DS2 Final

M. de Ferrante, K. Maciejewski, P. Batten

April 26, 2018

Intro

For our analysis we used Breast Cancer data from the Wisconsin Breast Cancer Database, compiled by Dr. William Wolberg MD, in the early 1990s, found in the `mlbench` R package. The objective of this dataset is to to use characteristics of tumors such as cell size and texture to classify the tumors as either benign or malignant. The goals of this analysis are to identify the best learning method for classifying tumors.

Methods used include logistic regression, K-means clustering, Support Vector Machine, LDA, Random Forests, and KNN. While it will be helpful to identify one specific method for predicting future classifications of tumor, it is still advisable to consider the output from all methods, albeit with weighted considerations based on how effective the models are. Ultimately, while machine learning is a very powerful tool for making highly important medical classifications, user discretion should be taken into account when considering interpretability, such as potentially using a simpler model that has nearly the same power as a more complex model. Since our research area is highly regulated and in the clinical setting, we chose to use an untouched testing set to evaluate the performance of our methods. We also examine the kappa statistic as a measure of prediction accuracy since our outcome is unbalanced.

Exploratory data analysis

```
### Summary stuff for DS2 final project

library(mlbench)
data(BreastCancer)
attach(BreastCancer)
library(dplyr)
BreastCancer <- BreastCancer[,-1] # remove ID column
summary(BreastCancer) # note that everything is factor
```

```
##   Cl.thickness   Cell.size     Cell.shape  Marg.adhesion  Epith.c.size
##  1      :145   1      :384   1      :353   1      :407   2      :386
##  5      :130   10     : 67   2      : 59   2      : 58   3      : 72
##  3      :108   3      : 52   10     : 58   3      : 58   4      : 48
##  4      : 80   2      : 45   3      : 56   10     : 55   1      : 47
##  10     : 69   4      : 40   4      : 44   4      : 33   6      : 41
##  2      : 50   5      : 30   5      : 34   8      : 25   5      : 39
##  (Other):117   (Other): 81   (Other): 95   (Other): 63   (Other): 66
##   Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses          Class
##  1      :402   2      :166   1      :443     1      :579   benign   :458
##  10     :132   3      :165   10     : 61     2      : 35   malignant:241
##  2      : 30   1      :152   3      : 44     3      : 33
##  5      : 30   7      : 73   2      : 36     10     : 14
##  3      : 28   4      : 40   8      : 24     4      : 12
##  (Other): 61   5      : 34   6      : 22     7      :  9
##  NA's   : 16   (Other): 69   (Other): 69     (Other): 17
```

Above, we load in the data and remove the ID column as it is not needed.

Each variable except Class is loaded as 11 numerical factors with values ranging from 0 through 10. Class is benign or malignant, and this is the variable of interest. There are 16 missing values in bare nuclei, as seen in the summary above.

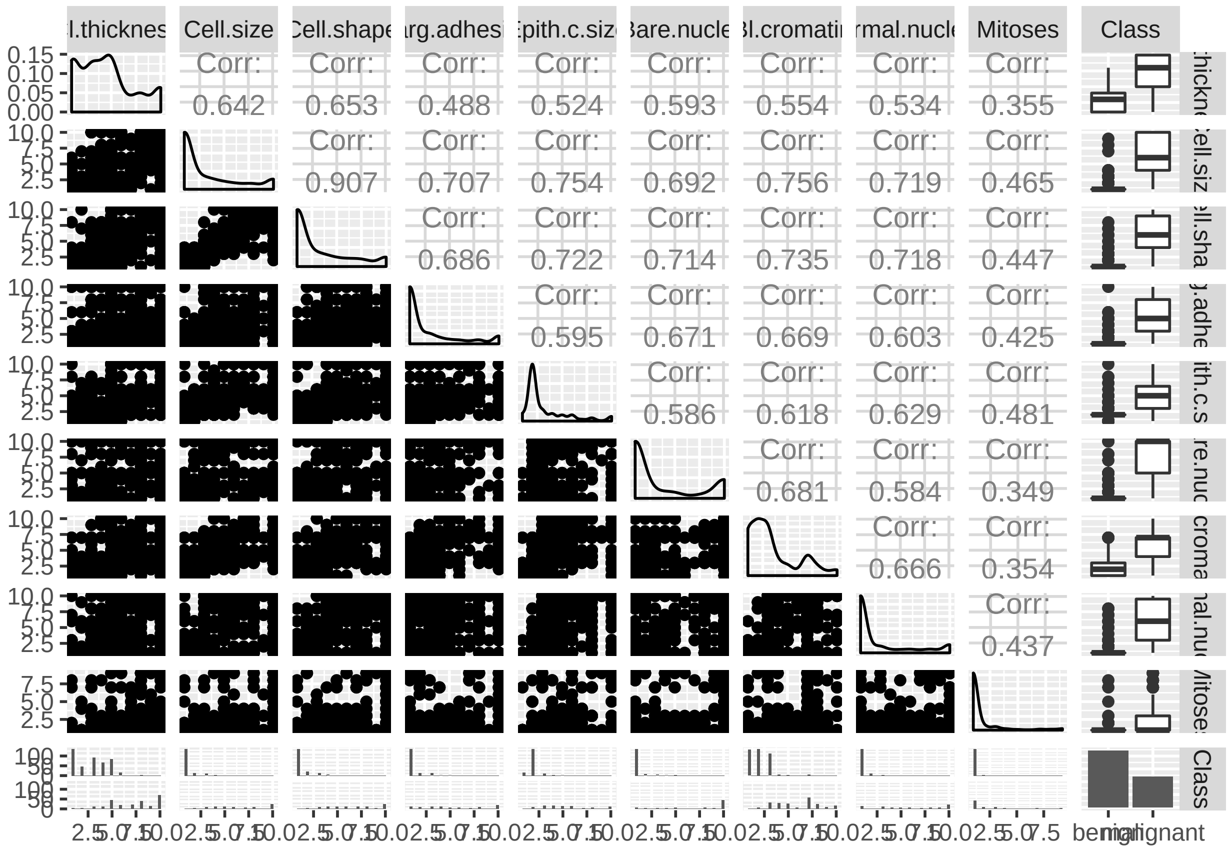
The variables included in the dataset are Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses, and Class. There are 458 subjects with benign growths and 241 with malignant.

