# Decission Trees - Diabetes

Sanjaya Mananage

Consider the Pima Indians Diabetes Database data set(https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database). This data set is created for prediction of whether a patient has diabetes or not. The data set contains several parameters which are considered important during the determination of diabetes. The sample size is 768 and all patients here are females at least 21 years old of Pima Indian heritage. Take Outcome as the binary response variable. I consider all predictors as quantitative variables and take all the data as training data.

For all the models I use leave-one-out cross-validation (LOOCV) to compute the estimated miss classification error rate.

## I fit a decision tree to the data and summarize the results

```
## Warning: package 'tree' was built under R version 4.2.3
```

```
##
## Classification tree:
## tree(formula = Outcome ~ ., data = diabetes.data)
## Variables actually used in tree construction:
## [1] "Glucose"                  "Age"
## [3] "BMI"                      "DiabetesPedigreeFunction"
## [5] "Pregnancies"
## Number of terminal nodes:  11
## Residual mean deviance:  0.8594 = 650.6 / 757
## Misclassification error rate: 0.2057 = 158 / 768
```

```
## [1] 158
```

The Variables actually used in tree construction are “Glucose”, “Age”, “BMI”, “DiabetesPedigreeFunction”, “Pregnancies”. There are 11 nodes and residual mean deviance is 0.8594 and miss classification error rate is 0.2057

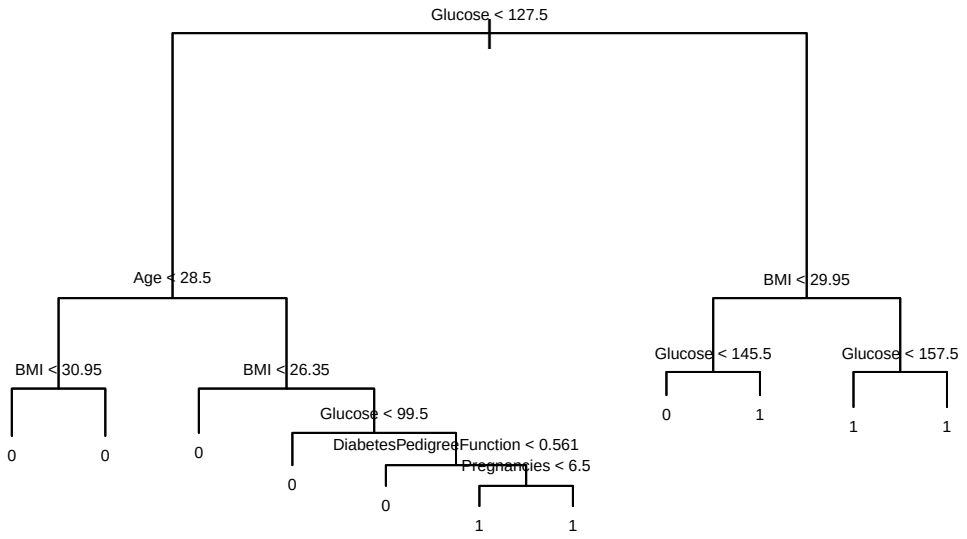


Figure 1: Classification tree for Admission data

Let $R_j$ be the partitions of the predictor space.

$$
\begin{aligned}
R_1 &= \{X \mid Glucose < 127.5, Age < 28.5, BMI < 30.95\} \\
R_2 &= \{X \mid Glucose < 127.5, Age < 28.5, BMI \geq 30.95\} \\
R_3 &= \{X \mid Glucose < 127.5, Age \geq 28.5, BMI < 26.35\} \\
R_4 &= \{X \mid Glucose < 127.5, Age \geq 28.5, BMI \geq 26.35, Glucose < 99.5\} \\
R_5 &= \{X \mid Glucose < 127.5, Age \geq 28.5, BMI \geq 26.35, Glucose \geq 99.5, DiabetesPedigreeFunction < 0.561\} \\
R_6 &= \{X \mid Glucose < 127.5, Age \geq 28.5, BMI \geq 26.35, Glucose \geq 99.5, DiabetesPedigreeFunction \geq 0.561, Pregnancies < 6.5\} \\
R_7 &= \{X \mid Glucose < 127.5, Age \geq 28.5, BMI \geq 26.35, Glucose \geq 99.5, DiabetesPedigreeFunction \geq 0.561, Pregnancies \geq 6.5\} \\
R_8 &= \{X \mid Glucose \geq 127.5, BMI < 29.95, Glucose < 145.5\} \\
R_9 &= \{X \mid Glucose \geq 127.5, BMI < 29.95, Glucose \geq 145.5\} \\
R_{10} &= \{X \mid Glucose \geq 127.5, BMI \geq 29.95, Glucose < 157.5\} \\
R_{11} &= \{X \mid Glucose \geq 127.5, BMI \geq 29.95, Glucose \geq 157.5\}
\end{aligned}
$$

```
##
## pred   0   1
##    0 383  82
##    1 117 186
```

```
miss.classification_rate_a=(117+85)/768
miss.classification_rate_a
```

```
## [1] 0.2630208
```

The test misclassification error rate using LOOCV is 0.2630208.

