Machine Learning Course: Project

Alex Zagorskiy
October 21, 2017

Abstract

Analysis of the data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants is presented. The goal of the analysis is to predict how well the participants are doing an exercise. An instant performance of the user was quantified using using 5 factored levels.

Analysis has been carried out using several prediction methods provided by the R machine learning packages (namely, regression tree rpart, Gradient Boosting Machines gbm, and rainforest rf). The latter two methods generate predictions with the accuarace of more than 95%, whereas the accuracy of rpart was unacceptable).

Depersonalizing the dataset by excluding the user names from the prediction model does not impact its predictive accuracy. Hence, the Combined Model can be applied to all participating individuals.

Preprocessing the data.

Reading data. Initial datasets were downloaded into the working directory.

```
setwd("C:/AZ/COURSES/DATA_science/MACHINE_LEARNING")
dat<-read.csv(file="pml-training.csv")
dat_test<-read.csv(file="pml-testing.csv")
```

Cleaning data; removing empty columns and those populated solely with NAs.

```
sel1<-apply(dat, 2, function(x){sum(is.na(x))})
dat2<-dat[, sel1<18000]
test2<-dat_test[,sel1<18000]
sel2<-apply(dat2, 2, function(x){sum(x=="")})
dat3<-dat2[, sel2<18000]
test3<-test2[,  sel2<18000]
```

Splitting the dataset

An available test dataset does not allow for verifying the fitted models as it contained only 20 datapoints. Let us split the training set into the training and validation subsets and make sure that the subsets for different users are uniformly represented in the training and validation sets.

```
set.seed(3433)
#Splitting the DS
inTrain = createDataPartition(dat3$classe, p =3/4)[[1]]
training = dat3[ inTrain,]
validat = dat3[-inTrain,]
table(training$user_name);table(validat$user_name)

##
##   adelmo carlitos  charles   eurico   jeremy    pedro
##     2910     2276     2667     2295     2586     1984

##
##   adelmo carlitos  charles   eurico   jeremy    pedro
##      982      836      869      775      816      626

#Ploting histograms of the parameter of interest vs user names

ggplot(training, aes(x=classe)) +
       stat_count( width=0.5,  colour="black", fill="lightblue") +
        facet_wrap(~user_name  )
```

