Practical Machine Learning Project

Yong-Hao Bai

7/10/2020

Load the required packages

```
library(caret); library(rattle); library(rpart); library(rpart.plot); library(randomForest); library(repmi
s);
library(lattice); library(ggplot2); library(readr); library(gbm)
```

Load Data

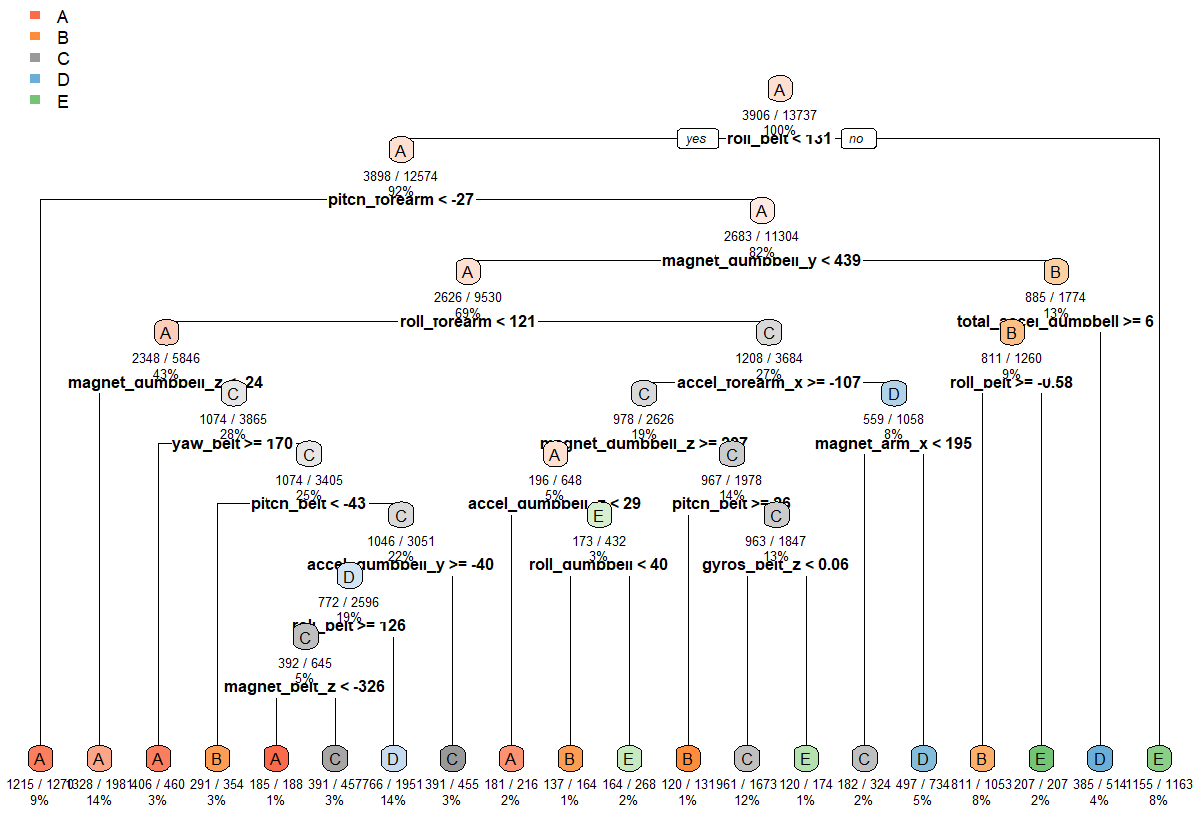
```
set.seed(717)
trainurl = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testurl = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(trainurl, "pml-training.csv")
download.file(testurl, "pml-testing.csv")
training <- read.csv("pml-training.csv",  na.strings=c("NA","#DIV/0!", ""))
testing <- read.csv("pml-testing.csv",  na.strings=c("NA","#DIV/0!", ""))
#update datasets to exclude those variables with NA values
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
```

