PSTAT 131 Final Project

Owen Philliber

2023-12-13

```
data <- read_csv('Healthcare-Diabetes.csv')
```

```
## Rows: 2768 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (10): Id, Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, B...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# removing the Unique Identifier as it is not important
data <- data[,-1]

# Converting the outcome to a factor
data$Outcome <- factor(data$Outcome)

summary(data)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:  0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median : 23.00
##  Mean   : 3.743   Mean   :121.1   Mean   : 69.13   Mean   : 20.82
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.: 32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :110.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:27.30   1st Qu.:0.2440           1st Qu.:24.00
##  Median : 37.00   Median :32.20   Median :0.3750           Median :29.00
##  Mean   : 80.13   Mean   :32.14   Mean   :0.4712           Mean   :33.13
##  3rd Qu.:130.00   3rd Qu.:36.62   3rd Qu.:0.6240           3rd Qu.:40.00
##  Max.   :846.00   Max.   :80.60   Max.   :2.4200           Max.   :81.00
##  Outcome
##  0:1816
##  1: 952
##
##
##
##
```

Introduction

This data set was found on Kaggle, produced by Nandita Pore. The data set can be found here: https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes/data. In our data frame, we have 8 explanatory variables and one response variable. All the explanatory variables are continuous variables and the response variable is a factor. Thus our models will be classification models attempting to model if a patient is diabetic or not. In this project, we will use various decision tree techniques in order to explain the presence of diabetes in a patient. The decision tree methods we will use is classification tree, pruned classification tree, bagged decision tree, random forest tree, and a boosted tree. We will then analyze the performance of the different methods using the test error rate estimate.

Exploritory Analysis of the Data Set

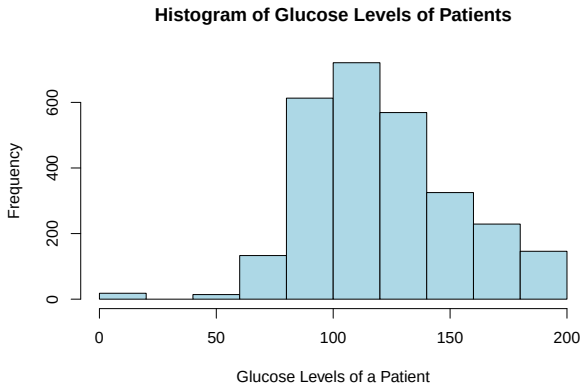
Explanatory Variables:

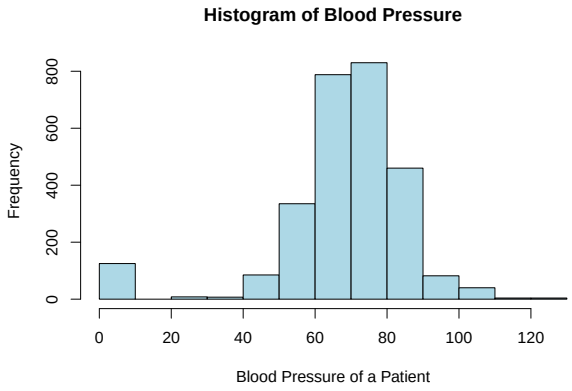
1) Pregnancies: how many times the patient has been pregnant
2) Glucose: The concentration of plasma glucose over a 2 hour glucose tolarance test
3) BloodPressure: Diallastic Blood Pressure of the patient (mm Hg)
4) SkinThickness: The thickness of the skin of the tricep skinfolds (mm)
5) Insulin: 2-Hour serum insulin (mu U/ml)
6) BMI: Body mass index of the patient (weight in kg / height in mˆ2)
7) DiabetesPedigreeFunction: A genetic score of diabetes
8) Age: Age of the patient

