Statistical Modelling in R

Semester 3 –

Assignment 2

```
# Importing necessary libraries
library(rpart)
library(rpart.plot)
library(tidyverse)
library(caTools)
library(caret)
library(MASS)
```

1. Code:

```
# Creating sample to split the msc data into testing and training
data samp = sample.split(msc$Grade, 0.7)

train = msc %>% subset(samp == T) # Creating training dataset
test = msc %>% subset(samp == F) # Creating testing dataset

tree = rpart(Grade ~ ., data = train) # Defining the model with Grade as explanatory variable
rpart.plot(tree, box.palette = "RdYlGn", main = "Decision Tree") # Plotting the decision tree

# Assigning predicted values to a new column in the testing dataset.
test["Predicted.Grade"] = predict(tree, test, type = "class")

# Building confusion matrix and checking accuracy.
confusionMatrix(test$Predicted.Grade, test$Grade)
```

Output:

Confusion Matrix and Statistics

Reference
Prediction A B C D A
6 0 0 0
B 0 13 0 0
C 0 0 8 0
D 0 0 0 5

Overall Statistics

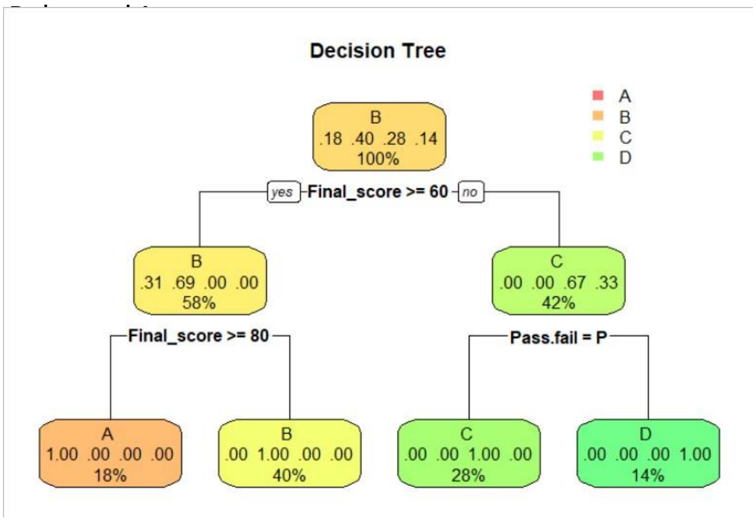
Accuracy : 1
95% CI : (0.8911, 1)
No Information Rate : 0.4062
P-Value [Acc > NIR] : 3.03e-13

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: A	Class: B	Class: C	Class: D
Sensitivity	1.0000	1.0000	1.00	1.0000
Specificity	1.0000	1.0000	1.00	1.0000
Pos Pred Value	1.0000	1.0000	1.00	1.0000
Neg Pred Value	1.0000	1.0000	1.00	1.0000
Prevalence	0.1875	0.4062	0.25	0.1562
Detection Rate	0.1875	0.4062	0.25	0.1562
Detection Prevalence	0.1875	0.4062	0.25	0.1562



2.

```
# Defining an initial model
model = glm(Final_score ~ ., data = train)

# Running step AIC to check for best model
stepAIC(
  model,
  scope = list(
    upper = ~ Exam1 * Exam2 * Exam3 * Exam4 * Final_score * Grade * Pass.fail, lower
    = ~1
  ),
  direction = "forward"
)

#Using model with lowest AIC from stepAIC.
final_model = glm(
  formula = Final_score ~ Exam1 + Exam2 + Exam3 + Exam4 + Grade +
    Pass.fail + Exam3:Grade + Exam2:Exam3 + Exam4:Pass.fail +
    Exam1:Grade + Exam2:Exam4 + Exam4:Grade + Exam2:Pass.fail +
    Exam2:Exam4:Pass.fail,
  data = train
)

summary(final_model)

# Prediction of final scores
final_pred = predict.glm(final_model, test, type = "response")

# Running t-test to test if the mean predicted values are different from actual
values
t.test(final_pred, test$Final_score)
```

Output:

```
Call:
glm(formula=Final_score~Exam1+Exam2+Exam3+Exam4+Grade+
   Pass.fail + Exam3:Grade + Exam2:Exam3 + Exam4:Pass.fail +
   Exam1:Grade + Exam2:Exam4 + Exam4:Grade + Exam2:Pass.fail +
   Exam2:Exam4:Pass.fail, data = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-13.175  -2.239   0.000   2.626  11.089

Coefficients: (2 not defined because of singularities)
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           97.583069   6.361748  15.339  < 2e-16 ***
Exam1                 -0.003353   0.038274  -0.088 0.930535
Exam2                  0.392310   0.445833   0.880 0.383015
Exam3                 -0.530283   0.253665  -2.090 0.041578 *
Exam4                  2.981650   0.850815   3.504 0.000962 ***
GradeB               -37.715882   7.104749  -5.309 2.43e-06 ***
GradeC               -39.383299   6.554371  -6.009 1.99e-07 ***
GradeD               -91.943123   7.500678 -12.258  < 2e-16 ***
Pass.failP                   NA         NA      NA       NA
Exam3:GradeB           0.644920   0.306719   2.103 0.040453 *
Exam3:GradeC           0.059195   0.280526   0.211 0.833718
Exam3:GradeD           2.354691   0.436383   5.396 1.78e-06 ***
Exam2:Exam3            0.027260   0.007117   3.830 0.000352 ***
Exam4:Pass.failP      -3.663756   0.928866  -3.944 0.000245 ***
Exam1:GradeB           0.103407   0.046206   2.238 0.029619 *
Exam1:GradeC           0.009528   0.049871   0.191 0.849241
Exam1:GradeD          -0.412450   0.165083  -2.498 0.015738 *
Exam2:Exam4           -0.157981   0.128296  -1.231 0.223828
Exam4:GradeB          -0.267402   0.389496  -0.687 0.495487
Exam4:GradeC           0.608206   0.390877   1.556 0.125891
Exam4:GradeD                 NA         NA      NA       NA
Exam2:Pass.failP      -1.019049   0.430253  -2.368 0.021688 *
Exam2:Exam4:Pass.failP 0.185424   0.128608   1.442 0.155479
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 25.68751)

   Null deviance: 29436.3 on 71 degrees of freedom
Residual deviance: 1310.1 on 51 degrees of freedom
AIC: 457.21
```

Number of Fisher Scoring iterations: 2

Welch Two Sample t-test

data: final_pred and test$Final_score
t = 0.49697, df = 61.849, p-value = 0.621
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-8.243648 13.698429
sample estimates:
mean of x mean of y
61.81552 59.08812