Prediction

FT

2022-11-28

We have 2 datasets, 1. first we will use for training of our model, 2. for predicting/testing. First we will load the datasets and perform some basic clearing of it.

```
library(dplyr)
library(caret)
library(rpart)
training <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""))
testing <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

However we want to keep only columns, that does not have any missing values.

```
training<-training[,colSums(is.na(training)) == 0]
testing <-testing[,colSums(is.na(testing)) == 0]
```

We still need to get rid of few columns, which we should not use for prediction (otherwise 100% prediction rate...)

```
training   <-training[,-c(1:7)]
testing <-testing[,-c(1:7)]
```

Lets train our model. We will stick with random forest, since we got plenty of data and random forest is very strong, if we got plenty of data. First we will divide our data on training and testing sets. We will use cross-validation, since we got ton of data, we will use only 5 fold cross validation. Afterwards we will print out confusion matrix to check, how well the model did.

```
inTrain <- createDataPartition(y=training$classe, p=0.5, list=FALSE)
train <- training[inTrain, ]
test <- training[-inTrain, ]

controlforrf <- trainControl(method="cv", number=5)
fit.rf <- train(classe~., data=train, method="rf", metric="Accuracy", trControl=controlforrf)

predictionRF <- predict(fit.rf, test)

confusionMatrix(factor(test$classe), factor(predictionRF))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
```

```
##          A 2789    0    0    0    1
##          B   14 1867   15    2    0
##          C    0   12 1687   12    0
##          D    0    1   29 1576    2
##          E    0    0    1    3 1799
##
## Overall Statistics
##
##                Accuracy : 0.9906
##                  95% CI : (0.9885, 0.9924)
##     No Information Rate : 0.2857
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9881
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9950   0.9931   0.9740   0.9893   0.9983
## Specificity            0.9999   0.9961   0.9970   0.9961   0.9995
## Pos Pred Value         0.9996   0.9837   0.9860   0.9801   0.9978
## Neg Pred Value         0.9980   0.9984   0.9944   0.9979   0.9996
## Prevalence             0.2857   0.1916   0.1766   0.1624   0.1837
## Detection Rate         0.2843   0.1903   0.1720   0.1607   0.1834
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9974   0.9946   0.9855   0.9927   0.9989
```

If we check the model accuracy, we see that the model predicts allmost too well. As we see also from the confusion matrix, the prediction is really good. (maybe some error that was missed) At last, we will predict on the test data to find our 20 predictions. They are as follows.

```
predict(fit.rf, testing)
```

```
##  [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
```