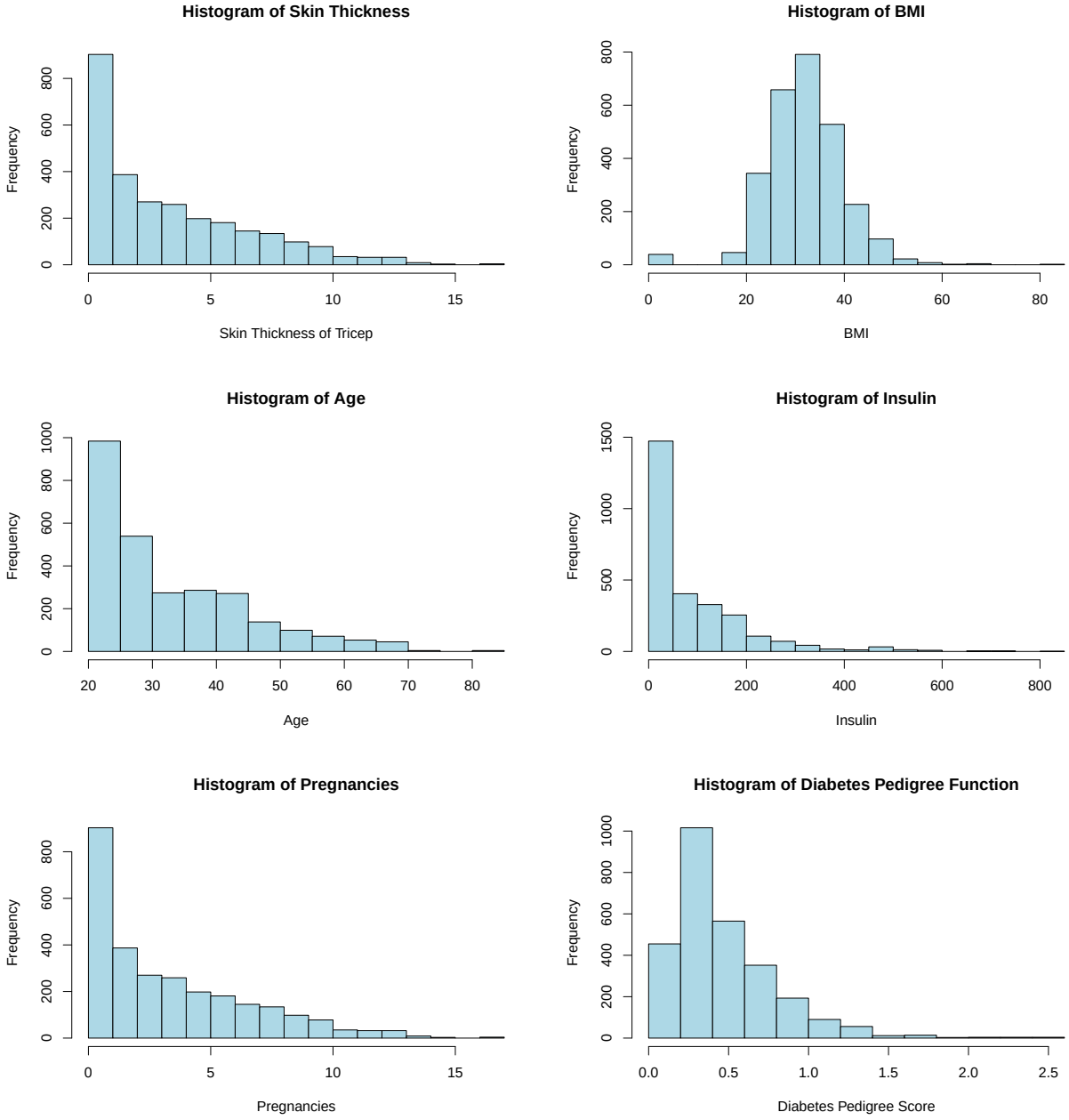
Response Variable:

Outcome: indicator if diabetes is present in the patient or not, 1 indicates there is presence and 0 indicates there is not

In order to run regressions on this data, it is important to understand the trends of the data. A histogram of each variable is important to understanding the distribution of each variable. Histograms are also a great way to see any potential high leverage points.

