

# Assignemnt Writeup

## Assignemnt Writeup

###Background Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: <http://groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset).

### ###Data

The training data for this project are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv>

The test data are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>

The data for this project come from this source: <http://groupware.les.inf.puc-rio.br/har>. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

### ###Goal

The goal of this project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set.

### ###Load all the required libraries

```
library(caret)
library(randomForest)
library(rpart)
library(rpart.plot)
library(rattle)
```

## Loading the data sets

```
Data_train <- read.csv("D:/Priyanka/Coursera/8.MachineLearning/pml-training.csv")
Data_test <- read.csv("D:/Priyanka/Coursera/8.MachineLearning/pml-testing.csv")
```

Partition the training into two parts

```
inTrain <- createDataPartition(Data_train$classe, p=0.7, list = FALSE)
TrainSet <- Data_train[inTrain,]
TestSet <- Data_train[-inTrain,]
dim(TrainSet)
```

```
## [1] 13737 159
```

```
dim(TestSet)
```

```
## [1] 5885 159
```

## Cleaning the data sets

Remove the variables which have negligible variability

```
nzv <- nearZeroVar(TrainSet)
TrainSet <- TrainSet[, -nzv]
TestSet <- TestSet[, -nzv]
dim(TrainSet)
```

```
## [1] 13737 104
```

```
dim(TestSet)
```

```
## [1] 5885 104
```

Remove the variables which has NA for more than 95% of the observations

```
AllNA <- sapply(TrainSet, function(x) mean(is.na(x))) > 0.95
TrainSet <- TrainSet[, AllNA == FALSE]
TestSet <- TestSet[, AllNA == FALSE]
dim(TrainSet)
```

```
## [1] 13737 58
```

```
dim(TestSet)
```

```
## [1] 5885 58
```

Remove the variables which behaves as an identifier only

```
TrainSet <- TrainSet[,-(1:4)]
```

```
TestSet <- TestSet[,-(1:4)]
```

```
dim(TrainSet)
```

```
## [1] 13737 54
```

```
dim(TestSet)
```

```
## [1] 5885 54
```

## Modelling starts

### Random Forest

```
ControlRF <- trainControl(method = "cv", number = 3)
```

```
m_rf <- train(classe ~., data = TrainSet, method = "rf", trControl = ControlRF)
```

```
m_rf$finalModel
```

```
##
```

```
## Call:
```

```
## randomForest(x = x, y = y, mtry = param$mtry)
```

```
## Type of random forest: classification
```

```
## Number of trees: 500
```

```
## No. of variables tried at each split: 27
```

```
##
```

```
## OOB estimate of error rate: 0.17%
```

```
## Confusion matrix:
```

```
## A B C D E class.error
```

```
## A 3904 1 0 0 1 0.0005120328
```

```
## B 5 2651 2 0 0 0.0026335591
```

```
## C 0 7 2389 0 0 0.0029215359
```

```
## D 0 0 2 2249 1 0.0013321492
```

```
## E 0 1 0 4 2520 0.0019801980
```

```
pred_rf <- predict(m_rf, TestSet)
```

```
conf_mat_rf <- confusionMatrix(pred_rf, TestSet$classe)
```

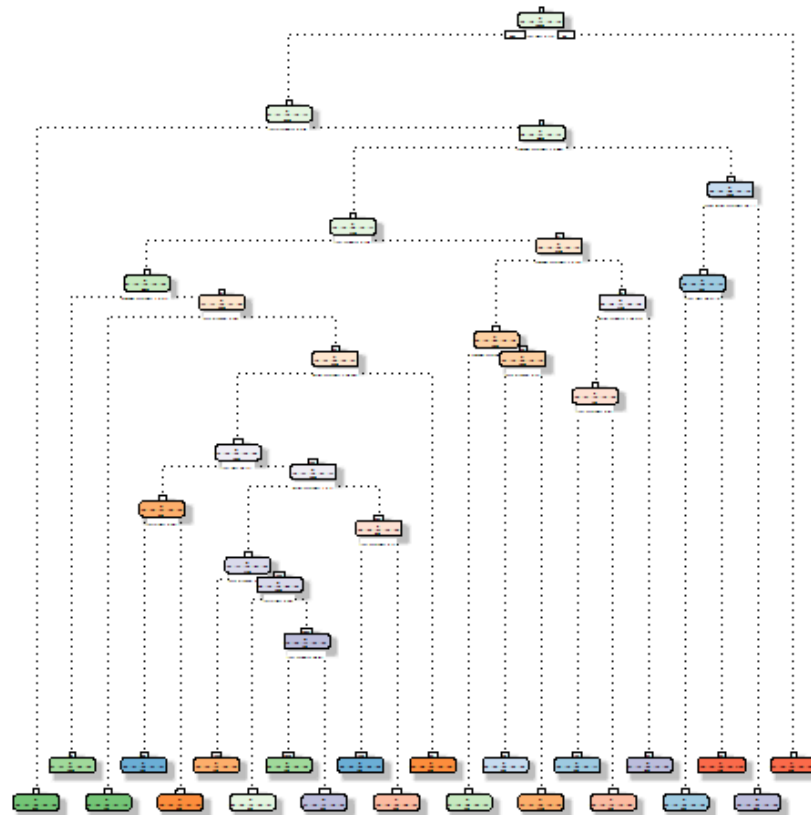
```
conf_mat_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A    B    C    D    E
##           A 1674    2    0    0    0
##           B    0 1132    2    0    0
##           C    0    5 1024    4    0
##           D    0    0    0  960    3
##           E    0    0    0    0 1079
##
## Overall Statistics
##
##           Accuracy : 0.9973
##           95% CI : (0.9956, 0.9984)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9966
##           McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity      1.0000  0.9939  0.9981  0.9959  0.9972
## Specificity      0.9995  0.9996  0.9981  0.9994  1.0000
## Pos Pred Value   0.9988  0.9982  0.9913  0.9969  1.0000
## Neg Pred Value   1.0000  0.9985  0.9996  0.9992  0.9994
## Prevalence       0.2845  0.1935  0.1743  0.1638  0.1839
## Detection Rate   0.2845  0.1924  0.1740  0.1631  0.1833
## Detection Prevalence 0.2848  0.1927  0.1755  0.1636  0.1833
## Balanced Accuracy 0.9998  0.9967  0.9981  0.9976  0.9986
```

