Diabetes prediction using R

code :-

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(mlbench)
library(tidyr)
library(Boruta)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(modelr)
library(cowplot)
```

```
##
## ************************************************************
```

```
## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ************************************************************
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```
library(e1071)
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(xgboost)
```

```
##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(mlr)
```

```
## Loading required package: ParamHelpers

## 'mlr' is in maintenance mode since July 2019. Future development
## efforts will go into its successor 'mlr3' (<https://mlr3.mlr-org.com>).

##
## Attaching package: 'mlr'

## The following object is masked from 'package:e1071':
##
##     impute

## The following object is masked from 'package:modelr':
##
##     resample

## The following object is masked from 'package:caret':
##
##     train
```

```r
library(MLeval)

# Reading the demographics, laboratory and examination .csv files using read_csv

demo <- read_csv("C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nutriti
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )

## See spec(...) for full column specifications.
```

```r
labs <- read_csv("C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nutriti
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
## See spec(...) for full column specifications.
```

```
exam <- read_csv("C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nutriti
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   BMIHEAD = col_logical(),
##   OHX02CTC = col_character(),
##   OHX03CTC = col_character(),
##   OHX04CTC = col_character(),
##   OHX05CTC = col_character(),
##   OHX06CTC = col_character(),
##   OHX07CTC = col_character(),
##   OHX08CTC = col_character(),
##   OHX09CTC = col_character(),
##   OHX10CTC = col_character(),
##   OHX11CTC = col_character(),
##   OHX12CTC = col_character(),
##   OHX13CTC = col_character(),
##   OHX14CTC = col_character(),
##   OHX15CTC = col_character(),
##   OHX18CTC = col_character(),
##   OHX19CTC = col_character(),
##   OHX20CTC = col_character(),
##   OHX21CTC = col_character(),
##   OHX22CTC = col_character()
##   # ... with 38 more columns
## )
## See spec(...) for full column specifications.
```

```
# Making new dataframes consisting of only selected columns
demo_df <- demo %>% select(SEQN , "GENDER" = RIAGENDR, "AGE" = RIDAGEYR)

# HBA1c is Hemoglobin A1c or Glycohemoglobin which is an indicator of diabetes
lab_df <- labs %>% select(SEQN, "HbA1c" = LBXGH)

# Creating another dataframe by inner joining the other data frames
df <- lab_df %>% inner_join(demo_df, by = "SEQN") %>%
                            inner_join(exam, by = "SEQN")

summary(df)
```

```
##       SEQN           HbA1c            GENDER           AGE
##  Min.   :73557   Min.   : 3.500   Min.   :1.000   Min.   : 0.00
##  1st Qu.:76092   1st Qu.: 5.200   1st Qu.:1.000   1st Qu.:10.00
##  Median :78643   Median : 5.400   Median :2.000   Median :27.00
##  Mean   :78645   Mean   : 5.643   Mean   :1.508   Mean   :31.63
##  3rd Qu.:81191   3rd Qu.: 5.800   3rd Qu.:2.000   3rd Qu.:52.00
##  Max.   :83731   Max.   :17.500   Max.   :2.000   Max.   :80.00
##                  NA's   :3170
##     PEASCST1        PEASCTM1         PEASCCT1          BPXCHR
##  Min.   :1.000   Min.   :   7.0   Min.   :  1.00   Min.   : 60.0
##  1st Qu.:1.000   1st Qu.: 578.0   1st Qu.: 56.00   1st Qu.: 88.0
```

```
##  Median :1.000   Median : 689.0   Median : 56.00   Median :102.0
##  Mean   :1.065   Mean   : 660.6   Mean   : 57.83   Mean   :105.5
##  3rd Qu.:1.000   3rd Qu.: 832.0   3rd Qu.: 72.00   3rd Qu.:120.0
##  Max.   :3.000   Max.   :2868.0   Max.   :122.00   Max.   :178.0
##                  NA's   :305      NA's   :9493     NA's   :7852
##      BPAARM          BPACSZ          BPXPLS          BPXPULS
##  Min.   :1.000   Min.   :1.000   Min.   : 40.00   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:3.000   1st Qu.: 66.00   1st Qu.:1.000
##  Median :1.000   Median :4.000   Median : 74.00   Median :1.000
##  Mean   :1.008   Mean   :3.675   Mean   : 74.42   Mean   :1.014
##  3rd Qu.:1.000   3rd Qu.:4.000   3rd Qu.: 82.00   3rd Qu.:1.000
##  Max.   :8.000   Max.   :5.000   Max.   :180.00   Max.   :2.000
##  NA's   :2278    NA's   :2271    NA's   :2264     NA's   :302
##      BPXPTY          BPXML1          BPXSY1          BPXDI1
##  Min.   :1.000   Min.   :100.0   Min.   : 66.0   Min.   :  0.00
##  1st Qu.:1.000   1st Qu.:130.0   1st Qu.:106.0   1st Qu.: 58.00
##  Median :1.000   Median :140.0   Median :116.0   Median : 66.00
##  Mean   :1.004   Mean   :144.7   Mean   :118.1   Mean   : 65.77
##  3rd Qu.:1.000   3rd Qu.:150.0   3rd Qu.:128.0   3rd Qu.: 76.00
##  Max.   :2.000   Max.   :888.0   Max.   :228.0   Max.   :122.00
##  NA's   :2249    NA's   :2260    NA's   :2641    NA's   :2641
##      BPAEN1          BPXSY2          BPXDI2          BPAEN2
##  Min.   :1.000   Min.   : 66.0   Min.   :  0.00   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:106.0   1st Qu.: 58.00   1st Qu.:2.000
##  Median :2.000   Median :116.0   Median : 66.00   Median :2.000
##  Mean   :1.989   Mean   :118.2   Mean   : 65.24   Mean   :1.954
##  3rd Qu.:2.000   3rd Qu.:128.0   3rd Qu.: 74.00   3rd Qu.:2.000
##  Max.   :2.000   Max.   :230.0   Max.   :116.00   Max.   :2.000
##  NA's   :2274    NA's   :2404    NA's   :2404     NA's   :2276
##      BPXSY3          BPXDI3          BPAEN3          BPXSY4
##  Min.   : 62    Min.   :  0.00   Min.   :1.000   Min.   : 80.0
##  1st Qu.:106    1st Qu.: 58.00   1st Qu.:2.000   1st Qu.:108.0
##  Median :114    Median : 68.00   Median :2.000   Median :126.0
##  Mean   :118    Mean   : 65.04   Mean   :1.963   Mean   :125.7
##  3rd Qu.:128    3rd Qu.: 74.00   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :228    Max.   :118.00   Max.   :2.000   Max.   :212.0
##  NA's   :2405   NA's   :2405     NA's   :2276    NA's   :9298
##      BPXDI4          BPAEN4          BMDSTATS        BMXWT
##  Min.   :  0.00   Min.   :1.000   Min.   :1.00   Min.   :  3.10
##  1st Qu.: 60.00   1st Qu.:2.000   1st Qu.:1.00   1st Qu.: 37.95
##  Median : 70.00   Median :2.000   Median :1.00   Median : 65.30
##  Mean   : 69.01   Mean   :1.879   Mean   :1.14   Mean   : 62.60
##  3rd Qu.: 78.00   3rd Qu.:2.000   3rd Qu.:1.00   3rd Qu.: 83.50
##  Max.   :128.00   Max.   :2.000   Max.   :4.00   Max.   :222.60
##  NA's   :9298     NA's   :9251                   NA's   :90
##      BMIWT           BMXRECUM        BMIRECUM       BMXHEAD       BMIHEAD
##  Min.   :1.000   Min.   : 48.60   Min.   :1      Min.   :33.80   Mode:logical
##  1st Qu.:3.000   1st Qu.: 69.70   1st Qu.:1      1st Qu.:39.70   NA's:9813
##  Median :3.000   Median : 82.80   Median :1      Median :41.80
##  Mean   :2.966   Mean   : 81.63   Mean   :1      Mean   :41.57
##  3rd Qu.:3.000   3rd Qu.: 93.20   3rd Qu.:1      3rd Qu.:43.50
##  Max.   :4.000   Max.   :115.10   Max.   :1      Max.   :46.80
##  NA's   :9429    NA's   :8748     NA's   :9782   NA's   :9584
##      BMXHT           BMIHT           BMXBMI         BMDBMIC
```

```
##  Min.   : 79.7   Min.   :1.000   Min.   :12.10   Min.   :1.000
##  1st Qu.:149.5   1st Qu.:3.000   1st Qu.:19.70   1st Qu.:2.000
##  Median :162.0   Median :3.000   Median :24.70   Median :2.000
##  Mean   :155.9   Mean   :2.647   Mean   :25.68   Mean   :2.489
##  3rd Qu.:171.1   3rd Qu.:3.000   3rd Qu.:30.20   3rd Qu.:3.000
##  Max.   :202.6   Max.   :3.000   Max.   :82.90   Max.   :4.000
##  NA's   :746     NA's   :9592    NA's   :758     NA's   :6290
##      BMXLEG          BMILEG        BMXARML         BMIARML        BMXARMC
##  Min.   :24.40   Min.   :1      Min.   : 9.90   Min.   :1      Min.   :10.40
##  1st Qu.:36.00   1st Qu.:1      1st Qu.:30.50   1st Qu.:1      1st Qu.:22.60
##  Median :38.60   Median :1      Median :35.50   Median :1      Median :29.30
##  Mean   :38.58   Mean   :1      Mean   :33.14   Mean   :1      Mean   :28.49
##  3rd Qu.:41.30   3rd Qu.:1      3rd Qu.:38.10   3rd Qu.:1      3rd Qu.:34.00
##  Max.   :51.90   Max.   :1      Max.   :47.90   Max.   :1      Max.   :59.40
##  NA's   :2411    NA's   :9463   NA's   :512     NA's   :9445   NA's   :512
##     BMIARMC        BMXWAIST        BMIWAIST       BMXSAD1         BMXSAD2
##  Min.   :1      Min.   : 40.20   Min.   :1      Min.   :10.00   Min.   :10.20
##  1st Qu.:1      1st Qu.: 71.20   1st Qu.:1      1st Qu.:17.30   1st Qu.:17.30
##  Median :1      Median : 87.80   Median :1      Median :20.70   Median :20.70
##  Mean   :1      Mean   : 87.27   Mean   :1      Mean   :21.11   Mean   :21.09
##  3rd Qu.:1      3rd Qu.:102.80   3rd Qu.:1      3rd Qu.:24.40   3rd Qu.:24.38
##  Max.   :1      Max.   :177.90   Max.   :1      Max.   :40.20   Max.   :40.20
##  NA's   :9441   NA's   :1152     NA's   :9374   NA's   :2595    NA's   :2595
##     BMXSAD3         BMXSAD4        BMDAVSAD        BMDSADCM        MGDEXSTS
##  Min.   :10.90   Min.   :11.0   Min.   :10.10   Min.   :1.000   Min.   :1.000
##  1st Qu.:18.50   1st Qu.:18.5   1st Qu.:17.30   1st Qu.:1.000   1st Qu.:1.000
##  Median :22.40   Median :22.4   Median :20.70   Median :1.000   Median :1.000
##  Mean   :22.23   Mean   :22.2   Mean   :21.11   Mean   :1.176   Mean   :1.132
##  3rd Qu.:25.40   3rd Qu.:25.3   3rd Qu.:24.40   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :35.60   Max.   :35.6   Max.   :40.10   Max.   :5.000   Max.   :3.000
##  NA's   :9455    NA's   :9455   NA's   :2595    NA's   :9329    NA's   :1522
##      MGD050          MGD060          MGQ070          MGQ080
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000
##  Median :2.000   Median :2.000   Median :2.000   Median :1.000
##  Mean   :1.967   Mean   :2.115   Mean   :1.883   Mean   :3.283
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:9.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
##  NA's   :4602    NA's   :9621    NA's   :2062    NA's   :8872
##      MGQ090          MGQ100          MGQ110          MGQ120
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000
##  Median :2.000   Median :2.000   Median :1.000   Median :2.000
##  Mean   :1.905   Mean   :1.909   Mean   :2.997   Mean   :1.852
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :2.000
##  NA's   :8872    NA's   :2061    NA's   :9058    NA's   :9058
##      MGD130         MGQ90DG         MGDSEAT        MGAPHAND       MGATHAND
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.0    Min.   :1.0
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.0    1st Qu.:1.0
##  Median :1.000   Median :1.000   Median :1.000   Median :1.0    Median :2.0
##  Mean   :1.142   Mean   :1.014   Mean   :1.041   Mean   :1.5    Mean   :1.5
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.0    3rd Qu.:2.0
##  Max.   :3.000   Max.   :4.000   Max.   :2.000   Max.   :2.0    Max.   :2.0
```

