Course Project: Practical Machine Learning

Mateopr

8/3/2020

Summary

The aim of the project is to create a model to predict the classe variable of the testing data set. The model was developed using the Random Forest method and pre processing the training data set. The model was applied on the testing data set obtaining all the answers right in the Prediction Quiz.

Data pre-processing

The first step is to load the files and the libraries. Then, I pre-processed the data transforming the characters into factor and removing those variables that are empty or almost empty. By doing that, the number of variables was reduced from 196 to 53.

```
library(caret)
```

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

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
training <- read.csv("pml-training.csv")
testing <- read.csv("pml-testing.csv")
training <- training %>% mutate_if(is.character,as.factor)
testing <- testing %>% mutate_if(is.character,as.factor)
training <- training[,colSums(is.na(training))==0]
training <- training[!sapply(training, function(x) all(x==""))]
training <- training[, -c(nearZeroVar(training), 1:7),]
```

```
inTrain = createDataPartition(training$classe, p = 3/4)[[1]]

training2 = training[ inTrain,]

testing2 = training[-inTrain,]
```

Model selection

The number of variables was high, so I did not do any plotting before the modeling. The method that I used was Random Forest because it can handle binary features, categorical features, and numerical features. The main drawback is that is like a black box and it is not easy to interpret the model, but it has high accuracy for this data set.

```
model<- train(classe ~ ., data=training2, method="rf",trControl=trainControl(method="none"), tuneGrid=da
model$finalmodel
```

```
## NULL
```

Cross-validation

- Divide the data into training and testing: The raw data was already divided by training and testing. But, the testing is just 20 samples for the quiz. So, I splited the data into training2 (75%) and testing2(25%) to evaluate the model.
- Build a model on the training set.
- Evaluate on the test set (testing2).

Results

The model obtained was tested in the testing2 set obtaining a 99.61% accuracy. I did the Project Quiz and I answered right the 20 questions.

```
pretrain <- predict(model, testing2)
confusionMatrix(testing2$classe,pretrain)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    2    0    0    0
##          B    8  937    4    0    0
##          C    0    6  849    0    0
##          D    0    0    7  797    0
##          E    0    0    0    3  898
##
## Overall Statistics
##
##                Accuracy : 0.9939
##                  95% CI : (0.9913, 0.9959)
##     No Information Rate : 0.2857
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9923
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9943   0.9915   0.9872   0.9962   1.0000
## Specificity            0.9994   0.9970   0.9985   0.9983   0.9993
## Pos Pred Value         0.9986   0.9874   0.9930   0.9913   0.9967
## Neg Pred Value         0.9977   0.9980   0.9973   0.9993   1.0000
## Prevalence             0.2857   0.1927   0.1754   0.1631   0.1831
## Detection Rate         0.2841   0.1911   0.1731   0.1625   0.1831
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9969   0.9943   0.9929   0.9973   0.9996
```

```
pretest <- predict(model, testing)
pretest
```

```
##  [1] 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
```