R Notebook

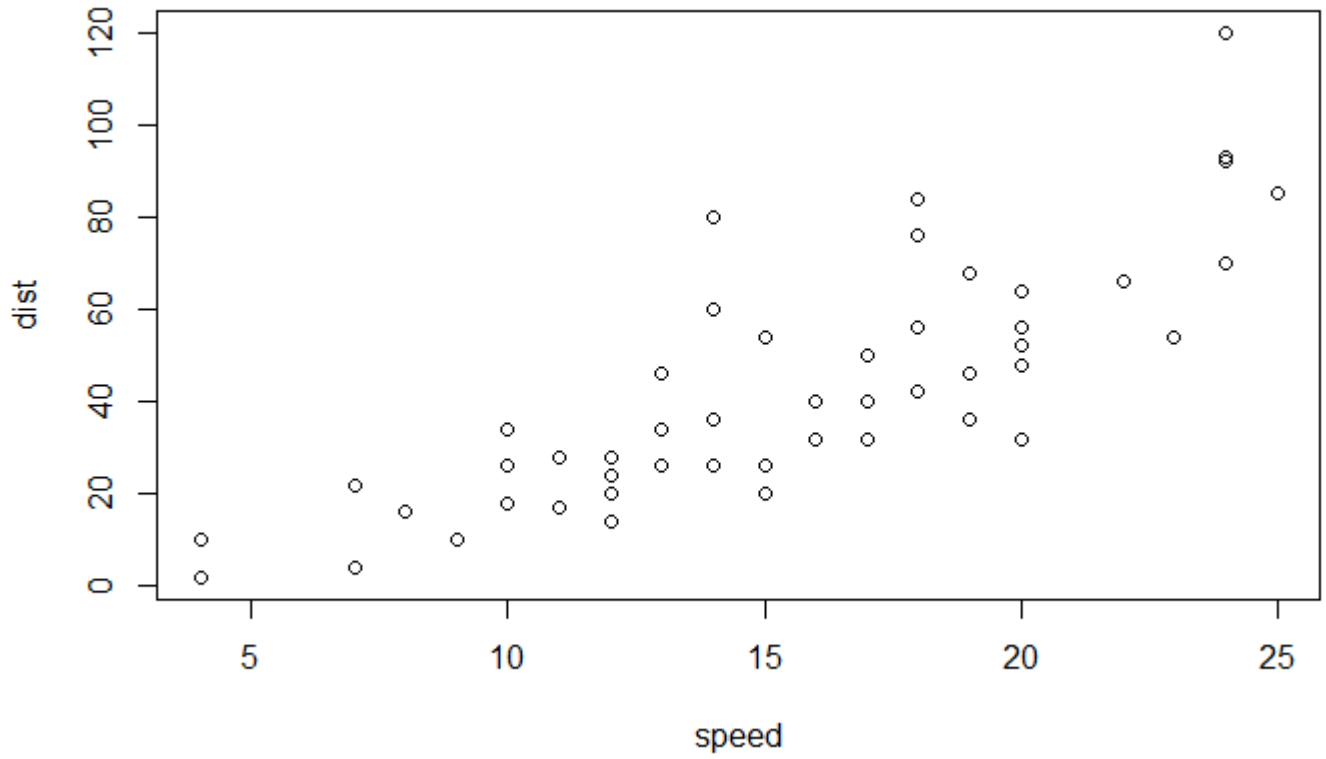
Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Hide

```
library('naivebayes')
```

```
package 愀笣naivebayes愀笮 was built under R version 3.6.3naivebayes 0.9.7 loaded
```

Hide

```
library('pROC')
```

```
package 'pROC' was built under R version 3.6.3Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

Hide

```
library('class')
library('rpart')
library('SwarmSVM')
```

```
package 'SwarmSVM' was built under R version 3.6.3
```

Hide

```
train_indessa <- read.csv("~/R/ML_Artivatic_dataset/train_indessa.csv", stringsAsFactors=FALSE)
```

Hide

```
term_total = train_indessa$term
term_num = c()
for (t in term_total) {
  term_num = c(term_num, as.integer(substr(t,1,2)))
}
```

Hide

```
home_own = c()
for (h in train_indessa$home_ownership) {
  if (h == "ANY") {
    o = 1
  } else if (h == "MORTGAGE") {
    o = 2
  } else if (h == "NONE") {
    o = 3
  } else if (h == "OTHER") {
    o = 4
  } else if (h == "OWN") {
    o = 5
  } else {
    o = 6
  }
  home_own = c(home_own, o)
}
```

Hide

