Homework 4

4375 Machine Learning with Dr. Mazidi

Cris Chou

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 is alot of predictors, most of which relate to the cell or parts of the cell.

```
# your code here
library(mlbench)
data(BreastCancer)
df <- 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 Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1             2
## 4 1016277            6         8          8             1             3
## 5 1017023            4         1          1             3             2
## 6 1017122            8        10         10             8             7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant
```

```
summary(BreastCancer[,c(11)])
```

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

```
total <- 458 + 241
benignAmount <- 458/total * 100
malignantAmount <- 241 / total * 100
print(paste(benignAmount,"% are benign class and ",malignantAmount,"% are malignant."))
```

```
## [1] "65.5221745350501 % are benign class and  34.4778254649499 % are malignant."
```

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: Because the dataset is whole and complete and we are not using training data, so the predictors have very higher accuracy leading to a very good model.

```
# your code here
glm0 <- glm(Class~Cell.size+Cell.shape, data = df, 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 = df)
##
## 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  757.06727   0.010    0.992
## Cell.size.L    10.45177  950.68968   0.011    0.991
## Cell.size.Q     0.04063 1479.65504   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 1011.79270  -0.003    0.998
## Cell.size^7     7.47104 1044.50458   0.007    0.994
## Cell.size^8     5.60143  830.93455   0.007    0.995
## Cell.size^9   -10.22144 1812.16582  -0.006    0.995
## Cell.shape.L   18.15803 2619.03235   0.007    0.994
## Cell.shape.Q    9.14381 1500.17053   0.006    0.995
## Cell.shape.C    5.50082 1302.51283   0.004    0.997
## Cell.shape^4   -2.23752 2679.86462  -0.001    0.999
## Cell.shape^5   -5.76978 3193.32564  -0.002    0.999
## Cell.shape^6   -5.58415 2713.54558  -0.002    0.998
## Cell.shape^7   -3.94569 1740.80748  -0.002    0.998
## Cell.shape^8   -1.82009  827.39666  -0.002    0.998
## Cell.shape^9   -0.77209  257.90960  -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: The distribution was good for the new columns having about a 50 50 split. This was probably a good idea since it gives us another predictor which has a balanced distribution.

```
# BreastCancer$Cell.small column
df$Cell.small <- 0
df$Cell.small[df$Cell.size==1] <- 1
df$Cell.small <- factor(df$Cell.small)

df$Cell.regular <- 0
df$Cell.regular[df$Cell.shape==1] <- 1
df$Cell.regular <- factor(df$Cell.regular)

df$Class <- factor(df$Class)

