**Session 18 Assignment-1**

5. Problem Statement

1. Use the below given data set

DataSet

2. Perform the below given activities:

a. Create classification model using different decision trees.

b. Verify model goodness of fit.

c. Apply all the model validation techniques.

d. Make conclusions

setwd("C:/Users/Seshan/Desktop/sv R related/acadgild/assignments/session 18 Assign/session18")

library(readr)

Weight\_lift <- read.csv("Weight lift.csv")

View(Weight\_lift)

data1<-Weight\_lift

Weight\_lift

# load libraries

library(caret)

library(randomForest)

library(rpart)

library(rpart.plot)

library(ggplot2)

library(lattice)

library(rattle)

summary(data1)

library(C50)

#install.package('devtools') # Only needed if you dont have this installed.

library(devtools)

install\_github('adam-m-mcelhinney/helpRFunctions')

library(helpRFunctions)

names(data)

dim(data)

library(caret)

library(zoo)

library(plyr)

data<-na.exclude(data1)

is.na(data)

which(is.na(data))

sum(is.na(data))

colSums(is.na(data))

#data[is.na(data)] <- mean(data, na.rm = TRUE)

str(data)

summary(data)

pairs(data[8:15])

# enable multi-core processing

library(doParallel)

#cl <- makeCluster(detectCores())

registerDoParallel()

set.seed(12345)

dataTrain<-data[1:800,]

dataTest<-data[805:4024,]

head(dataTrain)

head(dataTest)

indexNA <- as.vector(sapply(dataTrain[,1:152],function(x) {length(which(is.na(x)))!=0}))

dataTrain <- dataTrain[,!indexNA]

dataTrain<-na.exclude(dataTrain)

library(C50)

head(dataTrain)

head(dataTest)

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

library(tree)

fit <-tree(classe~.,data=dataTrain[,-1])

summary(fit)

#fit

plot(fit)

text(fit)

pred <-predict(fit,dataTest[,-1],type='class')

confusionMatrix(pred,dataTest$classe)

#----

library(rpart)

library(rpart.plot)

fit1 <- rpart(classe~.,data=dataTrain[,-1])

fit1

summary(fit1)

# make predictions

pred <- predict(fit1,dataTest[,-1],type='class')

confusionMatrix(pred,dataTest$classe)

rpart.plot::rpart.plot(fit1)

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

# load libraries

library(caret)

library(rpart)

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="rpart")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=churnTrain, trControl=train\_control, method="C5.0")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

#---------------# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=churnTrain, trControl=train\_control, method="bstTree")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5.0Cost")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5.0Rules")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5.0Tree")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="ctree")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

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

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(classe~., data=dataTrain, trControl=train\_control, method="ctree2")

model

# make predictions

predictions<- predict(model,dataTest)

# append predictions

pred<- cbind(dataTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

confusionMatrix

> setwd("C:/Users/Seshan/Desktop/sv R related/acadgild/assignments/session 18

Assign/session18")

> library(readr)

> Weight\_lift <- read.csv("Weight lift.csv")

> View(Weight\_lift)

> data1<-Weight\_lift

> Weight\_lift

accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z

1 184 -1160 1400 -876

2 182 -1150 1410 -871

3 185 -1130 1400 -863

4 188 -1120 1400 -855

5 188 -1100 1400 -843

6 190 -1090 1400 -838

accel\_forearm\_y.1 accel\_forearm\_z.1 magnet\_forearm\_x.1 magnet\_forearm\_y.

1

1 155 184 -1160 140

0

2 164 182 -1150 141

0

3 172 185 -1130 140

0

4 182 188 -1120 140

0

5 195 188 -1100 140

0

6 207 190 -1090 140

0

magnet\_forearm\_z.1 classe

1 -876 E

2 -871 E

3 -863 E

4 -855 E

5 -843 E

6 -838 E

[ reached getOption("max.print") -- omitted 4018 rows ]

> # load libraries

> library(caret)

> library(randomForest)

> library(rpart)

> library(rpart.plot)

> library(ggplot2)

> library(lattice)

> library(rattle)

Error in library(rattle) : there is no package called ‘rattle’

> summary(data1)

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_time

stamp

adelmo : 311 Min. :1.322e+09 Min. : 297 2/12/2011 13:35 :

311

carlitos:1580 1st Qu.:1.323e+09 1st Qu.:244321 28/11/2011 14:15:

88

eurico : 88 Median :1.323e+09 Median :492342 30/11/2011 17:12:

4

jeremy : 4 Mean :1.323e+09 Mean :490377 5/12/2011 11:23 :

337

pedro :2041 3rd Qu.:1.323e+09 3rd Qu.:736278 5/12/2011 11:25 :1

243

Max. :1.323e+09 Max. :996453 5/12/2011 14:22 :

456

new\_window num\_window roll\_belt pitch\_belt yaw\_belt

no :3936 Min. : 1.00 Min. :-28.90 Min. :-56.20 Min. :-179.00

0

yes: 88 1st Qu.:24.00 1st Qu.: 1.38 1st Qu.: 6.22 1st Qu.: -93.10

0

Median :46.00 Median :122.00 Median : 25.50 Median : -4.94

0

Mean :46.33 Mean : 73.31 Mean : 14.16 Mean : -30.97

5

3rd Qu.:69.00 3rd Qu.:124.00 3rd Qu.: 26.40 3rd Qu.: -2.69

5

Max. :91.00 Max. :159.00 Max. : 60.30 Max. : 179.00

0

total\_accel\_belt kurtosis\_roll\_belt kurtosis\_picth\_belt skewness\_roll\_belt