## Decision Tree

```
m_dt <- rpart(classe ~., data = TrainSet, method = "class")
fancyRpartPlot(m_dt)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2017-May-18 19:38:40 RKJ51882

```
pred_dt <- predict(m_dt, TestSet, type ="class")
conf_mat_dt <- confusionMatrix(pred_dt, TestSet$classe)
conf_mat_dt
```

## Confusion Matrix and Statistics

##

##           Reference

## Prediction    A    B    C    D    E

##           A 1553 286 44 112 94

##           B 53 621 41 24 92

##           C 10 82 805 138 65

##           D 40 70 62 581 52

##           E 18 80 74 109 779

##

## Overall Statistics

```
##
##          Accuracy : 0.7373
##          95% CI : (0.7259, 0.7485)
##      No Information Rate : 0.2845
##      P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.6647
##      McNemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##          Class: A Class: B Class: C Class: D Class: E
## Sensitivity      0.9277  0.5452  0.7846  0.60270  0.7200
## Specificity      0.8727  0.9558  0.9393  0.95448  0.9415
## Pos Pred Value   0.7434  0.7473  0.7318  0.72174  0.7349
## Neg Pred Value   0.9681  0.8975  0.9538  0.92461  0.9372
## Prevalence       0.2845  0.1935  0.1743  0.16381  0.1839
## Detection Rate   0.2639  0.1055  0.1368  0.09873  0.1324
## Detection Prevalence 0.3550  0.1412  0.1869  0.13679  0.1801
## Balanced Accuracy 0.9002  0.7505  0.8619  0.77859  0.8307
```

## Gradient Boosting Modelling

```
ControlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 2)
m_gbm <- train(classe ~., data = TrainSet, method = "gbm", trControl = ControlGBM)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1           1.6094             nan      0.1000    0.1305
##      2           1.5217             nan      0.1000    0.0888
##      3           1.4628             nan      0.1000    0.0646
##      4           1.4183             nan      0.1000    0.0563
##      5           1.3824             nan      0.1000    0.0519
##      6           1.3491             nan      0.1000    0.0407
##      7           1.3232             nan      0.1000    0.0394
##      8           1.2974             nan      0.1000    0.0348
##      9           1.2742             nan      0.1000    0.0353
##     10           1.2513             nan      0.1000    0.0322
##     20           1.0864             nan      0.1000    0.0170
##     40           0.9039             nan      0.1000    0.0103
##     60           0.7915             nan      0.1000    0.0063
##     80           0.7112             nan      0.1000    0.0048
##    100           0.6444             nan      0.1000    0.0044
##    120           0.5909             nan      0.1000    0.0029
##    140           0.5458             nan      0.1000    0.0030
##    150           0.5250             nan      0.1000    0.0020
```