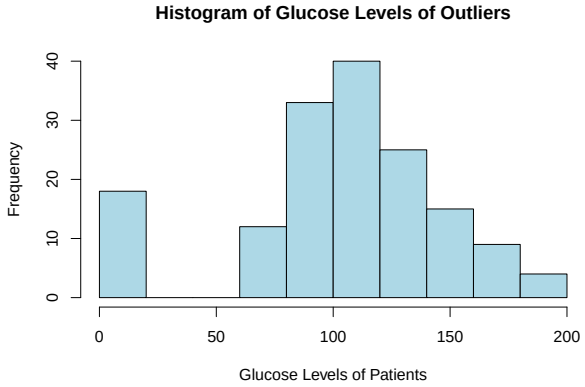


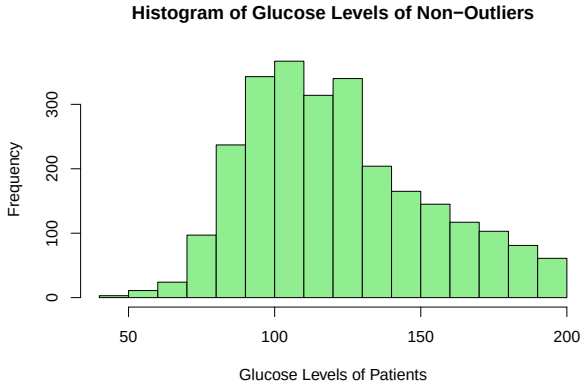


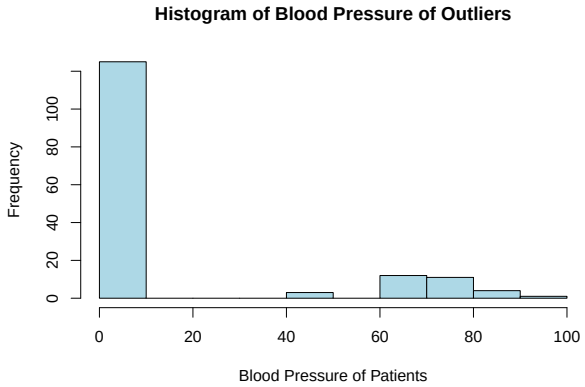


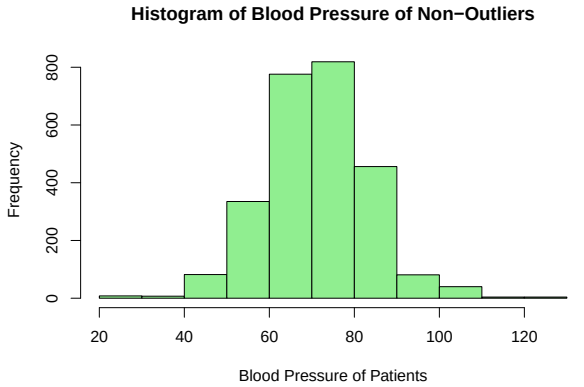
The graphs for glucose levels, blood pressure, skin thickness, and BMI all have observations that are zero. This is a concern since if any of these variables were zero on a patient, they would be dead. Based on the graph of the skin thickness, there are over 800 patients that have a skin thickness of zero. Since that is not possible we will assume that the measurement is not an outlier and occurs from the nature of the data collection system.

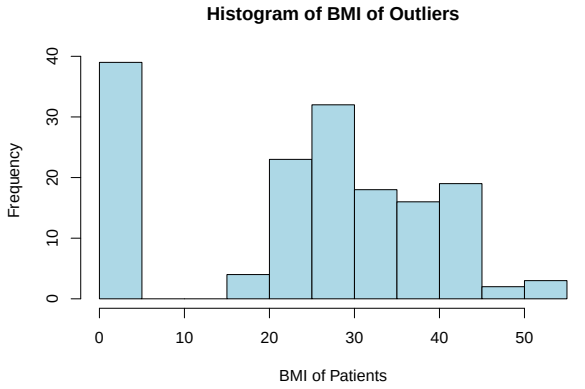
As for the other variables, we will produce histograms of the outlier data to the rest of the data. This is to see if there are any trends in the observations with zero glucose levels, blood pressure, or BMI.