**I used LOOCV to determine whether pruning is helpful and determine the optimal size for the pruned tree.**

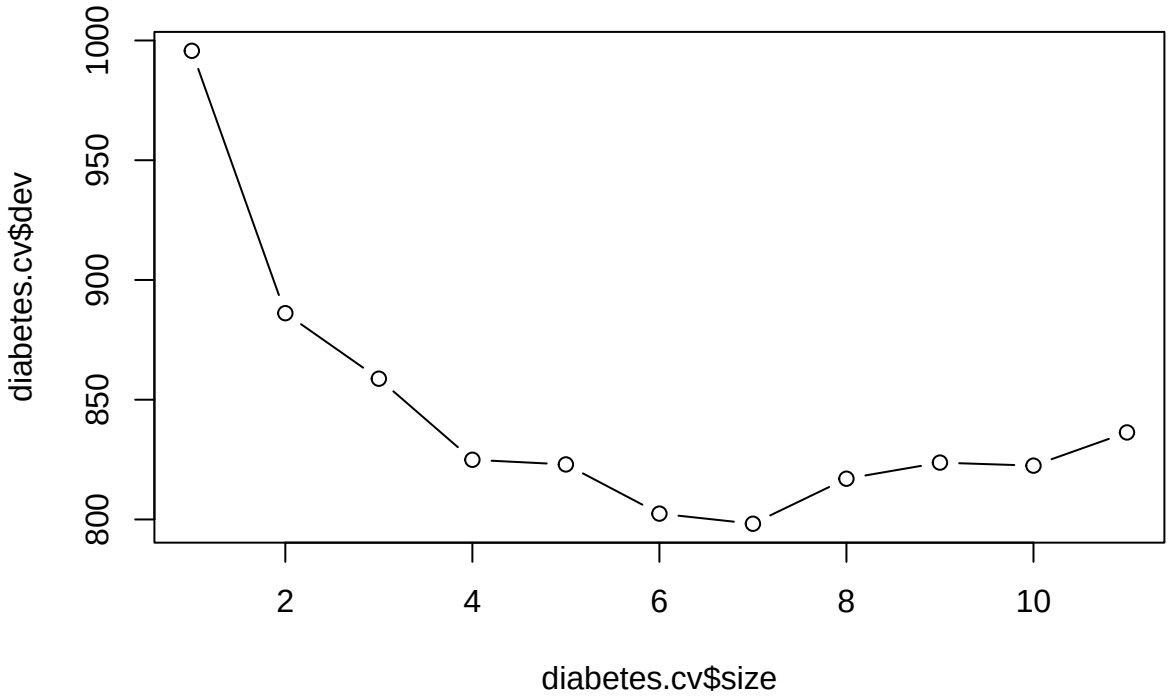


Figure 2: Plot the estimated test error rate

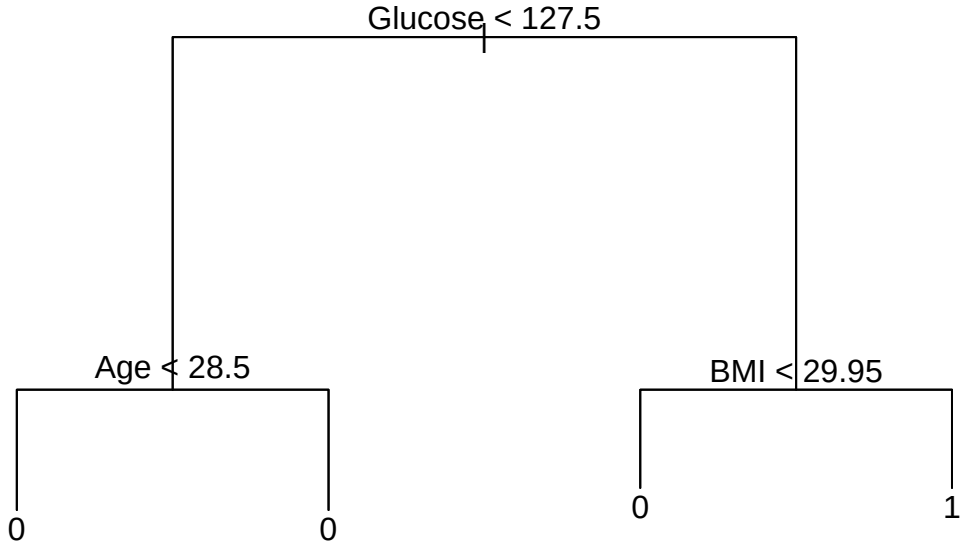


Figure 3: Classification prune Tree for cancer data

```
##
##       0   1
##   0 443 118
##   1  57 150
```

```
miss.classification_rate_b=(118+57)/768
miss.classification_rate_b
```

```
## [1] 0.2278646
```

The pruned tree has four(4) terminal nodes(Figure 2) and the actual used variable in tree construction are “Glucose”, “Age”, “BMI”(See Figure 3) and are seems to be most important predictors. Using LOOCV method the miss classification error rate for pruned tree with four terminal nodes is 0.2278646.

