**Learning outcomes:**

After solving these exercises, you should be able to understand the following:

1. Applying xgboost and Adaboost algorithms to solve classification problems.
2. Interpreting the results generated from each algorithm in R.
3. Comparison of the model performance in terms of precision, recall and accuracy

**Pre-processing: Universal Bank Dataset:**

The Universal Bank dataset has 14 variables and 5,000 records. Use “Personal.Loan” as target variable.

1. Import the data into R
2. Drop the features based on the data understanding.
3. Convert the features into appropriate data type.
4. Convert all categorical attributes into factors.

**Adaboost: Universal Bank Dataset**

1. After applying the pre-processing Step1-4
2. Split the data into train and evaluation data sets.
3. Build the classification model using Adaboost:

# build the classification model using Adaboost

library(ada)

model = ada(loan ~ ., iter = 20,data = trainR, loss="logistic")

1. Predict the values using model on test data sets.

# predict the values using model on test data sets.

pred = predict(model, testR);

pred

1. Evaluate the performance of the model

confusionMatrix(testR$loan,pred,positive = "1")

1. Experiment with different number of iterations and find the best.

**XGBoost: Universal Bank Dataset**

1. After applying the pre-processing Step1-4
2. Convert all the categorical features into numeric data.
3. Split the data into train and test data.
4. Standardize the data.
5. Convert the data into DMatrix form.

# Constructing the Dense matrix on the train and test data

dtrain = xgb.DMatrix(data = as.matrix(train\_Data[,ind\_Attr]),

label = train\_Data$loan)

dtest = xgb.DMatrix(data = as.matrix(test\_Data[,ind\_Attr]),

label = test\_Data$loan)

1. Creating a watchlist to evaluate the model performance on the test data

#Use watchlist parameter. It is a list of xgb.DMatrix, each of them tagged with a name.

watchlist = list(train=dtrain, test=dtest)

1. Build the xgboost model.

# Building the xgboost model

model = xgb.train(data=dtrain, max.depth=4,

eta=0.3, nthread = 2, nround=20,

watchlist=watchlist,

eval.metric = "error",

objective = "binary:logistic", verbose = 1)

1. Identify the important features.

importance <- xgb.importance(feature\_names = setdiff(names(final\_Data), "loan"), model = model)

print(importance)

xgb.plot.importance(importance\_matrix = importance)

1. Predict on the test data.

# prediction on test data

pred <- predict(model, as.matrix(test\_Data[,ind\_Attr]))

prediction <- as.numeric(pred > 0.5,1,0)

prediction <- as.factor(as.character(prediction))

1. Evaluate the performance of the model

confusionMatrix(testR$loan, prediction,positive = "1")