Predicting Cancer Malignancy with Logistic Regression

Cameron Mirhossaini

2022

Introduction

We are provided a data set which includes 10 features, one of which is malignancy, called Class. The purpose of this exploration is to identify how we can use the 9 other features to predict cancer malignancy. We will also perform some exploratory analysis on the data set. This study was conducted to learn if a new method called fine needle aspiration (which draws only a small tissue sample) could be effective in determining tumor status and prognosis. We take advantage of this study to explore the power of logistic regression. The features include:

Class - 0 if malignant, 1 if benign
Adhesion - marginal adhesion
BNuclei - bare nuclei
Chromat - bland chromatin
Epithel - epithelial cell size
Mitoses - mitoses
NNucleo - normal nucleoli
ClThick - clump thickness
UShape - cell shape uniformity
UCSize - cell size uniformity

Exploratory Data Analysis

Quick Look

First we will conduct EDA. Let's just see what the data looks like:

```
tumor <- read.table("brca.txt", header = T)
attach(tumor)
head(tumor)
```

```
##   Class Adhesion BNuclei Chromat Epithel Mitoses NNucleo ClThick UShape UCSize
## 1     1        5      10       3       7       1       2       5      4      4
## 2     1        1       2       3       2       1       1       3      1      1
## 3     1        1       4       3       3       1       7       6      8      8
## 4     1        3       1       3       2       1       1       4      1      1
## 5     0        8      10       9       7       1       7       8     10     10
## 6     1        1      10       3       2       1       1       1      1      1
```

We can see the 10 different features in the data set, with Class as the primary column.

NA Values

Now let's see if we have any missing values in our data:

```
cat("We have", sum(is.na(tumor)), "NA Value(s)")
```

```
## We have 0 NA Value(s)
```

```
apply(tumor, 2, function(x) any(is.na(x)))
```

```
##    Class Adhesion  BNuclei  Chromat  Epithel  Mitoses  NNucleo  ClThick
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
##   UShape   UCSize
##    FALSE    FALSE
```

Using two different methods, we've identified no NA values, so we will not have to impute them or ignore an entire feature altogether thankfully.

Summaries

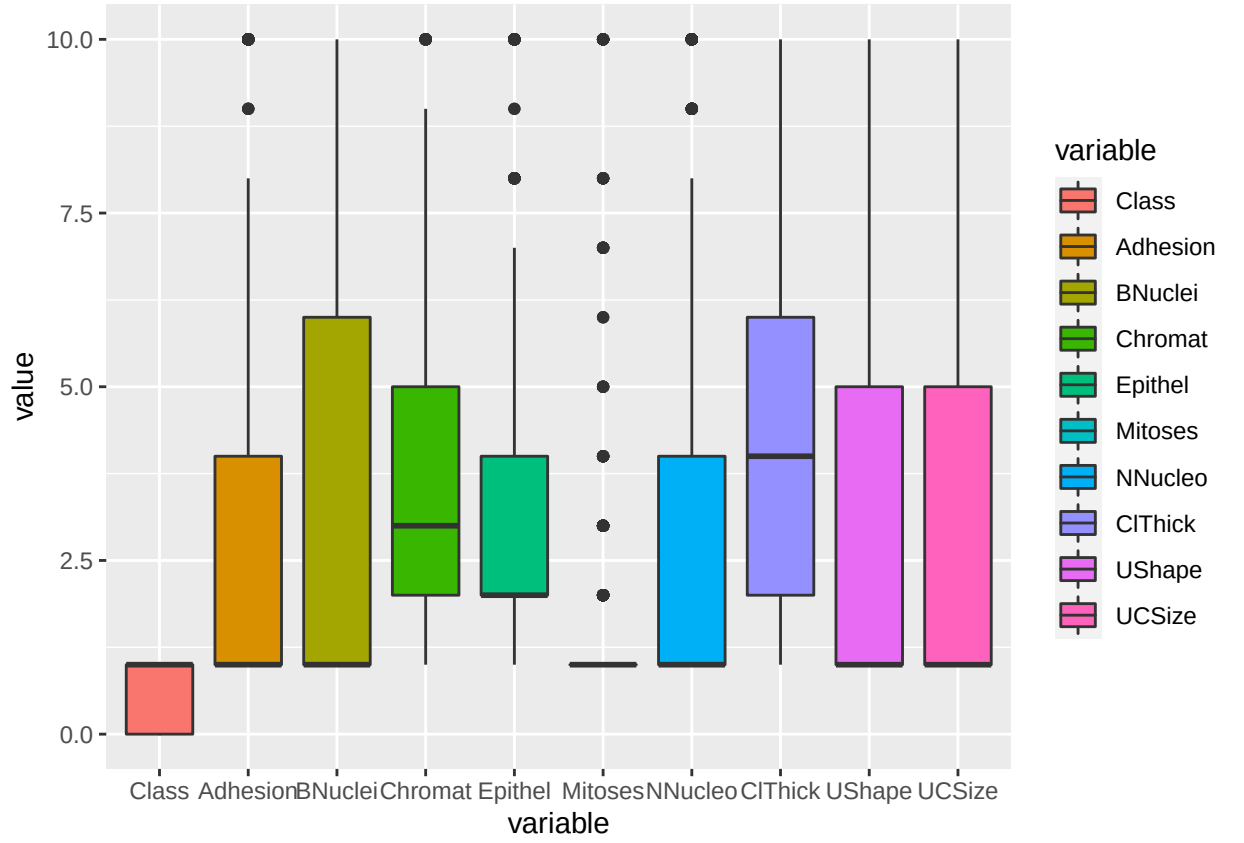
Now we will summarize the data

```
summary(tumor)
```

