Practical Machine Learning Final Project

Jeff B

4/8/2020

Summary

We trained a Random Forest model to predict how well a weight-lifting exercise was performed, based on data collected by wearables. The model used forty nine predictors. The model was cross-validated using five folds. The accuracy of the model, per a confusion matrix, is 0.9942.

How you built your model

First we set up the environment. The training and testing sets were already provided, so no data partitioning was necessary. This analysis uses the *caret* package.

```
# Download files
file1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv" # Training data
file2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv" # Testing data

if (!file.exists("pml-training.csv")) {download.file(file1, destfile = "./pml-training.csv")}
if (!file.exists("pml-testing.csv")) {download.file(file2, destfile = "./pml-testing.csv")}

# Load files into environment
training <- data.frame(read.csv("pml-training.csv"))
testing <- data.frame(read.csv("pml-testing.csv"))
```

The training and testing sets required some tidying. Many variables were removed. Statistical variables, such as averages and standard deviations, were removed so that only the raw underlying measures remained. This also removed all the N/A observations. Variables related to timestamps and rownames were also removed, because they were deemed irrelevant. This reduced the number of variables from 160 to 50.

```
# Remove summary statistics, timestamps, and windows from data
omitvars <- grep("avg|max|min|total|var|stddev|skewness|kurtosis|amplitude|timestamp|window",
     names(training), value = TRUE)
training <- select(training, !omitvars & !X)
```

```
## Note: Using an external vector in selections is ambiguous.
## ℹ Use `all_of(omitvars)` instead of `omitvars` to silence this message.
## ℹ See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
testing <- select(testing, !omitvars & !X)

# Convert relevant data to numeric class
training[,2:49] <- sapply(training[,2:49], as.numeric)
testing[,2:49] <- sapply(testing[,2:49], as.numeric)
```

Once the data were tidy, we chose to build a random forest model. Code is below.

How you used cross validation

Basic five-fold cross-validation was used as a training control. Doing dramatically reduced the runtime of the model and allowed us to predict the model's performance despite not having a test set with outcomes listed.

```
set.seed(1) # Set seed

# Configure resampling, 5-fold cross-validation
trControl <- trainControl(method = "cv", number = 5, allowParallel = TRUE)

# Run training model
rf <- train(classe ~ ., method="rf", data=training, trControl=trControl)
```

What you think the expected out of sample error is

A confusion matrix estimates accuracy of 0.9942. OOB estimate of error rate is 0.45% We expect the out of sample error to be larger than that of the training model, though not significantly so. We expect the model to predict ~89% of the test cases correctly, calculated as 0.9942^20.

```
# Assess model
rf
```

```
## Random Forest
##
## 19622 samples
##    49 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 15697, 15699, 15698, 15697, 15697
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9934765  0.9917472
##   27    0.9942412  0.9927151
##   53    0.9887881  0.9858154
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
rf$resample
```

```
##    Accuracy     Kappa Resample
## 1 0.9951592 0.9938765    Fold1
## 2 0.9943935 0.9929077    Fold3
## 3 0.9946470 0.9932280    Fold2
## 4 0.9933758 0.9916206    Fold5
## 5 0.9936306 0.9919429    Fold4
```

```
rf$finalModel
```

```
## 
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
## 
##         OOB estimate of  error rate: 0.45%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5575    3    1    0    1 0.0008960573
## B   19 3774    4    0    0 0.0060574137
## C    0    9 3402   11    0 0.0058445354
## D    0    1   26 3188    1 0.0087064677
## E    0    1    4    8 3594 0.0036041031
```

```
confusionMatrix.train(rf) # 0.9942 accuracy
```

```
## Cross-Validated (5 fold) Confusion Matrix 
## 
## (entries are percentual average cell counts across resamples)
##  
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.1  0.0  0.0  0.0
##          B  0.0 19.2  0.1  0.0  0.0
##          C  0.0  0.0 17.3  0.2  0.0
##          D  0.0  0.0  0.1 16.2  0.0
##          E  0.0  0.0  0.0  0.0 18.3
##                             
##  Accuracy (average) : 0.9942
```

Why you made the choices you did

The Random forest model type was chosen due to its reputation for predictive strength. It seemed especially relevant for a multi-factor outcome.

Five-fold cross-validation was used to provide an assessment of predictive performance and to limit the runtime while ensuring all data was used in the training.

No automated preprocessing was used, only the aforementioned manual variable selection and tidying.

Use your prediction model to predict 20 different test cases.

Predictions based on the testing data are provided below. Real performance will be assessed in the online quiz.

```
pred <- predict(rf, testing)
pred
```

```
##  [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
```