Homework 2

Team: Aishwarya Deshpande
Tanvi Anandpara
Ting Lang

Q1.

(a) We can use chi-square test to find a relationship between the variables before creating a decision tree.

chi_Sex<-table(income_data$Annual_Income, income_data$Sex) # p-value low so they are independent
chisq.test(chi_Sex)

```
Pearson's Chi-squared test

data:  chi_Sex
X-squared = 37.142, df = 8, p-value = 1.084e-05
```

p-value is low. Similarly, creating chi-square test for all variables.
All p-values are low so the null hypothesis that variables are independent is rejected.
From this we cannot isolate one particular variable which can be used to predict Annual_Income.

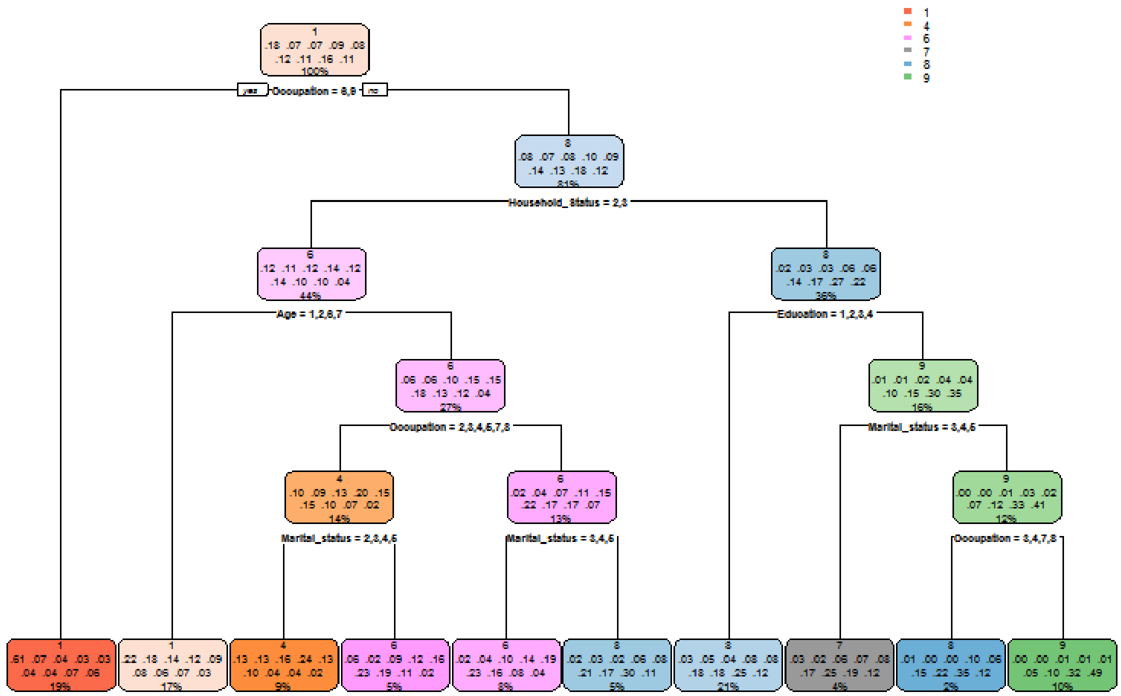
(b)

```
> income_rpart
n= 4125

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 4125 3376 1 (0.18 0.073 0.071 0.091 0.082 0.12 0.11 0.16 0.11)
   2) Occupation=6,9 804  312 1 (0.61 0.073 0.042 0.035 0.03 0.045 0.041 0.06
6 0.056) *
   3) Occupation=1,2,3,4,5,7,8 3321 2731 8 (0.077 0.073 0.078 0.1 0.095 0.14
0.13 0.18 0.12)
     6) Household_Status=2,3 1820 1560 6 (0.12 0.11 0.12 0.14 0.12 0.14 0.1 0
.1 0.041)
      12) Age=1,2,6,7 704  548 1 (0.22 0.18 0.14 0.12 0.087 0.081 0.055 0.074
0.034) *
      13) Age=3,4,5 1116  913 6 (0.064 0.064 0.1 0.15 0.15 0.18 0.13 0.12 0.0
45)
        26) Occupation=2,3,4,5,7,8 579  466 4 (0.1 0.09 0.13 0.2 0.15 0.15 0.
097 0.069 0.022)
          52) Marital_status=2,3,4,5 364  276 4 (0.13 0.13 0.16 0.24 0.13 0.0
99 0.041 0.044 0.022) *
          53) Marital_status=1 215  166 6 (0.056 0.023 0.088 0.12 0.16 0.23 0
.19 0.11 0.023) *
        27) Occupation=1 537  419 6 (0.02 0.035 0.069 0.11 0.15 0.22 0.17 0.1
7 0.069)
          54) Marital_status=3,4,5 324  251 6 (0.022 0.04 0.099 0.14 0.19 0.2
3 0.16 0.083 0.043) *
          55) Marital_status=1,2 213  150 8 (0.019 0.028 0.023 0.056 0.085 0.
21 0.17 0.3 0.11) *
     7) Household_Status=1 1501 1093 8 (0.02 0.031 0.029 0.061 0.061 0.14 0.1
7 0.27 0.22)
      14) Education=1,2,3,4 850  636 8 (0.028 0.049 0.036 0.079 0.08 0.18 0.1
8 0.25 0.12) *
      15) Education=5,6 651  426 9 (0.0092 0.0061 0.018 0.037 0.037 0.095 0.1
5 0.3 0.35)
```

```
          30) Marital_status=3,4,5 154  116 7 (0.032 0.019 0.058 0.071 0.084 0.
17 0.25 0.19 0.12) *
          31) Marital_status=1,2 497  291 9 (0.002 0.002 0.006 0.026 0.022 0.07
2 0.12 0.33 0.41)
            62) Occupation=3,4,7,8 103   67 8 (0.0097 0 0 0.097 0.058 0.15 0.22
0.35 0.12) *
            63) Occupation=1,2,5 394  200 9 (0 0.0025 0.0076 0.0076 0.013 0.053
0.099 0.32 0.49) *
> rpart.plot(income_rpart)
```



There are 10 leaves in this decision tree.

```
> income_rpart$frame
                var    n   wt  dev yval   complexity ncompete nsurrogate
1        Occupation 4125 4125 3376    1 0.0986374408        4          4
2            <leaf>  804  804  312    1 0.0000000000        0          0
3  Household_Status 3321 3321 2731    8 0.0262144550        4          5
6               Age 1820 1820 1560    6 0.0262144550        4          5
12           <leaf>  704  704  548    1 0.0011848341        0          0
13       Occupation 1116 1116  913    6 0.0082938389        4          5
26   Marital_status  579  579  466    4 0.0071090047        4          4
52           <leaf>  364  364  276    4 0.0038507109        0          0
53           <leaf>  215  215  166    6 0.0029620853        0          0
27   Marital_status  537  537  419    6 0.0053317536        4          5
54           <leaf>  324  324  251    6 0.0014810427        0          0
55           <leaf>  213  213  150    8 0.0002962085        0          0
7         Education 1501 1501 1093    8 0.0091824645        4          5
14           <leaf>  850  850  636    8 0.0041469194        0          0
15   Marital_status  651  651  426    9 0.0063684834        4          5
30           <leaf>  154  154  116    7 0.0017772512        0          0
```

```
31       Occupation  497  497  291    9 0.0063684834        4           1
62            <leaf>  103  103   67    8 0.0000000000        0           0
63            <leaf>  394  394  200    9 0.0000000000        0           0
```

(c)

```
Variable importance
            Occupation        Household_Status                     Age
                    32                      18                      16
             Education          Marital_status            Dual_Incomes
                    10                       9                       7
           Type_Of_Home     Person_In_Household Person_In_Household_U18
                     4                       2                       1
```

According to the variable importance table in the summary function, we can see that the variables Occupation, Household_Status and Age play a major role in predicting the income.

