COURSERA MACHINE LEARNING PROJECT - CONSTRUCTING A PREDICTION ALGORITHM

INTRODUCTION AND SUMMARY

The object of this project is to construct an algorithm using computer machine learning to predict five different methods of performing dumbell lifts. Two data files were provided, a training set (pml-training.csv) containing 19,622 observationsn of 160 variables derived from sensor measurements, and a target set (pml-test.csv) of 20 sets of observations from which predictions of the dumbbell lift method used are to be made.

After fitting various models using the 'caret' package in R, an algorithm based on the 'k-nearest neighbour' (knn) model applied to a limited set of predictors was found to give the best results.

TRAINING DATA SET ANALYSIS

Examination of the traning data showed that many columns were sparsely populated, containing missing values and division by zero errors. The data set was subset to remove these columns, leaving 60 columns x 19,622 rows containing x, y and z readings from sensors on belt, arm, forearm and the dumbbell; together with roll, pitch, yaw and acceleration rates derived from the sensor readings.

In order to investigate a variety of different models this data was further subset to extract the response variable ('classe') and either the sensor readings; the roll/pitch/yaw/acceleration measurements; or the two groups combined. The resulting data was partitioned to provide training and test sets (75% / 25%).

MODELING With 'caret' PACKAGE in 'R'

Decision tree training models (method = "rpart") were created from each from each of the three groups of varaibles and applied to the test set. The results (which are tabulated in Appendix A - table 1), were extremely disappointing, the model based on the x / y / z sensor readings performed best, but the accuracy and kappa (concordance) values were very poor (0.492 / 0.346) and the proportion of correct predictions on the test set was only about 50%. (See Appendix B - Figures 1 and 2 for charts showing variations in the structure of this model when using different groups of predictors).

A second set of models based on K-nearest neighbour (method = "knn") was added to the code, in this case the measurements on roll/yaw/pitch/acceleration rates performed best. The accuracy and kappa (concordance) values were 0.927 / 0.907 and the proportion of correct predictions on the test set was 94%. (These results are tabulated in Appendix A - table 2)

This model was judged to be satisfactory and was applied to the test set to produce the required predictions which are listed in Appendix A - Table 3.

Appendix A

Table 1 : Predictions of 'classe' using decision tree model

```
'rpart' model with set.seed(23543) on file 'pml-training.csv' :
classe  Actual   Roll/pitch/yaw   Sensor x/y/z   All Predictors
-----------------------------------------------------------------
A        1395         3032            1890             2557
B         949            0             692              645
C         855         1467            1184             1297
D         804            0             882                0
E         901          405             256              405

Correct (%)          43.19           50.65            48.86
Accuracy             0.437           0.492            0.509
Kappa                0.256           0.346            0.361
cp                   0.042           0.025            0.036
-----------------------------------------------------------------
```

Table 2 : Predictions of ‘classe’ using K-nearest neighbour model

```
'knn' model with set.seed(23543) on file 'pml-training.csv' :
classe  Actual   Roll/pitch/yaw   Sensor x/y/z   All Predictors
-----------------------------------------------------------------
A        1395         1369            1455             1413
B         949          949             890              902
C         855          893             885              887
D         804          814             840              831
E         901          879             834              861

Correct (%)          94.09           89.25            91.46
Accuracy (k = 5)     0.926           0.887            0.906
Kappa                0.906           0.857            0.881
-----------------------------------------------------------------
```

Table 3 : Predictions on the ‘Test’ data set with ‘knn’ model

```
PREDICTIONS with set.seed(23543) on file 'pml-testing.csv':

        Roll/pitch/yaw   Gyros/accel/magnets   All Predictors
-------------------------------------------------------------
 1              B                   B                   B
 2              A                   A                   A
 3              B                   B                   B
 4              A                   A                   A
 5              A                   A                   A
 6              E                   E                   E
 7              D                   D                   D
 8              B                   B                   B
 9              A                   A                   A
10              A                   A                   A
11              B                   B                   B
12              C                   C                   C
13              B                   E *                 D *
14              A                   A                   A
15              E                   E                   E
16              E                   E                   E
17              A                   A                   A
18              B                   B                   B
19              B                   C *                 B
20              B                   B                   B
-------------------------------------------------------------

* Inconsistent predictions between the three models
```

