Practical Machine Learning: Course Project

Rachel Smith

12/30/2020

1. Introduction

This document is the final report of the Peer Assessment project from Coursera's Practical Machine Learning as part of the data science specialization.

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, the goal was 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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

2. Load and clean data

Load

```
df_training <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv") %>% as_t

df_quiz <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv") %>% as_tibbl
```

Clean

```
dim(df_training)
```

```
## [1] 19622   160
```

```
# remove variables with missing values
df_training1 <- df_training %>% select(-which(colSums(is.na(.)) > 0))
dim(df_training1)
```

```
## [1] 19622    93
```

```r
# remove variables with near zero variance
df_training2 <- df_training1 %>% select(-c(df_training1 %>% nearZeroVar()))
dim(df_training2)
```

```
## [1] 19622    59
```

```r
# remove identification variables
df_training3 <- df_training2 %>% select(-c(1:5))
dim(df_training3)
```

```
## [1] 19622    54
```

3. Create data partition

```r
# set seed for reproducibility
set.seed(1234)

# create partition
inTrain <- createDataPartition(df_training3$classe, p = 0.7, list = FALSE)
df_train <- df_training3[c(inTrain),]
df_test <- df_training3[-c(inTrain),]
```

```r
# check dimensions
dim(df_train)
```

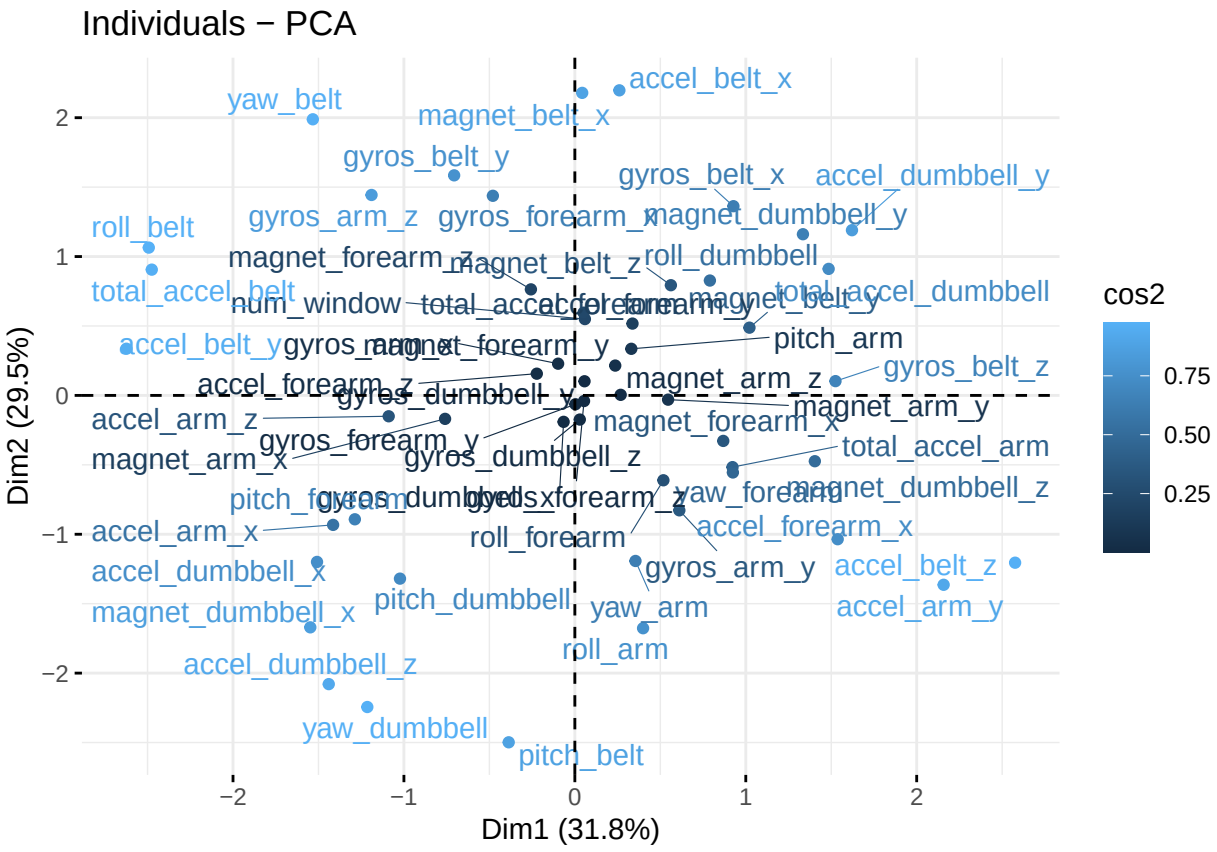
```
## [1] 13737    54
```

```r
dim(df_test)
```

```
## [1] 5885   54
```

```r
# visualize correlations among variables
pca_res <- princomp(cor(df_train %>% select(-classe)))

fviz_pca_ind(pca_res, col.ind = "cos2", repel = TRUE)
```