Min. : 0.00 Min. :-3.333 Min. :-2.1212 Min. :-3.031527

1st Qu.: 3.00 1st Qu.:-1.036 1st Qu.:-0.3913 1st Qu.: 0.005406

Median :19.00 Median :-1.036 Median :-0.3913 Median : 0.005406

Mean :12.77 Mean :-1.027 Mean :-0.3496 Mean : 0.003858

3rd Qu.:20.00 3rd Qu.:-1.036 3rd Qu.:-0.3913 3rd Qu.: 0.005406

Max. :26.00 Max. : 7.515 Max. :54.0000 Max. : 2.713152

skewness\_roll\_belt.1 max\_roll\_belt max\_picth\_belt max\_yaw\_belt

Min. :-6.63325 Min. :-94.400 Min. : 3.00 Min. :-3.3000

1st Qu.: 0.04512 1st Qu.: -4.100 1st Qu.:20.00 1st Qu.:-1.0000

Median : 0.04512 Median : -4.100 Median :20.00 Median :-1.0000

Mean : 0.04011 Mean : -4.626 Mean :19.87 Mean :-0.9917

3rd Qu.: 0.04512 3rd Qu.: -4.100 3rd Qu.:20.00 3rd Qu.:-1.0000

Max. : 7.34847 Max. :179.000 Max. :26.00 Max. : 7.5000

min\_roll\_belt min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt

Min. :-179.000 Min. : 0.00 Min. :-3.3000 Min. : 0.000

1st Qu.: -7.250 1st Qu.:18.00 1st Qu.:-1.0000 1st Qu.: 1.345

Median : -7.250 Median :18.00 Median :-1.0000 Median : 1.345

Mean : -7.838 Mean :17.86 Mean :-0.9917 Mean : 1.446

3rd Qu.: -7.250 3rd Qu.:18.00 3rd Qu.:-1.0000 3rd Qu.: 1.345

Max. : 157.000 Max. :20.00 Max. : 7.5000 Max. :358.000

amplitude\_pitch\_belt amplitude\_yaw\_belt var\_total\_accel\_belt avg\_roll\_belt

Min. : 0.000 Min. :0 Min. : 0.0000 Min. :-27.4

1st Qu.: 2.000 1st Qu.:0 1st Qu.: 0.3000 1st Qu.:121.9

Median : 2.000 Median :0 Median : 0.3000 Median :121.9

Mean : 2.014 Mean :0 Mean : 0.3148 Mean :120.8

3rd Qu.: 2.000 3rd Qu.:0 3rd Qu.: 0.3000 3rd Qu.:121.9

Max. :21.000 Max. :0 Max. :18.2000 Max. :154.5

avg\_yaw\_forearm stddev\_yaw\_forearm var\_yaw\_forearm gyros\_forearm\_x

Min. :-152.33 Min. : 0.00 Min. : 0 Min. :-1.8800

1st Qu.: 17.10 1st Qu.: 74.28 1st Qu.: 5542 1st Qu.:-0.1400

Median : 17.10 Median : 74.28 Median : 5542 Median : 0.0600

Mean : 17.13 Mean : 74.01 Mean : 5578 Mean : 0.1076

3rd Qu.: 17.10 3rd Qu.: 74.28 3rd Qu.: 5542 3rd Qu.: 0.4200

Max. : 132.59 Max. :197.51 Max. :39009 Max. : 1.8100

gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y

Min. :-5.730000 Min. :-2.58000 Min. :-328.000 Min. :-467.00

1st Qu.:-1.780000 1st Qu.:-0.31000 1st Qu.:-117.000 1st Qu.: 75.75

Median :-0.020000 Median :-0.02000 Median : -6.000 Median : 229.50

Mean :-0.004108 Mean : 0.09302 Mean : -6.445 Mean : 171.47

3rd Qu.: 1.830000 3rd Qu.: 0.48000 3rd Qu.: 113.000 3rd Qu.: 297.00

Max. : 5.170000 Max. : 3.35000 Max. : 279.000 Max. : 575.00

accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z

Min. :-366 Min. :-1160.0 Min. :-725.0 Min. :-876.0

1st Qu.:-210 1st Qu.: -589.0 1st Qu.: -76.0 1st Qu.: 370.8

Median :-181 Median : -330.5 Median : 653.0 Median : 560.0

Mean :-163 Mean : -348.7 Mean : 358.6 Mean : 475.2

3rd Qu.:-150 3rd Qu.: -152.0 3rd Qu.: 747.0 3rd Qu.: 670.0

Max. : 239 Max. : 413.0 Max. :1440.0 Max. :1040.0

accel\_forearm\_y.1 accel\_forearm\_z.1 magnet\_forearm\_x.1 magnet\_forearm\_y.1

Min. :-467.00 Min. :-366 Min. :-1160.0 Min. :-725.0

1st Qu.: 75.75 1st Qu.:-210 1st Qu.: -589.0 1st Qu.: -76.0

Median : 229.50 Median :-181 Median : -330.5 Median : 653.0

Mean : 171.47 Mean :-163 Mean : -348.7 Mean : 358.6

3rd Qu.: 297.00 3rd Qu.:-150 3rd Qu.: -152.0 3rd Qu.: 747.0

Max. : 575.00 Max. : 239 Max. : 413.0 Max. :1440.0

magnet\_forearm\_z.1 classe

Min. :-876.0 A:1365

1st Qu.: 370.8 B: 901

Median : 560.0 C: 112

Mean : 475.2 D: 276

3rd Qu.: 670.0 E:1370

Max. :1040.0

[ reached getOption("max.print") -- omitted 1 row ]

> library(C50)

> library(helpRFunctions)

> names(data)

