DS 300 Project

```
data = read.csv(’C:/Users/kyle.persin/Desktop/DS 300/student-mat.csv’)
```

#ELIMINATE UNNECESSARY VARIABLES

#CREATE A BINARY CLASSIFICAION VARIABLE

```
data$G4 = ifelse(data$G3 > 10,
                 yes = 0,
                 no = 1)
data$G4 = factor(data$G4)
```

#ELIMINATE UNNECESSARY VARIABLES

```
data = subset(data, select = -c(school, reason, traveltime, G1, G2, G3))
```

#SPLIT THE DATA AND TRAIN CLASSIFIERS

```
library(caTools)
set.seed(123)
split = sample.split(data$G4, SplitRatio = 0.75)
training_set = subset(data, split == TRUE)
test_set = subset(data, split == FALSE)
```

Log Reg:

```
logreg_class = glm(formula = G4 ~ .,
                   family = binomial,
                   data = training_set)

logreg_probs = predict(logreg_class,
                       type = ’response’,
                       newdata = test_set[,-28])
```

```
accuracy_logreg = ifelse(test = logreg_probs > .5,
                         yes = 1,
                         no = 0)
accuracy_logreg = factor(accuracy_logreg)
```

#decision tree

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.0.4
```

```
dt_class = rpart(formula = G4 ~ .,
                 data = training_set)

dt_preds = predict(dt_class, newdata = test_set[,-28], type = 'prob')
dt_probs = dt_preds[, 2]
```

```
accuracy_dt = ifelse(test = dt_probs > .5,
                     yes = 1,
                     no = 0)
accuracy_dt = factor(accuracy_dt)
```

#random forest

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.4
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
rf_class = randomForest(x = training_set[,-28],
                        y = training_set$G4,
                        ntree = 100)
rf_preds = predict(rf_class,
                   newdata = test_set[,-28],
                   type = 'prob')
rf_probs = rf_preds[, 2]
```

```
accuracy_rf = ifelse(test = rf_probs > .5,
                     yes = 1,
                     no = 0)
accuracy_rf = factor(accuracy_rf)
```

#CHECK ACCURACY

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.4
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
##
## Attaching package: ’ggplot2’

## The following object is masked from ’package:randomForest’:
##
##     margin
```

```
confusionMatrix(test_set$G4, accuracy_dt)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 33 19
##          1 23 23
##
##                Accuracy : 0.5714
##                  95% CI : (0.4675, 0.671)
##     No Information Rate : 0.5714
##     P-Value [Acc > NIR] : 0.5425
##
##                   Kappa : 0.1353
##
##  Mcnemar’s Test P-Value : 0.6434
##
##             Sensitivity : 0.5893
##             Specificity : 0.5476
##          Pos Pred Value : 0.6346
##          Neg Pred Value : 0.5000
##              Prevalence : 0.5714
##          Detection Rate : 0.3367
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 0.5685
##
##        ’Positive’ Class : 0
##
```

```
confusionMatrix(test_set$G4, accuracy_logreg)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 31 21
##          1 17 29
##
##                Accuracy : 0.6122
##                  95% CI : (0.5085, 0.709)
##     No Information Rate : 0.5102
##     P-Value [Acc > NIR] : 0.02705
##
##                   Kappa : 0.2255
```

```
##
##  Mcnemar’s Test P-Value : 0.62650
##
##             Sensitivity : 0.6458
##             Specificity : 0.5800
##          Pos Pred Value : 0.5962
##          Neg Pred Value : 0.6304
##              Prevalence : 0.4898
##          Detection Rate : 0.3163
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 0.6129
##
##        ’Positive’ Class : 0
##
```

