**Week4 : Prediction Assignment Write up**

The goal of this work is to predict the activity done by humans based on the analysis done in <http://groupware.les.inf.puc-rio.br/har>. To build this model, we are using pml training and testing datasets, which is being provided in the above mentioned website.

PACKAGES USED FOR ANALYSIS

For the analysis , the following packages are loaded

> library(ggplot2)

> library(scales)

> library(dplyr)

> library(caret)

READ THE DATA

Initially we have to read the csv files pml-training and pml-testing and also have to identify the missing values ,NA , Divison error values

> pmltrain<-read.csv("E:/Rprogram/mlweek4/pml-training.csv",header=T,na.strings=c("NA","","#DIV/0!"))

> dim(pmltrain)

[1] 19622 160

> colnames(pmltrain)

[1] "X" "user\_name"

[3] "raw\_timestamp\_part\_1" "raw\_timestamp\_part\_2"

[5] "cvtd\_timestamp" "new\_window"

[7] "num\_window" "roll\_belt"

[9] "pitch\_belt" "yaw\_belt"

[11] "total\_accel\_belt" "kurtosis\_roll\_belt"

[13] "kurtosis\_picth\_belt" "kurtosis\_yaw\_belt"

[15] "skewness\_roll\_belt" "skewness\_roll\_belt.1"

[17] "skewness\_yaw\_belt" "max\_roll\_belt"

[19] "max\_picth\_belt" "max\_yaw\_belt"

[21] "min\_roll\_belt" "min\_pitch\_belt"

[23] "min\_yaw\_belt" "amplitude\_roll\_belt"

[25] "amplitude\_pitch\_belt" "amplitude\_yaw\_belt"

[27] "var\_total\_accel\_belt" "avg\_roll\_belt"

[29] "stddev\_roll\_belt" "var\_roll\_belt"

[31] "avg\_pitch\_belt" "stddev\_pitch\_belt"

[33] "var\_pitch\_belt" "avg\_yaw\_belt"

[35] "stddev\_yaw\_belt" "var\_yaw\_belt"

[37] "gyros\_belt\_x" "gyros\_belt\_y"

[39] "gyros\_belt\_z" "accel\_belt\_x"

[41] "accel\_belt\_y" "accel\_belt\_z"

[43] "magnet\_belt\_x" "magnet\_belt\_y"

[45] "magnet\_belt\_z" "roll\_arm"

[47] "pitch\_arm" "yaw\_arm"

[49] "total\_accel\_arm" "var\_accel\_arm"

[51] "avg\_roll\_arm" "stddev\_roll\_arm"

[53] "var\_roll\_arm" "avg\_pitch\_arm"

[55] "stddev\_pitch\_arm" "var\_pitch\_arm"

[57] "avg\_yaw\_arm" "stddev\_yaw\_arm"

[59] "var\_yaw\_arm" "gyros\_arm\_x"

[61] "gyros\_arm\_y" "gyros\_arm\_z"

[63] "accel\_arm\_x" "accel\_arm\_y"

[65] "accel\_arm\_z" "magnet\_arm\_x"

[67] "magnet\_arm\_y" "magnet\_arm\_z"

[69] "kurtosis\_roll\_arm" "kurtosis\_picth\_arm"

[71] "kurtosis\_yaw\_arm" "skewness\_roll\_arm"

[73] "skewness\_pitch\_arm" "skewness\_yaw\_arm"

[75] "max\_roll\_arm" "max\_picth\_arm"

[77] "max\_yaw\_arm" "min\_roll\_arm"

[79] "min\_pitch\_arm" "min\_yaw\_arm"

[81] "amplitude\_roll\_arm" "amplitude\_pitch\_arm"

[83] "amplitude\_yaw\_arm" "roll\_dumbbell"

[85] "pitch\_dumbbell" "yaw\_dumbbell"

[87] "kurtosis\_roll\_dumbbell" "kurtosis\_picth\_dumbbell"

[89] "kurtosis\_yaw\_dumbbell" "skewness\_roll\_dumbbell"

[91] "skewness\_pitch\_dumbbell" "skewness\_yaw\_dumbbell"

[93] "max\_roll\_dumbbell" "max\_picth\_dumbbell"

[95] "max\_yaw\_dumbbell" "min\_roll\_dumbbell"

[97] "min\_pitch\_dumbbell" "min\_yaw\_dumbbell"

[99] "amplitude\_roll\_dumbbell" "amplitude\_pitch\_dumbbell"

[101] "amplitude\_yaw\_dumbbell" "total\_accel\_dumbbell"

[103] "var\_accel\_dumbbell" "avg\_roll\_dumbbell"

[105] "stddev\_roll\_dumbbell" "var\_roll\_dumbbell"

[107] "avg\_pitch\_dumbbell" "stddev\_pitch\_dumbbell"

[109] "var\_pitch\_dumbbell" "avg\_yaw\_dumbbell"

[111] "stddev\_yaw\_dumbbell" "var\_yaw\_dumbbell"

[113] "gyros\_dumbbell\_x" "gyros\_dumbbell\_y"