[1] "user\_name" "raw\_timestamp\_part\_1" "raw\_timestamp\_pa

rt\_2"

[4] "cvtd\_timestamp" "new\_window" "num\_window"

[7] "roll\_belt" "pitch\_belt" "yaw\_belt"

[10] "total\_accel\_belt" "kurtosis\_roll\_belt" "kurtosis\_picth\_b

elt"

[13] "skewness\_roll\_belt" "skewness\_roll\_belt.1" "max\_roll\_belt"

[16] "max\_picth\_belt" "max\_yaw\_belt" "min\_roll\_belt"

[19] "min\_pitch\_belt" "min\_yaw\_belt" "amplitude\_roll\_b

elt"

[22] "amplitude\_pitch\_belt" "amplitude\_yaw\_belt" "var\_total\_accel\_

belt"

[25] "avg\_roll\_belt" "stddev\_roll\_belt" "var\_roll\_belt"

[28] "avg\_pitch\_belt" "stddev\_pitch\_belt" "var\_pitch\_belt"

[31] "avg\_yaw\_belt" "stddev\_yaw\_belt" "var\_yaw\_belt"

[34] "gyros\_belt\_x" "gyros\_belt\_y" "gyros\_belt\_z"

[37] "accel\_belt\_x" "accel\_belt\_y" "accel\_belt\_z"

[157] "magnet\_forearm\_z.1" "classe"

> dim(data)

[1] 4024 158

> library(caret)

> library(zoo)

> library(plyr)

> data<-na.exclude(data1)

> is.na(data)

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp new\_w

indow

1 FALSE FALSE FALSE FALSE

FALSE

2 FALSE FALSE FALSE FALSE

FALSE

3 FALSE FALSE FALSE FALSE

FALSE

4 FALSE FALSE FALSE FALSE

FALSE

5 FALSE FALSE FALSE FALSE

FALSE

6 FALSE FALSE FALSE FALSE

FALSE

num\_window roll\_belt pitch\_belt yaw\_belt total\_accel\_belt kurtosis\_roll\_

belt

1 FALSE FALSE FALSE FALSE FALSE F

ALSE

2 FALSE FALSE FALSE FALSE FALSE F

ALSE

3 FALSE FALSE FALSE FALSE FALSE F

ALSE

4 FALSE FALSE FALSE FALSE FALSE F

ALSE

5 FALSE FALSE FALSE FALSE FALSE F

ALSE

6 FALSE FALSE FALSE FALSE FALSE F

ALSE

max\_roll\_arm max\_picth\_arm max\_yaw\_arm min\_roll\_arm min\_pitch\_arm min\_ya