(d)
Rule 1: If the occupation of a person is 8,9 (retired or unemployed), their household income will be in category 1 i.e. less than $10,000

Rule 2: If the person is the owner of the house, then they will not be in class 1 of annual income.

(e)

A surrogate split is a mimic or substitute for a variable that has been split.

```
Node number 1: 4125 observations,    complexity param=0.09863744

  Surrogate splits:
      Age                     splits as  LRRRRRR,    agree=0.864, adj=0.305, 
(0 split)
      Household_Status        splits as  RRL,        agree=0.856, adj=0.259, 
(0 split)
      Education               splits as  LLRRRR,     agree=0.848, adj=0.218, 
(0 split)
      Person_In_Household_U18 splits as  RRRRRRLL-R, agree=0.806, adj=0.002, 
(0 split)
```

Surrogate split of Age can be used instead of Occupation at the first node split since it will be 86.4% as accurate.

(f)

```
> table(predict(income_rpart,newdata = Testdata, type = "class"), Testdata$An
nual_Income, dnn = c("Predicted", "Actual"))
```

Predicted \ Actual	1	2	3	4	5	6	7	8	9
1	464	137	103	87	73	71	69	70	46
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	29	71	73	125	116	161	105	112	47
7	0	0	0	0	0	0	0	0	0
8	13	14	14	27	30	113	126	216	118
9	0	1	2	3	2	13	22	45	77

This gives us 32.84% accuracy in predictions.

(g)

People who have jobs and have education upto 1-3 years of college and are also home owners are likely to have high incomes.
Also, people with higher education and are either married or living together are likely to have higher income.

(h)

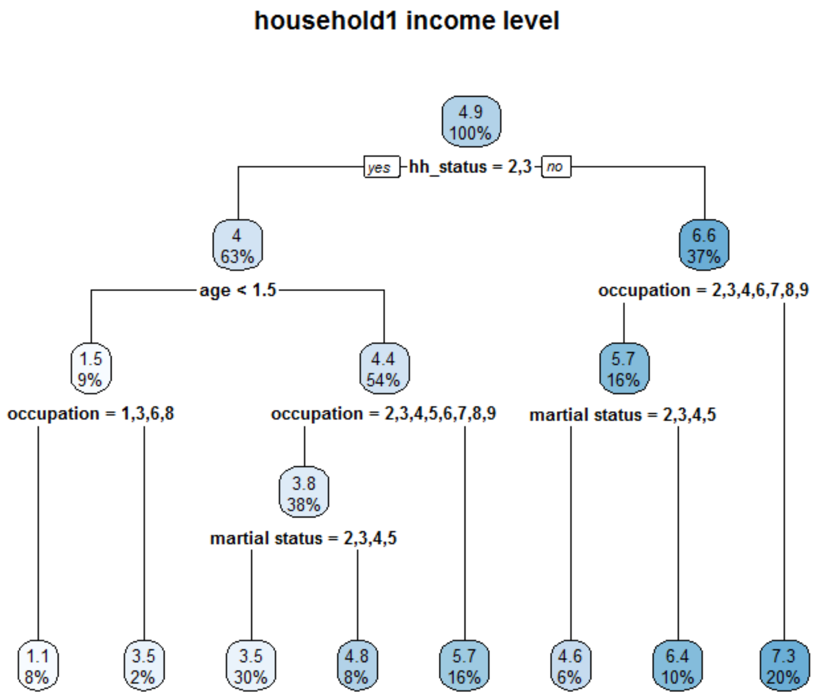
```
Variable importance
           Occupation        Household_Status                      Age
                   30                      16                       15
            Education          Marital_status             Dual_Incomes
                   10                       9                        8
           House_type     People_In_Household People_In_Household_U18
                    4                       3                        1
                 Var5                    Var2                     Var8
                    1                       1                        1
                 Var7                   Var10
                    1                       1
```

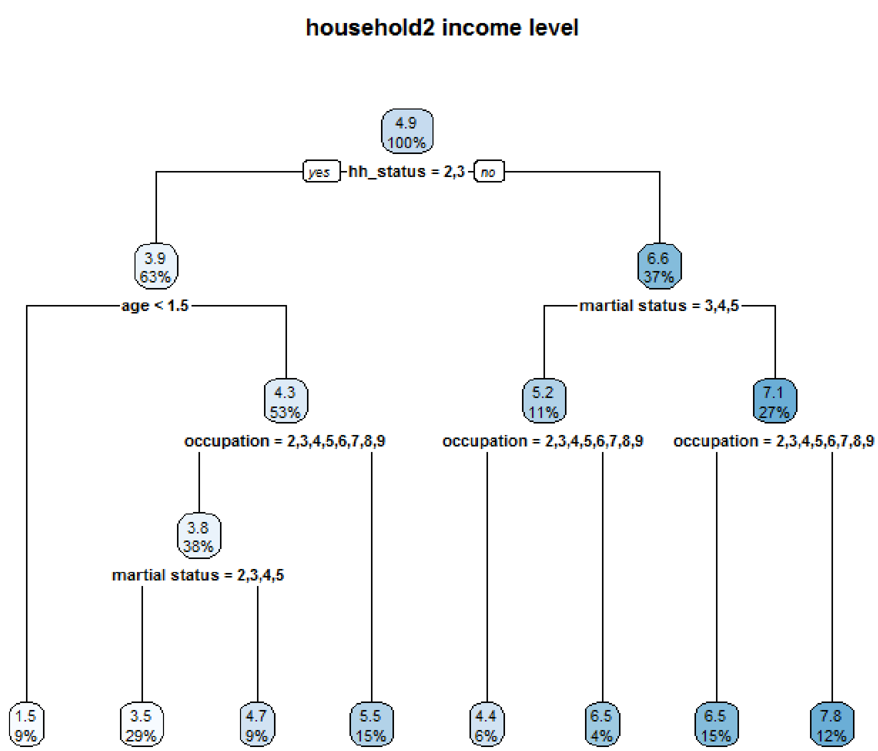
We can see that the importance of variables in both the models is similar. Occupation is the highest importance variable in order to predict income data.

Predicted \ Actual	1	2	3	4	5	6	7	8	9
1	314	58	45	27	30	18	28	40	31
2	14	39	18	18	6	8	7	4	2
3	7	8	23	7	6	6	5	2	1
4	13	16	21	53	29	16	8	5	4
5	4	8	8	5	15	3	12	7	4
6	13	9	20	29	30	77	36	26	16
7	0	8	14	15	24	30	51	17	20
8	6	6	14	22	19	53	69	171	67
9	0	0	1	2	4	7	14	29	72

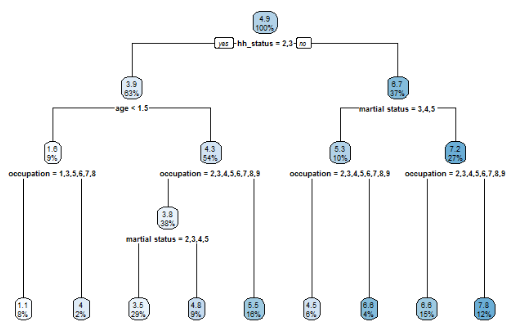
Also the accuracy of predictions in this decision tree is 41% which is a lot higher than the first decision tree.

