Tree Analysis and Results

Another approach we could take for our data set is tree regression. We will use the `rpart` library to make a regression tree model of the data set and then visualize it with the help of `rpart.plot`.

Data Preparation

Since our target variable, "Outcome", is a binary variable, we will make it of type factor. For better tree visualization, we will make the following changes to our variable names: `BloodPressure` will be renamed to `BP`, `SkinThickness` will be renamed to `ST`, and `DiabetesPredegreeFunction`will be renamed to `DPF` .

Now we split up the data to training and testing sets. This can allow us to train part of the data on tree regression and test its accuracy. After performing a 80/20 split, we can fit our model on the training data set. There is also an imbalance in our `Outcome` variable.

```
## Training
```

```
##
##   0   1
## 400 215
```

```
## Testing
```

```
##
##   0   1
## 100  53
```

We will use undersampling to reduce bias towards the majority.

```
## Training
```

```
##
##   0   1
## 215 215
```

```
## Testing
```

```
##
##  0  1
## 53 53
```

Tree Analysis

Since our target variable, `Outcome`, is a binary variable that classifies whether the patient has diabetes or not, we will create a classification tree on the training data set.

```
## n= 430
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
```

```
##
##  1) root 430 215 0 (0.50000000 0.50000000)
##    2) Glucose< 127.5 240  71 0 (0.70416667 0.29583333)
##      4) Age< 28.5 123  19 0 (0.84552846 0.15447154) *
##      5) Age>=28.5 117  52 0 (0.55555556 0.44444444)
##       10) Glucose< 99.5 39   7 0 (0.82051282 0.17948718) *
##       11) Glucose>=99.5 78  33 1 (0.42307692 0.57692308)
##         22) DPF< 0.2235 19   5 0 (0.73684211 0.26315789) *
##         23) DPF>=0.2235 59  19 1 (0.32203390 0.67796610)
##           46) Glucose>=119.5 20   9 0 (0.55000000 0.45000000)
##             92) Age< 41.5 13   4 0 (0.69230769 0.30769231) *
##             93) Age>=41.5 7   2 1 (0.28571429 0.71428571) *
##           47) Glucose< 119.5 39   8 1 (0.20512821 0.79487179) *
##    3) Glucose>=127.5 190  46 1 (0.24210526 0.75789474)
##      6) BMI< 29.8 42  18 0 (0.57142857 0.42857143)
##       12) Pregnancies< 1.5 13   1 0 (0.92307692 0.07692308) *
##       13) Pregnancies>=1.5 29  12 1 (0.41379310 0.58620690)
##         26) BP>=79 9   2 0 (0.77777778 0.22222222) *
##         27) BP< 79 20   5 1 (0.25000000 0.75000000) *
##      7) BMI>=29.8 148  22 1 (0.14864865 0.85135135) *
```

