Kelvyn Bladen

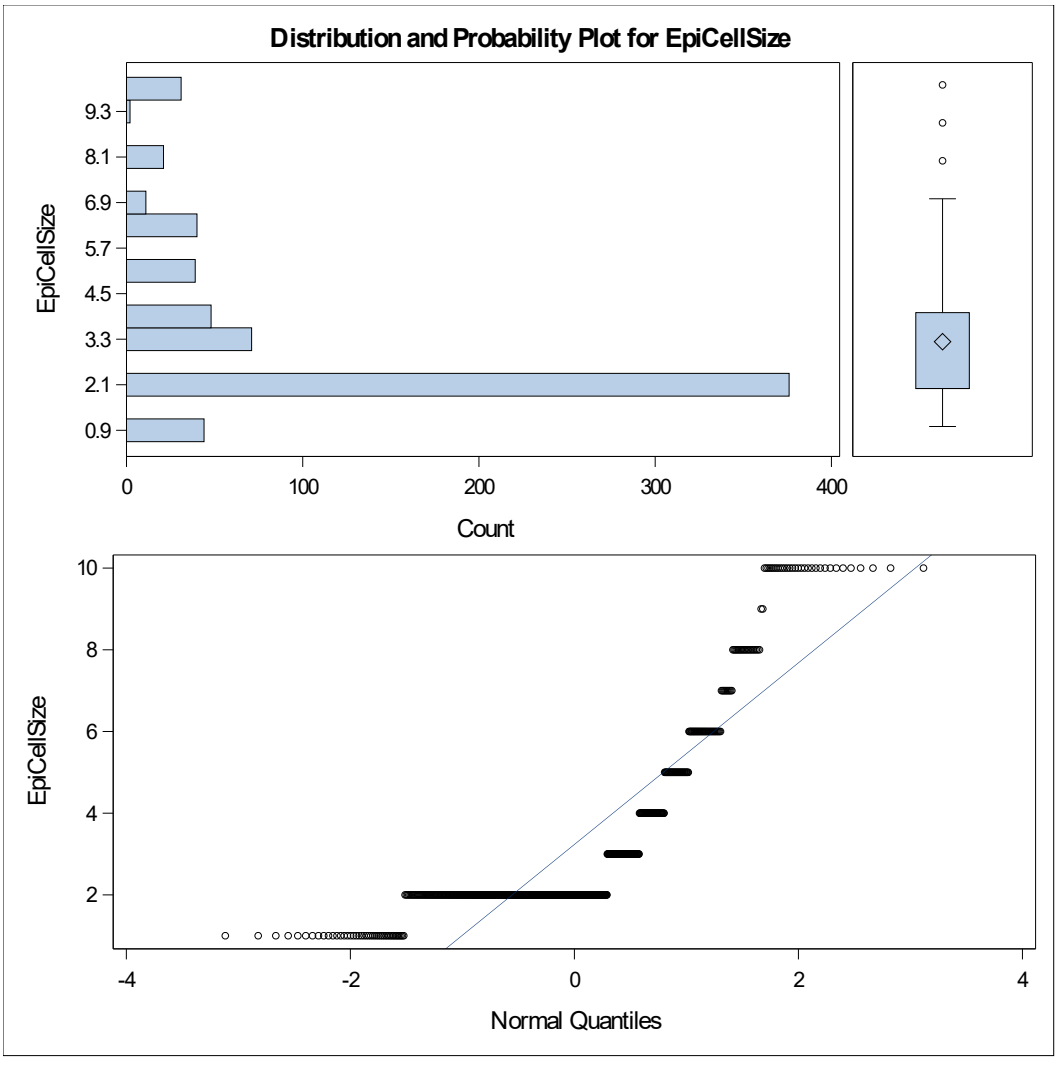
Stats 5810 – Dr. Cutler

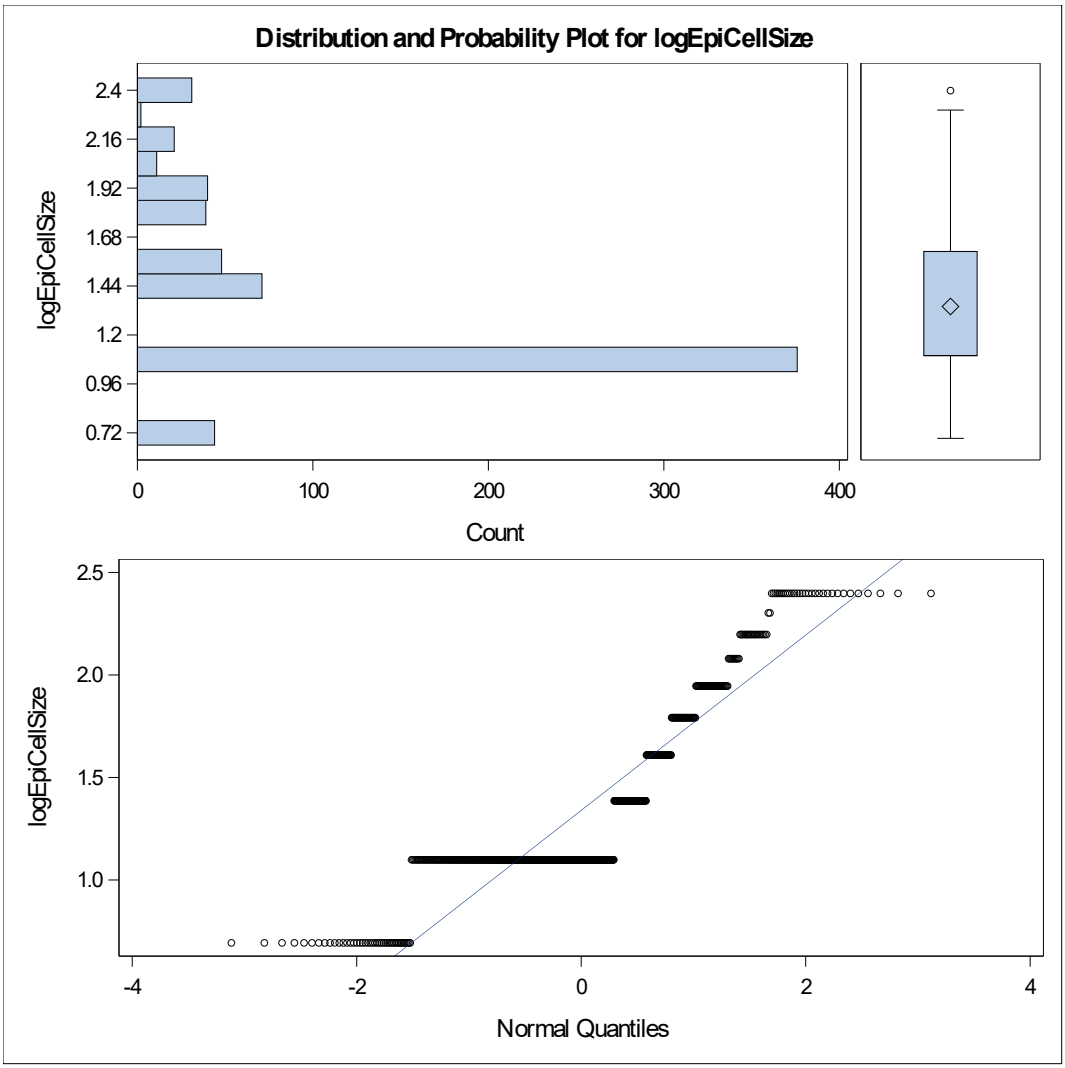
Project Report

Data Diagnostics

The data set that I choose for this project comes from the Wisconsin Breast Cancer sets in the UC Irvine Data Repository. The response variable is cancer type, coded as 0 for benign and 1 for malignant. There were 699 observations, but 16 had some missing value. Therefore to simplify it for performing my analysis, I trimmed the set to 683 observations, 444 benign and 239 malignant, with complete information for each of the 9 predictor variables. These predictor variables are as follows: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses. Each of these cancer attributes was observed or measured and quantitatively described using a relative scale from 1 to 10. As one might expect, many of the predictor values have extreme values, especially the value of 1. Consequently, before doing predictive analysis, I ran some basic diagnostics to see the distribution of the predictor variables. By building some histograms and quantile plots, I found that almost all of the predictors were strongly skewed to the right. To adjust for this, I applied a log transformation to all of them, and rebuilt the plots. However, I encountered an interesting predicament in doing this, because there are only 10 available values for each predictor. Many of the histograms maintained similar distributions after the transformation, but they still become more normal in shape and the corresponding quantile plots noticeably improved. Below is an example of this for just one variable, EpiCellSize. It follows the pattern I just described, where the histograms merely shift a

bit, but we see that the box plot looks betters for the transformed variable and outliers are pulled in by the transformation. Consequently, I choose to use the log transformations for each skewed variable when running those classification methods that are enhanced by having normalized predictors. With regard to those methods, although I won't include the error rates for running them with the raw predictors, it is noteworthy that the transformed data usually outperformed the raw by a small, but observable amount of roughly .005 to .03.





