Machine Learning Project

Channaveer Patil

4/14/2020

Summary:

This report is to summarize Practical Machine Learning Module Project work. Weight lifting data set is considered. This work tries to predict the way participants did exercise. Detailed description is provided for the prediction model that is developed, usage of cross validation, discuss sample error. Explain choices made, finally use model to predict results of 20 test cases. 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).

Acknowledgement:

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).

Data Provided:

Two worksheets: pml-training.csv and pml-testing.csv

The subjects were tracked during weightlifting exercises, and sensors were located in their arms, forearms, and belt areas, and a sensor was also positioned in the dumbbells. Several three-dimensional measurements were taken while the participants did dumbbell biceps curls, in five different fashions (classes):

- Classe A: exactly according to the specification.

- Classe B: throwing the elbows to the front.

- Classe C: lifting the dumbbell only halfway.

- Classe D: lowering the dumbbell only halfway.

- Classe E: throwing the hips to the front.

Model will be built based on training data (pml-training.csv) and tested on testing data (pml-testing.csv)

On a cursory review we know The training data from our set contains 19622 observations of 160 variables.

Required packages and libraries:

install.packages(“kableExtra”) - One time install.packages(“caret”) - One time

install.packages(“rattle”) - One time

install.packages(“rpart”) - One time

```
library(kableExtra); # data formatting and clean presentation
```

```
## Warning: package 'kableExtra' was built under R version 3.6.3
```

```
library(caret);
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rattle);
```

```
## Warning: package 'rattle' was built under R version 3.6.3
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart);
```

```
## Warning: package 'rpart' was built under R version 3.6.3
```

Loading given data

```
training <- read.csv("c:\\pml-training.csv", header = TRUE)
testing <- read.csv("c:\\pml-testing.csv", header = TRUE)
```

Cursory review of data structure

Training - 19622 observations of 160 variables

Testing - 20 observations of 160 variables

head(training) - avoiding execution to minimize report length

Many NAs, few initial columns with non vital for prediction

Formatting User and Classe Table

```
t1 <- table(training$user_name, training$classe)
kable(t1, caption = "Users-Classe") %>%
kable_styling(bootstrap_options = "striped", full_width = FALSE, latex_options = "hold
_position")
```

Users-Classe

	A	B	C	D	E
adelmo	1165	776	750	515	686
carlitos	834	690	493	486	609
charles	899	745	539	642	711
eurico	865	592	489	582	542
jeremy	1177	489	652	522	562
pedro	640	505	499	469	497

Identify and Remove predictors with near zero variance

```
nzv <- nearZeroVar(training)
training <- training[, -nzv]
```

160 variables reduced to 100

Remove variables with mostly NA values say > 90%

```
isNA <- sapply(training, function(x) mean(is.na(x))) > 0.90
training <- training[, isNA == FALSE]
```

100 variables reduced to 59

Remove identifier Variables - no vital for prediction

```
training <- training[, -(1:5)]
```

59 variables reduced to 54

Create validation set

```
set.seed(123)
inTrain  <- createDataPartition(training$classe, p = 0.7, list = FALSE)
trainSet <- training[inTrain, ]
valSet <- training[-inTrain, ]
dim(trainSet); dim(valSet)
```