## I use a bagging approach to analyze the data with $B = 1000$.

```
##                                0          1 MeanDecreaseAccuracy
## Pregnancies                30.769223 -2.5893091            28.505350
## Glucose                    63.692092 55.2226627            83.099899
## BloodPressure               9.996635  0.5889513             8.103179
## SkinThickness              12.883482 -3.3737147             8.839756
## Insulin                    19.232023 -5.9998613            12.237177
## BMI                        28.848724 33.5914919            44.120193
## DiabetesPedigreeFunction   12.866859  6.6259408            13.895832
## Age                        32.010407  9.9845558            33.934909
##                           MeanDecreaseGini
## Pregnancies                       23.17991
## Glucose                          113.67043
## BloodPressure                     30.12617
## SkinThickness                     17.02358
## Insulin                           18.27193
## BMI                               59.66284
## DiabetesPedigreeFunction          44.78770
## Age                               41.94566
```

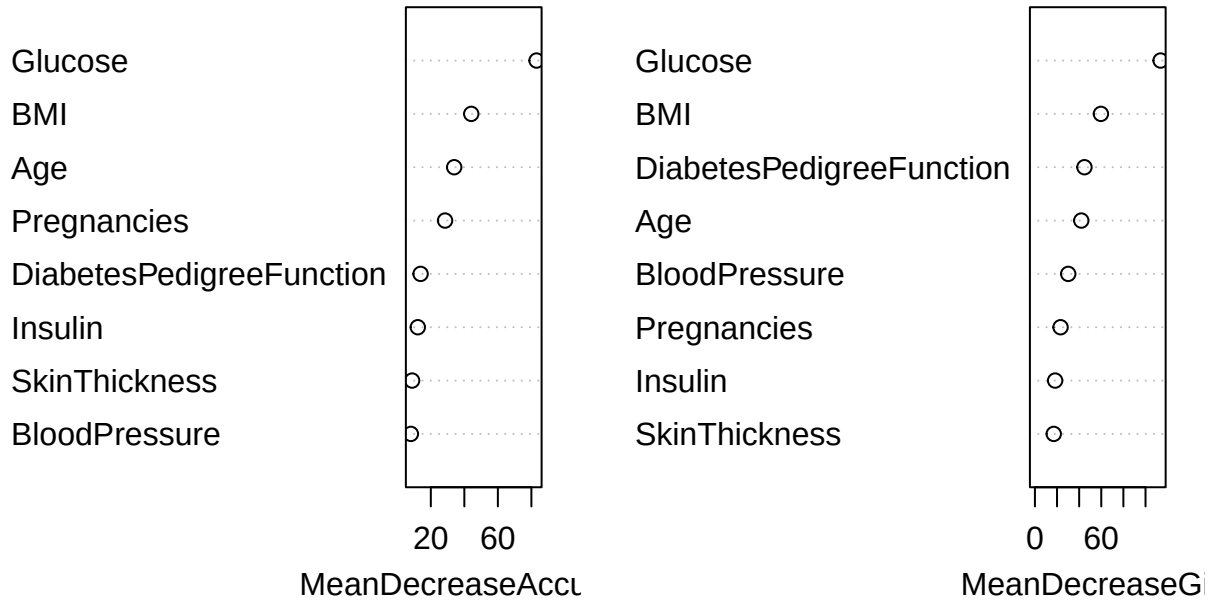


Figure 4: Variable importance measure for each predictor (Bagging)

```
miss.classification_rate_c=(103+78)/768
miss.classification_rate_c
```

```
## [1] 0.2356771
```

Using bagging approach with $B = 1000$, the Node purity plot (Figure 4) shows that the variables "Glucose" and "BMI are the most important predictors.

And the misclassification error rate using LOOCV method is 0.2356771.