summary(df[,c(3,4,12,13)])
```

```
##    Cell.size     Cell.shape  Cell.small Cell.regular
##  1      :384   1      :353   0:315      0:346
##  10     : 67   2      : 59   1:384      1:353
##  3      : 52   10     : 58
##  2      : 45   3      : 56
##  4      : 40   4      : 44
##  5      : 30   5      : 34
##  (Other): 81   (Other): 95
```

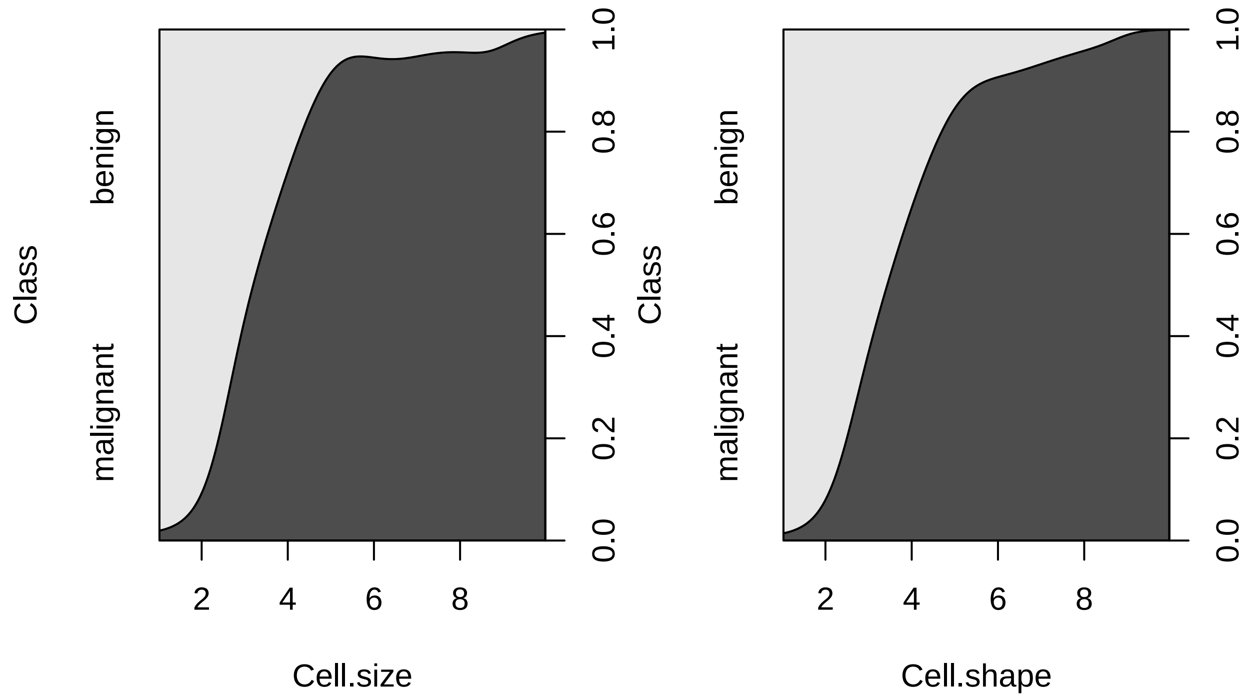
```
# BreastCancer$Cell.regular column
```

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 smaller cell sizes and non regular sized shapes usually correlated with being malignant. I think our cutoff was justified since the data was relatively balanced.

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



Step 5: Explore the new columns

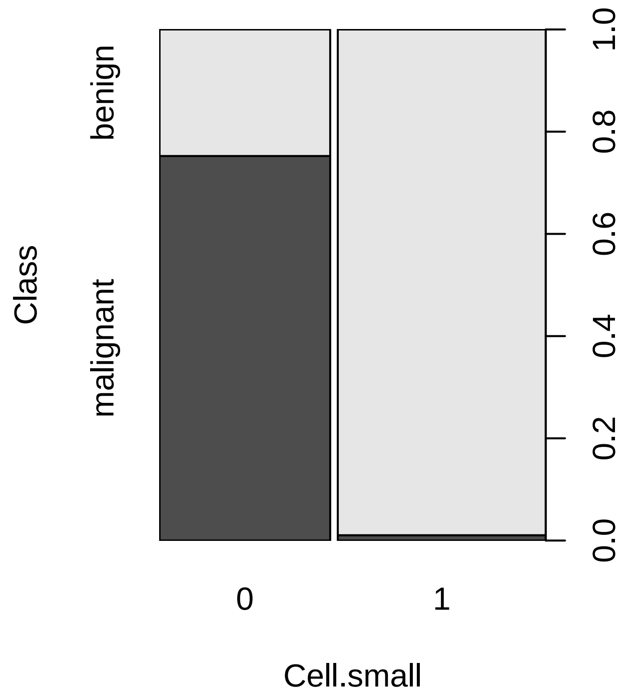
- Create plots (not cdplots) with the two new columns
- Again, use par(mfrow=c(1,2)) to set up a 1x2 grid for two plot() graphs with Class~Cell.small and Class~Cell.regular
- Now create two cdplot() graphs for the new columns
- Compute and output with labels the following: ((Examples on p. 142 may help)

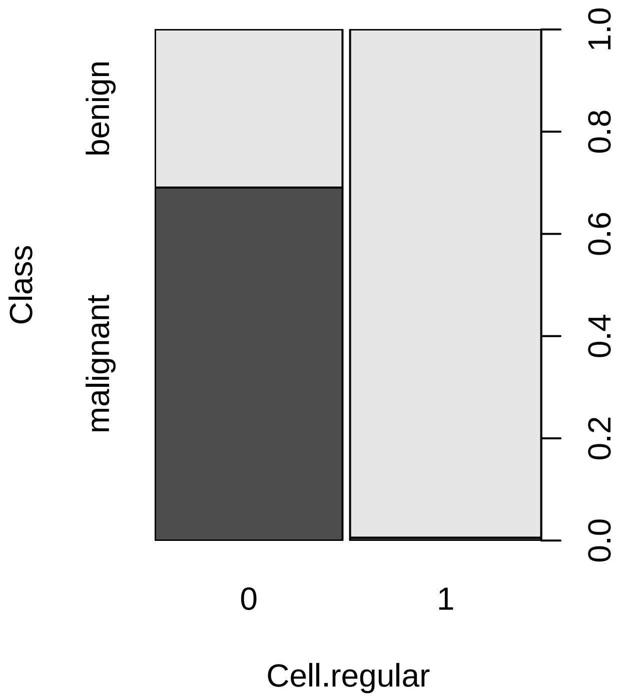
  a. calculate the percentage of malignant observations that are small
  b. calculate the percentage of malignant observations that are small
  c. calculate the percentage of malignant observations that are regular
  d. calculate the percentage of malignant observations that are not regular

- Write whether you think small and regular will be good predictors

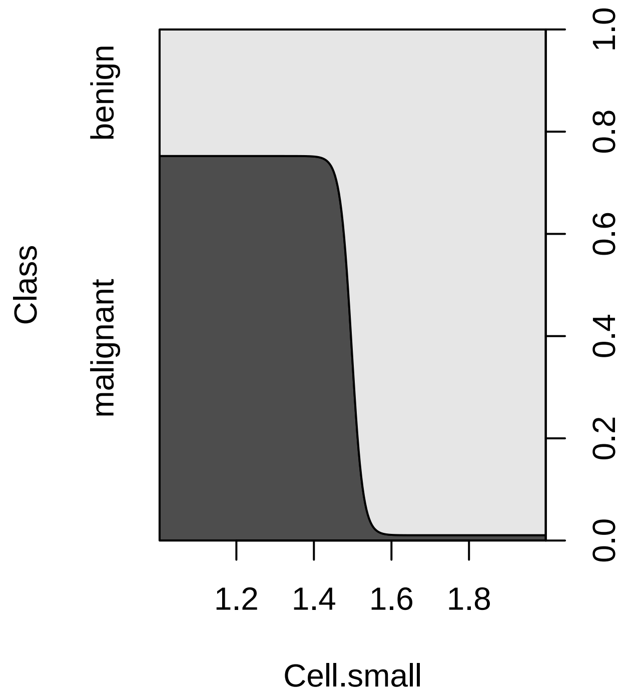
Your commentary here:Small and regular will probably be good predictors as it shows that most malignant cases have non small cells and irregular cell shapes.

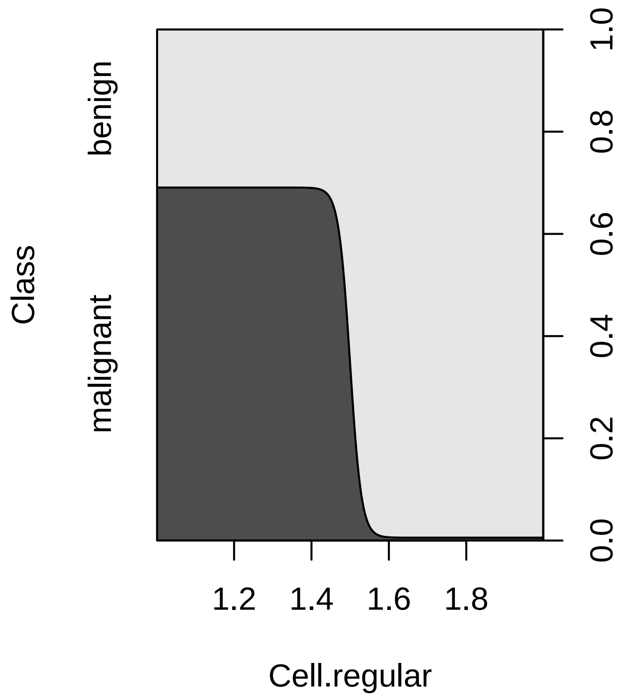
```r
# plots here
par(mfrow=c(1,2))
plot(Class~Cell.small,data = df)
plot(Class~Cell.regular,data = df)
```





