Machine Learning Course Project Report

Definition

In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. One can use any of the other variables to predict with. This report describing how the model is built, analysis of cross validation, expected out of sample error is, and justification of the model adapted. Apply prediction model to predict 20 different test cases and present the results.

Data

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

test and train data is read from the csv files into testd and traind variables respectively.

```
testd= read.csv("pml-testing.csv", sep=",", header=T,na.strings=c("NA",""))
traind= read.csv("pml-training.csv", sep=",", header=T)
```

Input Data files

- pml-training.csv
- pml-testing.csv

These files are placed in the working dirctory where the R script file also placed.

Output Data files

The output of the Model with the 20 test cases is the Data Array of 20 elments with each element indicting the classe variable value of A, B,C,D,E.

Data Cleaning

Given Data has about 160 variables with 19622 observations in the training set. On observation of the variables, there seems to be quite a bit of varibles with no data or has NAs. A few data cleanin steps as below have resuled in a variable set of 52.

- removed variables which has NA for all the observations
- removed variables which has timestamp, user informaton
- removed the num_window variable

```
traind = traind[, colSums(traind)==0]
traind = traind[,-c(grep("timestamp|X|user_name|new_window",names(traind)))]
traind = traind[,-1]
```

Data Exploration

A basic correlation analysis is performaed to determine the key set of variables. Variables which has more than 0.80 cross correlations have been listed. There are only 38 variabes that meet this criteria.

[1] “yaw_belt” “total_accel_belt” “accel_belt_y” “accel_belt_z”
[5] “accel_belt_x” “magnet_belt_x” “roll_belt” “roll_belt”
[9] “accel_belt_y” “accel_belt_z” “pitch_belt” “magnet_belt_x”
[13] “roll_belt” “total_accel_belt” “accel_belt_z” “roll_belt”
[17] “total_accel_belt” “accel_belt_y” “pitch_belt” “accel_belt_x”
[21] “gyros_arm_y” “gyros_arm_x” “magnet_arm_x” “accel_arm_x”
[25] “magnet_arm_z” “magnet_arm_y” “accel_dumbbell_x” “accel_dumbbell_z” [29] “gyros_dumbbell_z” “gyros_forearm_z” “gyros_dumbbell_x” “gyros_forearm_z” [33] “pitch_dumbbell” “yaw_dumbbell” “gyros_forearm_z” “gyros_dumbbell_x” [37] “gyros_dumbbell_z” “gyros_forearm_y”

Comparison of 38 variables seem to be the subset of the 52 variables that resulted as part of the data cleansing. Hence it is determined to keep the 52 variables as predictors to build the predictor model as below.

Final varialbles list of 52 predictors that are to be part of the prediction model are outlined below.

- [1] “roll_belt” “pitch_belt” “yaw_belt”
- [4] “total_accel_belt” “gyros_belt_x” “gyros_belt_y”
- [7] “gyros_belt_z” “accel_belt_x” “accel_belt_y”
- [10] “accel_belt_z” “magnet_belt_x” “magnet_belt_y”
- [13] “magnet_belt_z” “roll_arm” “pitch_arm”
- [16] “yaw_arm” “total_accel_arm” “gyros_arm_x”
- [19] “gyros_arm_y” “gyros_arm_z” “accel_arm_x”
- [22] “accel_arm_y” “accel_arm_z” “magnet_arm_x”
- [25] “magnet_arm_y” “magnet_arm_z” “roll_dumbbell”
- [28] “pitch_dumbbell” “yaw_dumbbell” “total_accel_dumbbell”
- [31] “gyros_dumbbell_x” “gyros_dumbbell_y” “gyros_dumbbell_z”
- [34] “accel_dumbbell_x” “accel_dumbbell_y” “accel_dumbbell_z”
- [37] “magnet_dumbbell_x” “magnet_dumbbell_y” “magnet_dumbbell_z”
- [40] “roll_forearm” “pitch_forearm” “yaw_forearm”
- [43] “total_accel_forearm” “gyros_forearm_x” “gyros_forearm_y”
- [46] “gyros_forearm_z” “accel_forearm_x” “accel_forearm_y”
- [49] “accel_forearm_z” “magnet_forearm_x” “magnet_forearm_y”
- [52] “magnet_forearm_z”

