coursera_ML

monique-97

22 02 2021

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

data source: 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.

aim of the project

The goal of your project is to predict the manner of exercises, this is the "classe" variable in the training set.

read libraries

```
library(caret)
```

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

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

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

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

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

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

```
library(rpart.plot)
```

```
## Loading required package: rpart
```

read data

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

We have 19622 observations and 160 variables in train data, and 20 observations and 160 variables in test data.

data pre-processing

remove NAs

```
train_data <- train_data[, colSums(is.na(train_data)) == 0]
test_data <- test_data[, colSums(is.na(test_data)) == 0]
```

take only columns of interest

```
training <- train_data[,-c(1:7)]
testing <-test_data[,-c(1:7)]
```

change values into numeric

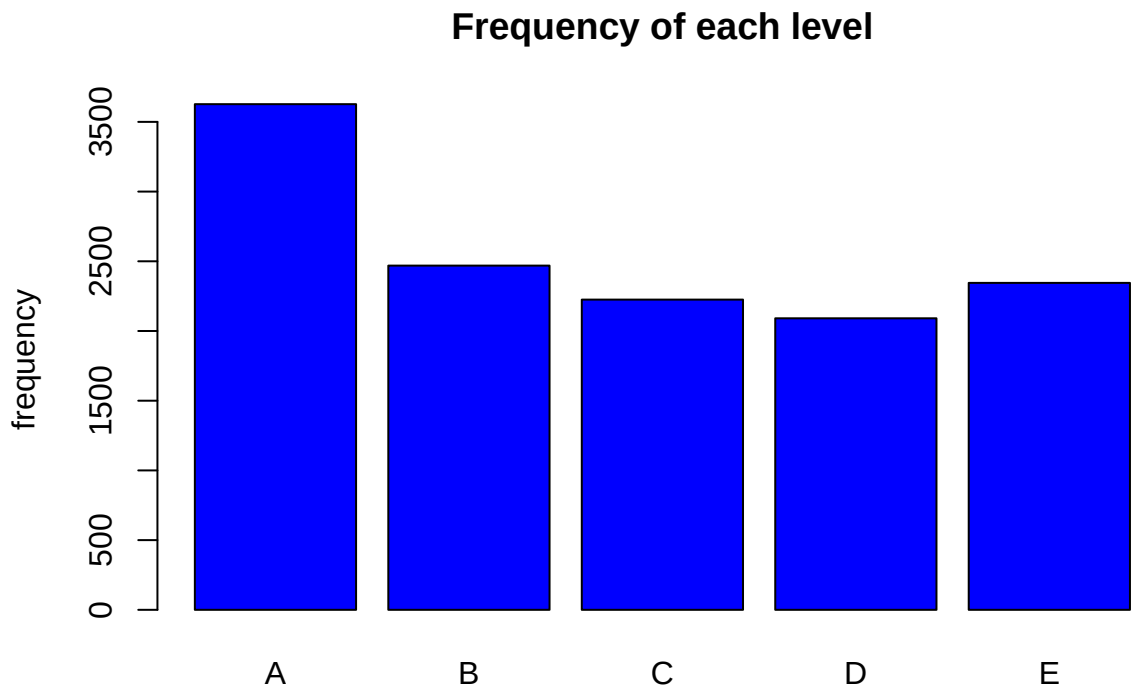
```
classe <- train_data$classe
train_clean <- train_data[, sapply(train_data, is.numeric)]
train_clean$classe <- classe
test_clean <- test_data[, sapply(test_data, is.numeric)]
```

split training data into a training dataset and a data for validation, set seed to get reproducible results

```
set.seed(199997)
training_part <- createDataPartition(train_clean$classe, p=0.65, list=FALSE)
train_values <- train_clean[training_part, ]
test_values <- train_clean[-training_part, ]
```

data visualization: frequency of each level

```
barplot(table(train_values$classe), col="blue", main="Frequency of each level", xlab=" ", ylab="frequen
```



Level A is the most frequent.

builting the model

Random forest prediction model will be applied.

run random forest

```
controltr <- trainControl(method="cv", 5)
model_random <- train(classe ~ ., data=train_values, method="rf", trControl=controltr, ntree=100)
model_random
```

```
## Random Forest
##
## 12757 samples
##    56 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10205, 10206, 10207, 10204, 10206
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9978836  0.997323
##   29    1.0000000  1.000000
##   56    1.0000000  1.000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 29.
```

cross validation

estimate model performance

```
prediction <- predict(model_random, test_values)
confusionMatrix(test_values$classe, prediction)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1952    1    0    0    0
##          B    0 1328    0    0    0
##          C    0    0 1197    0    0
##          D    0    0    1 1124    0
##          E    0    0    0    0 1262
##
## Overall Statistics
##
##                Accuracy : 0.9997
##                  95% CI : (0.9989, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                   Kappa : 0.9996
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9992   0.9992   1.0000   1.0000
## Specificity            0.9998   1.0000   1.0000   0.9998   1.0000
## Pos Pred Value         0.9995   1.0000   1.0000   0.9991   1.0000
## Neg Pred Value         1.0000   0.9998   0.9998   1.0000   1.0000
## Prevalence             0.2843   0.1936   0.1745   0.1637   0.1838
## Detection Rate         0.2843   0.1934   0.1744   0.1637   0.1838
## Detection Prevalence   0.2845   0.1934   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9999   0.9996   0.9996   0.9999   1.0000
```

out of sample error

calculate the expected out of sample error and accuracy

```
error <- 1 - as.numeric(confusionMatrix(test_values$classe, prediction)$overall[1])
acc <- postResample(prediction, test_values$classe)
```

Out-of-sample error is 0.015%.

apply model to test dataset

```
final_result <- predict(model_random, test_clean[, -length(names(test_clean))])
```

Machine learning algorithm was applied to the 20 test cases available in the test data.

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har.