Predicting the type of movement based on accelerator data

```
## Warning: package 'caret' was built under R version 3.1.2
```

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

```
## Warning: package 'AppliedPredictiveModeling' was built under R version
## 3.1.2
```

```
## Warning: package 'kernlab' was built under R version 3.1.2
```

```
## Warning: package 'e1071' was built under R version 3.1.2
```

executive summary

This report presents the build-up of a model based on a training set of several person exercising to predict the type of movement they are doing based on accelerometer data. The data used in this analysis is from groupware project on Human Activity Recognition [1]. The outcome is the column “classe” which is a factor of 5 levels, from A to E corresponding to a type of activity. The model chosen for this prediction was a CARE tree prediction without preprocessing of the data except some simple data cleaning. The accuracy of the model was around 50% on a subset of the training set for testing.

Data exploration and data cleaning

The objective is to predict the “classe”" parameter based on the other metrics. Looking at the data set, we have 160 columns, any of them are are statistically summary ofmore detailed parametrics taking during a time widow.

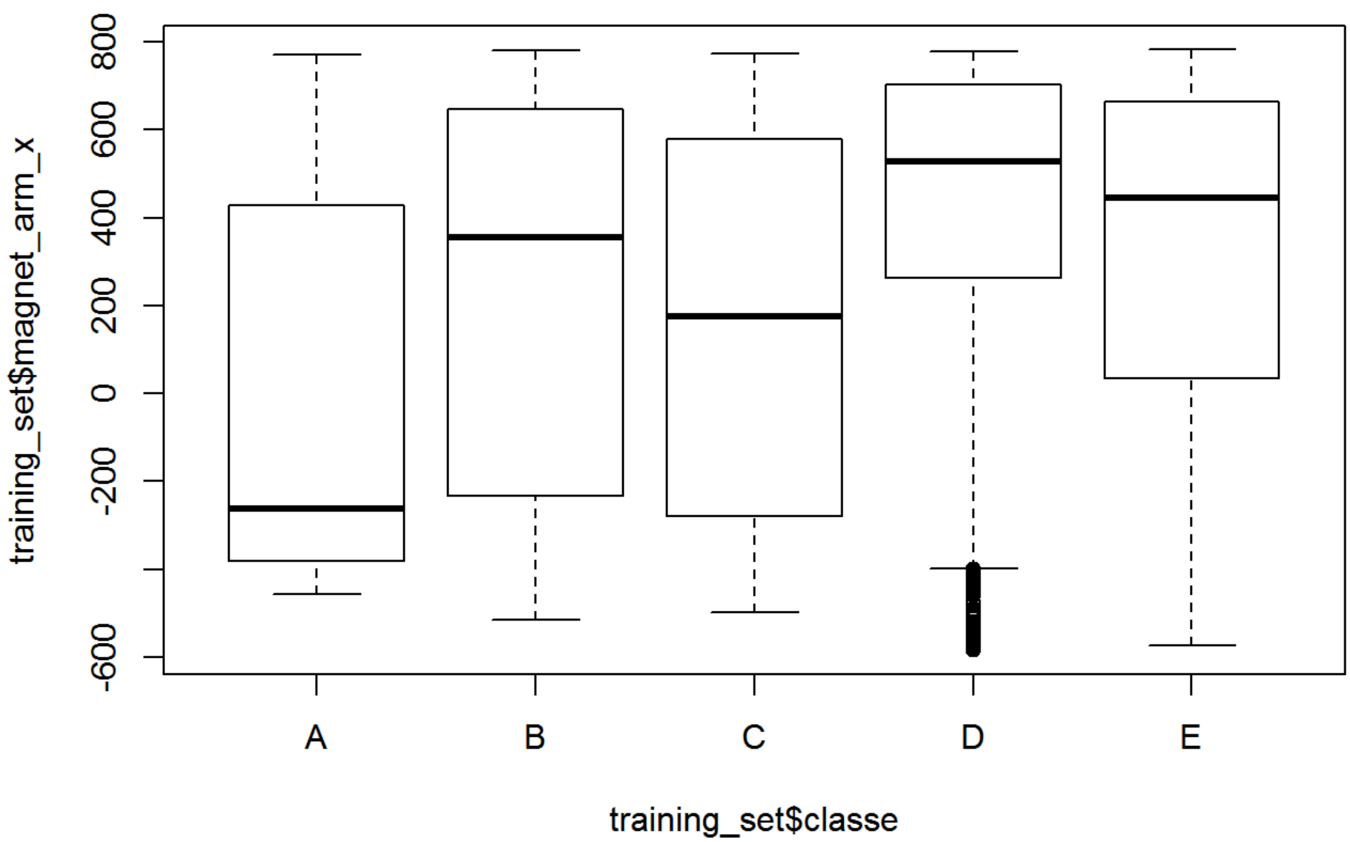
```
training <- read.csv("pml-training.csv",header = TRUE,sep = ",", quote = "\"'",na.strings = "NA",dec
= "." )
testing <- read.csv("pml-testing.csv",header = TRUE,sep = ",", quote = "\"'",na.strings = "NA",dec =
"." )
```

