Machine Learning Final Project

Synopsis

The goal of this project is to predict the exercise activity by studying data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants in an exercise study. The participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways. The variable “classe” was used in a blind study to determine which estimation approach (rpart, randomforest or gbm) would be the most accurate.

Data Preparation Steps

First, the train and final blind data set were loaded into the variables “train_data” and “blind_test”, assuming the downloaded files are in the working directory.

```
setwd("/Users/davidboberski/Desktop/Dad/R files/Machine Learning/")

train_data <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
blind_test <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))
```

Here the data is cleaned and prepared for analysis.

Next, the train data were split 60% into training and 40% into testing buckets using the variables “myTrain” and “myTest” respectively.

```
library(caret)
set.seed(1)
myTrain_index <- createDataPartition(y = train_data$classe, p = 0.6, list = FALSE)
myTrain <- train_data[myTrain_index, ]
myTest <- train_data[-myTrain_index, ]
```

Remove missing values.

```
# remove the index, user id, timestamp etc.
myTrain <- myTrain[,-(1:6)]
myTest <- myTest[,-(1:6)]
blind_test <- blind_test[,-(1:6)]

# remove the columns with all "NA"s
myTrain <- myTrain[,colSums(is.na(myTrain)) == 0]
myTest <- myTest[,colSums(is.na(myTest)) == 0]
blind_test <- blind_test[,colSums(is.na(blind_test)) == 0]
```

The five classes A-E seems to distribute reasonably even in the data set.

Model Analysis

The three models were tested with 5-fold cross validation, and to avoid over-fitting and relieve some computational burden the Principal Components Analysis (PCA) option is enabled for preprocessing.

```
library(randomForest)
library(rpart)
library(gbm)
library(plyr)

# 5-fold cross validation, PCA for data preprocessing, and parallel for speed up
set.seed(3)
tr_cntl = trainControl(method = "cv", number = 5, preProcOptions = "pca", allowParall
el = TRUE)

# rpart model
set.seed(5)
model_rpart <- train(classe ~ ., data = myTrain, method = "rpart", trControl = tr_cnt
l, na.action = na.omit)

# randomforest model
set.seed(7)
model_rf <- train(classe ~ ., data = myTrain, method = "rf", trControl = tr_cntl, na.
action = na.omit)

#gbm model
set.seed(9)
model_gbm <- train(classe ~ ., data = myTrain, method = "gbm", trControl = tr_cntl, n
a.action = na.omit)
```