```

##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.1902
##      2          1.4838          nan        0.1000    0.1362
##      3          1.3969          nan        0.1000    0.1060
##      4          1.3287          nan        0.1000    0.0901
##      5          1.2725          nan        0.1000    0.0770
##      6          1.2234          nan        0.1000    0.0591
##      7          1.1844          nan        0.1000    0.0626
##      8          1.1447          nan        0.1000    0.0600
##      9          1.1078          nan        0.1000    0.0507
##     10          1.0770          nan        0.1000    0.0458
##     20          0.8564          nan        0.1000    0.0249
##     40          0.6180          nan        0.1000    0.0135
##     60          0.4870          nan        0.1000    0.0046
##     80          0.3945          nan        0.1000    0.0061
##    100          0.3304          nan        0.1000    0.0053
##    120          0.2741          nan        0.1000    0.0035
##    140          0.2325          nan        0.1000    0.0014
##    150          0.2153          nan        0.1000    0.0021
##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.2495
##      2          1.4545          nan        0.1000    0.1684
##      3          1.3477          nan        0.1000    0.1211
##      4          1.2690          nan        0.1000    0.1115
##      5          1.1985          nan        0.1000    0.0966
##      6          1.1377          nan        0.1000    0.0839
##      7          1.0851          nan        0.1000    0.0640
##      8          1.0429          nan        0.1000    0.0788
##      9          0.9951          nan        0.1000    0.0551
##     10          0.9603          nan        0.1000    0.0595
##     20          0.6824          nan        0.1000    0.0254
##     40          0.4476          nan        0.1000    0.0091
##     60          0.3265          nan        0.1000    0.0081
##     80          0.2475          nan        0.1000    0.0039
##    100          0.1884          nan        0.1000    0.0015
##    120          0.1492          nan        0.1000    0.0033
##    140          0.1215          nan        0.1000    0.0014
##    150          0.1100          nan        0.1000    0.0017
##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.1236
##      2          1.5245          nan        0.1000    0.0880
##      3          1.4672          nan        0.1000    0.0652
##      4          1.4239          nan        0.1000    0.0533

```

##	5	1.3887	nan	0.1000	0.0494
##	6	1.3561	nan	0.1000	0.0418
##	7	1.3295	nan	0.1000	0.0399
##	8	1.3047	nan	0.1000	0.0311
##	9	1.2835	nan	0.1000	0.0363
##	10	1.2585	nan	0.1000	0.0349
##	20	1.0922	nan	0.1000	0.0177
##	40	0.9085	nan	0.1000	0.0087
##	60	0.7981	nan	0.1000	0.0079
##	80	0.7146	nan	0.1000	0.0042
##	100	0.6522	nan	0.1000	0.0039
##	120	0.5981	nan	0.1000	0.0031
##	140	0.5507	nan	0.1000	0.0025
##	150	0.5299	nan	0.1000	0.0029
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1930
##	2	1.4843	nan	0.1000	0.1282
##	3	1.4017	nan	0.1000	0.1072
##	4	1.3342	nan	0.1000	0.0849
##	5	1.2791	nan	0.1000	0.0726
##	6	1.2317	nan	0.1000	0.0637
##	7	1.1900	nan	0.1000	0.0671
##	8	1.1479	nan	0.1000	0.0541
##	9	1.1121	nan	0.1000	0.0505
##	10	1.0809	nan	0.1000	0.0418
##	20	0.8527	nan	0.1000	0.0190
##	40	0.6262	nan	0.1000	0.0171
##	60	0.4808	nan	0.1000	0.0096
##	80	0.3922	nan	0.1000	0.0034
##	100	0.3308	nan	0.1000	0.0031
##	120	0.2800	nan	0.1000	0.0027
##	140	0.2391	nan	0.1000	0.0033
##	150	0.2208	nan	0.1000	0.0019
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2333
##	2	1.4587	nan	0.1000	0.1670
##	3	1.3513	nan	0.1000	0.1306
##	4	1.2676	nan	0.1000	0.1057
##	5	1.1995	nan	0.1000	0.0907
##	6	1.1420	nan	0.1000	0.0843
##	7	1.0891	nan	0.1000	0.0611
##	8	1.0493	nan	0.1000	0.0758
##	9	1.0034	nan	0.1000	0.0664
##	10	0.9634	nan	0.1000	0.0546



##	20	0.6901	nan	0.1000	0.0212
##	40	0.4561	nan	0.1000	0.0153
##	60	0.3289	nan	0.1000	0.0092
##	80	0.2478	nan	0.1000	0.0058
##	100	0.1964	nan	0.1000	0.0030
##	120	0.1520	nan	0.1000	0.0016
##	140	0.1248	nan	0.1000	0.0015
##	150	0.1120	nan	0.1000	0.0014
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1299
##	2	1.5221	nan	0.1000	0.0885
##	3	1.4624	nan	0.1000	0.0670
##	4	1.4184	nan	0.1000	0.0552
##	5	1.3823	nan	0.1000	0.0450
##	6	1.3520	nan	0.1000	0.0416
##	7	1.3246	nan	0.1000	0.0451
##	8	1.2974	nan	0.1000	0.0371
##	9	1.2719	nan	0.1000	0.0348
##	10	1.2498	nan	0.1000	0.0321
##	20	1.0833	nan	0.1000	0.0180
##	40	0.8992	nan	0.1000	0.0091
##	60	0.7873	nan	0.1000	0.0055
##	80	0.7065	nan	0.1000	0.0043
##	100	0.6412	nan	0.1000	0.0031
##	120	0.5881	nan	0.1000	0.0024
##	140	0.5415	nan	0.1000	0.0027
##	150	0.5205	nan	0.1000	0.0013
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1942
##	2	1.4840	nan	0.1000	0.1376
##	3	1.3953	nan	0.1000	0.1080
##	4	1.3260	nan	0.1000	0.0848
##	5	1.2700	nan	0.1000	0.0695
##	6	1.2246	nan	0.1000	0.0820
##	7	1.1748	nan	0.1000	0.0583
##	8	1.1369	nan	0.1000	0.0564
##	9	1.1016	nan	0.1000	0.0429
##	10	1.0731	nan	0.1000	0.0461
##	20	0.8453	nan	0.1000	0.0233
##	40	0.6177	nan	0.1000	0.0121
##	60	0.4749	nan	0.1000	0.0062
##	80	0.3926	nan	0.1000	0.0049
##	100	0.3251	nan	0.1000	0.0044
##	120	0.2728	nan	0.1000	0.0022