remove irrelevant variables to the prediction

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

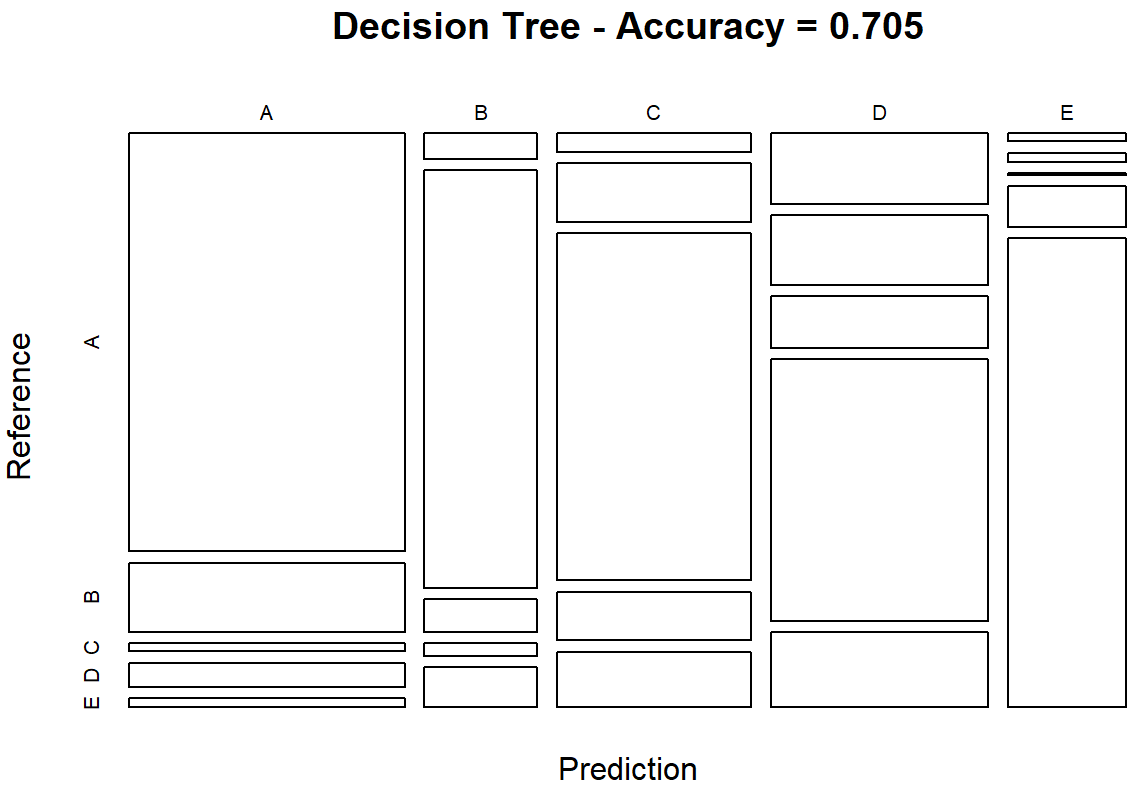
For cross validation purpose, the training data will be split into training training and training testing.