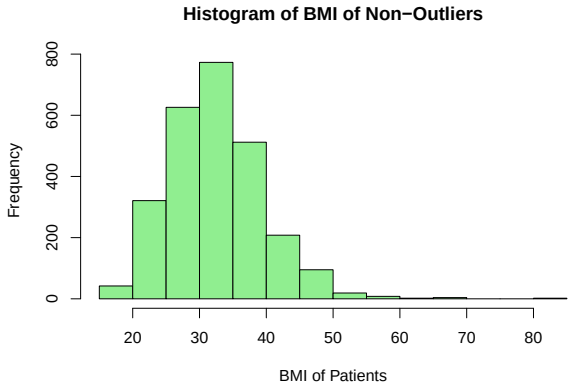






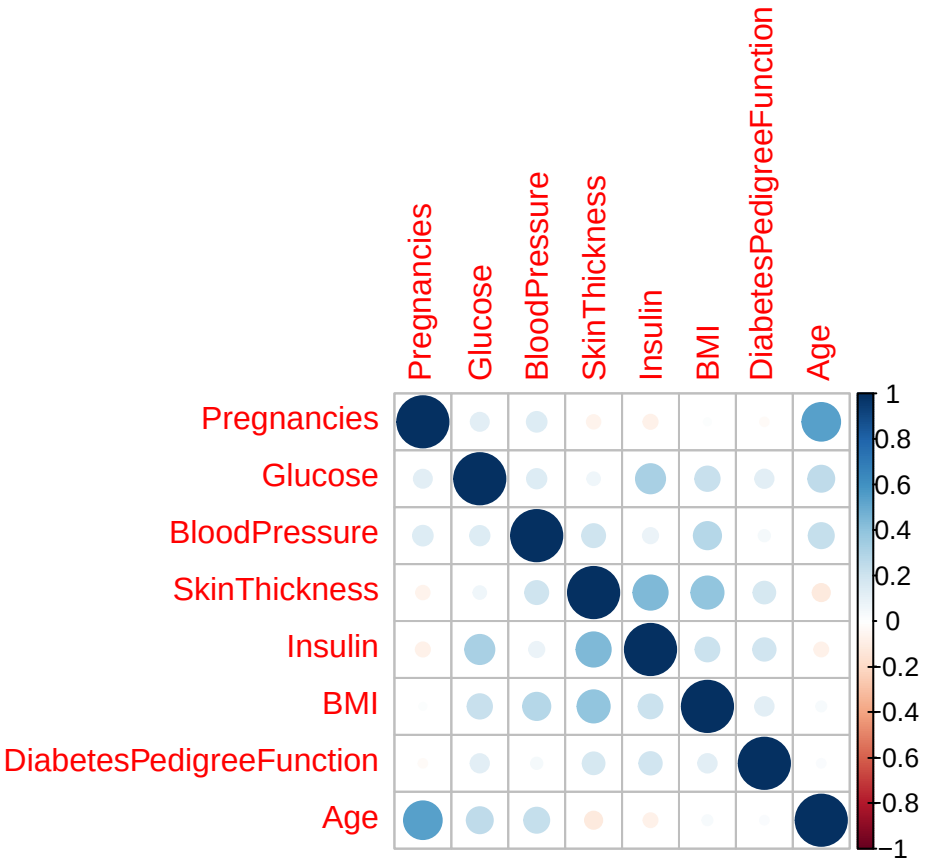






These graphs show that besides the variables that are zero, the observations with outliers are distributed similar to the rest of the observations. Since there are 156 observations, which is only 0.0563584% of the data set, the outliers do not impact the data set significantly. Thus, we will leave them in the dataset.

Another important measure to analyze the data is the correlation. The graph below is a correlation graph, so each box measures the correlation between the row variable and the column variable. The larger and darker the circle is the mor correlated the two variables are.



As shown by the graph, the variables pairs that have the most correlation are pregnancy and age, skin thickness and insulin, and skin thickness and BMI. All the other variable pairs are mostly uncorrelated.

Models

Since the data set has 2768 observations, we will split the data set into a training set and a test set. The test set will be for comparing the performances of the different models. We us roughly 80% of the data for the training data and the other 20% will be for a test set. In order to use as much of the data for training, we will use cross validation for tuning parameters. This allows us to only have a training and test set instead of a training, validation, and test set.

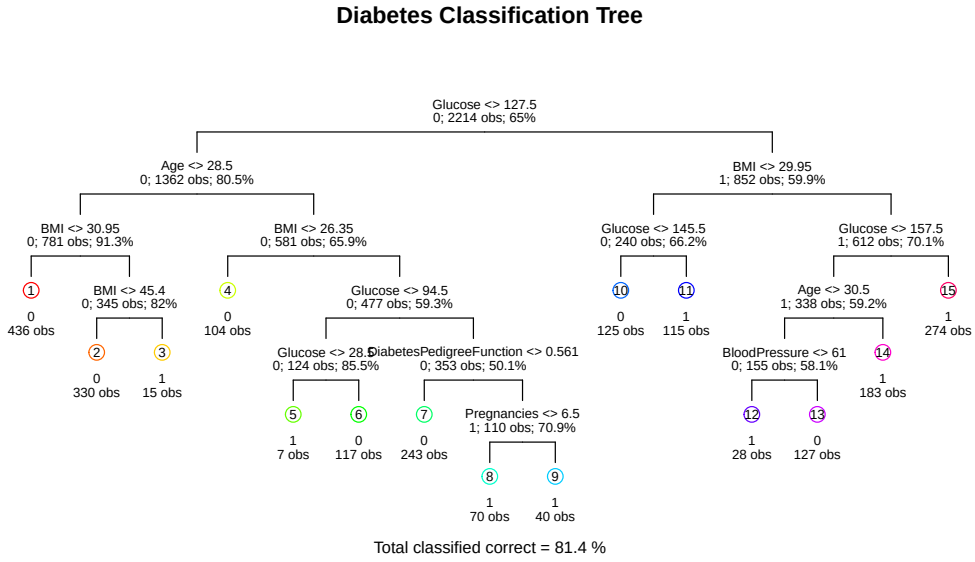
```
train <- sample(nrow(data), nrow(data)*.8)

data.train <- data[train, ]
data.test <- data[-train, ]
```

