Statistical Modelling in R - Assignment 2

- Palak Nagdev, Roll number 22

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

1. # 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)
```

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 step AIC.
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)
```