ML: Prediction Assignment

Course Project

Oscar Alonso

2024-03-14

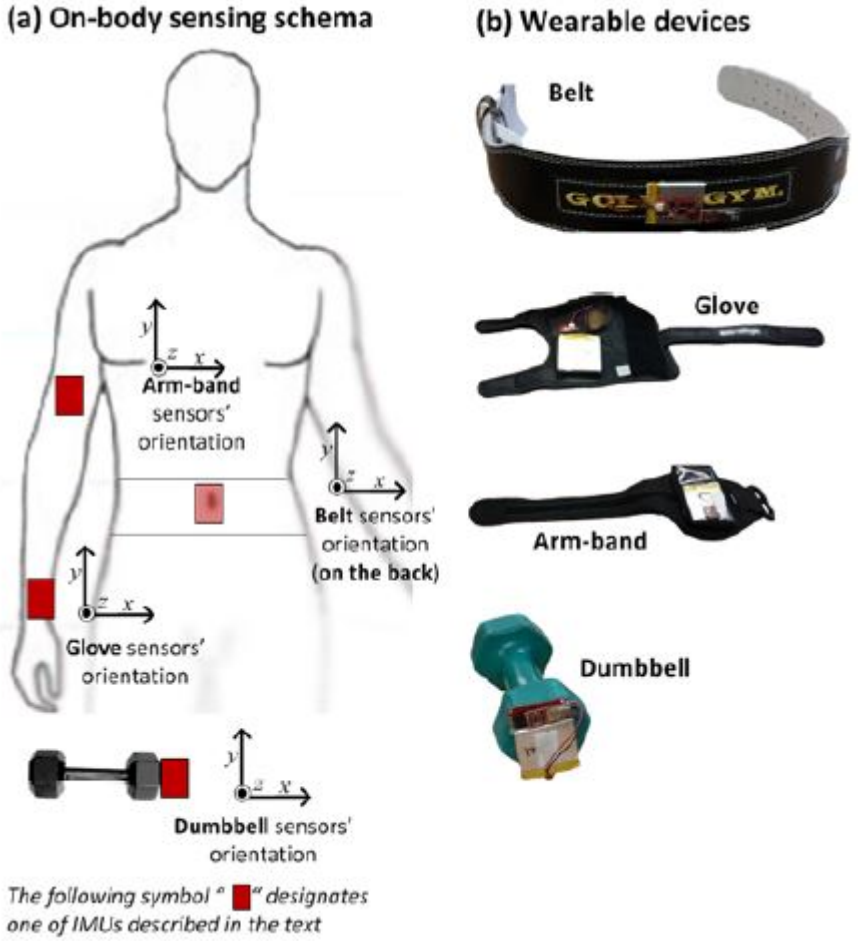
```
# Cargar librerias
library(readr)
library(caret)
library(dplyr)
library(ggplot2)
library(randomForest)
library(forecast)
library(rattle)
library(patchwork)
library(kableExtra)
```

INTRODUCTION AND BACKGROUND

Currently we are used to monitoring our daily activity using different smart devices (smart watches, smart phones, chest straps, pedometers…). This information is very valuable but we do not always obtain information about the quality with which we perform the exercises. This project tries to obtain a way to classify the quality with which the exercises have been carried out.

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test

cases.
The authors of the original study used: the sensors in the users' glove, armband, lumbar belt and dumbbell



The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Data collection and preparation

We collect the information they have collected and shared http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). They have been very generous in allowing their data to be used for this kind of assignment.

```
training <- read_csv("F:/COURSERA_FORMACION/Data Science_Statistics and Machine Learning Spec
ialization/Curso 03_Practical Machine Learning/Course_Project/pml-training.csv")
testing <- read_csv("F:/COURSERA_FORMACION/Data Science_Statistics and Machine Learning Speci
alization/Curso 03_Practical Machine Learning/Course_Project/pml-testing.csv")
```

