Practical Machine Learning - Final Project

ME

October 26, 2020

Project Description

From Project Description:
Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).
The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

0.Libraries

```r
        library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
        library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2

## Loading required package: lattice

## Loading required package: ggplot2
```

1.Get Data

```
        training<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")

        testing<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

3.Data Cleaning

There are various columns containing empty values and N/As. There are also columns with near zero variance which will have very little impact on modelling results. There columns will be removed as part of the data cleaning process.

```
        train_rmNA<-training[,colSums(is.na(training))==0]
        NZV<-nearZeroVar(train_rmNA)
        train_noNZV<-train_rmNA[,-NZV]
```

In additions, variables 1:6 (names, timestamps etc.) will be removed to simplify the modelling process.

```
        train_Clean<-train_noNZV[,-c(1:6)]
```

2.Splitting the Data

Here we will split the data into training and test sets. 75% of the cleaned dataset will be used for training and the remaining 25% will be used for testing accuracy of the final model.

```
#na remove


        inTrain<-createDataPartition(y=train_Clean$classe,p=0.75,list=F)
        train_data<-train_Clean[inTrain,]
        test_data<-train_Clean[-inTrain,]
```

3.Training the Model

Here we will start training some models to predict class. We will start with Random Forests as in a lot of cases the random forest algorithm makes good predictions relative to others. Random Forests also require less pre-processing relative to other models.

We can adjust the train control argument in the train function to add K-Fold cross validation. Here we add 3 folds.

```
        set.seed(123)
        rfMdl<-train(classe~.,data=train_data,method="rf",trControl=trainControl(method="cv",number = 3
        rfMdl
```

```
## Random Forest
##
## 14718 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 9812, 9812, 9812
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9889251  0.9859888
##   27    0.9891290  0.9862477
##   52    0.9820628  0.9773082
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The random forest model appears to show 98% accuracy on the training set (2% in sample error rate). We will now test this model against the test dataset.

4. Testing the Model

```
        Class_predict<-predict(rfMdl,newdata=test_data)
        rf_CM<-confusionMatrix(Class_predict,as.factor(test_data$classe))
        rf_CM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393   10    0    0    0
##          B    2  939    5    2    0
##          C    0    0  847   14    0
##          D    0    0    3  786    1
##          E    0    0    0    2  900
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.9891, 0.9943)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9899
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9895   0.9906   0.9776   0.9989
## Specificity            0.9972   0.9977   0.9965   0.9990   0.9995
## Pos Pred Value         0.9929   0.9905   0.9837   0.9949   0.9978
## Neg Pred Value         0.9994   0.9975   0.9980   0.9956   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1915   0.1727   0.1603   0.1835
## Detection Prevalence   0.2861   0.1933   0.1756   0.1611   0.1839
## Balanced Accuracy      0.9979   0.9936   0.9936   0.9883   0.9992
```

The random forest model predicts the test datset with 99% accuracy, resulting in an out-of-Sample error rate of 1%. This is a very good result. Hence, we can use this model to predict on the blind test (i.e. quiz)

5.Quiz Prediction

```
        Quiz_prediction<-predict(rfMdl,newdata = testing)
        Quiz_prediction
```

```
##  [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
```