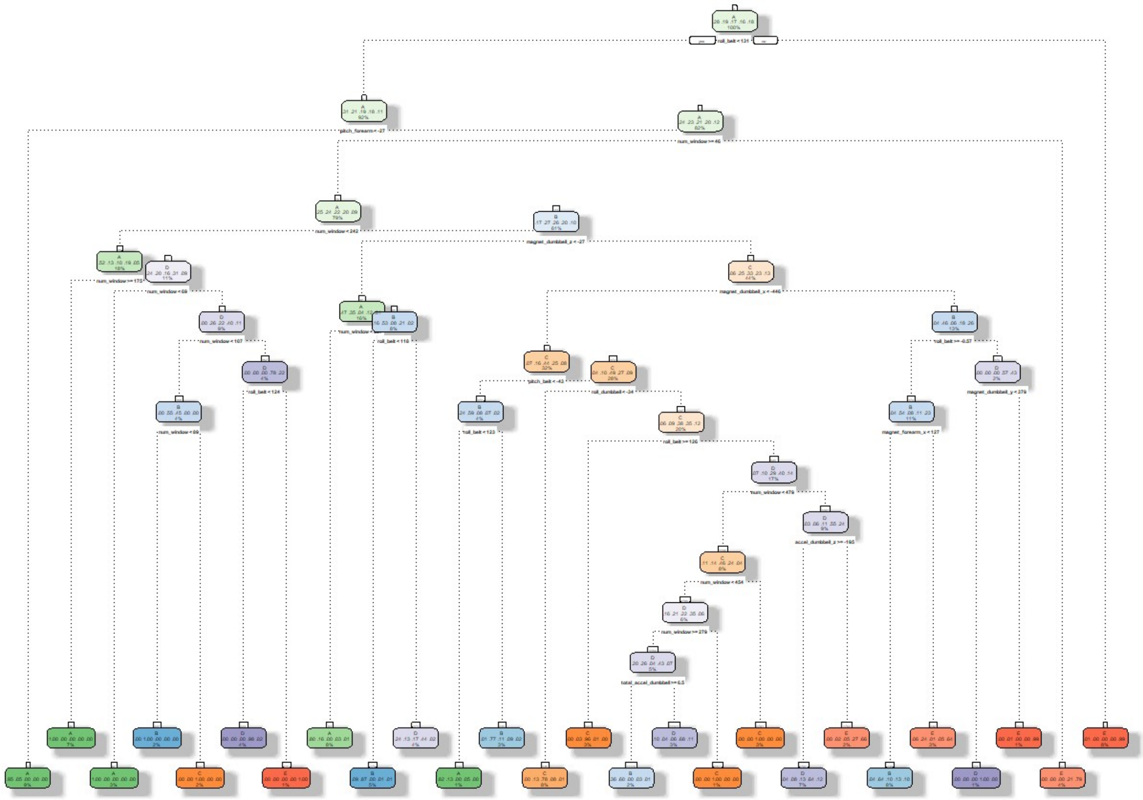
```
## [1] 13737    54
```

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

Regression Trees model

```
set.seed(223)
ModelTree <- rpart(classe ~ ., data = trainSet, method = "class")
fancyRpartPlot(ModelTree)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2020-Apr-14 11:06:08 H138973

# ModelTree # get text version # Classe summary table

```
predTree <- predict(ModelTree, newdata = valSet, type = "class")
confTree <- confusionMatrix(predTree, valSet$classe)
tree_overall <- as.data.frame(confTree$overall)
names(tree_overall) <- c("Value")

kable(tree_overall, caption = "Overall Statistics") %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE, latex_options = "ho
ld_position")
```

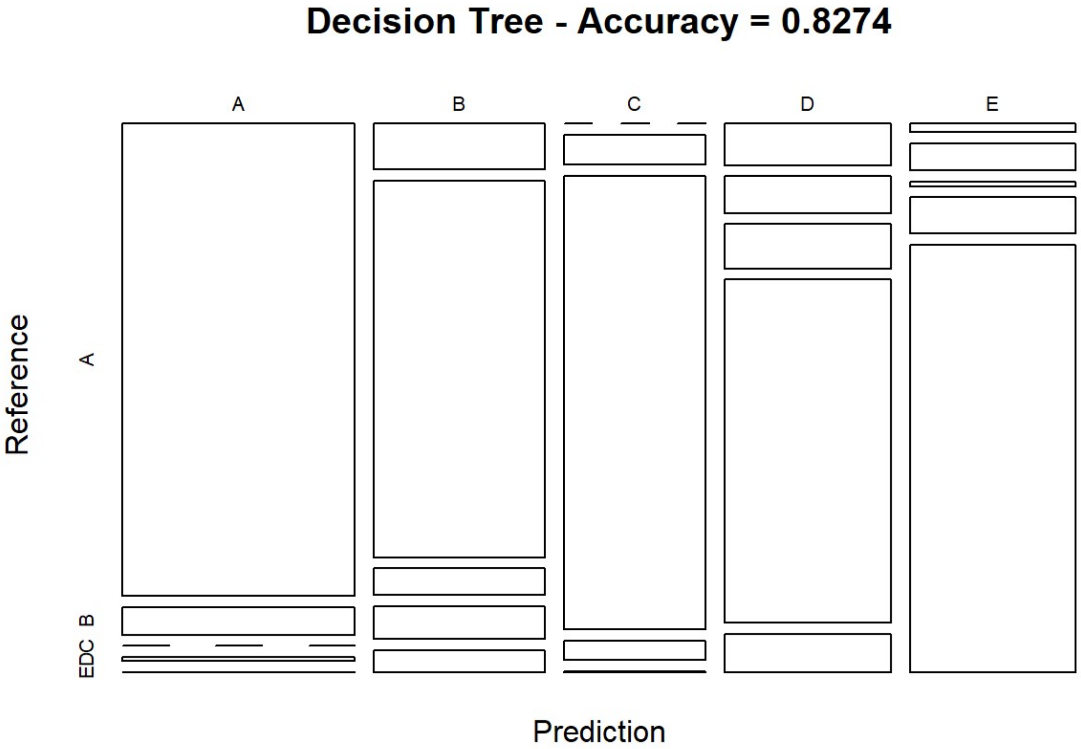
Overall Statistics

	Value
Accuracy	0.8273577
Kappa	0.7822750
AccuracyLower	0.8174552

	Value
AccuracyUpper	0.8369345
AccuracyNull	0.2844520
AccuracyPValue	0.0000000
McnemarPValue	NaN

Plotting decission tree