```
##  NA's   :2006    NA's   :2006    NA's   :4602    NA's   :2006   NA's   :2006
##     MGXH1T1         MGXH1T1E        MGXH2T1         MGXH2T1E
##  Min.   : 4.00   Min.   :1.000   Min.   : 4.00   Min.   :1.000
##  1st Qu.:20.40   1st Qu.:1.000   1st Qu.:20.70   1st Qu.:1.000
##  Median :27.70   Median :1.000   Median :28.20   Median :1.000
##  Mean   :28.63   Mean   :1.043   Mean   :29.27   Mean   :1.025
##  3rd Qu.:36.70   3rd Qu.:1.000   3rd Qu.:37.40   3rd Qu.:1.000
##  Max.   :81.50   Max.   :2.000   Max.   :79.50   Max.   :2.000
##  NA's   :2015    NA's   :2015    NA's   :2131    NA's   :2131
##     MGXH1T2         MGXH1T2E        MGXH2T2         MGXH2T2E
##  Min.   : 5.00   Min.   :1.000   Min.   : 5.00   Min.   :1.000
##  1st Qu.:21.20   1st Qu.:1.000   1st Qu.:21.60   1st Qu.:1.000
##  Median :28.80   Median :1.000   Median :29.00   Median :1.000
##  Mean   :29.89   Mean   :1.018   Mean   :30.36   Mean   :1.013
##  3rd Qu.:38.10   3rd Qu.:1.000   3rd Qu.:38.80   3rd Qu.:1.000
##  Max.   :77.60   Max.   :2.000   Max.   :81.40   Max.   :2.000
##  NA's   :2024    NA's   :2024    NA's   :2141    NA's   :2141
##     MGXH1T3         MGXH1T3E        MGXH2T3         MGXH2T3E
##  Min.   : 5.00   Min.   :1.000   Min.   : 5.10   Min.   :1.000
##  1st Qu.:21.60   1st Qu.:1.000   1st Qu.:21.80   1st Qu.:1.000
##  Median :29.10   Median :1.000   Median :29.50   Median :1.000
##  Mean   :30.43   Mean   :1.014   Mean   :30.69   Mean   :1.014
##  3rd Qu.:39.00   3rd Qu.:1.000   3rd Qu.:39.40   3rd Qu.:1.000
##  Max.   :81.20   Max.   :2.000   Max.   :82.80   Max.   :2.000
##  NA's   :2027    NA's   :2027    NA's   :2149    NA's   :2149
##     MGDCGSZ          OHDEXSTS        OHDDESTS         OHXIMP
##  Min.   :  8.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.: 45.60   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000
##  Median : 60.30   Median :1.000   Median :1.000   Median :2.000
##  Mean   : 63.05   Mean   :1.132   Mean   :1.097   Mean   :1.967
##  3rd Qu.: 80.70   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.000
##  Max.   :162.80   Max.   :3.000   Max.   :3.000   Max.   :2.000
##  NA's   :2136     NA's   :391     NA's   :391     NA's   :5494
##     OHX01TC         OHX02TC         OHX03TC         OHX04TC
##  Min.   :2.000   Min.   :2.000   Min.   :2.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :4.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :3.707   Mean   :2.885   Mean   :2.624   Mean   :2.168
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX05TC         OHX06TC         OHX07TC         OHX08TC         OHX09TC
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.00   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :2.00   Median :2.000   Median :2.000   Median :2.000
##  Mean   :2.171   Mean   :2.02   Mean   :2.101   Mean   :2.103   Mean   :2.104
##  3rd Qu.:2.000   3rd Qu.:2.00   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845     NA's   :845    NA's   :845     NA's   :845     NA's   :845
##     OHX10TC         OHX11TC         OHX12TC         OHX13TC         OHX14TC
##  Min.   :1.0   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :2.00
##  1st Qu.:2.0   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.00
##  Median :2.0   Median :2.000   Median :2.000   Median :2.000   Median :2.00
##  Mean   :2.1   Mean   :2.017   Mean   :2.172   Mean   :2.175   Mean   :2.62
```

```
##  3rd Qu.:2.0   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:4.00
##  Max.   :5.0   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.00
##  NA's   :845   NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX15TC         OHX16TC         OHX17TC         OHX18TC         OHX19TC
##  Min.   :2.00   Min.   :2.000   Min.   :2.000   Min.   :2.000   Min.   :2.000
##  1st Qu.:2.00   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.00   Median :4.000   Median :4.000   Median :2.000   Median :2.000
##  Mean   :2.89   Mean   :3.711   Mean   :3.669   Mean   :2.907   Mean   :2.688
##  3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845    NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX20TC         OHX21TC         OHX22TC         OHX23TC
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :2.102   Mean   :2.047   Mean   :1.955   Mean   :2.008
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX24TC         OHX25TC         OHX26TC         OHX27TC
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :2.028   Mean   :2.029   Mean   :2.012   Mean   :1.956
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX28TC         OHX29TC         OHX30TC         OHX31TC
##  Min.   :1.000   Min.   :1.000   Min.   :2.000   Min.   :2.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :2.000   Median :2.000   Median :2.000
##  Mean   :2.046   Mean   :2.106   Mean   :2.682   Mean   :2.894
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  NA's   :845     NA's   :845     NA's   :845     NA's   :845
##     OHX32TC        OHX02CTC           OHX03CTC           OHX04CTC
##  Min.   :2.000   Length:9813        Length:9813        Length:9813
##  1st Qu.:4.000   Class :character   Class :character   Class :character
##  Median :4.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :3.663
##  3rd Qu.:4.000
##  Max.   :5.000
##  NA's   :845
##    OHX05CTC           OHX06CTC           OHX07CTC           OHX08CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX09CTC           OHX10CTC           OHX11CTC           OHX12CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX13CTC           OHX14CTC           OHX15CTC           OHX18CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX19CTC           OHX20CTC           OHX21CTC           OHX22CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX23CTC           OHX24CTC           OHX25CTC           OHX26CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX27CTC           OHX28CTC           OHX29CTC           OHX30CTC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX31CTC           OHX02CSC           OHX03CSC           OHX04CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX05CSC           OHX06CSC           OHX07CSC           OHX08CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX09CSC           OHX10CSC           OHX11CSC           OHX12CSC
```

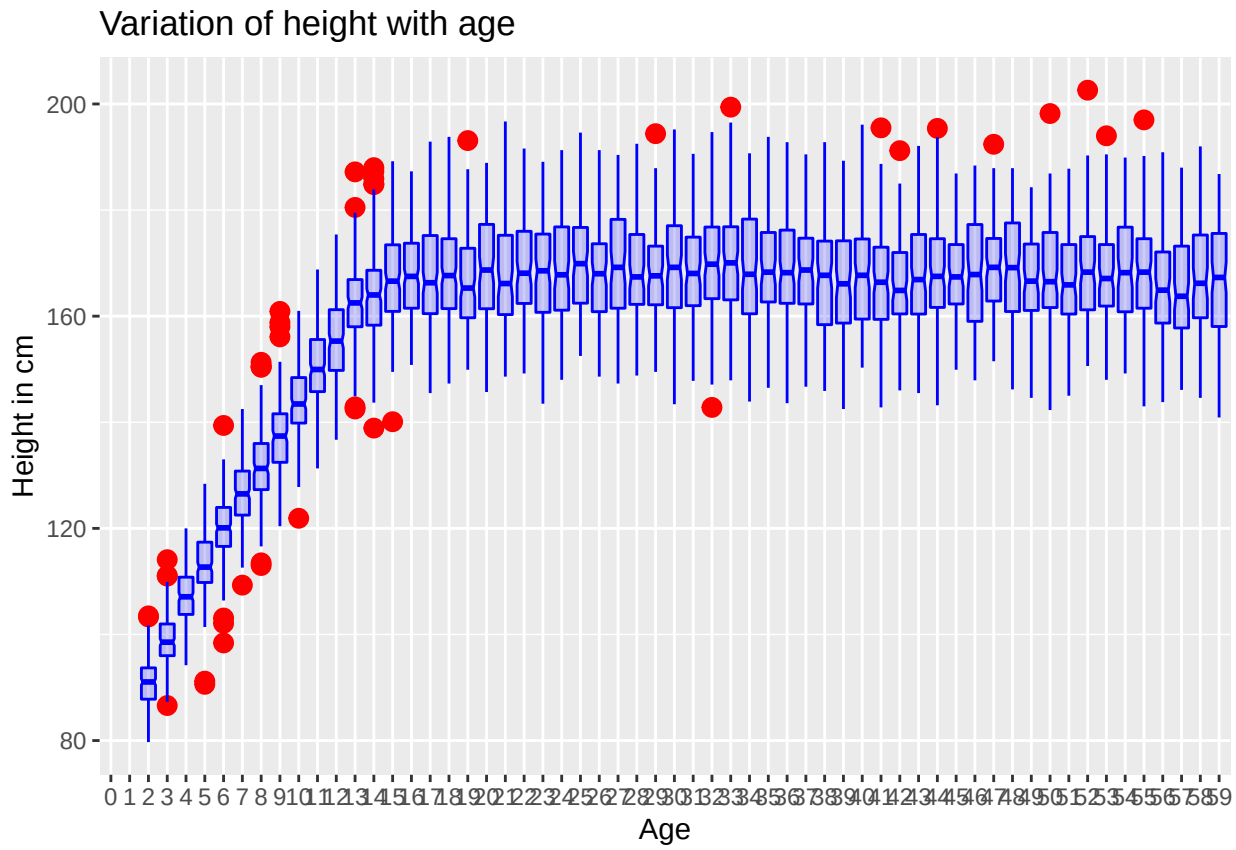
```
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX13CSC           OHX14CSC           OHX15CSC           OHX18CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX19CSC           OHX20CSC           OHX21CSC           OHX22CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX23CSC           OHX24CSC           OHX25CSC           OHX26CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX27CSC           OHX28CSC           OHX29CSC           OHX30CSC
##  Length:9813        Length:9813        Length:9813        Length:9813
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    OHX31CSC            OHX02SE           OHX03SE           OHX04SE
##  Length:9813        Min.   : 0.000   Min.   : 0.000   Min.   :0.000
##  Class :character   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.000
##  Mode  :character   Median : 9.000   Median : 1.000   Median :0.000
##                     Mean   : 5.719   Mean   : 3.057   Mean   :0.376
##                     3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.:0.000
##                     Max.   :13.000   Max.   :13.000   Max.   :9.000
##                     NA's   :6549     NA's   :6549     NA's   :6549
##     OHX05SE          OHX07SE          OHX10SE          OHX12SE          OHX13SE
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00
##  Median :0.000   Median :0.000   Median :0.000   Median :0.000   Median :0.00
##  Mean   :0.642   Mean   :3.185   Mean   :3.175   Mean   :0.624   Mean   :0.37
##  3rd Qu.:0.000   3rd Qu.:9.000   3rd Qu.:9.000   3rd Qu.:0.000   3rd Qu.:0.00
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.00
```

```
##  NA's   :6549    NA's   :6549    NA's   :6549    NA's   :6549    NA's   :6549
##     OHX14SE          OHX15SE          OHX18SE          OHX19SE
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 1.000   Median : 9.000   Median : 9.000   Median : 1.000
##  Mean   : 3.043   Mean   : 5.698   Mean   : 5.425   Mean   : 2.662
##  3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.: 9.000
##  Max.   :13.000   Max.   :13.000   Max.   :12.000   Max.   :12.000
##  NA's   :6549     NA's   :6549     NA's   :6549     NA's   :6549
##     OHX20SE         OHX21SE         OHX28SE          OHX29SE
##  Min.   :0.000   Min.   :0.0    Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.0    1st Qu.:0.000   1st Qu.:0.000
##  Median :0.000   Median :0.0    Median :0.000   Median :0.000
##  Mean   :0.464   Mean   :0.7    Mean   :0.695   Mean   :0.425
##  3rd Qu.:0.000   3rd Qu.:0.0    3rd Qu.:0.000   3rd Qu.:0.000
##  Max.   :9.000   Max.   :9.0    Max.   :9.000   Max.   :9.000
##  NA's   :6549    NA's   :6549   NA's   :6549    NA's   :6549
##     OHX30SE          OHX31SE          CSXEXSTS        CSXEXCMT
##  Min.   : 0.000   Min.   : 0.000   Min.   :1.000   Min.   :  1.00
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:1.000   1st Qu.:  7.00
##  Median : 1.000   Median : 9.000   Median :1.000   Median : 56.00
##  Mean   : 2.662   Mean   : 5.472   Mean   :1.207   Mean   : 45.26
##  3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.:1.000   3rd Qu.: 56.00
##  Max.   :12.000   Max.   :12.000   Max.   :3.000   Max.   :122.00
##  NA's   :6549     NA's   :6549     NA's   :6105    NA's   :9605
##      CSQ245          CSQ241          CSQ260A         CSQ260D        CSQ260G
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1      Min.   :1
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1      1st Qu.:1
##  Median :2.000   Median :2.000   Median :1.000   Median :1      Median :1
##  Mean   :2.221   Mean   :2.006   Mean   :1.198   Mean   :1      Mean   :1
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1      3rd Qu.:1
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :1      Max.   :1
##  NA's   :6256    NA's   :8834    NA's   :9631    NA's   :9739   NA's   :9530
##     CSQ260I         CSQ260N         CSQ260M         CSQ270          CSQ450
##  Min.   :1       Min.   :1       Min.   :1       Min.   :1.000   Min.   :  0.00
##  1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1.000   1st Qu.: 15.00
##  Median :1       Median :1       Median :1       Median :1.000   Median : 17.00
##  Mean   :1       Mean   :1       Mean   :1       Mean   :1.534   Mean   : 21.35
##  3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:2.000   3rd Qu.: 27.00
##  Max.   :1       Max.   :1       Max.   :1       Max.   :9.000   Max.   :100.00
##  NA's   :9708    NA's   :9392    NA's   :7059    NA's   :9530    NA's   :6393
##      CSQ460          CSQ470          CSQ480          CSQ490
##  Min.   :  0.00   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 30.00   1st Qu.: 17.0   1st Qu.:  5.00   1st Qu.: 48.00
##  Median : 35.00   Median : 27.0   Median :  9.00   Median : 55.00
##  Mean   : 41.08   Mean   : 29.8   Mean   : 11.46   Mean   : 60.95
##  3rd Qu.: 51.00   3rd Qu.: 37.0   3rd Qu.: 15.00   3rd Qu.: 80.00
##  Max.   :100.00   Max.   :100.0   Max.   :100.00   Max.   :100.00
##  NA's   :6397     NA's   :6401    NA's   :6402     NA's   :6402
##     CSXQUIPG         CSXQUIPT        CSXNAPG          CSXNAPT
##  Min.   :  0.00   Min.   :1.000   Min.   :  0.00   Min.   :1.000
##  1st Qu.:  5.00   1st Qu.:2.000   1st Qu.: 15.00   1st Qu.:1.000
##  Median : 10.00   Median :3.000   Median : 25.00   Median :1.000
##  Mean   : 14.85   Mean   :3.008   Mean   : 26.93   Mean   :1.466
```

