Practical Machine Learning Project

Jon Brophy

7/18/2020

Overview

"Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it."

The data set will consist of a training and test set, and will be used to predict the "classe" variable from the training set.

Load and Check Data

```r
library(caret)
```

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

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

```r
Train <- read.csv("pml-training.csv")
Test <- read.csv("pml-testing.csv")
NA_List_Train <- sapply(Train, function(x) sum(is.na(x))/length(x))
NA_List_Train <- NA_List_Train[NA_List_Train > 0 ]
NA_List_Test <- sapply(Test, function(x) sum(is.na(x))/length(x))
NA_List_Test <- NA_List_Test[NA_List_Test > 0 ]
```

Missing 98% of data in a number of fields in Train, and 100% in a number of fields in Test. I will exclude those fields from training and test. Additionally, first 7 fields are not related to the rest of the data so will also remove.

```r
Train <- Train[, -which(names(Train) %in% names(NA_List_Train))]
Train <- Train[, -which(names(Train) %in% names(NA_List_Test))]
Test <- Test[, -which(names(Test) %in% names(NA_List_Train))]
Test <- Test[, -which(names(Test) %in% names(NA_List_Test))]
Train <- Train[, -c(1:7)]
Test <- Test[, -c(1:7)]
```

Check Train for highly correlated variables, excluding classe:

```r
corVar <- cor(Train[, -53])
HighCor <- findCorrelation(corVar, .9)
HighCor <- names(Train)[HighCor]
Train <- Train[, -which(names(Train) %in% HighCor)]
Test <- Test[, -which(names(Test) %in% HighCor)]
```

Build Models

For this project I will try several different algorithms, including: Random Forest Gradient Boosted Machine (GBM) Support Vector Machine (SVM) I will use 5 fold cross validation to determine which model performs best and will be selected to be used on the test data.

```r
control <- trainControl(method = "cv", number = 5, allowParallel = T)
set.seed(100)
rf_model <- train(classe ~., data = Train, method = "rf", trControl = control); saveRDS(rf_model, "rf_m
set.seed(100)
gbm_model <- train(classe ~., data = Train, method = "gbm", trControl = control, verbose = F); saveRDS(
set.seed(100)
svm_model <- train(classe ~., data = Train, method = "svmRadial", trControl = control); saveRDS(svm_mod
```

```r
rf_model <-readRDS("rf_model.RDS"); gbm_model <- readRDS("gbm_model.RDS")
svm_model <-readRDS("svm_model.RDS")
results <- resamples(list(RF=rf_model, GBM=gbm_model, SVM=svm_model))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: RF, GBM, SVM
## Number of resamples: 5
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## RF  0.9920979 0.9938854 0.9946497 0.9945467 0.9949032 0.9971975    0
## GBM 0.9559011 0.9600000 0.9602548 0.9595859 0.9605096 0.9612640    0
## SVM 0.9270966 0.9273885 0.9314475 0.9318107 0.9342675 0.9388535    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## RF  0.9900032 0.9922649 0.9932329 0.9931017 0.9935522 0.9964552    0
## GBM 0.9442027 0.9493875 0.9497146 0.9488722 0.9500636 0.9509924    0
## SVM 0.9075844 0.9079745 0.9130823 0.9135731 0.9167005 0.9225236    0
```

Accuracy is highest with the RF model at over 99% (less than 1% expected out of sample error), so that will be chosen for the test data.

```r
TestPreds <- predict(rf_model, newdata=Test)
TestPreds
```

```
##  [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
```