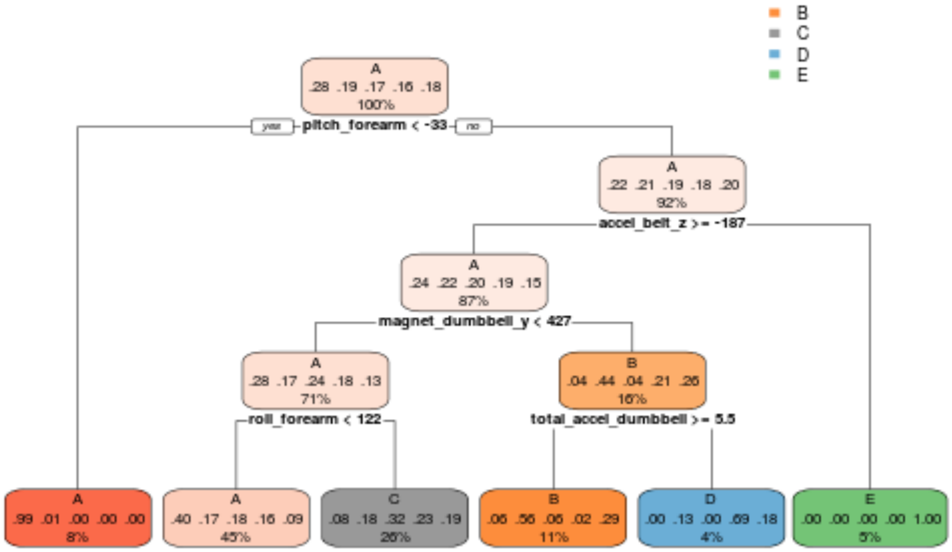
```
title: "ML assignment week 4"
author: "Siti Soraya"
date: "10/28/2019"

 # Loading the data
 train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
 test  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
 training_dat <- read.csv(url(train))
 testing_dat <- read.csv(url(test))

 dim(training_dat)
 [1] 19622   160
 dim(testing_dat)
 [1]  20 160

 # Cleaning the data
 # Removing variables that are having nearly zero variance
 non_zero_variance <- nearZeroVar(training_dat)
 training_dat <- training_dat[,-non_zero_variance]
 testing_dat <- testing_dat[,-non_zero_variance]

 dim(training_dat)
 [1] 19622   100
 dim(testing_dat)
 [1]  20 100

 # Removing variables that are having NA values, threshold is 95%
 na_val <- sapply(training_dat, function(x) mean(is.na(x))) > 0.95
 training_dat <- training_dat[,na_val == FALSE]
 testing_dat <- testing_dat[,na_val == FALSE]

 dim(training_dat)
 [1] 19622    59
 dim(testing_dat)
 [1] 20 59

 # Removing non-numeric variables which will not contribute into model
 training_dat <- training_dat[,8:59]
 testing_dat <- testing_dat[,8:59]

 dim(training_dat)
 [1] 19622    52
 dim(testing_dat)
 [1] 20 52

 # Partitioning the data
 partition <- createDataPartition(training_dat$classe, p=0.6, list=FALSE)
 training2 <- training_dat[partition,]
 testing2 <- training_dat[-partition,]

 dim(training2)
 [1] 11776    52
 dim(testing2)
 [1] 7846   52

 # Decision tree model
 DT_modfit <- train(classe ~ ., data = training2, method="rpart”)
 DT_prediction <- predict(DT_modfit, testing2)
 DT_pred_conf <- confusionMatrix(DT_prediction, testing2$classe)
 DT_pred_conf

 Confusion Matrix and Statistics

          Reference
 Prediction    A    B    C    D    E
          A 2032  614  589  560  327
          B   41  482   46   30  221
          C  155  371  731  499  413
          D    0   50    2  197   70
          E    4    1    0    0  411
```

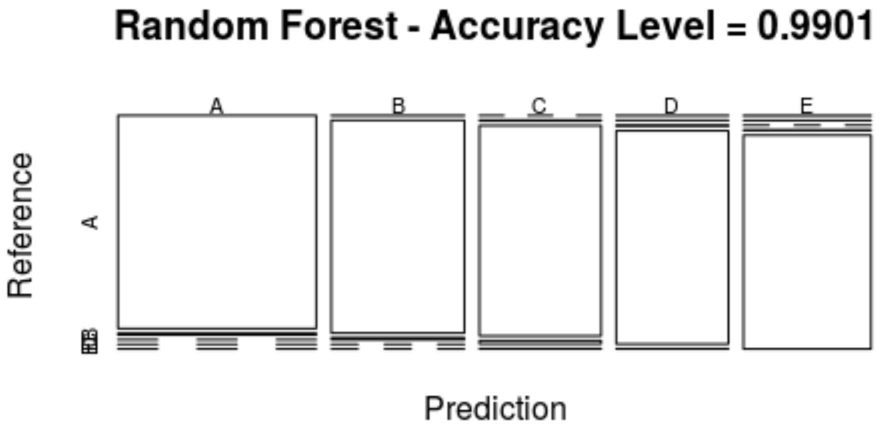
```
Overall Statistics

               Accuracy : 0.4911
                 95% CI : (0.48, 0.5022)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3354
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A Class: B Class: C Class: D
Sensitivity            0.9104  0.31752  0.53436  0.15319
Specificity            0.6277  0.94659  0.77802  0.98140
Pos Pred Value         0.4930  0.58780  0.33702  0.61755
