

Chronic Kidney Disease Analysis

Loading the libraries

tidyverse for easy data manipulation and visualization
caret for easy machine learning workflow

```
library(readxl)
library(caTools)
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.3

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

library(tidyverse)

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

## v ggplot2 3.2.1      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.4
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

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

library(caret)

## Warning: package 'caret' was built under R version 3.6.3

## Loading required package: lattice

##
## Attaching package: 'caret'

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

library(openxlsx)
```

Reading the dataset

Dataset description:

The dataset was obtained from Darden Business Publishing - University of Virginia.
Target variable - CKD Predictors - 32 Type of data - Rows and column data Dataset consists of 32 predictors, which consists of 10 continuous variables, and 22 categorical variables.

```
dataset <- read_excel("D:/UIC Spring 2020/Healthcare Analytics/Chronic Kidney Disease/Chronic Kidney Disease Dataset.xls", sheet = "All Data")
```

setting the seed to a number so we train on same data everytime

```
set.seed(12345)
```

description of the data

```
attach(dataset)
str(dataset)

## Classes 'tbl_df', 'tbl' and 'data.frame':   8819 obs. of  34 variables:
## $ ID          : num  1 2 3 4 5 6 7 8 9 10 ...
## $ Age         : num  65 36 66 54 63 26 66 59 53 78 ...
## $ Female      : num  1 1 1 1 1 0 0 1 1 1 ...
## $ Racegrp     : chr   "white" "hispa" "white" "white" ...
## $ Educ        : num  0 0 0 1 0 1 1 1 1 0 ...
## $ Unmarried   : num  0 NA 1 0 0 0 0 1 0 1 ...
## $ Income      : num  1 1 0 0 NA 0 0 0 1 0 ...
## $ CareSource  : chr   "other" "noplac" "noplac" "DrHMO" ...
## $ Insured     : num  1 0 1 1 1 1 0 1 1 1 ...
## $ Weight      : num  56 60.2 83.9 69.4 73.1 ...
## $ Height      : num  162 162 162 160 159 ...
## $ BMI         : num  21.3 22.9 31.8 26.9 28.8 ...
## $ Obese       : num  0 0 1 0 0 1 0 0 0 0 ...
## $ Waist       : num  83.6 76.6 113.2 77.9 89.3 ...
## $ SBP         : num  135 96 115 110 132 129 137 124 110 170 ...
## $ DBP         : num  71 52 57 57 73 70 92 73 74 78 ...
## $ HDL         : num  48 31 44 74 67 43 41 43 62 105 ...
## $ LDL         : num  249 135 211 156 154 159 143 140 110 90 ...
## $ Total Chol  : num  297 166 255 230 221 202 184 183 172 195 ...
## $ Dyslipidemia : num  0 0 1 0 0 0 0 0 0 0 ...
## $ PVD         : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Activity    : num  3 3 1 2 1 2 3 2 1 1 ...
## $ PoorVision  : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Smoker      : num  1 0 1 1 0 0 0 0 1 0 ...
## $ Hypertension : num  0 0 0 0 1 0 1 0 1 1 ...
## $ Fam Hypertension: num  0 0 0 0 0 1 0 1 1 0 ...
## $ Diabetes    : num  0 0 1 0 0 0 0 1 0 1 ...
## $ Fam Diabetes : num  1 0 0 0 0 0 0 1 0 1 ...
## $ Stroke      : num  0 0 0 0 0 0 0 0 0 1 ...
## $ CVD         : num  1 0 0 0 0 0 0 0 0 1 ...
```

```
## $ Fam CVD      : num  0 0 0 0 0 1 1 1 1 1 ...
## $ CHF          : num  0 0 0 0 0 0 0 0 0 NA ...
## $ Anemia       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ CKD          : num  0 0 0 0 0 0 0 0 0 0 ...
```

creating empty category variable

```
cat.ind <- c()
```

factorizing the categorical variables

```
n <- 0
#converting num to factors
for (i in 1:ncol(dataset)){
  dataset[[i]]=as.factor(dataset[[i]])
  if (length(levels(dataset[[i]]))>=5){
    dataset[[i]]=as.numeric(dataset[[i]])
  }
  if (is.factor(dataset[[i]])){
    n <- n + 1
    cat.ind[n] <- i
  }
}
```

