Predicting Breast Cancer Tumor Malignancy

Jean Chakmakas

May 4, 2017

Introduction

The goal of this project was to use the "Wisconsin Breast Cancer Database" to accurately predict the outcome of a tumor (malignant or benign) based on a set of descriptive attributes. To do so, the data was manipulated in R studio, and the random forest method was used to model the data.

Data Collection and Preparation

To obtain the data, the "breast-cancer-wisconsin.data" file was downloaded from the UCI website [1]. The data was then downloaded in R studio [2]. Initially, there were 699 records in the dataset.

```
# Load in the data
df = read.csv("proj1data.csv")
```

The ten attributes are the sample code number, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nuclei, mitosis, and class [1]. The first attribute, sample code number, is an integer ID number, and the class is the outcome variable. The class variable is either 2 for benign, or 4 for malignant. The remaining nine variables are scored on an integer scale ranging from one to ten. The code for displaying the column names and number of cases for each class outcome variable is shown below.

```
# Column names
colnames(df)
```

```
##  [1] "ID"            "Thickness"     "Uni_size"      "Uni_shape"
##  [5] "Marg_adhesion" "SECS"          "Bare_nuclei"   "Bland_chrom"
##  [9] "Normal_nuclei" "Mitoses"       "Class"
```

```
# Class column
table(df[11])
```

```
##
##   2   4
## 458 241
```

A small amount of data cleaning was required. In the seventh column representing the bare nuclei attribute, there were 16 records with a question mark instead of an integer value. This is interpreted as a missing value, and to handle this, removal of all rows where this was the case was necessary. The resulting dataset contains 683 records.

```
# Bare_nuclei column (16 records with '?')
table(df[7])
```

```
##
##   ?   1  10   2   3   4   5   6   7   8   9
##  16 402 132  30  28  19  30   4   8  21   9
```

```
# Remove records with question marks
df <- df[-grep("\\?", df$Bare_nuclei), ]
```

Following data cleaning, the data must be separated into a training set and a test set. The training set is used to learn the model, and the testing set is used to verify that the model works correctly and does not overfit the data. The amount of data chosen to be used for the training set was 80%, and the remaining 20% of data was used for the testing set. This was done so that the model would have enough examples to learn from, and also have enough testing examples to gauge whether the model performed well. The "createDataPartition" method was used from the "caret" package, because it splits the data based on the outcome variable [2]. This is important because there will be an even amount of benign and malignant outcomes in each set of data.

```
### Split data into training set and testing set
library(caret)
```

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

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

```
set.seed(1234)
partition <- createDataPartition(df$Class, p = 0.8, list = FALSE, times = 1)
trainingSet <- df[partition, ]
testingSet <- df[-partition, ]
```

