Machine Learning: Prediction Model Documentation

MD Alamgir Ph.D.

January 06, 2017

Synopsis

This Machine Learning project uses data from accelerometers on the belt, forearm, arm, and dumbell of six research study participants. Problem description and data sources are listed Appendix C.

- Training data (19622 observations) consist of accelerometer data and a label(variable=classe) identifying the type of the activity of the participant.
- The project goal is to predict the identifying labels(variable=classe) for the test data using machine learning models. Algorithms with multi-class prediction capability are needed since there are five classes (labels=A, B, C, D, and E) in the “classe” variable.
- Testing data consists of twenty observations of accelerometer data, without the identifying label.
- The training data cleaning and preparation is described below. The cleaned training data was partitioned into a training set (75% of original training data), and a validation set(25% of original training data). Basic Exploratory Analysis is covered in Appendix B.
- Two supervised machine learning models were trained: Random forest(rf), and Linear Discriminant Analysis(lda). Both rf and lda are appropriate for multiclass classification predictive modeling, and are appropriate candidates for this project. The lda model study is in Appendix A. The Random Forest model was found to have quite high accuracy of 99.7% while the lda model showed much lower accuracy of ~72%. Therefore, the Random Forest Model was used to predict variable “classe” of the test dataset (Table 1).

Data Loading and Prepration

- Exploratory Analysis is covered in Appendix B
- Data loading, cleaning, and data partitioning covered below.

```
#set Working Directory
setwd("C:/Users/MD/Documents/! Coursera/jhds/Week4_Machine_Learning/project")
library(ggplot2)
library(caret)
library(dplyr)
# LOad data
Training <- read.csv("pml-training.csv")
Data_test <- read.csv("pml-testing.csv")

set.seed(32987)

# Reduce variables(features) to fit, as indicated in )items 1, 2, and 3 below

# Item 1: Remove variables (columns in dataset) with nearly zero variance using the nearZeroVar
 function
nZVar <- nearZeroVar(Training)
Data_Training1 <- Training[, -nZVar]

# Item 2: Remove variables(columns in data set) that have > 95% "NA" entries
NA95pct <- sapply(Data_Training1, function(x) mean(is.na(x))) > 0.95
Data_Training2 <- Data_Training1[, NA95pct==F]

#Item 3: Remove first five columns as these are not relevant for prediction of "classe" labels.
Data_Training3 <- Data_Training2[, -(1:5)]


# Split training data into two sets: Data_Training (75%), and Data_validation (25%).

inTrain <- createDataPartition(y=Data_Training3$classe, p=0.75, list=F)
Data_Training <- Data_Training3[inTrain, ]
Data_validation <- Data_Training3[-inTrain, ]


# Models Fitted: Multiple models are fitted to Data_Training. Then prediction accuracy is determ
ined using the Data_validation dataset. The model with highest accuracy is applied to the test d
ata set. Random Forest model, to see if it would have acceptable performance. I fit the model on
 Data_Training1, and instruct the "train" function to use 3-fold cross-validation to select opti
mal tuning parameters for the model.

# Cross Validation: Use 3-fold cross-validation (CV) to select optimal tuning parameters.

fitControl <- trainControl(method="cv", number=3, verboseIter=F)

# Fit model on Data_Training
fit <- train(classe ~ ., data=Data_Training, method="rf", trControl=fitControl)

fit$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: 0.18%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4184    0    0    0    1 0.0002389486
## B    4 2842    1    1    0 0.0021067416
## C    0    4 2562    1    0 0.0019477990
## D    0    0    6 2406    0 0.0024875622
## E    0    0    0    9 2697 0.0033259424
```

```
# Model Evaluation: Use model to predict classe in validation set (Data_validation)
prediction <- predict(fit, newdata=Data_validation)

# Use confusion matrix to get estimate of out-of-sample error
confusionMatrix(Data_validation$classe, prediction)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    1    0    0    0
##          B    4  944    1    0    0
##          C    0    1  854    0    0
##          D    0    0    2  801    1
##          E    0    0    0    3  898
##
## Overall Statistics
##
##                Accuracy : 0.9973
##                  95% CI : (0.9955, 0.9986)
##     No Information Rate : 0.2851
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9966
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9971   0.9979   0.9965   0.9963   0.9989
## Specificity            0.9997   0.9987   0.9998   0.9993   0.9993
## Pos Pred Value         0.9993   0.9947   0.9988   0.9963   0.9967
## Neg Pred Value         0.9989   0.9995   0.9993   0.9993   0.9998
## Prevalence             0.2851   0.1929   0.1748   0.1639   0.1833
## Detection Rate         0.2843   0.1925   0.1741   0.1633   0.1831
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9984   0.9983   0.9981   0.9978   0.9991
```

```
# The accuracy of the Random Forest model in predicting the validation data set is seen to be ve
ry high (0.9973) or 99.7%.

# Test Set Predictions: Use the model fitted on Data_Training to predict the label for the obser
vations in Data_test, and output the predictions.

# Predict on test set
prediction <- predict(fit, newdata=Data_test)

# Convert predictions to character vector
prediction <- as.character(prediction)

# Output the predicted results
predresult<-data.frame(testdata_ID=Data_test$problem_id, Predicted_Classe=prediction)

# TABLE 1: Prediction Results

predresult
```

```
##    testdata_ID Predicted_Classe
## 1            1                B
## 2            2                A
## 3            3                B
## 4            4                A
## 5            5                A
## 6            6                E
## 7            7                D
## 8            8                B
## 9            9                A
## 10          10                A
## 11          11                B
## 12          12                C
## 13          13                B
## 14          14                A
## 15          15                E
## 16          16                E
## 17          17                A
## 18          18                B
## 19          19                B
## 20          20                B
```