CKD has 'NA' which needs to be predicted. Storing those rows separately. 'grep' function gives us the index of a column.

```
test.ind=which(is.na(dataset$CKD))
data.test=dataset[test.ind,]
p=grep("CKD", colnames(data.test))
```

removing the CKD column in the test data

```
data.test_without=data.test[, -p]
```

removing all the rows that have N/A in any of the columns to avoid injecting any artificial imputation in the test dataset

```
for (i in 1:ncol(data.test_without)){
  if (sum(is.na(data.test_without[[i]]))!=0){
    a=which(is.na(data.test_without[[i]]))
    data.test_without=data.test_without[-a,]
  }
}
```

training dataset after removing the N/A rows in the target variable 'CKD'

```
b=grep("CKD", colnames(dataset))
dataset=dataset[-test.ind,]
```

removing all the rows that have any missing values from the training data

```
for (i in 1:ncol(dataset)){
  if (sum(is.na(dataset[[i]]))!=0){
    a=which(is.na(dataset[[i]]))
    dataset=dataset[-a,]
  }
}
```

seperating the training data into categorical variables and continuous variables

```
cat.df <- dataset[,cat.ind]
cat.num<-dataset[, -cat.ind]
```

checking if there are any rows which are NA

```
na_dataset=lapply(dataset,function(x) {length(which(is.na(x)))})
unlist(na_dataset)
```

```
##          ID          Age          Female          Racegrp
##          0           0           0           0
##      Educ      Unmarried      Income      CareSource
##          0           0           0           0
##      Insured      Weight      Height      BMI
##          0           0           0           0
##      Obese      Waist      SBP      DBP
##          0           0           0           0
##      HDL      LDL      Total Chol      Dyslipidemia
##          0           0           0           0
##      PVD      Activity      PoorVision      Smoker
##          0           0           0           0
##      Hypertension Fam Hypertension      Diabetes      Fam Diabetes
##          0           0           0           0
##      Stroke      CVD      Fam CVD      CHF
##          0           0           0           0
##      Anemia      CKD
##          0           0
```

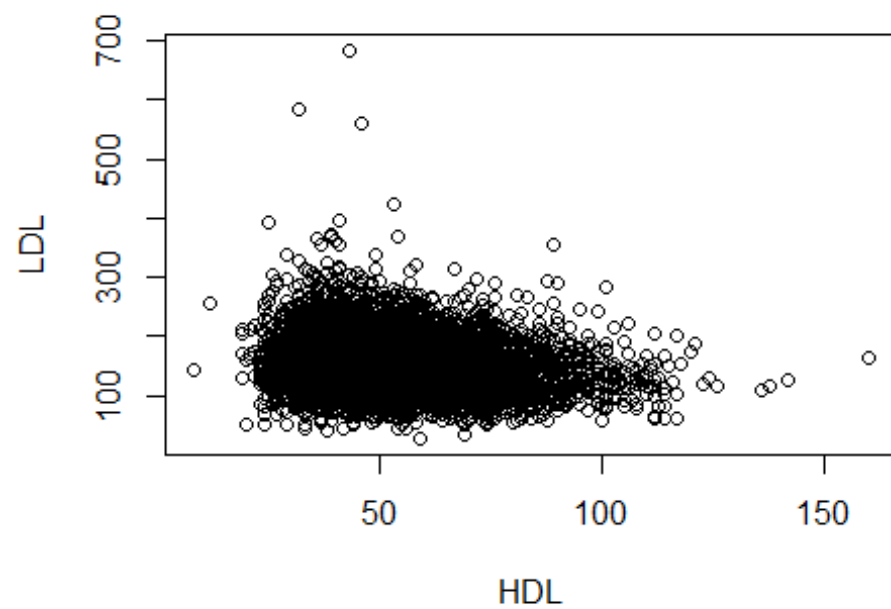
splitting the dataset into train and test

```
data.train=dataset[-test.ind,]
```