After viewing the data we clean the files.
- We eliminate those columns that contain more than 90% NA values. - The first 7 columns correspond to user information and do not correspond to information about the devices used.

```
#colMeans(is.na(training))
col_borra <- which(colMeans(is.na(training))>0.9)
training[,col_borra] <- NULL
#colMeans(is.na(testing))
col_borra <- which(colMeans(is.na(testing))>0.9)
testing[,col_borra] <- NULL
training_scale <- training[,8:59] %>%
  bind_cols(classe = training$classe)
training_scale$classe <- as.factor(training_scale$classe)
#str(training_scale)
testing_scale <- testing[,8:59]
testing_scale <- testing_scale %>%
  bind_cols(problem_id = testing$problem_id)
```

With the rescaled training data we partition them at 75% and 25%. With 75% of the data we will use it to train our models and algorithms, while with 25% of the data we will verify the accuracy of our predictions. With the best model obtained we are going to implement it in the 20 test cases.

```
inTrain <- createDataPartition(training_scale$classe, p = 3/4)[[1]]
entrenamiento <- training_scale[inTrain, ]
verificacion <- training_scale[-inTrain, ]
```

Training Models and Predictions

We are going to use 4 training models and then we use them for predictions.

- Radom Forest
- Gradiente Boosting Machine
- Linear Discriminant Analysis
- Tree Bag

Random Forest

```
mod_rf <- train(classe~., data = entrenamiento, method = "rf")
#summary(mod_rf$finalModel)
pred_rf <- predict(mod_rf, verificacion)
# unique(pred_rf)
```

After training the model, we can verify the accuracy of our model with the 25% of the data that we have allocated for this.

```
confusionMatrix(pred_rf, verificacion$classe)$overall[1]
```

```
##  Accuracy
## 0.9940865
```

Gradient Boosting Machine

After training the model, we can verify the accuracy of our model with the 25% of the data that we have allocated for this.

```
confusionMatrix(pred_gbm, verificacion$classe)$overall[1]
```

```
##  Accuracy
## 0.9626835
```

Linear Discriminant Analysis

```
mod_lda <- train(classe~., method = "lda", data = entrenamiento)
#summary(mod_lda)
pred_lda <- predict(mod_lda, verificacion)
```

After training the model, we can verify the accuracy of our model with the 25% of the data that we have allocated for this.

```
confusionMatrix(pred_lda, verificacion$classe)$overall[1]
```

```
##  Accuracy
## 0.7000408
```

Tree Bag

```
mod_TB <- train(classe~., method = "treebag", data = entrenamiento)
#summary(mod_TB)
pred_TB <- predict(mod_TB, verificacion)
```

After training the model, we can verify the accuracy of our model with the 25% of the data that we have allocated for this.

```
confusionMatrix(pred_TB, verificacion$classe)$overall[1]
```

```
##  Accuracy
## 0.9836868
```

Combing Predictors

```
predDF <- data.frame(pred_rf, pred_gbm, pred_lda, pred_TB, classe = verificacion$classe)
combModFit <- train(classe~., method ="rf", data = predDF)
combPred <- predict(combModFit, predDF)
```

After training the model, we can verify the accuracy of our model with the 25% of the data that we have allocated for this.

```
confusionMatrix(combPred, predDF$classe)$overall[1]
```

```
##  Accuracy
## 0.9944943
```

Summary and graphs of the models made

We present a summary table of the results of the different models studied.

