PMLAssignment.R

Gautham

2020-07-06

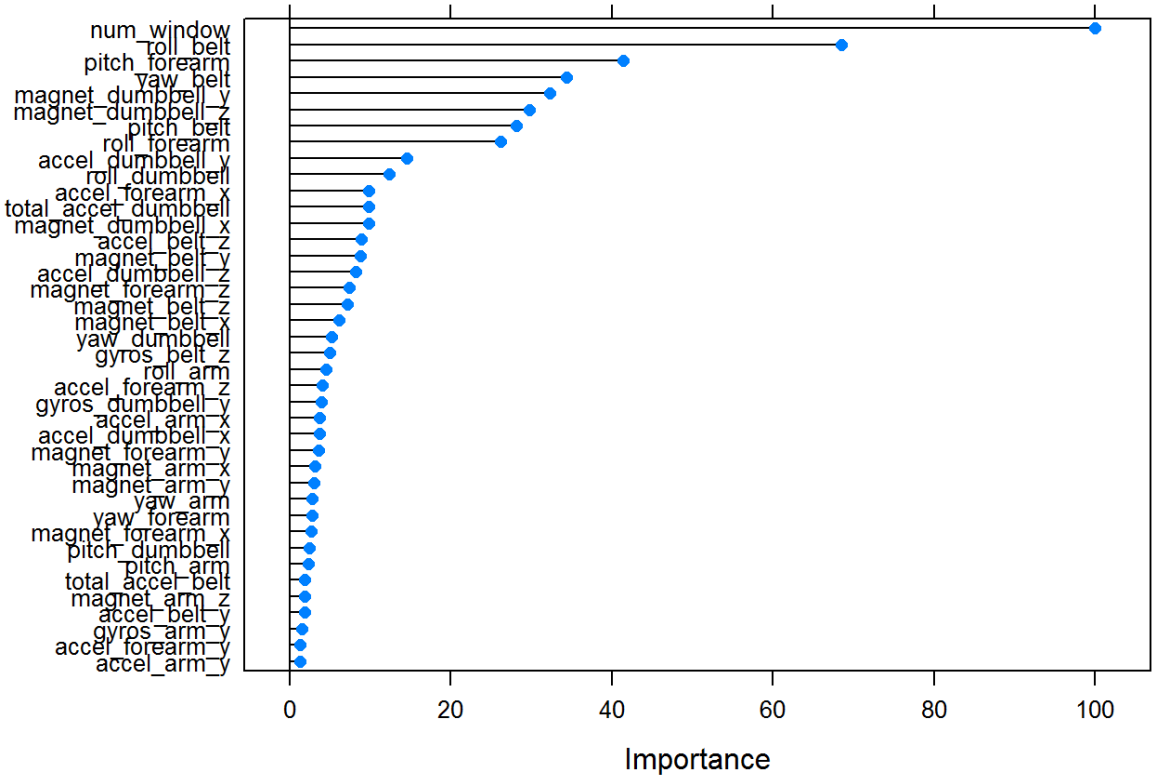
```
library(caret)

ibrary(ggplot2)
library(randomForest)

library(gbm)

library(doParallel)

library(survival)

library(splines)

library(plyr)

pml.training <- read.csv("C:/Users/91996/Downloads/pml-training.csv")

View(pml.training)

pml.testing <- read.csv("C:/Users/91996/Downloads/pml-testing.csv")

View(pml.testing)

training <- pml.training[, 6:dim(pml.training)[2]]

treshold <- dim(training)[1] * 0.95

goodColumns <- !apply(training, 2, function(x) sum(is.na(x)) > treshold  || sum(x=="") > treshold)

training <- training[, goodColumns]

library(caret)

badColumns <- nearZeroVar(training, saveMetrics = TRUE)

training <- training[, badColumns$nzv==FALSE]

training$classe = factor(training$classe)

inTrain <- createDataPartition(training$classe, p = 0.6)[[1]]

crossv <- training[-inTrain,]

training <- training[ inTrain,]

inTrain <- createDataPartition(crossv$classe, p = 0.75)[[1]]

crossv_test <- crossv[ -inTrain,]
```

```
crossv <- crossv[inTrain,]
testing <- pml.testing[, 6:dim(pml.testing)[2]]
testing <- testing[, goodColumns]
testing$classe <- NA
testing <- testing[, badColumns$nzv==FALSE]
mod1 <- train(classe ~ ., data=training, method="rf")
pred1 <- predict(mod1, crossv)
confusionMatrix(pred1, crossv$classe)
```