(i)

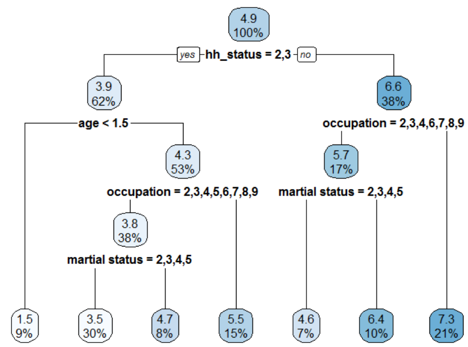

household1 income level
4.9
100%
yes
hh_status = 2,3
no
4
63%
6.6
37%
age < 1.5
occupation = 2,3,4,6,7,8,9
1.5
9%
4.4
54%
5.7
16%
occupation = 1,3,6,8
occupation = 2,3,4,5,6,7,8,9
martial status = 2,3,4,5
3.8
38%
martial status = 2,3,4,5
1.1
8%
3.5
2%
3.5
30%
4.8
8%
5.7
16%
4.6
6%
6.4
10%
7.3
20%


household2 income level
4.9
100%
yes
hh_status = 2,3
no
3.9
63%
6.6
37%
age < 1.5
martial status = 3,4,5
4.3
53%
5.2
11%
7.1
27%
occupation = 2,3,4,5,6,7,8,9
occupation = 2,3,4,5,6,7,8,9
occupation = 2,3,4,5,6,7,8,9
3.8
38%
martial status = 2,3,4,5
1.5
9%
3.5
29%
4.7
9%
5.5
15%
4.4
6%
6.5
4%
6.5
15%
7.8
12%