Decision Tree


Classification Tree
A
B
C
D
E
A
3906 / 13737
100%
yes
roll_belt < 131
no
A
3898 / 12574
92%
pitch_forearm < -27
A
2683 / 11304
82%
magnet_dumbbell_y < 439
A
2626 / 9530
69%
B
885 / 1774
13%
roll_forearm < 121
total_accel_dumbbell >= 6
A
2348 / 5846
43%
C
1208 / 3684
27%
B
811 / 1260
9%
magnet_dumbbell_z < 24
accel_forearm_x >= -107
roll_belt >= -0.58
C
1074 / 3865
28%
C
978 / 2626
19%
D
559 / 1058
8%
yaw_belt >= 170
magnet_dumbbell_z >= 307
magnet_arm_x < 195
C
1074 / 3405
25%
A
196 / 648
5%
C
967 / 1978
14%
pitch_belt < -43
accel_dumbbell_z < 29
pitch_belt >= 26
C
1046 / 3051
22%
E
173 / 432
3%
C
963 / 1847
13%
accel_dumbbell_y >= -40
roll_dumbbell < 40
gyros_belt_z < 0.06
D
772 / 2596
19%
roll_belt >= 126
C
392 / 645
5%
magnet_belt_z < -326
A
1215 / 1270
9%
A
328 / 1981
14%
A
406 / 460
3%
B
291 / 354
3%
A
185 / 188
1%
C
391 / 457
3%
D
766 / 1951
14%
C
391 / 455
3%
A
181 / 216
2%
B
137 / 164
1%
E
164 / 268
2%
B
120 / 131
1%
C
961 / 1673
12%
E
120 / 174
1%
C
182 / 324
2%
D
497 / 734
5%
B
811 / 1053
8%
E
207 / 207
2%
D
385 / 514
4%
E
155 / 1163
8%

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1399  232   26   81   30
##          B   35  571   44   17   55
##          C   43  139  818  115  131
##          D  186  184  136  693  198
##          E   11   13    2   58  668
##
## Overall Statistics
##
##                Accuracy : 0.705
##                  95% CI : (0.6932, 0.7166)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6273
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8357  0.50132   0.7973   0.7189   0.6174
## Specificity            0.9124  0.96818   0.9119   0.8569   0.9825
## Pos Pred Value         0.7913  0.79086   0.6565   0.4961   0.8883
## Neg Pred Value         0.9332  0.88999   0.9552   0.9396   0.9193
## Prevalence             0.2845  0.19354   0.1743   0.1638   0.1839
## Detection Rate         0.2377  0.09703   0.1390   0.1178   0.1135
## Detection Prevalence   0.3004  0.12268   0.2117   0.2374   0.1278
## Balanced Accuracy      0.8740  0.73475   0.8546   0.7879   0.7999
```

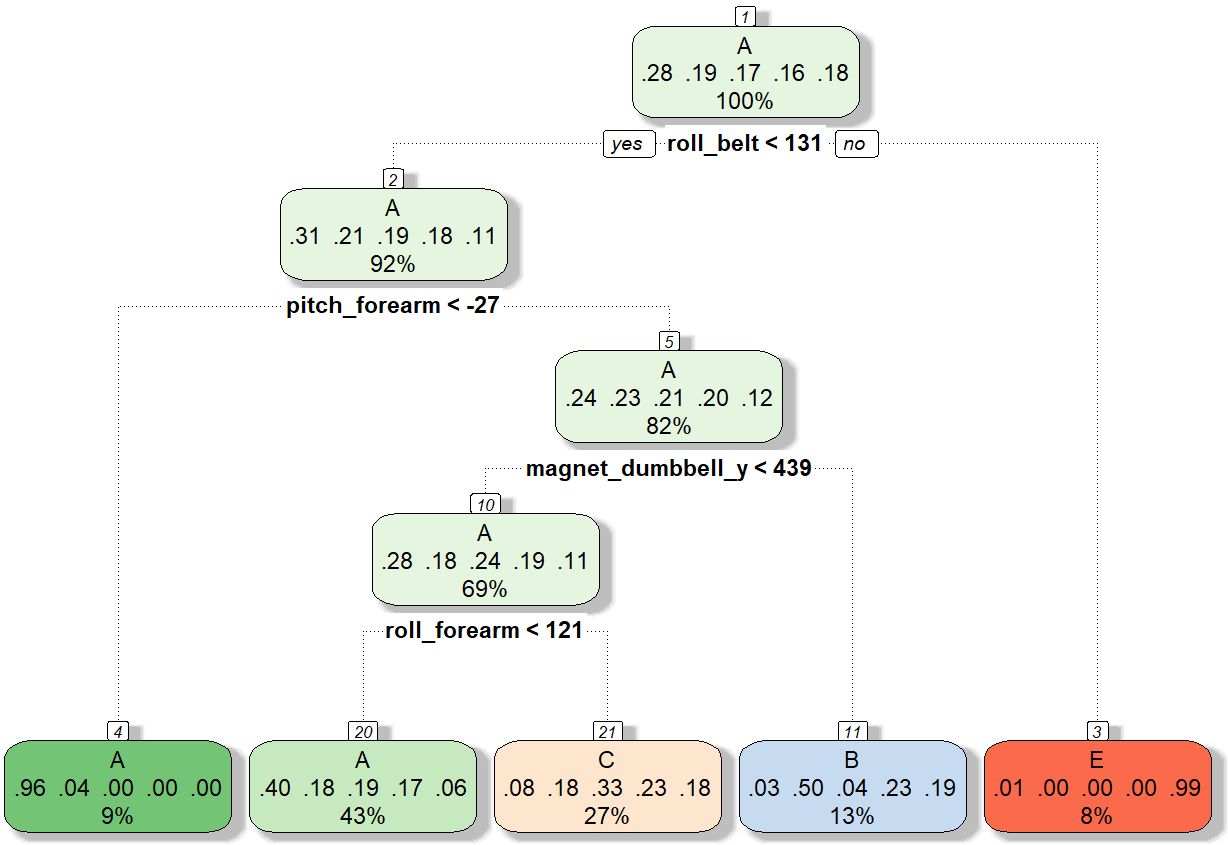
```
##  Accuracy
## 0.7050127
```



We see that the accuracy rate of the model is low: 0.7274,the out-of-sample-error is about 0.3 which is considerable.

Classification tree