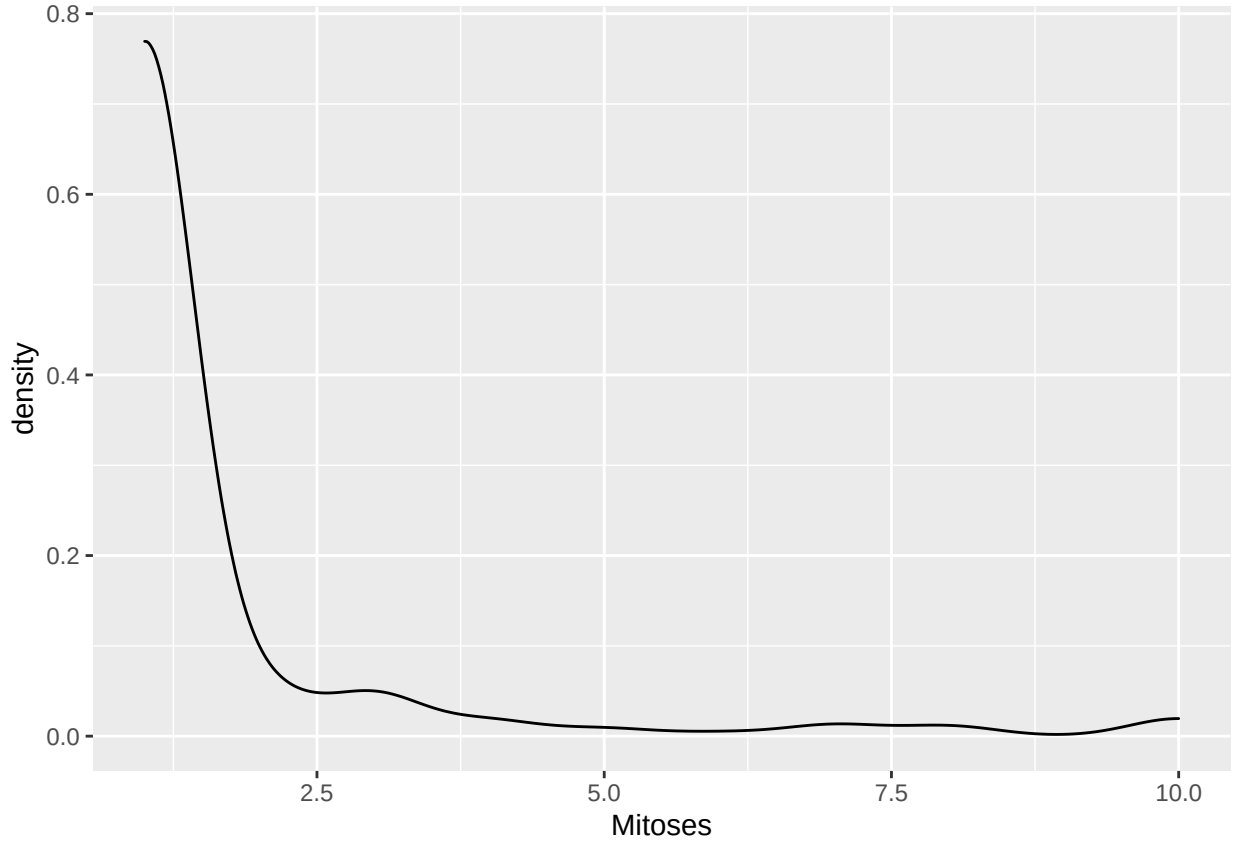
```
##      Class           Adhesion        BNuclei          Chromat
##  Min.   :0.0000   Min.   : 1.00   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:0.0000   1st Qu.: 1.00   1st Qu.: 1.000   1st Qu.: 2.000
##  Median :1.0000   Median : 1.00   Median : 1.000   Median : 3.000
##  Mean   :0.6487   Mean   : 2.84   Mean   : 3.544   Mean   : 3.435
##  3rd Qu.:1.0000   3rd Qu.: 4.00   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :1.0000   Max.   :10.00   Max.   :10.000   Max.   :10.000
##     Epithel          Mitoses          NNucleo          ClThick
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 2.000   Median : 1.000   Median : 1.000   Median : 4.000
##  Mean   : 3.235   Mean   : 1.613   Mean   : 2.864   Mean   : 4.439
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.: 4.000   3rd Qu.: 6.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##      UShape           UCSize
##  Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.000   Median : 1.000
##  Mean   : 3.206   Mean   : 3.149
##  3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000
```

```
ggplot(data = melt(tumor), aes(x=variable, y=value)) + geom_boxplot(aes(fill=variable))
```

```
## No id variables; using all as measure variables
```




```
ggplot(data = tumor, aes(x = Mitoses)) +
  geom_density()
```