```
ver_status = c()
for (v in train_indessa$verification_status) {
  if (v == "Not Verified") {
    s = 1
  } else if (h == "Source Verified") {
    s = 2
  } else {
    s = 3
  }
  ver_status = c(ver_status, s)
}
```

Hide

```
train_indessa$delinq_2yrs[is.na(train_indessa$delinq_2yrs)] <- 0
```

Hide

```
train_indessa$inq_last_6mths[is.na(train_indessa$inq_last_6mths)] <- 0
```

Hide

```
train_indessa$mths_since_last_delinq[is.na(train_indessa$mths_since_last_delinq)] <- 0
```

Hide

```
train_indessa$mths_since_last_record[is.na(train_indessa$mths_since_last_record)] <- 0
```

Hide

```
train_indessa$open_acc[is.na(train_indessa$open_acc)] <- 0
```

Hide

```
train_indessa$pub_rec[is.na(train_indessa$pub_rec)] <- 0
```

Hide

```
train_indessa$revol_util[is.na(train_indessa$revol_util)] <- 0
```

Hide

```
train_indessa$total_acc[is.na(train_indessa$total_acc)] <- 0
```

Hide

```
train_indessa$collections_12_mths_ex_med[is.na(train_indessa$collections_12_mths_ex_med)] <- 0
```

Hide

```
train_indessa$mths_since_last_major_derog[is.na(train_indessa$mths_since_last_major_derog)] <- 0
```

Hide

```
app_type = c()
for (a in train_indessa$application_type) {
  if (a == "INDIVIDUAL") {
    t = 1
  } else {
    t = 2
  }
  app_type = c(app_type, t)
}
```

Hide

```
gra = train_indessa$grade
gd = c()
for (g in gra) {
  if (g == "A") {
    d = 1
  } else if (g == "B") {
    d = 2
  } else if (g == "C") {
    d = 3
  } else if (g == "D") {
    d = 4
  } else {
    d = 5
  }
  gd = c(gd, d)
}
```

Hide

```
train_indessa$acc_now_delinq[is.na(train_indessa$acc_now_delinq)] <- 0
```

Hide

```
train_indessa$tot_coll_amt[is.na(train_indessa$tot_coll_amt)] <- 0
```

Hide

```
train_indessa$tot_cur_bal[is.na(train_indessa$tot_cur_bal)] <- 0
```

Hide

```
summary(train_indessa$tot_cur_bal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0   23208   65473  128544  196043 8000078 
```

Hide

```
train_indessa$total_rev_hi_lim[is.na(train_indessa$total_rev_hi_lim)] <- 0
```

Hide

```
train_indessa$annual_inc[is.na(train_indessa$annual_inc)] <- 40000
```

Hide

```
vanilla_train <-data.frame(train_indessa$member_id, train_indessa$loan_amnt, train_indessa$funded_amnt, train_indessa$funded_amnt_inv, term_num, train_indessa$int_rate, gd, home_own, train_indessa$annual_inc, ver_status, train_indessa$dti, train_indessa$delinq_2yrs, train_indessa$inq_last_6mths, train_indessa$mths_since_last_delinq, train_indessa$mths_since_last_record, train_indessa$open_acc, train_indessa$pub_rec, train_indessa$revol_bal, train_indessa$revol_util, train_indessa$total_acc, train_indessa$total_rec_int, train_indessa$total_rec_late_fee, train_indessa$recoveries, train_indessa$collection_recovery_fee, train_indessa$collections_12_mths_ex_med, train_indessa$mths_since_last_major_derog, app_type, l_w_p, train_indessa$acc_now_delinq, train_indessa$tot_coll_amt, train_indessa$tot_cur_bal, train_indessa$total_rev_hi_lim, train_indessa$loan_status)
```

Hide

```
for (i in 2:32) {
  print(colnames(vanilla_train)[i])
  print(colnames(vanilla_train)[33])
  print(cor(vanilla_train[,i], vanilla_train[,33]))
  print(abs(cor(vanilla_train[,i], vanilla_train[,33])))
}
```