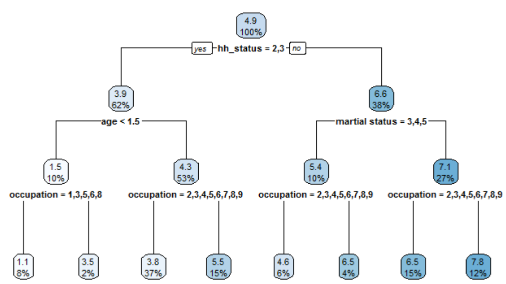
household6 income level



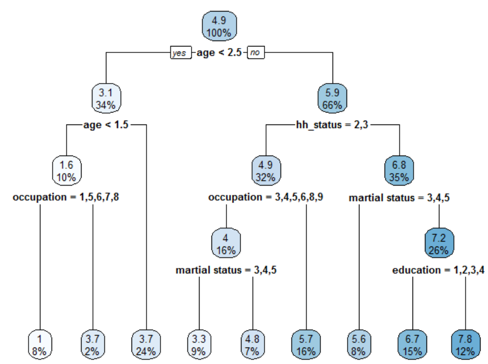
household7 income level



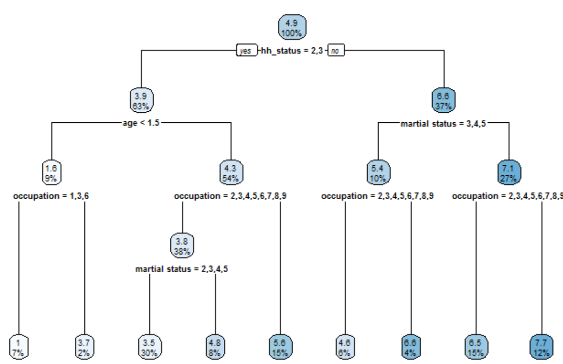
household8 income level



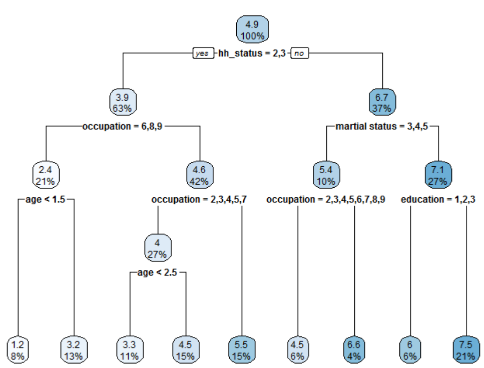
household4 income level



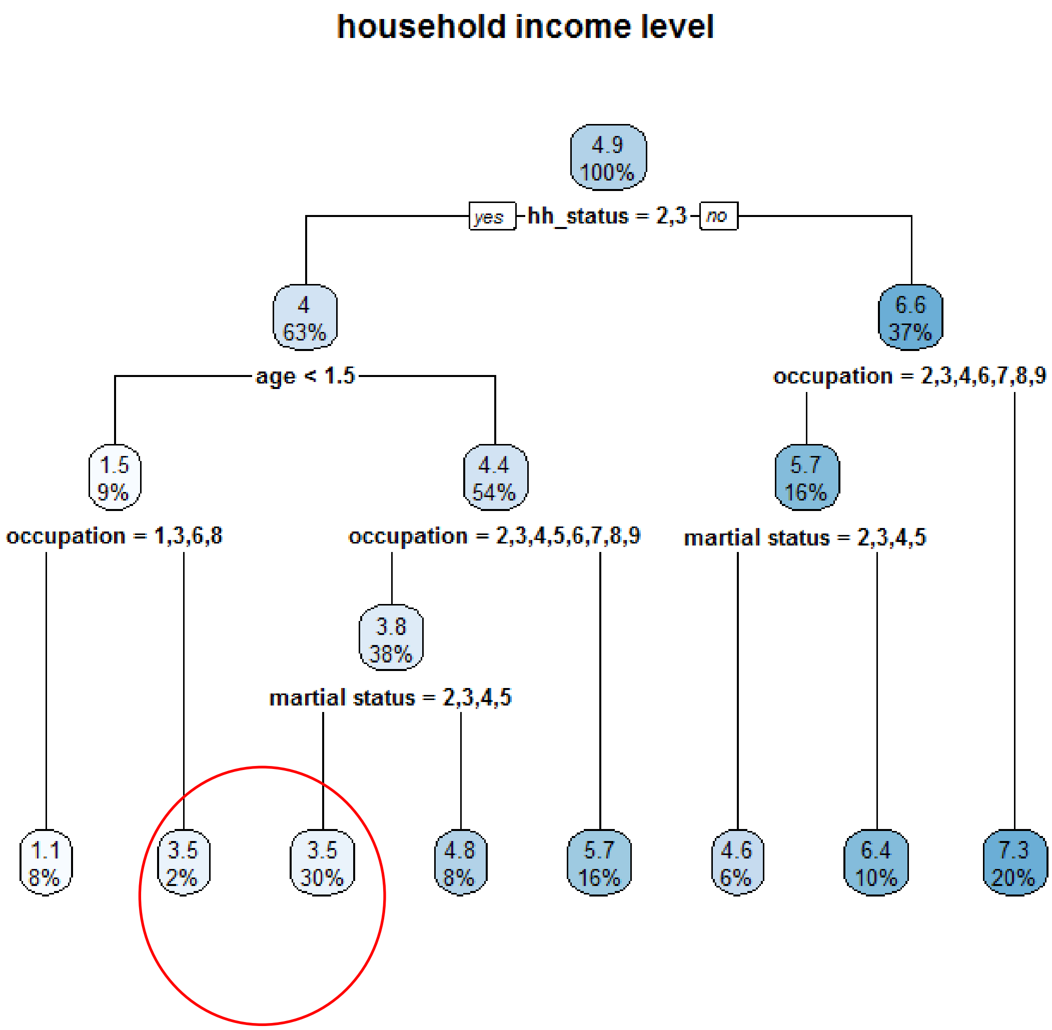
household3 income level

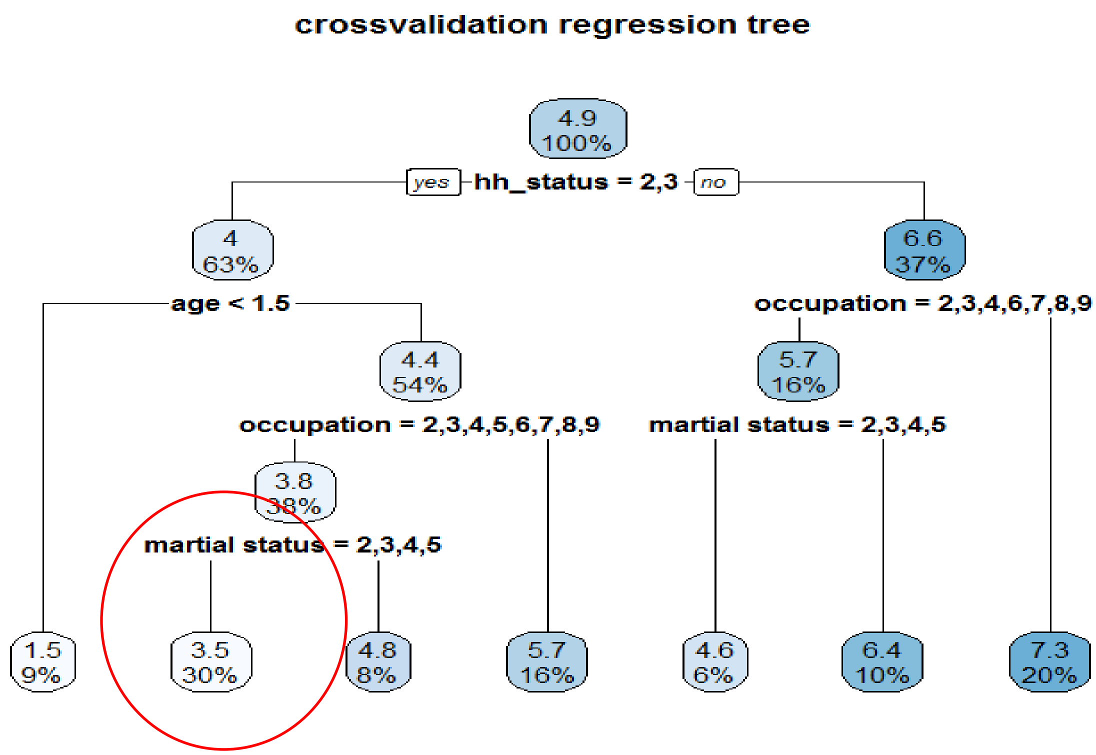


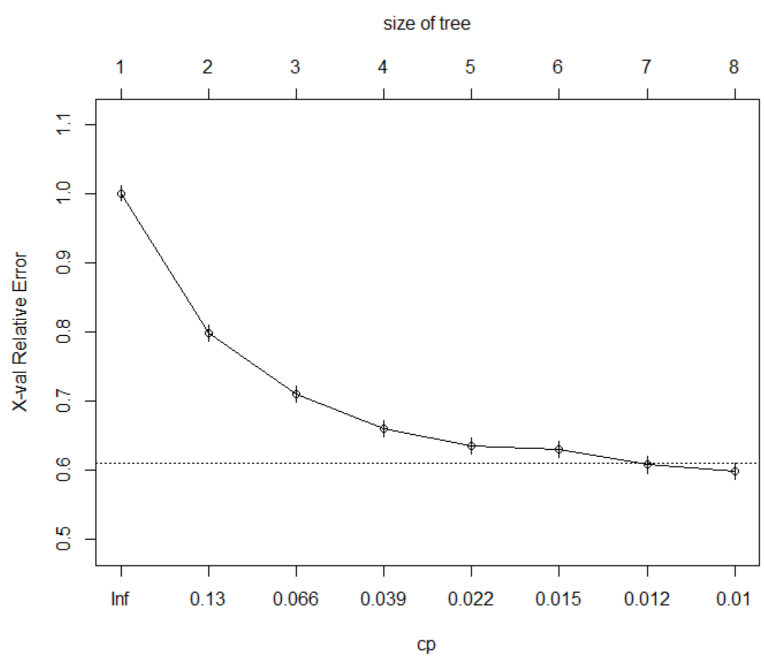
household5 income level



According to the graphs from the 8 training dataset. There is no significant difference between the estimation but only the depth of the tree would be different. In addition, the importance of variable for prediction is still the same.


household income level
4.9
100%
yes
hh_status = 2,3
no
4
63%
6.6
37%
age < 1.5
occupation = 2,3,4,6,7,8,9
1.5
9%
4.4
54%
5.7
16%
occupation = 1,3,6,8
occupation = 2,3,4,5,6,7,8,9
martial status = 2,3,4,5
3.8
38%
martial status = 2,3,4,5
1.1
8%
3.5
2%
3.5
30%
4.8
8%
5.7
16%
4.6
6%
6.4
10%
7.3
20%





1st decision tree:

For a larger tree that with a minimum pruning, we found that there are more leaves in this model without pruning.

2nd decision tree:

Since the cp table tells us that the optimum regression tree is when cp=0.01

To make sure we have 500 records of a parent node and 100 records for leaf nodes,

We setup the minisplit as 500 while minibucket= 100 for every leaf node, with cp value of 0.01

In terms of the improvement by using the pruning here, the minimum improvement for cross-validation tree is much higher than that of the least pruning one, which means that it's right to prune the tree.

```
    improve
 Min.    :0.01090
 1st Qu.:0.05247
 Median :0.16539
 Mean    :0.35086
 3rd Qu.:0.68336
 Max.    :0.95726
```

big.tree2(least pruning)

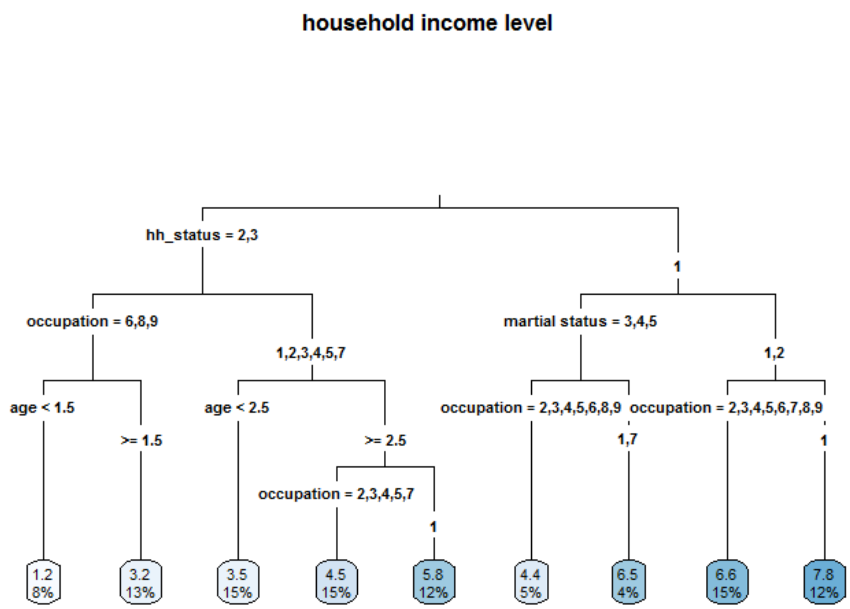
```
    improve
 Min.    :0.01413
 1st Qu.:0.05955
 Median :0.18014
 Mean    :0.35800
 3rd Qu.:0.67837
 Max.    :0.95726
```

