Homework 4

4375 Machine Learning with Dr. Mazidi

Garrett Strealy

9/19/2021

This script will run Logistic Regression and Naive Bayes on the BreastCancer data set which is part of package mlbench.

Step 1: Data exploration

- Load package mlbench, installing it at the console if necessary
- Load data(BreastCancer)
- Run str() and head() to look at the data
- Run summary() on the Class column
- Use R code to calculate and output the percentage in each class, with a label using paste()

Comment on the types of predictors available in terms of their data types:

There are five ordinal factors and four non-ordinal factors available as predictors of Class.

```
library(mlbench)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(e1071)

data("BreastCancer")
str(BreastCancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4
...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2
...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1
...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1
...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2
...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1
...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

```
head(BreastCancer)
```

Id <chr>	Cl.thickness <ord>	Cell.size <ord>	Cell.shape <ord>	Marg.adhesion <ord>	Epith.c.size <ord>	Bare.nuclei <fct>	Bl.c <fc
1 1000025	5	1	1	1	2	1	3
2 1002945	5	4	4	5	7	10	3
3 1015425	3	1	1	1	2	2	3
4 1016277	6	8	8	1	3	4	3
5 1017023	4	1	1	3	2	1	3
6 1017122	8	10	10	8	7	10	9

6 rows | 1-9 of 12 columns

```
summary(BreastCancer$Class)
```

```
##    benign malignant
##       458       241
```

```
ben <- sum(BreastCancer$Class == 'benign')
ben <- as.double(ben / (nrow(BreastCancer)))
print(paste("Percent benign: ", ben))
```

```
## [1] "Percent benign:  0.655221745350501"
```

```
mal <- sum(BreastCancer$Class == 'malignant')
mal <- as.double(mal / (nrow(BreastCancer)))
print(paste("Percent malignant: ", mal))
```

```
## [1] "Percent malignant:  0.344778254649499"
```

Step 2: First logistic regression model

- Cell.size and Cell.shape are in one of 10 levels
- Build a logistic regression model called glm0, where Class is predicted by Cell.size and Cell.shape
- Do you get any error or warning messages? Google the message and try to decide what happened
- Run summary on glm0 to confirm that it did build a model
- Write about why you think you got this warning message and what you could possibly do about it. List the source of your information in a simple markdown link.

Your commentary here:

I received the warning “glm.fit: fitted probabilities integerally 0 or 1 occurred.” This warning means that the model is predicting absolute probabilities like 0 and 1. If you have some variable(s) which perfectly separates zeroes and ones in target variable, R will yield this “perfect or quasi perfect separation” warning message. A possible solution to this is to use a Bayesian analysis with non-informative prior assumptions as proposed by Gelman et al (2008). Gelman recommends to put a Cauchy prior with median 0.0 and scale 2.5 on each coefficient (normalized to have mean 0.0 and a SD of 0.5). This will regularize the coefficients and pull them just slightly towards zero. Sources: 1) https://stats.stackexchange.com/questions/11109/how-to-deal-with-perfect-separation-in-logistic-regression (https://stats.stackexchange.com/questions/11109/how-to-deal-with-perfect-separation-in-logistic-regression) 2) https://projecteuclid.org/journals/annals-of-applied-statistics/volume-2/issue-4/A-weakly-informative-default-prior-distribution-for-logistic-and-other/10.1214/08-AOAS191.full (https://projecteuclid.org/journals/annals-of-applied-statistics/volume-2/issue-4/A-weakly-informative-default-prior-distribution-for-logistic-and-other/10.1214/08-AOAS191.full)

```
glm0 <- glm(Class ~ Cell.size + Cell.shape, data = BreastCancer, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm0)
```

