**ML – Assignment**

**PART I**

**[For Regression Dataset]**

**Q1.1 Compare the performance of 10 machine learning models for given regression data set for the data partition of 70-30% with acceptable error of ±100.**

A1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Method | Package | r | R2 | Error | Accuracy |
| M1 | nnet | nnet | 1 | 1 | 0 | 100 |
| M2 | lm | - | 1 | 1 | 0 | 100 |
| M3 | brnn | brnn | 1 | 1 | 0.18 | 100 |
| M4 | linearRidge | ridge | 1 | 1 | 0.44 | 100 |
| M5 | randomForest | randomForest | 1 | 1 | 48.84 | 95.56 |
| M6 | qrf | quantregForest | 1 | 1 | 63.82 | 76.81 |
| M7 | M5Rules | RWeka | 1 | 1 | 44.99 | 97.54 |
| M8 | pcr | pls | 1 | 1 | 0 | 100 |
| M9 | Bagging | RWeka | 1 | 1 | 44.46 | 96.48 |
| M10 | rpart | rpart | 1 | 1 | 50.08 | 95.06 |

**CODES**

**nnet**

library(nnet)

iterations = 10

arr\_a=list()

arr\_b=list()

arr\_c=list()

arr\_r=list()

arr\_R=list()

arr\_acc=list()

arr\_mae=list()

modelName <- "neuralNetwork"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a=round(runif(1,10,100),0)

b=round(runif(1,1,1000000),0)

c=round(runif(1,1,1000000),0)

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

arr\_mae[i] <- mae

arr\_a[i] <- a

arr\_b[i] <- b

arr\_c[i] <- c

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_a, arr\_b, arr\_c, arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**lm**

iterations = 10

arr\_r=list()

arr\_R=list()

arr\_mae=list()

arr\_acc=list()

modelName <- "lm"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- lm(formula, trainDataset)

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**brnn**

library(brnn)

library(arm)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "brnn"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x<- data.matrix(trainDataset)

model <- brnn(x, train\_labels, neurons= 10)

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**linearRidge**

library(ridge)

iterations = 1000

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "ridge"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- linearRidge(formula, trainDataSet, nPCs = n)

Predicted <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

r <- cor(Actual,Predicted)

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,2) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**randomForest**

library(randomForest)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "randomForest"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- randomForest(formula, trainDataset, ntree=700, mtry=2)

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**qrf**

library(quantregForest)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "qrf"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x <- data.matrix(trainDataset)

model <- quantregForest(x, train\_labels, nthreads = 5, keep.inbag = FALSE, mtry= round(runif(1,5,15)))

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**M5Rules**

library(RWeka)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "M5rules"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- M5Rules(formula, trainDataSet)

Predicted <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

r <- cor(Actual,Predicted)

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**pcr**

library(pls)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "pcr"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- pcr(formula, n, data = trainDataSet, validation = "CV", method = pls.options()$pcralg)

Predicted <- predict(model, testDataSet, ncomp=15)

Actual <- as.double(unlist(testDataSet[target]))

r <- cor(Actual,Predicted)

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Bagging**

library(RWeka)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "bagging"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- Bagging(formula, trainDataSet)

Predicted <- predict(model,testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**rpart**

library(rpart)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "decisionTree"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- rpart(formula, trainDataset, method="anova", control = rpart.control(minsplit = 90,cp=0.0001))

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Q1.2 Ensemble the models from Table 1.1 for data partition for given regression data set of 70-30%and with acceptable error of ±100.**

A1.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Combination of | r | R2 | Error | Accuracy |
| E1 | M1, M5, M6, M7, M10 | 1 | 1 | 37.67 | 99.92 |
| E2 | M1, M2, M4, M9, M10 | 1 | 1 | 36.99 | 94.41 |
| E3 | M2, M4, M6, M8, M10 | 1 | 1 | 40.16 | 96.37 |
| E4 | M1, M3, M5, M7 | 1 | 1 | 34.52 | 99.89 |
| E5 | M1, M2, M6, M8, M10 | 1 | 1 | 18.67 | 100 |

**CODES**

**M1, M5, M6, M7, M10**

library(quantregForest)

library(nnet)

library(randomForest)

library(RWeka)

library(rpart)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "combo1"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x<- data.matrix(trainDataset)

model <- quantregForest(x, train\_labels, nthreads = 5, keep.inbag = FALSE, mtry= round(runif(1,5,15)))