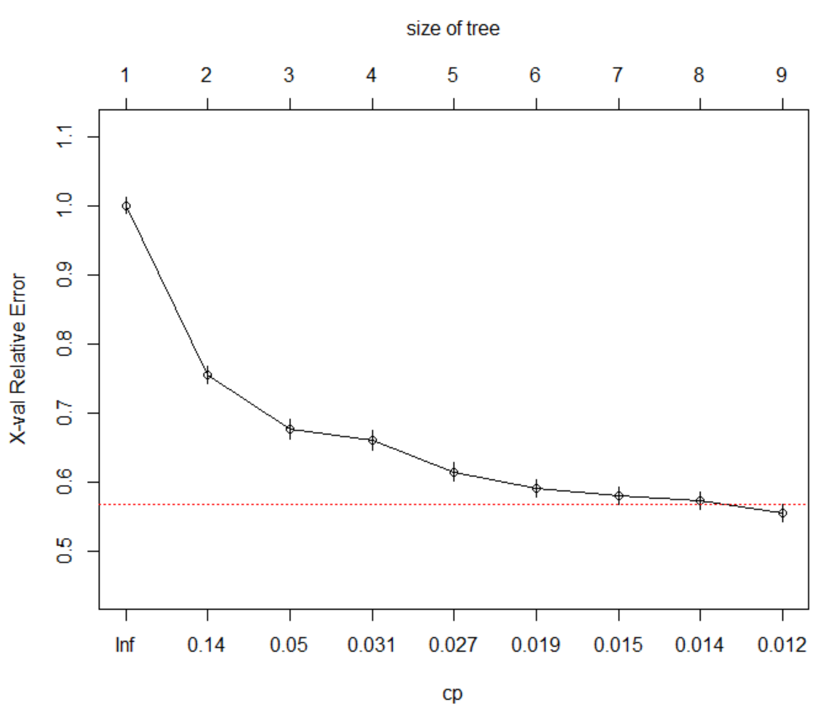
cv.finaltree(cp=0.01)

Obviously, the cross-validation decision tree at cp=0.01 is more accurate, compared to the less pruning ,by using cp= maximum.

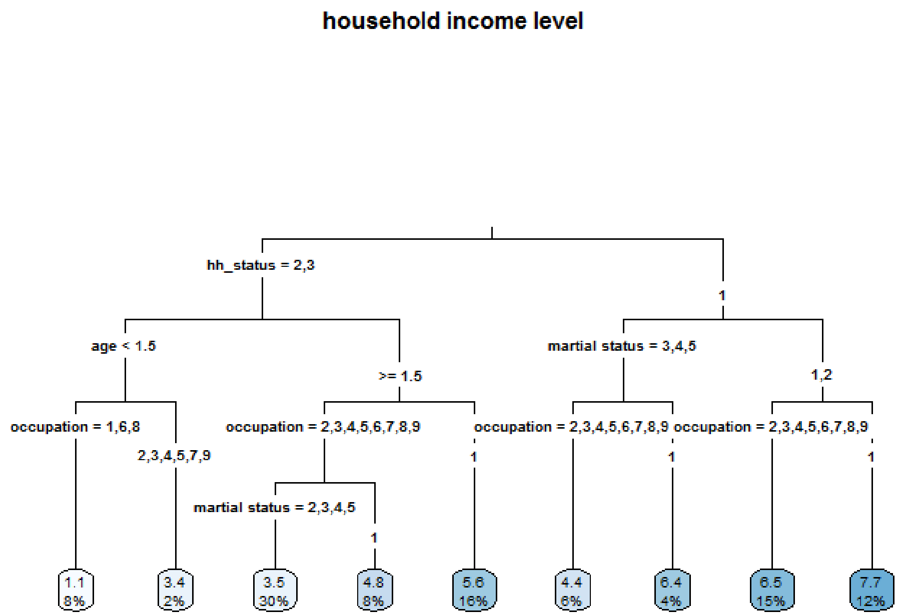
k) size of training set

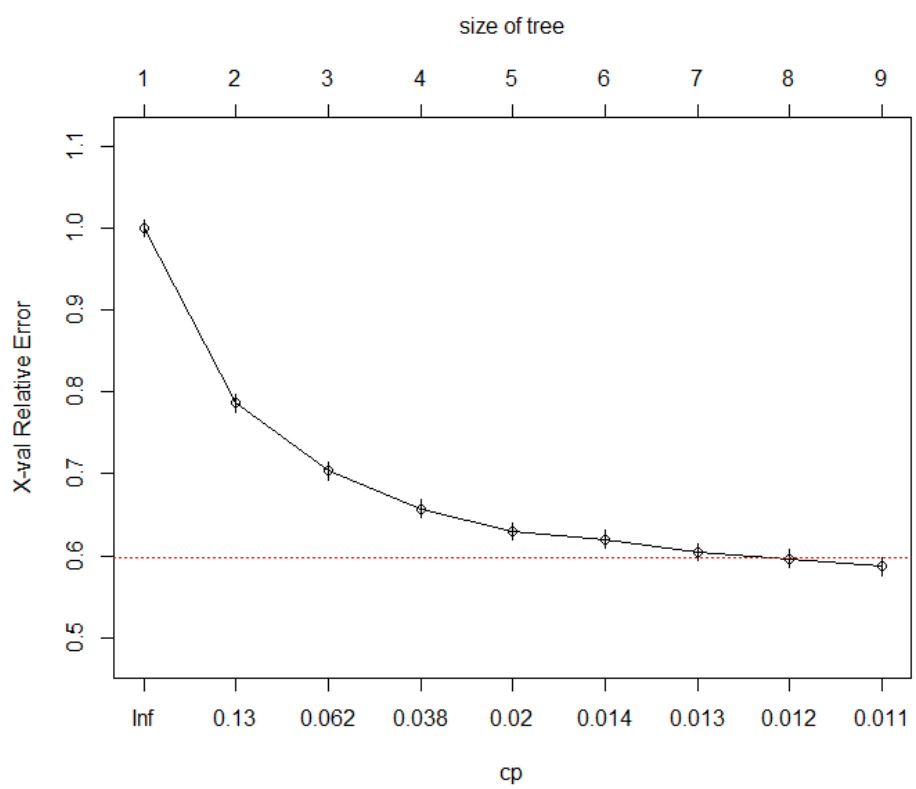
1. 50% training, 50% testing



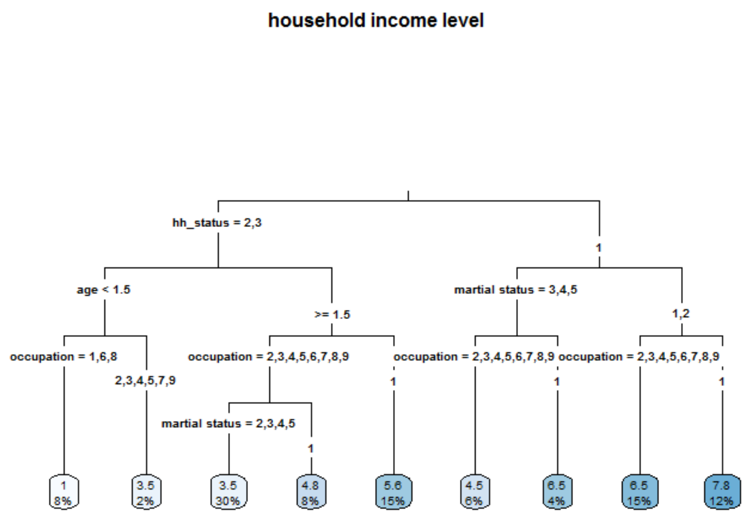


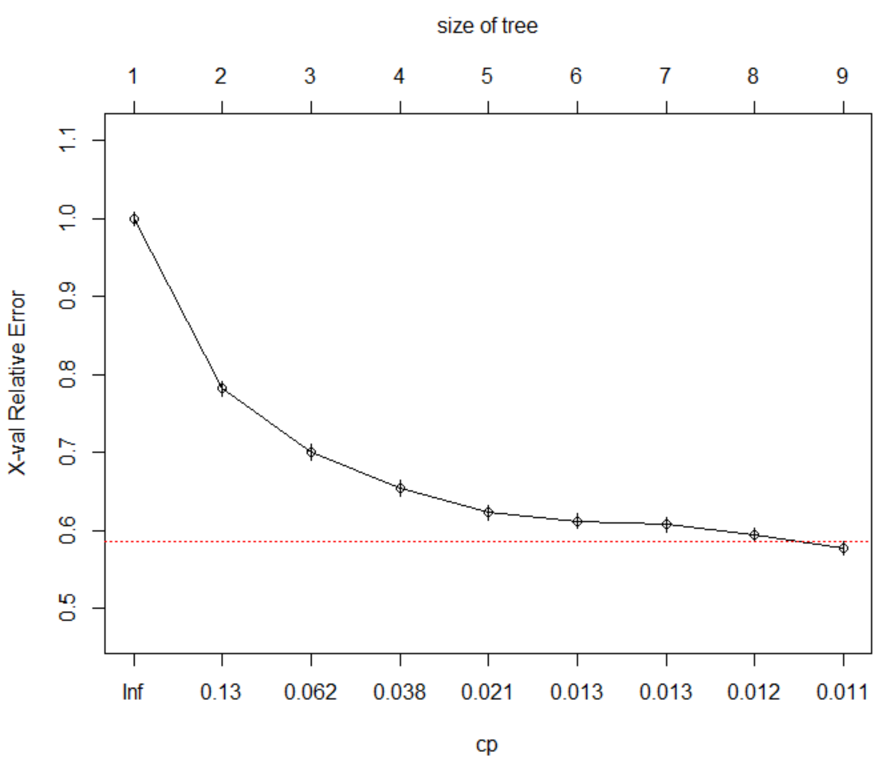
2. 70% training, 30% testing



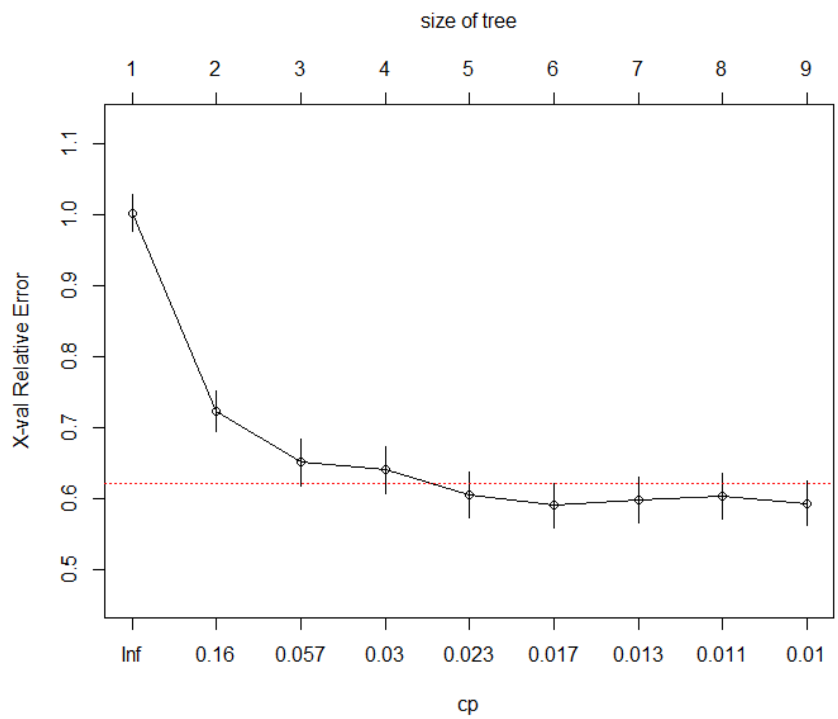


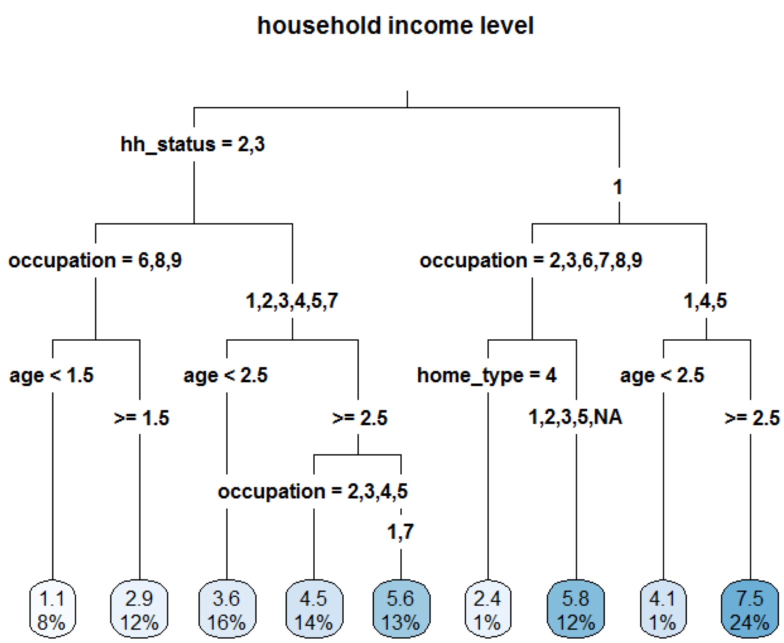
3. 90% training, 10% testing





4. 10% training, 90% testing





According to these 4 models, from training dataset size at 10% of income.data record to 90% of record, The model is getting better fit with smaller error rate. The larger the training data set we have, the smaller cp value we could have to achieve smaller error rate from the models, which is the difference between the models- varience is improved as the bias are smaller.

However, the error rate would not change that much as the training data set approaching the full size of the whole data. That is to say, increasing the size of training data set might not be helpful for model selection and it’s complexity would leave an overfitting.

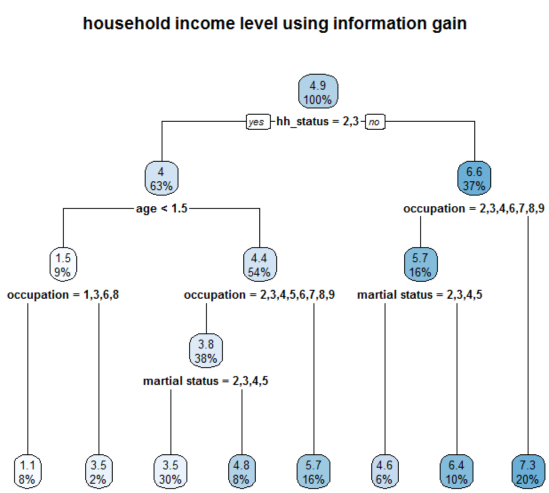
As a result, we prefer the model of training data at 70%

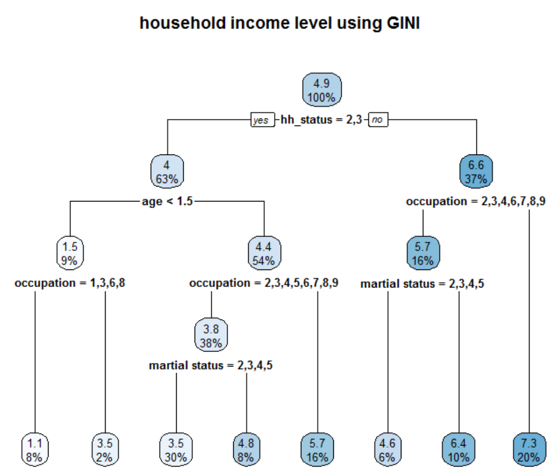
l)

Gini

```
          CP nsplit rel error    xerror       xstd
1 0.20703294      0 1.0000000 1.0007904 0.01067408
2 0.08517960      1 0.7929671 0.7937781 0.01135472
3 0.05098448      2 0.7077875 0.7219203 0.01195513
4 0.02930814      3 0.6568030 0.6597495 0.01196218
5 0.01660804      4 0.6274948 0.6353694 0.01157427
6 0.01416541      5 0.6108868 0.6239112 0.01156224
7 0.01086151      6 0.5967214 0.6068055 0.01178045
8 0.01000000      7 0.5858599 0.5977720 0.01136033
```

```
By going down to the whole tree using Gini index to split, we have lower
error rate from this model
```