Linear Discriminant Analysis

To begin my classification analysis on the Wisconsin Cancer data, I first performed a linear discriminant analysis on it in SAS using priors proportional for weighing the results according to the distribution of benign and malignant cancer types. As usual, the re-substitution error (.0249) slightly outperformed the cross-validated error rate (.0293). The resulting confusion matrix and corresponding error rate are show below for both the re-substitution and the cross-validation. It is immediately noticeable to me that these error rates are quite low, especially for a linear discriminant analysis.

Number of Observations and Percent Classified into Cancer by ReSub			
From Cancer	**0**	**1**	**Total**
0	435 97.97	9 2.03	444 100.00
1	8 3.35	231 96.65	239 100.00
Total	443 64.86	240 35.14	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer ReSub			
	0	**1**	**Total**
Rate	0.0203	0.0335	0.0249
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer by CV			
From Cancer	**0**	**1**	**Total**
0	435 97.97	9 2.03	444 100.00
1	11 4.60	228 95.40	239 100.00
Total	446 65.30	237 34.70	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer CV			
	0	1	Total
Rate	0.0203	0.0460	0.0293
Priors	0.6501	0.3499	

Consequently, I suspected that there are some very influential predictors that are splitting these two classes apart. Using the stepdisc SAS function to complete a selection of variables, I was surprised to find that none of them could be removed because each had a significant influence on the classification's effectiveness. The table below shows the retained variables and their p-values. From this, I gather that if I was to consider simplifying it, I would probably choose to remove Mitoses, EpiCellSize, and/or MargAdhesion. But for now, I will continue to use them since they were retained. It is my belief that many of the other methods will yield very low error rates as well. Although, this means that this data set might be a poor example for comparing different methods, there is a more important result to be realized here. The power to predict the type or strength of a cancerous growth is extremely valuable, and it is fantastic to see classification methods accomplishing this purpose with very low error rates.

Statistics for Removal, DF = 1, 673			
Variable	**Partial R-Square**	**F Value**	**Pr > F**
logClumpThick	0.0381	26.68	<.0001
logUniSize	0.0277	19.20	<.0001
logUniShape	0.0256	17.68	<.0001
logMargAdhesion	0.0047	3.16	0.0761
logEpiCellSize	0.0034	2.30	0.1296
logBareNuclei	0.2001	168.35	<.0001
logChromatin	0.0252	17.37	<.0001
logNormNucleoli	0.0221	15.19	0.0001
logMitoses	0.0033	2.19	0.1390

Quadratic Discriminant Analysis

In SAS, I ran a test to determine whether the covariance matrices could be treated as equal. The result of this test, shown below, was a chi-squared value of much less than .01. Consequently, SAS determined that a quadratic discriminant analysis would be more appropriate than a linear one.

Chi-Square	DF	Pr > ChiSq
1600.321444	45	<.0001

The confusion matrices and error rates (below) yielded error rate results that underperformed the output of the linear discriminant analysis by about a full percent. Again, the re-substitution did slightly better than the cross-validation for the quadradic discriminant analysis, just like it did for the linear. When comparing LDA and QDA for this data set, I'm interested in more than the total error rate. For both the re-substitution and cross-validation of the

linear discriminant analysis, the misclassifications are split evenly for both the benign and malignant types of cancer. However, for QDA, almost every misclassification is that of a benign being classified as malignant (25), rather than the other way around (1). In many instances, I suspect that this would be preferred because if some are to be wrong, we want them to be on the side of over-estimating the problem, rather than under-estimating. Consequently, I like the accuracy of LDA more, but in application, it would probably be wiser and more appropriate to use QDA.

