practical_machine_learning_final_project.R

lukas

Wed Oct 04 18:44:46 2017

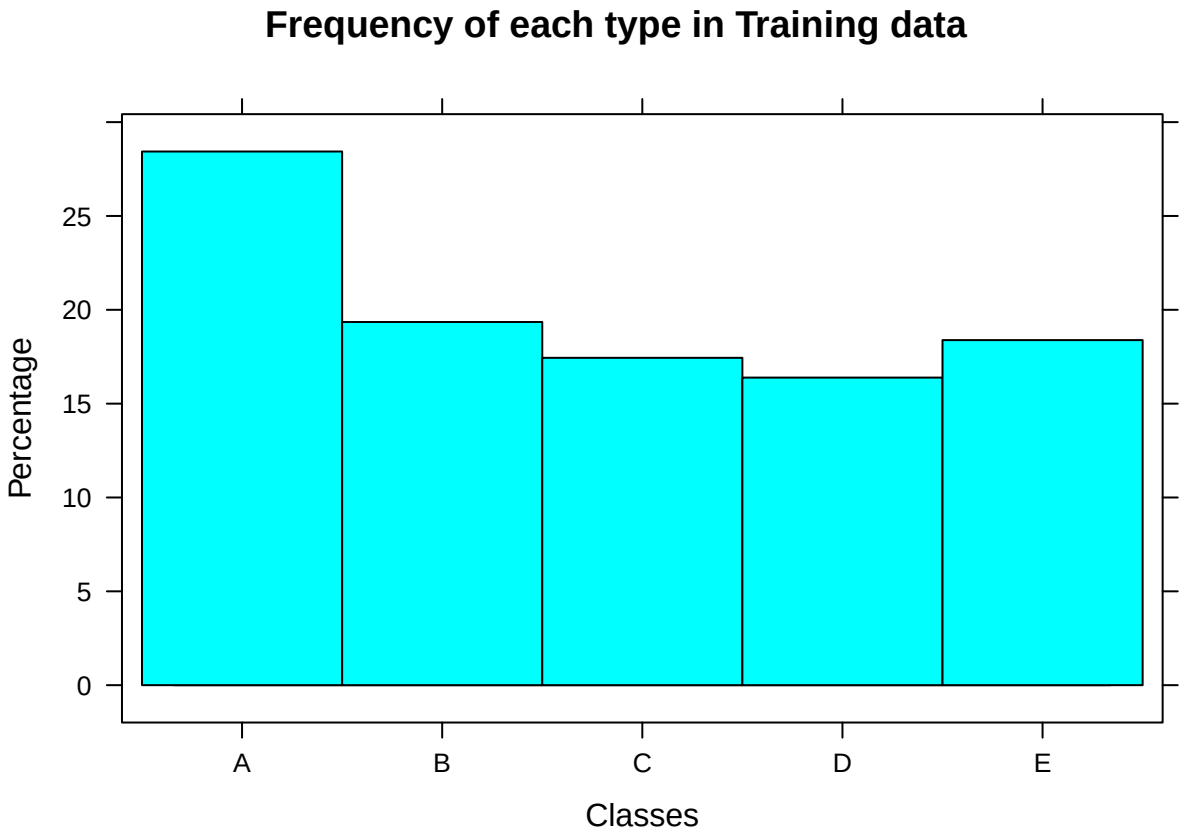
```
## Practical Machine Learning Project

## import data
pml_training <- read.csv("C:/Users/lukas/Downloads/pml-training.csv",header = T)
pml_testing <- read.csv("C:/Users/lukas/Downloads/pml-testing.csv",header = T)
#head(pml_training)

pml_training <- read.csv("C:/Users/lukas/Downloads/pml-training.csv",header = T,na.strings=c("NA",""))
pml_testing <- read.csv("C:/Users/lukas/Downloads/pml-testing.csv",header = T,na.strings=c("NA",""))
#head(pml_training)

pml_training <- pml_training[,colSums(is.na(pml_training)) == 0]
pml_testing <- pml_testing[,colSums(is.na(pml_testing)) == 0]
#head(pml_training)

## In this project, your goal will be to use data from accelerometers
## on the belt, forearm, arm, and dumbell of 6 participants. Thus we delete the
## unrelated variables.
pml_training <- pml_training[,-c(1:7)]
pml_testing <- pml_testing[,-c(1:7)]

## to do cross-validation
library(caret)
```

