**STT 811**

**In-Class Assignment 9**

This problem will use the OJ dataset. Note that you will use Purchase as the target (no need to convert to 0/1)

1. Split the data into training and test datasets (with a 75/25 split).

oj <- ISLR2::OJ

split\_pct <- 0.75

n <- length(oj$Purchase)\*split\_pct # train size

row\_samp <- sample(1:length(oj$Purchase), n, replace = FALSE)

train <- oj[row\_samp,]

test <- oj[-row\_samp,]

1. Build a Naïve Bayes model for your target based on PriceDiff and LoyalCH (using the naivebayes command). Compute the confusion matrix for both the train and test datasets. How do they compare?

naiveBayesModel <- naiveBayes(Purchase ~ PriceDiff+LoyalCH, data = train)

test\_pred <- predict(naiveBayesModel,test)

train\_cm <- confusionMatrix(as.factor(as.integer(2\*naiveBayesModel$fitted.values)), reference = as.factor(train$Purchase))

test\_cm <- confusionMatrix(as.factor(as.integer(2\*test\_pred)), reference = as.factor(test$Purchase))

print("train CM")

train\_cm$table

print("test CM")

test\_cm$table

1. Build another naïve Bayes model using Store as well (make sure it is categorical). Compare how the confusion matrix does (on test) vs. (2)

naiveBayesModel2 <- naiveBayes(Purchase ~ PriceDiff+LoyalCH+as.factor(STORE), data = train)

test\_pred <- predict(naiveBayesModel2,test, type = "response")

train\_cm <- confusionMatrix(as.factor(as.integer(2\*naiveBayesModel$fitted.values)), reference = as.factor(train$Purchase))

test\_cm <- confusionMatrix(as.factor(as.integer(2\*test\_pred)), reference = as.factor(test$Purchase))

print("train CM 2")

train\_cm$table

print("test CM 2")

test\_cm$table