Practical Machine Learning: Peer-graded Assignment

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. 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. More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Goal

The goal of this project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. Use the prediction model to predict 20 different test cases.

Data Loading and preprocessing

Loading required libraries

###Setting seed for reproducibility

```
set.seed(12345)
```

###Loading the data, removing Nans and keeping only required columns

```
if (!file.exists("pml-training.csv" )){
        fileUrl = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
        download.file(fileUrl, destfile="./pml-training.csv", method = "curl")
}

if (!file.exists("pml-testing.csv" )){
        fileUrl = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
        download.file(fileUrl, destfile="./pml-testing.csv", method = "curl")
}

#Read in the data:
read_train <- read.csv("pml-training.csv", header = TRUE, sep = ",", na.strings = c("NA", ""))
read_test <- read.csv("pml-testing.csv", header = TRUE, sep = ",", na.strings = c("NA", ""))


train_na <- sapply(read_train, function(x) {sum(is.na(x))})
read_train <- read_train[,which(train_na == 0)]

test_na <- sapply(read_test, function(x) {sum(is.na(x))})
read_test <- read_test[, which(test_na == 0)]

train_nzv <- nearZeroVar(read_train, saveMetrics = TRUE)
read_train <- read_train[,train_nzv$nzv == "FALSE"]
read_train$classe <- as.factor(read_train$classe)

test_nzv <- nearZeroVar(read_test, saveMetrics = TRUE)
read_test <- read_test[, test_nzv$nzv == "FALSE"]

train   <- read_train[,-c(1:6)]
test <- read_test[,-c(1:6)]

dim(train)
```

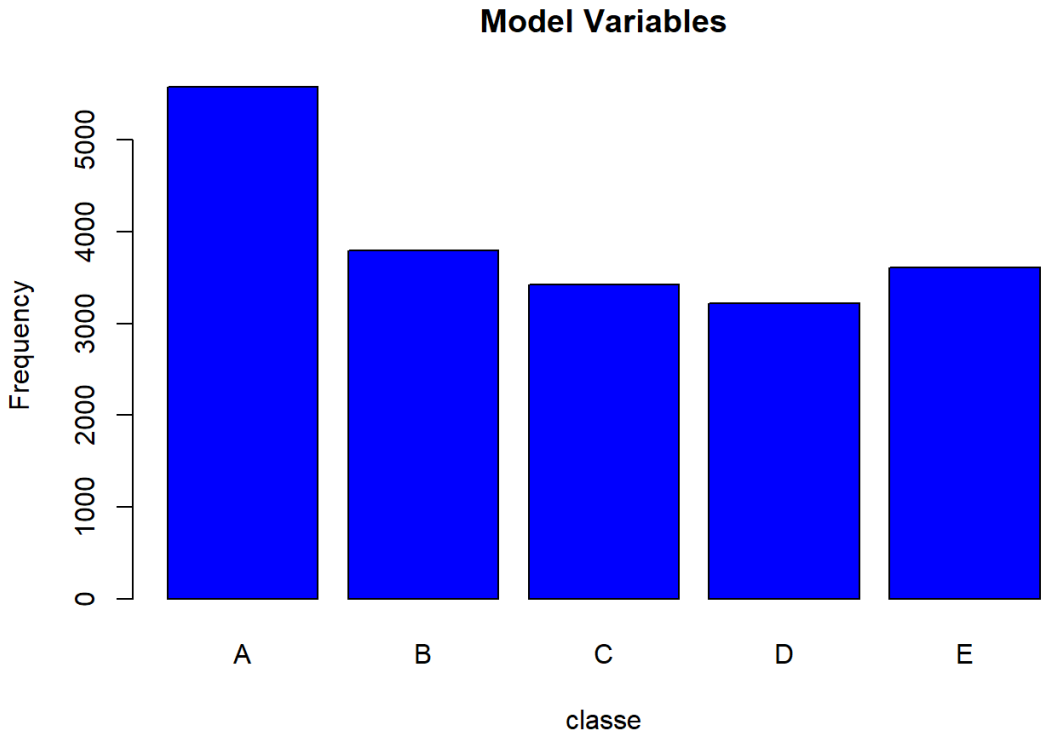
```
## [1] 19622    53
```

```
dim(test)
```

```
## [1] 20 53
```

Plotting the data

```
plot(train$classe, col="blue", main="Model Variables", xlab="classe", ylab="Frequency")
```



Cross-validation

Spliitting the train data set into 70% for training and 30% for testing. This will allow us to calculate out-of-sample errors

```
CVTrain <- createDataPartition(y=train$classe, p=0.7, list=FALSE)
cv_train <- train[CVTrain, ]
cv_test <- train[-CVTrain, ]

dim(cv_train)
```

```
## [1] 13737    53
```

```
dim(cv_test)
```