```
## CART
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10990, 10991, 10991, 10987
## Resampling results across tuning parameters:
##
##   cp       Accuracy  Kappa
##   0.03347  0.5216    0.37976
##   0.05961  0.4175    0.21078
##   0.11667  0.3331    0.07417
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03347.
```



Rattle 2020-Jul-11 01:35:54 josep

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1520   27  121    0    6
##          B  487  388  264    0    0
##          C  469   34  523    0    0
##          D  423  159  382    0    0
##          E  175  146  285    0  476
##
## Overall Statistics
##
##                Accuracy : 0.494
##                  95% CI : (0.4811, 0.5068)
##     No Information Rate : 0.5223
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3384
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4945  0.51459  0.33206       NA  0.98755
## Specificity            0.9452  0.85363  0.88329   0.8362  0.88784
## Pos Pred Value         0.9080  0.34065  0.50975       NA  0.43993
## Neg Pred Value         0.6310  0.92288  0.78349       NA  0.99875
## Prevalence             0.5223  0.12812  0.26763   0.0000  0.08190
## Detection Rate         0.2583  0.06593  0.08887   0.0000  0.08088
## Detection Prevalence   0.2845  0.19354  0.17434   0.1638  0.18386
## Balanced Accuracy      0.7198  0.68411  0.60768       NA  0.93770
```

```
##  Accuracy
## 0.4939677
```

Boosted Logistic Regression

```
## Boosted Logistic Regression
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10990, 10988, 10991, 10990, 10989
## Resampling results across tuning parameters:
##
##   nIter  Accuracy   Kappa
##   11     0.8148317  0.7634940
##   21     0.8729692  0.8382176
##   31     0.8965052  0.8681682
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was nIter = 31.
```

```
##  Accuracy
## 0.8968583
```