w\_arm

1 FALSE FALSE FALSE FALSE FALSE

FALSE

2 FALSE FALSE FALSE FALSE FALSE

FALSE

3 FALSE FALSE FALSE FALSE FALSE

FALSE

4 FALSE FALSE FALSE FALSE FALSE

FALSE

5 FALSE FALSE FALSE FALSE FALSE

FALSE

6 FALSE FALSE FALSE FALSE FALSE

FALSE

amplitude\_roll\_arm amplitude\_pitch\_arm amplitude\_yaw\_arm roll\_dumbbell

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell kurtosis\_picth\_dumbbe

ll

1 FALSE FALSE FALSE FAL

SE

2 FALSE FALSE FALSE FAL

SE

3 FALSE FALSE FALSE FAL

SE

4 FALSE FALSE FALSE FAL

SE

5 FALSE FALSE FALSE FAL

SE

6 FALSE FALSE FALSE FAL

SE

skewness\_roll\_dumbbell skewness\_pitch\_dumbbell max\_roll\_dumbbell

1 FALSE FALSE FALSE

2 FALSE FALSE FALSE

3 FALSE FALSE FALSE

4 FALSE FALSE FALSE

5 FALSE FALSE FALSE

6 FALSE FALSE FALSE

max\_picth\_dumbbell max\_yaw\_dumbbell min\_roll\_dumbbell min\_pitch\_dumbbell

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

min\_yaw\_dumbbell amplitude\_roll\_dumbbell amplitude\_pitch\_dumbbell

1 FALSE FALSE FALSE

2 FALSE FALSE FALSE

3 FALSE FALSE FALSE

4 FALSE FALSE FALSE

5 FALSE FALSE FALSE

6 FALSE FALSE FALSE

amplitude\_yaw\_dumbbell total\_accel\_dumbbell var\_accel\_dumbbell avg\_roll\_

dumbbell

1 FALSE FALSE FALSE

FALSE

2 FALSE FALSE FALSE

FALSE

3 FALSE FALSE FALSE

FALSE

4 FALSE FALSE FALSE

FALSE

5 FALSE FALSE FALSE

FALSE

6 FALSE FALSE FALSE

FALSE

stddev\_roll\_dumbbell var\_roll\_dumbbell avg\_pitch\_dumbbell stddev\_pitch\_d

umbbell

1 FALSE FALSE FALSE

FALSE

2 FALSE FALSE FALSE

FALSE

3 FALSE FALSE FALSE

FALSE

4 FALSE FALSE FALSE

FALSE

5 FALSE FALSE FALSE

FALSE

6 FALSE FALSE FALSE

FALSE

var\_pitch\_dumbbell avg\_yaw\_dumbbell stddev\_yaw\_dumbbell var\_yaw\_dumbbell

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

gyros\_dumbbell\_x gyros\_dumbbell\_y gyros\_dumbbell\_z accel\_dumbbell\_x

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

accel\_dumbbell\_y accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

magnet\_dumbbell\_z roll\_forearm pitch\_forearm yaw\_forearm kurtosis\_roll\_f

orearm

1 FALSE FALSE FALSE FALSE

FALSE

2 FALSE FALSE FALSE FALSE

FALSE

3 FALSE FALSE FALSE FALSE

FALSE

4 FALSE FALSE FALSE FALSE

FALSE

5 FALSE FALSE FALSE FALSE

FALSE

6 FALSE FALSE FALSE FALSE

FALSE

kurtosis\_picth\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm

1 FALSE FALSE FALSE

2 FALSE FALSE FALSE

3 FALSE FALSE FALSE

4 FALSE FALSE FALSE

5 FALSE FALSE FALSE

6 FALSE FALSE FALSE

max\_roll\_forearm max\_picth\_forearm max\_yaw\_forearm min\_roll\_forearm

1 FALSE FALSE FALSE FALSE

2 FALSE FALSE FALSE FALSE

3 FALSE FALSE FALSE FALSE

4 FALSE FALSE FALSE FALSE

5 FALSE FALSE FALSE FALSE

6 FALSE FALSE FALSE FALSE

magnet\_forearm\_z.1 classe

1 FALSE FALSE

2 FALSE FALSE

3 FALSE FALSE

4 FALSE FALSE

5 FALSE FALSE

6 FALSE FALSE

[ reached getOption("max.print") -- omitted 4018 rows ]

> which(is.na(data))

integer(0)

> sum(is.na(data))

[1] 0

> colSums(is.na(data))

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2

0 0 0

cvtd\_timestamp new\_window num\_window

0 0 0

roll\_belt pitch\_belt yaw\_belt

0 0 0

total\_accel\_belt kurtosis\_roll\_belt kurtosis\_picth\_belt

0 0 0

skewness\_roll\_belt skewness\_roll\_belt.1 max\_roll\_belt

0 0 0

max\_picth\_belt max\_yaw\_belt min\_roll\_belt

0 0 0

min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt

0 0 0

amplitude\_pitch\_belt amplitude\_yaw\_belt var\_total\_accel\_belt

0 0 0

avg\_roll\_belt stddev\_roll\_belt var\_roll\_belt

0 0 0

avg\_pitch\_belt stddev\_pitch\_belt var\_pitch\_belt

0 0 0

avg\_yaw\_belt stddev\_yaw\_belt var\_yaw\_belt

0 0 0

gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z

0 0 0

accel\_belt\_x accel\_belt\_y accel\_belt\_z

0 0 0

magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z

0 0 0

roll\_arm pitch\_arm yaw\_arm

0 0 0

total\_accel\_arm var\_accel\_arm avg\_roll\_arm

0 0 0

stddev\_roll\_arm var\_roll\_arm avg\_pitch\_arm

0 0 0

stddev\_pitch\_arm var\_pitch\_arm avg\_yaw\_arm

0 0 0

stddev\_yaw\_arm var\_yaw\_arm gyros\_arm\_x

0 0 0

gyros\_arm\_y gyros\_arm\_z accel\_arm\_x

0 0 0

accel\_arm\_y accel\_arm\_z magnet\_arm\_x

0 0 0

magnet\_arm\_y magnet\_arm\_z kurtosis\_roll\_arm

0 0 0

kurtosis\_picth\_arm kurtosis\_yaw\_arm skewness\_roll\_arm

0 0 0

skewness\_pitch\_arm skewness\_yaw\_arm max\_roll\_arm

0 0 0

max\_picth\_arm max\_yaw\_arm min\_roll\_arm

0 0 0

min\_pitch\_arm min\_yaw\_arm amplitude\_roll\_arm

0 0 0

amplitude\_pitch\_arm amplitude\_yaw\_arm roll\_dumbbell

0 0 0

pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell

0 0 0

kurtosis\_picth\_dumbbell skewness\_roll\_dumbbell skewness\_pitch\_dumbbell

0 0 0

max\_roll\_dumbbell max\_picth\_dumbbell max\_yaw\_dumbbell

0 0 0

min\_roll\_dumbbell min\_pitch\_dumbbell min\_yaw\_dumbbell

0 0 0

amplitude\_roll\_dumbbell amplitude\_pitch\_dumbbell amplitude\_yaw\_dumbbell

0 0 0

total\_accel\_dumbbell var\_accel\_dumbbell avg\_roll\_dumbbell

0 0 0

stddev\_roll\_dumbbell var\_roll\_dumbbell avg\_pitch\_dumbbell

0 0 0

stddev\_pitch\_dumbbell var\_pitch\_dumbbell avg\_yaw\_dumbbell

0 0 0

stddev\_yaw\_dumbbell var\_yaw\_dumbbell gyros\_dumbbell\_x

0 0 0

gyros\_dumbbell\_y gyros\_dumbbell\_z accel\_dumbbell\_x

0 0 0

accel\_dumbbell\_y accel\_dumbbell\_z magnet\_dumbbell\_x

0 0 0

magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm

0 0 0

pitch\_forearm yaw\_forearm kurtosis\_roll\_forearm

0 0 0

kurtosis\_picth\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm

0 0 0

max\_roll\_forearm max\_picth\_forearm max\_yaw\_forearm

0 0 0

min\_roll\_forearm min\_pitch\_forearm min\_yaw\_forearm

0 0 0

amplitude\_roll\_forearm amplitude\_pitch\_forearm amplitude\_yaw\_forearm

0 0 0

total\_accel\_forearm var\_accel\_forearm avg\_roll\_forearm

0 0 0

stddev\_roll\_forearm var\_roll\_forearm avg\_pitch\_forearm

0 0 0

stddev\_pitch\_forearm var\_pitch\_forearm avg\_yaw\_forearm

0 0 0

stddev\_yaw\_forearm var\_yaw\_forearm gyros\_forearm\_x

0 0 0

gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x

0 0 0

accel\_forearm\_y accel\_forearm\_z magnet\_forearm\_x

0 0 0

magnet\_forearm\_y magnet\_forearm\_z accel\_forearm\_y.1

0 0 0

accel\_forearm\_z.1 magnet\_forearm\_x.1 magnet\_forearm\_y.1

0 0 0

magnet\_forearm\_z.1 classe

0 0

> str(data)

'data.frame': 4024 obs. of 158 variables:

$ user\_name : Factor w/ 5 levels "adelmo","carlitos",..: 3 3 3

3 3 3 3 3 3 3 ...

