#29-11-17 Hbd Alice

#train <- read.csv("~/pml-training")

train <- read.csv("pml-training.csv")

test <- read.csv("pml-testing.csv")

library(caret)

intrain <- createDataPartition(y=train$classe, p = 0.7, list = FALSE)

train1 <- train[intrain,]

test1 < - train[-intrain,]

summary(train1)

titles(train1)

??titles

c<- intersect(colnames(test), colnames(train))

?c

cin<- intersect(colnames(test), colnames(train))

summary(train1[,-colnames(c((cin)))])

whatt <- test[which(!(colnames(test)%in%cin)),]

colnames(test)

test$problem\_id

colnames(train[which(!(colnames(test)==colnames(train)))])

#the last column of test data is labelled problems\_id

#put it all in?

#model1 <- train(classe~. , method = "rf", data = train1)

#bad idea..taking forever

predict\_model1 <- predict(model1, test1)

length(predict\_model1)

confusionMatrix(test1$classe, predict(model1, test1))

predict(model1, test1)

summary(train1$classe)

model1

test1\_1 <- test1

test1\_1 <- test1\_1[,-c(160)]

confusionMatrix(test1$classe, predict(model1, test1\_1))

predict\_model1 <- (predict(model1, newdata = test1\_1))

length(predict\_model)

summary(train1)

train1\_adelmo <- train1[which(train1$user\_name == "adelmo"),]

train1\_carlitos <- train1[which(train1$user\_name == "carlitos"),]

train1\_charles <- train1[which(train1$user\_name == "charles"),]

train1\_eurico <- train1[which(train1$user\_name == "eurico"),]

train1\_jeremy <- train1[which(train1$user\_name == "jeremy"),]

train1\_pedro <- train1[which(train1$user\_name == "pedro"),]

summary(train1$user\_name)

#rpart

library(caret)

model\_rpart1 <- train(classe ~ . , method = "rpart", data = train1\_adelmo)

model\_rpart1$finalModel

library(rattle)

fancyRpartPlot(model\_rpart1$finalModel)

test1\_adelmo <- test1[which(test1$user\_name == "adelmo"),]

predict1\_adelmo <- predict(model\_rpart1, test1\_adelmo)

confusionMatrix(predict1\_adelmo, test1\_adelmo$classe)

length(predict1\_adelmo)

view(test1\_adelmo)

#correlation

train2 <- train[,-c(1:7,160)]

corre<- abs(cor(train2[,]))

cn <- colnames(train)

length(train[!complete.cases(train),])

null1 <- data.frame(cn)

for(i in 1:160) {

null1[i,2] <- sum((train[,i]== ""))}

for(i in 1:160) {

null1[i,3] <- sum(is.na(train[,i]))}

train6 <- na.omit(train)

table(train6$user\_name)

#based on null deleted columns with large missing values ASSUMING limited/inconsistent predictive capabilities

train3 <- train[,-c(1,2:7,12:17,20,23,26,69:74,87:92,95,98,101,125:130,133,136,139)]

#actual correlation

corre <- abs(cor((train3[,-c(1,120)]),,))

diag(corre) <- 0

which(corre>0.8, arr.ind = T)

m.rp.2 <- train(classe ~ . , method = "rpart", data = train3)

library(rattle)

fancyRpartPlot(m.rp.2$finalModel)

m.rf.2 <- train(classe ~ . , method = "rf", data = train3)

p3 <- predict(m.rf.2,newdata = test1)

length(p3)

summary(p3)

#replacing NA with 0s (not sure if it's the right thing..hoping to solve the limited no of predictions problem)

train4 <- train3

for(i in 1:120){

for(j in 1:19622){

if(is.na(train4[j,i])){

train4[j,i] <- 0

}

}

}

library(caret)

m.rp.3 <- train(classe ~ . , method = "rpart", data = train4)

library(rattle)

fancyRpartPlot(m.rp.3$finalModel)

p4 <- predict(m.rp.3, test1)

length(p4)

#missing predictions problems not solved :/

train5 <- train1

for(i in 1:120){

for(j in 1:19622){

if(is.na(train5[j,i])){

train5[j,i] <- 0

}

}

}

m.rp.4 <- train(classe ~ . , method = "rpart", data = train5)

fancyRpartPlot(m.rp.4$finalModel)

p.rp.4 <- predict(m.rp.4, test1)

length(p.rp.4)

train6 <- na.omit(train)

m.rp.5 <- train(classe ~ . , method = "rpart", data = train6)

fancyRpartPlot(m.rp.5$finalModel)

p.rp.5 <- predict(m.rp.5, newdata = test)

m.ct.1<- train(classe ~ . , method = "cart", data = train6)

p.ct.1<- predict(m.ct.1, newdata = test)

length(p.rp.5)

length(p.ct.1)

test6 <- test

for(i in 1:120){

for(j in 1:6){

if(is.na(train5[j,i])){

test6[j,i] <- 0

}

}

}

p.rp.5 <- predict(m.rp.5, newdata = test6)

warnings()

null2 <- data.frame(cn)

for(i in 1:160) {

null2[i,2] <- sum((test[,i]== ""))}

for(i in 1:160) {

null2[i,3] <- sum(is.na(test[,i]))}

#removing columns not present in test data set

test7 <- test[,-c(12:36,50:59,69:83,87:101,103:112,125:139,141:150)]

train7 <- train[,-c(12:36,50:59,69:83,87:101,103:112,125:139,141:150)]

#rpart and rf

m.rp.6 <- train(classe ~ . , method = "rpart", data = train7)

library(rattle)

fancyRpartPlot(m.rp.6$finalModel)

p.rp.6 <- predict(m.rp.6, test7)

m.rf.3 <- train(classe ~ . , method = "rf", data = train7)

p.rf.3 <- predict(m.rf.3, test7)

length(p.rp.6)

length(p.rf.3)

#the predictions are all A :/

#removing X

train8 <- train7[,-1]

test8 <- test7[,-1]

#cross-validation 3 fold 5 times

train9 <- train8[,-59]

train.label <- as.factor(train8$classe)

cv.3folds <- createMultiFolds(train.label, k = 3, times = 5)

ctrl1 <- trainControl(method = "repeatedcv", number = 3, repeats = 5, index = cv.3folds)

library(doSNOW)

cl <- makeCluster(3, type = "SOCK")

registerDoSNOW(cl)

m.rf.5 <- train( x = train9, y = train.label, method = "rf", trControl = ctrl1)

stopCluster(cl)

test9 <- test8[,-59]

m.rf.5$finalModel

#Type of random forest: classification

#Number of trees: 500

#No. of variables tried at each split: 30

#OOB estimate of error rate: 0.07%

#Confusion matrix:

# A B C D E class.error

#A 5580 0 0 0 0 0.0000000000

#B 1 3795 1 0 0 0.0005267316

#C 0 3 3418 1 0 0.0011689071

#D 0 0 4 3210 2 0.0018656716

#E 0 0 0 2 3605 0.0005544774

m.rf.4 <- train(classe ~ . , method = "rf", data = train8)

p.rf.4 <- predict(m.rf.4, test8)

p.rf.4

#mtry Accuracy Kappa

#2 0.9905045 0.9879890

#41 0.9991620 0.9989403

#80 0.9983589 0.9979248

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

#The final value used for the model was mtry = 41.

#m.rf.4 has all the 20 test variables correctly predicted