Practical Machine Learning

Course Project

true

29 setembro 2020

Overview

Background

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.

The data for this project come from this source: More information is available from the website here: http://groupware.les.inf.puc-rio.br/har.

The goal of this project is to **predict the manner in which they did the exercise**.

Load Libraries and Data Download

```
##
## 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

## Loading required package: lattice

## Loading required package: ggplot2

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```r
urlTrain <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
urlTest <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"


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

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

DataTrain <- read.csv("./pml-training.csv")
DataTest <- read.csv("./pml-testing.csv")
```

Exploratory Data Analysis and Data Cleaning

Before start feature preprocessing, let's observe the descriptive statistics of each column in the training dataset.

```r
dim(DataTrain)
```

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

```r
dim(DataTest)
```

```
## [1]  20 160
```

```r
glimpse(DataTrain)
```

```
## Rows: 19,622
## Columns: 160
## $ X                        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...
## $ user_name                <chr> "carlitos", "carlitos", "carlitos", "carli...
## $ raw_timestamp_part_1     <int> 1323084231, 1323084231, 1323084231, 132308...
## $ raw_timestamp_part_2     <int> 788290, 808298, 820366, 120339, 196328, 30...
## $ cvtd_timestamp           <chr> "05/12/2011 11:23", "05/12/2011 11:23", "0...
## $ new_window               <chr> "no", "no", "no", "no", "no", "no", "no", ...
## $ num_window               <int> 11, 11, 11, 12, 12, 12, 12, 12, 12, 12, 12...
## $ roll_belt                <dbl> 1.41, 1.41, 1.42, 1.48, 1.48, 1.45, 1.42, ...
## $ pitch_belt               <dbl> 8.07, 8.07, 8.07, 8.05, 8.07, 8.06, 8.09, ...
## $ yaw_belt                 <dbl> -94.4, -94.4, -94.4, -94.4, -94.4, -94.4, ...
## $ total_accel_belt         <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ...
## $ kurtosis_roll_belt       <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_picth_belt      <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_yaw_belt        <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_roll_belt       <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_roll_belt.1     <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_yaw_belt        <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ max_roll_belt            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_picth_belt           <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_yaw_belt             <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ min_roll_belt            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_pitch_belt           <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_yaw_belt             <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ amplitude_roll_belt      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_pitch_belt     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_yaw_belt       <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ var_total_accel_belt     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_roll_belt            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_roll_belt         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_roll_belt            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_pitch_belt           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_pitch_belt        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_pitch_belt           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_yaw_belt             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_yaw_belt          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_yaw_belt             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ gyros_belt_x             <dbl> 0.00, 0.02, 0.00, 0.02, 0.02, 0.02, 0.02, ...
## $ gyros_belt_y             <dbl> 0.00, 0.00, 0.00, 0.00, 0.02, 0.00, 0.00, ...
## $ gyros_belt_z             <dbl> -0.02, -0.02, -0.02, -0.03, -0.02, -0.02, ...
## $ accel_belt_x             <int> -21, -22, -20, -22, -21, -21, -22, -22, -2...
## $ accel_belt_y             <int> 4, 4, 5, 3, 2, 4, 3, 4, 2, 4, 2, 2, 4, 4, ...
## $ accel_belt_z             <int> 22, 22, 23, 21, 24, 21, 21, 21, 24, 22, 23...
## $ magnet_belt_x            <int> -3, -7, -2, -6, -6, 0, -4, -2, 1, -3, -5, ...
## $ magnet_belt_y            <int> 599, 608, 600, 604, 600, 603, 599, 603, 60...
## $ magnet_belt_z            <int> -313, -311, -305, -310, -302, -312, -311, ...
## $ roll_arm                 <dbl> -128, -128, -128, -128, -128, -128, -128, ...
## $ pitch_arm                <dbl> 22.5, 22.5, 22.5, 22.1, 22.1, 22.0, 21.9, ...
## $ yaw_arm                  <dbl> -161, -161, -161, -161, -161, -161, -161, ...
## $ total_accel_arm          <int> 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34...
## $ var_accel_arm            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_roll_arm             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_roll_arm          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_roll_arm             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_pitch_arm            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_pitch_arm         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_pitch_arm            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_yaw_arm              <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_yaw_arm           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_yaw_arm              <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ gyros_arm_x              <dbl> 0.00, 0.02, 0.02, 0.02, 0.00, 0.02, 0.00, ...
## $ gyros_arm_y              <dbl> 0.00, -0.02, -0.02, -0.03, -0.03, -0.03, -...
## $ gyros_arm_z              <dbl> -0.02, -0.02, -0.02, 0.02, 0.00, 0.00, 0.0...
## $ accel_arm_x              <int> -288, -290, -289, -289, -289, -289, -289, ...
## $ accel_arm_y              <int> 109, 110, 110, 111, 111, 111, 111, 111, 10...
## $ accel_arm_z              <int> -123, -125, -126, -123, -123, -122, -125, ...
## $ magnet_arm_x             <int> -368, -369, -368, -372, -374, -369, -373, ...
## $ magnet_arm_y             <int> 337, 337, 344, 344, 337, 342, 336, 338, 34...
## $ magnet_arm_z             <int> 516, 513, 513, 512, 506, 513, 509, 510, 51...
## $ kurtosis_roll_arm        <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_picth_arm       <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_yaw_arm         <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_roll_arm        <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_pitch_arm       <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_yaw_arm         <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ max_roll_arm             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_picth_arm            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_yaw_arm              <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_roll_arm             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_pitch_arm            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_yaw_arm              <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_roll_arm       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_pitch_arm      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_yaw_arm        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ roll_dumbbell            <dbl> 13.05217, 13.13074, 12.85075, 13.43120, 13...
## $ pitch_dumbbell           <dbl> -70.49400, -70.63751, -70.27812, -70.39379...
## $ yaw_dumbbell             <dbl> -84.87394, -84.71065, -85.14078, -84.87363...
## $ kurtosis_roll_dumbbell   <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_picth_dumbbell  <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_yaw_dumbbell    <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_roll_dumbbell   <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_pitch_dumbbell  <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_yaw_dumbbell    <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ max_roll_dumbbell        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_picth_dumbbell       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_yaw_dumbbell         <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ min_roll_dumbbell        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_pitch_dumbbell       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_yaw_dumbbell         <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ amplitude_roll_dumbbell  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_pitch_dumbbell <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_yaw_dumbbell   <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ total_accel_dumbbell     <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37...
## $ var_accel_dumbbell       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_roll_dumbbell        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_roll_dumbbell     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_roll_dumbbell        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_pitch_dumbbell       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_pitch_dumbbell    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_pitch_dumbbell       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_yaw_dumbbell         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_yaw_dumbbell      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_yaw_dumbbell         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ gyros_dumbbell_x         <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ...
## $ gyros_dumbbell_y         <dbl> -0.02, -0.02, -0.02, -0.02, -0.02, -0.02, ...
## $ gyros_dumbbell_z         <dbl> 0.00, 0.00, 0.00, -0.02, 0.00, 0.00, 0.00,...
## $ accel_dumbbell_x         <int> -234, -233, -232, -232, -233, -234, -232, ...
## $ accel_dumbbell_y         <int> 47, 47, 46, 48, 48, 48, 47, 46, 47, 48, 47...
## $ accel_dumbbell_z         <int> -271, -269, -270, -269, -270, -269, -270, ...
## $ magnet_dumbbell_x        <int> -559, -555, -561, -552, -554, -558, -551, ...
## $ magnet_dumbbell_y        <int> 293, 296, 298, 303, 292, 294, 295, 300, 29...
## $ magnet_dumbbell_z        <dbl> -65, -64, -63, -60, -68, -66, -70, -74, -6...
## $ roll_forearm             <dbl> 28.4, 28.3, 28.3, 28.1, 28.0, 27.9, 27.9, ...
## $ pitch_forearm            <dbl> -63.9, -63.9, -63.9, -63.9, -63.9, -63.9, ...
## $ yaw_forearm              <dbl> -153, -153, -152, -152, -152, -152, -152, ...
## $ kurtosis_roll_forearm    <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_picth_forearm   <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ kurtosis_yaw_forearm     <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_roll_forearm    <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_pitch_forearm   <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ skewness_yaw_forearm     <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ max_roll_forearm         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_picth_forearm        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ max_yaw_forearm          <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ min_roll_forearm         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_pitch_forearm        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ min_yaw_forearm          <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ amplitude_roll_forearm   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_pitch_forearm  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ amplitude_yaw_forearm    <chr> "", "", "", "", "", "", "", "", "", "", ""...
## $ total_accel_forearm      <int> 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36...
## $ var_accel_forearm        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_roll_forearm         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_roll_forearm      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_roll_forearm         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_pitch_forearm        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_pitch_forearm     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_pitch_forearm        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ avg_yaw_forearm          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ stddev_yaw_forearm       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ var_yaw_forearm          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ gyros_forearm_x          <dbl> 0.03, 0.02, 0.03, 0.02, 0.02, 0.02, 0.02, ...
## $ gyros_forearm_y          <dbl> 0.00, 0.00, -0.02, -0.02, 0.00, -0.02, 0.0...
## $ gyros_forearm_z          <dbl> -0.02, -0.02, 0.00, 0.00, -0.02, -0.03, -0...
## $ accel_forearm_x          <int> 192, 192, 196, 189, 189, 193, 195, 193, 19...
## $ accel_forearm_y          <int> 203, 203, 204, 206, 206, 203, 205, 205, 20...
## $ accel_forearm_z          <int> -215, -216, -213, -214, -214, -215, -215, ...
## $ magnet_forearm_x         <int> -17, -18, -18, -16, -17, -9, -18, -9, -16,...
## $ magnet_forearm_y         <dbl> 654, 661, 658, 658, 655, 660, 659, 660, 65...
## $ magnet_forearm_z         <dbl> 476, 473, 469, 469, 473, 478, 470, 474, 47...
## $ classe                   <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A...
```