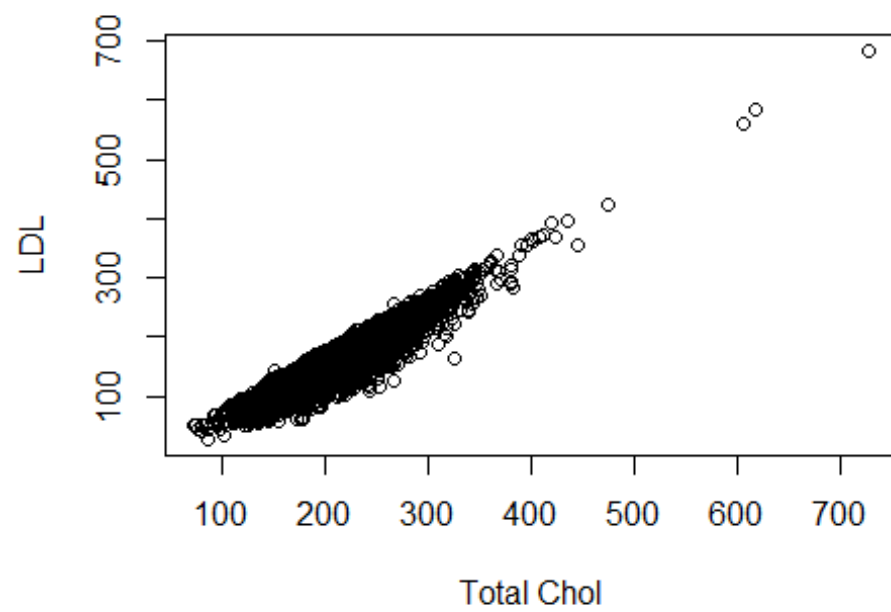
checked for independance of the unmarried variable by chi-square The p value came to be less than 0.1, so we include this variable as it is significant, but since it doesnt make sense in the context removed

By researching further, it is known that having low HDL and high LDL, is bad, and since Total Chol is sum of HDL, LDL, and 20% of total glycerides, I chose to remove the Total chol, because there is high correlation

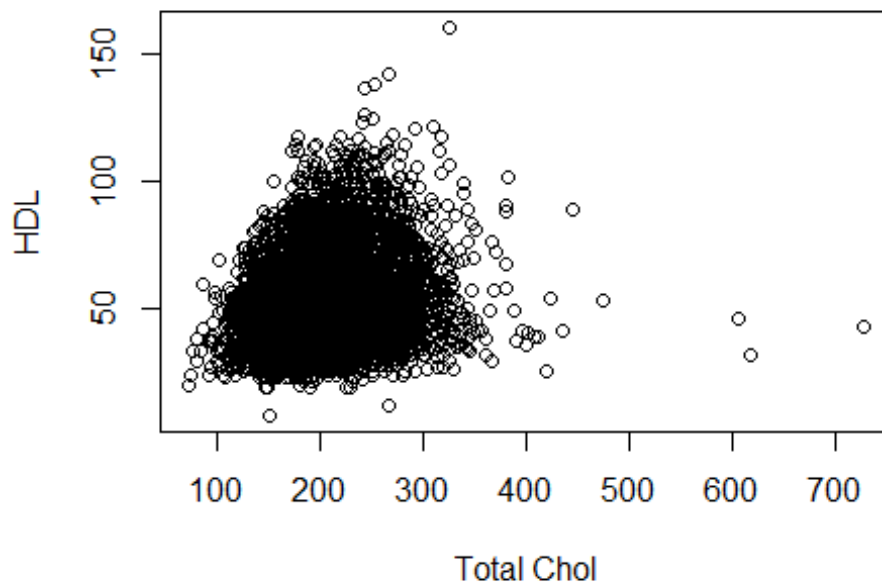
```
plot(HDL, LDL)
```



```
plot(`Total Chol`,LDL)
```



```
plot(`Total Chol`,HDL)
```



```
cor(dataset$LDL,dataset$`Total Chol`)
```

```
## [1] 0.9285436
```

```
cor(dataset$HDL,dataset$`Total Chol`)
```

```
## [1] 0.1711896
```

scaling the data of HDL and Total Chol and plotting gave the same plot

```
dat <- data.frame(HDL,`Total Chol`)
```

```
scaled.dat <- scale(dat)
```

```
colnames(scaled.dat)
```

```
## [1] "HDL" "Total.Chol"
```

