Election Data Preparation and Prediction

-- Chakradhar Varma

```
#set up of working directory
setwd("D:/BACP Program/Module 5/Mini Project")
getwd()

## [1] "D:/BACP Program/Module 5/Mini Project"

library("xlsx")

library("tidyverse")

library("data.table")

library("dplyr")

#Importing the data

EL_data <- read.xlsx(file.choose(),1)
#View(EL_data)

#Creating Data frame of Uttarkhand Data from EC Data
EL_data_uk <- EL_data %>% select(everything()) %>% filter(ST_NAME == "Uttarak
hand") %>% droplevels()
#View(EL_data_uk)

#removing NOTA category from the data
EL_data_uk <- filter(EL_data_uk, CAND_NAME != "None of the Above")
dim(EL_data_uk)

## [1] 637  15

str(EL_data_uk)

## 'data.frame':    637 obs. of  15 variables:
##  $ ST_CODE              : Factor w/ 1 level "S28": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ST_NAME              : Factor w/ 1 level "Uttarakhand": 1 1 1 1 1 1 1 1
1 1 ...
##  $ MONTH                : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ YEAR                 : num  2017 2017 2017 2017 2017 ...
##  $ DIST_NAME            : Factor w/ 13 levels "Almora","Bageshwar",..: 13
13 13 13 13 13 13 13 13 13 ...
##  $ AC_NO                : num  1 1 1 1 2 2 2 2 2 2 ...
##  $ AC_NAME              : Factor w/ 70 levels "Almora","B.H.E.L. ranipur",
..: 52 52 52 52 70 70 70 70 70 70 ...
```

```
##  $ AC_TYPE              : Factor w/ 3 levels "GEN","SC","ST": 2 2 2 2 1 1
1 1 1 1 ...
##  $ CAND_NAME            : Factor w/ 625 levels "(BRIG) GOVIND PRASAD BARTH
WAL",..: 415 280 129 444 226 473 366 447 399 198 ...
##  $ CAND_SEX             : Factor w/ 4 levels "F","M","NULL",..: 2 2 2 2 2
2 2 2 2 2 ...
##  $ CAND_CATEGORY        : Factor w/ 4 levels "GEN","NULL","SC",..: 3 3 3 3
1 1 1 1 3 3 ...
##  $ CAND_AGE             : Factor w/ 52 levels "25","26","27",..: 27 26 8 3
1 33 20 22 30 31 8 ...
##  $ PARTYABBRE           : Factor w/ 36 levels "AIFB","AVIRP",..: 17 5 18 1
0 5 17 18 18 10 30 ...
##  $ TOTALVALIDVOTESPOLLED: num  17798 16785 13508 679 19800 ...
##  $ POSITION             : num  1 2 3 5 1 2 3 4 5 7 ...
```

```
## Data Preparation of MyNetha data
#code to scrap data from MyNeta
library(rvest)

# helper to clean column names
mcga <- function(x) { make.unique(gsub("(^_|_$)", "", gsub("_+", "_",  gsub("
[[:punct:][:space:]]+", "_", tolower(x)))), sep = "_") }

pg <- read_html("http://www.myneta.info/uttarakhand2017/index.php?action=summ
ary&subAction=candidates_analyzed&sort=candidate#summary")

# target the table
tab <- html_node(pg, xpath=".//table[contains(thead, 'Liabilities')]")

# get the rows so we can target columns
rows <- html_nodes(tab, xpath=".//tr[td[not(@colspan)]]")

# make a data frame
do.call(
  cbind.data.frame,
  c(lapply(1:8, function(i) {
    html_text(html_nodes(rows, xpath=sprintf(".//td[%s]", i)), trim=TRUE)
  }), list(stringsAsFactors=FALSE))
) -> MN_data_uk

MN_data_uk <- setNames(MN_data_uk, mcga(html_text(html_nodes(tab, "th")))) #
get the header to get column names

#View(MN_data_uk)
dim(MN_data_uk)  # No of candidates is matching with EC data
```

```
## [1] 637   8
```

```
str(MN_data_uk)
```

```
## 'data.frame':    637 obs. of  8 variables:
##  $ sno             : chr  "1" "2" "3" "4" ...
##  $ candidate<U+2207>: chr  "(Dr) Dinesh" "(Dr) Harak Singh Rawat" "A.Hamee
d" "Aan Singh" ...
##  $ constituency    : chr  "KOTDWAR" "KOTDWAR" "DOIWALA" "BHIMTAL" ...
##  $ party           : chr  "UTTARAKHAND PARIVARTAN PARTY" "BJP" "BSP" "IND
" ...
##  $ criminal_case   : chr  "0" "2" "0" "0" ...
##  $ education       : chr  "Doctorate" "Doctorate" "12th Pass" "12th Pass"
...
##  $ total_assets    : chr  "Rs 52,92,000 ~ 52 Lacs+" "Rs 2,68,95,976 ~ 2 C
rore+" "Rs 2,00,74,378 ~ 2 Crore+" "Rs 8,12,936 ~ 8 Lacs+" ...
##  $ liabilities     : chr  "Rs 8,70,000 ~ 8 Lacs+" "Rs 0 ~" "Rs 14,80,000
~ 14 Lacs+" "Rs 0 ~" ...
```

```
#combining with previous winners data to see the incumbency
mcga <- function(x) { make.unique(gsub("(^_|_$)", "", gsub("_+", "_",  gsub("
[[:punct:][:space:]]+", "_", tolower(x)))), sep = "_") }

pg <- read_html("http://myneta.info/utt2012/index.php?action=summary&subActio
n=winner_analyzed&sort=candidate#summary")

# target the table
tab <- html_node(pg, xpath=".//table[contains(thead, 'Liabilities')]")

# get the rows so we can target columns
rows <- html_nodes(tab, xpath=".//tr[td[not(@colspan)]]")

# make a data frame
do.call(
  cbind.data.frame,
  c(lapply(1:8, function(i) {
    html_text(html_nodes(rows, xpath=sprintf(".//td[%s]", i)), trim=TRUE)
  }), list(stringsAsFactors=FALSE))
) -> MN_data_uk_12

MN_data_uk_12 <- setNames(MN_data_uk_12, mcga(html_text(html_nodes(tab, "th")
))) # get the header to get column names

#View(MN_data_uk_12)
dim(MN_data_uk_12)  # No of candidates is matching with EC data
```

```
## [1] 67  8
```

```
str(MN_data_uk_12)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ sno             : chr  "1" "2" "3" "4" ...