Conclusions

1. Two supervised machine learning models have been used in this project: Random Forest (rf), and Linear Discriminant Analysis (lda). The Random Forest model yielded very high accuracy ~ 0.997 when the trained model was applied to the validation dataset. The lda model yielded lower accuracy, ~ 0.72 (Appendix A). The Random Forest Model is recommended and used for this classification prediction study.
2. The prediction of the labels for the “classe” variable for the twenty observations in the test dataset is provided in TABLE 1.

APPENDICES

Appendix A - Fitting Linear Discriminant Analysis (“lda”) Model

```
library(ggplot2)
library(caret)
```

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

```
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
```

```
# Use 3-fold cross-validation (CV) to select optimal tuning parameters
fitControl_lda <- trainControl(method="cv", number=3, verboseIter=F)

# Fit model on Data_Training
fitlda <- train(classe ~ ., data=Data_Training, method="lda", trControl=fitControl_lda)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Use model to predict classe in validation set (Data_validation)
prediction_ldav <- predict(fitlda, newdata=Data_validation)

# Evaluate confusion matrix to get estimate of out-of-sample error
confusionMatrix(Data_validation$classe, prediction_ldav)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1148   47   86  108    6
##          B  121  624  119   41   44
##          C   69   94  553  104   35
##          D   38   32   97  612   25
##          E   33   99   80   97  592
##
## Overall Statistics
##
##                Accuracy : 0.7196
##                  95% CI : (0.7068, 0.7322)
##     No Information Rate : 0.2873
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6456
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8148   0.6964   0.5914   0.6362   0.8433
## Specificity            0.9293   0.9189   0.9239   0.9513   0.9265
## Pos Pred Value         0.8229   0.6575   0.6468   0.7612   0.6570
## Neg Pred Value         0.9256   0.9312   0.9057   0.9146   0.9725
## Prevalence             0.2873   0.1827   0.1907   0.1962   0.1431
## Detection Rate         0.2341   0.1272   0.1128   0.1248   0.1207
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.8720   0.8077   0.7577   0.7937   0.8849
```

```
# The accuracy is ~ 72%, less than Random Forest("rf") model
# Test Set Predictions are made below. [Note: these presictions will not be considered since the
 accuracy of the Linear Discriminant Analysis model is less than that of the Random Forest mode
l. Predictions using the "rf" model reported in Table 1 earlier will be considered as more accur
ate output of this project.
# Use the model fitted on Data_Training to predict the label for the observations in Data_test,
 and output the predictions.

# prediction of test set using lda model
prediction_lda <- predict(fitlda, newdata=Data_test)

# convert predictions to character vector
prediction_lda <- as.character(prediction_lda)

# Output the predicted results for the lda model
predresult_lda<-data.frame(testdata_ID=Data_test$problem_id, Predicted_Classe=prediction_lda)
predresult_lda
```

```
##    testdata_ID Predicted_Classe
## 1            1                B
## 2            2                A
## 3            3                B
## 4            4                A
## 5            5                A
## 6            6                E
## 7            7                D
## 8            8                D
## 9            9                A
## 10          10                A
## 11          11                D
## 12          12                A
## 13          13                E
## 14          14                A
## 15          15                B
## 16          16                A
## 17          17                A
## 18          18                B
## 19          19                B
## 20          20                B
```

Appendix B - Exploratory Analysis

```
#Number of Rows and columns:

# Before data cleaning:
dim(Training)
```

```
## [1] 19622   160
```

```
# After data cleaning have fewer columns (54 vs. 160). # Many columns have mostly "NA"s. Also nu
mber of rows is ~75% of original after partitioning 25% to the validation data set.
dim(Data_Training)
```

```
## [1] 14718    54
```

```
# Columns in training set after data cleaning, and used for training using "rf" and "lda" models
colnames(Data_Training)
```

```
##  [1] "num_window"           "roll_belt"            "pitch_belt"
##  [4] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
##  [7] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [10] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [13] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [16] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [19] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [22] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [25] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [28] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [31] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [34] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [37] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [40] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [43] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [46] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [49] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [52] "magnet_forearm_y"     "magnet_forearm_z"     "classe"
```

```
# Class Labels in training dataset (factors in classe)
levels(Data_Training$classe)
```

```
## [1] "A" "B" "C" "D" "E"
```

Appendix C - Project Description, Data Sources, and Requirements

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. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

*The training data for this project are available here:

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

*The test data are available here:

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

*The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

*The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set.

[End of Report]