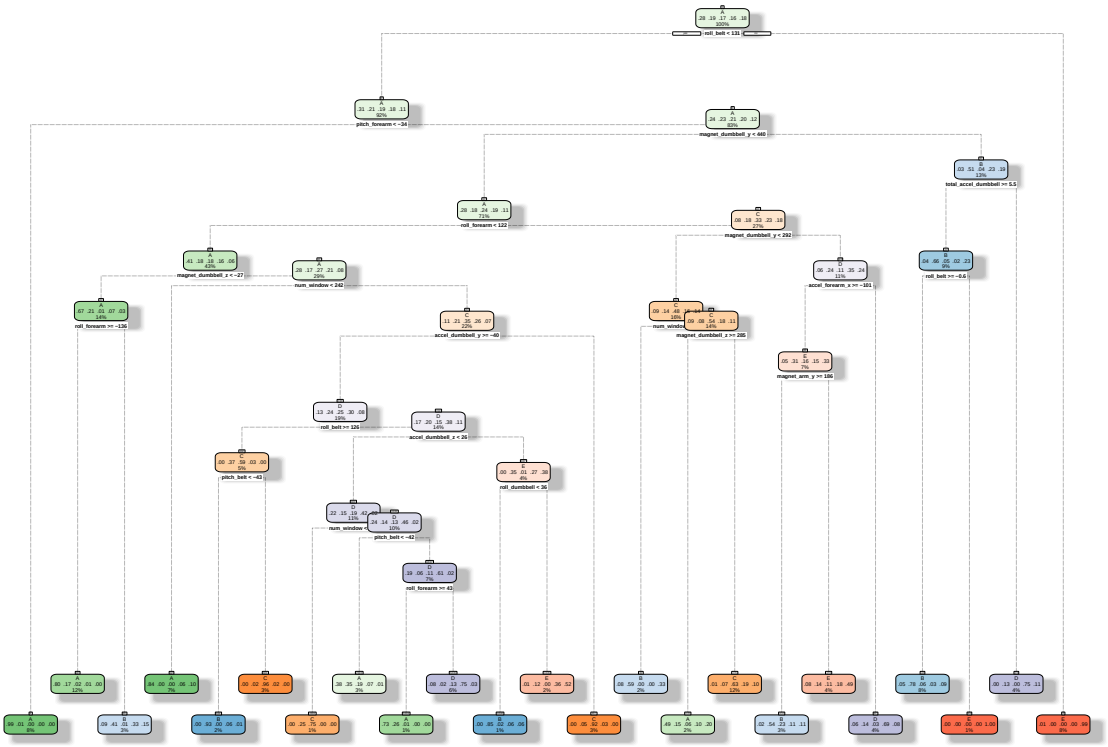
4. Fit models

A: Decision tree model

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

# fit model
model_dec_tree <- rpart(classe ~ ., data = df_train, method = "class")

# plot
fancyRpartPlot(model_dec_tree)
```



Rattle 2020−Dec−30 13:21:57 rachelsmith

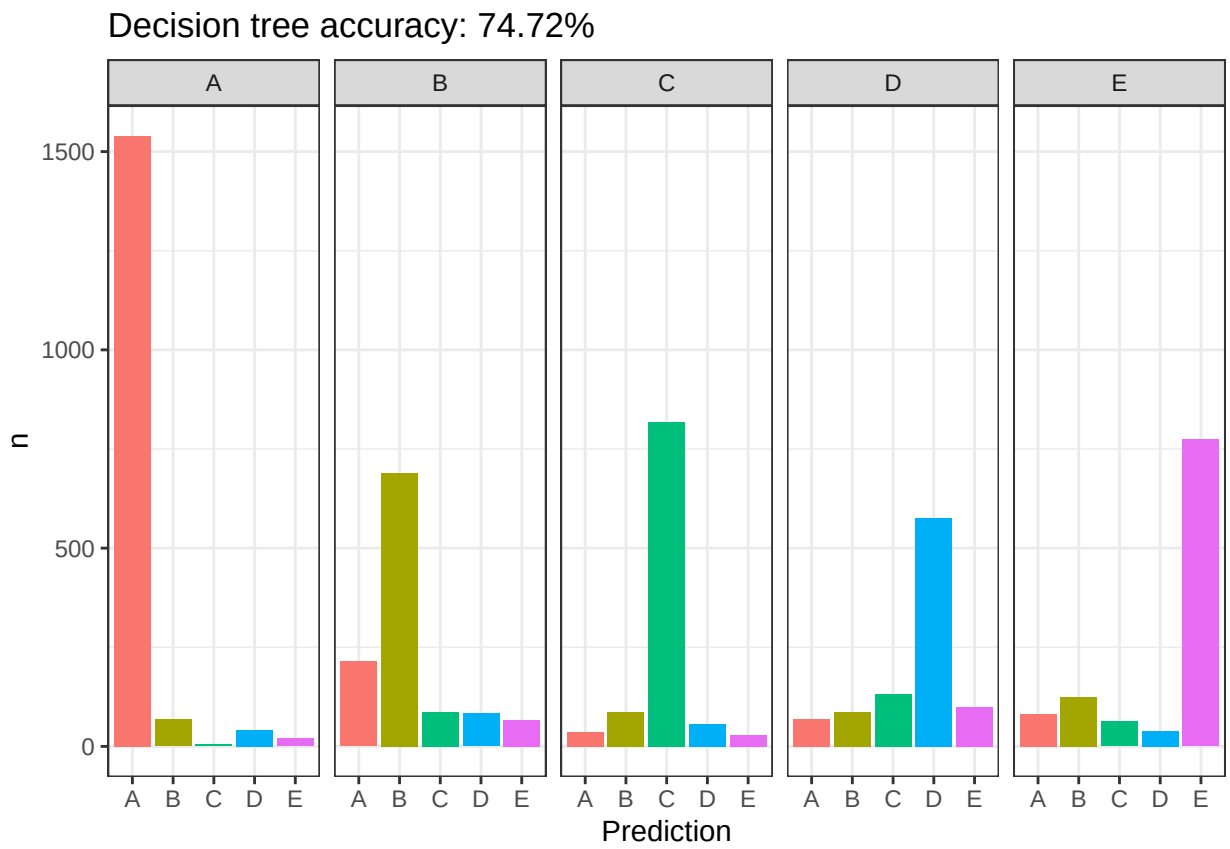
```
# validate on test data
predict_dec_tree <- predict(model_dec_tree, df_test, type = "class")

# confusion matrix
cm_dec_tree <- confusionMatrix(predict_dec_tree, df_test$classe)

# plot confusion matrix results
df_cm_dec_tree <- cm_dec_tree$table %>% as_tibble()

df_cm_dec_tree %>% ggplot(aes(x = Prediction, y = n, fill = Prediction)) +

       geom_bar(stat = "identity") +
       facet_grid(~Reference) +
       ggtitle(paste0("Decision tree accuracy: ", round(cm_dec_tree$overall["Accuracy"], 4)*100, "%")) +
       theme_bw() +
       theme(legend.position = "none")
```



Accuracy: 74.72%, out-of-sample error rate: 25.28%

B: Generalized boosted model

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

# training control
l_control_gbm <- trainControl(method = "repeatedcv", number = 5, repeats = 1)

# fit model
model_gbm <- train(classe ~ ., data = df_train, method = "gbm", trControl = l_control_gbm, verbose = FA
```