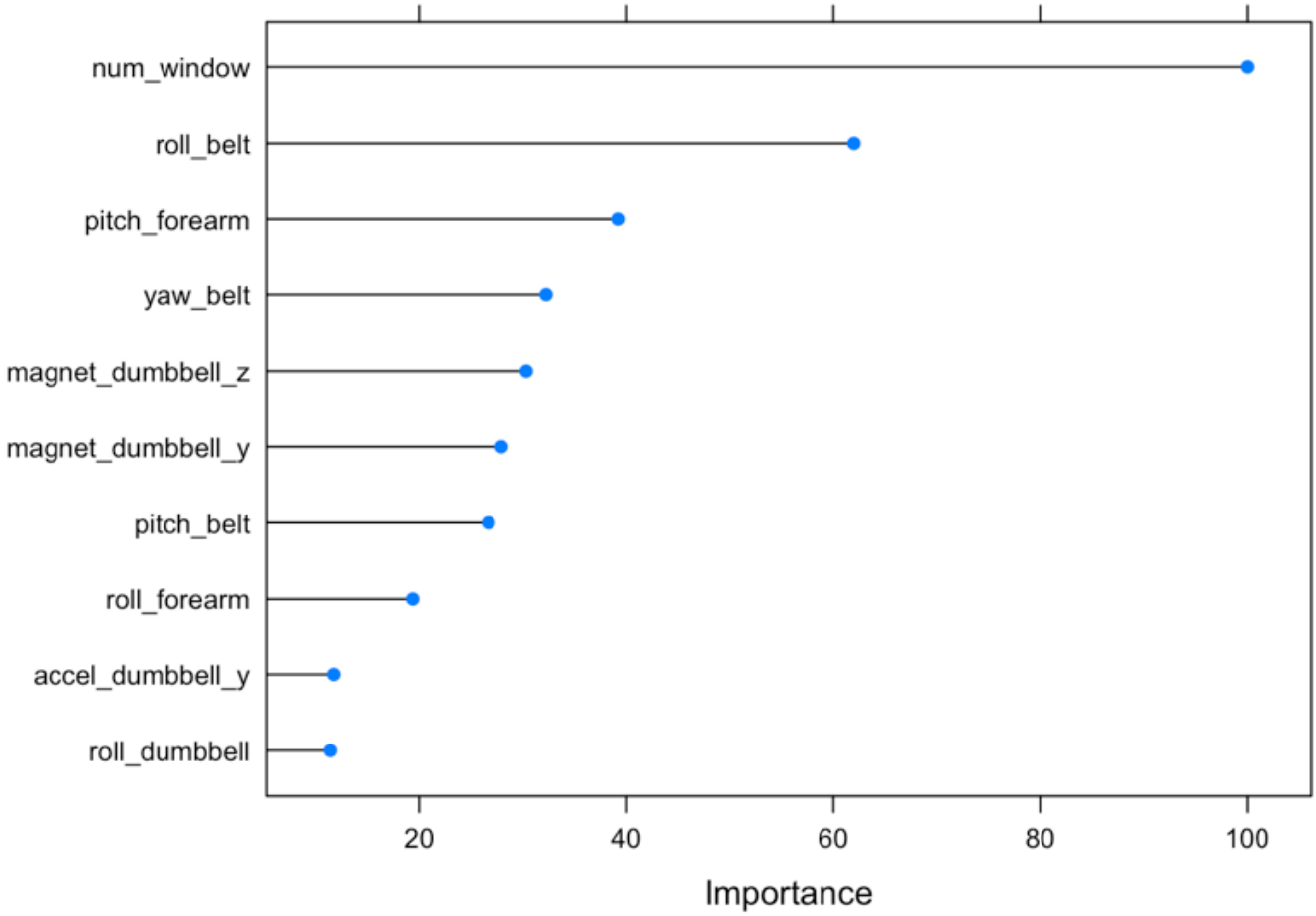
Then the 3 models were applied on the testing data set for comparing the corresponding accuracy.

```
# predict for the testing set
predict_rpart <- predict(model_rpart, myTest)
predict_rf <- predict(model_rf, myTest)
predict_gbm <- predict(model_gbm, myTest)

# accuracy for testing set
a1 <- confusionMatrix(as.factor(myTest$classe), predict_rpart)$overall[1]
a2 <- confusionMatrix(as.factor(myTest$classe), predict_rf)$overall[1]
a3 <- confusionMatrix(as.factor(myTest$classe), predict_gbm)$overall[1]

# predict for the training set
train_rpart <- predict(model_rpart, myTrain)
train_rf <- predict(model_rf, myTrain)
train_gbm <- predict(model_gbm, myTrain)

# accuracy for training set
b1 <- confusionMatrix(as.factor(myTrain$classe), train_rpart)$overall[1]
b2 <- confusionMatrix(as.factor(myTrain$classe), train_rf)$overall[1]
b3 <- confusionMatrix(as.factor(myTrain$classe), train_gbm)$overall[1]

# summerize the accuracy to a table
sum_table <- data.frame(Model = c("rpart", "randomForest", "gbm"))
sum_table$Testing_Accuracy <- c(a1, a2, a3)
sum_table$Training_Accuracy <- c(b1, b2, b3)

library(knitr)
kable(sum_table, format = "markdown", align = 'l')
```

Model	Testing_Accuracy	Training_Accuracy
rpart	0.5688249	0.5730299
randomForest	0.9975784	1.0000000
gbm	0.9861076	0.9938010

The training accuracy of ‘randomForest’ is highest, and therefore it is the model we should use. The top 10 variables from ‘randomForest’

```
plot(varImp(model_rf), top = 10)
```



Prediction

We ultimately apply the 'randomForest' model to the blind data set to predict the "classes" variable, and compare the predicted result to the actual result.

```
blind_rf_predict <- predict(model_rf, blind_test)

blind_rpart_predict <- predict(model_rpart, blind_test)
blind_gbm_predict <- predict(model_gbm, blind_test)

kable(cbind("randomForest" = blind_rf_predict, "gbm" = blind_gbm_predict,
            "rpart" = blind_rpart_predict),
            format = "markdown", align = 'l')
```

randomForest	gbm	rpart
2	2	1
1	1	1

2	2	3
1	1	1
1	1	1
5	5	3
4	4	3
2	2	3
1	1	1
1	1	1
2	2	3
3	3	3
2	2	3
1	1	1
5	5	3
5	5	3
1	1	1
2	2	1
2	2	1
2	2	3

Although ‘randomForest’ has better accuracy, it produced approximately as ‘gbm’ model.

Conclusion

The ‘randomForest’ model has the best accuracy prediction, around 0.998% on the testing data compared to all of the models evaluated. An alternative model ‘gbm’ produced approximately the same results on the blind test data.