Number of Observations and Percent Classified into Cancer by ReSub			
From Cancer	**0**	**1**	**Total**
0	420 94.59	24 5.41	444 100.00
1	0 0.00	239 100.00	239 100.00
Total	420 61.49	263 38.51	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer ReSub			
	0	**1**	**Total**
Rate	0.0541	0.0000	0.0351
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer by CV			
From Cancer	**0**	**1**	**Total**
0	419 94.37	25 5.63	444 100.00
1	1 0.42	238 99.58	239 100.00
Total	420 61.49	263 38.51	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer CV			
	0	**1**	**Total**
Rate	0.0563	0.0042	0.0381
Priors	0.6501	0.3499	

K – Nearest Neighbors Method

Next, I conducted a classification using K-Nearest Neighbor methods for values of k, 3 through 7. I found that for each of these, the misclassifications were distributed quite evenly, which is similar to the LDA sensitivity and specificity. The resulting cross-validation confusion matrices and their error rates are shown below. To summarize the output, I was surprised to find that the best error rate (.0249) occurred when k was set to be 6. In other data sets, it usually is found that lower values such as k=4 are the best. The resulting error rate of 6 k-nearest neighbors, is a slight improvement on LDA by about .005 and QDA by roughly .015. Impressively, the range of the errors for the K-nearest neighbor tests was a mere .0088, where k=3 had the worst error of .0337. The differences between them could quite possibly be subtle

enough to be white noise. Personally, I would suspect that for predicting onto a test set, using k-values of 4, 5, or 6 would be just fine, considering that these 3 values yielded the best errors and have been consistently more effective for other data sets.

Number of Observations and Percent Classified into Cancer k=3			
From Cancer	**0**	**1**	**Total**
0	433 97.52	11 2.48	444 100.00
1	12 5.02	227 94.98	239 100.00
Total	445 65.15	238 34.85	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer k=3			
	0	**1**	**Total**
Rate	0.0248	0.0502	0.0337
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer k=4			
From Cancer	**0**	**1**	**Total**
0	434 97.75	10 2.25	444 100.00
1	9 3.77	230 96.23	239 100.00
Total	443 64.86	240 35.14	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer k=4			
	0	**1**	**Total**
Rate	0.0225	0.0377	0.0278
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer k=5			
From Cancer	**0**	**1**	**Total**
0	434 97.75	10 2.25	444 100.00
1	9 3.77	230 96.23	239 100.00
Total	443 64.86	240 35.14	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer k=5			
	0	1	Total
Rate	0.0225	0.0377	0.0278
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer k=6			
From Cancer	0	1	Total
0	435 97.97	9 2.03	444 100.00
1	8 3.35	231 96.65	239 100.00
Total	443 64.86	240 35.14	683 100.00
Priors	0.65007	0.34993	

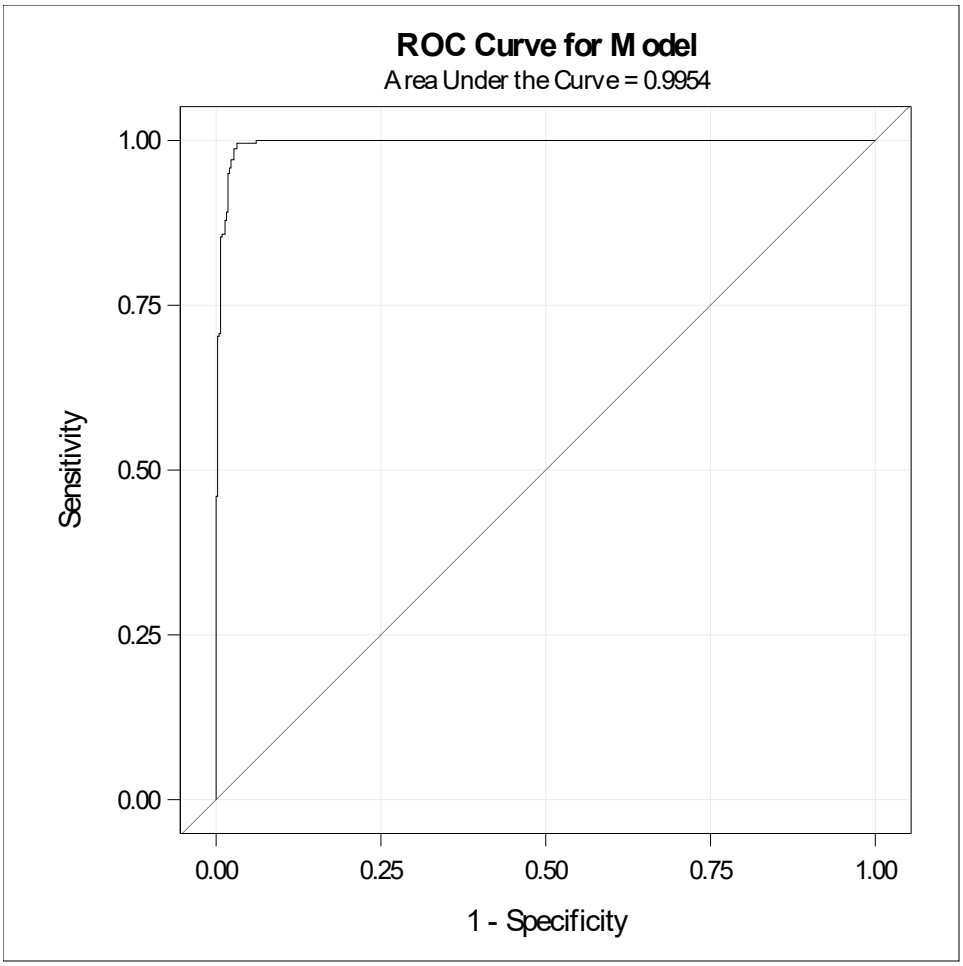
Error Count Estimates for Cancer k=6			
	0	1	Total
Rate	0.0203	0.0335	0.0249
Priors	0.6501	0.3499	

