PeerAssignment

Martin Alayon

6/15/2021

Introduction

The goal off this report is to create a model to predict the manner in which six participants, using accelerometers on a belt, did barbell lifts. There are 5 different ways (A,B,C,D and E) recorded in a variable named "classe" which the model should predict as a function of the accelerometers information. The data sert was obtained from Human Activity Recognition (puc-rio.br).

Exploration data

After downloading the data, it was loaded to explore its content.

```
training<- read.csv("pml-training.csv")
testing<- read.csv("pml-testing.csv")
```

There are a training and a testing data set with 160 variables and 19622 and 20 observation respectively. Some of the variables in both dataset have Nan's and were eliminated. Also, variables with participants names and date information were avoided.

```
VIdx<-!is.na(testing[1,])  #only no NA variables
VIdx[1:7]<-FALSE  #Eliminate 7 first variables
training_small<-training[,VIdx]
testing_small<-testing[,VIdx]
```

The resulting data set have the same number of observation but less variables (160). The variable named "classe" was treated as a factor variables with 5 levels

```
table(training_small$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Model fitting

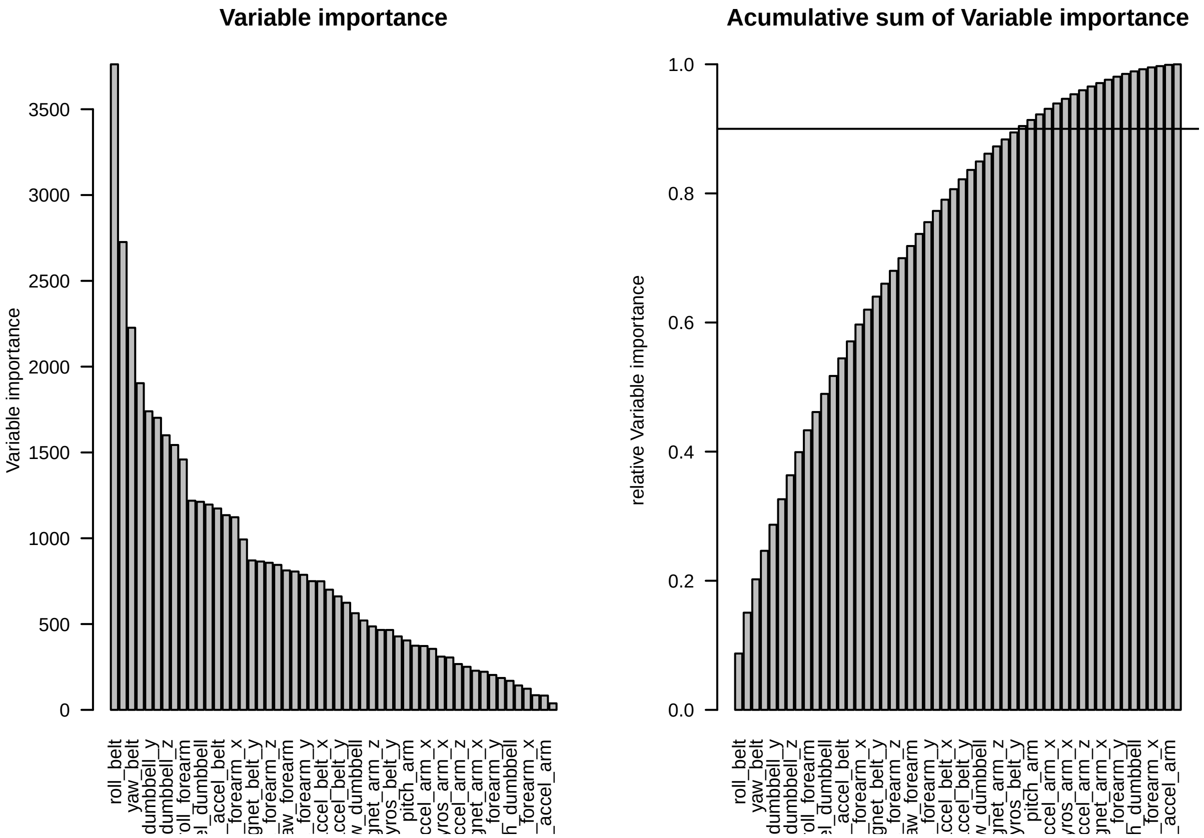
A 10 k fold validation method was used to fit a tree model.First, "classe" variable was predicted using all others variables and importance of each variable was evaluated in order to select the most important regressors.

```
library(caret)
```

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

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

```
set.seed(666)
train_control<-trainControl(method = "cv",number = 10)
fit_rpart <- train(as.factor(classe)~.,data = training_small,trControl=train_control,method ="rpart",tu
par(mfrow=c(1,2),cex=0.6)
barplot(fit_rpart$finalModel$variable.importance,ylab = "Variable importance",las=2,main = "Variable imp
barplot(cumsum(fit_rpart$finalModel$variable.importance)/sum(fit_rpart$finalModel$variable.importance),
        ylab = "relative Variable importance",las=2,main = "Acumulative sum of Variable importance")
abline(h=0.9)
```



The horizontal line shows 90% of the relative importance. Next, in order to build a simpler model, a tree was fitted using only the 90% most important variables.

```
Idx<-cumsum(fit_rpart$finalModel$variable.importance)/sum(fit_rpart$finalModel$variable.importance)<0.9
reg<-c(names(fit_rpart$finalModel$variable.importance[Idx]),"classe")
fit_rpart90 <- train(as.factor(classe)~.,data = training_small[,reg],trControl=train_control,
                     method ="rpart",tuneLength=80)
rbind(fit_rpart$results[1,],fit_rpart90$results[1,])
```

```
##             cp  Accuracy     Kappa  AccuracySD     KappaSD
## 1 3.560746e-05 0.9404748 0.9247073 0.005937223 0.007519048
## 2 1.281869e-04 0.9366538 0.9198442 0.006636850 0.008408144
```

Although some accuracy is lost, it is not too much and the model is much more simpler.

A quite high accuracy was achieved.Using the entire training data set, the confusion matrix shows very high level of accuracy, sensitivity and specificity.

```
y<-predict(fit_rpart90,training_small)
M90<-confusionMatrix(y,as.factor(training_small$classe))
M90
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 5483   98   17   14   12
##          B   46 3560   63   37   49
##          C   17   72 3281   56   14
##          D   15   32   44 3087   39
##          E   19   35   17   22 3493
## 
## Overall Statistics
## 
##                Accuracy : 0.9634
##                  95% CI : (0.9607, 0.966)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9537
## 
##  Mcnemar's Test P-Value : 0.0008075
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9826   0.9376   0.9588   0.9599   0.9684
## Specificity            0.9900   0.9877   0.9902   0.9921   0.9942
## Pos Pred Value         0.9749   0.9481   0.9538   0.9596   0.9741
## Neg Pred Value         0.9931   0.9851   0.9913   0.9921   0.9929
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2794   0.1814   0.1672   0.1573   0.1780
## Detection Prevalence   0.2866   0.1914   0.1753   0.1639   0.1828
## Balanced Accuracy      0.9863   0.9626   0.9745   0.9760   0.9813
```

Conclusions

- A high level of accuracy, sensitivity and specificity was achieve.
- 33 variable are enough to make high accuracy predictions.
- A tree model is a quite simple model, easy to train and tune and very fast.