Apparently there are a number of variables without relevant data, either empty or with NA. We will then remove variables with more than 90% of NA content.

```
CleanDataTrain <- DataTrain[,colMeans(is.na(DataTrain)) < .9]
```

Additionally let's **remove variables with low variance**, once predictors that only have a single unique value (i.e. a "zero-variance predictor") or have only a handful of unique values that occur with very low frequencies does not help on get a good prediction and may cause crashes on some models or the fit to be unstable.

```
## [1] 19622    52
```

Some models may benefit from **reducing the level of correlation** between the predictors.

```
descrCor <- cor(CleanDataTrain[,1:51])
highlyCorData <- findCorrelation(descrCor, cutoff = .75)
CleanDataTrain <- CleanDataTrain[,-highlyCorData]

dim(CleanDataTrain)
```

```
## [1] 19622    32
```

Training and Cross Validation Datasets

```
idxTrain <- createDataPartition(y=CleanDataTrain$classe, p=0.7, list=F)
DtTrain <- CleanDataTrain[idxTrain,]
DtCV <- CleanDataTrain[-idxTrain,]
dim(DtTrain)
```

```
## [1] 13737    32
```

```
dim(DtCV)
```

```
## [1] 5885   32
```

Models Tests

Let’s use the most probable successful models for **Classification** : Decision Trees, Support Vector Machine, Gradient Boosted Trees, and Random Forest.

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

CART - Classification And Regression Tree

```
CARTMeth <- train(classe ~ ., data=DtTrain, method="rpart",
                   trControl = control, tuneLength = 4)
pred_CART <- predict(CARTMeth, DtCV)
cmCART <- confusionMatrix(pred_CART, factor(DtCV$classe))
cmCART$overall[1]
```

```
##  Accuracy
## 0.5519116
```

SVN - Support Vector Machine

```
SVMMeth <- train(classe ~ ., data=DtTrain, method="svmLinear",
                 trControl = control, tuneLength = 4, verbose = F)


pred_SVM <- predict(SVMMeth, DtCV)

cmSVM <- confusionMatrix(pred_SVM, factor(DtCV$classe))

cmSVM$overall[1]
```

```
##  Accuracy
## 0.6351742
```

GBM - Gradient Boosting Machine

```
GBMMeth <- train(classe ~ ., data=DtTrain, method="gbm",
                 trControl = control, tuneLength = 5, verbose = F)


pred_GBM <- predict(GBMMeth, DtCV)

cmGBM <- confusionMatrix(pred_GBM, factor(DtCV$classe))

cmGBM$overall[1]
```

```
##  Accuracy
## 0.9802889
```

RF - Random Forest

```
RFMeth <- train(classe ~ ., data=DtTrain, method="rf",
                 trControl = control, tuneLength = 4, verbose = F)


pred_RF <- predict(RFMeth, DtCV)

cmRF <- confusionMatrix(pred_RF, factor(DtCV$classe))

cmRF$overall[1]
```

```
##  Accuracy
## 0.9884452
```

MOdels Accuracy Evaluation

```
cat(sprintf("Decision Trees:           %s\n", cmCART$overall[1]))
```

```
## Decision Trees:           0.551911639762107
```

```
cat(sprintf("Support Vector Machine:   %s\n", cmSVM$overall[1]))
```

```
## Support Vector Machine:   0.63517417162277
```

```
cat(sprintf("Gradient Boosted Machine: %s\n", cmGBM$overall[1]))
```

```
## Gradient Boosted Machine: 0.980288870008496
```

```
cat(sprintf("Random Forest:            %s\n", cmRF$overall[1]))
```

```
## Random Forest:            0.988445199660153
```

Random Forest presented the best accuracy, around 98.9% and it will be used in our predictions for test dataset.