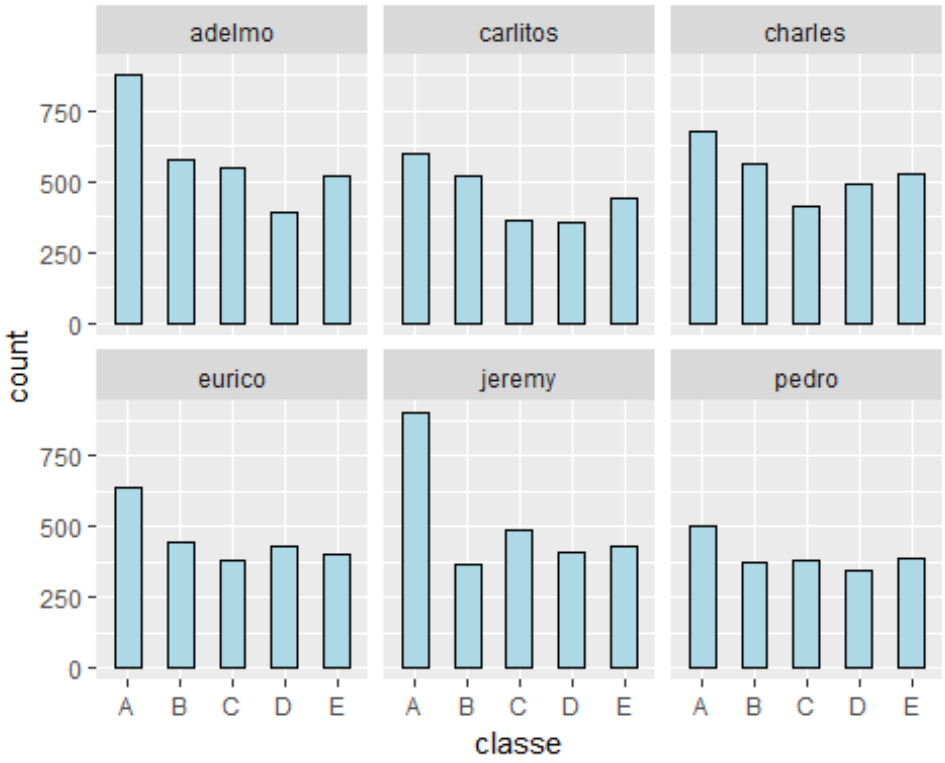
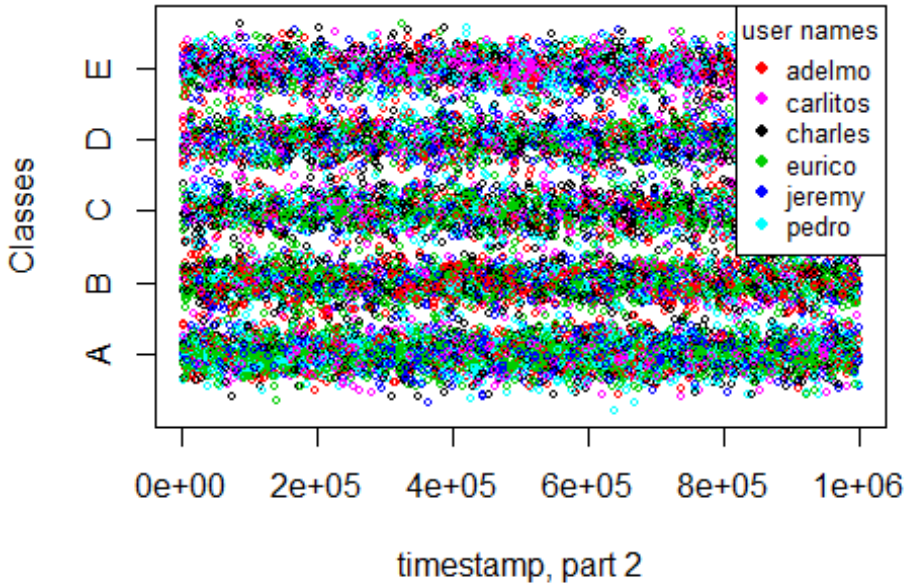


Figure 1: Histograms of the parameter of interest split per user names

One sees, that all six distributions are qualitatively similar, hence they can be treated in a similar way. Now we visually check, whether there exists a temporal dependence of the parameters of interest (per user).In order to separate the plots

visually , small random numbers are added to the numerical equivalents if the "classes".

```
xx<-as.numeric(training$classe)+rnorm(length(training$classe), 0.0, sd
= 0.2)
plot(training$raw_timestamp_part_2, xx,
     col=training$user_name,cex=0.5, yaxt="n" , ylab="Classes",
     xlab= "timestamp, part 2")
legend("topright", legend = levels(training$user_name),
        col= unique(training$user_name), pch=16,
       title = "user names",cex=0.8)
       axis(2, 1:5, levels(training$classe))
```



Apparently, there is no noticeable pattern, hence, no significant correlation with the time is to be expected.

Dimension reduction

It makes sent to attempt reducing the number of predictors. In the next only the numerical variables are selected for the principal component analysis reduction. Performing the PCA and generating the reduced data set, containing the variables responsible for 99% of variance. It is important to cache the pca model. Otherwise, we would need to recalculate all the prediction models (which will be also cached later) if we decide to change the random seed.

```
dat4_train<-apply(training[, 7:59], 2, as.numeric)
dat4_valid<-apply(validat[, 7:59], 2, as.numeric)
test4<-apply(test3[, 7:59], 2, as.numeric)

mod_nam<-paste("preproc_1",".pred", sep="")
if(!file.exists(mod_nam)){
        preProc<-preProcess(dat4_train, method="pca", thresh = 0.99)
        saveRDS(preProc, mod_nam)
} else preProc<-readRDS(mod_nam)
print(paste ("Now we have ", as.character(preProc$numComp),"
variables",
" instead of ",as.character(ncol(dat4_train))))

## [1] "Now we have  37  variables  instead of  53"

qq_train<-predict(preProc, dat4_train)
qq_valid<-predict(preProc, dat4_valid)
qq_test<-predict(preProc, test4)
```