```
##
## Call:
## glm(formula = Class ~ Cell.size + Cell.shape, family = "binomial",
##     data = BreastCancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6380  -0.0844  -0.0844   0.0000   3.3583
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     7.77977  757.06731   0.010    0.992
## Cell.size.L    10.45177  950.68968   0.011    0.991
## Cell.size.Q     0.04063 1479.65505   0.000    1.000
## Cell.size.C    10.70546  948.84001   0.011    0.991
## Cell.size^4    12.06582 1241.92612   0.010    0.992
## Cell.size^5     0.74199  792.70275   0.001    0.999
## Cell.size^6    -3.08210 1011.79271  -0.003    0.998
## Cell.size^7     7.47104 1044.50459   0.007    0.994
## Cell.size^8     5.60143  830.93455   0.007    0.995
## Cell.size^9   -10.22144 1812.16584  -0.006    0.995
## Cell.shape.L   18.15803 2619.03253   0.007    0.994
## Cell.shape.Q    9.14381 1500.17059   0.006    0.995
## Cell.shape.C    5.50082 1302.51288   0.004    0.997
## Cell.shape^4   -2.23752 2679.86482  -0.001    0.999
## Cell.shape^5   -5.76978 3193.32589  -0.002    0.999
## Cell.shape^6   -5.58415 2713.54579  -0.002    0.998
## Cell.shape^7   -3.94569 1740.80762  -0.002    0.998
## Cell.shape^8   -1.82009  827.39672  -0.002    0.998
## Cell.shape^9   -0.77209  257.90962  -0.003    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 900.53  on 698  degrees of freedom
## Residual deviance: 198.66  on 680  degrees of freedom
## AIC: 236.66
##
## Number of Fisher Scoring iterations: 19
```

Step 3: Data Wrangling

Notice in the summary() of glm0 that most of the levels of Cell.size and Cell.shape became predictors and that they had very high p-values, that is, they are not good predictors. We would need a lot more data to build a good logistic regression model this way. Many examples per factor level are generally required for model building. A better approach might be to just have 2 levels for each variable.

In this step:

- Add two new columns to BreastCancer as listed below:
    a. Cell.small which is a binary factor that is 1 if Cell.size==1 and 0 otherwise
    b. Cell.regular which is a binary factor that is 1 if Cell.shape==1 and 0 otherwise
- Run summary() on Cell.size and Cell.shape as well as the new columns

- Comment on the distribution of the new columns
- Do you think what we did is a good idea? Why or why not?

Your commentary here:

This was a good idea. The new columns are well balanced as opposed to the unbalanced original columns. We eliminated the high skewness towards value 1 of both original columns by grouping the less common values of 2-10 together.

```
BreastCancer$Cell.small <- ifelse(BreastCancer$Cell.size == 1, 1, 0)
BreastCancer$Cell.small <- as.factor(BreastCancer$Cell.small)

summary(BreastCancer$Cell.size)
```

```
##   1   2   3   4   5   6   7   8   9  10
## 384  45  52  40  30  27  19  29   6  67
```

```
summary(BreastCancer$Cell.small)
```

```
##   0   1
## 315 384
```

```
BreastCancer$Cell.regular <- ifelse(BreastCancer$Cell.shape == 1, 1, 0)
BreastCancer$Cell.regular <- as.factor(BreastCancer$Cell.regular)

summary(BreastCancer$Cell.shape)
```

```
##   1   2   3   4   5   6   7   8   9  10
## 353  59  56  44  34  30  30  28   7  58
```