Test Prediction

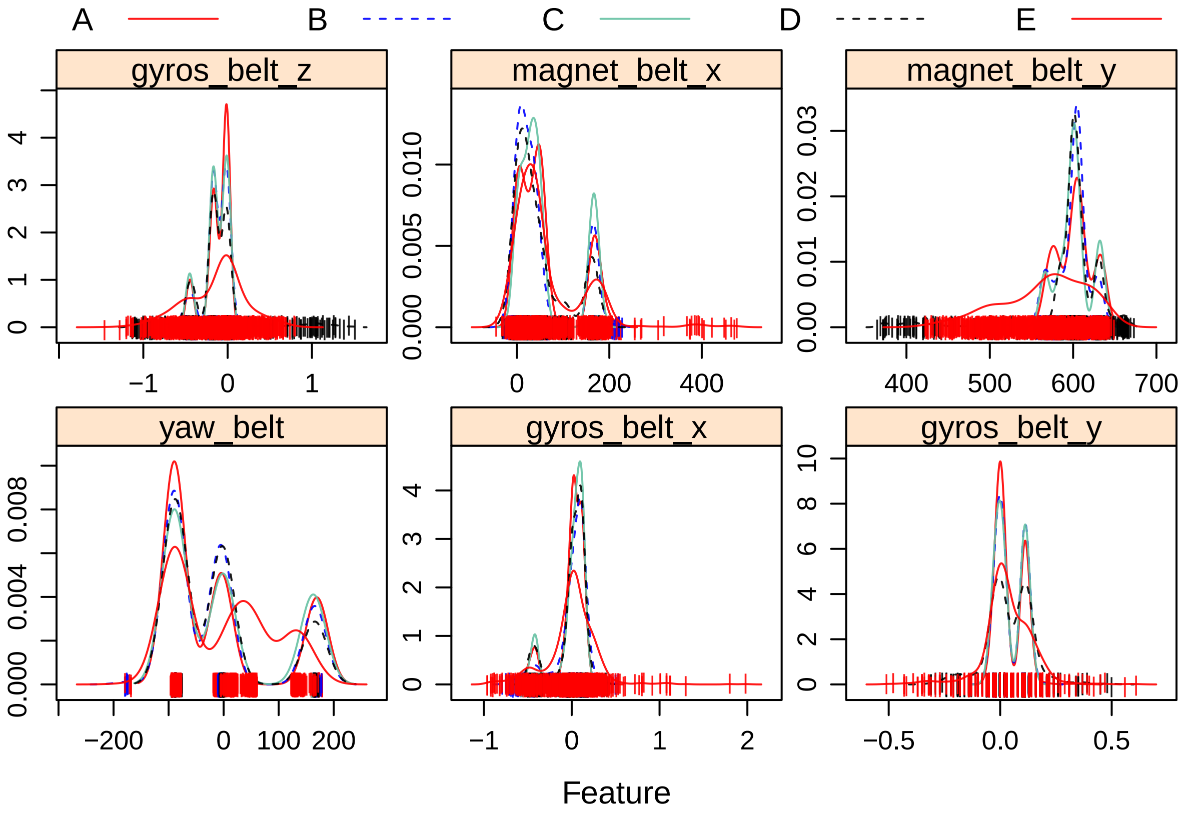
```
TestPred <- predict(RFMeth, DataTest)
TestPred
```