```

##      140      0.2320      nan      0.1000      0.0021
##      150      0.2109      nan      0.1000      0.0026
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1.6094      nan      0.1000      0.2440
##      2      1.4545      nan      0.1000      0.1667
##      3      1.3466      nan      0.1000      0.1230
##      4      1.2681      nan      0.1000      0.1195
##      5      1.1934      nan      0.1000      0.0911
##      6      1.1379      nan      0.1000      0.0726
##      7      1.0907      nan      0.1000      0.0788
##      8      1.0414      nan      0.1000      0.0670
##      9      1.0000      nan      0.1000      0.0664
##     10      0.9590      nan      0.1000      0.0673
##     20      0.6848      nan      0.1000      0.0274
##     40      0.4480      nan      0.1000      0.0099
##     60      0.3323      nan      0.1000      0.0042
##     80      0.2498      nan      0.1000      0.0031
##    100      0.1951      nan      0.1000      0.0018
##    120      0.1560      nan      0.1000      0.0021
##    140      0.1266      nan      0.1000      0.0011
##    150      0.1144      nan      0.1000      0.0016
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1.6094      nan      0.1000      0.1287
##      2      1.5247      nan      0.1000      0.0852
##      3      1.4675      nan      0.1000      0.0665
##      4      1.4228      nan      0.1000      0.0532
##      5      1.3880      nan      0.1000      0.0503
##      6      1.3548      nan      0.1000      0.0410
##      7      1.3279      nan      0.1000      0.0401
##      8      1.3020      nan      0.1000      0.0331
##      9      1.2802      nan      0.1000      0.0295
##     10      1.2597      nan      0.1000      0.0338
##     20      1.0929      nan      0.1000      0.0163
##     40      0.9105      nan      0.1000      0.0106
##     60      0.7965      nan      0.1000      0.0061
##     80      0.7127      nan      0.1000      0.0045
##    100      0.6491      nan      0.1000      0.0035
##    120      0.5952      nan      0.1000      0.0036
##    140      0.5483      nan      0.1000      0.0025
##    150      0.5265      nan      0.1000      0.0027
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1.6094      nan      0.1000      0.1840
##      2      1.4885      nan      0.1000      0.1370

```

##	3	1.4014	nan	0.1000	0.1048
##	4	1.3337	nan	0.1000	0.0825
##	5	1.2796	nan	0.1000	0.0754
##	6	1.2324	nan	0.1000	0.0752
##	7	1.1857	nan	0.1000	0.0539
##	8	1.1505	nan	0.1000	0.0484
##	9	1.1187	nan	0.1000	0.0496
##	10	1.0872	nan	0.1000	0.0480
##	20	0.8515	nan	0.1000	0.0253
##	40	0.6261	nan	0.1000	0.0146
##	60	0.4917	nan	0.1000	0.0058
##	80	0.3971	nan	0.1000	0.0051
##	100	0.3247	nan	0.1000	0.0036
##	120	0.2767	nan	0.1000	0.0028
##	140	0.2339	nan	0.1000	0.0021
##	150	0.2167	nan	0.1000	0.0018
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2334
##	2	1.4609	nan	0.1000	0.1622
##	3	1.3578	nan	0.1000	0.1343
##	4	1.2743	nan	0.1000	0.1062
##	5	1.2059	nan	0.1000	0.0987
##	6	1.1449	nan	0.1000	0.0837
##	7	1.0935	nan	0.1000	0.0674
##	8	1.0493	nan	0.1000	0.0610
##	9	1.0108	nan	0.1000	0.0704
##	10	0.9684	nan	0.1000	0.0484
##	20	0.7021	nan	0.1000	0.0349
##	40	0.4579	nan	0.1000	0.0106
##	60	0.3327	nan	0.1000	0.0077
##	80	0.2524	nan	0.1000	0.0044
##	100	0.1951	nan	0.1000	0.0034
##	120	0.1541	nan	0.1000	0.0037
##	140	0.1241	nan	0.1000	0.0013
##	150	0.1122	nan	0.1000	0.0018
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1266
##	2	1.5246	nan	0.1000	0.0850
##	3	1.4678	nan	0.1000	0.0663
##	4	1.4229	nan	0.1000	0.0545
##	5	1.3871	nan	0.1000	0.0493
##	6	1.3538	nan	0.1000	0.0430
##	7	1.3258	nan	0.1000	0.0364
##	8	1.3020	nan	0.1000	0.0330

##	9	1.2802	nan	0.1000	0.0369
##	10	1.2553	nan	0.1000	0.0326
##	20	1.0919	nan	0.1000	0.0208
##	40	0.9061	nan	0.1000	0.0097
##	60	0.7948	nan	0.1000	0.0075
##	80	0.7117	nan	0.1000	0.0054
##	100	0.6453	nan	0.1000	0.0037
##	120	0.5943	nan	0.1000	0.0036
##	140	0.5478	nan	0.1000	0.0028
##	150	0.5266	nan	0.1000	0.0018
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1881
##	2	1.4855	nan	0.1000	0.1293
##	3	1.3986	nan	0.1000	0.1055
##	4	1.3295	nan	0.1000	0.0833
##	5	1.2750	nan	0.1000	0.0766
##	6	1.2248	nan	0.1000	0.0636
##	7	1.1837	nan	0.1000	0.0667
##	8	1.1424	nan	0.1000	0.0553
##	9	1.1073	nan	0.1000	0.0441
##	10	1.0783	nan	0.1000	0.0388
##	20	0.8531	nan	0.1000	0.0213
##	40	0.6202	nan	0.1000	0.0139
##	60	0.4890	nan	0.1000	0.0114
##	80	0.3964	nan	0.1000	0.0073
##	100	0.3265	nan	0.1000	0.0046
##	120	0.2737	nan	0.1000	0.0027
##	140	0.2298	nan	0.1000	0.0011
##	150	0.2123	nan	0.1000	0.0012
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2403
##	2	1.4579	nan	0.1000	0.1714
##	3	1.3498	nan	0.1000	0.1281
##	4	1.2687	nan	0.1000	0.1056
##	5	1.2005	nan	0.1000	0.0877
##	6	1.1443	nan	0.1000	0.0824
##	7	1.0920	nan	0.1000	0.0767
##	8	1.0449	nan	0.1000	0.0710
##	9	1.0001	nan	0.1000	0.0656
##	10	0.9595	nan	0.1000	0.0601
##	20	0.6930	nan	0.1000	0.0271
##	40	0.4494	nan	0.1000	0.0136
##	60	0.3260	nan	0.1000	0.0075
##	80	0.2540	nan	0.1000	0.0048