Below, we convert the factors into numeric values and remove the NA's. Since there are few missing values compared to the number in the dataset, removing them should not effect our overall analysis power.

```
BreastCancer = BreastCancer %>%
  mutate(Cl.thickness=as.numeric(Cl.thickness)) %>%
  mutate(Cell.size=as.numeric(Cell.size)) %>%
  mutate(Cell.shape=as.numeric(Cell.shape)) %>%
  mutate(Marg.adhesion=as.numeric(Marg.adhesion)) %>%
  mutate(Epith.c.size=as.numeric(Epith.c.size)) %>%
  mutate(Bare.nuclei=as.numeric(Bare.nuclei)) %>%
  mutate(Bl.cromatin=as.numeric(Bl.cromatin)) %>%
  mutate(Normal.nucleoli=as.numeric(Normal.nucleoli)) %>%
  mutate(Mitoses=as.numeric(Mitoses)) %>% na.omit()
```

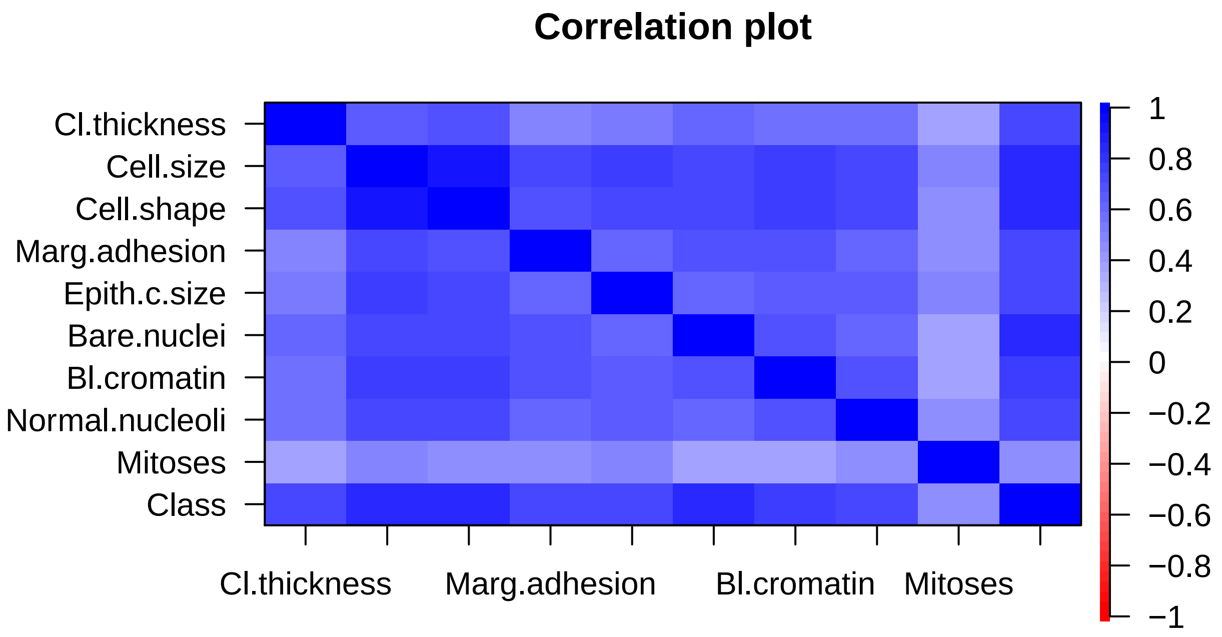
Summary plots

```
library(GGally)
ggpairs(BreastCancer) # all but response are numeric
```



The correlation plot below does not give much information. As expected, most measures have correlation with the outcome. Mitoses, however, has almost no corellation.

```r
library(psych)
BreastCancer_num <- BreastCancer %>%
  mutate(Class = as.numeric(Class)-1) # numeric response
cor.plot(BreastCancer_num[,])
```

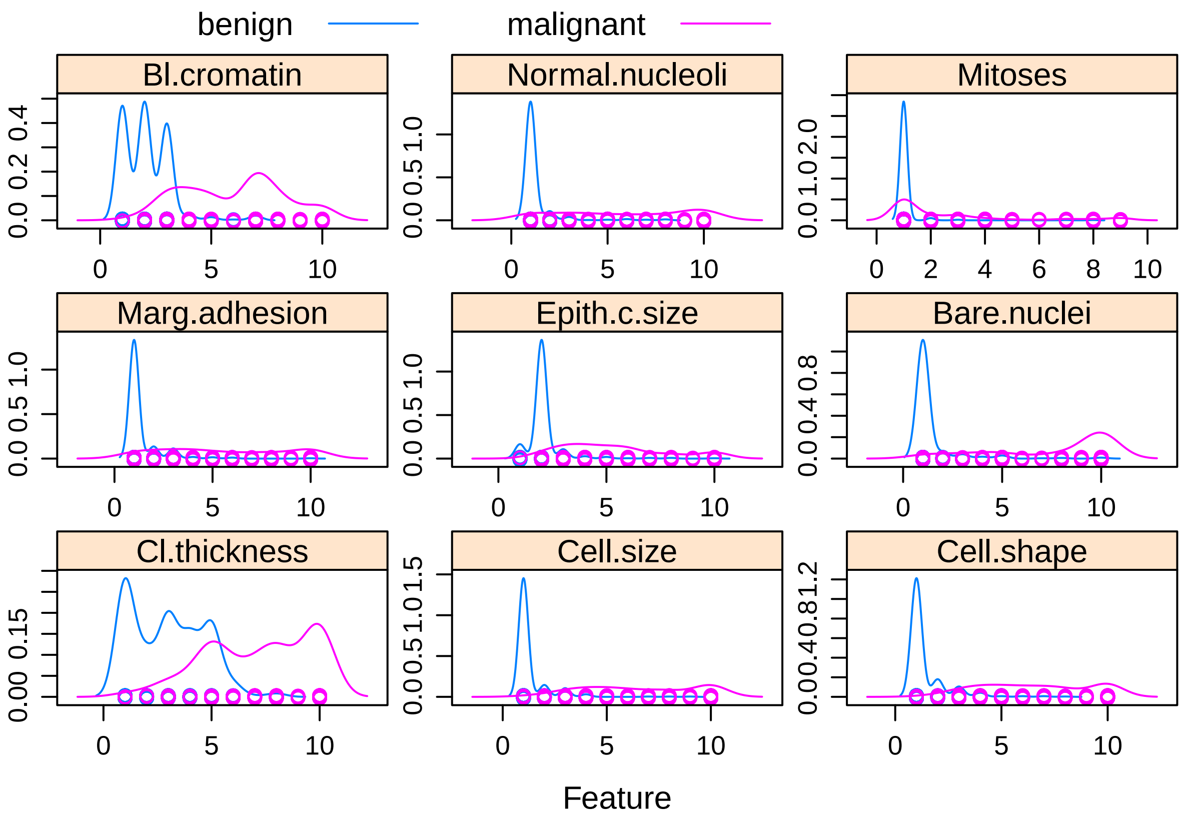


Below are the density plots for the variables. The blue line signifies benign and pink are malignant. We see that there are few malignant subjects with the following: normal nuclei, mitoses, marginal adhesion, single epithelial cell size, uniformity of cell size, uniformity of cell shape.

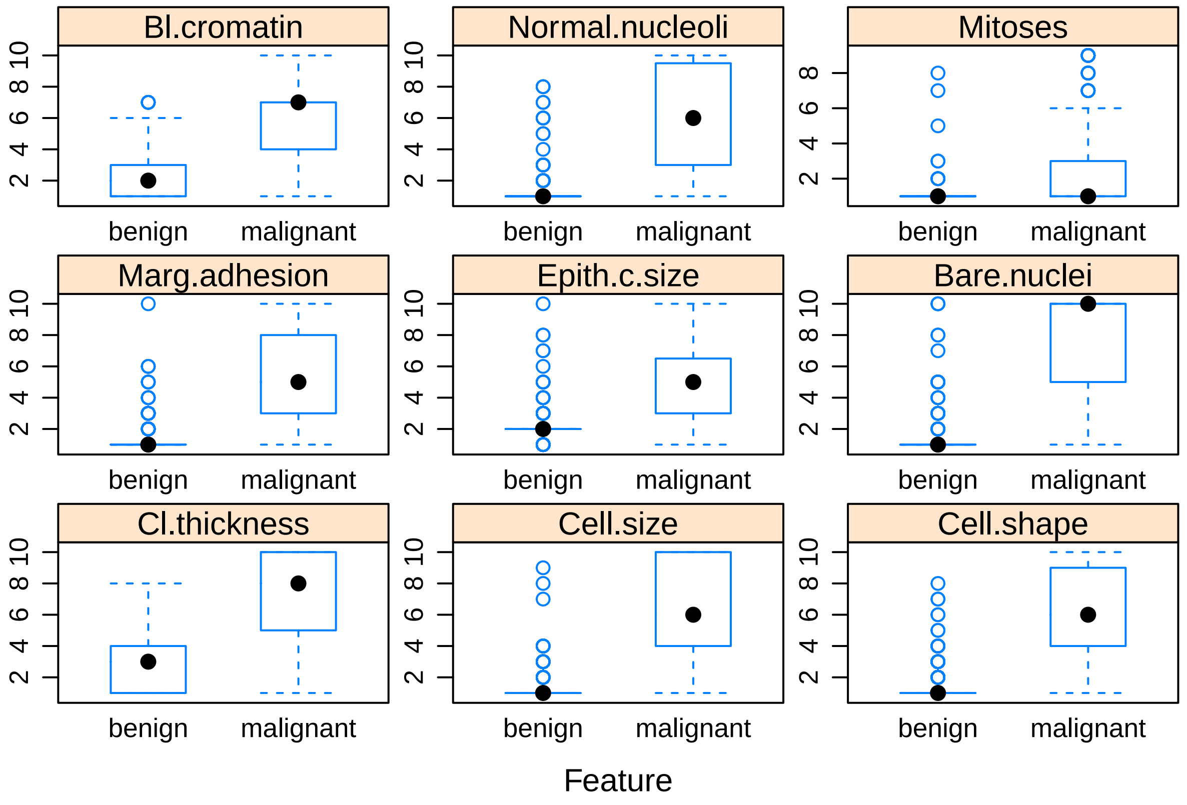
There are differences in the density plots of the following varaiables: bland chromatin, bare nucleoli, clump thickness.

The box plots below also show differences in the distributions for bland chromatin, bare nuclei, clump thickness.

```r
library(caret)
featurePlot(x=BreastCancer[,-10], y=BreastCancer[,10],
            plot="density",
            scales=list(x=list(relation="free"),
                        y=list(relation="free")),
            auto.key=list(columns=3),
            layout=c(3,3))
```



```r
featurePlot(x=BreastCancer[,-10], y=BreastCancer[,10],
            plot="box",
            scales=list(x=list(relation="free"),
                        y=list(relation="free")),
            auto.key=list(columns=3),
            layout=c(3,3))
```



Supervised Analyses

Logistic Analysis

Here is a generalized linear model with all variables included.

```
glm1 = glm(Class ~., data=BreastCancer,family=binomial)

summary(glm1)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial, data = BreastCancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4855  -0.1152  -0.0619   0.0222   2.4702
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -10.110096   1.173774  -8.613  < 2e-16 ***
## Cl.thickness     0.535256   0.141938   3.771 0.000163 ***
## Cell.size       -0.005943   0.209158  -0.028 0.977332
## Cell.shape       0.322136   0.230644   1.397 0.162510
## Marg.adhesion    0.330694   0.123462   2.679 0.007395 **
## Epith.c.size     0.096797   0.156568   0.618 0.536415
## Bare.nuclei      0.383015   0.093865   4.080 4.49e-05 ***
## Bl.cromatin       0.447401   0.171392   2.610 0.009044 **
## Normal.nucleoli   0.213074   0.112894   1.887 0.059109 .
## Mitoses           0.538551   0.325615   1.654 0.098138 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 884.35  on 682  degrees of freedom
## Residual deviance: 102.90  on 673  degrees of freedom
## AIC: 122.9
##
## Number of Fisher Scoring iterations: 8
```

At $\alpha = 0.05$ the following appear significant :

- Cl.thickness
- Marg.adhesion
- Bare.nuclei
- Bl.cromatin

We rerun the generalized linear model with only the significant values:

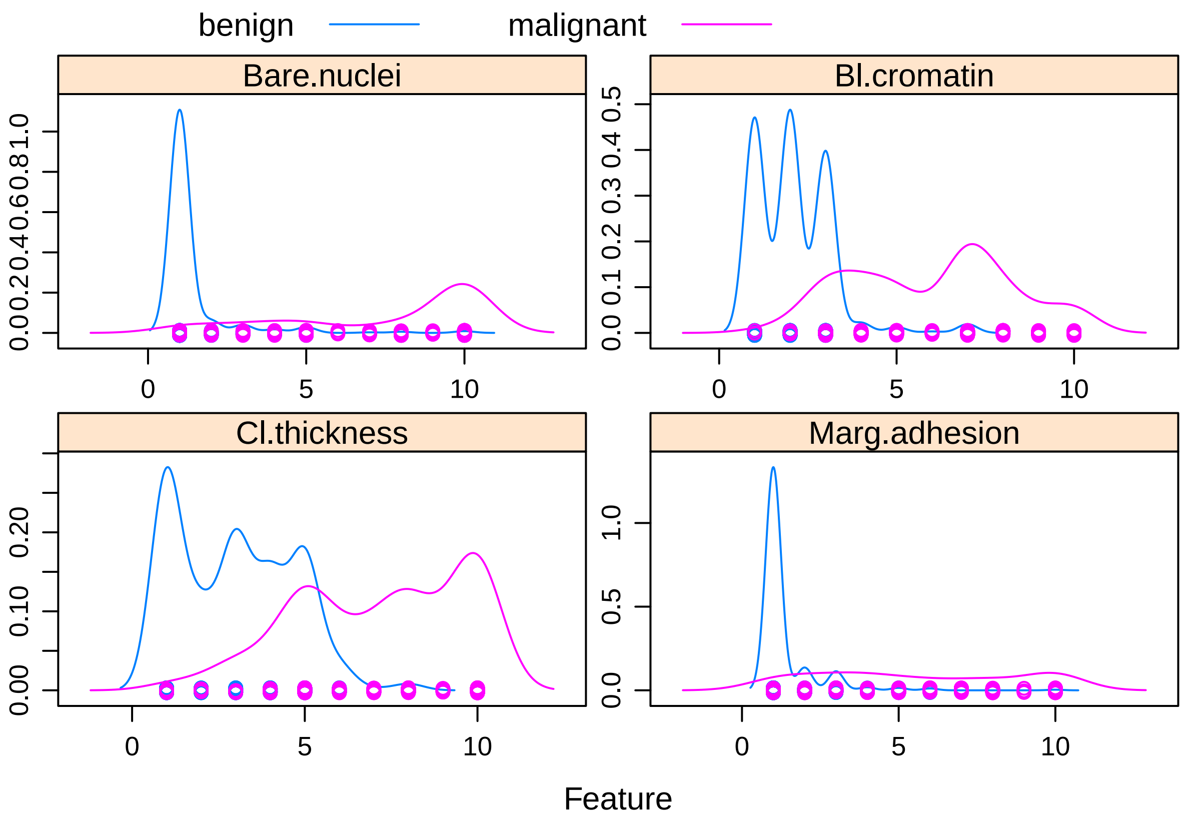
```
glm2 = glm(Class ~ Cl.thickness + Marg.adhesion +
             Bare.nuclei + Bl.cromatin,
           data=BreastCancer,family=binomial)

summary(glm2)
```

```
##
## Call:
## glm(formula = Class ~ Cl.thickness + Marg.adhesion + Bare.nuclei +
##     Bl.cromatin, family = binomial, data = BreastCancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6964  -0.1451  -0.0609   0.0232   2.4476
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -10.11370    1.03264  -9.794  < 2e-16 ***
## Cl.thickness    0.81166    0.12585   6.450 1.12e-10 ***
## Marg.adhesion   0.43412    0.11403   3.807 0.000141 ***
## Bare.nuclei     0.48136    0.08816   5.460 4.76e-08 ***
## Bl.cromatin     0.70154    0.15196   4.616 3.90e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 884.35  on 682  degrees of freedom
## Residual deviance: 125.77  on 678  degrees of freedom
## AIC: 135.77
##
## Number of Fisher Scoring iterations: 8
```

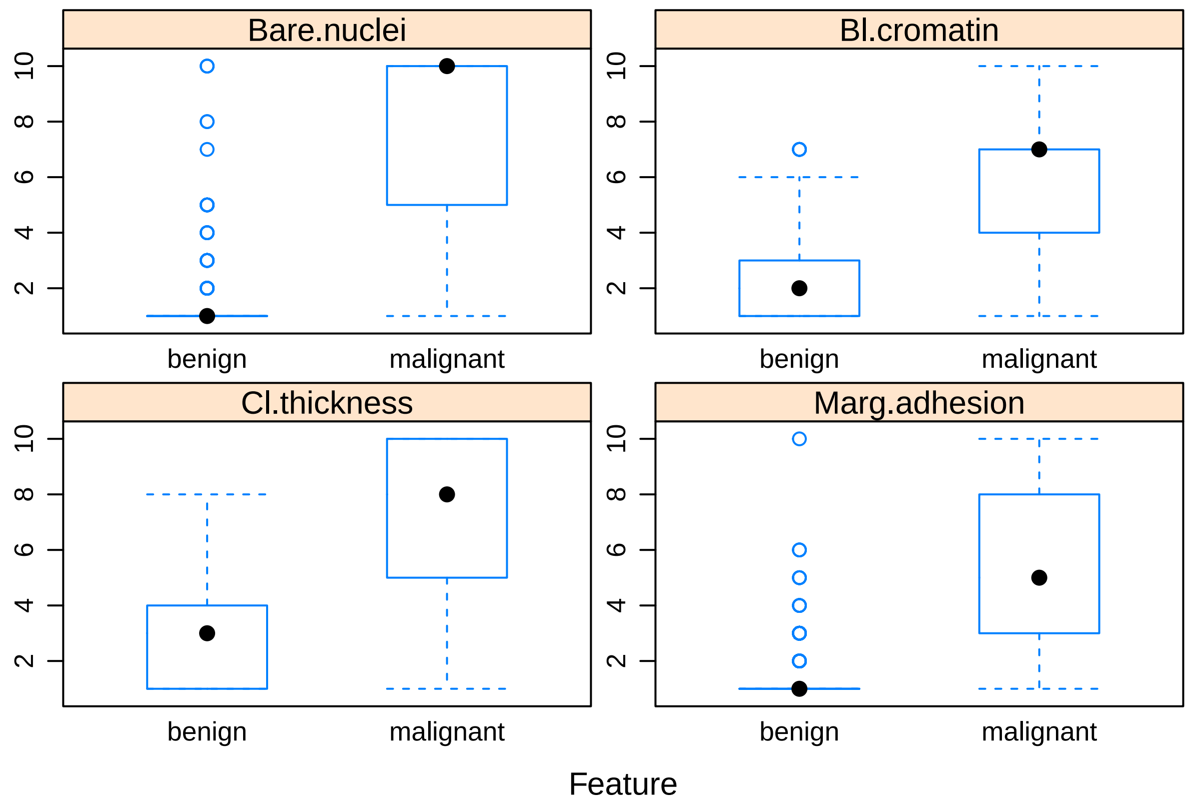
(conclusion/ explanation here)

Below are the feature plots for only the significant predictors, so that we may better see the difference in distributions between the variable and the outcome.

```r
featurePlot(x=BreastCancer[,c(1,4,6,7)], y=BreastCancer[,10],
            plot="density",
            scales=list(x=list(relation="free"),
                        y=list(relation="free")),
            auto.key=list(columns=3))
```



```r
featurePlot(x=BreastCancer[,c(1,4,6,7)], y=BreastCancer[,10],
            plot="box",
            scales=list(x=list(relation="free"),
                        y=list(relation="free")),
            auto.key=list(columns=3))
```



Now we split our data into a training and test set so we can make predictions

```r
set.seed(1)
BreastCancer.train <- sample(1:nrow(BreastCancer), 410)

BreastCancer.test=BreastCancer[-BreastCancer.train,] # test

Class.test=BreastCancer$Class[-BreastCancer.train]

glm.fits=glm(Class ~ Cl.thickness + Marg.adhesion +
             Bare.nuclei + Bl.cromatin,
           data=BreastCancer,family=binomial, subset=BreastCancer.train)

glm.probs = predict(glm.fits, BreastCancer.test, type = "response")
glm.pred=rep("benign",273)
glm.pred[glm.probs >.5]="malignant"
table(glm.pred, Class.test)
```

```
##            Class.test
## glm.pred     benign malignant
##   benign        176         8
##   malignant       5        84
```

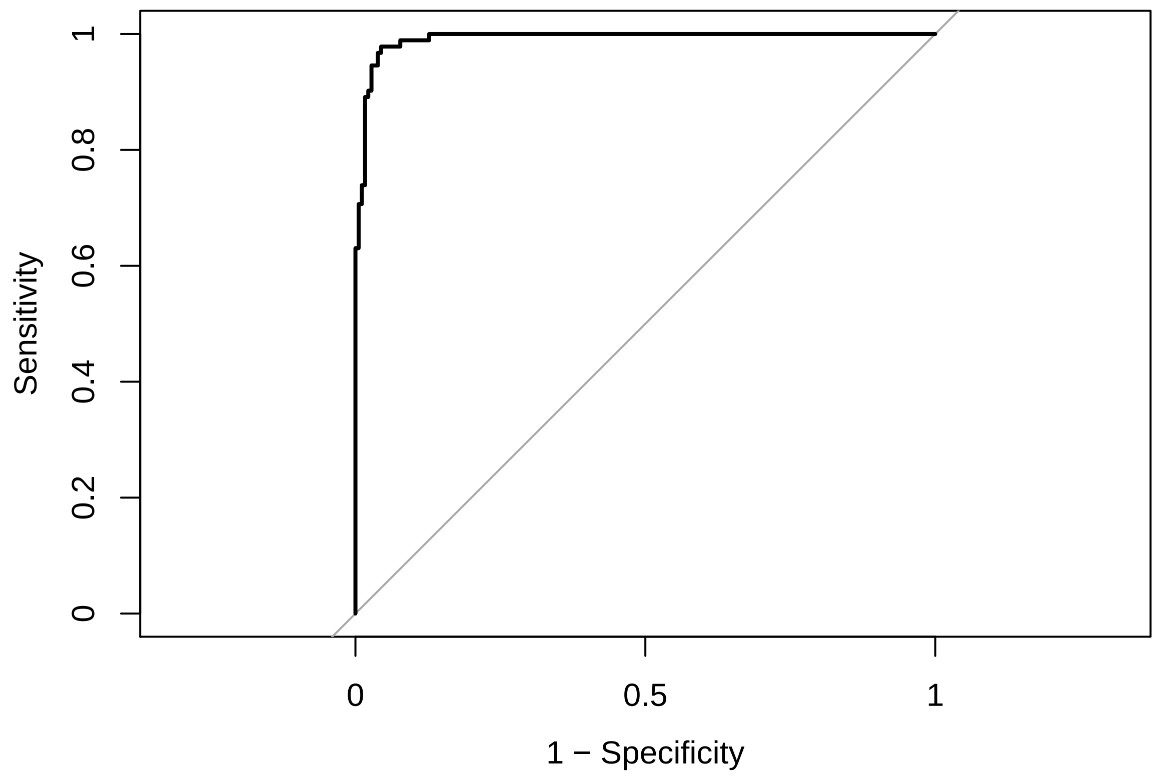
```r
confusionMatrix(glm.pred, Class.test, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       176         8
##   malignant      5        84
##
##                Accuracy : 0.9524
##                  95% CI : (0.9199, 0.9744)
##     No Information Rate : 0.663
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8926
##  Mcnemar's Test P-Value : 0.5791
##
##             Sensitivity : 0.9130
##             Specificity : 0.9724
##          Pos Pred Value : 0.9438
##          Neg Pred Value : 0.9565
##              Prevalence : 0.3370
##          Detection Rate : 0.3077
##    Detection Prevalence : 0.3260
##       Balanced Accuracy : 0.9427
##
##        'Positive' Class : malignant
##
```

```
mean(glm.pred == Class.test)
```

```
## [1] 0.952381
```

```
library(pROC)

roc.glm.train <- roc(BreastCancer.test$Class, glm.probs,
               levels = c("benign", "malignant"))
plot(roc.glm.train, legacy.axes = TRUE)
```



```r
auc(roc.glm.train)
```

```
## Area under the curve: 0.9917
```

There were only 13 incorrect predictions; 5 benign were predicted to be malignant and 8 that were truly malignant were predicted to be benign. Logistic has 95% correct response for test, which is pretty good.

The area under the ROC curve is 99%.

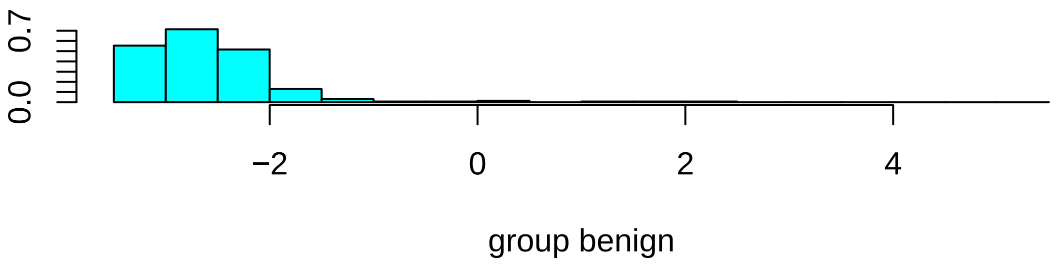
Sensitivity is 91%

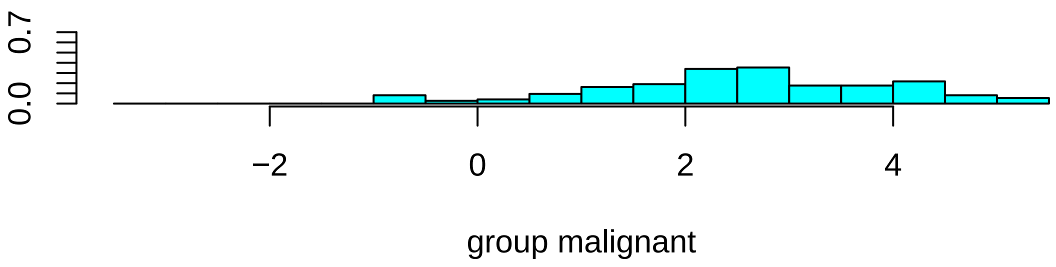
Specificity is 97%

PPV is 94% and NPV is 96%

LDA

```r
library(MASS)
library(pROC)
library(ISLR)
library(caret)

set.seed(1)
train_index <- sample(1:683, 410)

train <- BreastCancer[train_index,]
test <- BreastCancer[-train_index,]

lda.fit <- lda(Class ~ ., data = train)
plot(lda.fit)
```





```
lda.pred <- predict(lda.fit, newdata = test)

roc.lda <- roc(test$Class, lda.pred$posterior[,2],
               levels = c("benign", "malignant"))
#plot(roc.lda, legacy.axes = TRUE)



confusionMatrix(lda.pred$class, test$Class, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       178         9
##   malignant      3        83
##
##                Accuracy : 0.956
##                  95% CI : (0.9245, 0.9771)
##     No Information Rate : 0.663
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9022
##             Specificity : 0.9834
##          Pos Pred Value : 0.9651
##          Neg Pred Value : 0.9519
##              Prevalence : 0.3370
##          Detection Rate : 0.3040
##    Detection Prevalence : 0.3150
##       Balanced Accuracy : 0.9428
##
##        'Positive' Class : malignant
##
```

```r
diag(prop.table(table(lda.pred$class, test$Class), 1))
```

```
##    benign malignant
## 0.9518717 0.9651163
```

```r
set.seed(1)
qda.fit <- qda(Class ~ ., data = train)

qda.pred <- predict(qda.fit, newdata = test)
confusionMatrix(qda.pred$class, test$Class, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       172         3
##   malignant      9        89
##
##                Accuracy : 0.956
##                  95% CI : (0.9245, 0.9771)
##     No Information Rate : 0.663
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9032
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9674
##             Specificity : 0.9503
##          Pos Pred Value : 0.9082
##          Neg Pred Value : 0.9829
##              Prevalence : 0.3370
##          Detection Rate : 0.3260
##    Detection Prevalence : 0.3590
##       Balanced Accuracy : 0.9588
##
##        'Positive' Class : malignant
##
```

```r
diag(prop.table(table(qda.pred$class, test$Class), 1))
```

```
##    benign malignant
## 0.9828571 0.9081633
```

Linear Discriminant Analysis (LDA) resulted in 95.6% accuracy and a kappa statistic of .9 when comparing predictions from the model generated with the training data to the test data classes. Additionally sensitivity was 90.22% and specificity was 98.34% for LDA. For Quadratic Discriminant Analysis (QDA), there was also 95.6% prediction accuracy for the test data and the kappa statistic was slightly improved with a statistic of .9032. QDA had sensitivity of 96.74% and specificity of 95.03%. With higher sensitivity, we have a higher probability of the model accurately predicting malignancy (versus a tumor being benign) given that someone has a malignant tumor. This is important for being able to treat patients using chemotherapy or performing surgery if necessary. With high specificity, we have a higher probability of predicting that someone has a benign tumor given that they have a benign tumor. This is important because we do not want to subject patients to unnecessary risk by performing unnecessary surgery, or putting them on chemotherapy if they don't need to be since it can be very damaging to a patients health. Since both of these values are extremely important for patient health and safety, QDA would be a better choice for a model since we sacrifice too much in sensitivity with LDA. QDA ends up maximizing these values almost equally.

