithm assuming all the predictors are independent.

Classification Metrics

Throughout this assignment, we've used various classification metrics to gauge how the algorithm is performing on the data set. The last part of this write-up will discuss the significance of each of these metrics in the scope of classification.

In the Logistic Regression Model, the summary provided us with metrics similar to Linear Regression. This included the deviance residuals, as well as significance codes for the coefficients gathered on each predictor in the model. Since we went over what both mean in the Regression portion of the assignment, we'll skip over explaining them.

Where it differs from Linear Regression is in the bottom-most part of the summary. We get interesting details about the model's Null Deviance, as well as the Residual Deviance. The null deviance describes how little the model fits the data, in consideration of only the intercept, while the residual deviance describes how little the entire model fits the data. Generally, we're wanting the residual deviance to be much lower than the null deviance, and for both of these to be as low as possible. We also are given the AIC, standing for Akaike Information Criterion, which helps us draw comparisons between models. Generally, the lower this value is, the better. It'll be closest to an optimal value when the model isn't very complex, and has few predictors. Lastly, we get a count for Fisher Scoring iterations, which can be useful when solving the maximum likelihood problem.

In the Naïve Bayes Model, the summary provided us with the A-Priori Probabilities of the target variable, as well as all of the Conditional Probabilities of each predictor in relation to the target. The A-Prioi Probabilities simply tell us the general probability of each value of the target variable. In the model we made, we can see that we generally have about an 11% chance of getting someone to subscribe through our campaigns. We can break this down further by looking at our Conditional Probabilities, which break down the probabilities of getting each value of the target variable for each value of the particular predictor. When given a quantitative value, it will simply use the mean of the values that correspond. These are very useful in interpreting the model's basis for predicting, but the main drawback here, of course, is in the fact that it'll apply all of these conditional probabilities to all other variations of data outside of this data set. Additionally, it treats the predictors completely independent of one another, which can affect prediction accuracy with high predictor counts.