Modeling

To model this data, the random forest algorithm was chosen. This method creates a series of decision trees, and for classification problems, outputs the mode of the decision at each decision tree as the prediction. This method was chosen because it usually performs well with classification data, and it is resistant to overfitting (as long as the model is simple enough) because of the large amount of trees that are produced. The "randomForest" package in R was used, and all parameters were included in the model because there are only nine for this problem [2][3]. If there were many more, a feature selection algorithm would be important to use. The model performance on the training data is very good. The benign class has an error of about 2.2%, and the malignant class has an error of only about 1.6%.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(123)
# random forest importance -> importance of predictors is able to be accessed
# ntrees -> 2000 trees are produced in the forest
fit <- randomForest(as.factor(Class) ~ Thickness + Uni_size + Uni_shape + Marg_adhesion +
    SECS + Bare_nuclei + Bland_chrom + Normal_nuclei + Mitoses, data = trainingSet,
    importance = TRUE, ntree = 2000)
```

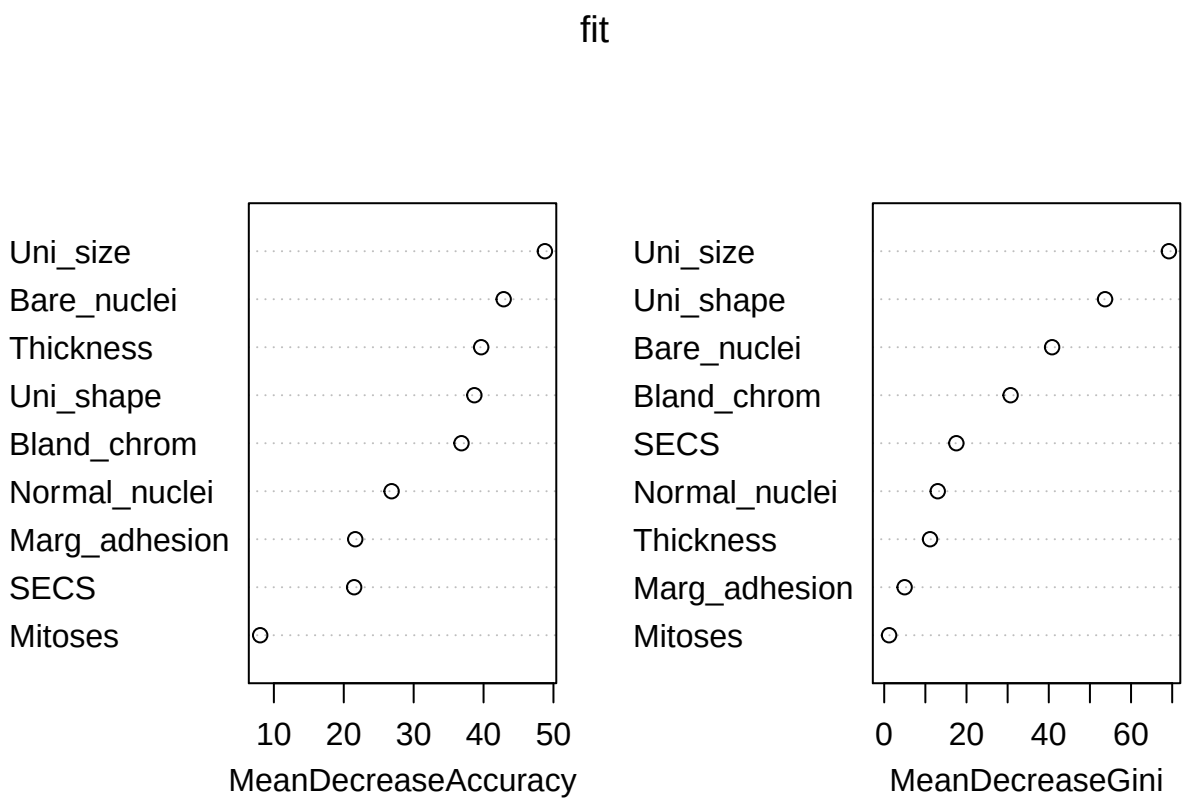
```
# display confusion matrix of fit on training data
fit$confusion
```

```
##     2   4 class.error
## 2 357   8  0.02191781
## 4   3 179  0.01648352
```

Evaluation

To assess the importance of each attribute, the method "varImpPlot" was used [2][3]. This method displays two graphs. The first shows the mean decrease in accuracy for each attribute, and the second shows the mean decrease in the Gini index. The mean decrease in accuracy measures the accuracy after one attribute has been removed [3]. If there is a high decrease in accuracy, it is a strong indication that the removed variable is very important in predicting the outcome. If it is very low, the variable is likely not important to the model. The mean decrease in the Gini index essentially does the same thing, but it is more of a measure of node purity [3]. The most important attribute in this example is uniformity of cell size, which is shown in both graphs. Mitosis is shown to be the least important, and could be removed from the model. The other attributes all appear to be relatively important, but vary based on which measure of importance is used.

```
# variable importance
varImpPlot(fit)
```



When evaluated with the testing set, the model performs well. It is not quite as good as the training data, but still confirms that the model is not overfitting to this data. The accuracy for the benign class is 94.9%, and the accuracy for the malignant class is 93.0%.

```r
# test with testing data
pred <- predict(fit, testingSet)
table(pred, testingSet$Class)
```

```
##
## pred  2  4
##    2 75  4
##    4  4 53
```

Conclusion

Although the model performs well, it is difficult to accept it in practice. The possibility that a case could be misclassified is too large of a risk. On one hand, if the case is benign and it is classified as malignant, the person would have to undergo unnessessary treatment. This is of course not desired, but it is worse to misclassify a malignant case as benign, for obvious reasons. This model should be used to reinfornce a diagnosis, but cannot be used to predict until it is perfect.

References

[1]"UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set", Archive.ics.uci.edu, 2017. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). [Accessed: 03- May- 2017].
[2]"Home", RStudio, 2017. [Online]. Available: https://www.rstudio.com. [Accessed: 03- May- 2017].
[3]T. Stephens, "Titanic: Getting Started With R - Part 5: Random Forests", Trevor Stephens, 2017. [Online]. Available: http://trevorstephens.com/kaggle-titanic-tutorial/r-part-5-random-forests/. [Accessed: 03- May- 2017].