```
summary(BreastCancer$Cell.regular)
```

```
##   0   1
## 346 353
```

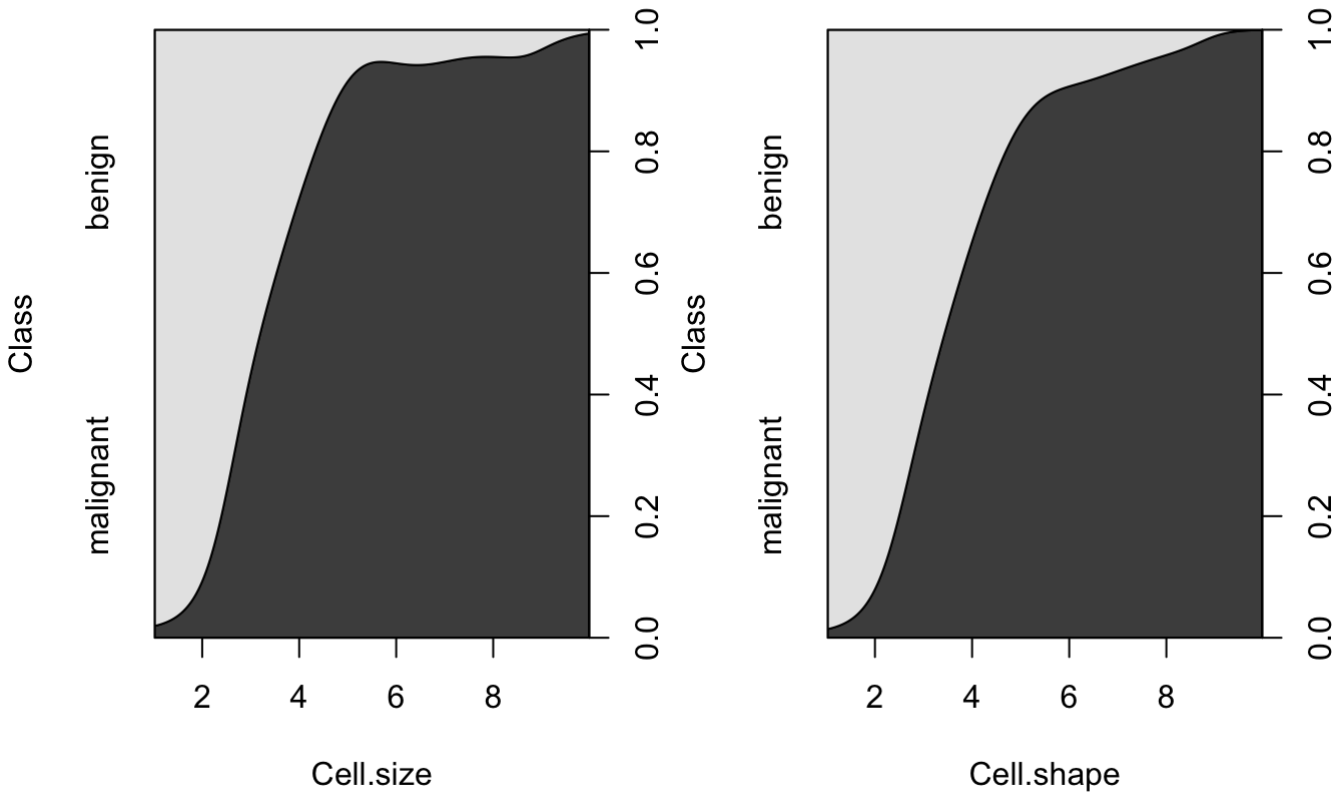
Step 4: Examine the relationship of malignancy to Cell.size and Cell.shape

- Create conditional density plots using the original Cell.size and Cell.shape, but first, attach() the data to reduce typing
- Then use par(mfrow=c(1,2)) to set up a 1x2 grid for two cdplot() graphs with Class~Cell.size and Class~Cell.shape
- Observing the plots, write a sentence or two comparing size and malignant, and shape and malignant
- Do you think our cutoff points for size==1 and shape==1 were justified now that you see this graph? Why or why not?

Your commentary here:

The curves rise sharply at approximately 1 on both graphs, so we can conclude that the cutoff points of 1 were justified based on the graphs.