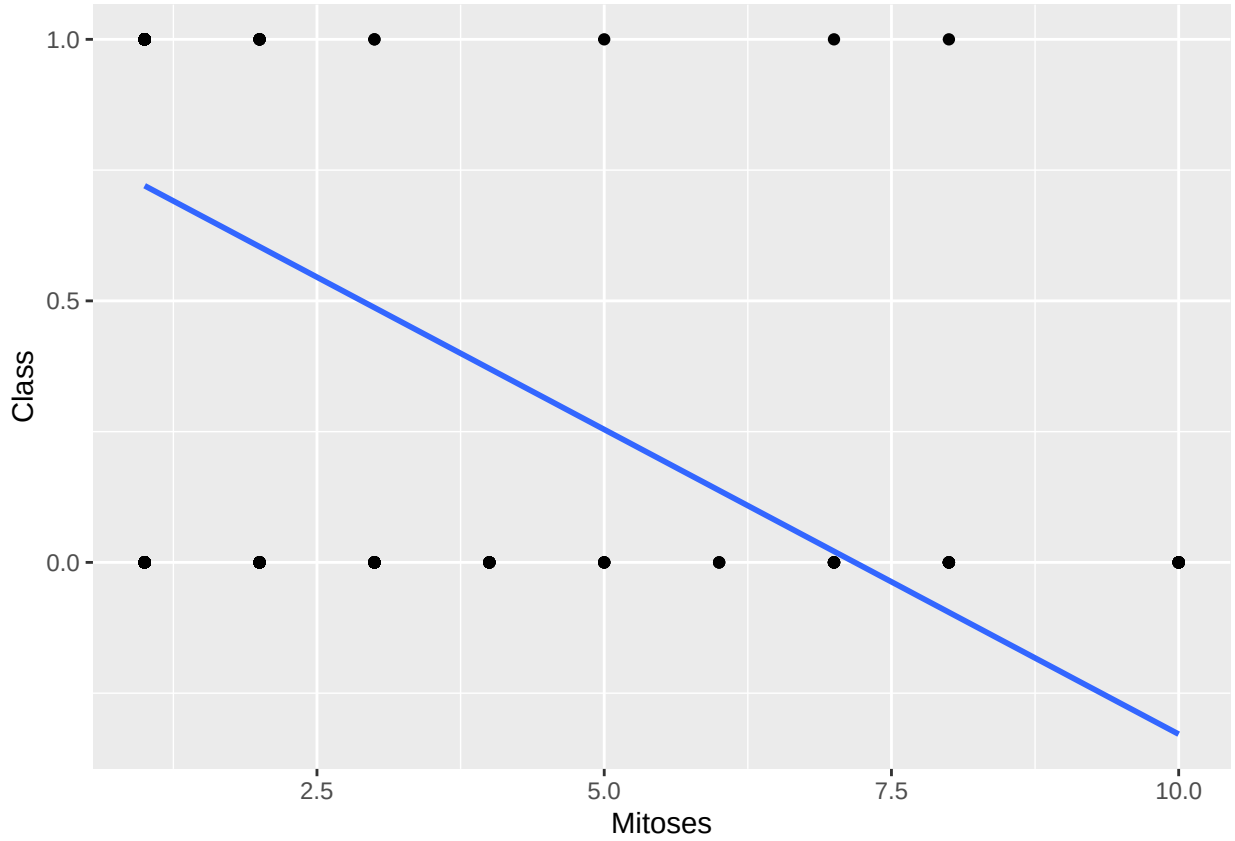
From our summary we can see that all variables max out at 10. The vast majority of cells only have 1 round of mitosis (80% of them almost). This could perhaps either indicate that mitosis is a weak indicator of non-cancerous cells or an extremely strong predictor of cancerous cells. We can actually explore this further by looking at the correlation between mitoses and class.

```
mit <- lm(Class~Mitoses)
sumary(mit)
```

```
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.8366279  0.0227518  36.772 < 2.2e-16
## Mitoses     -0.1165005  0.0095678 -12.176 < 2.2e-16
##
## n = 669, p = 2, Residual SE = 0.43243, R-Squared = 0.18
```

```
ggplot(tumor, aes(Mitoses, Class)) +
  geom_point() +
  geom_smooth(method='lm', se=F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We can see from this quick regression that Mitoses is a significant predictor for Class, with more mitoses indicating higher likelihood of Class=0, or malignancy. The presence of reduced SE with fewer mitoses indicates that the feature is better at predicting benign tumors than malignant ones. I hypothesize that after feature selection, Mitoses will still be included.

Additionally, from the summary statistics we can see that UShape and UCSize have nearly identical distributions, which may indicate a level of linear dependence.

```
anova(lm(Class ~ . - UShape, data = tumor), lm(Class~., data = tumor))
```

```
## Analysis of Variance Table
##
## Model 1: Class ~ (Adhesion + BNuclei + Chromat + Epithel + Mitoses + NNucleo +
##     ClThick + UShape + UCSize) - UShape
## Model 2: Class ~ Adhesion + BNuclei + Chromat + Epithel + Mitoses + NNucleo +
##     ClThick + UShape + UCSize
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    660 23.115
## 2    659 22.924  1   0.19024 5.4688 0.01966 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-test indicates that UShape is still somewhat significant with a p value of 2%, so it may be included in the final model selection.