Tree plot

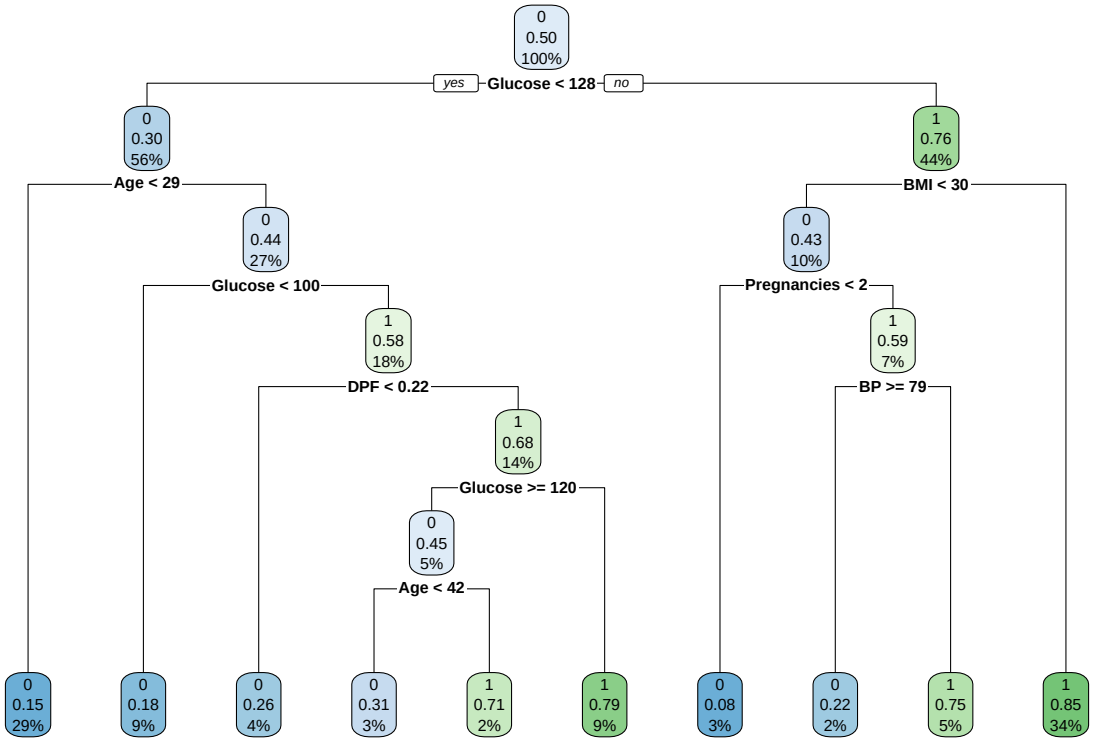

Figure 1: Tree Diagram

There are a total of 9 splits in this tree. The first split is on the Glucose level of the patient. If glucose exceeds 128, we go to the right, node 3. If we look at the tree details from above, we can see that node three has 190 cases and 46 of those would be miss classified as 0, or non diabetic. This can be compared to the accuracy of 76% being correctly classified as diabetic because $\frac{46}{190} = 24\%$ is the remainder that was miss classified. The percentage shown at the bottom of the third node, 44%, is how much of the original observations fit the case. In other words, $\frac{190}{430}$ or 44% of the total patients observed had glucose exceeding 128.

Interestingly, Insulin levels measured in the 2 hour serum test is not used shown on the tree to help determine whether the patient has diabetes or not. With an initial assumption this can be surprising because with diabetes, insulin levels are not high enough to manage the blood sugar in the body. A possible reasoning behind insulin levels not being used as splitting criterion could be that the patients who are diabetic already are using some medication to help maintain a healthy amount of insulin in their body. This would lead to normal/similar insulin levels among non diabetic patients and diabetic patients with treatment to have similar insulin levels.

With the help of the variable importance method in rpart library, we can see the level of importance for each predictor. Notice that although Insulin was not used as a criterion for branching the classification tree, it still has a higher level of importance than Blood Pressure, Pregancies, Diabetes Predegree Function, and Skin Thickness.

Tree Predictions and Results

Now that we have a tree model for classifying whether a patient has diabetes or not, we can look the accuracy of our model through the test data set.There are 14 false negatives and 15 false positives, giving us a $\frac{(14+15)}{(39+15+14+38)} = \frac{29}{106} = 27\%$ error rate and a 73% accuracy.

Tree Pruning

The accuracy of this current tree is pretty good. However, we can see if pruning the tree will lead to a lower error rate and more accurate predictions. We first check the `cp`, complexity parameter, of the shorter trees. Then we select the lowest `cp` with the lowest relative error.

```
##
## Classification tree:
## rpart(formula = Outcome ~ ., data = diabetes.training, method = "class")
##
## Variables actually used in tree construction:
## [1] Age         BMI         BP          DPF         Glucose     Pregnancies
##
## Root node error: 215/430 = 0.5
##
## n= 430
##
##         CP nsplit rel error  xerror     xstd
## 1 0.455814      0   1.00000 1.07442 0.048091
## 2 0.027907      1   0.54419 0.56744 0.043479
## 3 0.023256      5   0.41860 0.52093 0.042330
## 4 0.011628      7   0.37209 0.48372 0.041300
## 5 0.010000      9   0.34884 0.48372 0.041300

## CP with lowest cross validation error: 0.01162791
```