```
##  3rd Qu.: 20.00   3rd Qu.:4.000   3rd Qu.: 35.00   3rd Qu.:1.000
##  Max.   :100.00   Max.   :5.000   Max.   :100.00   Max.   :5.000
##  NA's   :6680     NA's   :6680    NA's   :6684     NA's   :6596
##     CSXQUISG         CSXQUIST        CSXSLTSG         CSXSLTST
##  Min.   :  0.00   Min.   :1.000   Min.   :  2.00   Min.   :1.000
##  1st Qu.: 35.00   1st Qu.:2.000   1st Qu.: 35.00   1st Qu.:1.000
##  Median : 52.00   Median :2.000   Median : 50.00   Median :1.000
##  Mean   : 53.21   Mean   :2.251   Mean   : 52.06   Mean   :1.064
##  3rd Qu.: 65.00   3rd Qu.:2.000   3rd Qu.: 63.00   3rd Qu.:1.000
##  Max.   :100.00   Max.   :5.000   Max.   :100.00   Max.   :5.000
##  NA's   :6699     NA's   :6699    NA's   :6596     NA's   :6596
##     CSXNASG          CSXNAST        CSXTSEQ             CSXCHOOD
##  Min.   :  0.00   Min.   :1.000   Length:9813        Min.   :1.000
##  1st Qu.: 18.25   1st Qu.:1.000   Class :character   1st Qu.:2.000
##  Median : 30.00   Median :1.000   Mode  :character   Median :2.000
##  Mean   : 33.08   Mean   :1.242                      Mean   :2.152
##  3rd Qu.: 42.00   3rd Qu.:1.000                      3rd Qu.:2.000
##  Max.   :100.00   Max.   :5.000                      Max.   :4.000
##  NA's   :6595     NA's   :6595                       NA's   :6286
##     CSXSBOD         CSXSMKOD        CSXLEAOD        CSXSOAOD
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:1.000
##  Median :1.000   Median :3.000   Median :3.000   Median :1.000
##  Mean   :1.428   Mean   :2.903   Mean   :2.816   Mean   :1.145
##  3rd Qu.:1.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:1.000
##  Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##  NA's   :6288    NA's   :6290    NA's   :6293    NA's   :6293
##     CSXGRAOD        CSXONOD         CSXNGSOD        CSXSLTRT
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :  1.00
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:4.000   1st Qu.: 35.00
##  Median :2.000   Median :3.000   Median :4.000   Median : 53.00
##  Mean   :2.156   Mean   :2.992   Mean   :3.751   Mean   : 55.75
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.: 74.00
##  Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :100.00
##  NA's   :6294    NA's   :6294    NA's   :6294    NA's   :8218
##     CSXSLTRG        CSXNART          CSXNARG         CSAEFFRT
##  Min.   :1.000   Min.   :  0.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.: 16.00   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median : 26.00   Median :1.000   Median :1.000
##  Mean   :1.066   Mean   : 31.53   Mean   :1.199   Mean   :1.709
##  3rd Qu.:1.000   3rd Qu.: 42.00   3rd Qu.:1.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :100.00   Max.   :5.000   Max.   :5.000
##  NA's   :8218    NA's   :8200     NA's   :8200    NA's   :6276
```

```r
# Keeping columns which have less than or equal to 50% missing values (NAs)

DF <- df[apply(df,2,function(x) mean(is.na(x))<=.50)]

# We can see that there are no more columns which have more than 50% missing
#values

write_csv(DF, "C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nutrition-

# Discarding remaining irrelevant features from dataframe and selecting only
```

```r
#useful features
# Gender not mentioned in documentation

DF1 <- DF %>% select(SEQN, HbA1c,  AGE, "BP_SYS" = BPXSY1,
                     "BP_DIA" = BPXDI1, "WEIGHT" = BMXWT, "HEIGHT" = BMXHT,
                     "BMI" = BMXBMI, "UPPER_LEG_LENGTH" = BMXLEG,
                     "UPPER_ARM_LENGTH" = BMXARML, "ARM_CIRCUM" = BMXARMC,
                     "WAIST_CIRCUMF" = BMXWAIST, "SAG_ABDOMINAL_DIA" = BMDAVSAD)


# 1) How does height vary with age? Are there any anomalies?
# Plot an age vs Height Graph
DF1 %>%
  filter(AGE<60) %>%
  ggplot(aes(x = as.factor(AGE), y = HEIGHT)) +
  geom_boxplot(color = "blue", fill = "blue",
               alpha = 0.2, notch = TRUE, notchwidth = 0.8, outlier.colour = "red",
               outlier.fill = "red", outlier.size = 3, outlier.alpha = 1) +
  labs(title = "Variation of height with age",
       x= "Age",
       y = "Height in cm")
```




```r
# Till about 18 years height increases with age, but after that it seems to
#be constant which is as expected
```

```r
# Removing observations less than 5 years old
DF1 <- subset(DF1, AGE>=5)

# Imputing missing values of remaining columns with median
f=function(x){
  x<-as.numeric(as.character(x)) #first convert each column into numeric if it
                                 #is from factor
  x[is.na(x)] =median(x, na.rm=TRUE) #convert the item with NA to median value
                                     #from the column
  x #display the column
}

Final_DF = data.frame(apply(DF1,2,f))
f_df <- data.frame(apply(DF1,2,f))

summary(Final_DF)
```

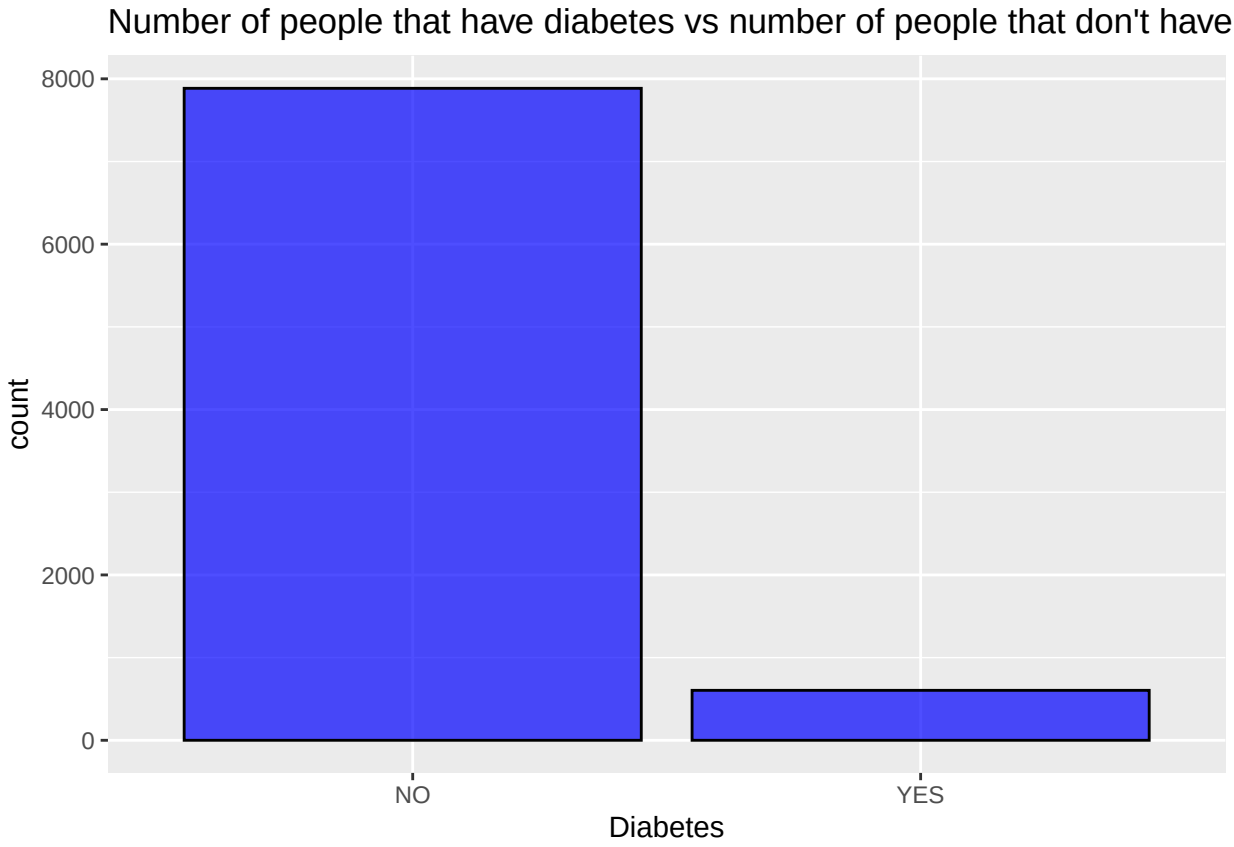
```
##       SEQN           HbA1c             AGE           BP_SYS
##  Min.   :73557   Min.   : 3.50   Min.   : 5.0   Min.   : 66.0
##  1st Qu.:76118   1st Qu.: 5.20   1st Qu.:15.0   1st Qu.:108.0
##  Median :78682   Median : 5.40   Median :34.0   Median :116.0
##  Mean   :78669   Mean   : 5.59   Mean   :36.3   Mean   :117.8
##  3rd Qu.:81210   3rd Qu.: 5.60   3rd Qu.:56.0   3rd Qu.:124.0
##  Max.   :83731   Max.   :17.50   Max.   :80.0   Max.   :228.0
##      BP_DIA           WEIGHT           HEIGHT           BMI
##  Min.   :  0.00   Min.   : 13.10   Min.   : 90.6   Min.   :12.10
##  1st Qu.: 60.00   1st Qu.: 53.30   1st Qu.:153.9   1st Qu.:20.90
##  Median : 66.00   Median : 69.80   Median :163.2   Median :25.50
##  Mean   : 65.81   Mean   : 70.38   Mean   :160.4   Mean   :26.39
##  3rd Qu.: 74.00   3rd Qu.: 86.60   3rd Qu.:171.8   3rd Qu.:30.60
##  Max.   :122.00   Max.   :222.60   Max.   :202.6   Max.   :82.90
##  UPPER_LEG_LENGTH UPPER_ARM_LENGTH   ARM_CIRCUM    WAIST_CIRCUMF
##  Min.   :24.40    Min.   :17.0     Min.   :13.90   Min.   : 41.50
##  1st Qu.:36.40    1st Qu.:33.5     1st Qu.:26.10   1st Qu.: 76.00
##  Median :38.60    Median :36.0     Median :30.50   Median : 89.80
##  Mean   :38.58    Mean   :35.4     Mean   :30.31   Mean   : 90.01
##  3rd Qu.:41.00    3rd Qu.:38.4     3rd Qu.:34.50   3rd Qu.:103.20
##  Max.   :51.90    Max.   :47.9     Max.   :59.40   Max.   :177.90
##  SAG_ABDOMINAL_DIA
##  Min.   :10.10
##  1st Qu.:17.90
##  Median :20.70
##  Mean   :21.05
##  3rd Qu.:23.60
##  Max.   :40.10
```

```r
# Changing the HbA1c values to yes or no diabetes

#No Diabetes 6.4 and lower Yes Diabetes 6.5 or higher

Final_DF$HbA1c <- cut(Final_DF$HbA1c,
                      breaks = c(-Inf, 6.5, Inf),
                      labels = c("NO", "YES"),
```

```
                      right = FALSE) # Interval is closed on the left and open
                                     #on right

names(Final_DF)[names(Final_DF) == "HbA1c"] <- "Diabetes" # Renaming column
str(Final_DF)
```

```
## 'data.frame':    8489 obs. of  13 variables:
##  $ SEQN             : num  73557 73558 73559 73560 73561 ...
##  $ Diabetes         : Factor w/ 2 levels "NO","YES": 2 2 2 1 1 1 1 1 1 1 ...
##  $ AGE              : num  69 54 72 9 73 56 61 56 65 26 ...
##  $ BP_SYS           : num  122 156 140 108 136 160 118 128 140 106 ...
##  $ BP_DIA           : num  72 62 90 38 86 84 80 74 78 60 ...
##  $ WEIGHT           : num  78.3 89.5 88.9 32.2 52 105 93.4 61.8 65.3 47.1 ...
##  $ HEIGHT           : num  171 177 175 137 162 ...
##  $ BMI              : num  26.7 28.6 28.9 17.1 19.7 41.7 35.7 26.5 22 20.3 ...
##  $ UPPER_LEG_LENGTH : num  39.2 40 40 33.5 36.3 34.2 37.1 32.4 40 34.4 ...
##  $ UPPER_ARM_LENGTH : num  40.2 41.5 41 29.5 37.5 36.2 39.3 33.5 40.3 32.6 ...
##  $ ARM_CIRCUM       : num  35.3 34.7 33.5 21 25.2 41.8 38 29 27.5 25.8 ...
##  $ WAIST_CIRCUMF    : num  100 107.6 109.2 61 89.8 ...
##  $ SAG_ABDOMINAL_DIA: num  20.6 24.4 25.6 14.9 20.7 29.1 26.7 19.9 20 14.5 ...
```

