R Notebook

Importing Libraries

```
library(caret)
```

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

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

```
library(rpart)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(pROC)
```

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

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

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

Importing Data

```
data <- read.csv("Bike_Buying_Costumers.csv")
```

Cleaning Data

```
subset_data <- data[complete.cases(data[, c("DateFirstPurchase", "MaritalStatus", "YearlyIncome",     "Gender", "TotalChildren", "NumberChildrenAtHome", "Education",  "Occupation", "HomeOwnerFlag",     "NumberCarsOwned", "CommuteDistance", "BirthDate")]), ]
```

Incorporating Age

```
subset_data$BirthDate <- as.Date(subset_data$BirthDate, format = "%m/%d/%Y")
subset_data$Age <- as.numeric(difftime(as.Date("2007-10-31"), subset_data$BirthDate, units = "days"))
subset_data$Age <- round(subset_data$Age / 365.25)
```

```
head(subset_data)
```

	CustomerID <int>	TerritoryID <int>	TotalPurchaseYTD <dbl>	DateFirstPurchase <chr>
702	11000	9	0.00	01/15/2004
703	11001	9	4.99	09/28/2003
704	11002	9	2319.99	03/19/2004
705	11003	9	2482.23	05/19/2004
706	11004	9	69.99	07/31/2004
707	11005	9	2384.07	11/28/2003