The first step is to remove the window summary data since it is not defined for the test step.In addition, all the none critical parameters not correlated to the activity are removed such as time, window numbers. The resulting data set has 52 numberical culomns.

```
#data preparation
index <- (training$new_window == "yes")
training_set <- training[-index,]
# integrated parameters
training_ave <- training[index,]
colremove <- c(1:7,11:36,50:59,69:83,87:101,103:112,125:139,141:150)
training_set <- training_set[,-colremove]
testing_set <- testing[,-colremove]
```

For first level analysis, a box plot of the different parameters as a function of the classe outcome can help to detect any issues with the data. As an examplem the magnet_arm_x was plaotting as a function of the classes.

```
plot(training_set$magnet_arm_x ~ training_set$classe)
```

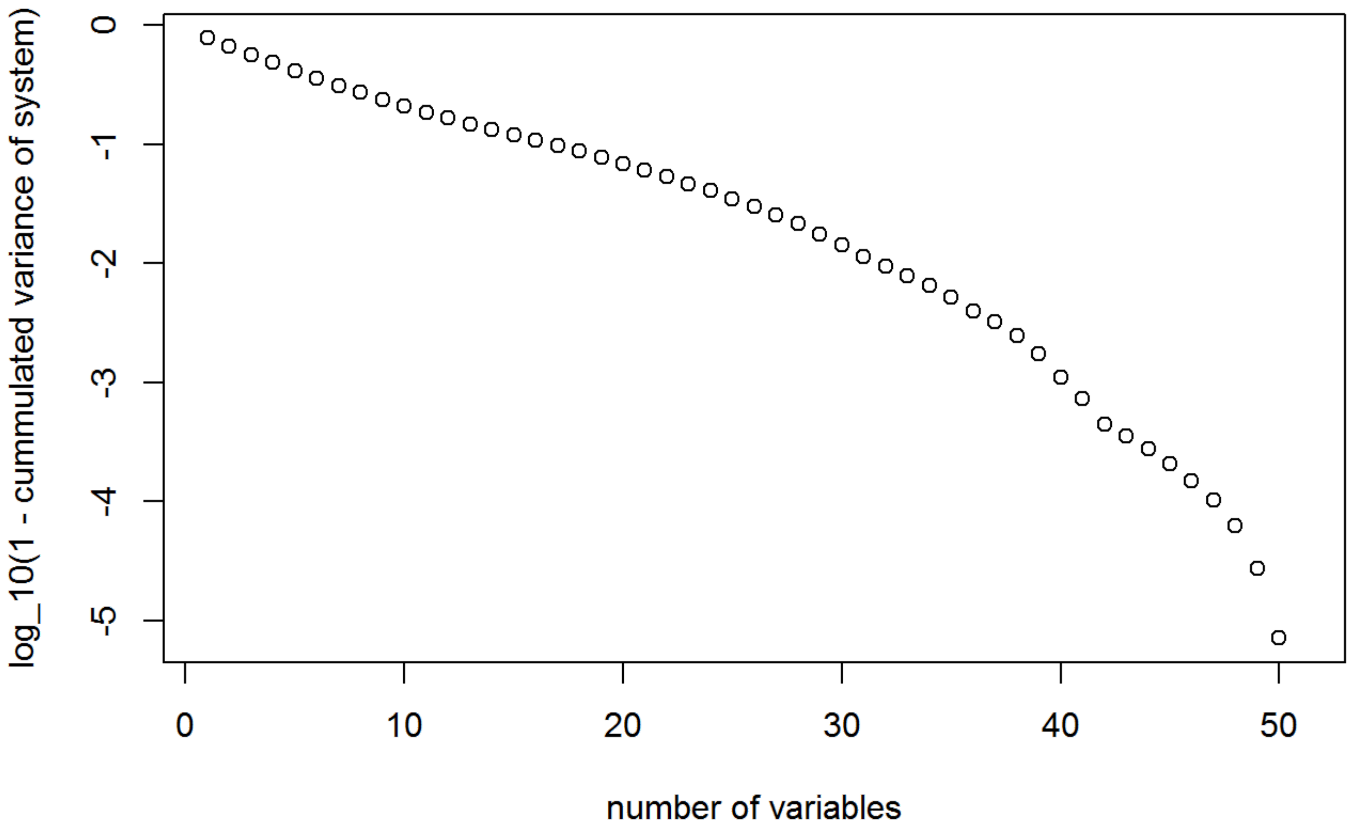


Preprocessing of data

Due to the number of numerical variables, a singular value decomposition was tried to see if we can reduce the number of regressor. the cummulated

The following plot of the residual variance as a function of the number of variables.

```
plot(log10(1.0-cummul_d), ylab = "log_10(1 - cummulated variance of system)",xlab = "number of variab
les")
```



The decay of the diagonal factor is very slow. To maintain 99% of the variance of the system, we need to include up 30 parameters. this suggests that principal component analysis method might not provide an improvement for the model. Other simple preproceesing methods were used and tested to see if the model accuracy ws improved. Unfortunately, no simple preprocessing was found effective.

Model development.

The training set is split between a training set and a test set in order to evaluate the accuracy of the modeland to pick the best model. 75% of the set is used to train the model.

The outcome is a factor so we need to choice a model for categorization. A standard linear regression model cannot be used. A CARE tree analysis was tested and seems to provide the best accurary.

```
## Loading required package: rpart
```

Different method of pre-processing the data was tested (pca, BoxCox) to see if the accuracy of the model could be improved.

verification of the accuracy of the model

The training set split is used to evaluate the accuracy of the model. This training set has seen the same preprocess. The outcome of the prediction using the training set is compared to the actual values.