```
[1] "train_indessa.loan_amnt"
[1] "train_indessa.loan_status"
[1] -0.09571273
[1] 0.09571273
[1] "train_indessa.funded_amnt"
[1] "train_indessa.loan_status"
[1] -0.09756343
[1] 0.09756343
[1] "train_indessa.funded_amnt_inv"
[1] "train_indessa.loan_status"
[1] -0.1018775
[1] 0.1018775
[1] "term_num"
[1] "train_indessa.loan_status"
[1] -0.1309594
[1] 0.1309594
[1] "train_indessa.int_rate"
[1] "train_indessa.loan_status"
[1] 0.002589613
[1] 0.002589613
[1] "gd"
[1] "train_indessa.loan_status"
[1] -0.05941464
[1] 0.05941464
[1] "home_own"
[1] "train_indessa.loan_status"
[1] -0.003807015
[1] 0.003807015
[1] "train_indessa.annual_inc"
[1] "train_indessa.loan_status"
[1] -0.008601625
[1] 0.008601625
[1] "ver_status"
[1] "train_indessa.loan_status"
[1] -0.06886676
[1] 0.06886676
[1] "train_indessa.dti"
[1] "train_indessa.loan_status"
[1] -0.1340201
[1] 0.1340201
[1] "train_indessa.delinq_2yrs"
[1] "train_indessa.loan_status"
[1] -0.04573019
[1] 0.04573019
[1] "train_indessa.inq_last_6mths"
[1] "train_indessa.loan_status"
[1] 0.08730634
[1] 0.08730634
[1] "train_indessa.mths_since_last_delinq"
[1] "train_indessa.loan_status"
[1] -0.02330996
[1] 0.02330996
[1] "train_indessa.mths_since_last_record"
```

```
[1] "train_indessa.loan_status"
[1] -0.02722468
[1] 0.02722468
[1] "train_indessa.open_acc"
[1] "train_indessa.loan_status"
[1] -0.06512924
[1] 0.06512924
[1] "train_indessa.pub_rec"
[1] "train_indessa.loan_status"
[1] -0.04889524
[1] 0.04889524
[1] "train_indessa.revol_bal"
[1] "train_indessa.loan_status"
[1] -0.0400513
[1] 0.0400513
[1] "train_indessa.revol_util"
[1] "train_indessa.loan_status"
[1] -0.04742907
[1] 0.04742907
[1] "train_indessa.total_acc"
[1] "train_indessa.loan_status"
[1] -0.002717927
[1] 0.002717927
[1] "train_indessa.total_rec_int"
[1] "train_indessa.loan_status"
[1] 0.038344
[1] 0.038344
[1] "train_indessa.total_rec_late_fee"
[1] "train_indessa.loan_status"
[1] -0.0043528
[1] 0.0043528
[1] "train_indessa.recoveries"
[1] "train_indessa.loan_status"
[1] -0.06208382
[1] 0.06208382
[1] "train_indessa.collection_recovery_fee"
[1] "train_indessa.loan_status"
[1] -0.04282325
[1] 0.04282325
[1] "train_indessa.collections_12_mths_ex_med"
[1] "train_indessa.loan_status"
[1] -0.03393834
[1] 0.03393834
[1] "train_indessa.mths_since_last_major_derog"
[1] "train_indessa.loan_status"
[1] -0.07474282
[1] 0.07474282
[1] "app_type"
[1] "train_indessa.loan_status"
[1] -0.01313346
[1] 0.01313346
[1] "l_w_p"
[1] "train_indessa.loan_status"
[1] 0.2680722
```

```
[1] 0.2680722
[1] "train_indessa.acc_now_delinq"
[1] "train_indessa.loan_status"
[1] -0.01441698
[1] 0.01441698
[1] "train_indessa.tot_coll_amt"
[1] "train_indessa.loan_status"
[1] -0.02337655
[1] 0.02337655
[1] "train_indessa.tot_cur_bal"
[1] "train_indessa.loan_status"
[1] -0.0822156
[1] 0.0822156
[1] "train_indessa.total_rev_hi_lim"
[1] "train_indessa.loan_status"
[1] -0.1052294
[1] 0.1052294
```

Select features with absolute correlation with loan_status > 0.1.

Hide

```
vanilla_train.feature_selection <-data.frame(train_indessa$member_id, train_indessa$funded_amnt_
inv, term_num, train_indessa$dti, l_w_p, train_indessa$total_rev_hi_lim, train_indessa$loan_stat
us)
```

Hide

```
test_indessa <- read.csv("~/R/ML_Artivatic_dataset/test_indessa.csv", stringsAsFactors=FALSE)
summary(test_indessa)
```

