DS8-PROJECT-a

VIOLET-TAN

12/12/2020

The goal for this project will be to use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants to predict the manner in which they did the exercise. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways A, B, C, D, E. SUBMITTING REQUIREMENTS

This is the “classe” variable in the training set.

1. Use any of the other variables to predict with.
2. Create a report describing how to built the model using cross validation
3. What is the expected out of sample error is, and why it is selected.
4. Use the prediction model to predict 20 different test cases.

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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

ACKNOWLEDGE: The dataset used in this project is a courtesy of “Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers’ Data Classification of Body Postures and Movements”

##LOADING REQUIRED PACKAGES

```
#install.packages(caret,dependencies = c("depends", "suggests"))
#library(mlbench)
library(data.table)
library(rpart)
library(rpart.plot)
library(lattice)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(plyr)
library(survival)
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
#library(rattle)
```

LOADING DATA FROM DATA SOURCE

```
pmltrain <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", head
er= TRUE)
pmltest <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", header
=TRUE)
```

DATA CLEANING AND EXPLORING

```
dim(pmltrain)
```

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

```
dim(pmltest)
```

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

```
# There are 19622 observations and 160 variables in the Training dataset while 20 observations a
nd 160 variables in testing dataset
sum(is.na(pmltrain))
```

```
## [1] 1287472
```

```
sum(is.na(pmltest))
```

```
## [1] 2000
```

```
# Getting rid of unrelated and near zero variable(NZV)
trainingset <- pmltrain[, colSums(is.na(pmltrain)) == 0]
testingset <- pmltest[, colSums(is.na(pmltest)) == 0]
trainingset <- trainingset[, -c(1:7)]
testingset <- testingset[, -c(1:7)]
dim(trainingset)
```

```
## [1] 19622    86
```

```
dim(testingset)
```

```
## [1] 20 53
```

```
set.seed(1111)
inTrain <- createDataPartition(trainingset$classe, p = 0.8, list = FALSE)
trainData <- trainingset[inTrain, ]
testData <- trainingset[-inTrain, ]
dim(trainData)
```

```
## [1] 15699    86
```

```
dim(testData)
```

```
## [1] 3923   86
```

```
NZV <- nearZeroVar(trainData)
trainData <- trainData[, -NZV]
testData  <- testData[, -NZV]
dim(trainData)
```

```
## [1] 15699    53
```

```
dim(testData)
```

```
## [1] 3923   53
```

MODEL SELECTION

```
It is determined that this is a classification problem and the goal of the below comparison is t
o discover which algorithm suits the data better.
**A decision tree(dt) is a simple, decision making-diagram.
**Random forests are a large number of trees, combined (using averages or "majority rules") at t
he end of the process.
**Gradient boosting machines(gbm) also combine decision trees, but start the combining process a
t the beginning, instead of at the end.
The Kappa metric is selected as the comparison criteria.
To reduce the risk of overfitting, a 10-fold cross validation is employed during model building.
(Refer to lectures and [2])
```

Model Comparison

```
# k-fold validation - 10-fold validation, use kappa as metric
set.seed(1111)
fitControl <- trainControl(method = "cv",
                           number = 10)
gbmFit <- train(classe~., data=trainData, method="gbm", metric="Kappa", trControl=fitControl,ver
bose=FALSE)
rfFit <- train(classe~., data=trainData, method="rf", metric="Kappa", trControl=fitControl)
dtFit <- train(classe~., data=trainData, method="rpart", metric="Kappa", trControl=fitControl)
```