Information gain:

```
          CP nsplit rel error    xerror       xstd
1 0.20703294      0 1.0000000 1.0007904 0.01067408
2 0.08517960      1 0.7929671 0.7937781 0.01135472
3 0.05098448      2 0.7077875 0.7219203 0.01195513
4 0.02930814      3 0.6568030 0.6597495 0.01196218
```

```
5 0.01660804      4 0.6274948 0.6353694 0.01157427
6 0.01416541      5 0.6108868 0.6239112 0.01156224
7 0.01086151      6 0.5967214 0.6068055 0.01178045
8 0.01000000      7 0.5858599 0.5977720 0.01136033
```

```
Variable importance
     hh_status           age     occupation martial status     dualincomes
            27            22             16             14              11
     education     home_type        hh_size
             7             3              1
```


```
Variable importance
     hh_status           age     occupation martial status     dualincomes
            27            22             16             14              11
     education     home_type        hh_size
             7             3              1
```

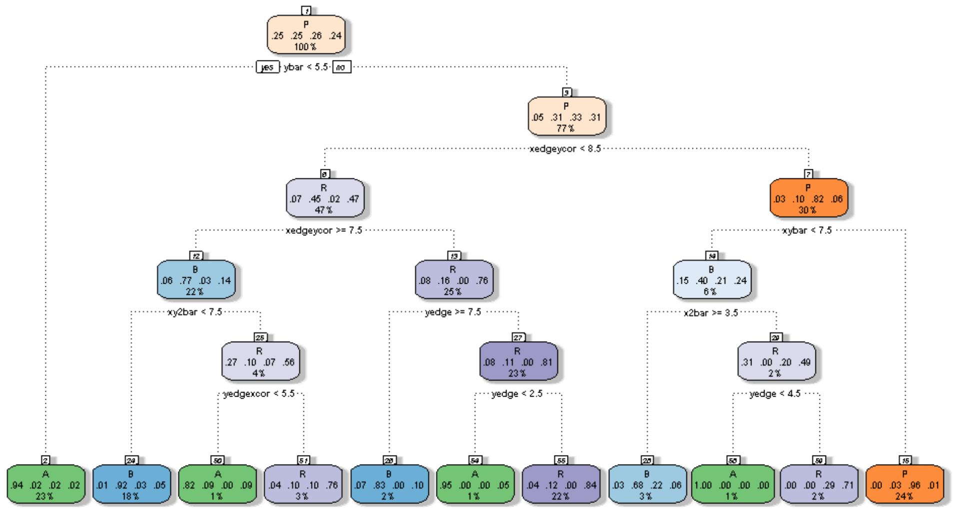

By comparing with these two method for splitting, there are no difference between the information gain and Gini index split method in this case.

For Gini, we use that to test impurity.

Q2.

1. <u>Decision Tree - dtree **Accuracy: 89.845**</u>

```
# Train a decision tree classifier using the rpart() function
dtree <- rpart(letter ~ ., data = abpr_train)
plot(dtree, uniform=TRUE, compress=TRUE, branch=0.5)
text(dtree, use.n=TRUE, cex=.8)
fancyRpartPlot(dtree)
letter_pred <- table(predict(dtree,type="class",newdata =
abpr_train),abpr_train$letter,dnn=c('Actual','Predicted'))
Accuracy_letter <- sum(diag(letter_pred)/sum(letter_pred))
View(Accuracy_letter)
```



Rattle 2017-Oct-04 12:13:02 Tanvi Anandpara

2. <u>Decision Tree - dtr1 **Accuracy: 97.54142**</u>

   # Decision Tree - 01

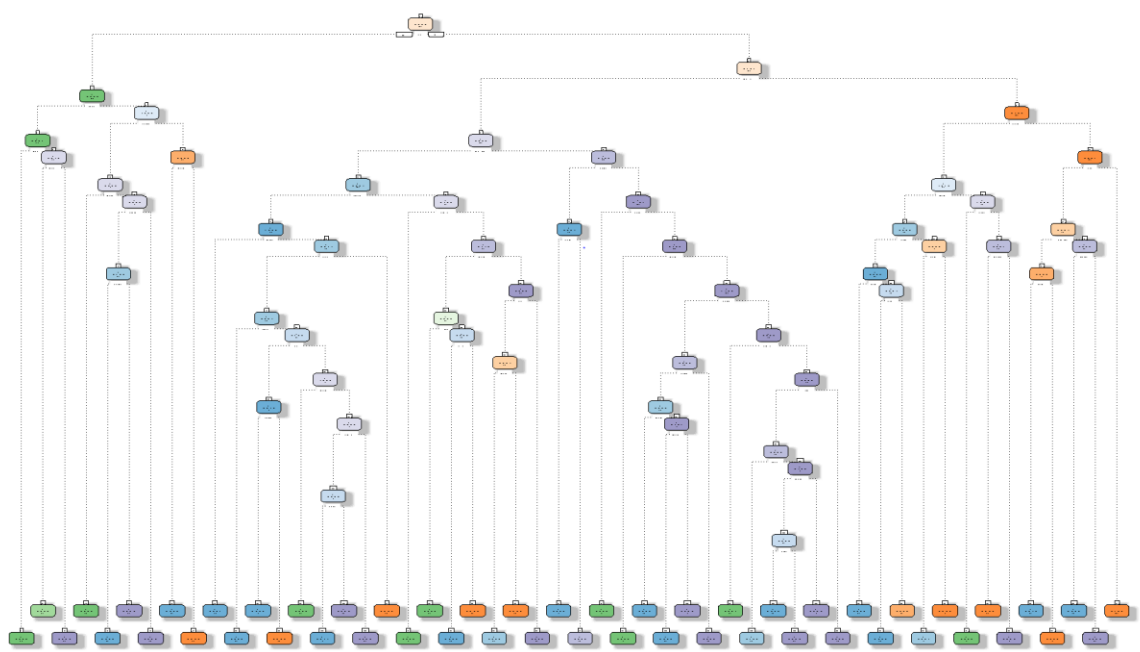
   dtr1 <- rpart(letter ~ ., data = abpr_train,method = "class", control = rpart.control(minsplit = 1, cp = 0.001))

   fancyRpartPlot(dtr1)

   letter_pred <- table(predict(dtr1,type="class",newdata = abpr_train),abpr_train$letter,dnn=c('Actual','Predicted'))

   Accuracy_letter <- sum(diag(letter_pred)/sum(letter_pred))

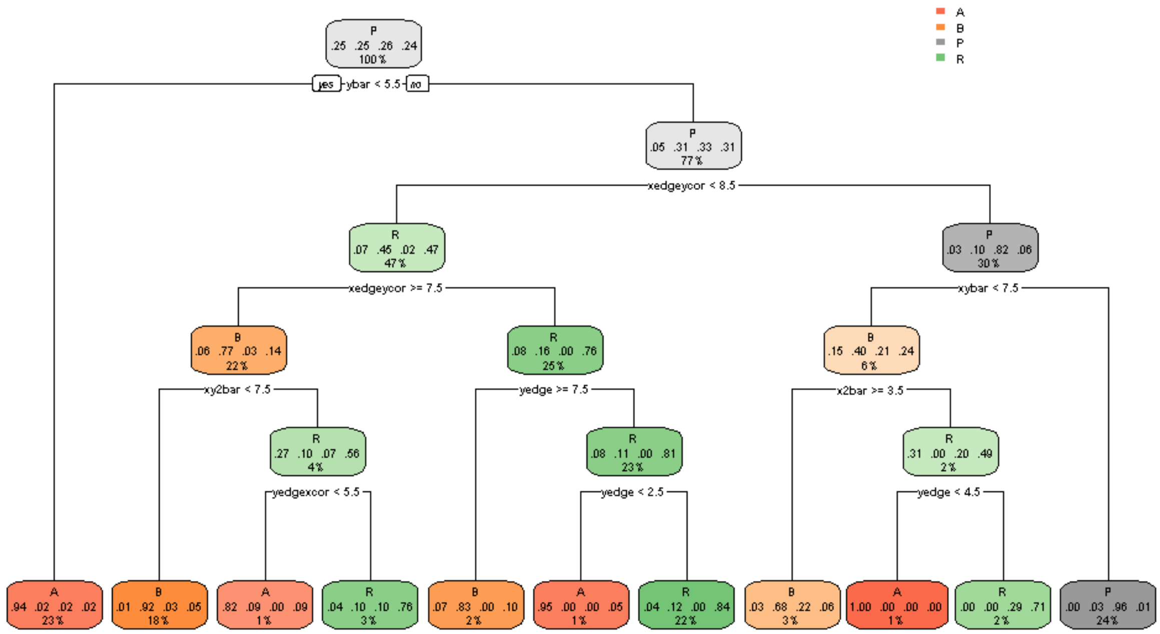
   View(Accuracy_letter)