```
   member_id          loan_amnt      funded_amnt    funded_amnt_inv     term           batch_enr
olled        int_rate
 Min.   :   70626   Min.   :  500   Min.   :  500   Min.   :    0   Length:354951      Length:35
4951      Min.   : 5.32
 1st Qu.:10889411   1st Qu.: 8000   1st Qu.: 8000   1st Qu.: 8000   Class :character   Class :ch
aracter   1st Qu.: 9.99
 Median :37086503   Median :13000   Median :13000   Median :13000   Mode  :character   Mode  :ch
aracter   Median :12.99
 Mean   :34996354   Mean   :14752   Mean   :14738   Mean   :14699
Mean   :13.25
 3rd Qu.:58448923   3rd Qu.:20000   3rd Qu.:20000   3rd Qu.:20000
3rd Qu.:16.20
 Max.   :73544827   Max.   :35000   Max.   :35000   Max.   :35000
Max.   :28.99

    grade            sub_grade          emp_title          emp_length        home_ownership
annual_inc      verification_status
 Length:354951      Length:354951      Length:354951      Length:354951      Length:354951
Min.   :      0   Length:354951
 Class :character   Class :character   Class :character   Class :character   Class :character
1st Qu.:  45000   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
Median :  65000   Mode  :character

Mean   :  75024

3rd Qu.:  90000

Max.   :9000000

NA's   :1
  pymnt_plan            desc             purpose             title             zip_code
addr_state             dti
 Length:354951      Length:354951      Length:354951      Length:354951      Length:354951
Length:354951      Min.   :   0.00
 Class :character   Class :character   Class :character   Class :character   Class :character
Class :character   1st Qu.:  11.89
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
Mode  :character   Median :  17.65

Mean   :  18.18

3rd Qu.:  23.94

Max.   :9999.00

  delinq_2yrs      inq_last_6mths    mths_since_last_delinq mths_since_last_record    open_acc
pub_rec         revol_bal
 Min.   : 0.0000   Min.   : 0.0000   Min.   :  0.00         Min.   :  0.00         Min.   : 0.00
Min.   : 0.000   Min.   :      0
 1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 15.00         1st Qu.: 51.00         1st Qu.: 8.00
1st Qu.: 0.000   1st Qu.:   6441
 Median : 0.0000   Median : 0.0000   Median : 31.00         Median : 70.00         Median :11.00
```

```
Median : 0.000   Median :  11873
 Mean   : 0.3144   Mean   : 0.6946   Mean   : 34.08         Mean   : 70.16         Mean   :11.55
Mean   : 0.196   Mean   :  16920
 3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 50.00         3rd Qu.: 92.00         3rd Qu.:14.00
3rd Qu.: 0.000   3rd Qu.:  20811
 Max.   :39.0000   Max.   :33.0000   Max.   :188.00         Max.   :129.00         Max.   :76.00
Max.   :54.000   Max.   :2904836
 NA's   :13        NA's   :13        NA's   :181758         NA's   :300021         NA's   :13
NA's   :13
   revol_util       total_acc      initial_list_status total_rec_int     total_rec_late_fee   re
coveries       collection_recovery_fee
 Min.   :  0.00   Min.   :  1.00   Length:354951       Min.   :    0.0   Min.   :  0.0000   Min.
:    0.00   Min.   :   0.000
 1st Qu.: 37.70   1st Qu.: 17.00   Class :character    1st Qu.:  441.3   1st Qu.:  0.0000   1st
Qu.:    0.00   1st Qu.:   0.000
 Median : 56.00   Median : 24.00   Mode  :character    Median : 1074.1   Median :  0.0000   Medi
an :    0.00   Median :   0.000
 Mean   : 55.08   Mean   : 25.27                       Mean   : 1756.9   Mean   :  0.3993   Mean
:   46.22   Mean   :   4.913
 3rd Qu.: 73.60   3rd Qu.: 32.00                       3rd Qu.: 2243.1   3rd Qu.:  0.0000   3rd
Qu.:    0.00   3rd Qu.:   0.000
 Max.   :182.80   Max.   :169.00                       Max.   :23062.5   Max.   :286.7476   Max.
:29282.07   Max.   :5569.920
 NA's   :215      NA's   :13
 collections_12_mths_ex_med mths_since_last_major_derog application_type   verification_status_j
oint last_week_pay
 Min.   : 0.0000            Min.   :  0.00              Length:354951      Length:354951
Length:354951
 1st Qu.: 0.0000            1st Qu.: 27.00              Class :character   Class :character
Class :character
 Median : 0.0000            Median : 44.00              Mode  :character   Mode  :character
Mode  :character
 Mean   : 0.0145            Mean   : 44.08
 3rd Qu.: 0.0000            3rd Qu.: 61.00
 Max.   :20.0000            Max.   :188.00
 NA's   :50                 NA's   :266228
 acc_now_delinq      tot_coll_amt      tot_cur_bal      total_rev_hi_lim
 Min.   :0.000000   Min.   :      0   Min.   :      0   Min.   :      0
 1st Qu.:0.000000   1st Qu.:      0   1st Qu.:  29874   1st Qu.:  13900
 Median :0.000000   Median :      0   Median :  80369   Median :  23700
 Mean   :0.004956   Mean   :    244   Mean   : 139314   Mean   :  32051
 3rd Qu.:0.000000   3rd Qu.:      0   3rd Qu.: 207800   3rd Qu.:  39700
 Max.   :5.000000   Max.   :9152545   Max.   :4447397   Max.   :9999999
 NA's   :13         NA's   :28272     NA's   :28272     NA's   :28272
```

Hide