##  $ candidate<U+2207>: chr  "Adesh Chauhan" "Ajay Bhatt" "Ajay Tamta" "Amri
ta Rawat" ...
```

```
##  $ constituency     : chr  "BHEL RANIPUR" "RANIKHET" "SOMESHWAR (SC)" "RAM
NAGAR" ...
##  $ party            : chr  "BJP" "BJP" "BJP" "INC" ...
##  $ criminal_case    : chr  "0" "0" "0" "0" ...
##  $ education        : chr  "12th Pass" "Graduate Professional" "12th Pass"
"Graduate" ...
##  $ total_assets     : chr  "Rs 77,02,745 ~ 77 Lacs+" "Rs 31,37,910 ~ 31 La
cs+" "Rs 31,27,175 ~ 31 Lacs+" "Rs 13,57,86,327 ~ 13 Crore+" ...
##  $ liabilities      : chr  "Rs 95,180 ~ 95 Thou+" "Rs 5,00,000 ~ 5 Lacs+"
"Rs 7,45,253 ~ 7 Lacs+" "Rs 0 ~" ...

MN_data_uk_12$INCUMBENCY_FACTOR <- 1

names(MN_data_uk)[2]="candidate"
names(MN_data_uk_12)[2]="candidate"

#removing unnecessary columns
MN_data_uk_12_new <- MN_data_uk_12[c(-1, -3:-8)]

MN_data_uk_12_new$candidate <- str_replace(MN_data_uk_12_new$candidate, "\\("
, "")
MN_data_uk_12_new$candidate <- str_replace(MN_data_uk_12_new$candidate, "\\)"
, "")

#merging both the 2017 and 2012 winners data
MN_data_uk_comb <- left_join(MN_data_uk, MN_data_uk_12_new, by=c("candidate")
, all=TRUE)

#View(MN_data_uk_comb)
str(MN_data_uk_comb)

## 'data.frame':    637 obs. of  9 variables:
##  $ sno              : chr  "1" "2" "3" "4" ...
##  $ candidate        : chr  "(Dr) Dinesh" "(Dr) Harak Singh Rawat" "A.Hamee
d" "Aan Singh" ...
##  $ constituency     : chr  "KOTDWAR" "KOTDWAR" "DOIWALA" "BHIMTAL" ...
##  $ party            : chr  "UTTARAKHAND PARIVARTAN PARTY" "BJP" "BSP" "IND
" ...
##  $ criminal_case    : chr  "0" "2" "0" "0" ...
##  $ education        : chr  "Doctorate" "Doctorate" "12th Pass" "12th Pass"
...
##  $ total_assets     : chr  "Rs 52,92,000 ~ 52 Lacs+" "Rs 2,68,95,976 ~ 2 C
rore+" "Rs 2,00,74,378 ~ 2 Crore+" "Rs 8,12,936 ~ 8 Lacs+" ...
##  $ liabilities      : chr  "Rs 8,70,000 ~ 8 Lacs+" "Rs 0 ~" "Rs 14,80,000
~ 14 Lacs+" "Rs 0 ~" ...
##  $ INCUMBENCY_FACTOR: num  NA NA NA NA NA NA 1 NA NA NA ...

#replace NA value with 0 in the final data
MN_data_uk_comb$INCUMBENCY_FACTOR[is.na(MN_data_uk_comb$INCUMBENCY_FACTOR)] <
- 0
```

```
table(MN_data_uk_comb$INCUMBENCY_FACTOR)

##
##   0   1
## 598  39

#cleaning MyNeta data and to make it compatiable with EC Data

#removing s.no column and liabilities
MN_data_uk_comb <- MN_data_uk_comb[c(-1,-8)]

#Renaming the columns
names(MN_data_uk_comb)[1] <- "CAND_NAME"
names(MN_data_uk_comb)[2] <- "CONS_NAME"
names(MN_data_uk_comb)[3] <- "PARTYABBRE"
names(MN_data_uk_comb)[4] <- "CRIMINAL_CASE"
names(MN_data_uk_comb)[5] <- "EDUCATION"
names(MN_data_uk_comb)[6] <- "TOTAL_ASSESTS"


#merging with candidates data who filed ITR

# helper to clean column names
mcga <- function(x) { make.unique(gsub("(^_|_$)", "", gsub("_+", "_",  gsub("
[[:punct:][:space:]]+", "_", tolower(x)))), sep = "_") }

pg <- read_html("http://www.myneta.info/uttarakhand2017/index.php?action=summ
ary&subAction=filed_itr&sort=candidate#summary")

# target the table
tab <- html_node(pg, xpath=".//table[contains(thead, 'Liabilities')]")

# get the rows so we can target columns
rows <- html_nodes(tab, xpath=".//tr[td[not(@colspan)]]")

# make a data frame
do.call(
  cbind.data.frame,
  c(lapply(1:8, function(i) {
    html_text(html_nodes(rows, xpath=sprintf(".//td[%s]", i)), trim=TRUE)
  }), list(stringsAsFactors=FALSE))
) -> MN_data_ITR_uk

MN_data_ITR_uk <- setNames(MN_data_ITR_uk, mcga(html_text(html_nodes(tab, "th
")))) # get the header to get column names

MN_data_ITR_uk$ITR_FILED <- 1
```

```
dim(MN_data_ITR_uk)
```

```
## [1] 358   9
```

```
#View(MN_data_ITR_uk)

#remove unused columns
MN_data_ITR_uk <- MN_data_ITR_uk[c(-1,-5:-8)]

names(MN_data_ITR_uk)[1] <- "CAND_NAME"
names(MN_data_ITR_uk)[2] <- "CONS_NAME"
names(MN_data_ITR_uk)[3] <- "PARTYABBRE"

#MN_data_ITR_uk$CAND_NAME <- as.factor(toupper(MN_data_ITR_uk$CAND_NAME))

MN_data_uk_final <- merge(MN_data_uk_comb, MN_data_ITR_uk, by= c("CAND_NAME",
"CONS_NAME", "PARTYABBRE"), all=TRUE)

dim(MN_data_uk_final)
```

```
## [1] 637   8
```

```
MN_data_uk_final$ITR_FILED[is.na(MN_data_uk_final$ITR_FILED)] <- 0

#View(MN_data_uk_final)

#checking candidate column
MN_data_uk_final$CAND_NAME <- as.factor(toupper(MN_data_uk_final$CAND_NAME))

#checking party column
#to see if EC data and MyNeta data is matching
#table(MN_data_uk_final$PARTYABBRE)

table(EL_data_uk$PARTYABBRE)
```

```
## 
##       AIFB      AVIRP     BaSaPa       BASD        BJP       bkdl 
##          1          1          4          1         70          1 
##      BKLJP        BMF       BMUP        BSP       BSRD        CPI 
##          1          1          3         69          2          4 
## CPI(ML)(L)        CPM   HAMJANPA      IBusP        INC        IND 
##          2          6          3          3         70        261 
##       LSPS        NCP       NOTA       PECP     PEPART    PRAJAPA 
##          1          2          0          3          1          3 
##       RaAP       raup       RJSD        RLD       SaSP     SAVIPA 
##          3          1          1          6          4          6 
##        SHS         SP       UKDD       UKKD        UPP       UtRM 
##          9         20          9         54          9          2 
```

```
#correcting the wrong data
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$CAND_NAME == "SHASHTRI PAW
AN MALETHA")] = "Uttarakhand Kranti Dal (Democratic)"