Significance Test- Chi-Square test:

#checking the significance of the column "Dyslipidemia"

```
chi.Dyslipidemia=table(data.train$Dyslipidemia,data.train$CKD)
```

```
print(chi.Dyslipidemia)
```

```
##
```

```
##      0      1
```

```
## 0 3468 238
```

```
## 1  401  28
```

```
print(chisq.test(chi.Dyslipidemia))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.Dyslipidemia
## X-squared = 8.7463e-29, df = 1, p-value = 1
```

#checking the significance of the column "PoorVision"

```
chi.Poorvision=table(dataset$PoorVision,dataset$CKD)
print(chi.Poorvision)
```

```
##
##      0      1
## 0 3685  229
## 1  184   37
```

```
print(chisq.test(chi.Poorvision))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.Poorvision
## X-squared = 39.436, df = 1, p-value = 3.39e-10
```

#chi square between Female and CKD

```
chi.female = table(Female,CKD)
print(chi.female)
```

```
##      CKD
## Female  0      1
##      0 2655  210
##      1 2881  254
```

```
print(chisq.test(chi.female))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.female
## X-squared = 1.1452, df = 1, p-value = 0.2846
```

Checking for correlation-

#correlation with the SBP and DBP

```
cor(dataset$SBP,dataset$DBP)
```

```
## [1] 0.3512769
```

#checking correlation between Fam CVD and CVD

```
chi.FamCVD=table(dataset$`Fam CVD`,dataset$CVD)
print(chi.FamCVD)
```

```
##
##      0      1
## 0 2520  144
## 1 1383   88
```

```
print(chisq.test(chi.FamCVD))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.FamCVD
## X-squared = 0.49166, df = 1, p-value = 0.4832
```

#checking correlation between Fam Diabetes and Diabetes

```
chi.Diabetes=table(dataset$Diabetes,dataset$`Fam Diabetes`)
print(chi.Diabetes)
```

```
##
##      0      1
## 0 2683 1021
## 1  157  274
```

```
print(chisq.test(chi.Diabetes))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.Diabetes
## X-squared = 231.05, df = 1, p-value < 2.2e-16
```

#checking correlation between Hypertension and Fam Hypertension

```
chi.Hypertension = table(Hypertension,`Fam Hypertension`)
print(chi.Hypertension)
```

```
##           Fam Hypertension
## Hypertension    0      1
##           0 4104 1127
##           1 2589  919
```

```
print(chisq.test(chi.Hypertension))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.Hypertension
## X-squared = 25.091, df = 1, p-value = 5.469e-07
```

#checking correlation between education and CKD


```

chi.edu=table(dataset$Educ,dataset$CKD)
print(chi.edu)

##
##      0      1
## 0 2008  173
## 1 1861   93

print(chisq.test(chi.edu))

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.edu
## X-squared = 16.713, df = 1, p-value = 4.349e-05

```

#checking correlation between race and CKD

```

chi.race=table(dataset$Racegrp,dataset$CKD)
print(chi.race)

##
##      0      1
## black  637   44
## hispa 1100   30
## other  120    4
## white 2012  188

print(chisq.test(chi.race))

##
##  Pearson's Chi-squared test
##
## data:  chi.race
## X-squared = 45.229, df = 3, p-value = 8.274e-10

```

#checking correlation between caresouce and CKD

```

chi.care=table(dataset$CareSource,dataset$CKD)
print(chi.care)

##
##      0      1
## clinic  809   54
## DrHMO   2268  189
## noplac  613    9
## other   179   14

print(chisq.test(chi.care))

##
##  Pearson's Chi-squared test
##

```

```
## data:  chi.care
## X-squared = 32.425, df = 3, p-value = 4.259e-07
```