Gradient Boosting

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

```
## Stochastic Gradient Boosting
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 10989, 10988, 10990, 10991, 10990
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7558426  0.6904061
##   1                  100      0.8240513  0.7773866
##   1                  150      0.8546979  0.8161247
##   2                   50      0.8541156  0.8152252
##   2                  100      0.9069654  0.8822838
##   2                  150      0.9302607  0.9117700
##   3                   50      0.8985940  0.8716343
##   3                  100      0.9419814  0.9265947
##   3                  150      0.9615635  0.9513768
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1654   39    0    0    6
##          B   14 1066   32    9   16
##          C    5   30  980   41   11
##          D    1    3   11  911    8
##          E    0    1    3    3 1041
##
## Overall Statistics
##
##                Accuracy : 0.9604
##                  95% CI : (0.9551, 0.9652)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9499
##
##  Mcnemar's Test P-Value : 5.653e-10
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9881   0.9359   0.9552   0.9450   0.9621
## Specificity            0.9893   0.9850   0.9821   0.9953   0.9985
## Pos Pred Value         0.9735   0.9376   0.9185   0.9754   0.9933
## Neg Pred Value         0.9952   0.9846   0.9905   0.9893   0.9915
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2811   0.1811   0.1665   0.1548   0.1769
## Detection Prevalence   0.2887   0.1932   0.1813   0.1587   0.1781
## Balanced Accuracy      0.9887   0.9605   0.9686   0.9702   0.9803
```

```
##  Accuracy
## 0.9604078
```