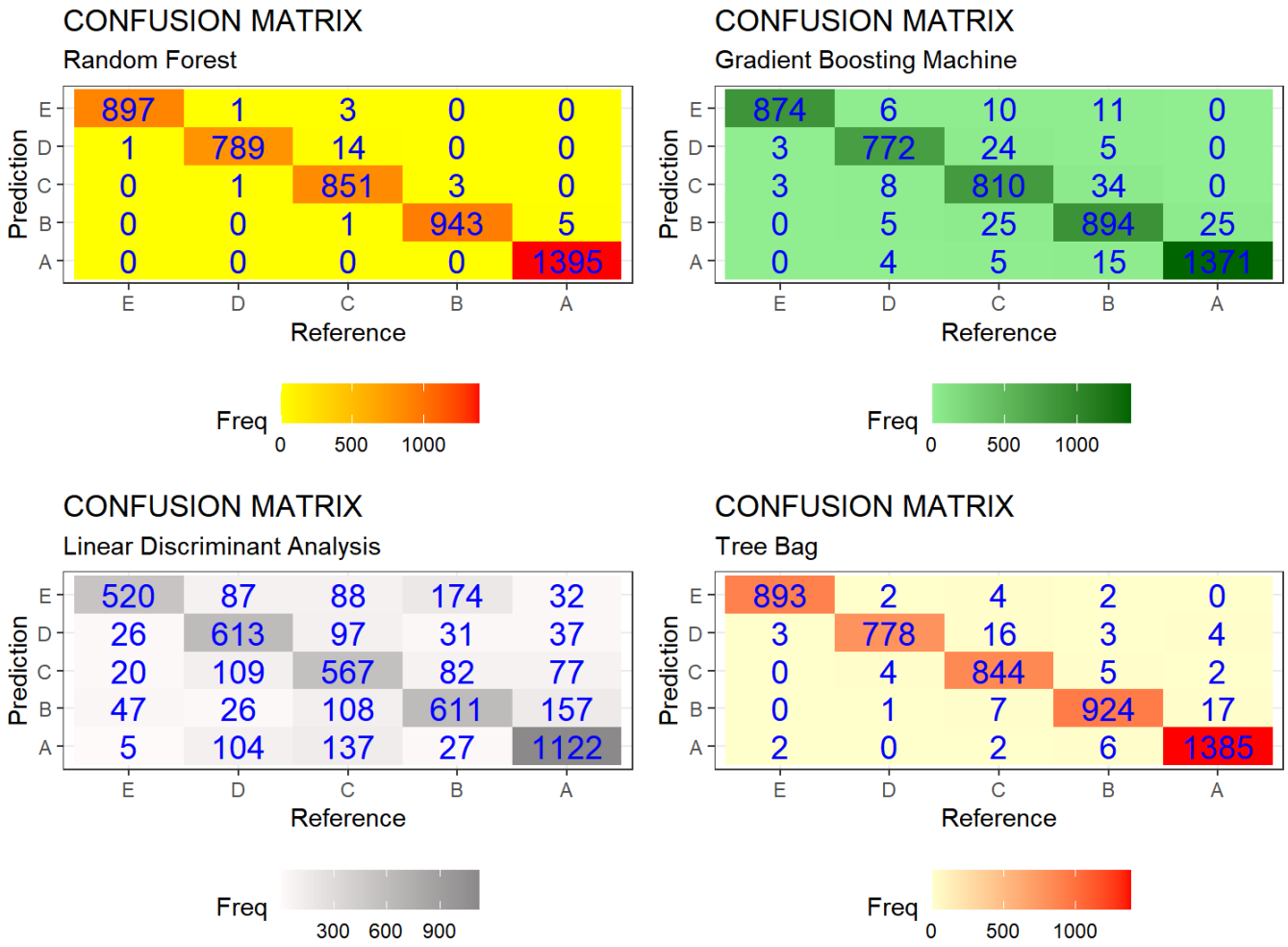
```
#### Tabla de Accuracy según modelos
Tabla_Accuracy <- data.frame(
  "Random Forest" = round(confusionMatrix(pred_rf, verificacion$classe)$overall[1],4),
  "Boosted Tree" = round(confusionMatrix(pred_gbm, verificacion$classe)$overall[1],4),
  "Linear Discriminant Analysis" = round(confusionMatrix(pred_lda, verificacion$classe)$overa
ll[1],4),
  "Tree Bag" = round(confusionMatrix(pred_TB, verificacion$classe)$overall[1],4),
  "Combining Predictors" = round(confusionMatrix(combPred, predDF$classe)$overall[1],4)
 )
kbl(Tabla_Accuracy,
    caption = "Accuracy Comparative Table",
    col.names = c("RANDOM FOREST", "BOOSTED TREE", "LINEAR DISCRIMINANT ANALYSIS", "TREE BA
G", "COMBINING PREDICTORS")) %>%
  kable_classic(bootstrap_options = "condensed" , full_width = F, font_size=14) %>%
  row_spec(0, monospace = F, color = "white",background = "navy", font_size = 16)%>%
  row_spec(row = 1, background = "lightyellow")%>%
  column_spec(column = c(1), background = "navy", color = "white") %>%
  column_spec(column = c(2,6), background = "tomato")
```