Appendix B

Figure 1 :

Structure of ‘rpart’ Decision Tree model (all predictors)

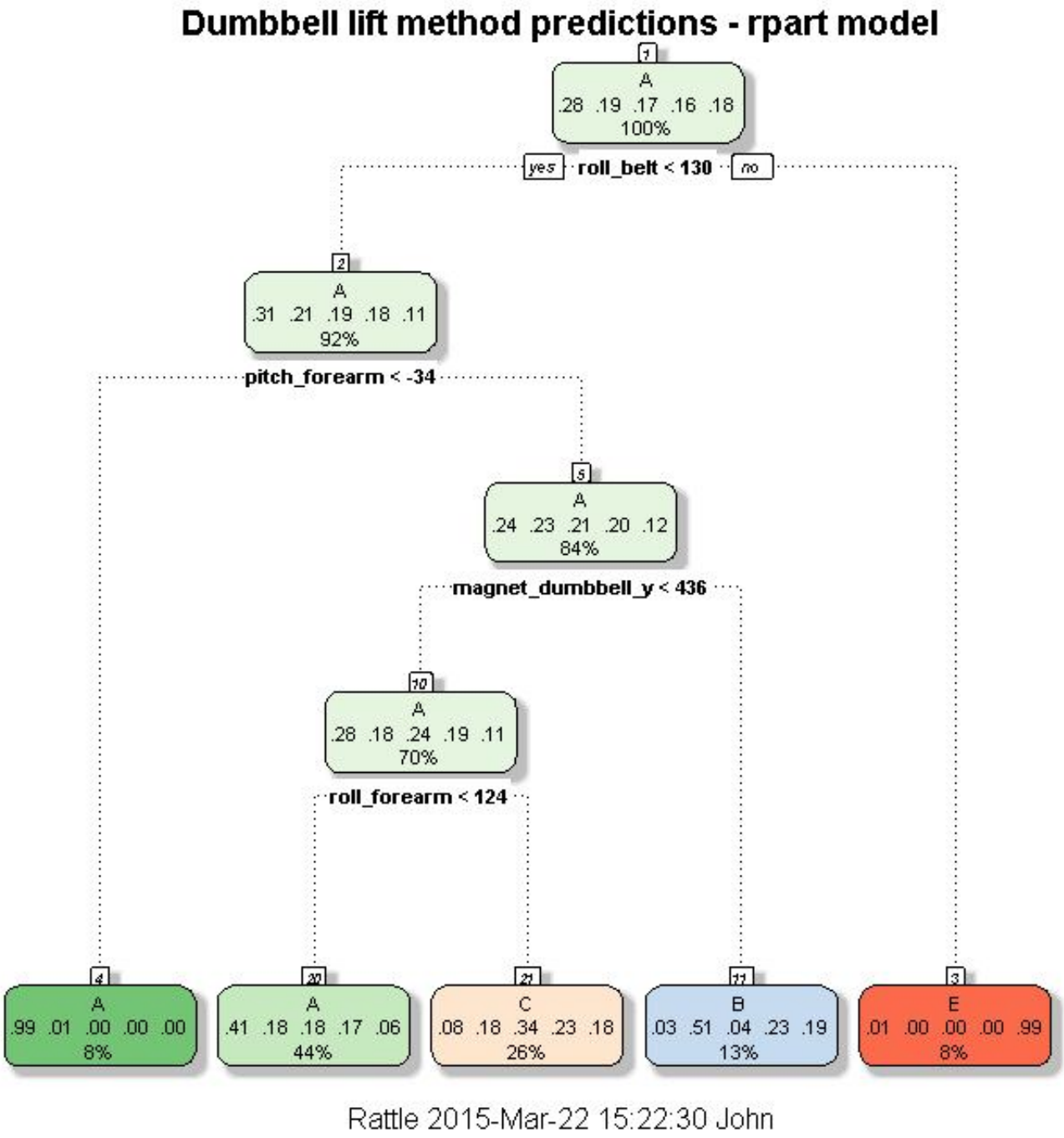
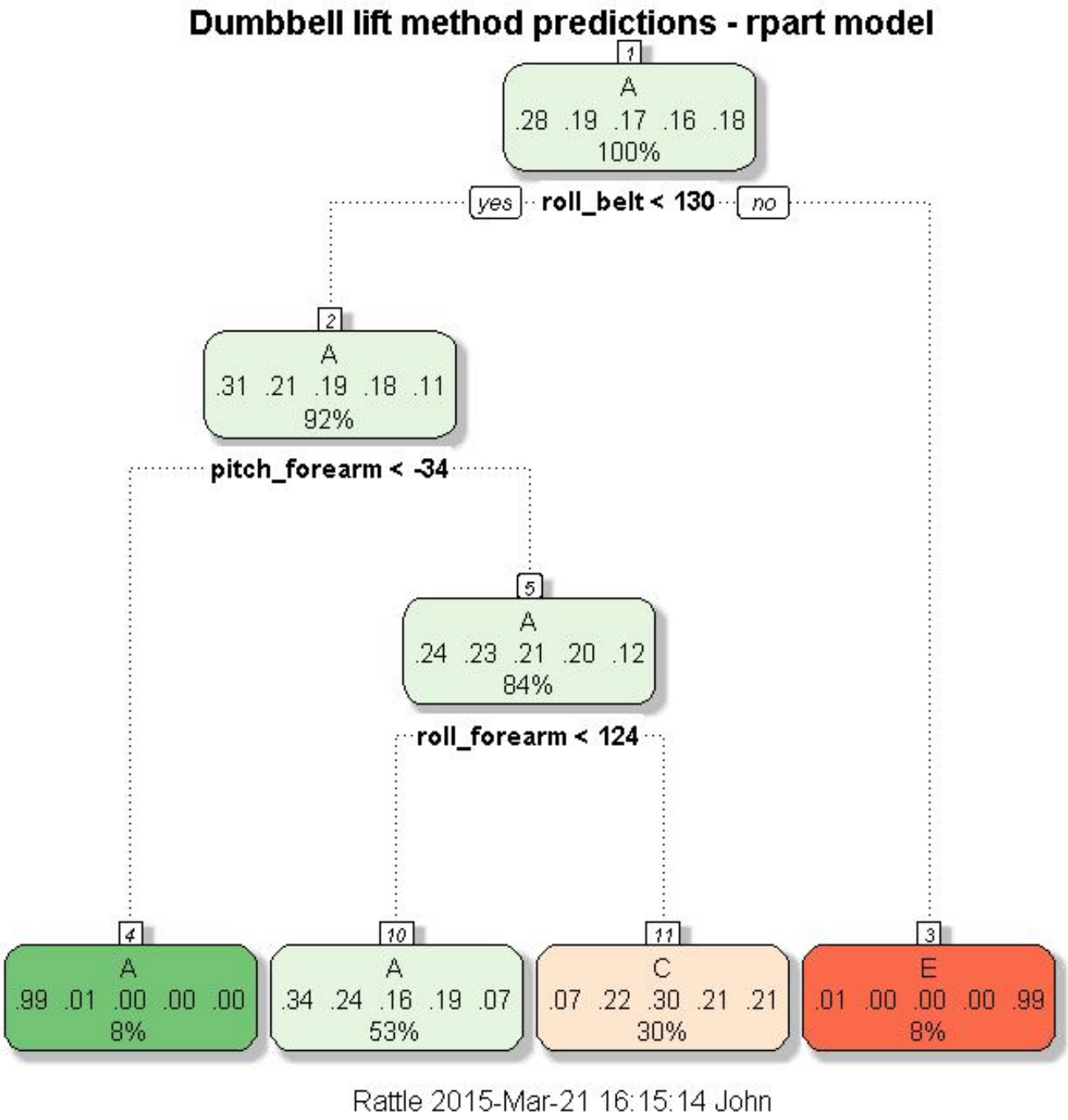