$ raw\_timestamp\_part\_1 : int 1322489729 1322489729 1322489729 1322489729

1322489729 1322489729 1322489729 1322489729 1322489729 1322489729 ...

$ raw\_timestamp\_part\_2 : int 34670 62641 70653 82654 90637 170626 190665

242723 267551 274689 ...

$ cvtd\_timestamp : Factor w/ 7 levels "2/12/2011 13:35",..: 2 2 2 2

2 2 2 2 2 2 ...

$ new\_window : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1

1 ...

$ num\_window : int 1 1 1 1 1 1 1 1 1 1 ...

$ roll\_belt : num 3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31

2 ...

$ pitch\_belt : num 41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.

4 47.7 ...

$ yaw\_belt : num -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -

82.2 -82.6 -82.8 ...

$ total\_accel\_belt : int 3 2 1 1 1 1 3 4 2 3 ...

$ kurtosis\_roll\_belt : num -1.04 -1.04 -1.04 -1.04 -1.04 ...

$ kurtosis\_picth\_belt : num -0.391 -0.391 -0.391 -0.391 -0.391 ...

$ skewness\_roll\_belt : num 0.00541 0.00541 0.00541 0.00541 0.00541 ...

$ skewness\_roll\_belt.1 : num 0.0451 0.0451 0.0451 0.0451 0.0451 ...

$ max\_roll\_belt : num -4.1 -4.1 -4.1 -4.1 -4.1 -4.1 -4.1 -4.1 -4.

1 -4.1 ...

$ max\_picth\_belt : int 20 20 20 20 20 20 20 20 20 20 ...

$ max\_yaw\_belt : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

$ min\_roll\_belt : num -7.25 -7.25 -7.25 -7.25 -7.25 -7.25 -7.25 -

7.25 -7.25 -7.25 ...

$ min\_pitch\_belt : int 18 18 18 18 18 18 18 18 18 18 ...

$ min\_yaw\_belt : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

$ amplitude\_roll\_belt : num 1.34 1.34 1.34 1.34 1.34 ...

$ amplitude\_pitch\_belt : int 2 2 2 2 2 2 2 2 2 2 ...

$ amplitude\_yaw\_belt : int 0 0 0 0 0 0 0 0 0 0 ...

$ var\_total\_accel\_belt : num 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 ...

$ avg\_roll\_belt : num 122 122 122 122 122 ...

$ stddev\_roll\_belt : num 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 ...

$ var\_roll\_belt : num 0.35 0.35 0.35 0.35 0.35 0.35 0.35 0.35 0.3

5 0.35 ...

$ avg\_pitch\_belt : num 25.8 25.8 25.8 25.8 25.8 ...

$ stddev\_pitch\_belt : num 0.35 0.35 0.35 0.35 0.35 0.35 0.35 0.35 0.3

5 0.35 ...

$ var\_pitch\_belt : num 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...

$ avg\_yaw\_belt : num -4.95 -4.95 -4.95 -4.95 -4.95 -4.95 -4.95 -

4.95 -4.95 -4.95 ...

$ stddev\_yaw\_belt : num 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 ...

$ var\_yaw\_belt : num 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.1

7 0.17 ...

$ gyros\_belt\_x : num 2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.65

1.48 ...

$ min\_yaw\_dumbbell : num -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1 -0.1

-0.1 ...

$ amplitude\_roll\_dumbbell : num 55.7 55.7 55.7 55.7 55.7 ...

$ amplitude\_pitch\_dumbbell: num 54.7 54.7 54.7 54.7 54.7 ...

$ amplitude\_yaw\_dumbbell : int 0 0 0 0 0 0 0 0 0 0 ...

$ total\_accel\_dumbbell : int 4 4 4 5 4 4 4 4 4 4 ...

$ var\_accel\_dumbbell : num 2.42 2.42 2.42 2.42 2.42 ...

$ avg\_roll\_dumbbell : num -5.12 -5.12 -5.12 -5.12 -5.12 ...

[list output truncated]

> summary(data)

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_time

stamp

adelmo : 311 Min. :1.322e+09 Min. : 297 2/12/2011 13:35 :

311

carlitos:1580 1st Qu.:1.323e+09 1st Qu.:244321 28/11/2011 14:15:

88

eurico : 88 Median :1.323e+09 Median :492342 30/11/2011 17:12:

4

jeremy : 4 Mean :1.323e+09 Mean :490377 5/12/2011 11:23 :

337

pedro :2041 3rd Qu.:1.323e+09 3rd Qu.:736278 5/12/2011 11:25 :1

243

Max. :1.323e+09 Max. :996453 5/12/2011 14:22 :