#Changing the names of party to ABBREVATIONS as in EC Data
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "AARAKSHAN V
IRODHI PARTY")] = "AVIRP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bahujan Muk
ti Party")] = "BMUP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bahujan San
gharshh Dal")] = "BASD"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bhartiya Sa
rvodaya Party")] = "BaSaPa"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bharat Kaum
i Dal")] = "bkdl"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bharat Ki L
ok Jimmedar Party")] = "BKLJP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bharatiya M
omin Front")] = "BMF"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Bharatiya S
ubhash Sena")] = "BSRD"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "CPI(M)")] =
"CPM"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Hamari Janm
anch Party")] = "HAMJANPA"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Indian Busi
ness Party")] = "IBusP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Lok Shahi P
arty (Secular)")] = "LSPS"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Peace Party
")] = "PECP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "People's Pa
rty")] = "PEPART"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Prajamandal
Party")] = "PRAJAPA"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Rashtriya A
darsh Party")] = "RaAP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Rashtriya J
an Sahay Dal")] = "RJSD"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Rashtriya U
ttarakhand Party")] = "raup"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Sainik Sama
j Party")] = "SaSP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Sarv Vikas
Party")] = "SAVIPA"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "UKD")] = "U
KKD"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Uttarakhand
Kranti Dal (Democratic)")] = "UKDD"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "UTTARAKHAND
```

```r
PARIVARTAN PARTY")] = "UPP"
MN_data_uk_final$PARTYABBRE[which(MN_data_uk_final$PARTYABBRE == "Uttarakhand
Raksha Morcha")] = "UtRM"

#converting character class to appropriate class in MN_data_uk data set

#cleaning Total Assets Column
MN_data_uk_final$TOTAL_ASSESTS<-as.character(MN_data_uk_final$TOTAL_ASSESTS)
for(i in 1:nrow(MN_data_uk_final)){
  if(MN_data_uk_final$TOTAL_ASSESTS[i]=="Nil"){
    MN_data_uk_final$TOTAL_ASSESTS[i]=0
  }
  else{
    str<-MN_data_uk_final$TOTAL_ASSESTS[i]
    cleanstr<-gsub(",","",substr(str,start=4,stop=regexpr("~",str)-2))
    MN_data_uk_final$TOTAL_ASSESTS[i]<-cleanstr
  }
}
#Assets value is converted to numeric
MN_data_uk_final$TOTAL_ASSESTS<-as.numeric(MN_data_uk_final$TOTAL_ASSESTS)
sum(is.na(MN_data_uk_final$TOTAL_ASSESTS))
```

```
## [1] 0
```

```r
#No NA values

#checking education column
MN_data_uk_final$EDUCATION <- as.factor(MN_data_uk_final$EDUCATION)
str(MN_data_uk_final)
```

```
## 'data.frame':    637 obs. of  8 variables:
##  $ CAND_NAME        : Factor w/ 617 levels "(DR) DINESH",..: 1 2 3 4 5 6 7
8 8 9 ...
##  $ CONS_NAME        : chr  "KOTDWAR" "KOTDWAR" "DOIWALA" "BHIMTAL" ...
##  $ PARTYABBRE       : chr  "UPP" "BJP" "BSP" "IND" ...
##  $ CRIMINAL_CASE    : chr  "0" "2" "0" "0" ...
##  $ EDUCATION        : Factor w/ 11 levels "10th Pass","12th Pass",..: 5 5
2 2 3 9 2 1 6 11 ...
##  $ TOTAL_ASSESTS    : num  5292000 26895976 20074378 812936 14844166 ...
##  $ INCUMBENCY_FACTOR: num  0 0 0 0 0 0 1 0 0 0 ...
##  $ ITR_FILED        : num  0 1 1 0 1 0 1 0 1 1 ...
```

```r
unique(MN_data_uk_final$EDUCATION)
```

```
##  [1] Doctorate             12th Pass             5th Pass
##  [4] Literate              10th Pass             Graduate
##  [7] Post Graduate         Graduate Professional 8th Pass
## [10] Others                Illiterate
## 11 Levels: 10th Pass 12th Pass 5th Pass 8th Pass Doctorate ... Post Gradua
te
```

```r
#table(MN_data_uk_final$EDUCATION)

#converting criminal_case column to numeric
MN_data_uk_final$CRIMINAL_CASE <- as.numeric(MN_data_uk_final$CRIMINAL_CASE)


#Merging MyNeta data with EC data
library(stringr)

#converting EC data AC_NAME to upper case
EL_data_uk$AC_NAME <- toupper(EL_data_uk$AC_NAME)
#str(EL_data_uk)

#removing additional characters in the brackets to match with MyNeta data's c
andiate name column
EL_data_uk$CAND_NAME <- str_replace(EL_data_uk$CAND_NAME, " \\(.*\\)", "")
MN_data_uk_final$CAND_NAME <- str_replace(MN_data_uk_final$CAND_NAME, " \\(.*
\\)", "")

EL_data_uk_final <- merge(EL_data_uk, MN_data_uk_final, by= c("CAND_NAME", "P
ARTYABBRE"), all = TRUE)

EL_data_uk_final <- filter(EL_data_uk_final, AC_NAME != 'NA')

#View(EL_data_uk_final)


##EDA on the final Data

EL_Model_data <- EL_data_uk_final
dim(EL_Model_data)
```

```
## [1] 642  21
```

```r
#str(EL_Model_data)

attach(EL_Model_data)
#checking individual columns
#CAND_NAME is character we can remove it
EL_Model_data$CAND_NAME <- NULL

#ST_CODE & ST_NAME are statis values and can be removed
EL_Model_data$ST_CODE <- NULL
EL_Model_data$ST_NAME <- NULL

#Removing other redundant columns
EL_Model_data$MONTH <- NULL
EL_Model_data$YEAR <- NULL
EL_Model_data$DIST_NAME <- NULL
EL_Model_data$AC_NO <- NULL
```

```
EL_Model_data$AC_NAME <- NULL
EL_Model_data$CONS_NAME <- NULL
#candidates are mostly belong to the similar category as that of the constiue
ncy type.