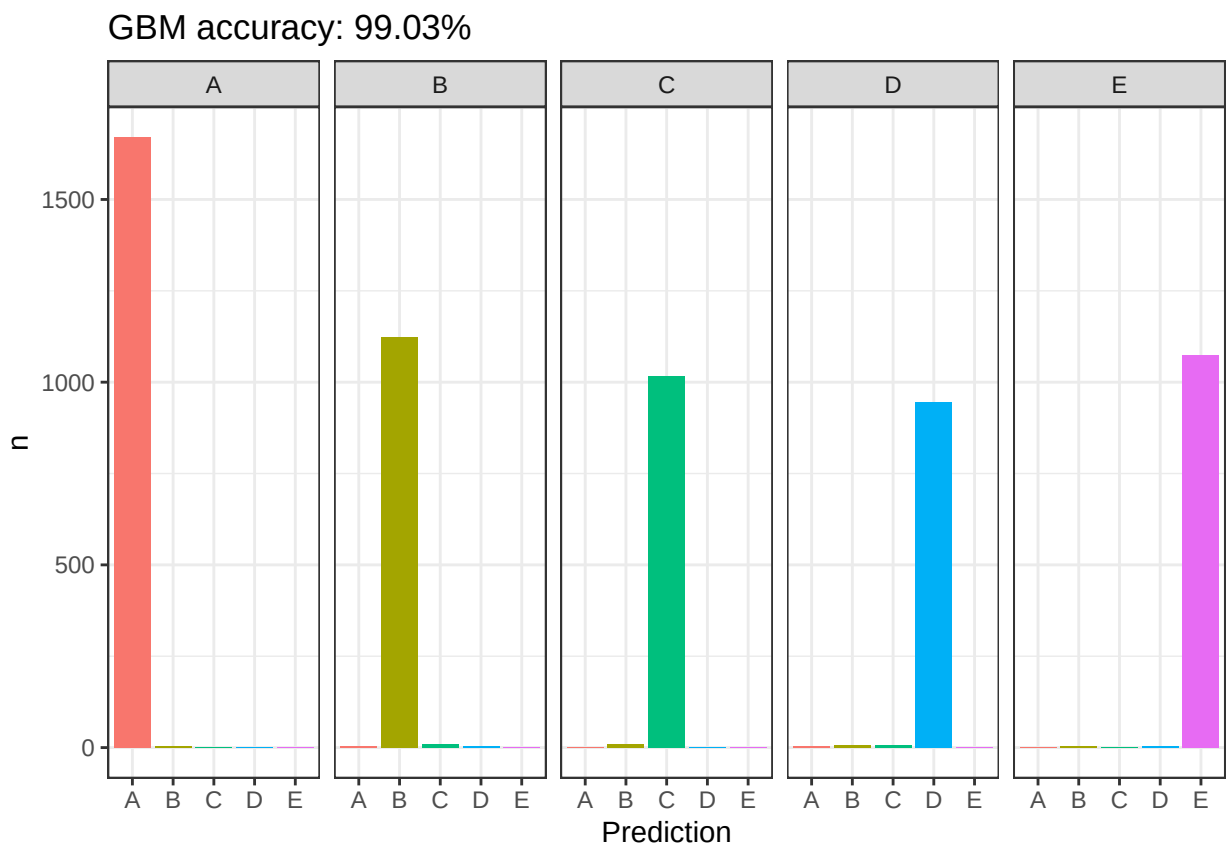
```
# validate on test data
predict_gbm <- predict(model_gbm, df_test)

# confusion matrix
cm_gbm <- confusionMatrix(predict_gbm, df_test$classe)

# plot confusion matrix results
df_cm_gbm <- cm_gbm$table %>% as_tibble()

df_cm_gbm %>% ggplot(aes(x = Prediction, y = n, fill = Prediction)) +
```

```
        geom_bar(stat = "identity") +
        facet_grid(~Reference) +
        ggtitle(paste0("GBM accuracy: ", round(cm_gbm$overall["Accuracy"], 4)*100, "%")) +
        theme_bw() +
        theme(legend.position = "none")
```