```
plot(confTree$table, col = confTree$byClass, main = paste("Decision Tree - Accuracy
=", round(confTree$overall["Accuracy"], 4)))
```



# Decision Tree Accuracy = 0.8274 # # Random Forests model and predictions

```
set.seed(323)
tControl <- trainControl(method="cv", number=3, verboseIter=FALSE)
ModelRF <- train(classe ~ ., data=trainSet, method="rf", trControl = tControl)
```

Predicting

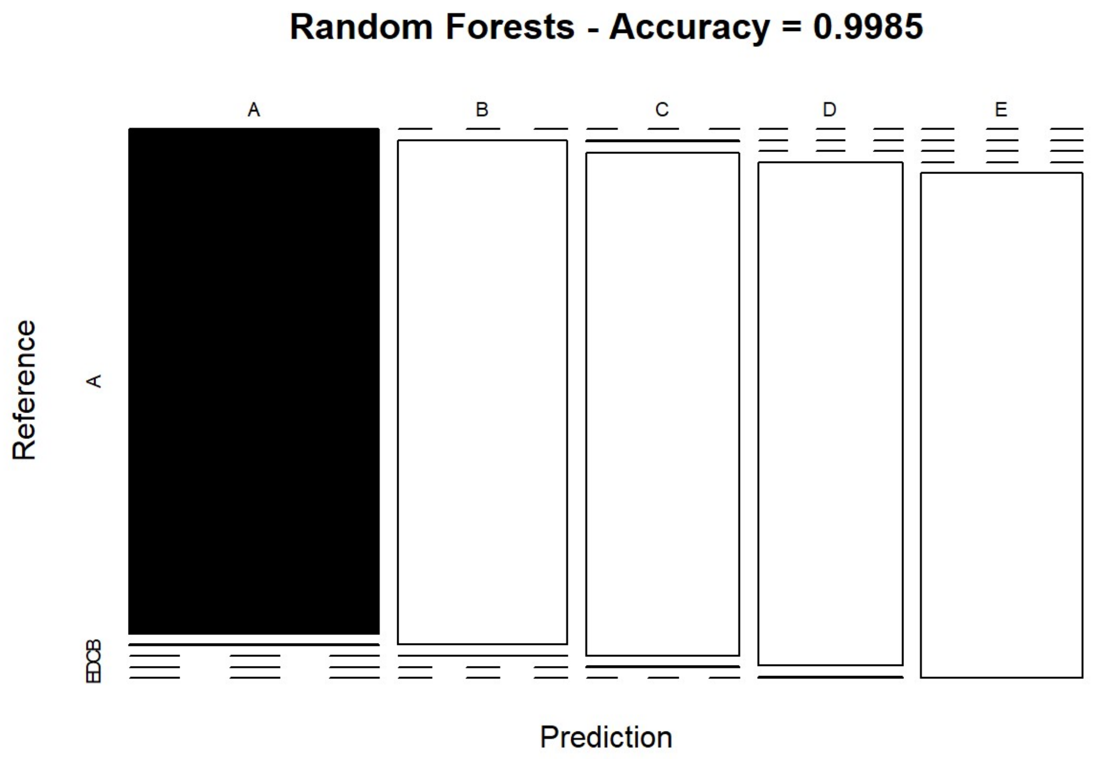
```
predRF <- predict(ModelRF, newdata = valSet)
confRF <- confusionMatrix(predRF, valSet$classe)
RF_overall <- as.data.frame(confRF$overall)
names(RF_overall) <- c("Value")

kable(RF_overall, caption = "Overall Statistics") %>%
kable_styling(bootstrap_options = "striped", full_width = FALSE, latex_options = "hold
_position")
```

Overall Statistics

	Value
Accuracy	0.9984707
Kappa	0.9980656
AccuracyLower	0.9970989
AccuracyUpper	0.9993005
AccuracyNull	0.2844520
AccuracyPValue	0.0000000
McnemarPValue	NaN

Plotting decission tree