KNN

```
library(class)
set.seed(1)

area_under_curve <- rep(0, 10)
accuracy <- rep(0, 10)
for(i in 1:10) {
  set.seed(1)
  knn_pred <- knn(train[,-10], test[,-10], train$Class, k = i, prob = TRUE)
  scores.knn <- attr(knn_pred,"prob")
  knnROC <- roc(test$Class, scores.knn,
               levels = c("benign", "malignant"))

  area_under_curve[i] <- auc(knnROC)
  accuracy[i] <- confusionMatrix(knn_pred, test$Class)$overall[1]
}

area_under_curve
```

```
##  [1] 0.4891304 0.4513872 0.4377853 0.4267956 0.4075486 0.4034951 0.3948475
##  [8] 0.3602570 0.3540115 0.3434122
```

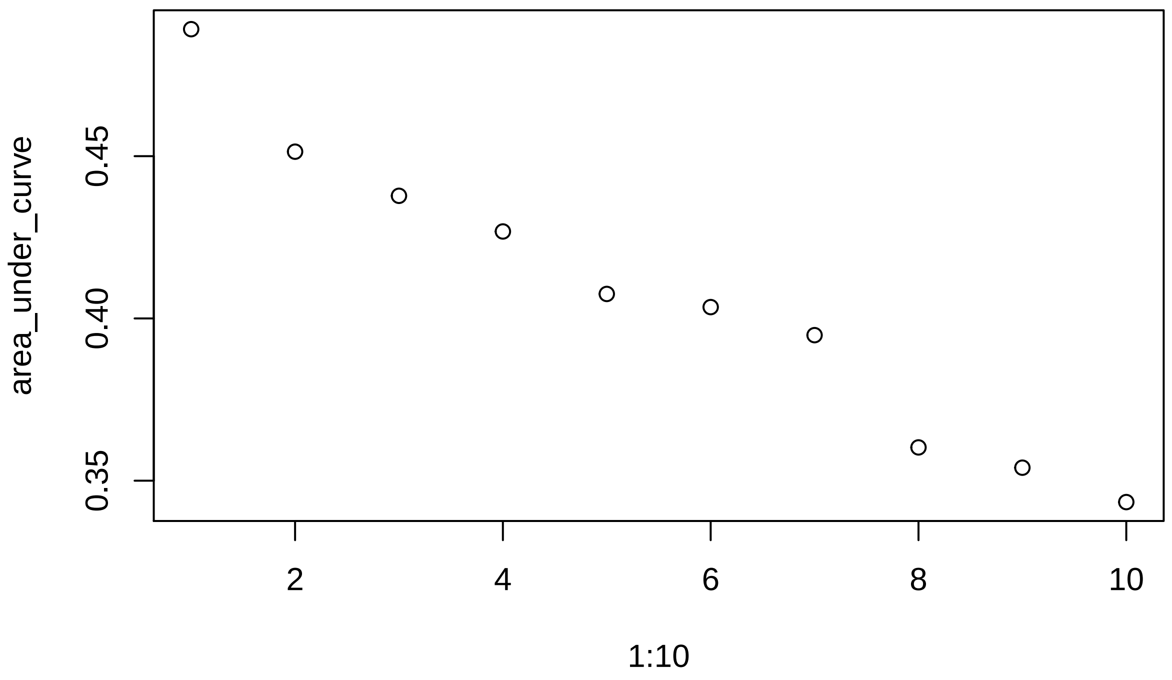
```
accuracy
```

```
##  [1] 0.9523810 0.9413919 0.9670330 0.9670330 0.9633700 0.9597070 0.9633700
##  [8] 0.9633700 0.9633700 0.9633700
```

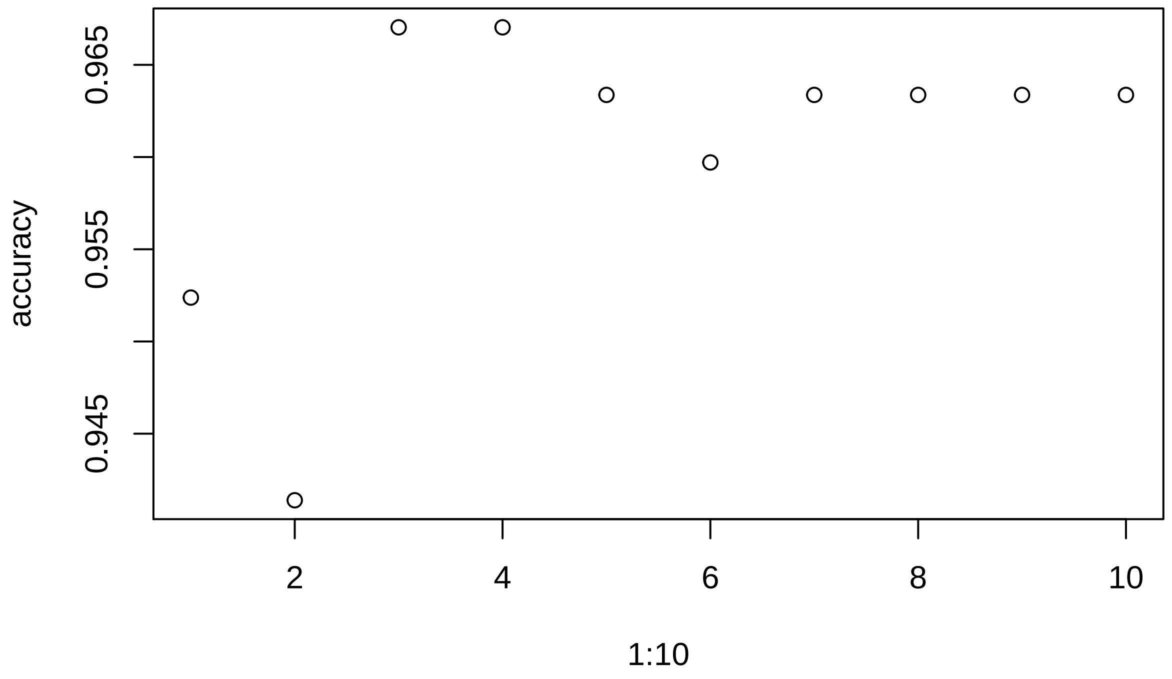
```
confusionMatrix(knn_pred, test$Class)$overall[1]
```

```
## Accuracy
##  0.96337
```

```
knn_output <- data_frame(1:10, area_under_curve, accuracy)

plot(1:10, area_under_curve)
```



```r
plot(1:10, accuracy)
```



```r
set.seed(1)
knn_pred <- knn(train[,-10], test[,-10], train$Class, k = 3, prob = TRUE)
scores.knn <- attr(knn_pred,"prob")
knnROC <- roc(test$Class, scores.knn,
              levels = c("benign", "malignant"))

auc(knnROC)
```

```
## Area under the curve: 0.4378
```

```r
confusionMatrix(knn_pred, test$Class, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       178         6
##   malignant      3        86
##
##                Accuracy : 0.967
##                  95% CI : (0.9383, 0.9848)
##     No Information Rate : 0.663
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9256
##  Mcnemar's Test P-Value : 0.505
##
##             Sensitivity : 0.9348
##             Specificity : 0.9834
##          Pos Pred Value : 0.9663
##          Neg Pred Value : 0.9674
##              Prevalence : 0.3370
##          Detection Rate : 0.3150
##    Detection Prevalence : 0.3260
##       Balanced Accuracy : 0.9591
##
##        'Positive' Class : malignant
##
```

After comparing the area under the ROC curve and the percentage of acccuracy in predicting tumor type, the best number of neighbors was chosen to be 3. This value appears to maximize accuracy without losing too much in area under the curve. The chosen number of neighbors has area under the curve equal to .4378 and prediction accuracy against the test set of 96.7%. The kappa statistic is .9256. The sensitivity and specificity are 93.48% and 98.34%, respectively.