```
# your code here
attach(BreastCancer)
par(mfrow=c(1,2))
cdplot(Class~Cell.size)
cdplot(Class~Cell.shape)
```



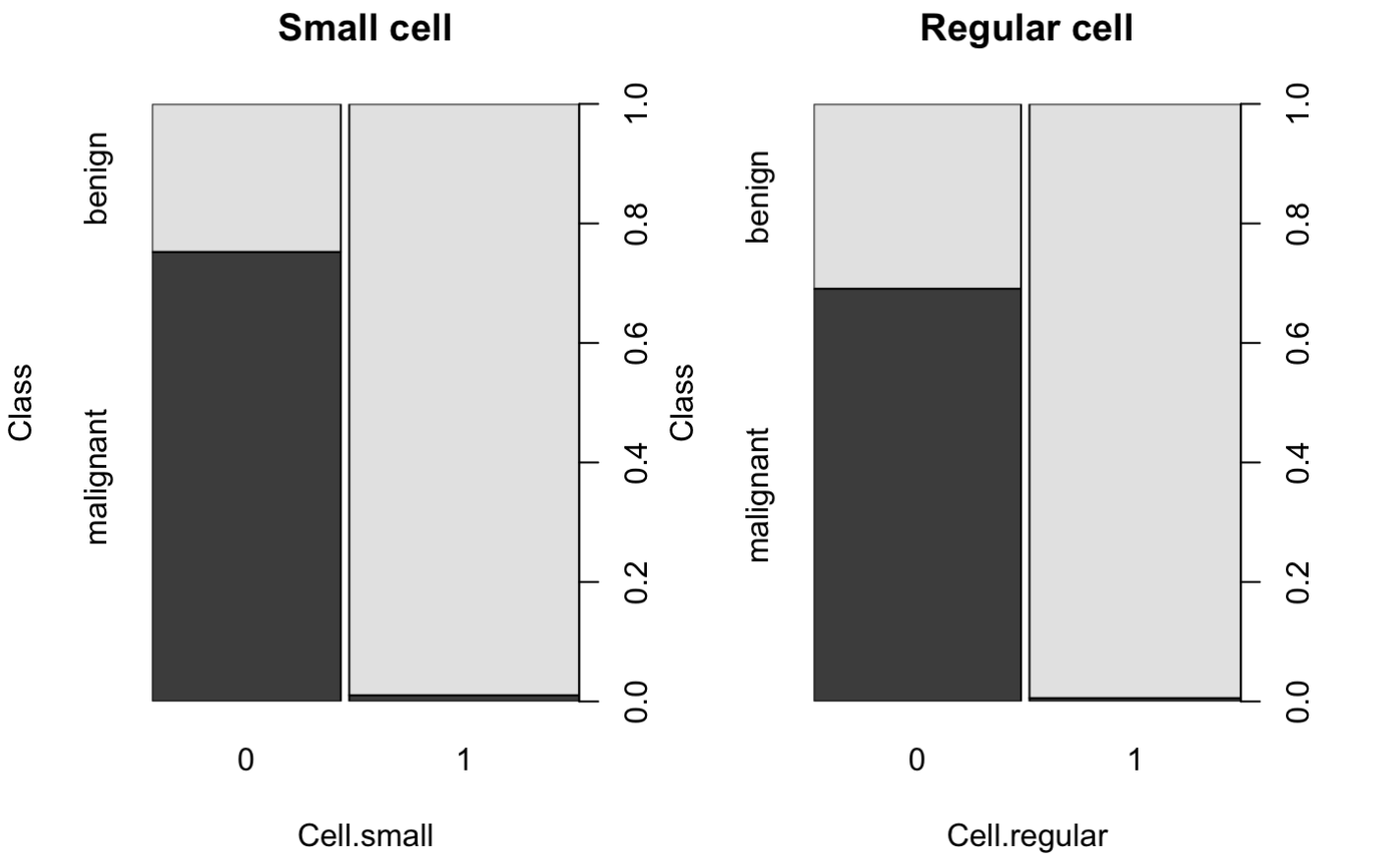
Step 5: Explore the new columns

- Create plots (not cdplots) with the two new columns
- Again, use par(mfrow=c(1,2)) to set up a 1x2 grid for two plot() graphs with Class~Cell.small and Class~Cell.regular
- Now create two cdplot() graphs for the new columns
- Compute and output with labels the following: ((Examples on p. 142 may help)
    a. calculate the percentage of malignant observations that are small