```
plot(confRF$table, col = confRF$byClass,
main = paste("Random Forests - Accuracy =",
round(confRF$overall["Accuracy"], 4)))
```



# Boosting model and predictions

```
set.seed(423)
bControl <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
ModelGBM <- train(classe ~ ., data=trainSet, method="gbm",
                  trControl = tControl, verbose = FALSE)
```

Predicting

```
predGBM <- predict(ModelGBM, newdata = valSet)
confGBM <- confusionMatrix(predGBM, valSet$classe)
GBM_overall <- as.data.frame(confGBM$overall)
names(GBM_overall) <- c("Value")

kable(GBM_overall, caption = "Overall Statistics") %>%
kable_styling(bootstrap_options = "striped", full_width = FALSE,
latex_options = "hold_position")
```

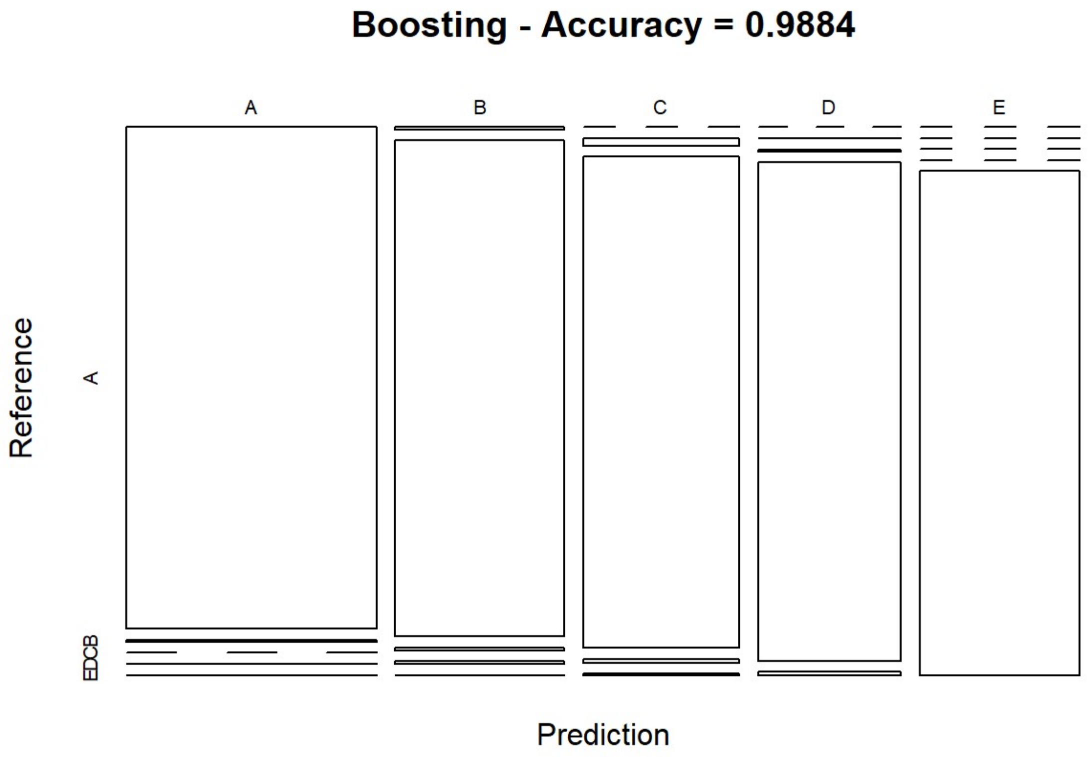
Overall Statistics

	Value

	Value
Accuracy	0.9884452
Kappa	0.9853839
AccuracyLower	0.9853742
AccuracyUpper	0.9910164
AccuracyNull	0.2844520
AccuracyPValue	0.0000000
McnemarPValue	NaN

Plotting decission tree

```
plot(confGBM$table, col = confGBM$byClass, main = paste("Boosting - Accuracy =",
round(confGBM$overall["Accuracy"], 4)))
```



# Model Comparison

```
Accuracy <- data.frame(Model = c("Regression Trees","Random Forests",
"Boosting"), Accuracy = c(round(confTree$overall["Accuracy"], 4),
round(confRF$overall["Accuracy"], 4), round(confGBM$overall["Accuracy"], 4)))

kable(Accuracy, caption = "Models' accuracy") %>%
kable_styling(bootstrap_options = "striped", full_width = FALSE,
latex_options = "hold_position")
```

Models’ accuracy

Model	Accuracy
Regression Trees	0.8274
Random Forests	0.9985
Boosting	0.9884

Conclusion

Among the models developed, Random Forest based model is providing maximum accuracy. This will be used to respond to 20 Quiz questions.