[115] "gyros\_dumbbell\_z" "accel\_dumbbell\_x"

[117] "accel\_dumbbell\_y" "accel\_dumbbell\_z"

[119] "magnet\_dumbbell\_x" "magnet\_dumbbell\_y"

[121] "magnet\_dumbbell\_z" "roll\_forearm"

[123] "pitch\_forearm" "yaw\_forearm"

[125] "kurtosis\_roll\_forearm" "kurtosis\_picth\_forearm"

[127] "kurtosis\_yaw\_forearm" "skewness\_roll\_forearm"

[129] "skewness\_pitch\_forearm" "skewness\_yaw\_forearm"

[131] "max\_roll\_forearm" "max\_picth\_forearm"

[133] "max\_yaw\_forearm" "min\_roll\_forearm"

[135] "min\_pitch\_forearm" "min\_yaw\_forearm"

[137] "amplitude\_roll\_forearm" "amplitude\_pitch\_forearm"

[139] "amplitude\_yaw\_forearm" "total\_accel\_forearm"

[141] "var\_accel\_forearm" "avg\_roll\_forearm"

[143] "stddev\_roll\_forearm" "var\_roll\_forearm"

[145] "avg\_pitch\_forearm" "stddev\_pitch\_forearm"

[147] "var\_pitch\_forearm" "avg\_yaw\_forearm"

[149] "stddev\_yaw\_forearm" "var\_yaw\_forearm"

[151] "gyros\_forearm\_x" "gyros\_forearm\_y"

[153] "gyros\_forearm\_z" "accel\_forearm\_x"

[155] "accel\_forearm\_y" "accel\_forearm\_z"

[157] "magnet\_forearm\_x" "magnet\_forearm\_y"

[159] "magnet\_forearm\_z" "classe"

>pmltest<-read.csv("E:/Rprogram/mlweek4/pml-testing.csv",na.strings=c("NA","","#DIV/0!"))

SPLITTING THE TRAINING DATA

To build the model and to create out of sample error data partition is done on pml-training data in a ration 70%:30%

> set.seed(32234)

> inTrain<-createDataPartition(y=pmltrain$classe,list=F,p=0.7)

> trainData<-pmltrain[inTrain,]

> testData<-pmltrain[-inTrain,]

> table(is.na(trainData))

Find which variables (if any) that are mostly na values

> naprops <- colSums(is.na(trainData))/nrow(trainData)

> mostlyNAs <- names(naprops[naprops > 0.75]) # mostly being 75%

> mostlyNACols <- which(naprops > 0.75) # there's about 100 of them

Taking a small sample data for training where all na’s are removed

> set.seed(1256)

> smalltrain <- trainData %>% tbl\_df %>% sample\_n(size=1000)

> smalltrain <- smalltrain[,-mostlyNACols]

> smalltrain <- smalltrain[,-grep("X|user\_name",names(smalltrain))]

> smalltrain <- smalltrain[,-grep("cvtd\_timestamp",names(smalltrain))]

> smalltrain <- smalltrain[,-nearZeroVar(smalltrain)]

> modelVars <- names(smalltrain)

> modelVars1 <- modelVars[-grep("classe",modelVars)] # remove the classe var

Using random forest with the predictors in the model variable we are predicting the classe variable

> set.seed(57)

> cleanedTrainData <- trainData[,modelVars]

> modelFit <- randomForest(classe ~., data=cleanedTrainData, type="class")

The values of the predictive model is stored in predTrain and confusion matrix is formed from predicted data to obtain the sample error

> predTrain <- predict(modelFit,newdata=trainData)

> confusionMatrix(predTrain,trainData$classe)$table

Reference

Prediction A B C D E

A 3906 0 0 0 0

B 0 2658 0 0 0

C 0 0 2396 0 0

D 0 0 0 2252 0

E 0 0 0 0 2525

To get an out sample error from the 30% pml-training data which is given as testing data

> classe\_col <- grep("classe",names(testData))

> predTest <- predict(modelFit, newdata = testData[,-classe\_col], type="class")

>

> confusionMatrix(predTest,testData$classe)

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1672 3 0 0 0

B 2 1135 5 0 0

C 0 1 1021 1 0

D 0 0 0 963 1

E 0 0 0 0 1081

Overall Statistics

Accuracy : 0.9978

95% CI : (0.9962, 0.9988)

No Information Rate : 0.2845

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9972

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 0.9988 0.9965 0.9951 0.9990 0.9991

Specificity 0.9993 0.9985 0.9996 0.9998 1.0000

Pos Pred Value 0.9982 0.9939 0.9980 0.9990 1.0000

Neg Pred Value 0.9995 0.9992 0.9990 0.9998 0.9998

Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839

Detection Rate 0.2841 0.1929 0.1735 0.1636 0.1837

Detection Prevalence 0.2846 0.1941 0.1738 0.1638 0.1837

Balanced Accuracy 0.9990 0.9975 0.9974 0.9994 0.9995

**Predicting exercise activity using the model**

Perform the prediction

> predpmltest <- predict(modelFit, newdata = pmltest, type="class")

> print(predpmltest)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

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