#considering only candidate category
EL_Model_data$AC_TYPE <- NULL

#POSITION
EL_Model_data$WINNER <- ifelse(EL_Model_data$POSITION=="1",1,0)
table(EL_Model_data$WINNER)
```

```
##
##   0   1
## 572  70
```

```
#CAND_SEX
table(CAND_SEX)
```

```
## CAND_SEX
##    F    M NULL    O
##   62  578    0    2
```

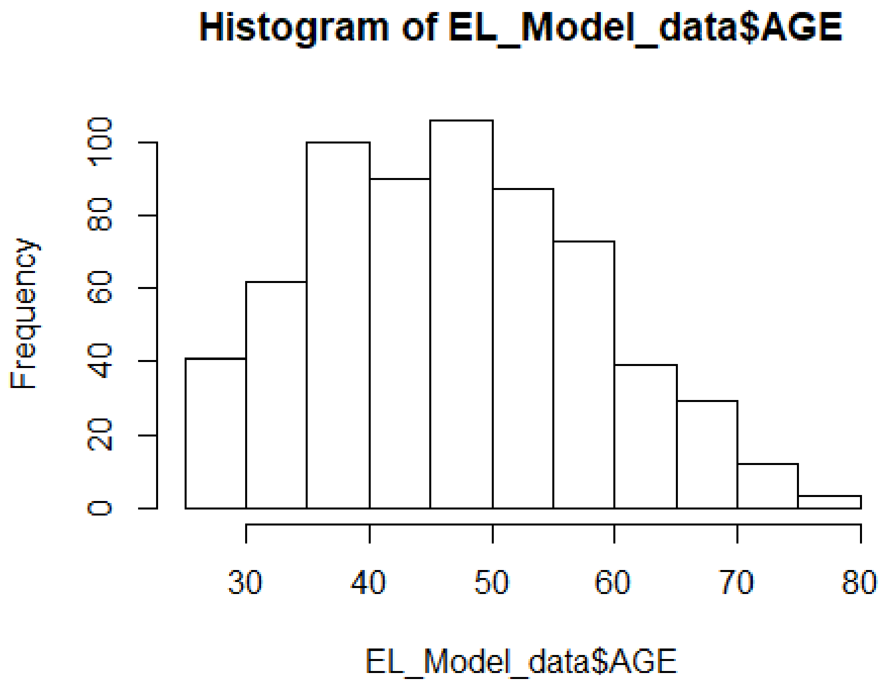
```
#90% are the Male candidates
EL_Model_data$GENDER_MALE<-ifelse(EL_Model_data$CAND_SEX=="M",1,0)
EL_Model_data$GENDER_FEMALE<-ifelse(EL_Model_data$CAND_SEX=="F",1,0)

#CAND_CATEGORY
table(CAND_CATEGORY)
```

```
## CAND_CATEGORY
##  GEN NULL   SC   ST
##  498    0  129   15
```

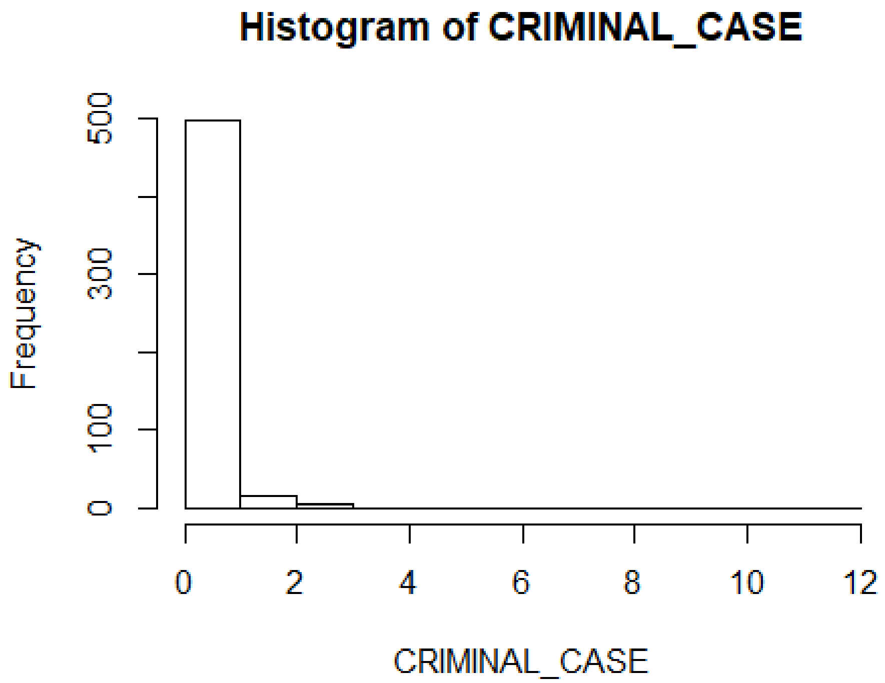
```
#77% of the candidates belong to Genearal category
EL_Model_data$CATEGORY_GEN<-ifelse(EL_Model_data$CAND_CATEGORY=="GEN",1,0)

#CAND_AGE
EL_Model_data$AGE <- as.numeric(as.character(EL_Model_data$CAND_AGE))
summary(EL_Model_data$AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   39.00   47.00   47.18   55.00   77.00
```

```
hist(EL_Model_data$AGE)
```



```
#Education
#table(EL_Model_data$EDUCATION)

ED <- EL_Model_data$EDUCATION
EL_Model_data$GRADUATE <- ifelse(ED=="Graduate"|ED=="Doctorate"|ED=="Post Gra
duate"|ED=="Graduate Professional",1,0)

#TOTAL_ASSESTS
summary(EL_Model_data$TOTAL_ASSESTS)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's 
##       500    709750   3892130  15986085  12724890 802555607       119

EL_Model_data$IS_RICH<-ifelse(EL_Model_data$TOTAL_ASSESTS>10000000,1,0)

#CRIMINAL_CASE
summary(EL_Model_data$CRIMINAL_CASE)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0000  0.0000  0.0000  0.2447  0.0000 12.0000     119

hist(CRIMINAL_CASE)
```



```
#party
#table(EL_Model_data$PARTYABBRE)

# Following are considered as national party
# INC, BJP, BSP, SP, NCP, CPI,CPM

NP <- EL_Model_data$PARTYABBRE

EL_Model_data$IS_NATIONALPARTY <- ifelse(NP=="INC" | NP =="BJP" | NP=="BSP" |
NP=="SP" | NP=="NCP" | NP=="CPI" | NP=="CPM",1,0)

#removing the columns
EL_Model_data <- EL_Model_data[c(-1:-6, -8, -9)]

EL_samp <- EL_Model_data

EL_Model_data$WINNER <- as.factor(EL_Model_data$WINNER)

attach(EL_Model_data)

## The following objects are masked from EL_Model_data (pos = 3):
##
##     CRIMINAL_CASE, INCUMBENCY_FACTOR, ITR_FILED

library(Boruta)

## Warning: package 'Boruta' was built under R version 3.5.3
## Loading required package: ranger

## Warning: package 'ranger' was built under R version 3.5.3

#Feature Selection (Wrapper Method)
set.seed(123)
boruta.train <- Boruta(WINNER~. ,data=EL_Model_data, doTrace = 2)

print(boruta.train)

library(rattle)

library(ROCR)

library(ineq)

library(car)

library(caret)

library(class)
library(rpart)
library(SDMTools)

library(pROC)

library(Hmisc)

library(psych)

library(devtools)

library(e1071)