Support Vector Machine

```r
library(e1071)

set.seed(1)
tune.out <- e1071::tune(svm, Class ~ ., data = train,
                 type = "C-classification", kernel = "linear",
                 ranges = list(cost = c(0.001 , 0.01, 0.1, 1,5,10,100)))

svmfit <- e1071::svm(Class ~ ., data = train, kernel = "linear", cost = 0.01)

pred.svm.train <- predict(svmfit, test )
confusionMatrix(test$Class, pred.svm.train, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       176         5
##   malignant      7        85
##
##                Accuracy : 0.956
##                  95% CI : (0.9245, 0.9771)
##     No Information Rate : 0.6703
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9011
##  Mcnemar's Test P-Value : 0.7728
##
##             Sensitivity : 0.9444
##             Specificity : 0.9617
##          Pos Pred Value : 0.9239
##          Neg Pred Value : 0.9724
##              Prevalence : 0.3297
##          Detection Rate : 0.3114
##    Detection Prevalence : 0.3370
##       Balanced Accuracy : 0.9531
##
##        'Positive' Class : malignant
##
```

```
tune.out <- tune(svm, Class ~ ., data = train,
                 kernel = "radial",
                 ranges = list(cost = c(0.1,1,10,100,1000),
                          gamma = c(0.0001,0.001,0.01,0.1,0.5,1)))

svmfit <- e1071::svm(Class ~ ., data = train, kernel = "radial", cost = 10, gamma = .1)

svm.tune.pred.train <- predict(svmfit,
                         newdata = test)


confusionMatrix(test$Class, svm.tune.pred.train, positive = "malignant")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       173         8
##   malignant      6        86
##
##                Accuracy : 0.9487
##                  95% CI : (0.9155, 0.9717)
##     No Information Rate : 0.6557
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8858
##  Mcnemar's Test P-Value : 0.7893
##
##             Sensitivity : 0.9149
##             Specificity : 0.9665
##          Pos Pred Value : 0.9348
##          Neg Pred Value : 0.9558
##              Prevalence : 0.3443
##          Detection Rate : 0.3150
##    Detection Prevalence : 0.3370
##       Balanced Accuracy : 0.9407
##
##        'Positive' Class : malignant
##
```

The linear kernel with tuning resulted in better performance using cost = .01, and with this model the prediction accuracy on the test data was 95.6% with a kappa statistic of .9011. The sensitivity for the linear kernel was 94.44% and the specificity was 96.17%. Support vector classification with a radial kernel had test set accuracy of 94.87% and a kappa statistic of .8858 (after tuning the parameters). The sensitivity for the radial kernel was 91.49% and the specificity was 96.65%. In comparing prediction accuracy of support vector classification using a linear kernel versus a radial kernel after tuning parameters, the linear kernel performed better on this data.

Random Forrest

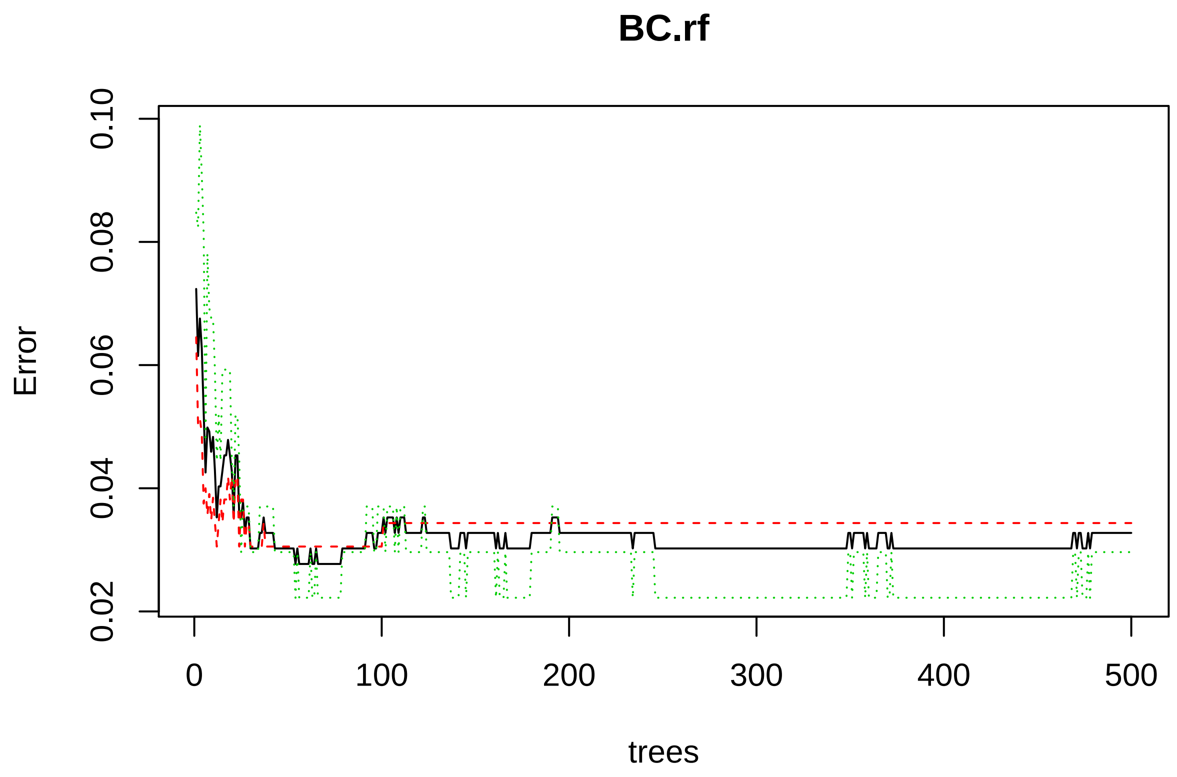
```r
library(mlbench)
data(BreastCancer)
library(tidyverse)
library(randomForest)
```

```r
set.seed(1)
train_sample <- sample(1:nrow(BreastCancer), 410)
BreastCancer.train=BreastCancer[train_sample,]
BreastCancer.test=BreastCancer[-train_sample,]
```

```r
set.seed(10)
BC.rf=randomForest(Class ~ . , data = BreastCancer, subset = train_sample, na.action = na.exclude, impo
BC.rf #RF chooses 3 variables
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = BreastCancer, importance = T,      subset = train_sample, 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 3.27%
## Confusion matrix:
##           benign malignant class.error
## benign       253         9  0.03435115
## malignant      4       131  0.02962963
```

```r
plot(BC.rf)
```



```r
summary(BC.rf)
```

```
##                 Length Class   Mode
## call               6   -none-  call
## type               1   -none-  character
## predicted        397   factor  numeric
## err.rate        1500   -none-  numeric
## confusion          6   -none-  numeric
## votes            794   matrix  numeric
## oob.times        397   -none-  numeric
## classes            2   -none-  character
## importance        40   -none-  numeric
## importanceSD      30   -none-  numeric
## localImportance    0   -none-  NULL
## proximity          0   -none-  NULL
## ntree              1   -none-  numeric
## mtry               1   -none-  numeric
## forest            14   -none-  list
## y                397   factor  numeric
## test               0   -none-  NULL
## inbag              0   -none-  NULL
## terms              3   terms   call
## na.action         13   exclude numeric
```

```r
importance(BC.rf, type=1)
```

```
##                 MeanDecreaseAccuracy
## Id                         0.7952059
## Cl.thickness              13.2657867
## Cell.size                 21.6897212
## Cell.shape                17.1802940
## Marg.adhesion              14.5420982
## Epith.c.size                9.1619407
## Bare.nuclei                18.9476062
## Bl.cromatin                17.3514116
## Normal.nucleoli            12.0412952
## Mitoses                     9.5116092
```

```
test.err=double(10)
oob.err=double(10)
for(mtry in 1:10)
{
  rf=randomForest(Class ~ . , data = BreastCancer.train,
                  mtry=mtry, ntree=200, na.action = na.exclude)
  oob.err[mtry] = rf$err.rate[200]

  pred=predict(rf, BreastCancer.test[1:10]) #Predictions on Test Set for each Tree
  test.err[mtry]= mean(((as.numeric(BreastCancer.test$Class)-1) - (as.numeric(pred)-1)^2),
                       na.rm = T) #Mean Squared Test Error

  cat(mtry," ") #printing the output to the console

}
```

```
## 1  2  3  4  5  6  7  8  9  10
```

```
oob.err
```

```
##  [1] 0.03526448 0.03274559 0.03526448 0.03274559 0.03778338 0.03274559
##  [7] 0.03778338 0.03526448 0.04030227 0.03778338
```

```
test.err
```

```
##  [1] 0.000000000 0.000000000 0.003496503 0.010489510 0.003496503
##  [6] 0.006993007 0.006993007 0.010489510 0.006993007 0.010489510
```

```
importance(rf)
```