Number of Observations and Percent Classified into Cancer k=7			
From Cancer	**0**	**1**	**Total**
0	433 97.52	11 2.48	444 100.00
1	9 3.77	230 96.23	239 100.00
Total	442 64.71	241 35.29	683 100.00
Priors	0.65007	0.34993	

Error Count Estimates for Cancer k=7			
	0	**1**	**Total**
Rate	0.0248	0.0377	0.0293
Priors	0.6501	0.3499	

Logistic Regression

Following the k-nearest neighbor method, I employed logistic regression on my dataset. The generated ROC curve below looks truly fantastic. I don't think I've seen a better one in any previous projects or examples. The classification table is also very impressive compared to most others. Evaluating the error at the standard c=.5, I found a total error of .032. This is not quite as good as LDA or k-NN, but I think logistic regression could be extremely useful to this problem. By using different values of c, we can alter the specificities and sensitivities to reduce the number of missed malignant cancer growths. This would benefit medical performance greatly, and it appears to have a better error rate than QDA and comparable to our other methods (c=.3, error = .026).



Classification Table									
	Correct		Incorrect		Percentages				
Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensi-tivity	Speci-ficity	False POS	False NEG
0.300	233	432	12	6	97.4	97.5	97.3	4.9	1.4
0.500	229	432	12	10	96.8	95.8	97.3	5.0	2.3
0.700	222	435	9	17	96.2	92.9	98.0	3.9	3.8

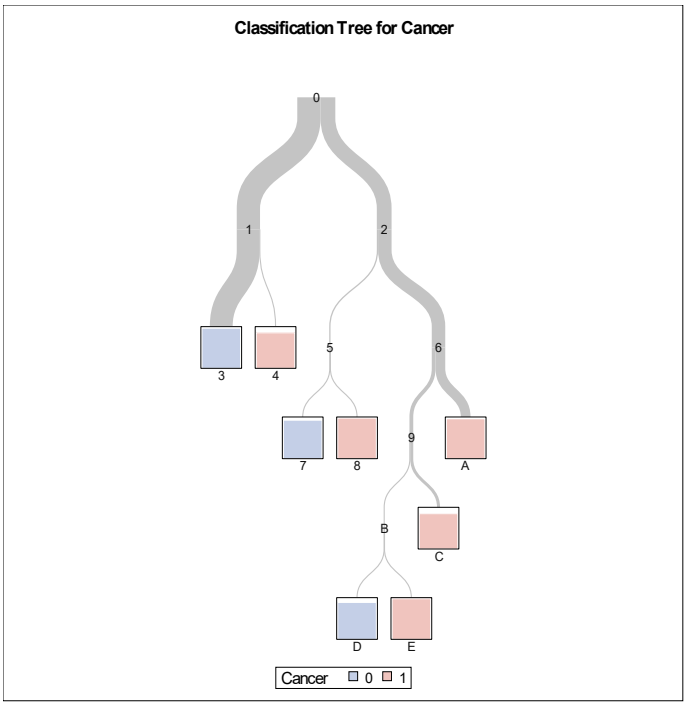
Using a selection of variables, this time EpiCellSize, UniSize, and NormNucleoli were removed. The new ROC curve looked almost identical to the curve above, with an area of .9955. As we can see from the table below, variable selection actually improved the error rates slightly

for the lower c-values. Consequently, I suspect that some of those removed variables were simply noise that had almost no value to the logistic regression.