Accuracy Comparative Table

	RANDOM FOREST	BOOSTED TREE	LINEAR DISCRIMINANT ANALYSIS	TREE BAG	COMBINING PREDICTORS
Accuracy	0.9941	0.9627	0.7	0.9837	0.9945

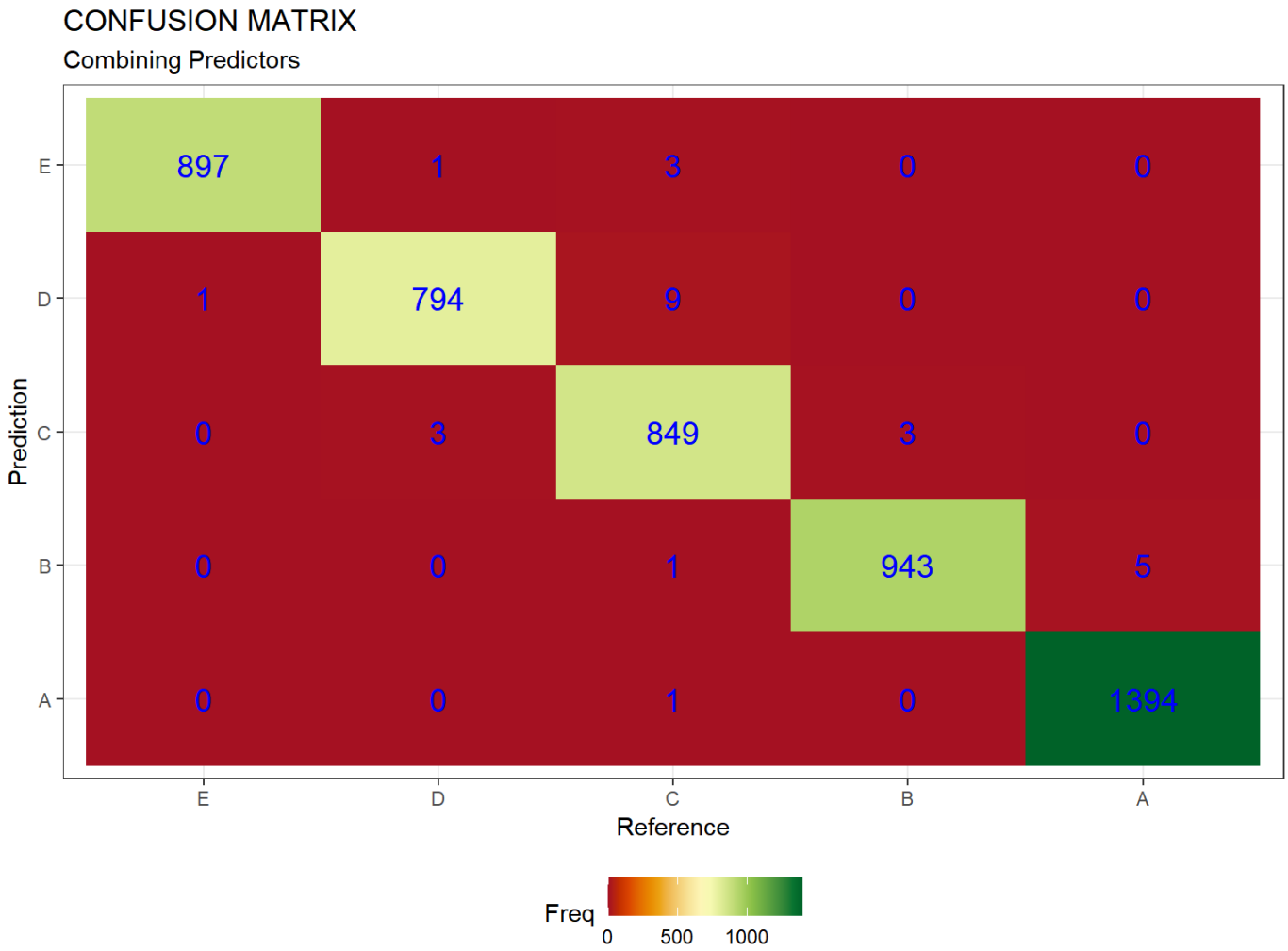
We represent the graphs of the confusion matrix of the different models that have been studied.

```
cm_RF <- confusionMatrix(pred_rf, verificacion$classe, dnn = c("Prediction", "Reference"))
plt <- as.data.frame(cm_RF$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))
g1 <- ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
  geom_tile() +  geom_text(aes(label=Freq), size = 5, color = "blue") +
  scale_fill_gradient(low="yellow", high="red") +
  labs(x = "Reference",y = "Prediction", title = "CONFUSION MATRIX", subtitle ="Random Fores
t")+
  theme_bw() + theme(legend.position = "bottom")

cm_gbm <- confusionMatrix(pred_gbm, verificacion$classe, dnn = c("Prediction", "Reference"))
plt <- as.data.frame(cm_gbm$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))
g2<-ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
  geom_tile() +  geom_text(aes(label=Freq), size = 5, color = "blue") +
  scale_fill_gradient(low="lightgreen", high="darkgreen") +
  labs(x = "Reference",y = "Prediction", title = "CONFUSION MATRIX", subtitle ="Gradient Boos
ting Machine")+
  theme_bw() +  theme(legend.position = "bottom")

cm_lda <- confusionMatrix(pred_lda, verificacion$classe, dnn = c("Prediction", "Reference"))
plt <- as.data.frame(cm_lda$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))
g3<-ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
  geom_tile() +  geom_text(aes(label=Freq), size = 5, color = "blue") +
  scale_fill_gradient(low="snow", high="snow4") +
  labs(x = "Reference",y = "Prediction", title = "CONFUSION MATRIX", subtitle ="Linear Discri
minant Analysis")+
  theme_bw() +  theme(legend.position = "bottom")

cm_TB <- confusionMatrix(pred_TB, verificacion$classe, dnn = c("Prediction", "Reference"))
plt <- as.data.frame(cm_TB$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))
g4<- ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
  geom_tile() +  geom_text(aes(label=Freq), size = 5, color = "blue") +
  scale_fill_gradient2(low = "#075AFF", mid = "#FFFFCC", high = "#FF0000") +
  labs(x = "Reference",y = "Prediction", title = "CONFUSION MATRIX", subtitle ="Tree Bag")+
  theme_bw() +  theme(legend.position = "bottom")

g_1_4 <- (g1+g2)/(g3+g4)
g_1_4
```