6 rows | 1-5 of 18 columns

Descriptive Statistics

```
summary(subset_data)
```

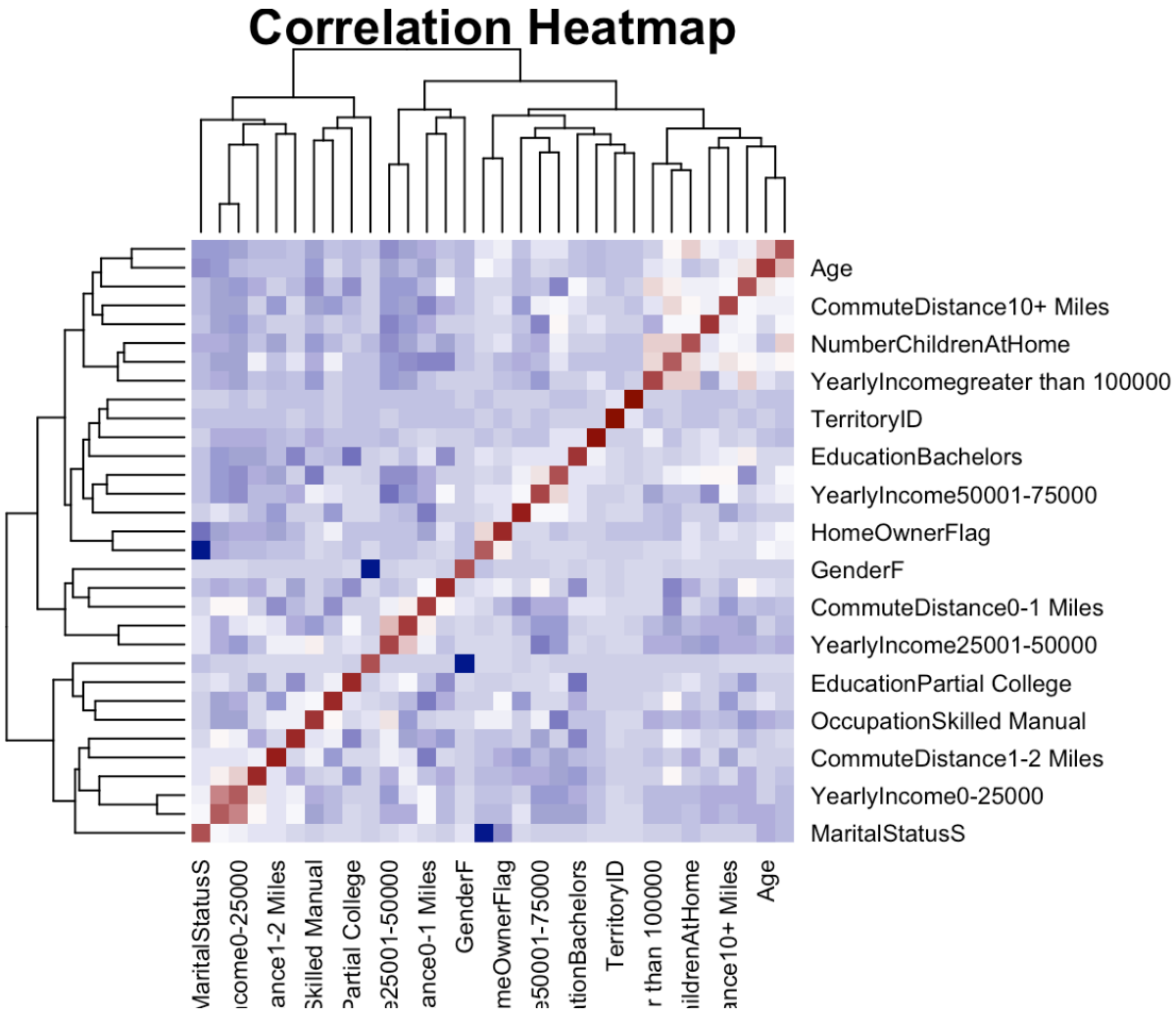
```
##    CustomerID     TerritoryID     TotalPurchaseYTD    DateFirstPurchase
##  Min.   :11000   Min.   : 1.000   Min.   :   -0.002   Length:9778
##  1st Qu.:13444   1st Qu.: 4.000   1st Qu.:   14.980   Class :character
##  Median :15888   Median : 6.000   Median :   79.970   Mode  :character
##  Mean   :15888   Mean   : 5.854   Mean   : 1139.952
##  3rd Qu.:18333   3rd Qu.: 9.000   3rd Qu.: 1759.970
##  Max.   :20777   Max.   :10.000   Max.   :13293.090
##  MaritalStatus      YearlyIncome          Gender           TotalChildren
##  Length:9778        Length:9778        Length:9778        Min.   :0.000
##  Class :character   Class :character   Class :character   1st Qu.:0.000
##  Mode  :character   Mode  :character   Mode  :character   Median :2.000
##                                                           Mean   :1.855
##                                                           3rd Qu.:3.000
##                                                           Max.   :5.000
##  NumberChildrenAtHome  Education          Occupation         HomeOwnerFlag
##  Min.   :0.000        Length:9778        Length:9778        Min.   :0.0000
##  1st Qu.:0.000        Class :character   Class :character   1st Qu.:0.0000
##  Median :0.000        Mode  :character   Mode  :character   Median :1.0000
##  Mean   :1.034                                              Mean   :0.6732
##  3rd Qu.:2.000                                              3rd Qu.:1.0000
##  Max.   :5.000                                              Max.   :1.0000
##  NumberCarsOwned CommuteDistance      BirthDate              Age
##  Min.   :0.000   Length:9778        Min.   :1912-03-23   Min.   :27.00
##  1st Qu.:1.000   Class :character   1st Qu.:1954-08-21   1st Qu.:37.00
##  Median :2.000   Mode  :character   Median :1963-10-05   Median :44.00
##  Mean   :1.516                      Mean   :1962-02-10   Mean   :45.71
##  3rd Qu.:2.000                      3rd Qu.:1970-10-02   3rd Qu.:53.00
##  Max.   :4.000                      Max.   :1980-12-26   Max.   :96.00
##  BikeBuyingCustomer
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4927
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

One Hot Encoding

