

Majority of Project Code

```
#open csv files
test <- read.csv('machine-testing.csv')
train <- read.csv('machine-training.csv')

# Explore the dataset and determine which of the predictors are worth keeping in the
analysis.
# Eliminate NA columns and create data set to test
ds <- train[ , colSums(is.na(train)) < 9811]
# Convert classe into dichotomous outcome
ds$classe <- ifelse(ds$classe=='A', 1, 0)

#long method
dv <- ds
summary(dv)
  #individually remove all empty variables
dv$kurtosis_yaw_belt<-NULL
dv$kurtosis_roll_belt<-NULL
dv$skewness_roll_belt<-NULL
dv$skewness_roll_belt<-NULL
dv$yaw_belt<-NULL
dv$max_yaw_belt<-NULL
dv$min_yaw_belt<-NULL
dv$amplitude_yaw_belt<-NULL
dv$kurtosis_roll_arm<-NULL
dv$kurtosis_pitch_arm<-NULL
dv$kurtosis_yaw_arm<-NULL
dv$skewness_roll_arm<-NULL
dv$skewness_pitch_arm<-NULL
dv$skewness_yaw_arm<-NULL
dv$kurtosis_roll_dumbbell<-NULL
dv$kurtosis_pitch_dumbbell<-NULL
dv$kurtosis_yaw_dumbbell<-NULL
dv$skewness_roll_dumbbell<-NULL
dv$skewness_pitch_dumbbell<-NULL
dv$skewness_yaw_dumbbell<-NULL
dv$max_yaw_dumbbell<-NULL
dv$min_yaw_dumbbell<-NULL
dv$amplitude_yaw_dumbbell<-NULL
dv$kurtosis_roll_forearm<-NULL
dv$kurtosis_pitch_forearm<-NULL
dv$kurtosis_yaw_forearm<-NULL
dv$skewness_roll_forearm<-NULL
dv$skewness_pitch_forearm<-NULL
dv$skewness_yaw_forearm<-NULL
dv$max_yaw_forearm<-NULL
dv$min_yaw_forearm<-NULL
dv$amplitude_yaw_forearm<-NULL
#missed variables
```

Majority of Project Code

```
dv$kurtosis_pitch_belt<-NULL
dv$skewness_roll_belt.1<-NULL
dv$skewness_yaw_belt<-NULL

#determine if non-variable/integer variables should stay
theory = dv
table(theory$classe[theory$user_name=="adelmo"])
table(theory$classe[theory$user_name=="carlitos"])
table(theory$classe[theory$user_name=="charles"])
table(theory$classe[theory$user_name=="eurico"])
table(theory$classe[theory$user_name=="jeremy"])
table(theory$classe[theory$user_name=="pedro"])
theory$raw_timestamp_part_1 <-NULL
theory$user_name<-NULL
theory$raw_timestamp_part_2<-NULL
theory$cvtd_timestamp<-NULL
theory$new_window<-NULL
theory$num_window<-NULL
theory$x<-NULL

dv = theory
# Data Partition
set.seed(425)
ind_dv <- sample(2, nrow(dv), replace = TRUE, prob = c(0.7, 0.3))
train_dv <- dv[ind_dv==1,]
test_dv <- dv[ind_dv==2,]
train_dv <- as.data.frame(train_dv)
train_dv$classe <- as.factor(train_dv$classe)
#train_2 <- train_dv.combine[1:13816, c("pitch_forearm", "magnet_dumbbell_z",
    "roll_forearm", "magnet_dumbbell_y", "roll_arm magnet_arm_x")]

#Variable Selection
var.select(classe~, train_dv)
#dv_var1 <- data.frame(dv$pitch_forearm,dv$magnet_dumbbell_z,dv$roll_forearm,
#                      dv$magnet_dumbbell_y,dv$roll_arm,dv$magnet_arm_x,
#                      dv$classe)
dv_var1 <- train_dv[c("pitch_forearm","magnet_dumbbell_z","roll_forearm",
    "magnet_dumbbell_y","roll_arm","magnet_arm_x","total_accel_dumbbell",
    "classe")]
dv_var2 <- data.frame(dv$gyros_belt_z,dv$gyros_forearm_x,dv$gyros_belt_z,dv$classe)
dv_var1$classe <- as.factor(dv_var1$classe)
# Random Forest
library(randomForest)
set.seed(573)
rf <- randomForest(classe~, data = train_dv)
print(rf)
dim(rf)
```

Majority of Project Code

```
#Testing Variables
set.seed(835)
rf_var1 <- randomForest(classe~., data=dv_var1)
rf_var2 <- randomForest(dv.classe~., data=dv_var2)

# Prediction & Confusion Matrix - train data
library(caret)
p1 <- predict(rf, train_dv)
confusionMatrix(p1, train_dv$classe)
#Variable Selection
p_var1 <- predict(rf_var1, dv_var1)
p_var2 <- predict(rf_var2, dv_var2)
confusionMatrix(p_var1, dv_var1$classe)
confusionMatrix(p_var2, dv_var2$dv.classe)#Var1 ended up way better
# Prediction & Confusion Matrix - test data
p2 <- predict(rf, test_dv)
confusionMatrix(p2, test_dv$classe)
p2_var1 <- predict(rf_var1, test_dv)
confusionMatrix(p2_var1, test_dv$classe )
# Error rate of Random Forest
plot(rf)

# Cross-Validation
set.seed(6872)
cv.10.folds <- createMultiFolds(dv$classe, k=10, times = 10)

# Set up caret's trainControl object
ctrl.1 <- trainControl(method = "repeatedcv", number = 10, repeats = 10, index =
cv.10.folds)

#Start-up doSnow
library(doSNOW)
cl <- makeCluster(8, type = "SOCK")
registerDoSNOW(cl)
#shutdown doSnow after train
stopCluster(cl)

# Set seed for reproducibility and train
set.seed(4762)
rf.cv.1 <- train(x = dv, y=dv$classe, method="rf", tuneLenght = 3,
               ntree = 1000, trControl = ctrl.1)
```

Majority of Project Code

```
#Build Model
model <- rpart(classe ~ pitch_forearm+magnet_dumbbell_z+roll_forearm+
               magnet_dumbbell_y+roll_arm+magnet_arm_x+
               total_accel_dumbbell, method = "class", data = dv_var1)

#Test for Out of Sample Error
set.seed(7358)
tr_control <- trainControl(method = "boot", number = 100)
boot_strap <- train(classe~., data = dv_var1, trControl=tr_control, method='nb')
print(boot_strap)

# Test the test data
ptest <- predict(rf,test)
confusionMatrix(ptest, test, type="response")

prediction <- predict(model,test)

var1_test <- predict(rf_va)
```