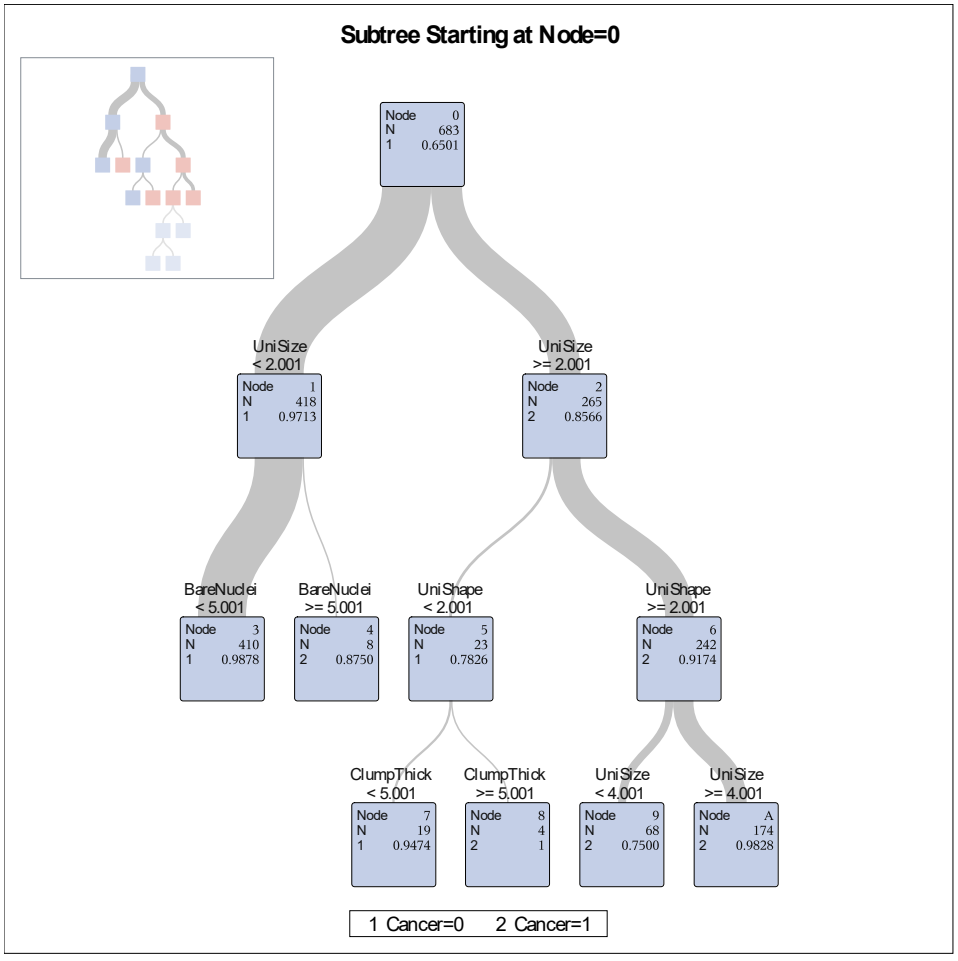
Classification Table									
	Correct		Incorrect		Percentages				
Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensi-tivity	Speci-ficity	False POS	False NEG
0.300	234	432	12	5	97.5	97.9	97.3	4.9	1.1
0.500	230	432	12	9	96.9	96.2	97.3	5.0	2.0
0.700	220	435	9	19	95.9	92.1	98.0	3.9	4.2