```
term_total_test = test_indessa$term
term_num_test = c()
for (t_test in term_total_test) {
  term_num_test = c(term_num_test, as.integer(substr(t_test,1,2)))
}
```

Hide

```
test_indessa$total_rev_hi_lim[is.na(test_indessa$total_rev_hi_lim)] <- 0
```

Form a similar test_set.

Hide

```
vanilla_test.feature_selection <-data.frame(test_indessa$member_id, test_indessa$funded_amnt_in
v, term_num_test, test_indessa$dti, l_w_p_test, test_indessa$total_rev_hi_lim)
```

75% of the sample size

Hide

```
smp_size <- floor(0.675 * nrow(vanilla_train.feature_selection))
```

set the seed to make your partition reproducible

Hide

```
set.seed(123)
train_ind <- sample(seq_len(nrow(vanilla_train.feature_selection)), size = smp_size)
```

Hide

```
train <- vanilla_train.feature_selection[train_ind, ]
validation <- vanilla_train.feature_selection[-train_ind, ]
```

Hide

```
normalize <- function(x)
{
    return((x- min(x)) /(max(x)-min(x)))
}
```

Hide

```
train.normalized = data.frame(train$train_indessa.member_id)
validation.normalized = data.frame(validation$train_indessa.member_id)
```

Hide

```
for (i in 2:6) {
  train.normalized[,i] <- normalize(train[,i])
  validation.normalized[,i] <- normalize(validation[,i])
}
```

Hide

```
train.normalized[,7] = data.frame(train$train_indessa.loan_status)
validation.normalized[,7] = data.frame(validation$train_indessa.loan_status)
```

Hide

```
colnames(train.normalized) = colnames(train)
colnames(validation.normalized) = colnames(train)
```

GNB on original data

Hide

```
M = matrix(as.numeric(unlist(train[,2:6])), ncol=5, byrow=F)
y <- factor(train$train_indessa.loan_status)
```

Hide

```
colnames(M) <- c("funded_amnt_inv", "term_num", "dti", "l_w_p", "total_rev_hi_lim")
print(colnames(M))
```

```
[1] "funded_amnt_inv"  "term_num"         "dti"              "l_w_p"            "total_rev_hi_li
m"
```

Train the Gaussian Naive Bayes

Hide

```
gnb <- gaussian_naive_bayes(x = M, y = y)
```

Hide

```
summary(gnb)
```

```


========================================================= Gaussian Naive Bayes =================
=======================================

- Call: gaussian_naive_bayes(x = M, y = y)
- Samples: 359388
- Features: 5
- Prior probabilities:
    - 0: 0.764
    - 1: 0.236

-------------------------------------------------------------------------------------------------
---------------------------------------
```

Hide

```
N = matrix(as.numeric(unlist(validation[,2:6])), ncol=5, byrow=F)
```

Hide

```
colnames(N) <- c("funded_amnt_inv", "term_num", "dti", "l_w_p", "total_rev_hi_lim")
print(colnames(N))
```

```
[1] "funded_amnt_inv"  "term_num"         "dti"              "l_w_p"            "total_rev_hi_li
m"
```

Classification

Hide

