Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | getwd() |
|  | download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", "training.csv") |
|  | download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", "testing.csv") |
|  | training.csv <- read.csv("training.csv") |
|  | testing.csv <- read.csv("testing.csv") |
|  | require("caret") |
|  | require("tidyverse") |
|  | require("knitr") |
|  |  |
|  | ## The goal of your project is to predict the manner in which they did the exercise. |
|  | ## This is the "classe" variable in the training set. |
|  | ## You may use any of the other variables to predict with. |
|  | ## You should create a report describing how you built your model, |
|  | ## how you used cross validation, |
|  | ## what you think the expected out of sample error is, |
|  | ## and why you made the choices you did. |
|  | ## You will also use your prediction model to predict 20 different test cases. |
|  | ## Peer Review Portion |
|  |  |
|  | ## Your submission for the Peer Review portion should consist of a link to a Github repo with your R markdown |
|  | ## and compiled HTML file describing your analysis. |
|  | ## Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. |
|  | ## It will make it easier for the graders if you submit a repo with a gh-pages branch |
|  | ## so the HTML page can be viewed online (and you always want to make it easy on graders :-). |
|  | ## Course Project Prediction Quiz Portion |
|  |  |
|  | ## Apply your machine learning algorithm to the 20 test cases available in the test data above |
|  | ## and submit your predictions in appropriate format to the Course Project Prediction Quiz for automated grading. |
|  |  |
|  | ## explore the training data |
|  | training.csv$classe <- as.factor(training.csv$classe) |
|  | head(training.csv) |
|  | summary(training.csv) |
|  | head(training.csv$classe) |
|  |  |
|  |  |
|  | ## set seed for reproducibility |
|  | set.seed(1024) |
|  |  |
|  |  |
|  |  |
|  | ## eliminte missing values by imputation |
|  |  |
|  | sum(is.na(training.csv$classe)) |
|  | sum(is.na(training.csv))/67 |
|  | 19622-sum(is.na(training.csv))/67 |
|  | ## for 19216 of 19622 cases the same 67 variables are na. |
|  | ## that is too much, so we exclude those variables |
|  | training.csv.old <- training.csv |
|  | training.csv <- select(training.csv, - which(is.na(training.csv[1,])) ) |
|  | ## now the dataset contains no missing values |
|  | sum(is.na(training.csv)) |
|  |  |
|  | ?train |
|  | ?trControl |
|  | ## apply the same selection to testing.csv |
|  | testing.csv <- select(testing.csv, - which(is.na(training.csv.old[1,])) ) |
|  | sum(is.na(testing.csv)) |
|  |  |
|  | ## unfortunately there are still 66 variables in the testing set which only contain NAs. |
|  | ## let's exclude those as well |
|  | testing.csv.old <- testing.csv |
|  | testing.csv <- select(testing.csv, - which(is.na(testing.csv[1,])) ) |
|  | ## now the dataset contains no missing values |
|  | sum(is.na(testing.csv)) |
|  |  |
|  | ## and now apply thisselection to training.csv |
|  | training.csv <- select(training.csv, - which(is.na(testing.csv.old[1,])) ) |
|  | sum(is.na(testing.csv)) |
|  |  |
|  | ## this leaves 60 variables for our model. |
|  |  |
|  | ## there are vars with near zero variance, remve those |
|  | nsv <- nearZeroVar(training.csv,saveMetrics=TRUE) |
|  | training.csv <- training.csv[,!nsv$nzv] |
|  | testing.csv <- testing.csv[,!nsv$nzv] |
|  |  |
|  | ## this leaves 59 variables for our model. |
|  |  |
|  | ## and finally remove the unnecassary variables |
|  | # Remove unnecessary columns |
|  | training.csv <- select(training.csv,-X, - user\_name,-raw\_timestamp\_part\_1,-raw\_timestamp\_part\_2,-cvtd\_timestamp,-num\_window ) |
|  |  |
|  | testing.csv <- select(testing.csv, -X, - user\_name,-raw\_timestamp\_part\_1,-raw\_timestamp\_part\_2,-cvtd\_timestamp,-num\_window,-problem\_id ) |
|  |  |
|  | ## this leaves 54 variables for our model. |
|  |  |
|  |  |
|  |  |
|  |  |
|  | ## check correlation between all vars and our outcome var classe |
|  | cor <- abs(sapply(colnames(training[, -ncol(training)]), function(x) cor(as.numeric(training[, x]), as.numeric(training$classe), method = "spearman"))) |
|  | cor |
|  | summary(training.csv) |
|  |  |
|  | ## two sets |
|  | set.seed(1024) |
|  | inTrain <- createDataPartition(y = training.csv$classe, p = .75, list = FALSE) |
|  | training <- training.csv[ inTrain,] |
|  | validation <- training.csv[-inTrain,] |
|  | nrow(training) |
|  |  |
|  | ## vgl quiz week 4 |
|  | train\_ex <- training[abs(runif(1000, 1, length(training[,1]))),] |
|  | rf = train(classe ~ ., data = train\_ex, method = "rf") |
