Practical Machine Learning Project

Introduction

This is the course project of Practical Machine Learning from coursera. The goal of this project is to predict the manner in which they did the exercise.

Loading data

```
raw_training <- read.csv('pml-training.csv')
raw_testing <- read.csv('pml-testing.csv')

set.seed(8888)
inTrain <- createDataPartition(raw_training$classe, list=FALSE, p=.9)
training = raw_training[inTrain,]
testing = raw_training[-inTrain,]
```

Preprocessing

```
nzv <- nearZeroVar(training)

training <- training[-nzv]
testing <- testing[-nzv]
raw_testing <- raw_testing[-nzv]

training <- training[-5]
testing <- testing[-5]
raw_testing <- raw_testing[-5]

num_features_idx = which(lapply(training,class) %in% c('numeric')  )

preModel <- preProcess(training[,num_features_idx], method=c('knnImpute'))
```

In some situations, the data generating mechanism can create predictors that only have a single unique value (i.e. a "zero-variance predictor"). For many models (excluding tree-based models), this may cause the model to crash or the fit to be unstable. PreProcess can be used to impute data sets based only on information in the training set. One method of doing this is with K-nearest neighbors.

Get preprocessed data

```
ptraining <- cbind(training$classe, predict(preModel, training[,num_features_idx]))
ptesting <- cbind(testing$classe, predict(preModel, testing[,num_features_idx]))
prtesting <- predict(preModel, raw_testing[,num_features_idx])

names(ptraining)[1] <- 'classe'
names(ptesting)[1] <- 'classe'

ptraining[is.na(ptraining)] <- 0
ptesting[is.na(ptesting)] <- 0
prtesting[is.na(prtesting)] <- 0
```

Fit model and corss validation

```
rf_model  <- randomForest(classe ~ ., ptraining)
```

In-sample accuracy

```
training_pred <- predict(rf_model, ptraining)
print(table(training_pred, ptraining$classe))
```

```
##
## training_pred    A    B    C    D    E
##             A 5022    0    0    0    0
##             B    0 3418    0    0    0
##             C    0    0 3080    0    0
##             D    0    0    0 2895    0
##             E    0    0    0    0 3247
```

```
print(mean(training_pred == ptraining$classe))
```

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

Out-of-sample accuracy

```
testing_pred <- predict(rf_model, ptesting)
print(table(testing_pred, ptesting$classe))
```

```
##
## testing_pred   A   B   C   D   E
##            A 557   1   1   1   0
##            B   1 377   4   0   0
##            C   0   1 335   2   2
##            D   0   0   2 317   0
##            E   0   0   0   1 358
```

```
print(mean(testing_pred == ptesting$classe))
```

```
## [1] 0.9918
```

Confusion Matrix:

```
print(confusionMatrix(testing_pred, ptesting$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 557   1   1   1   0
##          B   1 377   4   0   0
##          C   0   1 335   2   2
##          D   0   0   2 317   0
##          E   0   0   0   1 358
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.987, 0.995)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.998    0.995    0.980    0.988    0.994
## Specificity             0.998    0.997    0.997    0.999    0.999
## Pos Pred Value          0.995    0.987    0.985    0.994    0.997
## Neg Pred Value          0.999    0.999    0.996    0.998    0.999
## Prevalence              0.285    0.193    0.174    0.164    0.184
## Detection Rate          0.284    0.192    0.171    0.162    0.183
## Detection Prevalence    0.286    0.195    0.173    0.163    0.183
## Balanced Accuracy       0.998    0.996    0.988    0.993    0.997
```

Apply model to the test set

```
answers <- predict(rf_model, prtesting)
answers
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

Conclusion

We are able to provide very good prediction of weight lifting style as measured with accelerometers.