Predicted1 <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a= 16

b= 656689

c= 632762

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted2 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- randomForest(formula, trainDataset, ntree=700, mtry=2)

Predicted3 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- M5Rules(formula, trainDataSet)

Predicted4 <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- rpart(formula, trainDataset, method="anova", control = rpart.control(minsplit = 90,cp=0.0001))

Predicted5 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

Predicted <- (Predicted1[,2] + Predicted2 + Predicted3 + Predicted4 +Predicted5)/5

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**M1, M2, M4, M9, M10**

library(nnet)

library(ridge)

library(RWeka)

library(rpart)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "combo2"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- lm(formula, trainDataset)

Predicted1 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a= 16

b= 656689

c= 632762

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted2 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- linearRidge(formula, trainDataSet, nPCs = n)

Predicted3 <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- Bagging(formula, trainDataSet)

Predicted4 <- predict(model,testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- rpart(formula, trainDataset, method="anova", control = rpart.control(minsplit = 90,cp=0.0001))

Predicted5 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

Predicted <- (Predicted1 + Predicted2 + Predicted3 + Predicted4 +Predicted5)/5

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**M2, M4, M6, M8, M10**

library(quantregForest)

library(ridge)

library(pls)

library(rpart)

iterations = 20

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "combo3"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

dataset <- dataset[sample(nrow(dataset)),]

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- lm(formula, trainDataset)

Predicted1 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x<- data.matrix(trainDataset)

model <- quantregForest(x, train\_labels, nthreads = 5, keep.inbag = FALSE, mtry= round(runif(1,5,15)))

Predicted2 <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- linearRidge(formula, trainDataSet, nPCs = n)

Predicted3 <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- pcr(formula, n, data = trainDataSet, validation = "CV", method = pls.options()$pcralg)

Predicted4 <- predict(model, testDataSet, ncomp=15)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- rpart(formula, trainDataset, method="anova", control = rpart.control(minsplit = 90,cp=0.0001))

Predicted5 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

Predicted <- (Predicted1 + Predicted2[,2] + Predicted3 + Predicted4 +Predicted5)/5

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**M1, M3, M5, M7**

library(quantregForest)

library(nnet)

library(randomForest)

library(RWeka)

library(rpart)

library(arm)

library(brnn)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "combo4"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

#-------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x<- data.matrix(trainDataset)

model <- brnn(x, train\_labels, neurons= 10)

Predicted1 <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a= 16

b= 656689

c= 632762

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted2 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- randomForest(formula, trainDataset, ntree=700, mtry=2)

Predicted3 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- M5Rules(formula, trainDataSet)

Predicted4 <- predict(model, testDataSet)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

Predicted <- (Predicted1 + Predicted2 + Predicted3 + Predicted4)/4

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**M1, M2, M6, M8, M10**

library(quantregForest)

library(nnet)

library(pls)

library(rpart)

iterations = 10

arr\_r=list()

arr\_mae=list()

arr\_R=list()

arr\_acc=list()

modelName <- "combo5"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

dataset <- dataset[sample(nrow(dataset)),]

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n=15

inputs <-sample(inputs,n)

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- lm(formula, trainDataset)

Predicted1 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#--------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs)]

train\_labels <- dataset[1:(totalDataset \* training/100),c(target)]

test\_labels <- dataset[(totalDataset \* training/100):totalDataset,c(target)]

x<- data.matrix(trainDataset)

model <- quantregForest(x, train\_labels, nthreads = 5, keep.inbag = FALSE, mtry= round(runif(1,5,15)))

Predicted2 <- predict(model, testDataset)

Actual <- as.double(unlist(test\_labels))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a= 16

b= 656689

c= 632762

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted3 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataSet <- dataset[1:(totalDataset \* training/100),c(inputs,target)]

testDataSet <- dataset[(totalDataset \* training/100):totalDataset,c(inputs,target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- pcr(formula, n, data = trainDataSet, validation = "CV", method = pls.options()$pcralg)

Predicted4 <- predict(model, testDataSet, ncomp=15)

Actual <- as.double(unlist(testDataSet[target]))

#---------------------------------------------------------------------------------------------------------#

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

model <- rpart(formula, trainDataset, method="anova", control = rpart.control(minsplit = 90,cp=0.0001))

Predicted5 <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

#---------------------------------------------------------------------------------------------------------#

