**Overview:**

In this project, I analyzed how well the people do a particular activity based on the dataset from <http://groupware.les.inf.puc-rio.br/har> . The training data is from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. There are 19622 samples with 160 features\measurements in the dataset. These samples belong to 5 diferent classes. Machine learning models are built to predict the quality of an activity which is represented in 5 classes(A,B,C,D,E). Then I use these models to predict the classes of 20 out-of-sample observations.

**Method:**

After a brief exploration on the dataset, some columns that are entirely empty or NA are eliminated from the training set. The domain knowledge that we obtained from the authors of the study allowed us to dismiss features like *timestamp* and *username*, as this would help our models to concentrate on more relevant features.

A total of 4 algorithms from two different family of machine learning approaches were used. Basic decision tree (rpart), boosted tee (gbm) and random forest (rf) are used because of ease of interpretation of the models and high level of accuracy in the face of complex data, respectively. Linear discriminant analysis(lda) is used because it is known to work well in general when the measurements made on independent variables for each observation are continuous quantities. Our data is mostly comprised of continuous variables.

Data is partitioned into %80 trainingset and %20 test set. Then the training set is k-folded for cross validation accross 4 algorithms we use. After training, each model's performance is evaluated against the test set.

**Result:**

The random forest model showed the best predictive power even though it was computationaly more costly to train. The table below sumarizes these results:

lda rpart gbm rf

accuracy 0.6962 0.4881 0.9651 0.9962

time to train(sec) 6 17 597 1725

The rf model made these predictions for 20 samples in pml-testing.csv:

> predict(model\_rf, testData)

[1] B A B A A E D B A A B C B A E E A B B B

Levels: A B C D E

**R Code Reference:**

#

# This R program learns from human body activity data and makes predictions as described in

# <https://www.coursera.org/learn/practical-machine-learning/supplement/PvInj/course-project-instructions-read-first>

#

library(caret); library(rpart); library(gbm) ; library(randomForest)

# Read, explore and clean data

training <- read.csv("C:\\Users\\Emin\\Documents\\data\_scientist\_toolkit\_R\\course8\_PracticalML\\pml-training.csv", na.strings = c("NA", ""))

testing <- read.csv("C:\\Users\\Emin\\Documents\\data\_scientist\_toolkit\_R\\course8\_PracticalML\\pml-testing.csv", na.strings = c("NA", ""))

dim(testing)

dim(training)

summary(training)

head(colnames(training),10)

not\_predictor <- c("X","user\_name","raw\_timestamp\_part\_1","raw\_timestamp\_part\_2","cvtd\_timestamp","new\_window","num\_window")

training[,not\_predictor] <- NA

testing[,not\_predictor] <- NA

training <- training[, colSums(is.na(training)) == 0]

testing <- testing[, colSums(is.na(testing)) == 0]

dim(training)

dim(testing)

# Partition data as training and validation sets

set.seed(71823)

partitionD <- createDataPartition(training$classe, p = 0.8, list = FALSE)

train <- training[partitionD, ]

valid <- training[-partitionD, ]

control <- trainControl(method = "cv", number = 5) # Use 5-fold cross validation

Sys.time()

#

# Basic Decision Tree

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

print(model\_rpart)

predict\_rpart <- predict(model\_rpart, valid)

confusionMatrix(valid$classe, predict\_rpart)

Sys.time()

#

# Linear Discriminant Analysis model

model\_lda <- train(classe ~ ., data = train, method = "lda", trControl = control)

print(model\_rlda)

print(model\_lda)

predict\_lda <- predict(model\_lda, valid)

confusionMatrix(valid$classe, predict\_lda)

Sys.time()

#

# Boosted Tree model

model\_gbm <- train(classe ~ ., data = train, method = "gbm", trControl = control)

print(model\_gbm)

predict\_gbm <- predict(model\_gbm, valid)

confusionMatrix(valid$classe, predict\_gbm)

predict(model\_gbm, testData)

Sys.time()

#

# Random Forest model

model\_rf <- train(classe ~ ., data = train, method = "rf", trControl = control)

print(model\_rf, digits = 4)

print(model\_rf)

predict\_rf <- predict(model\_rf, valid)

confusionMatrix(valid$classe, predict\_rf)

# save.image("C:\\Users\\Emin\\Documents\\PML1")

predict(model\_rf, testData)

Sys.time()