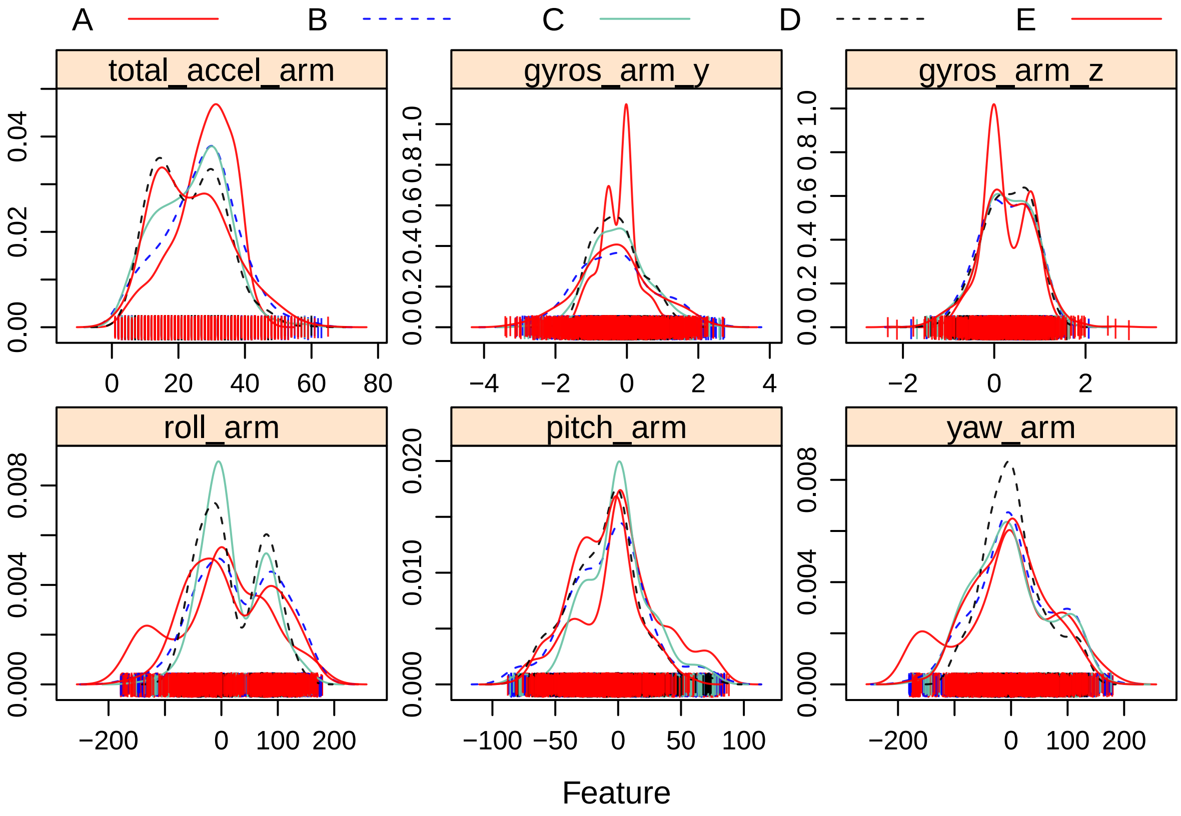
```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

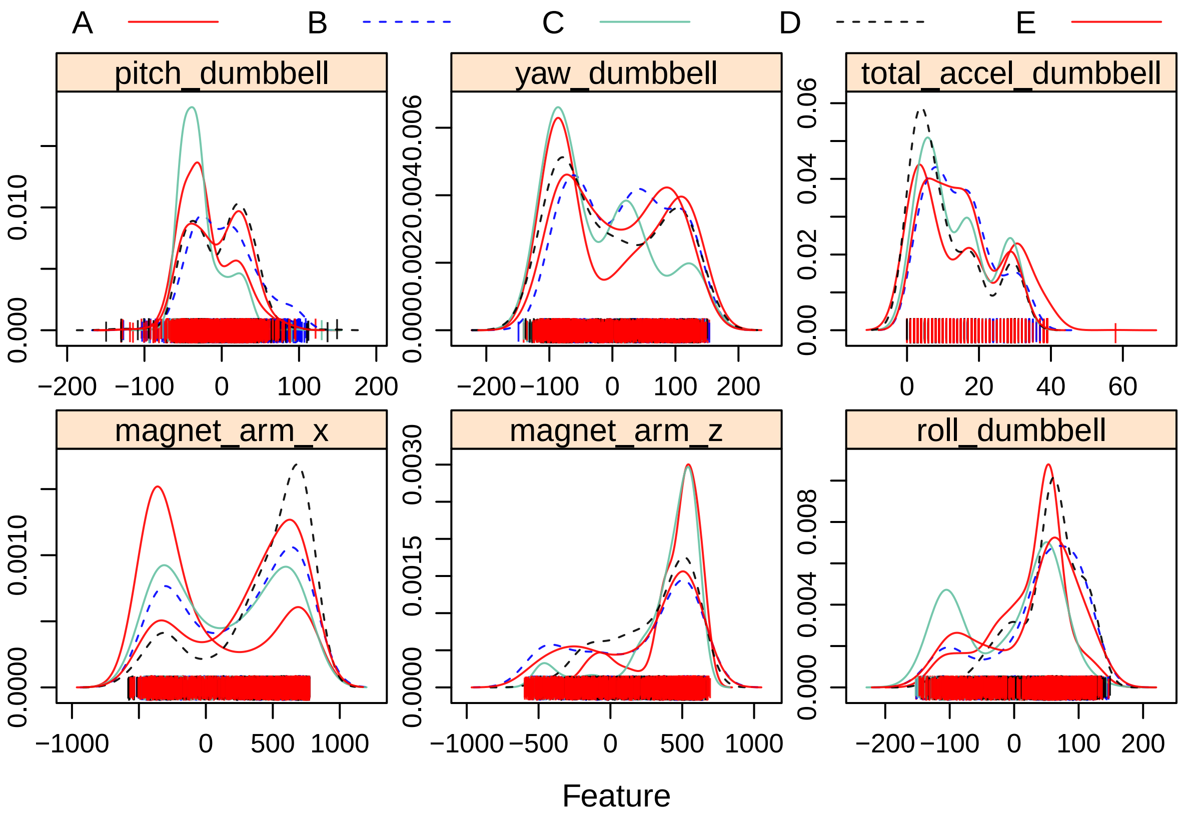
Appendix

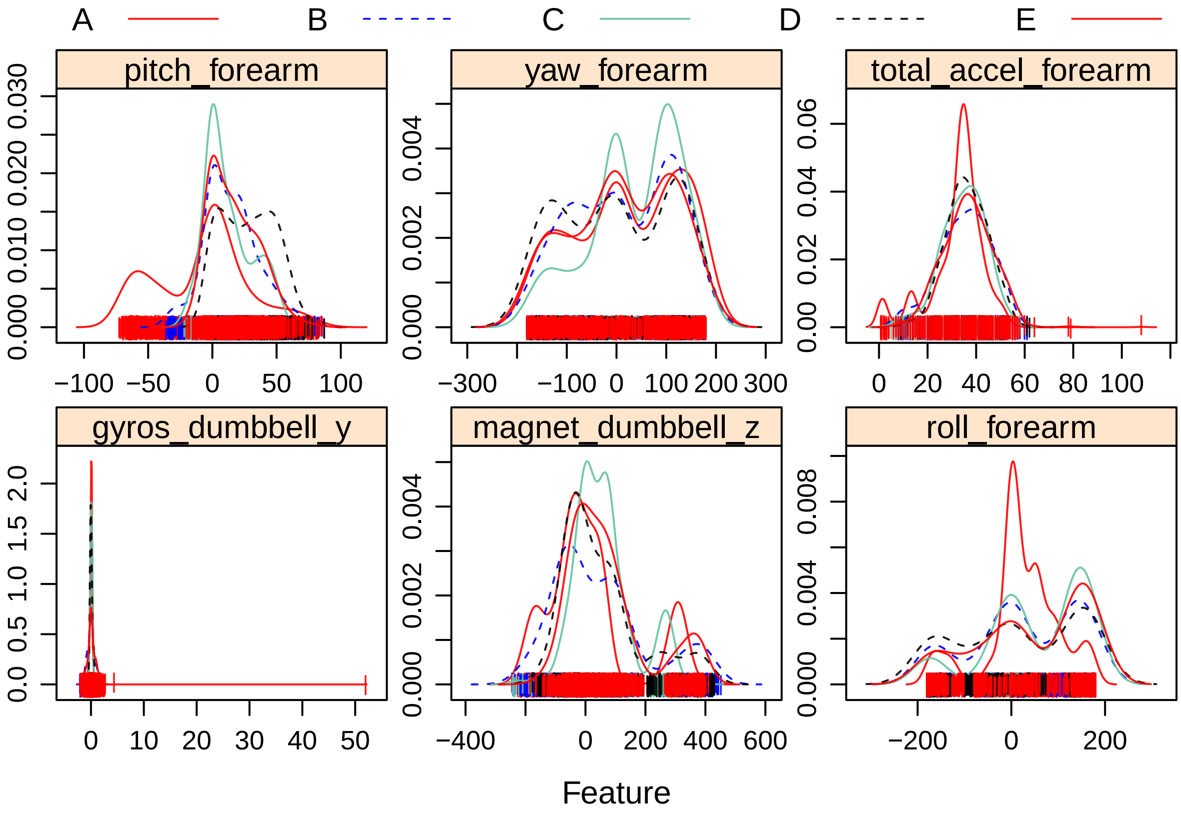
Verify variables density, quantiles, outliers and best methods performance.

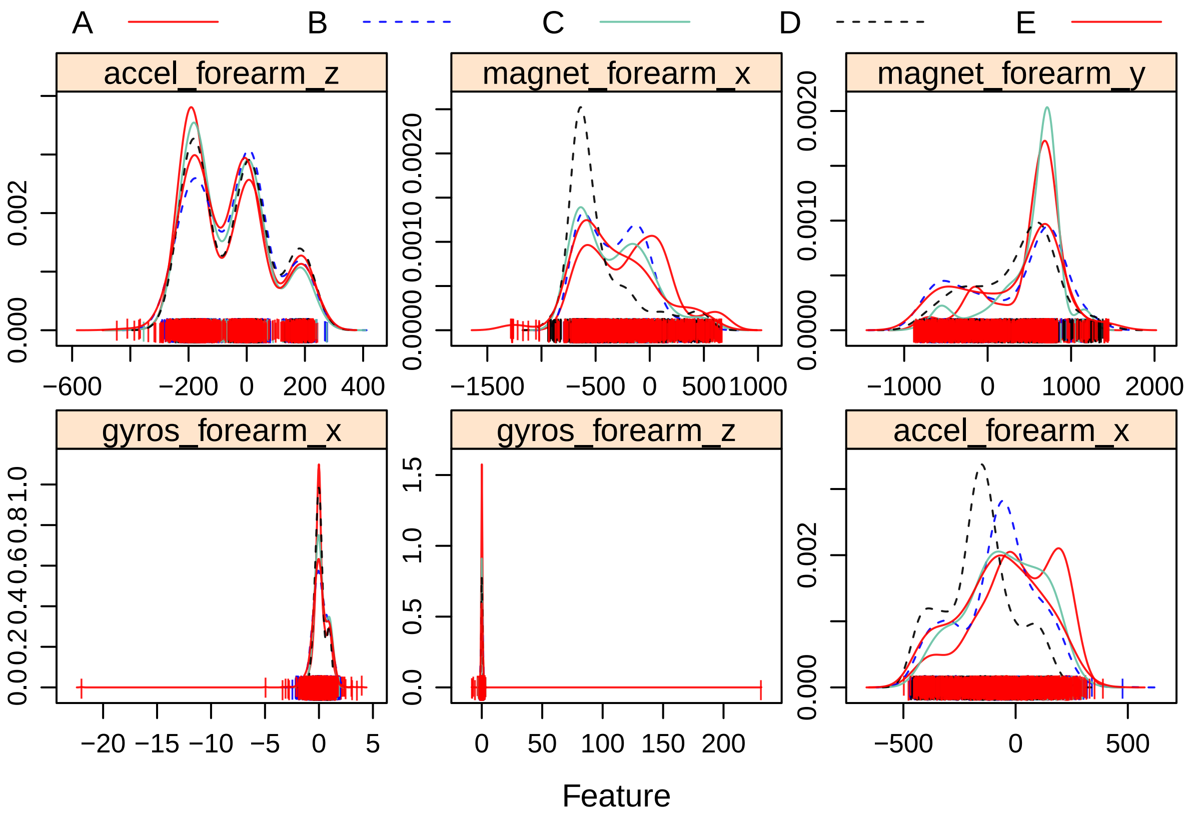
```
transparentTheme(trans = .9)
featurePlot(x = DtCV[, 1:31],
            y = as.factor(DtCV$classe),
            plot = "density",
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(3, 2),
            auto.key = list(columns = 5))
```