Rattle 2017-Oct-04 13:33:39 Tanvi Anandpara

3. Decision Tree – letterGini **Accuracy: 88.99598**

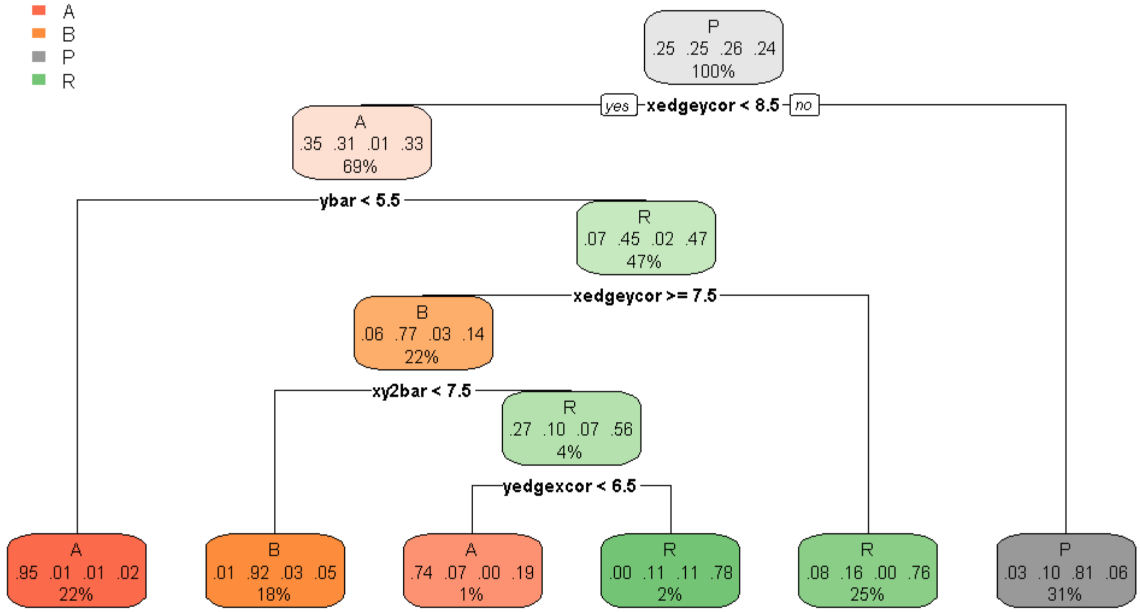
```
# Decision Tree - Gini
letterGini = rpart(letter ~ ., data = abpr_train, parms = list(split = 'gini'))
rpart.plot(letterGini)
Gini_predict <- table(predict(letterGini,type="class",newdata =
abpr_test),abpr_test$letter,dnn=c('Actual','Predicted'))
AccuracyGini <- sum(diag(Gini_predict)/sum(Gini_predict))
View(AccuracyGini)
```



4. Decision Tree – letterInfo **84.25703**

```
# Decision Tree - Information
letterInfo = rpart(letter ~ ., data = abpr_train, parms = list(split = 'information'))
rpart.plot(letterInfo)
Info_pred <- table(predict(letterInfo,type="class",newdata = abpr_test), abpr_test$letter,
dnn=c('Actual','Predicted'))
AccuracyInfo <- sum(diag(Info_pred)/sum(Info_pred))
```

View(AccuracyInfo)



5. <u>Random Forest - dforest **Accuracy: 99.51807**</u>

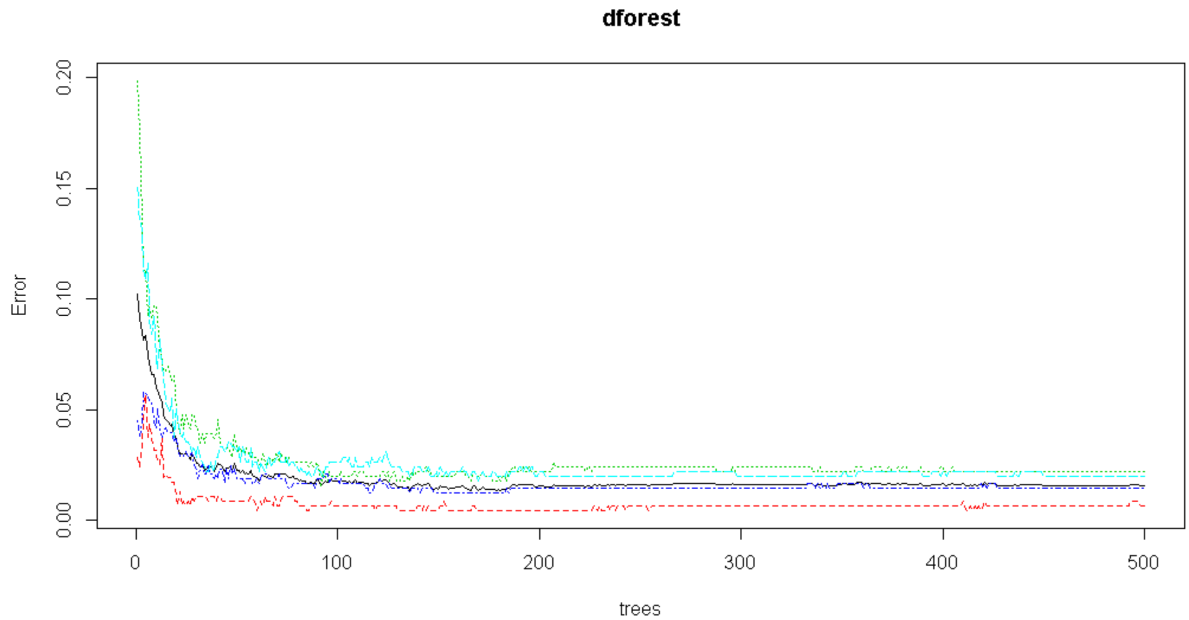
```
# Generate Random Forest
dforest <- randomForest(letter ~ ., data = abpr_train)
print(dforest)
attributes(dforest)
plot(dforest)

# error
dforest$err.rate
rf.lgnd <- if (is.null(dforest$abpr_test$err.rate)) {colnames(dforest$err.rate)} else
{colnames(dforest$abpr_test$err.rate)}
legend("top", cex =0.5, legend=rndF1.legend, lty=c(1,2,3), col=c(1,2,3), horiz=T)


#plot variable importance
varImpPlot(dforest)

### get accuracy of prediction
table(predict(dforest), abpr_train$letter)

abpr_Pred = predict(dforest, newdata = abpr_test)
t <- table(abpr_Pred, abpr_test$letter,dnn=c('Actual','Prediction'))
accuracy <- sum(diag(t)/sum(t))
View(accuracy)
```



Conclusion: When we compare Accuracy of all four decision trees modelled in R, we observe that Decision Tree - dtr1 gives the best result with 97.54142%. When we compare the results of decision trees against Random Forest model, we find that Random Forest gives 99.51807 accuracy. Hence, we select Random Forest model.