Building our model

After analyzing the data a bit to get a better grasp of it, we'll start by creating our model and looking at the Residual and Null Deviance to see if logistic regression is even a good tool to use.

```
t.glm <- glm(formula = Class ~ ., family = binomial, tumor)
summary(t.glm)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial, data = tumor)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.47083  -0.01285   0.05098   0.10020   3.05419
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 10.863734   1.375500   7.898 2.83e-15 ***
## Adhesion    -0.345357   0.135955  -2.540 0.011078 *
## BNuclei     -0.437654   0.107416  -4.074 4.61e-05 ***
## Chromat     -0.501644   0.194420  -2.580 0.009874 **
## Epithel     -0.066161   0.168754  -0.392 0.695018
## Mitoses     -0.611383   0.368608  -1.659 0.097191 .
## NNucleo     -0.274353   0.128403  -2.137 0.032626 *
## ClThick     -0.589115   0.159843  -3.686 0.000228 ***
## UShape      -0.317574   0.266439  -1.192 0.233292
## UCSize      -0.006077   0.247662  -0.025 0.980424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 867.331  on 668  degrees of freedom
## Residual deviance:  86.187  on 659  degrees of freedom
## AIC: 106.19
##
## Number of Fisher Scoring iterations: 8
```

```
cpval <- round(pchisq(t.glm$null.deviance - t.glm$deviance, df=t.glm$df.null - t.glm$df.residual,
                      lower.tail=FALSE),5)
cat("After our Chi Squared Test between Residual and Null Deviance, our P-value is",
    cpval)
```

```
## After our Chi Squared Test between Residual and Null Deviance, our P-value is 0
```

Since our P-value is 0, we can safely conclude that the model is more useful than simply the intercept column in predicting the effect.

Choosing our Model

Let's use step selection along with the AIC score to see which features we should include, and which ones we shouldn't

```
AIC_b <- step(t.glm, scope=list(lower= ~ BNuclei,
                                upper=~ Adhesion + BNuclei + Chromat + Epithel + Mitoses
                               + NNucleo + ClThick + UShape + UCSize),
              direction="both", data=tumor)
```

```
## Start:  AIC=106.19
## Class ~ Adhesion + BNuclei + Chromat + Epithel + Mitoses + NNucleo +
##     ClThick + UShape + UCSize
##
##             Df Deviance    AIC
## - UCSize     1   86.187 104.19
## - Epithel    1   86.338 104.34
## - UShape     1   87.543 105.54
## <none>           86.187 106.19
## - Mitoses    1   89.740 107.74
## - NNucleo    1   91.208 109.21
## - Adhesion   1   92.800 110.80
## - Chromat    1   93.457 111.46
## - ClThick    1  103.740 121.74
##
## Step:  AIC=104.19
## Class ~ Adhesion + BNuclei + Chromat + Epithel + Mitoses + NNucleo +
##     ClThick + UShape
##
##             Df Deviance    AIC
## - Epithel    1   86.341 102.34
## <none>           86.187 104.19
## - UShape     1   89.132 105.13
## - Mitoses    1   89.818 105.82
## + UCSize     1   86.187 106.19
## - NNucleo    1   91.370 107.37
## - Adhesion   1   93.339 109.34
## - Chromat    1   94.116 110.12
## - ClThick    1  104.250 120.25
##
## Step:  AIC=102.34
## Class ~ Adhesion + BNuclei + Chromat + Mitoses + NNucleo + ClThick +
##     UShape
##
##             Df Deviance    AIC
## <none>           86.341 102.34
## - UShape     1   89.831 103.83
## - Mitoses    1   89.991 103.99
## + Epithel    1   86.187 104.19
## + UCSize     1   86.338 104.34
## - NNucleo    1   91.995 106.00
## - Adhesion   1   94.347 108.35
## - Chromat    1   94.788 108.79
## - ClThick    1  104.473 118.47
```

It appears that our final model has an AIC of 102.34, slightly better than our previous, full model with and AIC of 106.2. The model is: Class ~ Adhesion + BNuclei + Chromat + Mitoses + NNucleo + ClThick + UShape As hypothesized by exploratory data analysis, Mitoses is included, and only one of UShape and UCSize is present.