```
columns_to_encode <- c("MaritalStatus", "YearlyIncome", "Gender", "Education", "Oc
cupation", "CommuteDistance")

formula <- as.formula(paste("~", paste(columns_to_encode, collapse = " + ")))

encoded_data <- predict(dummyVars(formula, data = subset_data), newdata = subset_d
ata)

final_data <- cbind(subset_data[, -which(names(data) %in% columns_to_encode)], enc
oded_data)

final_data <- final_data[, -which(names(final_data) %in% c("BirthDate", "DateFirst
Purchase", "CustomerID"))]
```

Correlation

```
cor_matrix <- cor(final_data)
heatmap(cor_matrix,
        col = colorRampPalette(c("darkblue", "white", "darkred"))(50),
        main = "Correlation Heatmap")
```



Feature Importance

```
colnames(final_data)[colnames(final_data) %in% c("YearlyIncome0-25000", "YearlyInc
ome25001-50000", "YearlyIncome50001-75000", "YearlyIncome75001-100000", "YearlyInc
omegreater than 100000", "EducationGraduate Degree", "EducationHigh School", "Educ
ationPartial College", "EducationPartial High School", "OccupationSkilled Manual",
"CommuteDistance0-1 Miles", "CommuteDistance1-2 Miles", "CommuteDistance10+ Miles"
, "CommuteDistance2-5 Miles", "CommuteDistance10+ Miles", "CommuteDistance2-5 Mile
s", "CommuteDistance5-10 Miles")] <- c("YearlyIncome0", "YearlyIncome1", "YearlyIn
come2", "YearlyIncome3", "YearlyIncome4", "EducationGraduateDegree", "EducationHig
hSchool", "EducationPartialCollege", "EducationPartialHighSchool", "OccupationSkil
ledManual", "CommuteDistance0", "CommuteDistance1", "CommuteDistance2", "CommuteDi
stance3", "CommuteDistance4")

model <- randomForest(BikeBuyingCustomer ~ ., data = final_data)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
importance <- importance(model)
```

Eliminating Features

```
final_data <- final_data[, -which(names(final_data) %in% c("YearlyIncome4", "Occup
ationManual", "EducationPartialHighSchool", "YearlyIncome0", "OccupationManagement
", "CommuteDistance2", "OccupationClerical", "EducationGraduateDegree", "Education
HighSchool", "TotalChildren", "OccupationProfessional"))]
```

Scaling Data

```
normalized_data <- preProcess(final_data, method = c("range"))
scaled_data <- predict(normalized_data, newdata = final_data)
```

Test-Train Split

```
set.seed(12345)
indices <- sample(nrow(scaled_data), size = round(0.8 * nrow(scaled_data)), replac
e = FALSE)
train_data <- scaled_data[indices, ]
test_data <- scaled_data[-indices, ]
```

Logistic Regression

```
model_LR <- glm(BikeBuyingCustomer ~ ., data = train_data, family = binomial)
predictions_LR <- predict(model_LR, newdata = test_data, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
predicted_labels_LR <- ifelse(predictions_LR >= 0.5, 1, 0)
```

Accuracy - Logistic Regression

```
actual_labels <- test_data$BikeBuyingCustomer
accuracy_LR <- sum(predicted_labels_LR == actual_labels) / length(actual_labels)
```

Confusion Matrix - Logistic Regression