```
# 2) Number of people who have vs don't have diabetes
Final_DF %>%
  ggplot(aes(x = Diabetes)) + geom_bar(data = Final_DF, fill = "blue",
                                       color = "black", alpha  = 0.7) +
  labs(title = "Number of people that have diabetes vs number of people that don't have diabetes")
```



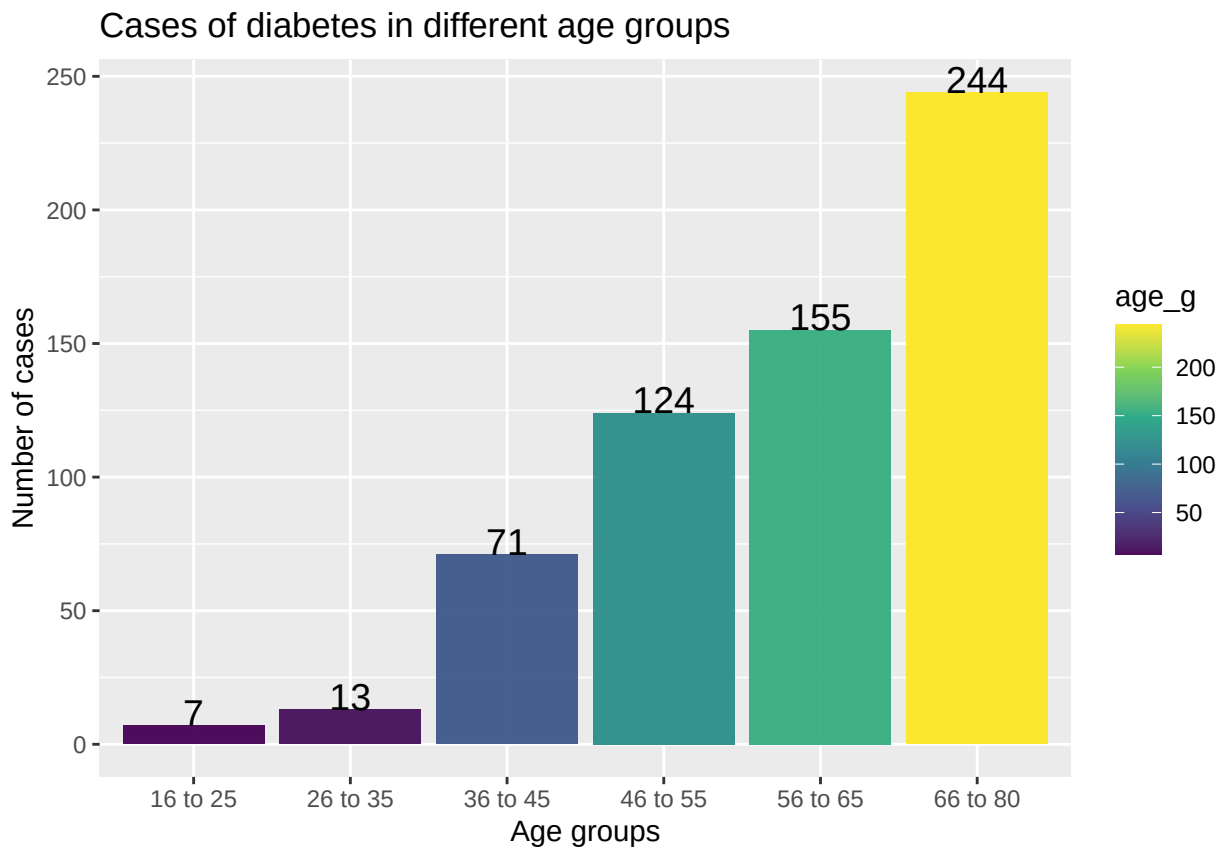

```
###
#EDA
###


#SEQN, HbA1c,  AGE, "BP_SYS" = BPXSY1,
#"BP_DIA" = BPXDI1, "WEIGHT" = BMXWT, "HEIGHT" = BMXHT,
#"BMI" = BMXBMI, "UPPER_LEG_LENGTH" = BMXLEG,
#"UPPER_ARM_LENGTH" = BMXARML, "ARM_CIRCUM" = BMXARMC,
#"WAIST_CIRCUMF" = BMXWAIST, "SAG_ABDOMINAL_DIA" = BMDAVSAD


# Make a dataset with people with only diabetes

diabetes_data <- Final_DF %>%
  filter(Diabetes == "YES")

x1 <- diabetes_data %>%
  filter(AGE>5) %>%
  filter(AGE<=15) %>%
  count(Diabetes == "YES")

x2 <- diabetes_data %>%
  filter(AGE>15) %>%
  filter(AGE<=25) %>%
  count(Diabetes == "YES")
```

```r
x3 <- diabetes_data %>%
  filter(AGE>25) %>%
  filter(AGE<=35) %>%
  count(Diabetes == "YES")

x4 <- diabetes_data %>%
  filter(AGE>35) %>%
  filter(AGE<=45) %>%
  count(Diabetes == "YES")

x5 <- diabetes_data %>%
  filter(AGE>45) %>%
  filter(AGE<=55) %>%
  count(Diabetes == "YES")

x6 <- diabetes_data %>%
  filter(AGE>55) %>%
  filter(AGE<=65) %>%
  count(Diabetes == "YES")

x7 <- diabetes_data %>%
  filter(AGE>65) %>%
  count(Diabetes == "YES")

age_groups <- x1 %>% full_join(x2)%>% full_join(x3)%>%
  full_join(x4)%>% full_join(x5)%>% full_join(x6)%>% full_join(x7)
```

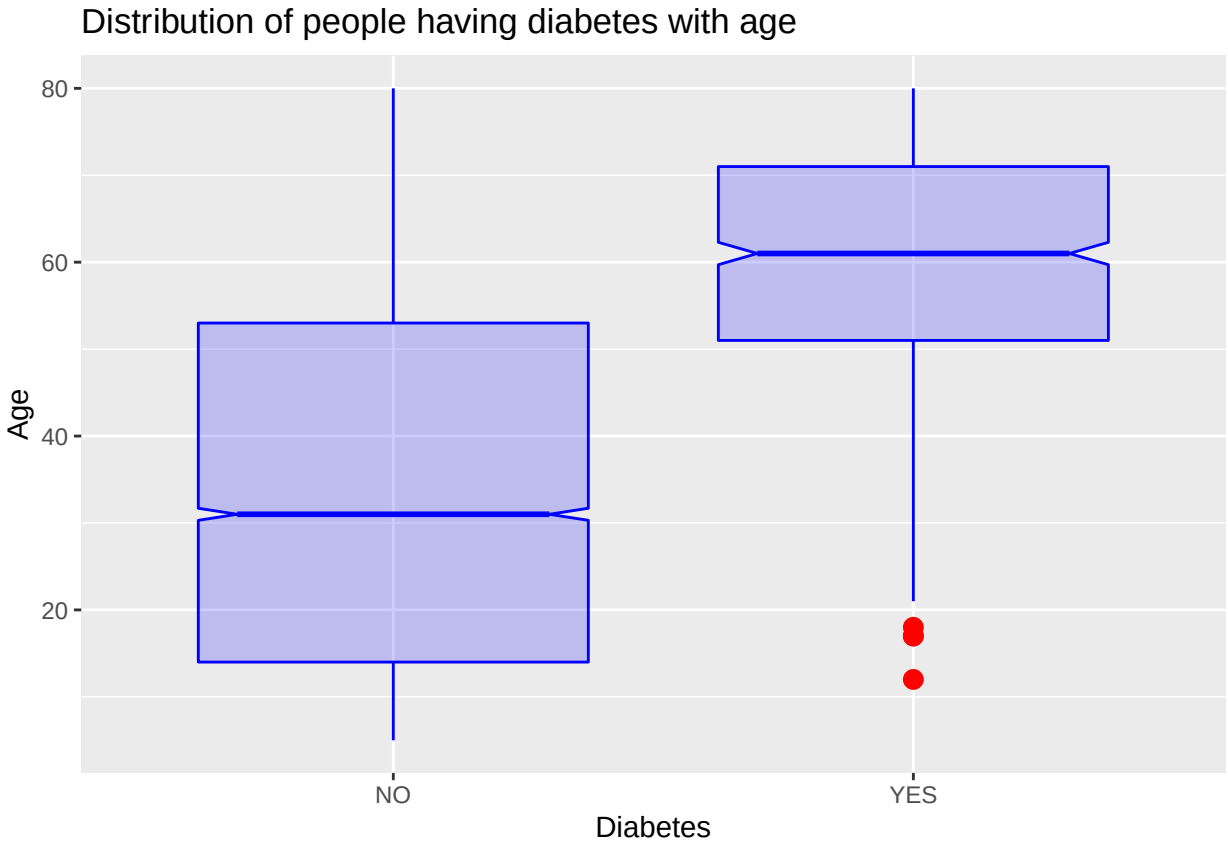
```
## Joining, by = c("Diabetes == \"YES\"", "n")
```

```
## Joining, by = c("Diabetes == \"YES\"", "n")
## Joining, by = c("Diabetes == \"YES\"", "n")
## Joining, by = c("Diabetes == \"YES\"", "n")
## Joining, by = c("Diabetes == \"YES\"", "n")
## Joining, by = c("Diabetes == \"YES\"", "n")
```

```r
col_names <- c("16 to 25", "26 to 35", "36 to 45", "46 to 55", "56 to 65",
               "66 to 80" )
age_g <- c( 7, 13, 71, 124, 155, 244)

x8 <- data.frame(col_names, age_g)
x8 <- x8 %>%
  group_by(age_g)


# 3) What age group seems to have diabetes more frequently

ggplot(x8, aes(x=col_names,y = age_g, fill = age_g)) +  geom_col() +
  scale_fill_viridis_c(alpha = 0.95) +
  geom_text(aes(label = age_g), vjust = 0.01, size = 5) +
  labs(title = "Cases of diabetes in different age groups",
       x = "Age groups",
       y = "Number of cases")
```




```r
# 4) Distribution of people having diabetes with AGE

ggplot(Final_DF, aes(x = Diabetes, y = AGE)) +
  geom_boxplot(color = "blue", fill = "blue", alpha = 0.2,
               notch = TRUE, notchwidth = 0.8, outlier.colour = "red",
                                                    outlier.fill = "red",
               outlier.size = 3, outlier.alpha = 1) +
  labs(title = "Distribution of people having diabetes with age",
       x = "Diabetes",
       y = "Age")
```

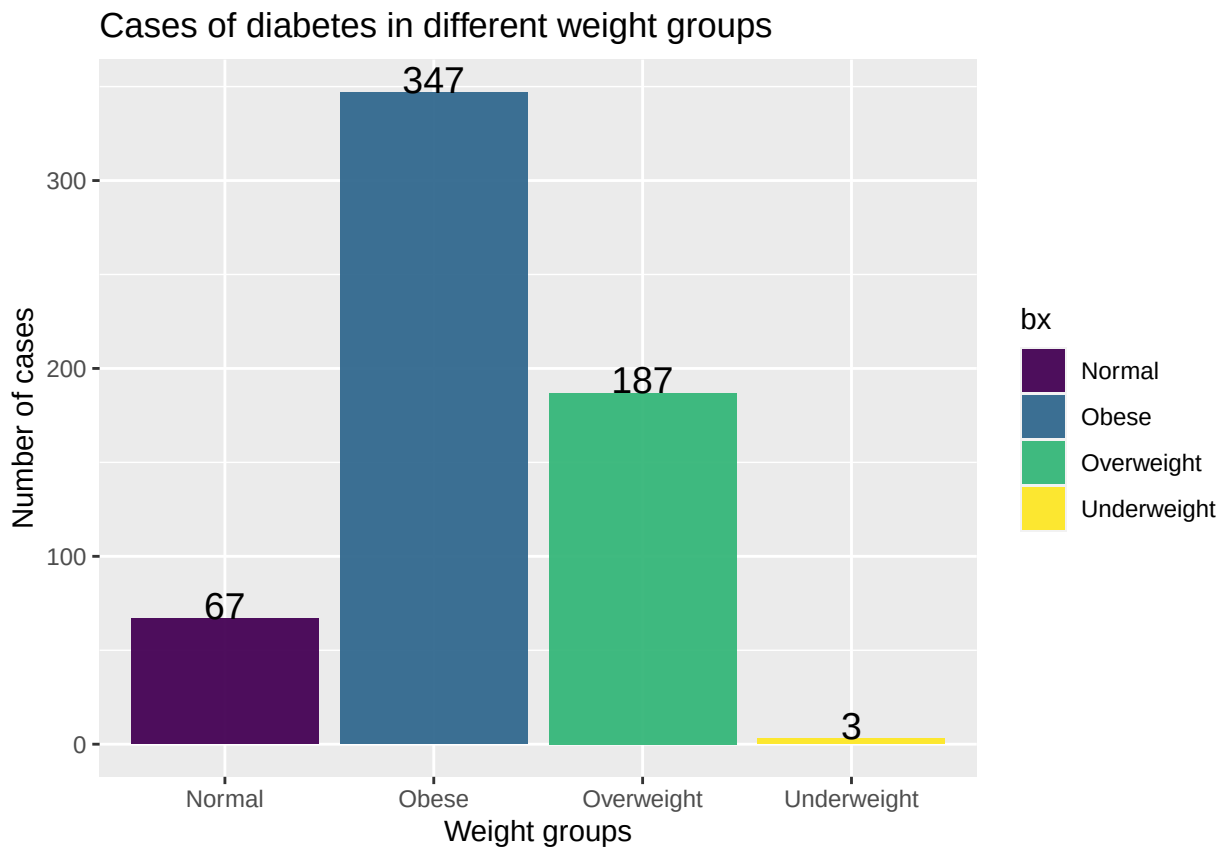



```
# 5) Relation between BMI and diabetes


bmi_under <- diabetes_data %>%
  filter(BMI<18.5) %>%
  count(Diabetes == "YES")

bmi_normal <- diabetes_data %>%
  filter(BMI>=18.5) %>%
  filter(BMI<25) %>%
  count(Diabetes == "YES")

bmi_over <- diabetes_data %>%
  filter(BMI>=25) %>%
  filter(BMI<30) %>%
  count(Diabetes == "YES")

bmi_obese <- diabetes_data %>%
  filter(BMI>=30) %>%
  count(Diabetes == "YES")

bmi_groups <- bmi_under %>% full_join(bmi_normal)%>% full_join(bmi_over)%>%
  full_join(bmi_obese)
```