456

new\_window num\_window roll\_belt pitch\_belt yaw\_belt

no :3936 Min. : 1.00 Min. :-28.90 Min. :-56.20 Min. :-179.00

0

yes: 88 1st Qu.:24.00 1st Qu.: 1.38 1st Qu.: 6.22 1st Qu.: -93.10

0

Median :46.00 Median :122.00 Median : 25.50 Median : -4.94

0

Mean :46.33 Mean : 73.31 Mean : 14.16 Mean : -30.97

5

3rd Qu.:69.00 3rd Qu.:124.00 3rd Qu.: 26.40 3rd Qu.: -2.69

5

Max. :91.00 Max. :159.00 Max. : 60.30 Max. : 179.00

0

total\_accel\_belt kurtosis\_roll\_belt kurtosis\_picth\_belt skewness\_roll\_belt

Min. : 0.00 Min. :-3.333 Min. :-2.1212 Min. :-3.031527

1st Qu.: 3.00 1st Qu.:-1.036 1st Qu.:-0.3913 1st Qu.: 0.005406

Median :19.00 Median :-1.036 Median :-0.3913 Median : 0.005406

Mean :12.77 Mean :-1.027 Mean :-0.3496 Mean : 0.003858

3rd Qu.:20.00 3rd Qu.:-1.036 3rd Qu.:-0.3913 3rd Qu.: 0.005406

Max. :26.00 Max. : 7.515 Max. :54.0000 Max. : 2.713152

skewness\_roll\_belt.1 max\_roll\_belt max\_picth\_belt max\_yaw\_belt

Min. :-6.63325 Min. :-94.400 Min. : 3.00 Min. :-3.3000

1st Qu.: 0.04512 1st Qu.: -4.100 1st Qu.:20.00 1st Qu.:-1.0000

Median : 0.04512 Median : -4.100 Median :20.00 Median :-1.0000

Mean : 0.04011 Mean : -4.626 Mean :19.87 Mean :-0.9917

3rd Qu.: 0.04512 3rd Qu.: -4.100 3rd Qu.:20.00 3rd Qu.:-1.0000

Max. : 7.34847 Max. :179.000 Max. :26.00 Max. : 7.5000

min\_roll\_belt min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt

Min. :-179.000 Min. : 0.00 Min. :-3.3000 Min. : 0.000

1st Qu.: -7.250 1st Qu.:18.00 1st Qu.:-1.0000 1st Qu.: 1.345

Median : -7.250 Median :18.00 Median :-1.0000 Median : 1.345

Mean : -7.838 Mean :17.86 Mean :-0.9917 Mean : 1.446

3rd Qu.: -7.250 3rd Qu.:18.00 3rd Qu.:-1.0000 3rd Qu.: 1.345

Max. : 575.00 Max. : 239 Max. : 413.0 Max. :1440.0

magnet\_forearm\_z.1 classe

Min. :-876.0 A:1365

1st Qu.: 370.8 B: 901

Median : 560.0 C: 112

Mean : 475.2 D: 276

3rd Qu.: 670.0 E:1370

Max. :1040.0

[ reached getOption("max.print") -- omitted 1 row ]

> pairs(data[8:15])

> # enable multi-core processing

> library(doParallel)

> #cl <- makeCluster(detectCores())

> set.seed(12345)

> dataTrain<-data[1:800,]

> dataTest<-data[805:4024,]

> head(dataTrain)

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp new\_wi

ndow

1 eurico 1322489729 34670 28/11/2011 14:15

no

2 eurico 1322489729 62641 28/11/2011 14:15

no

3 eurico 1322489729 70653 28/11/2011 14:15

no

4 eurico 1322489729 82654 28/11/2011 14:15

no

5 eurico 1322489729 90637 28/11/2011 14:15

no

6 eurico 1322489729 170626 28/11/2011 14:15

no

num\_window roll\_belt pitch\_belt yaw\_belt total\_accel\_belt kurtosis\_roll\_bel

t

1 1 3.70 41.6 -82.8 3 -1.0356

6

2 1 3.66 42.8 -82.5 2 -1.0356

6

3 1 3.58 43.7 -82.3 1 -1.0356

6

4 1 3.56 44.4 -82.1 1 -1.0356

6

5 1 3.57 45.1 -81.9 1 -1.0356

6

6 1 3.45 45.6 -81.9 1 -1.0356

6

kurtosis\_picth\_belt skewness\_roll\_belt skewness\_roll\_belt.1 max\_roll\_belt

1 -0.39133 0.005406 0.045115 -4.1

2 -0.39133 0.005406 0.045115 -4.1

3 -0.39133 0.005406 0.045115 -4.1

4 -0.39133 0.005406 0.045115 -4.1

5 -0.39133 0.005406 0.045115 -4.1

6 -0.39133 0.005406 0.045115 -4.1

max\_picth\_belt max\_yaw\_belt min\_roll\_belt min\_pitch\_belt min\_yaw\_belt

1 20 -1 -7.25 18 -1

2 20 -1 -7.25 18 -1

3 20 -1 -7.25 18 -1

4 20 -1 -7.25 18 -1

magnet\_forearm\_y magnet\_forearm\_z accel\_forearm\_y.1 accel\_forearm\_z.1

805 -420 239 -104 -199

806 -441 219 -123 -204

807 -463 209 -137 -210

808 -477 206 -142 -216

809 -488 188 -152 -216

810 -502 183 -174 -211

magnet\_forearm\_x.1 magnet\_forearm\_y.1 magnet\_forearm\_z.1 classe

805 -335 -420 239 D

806 -293 -441 219 D

807 -275 -463 209 D

808 -247 -477 206 D

809 -212 -488 188 D

810 -201 -502 183 D

> indexNA <- as.vector(sapply(dataTrain[,1:152],function(x) {length(which(is.

na(x)))!=0}))

> dataTrain <- dataTrain[,!indexNA]

> dataTrain<-na.exclude(dataTrain)

> library(C50)