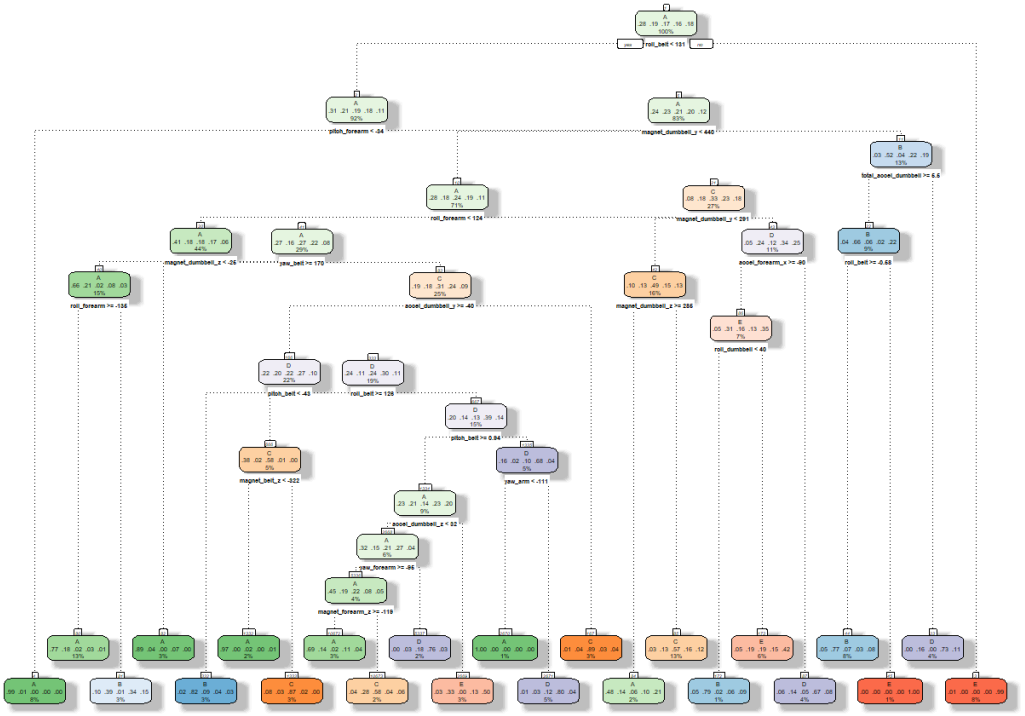
```
## [1] 5885   53
```

Model Fitting

I will fit 5 different models in order to choose the most accurate one for data prediction.

Model 1: Decision Trees

```
DT_fit <- rpart(classe ~ ., data=cv_train, method="class")
DT_predict <- predict(DT_fit, cv_test, type = "class")
DT_accuracy <- confusionMatrix(cv_test$classe, DT_predict)$overall[1]
DT_OoSE <- 1 - DT_accuracy
fancyRpartPlot(DT_fit)
```



Rattle 2020-Oct-14 15:36:32 Patty

The estimated accuracy for Decision Trees is 0.7391674 and Out-of-Sample Error is 0.2608326

Model 2: Random Forest

```
RF_fit <- randomForest(classe ~ ., data=cv_train, method="class")
RF_predict <- predict(RF_fit, cv_test, type = "class")
RF_accuracy <- confusionMatrix(cv_test$classe, RF_predict)$overall[1]
RF_OoSE <- 1 - RF_accuracy
```

The estimated accuracy for Random Forest is 0.9966015 and Out-of-Sample Error is 0.0033985

Model 3: Support Vector Machines

```
SVM_fit <- svm(classe ~ ., data=cv_train)
SVM_predict <- predict(SVM_fit, cv_test)
SVM_accuracy <- confusionMatrix(cv_test$classe, SVM_predict)$overall[1]
SVM_OoSE <- 1- SVM_accuracy
```

The estimated accuracy for Support Vector Machines is 0.9413764 and Out-of-Sample Error is 0.0586236

Model 4: Generalized Boosted Regression Models

The estimated accuracy for Generalized Boosted Regression Models is 0.957859 and Out-of-Sample Error is 0.042141

Model 5: Linear Discriminant Analysis

```
LDA_fit <- train(classe ~ ., data=cv_train, method = "lda")
LDA_predict <- predict(LDA_fit, cv_train)
LDA_accuracy <- confusionMatrix(cv_train$classe, LDA_predict)$overall[1]
LDA_OoSE <- 1-LDA_accuracy
```

The estimated accuracy for Linear Discriminant Analysis is 0.7059038 and Out-of-Sample Error is 0.2940962

Conclusion

The highest accuracy and lower Out-of-Sample errors are found using Random Forest (0.9966015, 0.0033985 Respectively). I will use this model for the project prediction portion.

Random Forest predictions

```
predict(RF_fit, test, type="class")
```

```
##  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
```