```
## Joining, by = c("Diabetes == \"YES\"", "n")
```

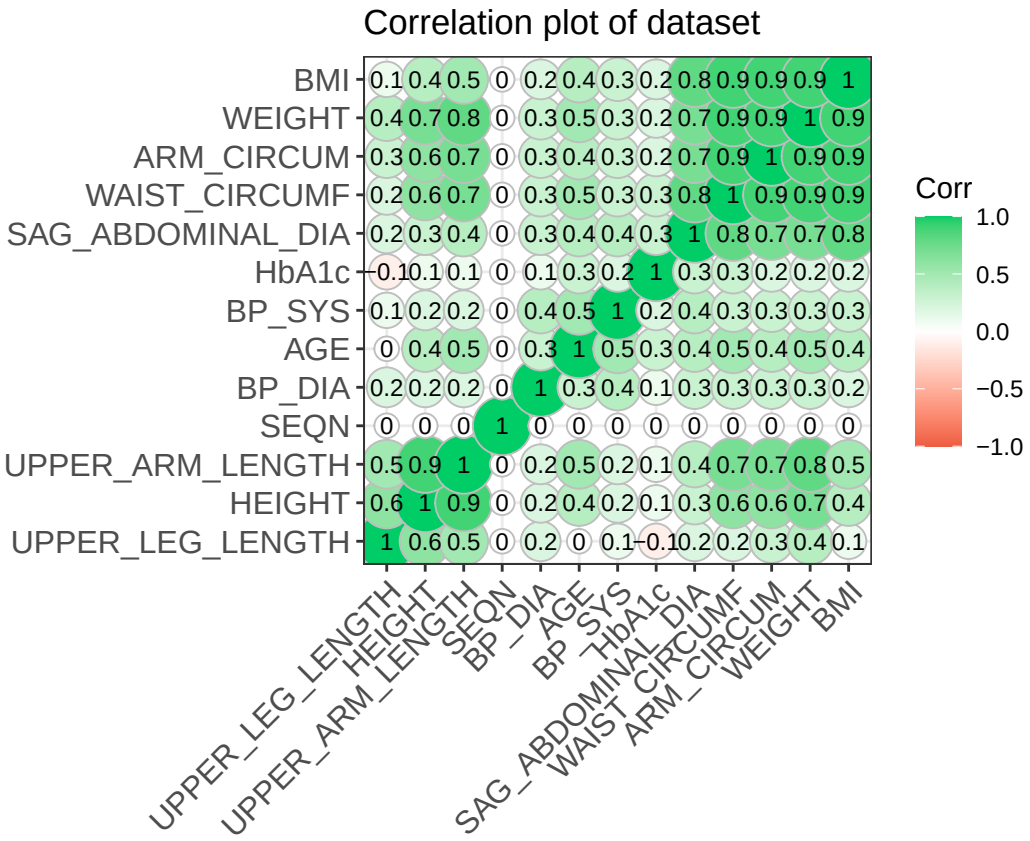
```
## Joining, by = c("Diabetes == \"YES\"", "n")
## Joining, by = c("Diabetes == \"YES\"", "n")
```

```
bx <- c("Underweight", "Normal", "Overweight", "Obese")
by <- c( 3, 67,187, 347)

bd <- data.frame(bx, by)

ggplot(bd, aes(x=bx,y = by, fill = bx)) +  geom_col() +
  scale_fill_viridis_d(alpha = 0.95) +
  geom_text(aes(label = by), vjust = 0.01, size = 5) +
  labs(title = "Cases of diabetes in different weight groups",
       x = "Weight groups",
       y = "Number of cases")
```




```
#Correlation Plot
library(ggcorrplot)

corr <- round(cor(f_df), 1)

ggcorrplot(corr, hc.order = TRUE,
           lab = TRUE,
           lab_size = 3,
           method = "circle",
           colors = c("tomato2", "white", "springgreen3"),
```

```
          title = "Correlation plot of dataset",
          ggtheme = theme_bw())
```




```
## Feature importance using Boruta Package

boruta_output <- Boruta(Diabetes ~ . , data=na.omit(Final_DF), maxRuns = 200,
                        doTrace=2)
```

```
##  1. run of importance source...

##  2. run of importance source...

##  3. run of importance source...

##  4. run of importance source...

##  5. run of importance source...

##  6. run of importance source...

##  7. run of importance source...

##  8. run of importance source...
```

```
##  9. run of importance source...

##  10. run of importance source...

##  11. run of importance source...

## After 11 iterations, +48 secs:

##  confirmed 11 attributes: AGE, ARM_CIRCUM, BMI, BP_DIA, BP_SYS and 6 more;

##  still have 1 attribute left.

##  12. run of importance source...

##  13. run of importance source...

##  14. run of importance source...

##  15. run of importance source...

##  16. run of importance source...

##  17. run of importance source...

##  18. run of importance source...

##  19. run of importance source...

##  20. run of importance source...

##  21. run of importance source...

## After 21 iterations, +1.5 mins:

##  rejected 1 attribute: SEQN;

##  no more attributes left.
```

```
print(boruta_output)
```

```
## Boruta performed 21 iterations in 1.48958 mins.
##  11 attributes confirmed important: AGE, ARM_CIRCUM, BMI, BP_DIA,
## BP_SYS and 6 more;
##  1 attributes confirmed unimportant: SEQN;
```

```
#names(boruta_output)

boruta_signif <- getSelectedAttributes(boruta_output, withTentative = TRUE)
print(boruta_signif)
```

```
##  [1] "AGE"               "BP_SYS"            "BP_DIA"
##  [4] "WEIGHT"            "HEIGHT"            "BMI"
##  [7] "UPPER_LEG_LENGTH"  "UPPER_ARM_LENGTH"  "ARM_CIRCUM"
## [10] "WAIST_CIRCUMF"     "SAG_ABDOMINAL_DIA"
```

```
roughFixMod <- TentativeRoughFix(boruta_output)
boruta_signif <- getSelectedAttributes(roughFixMod)
print(boruta_signif)
```

```
##  [1] "AGE"               "BP_SYS"            "BP_DIA"
##  [4] "WEIGHT"            "HEIGHT"            "BMI"
##  [7] "UPPER_LEG_LENGTH"  "UPPER_ARM_LENGTH"  "ARM_CIRCUM"
## [10] "WAIST_CIRCUMF"     "SAG_ABDOMINAL_DIA"
```

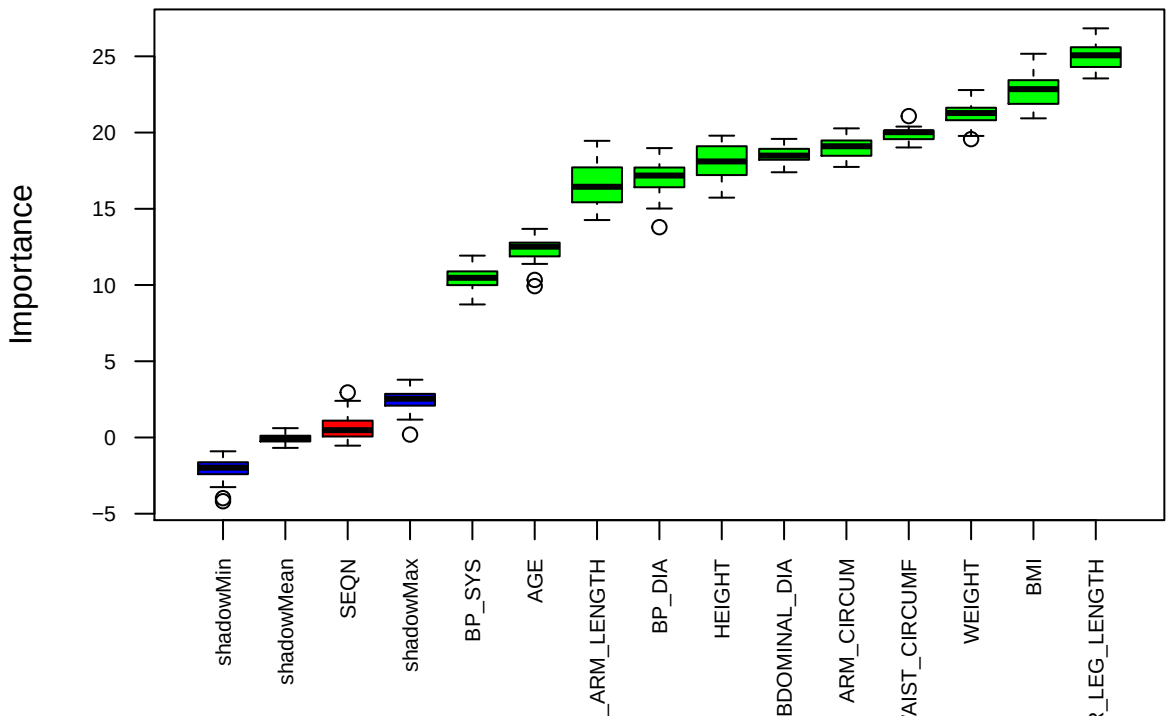
```
# Variable Importance Scores
imps <- attStats(roughFixMod)
imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]
imps2[order(-imps2$meanImp), ]  # descending sort
```

```
##                    meanImp  decision
## UPPER_LEG_LENGTH  25.05469 Confirmed
## BMI               22.80465 Confirmed
## WEIGHT            21.22875 Confirmed
## WAIST_CIRCUMF     19.89443 Confirmed
## ARM_CIRCUM        19.01839 Confirmed
## SAG_ABDOMINAL_DIA 18.52763 Confirmed
## HEIGHT            18.12503 Confirmed
## BP_DIA            16.99411 Confirmed
## UPPER_ARM_LENGTH  16.55303 Confirmed
## AGE               12.32013 Confirmed
## BP_SYS            10.36542 Confirmed
```

```
setDT(imps2, keep.rownames = TRUE)[]
```

```
##                    rn  meanImp  decision
##  1:               AGE 12.32013 Confirmed
##  2:            BP_SYS 10.36542 Confirmed
##  3:            BP_DIA 16.99411 Confirmed
##  4:            WEIGHT 21.22875 Confirmed
##  5:            HEIGHT 18.12503 Confirmed
##  6:               BMI 22.80465 Confirmed
##  7:  UPPER_LEG_LENGTH 25.05469 Confirmed
##  8:  UPPER_ARM_LENGTH 16.55303 Confirmed
##  9:        ARM_CIRCUM 19.01839 Confirmed
## 10:     WAIST_CIRCUMF 19.89443 Confirmed
## 11: SAG_ABDOMINAL_DIA 18.52763 Confirmed
```

```
# Plot variable importance
plot(boruta_output, cex.axis=.7, las=2, xlab="", main="Variable Importance")
```




```
## Remove SEQN from df

Final_DF_1 <- subset(Final_DF , select  = -c(SEQN))

#####################################
#Fitting Models
#####################################


#################
##Splitting Data
#################

dat_part <- resample_partition(Final_DF_1,
                               c(train = 0.8,
                               test = 0.2))


train <- as.tibble(dat_part$train)
test <- as.tibble(dat_part$test)

##############
#Oversampling
##############

train.rose <- ovun.sample(Diabetes~. , data = train, method = "over",
```

```
                          seed = 1)$data

table(train.rose$Diabetes)
```

```
##
##   NO  YES
## 6320 6268
```

```
table(train$Diabetes)
```

```
##
##   NO  YES
## 6320  471
```

```
summary(train)
```

```
##  Diabetes        AGE            BP_SYS          BP_DIA           WEIGHT
##  NO :6320   Min.   : 5.00   Min.   : 66.0   Min.   :  0.00   Min.   : 13.10
##  YES: 471   1st Qu.:15.00   1st Qu.:108.0   1st Qu.: 60.00   1st Qu.: 53.60
##             Median :34.00   Median :116.0   Median : 66.00   Median : 69.80
##             Mean   :36.34   Mean   :117.7   Mean   : 65.69   Mean   : 70.58
##             3rd Qu.:56.00   3rd Qu.:124.0   3rd Qu.: 74.00   3rd Qu.: 86.75
##             Max.   :80.00   Max.   :228.0   Max.   :122.00   Max.   :222.60
##      HEIGHT           BMI         UPPER_LEG_LENGTH UPPER_ARM_LENGTH
##  Min.   : 90.6   Min.   :12.10   Min.   :24.40    Min.   :17.00
##  1st Qu.:154.1   1st Qu.:21.00   1st Qu.:36.40    1st Qu.:33.50
##  Median :163.2   Median :25.50   Median :38.60    Median :36.00
##  Mean   :160.5   Mean   :26.43   Mean   :38.59    Mean   :35.43
##  3rd Qu.:171.9   3rd Qu.:30.60   3rd Qu.:41.00    3rd Qu.:38.40
##  Max.   :202.6   Max.   :82.90   Max.   :51.90    Max.   :47.90
##    ARM_CIRCUM    WAIST_CIRCUMF    SAG_ABDOMINAL_DIA
##  Min.   :13.90   Min.   : 41.50   Min.   :10.10
##  1st Qu.:26.20   1st Qu.: 76.20   1st Qu.:17.90
##  Median :30.50   Median : 89.80   Median :20.70
##  Mean   :30.34   Mean   : 90.14   Mean   :21.07
##  3rd Qu.:34.50   3rd Qu.:103.30   3rd Qu.:23.70
##  Max.   :59.40   Max.   :172.50   Max.   :40.10
```

```
summary(train.rose)
```

```
##  Diabetes        AGE            BP_SYS          BP_DIA           WEIGHT
##  NO :6320   Min.   : 5.00   Min.   : 66.0   Min.   :  0.00   Min.   : 13.10
##  YES:6268   1st Qu.:29.00   1st Qu.:112.0   1st Qu.: 60.00   1st Qu.: 65.00
##             Median :51.00   Median :118.0   Median : 66.00   Median : 78.40
##             Mean   :47.33   Mean   :123.3   Mean   : 67.01   Mean   : 80.41
##             3rd Qu.:66.00   3rd Qu.:134.0   3rd Qu.: 76.00   3rd Qu.: 95.70
##             Max.   :80.00   Max.   :228.0   Max.   :122.00   Max.   :222.60
##      HEIGHT           BMI         UPPER_LEG_LENGTH UPPER_ARM_LENGTH
##  Min.   : 90.6   Min.   :12.1   Min.   :24.4     Min.   :17.00
##  1st Qu.:156.6   1st Qu.:23.9   1st Qu.:35.7     1st Qu.:34.80
##  Median :164.7   Median :28.4   Median :38.6     Median :36.70
```