Predicted <- (Predicted1 + Predicted2[,2] + Predicted3 + Predicted4[1:3581] +Predicted5)/5

r <- cor(Actual,Predicted )

r <- round(r,2)

r

R <- r \* r

R <- round(R,2)

R

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

mae

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

accuracy

arr\_mae[i] <- mae

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_r, arr\_R, arr\_acc, arr\_mae)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

**Q1.3 Study 5 feature selection techniques for given regression data set and report Top five features.**

**[Hint: Use FSelecter, Boruta, etc]**

A1.3

|  |  |
| --- | --- |
| Feature Selection Technique | Top 5 Features |
| Random Approach | F13, F7, F12, F6, F14 |
| information.gain | F6, F5, F4, F9, F3 |
| relief | F3, F14, F8, F7, F10 |
| gain.ratio | F4, F10, F9, F6, F5 |
| oneR | F4, F10, F9, F6, F5 |

**CODES**

**Random Approach (using nnet)**

library(nnet)

iterations = 10

f1= list()

f2= list()

f3= list()

f4= list()

f5= list()

arr\_a=list()

arr\_b=list()

arr\_c=list()

arr\_r=list()

arr\_R=list()

arr\_acc=list()

arr\_mae=list()

modelName <- "random"

InputDataFileName="E:/101510028/regressionDataSet.csv"

training = 70

dataset <- read.csv(InputDataFileName)

dataset <- dataset[sample(nrow(dataset)),]

totalDataset <- nrow(dataset)

for(i in 0:iterations){

target <- names(dataset)[1]

inputs <- setdiff(names(dataset),target)

n= round(runif(1,1,15),0)

inputs <-sample(inputs,n)

trainDataset <- dataset[1:(totalDataset \* training/100),c(inputs, target)]

testDataset <- dataset[(totalDataset \* training/100):totalDataset,c(inputs, target)]

formula <- as.formula(paste(target, "~", paste(c(inputs), collapse = "+")))

a=round(runif(1,10,100),0)

b=round(runif(1,1,1000000),0)

c=round(runif(1,1,1000000),0)

model <- nnet(formula, trainDataset, size=a, linout=TRUE, skip=TRUE, MaxNWts=b, trace=FALSE, maxit=c)

Predicted <- predict(model, testDataset)

Actual <- as.double(unlist(testDataset[target]))

r <- cor(Actual,Predicted )

r <- round(r,2)

R <- r \* r

R <- round(R,2)

mae <- mean(abs(Actual-Predicted))

mae <- round(mae,2)

accuracy <- mean(abs(Actual-Predicted) <=100)

accuracy <- round(accuracy,4) \*100

arr\_mae[i] <- mae

f1[i] <- inputs[1]

f2[i] <- inputs[2]

f3[i] <- inputs[3]

f4[i] <- inputs[4]

f5[i] <- inputs[5]

arr\_a[i] <- a

arr\_b[i] <- b

arr\_c[i] <- c

arr\_r[i] <- r

arr\_R[i] <- R

arr\_acc[i] <- accuracy

}

data <- cbind(arr\_a, arr\_b, arr\_c, arr\_r, arr\_R, arr\_acc, arr\_mae, f1, f2, f3, f4, f5)

write.csv(data, file=paste("E:/101510028/",modelName,"-regression.csv",sep=''), row.names=TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**information.gain + relief + gain.ratio + oneR**

library(FSelector)

InputDataFileName= "E:/101510028/regressionDataSet.csv"

data <- read.csv(InputDataFileName)

data <- data[sample(nrow(data)),]

wts <- information.gain(Class~., data)

write.csv(wts, file=paste("E:/101510028/fselector\_results.csv", sep = ''), row.names = TRUE)

wts <- relief(Class~., data)

write.csv(wts, file=paste("E:/101510028/relief\_results.csv", sep = ''), row.names = TRUE)

wts <- gain.ratio(Class~., data)

write.csv(wts, file=paste("E:/101510028/gain\_ratio\_results.csv", sep = ''), row.names = TRUE)

wts <- symmetrical.uncertainty(Class~., data)

write.csv(wts, file=paste("E:/101510028/symmetrical\_uncertainity\_results.csv", sep = ''), row.names = TRUE)

wts <- oneR(Class~., data)

write.csv(wts, file=paste("E:/101510028/oneR\_results.csv", sep = ''), row.names = TRUE)

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------