Basic Binary Tree

A basic binary tree is a non-parametric method that splits the predictor space into regions. For classification, the mode of the region is used for the prediction for any variable that falls into that region. R uses a top down greedy approach which starts with all observations belonging to a single region. Then the function calculates the best way to split the region using the specified criterion. The default splitting criterion for the tree function in R is the deviance criterion.

```
##
## Classification tree:
## tree(formula = Outcome ~ ., data = data.train)
## Variables actually used in tree construction:
## [1] "Glucose"                  "Age"
## [3] "BMI"                      "DiabetesPedigreeFunction"
## [5] "Pregnancies"              "BloodPressure"
## Number of terminal nodes:  15
## Residual mean deviance:  0.8069 = 1774 / 2199
## Misclassification error rate: 0.1861 = 412 / 2214
```



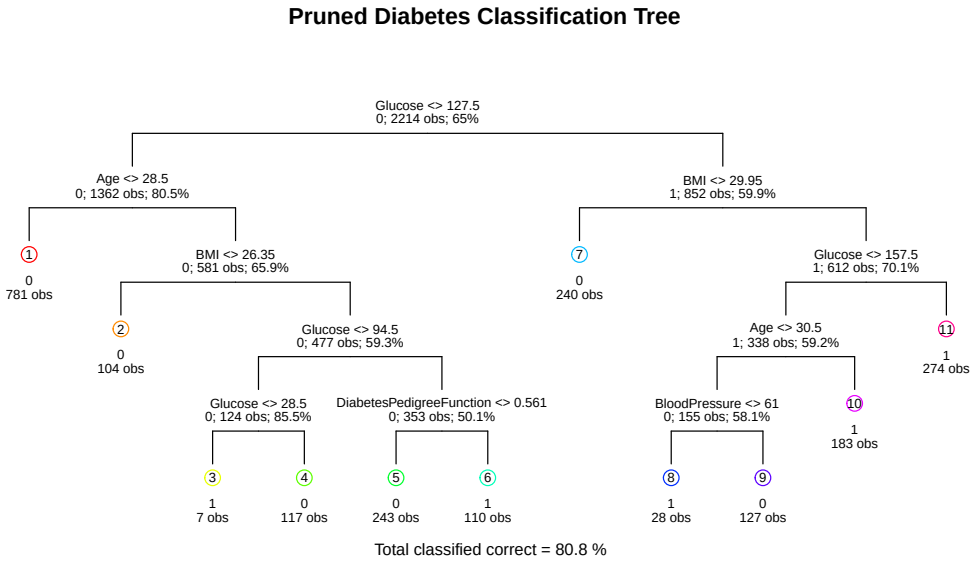
The basic tree has many flaws. For example, the lowest split of pregnancies is redundant since both sides of the split lead to a prediction of 1.

Pruning the Tree

Often times an ordinary binary classification tree will over fit the data, thus we must prune the data. Pruning is achieved through minimizing the splitting criterion after adding a cost of complexity to the splitting criterion. The way we do this is by limiting the size and preforming a cross validation changing the size of the trees used.

```
##
## Classification tree:
## snip.tree(tree = tree.df, nodes = c(47L, 6L, 4L))
## Variables actually used in tree construction:
## [1] "Glucose"                  "Age"
## [3] "BMI"                      "DiabetesPedigreeFunction"
## [5] "BloodPressure"
## Number of terminal nodes:  11
## Residual mean deviance:  0.8827 = 1945 / 2203
```

```
## Misclassification error rate: 0.1915 = 424 / 2214
```



As we can see above, the pruned tree is much more simple. This is better for readability and interpretability since every split has a purpose. Also the pruned tree should theoretically have a lower test error rate since the pruned tree should be more generalized.