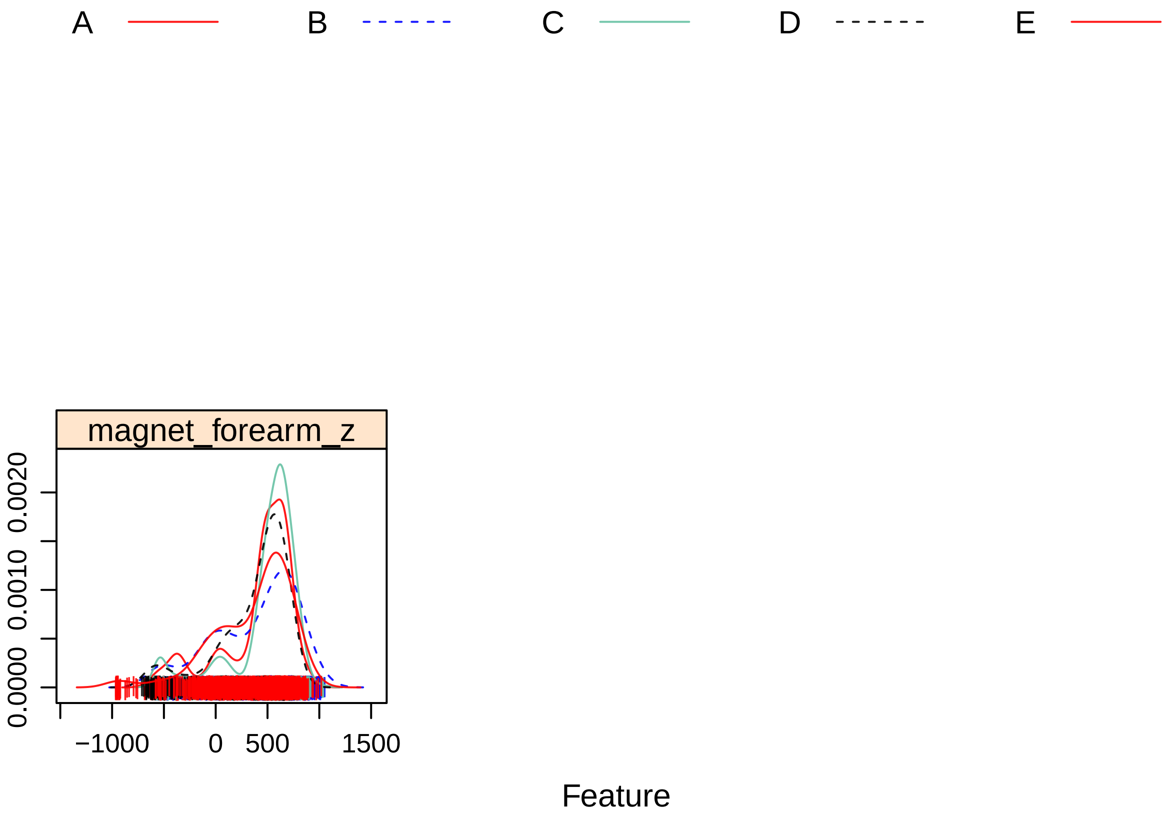

A
B
C
D
E
gyros_belt_z
magnet_belt_x
magnet_belt_y
yaw_belt
gyros_belt_x
gyros_belt_y
0
1
2
3
4
−1
0
1
0.000
0.005
0.010
0
200
400
0.00
0.01
0.02
0.03
400
500
600
700
0.000
0.004
0.008
−200
0
100
200
0
1
2
3
4
−1
0
1
2
0
2
4
6
8
10
−0.5
0.0
0.5
Feature


A
B
C
D
E
total_accel_arm
gyros_arm_y
gyros_arm_z
roll_arm
pitch_arm
yaw_arm
0.00
0.02
0.04
0 20 40 60 80
0.0 0.2 0.4 0.6 0.8 1.0
−4 −2 0 2 4
0.0 0.2 0.4 0.6 0.8 1.0
−2 0 2
0.000
0.004
0.008
−200 0 100 200
0.000
0.010
0.020
−100 −50 0 50 100
0.000
0.004
0.008