```
confusionMatrix(test_set$G4, accuracy_rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 32 20
##          1 20 26
##
##                Accuracy : 0.5918
##                  95% CI : (0.4879, 0.6901)
##     No Information Rate : 0.5306
##     P-Value [Acc > NIR] : 0.1326
##
##                   Kappa : 0.1806
##
##  Mcnemar’s Test P-Value : 1.0000
##
##             Sensitivity : 0.6154
##             Specificity : 0.5652
##          Pos Pred Value : 0.6154
##          Neg Pred Value : 0.5652
##              Prevalence : 0.5306
##          Detection Rate : 0.3265
##    Detection Prevalence : 0.5306
##       Balanced Accuracy : 0.5903
##
##        ’Positive’ Class : 0
##
```

#CREATING ROC CURVES

```
library(pROC)
```

```
## Warning: package ’pROC’ was built under R version 4.0.4
```

```
## Type ’citation("pROC")’ for a citation.
```

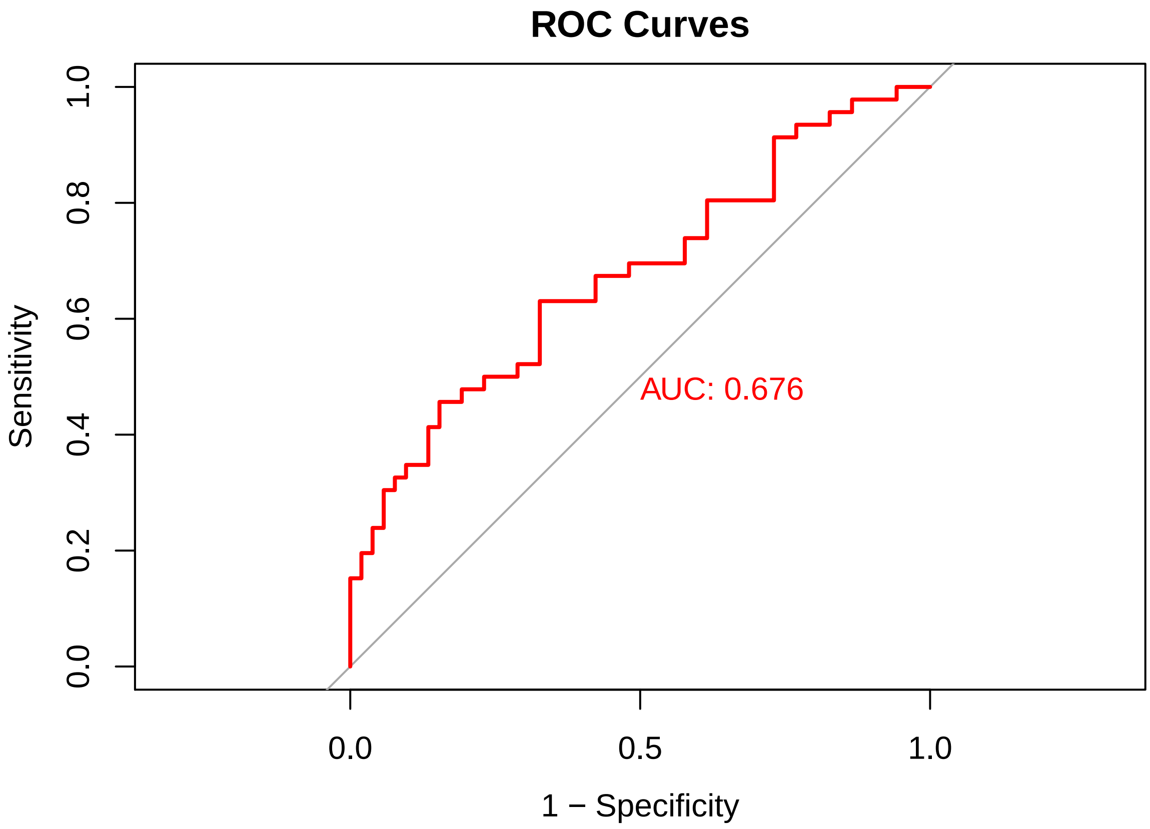
```
##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
logregROC = roc(test_set$G4 ~ logreg_probs, plot=TRUE, print.auc=TRUE, col="red", lwd =2, legacy.axes=T
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