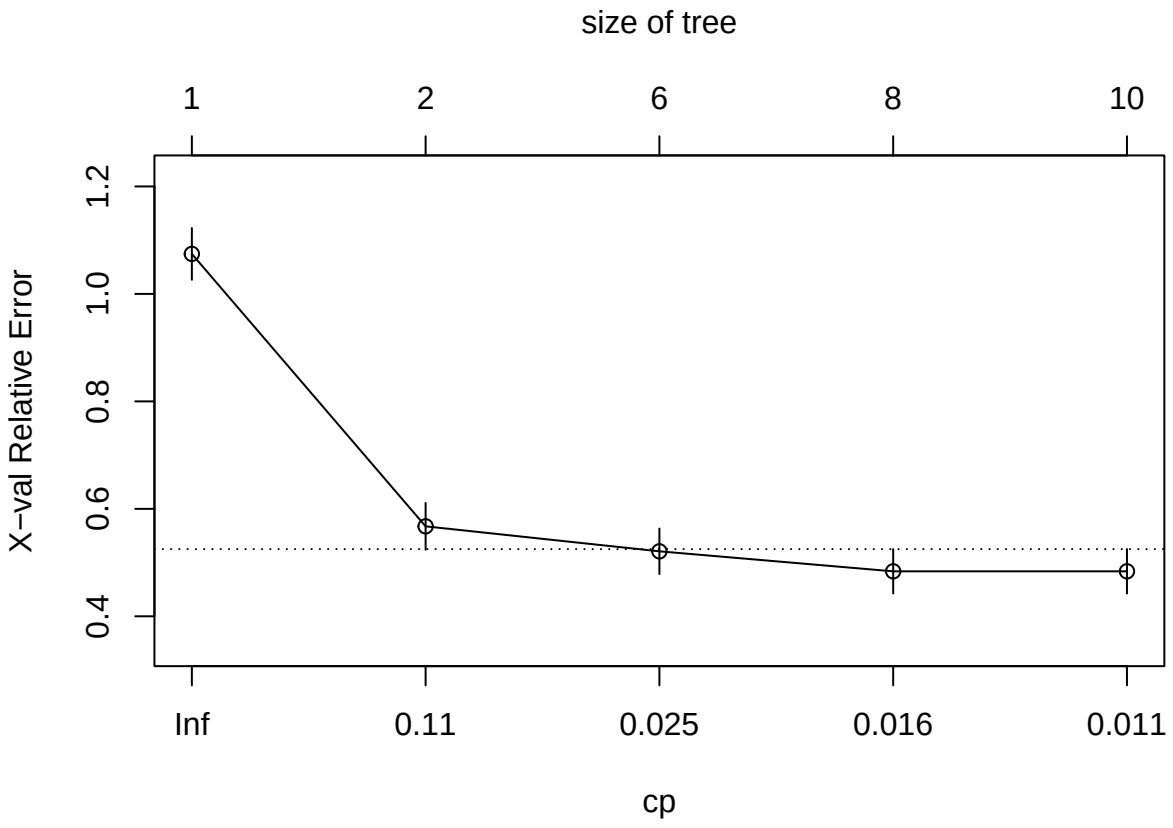

Figure 2: Cross Validation Error vs. Complexity Parameeter

Now we make a tree with the lowest `cp` and look at the new pruned tree. There 7 splits in the pruned tree, two less from the original.