```
gnb.original.val_output = predict(gnb, newdata = N, type = "prob") # head(gnb %class% M)
```

Hide

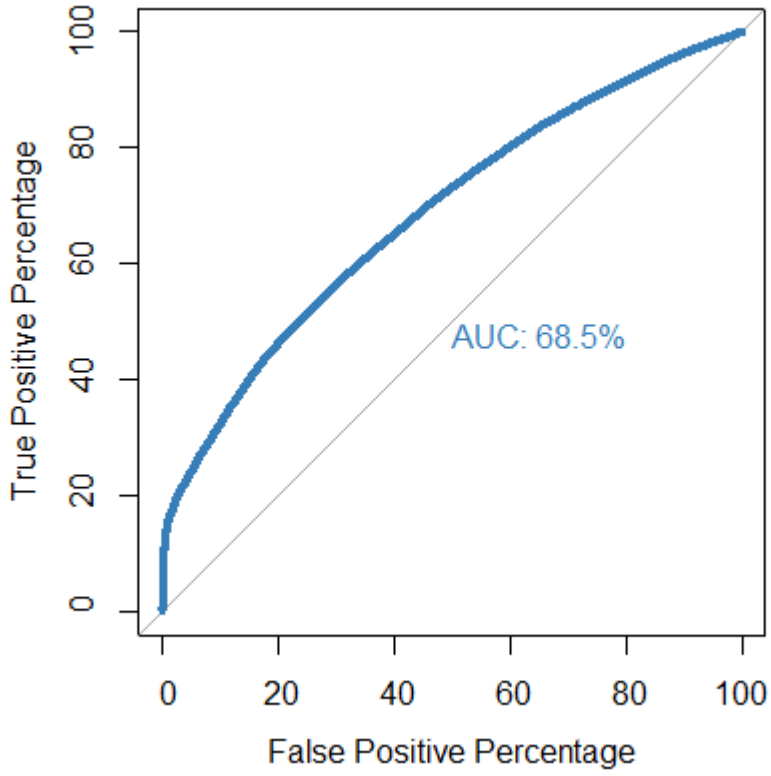
```
par(pty = 's')
roc(factor(validation$train_indessa.loan_status), gnb.original.val_output[,1], plot = TRUE, lega
cy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Perce
ntage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```


Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = gnb.origina
l.val_output[, 1], percent = TRUE,     plot = TRUE, legacy.axes = TRUE, xlab = "False Positive P
ercentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     print.auc = TRU
E)

Data: gnb.original.val_output[, 1] in 132036 controls (factor(validation$train_indessa.loan_stat
us) 0) > 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 68.5%
```



GNB on normalized data

Hide

```
M = matrix(as.numeric(unlist(train.normalized[,2:6])), ncol=5, byrow=F)
y <- factor(train$train_indessa.loan_status)
```

Hide

```
colnames(M) <- c("funded_amnt_inv", "term_num", "dti", "l_w_p", "total_rev_hi_lim")
print(colnames(M))
```

```
[1] "funded_amnt_inv"  "term_num"         "dti"              "l_w_p"            "total_rev_hi_li
m"
```

Train the Gaussian Naive Bayes

Hide

```
gnb <- gaussian_naive_bayes(x = M, y = y)
```

Hide

```
summary(gnb)
```

```
========================================================= Gaussian Naive Bayes ================
========================================

- Call: gaussian_naive_bayes(x = M, y = y)
- Samples: 359388
- Features: 5
- Prior probabilities:
    - 0: 0.764
    - 1: 0.236

-------------------------------------------------------------------------------------------------
----------------------------------------
```

Hide

```
N = matrix(as.numeric(unlist(validation.normalized[,2:6])), ncol=5, byrow=F)
```

Hide

```
colnames(N) <- c("funded_amnt_inv", "term_num", "dti", "l_w_p", "total_rev_hi_lim")
print(colnames(N))
```

```
[1] "funded_amnt_inv"  "term_num"         "dti"              "l_w_p"            "total_rev_hi_li
m"
```

Classification

Hide

```
gnb.normalized.val_output = predict(gnb, newdata = N, type = "prob") # head(gnb %class% M)
```

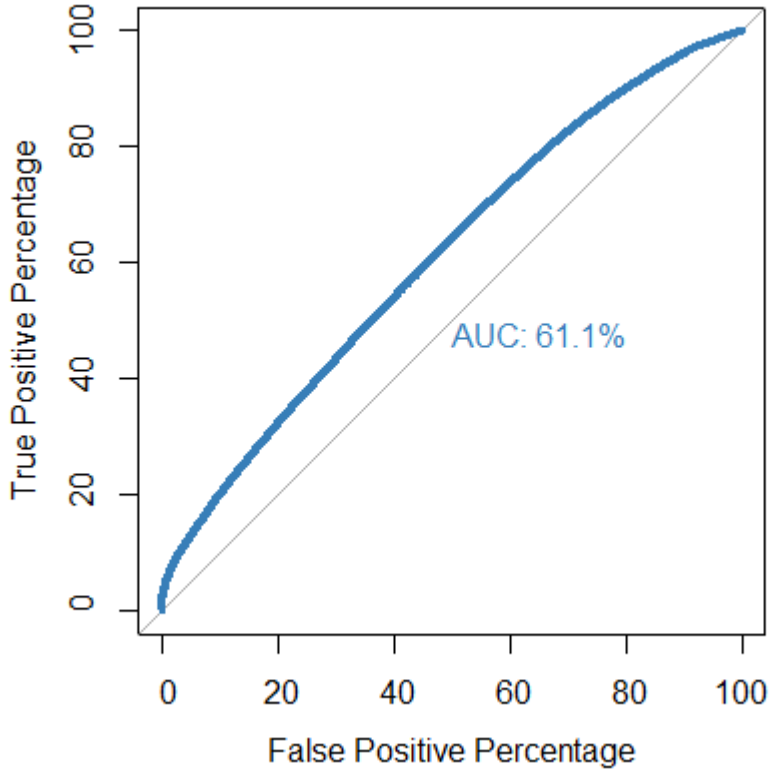
Hide

