Executive Summary

This project is to demonstrate if the type of exercies could be determined from the readings of different sensors attached to four location (arm, forearm, dumbbell and belt). Cross validation is done on testing data set created by 70/30 split random subsampling on the training data set. 20 features have been selected by variable importance and used to build the final model using Random Forest classification. The out-of-bags accuracy of the final model is 99.1%. The project shows that it is possible to recognize five types of exercise if the sensor readings are presented.

Data Source and Citation

The training data for this project are available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

Original Data: Human Activity Recognition
http://groupware.les.inf.puc-rio.br/har

Citation:
Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6.

Implementation

Download Files

```
dataDir <- './data'
dir.create(dataDir, showWarnings=FALSE, recursive=TRUE)
fileUrl <- 'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
destFile <- file.path(dataDir, 'training.csv')
#download.file(fileUrl, destfile=destFile)
fileUrl <- 'https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'
destFile <- file.path(dataDir, 'testing.csv')
#download.file(fileUrl, destfile=destFile)
dateDownloaded<-date()
dateDownloaded
```

Load Data

```
dataDir <- './data'
srcFile <- file.path(dataDir, 'training.csv')
                                    training <- read.csv(srcFile, header=T)

srcFile <- file.path(dataDir, 'testing.csv')
testing <- read.csv(srcFile, header=T)
```

Features Selections

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
# Exclude all the summary variables such as max_roll_belt, max_picth_belt, max_yaw_belt, min_roll_belt
ds <- training[,c(1,2,3,4,5,6,7,8,9,10,11,37,38,39,40,41,42,43,44,45,46,47,48,49,60,61,62,63,64,65,66,67,68,84,85,86,113,114
,115,116,117,118,119,120,121,122,123,124,151,152,153,154,155,156,157,158,159,160)]

# Reorder the data set
ds_try <- ds[,c(58, 8:57)]
inTrain <- createDataPartition(y=ds_try$classe, p=0.5, list=FALSE)
training_set <- ds_try[inTrain,]

# Enable Parallel Processing
library("doSNOW")
```

```
## Loading required package: foreach
## foreach: simple, scalable parallel programming from Revolution Analytics
## Use Revolution R for scalability, fault tolerance and more.
## http://www.revolutionanalytics.com
```

```
## Loading required package: iterators
## Loading required package: snow
```

```
cl<-makeCluster(4) #change the 4 to your number of CPU cores
registerDoSNOW(cl)

# Model Training
modFit <- train((training_set[,-1]), training_set$classe ,method="rf",prox=TRUE, ntree=100)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
modFit
```

```
## Random Forest
##
## 9812 samples
##   50 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 9812, 9812, 9812, 9812, 9812, 9812, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa   Accuracy SD  Kappa SD
##    2    0.9814    0.9764  0.003145     0.003983
##   26    0.9840    0.9798  0.003062     0.003870
##   50    0.9767    0.9705  0.004197     0.005299
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 26.
```

```
# Getting the importance of the variables.
imp <- varImp(modFit)
imp
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 50)
##
##                   Overall
## roll_belt          100.00
## pitch_forearm       57.65
## magnet_dumbbell_z   49.70
## yaw_belt            49.35
## magnet_dumbbell_y   43.23
## pitch_belt          43.11
## roll_forearm        39.63
## accel_dumbbell_y    29.37
## roll_dumbbell       18.07
## accel_forearm_x     17.96
## magnet_dumbbell_x   17.26
## accel_dumbbell_z    15.34
## accel_belt_z        15.31
## magnet_belt_z       15.06
## magnet_belt_y       11.82
## magnet_forearm_z    11.63
## magnet_belt_x       10.30
```

```
## yaw_arm              9.89
## gyros_belt_z         9.38
## accel_forearm_z      9.05
```

```
# Clean up Memory
modFit<-NULL
```

Model Run

```
# Subset data set with top 20 most important variables and predicted variable (classe)
ds_try = with(ds,
              data.frame
                    (
                      classe,
                      roll_belt, pitch_forearm, yaw_belt, pitch_belt, magnet_dumbbell_z,
                      roll_forearm, magnet_dumbbell_y, accel_dumbbell_y, roll_dumbbell, accel_forearm_x,
                      magnet_dumbbell_x, magnet_belt_z, accel_dumbbell_z, accel_belt_z, magnet_belt_y,
                      gyros_belt_z, magnet_belt_x, roll_arm, yaw_arm, magnet_forearm_y
                    )
              )

inTrain <- createDataPartition(y=ds_try$classe, p=0.7, list=FALSE)
training_set <- ds_try[inTrain,]
testing_set <- ds_try[-inTrain,]

modFit <- train((training_set[,-1]), training_set$classe ,method="rf",prox=TRUE, ntree=500)
modFit
```

```
## Random Forest
##
## 13737 samples
##    20 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 13737, 13737, 13737, 13737, 13737, 13737, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa   Accuracy SD  Kappa SD
##    2    0.9868    0.9833  0.002220     0.002802
##   11    0.9843    0.9802  0.001849     0.002329
##   20    0.9766    0.9704  0.004477     0.005667
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

```
# Terminate Parallel Processing
stopCluster(cl)
```

Results

```
predictions <- predict(modFit,newdata=testing_set)

confusionMatrix(predictions,testing_set$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    6    0    0    0
```

```
##          B    2 1124    3    1    0
##          C    0    9 1022   12    2
##          D    0    0    1  951    5
##          E    0    0    0    0 1075
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.991, 0.995)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    0.987    0.996    0.987    0.994
## Specificity             0.999    0.999    0.995    0.999    1.000
## Pos Pred Value          0.996    0.995    0.978    0.994    1.000
## Neg Pred Value          1.000    0.997    0.999    0.997    0.999
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.191    0.174    0.162    0.183
## Detection Prevalence    0.285    0.192    0.178    0.163    0.183
## Balanced Accuracy       0.999    0.993    0.996    0.993    0.997
```

Conclusions

The Project successfully demonstrated that random forest classification can recognize five types of exercise if selected 20 sensor readings are presented. The selected sensor readings are roll_belt, pitch_forearm, yaw_belt, pitch_belt, magnet_dumbbell_z, roll_forearm, magnet_dumbbell_y, accel_dumbbell_y, roll_dumbbell, accel_forearm_x, magnet_dumbbell_x, magnet_belt_z, accel_dumbbell_z, accel_belt_z, magnet_belt_y, gyros_belt_z, magnet_belt_x, roll_arm, yaw_arm, magnet_forearm_y.