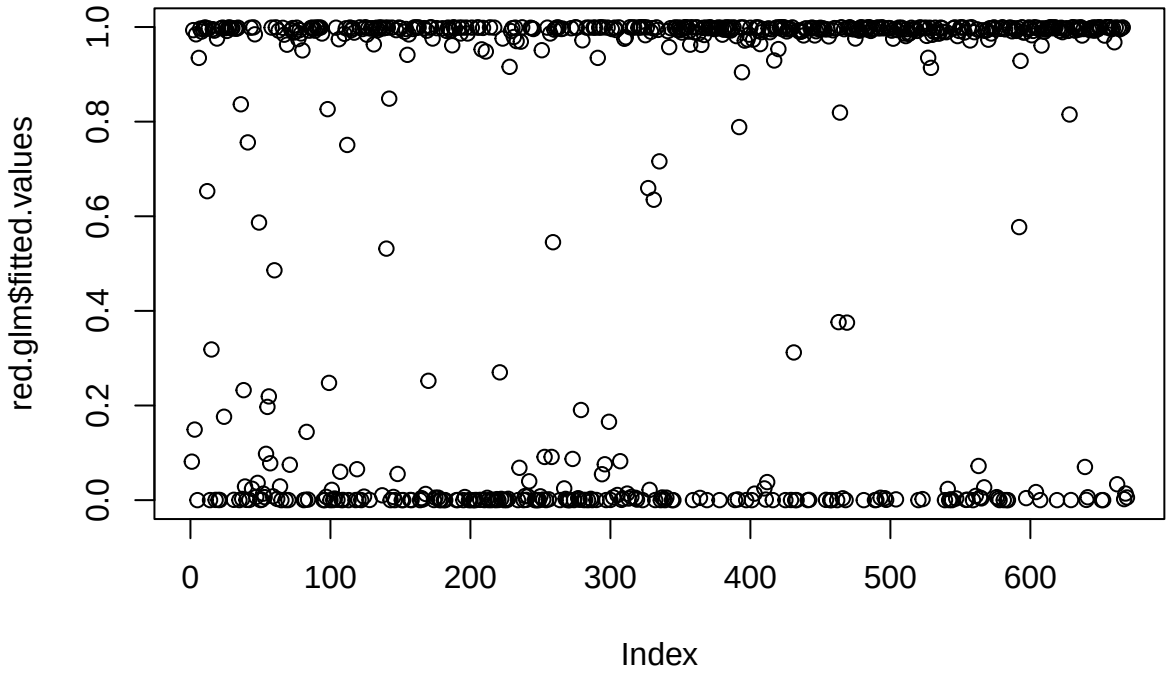
Using the best model to find a cutoff

Since we are using binary logistic regression, we must determine an optimal cutoff probability to minimize the number of false positives and false negatives. We can start by arbitrarily choosing two cutoffs, assessing them, and then using more formal methods (like ROC analysis).

.5 cutoff

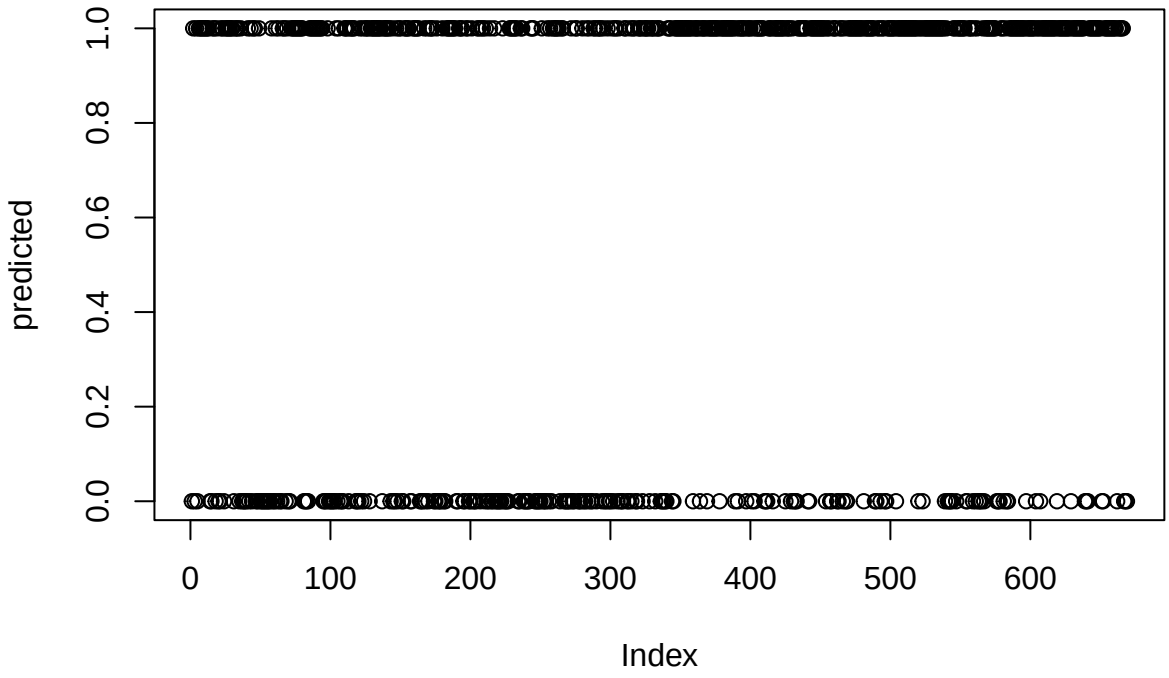
Let p < 5 be benign cells and p >= 5 be malignant cells. Let's see how the reduced model works to

```
red.glm <- glm(Class ~ Adhesion + BNuclei + Chromat + Mitoses +
                    NNucleo + ClThick + UShape, family = binomial, tumor)
plot(red.glm$fitted.values)
```




```
#plot misclassified values

predicted <- red.glm$fitted.values
for(i in 1:length(predicted)) {
  if(predicted[i] < .5) predicted[i] <- 0
  else predicted[i] <- 1
}
plot(predicted)
```




```
confusionMatrix(data=as.factor(predicted), reference = as.factor(tumor$Class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 225   8
##          1  10 426
##
##                Accuracy : 0.9731
##                  95% CI : (0.9578, 0.984)
##     No Information Rate : 0.6487
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9408
##
##  Mcnemar's Test P-Value : 0.8137
##
##             Sensitivity : 0.9574
##             Specificity : 0.9816
##          Pos Pred Value : 0.9657
##          Neg Pred Value : 0.9771
##              Prevalence : 0.3513
##          Detection Rate : 0.3363
##    Detection Prevalence : 0.3483
```

```
##       Balanced Accuracy : 0.9695
##
##        'Positive' Class : 0
##
```

We can see that our accuracy score is .97 if p = .5.