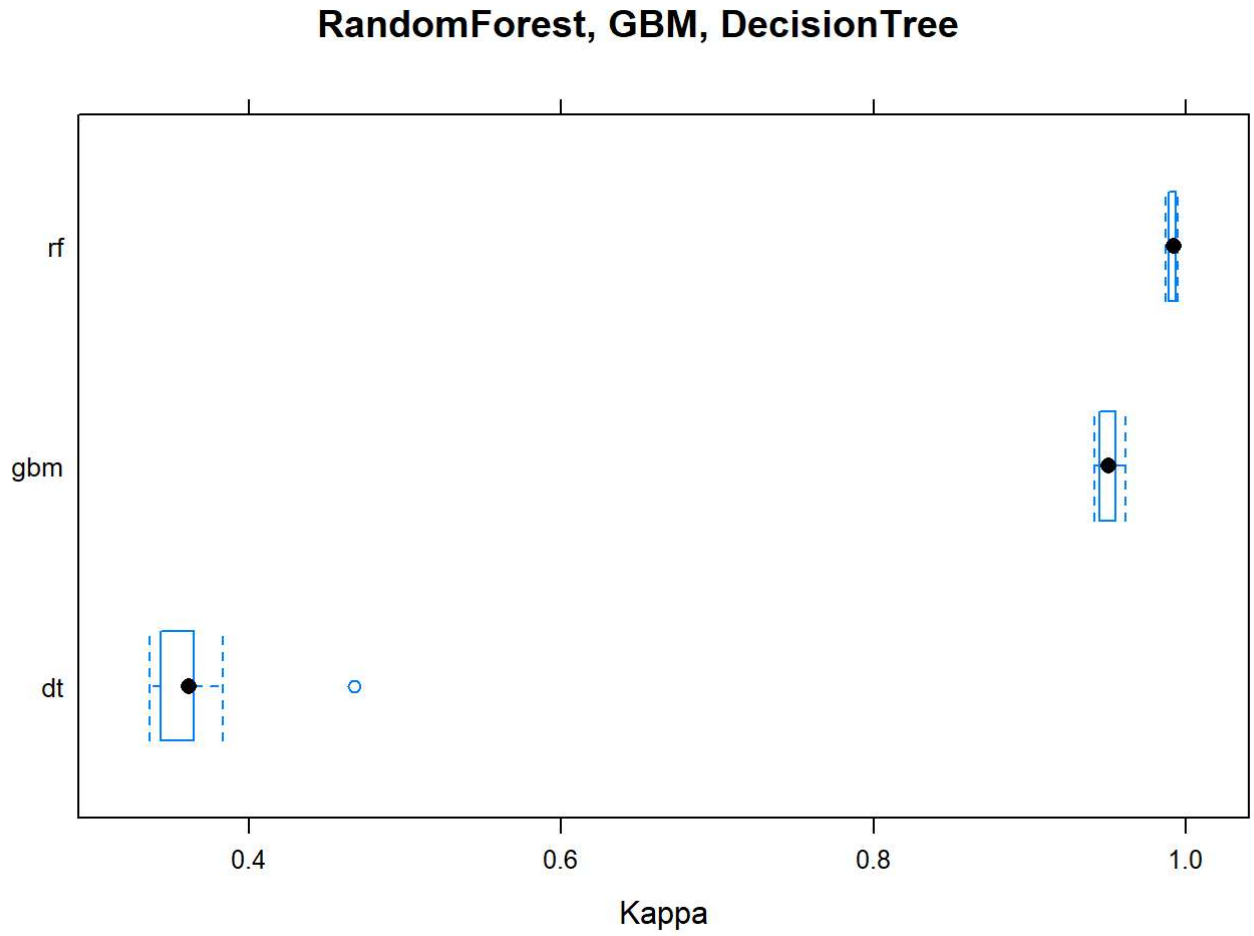
#Model Selection

```
The models are then compared using the resamples function from the Caret package.
Based on the plot below, it can be determined that the RandomForest algorithm performs better th
an the dt and gbm algorithm for this dataset, achieving a Kappa mean value of 0.9961. It can als
o be seen that the RandomForest algorithm also displays less spread than Gradient Boosting.
Therefore, the RandomForest model is selected for this dataset.
```

```
reValues <- resamples(list(rf=rfFit,gbm=gbmFit, dt=dtFit))
summary(reValues)
```

```
##
## Call:
## summary.resamples(object = reValues)
##
## Models: rf, gbm, dt
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf  0.9898089 0.9915594 0.9936285 0.9931838 0.9947458 0.9955471    0
## gbm 0.9535623 0.9569854 0.9608156 0.9605710 0.9635350 0.9694268    0
## dt  0.4939529 0.4993613 0.5116205 0.5143048 0.5128938 0.5768005    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf  0.9871083 0.9893208 0.9919385 0.9913769 0.9933528 0.9943676    0
## gbm 0.9412577 0.9455880 0.9504063 0.9501188 0.9538565 0.9613383    0
## dt  0.3368049 0.3453269 0.3618000 0.3671281 0.3645468 0.4674488    0
```

```
bwplot(reValues,metric="Kappa",main="RandomForest, GBM, DecisionTree")
```



MODEL VALIDATION

1. using the selected RandomForest model for model validation.

2. The details of the selected model is shown below.

```
rfFit
```

```
## Random Forest
##
## 15699 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14130, 14128, 14129, 14129, 14131, 14128, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9931838  0.9913769
##   27    0.9931202  0.9912971
##   52    0.9875147  0.9842057
##
## Kappa was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

3. Using the confusionMatrix function in the Caret package to validate the selected model with the testData set. The corresponding statistics and error rates are shown.

```
confusionMatrix(as.factor(testData$classe), predict(rfFit,testData))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    0    0    0    0
##          B    1  758    0    0    0
##          C    0    1  682    1    0
##          D    0    0    6  636    1
##          E    0    0    0    2  719
##
## Overall Statistics
##
##                Accuracy : 0.9969
##                  95% CI : (0.9947, 0.9984)
##     No Information Rate : 0.2847
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9961
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9991   0.9987   0.9913   0.9953   0.9986
## Specificity            1.0000   0.9997   0.9994   0.9979   0.9994
## Pos Pred Value         1.0000   0.9987   0.9971   0.9891   0.9972
## Neg Pred Value         0.9996   0.9997   0.9981   0.9991   0.9997
## Prevalence             0.2847   0.1935   0.1754   0.1629   0.1835
## Detection Rate         0.2845   0.1932   0.1738   0.1621   0.1833
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9996   0.9992   0.9953   0.9966   0.9990
```

4. From the above validation result, it can be determined that the selected Model performs at a Kappa value of 0.9961, with an accuracy of 0.9969.

FINAL MODEL TESTING

```
#1. Finally, using the selected model to predict the classification of the testing set provided. In addition, in accordance to submission instructions, the pml_write_files function is used to generate submission files.
testresults <- predict(rfFit, newdata=testingset)
print(as.data.frame(testresults))
```

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

##References [1] https://topepo.github.io/caret/featureselection.html (https://topepo.github.io/caret/featureselection.html) [2] https://topepo.github.io/caret/training.html (https://topepo.github.io/caret/training.html)