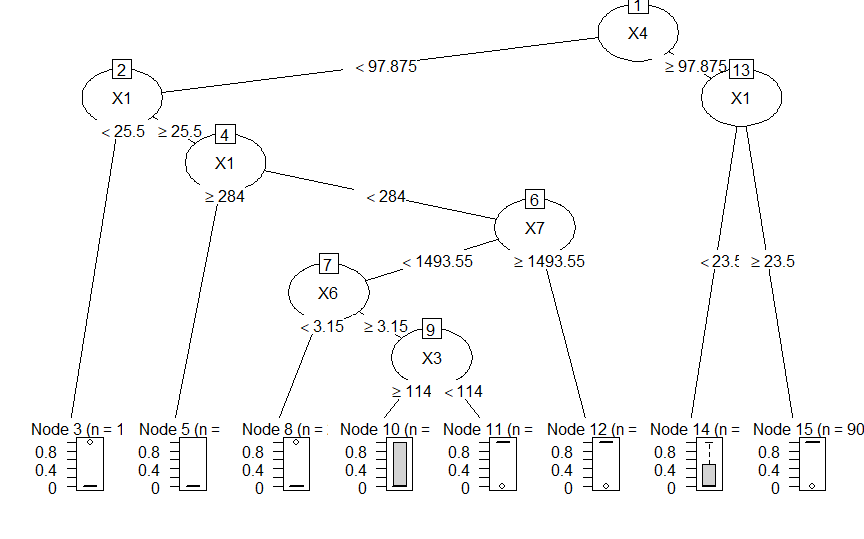
**Project Report for CA 2**

By Arambakam Mukesh – 19301497

The code for this can be found on my GitHub, please find the link to the repo below.

<https://github.com/mukeshmk/r-project>

Decision tree using X data without pruning



Decision tree using X data with pruning

A screenshot of a cell phone

Description automatically generated

Decision tree using Y data without pruning

A close up of a map

Description automatically generated

Decision tree using Y data with pruning

A close up of a map

Description automatically generated

Decision tree using X & Y data without pruning

A screenshot of a cell phone

Description automatically generated

Decision tree using X & Y data with pruning

A screenshot of a cell phone

Description automatically generated

Decision tree using X data for group 0 without pruning

A close up of a map

Description automatically generated

Decision tree using X data for group 0 with pruning

A screenshot of a social media post

Description automatically generated

Decision tree using Y data for group 0 without pruning

A screenshot of a social media post

Description automatically generated

Decision tree using Y data for group 0 with pruning

A screenshot of a social media post

Description automatically generated

Decision tree using X & Y data for group 0 without Pruning

A close up of a map

Description automatically generated

Decision tree using X & Y data for group 0 with Pruning

A close up of a map

Description automatically generated

Decision tree using X data for group 1 without Pruning

A close up of a map

Description automatically generated

Decision tree using X data for group 1 with Pruning

A screenshot of a cell phone

Description automatically generated

Decision tree using Y data for group 1 without Pruning

A close up of a map

Description automatically generated

Decision tree using Y data for group 1 with Pruning

A screenshot of a cell phone

Description automatically generated

Decision tree using X & Y data for group 1 without Pruning

A close up of a map

Description automatically generated

Decision tree using X & Y data for group 1 with Pruning

A close up of a map

Description automatically generated

**Accuracies for the above generated graphs:**

> print(paste("Accuracy for Decision tree using X data: ",accuracy\_Test\_X))

[1] "Accuracy for Decision tree using X data: 0.753378378378378"

> print(paste("Accuracy for Decision tree using Y data: ",accuracy\_Test\_Y))

[1] "Accuracy for Decision tree using Y data: 0.766891891891892"

> print(paste("Accuracy for Decision tree using X & Y data: ",accuracy\_Test\_XY))

[1] "Accuracy for Decision tree using X & Y data: 0.753378378378378"

> print(paste("Accuracy for Decision tree using X data for group 0: ",accuracy\_Test\_X\_G0))

[1] "Accuracy for Decision tree using X data for group 0: 0.736486486486487"

> print(paste("Accuracy for Decision tree using Y data for group 0: ",accuracy\_Test\_Y\_G0))

[1] "Accuracy for Decision tree using Y data for group 0: 0.668918918918919"

> print(paste("Accuracy for Decision tree using X & Y data for group 0: ",accuracy\_Test\_XY\_G0))

[1] "Accuracy for Decision tree using X & Y data for group 0: 0.72972972972973"

> print(paste("Accuracy for Decision tree using X data for group 1: ",accuracy\_Test\_X\_G1))

