Project

The goal of this project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set.

We should create a report describing how you built our model, how you used cross validation, what we think the expected out of sample error is, and why we made the choices we did. We will also use ourprediction model to predict 20 different test cases.

1.Preparing 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)

2.Loading Data

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

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

```
dim(testing.csv)
```

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

3.Cleaning Data

```
# remove near zero variance
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
nearZero<-nearZeroVar(training.csv, saveMetrics=TRUE)
training.csv<-training.csv[, nearZero$nzv==FALSE]
# remove NAs
training.csv<-training.csv[ , colSums(is.na(training.csv))==0]
# remove not relevant columns
training.csv<-training.csv[,7:ncol(training.csv)]
dim(training.csv)
```

```
## [1] 19622    53
```

4.Random Forest

We create a model by using Random Forests.

```
# split data
set.seed(33833)
inTrain<-createDataPartition(training.csv$classe,p=0.7,list=FALSE)
training<-training.csv[inTrain,]
testing<-training.csv[-inTrain,]

# train model
ctrl<-trainControl(method="cv", number=4, allowParallel=TRUE)
modFit<-train(classe ~ ., data=training, method="rf", trControl=ctrl)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
modFit$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.71%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    1    0    0    1 0.0005120328
## B   16 2636    6    0    0 0.0082768999
## C    0   22 2370    4    0 0.0108514190
## D    0    0   40 2209    3 0.0190941385
## E    0    0    1    4 2520 0.0019801980
```

```
# in sample error
inError<-sum(diag(modFit$finalModel$confusion))/sum(modFit$finalModel$confusion)
inError
```

```
## [1] 0.992863
```

```
# cross validation
testingPredict<-predict(modFit, testing)
cm<-confusionMatrix(testingPredict,testing$classe)
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    2    0    0    0
##          B    2 1135   19    0    0
##          C    0    2 1006   28    3
##          D    0    0    1  933    3
##          E    0    0    0    3 1076
##
## Overall Statistics
##
##                Accuracy : 0.9893
##                  95% CI : (0.9863, 0.9918)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9865
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9988   0.9965   0.9805   0.9678   0.9945
## Specificity            0.9995   0.9956   0.9932   0.9992   0.9994
## Pos Pred Value         0.9988   0.9818   0.9682   0.9957   0.9972
## Neg Pred Value         0.9995   0.9992   0.9959   0.9937   0.9988
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2841   0.1929   0.1709   0.1585   0.1828
## Detection Prevalence   0.2845   0.1964   0.1766   0.1592   0.1833
## Balanced Accuracy      0.9992   0.9960   0.9869   0.9835   0.9969
```

```
# out of sample error
outError<-sum(diag(cm$table))/sum(cm$table)
outError
```

```
## [1] 0.9892948
```

The out of sample error is less than 1%.

5.Predicting 20 Test Cases

```
predict20<-predict(modFit, testing.csv)
predict20
```

```
##  [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
```

```
# create one file for each submission
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=FALSE)
  }
}
pml_write_files(predict20)
```

Thank you.