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

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

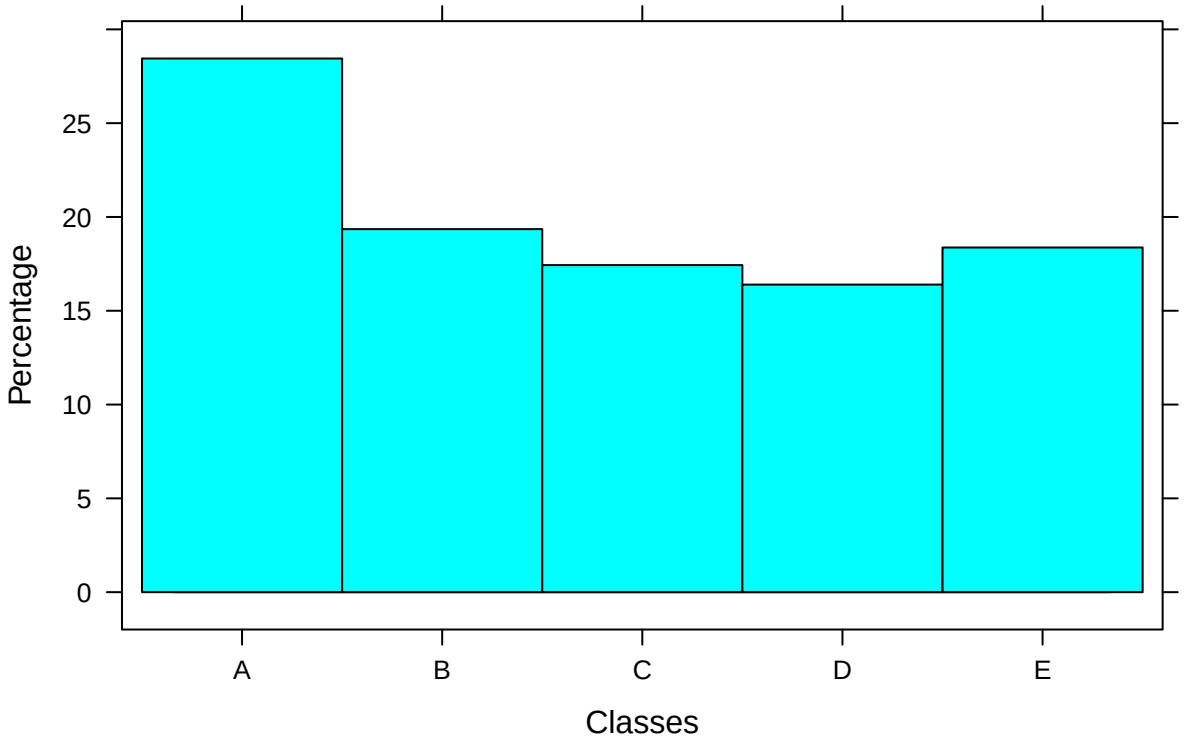
```
set.seed(1001)
inTrain <- createDataPartition(y=pml_training$classe, p=3/4, list=FALSE)
pml_training_train <- pml_training[inTrain,]
pml_training_test <- pml_training[-inTrain,]

histogram(pml_training_train$classe,pml_training_train,xlab = "Classes",
                ylab="Percentage",main="Frequency of each type in Training data")
```

```
## Warning in histogram.factor(pml_training_train$classe,
## pml_training_train, : explicit 'data' specification ignored
```




```
histogram(pml_training_test$classe,pml_training_test,xlab = "Classes",
                ylab="Percentage",main="Frequency of each type in Testing data")
```

```
## Warning in histogram.factor(pml_training_test$classe, pml_training_test, :
## explicit 'data' specification ignored
```




```
## rpart: Regressive Partitioning and Regression trees
library(rpart)
#install.packages("RGtk2") if necessary
#library(RGtk2)
#library(rattle)
modFit <- rpart(classe ~ .,data=pml_training_train,method="class")
#fancyRpartPlot(modFit)
train_pred <- predict(modFit,pml_training_test,type = "class")
confusionMatrix(train_pred,pml_training_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1269  142   20   33   21
##          B   49  590   92   87   90
##          C   34  124  699  116  122
##          D   17   67   44  509   53
##          E   26   26    0   59  615
##
## Overall Statistics
##
##                Accuracy : 0.7508
##                  95% CI : (0.7385, 0.7629)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.6841
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9097   0.6217   0.8175   0.6331   0.6826
## Specificity            0.9384   0.9196   0.9022   0.9559   0.9723
## Pos Pred Value         0.8545   0.6498   0.6384   0.7377   0.8471
## Neg Pred Value         0.9631   0.9102   0.9590   0.9300   0.9315
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2588   0.1203   0.1425   0.1038   0.1254
## Detection Prevalence   0.3028   0.1852   0.2233   0.1407   0.1480
## Balanced Accuracy      0.9241   0.7707   0.8599   0.7945   0.8274
```

```
## Now we try random forest.
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
modFit <- randomForest(classe ~., data=pml_training_train, method="class")
train_pred <- predict(modFit,newdata = pml_training_test,type = "class")
confusionMatrix(train_pred,pml_training_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    5    0    0    0
##          B    0  943    5    0    0
##          C    0    1  850    6    4
##          D    0    0    0  798    3
##          E    1    0    0    0  894
##
## Overall Statistics
##
##                Accuracy : 0.9949
##                  95% CI : (0.9925, 0.9967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9936
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
```

```
## Sensitivity            0.9993   0.9937   0.9942   0.9925   0.9922
## Specificity            0.9986   0.9987   0.9973   0.9993   0.9998
## Pos Pred Value         0.9964   0.9947   0.9872   0.9963   0.9989
## Neg Pred Value         0.9997   0.9985   0.9988   0.9985   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1923   0.1733   0.1627   0.1823
## Detection Prevalence   0.2853   0.1933   0.1756   0.1633   0.1825
## Balanced Accuracy      0.9989   0.9962   0.9957   0.9959   0.9960
```

```
## Finally, we predict the 20 predicting tests.
test_pred <- predict(modFit,newdata = pml_testing)
test_pred
```

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