Practical Machine Learning Assignment

steven157

Monday, May 25, 2015

Project Description

- We were given 6 participants' data from accelerometers on the belt, forearm, arm, and dumbell by performing barbell lifts correctly and incorrectly in 5 different ways.
- Based on these data, we are going to predict the manner in which they did the exercise.

Activate required packages.

```
library(abind)
library(arm)
library(caret)
library(caTools)
library(kernlab)
library(klaR)
library(nnet)
library(rattle)
library(randomForest)
library(rpart)
```

Set a seed for pseudo-random generator.

```
set.seed(777)
```

Read data from working directory

```
train <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""))
test <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Process of cleaning and pre-processing traning dataset

- Check names' coherence.

```
all.equal(colnames(test)[1:length(colnames(test))-1], colnames(train)[1:length(colname
s(train))-1])
```

```
## [1] TRUE
```

- Clean out variables with high proportion of NAs and low variance.
- Specifically, variables with NAs more than half or related with data acquisition were removed.
- For easier computation with low informativity loss.

```
nearzero <- nearZeroVar(train, saveMetrics = TRUE)
train <- train[, !nearzero$nzv]
toberem <- sapply(colnames(train), function(x) if(sum(is.na(train[, x])) > 0.50*nrow(tr
ain)) {return(TRUE)} else{return(FALSE)})
train <- train[, !toberem]
train <- train[, -(1:6)]
```

- Correlation analysis showed that majority of the variables are highly orrelated.
- Hence, PCA was used for the pre-processing process.
- The variables selected for model specification is presented below.

```
HC <- caret::findCorrelation(cor(train[, -53]), cutoff=0.8)
names(train)[HC]
```

```
##  [1] "accel_belt_z"     "roll_belt"        "accel_belt_y"
##  [4] "accel_dumbbell_z" "accel_belt_x"     "pitch_belt"
##  [7] "accel_arm_x"      "accel_dumbbell_x" "magnet_arm_y"
## [10] "gyros_arm_y"      "gyros_forearm_z"  "gyros_dumbbell_x"
```

```
names(train)
```

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

Cross Validation

- TrainControl is used to perform 7-fold cross validation.
- Reduce out of sample error and avoid overfitting

```
tc <- trainControl(method = "cv", number = 7, verboseIter=FALSE , preProcOptions="pca",
allowParallel=TRUE)
```

Model Specification

- Bayes Generalized Linear model.

```
BGLM <- train(classe ~ ., data = train, method = "bayesglm", trControl= tc)
```

- Logit Boosted model.

```
LB <- train(classe ~ ., data = train, method = "LogitBoost", trControl= tc)
```

- Neural Net.

```
NN <- train(classe ~ ., data = train, method = "nnet", trControl= tc, verbose=FALSE)
```

- Random Forest.

```
RF <- train(classe ~ ., data = train, method = "rf", trControl= tc)
```

- Support Vector Machine (linear)

```
SVML <- train(classe ~ ., data = train, method = "svmLinear", trControl= tc)
```

- Support Vector Machine (radial)

```
SVMR <- train(classe ~ ., data = train, method = "svmRadial", trControl= tc)
```

- Accuracy comparison among these models.

```
model <- c("Bayes GLM", "LogitBoost", "Neural Net", "Random Forest", "SVM (linear)", "S
VM (radial)")
ACC <- c(max(BGLM$results$Accuracy), max(LB$results$Accuracy),
        max(NN$results$Accuracy), max(RF$results$Accuracy),
        max(SVML$results$Accuracy), max(SVMR$results$Accuracy))
KPP <- c(max(BGLM$results$Kappa), max(LB$results$Kappa),
        max(NN$results$Kappa), max(RF$results$Kappa),
        max(SVML$results$Kappa), max(SVMR$results$Kappa))
performance <- cbind(model,ACC,KPP)
knitr::kable(performance)
```

model	ACC	KPP
Bayes GLM	0.402405338464838	0.235754704805116
LogitBoost	0.895518192432842	0.866855688388553
Neural Net	0.437825863039573	0.296946869503721

Random Forest	0.995056771497045	0.993746901020635
SVM (linear)	0.786005069669905	0.727931165180934
SVM (radial)	0.934462188715825	0.916945783465477

- Based on the above table, Random Forest & Support Vector Machine (radial) provide the most accurate results and will be used for prediction on testing dataset.

Prediction on testing dataset

- Predict class variables on testing dataset
- Compare prediction similarity from both models

```
RFP <- predict(RF, test)
SVMRP <- predict(SVMR, test)
prediction <- data.frame(cbind(RFP, SVMRP))
prediction$match <- with(prediction, RFP == SVMRP)
colnames(prediction) <- c("Random Forest", "SVM (radial)", "Prediction Match?")
knitr::kable(prediction)
```

Random Forest	SVM (radial)	Prediction Match?
2	2	TRUE
1	1	TRUE
2	2	TRUE
1	1	TRUE
1	1	TRUE
5	5	TRUE
4	4	TRUE
2	2	TRUE
1	1	TRUE
1	1	TRUE
2	2	TRUE
3	3	TRUE
2	2	TRUE
1	1	TRUE
5	5	TRUE
5	5	TRUE
1	1	TRUE

	2	2	TRUE
	2	2	TRUE
	2	2	TRUE

- THe prediction result is generated by the code below for each test case

```
pml_write_files = function(x){
        n = length(x)
        for(i in 1:n){
                filename = paste0("problem_id_",i,".txt")
                write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FA
LSE)
        }
}
pml_write_files(RFP)
pml_write_files(SVMRP)
```