```
par(pty = 's')
roc(factor(validation$train_indessa.loan_status), gnb.normalized.val_output[,1], plot = TRUE, le
gacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Per
centage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```
Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = gnb.normali
zed.val_output[, 1], percent = TRUE,     plot = TRUE, legacy.axes = TRUE, xlab = "False Positive
Percentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     print.auc = TR
UE)

Data: gnb.normalized.val_output[, 1] in 132036 controls (factor(validation$train_indessa.loan_st
atus) 0) > 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 61.06%
```



Train the Logistic Regression

Hide

```
glm.fit <- glm(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv + term_num +
train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim, data = train) # , family = binomial,
model = TRUE)
```

Hide

```
summary(glm.fit)
```

```
Call:
glm(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv +
    term_num + train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7890  -0.2566  -0.1604   0.0147   2.8246

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    3.902e-01  3.184e-03  122.53   <2e-16 ***
train_indessa.funded_amnt_inv -1.086e-06  9.385e-08  -11.57   <2e-16 ***
term_num                      -3.785e-03  6.765e-05  -55.95   <2e-16 ***
train_indessa.dti             -4.587e-03  8.129e-05  -56.42   <2e-16 ***
l_w_p                          2.330e-03  1.528e-05  152.46   <2e-16 ***
train_indessa.total_rev_hi_lim -8.183e-07  2.285e-08  -35.81   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.1622419)

    Null deviance: 64803  on 359387  degrees of freedom
Residual deviance: 58307  on 359382  degrees of freedom
AIC: 366300

Number of Fisher Scoring iterations: 2
```

Classification

Hide

```
glm.original.val_output = predict.glm(glm.fit, newdata = validation, type = "terms") # head(gnb
 %class% M)
```

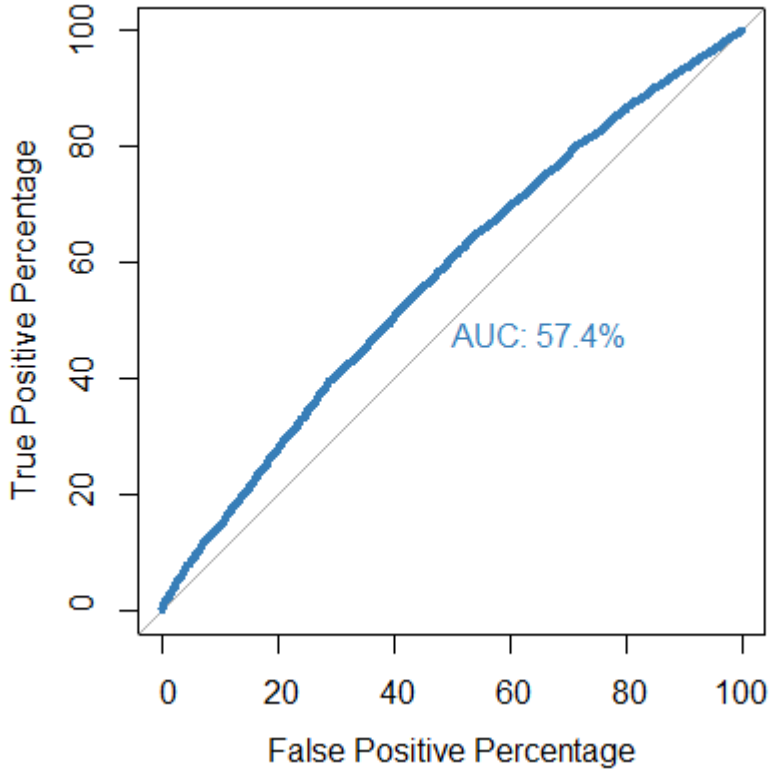
Hide

