**Quiz 6 – Week 8**

*Content: Naïve Bayes Classifier*

Data files <bank-sample.csv> and <bank-sample-test.csv> are provided on Canvas.

Please open the R code file attached <Quiz6.R> and run the code to answer the questions below.

*Note: to run the code file smoothly, you might need to:*

*Set working directory: setwd()*

*Install packages: install.packages(“e1071”)*

**Q1.** How many explanatory variables are there in the Naïve Bayes classifier “nb\_model”?

6,7,8,9,10

Ans: 10

Explanation: data set “banktrain” has 11 columns (after dropping 6 unwanted columns).

Model was formed as

nb\_model <- naiveBayes( subscribed ~ . , data = banktrain)

where the response variable “subscribed” depends on a dot. This dot is to replace for “ALL OTHER COLUMNS” in data = banktrain. Hence, there are 10 columns used as explanatories in the model.

**Q2.** The values in “nb\_prediction” are the class labels (no/yes) for each test point in the test data. True or False?

Ans: False. It’s the probabilities, since type = “raw” is specified in the command

nb\_prediction <- predict(nb\_model, newdata = banktest[ , -11], type ='raw')

If we want to get the class labels (no/yes) then we should specify type = “class”.

**Q3.** Write R code to **get the predicted class labels** for the data points in banktest and report the number of points be predicted as “yes” in the blank.

Note: only report the answer, not include the code in the blank provided.

Ans: 5

predicted.class <- predict(nb\_model, newdata = banktest[ , -11], type ='class')

table(predicted.class)

# 5 out of 100 points (or customers) be predicted as “yes”.

**Q4.** Continuing with the R code for Question 4 above, write R code to form a confusion matrix between the **predicted class label** and the **real class label** of the test set. Calculate and report the accuracy in the blank.

Note: only report the answer, not include the code in the blank provided.

Ans: 0.9 or 90%

confusion.matrix = table(predicted.class, banktest[ , 11])

accuracy = sum(diag(confusion.matrix)) / sum(confusion.matrix)

accuracy

# 0.9