library(klaR)

library(MASS)
library(plyr)

library(psych)
library(ElemStatLearn)

library(rpart)
library(rpart.plot)

library(nnet)
library(stats)
library(randomForest)

#Generating n random numbers b/w 0 and 1
EL_Model_data$random <- runif(nrow(EL_Model_data),0,1)
#Adding these randomly generated numbers to the data as a new column
EL_Model_data <- EL_Model_data[order(EL_Model_data$random),]
#Splitting the data into dev and testing sample based on the random number
```

```
EL_Model_data.train <- EL_Model_data[which(EL_Model_data$random <= 0.7),]
EL_Model_data.val <- EL_Model_data[which(EL_Model_data$random > 0.7),]

dim(EL_Model_data.train)

## [1] 453  12

dim(EL_Model_data.val)

## [1] 189  12

#Considering all variables intially
logit.eq <- WINNER ~ INCUMBENCY_FACTOR+CRIMINAL_CASE+ITR_FILED+GRADUATE+IS_RI
CH+IS_NATIONALPARTY+CATEGORY_GEN+AGE+GENDER_MALE+GENDER_FEMALE

model.LR.all <- glm(logit.eq, EL_Model_data.train, family = binomial)
summary(model.LR.all)

## 
## Call:
## glm(formula = logit.eq, family = binomial, data = EL_Model_data.train)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.82999  -0.24526  -0.07533  -0.03722   2.97773  
## 
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)    
## (Intercept)        -18.77563 1455.39813  -0.013 0.989707    
## INCUMBENCY_FACTOR    0.42018    0.55184   0.761 0.446410    
## CRIMINAL_CASE        0.51425    0.28283   1.818 0.069030 .  
## ITR_FILED            1.34960    0.69801   1.933 0.053175 .  
## GRADUATE             1.16729    0.48104   2.427 0.015241 *  
## IS_RICH              0.69543    0.50244   1.384 0.166329    
## IS_NATIONALPARTY     3.90306    1.04442   3.737 0.000186 ***
## CATEGORY_GEN        -0.06065    0.58128  -0.104 0.916898    
## AGE                  0.01481    0.02169   0.683 0.494680    
## GENDER_MALE         10.74678 1455.39797   0.007 0.994108    
## GENDER_FEMALE        9.75979 1455.39827   0.007 0.994649    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 255.53  on 360  degrees of freedom
## Residual deviance: 147.83  on 350  degrees of freedom
##   (92 observations deleted due to missingness)
## AIC: 169.83
## 
## Number of Fisher Scoring iterations: 14
```

```r
vif(model.LR.all)
```

```
## INCUMBENCY_FACTOR     CRIMINAL_CASE         ITR_FILED          GRADUATE
##      1.087710e+00      1.065461e+00      1.158616e+00      1.115436e+00
##           IS_RICH  IS_NATIONALPARTY      CATEGORY_GEN               AGE
##      1.302583e+00      1.031393e+00      1.121572e+00      1.174438e+00
##       GENDER_MALE     GENDER_FEMALE
##      3.045963e+06      3.045964e+06
```

```r
#removing insignificant variables
logit.eq <- WINNER ~ INCUMBENCY_FACTOR+GRADUATE+IS_NATIONALPARTY+IS_RICH

model.LR.imp <- glm(logit.eq, EL_Model_data.train, family = binomial)