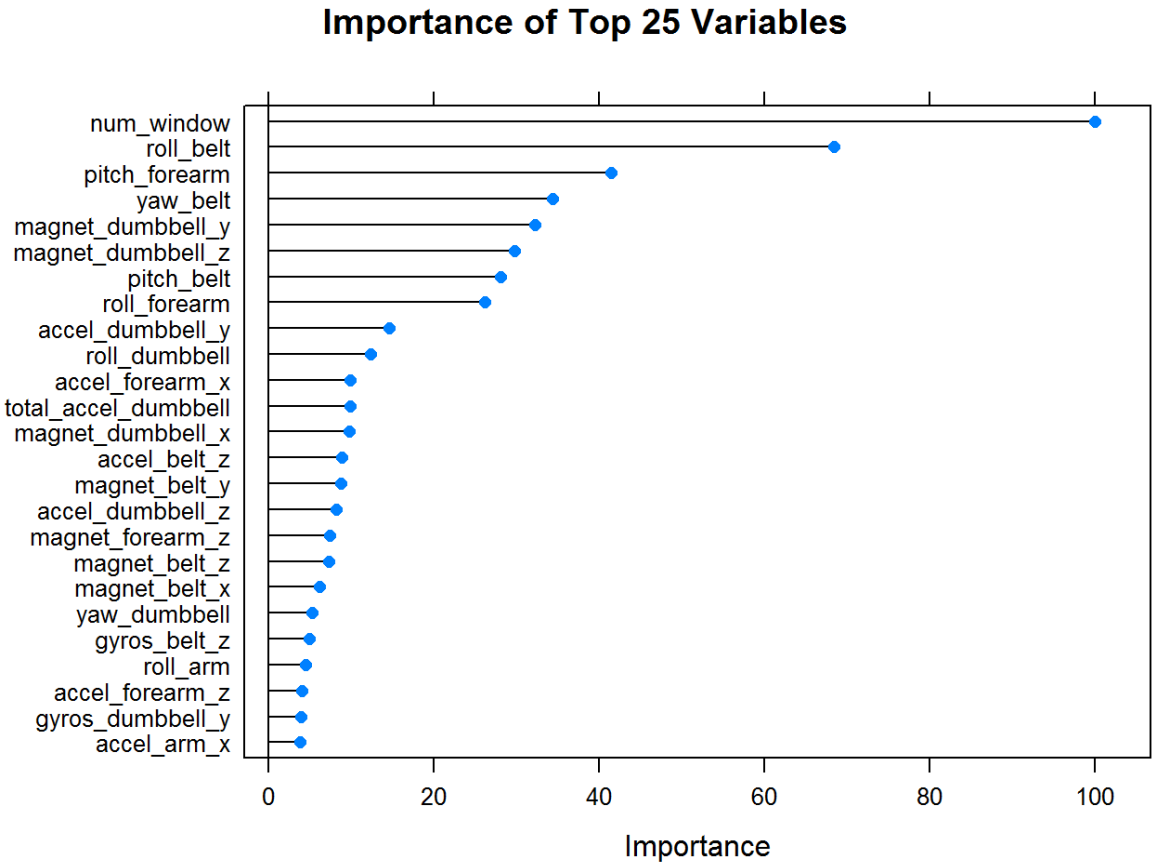
```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1672    3    0    0    0
         B    1 1135    6    0    0
         C    0    1 1020    4    0
         D    0    0    0  960    1
         E    1    0    0    1 1081

Overall Statistics

               Accuracy : 0.997
                 95% CI : (0.995, 0.998)
    No Information Rate : 0.284
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.996
 Mcnemar's Test P-Value : NA

Statistics by Class:
```

	Class: A	Class: B	Class: C	Class: D	Class: E
Sensitivity	0.999	0.996	0.994	0.995	0.999
Specificity	0.999	0.999	0.999	1.000	1.000
Pos Pred Value	0.998	0.994	0.995	0.999	0.998
Neg Pred Value	1.000	0.999	0.999	0.999	1.000
Prevalence	0.284	0.194	0.174	0.164	0.184
Detection Rate	0.284	0.193	0.173	0.163	0.184
Detection Prevalence	0.285	0.194	0.174	0.163	0.184
Balanced Accuracy	0.999	0.998	0.997	0.997	0.999

```
pred1 <- predict(mod1, crossv_test)
accuracy <- sum(pred1 == crossv_test$classe) / length(pred1)
varImpRF <- train(classe ~ ., data = training, method = "rf")
varImpObj <- varImp(varImpRF)
plot(varImpObj, main = "Importance of Top 40 Variables", top = 40)
```



plot(varImpObj, main = "Importance of Top 25 Variable s", top = 25)


Importance of Top 25 Variables
num_window
roll_belt
pitch_forearm
yaw_belt
magnet_dumbbell_y
magnet_dumbbell_z
pitch_belt
roll_forearm
accel_dumbbell_y
roll_dumbbell
accel_forearm_x
total_accel_dumbbell
magnet_dumbbell_x
accel_belt_z
magnet_belt_y
accel_dumbbell_z
magnet_forearm_z
magnet_belt_z
magnet_belt_x
yaw_dumbbell
gyros_belt_z
roll_arm
accel_forearm_z
gyros_dumbbell_y
accel_arm_x
0
20
40
60
80
100
Importance