```
cdplot(Class~Cell.small,data = df)
cdplot(Class~Cell.regular,data = df)
```





```
# calculations and output here
newList1 <- df[,c(11,12)]
newList2 <- df[,c(11,13)]

newList1$mal<- FALSE
newList1$mal[newList1$Class =="malignant"] <- TRUE
newList2$mal<- FALSE
newList2$mal[newList2$Class =="malignant"] <- TRUE
malSmall1 <- subset(newList1,mal==TRUE)
summary(malSmall1$Cell.small)
```

```
##   0   1
## 237   4
```

```
malReg1 <- subset(newList2,mal==TRUE)
summary(malReg1$Cell.regular)
```

```
##   0   1
## 239   2
```

```
print(paste("Malignant and Small percentage: ", 4/241 * 100, "%"))
```

```
## [1] "Malignant and Small percentage:  1.6597510373444 %"
```

```
print(paste("Malignant and not Small percentage", 237/241 * 100, "%"))
```

```
## [1] "Malignant and not Small percentage 98.3402489626556 %"
```

```
print(paste("Malignant and Regular percentage: ", 2/241 * 100, "%"))
```

```
## [1] "Malignant and Regular percentage:  0.829875518672199 %"
```

```
print(paste("Malignant and not Regular percentage", 237/241 * 100, "%"))
```

```
## [1] "Malignant and not Regular percentage 98.3402489626556 %"
```

Step 6: Train/test split

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

```
# your code here
set.seed(1234)
i <- sample(1:nrow(df), .8*nrow(df),replace=FALSE)
train <- df[i,]
test <- df[-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: The residual deviance is significantly lower than the null deviance which is a good sign. The AIC is a little high but not the highest.

```
# your code here
glm1 <- glm(Class == "malignant" ~Cell.regular+Cell.small,data = train, family = "binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = Class == "malignant" ~ 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 misclassifications were more false negatives than false positives.