summary(model.LR.imp)
```

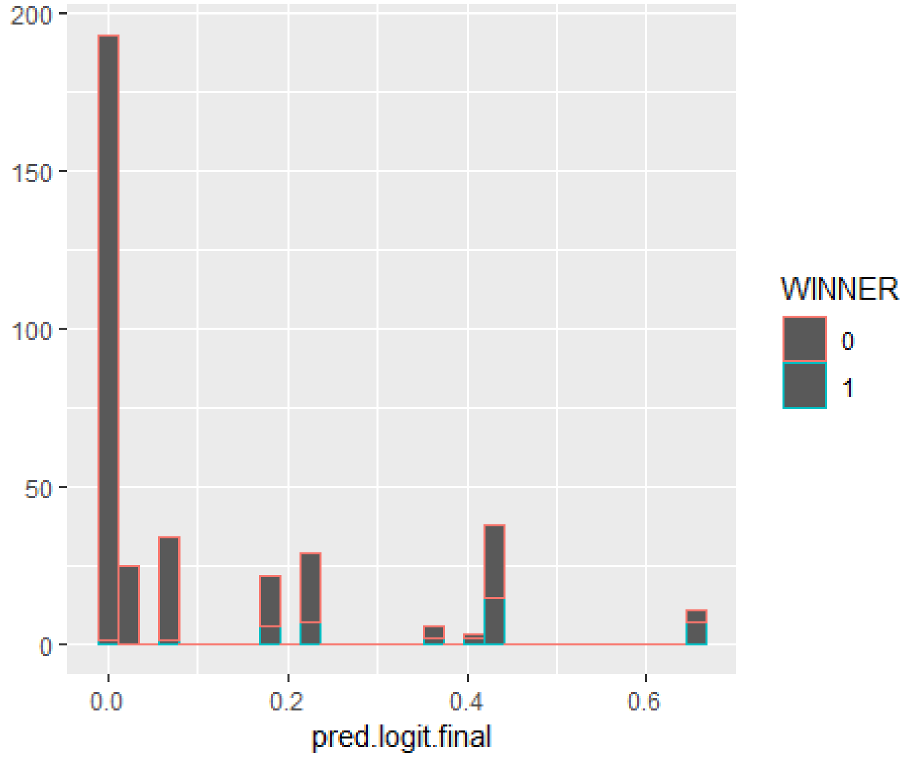
```
##
## Call:
## glm(formula = logit.eq, family = binomial, data = EL_Model_data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4619  -0.3974  -0.0998  -0.0545   3.2573
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -6.5097     1.0770  -6.044  1.5e-09 ***
## INCUMBENCY_FACTOR     0.9150     0.5098   1.795 0.072670 .
## GRADUATE              1.2095     0.4408   2.744 0.006065 **
## IS_NATIONALPARTY      4.0107     1.0332   3.882 0.000104 ***
## IS_RICH               1.0221     0.4227   2.418 0.015600 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 255.53  on 360  degrees of freedom
## Residual deviance: 159.79  on 356  degrees of freedom
##   (92 observations deleted due to missingness)
## AIC: 169.79
##
## Number of Fisher Scoring iterations: 8
```

```r
vif(model.LR.imp)
```

```
## INCUMBENCY_FACTOR          GRADUATE  IS_NATIONALPARTY           IS_RICH
##          1.030693          1.009470          1.017545          1.035098
```

```r
#predict the train set
pred.logit.final <- predict.glm(model.LR.imp, newdata=EL_Model_data.train, type="response")
qplot( pred.logit.final,data=EL_Model_data.train, color=WINNER )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 92 rows containing non-finite values (stat_bin).
```



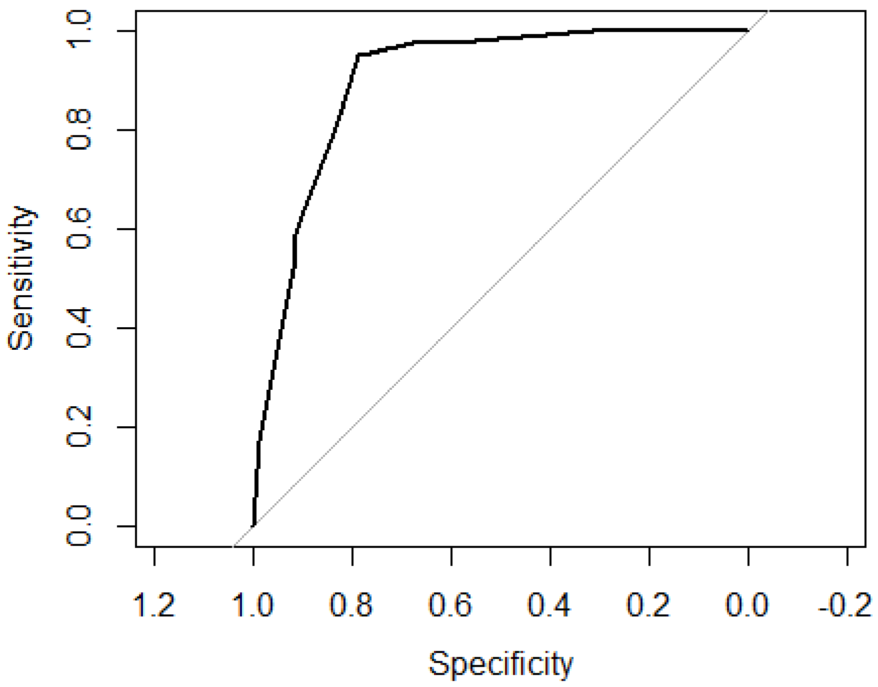
```
tab.LR.imp = data.frame(Target = EL_Model_data.train$WINNER, Prediction = pre
dict.glm(model.LR.imp, newdata=EL_Model_data.train, type="response") )
tab.LR.imp$Classification = ifelse(tab.LR.imp$Prediction>0.5,1,0)
with(tab.LR.imp, table(Target, Classification))

##       Classification
## Target   0   1
##      0 316   4
##      1  34   7

confusionMatrix(table(tab.LR.imp$Target, tab.LR.imp$Classification))

## Confusion Matrix and Statistics
##
##
##       0   1
##   0 316   4
##   1  34   7
##
##                Accuracy : 0.8947
##                  95% CI : (0.8584, 0.9244)
##     No Information Rate : 0.9695
##     P-Value [Acc > NIR] : 1
##
```

```
##                   Kappa : 0.2323
##  Mcnemar's Test P-Value : 2.546e-06
##
##             Sensitivity : 0.9029
##             Specificity : 0.6364
##          Pos Pred Value : 0.9875
##          Neg Pred Value : 0.1707
##              Prevalence : 0.9695
##          Detection Rate : 0.8753
##    Detection Prevalence : 0.8864
##       Balanced Accuracy : 0.7696
##
##        'Positive' Class : 0
##
```

```
accuracy.logit<- roc.logit<-roc(EL_Model_data.train$WINNER,pred.logit.final )
roc.logit #0.9074
```

```
##
## Call:
## roc.default(response = EL_Model_data.train$WINNER, predictor = pred.logit.
final)
##
## Data: pred.logit.final in 320 controls (EL_Model_data.train$WINNER 0) < 41
cases (EL_Model_data.train$WINNER 1).