Bringing the factor variables back to the PCA-transformed data frames.

```
qq_train<-data.frame(qq_train,
training$user_name,training$new_window,training$classe)
qq_valid<-data.frame(qq_valid,
validat$user_name,validat$new_window,validat$classe)
qq_test<-data.frame(qq_test,
test3$user_name,test3$new_window,test3$problem_id)
names(qq_train)<-gsub("training.", "", names(qq_train))
names(qq_valid)<-names(qq_train)
names(qq_test)<-names(qq_train); names(qq_test)
[ncol(qq_test)]<-"problem_id"
```

Prediction modelling

We will consider two approaches, namely building a depersonalized general models for all the participants and, secondly generating a set of user-specific models. The idea is to understand, whether we can use a unified approach or have to treat each person separately.

Let us define a function, which would make model fits for general dataset as well as for the individual user subsets and assess its accuracy on the training, validation subsets and on the given test data set. The exact "classe" parameters for the latter one are not shown in this file as the "Coursera Code of Honor" is to be respected. They are hidden in the file "mod_fit_0.txt", which was generated based on the output from one of the prediction models, which resulted in 100% success rate in Quiz 4 of this course.

Advance prediction algorithms may require significant computational resourses. We will cache all intermediate models, in order to avoid re-calculating while editing and

debugging the Rmd file. Prediction method (character meth) and the request to generate separate models for each user name (logical loop=...) are the input parameters.

```
        #reading the hidden output of the test dataset.
        pred_0<-read.csv(file="mod_fit_0.txt")$x
```

Below is the function respocible for prediction and verification

```
        prediction_z<-function(meth, loop=T, variant=""){
        acc<-c(1:6); RMS<-c(1:6);cm<-c(1:6)
        pred_test_fin<-pred_0; acc_test<-rep(NA, 6)
        us_nam<-table((qq_train$user_name))
        #de-personalizing the dataset
        xx<-names(qq_train)!="user_name"
        if(loop) {ii<-6; switch<-F
        } else {ii<-1; switch<-T}
        for (i in c(1:ii)){
        yy<-qq_train$user_name==names(us_nam)[i] | switch
        yy_valid<-qq_valid$user_name==names(us_nam)[i] | switch
        yy_test<-qq_test$user_name==names(us_nam)[i] | switch
        mod_nam<-paste("mod_", meth,"_",as.character(i*loop),
                        variant,".rds",sep="")
                if(!file.exists(mod_nam)){
                modfit_1<- train(classe ~ ., data=qq_train[,xx],
method=meth)
                saveRDS(modfit_1, mod_nam)
                } else modfit_1<-readRDS(mod_nam)
        #calcualting preditions
        pred_1<-predict(modfit_1, qq_train[yy,xx])
        pred_valid<-predict(modfit_1, qq_valid[yy_valid,xx])
        pred_test_1<-predict(modfit_1, qq_test[yy_test, xx])
        pred_test_fin[qq_test$problem_id[yy_test]]<-pred_test_1
        #accuracy
        acc[i]<- max(modfit_1$results$Accuracy)
        cm[i]<-confusionMatrix(pred_valid, qq_valid$classe[yy_valid])
$overall[1]
        acc_test<-rep(sum(pred_test_fin==pred_0)/20,ii)
        rm(modfit_1)}   # freeing the memory
        #forming a consolidated output
        outp<-rbind.data.frame(round(acc[1:ii],3),round(cm[1:ii],3),
                               round(acc_test[1:ii],2))
        row.names(outp)<-c("training", "validation", "test")
        if(ii==6)  {
                names(outp)<-names(us_nam)[1:6]
                print(paste("Split model fit (per user), Method= ",
meth))
        } else {
                names(outp)<-"All users"
                print(paste("Combined model fit (all users), Method= ",
```

```
meth))
        }
        write.csv(pred_test_fin, file=paste("mod_test", meth,".txt"))
        print( "Accuracy on the training,validation and test sets " )
        print(outp)
        list(pred_test_fin, outp)
        }
```