−200 0 100 200
Feature

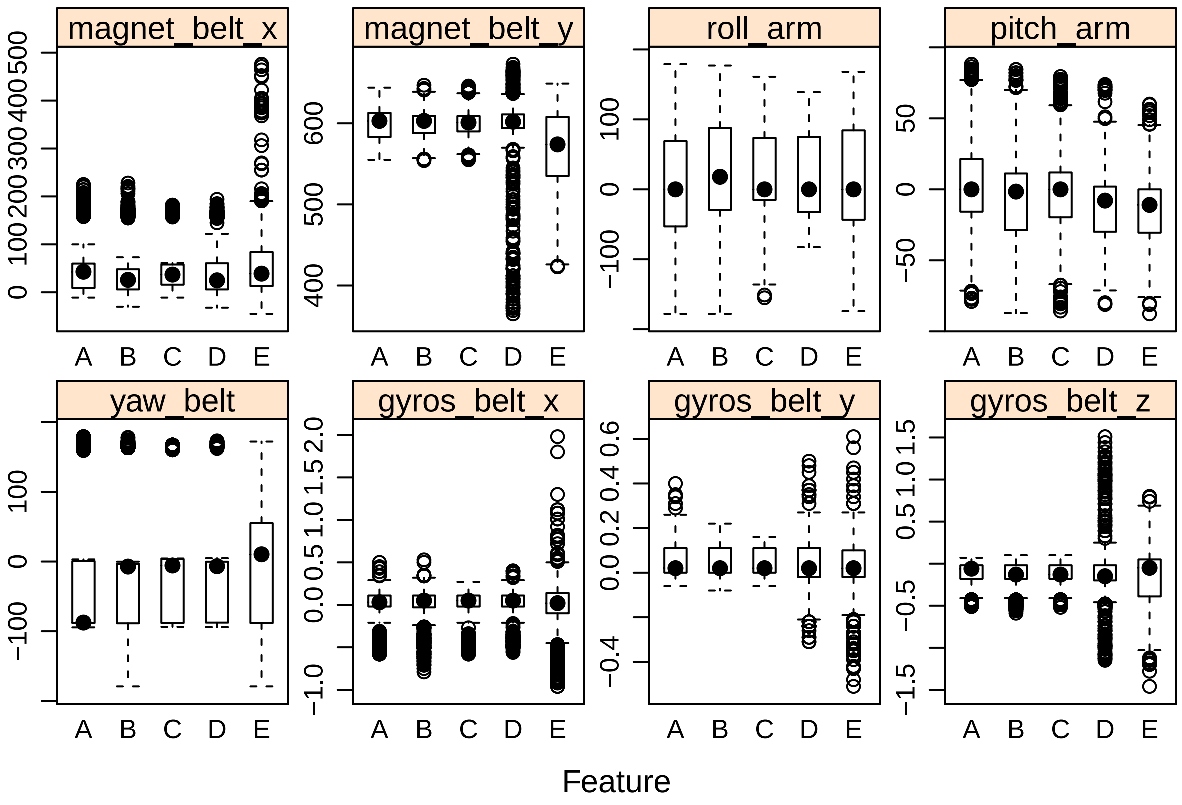

A
B
C
D
E
pitch_dumbbell
yaw_dumbbell
total_accel_dumbbell
magnet_arm_x
magnet_arm_z
roll_dumbbell
0.000
0.010
0.0000.0020.0040.006
0.00
0.02
0.04
0.06
−200
−100
0
100
200
20
40
60
0.0000
0.0010
0.0000
0.0015
0.0030
0.000
0.004
0.008
−1000
500
1000
−500
Feature

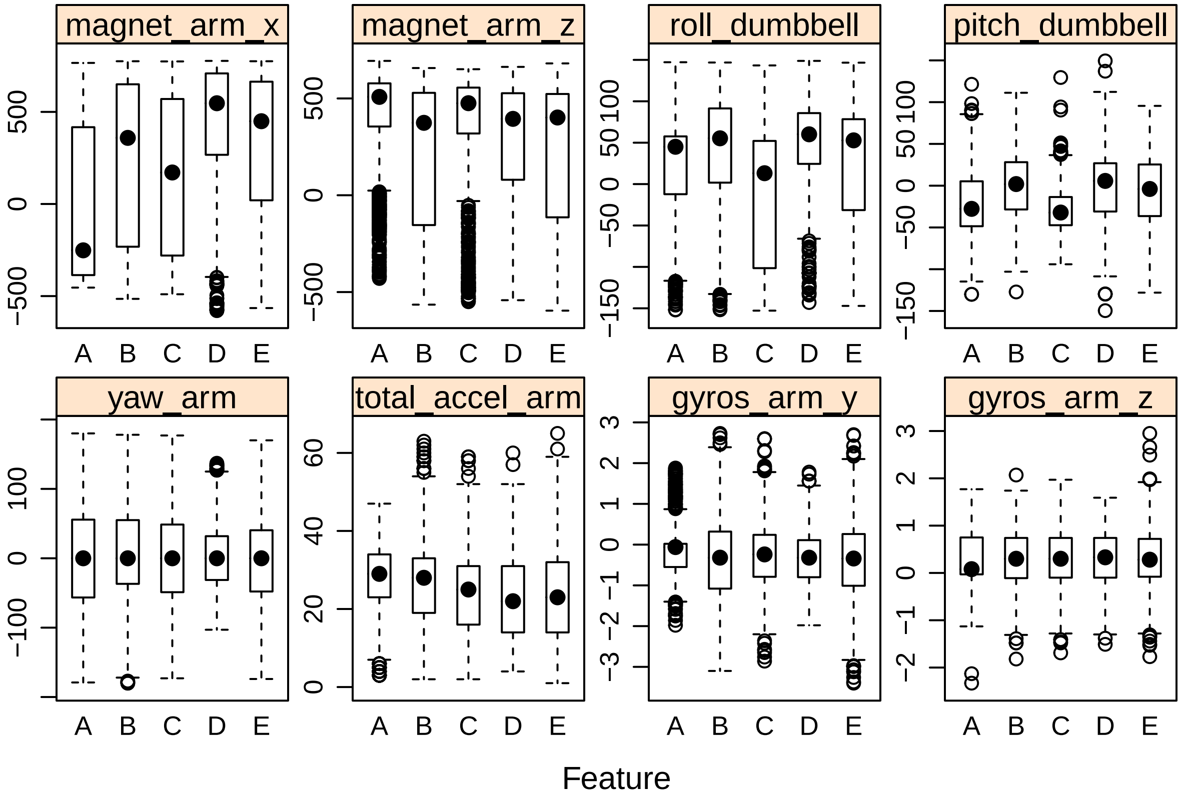

A
B
C
D
E
pitch_forearm
yaw_forearm
total_accel_forearm
gyros_dumbbell_y
magnet_dumbbell_z
roll_forearm
0.000 0.010 0.020 0.030
−100 −50 0 50 100
0.000 0.002 0.004
−300 −100 0 100 200 300