.9 cutoff

```
red.glm <- glm(Class ~ Adhesion + BNuclei + Chromat + Mitoses +
                    NNucleo + ClThick + UShape, family = binomial, tumor)


predicted <- red.glm$fitted.values
for(i in 1:length(predicted)) {
  if(predicted[i] < .9) predicted[i] <- 0
  else predicted[i] <- 1
}
confusionMatrix(data=as.factor(predicted), reference = as.factor(tumor$Class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 234  15
##          1   1 419
##
##                Accuracy : 0.9761
##                  95% CI : (0.9615, 0.9863)
##     No Information Rate : 0.6487
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9482
##
##  Mcnemar's Test P-Value : 0.001154
##
##             Sensitivity : 0.9957
##             Specificity : 0.9654
##          Pos Pred Value : 0.9398
##          Neg Pred Value : 0.9976
##              Prevalence : 0.3513
##          Detection Rate : 0.3498
##    Detection Prevalence : 0.3722
##       Balanced Accuracy : 0.9806
##
##        'Positive' Class : 0
##
```

Just doing a back-of-the-envelope calculation, it seems like the optimal cutoff may be somewhere between .5 and .9. Let's apply a more rigorous method of finding our optimal cutoff using a training/testing split.

Performing a logistic regression by splitting data

Here we will split the data into a testing and training set. The training set (2/3) will be used to select the model, and the testing set will be used to select the appropriate cutoff by minimizing the sum of errors, and to evaluate the performance of our model.

```
######Create two datasets######
testing.data <- as.data.frame(matrix(nrow = 0, ncol= ncol(tumor)))
training.data <- as.data.frame(matrix(nrow = 0, ncol = ncol(tumor)))
colnames(testing.data) <- colnames(tumor); colnames(training.data) <- colnames(tumor)


for(i in 1:length(tumor$Class)) {
  if(i %% 3 != 0) training.data[nrow(training.data) + 1,] = tumor[i,]
  else {
    testing.data[nrow(testing.data) + 1,] = tumor[i,]
  }
}

###Find model###
train.start <- glm(formula = Class~., family = binomial, data = training.data)
train.glm <- step(train.start, scope=list(upper=~ Adhesion + BNuclei + Chromat +
                                            Epithel + Mitoses
                                          + NNucleo + ClThick + UShape + UCSize),
              direction="both", data=training.data)
```

```
## Start:  AIC=83.87
## Class ~ Adhesion + BNuclei + Chromat + Epithel + Mitoses + NNucleo +
##     ClThick + UShape + UCSize
##
##            Df Deviance     AIC
## - Epithel   1   64.151  82.151
## - Adhesion  1   64.276  82.276
## - UShape    1   64.825  82.825
## - Chromat   1   65.062  83.062
## - UCSize    1   65.094  83.094
## - NNucleo   1   65.257  83.257
## <none>          63.867  83.867
## - Mitoses   1   66.120  84.120
## - ClThick   1   70.917  88.917
## - BNuclei   1   84.243 102.243
##
## Step:  AIC=82.15
## Class ~ Adhesion + BNuclei + Chromat + Mitoses + NNucleo + ClThick +
##     UShape + UCSize
##
##            Df Deviance     AIC
## - Adhesion  1   64.568  80.568
## - UShape    1   64.942  80.942
## - UCSize    1   65.233  81.233
## - Chromat   1   65.342  81.342
## - NNucleo   1   65.392  81.392
## <none>          64.151  82.151
## - Mitoses   1   66.394  82.394
```

```
## + Epithel   1   63.867  83.867
## - ClThick   1   71.814  87.814
## - BNuclei   1   84.331 100.331
##
## Step:  AIC=80.57
## Class ~ BNuclei + Chromat + Mitoses + NNucleo + ClThick + UShape +
##     UCSize
##
##            Df Deviance     AIC
## - UShape    1   65.367  79.367
## - Chromat   1   65.677  79.677
## - NNucleo   1   65.817  79.817
## - UCSize    1   66.326  80.326
## <none>          64.568  80.568
## - Mitoses   1   66.899  80.899
## + Adhesion  1   64.151  82.151
## + Epithel   1   64.276  82.276
## - ClThick   1   71.814  85.814
## - BNuclei   1   86.884 100.884
##
## Step:  AIC=79.37
## Class ~ BNuclei + Chromat + Mitoses + NNucleo + ClThick + UCSize
##
##            Df Deviance     AIC
## - Chromat   1   66.645  78.645
## - NNucleo   1   67.051  79.051
## <none>          65.367  79.367
## - Mitoses   1   67.445  79.445
## + UShape    1   64.568  80.568
## + Adhesion  1   64.942  80.942
## + Epithel   1   65.246  81.246
## - UCSize    1   72.221  84.221
## - ClThick   1   74.177  86.177
## - BNuclei   1   94.042 106.042
##
## Step:  AIC=78.65
## Class ~ BNuclei + Mitoses + NNucleo + ClThick + UCSize
##
##            Df Deviance     AIC
## - Mitoses   1   68.442  78.442
## <none>          66.645  78.645
## - NNucleo   1   68.679  78.679
## + Chromat   1   65.367  79.367
## + UShape    1   65.677  79.677
## + Adhesion  1   66.344  80.344
## + Epithel   1   66.541  80.541
## - ClThick   1   75.607  85.607
## - UCSize    1   82.909  92.909
## - BNuclei   1  107.574 117.574
##
## Step:  AIC=78.44
## Class ~ BNuclei + NNucleo + ClThick + UCSize
##
##            Df Deviance     AIC
```

```
## <none>           68.442  78.442
## - NNucleo   1   70.504  78.504
## + Mitoses   1   66.645  78.645
## + Chromat   1   67.445  79.445
## + UShape    1   67.821  79.821
## + Adhesion  1   68.043  80.043
## + Epithel   1   68.321  80.321
## - ClThick   1   81.664  89.664
## - UCSize    1   86.967  94.967
## - BNuclei   1  110.321 118.321
```