    b. calculate the percentage of malignant observations that are small
    c. calculate the percentage of malignant observations that are regular
    d. calculate the percentage of malignant observations that are not regular
- Write whether you think small and regular will be good predictors

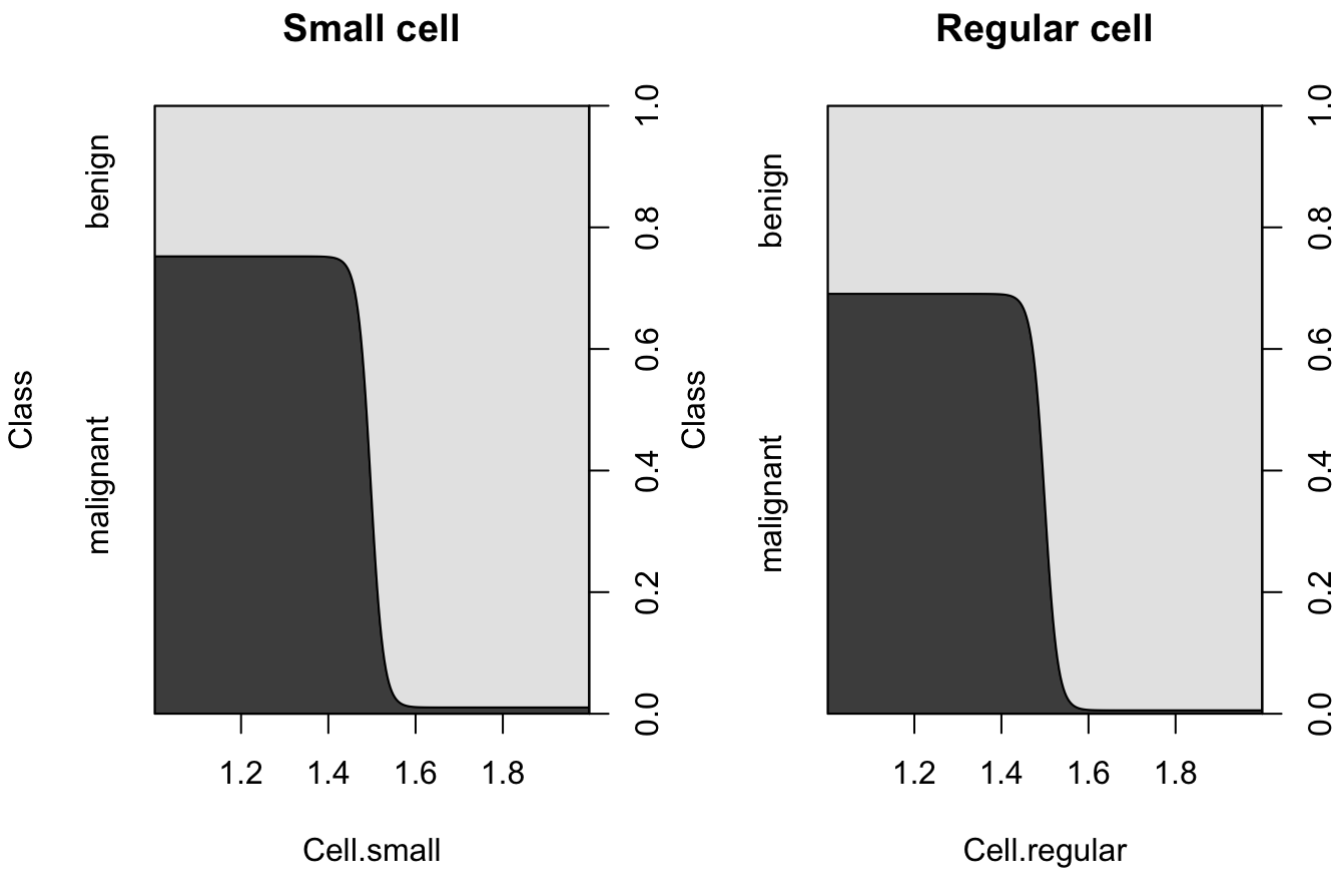
Your commentary here:

82% of benign cases are small, while 98% of malignant cases are not small. 76% of benign cases are regular, while 99% of malignant cases are not regular. Both small and regular should be good predictors based on this.