0.00 0.02 0.04 0.06
0 20 40 60 80 100
0.0 0.5 1.0 1.5 2.0
0 10 20 30 40 50
0.000 0.002 0.004
−400 0 200 400 600
0.000 0.004 0.008
−200 0 200
Feature

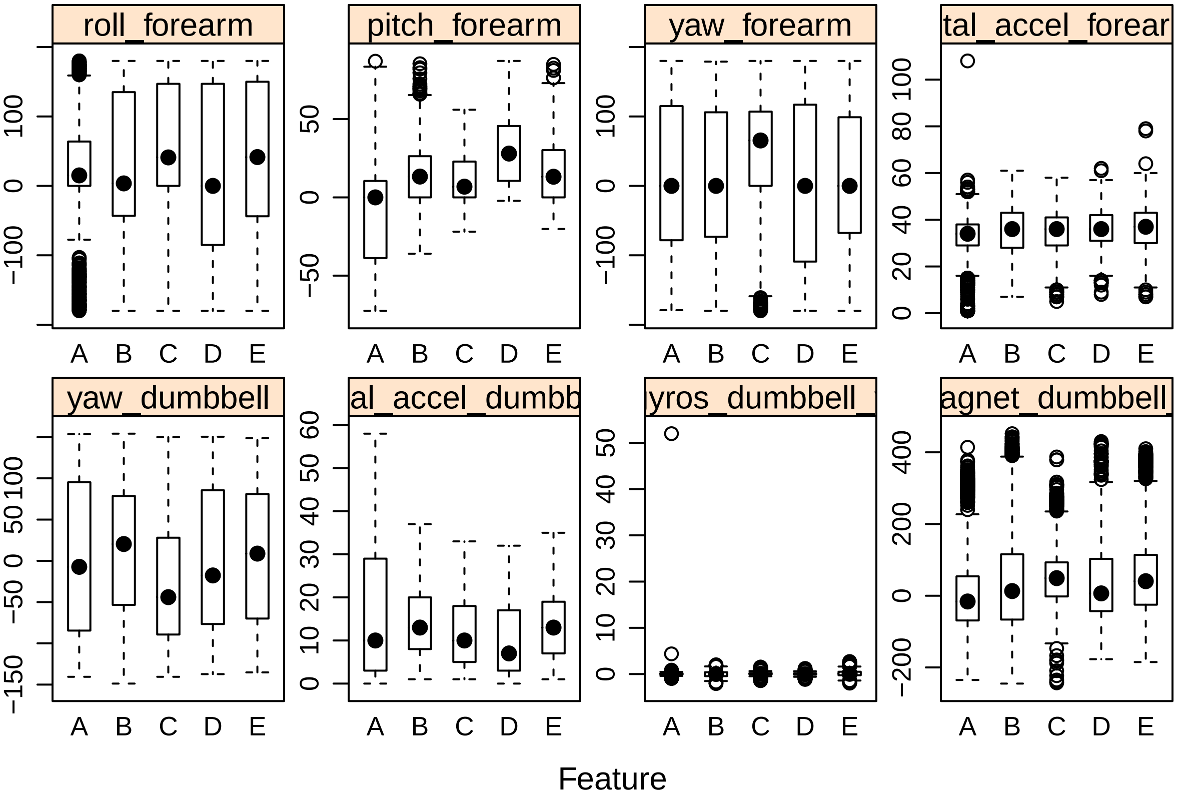

A
B
C
D
E
accel_forearm_z
magnet_forearm_x
magnet_forearm_y
gyros_forearm_x
gyros_forearm_z
accel_forearm_x
0.000
0.002
0.0000 0.0010 0.0020
0.0000
0.0010
0.0020
0.00.20.40.60.81.0
0.0
0.5
1.0
1.5
0.000
0.002
−600
−200
0
200
400
−1500
−500
0
500
1000
−1000
0
1000
2000
−20
−15
−10
−5
0
5
0
50
100
150
200
−500
0
500
Feature

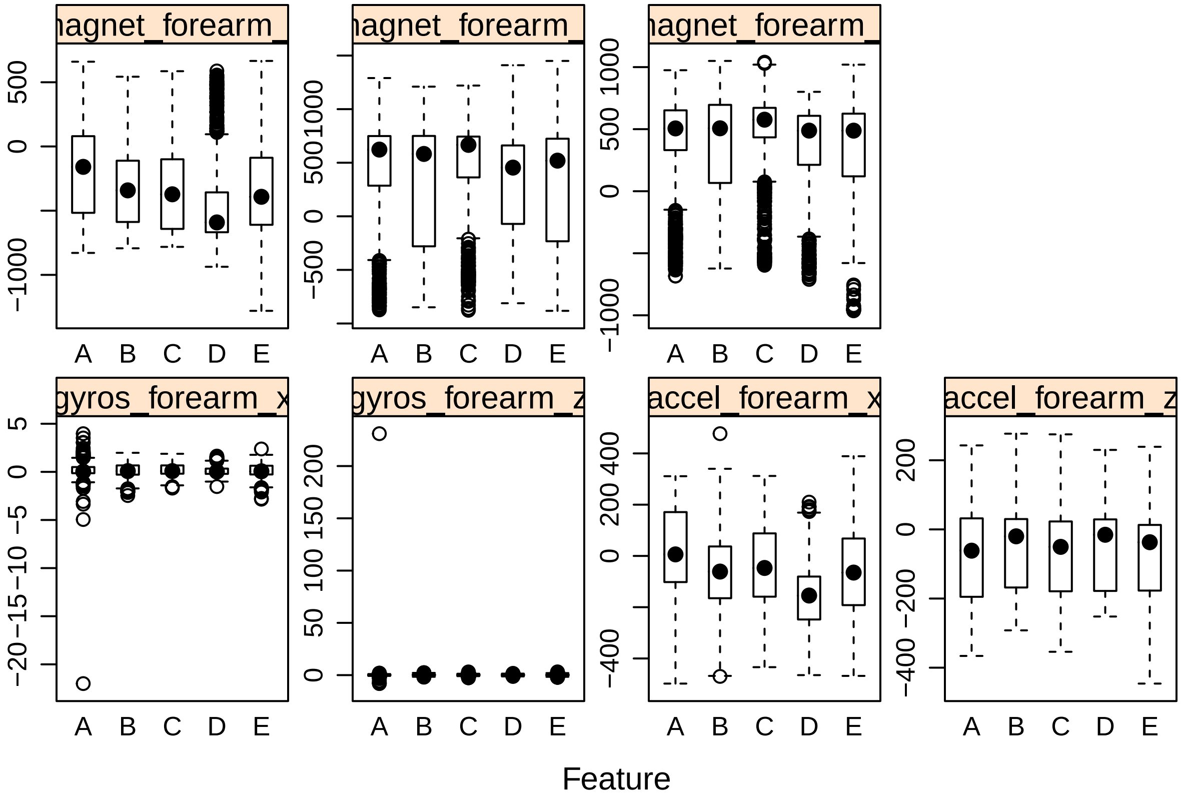