[1] "Accuracy for Decision tree using X data for group 1: 0.753378378378378"

> print(paste("Accuracy for Decision tree using Y data for group 1: ",accuracy\_Test\_Y\_G1))

[1] "Accuracy for Decision tree using Y data for group 1: 0.712837837837838"

> print(paste("Accuracy for Decision tree using X & Y data for group 1: ",accuracy\_Test\_XY\_G1))

[1] "Accuracy for Decision tree using X & Y data for group 1: 0.760135135135135"

**Conclusion:**

The best Decision Tree generated is the DT generated over **Entire Dataset with the Predictors X1-X7 and Y1-Y7 over Group 1** as it has the highest **accuracy of 76%** - see **Plot 18** for the DT. The Decision Tree’s summary, indicating the splits can be found in the next page.

Though the other DT’s also have an accuracy of 75.6% - the DT in Plot 18 is better because it gives a consistent accuracy of 76% when tested with different `rpart` configurations like minsplit, minbucket and maxdepth, more over it has a **Low Variance** between different decision tree when run with different seed values for the data and different rpart configuration. We can also clearly say that X’s are the features with **High Information Gain** compared to Y, and also splitting the data into Group’s do not show a significant improvement accuracy.

The code for this can be found on my GitHub, please find the link to the repo below.

<https://github.com/mukeshmk/r-project>

> summary(DT\_Model\_XY\_G1\_pruned)

Call:

rpart(formula = Response ~ ., data = input\_XY\_G1, method = "class",

control = rpart.control(minsplit = 30, minbucket = 10, maxdepth = 8))

n= 200

CP nsplit rel error xerror xstd

1 0.54081633 0 1.0000000 1.0612245 0.07209601

2 0.06122449 1 0.4591837 0.5102041 0.06248698

3 0.02040816 2 0.3979592 0.4897959 0.06163130

Variable importance

X4 X5 X7 X1 Y5 Y4 Y7

24 17 16 13 13 12 4

Node number 1: 200 observations, complexity param=0.5408163

predicted class=0 expected loss=0.49 P(node) =1

class counts: 102 98

probabilities: 0.510 0.490

left son=2 (117 obs) right son=3 (83 obs)

Primary splits:

X4 < 97.875 to the left, improve=30.76630, (0 missing)

X5 < 32.39 to the left, improve=30.29774, (0 missing)

Y5 < 0.5 to the left, improve=25.97221, (0 missing)

Y1 < 0.5 to the left, improve=24.09052, (0 missing)

X1 < 33.5 to the left, improve=23.23900, (0 missing)

Surrogate splits:

X5 < 44.545 to the left, agree=0.855, adj=0.651, (0 split)

Y4 < 0.5 to the left, agree=0.795, adj=0.506, (0 split)

X1 < 80 to the left, agree=0.785, adj=0.482, (0 split)

Y5 < 0.5 to the left, agree=0.780, adj=0.470, (0 split)

X7 < 1576.35 to the left, agree=0.735, adj=0.361, (0 split)

Node number 2: 117 observations

predicted class=0 expected loss=0.2564103 P(node) =0.585

class counts: 87 30

probabilities: 0.744 0.256

Node number 3: 83 observations, complexity param=0.06122449

predicted class=1 expected loss=0.1807229 P(node) =0.415

class counts: 15 68

probabilities: 0.181 0.819

left son=6 (16 obs) right son=7 (67 obs)

Primary splits:

X7 < 605.4 to the left, improve=10.180930, (0 missing)

X5 < 32.39 to the left, improve= 6.437137, (0 missing)

X1 < 23.5 to the left, improve= 6.131738, (0 missing)

Y5 < 0.5 to the left, improve= 4.552823, (0 missing)

X4 < 183.75 to the left, improve= 4.137415, (0 missing)

Surrogate splits:

Y7 < 0.5 to the left, agree=0.904, adj=0.500, (0 split)

X1 < 21 to the left, agree=0.855, adj=0.250, (0 split)

X5 < 25.895 to the left, agree=0.855, adj=0.250, (0 split)

Y5 < 0.5 to the left, agree=0.843, adj=0.188, (0 split)

X2 < 3.5 to the left, agree=0.819, adj=0.062, (0 split)

Node number 6: 16 observations

predicted class=0 expected loss=0.3125 P(node) =0.08

class counts: 11 5

probabilities: 0.688 0.312

Node number 7: 67 observations

predicted class=1 expected loss=0.05970149 P(node) =0.335

class counts: 4 63

probabilities: 0.060 0.940