```
# plots here
par(mfrow=c(1,2))
plot(Class~Cell.small, data=BreastCancer, main="Small cell")
plot(Class~Cell.regular, data=BreastCancer, main="Regular cell")
```




```
cdplot(Class~Cell.small, main="Small cell")
cdplot(Class~Cell.regular, main="Regular cell")
```




```
# calculations and output here
count_benign <- c(
         length(Class[Class=="benign"]),
         length(Class[Class=="malignant"])
       )

# likelihood for small
lh_small <- matrix(rep(0, 4), ncol = 2)
       for (bn in c(1, 2)) {
         for (sm in c("0","1")) {
           lh_small[bn, as.integer(sm) + 1] <-
             nrow(BreastCancer[Cell.small == sm & as.numeric(Class) == bn,]) /
              count_benign[bn]
          }
       }

#print("Likelihood values for p(Class|Cell.small):")
#lh_small
print(paste("a. Percentage of malignant observations that are small: ", lh_small[2,2]))
```

```
## [1] "a. Percentage of malignant observations that are small:  0.016597510373444"
```

```
print(paste("b. Percentage of malignant observations that are not small: ", lh_small[2,1
]))
```

```
## [1] "b. Percentage of malignant observations that are not small:  0.983402489626556"
```

```
# likelihood for regular
lh_regular <- matrix(rep(0, 4), ncol = 2)
       for (bn in c(1, 2)) {
         for (rg in c("0","1")) {
           lh_regular[bn, as.integer(rg) + 1] <-
             nrow(BreastCancer[Cell.regular == rg & as.numeric(BreastCancer$Class) == b
n,]) /
              count_benign[bn]
         }
       }

#print("Likelihood values for p(Class|Cell.regular):")
#lh_regular
print(paste("c. Percentage of malignant observations that are regular: ", lh_regular[2,2
]))
```

```
## [1] "c. Percentage of malignant observations that are regular:  0.00829875518672199"
```

```
print(paste("d. Percentage of malignant observations that are not regular: ", lh_regular
[2,1]))
```

```
## [1] "d. Percentage of malignant observations that are not regular:  0.99170124481327
8"
```

Step 6: Train/test split

- Divide the data into 80/20 train/test sets, using seed 1234

```
set.seed(1234)

i <- sample(1:nrow(BreastCancer), 0.8 * nrow(BreastCancer), replace = FALSE)

train <- BreastCancer[i,]
test <- BreastCancer[-i,]
```

Step 7: Build a logistic regression model

- Build a logistic regression model predicting malignant with two preditors: Cell.small and Cell. regular
- Run summary() on the model
- Which if any of the predictors are good predictors?
- Comment on the model null variance versus residual variance and what it means
- Comment on the AIC score

Your commentary here:

Both small and regular are good predictors with low P values. The null deviance measures the lack of fit of the model, considering only the intercept. The residual deviance measures the lack of fit of the entire model. We want to see a residual deviance much lower than null deviance, which this model satisfies (721 vs 255). AIC is useful in comparing models. The lower the AIC, the better. AIC shows a preference for less complex models with fewer predictors.

```
glm1 <- glm(Class ~ Cell.regular + Cell.small, data = train, family = binomial)

summary(glm1)
```

```
##
## Call:
## glm(formula = Class ~ Cell.regular + Cell.small, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8314  -0.0445  -0.0445   0.6433   3.7198
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.4701     0.1672   8.791  < 2e-16 ***
## Cell.regular1  -3.7044     0.7603  -4.873 1.10e-06 ***
## Cell.small1    -4.6830     0.7411  -6.319 2.64e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 721.78  on 558  degrees of freedom
## Residual deviance: 255.73  on 556  degrees of freedom
## AIC: 261.73
##
## Number of Fisher Scoring iterations: 8
```

Step 8: Evaluate on the test data

- Test the model on the test data
- Compute and output accuracy
- Output the confusion matrix and related stats using the confusionMatrix() function in the caret package
- Were the mis-classifications more false positives or false negatives?

Your commentary here:

The mis-classifications were more false negatives than false positives.

```
probs <- predict(glm1, newdata = test, type = "response")
pred <- ifelse(probs > 0.5, 2, 1)
acc <- mean(pred == as.integer(test$Class))
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.885714285714286"
```

```
confusionMatrix(as.factor(pred), as.factor(as.integer(test$Class)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 79  2
##          2 14 45
##
##                Accuracy : 0.8857
##                  95% CI : (0.821, 0.9332)
##     No Information Rate : 0.6643
##     P-Value [Acc > NIR] : 1.386e-09
##
##                   Kappa : 0.759
##
##  Mcnemar's Test P-Value : 0.00596
##
##             Sensitivity : 0.8495
##             Specificity : 0.9574
##          Pos Pred Value : 0.9753
##          Neg Pred Value : 0.7627
##              Prevalence : 0.6643
##          Detection Rate : 0.5643
##    Detection Prevalence : 0.5786
##       Balanced Accuracy : 0.9035
##
##        'Positive' Class : 1
##
```

Step 9: Model coefficients

- The coefficients from the model are in units of logits. Extract and output the coefficient of Cell.small with glm1$coefficients[]
- Find the estimated probability of malignancy if Cell.small is true using exp(). See the example on p. 107 of the pdf.
- Find the probability of malignancy if Cell.small is true over the whole BreastCancer data set and compare results. Are they close? Why or why not?

Your commentary here:

The real probability is nearly double the estimated probability (.0165 vs .009). This inaccuracy reflects the limitations of the model glm1, although the values are quite close and are a good sign for the model's prediction capabilities.