```
confusionMatrix(training_t$classe,predict(modelFit,training_t))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1265   16  111    0    2
##          B  415  296  238    0    0
##          C  412   25  418    0    0
##          D  366  135  303    0    0
##          E  135  124  242    0  400
##
## Overall Statistics
##
##                Accuracy : 0.4852
##                  95% CI : (0.4711, 0.4993)
##     No Information Rate : 0.5289
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3265
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4879  0.49664  0.31860       NA  0.99502
## Specificity            0.9442  0.84839  0.87831    0.836  0.88869
## Pos Pred Value         0.9075  0.31191  0.48889       NA  0.44395
## Neg Pred Value         0.6215  0.92413  0.77915       NA  0.99950
## Prevalence             0.5289  0.12156  0.26759    0.000  0.08199
## Detection Rate         0.2580  0.06037  0.08525    0.000  0.08158
## Detection Prevalence   0.2843  0.19355  0.17438    0.164  0.18377
## Balanced Accuracy      0.7160  0.67252  0.59845       NA  0.94186
```

```
#confusionMatrix(training_t$classe,predict(modelFit,testPC))
modelFit$finalModel
```

```
## n= 14718
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 14718 10533 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 130.5 13475  9302 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -33.65 1180    10 A (0.99 0.0085 0 0 0) *
##      5) pitch_forearm>=-33.65 12295  9292 A (0.24 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 439.5 10362  7424 A (0.28 0.18 0.24 0.19 0.11)
##         20) roll_forearm< 123.5 6454  3808 A (0.41 0.18 0.18 0.17 0.06) *
##         21) roll_forearm>=123.5 3908  2599 C (0.075 0.18 0.33 0.23 0.19) *
##       11) magnet_dumbbell_y>=439.5 1933   943 B (0.034 0.51 0.043 0.22 0.19) *
##    3) roll_belt>=130.5 1243    12 E (0.0097 0 0 0 0.99) *
```

The accuracy of the model is around 50%. This is not very high. However, with 5 different categories, a random assessement would be 20%.

predicting the test set

the final step is to use our current model for predicting the values on a new set of data.

the result of the prediction is given by

```
predict(modelFit,testing_set)
```

```
##  [1] C A C A A C C A A A C C C A C A A A A C
## Levels: A B C D E
```

reference

[1] Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers’ Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6