```
cm_LR <- confusionMatrix(data = as.factor(predicted_labels_LR), reference = as.fac
tor(actual_labels))
print(cm_LR)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 549 509
##          1 416 482
##
##                Accuracy : 0.5271
##                  95% CI : (0.5047, 0.5494)
##     No Information Rate : 0.5066
##     P-Value [Acc > NIR] : 0.036985
##
##                   Kappa : 0.0552
##
##  Mcnemar's Test P-Value : 0.002487
##
##             Sensitivity : 0.5689
##             Specificity : 0.4864
##          Pos Pred Value : 0.5189
##          Neg Pred Value : 0.5367
##              Prevalence : 0.4934
##          Detection Rate : 0.2807
##    Detection Prevalence : 0.5409
##       Balanced Accuracy : 0.5276
##
##        'Positive' Class : 0
##
```

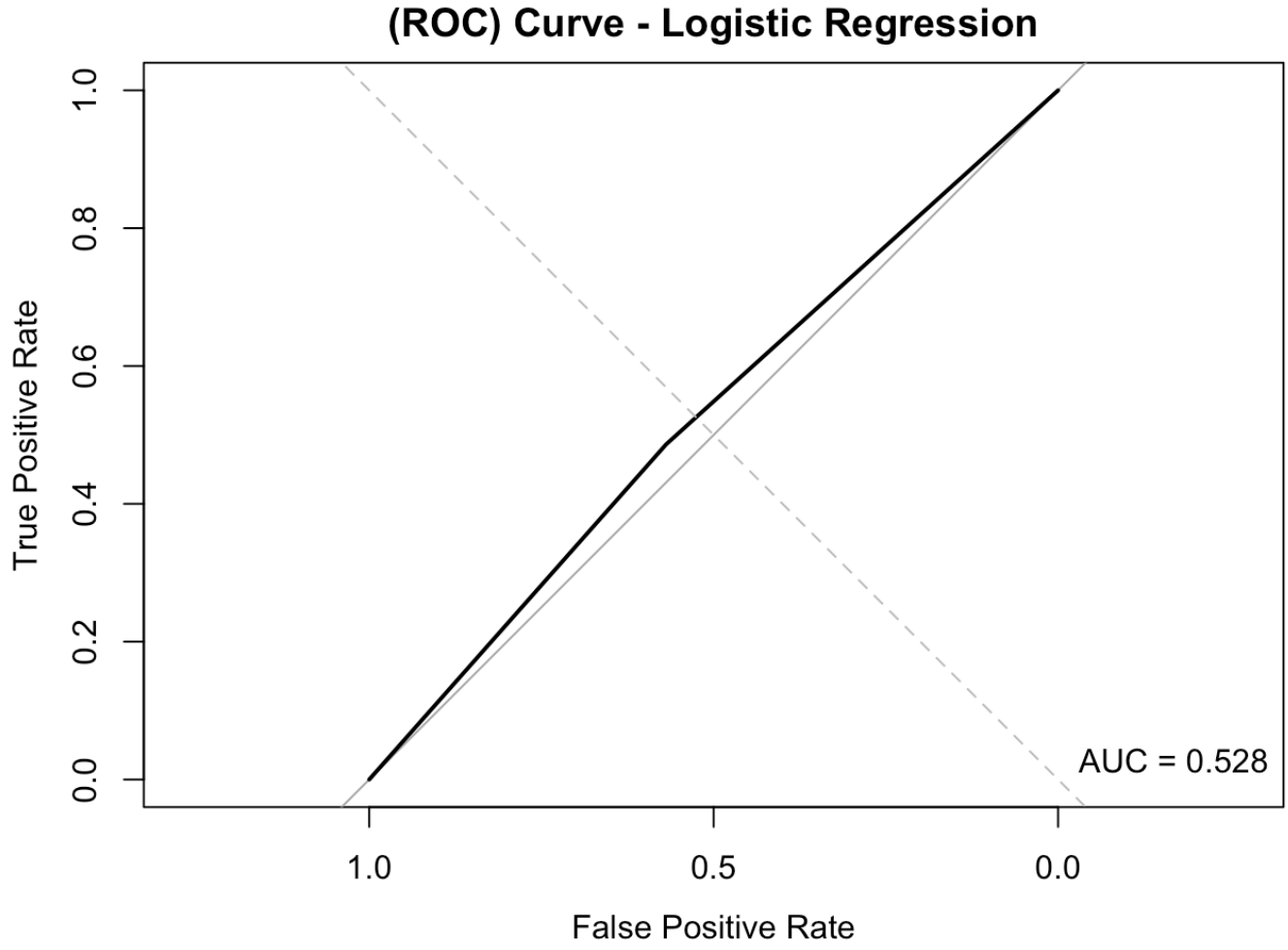
ROC_AUC - Logistic Regression

