MachineLearningProject

RM

Saturday, August 22, 2015

How the model is built Exploring he data, it is noticed that out of the 159 columns, only 53 are useful variables/columns are going to be useful. Given that random forest algorithm tends to perform better when so many potential variables are oinvolved, that model will be tried first to see if the accuracy will be satisfactory or we need to exolore other models.Removing the columns not adding value to the model, the model is built with training data provided after splitting the whole data into two randomly selected sets. 65% is assigned to train the model and 35% to test the model. It is noticed from the results presented below that the accuracy is more than 99% which is pretty high and no further exploration of alternate models is performed.

How you used cross validation Cross validation is performed to ensure there is no over fitting and to see if any further improvement in accuracy is preferred. The K-fold method, with k as 5 is used on the training set. The result shows only a very minor improvement in accuracy.

What you think the expected out of sample error is The out of sample erro will be (1-accuracy) in the cross-validatoin data.

Why you made the choices you did The splitting of the data 65:35 is not too far from the norm 75:25 Random forest models tend to provide fairly good accuracy where large number of variables are part of the data. A high 99.2% accuracy confirmed this for this dataset. Cross validation only made a slight improvement in accuracy to 99.3%

```
# Load required packages
library(caret)
library(randomForest)
library(corrplot)
library(parallel)
library(doParallel)
library(png)
library(grid)
# Ingest the data replacing all missing data with NA
trainData <- read.csv("C:/Users/Ravi/zPersonal/Data Science+Berkeley/Coursera/Data Science Specialization/08 Practical Machine Learning/Data/pml-training.csv", , na.strings=c("NA","#DIV/0!", ""), header=TRUE)
testData <- read.csv("C:/Users/Ravi/zPersonal/Data Science+Berkeley/Coursera/Data Science Specialization/08 Practical Machine Learning/Data/pml-testing.csv", , na.strings=c("NA","#DIV/0!", ""), header=TRUE)

# Review the Data

# Clean the data
trainData <- trainData[,colSums(is.na(trainData))  == 0]
testData <- testData[,colSums(is.na(testData))  == 0]
#Remove the following columns(1-7) that do not contribute to the moel in any meaningful way
#[1] "X"                    "user_name"            "raw_timestamp_part_1" "raw_timestamp_part_2"
#[5] "cvtd_timestamp"       "new_window"           "num_window"
trainData <- trainData[-c(1:7)]
dim(trainData )
#[1] 19622    53
```