```
sm <- coef(glm1)["Cell.small1"]
sm
```

```
## Cell.small1
##   -4.682999
```

```
p <- exp(sm) / (1 + exp(sm))
p
```

```
## Cell.small1
##  0.00916643
```

```
# from step 5
lh_small[2, 2]
```

```
## [1] 0.01659751
```

Step 10: More logistic regression models

- Build two more models, glm_small using only Cell.small, and glm_regular using Cell.regular as the predictor
- Use anova(glm_small, glm_regular, glm1) to compare all 3 models, using whatever names you used for your models. Analyze the results of the anova().
- Also, compare the 3 AIC scores of the models. Feel free to use the internet to help you interpret AIC scores.

Your commentary here:

glm1 has an AIC of 261, while glm_small is 304 and glm_regular is 374. glm1 has the best AIC score, meaning it is the most parsimonious model of the three.

```
glm_small <- glm(Class ~ Cell.small, data = train, family = binomial)
#summary(glm_small)
glm_regular <- glm(Class ~ Cell.regular, data = train, family = binomial)
#summary(glm_regular)
anova(glm_small, glm_regular, glm1)
```

	Resid. Df <dbl>	Resid. Dev <dbl>	Df <dbl>	Deviance <dbl>
1	557	300.7544	*NA*	*NA*
2	557	370.0222	0	-69.26775
3	556	255.7342	1	114.28797

3 rows

Step 11: A Naive Bayes model

- Build a Naive Bayes Model Class ~ Cell.small + Cell.regular on the training data using library e1071
- Output the model parameters
- Aand nswer the following questions:
    a. What percentage of the training data is benign?

b. What is the likelihood that a malignant sample is not small?
c. What is the likelihood that a malignant sample is not regular?

Your commentary here: a. 65% of the training data is benign. b. A malignant sample has a 98% likelihood of not being small. c. A malignant sample has a 98% likelihood of not being regular.

```
nb1 <- naiveBayes(Class ~ Cell.small + Cell.regular, data = train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##    benign malignant
## 0.6529517 0.3470483
##
## Conditional probabilities:
##            Cell.small
## Y                    0          1
##   benign    0.16438356 0.83561644
##   malignant 0.98969072 0.01030928
##
##            Cell.regular
## Y                    0          1
##   benign    0.23835616 0.76164384
##   malignant 0.98969072 0.01030928
```

Step 12: Evaluate the model

- Predict on the test data with Naive Bayes model
- Output the confusion matrix
- Are the results the same or different? Why do you think that is the case?

Your commentary here:

The results are the same as the logistic regression model. Naive Bayes and Logistic Regression models can produce the same results as the training set size approaches infinity IF the Naive Bayes assumption holds that the x(sub i)'s are conditionally independent of one another given y. The difference between the two is that Naive Bayes uses a restrictive generative model to come up with σ (w0 + w · x), while Logistic Regression is less restrictive. Source: https://www.cs.princeton.edu/courses/archive/spr07/cos424/scribe_notes/0410.pdf (https://www.cs.princeton.edu/courses/archive/spr07/cos424/scribe_notes/0410.pdf)

```
p1 <- predict(nb1, newdata = test, type = "class")
confusionMatrix(p1, test$Class)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign        79         2
##   malignant     14        45
##
##                Accuracy : 0.8857
##                  95% CI : (0.821, 0.9332)
##     No Information Rate : 0.6643
##     P-Value [Acc > NIR] : 1.386e-09
##
##                   Kappa : 0.759
##
##  Mcnemar's Test P-Value : 0.00596
##
##             Sensitivity : 0.8495
##             Specificity : 0.9574
##          Pos Pred Value : 0.9753
##          Neg Pred Value : 0.7627
##              Prevalence : 0.6643
##          Detection Rate : 0.5643
##    Detection Prevalence : 0.5786
##       Balanced Accuracy : 0.9035
##
##        'Positive' Class : benign
##
```