Human Activity: Machine Learning

Johannes Tynes

May 5th 2017

Summary

This is a machine learning modeling exercised aimed at developing a predictive ML model for human activity identification.

The data used by the model encompasses over 19k training examples with physical characteristics of a training exercise, as well as data on 5 different ways in which the exercise can be done.

The purpose of the model was to predict in which of the 5 ways the exercise was done.

A gradient boosting machine was developd for classification based on data with near-zero variability variables omitted along with variables with a significant missing-data percentage. The model was developed using 4-fold randomly sampled cross validation and achieved an accuracy of around 98.2%.

Loading Data

```
# Data is read from CSV files using `readr::read_csv`:
read <- function(fp) {
  read_csv(fp) %>%
  mutate_each_(funs(factor),
    vars=intersect(colnames(.), c("user_name", "new_window", "classe"))) %>%
  mutate_if("is.character", "as.numeric") %>%
  select(-1)
}

# Loading the training and testing data into `d` variable:
  list(test="pml-testing.csv", train="pml-training.csv") %>%
  llply(read) ->
d
```

Data PreProcessing

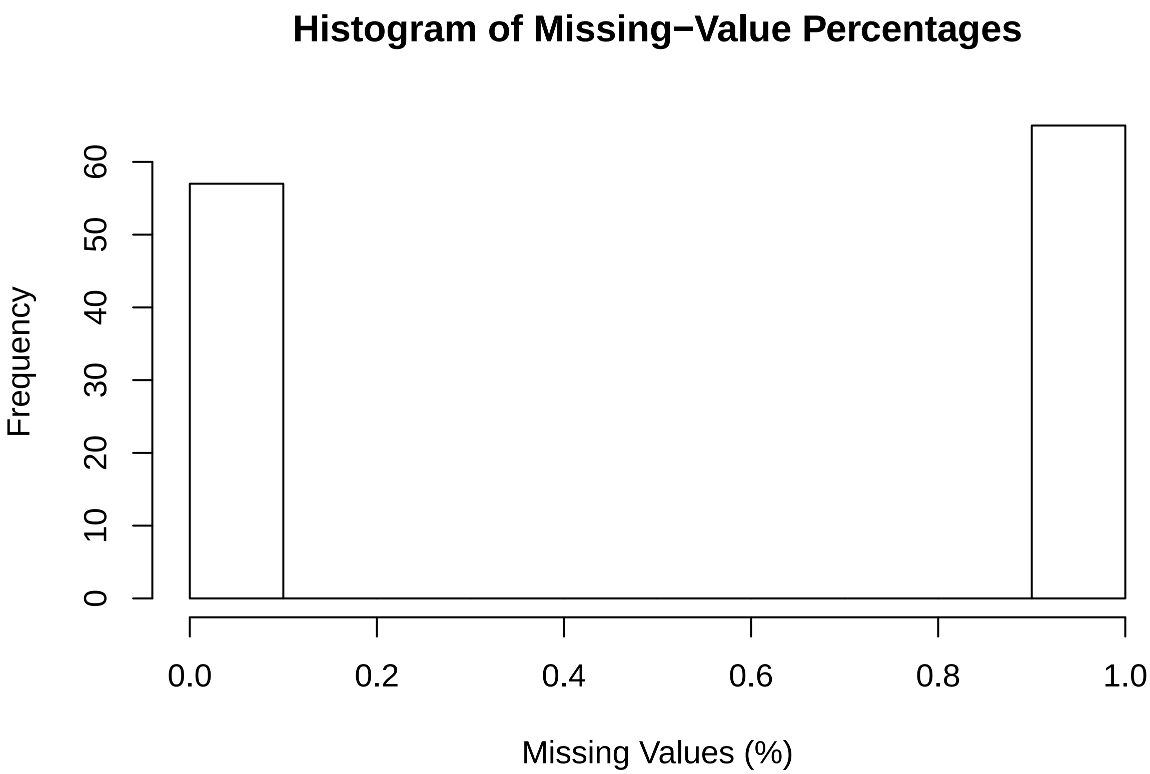
The first step taken in data pre-processing has been to eliminate potential explanatory variables with very low variability and consequently limited potential to add predictive value to the model.

In order to identify near-zero-variability variables the `caret::nearZeroVar` function was used:

```
d[["train"]] %<>% {.[,-nearZeroVar(.)]}
#f <- b %>% {.[,-nearZeroVar(.)]}
#g <- f[, f %>% apply(2, . %>% is.na %>% mean) %>% `<=`(.3)]
```

As a second step in data pre-processing was investigating whether there were any variables with large numbers of missing values. The figure below shows that the distribution of missing value percentages across variables.

```r
d[["train"]] %>% apply(2, . %>% is.na %>% mean) %>%
hist(main="Histogram of Missing-Value Percentages", xlab="Missing Values (%)")
```



Based on this it can be seen that: * A majority of variables suffer from a very high missing-values percentage. * Nearly half of the variables have very few missing value.

It was decided to omit variables with over 30% missing values.

```r
d[["train"]] %<>% (function(i) {
  i[, apply(i, 2, . %>% is.na %>% mean) %>% `<=`(.3)]})
```

As a final pre-processing step, variables with unlikely predictive power were omitted (user-name, time stamps):

```r
d[["train"]] %<>% select(-user_name, -matches("_timestamp_"))
```

Training the Predictive Model

The predictive model was fitted and assessed using a gradient boosting machine because it is a versatile algorithm for classification that can fit a wide range of classification problems and, in the author's experience, is usually at least comparable in performance terms to tuned random forest and support vector machine implementations.

```r
# Setting seed for reproducibility:
set.seed(1)
# Training the model
  train(
    classe ~ .,
```

```
    data=d[["train"]] %>% sample_n(10000),
    method="gbm",
    trControl=trainControl(method="cv", number=4),
    verbose=FALSE,
    na.action=na.pass) ->
m
```

The model was trained: * using the `caret::train` function as a wrapper around `gbm`. * with validation through a 4-fold cross validation on 10k-observation samples (for reasons of computational speed)

Model Performance

The tuned GBM model has an achieved accuracy of 0.7605968, 0.8788989, 0.9285002, 0.8307989, 0.9366001, 0.9672999, 0.8684992, 0.9579, 0.9823995 using an interaction depth of 3 and 150 trees.

Applying the Model to Predict Test Cases

```
p <- predict(m, newdata=d[["test"]])
write.csv(cbind(d[["test"]], classe=p), file="ml_predictions.csv")
```