Bagging

Bagging is a tree building technique that is built open using bootstrapping to create more training sets. From the bootstrap sets, a model is built upon each of the bootstrap set and then the mode of all the model outcomes is the result. The aim of bagging is to reduce the overall variance of the model. This is because the variance of an average reduces as the number of variables being averaged increases. This is shown in the equation $Var(\frac{1}{B}\sum V_i) = \frac{\sigma^2}{B}$ where B is the number of independent random variables.

```
##
## Call:
##  randomForest(formula = Outcome ~ ., data = data.train, mtry = 8)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 0.81%
## Confusion matrix:
##      0   1 class.error
## 0 1430   8 0.005563282
## 1   10 766 0.012886598
```

```
importance(bag.tree)
```

```
##                           MeanDecreaseGini
## Pregnancies                        61.28442
## Glucose                           306.97156
## BloodPressure                      90.15323
## SkinThickness                      48.42909
## Insulin                            57.65054
## BMI                               180.33109
## DiabetesPedigreeFunction          127.80993
## Age                               135.50137
```

The importance of each variable is found through the average of how much the gini index is decreased from the splits from that predictor. As we can see, Glucose, BMI, and Age are the three most important variables from the bagged tree.

Random Forest

Random Forest is similar to bagging, but when creating each tree random forests selects $m < p$ predictors at random. This helps decorrelate the trees. For classification typically $m = \sqrt{p}$, so in this case $m = \sqrt{8} \approx 3$

```
##
## Call:
##  randomForest(formula = Outcome ~ ., data = data.train, mtry = 3)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 0.86%
## Confusion matrix:
##      0   1 class.error
## 0 1431   7 0.004867872
## 1   12 764 0.015463918
```

As we can see the results are comparable to bagging.

```
importance(rf.tree)
```

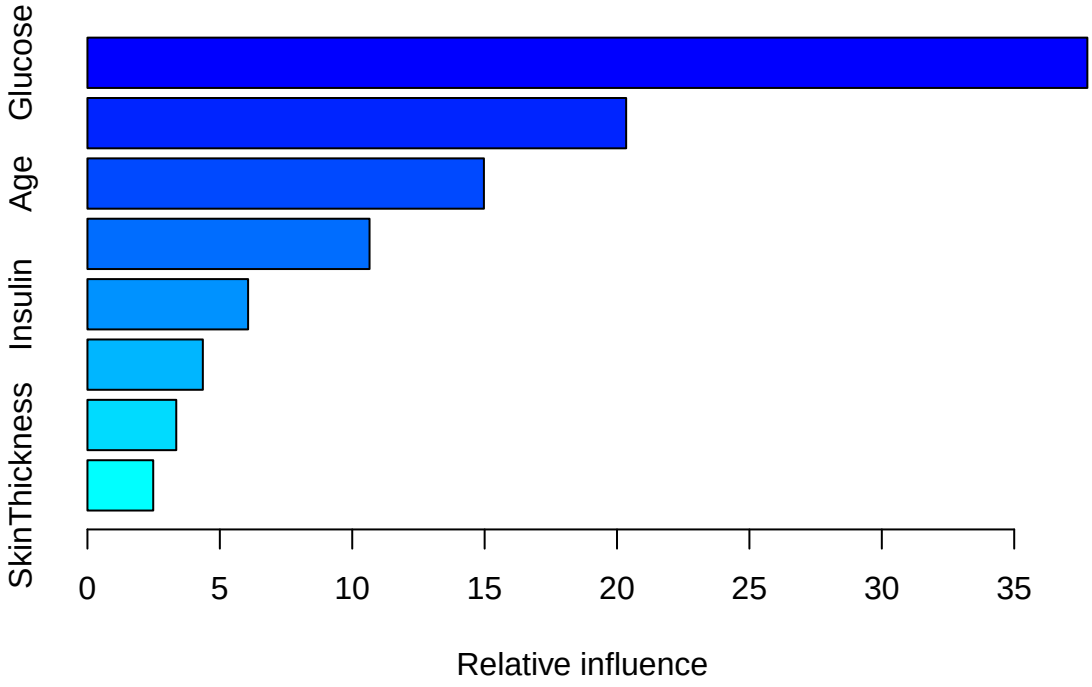
```
##                           MeanDecreaseGini
## Pregnancies                        75.13596
## Glucose                           271.63249
## BloodPressure                      90.10466
## SkinThickness                      62.43977
## Insulin                            65.76347
## BMI                               173.87838
## DiabetesPedigreeFunction          127.50971
## Age                               140.50584
```

To be expected, the order of importance for each variable from the random forest tree is the same as the bagged tree.

