MLProj

Pradeep Gurav

23/05/2020

Executive Summary

The goal of this project is to predict the manner in which subjects did the exercise. The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. I have used RandomForest method to build the model. This report describes:
* how the model is built
* use of cross validation
* an estimate of expected out of sample error
* Predictied values for the Testdata provided

Getting and cleaning the Data

```r
set.seed(123)

train.url <-
        "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test.url <-
        "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

path <- paste(getwd(),"/", sep="")
train.file <- file.path(path, "machine-train-data.csv")
test.file <- file.path(path, "machine-test-data.csv")
if (!file.exists(train.file)) {
        download.file(train.url, destfile=train.file)
}
if (!file.exists(test.file)) {
        download.file(test.url, destfile=test.file)
}

train.data.raw <- read.csv(train.file, na.strings=c("NA","#DIV/0!",""))
test.data.raw <- read.csv(test.file, na.strings=c("NA","#DIV/0!",""))

## Remove irrelevant colums

# Drop the first 7 columns as they're not relevant for predicting.
train_data <- train.data.raw[,8:length(colnames(train.data.raw))]
test_data <- test.data.raw[,8:length(colnames(test.data.raw))]

# Drop colums with NAs
```

```
train_data <- train_data[, colSums(is.na(train_data)) == 0]
test_data <- test_data[, colSums(is.na(test_data)) == 0]

# Check for near zero variance predictors and drop them if necessary
nzv <- nearZeroVar(train_data,saveMetrics=TRUE)
zero.var.ind <- sum(nzv$nzv)

if ((zero.var.ind>0)) {
        train_data <- train_data[,nzv$nzv==FALSE]
}

# Check for near zero variance predictors and drop them if necessary
```

Split the data for cross validation

The training data is divided into two sets. This first is a training set with 70% of the data which is used to train the model. The second is a cross validation set used to assess model performance.

```
in.training <- createDataPartition(train_data$classe, p=0.70, list=F)
train.data.final <- train_data[in.training, ]
crossvalidata <- train_data[-in.training, ]
```

Model Development

We will use random forest as the model as implemented in the randomForest package.

Why we will use RandomForest method to build a model

Because
* it automatically selects important variables and
* is robust to correlated covariates & outliers in general
* 5-fold cross validation is used in the algorithm.
* averages multiple deep decision trees

```
control.parms <- trainControl(method="cv", 5)
rf.model <- train(classe ~ ., data=train.data.final, method="rf",
                  trControl=control.parms, ntree=100)

# Training set accuracy
#ptraining <- predict(rf.model, train.data.final)
#print(confusionMatrix(ptraining, train.data.final$classe))
```

Obviously the model performs excellent against the training set, but we need to cross validate the performance against the held out set and see if we have avoided overfitting.

Cross Validation using the Validation dataset (Out of Sample)

Let us now see how the model performs on the cross validation set that we held out from training.

```r
rf.predict <- predict(rf.model, crossvalidata)
print(confusionMatrix(crossvalidata$classe, rf.predict))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    1    0    0    0
##          B    5 1127    7    0    0
##          C    0    5 1020    1    0
##          D    0    0   10  954    0
##          E    0    0    5    6 1071
##
## Overall Statistics
##
##                Accuracy : 0.9932
##                  95% CI : (0.9908, 0.9951)
##     No Information Rate : 0.2851
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9914
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9970   0.9947   0.9789   0.9927   1.0000
## Specificity            0.9998   0.9975   0.9988   0.9980   0.9977
## Pos Pred Value         0.9994   0.9895   0.9942   0.9896   0.9898
## Neg Pred Value         0.9988   0.9987   0.9955   0.9986   1.0000
## Prevalence             0.2851   0.1925   0.1771   0.1633   0.1820
## Detection Rate         0.2843   0.1915   0.1733   0.1621   0.1820
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9984   0.9961   0.9888   0.9953   0.9989
```

The cross validation accuracy is 99.32% and the out-of-sample error is therefore 0.68% so the model performs rather good.

Out of sample error with the model is .68%

Test set prediction

The prediction of the algorithm for the test set is:

```r
results <- predict(rf.model,
                   test_data[, -length(names(test_data))])
results
```

```
##  [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
```

We then save the output to files according to instructions and post it to the submission page.

Reference

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13). Stuttgart, Germany: ACM SIGCHI, 2013

Annexure Graph

```r
ImpObj <- varImp(rf.model)
plot(ImpObj, main = "Top 25 influencing Variables", top = 25)
```