Human Activity Recognition

```
library(rpart)
library(caret)
```

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

```
pml.training <- read.csv("C:/coursera/machine_Learning/Project/pml-training.csv")
pml.testing <- read.csv("C:/coursera/machine_Learning/Project/pml-testing.csv")
```

Keep only the variables associated with accelerometers on the belt, forearm, arm, and dumbell.

```
training_pml <- pml.training[,c('accel_belt_x','accel_belt_y','accel_belt_z','accel_arm_x','accel_arm_y','accel_arm_z','accel_dumbbell_x','accel_dumbbell_y','accel_dumbbell_z','accel_forearm_x','accel_forearm_y','accel_forearm_z','classe')]
testing_pml <- pml.testing[,c('accel_belt_x','accel_belt_y','accel_belt_z','accel_arm_x','accel_arm_y','accel_arm_z','accel_dumbbell_x','accel_dumbbell_y','accel_dumbbell_z','accel_forearm_x','accel_forearm_y','accel_forearm_z','problem_id')]
```

In order to minimise number of predictors Test for highlly correlated variables .

```
corr_pml <- cor(training_pml[, -13])
highCorr <- findCorrelation(corr_pml, 0.90)
highCorr
```

```
## [1] 3
```

```
training_pml_uncrl <- training_pml[,-highCorr ]
testing_pml_uncrl <- testing_pml[, -highCorr]
```

Inspecting model based on rpart function

```
summary (training_pml)
```

```
##   accel_belt_x      accel_belt_y    accel_belt_z     accel_arm_x
##  Min.   :-120.00   Min.   :-69.0   Min.   :-275.0   Min.   :-404.0
##  1st Qu.: -21.00   1st Qu.:  3.0   1st Qu.:-162.0   1st Qu.:-242.0
##  Median : -15.00   Median : 35.0   Median :-152.0   Median : -44.0
##  Mean   :  -5.59   Mean   : 30.1   Mean   : -72.6   Mean   : -60.2
##  3rd Qu.:  -5.00   3rd Qu.: 61.0   3rd Qu.:  27.0   3rd Qu.:  84.0
##  Max.   :  85.00   Max.   :164.0   Max.   : 105.0   Max.   : 437.0
##   accel_arm_y      accel_arm_z     accel_dumbbell_x accel_dumbbell_y
##  Min.   :-318.0   Min.   :-636.0   Min.   :-419.0   Min.   :-189.0
##  1st Qu.: -54.0   1st Qu.:-143.0   1st Qu.: -50.0   1st Qu.:  -8.0
##  Median :  14.0   Median : -47.0   Median :  -8.0   Median :  41.5
##  Mean   :  32.6   Mean   : -71.2   Mean   : -28.6   Mean   :  52.6
##  3rd Qu.: 139.0   3rd Qu.:  23.0   3rd Qu.:  11.0   3rd Qu.: 111.0
##  Max.   : 308.0   Max.   : 292.0   Max.   : 235.0   Max.   : 315.0
##  accel_dumbbell_z accel_forearm_x  accel_forearm_y accel_forearm_z
##  Min.   :-334.0   Min.   :-498.0   Min.   :-632    Min.   :-446.0
##  1st Qu.:-142.0   1st Qu.:-178.0   1st Qu.:  57    1st Qu.:-182.0
##  Median :  -1.0   Median : -57.0   Median : 201    Median : -39.0
##  Mean   : -38.3   Mean   : -61.7   Mean   : 164    Mean   : -55.3
##  3rd Qu.:  38.0   3rd Qu.:  76.0   3rd Qu.: 312    3rd Qu.:  26.0
##  Max.   : 318.0   Max.   : 477.0   Max.   : 923    Max.   : 291.0
##  classe
##  A:5580
##  B:3797
##  C:3422
##  D:3216
##  E:3607
##
```

```
tc <- trainControl("cv",10)
rpart.grid <- expand.grid(.cp=0.2)
modfit_pm_rpart <- train(classe ~.,method = 'rpart', data = training_pml_uncrl,trControl=tc,tuneGrid=rpart.grid)

modfit_pm_rpart
```

```
## CART
##
```

```
## 19622 samples
##    11 predictors
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 17660, 17660, 17660, 17658, 17661, 17660, ...
##
## Resampling results
##
##   Accuracy  Kappa  Accuracy SD  Kappa SD
##   0.3       0      1e-04        0
##
## Tuning parameter 'cp' was held constant at a value of 0.2
##
```