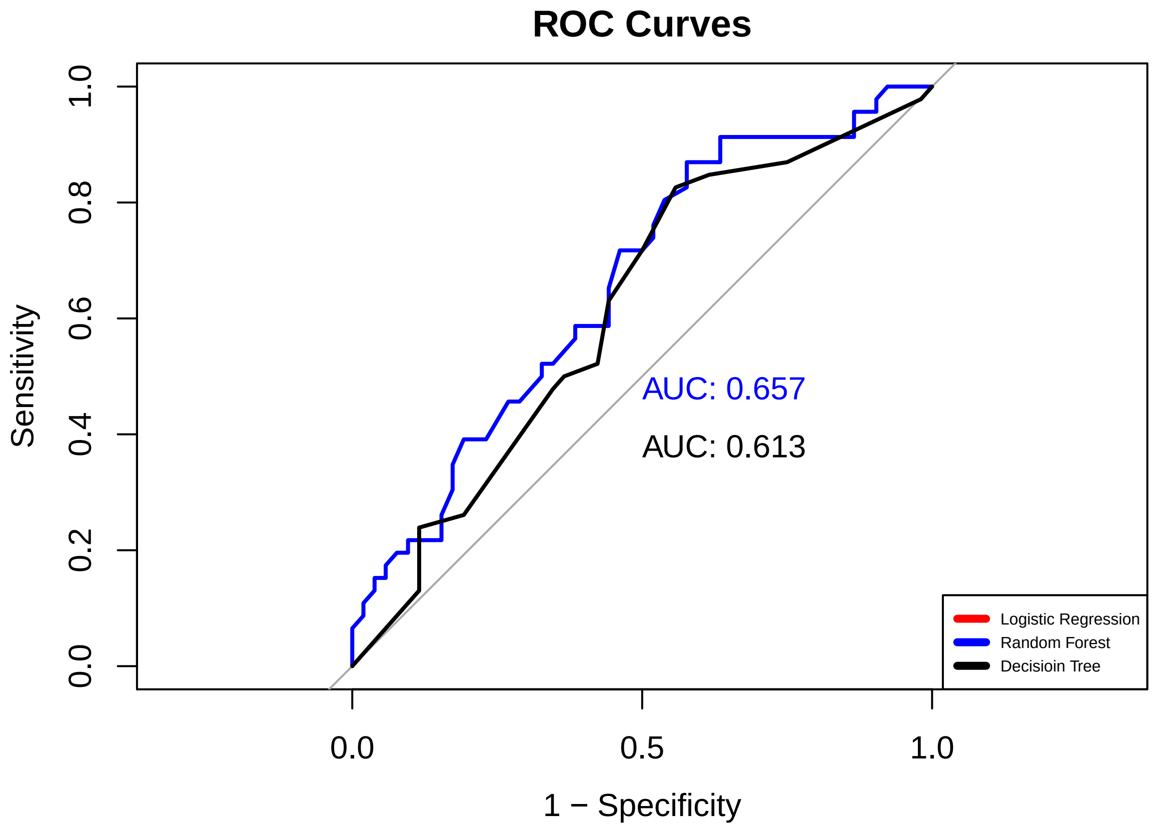
```
rfROC = roc(test_set$G4 ~ rf_probs, plot=TRUE, print.auc=TRUE, col="blue", lwd =2, legacy.axes=TRUE, ma
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
dtROC = roc(test_set$G4 ~ dt_probs, plot=TRUE, print.auc=TRUE, col="black", lwd = 2, print.auc.y=0.4, l
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
legend("bottomright",legend=c("Logistic Regression","Random Forest", "Decisioin Tree"), col=c("red", "bl
```



#CHECKING AUC

```
auc(dtROC)
```

```
## Area under the curve: 0.6131
```

```
auc(rfROC)
```

```
## Area under the curve: 0.6572
```

```
auc(logregROC)
```

```
## Area under the curve: 0.676
```