Model 1- method= rpart (randomTrees)

```
trainIndex <- createDataPartition(y = traind$classe, p=0.7,list=FALSE)
trainData <- traind[trainIndex,]
modFit1 <- train(trainData$classe ~.,data = trainData,method="rpart")
modFit1
```

CART

13737 samples 52 predictors 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:

- cp Accuracy Kappa Accuracy SD Kappa SD
- 0.0252 0.556 0.427 0.02 0.0339
- 0.0464 0.462 0.286 0.0631 0.106
- 0.118 0.325 0.0605 0.0425 0.0643

Accuracy was used to select the optimal model using the largest value. The final value used for the model was cp = 0.0252.

- Model 1 Accuracy is 0.556 and hence is rejected

Model 2 method = rf (RandomForest)

```
trainIndex <- createDataPartition(y = testd$classe, p=0.7,list=FALSE)
trainData <- traind[trainIndex,]
trainData_test <- traind[-trainIndex,]
tcrtl <- trainControl(method = "repeatedcv", number = 3, repeats = 3, verboseIter=T, return
Resamp='all')
modFit2 <- train(trainData$classe ~.,data = trainData,method="rf",trControl=tcrtl )
```

Resulting Model as presented below.

Random Forest

13737 samples 52 predictors 5 classes: ‘A’, ‘B’, ‘C’, ‘D’, ‘E’

No pre-processing Resampling: Cross-Validated (3 fold, repeated 3 times)

Summary of sample sizes: 9157, 9158, 9159, 9158, 9158, 9158, …

- Resampling results across tuning parameters:

- mtry Accuracy Kappa Accuracy SD Kappa SD
- 2 0.989 0.986 0.00146 0.00184
- 27 0.989 0.986 0.00136 0.00172
- 52 0.985 0.98 0.00293 0.00371

Accuracy was used to select the optimal model using the largest value. The final value used for the model was mtry = 27.

- Model 2 is not applied to predict function against the test data set with in the train data set partion and then prediction results are tabulated against the actual outcome values with in the test data set of the traingdata.

```
predFit = predict(modFit2, data=trainData_test)
table(predFit, traindData_test$classe)
```

- PredFit A B C D E
- A 1671 11 0 0 0
- B 2 1128 3 0 2
- C 1 0 1019 9 4
- D 0 0 4 954 1
- E 0 0 0 1 1075

Accuracy is 0.985 and hence a good prediction model.

Assumptions and Comments

- Only sensor read/ observed data is assumed to be important observation. Other variables such as timestamp, Window numbers, user names are not included in the Model
- Addtioanl tests are not carried to determine the importance or the influence of the excluded variables have on outcome-classe
- There is likely that a further reduction of variables may result in a stronger prediciton strength.

Model Cross Validaion Testing

In the trainign control options, repeatedCV method is set with 3 repeats and 3 numbers for subsamples to obtain a faster response at the same time not sacrificing the accuracy of the model. Based on the observations, it is strongly felt that the choise of the Model Fit and train control options have resulted in a accurate prediction model.

Model Test Resuls

```
test_results = predFit(modFit2, testd)
test_results
```

- [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

Test results have predicted the classe value with 100% accurcy againt the test data set with 20 observations as presented above.

Error estimates

Confusion Matrix or the comparison table betweent the Model 2 prediction againt the actual classe outcome in the testing subset in training set is a good indication of the out of sample error rate

Error rate is = (1-(accuratePredictions/TotalSampleSize))*100

- Out of Sample Error estimate is about 0.646% for Model 2

Conclusion

Based on the above analysis and the end results on the test data, Model 2 is fairly accuare and can predict the classe outcome fro 52 predictor varisbles wit 99% accuracy. RandomForest method applied in the training with the crossfit validation has resulted in best possible prediction model. As we can observe ther method rpart is quite week in building the prodiciton model.