```
##  Mean   :163.4   Mean   :29.4   Mean   :38.2     Mean   :36.47
##  3rd Qu.:172.9   3rd Qu.:33.7   3rd Qu.:40.7     3rd Qu.:39.00
##  Max.   :202.6   Max.   :82.9   Max.   :51.9     Max.   :47.90
##    ARM_CIRCUM    WAIST_CIRCUMF    SAG_ABDOMINAL_DIA
##  Min.   :13.90   Min.   : 41.50   Min.   :10.10
##  1st Qu.:29.20   1st Qu.: 87.70   1st Qu.:20.40
##  Median :32.60   Median : 99.05   Median :22.50
##  Mean   :32.62   Mean   : 99.28   Mean   :23.28
##  3rd Qu.:36.40   3rd Qu.:112.60   3rd Qu.:26.60
##  Max.   :59.40   Max.   :172.50   Max.   :40.10
```

```
##############
#Undersampling
##############


train.rose_under <- ovun.sample(Diabetes~. , data = train, method = "under",
                                N = 964, seed = 1)$data

table(train.rose_under$Diabetes)
```

```
##
##  NO YES
## 493 471
```

```
table(train$Diabetes)
```

```
##
##   NO  YES
## 6320  471
```

```
summary(train)
```

```
##  Diabetes        AGE            BP_SYS          BP_DIA           WEIGHT
##  NO :6320   Min.   : 5.00   Min.   : 66.0   Min.   :  0.00   Min.   : 13.10
##  YES: 471   1st Qu.:15.00   1st Qu.:108.0   1st Qu.: 60.00   1st Qu.: 53.60
##             Median :34.00   Median :116.0   Median : 66.00   Median : 69.80
##             Mean   :36.34   Mean   :117.7   Mean   : 65.69   Mean   : 70.58
##             3rd Qu.:56.00   3rd Qu.:124.0   3rd Qu.: 74.00   3rd Qu.: 86.75
##             Max.   :80.00   Max.   :228.0   Max.   :122.00   Max.   :222.60
##      HEIGHT           BMI         UPPER_LEG_LENGTH UPPER_ARM_LENGTH
##  Min.   : 90.6   Min.   :12.10   Min.   :24.40    Min.   :17.00
##  1st Qu.:154.1   1st Qu.:21.00   1st Qu.:36.40    1st Qu.:33.50
##  Median :163.2   Median :25.50   Median :38.60    Median :36.00
##  Mean   :160.5   Mean   :26.43   Mean   :38.59    Mean   :35.43
##  3rd Qu.:171.9   3rd Qu.:30.60   3rd Qu.:41.00    3rd Qu.:38.40
##  Max.   :202.6   Max.   :82.90   Max.   :51.90    Max.   :47.90
##    ARM_CIRCUM    WAIST_CIRCUMF    SAG_ABDOMINAL_DIA
##  Min.   :13.90   Min.   : 41.50   Min.   :10.10
##  1st Qu.:26.20   1st Qu.: 76.20   1st Qu.:17.90
##  Median :30.50   Median : 89.80   Median :20.70
##  Mean   :30.34   Mean   : 90.14   Mean   :21.07
##  3rd Qu.:34.50   3rd Qu.:103.30   3rd Qu.:23.70
##  Max.   :59.40   Max.   :172.50   Max.   :40.10
```

```r
summary(train.rose_under)
```

```
##  Diabetes       AGE            BP_SYS          BP_DIA           WEIGHT
##  NO :493   Min.   : 5.00   Min.   : 72.0   Min.   :  0.00   Min.   : 14.2
##  YES:471   1st Qu.:30.00   1st Qu.:114.0   1st Qu.: 60.00   1st Qu.: 65.1
##            Median :52.00   Median :118.0   Median : 66.00   Median : 77.5
##            Mean   :47.74   Mean   :123.6   Mean   : 66.96   Mean   : 79.9
##            3rd Qu.:66.00   3rd Qu.:134.0   3rd Qu.: 74.00   3rd Qu.: 95.2
##            Max.   :80.00   Max.   :204.0   Max.   :112.00   Max.   :195.4
##      HEIGHT           BMI         UPPER_LEG_LENGTH UPPER_ARM_LENGTH
##  Min.   :103.1   Min.   :12.60   Min.   :26.00    Min.   :20.10
##  1st Qu.:156.6   1st Qu.:24.15   1st Qu.:36.00    1st Qu.:34.90
##  Median :164.4   Median :28.30   Median :38.60    Median :36.60
##  Mean   :163.2   Mean   :29.28   Mean   :38.26    Mean   :36.45
##  3rd Qu.:172.4   3rd Qu.:33.42   3rd Qu.:40.50    3rd Qu.:39.00
##  Max.   :202.6   Max.   :71.50   Max.   :51.90    Max.   :46.80
##    ARM_CIRCUM    WAIST_CIRCUMF    SAG_ABDOMINAL_DIA
##  Min.   :14.20   Min.   : 41.50   Min.   :11.90
##  1st Qu.:29.20   1st Qu.: 87.95   1st Qu.:20.50
##  Median :32.40   Median : 98.50   Median :22.40
##  Mean   :32.52   Mean   : 99.07   Mean   :23.28
##  3rd Qu.:36.20   3rd Qu.:112.03   3rd Qu.:26.32
##  Max.   :55.70   Max.   :161.00   Max.   :40.10
```

```r
#########################
#Logistic Regression
#########################

##################
# Normal Sample
##################

log_train <- train
log_test <- test

fit_log_normal <- glm(Diabetes ~ . , family = binomial(link = "logit"),
                      data = log_train)

summary(fit_log_normal)
```

```
##
## Call:
## glm(formula = Diabetes ~ ., family = binomial(link = "logit"),
##     data = log_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6246  -0.3633  -0.1743  -0.0768   3.2927
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -11.532152   2.592321  -4.449 8.64e-06 ***
```

```
## AGE                  0.042433   0.003771  11.253  < 2e-16 ***
## BP_SYS               0.011741   0.003109   3.776 0.000159 ***
## BP_DIA              -0.008955   0.004095  -2.187 0.028770 *
## WEIGHT              -0.007088   0.014427  -0.491 0.623196
## HEIGHT               0.037294   0.016703   2.233 0.025567 *
## BMI                  0.011801   0.039505   0.299 0.765159
## UPPER_LEG_LENGTH    -0.100827   0.022078  -4.567 4.95e-06 ***
## UPPER_ARM_LENGTH    -0.028332   0.033103  -0.856 0.392070
## ARM_CIRCUM           0.033358   0.022002   1.516 0.129485
## WAIST_CIRCUMF        0.004263   0.009240   0.461 0.644549
## SAG_ABDOMINAL_DIA    0.146868   0.028026   5.240 1.60e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3422.3  on 6790  degrees of freedom
## Residual deviance: 2538.0  on 6779  degrees of freedom
## AIC: 2562
##
## Number of Fisher Scoring iterations: 7
```

```
log_test <-
  log_test %>%
  add_predictions(fit_log_normal, type = "response") %>%
  mutate(pred_dia = ifelse(pred > 0.12,
                           "YES", "NO"),
         correct = Diabetes == pred_dia )


table(log_test$Diabetes, log_test$pred_dia)[,1:2] # Confusion Matrix
```

```
##
##         NO  YES
##   NO  1343  222
##   YES   60   73
```

```
# Sensitivity = 0.6885246

# Specificity = 0.8261421

# Accuracy = 0.8162544

###################
# Oversampling
#################

log_train_over <- train.rose
log_test_over <- test

fit_log_over <- glm(Diabetes ~ . , family = binomial(link = "logit"),
                    data = log_train_over)
summary(fit_log_over)
```

```
##
## Call:
## glm(formula = Diabetes ~ ., family = binomial(link = "logit"),
##     data = log_train_over)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.78023  -0.61224  -0.08069   0.74336   2.58414
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -10.869641   1.318447  -8.244  < 2e-16 ***
## AGE                 0.049885   0.001627  30.653  < 2e-16 ***
## BP_SYS              0.011280   0.001573   7.170 7.49e-13 ***
## BP_DIA             -0.004989   0.001977  -2.523   0.0116 *
## WEIGHT             -0.012812   0.007715  -1.661   0.0968 .
## HEIGHT              0.047557   0.008503   5.593 2.24e-08 ***
## BMI                 0.052907   0.021286   2.486   0.0129 *
## UPPER_LEG_LENGTH   -0.118967   0.009697 -12.268  < 2e-16 ***
## UPPER_ARM_LENGTH   -0.025298   0.015423  -1.640   0.1009
## ARM_CIRCUM          0.015696   0.009922   1.582   0.1137
## WAIST_CIRCUMF       0.008598   0.003783   2.273   0.0230 *
## SAG_ABDOMINAL_DIA   0.129030   0.012592  10.247  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 17450  on 12587  degrees of freedom
## Residual deviance: 11046  on 12576  degrees of freedom
## AIC: 11070
##
## Number of Fisher Scoring iterations: 6
```

```
log_test_over <-
  log_test_over %>%
  add_predictions(fit_log_over, type = "response") %>%
  mutate(pred_dia = ifelse(pred > 0.5,
                           "YES", "NO"),
         correct = Diabetes == pred_dia )


table(log_test_over$Diabetes, log_test_over$pred_dia)[,1:2] # Confusion Matrix
```

```
##
##         NO  YES
##   NO  1171  394
##   YES   26  107
```

```
#sensitivity = 0.852459

#specificity = 0.732868
```

```r
#accuracy = 0.0.7414065

##################
# Undersampling
#################

log_train_under <- train.rose_under
log_test_under <- test

fit_log_under <- glm(Diabetes ~ . , family = binomial(link = "logit"),
                     data = log_train_under)
summary(fit_log_under)
```

```
##
## Call:
## glm(formula = Diabetes ~ ., family = binomial(link = "logit"),
##     data = log_train_under)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8199  -0.6447  -0.0817   0.7464   2.6018
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -11.843795   3.852714  -3.074  0.00211 **
## AGE                 0.044722   0.005878   7.608 2.78e-14 ***
## BP_SYS              0.010542   0.005626   1.874  0.06096 .
## BP_DIA             -0.003661   0.007015  -0.522  0.60172
## WEIGHT             -0.015189   0.021990  -0.691  0.48973
## HEIGHT              0.061413   0.025104   2.446  0.01443 *
## BMI                 0.075009   0.058514   1.282  0.19988
## UPPER_LEG_LENGTH   -0.151973   0.036268  -4.190 2.79e-05 ***
## UPPER_ARM_LENGTH   -0.025603   0.055616  -0.460  0.64527
## ARM_CIRCUM          0.016594   0.034555   0.480  0.63108
## WAIST_CIRCUMF       0.004152   0.013954   0.298  0.76606
## SAG_ABDOMINAL_DIA   0.134150   0.046995   2.855  0.00431 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1335.89  on 963  degrees of freedom
## Residual deviance:  856.14  on 952  degrees of freedom
## AIC: 880.14
##
## Number of Fisher Scoring iterations: 6
```

```r
log_test_under <-
  log_test_under %>%
  add_predictions(fit_log_under, type = "response") %>%
  mutate(pred_dia = ifelse(pred > 0.5,
                           "YES", "NO"),
         correct = Diabetes == pred_dia )
```

```r
table(log_test_under$Diabetes, log_test_under$pred_dia)[,1:2] # Confusion Matrix
```

```
##
##         NO  YES
##   NO  1186  379
##   YES   30  103
```

```r
#sensitivity = 0.8852459

#specificity = 0.7277919

#accuracy = 0.7391048



######################
##Random Forests
######################

## Normal Sample


summary(train)
```

```
##  Diabetes        AGE            BP_SYS          BP_DIA           WEIGHT
##  NO :6320   Min.   : 5.00   Min.   : 66.0   Min.   :  0.00   Min.   : 13.10
##  YES: 471   1st Qu.:15.00   1st Qu.:108.0   1st Qu.: 60.00   1st Qu.: 53.60
##             Median :34.00   Median :116.0   Median : 66.00   Median : 69.80
##             Mean   :36.34   Mean   :117.7   Mean   : 65.69   Mean   : 70.58
##             3rd Qu.:56.00   3rd Qu.:124.0   3rd Qu.: 74.00   3rd Qu.: 86.75
##             Max.   :80.00   Max.   :228.0   Max.   :122.00   Max.   :222.60
##      HEIGHT           BMI        UPPER_LEG_LENGTH UPPER_ARM_LENGTH
##  Min.   : 90.6   Min.   :12.10   Min.   :24.40    Min.   :17.00
##  1st Qu.:154.1   1st Qu.:21.00   1st Qu.:36.40    1st Qu.:33.50
##  Median :163.2   Median :25.50   Median :38.60    Median :36.00
##  Mean   :160.5   Mean   :26.43   Mean   :38.59    Mean   :35.43
##  3rd Qu.:171.9   3rd Qu.:30.60   3rd Qu.:41.00    3rd Qu.:38.40
##  Max.   :202.6   Max.   :82.90   Max.   :51.90    Max.   :47.90
##    ARM_CIRCUM    WAIST_CIRCUMF    SAG_ABDOMINAL_DIA
##  Min.   :13.90   Min.   : 41.50   Min.   :10.10
##  1st Qu.:26.20   1st Qu.: 76.20   1st Qu.:17.90
##  Median :30.50   Median : 89.80   Median :20.70
##  Mean   :30.34   Mean   : 90.14   Mean   :21.07
##  3rd Qu.:34.50   3rd Qu.:103.30   3rd Qu.:23.70
##  Max.   :59.40   Max.   :172.50   Max.   :40.10
```

```r
model_normal_rf <- randomForest(Diabetes ~ ., data = train, ntree = 1000,
                                proximity = TRUE) # Prox = true returns
                                                  #proximity matrix

model_normal_rf
```

```
##
## Call:
##  randomForest(formula = Diabetes ~ ., data = train, ntree = 1000,      proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 7.16%
## Confusion matrix:
##       NO YES class.error
## NO  6289  31 0.004905063
## YES  455  16 0.966029724
```