Random Forest

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training_train, method = "class")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.5%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    2    0    0    0 0.0005120328
## B    8 2642    8    0    0 0.0060195636
## C    0   15 2378    3    0 0.0075125209
## D    0    0   24 2225    3 0.0119893428
## E    0    0    1    5 2519 0.0023762376
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    3    0    0    0
##          B    0 1135    3    0    0
##          C    0    1 1023   11    2
##          D    1    0    0  953    0
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.9964
##                  95% CI : (0.9946, 0.9978)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9955
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9965   0.9971   0.9886   0.9982
## Specificity            0.9993   0.9994   0.9971   0.9998   1.0000
## Pos Pred Value         0.9982   0.9974   0.9865   0.9990   1.0000
## Neg Pred Value         0.9998   0.9992   0.9994   0.9978   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1929   0.1738   0.1619   0.1835
## Detection Prevalence   0.2848   0.1934   0.1762   0.1621   0.1835
## Balanced Accuracy      0.9993   0.9979   0.9971   0.9942   0.9991
```

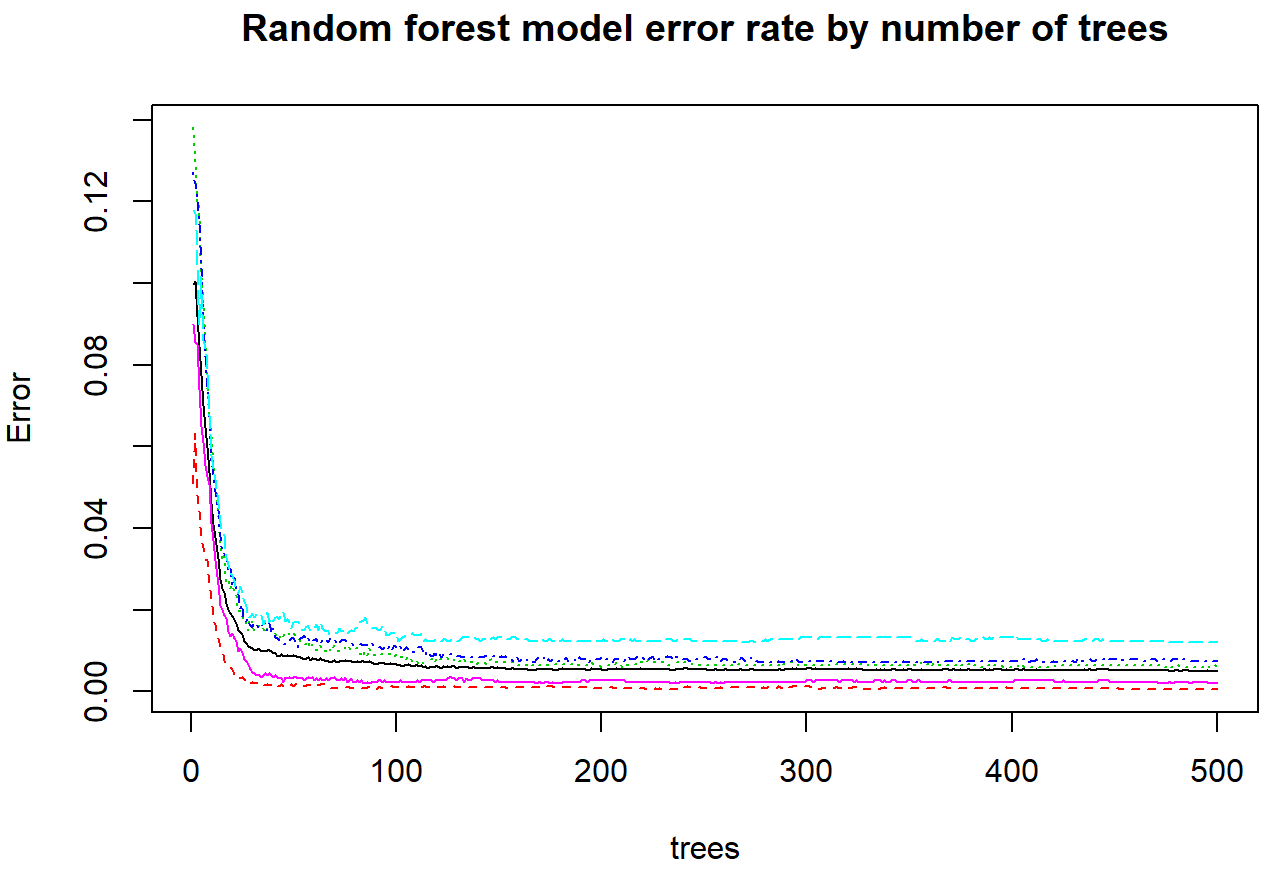
```
##  Accuracy
## 0.9964316
```

Looking at the results, clearly, the random forest model provides a more accurate prediction of classe with 0.9955 compare to decision tree's 0.7488. The expected out-of-sample error is estimated at 0.005.

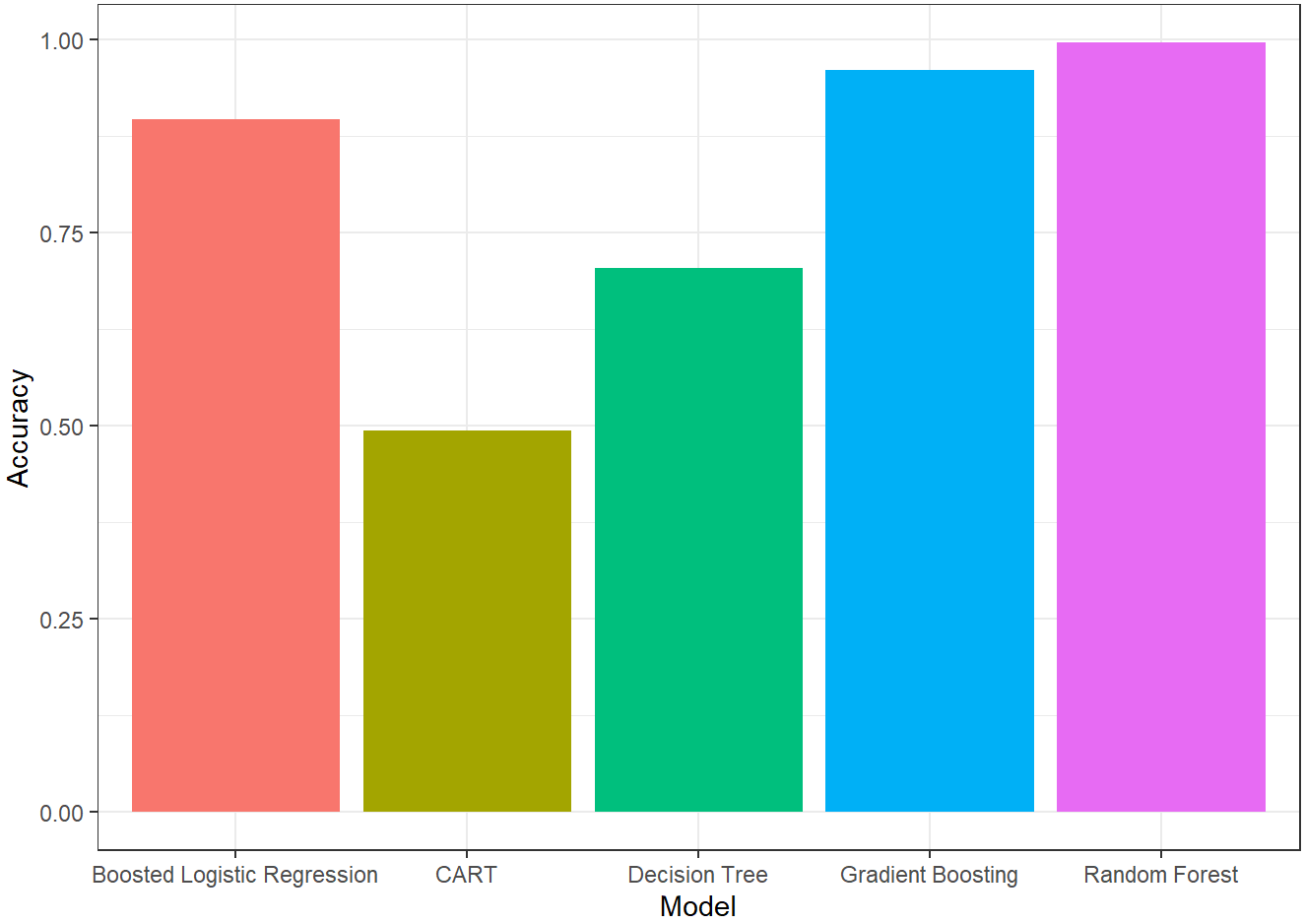
Variable Importance