```

##      100      0.1995      nan      0.1000      0.0018
##      120      0.1585      nan      0.1000      0.0017
##      140      0.1258      nan      0.1000      0.0015
##      150      0.1126      nan      0.1000      0.0008
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1.6094      nan      0.1000      0.1296
##      2      1.5233      nan      0.1000      0.0883
##      3      1.4646      nan      0.1000      0.0661
##      4      1.4217      nan      0.1000      0.0554
##      5      1.3846      nan      0.1000      0.0493
##      6      1.3517      nan      0.1000      0.0408
##      7      1.3258      nan      0.1000      0.0420
##      8      1.2992      nan      0.1000      0.0317
##      9      1.2787      nan      0.1000      0.0366
##     10      1.2526      nan      0.1000      0.0356
##     20      1.0896      nan      0.1000      0.0176
##     40      0.9063      nan      0.1000      0.0103
##     60      0.7940      nan      0.1000      0.0061
##     80      0.7127      nan      0.1000      0.0042
##    100      0.6467      nan      0.1000      0.0038
##    120      0.5921      nan      0.1000      0.0039
##    140      0.5455      nan      0.1000      0.0035
##    150      0.5242      nan      0.1000      0.0029
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      1.6094      nan      0.1000      0.1871
##      2      1.4865      nan      0.1000      0.1318
##      3      1.3991      nan      0.1000      0.1062
##      4      1.3310      nan      0.1000      0.0861
##      5      1.2749      nan      0.1000      0.0760
##      6      1.2264      nan      0.1000      0.0762
##      7      1.1787      nan      0.1000      0.0535
##      8      1.1434      nan      0.1000      0.0597
##      9      1.1067      nan      0.1000      0.0544
##     10      1.0724      nan      0.1000      0.0469
##     20      0.8618      nan      0.1000      0.0233
##     40      0.6172      nan      0.1000      0.0158
##     60      0.4827      nan      0.1000      0.0053
##     80      0.3884      nan      0.1000      0.0097
##    100      0.3222      nan      0.1000      0.0041
##    120      0.2714      nan      0.1000      0.0045
##    140      0.2291      nan      0.1000      0.0029
##    150      0.2121      nan      0.1000      0.0018
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve

```

##	1	1.6094	nan	0.1000	0.2423
##	2	1.4574	nan	0.1000	0.1714
##	3	1.3492	nan	0.1000	0.1186
##	4	1.2725	nan	0.1000	0.1003
##	5	1.2074	nan	0.1000	0.1048
##	6	1.1434	nan	0.1000	0.0903
##	7	1.0872	nan	0.1000	0.0657
##	8	1.0452	nan	0.1000	0.0518
##	9	1.0111	nan	0.1000	0.0635
##	10	0.9715	nan	0.1000	0.0705
##	20	0.6942	nan	0.1000	0.0283
##	40	0.4494	nan	0.1000	0.0085
##	60	0.3250	nan	0.1000	0.0066
##	80	0.2520	nan	0.1000	0.0053
##	100	0.1942	nan	0.1000	0.0030
##	120	0.1525	nan	0.1000	0.0029
##	140	0.1238	nan	0.1000	0.0022
##	150	0.1108	nan	0.1000	0.0010
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1262