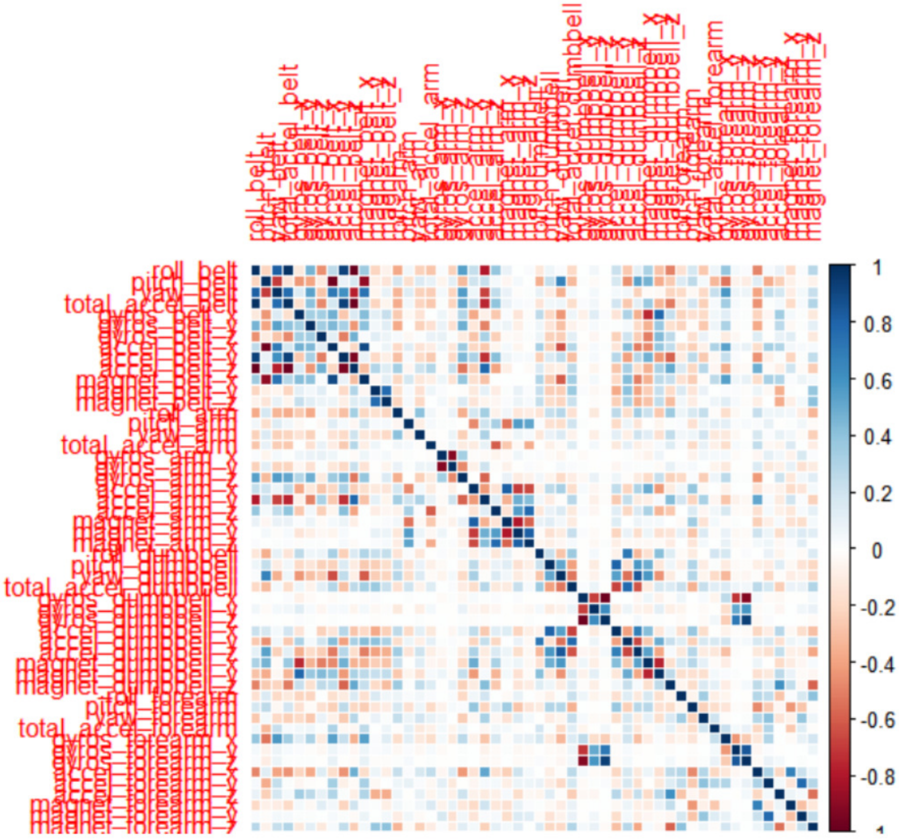
```
testData <- testData[-c(1:7)]
dim(testData )
#[1] 20 53

# Given that we have plenty of data(19622 records), Partition the training data
# into sub-training and sub-testing to do some cross validation
# We'll use a rough 2:1 ratio
trainIndex <- createDataPartition(y=trainData$classe, p=0.65, list=FALSE)
training <- trainData[trainIndex,]
testing <- trainData[-trainIndex,]

# Select a machine learning algorithm
# Lets try random forest method and see if the accuracy is satisfactory
modelrf <-randomForest(classe ~. , data=training, method="class")

# Train the model
# Predict - test the model
predictionrf <- predict(modelrf, testing, type = "class")
modelrf

Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1950    6    0    0    0
         B    0 1318   15    0    0
         C    2    4 1182   19    0
         D    0    0    0 1105    4
         E    1    0    0    1 1258

Overall Statistics

               Accuracy : 0.9924
                 95% CI : (0.9901, 0.9943)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9904
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9985   0.9925   0.9875   0.9822   0.9968
Specificity            0.9988   0.9973   0.9956   0.9993   0.9996
Pos Pred Value         0.9969   0.9887   0.9793   0.9964   0.9984
Neg Pred Value         0.9994   0.9982   0.9973   0.9965   0.9993
Prevalence             0.2845   0.1934   0.1744   0.1639   0.1838
Detection Rate         0.2840   0.1920   0.1722   0.1610   0.1832
Detection Prevalence   0.2849   0.1942   0.1758   0.1615   0.1835
Balanced Accuracy      0.9986   0.9949   0.9915   0.9908   0.9982
```

```
#plot the correlation
require(corrplot)
corrData <- cor(trainData[, -length(names(trainData))])
corrplot(corrData, method="color")
```

```
library(png)
```

```
## Warning: package 'png' was built under R version 3.1.3
```

```
library(grid)
img <- readPNG("C:/CorrImg.png")
 grid.raster(img)
```



```
#Cross validation
# to ensure repeatability - random seed
set.seed(4321)
#parallel computing for multi-core
registerDoParallel(makeCluster(detectCores()))
controlF <- trainControl(method = "repeatedcv", number = 10, repeats = 10)
modelrf_CV <- train(classe ~ ., method="rf",  data=training, trControl = controlF)


#check accuracy after cross validation
print("check accuracy after cross validation")
```

```
new_accuracy<<- predict(modelrf_CV, testing)
print(confusionMatrix(new_accuracy, testing$classe))
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1950   12    0    0    0
         B    3 1314    2    0    0
         C    0    2 1190   16    0
         D    0    0    5 1107    6
         E    0    0    0    2 1256

Overall Statistics
                                          
               Accuracy : 0.993           
                 95% CI : (0.9907, 0.9948)
    No Information Rate : 0.2845          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.9912          
 Mcnemar's Test P-Value : NA              

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9985   0.9895   0.9942   0.9840   0.9952
Specificity            0.9976   0.9991   0.9968   0.9981   0.9996
Pos Pred Value         0.9939   0.9962   0.9851   0.9902   0.9984
Neg Pred Value         0.9994   0.9975   0.9988   0.9969   0.9989
Prevalence             0.2845   0.1934   0.1744   0.1639   0.1838
Detection Rate         0.2840   0.1914   0.1733   0.1613   0.1830
Detection Prevalence   0.2858   0.1921   0.1760   0.1629   0.1832
Balanced Accuracy      0.9980   0.9943   0.9955   0.9910   0.9974
```

We notice that the accuracy after cross validation has not much improved. This model will be used to submit the responses to the rest of the project tasks.