```
prediction <- as.data.frame((predict(model_normal_rf, newdata = test)))

confusionMatrix((predict(model_normal_rf, newdata = test)), test$Diabetes,
                positive = "YES") # Bad Sensitivity but good specificity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1562  128
##        YES    3    5
##
##                Accuracy : 0.9229
##                  95% CI : (0.9091, 0.9351)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 0.4511
##
##                   Kappa : 0.0626
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.037594
##             Specificity : 0.998083
##          Pos Pred Value : 0.625000
##          Neg Pred Value : 0.924260
##              Prevalence : 0.078327
##          Detection Rate : 0.002945
##    Detection Prevalence : 0.004711
##       Balanced Accuracy : 0.517839
##
##        'Positive' Class : YES
##
```

```
#sensitivity = 0.049180

#specificity = 0.995558

#accuracy = 0.9276

# save the model to disk
```

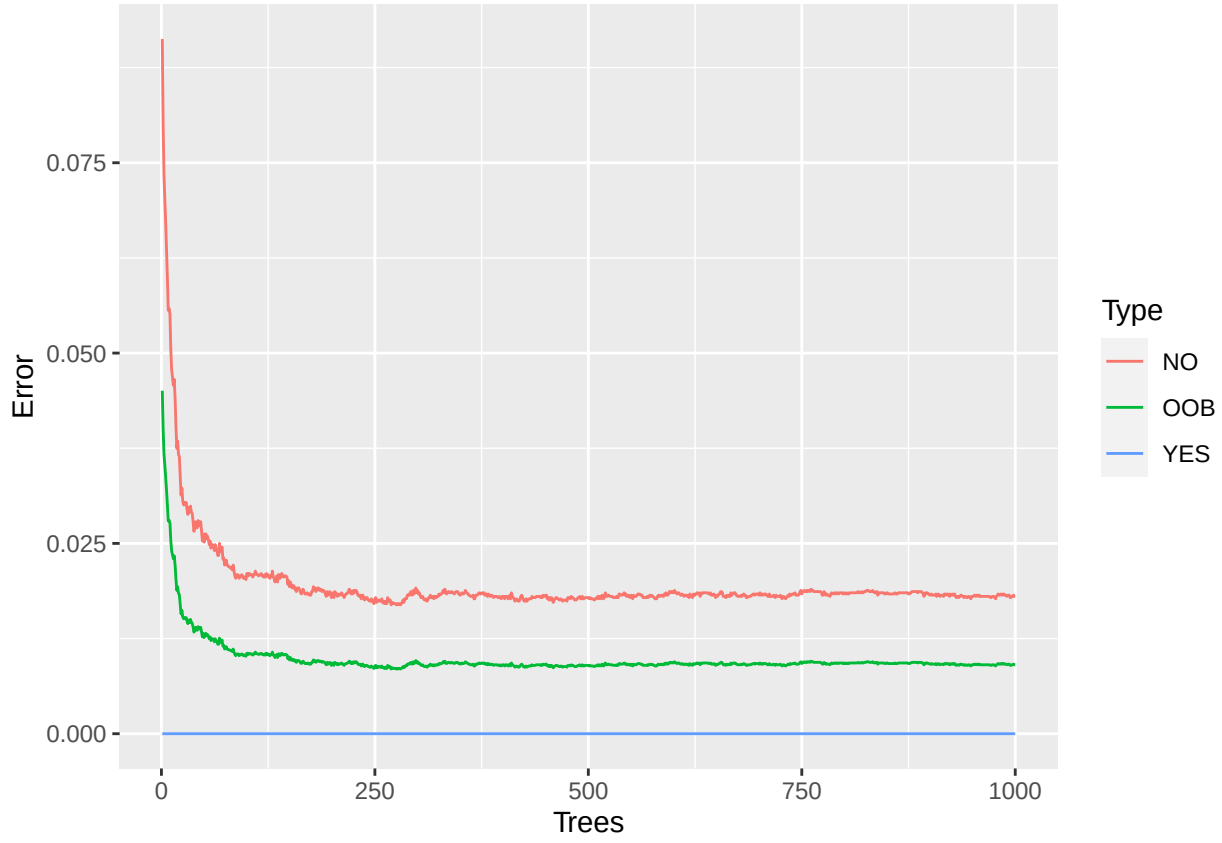
```r
saveRDS(model_normal_rf, "C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-

## Oversampling


model <- randomForest(Diabetes ~ ., data = train.rose, ntree = 1000,
                      proximity = TRUE) # Prox = true returns proximity matrix

model
```

```
##
## Call:
##  randomForest(formula = Diabetes ~ ., data = train.rose, ntree = 1000,      proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 0.91%
## Confusion matrix:
##       NO  YES class.error
## NO  6206  114  0.01803797
## YES    0 6268  0.00000000
```

```r
confusionMatrix((predict(model, newdata = test)), test$Diabetes,
                positive = "YES") # Bad Sensitivity but good specificity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1541  116
##        YES   24   17
##
##                Accuracy : 0.9176
##                  95% CI : (0.9034, 0.9302)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 0.7531
##
##                   Kappa : 0.1646
##
##  Mcnemar's Test P-Value : 1.461e-14
##
##             Sensitivity : 0.12782
##             Specificity : 0.98466
##          Pos Pred Value : 0.41463
##          Neg Pred Value : 0.92999
##              Prevalence : 0.07833
##          Detection Rate : 0.01001
##    Detection Prevalence : 0.02415
##       Balanced Accuracy : 0.55624
##
##        'Positive' Class : YES
##
```

```r
#sensitivity = 0.07377

#specificity = 0.98731

#accuracy = 0.9217

oob.error.data <- data.frame(
  Trees = rep(1:nrow(model$err.rate), times=3),
  Type = rep(c("OOB", "NO", "YES"), each = nrow(model$err.rate)),
  Error=c(model$err.rate[,"OOB"],
          model$err.rate[,"NO"],
          model$err.rate[,"YES"]))

ggplot(data = oob.error.data, aes(x=Trees, y=Error)) +
  geom_line(aes(color = Type))
```




```r
oob.values <- vector(length = 12)
for(i in 1:12){
  temp.model <- randomForest(Diabetes~., data = train.rose, mtry=i,
                             ntree = 1000)
  oob.values[i] <- temp.model$err.rate[nrow(temp.model$err.rate),1]
}

oob.values # mtry = 1 gave the best oob
```

```
##  [1] 0.005878615 0.008102955 0.008817922 0.010168414 0.010962822 0.012313314
##  [7] 0.012472196 0.013504925 0.015729266 0.016523673 0.017556403 0.017635844
```

```
# save the model to disk
saveRDS(model, "C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nutrition-


#code to load model
#super_model <- readRDS("")
#print(super_model)

## Undersampling


model_under <- randomForest(Diabetes ~ ., data = train.rose_under, ntree = 1000,
                            proximity = TRUE) # Prox = true returns proximity
                                                                    #matrix

model_under
```

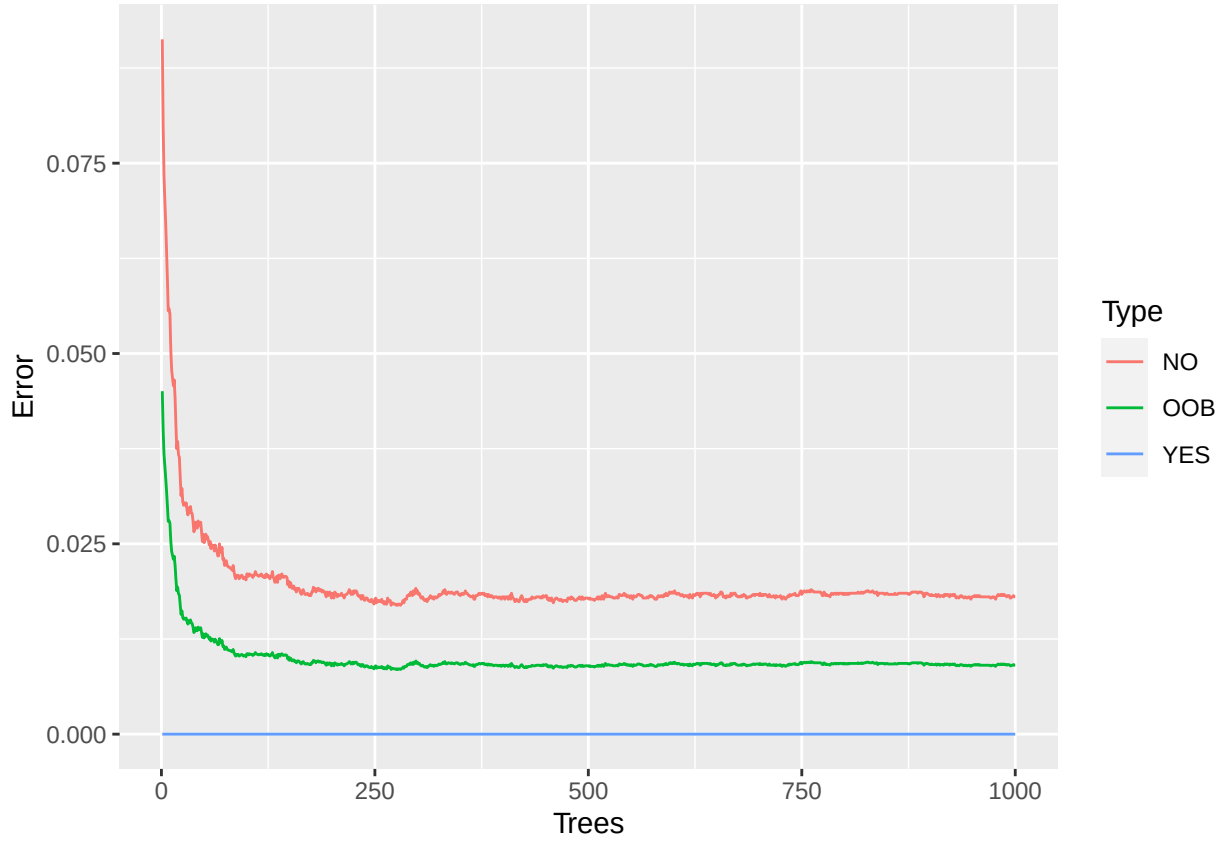
```
##
## Call:
##  randomForest(formula = Diabetes ~ ., data = train.rose_under,      ntree = 1000, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 20.75%
## Confusion matrix:
##      NO YES class.error
## NO  364 129   0.2616633
## YES  71 400   0.1507431
```

```
confusionMatrix(predict(model_under, test), test$Diabetes, positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1160   31
##        YES  405  102
##
##               Accuracy : 0.7432
##                 95% CI : (0.7217, 0.7639)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1
##
##                  Kappa : 0.2222
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.76692
##             Specificity : 0.74121
```

```
##          Pos Pred Value : 0.20118
##          Neg Pred Value : 0.97397
##              Prevalence : 0.07833
##          Detection Rate : 0.06007
##    Detection Prevalence : 0.29859
##       Balanced Accuracy : 0.75407
##
##        'Positive' Class : YES
##
```

```r
# Good sensitivity

#sensitivity = Sensitivity : 0.90164

#specificity = 0.71447

#accuracy = 0.7279

oob.error.data <- data.frame(
  Trees = rep(1:nrow(model$err.rate), times=3),
  Type = rep(c("OOB", "NO", "YES"), each = nrow(model$err.rate)),
  Error=c(model$err.rate[,"OOB"],
          model$err.rate[,"NO"],
          model$err.rate[,"YES"]))

ggplot(data = oob.error.data, aes(x=Trees, y=Error)) +
  geom_line(aes(color = Type))
```




```
oob.values <- vector(length = 12)
for(i in 1:12){
  temp.model <- randomForest(Diabetes~., data = train.rose, mtry=i,
                             ntree = 1000)
  oob.values[i] <- temp.model$err.rate[nrow(temp.model$err.rate),1]
}  ##ntrees = 630 seems to be the best

oob.values # mtry = 1 gave the best oob
```

```
##  [1] 0.006037496 0.007785192 0.009056244 0.010168414 0.010565618 0.012233874
##  [7] 0.013584366 0.013902129 0.014617096 0.015252622 0.017079759 0.017079759
```

```
saveRDS(model_under, "C:/Semester 2/Intro to Data Mining and Processing/Project/national-health-and-nut


#tuning the model

model_under_tuned <- randomForest(Diabetes ~ ., data = train.rose_under,
                                  ntree = 1000, proximity = TRUE)
# Prox = true returns proximity matrix
model_under_tuned
```

```
##
## Call:
##  randomForest(formula = Diabetes ~ ., data = train.rose_under,      ntree = 1000, proximity = TRUE)
```

```
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 20.85%
## Confusion matrix:
##      NO YES class.error
## NO  365 128   0.2596349
## YES  73 398   0.1549894
```

```
confusionMatrix(predict(model_under_tuned, test), test$Diabetes,
                positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1160   27
##        YES  405  106
##
##                Accuracy : 0.7456
##                  95% CI : (0.7242, 0.7662)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.234
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.79699
##             Specificity : 0.74121
##          Pos Pred Value : 0.20744
##          Neg Pred Value : 0.97725
##              Prevalence : 0.07833
##          Detection Rate : 0.06243
##    Detection Prevalence : 0.30094
##       Balanced Accuracy : 0.76910
##
##        'Positive' Class : YES
##
```

```
#Sensitivity : 0.90984
#Specificity : 0.71256
#Accuracy : 0.7267


###################
##xgboost
###################

###################
##NORMAL DATA
###################
```

```r
df_train_0 <- train.rose
df_test_0 <- test

setDT(df_train_0)
setDT(df_test_0)

# Using one hot encoding

labels_0 <- df_train_0$Diabetes
ts_label_0 <- df_test_0$Diabetes

new_tr_0 <- model.matrix(~.+0, data = df_train_0[,-c("Diabetes"), with=F])

new_ts_0 <- model.matrix(~.+0, data = df_test_0[,-c("Diabetes"), with=F])

#convert factor to numeric
labels_0 <- as.numeric(labels_0)-1
ts_label_0 <- as.numeric(ts_label_0)-1

dtrain_0 <- xgb.DMatrix(data = new_tr_0,label = labels_0)
dtest_0 <- xgb.DMatrix(data = new_ts_0,label=ts_label_0)