#checking correlation between Insured and CKD

```
chi.insurance=table(dataset$Insured,dataset$CKD)
print(chi.insurance)
```

```
##
##      0      1
## 0  776      9
## 1 3093     257
```

```
print(chisq.test(chi.insurance))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.insurance
## X-squared = 43.91, df = 1, p-value = 3.439e-11
```

#checking correlation between PVD and CKD

```
chi.pvd=table(dataset$PVD,dataset$CKD)
print(chi.pvd)
```

```
##
##      0      1
## 0 3766     228
## 1   103      38
```

```
print(chisq.test(chi.pvd))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi.pvd
## X-squared = 98.596, df = 1, p-value < 2.2e-16
```

#checking correlation between activity and CKD

```
chi.activity=table(dataset$Activity,dataset$CKD)
print(chi.activity)
```

```
##
##      0      1
## 1   901     99
## 2  2092    144
## 3   619     21
## 4   257      2
```

```
print(chisq.test(chi.activity))
```

```
##
## Pearson's Chi-squared test
##
## data: chi.activity
## X-squared = 44.321, df = 3, p-value = 1.29e-09

#checking correlation between smoker and CKD

chi.smoker=table(dataset$Smoker,dataset$CKD)
print(chi.smoker)

##
##      0      1
## 0 2713  160
## 1 1156  106

print(chisq.test(chi.smoker))

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: chi.smoker
## X-squared = 11.204, df = 1, p-value = 0.0008163

str(dataset)

## Classes 'tbl_df', 'tbl' and 'data.frame': 4135 obs. of 34 variables:
## $ ID : num 1 3 4 6 7 8 9 13 14 15 ...
## $ Age : num 46 47 35 7 47 40 34 43 17 41 ...
## $ Female : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 2 1 1 2 ...
## $ Racegrp : Factor w/ 4 levels "black","hispa",...: 4 4 4 4 1 4 4
2 2 2 ...
## $ Educ : Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 1 2 1 ...
## $ Unmarried : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 1 1 2 1 ...
## $ Income : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
## $ CareSource : Factor w/ 4 levels "clinic","DrHMO",...: 4 3 2 2 2 2 2
2 1 2 ...
## $ Insured : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 2 2 2 ...
## $ Weight : num 167 526 339 918 539 345 412 169 459 322 ...
## $ Height : num 222 226 206 525 322 210 342 256 439 68 ...
## $ BMI : num 361 1397 920 1416 1098 ...
## $ Obese : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 1 1 1 2 ...
## $ Waist : num 207 503 150 550 363 386 227 158 239 256 ...
## $ SBP : num 58 38 33 52 60 47 33 124 31 76 ...
## $ DBP : num 55 41 41 54 76 57 58 102 46 61 ...
## $ HDL : num 32 28 58 27 25 27 46 39 32 22 ...
## $ LDL : num 205 167 112 115 99 96 66 127 80 97 ...
## $ Total Chol : num 209 167 142 114 96 95 84 138 84 91 ...
## $ Dyslipidemia : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
## $ PVD : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ Activity : Factor w/ 4 levels "1","2","3","4": 3 1 2 2 3 2 1 4 2
```

```

2 ...
## $ PoorVision      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ Smoker          : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 2 1 2 ...
## $ Hypertension     : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 2 1 2 ...
## $ Fam Hypertension: Factor w/ 2 levels "0","1": 1 1 1 2 1 2 2 1 1 1 ...
## $ Diabetes         : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 1 1 1 1 ...
## $ Fam Diabetes     : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 1 2 2 ...
## $ Stroke           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ CVD              : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
## $ Fam CVD          : Factor w/ 2 levels "0","1": 1 1 1 2 2 2 2 1 1 1 ...
## $ CHF              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ Anemia           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ CKD              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

```

t-test for continuous variable

```

t.test(dataset$SBP~dataset$CKD)

##
## Welch Two Sample t-test
##
## data: dataset$SBP by dataset$CKD
## t = -11.117, df = 289.41, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -19.50560 -13.63771
## sample estimates:
## mean in group 0 mean in group 1
## 46.19902 62.77068

t.test(dataset$DBP~dataset$CKD)

##
## Welch Two Sample t-test
##
## data: dataset$DBP by dataset$CKD
## t = 2.6927, df = 294.23, p-value = 0.007493
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 0.6181552 3.9756354
## sample estimates:
## mean in group 0 mean in group 1
## 55.86456 53.56767

t.test(dataset$HDL~dataset$CKD)

##
## Welch Two Sample t-test
##
## data: dataset$HDL by dataset$CKD
## t = 3.0937, df = 308.67, p-value = 0.002158