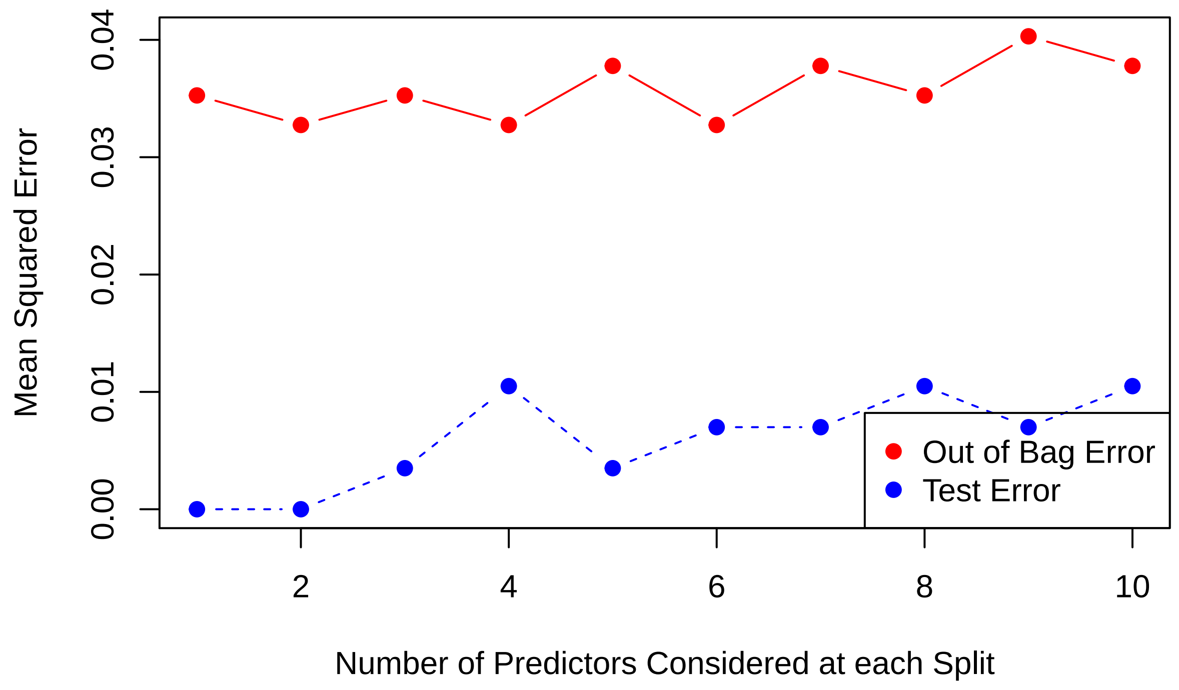
```
##                 MeanDecreaseGini
## Id                      1.401452
## Cl.thickness            2.513398
## Cell.size             131.474324
## Cell.shape              4.705948
## Marg.adhesion           3.685853
## Epith.c.size            0.906835
## Bare.nuclei            11.163752
## Bl.cromatin            10.328135
## Normal.nucleoli         9.356854
## Mitoses                 1.372466
```

```
rf
```

```
##
## Call:
##  randomForest(formula = Class ~ ., data = BreastCancer.train,      mtry = mtry, ntree = 200, na.acti
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 10
##
```

```
##         OOB estimate of  error rate: 3.78%
## Confusion matrix:
##           benign malignant class.error
## benign       252        10  0.03816794
## malignant      5       130  0.03703704
```

```
matplot(1:mtry , cbind(oob.err,test.err), pch=19 , col=c("red","blue"),type="b",ylab="Mean Squared Erro
legend("bottomright",legend=c("Out of Bag Error","Test Error"), pch=19, col=c("red","blue"))
```



Applying random forests techniques to predict tumor classification results in 3 variables being deemed important in contributing to this classification. Restricting the number of variables used at each split per tree to three yields the best out-of-bag error, as well as the (tied-for) best test error. 3 variables is also relatively simple, which is important to consider when building classifier models. The 3 variables that most effectively decrease impurity at each decision node are cell size, bland chromatin (which measures texture of thhe nucleus), and bare nuclei, a term used to describe the composition of the nucleus.

Conclusions

Overall Comparison of Methods

```
library(knitr)

accuracy <- c("95.24%" , "95.45%", '96.7%', "95.6%", "95.6%", "95.6%")
kappa_stat <- c(.8926 , .9012, .9256, .9, .9032, .9011)
sensitivity <- c("91.30%", '92.31%', "93.48%", "90.22%", "96.74%", "94.4%")
specificity <- c("97.24%", "97.25%", "98.34%", "98.34%", "95.03%", "96.17%")


summary <- data_frame(accuracy, kappa_stat, sensitivity, specificity)

row.names(summary) <- c("Logistic Regression", "Random Forest", "KNN","LDA","QDA", "Support Vector Clas
```

```r
colnames(summary) <- c("Test Accuracy", "Kappa Statistic", "Sensitivity", "Specificity")

kable(summary, align = "c")
```

	Test Accuracy	Kappa Statistic	Sensitivity	Specificity
Logistic Regression	95.24%	0.8926	91.30%	97.24%
Random Forest	95.45%	0.9012	92.31%	97.25%
KNN	96.7%	0.9256	93.48%	98.34%
LDA	95.6%	0.9000	90.22%	98.34%
QDA	95.6%	0.9032	96.74%	95.03%
Support Vector Classification	95.6%	0.9011	94.4%	96.17%