Prediction using the Recursive partitioning algorithm

```
pred<-prediction_z("rpart", loop=F,variant="")

## [1] "Combined model fit (all users), Method=  rpart"
## [1] "Accuracy on the training,validation and test sets "
##            All users
## training       0.380
## validation     0.359
## test           0.450

pred<-prediction_z("rpart",  loop=T,variant="")

## [1] "Split model fit (per user), Method=  rpart"
## [1] "Accuracy on the training,validation and test sets "
##            adelmo carlitos charles eurico jeremy pedro
## training    0.373    0.364   0.373  0.383  0.370 0.375
## validation  0.339    0.404   0.417  0.307  0.423 0.232
## test        0.450    0.450   0.450  0.450  0.450 0.450
```

One sees that the prediction with the rpart model generates unacceptable results, actually worse, than it would be expected from the coin flipping. A further unalysis is required in order to understand the root cause. Meanwhile let us try some advanced methods.

Prediction with the use of the rainforest method.

```
pred<-prediction_z("rf", loop = F,variant="")

## [1] "Combined model fit (all users), Method=  rf"
## [1] "Accuracy on the training,validation and test sets "
##            All users
## training       0.973
## validation     0.979
## test           1.000

pred<-prediction_z("rf", loop = T,variant="")

## [1] "Split model fit (per user), Method=  rf"
## [1] "Accuracy on the training,validation and test sets "
##            adelmo carlitos charles eurico jeremy pedro
## training    0.973    0.973   0.973  0.972  0.972 0.973
## validation  0.966    0.982   0.987  0.995  0.968 0.982
## test        1.000    1.000   1.000  1.000  1.000 1.000
```

This algorithm shows the best performance. One sees that there is no apparent difference in the accuracy of the general and user-tailored algorithms. An obvious disadvantage is an extremly high computational effort required to generate the fit model.

Prediction with the use of the Gradient Boosting Machines (gbm) method.

```
pred<-prediction_z("gbm", loop = F,variant="")
```

```
## [1] "Combined model fit (all users), Method=  gbm"
## [1] "Accuracy on the training,validation and test sets "
##            All users
## training       0.854
## validation     0.868
## test           0.950
```

```
pred<-prediction_z("gbm", loop = T,variant="")
```

```
## [1] "Split model fit (per user), Method=  gbm"
## [1] "Accuracy on the training,validation and test sets "
##            adelmo carlitos charles eurico jeremy pedro
## training    0.854    0.855   0.854  0.854  0.854 0.854
## validation  0.813    0.882   0.934  0.894  0.839 0.859
## test        0.950    0.950   0.950  0.950  0.950 0.950
```

This algorithm is faster than RF but still shows quite a good accuracy.

Cross-validation using the reduced size of the training set.

```
interim<-qq_train
inTrain = createDataPartition(interim$classe, p =0.15)[[1]]
qq_train = interim[ inTrain,]
#pred<-prediction_z("rf", loop = F,variant="X_valid")
pred<-prediction_z("rf", loop = T,variant="X_valid")
```

```
## [1] "Split model fit (per user), Method=  rf"
## [1] "Accuracy on the training,validation and test sets "
##            adelmo carlitos charles eurico jeremy pedro
## training    0.881    0.881   0.879  0.880  0.880 0.885
## validation  0.866    0.903   0.946  0.907  0.892 0.907
## test        0.900    0.900   0.900  0.900  0.900 0.900
```

As it could be expected the model trained on a significantly reduced dataset showed slightly lower (but comparable!!!) accuracy

Conclusions

Advanced classification models such as rainforest and gradient boosting machine exhibit very good predictive accuracy both on validation and test datasets. Nonetheless they require significant computation effort.

Depersonalised predictive model does show a similar accuracy as those customized for each individual. Hence, the combined model can potentially be used to predict (monitor) performace of the entire population of interest.