```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.045351 4.698911
## sample estimates:
## mean in group 0 mean in group 1
##      35.89093      33.01880

t.test(dataset$LDL~dataset$CKD)

##
##  Welch Two Sample t-test
##
## data:  dataset$LDL by dataset$CKD
## t = -2.3321, df = 296.13, p-value = 0.02037
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -12.244989 -1.036745
## sample estimates:
## mean in group 0 mean in group 1
##      107.6937      114.3346
```

removing the columns that were not found significant with the target variable

```
grep("Income",colnames(dataset))
## [1] 7

grep("ID",colnames(dataset))
## [1] 1

grep("Unmarried",colnames(dataset))
## [1] 6

grep("Weight",colnames(dataset))
## [1] 10

grep("Height",colnames(dataset))
## [1] 11

grep("BMI",colnames(dataset))
## [1] 12

grep("Total Chol",colnames(dataset))
## [1] 19

grep("Dyslipidemia",colnames(dataset))
## [1] 20
```

```

grep("PoorVision", colnames(dataset))
## [1] 23

grep("Waist", colnames(dataset))
## [1] 14

grep("Educ", colnames(dataset))
## [1] 5

grep("CareSource", colnames(dataset))
## [1] 8

#col index of the dataset which are not needed
col=c(7,1,5,6,8,10,11,12,14,19,20,23)

dataset=dataset[,-col]

```

since we want to check for the recall, I split the data into train and test after removing the N/A's from the target variable 'CKD'

```

require(caTools)
set.seed(123)

sample <- sample.split(dataset, SplitRatio = 0.75)
data.train1 <- subset(dataset, sample ==TRUE)

## Warning: Length of logical index must be 1 or 4135, not 22

data.test1 <- subset(dataset, sample ==FALSE)

## Warning: Length of logical index must be 1 or 4135, not 22

```

keeping the variables that are significant and after removing multi-collinearity.

```

keeps <- c("Age", "Female",
"Racegrp", "HDL", "LDL", "PVD", "Hypertension", "Diabetes", "CHF", "CKD")
data.train1 <- data.train1[keeps]
data.test1 <- data.test1[keeps]

```

Building the Logistic Regression model

```
log.model = glm( data.train1$CKD ~ ., family = binomial, data = data.train1)
```

summary of the logistic regression model

```

summary(log.model)

##
## Call:
## glm(formula = data.train1$CKD ~ ., family = binomial, data = data.train1)

```

```
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7074  -0.2743  -0.1147  -0.0599   3.4497
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.535e+00  5.552e-01 -11.772  < 2e-16 ***
## Age          8.771e-02  7.707e-03  11.381  < 2e-16 ***
## Female1      3.425e-01  1.921e-01   1.783  0.074572 .
## Racegrphispa -9.731e-01  3.100e-01  -3.139  0.001697 **
## Racegrpothor -1.498e+01  6.057e+02  -0.025  0.980268
## Racegrpwhite -8.686e-02  2.456e-01  -0.354  0.723540
## HDL          -1.749e-02  6.466e-03  -2.705  0.006839 **
## LDL          5.917e-04  2.214e-03   0.267  0.789256
## PVD1         5.764e-01  2.674e-01   2.156  0.031112 *
## Hypertension1 7.513e-01  2.180e-01   3.446  0.000569 ***
## Diabetes1     8.739e-01  2.041e-01   4.282  1.86e-05 ***
## CHF1         5.463e-01  3.306e-01   1.653  0.098401 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1362.67  on 3006  degrees of freedom
## Residual deviance:  919.79  on 2995  degrees of freedom
## AIC: 943.79
##
## Number of Fisher Scoring iterations: 17
```

For the better interpretation, we are finding the odds ratio below

```
exp(coefficients(log.model))

