Machine Learning Project

DI CAI

**Data Cleaning**

#Data loading

pmlTrain<-read.csv("pml-training.csv", header=T, na.strings=c("NA", "#DIV/0!"))

pmlTest<-read.csv("pml-testing.csv", header=T, na.string=c("NA", "#DIV/0!"))

Training data was partitioned and preprocessed using the code described below. In brief, all variables with at least one “NA” were excluded from the analysis. Variables related to time and user information were excluded for a total of 51 variables and 19622 class measurements. Same variables were maintained in the test data set (Validation dataset) to be used for predicting the 20 test cases provided.

## NA exclusion for all available variables

noNApmlTrain<-pmlTrain[, apply(pmlTrain, 2, function(x) !any(is.na(x)))]

dim(noNApmlTrain)

[1] 19622 60

## variables with user information, time and undefined

cleanpmlTrain<-noNApmlTrain[,-c(1:8)]

dim(cleanpmlTrain)

[1] 19622 52

## 20 test cases provided clean info - Validation data set

cleanpmltest<-pmlTest[,names(cleanpmlTrain[,-52])]

dim(cleanpmltest)

[1] 20 51

**Data Partitioning and Prediction Process**

The cleaned downloaded data set was subset in order to generate a test set independent from the 20 cases provided set. Partitioning was performed to obtain a 75% training set and a 25% test set.

#data cleaning

library(caret)

inTrain<-createDataPartition(y=cleanpmlTrain$classe, p=0.75,list=F)

training<-cleanpmlTrain[inTrain,]

test<-cleanpmlTrain[-inTrain,]

#Training and test set dimensions

dim(training)

[1] 14718 52

dim(test)

[1] 4904 52

**Results and Conclusions**

Random forest trees were generated for the training dataset using cross-validation. Then the generated algorithm was examnined under the partitioned training set to examine the accuracy and estimated error of prediction. By using 51 predictors for five classes using cross-validation at a 5-fold an accuracy of 99.2% with a 95% CI [0.989-0.994] was achieved accompanied by a Kappa value of 0.99.

library(caret)

set.seed(13333)

fitControl2<-trainControl(method="cv", number=5, allowParallel=T, verbose=T)

rffit<-train(classe~.,data=training, method="rf", trControl=fitControl2, verbose=F)

+ Fold1: mtry= 2

- Fold1: mtry= 2

+ Fold1: mtry=26

- Fold1: mtry=26

+ Fold1: mtry=51

- Fold1: mtry=51

+ Fold2: mtry= 2

- Fold2: mtry= 2

+ Fold2: mtry=26

- Fold2: mtry=26

+ Fold2: mtry=51

- Fold2: mtry=51

+ Fold3: mtry= 2

- Fold3: mtry= 2

+ Fold3: mtry=26

- Fold3: mtry=26

+ Fold3: mtry=51

- Fold3: mtry=51

+ Fold4: mtry= 2

- Fold4: mtry= 2

+ Fold4: mtry=26

- Fold4: mtry=26

+ Fold4: mtry=51

- Fold4: mtry=51

+ Fold5: mtry= 2

- Fold5: mtry= 2

+ Fold5: mtry=26

- Fold5: mtry=26

+ Fold5: mtry=51

- Fold5: mtry=51

Aggregating results

Selecting tuning parameters

Fitting mtry = 26 on full training set

predrf<-predict(rffit, newdata=test)

confusionMatrix(predrf, test$classe)

Confusion Matrix and Statistics

Reference

Prediction A B C D E

A 1395 1 0 0 0

B 0 943 5 0 0

C 0 3 848 12 0

D 0 0 2 792 3

E 0 2 0 0 898

Overall Statistics

Accuracy : 0.9943

95% CI : (0.9918, 0.9962)

No Information Rate : 0.2845

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

Kappa : 0.9928

Mcnemar's Test P-Value : NA

Statistics by Class:

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

Sensitivity 1.0000 0.9937 0.9918 0.9851 0.9967

Specificity 0.9997 0.9987 0.9963 0.9988 0.9995

Pos Pred Value 0.9993 0.9947 0.9826 0.9937 0.9978

Neg Pred Value 1.0000 0.9985 0.9983 0.9971 0.9993

Prevalence 0.2845 0.1935 0.1743 0.1639 0.1837

Detection Rate 0.2845 0.1923 0.1729 0.1615 0.1831

Detection Prevalence 0.2847 0.1933 0.1760 0.1625 0.1835

Balanced Accuracy 0.9999 0.9962 0.9941 0.9919 0.9981

pred20<-predict(rffit, newdata=cleanpmltest)

# Output for the prediction of the 20 cases provided

pred20

[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