## Area under the curve: 0.9018
```

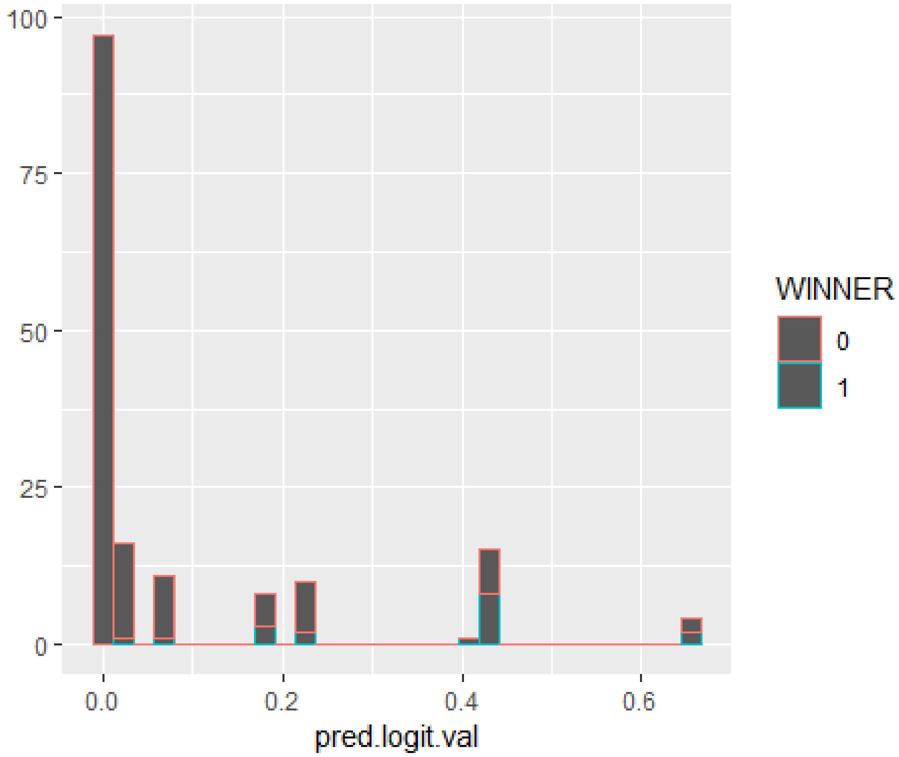
```
plot(roc.logit)
```



```
tab.LR.imp$Target<- as.character(tab.LR.imp$Target)
tab.LR.imp$Target[tab.LR.imp$Target == "0"] <- 0
tab.LR.imp$Target[tab.LR.imp$Target== "1"] <- 1

tab.LR.imp$Target <- as.numeric(tab.LR.imp$Target)
#Deciling
decile <- function(x){
  deciles <- vector(length=10)
  for (i in seq(0.1,1,.1)){
    deciles[i*10] <- quantile(x, i, na.rm=T)
  }
  return (
    ifelse(x<deciles[1], 1,
           ifelse(x<deciles[2], 2,
                  ifelse(x<deciles[3], 3,
                         ifelse(x<deciles[4], 4,
                                ifelse(x<deciles[5], 5,
                                       ifelse(x<deciles[6], 6,
                                              ifelse(x<deciles[7], 7,
                                                     ifelse(x<deciles[8], 8,
                                                            ifelse(x<deciles[
9], 9, 10
                                                            ))))))))))
}

#Assigning deciles to the data
```

```
tab.LR.imp$deciles <- decile(tab.LR.imp$Prediction)

##Ranking the data
library(data.table)
#Creating rank table
tmp_DT = data.table(tab.LR.imp)
rank <- tmp_DT[, list(
  cnt = length(Target),
  cnt_resp = sum(Target),
  cnt_non_resp = sum(Target == 0)) ,
  by=deciles][order(-deciles)]
rank$rrate <- round(rank$cnt_resp * 100 / rank$cnt,2);
rank$cum_resp <- cumsum(rank$cnt_resp)
rank$cum_non_resp <- cumsum(rank$cnt_non_resp)
rank$cum_perct_resp <- round(rank$cum_resp * 100 / sum(rank$cnt_resp),2);
rank$cum_perct_non_resp <- round(rank$cum_non_resp * 100 / sum(rank$cnt_non_r
esp),2);
rank$ks <- abs(rank$cum_perct_resp - rank$cum_perct_non_resp);
rank

##    deciles cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
## 1:      10  49       22           27 44.90       22           27
## 2:       9  38       11           27 28.95       33           54
## 3:       8  22        6           16 27.27       39           70
## 4:       7  59        1           58  1.69       40          128
## 5:       6  92        1           91  1.09       41          219
## 6:       3 101        0          101  0.00       41          320
## 7:      NA  92        8           84  8.70       49          404
##    cum_perct_resp cum_perct_non_resp    ks
## 1:          44.90               6.68 38.22
## 2:          67.35              13.37 53.98
## 3:          79.59              17.33 62.26
## 4:          81.63              31.68 49.95
## 5:          83.67              54.21 29.46
## 6:          83.67              79.21  4.46
## 7:         100.00             100.00  0.00

#ks ===> 71.49

#detach(package:neuralnet)
pred <- prediction(tab.LR.imp$Prediction, tab.LR.imp$Target)
perf <- performance(pred, "tpr", "fpr")
#plot(perf)
KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])
auc <- performance(pred,"auc");
auc <- as.numeric(auc@y.values)

gini = ineq(tab.LR.imp$Prediction, type="Gini")

auc
```

```
## [1] 0.9017912

KS

## [1] 0.7387195

gini

## [1] 0.716105

#prediction on validation set

dim(EL_Model_data.val)

## [1] 189  12

pred.logit.val <- predict.glm(model.LR.imp, newdata=EL_Model_data.val, type="
response")
qplot( pred.logit.val,data=EL_Model_data.val, color=WINNER )

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 27 rows containing non-finite values (stat_bin).