##      (Intercept)           Age           Female1 Racegrphispa Racegrpothor
## 1.451095e-03 1.091673e+00 1.408418e+00 3.778971e-01 3.117390e-07
## Racegrpwhite           HDL           LDL           PVD1 Hypertension1
## 9.168053e-01 9.826643e-01 1.000592e+00 1.779627e+00 2.119719e+00
## Diabetes1           CHF1
## 2.396216e+00 1.726850e+00
```

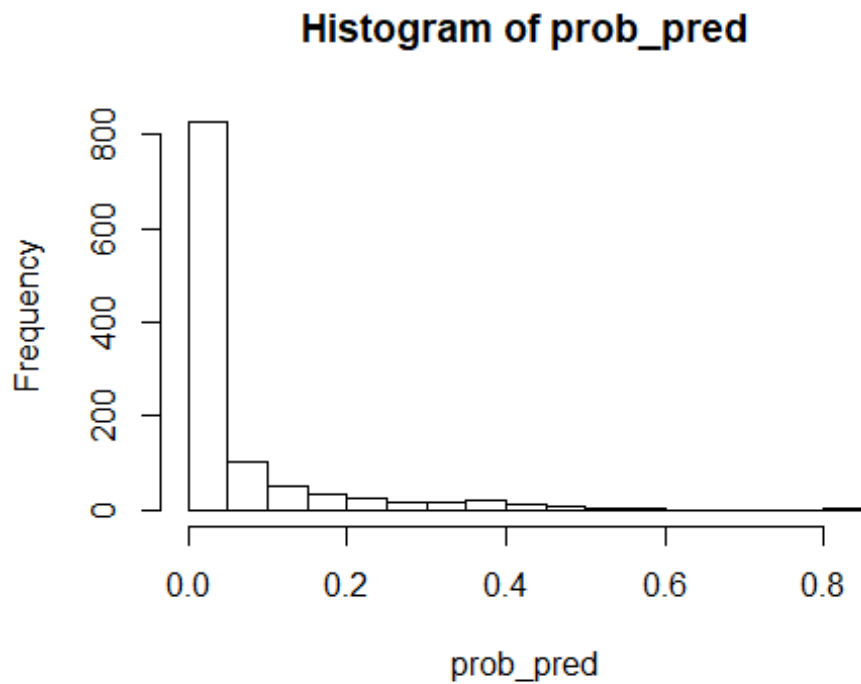
predicting the results on the test data. Using a threshold value to classify the target variable if a person has chronic kidney disease or not so we can have less false negatives in the prediction. I focused on getting better recall for this reason

```
# Predicting the Test set results
prob_pred = predict(log.model, type = 'response', newdata = data.test1)
y_pred = ifelse(prob_pred > 0.01, 1, 0)
```

Making the Confusion Matrix to get the recall value for the prediction #Recall = tp/(tp+fn)

```
cm = table(data.test1$CKD, y_pred > 0.01)
cm

##
##      FALSE TRUE
##    0    562  480
##    1      5   81
hist(prob_pred)
```



```
table(data.test1$CKD)

##
##    0    1
## 1042   86

cm[3]

## [1] 480

Recall = cm[4]/(cm[4]+cm[2])
print(Recall)

## [1] 0.9418605
```

Recall-Precision curve

```
#RP.perf <- performance(pred, "prec", "rec")
```


Saving the RDS file to load in RShiny for the screening tool

```
saveRDS(log.model, file = "D:/UIC Spring 2020/Healthcare Analytics/Chronic  
Kidney Disease/CKD_v2.rds")  
write.xlsx(data.test1, 'D:/UIC Spring 2020/Healthcare Analytics/Chronic  
Kidney Disease/testdata_CKD_v2.xlsx')  
data.frame(min=sapply(cat.num,min),max=sapply(cat.num,max))
```

```
##           min  max  
## ID           1 6000  
## Age           1   66  
## Weight        3 1121  
## Height        1   553  
## BMI           2 2403  
## Waist         2   800  
## SBP           2  138  
## DBP           1  105  
## HDL           1  111  
## LDL           3  277  
## Total Chol    2  282
```