Classification and Regression Trees

A Classification and Regression Tree is possibly the best method for understanding and interpreting. By running a test to find how many terminal nodes I should use, I found that a 14 node tree was the comparison for the 1-SE rule. Possible sizes of 3, 5, 8, and 9 all met the criteria of the 1-SE rule. By running each of these, I found that a 3-node tree was oversimplified, while 5 and 9 node trees each had a split where benign was the majority at both nodes of the split. The 8-node tree is the best in my opinion. It has no such problem at any of it's splits, it's not obviously over-simplified, and it's still relatively easy to understand. The tree and confusion matrix below indicate that the nodes appear to be extremely pure. However, the error rate for this method comes out to be .0483. This is the worst error of any of the methods so far. However, the tree does help to see which variables are most important and where they are splitting on. This information could certainly help medical staff to have a quick idea of whether a cancer growth is malignant by checking just a few characteristics, such as UniShape > 2, BareNuclei & ClumpThick > 5, and a high magnitude for UniSize.



Confusion Matrices				
		Predicted		Error
	Actual	0	1	Rate
Model Based	**0**	433	11	0.0248
	1	7	232	0.0293
Cross Validation	**0**	427	17	0.0383
	1	16	223	0.0669



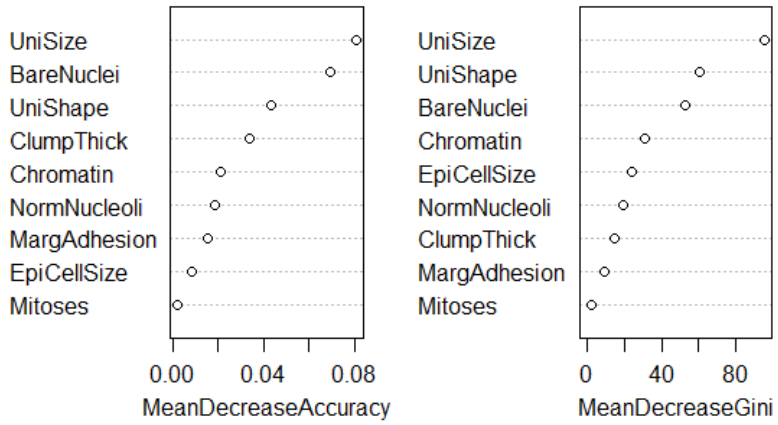
Random Forest

Switching to R, I ran my data through the random forest algorithm. From the confusion matrix and percent correctly classified, I can see this method has an error of .0279. This is better than almost all the cross-validated errors from previous methods, with the exception of the best k-nearest neighbors. I also like that it seems to favor misclassifying more benign growths, than malignant ones. It's quite effective and tends toward the implementation I'm looking for.

```
    0   1 class.error
0 432  12  0.02702703
1   7 232  0.02928870
```

```
PCC: 97.21014
```

I then used random forest to evaluate variable importance. The chart below suggests that the least important variables are Mitoses and EpiCellSize. This is very consistent with the stepdisc output, however the logistic selection is quite contradicting. Perhaps, It may be that UniSize is quite collinear with other variables, resulting in it's removal earlier. I'm inclined to suspect that random forest's variable importance is more reliable though, especially with stepdisc's selection method and CART's variables to split on validating it.



With this plot in mind, I tried a few variations of my original random forest where I removed a few of the less important variables. By eliminating Mitoses and EpiCellSize from the classification, I was able to get an error of .0323 from the matrix below. This seems to imply that these 2 variables are mostly noise and have little influence on improving our classification.

```
    0   1 class.error
0 432  12  0.02702703
1  10 229  0.04184100
```

```
PCC: 96.76916
```

Ada Boosting

My next method to try was Ada Boosting. I can clearly see from this matrix, that the initial error for this method is the best yet (.0117). Unfortunately, this is likely over-predicting, because of the following matrix which represents the accuracy when doing cross-validation on Ada Boosting. This error of .0307 is right in the normal range of errors I've seen and started to expect. Thus, I think ada-boosting does fine, but is probably not the best for this problem unless it can be proven to be just as accurate on a test dataset without doing any cross-validation.

```
          Final Prediction
True value   0   1
         0 437   7
         1   1 238
```

```
PCC: 98.8287
```

```
Cross-Validated
      0   1
  0 433  11
  1  10 229
```

```
PCC: 96.9252
```