```



```
tab.LR.imp = data.frame(Target = EL_Model_data.val$WINNER, Prediction = predi
ct.glm(model.LR.imp, newdata=EL_Model_data.val, type="response") )
tab.LR.imp$Classification = ifelse(tab.LR.imp$Prediction>0.5,1,0)
with(tab.LR.imp, table(Target, Classification))
```

```
##       Classification
## Target   0   1
##      0 142   2
##      1  16   2
```

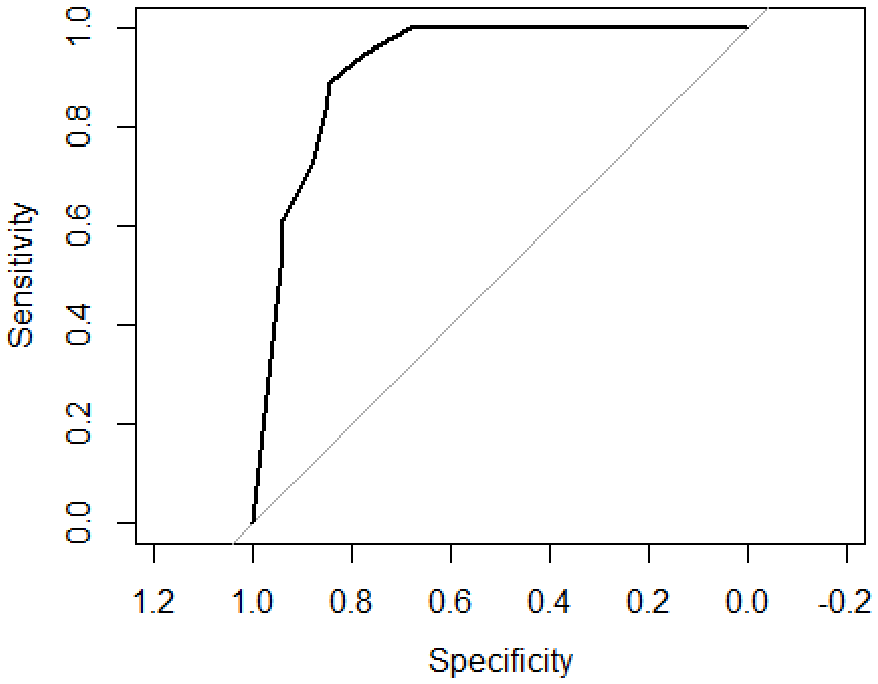
```
confusionMatrix(table(tab.LR.imp$Target, tab.LR.imp$Classification))
```

```
## Confusion Matrix and Statistics
## 
##     
##       0   1
##   0 142   2
##   1  16   2
##                                           
##                Accuracy : 0.8889
##                  95% CI : (0.8301, 0.9328)
##     No Information Rate : 0.9753
##     P-Value [Acc > NIR] : 1.000000
##                                           
##                   Kappa : 0.1474
##  Mcnemar's Test P-Value : 0.002183
##                                           
##             Sensitivity : 0.8987
##             Specificity : 0.5000
##          Pos Pred Value : 0.9861
##          Neg Pred Value : 0.1111
##              Prevalence : 0.9753
##          Detection Rate : 0.8765
##    Detection Prevalence : 0.8889
##       Balanced Accuracy : 0.6994
##                                           
##        'Positive' Class : 0
## 
```

```
accuracy.logit<- roc.logit<-roc(EL_Model_data.val$WINNER,pred.logit.val )
roc.logit
```

```
## 
## Call:
## roc.default(response = EL_Model_data.val$WINNER, predictor = pred.logit.va
l)
## 
## Data: pred.logit.val in 144 controls (EL_Model_data.val$WINNER 0) < 18 cas
es (EL_Model_data.val$WINNER 1).
## Area under the curve: 0.9203
```

```
plot(roc.logit)
```



```
tab.LR.imp$Target<- as.character(tab.LR.imp$Target)
tab.LR.imp$Target[tab.LR.imp$Target == "0"] <- 0
tab.LR.imp$Target[tab.LR.imp$Target== "1"] <- 1

tab.LR.imp$Target <- as.numeric(tab.LR.imp$Target)

#Assigning deciles to the data
tab.LR.imp$deciles <- decile(tab.LR.imp$Prediction)

#Creating rank table
tmp_DT = data.table(tab.LR.imp)
rank <- tmp_DT[, list(
  cnt = length(Target),
  cnt_resp = sum(Target),
  cnt_non_resp = sum(Target == 0)) ,
  by=deciles][order(-deciles)]
rank$rrate <- round(rank$cnt_resp * 100 / rank$cnt,2);
rank$cum_resp <- cumsum(rank$cnt_resp)
rank$cum_non_resp <- cumsum(rank$cnt_non_resp)
rank$cum_perct_resp <- round(rank$cum_resp * 100 / sum(rank$cnt_resp),2);
rank$cum_perct_non_resp <- round(rank$cum_non_resp * 100 / sum(rank$cnt_non_r
esp),2);
rank$ks <- abs(rank$cum_perct_resp - rank$cum_perct_non_resp);
rank
```

```
##    deciles cnt cnt_resp cnt_non_resp rrate cum_resp cum_non_resp
## 1:      10  19       10            9 52.63       10            9
## 2:       9  17        5           12 29.41       15           21
## 3:       8  13        2           11 15.38       17           32
## 4:       7  16        1           15  6.25       18           47
## 5:       6  37        0           37  0.00       18           84
## 6:       4  60        0           60  0.00       18          144
## 7:      NA  27        3           24 11.11       21          168
##    cum_perct_resp cum_perct_non_resp    ks
## 1:          47.62               5.36 42.26
## 2:          71.43              12.50 58.93
## 3:          80.95              19.05 61.90
## 4:          85.71              27.98 57.73
## 5:          85.71              50.00 35.71
## 6:          85.71              85.71  0.00
## 7:         100.00             100.00  0.00
```

```
pred <- prediction(tab.LR.imp$Prediction, tab.LR.imp$Target)
perf <- performance(pred, "tpr", "fpr")
#plot(perf)
KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])
auc <- performance(pred,"auc");
auc <- as.numeric(auc@y.values)

gini = ineq(tab.LR.imp$Prediction, type="Gini")

auc
```

```
## [1] 0.9203318
```

```
KS
```

```
## [1] 0.7361111
```

```
gini
```

```
## [1] 0.7703593
```