Figure 2 :

Structure of ‘rpart’ Decision Tree model (sensor x/y/z predictors)



Appendix C

R code used in the project

```
## Machine Learning Project - Predicting multiple factor variables

## Column numbers in files pml-training.csv and pml-testing.csv
## after removal of sparsely populated columns :

##               Movement    Sensors
## Belt          :  8:11      12:20
## Arm           : 21:24      25:33
## Dumbbell      : 34:37      38:46
## Forearm       : 47:50      51:59
```

```
## classe       : 60

library(caret)
library(rpart)
library(rattle)

## Select predictor and response variables
COLUMNS <- 8:60                           ## all predictors
## COLUMNS <- c(8:11, 21:24, 34:37, 47:50, 60)   ## rates
## COLUMNS <- c(12:20, 25:33, 38:46, 51:59, 60)  ## sensor x/y/z
set.seed(23543)   ## for repeatability

## Get TARGET data set and extract the required columns
DATA <- read.csv("pml-testing.csv")
COLS <- length(DATA)
FLAG <- rep(TRUE, COLS)
for(i in 1:COLS)
{  if(is.na(DATA[1, i])) FLAG[i] <- FALSE
}
PRED <- subset(DATA, select=(FLAG==TRUE))
PRED <- data.frame(PRED[ , COLUMNS])
sum(complete.cases(PRED)*1)-dim(PRED[ ,1]) ## verify complete cases

## Get TRAINING data set, extract matching columns and
## partition into TRAINING & TEST sets (75%/25%)
TEMP <- read.csv("pml-training.csv")
TEMP <- subset(TEMP, select=(FLAG==TRUE))
DATA <- data.frame(TEMP[ , COLUMNS])
sum(complete.cases(DATA)*1)-dim(DATA[ ,1]) ## verify complete cases
LIST <- createDataPartition(y=DATA$classe, p=0.75, list=FALSE)
TRAIN <- DATA[LIST, ]
TEST <- DATA[-LIST, ]
rm(DATA); rm(TEMP)

COLNAMES <- cbind(colnames(TRAIN), colnames(PRED))
View(COLNAMES)   ## check same columns selected in TARGET & TRAINING sets

## Construct and test prediction algorithms with cross validation
## on partitioned training data
CTRL <- trainControl(method="cv", repeats=20, seeds=NULL)

## 'rpart' model
modFit1 <- train(classe ~ ., data=TRAIN, method="rpart", trControl=CTRL)
Prediction1 <- predict(modFit1, newdata=TEST)

## 'k-nearest-neighbour' model
modFit2 <- train(classe ~ ., data=TRAIN, method="knn", trControl=CTRL)
Prediction2 <- predict(modFit2, newdata=TEST)

Table1 <- data.frame(ObsNo=1:length(TEST$classe), Classe=TEST$classe, P_rpart=Prediction1,
```

```
P_knn=Prediction2)
View(Table1)
View(data.frame(Observed=summary(TEST$classe), P_rpart=summary(Prediction1), P_knn=summary(P
rediction2)))

## Predict classe in TARGET data ('knn' model)
Prediction <- predict(modFit2, newdata=PRED)
TableA <- data.frame(ObsNo=1:20, Predict=Prediction)
View(TableA)

modFit1
sum((TEST$classe == Prediction1)*1) / length(TEST$classe) * 100 ## accuracy rpart
modFit2
sum((TEST$classe == Prediction2)*1) / length(TEST$classe) * 100 ## accuracy knn

## Plot the'rpart' model
fancyRpartPlot(modFit1$finalModel, main="Dumbbell lift method predictions - rpart model\n")
```

CITATION : the data set used in this project is the Weightlifting Exercises data published by :-

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.