Machine Learning Course Project

Project summary

Six participants were asked to perform barbell lifts correctly and incorrectly in five different ways. Use data from accelerometers on the belt, forearm, arm, and dumbell to predict the lift method.

http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har)

1. Load trainig data

```
library(data.table)
training <- fread("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
```

2. Remove columns that do not add to the model to reduce size of dataset. Several variables are derivatives of the other variables (max, min, avg, stddev, etc), and mostly (> 97%) NA. Remove columns with user name, time stamp, window

```
library(dplyr)
library(caret)
remove <- c("name", "timestamp", "window", "max", "min", "kurtosis", "skewness",
                  "amplitude", "var", "avg", "std", "V1")
train1 <- select(training, !contains(remove))
```

3. Check remaining predictors to see if any others can be removed

```
nearZeroVar(train1, saveMetrics = FALSE)
```

```
## integer(0)
```

outcome shows no additional removal or variables appropriate

4. Create a train and test subset of training dataset. This allows testing of accuracy of model without using the final test data

```
inTrain <- createDataPartition(y = train1$classe, p = 0.70, list = FALSE)
subset_train <- train1[inTrain,]
subset_test <- train1[-inTrain,]

dim(subset_train)
```

```
## [1] 13737    53
```

```
dim(subset_test)
```

```
## [1] 5885   53
```

5. Fit a model using boost method (gbm) on train subset.

```
modFitgbm <- train(classe ~ ., method="gbm",data=subset_train,verbose=FALSE)
modFitgbm$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 52 had non-zero influence.
```

6. Show results

```
predResultgbm <- predict(modFitgbm, subset_test)
confusionMatrix(table(subset_test$classe, predResultgbm))
```

```
## Confusion Matrix and Statistics
##
##    predResultgbm
##        A    B    C    D    E
##   A 1641   18    6    4    5
##   B   41 1064   33    1    0
##   C    0   25  988   12    1
##   D    2    3   42  909    8
##   E    6   13   10   24 1029
##
## Overall Statistics
##
##                Accuracy : 0.9568
##                  95% CI : (0.9513, 0.9619)
##     No Information Rate : 0.2872
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9454
##
##  Mcnemar's Test P-Value : 1.039e-09
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9710   0.9475   0.9157   0.9568   0.9866
## Specificity            0.9921   0.9843   0.9921   0.9889   0.9891
## Pos Pred Value         0.9803   0.9342   0.9630   0.9429   0.9510
## Neg Pred Value         0.9884   0.9876   0.9813   0.9917   0.9971
## Prevalence             0.2872   0.1908   0.1833   0.1614   0.1772
## Detection Rate         0.2788   0.1808   0.1679   0.1545   0.1749
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9816   0.9659   0.9539   0.9728   0.9878
```

Model accuracy is ```{r}confusionMatrix(table(subset_test$classe, predResultgbm))$overall]Accuracy