Neg Pred Value         0.9463  0.85255  0.88779  0.85532
Prevalence             0.2845  0.19347  0.17436  0.16391
Detection Rate         0.2590  0.06143  0.09317  0.02511
Detection Prevalence   0.5254  0.10451  0.27645  0.04066
Balanced Accuracy      0.7691  0.63205  0.65619  0.56730

                     Class: E
Sensitivity           0.28502
Specificity           0.99922
Pos Pred Value        0.98798
Neg Pred Value        0.86124
Prevalence            0.18379
Detection Rate        0.05238
Detection Prevalence  0.05302
Balanced Accuracy     0.64212
```

```
rpart.plot(DT_modfit$finalModel, roundint=FALSE)
```



```
# Random forest model
RF_modfit <- train(classe ~ ., data = training2, method = "rf", ntree = 100)
RF_prediction <- predict(RF_modfit, testing2)
RF_pred_conf <- confusionMatrix(RF_prediction, testing2$classe)
RF_pred_conf

Confusion Matrix and Statistics

         Reference
Prediction    A    B    C    D    E
         A 2228   18    0    0    0
         B    1 1493   13    0    0
         C    0    5 1349   21    3
         D    1    1    6 1261    2
         E    2    1    0    4 1437

Overall Statistics

              Accuracy : 0.9901
                95% CI : (0.9876, 0.9921)
   No Information Rate : 0.2845
   P-Value [Acc > NIR] : < 2.2e-16
                 Kappa : 0.9874
Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D
Sensitivity            0.9982   0.9835   0.9861   0.9806
Specificity            0.9968   0.9978   0.9955   0.9985
Pos Pred Value         0.9920   0.9907   0.9790   0.9921
Neg Pred Value         0.9993   0.9961   0.9971   0.9962
Prevalence             0.2845   0.1935   0.1744   0.1639
Detection Rate         0.2840   0.1903   0.1719   0.1607
Detection Prevalence   0.2863   0.1921   0.1756   0.1620
Balanced Accuracy      0.9975   0.9907   0.9908   0.9895

                     Class: E
Sensitivity            0.9965
Specificity            0.9989
Pos Pred Value         0.9952
Neg Pred Value         0.9992
Prevalence             0.1838
Detection Rate         0.1832
Detection Prevalence   0.1840
Balanced Accuracy      0.9977
```

```
plot(RF_pred_conf$table, col = RF_pred_conf$byClass,
     main = paste("Random Forest - Accuracy Level =",
                  round(RF_pred_conf$overall['Accuracy'], 4)))
```



```
Final_RF_prediction <- predict(RF_modfit, testing_dat)
Final_RF_prediction
[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
```