Practical Machine Learning Week 4 Project

Executive Summary

Recent availability of devices like *Jawbone Up*, *Nike FuelBand*, and *Fitbit* made possible to collect a large amounts of data about people's exercises. This project analyzes these large datasets to predict whether people exercise *correctly* or *incorrectly*. For this goal, we use data from accelerometers on the participants' belt, forearm, arm, and dumbell as described in [1].

As we will show in the following lines, we have loaded and cleaned the data, and predicted three common type of models, Random Forest, CART and Stochastic Gradient Boosting Trees. The most accurate is Random Forest, and that's why we used it as our final model. At the end we predicted outcomes of the validation dataset.

Data Preparation

Dataset are from the Weight Lifting Exercise Dataset by (see more in [1], [2], [3])

First we load the data and make the general consistency check (on top of displayed, we also used *summary*, *str* and *table* commands). We also found that "#DIV/0!" is also NA for us

```
library(tidyverse); library(caret); library(plyr); library(dplyr)
trainingdata <- read.csv('pml-training.csv', na.strings = c("","NA","#DIV/0!"))
val <- read.csv('pml-testing.csv', na.strings = c("","NA","#DIV/0!"))
```

We also need to prepare our data for the analysis. First, as per assignment, we should use only belt, arm, dumbbell and forearm sensors. We select them in the first part of the code. Second, we also need to delete near zero variables from the dataset. Lastly, we impute missing values using the *caret* package.

```
prepareData <- function(dts){dts <- dts[-1]}
trainingdata <- prepareData(trainingdata)
val <- prepareData(val)

logicalBADF <- grep("_belt|_arm|_dumbbell|_forearm", names(trainingdata))
trainingdata <- trainingdata[,c(logicalBADF,which(names(trainingdata)=="classe"))]

trainZeros <- nearZeroVar(trainingdata, saveMetrics=TRUE)
trainingdata <- trainingdata[,trainZeros$nzv == FALSE]

preProc <- preProcess(trainingdata, method = c("knnImpute"))
trainingdata2 <- predict(preProc,trainingdata)
```

At the end we create a training and testing partition using the *caret* package.

```
set.seed(123)
inTrain <- createDataPartition(y=trainingdata2$classe, p=0.7, list=FALSE)
train <- trainingdata2[inTrain,]
testing <- trainingdata2[-inTrain,]
dim(train);dim(testing)
```

```
## [1] 13737   118
```

```
## [1] 5885  118
```

As we can see, only 118 potential predictors remain in the dataset.

Analytics

We are going to use multiple methods using and then we'll see which one is the best.

Random Forest (rf)

We will use cross-validation with three parts what looks like a good choice.

```
randformod <- train(classe ~ ., data=train, method='rf', trControl=trainControl(method='cv', number = 3)
```

Classification and Regression Trees (rpart)

Here we run standard CART model with maximal depth of 5.

```
cartmod <- train(classe ~ ., data=train, method='rpart',control = rpart.control(maxdepth = 5))
```

Stochastic Gradient Boosting Trees (gbm)

```
gbmmod <- train(classe ~ .,data=train,method='gbm', trControl=trainControl(method='cv', number = 10), v
```

Testing on validation data

Now we test the model on validation part of the dataset (we created as a partition), and we show the Accuracy of different methods.

Model	Accuracy
Random Forests	0.9915038
CART	0.4451997
GBM	0.9597281

Because the highest accuracy is achieved for Random Forest, we chose it as the final model.

Predicting Validation Dataset

The following is the prediction for the validation dataset.

```
##    id predicted
## 1   1         B
## 2   2         A
## 3   3         B
## 4   4         A
## 5   5         A
## 6   6         E
## 7   7         D
## 8   8         B
## 9   9         A
## 10 10         A
## 11 11         B
## 12 12         C
```

```
## 13 13         B
## 14 14         A
## 15 15         E
## 16 16         E
## 17 17         A
## 18 18         B
## 19 19         B
## 20 20         B
```

Conclusion

We have loaded and cleaned the data, and predicted three common type of models. The most accurate seems to be the Random Forest model, and that's why we used it as our final model. At the end we predicted outcomes of the validation dataset.

References

[1] Data description: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

[2] Train data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

[3] Test data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv