**Assignment 1 - Introduction to Machine Learning**

For this assignment, you will be using the Breast Cancer Wisconsin (Diagnostic) Database to create a classifier that can help diagnose patients. First, read through the description of the dataset (below).

import numpy as np import pandas as pd from sklearn.datasets import load\_breast\_cancer

cancer = load\_breast\_cancer()

print(cancer)

print(cancer.DESCR) # Print the data set description

The object returned by load\_breast\_cancer() is a scikit-learn Bunch object, which is similar to a dictionary.

In [1]:**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**from** **sklearn.datasets** **import** load\_breast\_cancer

cancer = load\_breast\_cancer()

*# print(cancer)*

print(cancer.DESCR) *# Print the data set description*

cancer.keys()

Breast Cancer Wisconsin (Diagnostic) Database

=============================================

Notes

Data Set Characteristics:

:Number of Instances: 569

:Number of Attributes: 30 numeric, predictive attributes and the class

## **\*\* 1. Loading the Dataset\*\***

The Dataset has been downloaded from the internet and has been loaded into two seperate dataframes, \_\_“training\_\_ and **“testing”**. The \_\_“training\_\_ data set has 19622 number of records and the \_\_“testing\_\_ data set has 20 records. The number of variables is 160.

Here are the datasets, loaded directly from web and then downloaded.

train\_url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

test\_url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

init\_org\_training\_data <- read.csv(url(train\_url))

init\_org\_testing\_data <- read.csv(url(test\_url))

dim(init\_org\_training\_data)

## [1] 19622 160

dim(init\_org\_testing\_data)

## [1] 20 160

# \*\* 2. Data Cleansing \*\*

There are 3 parts in Data Cleansing.

### **A. Removing Variables which are having nearly zero variance.**

non\_zero\_var <- nearZeroVar(init\_org\_training\_data)

org\_training\_data <- init\_org\_training\_data[,-non\_zero\_var]

org\_testing\_data <- init\_org\_testing\_data[,-non\_zero\_var]

dim(org\_training\_data)

## [1] 19622 100

dim(org\_testing\_data)

## [1] 20 100

### **B. Removing Variables which are having NA values. Our threshhold is 95%.**

na\_val\_col <- sapply(org\_training\_data, **function**(x) mean(is.na(x))) > 0.95

org\_training\_data <- org\_training\_data[,na\_val\_col == FALSE]

org\_testing\_data <- org\_testing\_data[,na\_val\_col == FALSE]

dim(org\_training\_data)

## [1] 19622 59

dim(org\_testing\_data)

## [1] 20 59

### **C. Removing variables which are non-numeric and hence will not contribute into our model. The very first 7 variables are of that kind only. Hence those needs to be removed from the datasets.**

org\_training\_data <- org\_training\_data[,8:59]

org\_testing\_data <- org\_testing\_data[,8:59]

dim(org\_training\_data)

## [1] 19622 52

dim(org\_testing\_data)

## [1] 20 52

## **\*\* 3. Data Partitioning \*\***

As per recommendation of the course \_\_ Practical Machine Learning\_\_ , we will be seggregating our **org\_training\_data** into 2 different parts, one is the training set (consisiting 60% of the total data) and test set (consisting 40% of the total data)

inTrain <- createDataPartition(org\_training\_data$classe, p=0.6, list=FALSE)

training <- org\_training\_data[inTrain,]

testing <- org\_training\_data[-inTrain,]

dim(training)

## [1] 11776 52

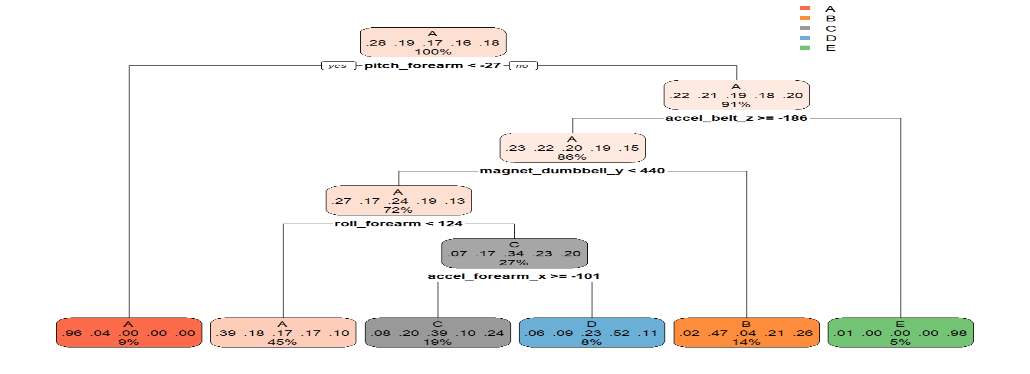
dim(testing)

## [1] 7846 52

## **\*\* 4. Decision Tree Model \*\***

DT\_modfit <- train(classe ~ ., data = training, method="rpart")

Prediction in terms of Decision Tree Model



## **\*\* 5. Random Forest Model \*\***

RF\_modfit <- train(classe ~ ., data = training, method = "rf", ntree = 100)

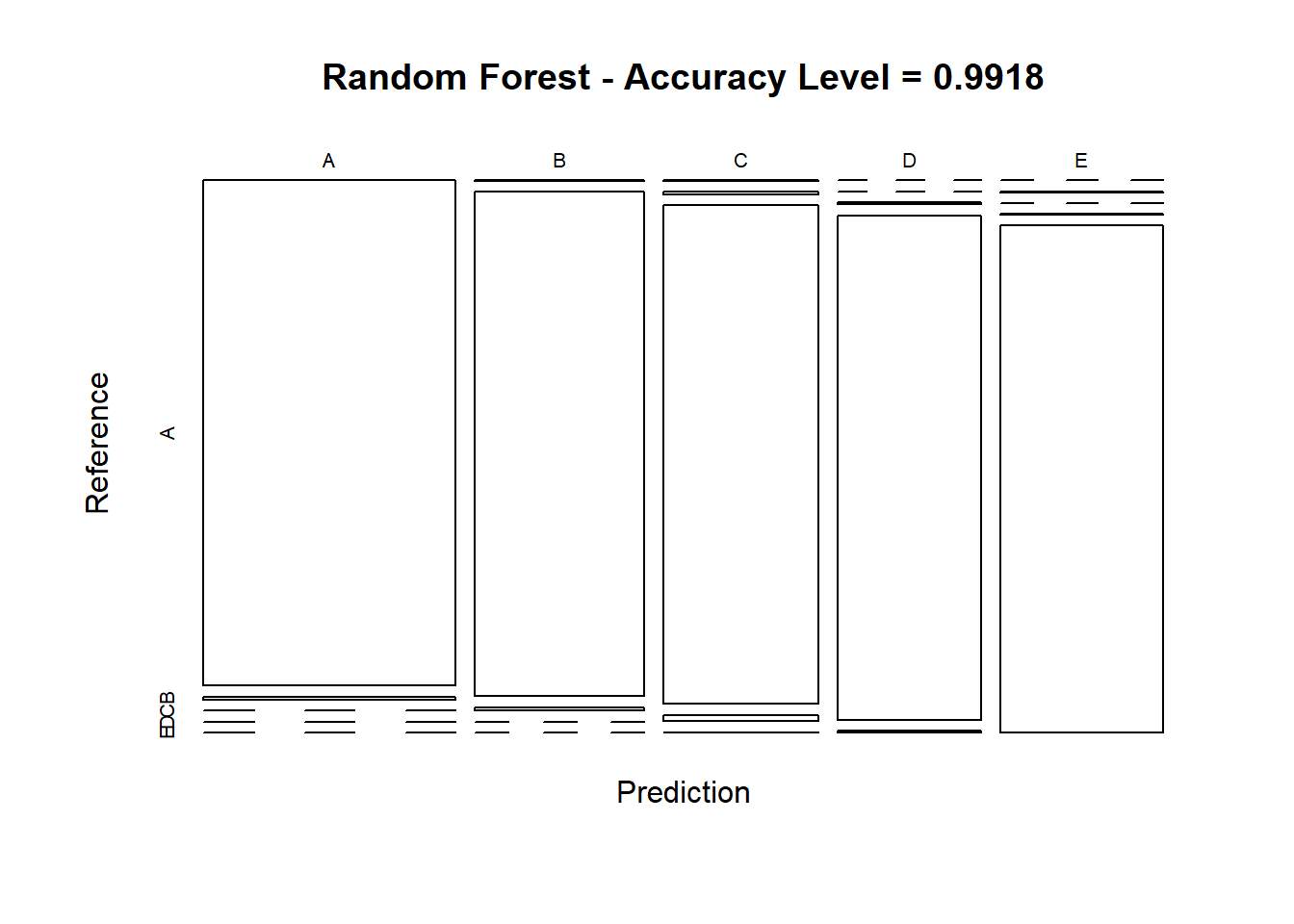
Prediction in terms of Random Forest Model

Here is the plot

plot(RF\_pred\_conf$table, col = RF\_pred\_conf$byClass,

main = paste("Random Forest - Accuracy Level =",

round(RF\_pred\_conf$overall['Accuracy'], 4)))



From the Confusion Matrix, we can clearly see that the prediction accuracy of Random Forest model is 99% which is satisfactory.

## **\*\* 6. Gradient Boosting Model \*\***

GBM\_modfit <- train(classe ~ ., data = training, method = "gbm", verbose = FALSE)

GBM\_modfit$finalModel

## A gradient boosted model with multinomial loss function.

## 150 iterations were performed.

## There were 51 predictors of which 42 had non-zero influence.

Prediction in terms of GBM Model