#default parameters
params <- list(booster = "gbtree", objective = "binary:logistic", eta=0.3,
               gamma=0, max_depth=6, min_child_weight=1, subsample=1,
               colsample_bytree=1)

xgbcv_0 <- xgb.cv( params = params, data = dtrain_0, nrounds = 100, nfold = 5,
                   showsd = T, stratified = T, print.every.n = 10,
                   early.stop.round = 20, maximize = F)
```

```
## [1]  train-error:0.152546+0.002483   test-error:0.160630+0.006881
## Multiple eval metrics are present. Will use test_error for early stopping.
## Will train until test_error hasn't improved in 20 rounds.
##
## [11] train-error:0.100652+0.002405   test-error:0.113283+0.005454
## [21] train-error:0.073006+0.003341   test-error:0.093503+0.005552
## [31] train-error:0.048677+0.002178   test-error:0.073881+0.006044
## [41] train-error:0.028837+0.001852   test-error:0.059104+0.004167
## [51] train-error:0.018529+0.001077   test-error:0.052431+0.003392
## [61] train-error:0.011618+0.000785   test-error:0.045758+0.003083
## [71] train-error:0.006117+0.000324   test-error:0.039641+0.002927
## [81] train-error:0.003555+0.000291   test-error:0.035669+0.002546
## [91] train-error:0.001966+0.000368   test-error:0.032412+0.002671
## [100]    train-error:0.001231+0.000342   test-error:0.031618+0.002422
```

```r
# lowest in 100th iteration

#first default - model training
xgb1_0 <- xgb.train (params = params, data = dtrain_0, nrounds = 79, watchlist =
                     list(val=dtest_0,train=dtrain_0), print.every.n = 10,
                     early.stop.round = 10, maximize = F ,
                     eval_metric = "error")
```

```
## [17:19:17] WARNING: amalgamation/../src/learner.cc:480:
## Parameters: { early_stop_round, print_every_n } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
##
##
## [1]  val-error:0.272674  train-error:0.151970
## Multiple eval metrics are present. Will use train_error for early stopping.
## Will train until train_error hasn't improved in 10 rounds.
##
## [11] val-error:0.220848  train-error:0.100493
## [21] val-error:0.192580  train-error:0.080473
## [31] val-error:0.163722  train-error:0.049571
## [41] val-error:0.146054  train-error:0.033445
## [51] val-error:0.133098  train-error:0.020416
## [61] val-error:0.118963  train-error:0.012313
## [71] val-error:0.108363  train-error:0.006435
## [79] val-error:0.107774  train-error:0.004131
```

```
#model prediction
xgbpred_0 <- predict (xgb1_0,dtest_0)
xgbpred_0 <- ifelse (xgbpred_0 > 0.5,'YES','NO')
xgbpred1_0 <- as.factor(xgbpred_0)
confusionMatrix(xgbpred1_0 , df_test_0$Diabetes, positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1469   86
##        YES   96   47
##
##                Accuracy : 0.8928
##                  95% CI : (0.8771, 0.9071)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.2823
##
##  Mcnemar's Test P-Value : 0.5047
##
##             Sensitivity : 0.35338
##             Specificity : 0.93866
##          Pos Pred Value : 0.32867
##          Neg Pred Value : 0.94469
##              Prevalence : 0.07833
##          Detection Rate : 0.02768
##    Detection Prevalence : 0.08422
##       Balanced Accuracy : 0.64602
##
##        'Positive' Class : YES
##
```

```r
# Bad sensitivity

#Sensitivity : 0.27049
#Specificity : 0.93147
#Accuracy : 0.884

###################
##OVERSAMPLED DATA
###################
df_train <- train.rose
df_test <- test

setDT(df_train)
setDT(df_test)

# Using one hot encoding

labels <- df_train$Diabetes
ts_label <- df_test$Diabetes

new_tr <- model.matrix(~.+0, data = df_train[,-c("Diabetes"), with=F])

new_ts <- model.matrix(~.+0, data = df_test[,-c("Diabetes"), with=F])

#convert factor to numeric
labels <- as.numeric(labels)-1
ts_label <- as.numeric(ts_label)-1

dtrain <- xgb.DMatrix(data = new_tr,label = labels)
dtest <- xgb.DMatrix(data = new_ts,label=ts_label)

#default parameters
params <- list(booster = "gbtree", objective = "binary:logistic", eta=0.3,
               gamma=0, max_depth=6, min_child_weight=1, subsample=1,
               colsample_bytree=1)

xgbcv <- xgb.cv( params = params, data = dtrain, nrounds = 100, nfold = 5,
                 showsd = T, stratified = T, print.every.n = 10,
                 early.stop.round = 20, maximize = F)
```

```
## [1]  train-error:0.155009+0.003644   test-error:0.163012+0.004003
## Multiple eval metrics are present. Will use test_error for early stopping.
## Will train until test_error hasn't improved in 20 rounds.
##
## [11] train-error:0.098526+0.003562   test-error:0.113758+0.007134
## [21] train-error:0.074198+0.001372   test-error:0.093104+0.002586
## [31] train-error:0.046751+0.002370   test-error:0.071893+0.005021
## [41] train-error:0.028539+0.001770   test-error:0.058865+0.004396
## [51] train-error:0.018212+0.000991   test-error:0.051954+0.005838
## [61] train-error:0.010605+0.001297   test-error:0.044645+0.005286
## [71] train-error:0.005680+0.000484   test-error:0.038846+0.004493
## [81] train-error:0.002900+0.000500   test-error:0.035907+0.004191
## [91] train-error:0.001708+0.000346   test-error:0.034000+0.003694
```

```
## [100]    train-error:0.001251+0.000204   test-error:0.033206+0.003949
```

```
# lowest in 100th iteration

#first default - model training
xgb1 <- xgb.train (params = params, data = dtrain, nrounds = 79, watchlist =
                     list(val=dtest,train=dtrain), print.every.n = 10,
                   early.stop.round = 10, maximize = F , eval_metric = "error")
```

```
## [17:19:21] WARNING: amalgamation/../src/learner.cc:480:
## Parameters: { early_stop_round, print_every_n } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
##
##
## [1]  val-error:0.272674  train-error:0.151970
## Multiple eval metrics are present. Will use train_error for early stopping.
## Will train until train_error hasn't improved in 10 rounds.
##
## [11] val-error:0.220848  train-error:0.100493
## [21] val-error:0.192580  train-error:0.080473
## [31] val-error:0.163722  train-error:0.049571
## [41] val-error:0.146054  train-error:0.033445
## [51] val-error:0.133098  train-error:0.020416
## [61] val-error:0.118963  train-error:0.012313
## [71] val-error:0.108363  train-error:0.006435
## [79] val-error:0.107774  train-error:0.004131
```

```
#model prediction
xgbpred <- predict (xgb1,dtest)
xgbpred <- ifelse (xgbpred > 0.5,'YES','NO')
xgbpred1 <- as.factor(xgbpred)
confusionMatrix(xgbpred1 , df_test$Diabetes, positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1469   86
##        YES   96   47
##
##                Accuracy : 0.8928
##                  95% CI : (0.8771, 0.9071)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.2823
##
##  Mcnemar's Test P-Value : 0.5047
##
##             Sensitivity : 0.35338
```

```
##             Specificity : 0.93866
##          Pos Pred Value : 0.32867
##          Neg Pred Value : 0.94469
##              Prevalence : 0.07833
##          Detection Rate : 0.02768
##    Detection Prevalence : 0.08422
##       Balanced Accuracy : 0.64602
##
##        'Positive' Class : YES
##
```

```
# very low sensitivity but high specificity

#Sensitivity : 0.27049
#Specificity : 0.93147
#Accuracy : 0.884

####################
##UNDERSAMPLED DATA
####################

df_train1 <- train.rose_under
df_test1 <- test

setDT(df_train1)
setDT(df_test1)

# Using one hot encoding

labels1 <- df_train1$Diabetes
ts_label1 <- df_test1$Diabetes

new_tr1 <- model.matrix(~.+0, data = df_train1[,-c("Diabetes"), with=F])

new_ts1 <- model.matrix(~.+0, data = df_test1[,-c("Diabetes"), with=F])
#convert factor to numeric
labels1 <- as.numeric(labels1)-1
ts_label1 <- as.numeric(ts_label1)-1

dtrain1 <- xgb.DMatrix(data = new_tr1,label = labels1)
dtest1 <- xgb.DMatrix(data = new_ts1,label=ts_label1)

#default parameters

xgbcv1 <- xgb.cv( params = params, data = dtrain1, nrounds = 100,
                  nfold = 5, showsd = T, stratified = T, print.every.n = 10,
                  early.stop.round = 20, maximize = F)
```

```
## [1]  train-error:0.148340+0.010841   test-error:0.243771+0.036014
## Multiple eval metrics are present. Will use test_error for early stopping.
## Will train until test_error hasn't improved in 20 rounds.
##
## [11] train-error:0.068983+0.012089   test-error:0.220947+0.027666
## [21] train-error:0.029043+0.006480   test-error:0.205397+0.020386
```

```
## [31] train-error:0.006224+0.001720   test-error:0.217832+0.030496
## [41] train-error:0.002334+0.001271   test-error:0.218890+0.024266
## Stopping. Best iteration:
## [21] train-error:0.029043+0.006480   test-error:0.205397+0.020386
```

```
# lowest in 6th iteration

#first default - model training
xgb1_1 <- xgb.train (params = params, data = dtrain1, nrounds = 79, watchlist =
                     list(val=dtest1,train=dtrain1), print.every.n = 10,
                     early.stop.round = 10, maximize = F ,
                     eval_metric = "error")
```

```
## [17:19:23] WARNING: amalgamation/../src/learner.cc:480:
## Parameters: { early_stop_round, print_every_n } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
##
##
## [1]  val-error:0.269729  train-error:0.153527
## Multiple eval metrics are present. Will use train_error for early stopping.
## Will train until train_error hasn't improved in 10 rounds.
##
## [11] val-error:0.263251  train-error:0.097510
## [21] val-error:0.259717  train-error:0.040456
## [31] val-error:0.256773  train-error:0.022822
## [41] val-error:0.249706  train-error:0.007261
## [51] val-error:0.254417  train-error:0.003112
## [61] val-error:0.252061  train-error:0.002075
## Stopping. Best iteration:
## [56] val-error:0.247939  train-error:0.002075
```

```
#model prediction
xgbpred11 <- predict (xgb1_1,dtest1)
xgbpred11 <- ifelse (xgbpred11 > 0.5,'YES','NO')
xgbpred12 <- as.factor(xgbpred11)
confusionMatrix(xgbpred12 , df_test$Diabetes, positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1175   31
##        YES  390  102
##
##                Accuracy : 0.7521
##                  95% CI : (0.7308, 0.7724)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2316
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.76692
##             Specificity : 0.75080
##          Pos Pred Value : 0.20732
##          Neg Pred Value : 0.97430
##              Prevalence : 0.07833
##          Detection Rate : 0.06007
##    Detection Prevalence : 0.28975
##       Balanced Accuracy : 0.75886
##
##        'Positive' Class : YES
##
```

```
#Sensitivity : 0.86885
#Specificity : 0.72145
#Accuracy : 0.732
```

```
###################
##SVM
###################

##################
#normal sample
#################
svm_model_normal <- svm(Diabetes ~ ., data=train)
summary(svm_model_normal)
```

```
##
## Call:
## svm(formula = Diabetes ~ ., data = train)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  1244
##
##  ( 471 773 )
##
##
## Number of Classes:  2
##
## Levels:
##  NO YES
```

```
confusionMatrix(predict (svm_model_normal, test) , df_test$Diabetes,
                positive = "YES")
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   NO  YES
##        NO  1565  133
##        YES    0    0
##
##                Accuracy : 0.9217
##                  95% CI : (0.9079, 0.934)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 0.5231
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.92167
##              Prevalence : 0.07833
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        'Positive' Class : YES
##
```

```r
# Sensitivity : 0.00000
# Specificity : 1.00000
# Accuracy : 0.9282

###################
#oversampled
#################

svm_model_over <- svm(Diabetes ~ ., data=train.rose)
summary(svm_model_over)
```

```
##
## Call:
## svm(formula = Diabetes ~ ., data = train.rose)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  5483
##
##  ( 2654 2829 )
##
##
## Number of Classes:  2
```

```
##
## Levels:
##  NO YES
```

```
confusionMatrix(predict (svm_model_over, test) , df_test$Diabetes,
                positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1156   30
##        YES  409  103
##
##                Accuracy : 0.7415
##                  95% CI : (0.7199, 0.7621)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2227
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.77444
##             Specificity : 0.73866
##          Pos Pred Value : 0.20117
##          Neg Pred Value : 0.97470
##              Prevalence : 0.07833
##          Detection Rate : 0.06066
##    Detection Prevalence : 0.30153
##       Balanced Accuracy : 0.75655
##
##        'Positive' Class : YES
##
```

```
#Sensitivity : 0.88525
#Specificity : 0.71510
#Accuracy : 0.7273


##################
#undersampled
#################

svm_model_under <- svm(Diabetes ~ ., data=train.rose_under)
summary(svm_model_under)
```

```
##
## Call:
## svm(formula = Diabetes ~ ., data = train.rose_under)
##
##
## Parameters:
```

```
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  513
##
##  ( 245 268 )
##
##
## Number of Classes:  2
##
## Levels:
##  NO YES
```

```
confusionMatrix(predict (svm_model_under, test) , df_test$Diabetes,
                positive = "YES")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   NO  YES
##        NO  1153   25
##        YES  412  108
##
##                Accuracy : 0.7426
##                  95% CI : (0.7211, 0.7633)
##     No Information Rate : 0.9217
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2354
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.81203
##             Specificity : 0.73674
##          Pos Pred Value : 0.20769
##          Neg Pred Value : 0.97878
##              Prevalence : 0.07833
##          Detection Rate : 0.06360
##    Detection Prevalence : 0.30624
##       Balanced Accuracy : 0.77439
##
##        'Positive' Class : YES
##
```

```
# Sensitivity : 0.92623
# Specificity : 0.67893
# Accuracy : 0.6967

#########################
#Comparing Results
#########################
```