```
roc_obj <- roc(actual_labels, predicted_labels_LR)
```

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

```
## Setting direction: controls < cases
```

```
plot(roc_obj, main = "(ROC) Curve - Logistic Regression",
     xlab = "False Positive Rate", ylab = "True Positive Rate")

abline(0, 1, lty = 2, col = "gray")

auc_val <- auc(roc_obj)
legend("bottomright", paste("AUC =", round(auc_val, 3)), bty = "n")
```



Decision Tree

```
model_DT <- rpart(train_data$BikeBuyingCustomer ~ ., data = subset(train_data, sel
ect = -BikeBuyingCustomer), method = "class")
predictions_DT <- predict(model_DT, newdata = test_data)
predicted_labels_DT <- ifelse(predictions_DT[,1] >= 0.5, 1, 0)
```

Accuracy - Decision Tree

```
accuracy_DT <- sum(predicted_labels_DT == actual_labels) / length(actual_labels)
print(accuracy_DT)
```

```
## [1] 0.4698364
```

Confusion Matrix - Decision Tree

```
cm_DT <- confusionMatrix(data = as.factor(predicted_labels_DT), reference = as.fac
tor(actual_labels))
print(cm_DT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 263 335
##          1 702 656
##
##                Accuracy : 0.4698
##                  95% CI : (0.4475, 0.4922)
##     No Information Rate : 0.5066
##     P-Value [Acc > NIR] : 0.9995
##
##                   Kappa : -0.0658
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.2725
##             Specificity : 0.6620
##          Pos Pred Value : 0.4398
##          Neg Pred Value : 0.4831
##              Prevalence : 0.4934
##          Detection Rate : 0.1345
##    Detection Prevalence : 0.3057
##       Balanced Accuracy : 0.4672
##
##        'Positive' Class : 0
##
```

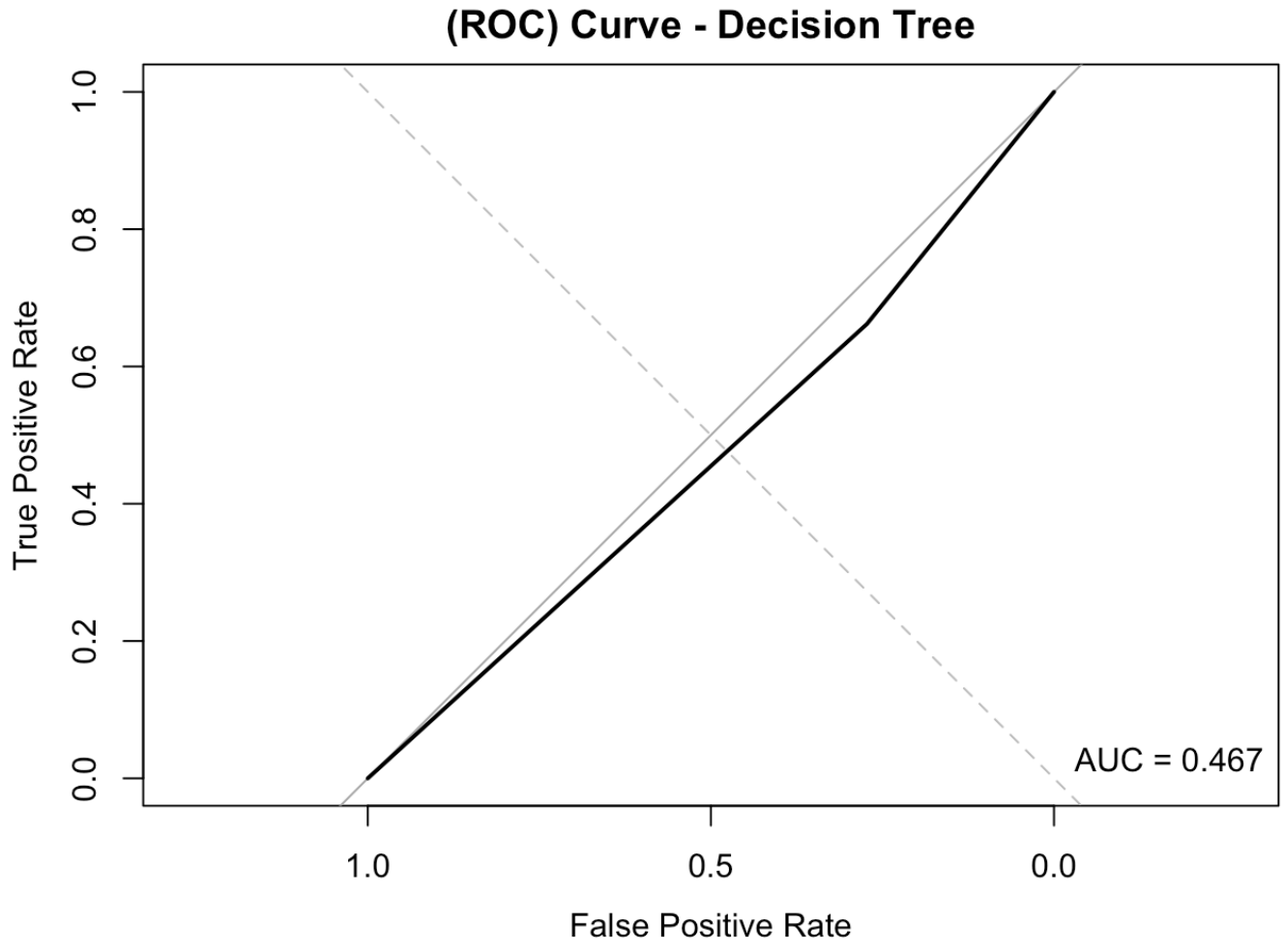
ROC_AUC - Decision Tree