##	2	1.5243	nan	0.1000	0.0859
##	3	1.4664	nan	0.1000	0.0654
##	4	1.4238	nan	0.1000	0.0513
##	5	1.3890	nan	0.1000	0.0438
##	6	1.3595	nan	0.1000	0.0387
##	7	1.3329	nan	0.1000	0.0447
##	8	1.3055	nan	0.1000	0.0405
##	9	1.2793	nan	0.1000	0.0368
##	10	1.2531	nan	0.1000	0.0284
##	20	1.0915	nan	0.1000	0.0193
##	40	0.9056	nan	0.1000	0.0093
##	60	0.7977	nan	0.1000	0.0048
##	80	0.7163	nan	0.1000	0.0054
##	100	0.6502	nan	0.1000	0.0050
##	120	0.5953	nan	0.1000	0.0021
##	140	0.5503	nan	0.1000	0.0024
##	150	0.5291	nan	0.1000	0.0029
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1884
##	2	1.4880	nan	0.1000	0.1281
##	3	1.4044	nan	0.1000	0.1068
##	4	1.3375	nan	0.1000	0.0841
##	5	1.2831	nan	0.1000	0.0740
##	6	1.2363	nan	0.1000	0.0736

##	7	1.1899	nan	0.1000	0.0599
##	8	1.1517	nan	0.1000	0.0531
##	9	1.1173	nan	0.1000	0.0591
##	10	1.0804	nan	0.1000	0.0499
##	20	0.8574	nan	0.1000	0.0242
##	40	0.6194	nan	0.1000	0.0119
##	60	0.4880	nan	0.1000	0.0122
##	80	0.3886	nan	0.1000	0.0053
##	100	0.3257	nan	0.1000	0.0045
##	120	0.2730	nan	0.1000	0.0027
##	140	0.2301	nan	0.1000	0.0022
##	150	0.2130	nan	0.1000	0.0019
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2334
##	2	1.4578	nan	0.1000	0.1669
##	3	1.3534	nan	0.1000	0.1327
##	4	1.2694	nan	0.1000	0.1032
##	5	1.2032	nan	0.1000	0.0880
##	6	1.1458	nan	0.1000	0.0892
##	7	1.0891	nan	0.1000	0.0747
##	8	1.0420	nan	0.1000	0.0796
##	9	0.9932	nan	0.1000	0.0609
##	10	0.9548	nan	0.1000	0.0623
##	20	0.6935	nan	0.1000	0.0201
##	40	0.4515	nan	0.1000	0.0124
##	60	0.3309	nan	0.1000	0.0065
##	80	0.2475	nan	0.1000	0.0072
##	100	0.1910	nan	0.1000	0.0020
##	120	0.1509	nan	0.1000	0.0016
##	140	0.1221	nan	0.1000	0.0010
##	150	0.1105	nan	0.1000	0.0007
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1238
##	2	1.5233	nan	0.1000	0.0840
##	3	1.4662	nan	0.1000	0.0663
##	4	1.4212	nan	0.1000	0.0559
##	5	1.3853	nan	0.1000	0.0495
##	6	1.3533	nan	0.1000	0.0411
##	7	1.3265	nan	0.1000	0.0363
##	8	1.3035	nan	0.1000	0.0350
##	9	1.2813	nan	0.1000	0.0351
##	10	1.2592	nan	0.1000	0.0390
##	20	1.0872	nan	0.1000	0.0190
##	40	0.9021	nan	0.1000	0.0099

##	60	0.7911	nan	0.1000	0.0068
##	80	0.7066	nan	0.1000	0.0035
##	100	0.6443	nan	0.1000	0.0036
##	120	0.5904	nan	0.1000	0.0039
##	140	0.5437	nan	0.1000	0.0024
##	150	0.5218	nan	0.1000	0.0021
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1966
##	2	1.4848	nan	0.1000	0.1283
##	3	1.4012	nan	0.1000	0.1130
##	4	1.3292	nan	0.1000	0.0875
##	5	1.2725	nan	0.1000	0.0777
##	6	1.2229	nan	0.1000	0.0753
##	7	1.1753	nan	0.1000	0.0576
##	8	1.1380	nan	0.1000	0.0558
##	9	1.1035	nan	0.1000	0.0451
##	10	1.0738	nan	0.1000	0.0431
##	20	0.8397	nan	0.1000	0.0221
##	40	0.6171	nan	0.1000	0.0087
##	60	0.4808	nan	0.1000	0.0081
##	80	0.3925	nan	0.1000	0.0048
##	100	0.3272	nan	0.1000	0.0045
##	120	0.2754	nan	0.1000	0.0027
##	140	0.2343	nan	0.1000	0.0017
##	150	0.2147	nan	0.1000	0.0017
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2409
##	2	1.4586	nan	0.1000	0.1667
##	3	1.3528	nan	0.1000	0.1336
##	4	1.2684	nan	0.1000	0.1014
##	5	1.2042	nan	0.1000	0.0883
##	6	1.1478	nan	0.1000	0.0908
##	7	1.0915	nan	0.1000	0.0750
##	8	1.0436	nan	0.1000	0.0744
##	9	0.9971	nan	0.1000	0.0652
##	10	0.9572	nan	0.1000	0.0532
##	20	0.6856	nan	0.1000	0.0199
##	40	0.4456	nan	0.1000	0.0110
##	60	0.3310	nan	0.1000	0.0082
##	80	0.2514	nan	0.1000	0.0073
##	100	0.1967	nan	0.1000	0.0041
##	120	0.1573	nan	0.1000	0.0017
##	140	0.1263	nan	0.1000	0.0030
##	150	0.1136	nan	0.1000	0.0010



