Classification Algorithms

# Read Data

data<-read.csv("UniversalBank.csv")

#Structure of Data

str(data)

summary(data) #Summary of data

# converting data into factors

data$Education<-factor(data$Education)

data$Personal.Loan<-factor(data$Personal.Loan)

data$Securities.Account<-factor(data$Securities.Account)

data$CD.Account<-factor(data$CD.Account)

data$Online<-factor(data$Online)

data$CreditCard<-factor(data$CreditCard)

str(data)

head(data)

summary(data)

# Creating Training and validation dataset

set.seed(1)

train.index <- sample(c(1:dim(data)[1]), dim(data)[1]\*0.6)

valid.index <- setdiff(c(1:dim(data)[1]), train.index)

train.df <- data[train.index, ]

valid.df <- data[valid.index, ]

dim(train.df)

t<- table(train.df$Personal.Loan); t

dim(valid.df)

v<-table(valid.df$Personal.Loan); v

############################################

#C4.5

############################################

install.packages("RWeka")

install.packages("party")

library(RWeka)

library(party)

fit1 <- J48(Personal.Loan ~ Age + Experience + Income +Family +CCAvg +Education +Mortgage +Securities.Account+CD.Account+Online+CreditCard, data=train.df)

# summarize the fit

summary(fit1)

install.packages("partykit")

library(partykit)

plot(fit1)

print(fit1)

#Model Evaluation

library(caret)

library(ggplot2)

# training

pred.class <- predict(fit1, newdata = train.df)

confusionMatrix(pred.class, train.df$Personal.Loan, positive="1")

F1\_Score(y\_true=train.df$Personal.Loan, y\_pred=pred.class, positive = "1") # F1 SCore calculation

# validation

pred.class1 <- predict(fit1, newdata = valid.df)

confusionMatrix(pred.class1, valid.df$Personal.Loan, positive="1")

F1\_Score(y\_true=valid.df$Personal.Loan, y\_pred=pred.class1, positive = "1")

###############################################

#CART

##############################################

install.packages("rpart")

library(rpart)

fit <- rpart(Personal.Loan ~ Age + Experience + Income +Family +CCAvg +Education +Mortgage +Securities.Account+CD.Account+Online+CreditCard, data=train.df,

control=rpart.control(minsplit=1))

summary(fit)

print(fit)

install.packages("rpart.plot")

library(rpart.plot)

rpart.plot(fit, extra = 101)

# predict

predict(fit,valid.df)

# set argument type = "class" in predict() to generate predicted class membership.

pred.train <- predict(fit,train.df,type = "class")

# generate confusion matrix for training data

library(caret)

library(ggplot2)

library(lattice)

confusionMatrix(pred.train, as.factor(train.df$Personal.Loan),positive="1")

F1\_Score(y\_true=train.df$Personal.Loan, y\_pred=pred.train, positive = "1") # F1 SCore calculation

# generate confusion matrix for validation data

pred.valid <- predict(fit,valid.df,type = "class")

confusionMatrix(pred.valid, as.factor(valid.df$Personal.Loan),positive="1")

F1\_Score(y\_true=valid.df$Personal.Loan, y\_pred=pred.valid, positive = "1")

###################################################

#JRip

###################################################

library(RWeka)

fit\_JRip <- JRip(Personal.Loan ~ Age + Experience + Income +Family +CCAvg +Education +Mortgage +Securities.Account+CD.Account+Online+CreditCard, data=train.df)

fit\_JRip

summary(fit\_JRip)

# predict

predict(fit\_JRip,valid.df)

# set argument type = "class" in predict() to generate predicted class membership.

pred.train1 <- predict(fit\_JRip,train.df,type = "class")

# generate confusion matrix for training data

library(caret)

library(ggplot2)

library(lattice)

confusionMatrix(pred.train1, as.factor(train.df$Personal.Loan),positive="1")

F1\_Score(y\_true=train.df$Personal.Loan, y\_pred=pred.train1, positive = "1") # F1 SCore calculation

# generate confusion matrix for validation data

pred.valid1 <- predict(fit\_JRip,valid.df,type = "class")

confusionMatrix(pred.valid1, as.factor(valid.df$Personal.Loan),positive="1")

F1\_Score(y\_true=valid.df$Personal.Loan, y\_pred=pred.valid1, positive = "1")

###################################################################

#Naive Bayes

##################################################################

library(e1071)

bayes.model <- naiveBayes(Personal.Loan ~ Age + Experience + Income +Family +CCAvg +Education +Mortgage +Securities.Account+CD.Account+Online+CreditCard, data=train.df)

bayes.model

pred.train2 <- predict(bayes.model,train.df,type = "class")

confusionMatrix(pred.train2, as.factor(train.df$Personal.Loan),positive="1")

F1\_Score(y\_true=train.df$Personal.Loan, y\_pred=pred.train2, positive = "1")

# generate confusion matrix for validation data

pred.valid2 <- predict(bayes.model,valid.df,type = "class")

confusionMatrix(pred.valid2, as.factor(valid.df$Personal.Loan),positive="1")

F1\_Score(y\_true=valid.df$Personal.Loan, y\_pred=pred.valid2, positive = "1")