```
roc_obj <- roc(actual_labels, predicted_labels_DT)
```

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

```
## Setting direction: controls < cases
```

```
plot(roc_obj, main = "(ROC) Curve - Decision Tree",
     xlab = "False Positive Rate", ylab = "True Positive Rate")

abline(0, 1, lty = 2, col = "gray")

auc_val <- auc(roc_obj)
legend("bottomright", paste("AUC =", round(auc_val, 3)), bty = "n")
```



CART

```
model_CART <- rpart(BikeBuyingCustomer ~ ., data = train_data, method = "class")
predictions_CART <- predict(model_CART, newdata = test_data, type = "class")
```

Accuracy - CART

```
accuracy_CART <- sum(predictions_CART == actual_labels) / length(actual_labels)
print(accuracy_CART)
```

```
## [1] 0.5301636
```

Confusion Matrix - CART

```
cm_CART <- confusionMatrix(data = predictions_CART, reference = as.factor(actual_l
abels))
print(cm_CART)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 702 656
##          1 263 335
##
##                Accuracy : 0.5302
##                  95% CI : (0.5078, 0.5525)
##     No Information Rate : 0.5066
##     P-Value [Acc > NIR] : 0.01978
##
##                   Kappa : 0.0652
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.7275
##             Specificity : 0.3380
##          Pos Pred Value : 0.5169
##          Neg Pred Value : 0.5602
##              Prevalence : 0.4934
##          Detection Rate : 0.3589
##    Detection Prevalence : 0.6943
##       Balanced Accuracy : 0.5328
##
##        'Positive' Class : 0
##
```

ROC_AUC - CART

```
roc_obj <- roc(actual_labels, as.numeric(predictions_CART))
```

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

```
## Setting direction: controls < cases
```

```
plot(roc_obj, main = "(ROC) Curve - CART",
     xlab = "False Positive Rate", ylab = "True Positive Rate")

abline(0, 1, lty = 2, col = "gray")

auc_val <- auc(roc_obj)
legend("bottomright", paste("AUC =", round(auc_val, 3)), bty = "n")
```