```
##                        Overall
## roll_belt            876.72212
## pitch_belt           489.54802
## yaw_belt             620.44650
## total_accel_belt     148.20012
## gyros_belt_x          62.78619
## gyros_belt_y          80.70786
## gyros_belt_z         214.33133
## accel_belt_x          80.43616
## accel_belt_y          83.97932
## accel_belt_z         282.47151
## magnet_belt_x        165.65756
## magnet_belt_y        268.62704
## magnet_belt_z        286.17267
## roll_arm             228.95367
## pitch_arm            118.44882
## yaw_arm              168.12837
## total_accel_arm       72.31039
## gyros_arm_x           94.46041
## gyros_arm_y           95.15750
## gyros_arm_z           44.09700
## accel_arm_x          174.85325
## accel_arm_y          114.22182
## accel_arm_z           93.34785
## magnet_arm_x         193.92654
## magnet_arm_y         156.82041
## magnet_arm_z         134.36351
## roll_dumbbell        291.25438
## pitch_dumbbell       124.43499
## yaw_dumbbell         189.97041
## total_accel_dumbbell 190.33350
## gyros_dumbbell_x      90.88199
## gyros_dumbbell_y     179.20113
## gyros_dumbbell_z      59.99347
## accel_dumbbell_x     175.90178
## accel_dumbbell_y     289.37174
## accel_dumbbell_z     228.14040
## magnet_dumbbell_x    346.57015
## magnet_dumbbell_y    453.86872
## magnet_dumbbell_z    520.04673
## roll_forearm         396.67737
## pitch_forearm        580.58801
## yaw_forearm          115.33879
## total_accel_forearm   80.08305
## gyros_forearm_x       58.20348
## gyros_forearm_y       93.23729
## gyros_forearm_z       60.11269
## accel_forearm_x      219.47732
## accel_forearm_y       98.00102
## accel_forearm_z      162.33309
## magnet_forearm_x     161.27334
## magnet_forearm_y     154.31374
## magnet_forearm_z     191.56299
```

plot of the model error rate by number of trees and 20 most important variables (out of 52)



Accurary comparison among models



Random forest has the highest accurary.

Predictio on Testing

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```