```
# your code here
library(caret)
```

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

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

```
pred <- predict(glm1, newdata=test, type = "response")
pr <- ifelse(pred > .5, "malignant", "benign")
pr1 <- ifelse(pred >.5, 2,1 )#if use string for acc doesn't work
acc1 <- mean(pr1==as.integer(test$Class))
print(paste("glm1 accuracy = ",acc1))
```

```
## [1] "glm1 accuracy =  0.885714285714286"
```

```
confusionMatrix(as.factor(pr),test$Class,positive="malignant")
```

```
## 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 : 0.9574
##             Specificity : 0.8495
##          Pos Pred Value : 0.7627
##          Neg Pred Value : 0.9753
##              Prevalence : 0.3357
##          Detection Rate : 0.3214
##    Detection Prevalence : 0.4214
##       Balanced Accuracy : 0.9035
##
##        'Positive' Class : malignant
##
```

```
#table(pr,test$Class)
```

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 probability of malignancy was 1.6597510373444 from step 5. It is sort of close to the predicted possiblity but a bit lower. However in the 2nd model where I used Cell.regular and Cell.small as predictors the estimated was much closer.

```
# your code here
glm1$coefficients[3]
```

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

```
glmTest <- glm(Class~Cell.regular+Cell.small, data = df, family = "binomial")

glmTest$coefficients[3]
```

```
## Cell.small1
##   -4.040546
```

```
estProb <- exp(glm1$coefficients[3])/(1+exp(glm1$coefficients[3]))
#first probablity based off of Cell.shape and Cell.size, 2nd one based off of Cell.regular and Cell.sma
print(paste("The estimated probablity for malignancy based of regular cells is ",estProb * 100,"% (usin
```

```
## [1] "The estimated probablity for malignancy based of regular cells is  0.916643037925551 % (using C
estProb2 <- exp(glmTest$coefficients[3])/(1+exp(glmTest$coefficients[3]))
print(paste("The estimated probablity for malignancy based of regular cells is ",estProb2 * 100,"% (usi
```

```
## [1] "The estimated probablity for malignancy based of regular cells is  1.72838857161458 % (using Cel
```

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: The comparison shows that the 3rd model has the lowest residual deviation. Its AIC was also the lowest showing that it was the best model.

```
# your code here
glm_small <- glm(Class== "malignant"~Cell.small,data = train, family="binomial")
glm_regular <- glm(Class=="malignant"~Cell.regular, data = train, family="binomial")
anova(glm_small, glm_regular, glm1)
```

```
## Analysis of Deviance Table
##
## Model 1: Class == "malignant" ~ Cell.small
## Model 2: Class == "malignant" ~ Cell.regular
## Model 3: Class == "malignant" ~ Cell.regular + Cell.small
##   Resid. Df Resid. Dev Df Deviance
## 1       557     300.75
## 2       557     370.02  0  -69.268
## 3       556     255.73  1  114.288
```

```
summary(glm_small)
```

```
##
## Call:
## glm(formula = Class == "malignant" ~ Cell.small, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6942  -0.1143  -0.1143   0.7375   3.1729
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1632     0.1479   7.864 3.71e-15 ***
## Cell.small1  -6.1903     0.7246  -8.544  < 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: 721.78  on 558  degrees of freedom
## Residual deviance: 300.75  on 557  degrees of freedom
## AIC: 304.75
##
## Number of Fisher Scoring iterations: 7
```

```
summary(glm_regular)
```

```
##
## Call:
## glm(formula = Class == "malignant" ~ Cell.regular, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5266  -0.1197  -0.1197   0.8645   3.1438
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.7916     0.1292   6.125 9.07e-10 ***
## Cell.regular1  -5.7261     0.7212  -7.939 2.04e-15 ***
## ---
## 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: 370.02  on 557  degrees of freedom
## AIC: 374.02
##
## Number of Fisher Scoring iterations: 7
```

```
summary(glm1)
```

```
##
## Call:
## glm(formula = Class == "malignant" ~ 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 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. 65.29517% is benign b. 98.969072% c. 98.969072%

```
# your code here
library(e1071)
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 confusion matrix is the same. Its the same because they are both classifying and since the data is well balanced

```r
# your code here
predNB <- predict(nb1,  newdata=test)
#head(predNB, n=2)
library(caret)
confusionMatrix(predNB,test$Class,positive="malignant")
```

```
## 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 : 0.9574
##             Specificity : 0.8495
##          Pos Pred Value : 0.7627
##          Neg Pred Value : 0.9753
##              Prevalence : 0.3357
##          Detection Rate : 0.3214
##    Detection Prevalence : 0.4214
##       Balanced Accuracy : 0.9035
##
##        'Positive' Class : malignant
##
```