> head(dataTrain)

user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp new\_wi

ndow

1 eurico 1322489729 34670 28/11/2011 14:15

no

2 eurico 1322489729 62641 28/11/2011 14:15

no

3 eurico 1322489729 70653 28/11/2011 14:15

no

4 eurico 1322489729 82654 28/11/2011 14:15

no

5 eurico 1322489729 90637 28/11/2011 14:15

no

6 eurico 1322489729 170626 28/11/2011 14:15

no

num\_window roll\_belt pitch\_belt yaw\_belt total\_accel\_belt kurtosis\_roll\_bel

t

1 1 3.70 41.6 -82.8 3 -1.0356

6

2 1 3.66 42.8 -82.5 2 -1.0356

6

3 1 3.58 43.7 -82.3 1 -1.0356

6

4 1 3.56 44.4 -82.1 1 -1.0356

6

5 1 3.57 45.1 -81.9 1 -1.0356

6

6 1 3.45 45.6 -81.9 1 -1.0356

6

kurtosis\_picth\_belt skewness\_roll\_belt skewness\_roll\_belt.1 max\_roll\_belt

1 -0.39133 0.005406 0.045115 -4.1

2 -0.39133 0.005406 0.045115 -4.1

3 -0.39133 0.005406 0.045115 -4.1

4 -0.39133 0.005406 0.045115 -4.1

5 -0.39133 0.005406 0.045115 -4.1

6 -0.39133 0.005406 0.045115 -4.1

magnet\_forearm\_y magnet\_forearm\_z accel\_forearm\_y.1 accel\_forearm\_z.1

805 -420 239 -104 -199

806 -441 219 -123 -204

807 -463 209 -137 -210

808 -477 206 -142 -216

809 -488 188 -152 -216

810 -502 183 -174 -211

magnet\_forearm\_x.1 magnet\_forearm\_y.1 magnet\_forearm\_z.1 classe

805 -335 -420 239 D

806 -293 -441 219 D

807 -275 -463 209 D

808 -247 -477 206 D

809 -212 -488 188 D

810 -201 -502 183 D

> #------------

> library(tree)

> fit <-tree(classe~.,data=dataTrain[,-1])

> summary(fit)

Classification tree:

tree(formula = classe ~ ., data = dataTrain[, -1])

Variables actually used in tree construction:

[1] "raw\_timestamp\_part\_1"

Number of terminal nodes: 3

Residual mean deviance: 0 = 0 / 797

Misclassification error rate: 0 = 0 / 800

> #fit

> plot(fit)

> text(fit)

> pred <-predict(fit,dataTest[,-1],type='class')

> confusionMatrix(pred,dataTest$classe)

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 0 0 0 0 0

B 0 0 0 0 0

C 0 0 0 0 0

D 1028 901 112 212 967

E 0 0 0 0 0

Overall Statistics

Accuracy : 0.0658

95% CI : (0.0575, 0.075)

No Information Rate : 0.3193

P-Value [Acc > NIR] : 1

Kappa : 0

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 0.0000 0.0000 0.00000 1.00000 0.0000

Specificity 1.0000 1.0000 1.00000 0.00000 1.0000

Pos Pred Value NaN NaN NaN 0.06584 NaN

Neg Pred Value 0.6807 0.7202 0.96522 NaN 0.6997

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.0000 0.0000 0.00000 0.06584 0.0000

Detection Prevalence 0.0000 0.0000 0.00000 1.00000 0.0000

Balanced Accuracy 0.5000 0.5000 0.50000 0.50000 0.5000

>

> #----

> library(rpart)

> library(rpart.plot)

> fit1 <- rpart(classe~.,data=dataTrain[,-1])

> fit1

n= 800

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 800 397 E (0.42 0 0 0.075 0.5)

2) raw\_timestamp\_part\_1>=1.322959e+09 397 60 A (0.85 0 0 0.15 0)

4) raw\_timestamp\_part\_1< 1.323084e+09 337 0 A (1 0 0 0 0) \*

5) raw\_timestamp\_part\_1>=1.323084e+09 60 0 D (0 0 0 1 0) \*

3) raw\_timestamp\_part\_1< 1.322959e+09 403 0 E (0 0 0 0 1) \*

> summary(fit1)

Call:

rpart(formula = classe ~ ., data = dataTrain[, -1])

n= 800

CP nsplit rel error xerror xstd

1 0.8488665 0 1.0000000 1.0000000 0.03562151

2 0.1511335 1 0.1511335 0.1511335 0.01876532

3 0.0100000 2 0.0000000 0.0000000 0.00000000

Variable importance

num\_window raw\_timestamp\_part\_1 accel\_forearm\_z accel\_forea

rm\_z.1

17 17 13

13

magnet\_belt\_x magnet\_dumbbell\_z pitch\_arm pitc

h\_belt

13 13 4

4

roll\_arm yaw\_belt

4 4

Node number 1: 800 observations, complexity param=0.8488665

predicted class=E expected loss=0.49625 P(node) =1

class counts: 337 0 0 60 403

probabilities: 0.421 0.000 0.000 0.075 0.504

left son=2 (397 obs) right son=3 (403 obs)

Primary splits:

raw\_timestamp\_part\_1 < 1322959000 to the right, improve=348.6635, (0 mi

ssing)

cvtd\_timestamp splits as RRRLL--, improve=348.6635, (0 missing)

num\_window < 10.5 to the right, improve=348.6635, (0 mi

ssing)

magnet\_belt\_x < 29.5 to the left, improve=348.6635, (0 mi

ssing)

accel\_forearm\_z < -168 to the left, improve=348.6635, (0 mi

ssing)