```

##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.1286
##      2          1.5232          nan        0.1000    0.0857
##      3          1.4644          nan        0.1000    0.0661
##      4          1.4199          nan        0.1000    0.0542
##      5          1.3839          nan        0.1000    0.0492
##      6          1.3517          nan        0.1000    0.0391
##      7          1.3259          nan        0.1000    0.0387
##      8          1.3010          nan        0.1000    0.0318
##      9          1.2801          nan        0.1000    0.0364
##     10          1.2559          nan        0.1000    0.0330
##     20          1.0909          nan        0.1000    0.0193
##     40          0.9083          nan        0.1000    0.0085
##     60          0.7962          nan        0.1000    0.0071
##     80          0.7165          nan        0.1000    0.0043
##    100          0.6520          nan        0.1000    0.0034
##    120          0.5970          nan        0.1000    0.0033
##    140          0.5507          nan        0.1000    0.0041
##    150          0.5278          nan        0.1000    0.0024
##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.1920
##      2          1.4855          nan        0.1000    0.1385
##      3          1.3973          nan        0.1000    0.1037
##      4          1.3306          nan        0.1000    0.0869
##      5          1.2745          nan        0.1000    0.0724
##      6          1.2281          nan        0.1000    0.0694
##      7          1.1834          nan        0.1000    0.0630
##      8          1.1439          nan        0.1000    0.0566
##      9          1.1081          nan        0.1000    0.0487
##     10          1.0770          nan        0.1000    0.0432
##     20          0.8496          nan        0.1000    0.0238
##     40          0.6235          nan        0.1000    0.0143
##     60          0.4897          nan        0.1000    0.0060
##     80          0.3982          nan        0.1000    0.0059
##    100          0.3261          nan        0.1000    0.0048
##    120          0.2757          nan        0.1000    0.0027
##    140          0.2337          nan        0.1000    0.0021
##    150          0.2155          nan        0.1000    0.0029
##
## Iter    TrainDeviance    ValidDeviance    StepSize    Improve
##      1          1.6094          nan        0.1000    0.2375
##      2          1.4571          nan        0.1000    0.1652
##      3          1.3545          nan        0.1000    0.1322
##      4          1.2718          nan        0.1000    0.1050

```

##	5	1.2054	nan	0.1000	0.0870
##	6	1.1504	nan	0.1000	0.0810
##	7	1.0996	nan	0.1000	0.0678
##	8	1.0561	nan	0.1000	0.0773
##	9	1.0094	nan	0.1000	0.0517
##	10	0.9757	nan	0.1000	0.0593
##	20	0.6993	nan	0.1000	0.0252
##	40	0.4547	nan	0.1000	0.0130
##	60	0.3314	nan	0.1000	0.0079
##	80	0.2491	nan	0.1000	0.0029
##	100	0.1953	nan	0.1000	0.0038
##	120	0.1571	nan	0.1000	0.0015
##	140	0.1239	nan	0.1000	0.0013
##	150	0.1099	nan	0.1000	0.0006
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1288
##	2	1.5227	nan	0.1000	0.0870
##	3	1.4652	nan	0.1000	0.0644
##	4	1.4211	nan	0.1000	0.0544
##	5	1.3842	nan	0.1000	0.0509
##	6	1.3516	nan	0.1000	0.0389
##	7	1.3262	nan	0.1000	0.0405
##	8	1.3004	nan	0.1000	0.0363
##	9	1.2775	nan	0.1000	0.0354
##	10	1.2529	nan	0.1000	0.0316
##	20	1.0902	nan	0.1000	0.0193
##	40	0.9054	nan	0.1000	0.0088
##	60	0.7946	nan	0.1000	0.0064
##	80	0.7126	nan	0.1000	0.0040
##	100	0.6466	nan	0.1000	0.0037
##	120	0.5925	nan	0.1000	0.0041
##	140	0.5466	nan	0.1000	0.0022
##	150	0.5255	nan	0.1000	0.0024
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.1916
##	2	1.4835	nan	0.1000	0.1333
##	3	1.3952	nan	0.1000	0.1039
##	4	1.3279	nan	0.1000	0.0897
##	5	1.2710	nan	0.1000	0.0705
##	6	1.2253	nan	0.1000	0.0643
##	7	1.1843	nan	0.1000	0.0624
##	8	1.1445	nan	0.1000	0.0539
##	9	1.1101	nan	0.1000	0.0445
##	10	1.0817	nan	0.1000	0.0452

##	20	0.8463	nan	0.1000	0.0254
##	40	0.6141	nan	0.1000	0.0108
##	60	0.4793	nan	0.1000	0.0075
##	80	0.3924	nan	0.1000	0.0066
##	100	0.3244	nan	0.1000	0.0025
##	120	0.2692	nan	0.1000	0.0020
##	140	0.2295	nan	0.1000	0.0026
##	150	0.2119	nan	0.1000	0.0011
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2424
##	2	1.4525	nan	0.1000	0.1701
##	3	1.3473	nan	0.1000	0.1223
##	4	1.2714	nan	0.1000	0.1107
##	5	1.2009	nan	0.1000	0.0948
##	6	1.1406	nan	0.1000	0.0792
##	7	1.0899	nan	0.1000	0.0677
##	8	1.0456	nan	0.1000	0.0557
##	9	1.0094	nan	0.1000	0.0620
##	10	0.9711	nan	0.1000	0.0643
##	20	0.6965	nan	0.1000	0.0406
##	40	0.4508	nan	0.1000	0.0127
##	60	0.3247	nan	0.1000	0.0075
##	80	0.2483	nan	0.1000	0.0050
##	100	0.1952	nan	0.1000	0.0035
##	120	0.1543	nan	0.1000	0.0036
##	140	0.1238	nan	0.1000	0.0012
##	150	0.1106	nan	0.1000	0.0010
##					
##	Iter	TrainDeviance	ValidDeviance	StepSize	Improve
##	1	1.6094	nan	0.1000	0.2406
##	2	1.4542	nan	0.1000	0.1705
##	3	1.3474	nan	0.1000	0.1254
##	4	1.2696	nan	0.1000	0.1097
##	5	1.1998	nan	0.1000	0.0906
##	6	1.1425	nan	0.1000	0.0820
##	7	1.0909	nan	0.1000	0.0737
##	8	1.0450	nan	0.1000	0.0686
##	9	1.0014	nan	0.1000	0.0523
##	10	0.9688	nan	0.1000	0.0672
##	20	0.7014	nan	0.1000	0.0266
##	40	0.4644	nan	0.1000	0.0098
##	60	0.3341	nan	0.1000	0.0069
##	80	0.2526	nan	0.1000	0.0057
##	100	0.1954	nan	0.1000	0.0027
##	120	0.1578	nan	0.1000	0.0020