Using our training data and step selection, our model is: Class ~ BNuclei + NNucleo + ClThick + UCSize Step: AIC=78.44 Now we test our data and find the optimal cutoff

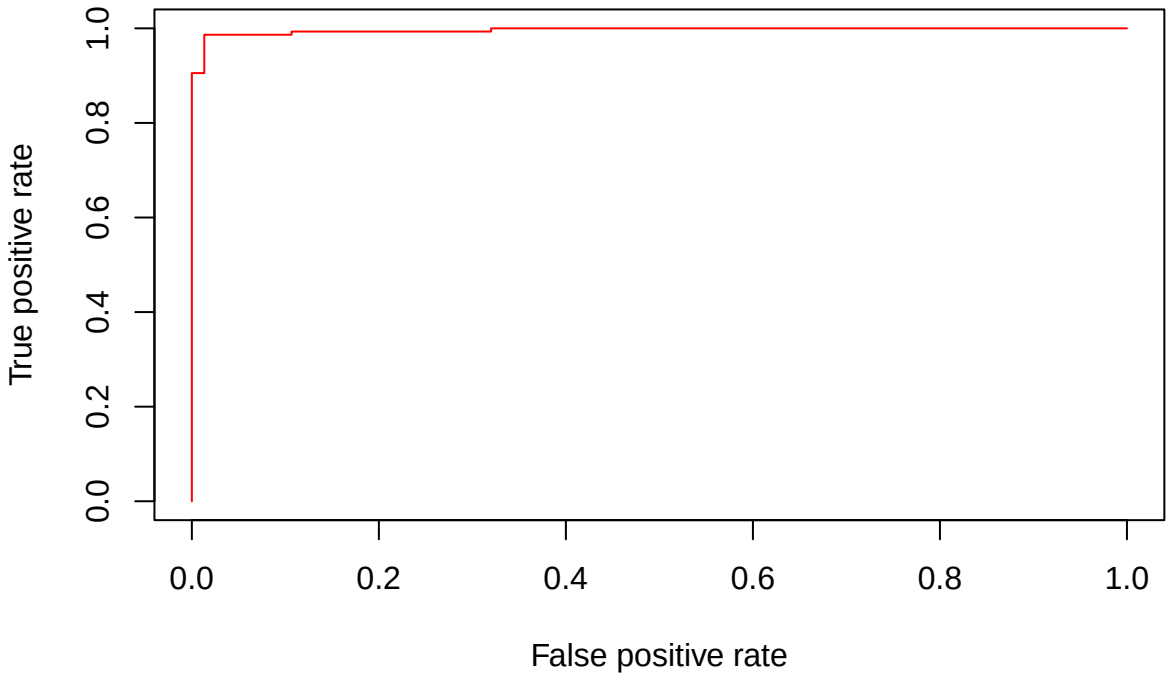
```
glm.testing <- glm(formula = Class ~ BNuclei + NNucleo + ClThick + UCSize, family = binomial, data = tes

fittedvals <- as.vector(glm.testing$fitted.values)

roc.1 <- ROCR::prediction(predictions = fittedvals, labels = testing.data$Class)
roc2 <- ROCR::performance(roc.1, measure = "tpr", x.measure = "fpr")
roc2
```

```
## A performance instance
##   'False positive rate' vs. 'True positive rate' (alpha: 'Cutoff')
##   with 114 data points
```

```
plot(roc2, col=rainbow(10))
```




```
#######Test Model######
##Create Dataframe##
pos.cut <-seq(0,1,by=.05)
pos.cut.df <- as.data.frame(matrix(nrow = length(pos.cut), ncol= 2))
colnames(pos.cut.df) <- c("Possible Cutoff", "Sum of Errors")
pos.cut.df["Possible Cutoff"] <- pos.cut
pos.cut.df$`Sum of Errors` <- rep(0, length(pos.cut.df$`Sum of Errors`))


error.matrix <- as.data.frame(cbind(testing.data$Class, fittedvals))
colnames(error.matrix)[1] = "Class"

#weighted fp & fn because a false negative is worse
fpw <- 1
fnw <- 1

for (j in 1:length(pos.cut)) {
  false.neg <- 0
  false.pos <- 0
  benign_prop = pos.cut[j]
  for (i in 1:length(error.matrix$Class)) {
    if ((error.matrix$Class[i] == 0) && (error.matrix$fittedvals[i] > benign_prop)) false.neg = false.n
    if ((error.matrix$Class[i] == 1) && (error.matrix$fittedvals[i] < benign_prop)) false.pos = false.p
  }

  pos.cut.df[j,2] = (fnw*false.neg + fpw*false.pos)
```

