> library(caret)

> library(ggplot2)

> library(dummies)

> library(pROC)

>

> #Importing dataset

> df <- read.csv("HR\_data.csv", header = TRUE)

>

> #Renaming columns and cleaning the dataset

> colnames(df)[colnames(df) == "sales"] <- "department"

>

> #Creating factors and partitions

> df$department <- as.factor(df$department)

> df$salary <- as.factor(df$salary)

> df$left <- as.factor(df$left)

>

> set.seed(123457)

> train\_index <- caret::createDataPartition(df$department, p = 0.7, list = FALSE)

> df.train <- df[train\_index,]

> df.test <- df[-train\_index,]

>

> #Logistic regression without feature selection

> fit1 <- glm(left ~ ., data = df.train, family = "binomial")

> df.train$predicted\_prob <- predict(fit1, df.train, type = "response")

> df.test$predicted\_prob <- predict(fit1, df.test, type = "response")

>

> #Classification with cutoff=0.5

> df.train$predicted\_outcome <- ifelse((df.train$predicted\_prob > 0.5), 1, 0)

> conf.matrix.train <- caret::confusionMatrix(df.train$predicted\_outcome, df.train$left)

> conf.matrix.train

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 7395 1664

1 586 857

Accuracy : 0.7858

95% CI : (0.7778, 0.7936)

No Information Rate : 0.76

P-Value [Acc > NIR] : 1.95e-10

Kappa : 0.3122

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9266

Specificity : 0.3399

Pos Pred Value : 0.8163

Neg Pred Value : 0.5939

Prevalence : 0.7600

Detection Rate : 0.7042

Detection Prevalence : 0.8626

Balanced Accuracy : 0.6333

'Positive' Class : 0

>

> df.test$predicted\_outcome <- ifelse((df.test$predicted\_prob > 0.5), 1, 0)

> conf.matrix.test <- caret::confusionMatrix(df.test$predicted\_outcome, df.test$left)

> conf.matrix.test

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 3186 692

1 261 358

Accuracy : 0.7881

95% CI : (0.7758, 0.7999)

No Information Rate : 0.7665

P-Value [Acc > NIR] : 0.0002949

Kappa : 0.3094

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9243

Specificity : 0.3410

Pos Pred Value : 0.8216

Neg Pred Value : 0.5784

Prevalence : 0.7665

Detection Rate : 0.7085

Detection Prevalence : 0.8624

Balanced Accuracy : 0.6326

'Positive' Class : 0

>

> #ROC for Logistic regerssion with all predictors

> roc.train <- pROC::roc(df.train$left, df.train$predicted\_prob)

> pROC::plot.roc(roc.train)

>

> roc.test1 <- pROC::roc(df.test$left, df.test$predicted\_prob)

> pROC::plot.roc(roc.test1)

>

> #Plot cutoff vs accuracy

> cutoff <- seq(0, 1, length = 10000)

> acc <- numeric(10000)

> accPlot.dataFrame <- data.frame(CUTOFF = cutoff, ACCURACY = acc)

>

> #Plot for training data set

> for (index in 1:10000) {

+ pred <- ifelse((df.train$predicted\_prob > cutoff[index]), 1, 0)

+ true.positives <- sum(pred == 1 & df.train$left == 1)

+ true.negatives <- sum(pred == 0 & df.train$left == 0)

+ accPlot.dataFrame$ACCURACY[index] <- ((true.positives + true.negatives) / length(df.train$left)) \* 100

+ }

> ggplot2::ggplot(data = accPlot.dataFrame, mapping = aes(x = CUTOFF, y = ACCURACY, col)) + geom\_line(size = 1)

> idealCutoff <- accPlot.dataFrame$CUTOFF[which.max(accPlot.dataFrame$ACCURACY)]

> acc.max <- accPlot.dataFrame$ACCURACY[which.max(accPlot.dataFrame$ACCURACY)]

> accData <- data.frame(CUTOFF = idealCutoff, ACCURACY = acc.max)

>

> #Plot for test data set

> for (index in 1:10000) {

+ pred <- ifelse((df.test$predicted\_prob > cutoff[index]), 1, 0)

+ true.positives <- sum(pred == 1 & df.test$left == 1)

+ true.negatives <- sum(pred == 0 & df.test$left == 0)

+ accPlot.dataFrame$ACCURACY[index] <- ((true.positives + true.negatives) / length(df.test$left)) \* 100

+ }

> ggplot2::ggplot(data = accPlot.dataFrame, mapping = aes(x = CUTOFF, y = ACCURACY, col)) + geom\_line(size = 1)

> idealCutoff <- accPlot.dataFrame$CUTOFF[which.max(accPlot.dataFrame$ACCURACY)]

> acc.max <- accPlot.dataFrame$ACCURACY[which.max(accPlot.dataFrame$ACCURACY)]

>

> accData <- rbind.data.frame(accData, c(idealCutoff, acc.max), make.row.names = FALSE)

> row.names(accData) <- c("TRAINING", "TEST")

> print(accData)

CUTOFF ACCURACY

TRAINING 0.3989399 80.32756

TEST 0.4139414 79.96442

>

> #Confusion matrices with optimal cutoff

> df.train$predicted\_outcome <- ifelse((df.train$predicted\_prob > accData$CUTOFF[rownames(accData) == "TRAINING"]), 1, 0)

> conf.matrix.train <- caret::confusionMatrix(df.train$predicted\_outcome, df.train$left)

> conf.matrix.train

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 7067 1152

1 914 1369

Accuracy : 0.8033

95% CI : (0.7955, 0.8108)

No Information Rate : 0.76

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.4428

Mcnemar's Test P-Value : 1.847e-07

Sensitivity : 0.8855

Specificity : 0.5430

Pos Pred Value : 0.8598

Neg Pred Value : 0.5996

Prevalence : 0.7600

Detection Rate : 0.6729

Detection Prevalence : 0.7826

Balanced Accuracy : 0.7143

'Positive' Class : 0

>

> df.test$predicted\_outcome <- ifelse((df.test$predicted\_prob > accData$CUTOFF[rownames(accData) == "TEST"]), 1, 0)

> conf.matrix.test <- caret::confusionMatrix(df.test$predicted\_outcome, df.test$left)

> conf.matrix.test

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 3061 515

1 386 535

Accuracy : 0.7996

95% CI : (0.7876, 0.8113)

No Information Rate : 0.7665

P-Value [Acc > NIR] : 5.085e-08

Kappa : 0.4153

Mcnemar's Test P-Value : 2.005e-05

Sensitivity : 0.8880

Specificity : 0.5095

Pos Pred Value : 0.8560

Neg Pred Value : 0.5809

Prevalence : 0.7665

Detection Rate : 0.6807

Detection Prevalence : 0.7952

Balanced Accuracy : 0.6988

'Positive' Class : 0