The rpart model has very low accuracy. Hence it cannot be used for prediction.

Inspect rf model on small sample of training data 20%, due mainlly to memory limit.

```
inTrain_pml_2 <- createDataPartition(y=training_pml_uncrl$classe, p=0.2, list=FALSE)
training_pml_2  <- training_pml_uncrl[inTrain_pml_2,]

modFitRF_pml_2 <- train(classe~., data= training_pml_2, method ="rf", prox=TRUE)
```

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

```
modFitRF_pml_2
```

```
## Random Forest
##
## 3927 samples
##   11 predictors
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3927, 3927, 3927, 3927, 3927, 3927, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa  Accuracy SD  Kappa SD
##   2     0.8       0.8    0.01         0.01
##   6     0.8       0.8    0.009        0.01
##   10    0.8       0.7    0.01         0.01
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

Sample validation data for confusion matrix evaluation.

In order to avoid using same observation in training and Validation I only used the data that was not used for training

```
valid_pml  <- training_pml_uncrl[-inTrain_pml_2,]

#the data in the validation size is limited to 0.8*0.1 = 0.08 % of the original trining data set
invalid_pml_2 <- createDataPartition(y=valid_pml$classe, p=0.1, list=FALSE)
valid_pml_2 <- valid_pml[invalid_pml_2,]
```

Inspect the confusion Matrix.

```
valid_pml_2$Prediction <- predict(modFitRF_pml_2, newdata=valid_pml_2)
confusionMatrix(data=valid_pml_2$Prediction, valid_pml_2$classe)
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   A   B   C   D   E
##          A 412  23   8   9   1
##          B   6 240  10   4  10
##          C   9  17 246  17  13
##          D  14  11   4 220  11
##          E   6  13   6   8 254
##
## Overall Statistics
##
##                Accuracy : 0.873
##                  95% CI : (0.855, 0.889)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.839
##  Mcnemar's Test P-Value : 0.000532
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.922    0.789    0.898    0.853    0.879
## Specificity             0.964    0.976    0.957    0.970    0.974
## Pos Pred Value          0.909    0.889    0.815    0.846    0.885
## Neg Pred Value          0.969    0.951    0.978    0.971    0.973
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.262    0.153    0.156    0.140    0.162
## Detection Prevalence    0.288    0.172    0.192    0.165    0.183
## Balanced Accuracy       0.943    0.883    0.927    0.911    0.927
```

Aplly the predictor model to new data given in the test dataset

```
pred_pml <- predict(modFitRF_pml_2,testing_pml_uncrl )
pred_pml
```

```
##  [1] B C C A A E D B A A B C B A E B A B C B
## Levels: A B C D E
```

Attach predicted values to the test dataset

```
testing_pml_uncrl_2 <- testing_pml_uncrl
testing_pml_uncrl_2$pred_class <- pred_pml
#view newe predicted value and row identifier. Needed for the submission part of the course project assignement
testing_pml_uncrl_2[c(12,13)]
```

```
##    problem_id pred_class
## 1           1          B
## 2           2          C
## 3           3          C
## 4           4          A
## 5           5          A
## 6           6          E
## 7           7          D
## 8           8          B
## 9           9          A
## 10         10          A
## 11         11          B
## 12         12          C
## 13         13          B
## 14         14          A
## 15         15          E
## 16         16          B
## 17         17          A
## 18         18          B
## 19         19          C
## 20         20          B
```