Surrogate splits:

num\_window < 10.5 to the right, agree=1.000, adj=1.000, (0

split)

magnet\_belt\_x < 29.5 to the left, agree=1.000, adj=1.000, (0

split)

accel\_forearm\_z < -168 to the left, agree=1.000, adj=1.000, (0

split)

accel\_forearm\_z.1 < -168 to the left, agree=1.000, adj=1.000, (0

split)

magnet\_dumbbell\_z < 10.5 to the left, agree=0.996, adj=0.992, (0

split)

Node number 2: 397 observations, complexity param=0.1511335

predicted class=A expected loss=0.1511335 P(node) =0.49625

class counts: 337 0 0 60 0

probabilities: 0.849 0.000 0.000 0.151 0.000

left son=4 (337 obs) right son=5 (60 obs)

Primary splits:

raw\_timestamp\_part\_1 < 1323084000 to the left, improve=101.864, (0 mis

sing)

cvtd\_timestamp splits as ---LR--, improve=101.864, (0 missing)

num\_window < 18.5 to the left, improve=101.864, (0 mis

sing)

pitch\_belt < 6.925 to the right, improve=101.864, (0 mis

sing)

yaw\_belt < -93.6 to the left, improve=101.864, (0 mis

sing)

Surrogate splits:

num\_window < 18.5 to the left, agree=1, adj=1, (0 split)

pitch\_belt < 6.925 to the right, agree=1, adj=1, (0 split)

yaw\_belt < -93.6 to the left, agree=1, adj=1, (0 split)

roll\_arm < -37.7 to the left, agree=1, adj=1, (0 split)

pitch\_arm < 5.175 to the right, agree=1, adj=1, (0 split)

Node number 3: 403 observations

predicted class=E expected loss=0 P(node) =0.50375

class counts: 0 0 0 0 403

probabilities: 0.000 0.000 0.000 0.000 1.000

Node number 4: 337 observations

predicted class=A expected loss=0 P(node) =0.42125

class counts: 337 0 0 0 0

probabilities: 1.000 0.000 0.000 0.000 0.000

Node number 5: 60 observations

predicted class=D expected loss=0 P(node) =0.075

class counts: 0 0 0 60 0

probabilities: 0.000 0.000 0.000 1.000 0.000

> # make predictions

> pred <- predict(fit1,dataTest[,-1],type='class')

> confusionMatrix(pred,dataTest$classe)

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 0 0 0 0 0

B 0 0 0 0 0

C 0 0 0 0 0

D 1028 901 112 212 967

E 0 0 0 0 0

Overall Statistics

Accuracy : 0.0658

95% CI : (0.0575, 0.075)

No Information Rate : 0.3193

P-Value [Acc > NIR] : 1

Kappa : 0

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 0.0000 0.0000 0.00000 1.00000 0.0000

Specificity 1.0000 1.0000 1.00000 0.00000 1.0000

Pos Pred Value NaN NaN NaN 0.06584 NaN

Neg Pred Value 0.6807 0.7202 0.96522 NaN 0.6997

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.0000 0.0000 0.00000 0.06584 0.0000

Detection Prevalence 0.0000 0.0000 0.00000 1.00000 0.0000

Balanced Accuracy 0.5000 0.5000 0.50000 0.50000 0.5000

> rpart.plot::rpart.plot(fit1)

> # load libraries

> library(caret)

> library(rpart)

>

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="rp

art")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=churnTrain, trControl=train\_control, method="C

5.0")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

> #---------------# define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=churnTrain, trControl=train\_control, method="b

stTree")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

>

> #---------------

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5

.0Cost")

Error: One or more factor levels in the outcome has no data: 'B', 'C'

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

> #---------------

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5

.0Rules")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

> #---------------

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="C5

.0Tree")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

>

> #---------------

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="ct

ree")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530

>

> #---------------

> # define training control

> train\_control<- trainControl(method="cv", number=10)

>

> # train the model

> model<- train(classe~., data=dataTrain, trControl=train\_control, method="ct

ree2")

> model

Conditional Inference Tree

4000 samples

157 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 3602, 3600, 3600, 3599, 3601, 3599, ...

Resampling results across tuning parameters:

maxdepth mincriterion Accuracy Kappa

1 0.01 0.5995049 0.3914677

1 0.50 0.5995049 0.3914677

1 0.99 0.5995049 0.3914677

2 0.01 0.7666999 0.6506937

2 0.50 0.7666999 0.6506937

2 0.99 0.7666999 0.6506937

3 0.01 0.9307515 0.8998373

3 0.50 0.9307515 0.8998373

3 0.99 0.9307515 0.8998373

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were maxdepth = 3 and mincriterion = 0.01

.

> # make predictions

> predictions<- predict(model,dataTest)

>

> # append predictions

> pred<- cbind(dataTest,predictions)

>

> # summarize results

> confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)

> confusionMatrix

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1028 0 0 0 0

B 0 901 0 0 0

C 0 0 112 0 0

D 0 0 0 0 0

E 0 0 0 212 967

Overall Statistics

Accuracy : 0.9342

95% CI : (0.925, 0.9425)

No Information Rate : 0.3193

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9071

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.00000 0.00000 1.0000

Specificity 1.0000 1.0000 1.00000 1.00000 0.9059

Pos Pred Value 1.0000 1.0000 1.00000 NaN 0.8202

Neg Pred Value 1.0000 1.0000 1.00000 0.93416 1.0000

Prevalence 0.3193 0.2798 0.03478 0.06584 0.3003

Detection Rate 0.3193 0.2798 0.03478 0.00000 0.3003

Detection Prevalence 0.3193 0.2798 0.03478 0.00000 0.3661

Balanced Accuracy 1.0000 1.0000 1.00000 0.50000 0.9530