Boosting

Boosting is another decision tree method. One major difference between random forests and bagging compared to boosting is that boosting does not rely on bootstrap sampling. Instead, boosting combines trees sequentially.

```
## gbm(formula = Outcome ~ ., distribution = "bernoulli", data = boosting.data,
##     n.trees = 500)
## A gradient boosted model with bernoulli loss function.
## 500 iterations were performed.
## There were 8 predictors of which 8 had non-zero influence.
```




```
##                                                 var   rel.inf
## Glucose                                     Glucose 37.762492
## BMI                                             BMI 20.344850
## Age                                             Age 14.973209
## DiabetesPedigreeFunction DiabetesPedigreeFunction 10.654385
## Insulin                                     Insulin  6.067224
## Pregnancies                             Pregnancies  4.358555
## BloodPressure                         BloodPressure  3.354646
## SkinThickness                         SkinThickness  2.484640
```

As we can see, the training error rate for the boosting tree is higher than the bagging and random forest trees.

Evaluation and Comparison

In order to properly compare the models, a separate test set must be used. This is because if a model is too flexible and overfits the data, then the training MSE will be small while the test MSE could potentially be large.

The test MSE is approximated by the error rate of the model on the test data set. Below is the confusion matrix on the test set along with the test MSE estimate for the basic tree.

```
##
## test.yhat.basic   0   1
##               0 342  55
##               1  36 121
```

```
## [1] "Test Error Rate:"
```

```
## [1] 0.1642599
```

Below is the confusion matrix on the test set along with the test MSE estimate for the pruned tree.

```
##
## test.yhat.prune   0   1
##               0 355  63
##               1  23 113
```

```
## [1] "Test Error Rate:"
```

```
## [1] 0.1552347
```

Below is the confusion matrix on the test set along with the test MSE estimate for the bagged tree.

```
##
## test.yhat.bag   0   1
##             0 378   1
##             1   0 175
```

```
## [1] "Test Error Rate:"
```

```
## [1] 0.001805054
```

Below is the confusion matrix on the test set along with the test MSE estimate for the random forest tree

```
##
## test.yhat.rf   0   1
##            0 378   1
##            1   0 175
```

```
## [1] "Test Error Rate:"
```

```
## [1] 0.001805054
```

Below is the confusion matrix on the test set along with the test MSE estimate for the boosted tree.

```
##
## boost.yhat   0   1
##          0 348  53
##          1  30 123
```

```
## [1] "Test Error Rate:"
```

```
## [1] 0.1498195
```

One observation from the test MSE estimates is that while the basic tree had a lower training error rate than the pruned tree, the pruned tree has a lower test MSE estimate. This is due to the fact that basic trees often over fits the training data. This is an example of why the training MSE is not an indicator for the test MSE. Another observation is that the the random forests and the bagging trees have the exact same low test MSE estimates. My hypothesis for this is because the correlation between the observation variables is low, so the decorrelation of the random forest does not have a large difference.

Results

Through out this project my goal was to accurately model the presence of diabetes according to the data set. Each of the decision tree models had their own strengths and weaknesses. For example, the pruned tree is more understandable and interpretable that the random forest, bagged, and boosted decision trees. And the random forest and bagged decision trees have a low test MSE estimate, so they are more accurate. One room for improvement would be to tune the more of the parameters using cross validation in order to optimize each model further. The number of trees could have been more optimized for the random forest, bagged, and boosted tree as well as the shrinkage factor for the boosted tree. Another improvement would be to fit other models such as support vector machines. Each method we learned in the class can bring insight into analyzing data sets, so the more models the more the data set can be analyzed. The importance measures that the decision trees provide tell us that it is important to further study the interaction between glucose and diabetes, along with BMI and Age and diabetes. Overall, the bagged and random forest models can correctly predict diabetes given the observation variables with a high level of accuracy, which indicates a success on analyzing the data set.