|  | gbm = train(classe ~ ., data = train\_ex, method = "gbm") |
|  |  |
|  | pred\_rf <- predict(rf, testing) |
|  | pred\_gbm <- predict(gbm, testing) |
|  |  |
|  | confusionMatrix(pred\_rf, testing$classe)$overall[1] |
|  | confusionMatrix(pred\_gbm, testing$classe)$overall[1] |
|  |  |
|  | predDF <- data.frame(pred\_rf, pred\_gbm, testing$classe) |
|  |  |
|  | testing\_20 <- read.csv("testing.csv") |
|  | pred\_20 <- predict(rf, testing\_20) |
|  |  |
|  | ##overnieuw met alletestdata |
|  | rf = train(classe ~ ., data = training, method = "rf") |
|  | pred\_rf <- predict(rf, testing) |
|  | confusionMatrix(pred\_rf, testing$classe)$overall[1] |
|  | predDF <- data.frame(pred\_rf, pred\_gbm, testing$classe) |
|  |  |
|  | testing\_20 <- read.csv("testing.csv") |
|  | pred\_20 <- predict(rf, testing\_20) |
|  |  |
|  | imp <- varImp(rf)$importance |
|  | varImpPlot(rf$finalModel, sort = TRUE, type = 1, pch = 19, col = 1, cex = 1, main = "Importance of the Predictors") |
|  | varImpPlot(rf$finalModel, sort = TRUE) |
|  |  |
|  | sum(pred\_rf[predDF$pred\_rf == predDF$pred\_gbm] == |
|  | predDF$testing.classe[predDF$pred\_rf == predDF$pred\_gbm]) / |
|  | sum(predDF$pred\_rf == predDF$pred\_gbm) |
|  |  |
|  |  |
|  |  |
|  |  |
|  | ## rf |
|  | training[abs(runif(1000, 1, length(training[,1]))),] |
|  | rfFit <- train(classe ~ ., method = "rf", data = training[abs(runif(1000, 1, length(training[,1]))),], importance = T, trControl = trainControl(method = "cv", number = 4)) |
|  |  |
|  | rfFit <- train(classe ~ ., method = "rf", data = training, importance = T, trControl = trainControl(method = "cv", number = 4)) |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  | # Check model performance |
|  | confusionMatrix(testing\_pred,testing$classe) |
|  |  |
|  | ## fastAdaboost |
|  | adaboostFit <- train(classe ~ ., method = "adaboost", data = training) |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  |  |
|  | ## cforest |
|  | rfFit <- train(classe ~ ., method = "cforest", data = training) |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  |  |
|  |  |
|  |  |
|  |  |
|  | # Fit rf model |
|  | rfFit <- train(classe ~ ., method = "rf", data = training, importance = T, trControl = trainControl(method = "cv", number = 4)) |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  | # Check model performance |
|  | confusionMatrix(testing\_pred,testing$classe) |
|  |  |
|  |  |
|  | ## rf |
|  |  |
|  | set.seed(5678) |
|  | inTrain <- createDataPartition(y = training.csv$classe, p = .75, list = FALSE) |
|  | training <- training.csv[ inTrain,] |
|  | testing <- training.csv[-inTrain,] |
|  | nrow(training) |
|  | # Fit rf model |
|  | rfFit <- train(classe ~ ., method = "rf", data = training) |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  | # Check model performance |
|  | confusionMatrix(testing\_pred,testing$classe) |
|  |  |
|  |  |
|  | # Check important variable |
|  | imp <- varImp(rfFit)$importance |
|  | varImpPlot(rfFit$finalModel, sort = TRUE, type = 1, pch = 19, col = 1, cex = 1, main = "Importance of the Predictors") |
|  |  |
|  |  |
|  | testing\_pred <- predict(rfFit, newdata=testing) |
|  | summary(testing\_pred) |
|  |  |
|  | # and then the 20 subj testing set |
|  |  |
|  | testing\_pred <- predict(rfFit, newdata=testing.csv) |
|  | summary(testing\_pred) |
|  |  |
|  | write\_files <- function(x) { |
|  | n <- length(x) |
|  | for (i in 1:n) { |
|  | filename <- paste0("problem\_id", i, ".txt") |
|  | write.table(x[i], file=filename, quote=FALSE, row.names=FALSE,col.names=FALSE) |
|  | } |
|  | } |
|  | write\_files(testing\_pred) |
|  |  |
|  | ## decision tree. |
|  | ?rpart |
|  | modFitDT <- rpart(classe ~ ., data = training, method="class") |
|  | fancyRpartPlot(modFitDT) |
|  |  |
|  | set.seed(12345) |
|  |  |
|  | prediction <- predict(modFitDT, testing, type = "class") |
|  | confusionMatrix(prediction, testing$classe) |
|  |  |
|  | ## rf |
|  | set.seed(12345) |
|  | modFitRF <- randomForest(classe ~ ., data = training, ntree = 1000) |
|  |  |
|  | prediction <- predict(modFitRF, testing, type = "class") |
|  | confusionMatrix(prediction, testing$classe) |
|  |  |
|  | predictionDT <- predict(modFitDT, dt\_testing, type = "class") |
|  | predictionDT |
|  |  |
|  | predictionRF <- predict(modFitRF, dt\_testing, type = "class") |
|  | predictionRF |
|  |  |
|  | pml\_write\_files = function(x){ |
|  | n = length(x) |
|  | for(i in 1:n){ |
|  | filename = paste0("problem\_id\_",i,".txt") |
|  | write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE) |
|  | } |
|  | } |
|  |  |
|  | pml\_write\_files(predictionRF) |
|  |  |
|  | ## boost? |
|  |  |
|  | ?fancyRpartPlot |