```
##      140      0.1250      nan      0.1000      0.0017
##      150      0.1123      nan      0.1000      0.0007

m_gbm$finalModel

## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 43 had non-zero influence.

pred_gbm <- predict(m_gbm, TestSet)
conf_mat_gbm <- confusionMatrix(pred_gbm, TestSet$classe)
conf_mat_gbm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##           A 1671   11    0    0    0
##           B    3 1116    7    6    5
##           C    0   12 1018   14    1
##           D    0    0    1  944    7
##           E    0    0    0    0 1069
##
## Overall Statistics
##
##           Accuracy : 0.9886
##           95% CI : (0.9856, 0.9912)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9856
##           McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity      0.9982   0.9798   0.9922   0.9793   0.9880
## Specificity      0.9974   0.9956   0.9944   0.9984   1.0000
## Pos Pred Value   0.9935   0.9815   0.9742   0.9916   1.0000
## Neg Pred Value   0.9993   0.9952   0.9983   0.9959   0.9973
## Prevalence       0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate   0.2839   0.1896   0.1730   0.1604   0.1816
## Detection Prevalence 0.2858   0.1932   0.1776   0.1618   0.1816
## Balanced Accuracy 0.9978   0.9877   0.9933   0.9888   0.9940
```

###Model Accuracy Comparison and Predicting using Test Data set The accuracies for all the above models are : Random Forest: 99.6% Decision Tree : 76.3% GBM : 98.6%

Hence random forest comes out to be the most efficient model.

Using random forest model, predicting the test data.

```
Pred_test <- predict(m_rf, Data_test)
```

```
Pred_test
```

```
## [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
```