```
featurePlot(x = DtCV[, 1:31],
            y = as.factor(DtCV$classe),
            plot = "box",
            type = c("p", "smooth"),
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            span = .5,
            layout = c(4, 2))
```

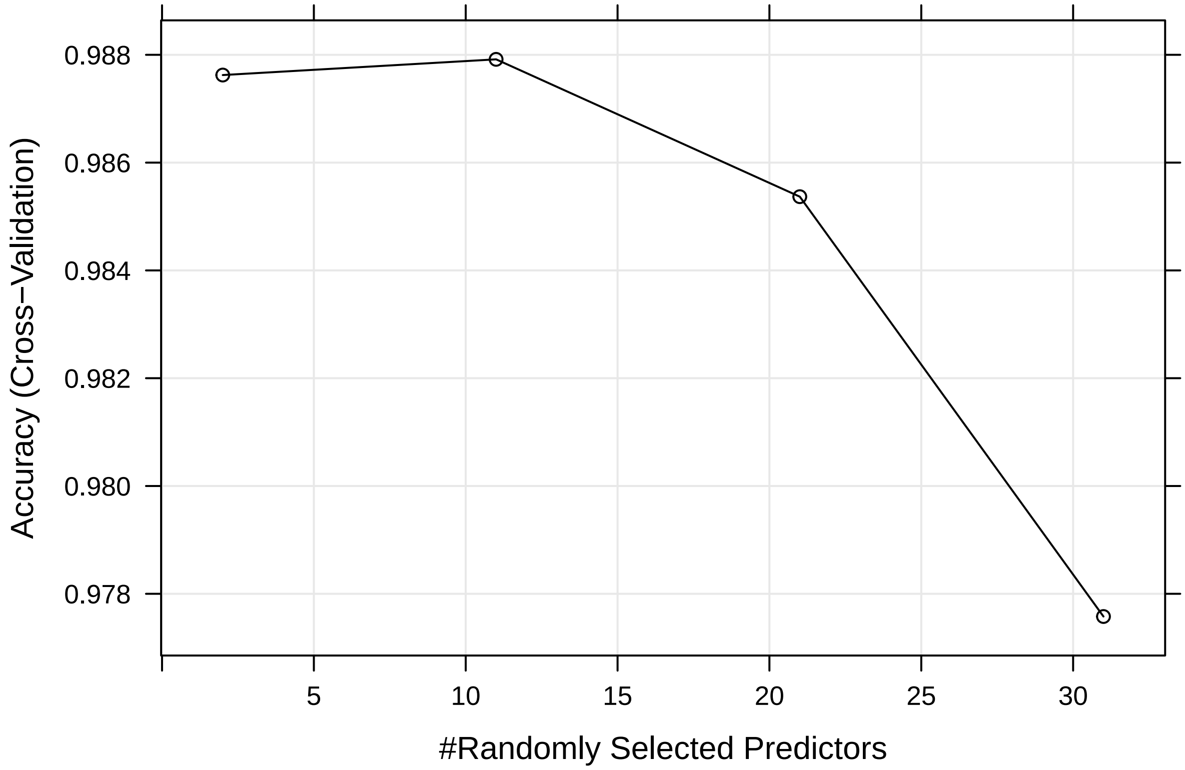

magnet_belt_x
0
100
200
300
400
500
magnet_belt_y
400
500
600
roll_arm
−100
0
100
pitch_arm
−50
0
50
yaw_belt
−100
0
100
gyros_belt_x
−1.0
0.0
0.5
1.0
1.5
2.0
gyros_belt_y
−0.4
0.0
0.2
0.4
0.6
gyros_belt_z
−1.5
−0.5
0.5
1.0
1.5
A
B
C
D
E
Feature


magnet_arm_x
magnet_arm_z
roll_dumbbell
pitch_dumbbell
yaw_arm
total_accel_arm
gyros_arm_y
gyros_arm_z
A
B
C
D
E
Feature

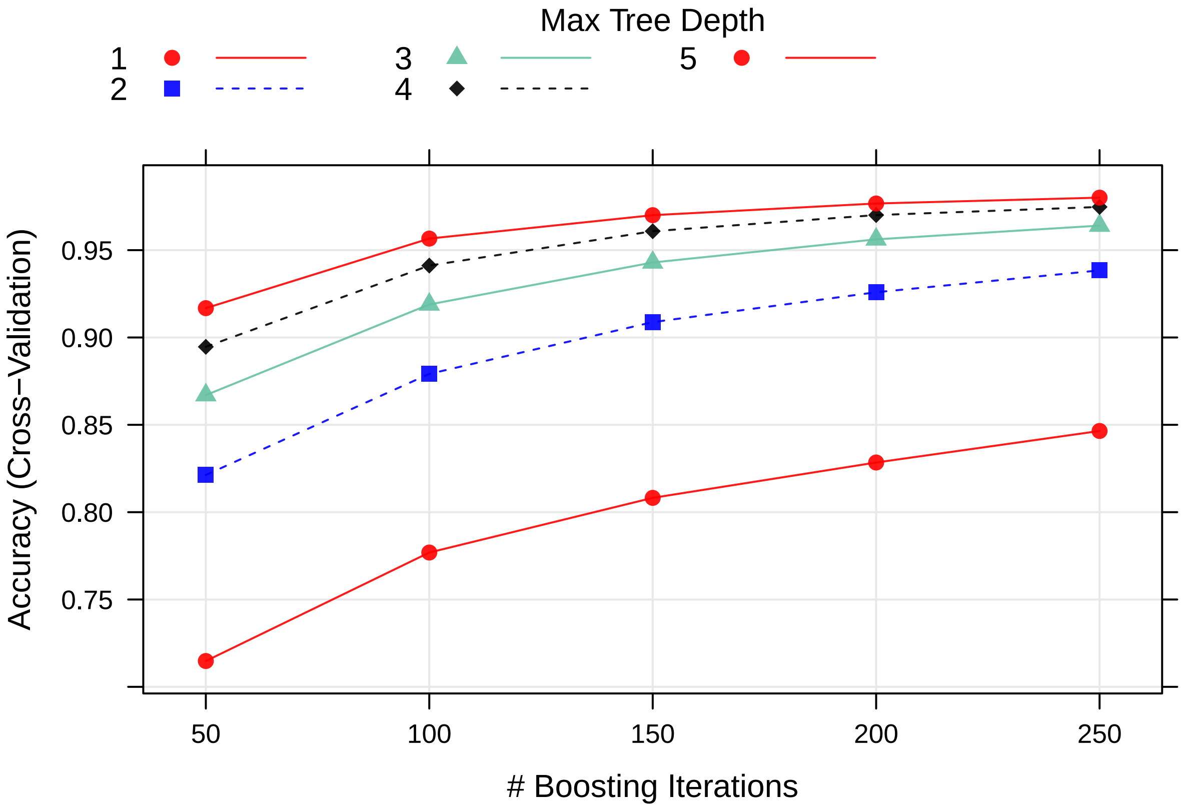

roll_forearm
pitch_forearm
yaw_forearm
tal_accel_forear
yaw_dumbbell
al_accel_dumbb
yros_dumbbell_
agnet_dumbbell_
A
B
C
D
E
-100
0
100
-50
0
50
0
20
40
60
80
100
-150
-50
0
50
100
0
10
20
30
40
50
60
-200
0
200
400
Feature



```r
plot(RFMeth)
```



```
plot(GBMMeth)
```


Max Tree Depth
1
2
3
4
5
Accuracy (Cross−Validation)
0.95
0.90
0.85
0.80
0.75
50
100
150
200
250
# Boosting Iterations