Gradient Boosting Machines

Gradient Boosting Machines are often a very accurate form of classification if they can be tuned properly. To preform this analysis, I first obtained the out-of-bag error rates for the method and for when it is cross-validated. The resulting cross-validation error of .0351 is not particularly great when compared to some of the others which have been between .03 and .025.

```
"Percent Correctly Classified = " "97.22"
 "Specificity = "                 "97.75"
 "Sensitivity = "                 "96.23"
```

```
Cross-Validated
"Percent Correctly Classified = " "96.49"
 "Specificity = "                 "97.52"
 "Sensitivity = "                 "94.56"
```

So using tuning procedures, I found the optimal shrinkage (.1), number of trees (25), and interaction depth (16). Implementing these criteria back into the gradient boosting machine algorithm, yielded the following improved confusion matrices for the raw and cross-validated analysis. Both of these error rates outperform their corresponding errors from above, and the cross-validated error of .0307 shows that this method is yielding similar results to the others. Perhaps gradient boosting machines aren't the best for this dataset, for some reason. While they often are very accurate, the results were merely average for classifying cancer types.

```
     0   1
  0 434  10
  1   3 236
```

```
"Percent Correctly Classified = " "98.1"
 "Specificity = "                 "97.75"
 "Sensitivity = "                 "98.74"
```

```
 Cross-validated
     0   1
  0 431  13
  1   8 231


"Percent Correctly Classified = " "96.93"
 "Specificity = "                 "97.07"
 "Sensitivity = "                 "96.65"
```

Support Vector Machines

My last method to try was a Support Vector Machine. Surprisingly, it's cross-validated error is only .0293, which is better than the final results from both ada-boosting and gradient boosting. The initial output also doesn't appear to over-predict as much, having a more reasonable error of .0249, rather than .019 (GBM) and .012 (ADA). It doesn't seem to be quite as accurate as random forest and some of the K-NN methods have been. However, I'd say it's fairly effective for the job and worth running regardless in case it does out-perform other methods.

```
      0   1
  0 433  11
  1   6 233


"Percent Correctly Classified = " "97.51"
 "Specificity = "                 "97.52"
 "Sensitivity = "                 "97.49"


Cross-validated
     0   1
  0 431  13
  1   7 232


"Percent Correctly Classified = " "97.07"
 "Specificity = "                 "97.07"
 "Sensitivity = "                 "97.07"
```

Conclusion

The goal of this project has been to correctly classify cancerous growths as benign or malignant. After running each classification method, I observed that the method with the best out-of-bag re-substitution error was Ada-Boosting (.0117) and the best cross-validated error was using K-Nearest Neighbors where k was set to 6 (.0249). It may seem tempting to conclude that these are the best methods for prediction and therefore primarily use them for predicting onto un-known observations. However, I don't believe that this is the case after evaluating each method. The fantastic error from Ada-Boosting is likely over-fit and would require a great deal of testing to prove otherwise. K-NN methods did a good job of reducing the total number of misclassifications, but it could well be random chance that this method came out on top considering how close all of the error rates were. I actually believe other methods would have more value in their application.

While the Classification Tree is not quite as accurate, it is very easy to understand and apply for medical professionals that may not be able to collect data for each of the predictor variables. The Random Forest method is also extremely valuable for this purpose. It has an error rate (.0323) superior to the Classification Tree, but since most other methods are performing just as well or better, it's worth is actually found in the way it does variable selection. From the Random Forest method, I'm able to rank the predictor variables from most to least important as follows: Uniformity of Cell Size, Bare Nuclei, Uniformity of Cell Shape, Clump Thickness, Bland Chromatin, Normal Nucleoli, Marginal Adhesion, Single Epithelial Cell Size and finally Mitoses. With this information, other classification methods can be simplified by removing some of the least influential predictors. It can also help those collecting data or evaluating patients to know which predictors to pay more attention.

Finally, if I had to recommend one tunable method for minimizing the classification error rate, it would be Logistic Regression. Usually, this is interpreted for when the parameter c is .5. However, as I mentioned with regard to QDA and Logistic, it could be in the best interest of cancer patients to focus on improving the sensitivity of the method. As we saw in the confusion table for logistic regression, using a lower value of c such as .3 yielded a total error rate of only .025 and a sensitivity error of .021. I suspect that by looking at even more values for c, an even better error could be found for both of these. With this level of total accuracy while emphasizing reducing the sensitivity error, Logistic Regression should be the best method for applying to future data sets where the type of cancer is un-known.