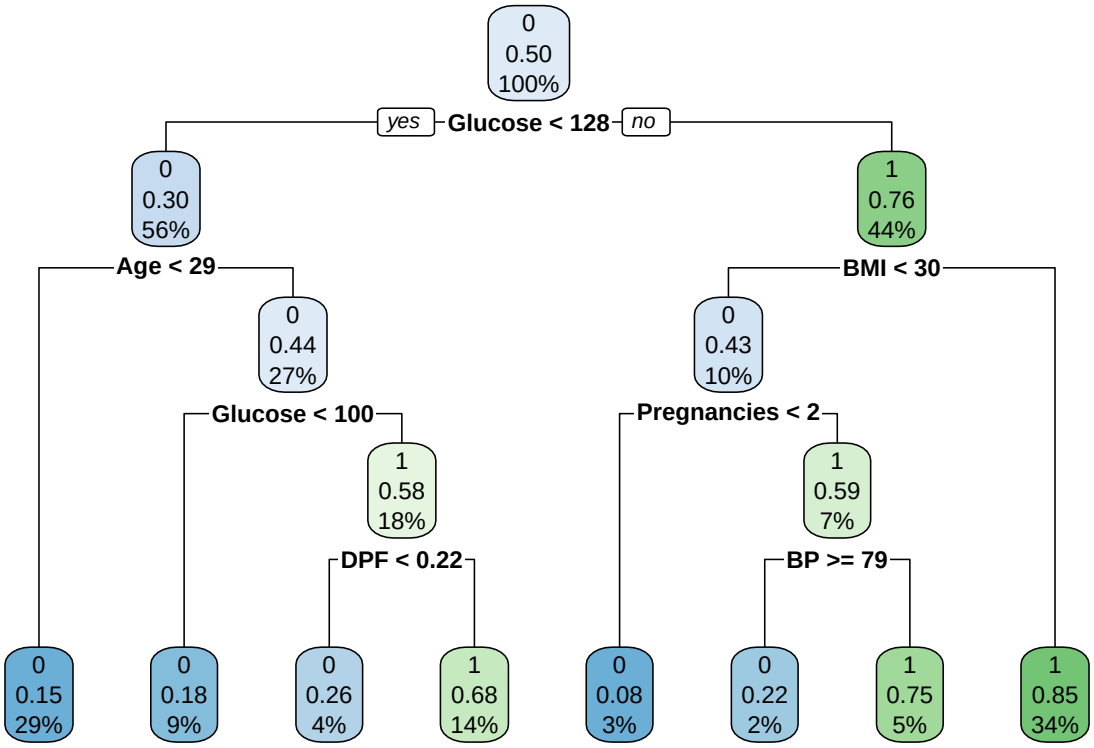

Figure 3: Pruned Tree

The accuracy actually increased by a percent. The total false negatives in the pruned tree is 14 and 12 false negatives giving us a $\frac{(14+12)}{(39+12+14+41)} = \frac{26}{106} = 25\%$ error and 75% accuracy.

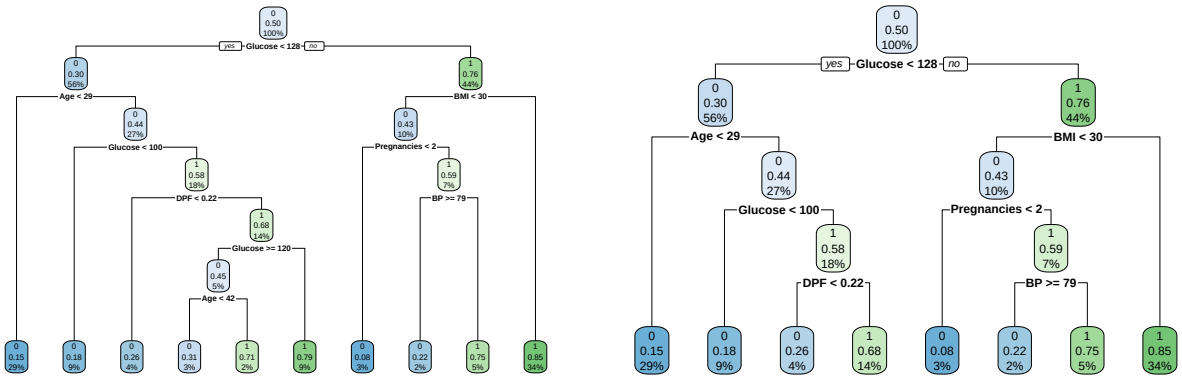

Figure 4: Original Tree (left) vs. Pruned Tree (right)

There was removal of unnecessary splits and roots which we can see when comparing the pruned tree with the original. There are two more splits that occur after $DPF < 0.63$ in the original split. The pruned tree removes them and increases the accuracy of the model predictions.