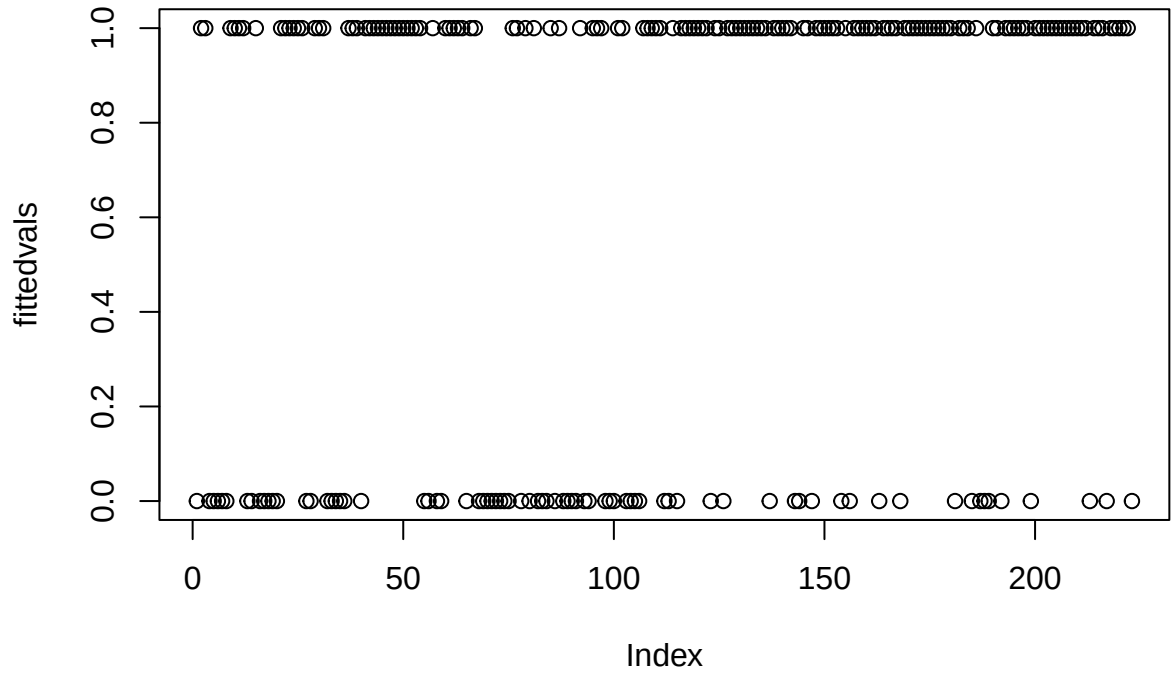
```
}
##test##
cat("The cut-off should be: ")
```

```
## The cut-off should be:
```

```
pos.cut.df$`Possible Cutoff`[which.min(pos.cut.df$`Sum of Errors`)]
```

```
## [1] 0.6
```

```
for(i in 1:length(fittedvals)) {
  if(fittedvals[i] < .6) fittedvals[i] <- 0
  else fittedvals[i] <- 1
}
plot(fittedvals)
```




```
confusionMatrix(data=as.factor(fittedvals), reference = as.factor(testing.data$Class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  74   2
```

```
##          1   1 146
##
##                Accuracy : 0.9865
##                  95% CI : (0.9612, 0.9972)
##     No Information Rate : 0.6637
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.97
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9867
##             Specificity : 0.9865
##          Pos Pred Value : 0.9737
##          Neg Pred Value : 0.9932
##              Prevalence : 0.3363
##          Detection Rate : 0.3318
##    Detection Prevalence : 0.3408
##       Balanced Accuracy : 0.9866
##
##        'Positive' Class : 0
##
```

We have found that our optimal cutoff is at p = .6. Using this probability, our model does extremely well: we have 1 Type I error and 2 Type II errors, giving us nearly an AUC ~ 1.

Final Thoughts

Let's put our model to the test by seeing how well it predicts a new value. Given the following sample:
Adhesion = 1, BNuclei=1,
Chromat=3,
Epithel=2,
Mitoses=1,
NNucleo=1,
ClThick=4,
UShape=1,
UCSize=1
Will it be malignant or not?

```
sample <- data.frame(Adhesion = 1,
                     BNuclei=1,
                     Chromat=3,
                     Epithel=2,
                     Mitoses=1,
                     NNucleo=1,
                     ClThick=4,
                     UShape=1,
                     UCSize=1)

pred <- predict(glm.testing, newdata= sample, se.fit = T,type = "link")

pred.low <- pred$fit - 1.96*pred$se
pred.high <- pred$fit + 1.96*pred$se
print(c(ilogit(pred.low), ilogit(pred.high)))
```

```
##         1         1
## 0.9543232 0.9984506
```

Give our 95% confidence interval is around .976 (well above the .6 cutoff), we can say with high certainty that this sample is benign. We could include some PCA plots to show where, depending on the features with the most variance, this sample would fall and see if it is grouped with the other benign samples based on the profile to confirm further the power of our model. We could have also done additional EDA on the other features and used them to hypothesize which class this sample would fall into before running the model. All these additions would only add to our understanding of our data and our model.