Accuracy: 99.03%, out-of-sample error rate: 0.97%

C: Random forest model

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

# training control
l_control_rf <- trainControl(method = "cv", number = 3, verboseIter = FALSE)

# fit model
model_rf <- train(classe ~ ., data = df_train, method = "rf", trControl = l_control_rf)
```

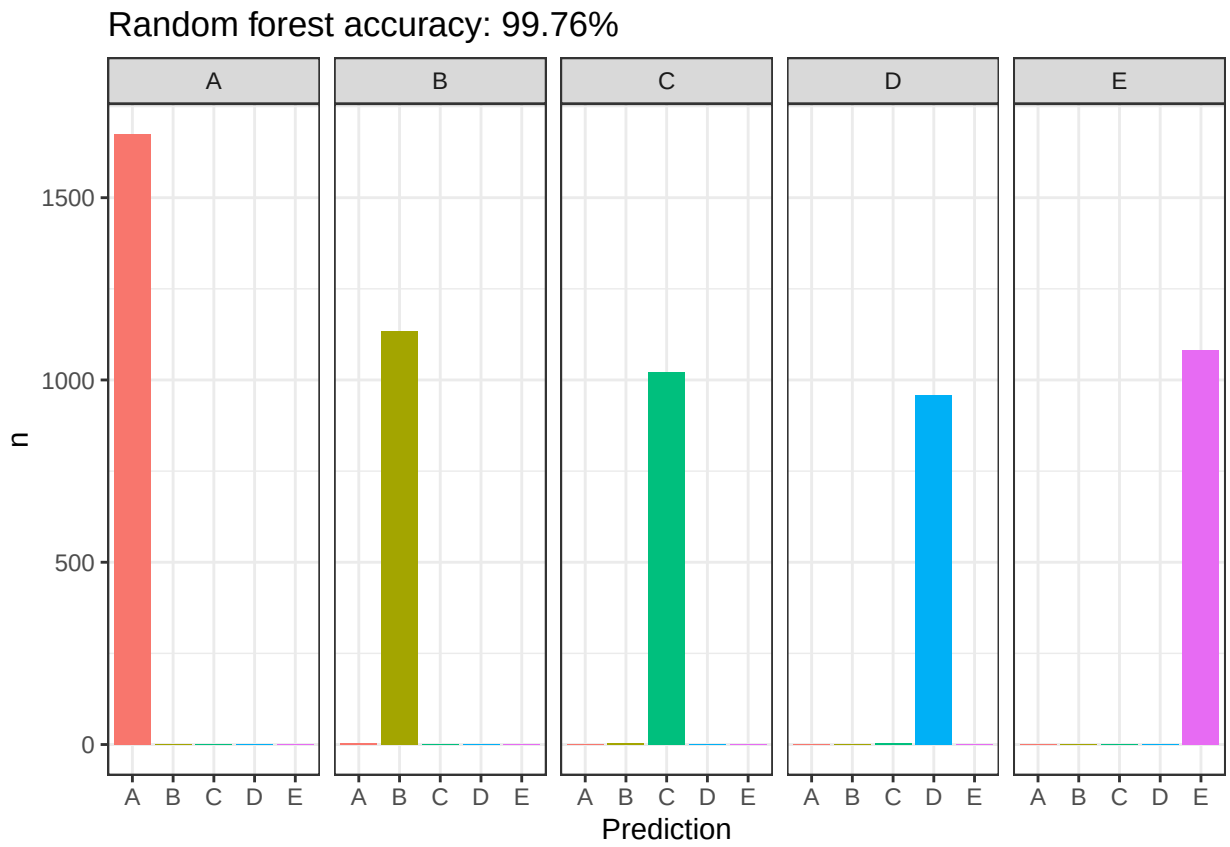
```
# validate on test data
predict_rf <- predict(model_rf, df_test)

# confusion matrix
cm_rf <- confusionMatrix(predict_rf, df_test$classe)
```

```
# plot confusion matrix results
df_cm_rf <- cm_rf$table %>% as_tibble()

df_cm_rf %>% ggplot(aes(x = Prediction, y = n, fill = Prediction)) +

       geom_bar(stat = "identity") +
       facet_grid(~Reference) +
       ggtitle(paste0("Random forest accuracy: ", round(cm_rf$overall["Accuracy"], 4)*100, "%")) +
       theme_bw() +
       theme(legend.position = "none")
```



Accuracy: 99.76%, out-of-sample error rate: 0.24% Very high accuracy possibly due to overfitting?

5. Apply model to quiz data

```
# RF had highest accuracy, use for quiz data

predict(model_rf, df_quiz)
```

```
##  [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
```