### Use a random forest approach to analyze the data with $B = 1000$ and $m \approx p/3$.

```
##                                0          1 MeanDecreaseAccuracy
## Pregnancies              25.085449 -1.0099025            22.973021
## Glucose                  52.895991 54.5718785            71.488969
## BloodPressure             7.470102 -0.5869646             5.352824
## SkinThickness             6.663836  0.5214837             6.106488
## Insulin                  11.402725 -0.0228214             8.831540
## BMI                      23.830895 28.4773072            35.511897
## DiabetesPedigreeFunction 11.342835  4.4407736            11.531671
## Age                      28.006820 10.3261264            30.492379
##                          MeanDecreaseGini
## Pregnancies                      26.84585
## Glucose                          98.72177
## BloodPressure                    30.17176
## SkinThickness                    20.84183
## Insulin                          22.50911
## BMI                              59.17288
## DiabetesPedigreeFunction         43.21700
## Age                              47.11519
```

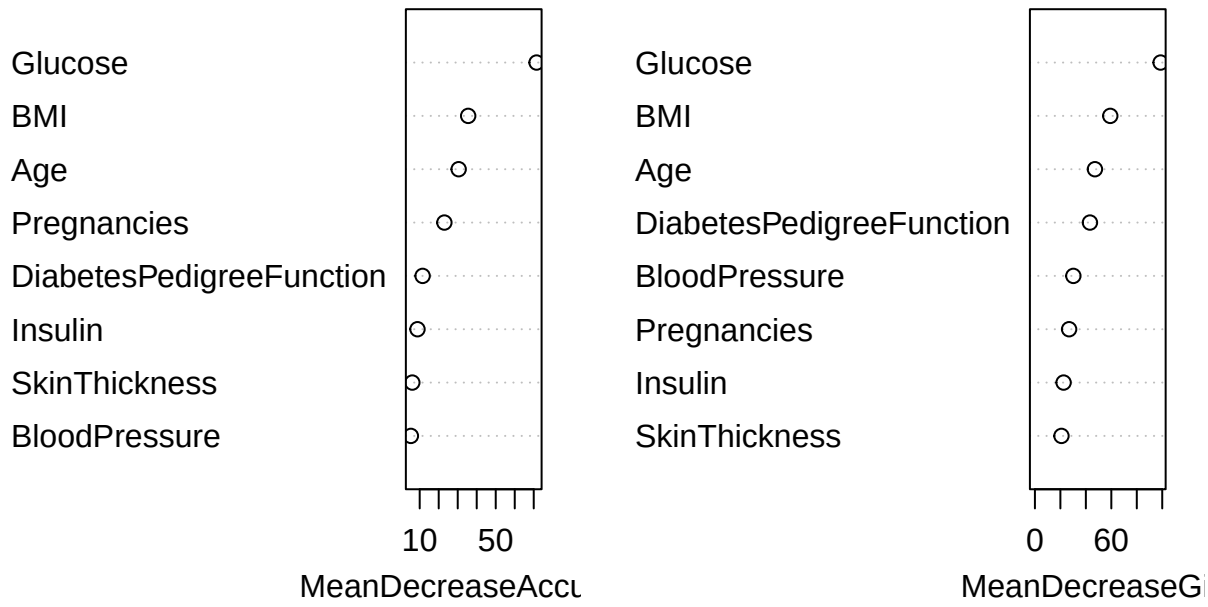


Figure 5: Variable importance measure for each predictor (Random forest)

```
##    Outcome
##       0   1
##   0 425 107
##   1  75 161
```

```
miss.classification_rate_d=(107+75)/768
miss.classification_rate_d
```

```
## [1] 0.2369792
```

Using random forest approach with $B = 1000$ the Node purity plot (Figure 5) shows that the variables "Glucose" and "BMI" are most important predictors.

And the miss classification error rate using LOOCV method is 0.2369792.

### Use a boosting approach to analyze the data with $mfinal = 1000$ and $d = 1$.

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

# Define the training control for cross-validation
train_control <- trainControl(method = "cv", number = 10)

# Create the model using a boosting algorithm
model <- train(Outcome ~ ., data = diabetes.data, method = "gbm",
               trControl = train_control,
               verbose = FALSE)

# Get the predictions on the training data using 10-fold cross-validation
predictions <- predict(model, newdata = diabetes.data)
```

```r
# Calculate the misclassification rate
misclassification_rate <- mean(predictions != diabetes.data$Outcome)
print("Misclassification Rate:")
```

```
## [1] "Misclassification Rate:"
```

```r
print(misclassification_rate)
```

```
## [1] 0.2070312
```

Using boosting approach with $mfinal = 1000$ and $d = 1$the miss classification error rate using 10-fold cross validation method is 0.2070312.

## Compare the results from the various methods.

| | un-pruned tree | pruned tree | bagging | random-forest | boosting |
|---|---|---|---|---|---|
| Miss classification error rate | 0.2630208 | 0.2278646 | 0.2356771 | 0.2369792 | 0.2070312 |

Table 1: Miss classification error rate for different approches

When consider the four different approaches discussed above, un-pruned tree approach gives large Miss classification error rate(0.2630208) and boosting approach gives the small Miss classification error rate(0.2070312). So boosting approach should be recommended to analyse diabetes data.