We also present the graph as a combination of models.

```
cm_Comb <- confusionMatrix(combPred, verificacion$classe, dnn = c("Prediction", "Reference"))
plt <- as.data.frame(cm_Comb$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))
g5 <- ggplot(plt, aes(Prediction,Reference, fill= Freq)) +
  geom_tile() +  geom_text(aes(label=Freq), size = 5, color = "blue") +
  scale_fill_gradientn(colors = hcl.colors(20, "RdYlGn")) +
  labs(x = "Reference",y = "Prediction", title = "CONFUSION MATRIX", subtitle ="Combining Pre
dictors")+
  theme_bw() +  theme(legend.position = "bottom")
g5
```



Model validation with 20 test cases We observe that the Random Forest model is very similar to the combination of predictors that we have made. For reasons of processing capacity of the PC used, we are going to classify the 20 test cases with this model.

```
pred_rf1 <- predict(mod_rf, testing_scale)
pred_rf1<- as.data.frame(pred_rf1) %>%
  rename("Random Forest Prediction" = "pred_rf1")
print(pred_rf1)
```

```
##    Random Forest Prediction
## 1                         B
## 2                         A
## 3                         B
## 4                         A
## 5                         A
## 6                         E
## 7                         D
## 8                         B
## 9                         A
## 10                        A
## 11                        B
## 12                        C
## 13                        B
## 14                        A
## 15                        E
## 16                        E
## 17                        A
## 18                        B
## 19                        B
## 20                        B
```