```
par(pty = 's')
roc(factor(validation$train_indessa.loan_status), glm.original.val_output[,1], plot = TRUE, lega
cy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Perce
ntage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = glm.origina
l.val_output[, 1], percent = TRUE,     plot = TRUE, legacy.axes = TRUE, xlab = "False Positive P
ercentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     print.auc = TRU
E)

Data: glm.original.val_output[, 1] in 132036 controls (factor(validation$train_indessa.loan_stat
us) 0) < 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 57.36%
```



Train the Logistic Regression

Hide

```
glm.fit <- glm(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv + term_num +
train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim, data = train.normalized) # , family
 = binomial, model = TRUE)
```

Hide

```
summary(glm.fit)
```

```
Call:
glm(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv +
    term_num + train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim,
    data = train.normalized)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7890  -0.2566  -0.1604   0.0147   2.8246

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    0.253947   0.002234  113.65   <2e-16 ***
train_indessa.funded_amnt_inv -0.037997   0.003285  -11.57   <2e-16 ***
term_num                      -0.090835   0.001624  -55.95   <2e-16 ***
train_indessa.dti             -3.084640   0.054668  -56.42   <2e-16 ***
l_w_p                          0.708297   0.004646  152.46   <2e-16 ***
train_indessa.total_rev_hi_lim -1.635596   0.045674  -35.81   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.1622419)

    Null deviance: 64803  on 359387  degrees of freedom
Residual deviance: 58307  on 359382  degrees of freedom
AIC: 366300

Number of Fisher Scoring iterations: 2
```

Classification

Hide

```
glm.normalized.val_output = predict.glm(glm.fit, newdata = validation.normalized, type = "terms"
) # head(gnb %class% M)
```

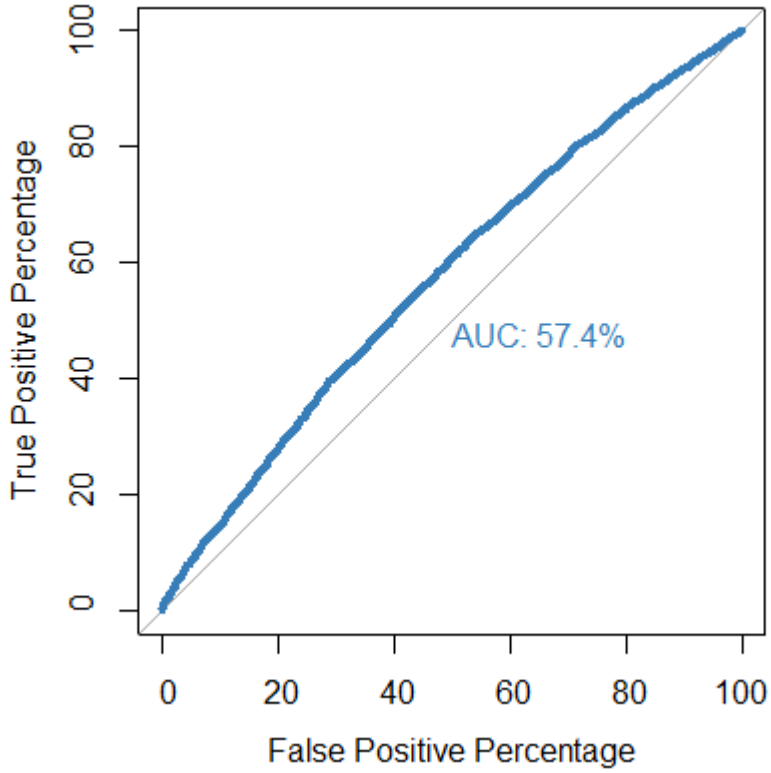
Hide

```
par(pty = 's')
roc(factor(validation.normalized$train_indessa.loan_status), glm.normalized.val_output[,1], plot
= TRUE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Po
sitive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
Call:
roc.default(response = factor(validation.normalized$train_indessa.loan_status),     predictor =
glm.normalized.val_output[, 1], percent = TRUE,     plot = TRUE, legacy.axes = TRUE, xlab = "Fal
se Positive Percentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     pr
int.auc = TRUE)

Data: glm.normalized.val_output[, 1] in 132036 controls (factor(validation.normalized$train_inde
ssa.loan_status) 0) < 41004 cases (factor(validation.normalized$train_indessa.loan_status) 1).
Area under the curve: 57.36%
```



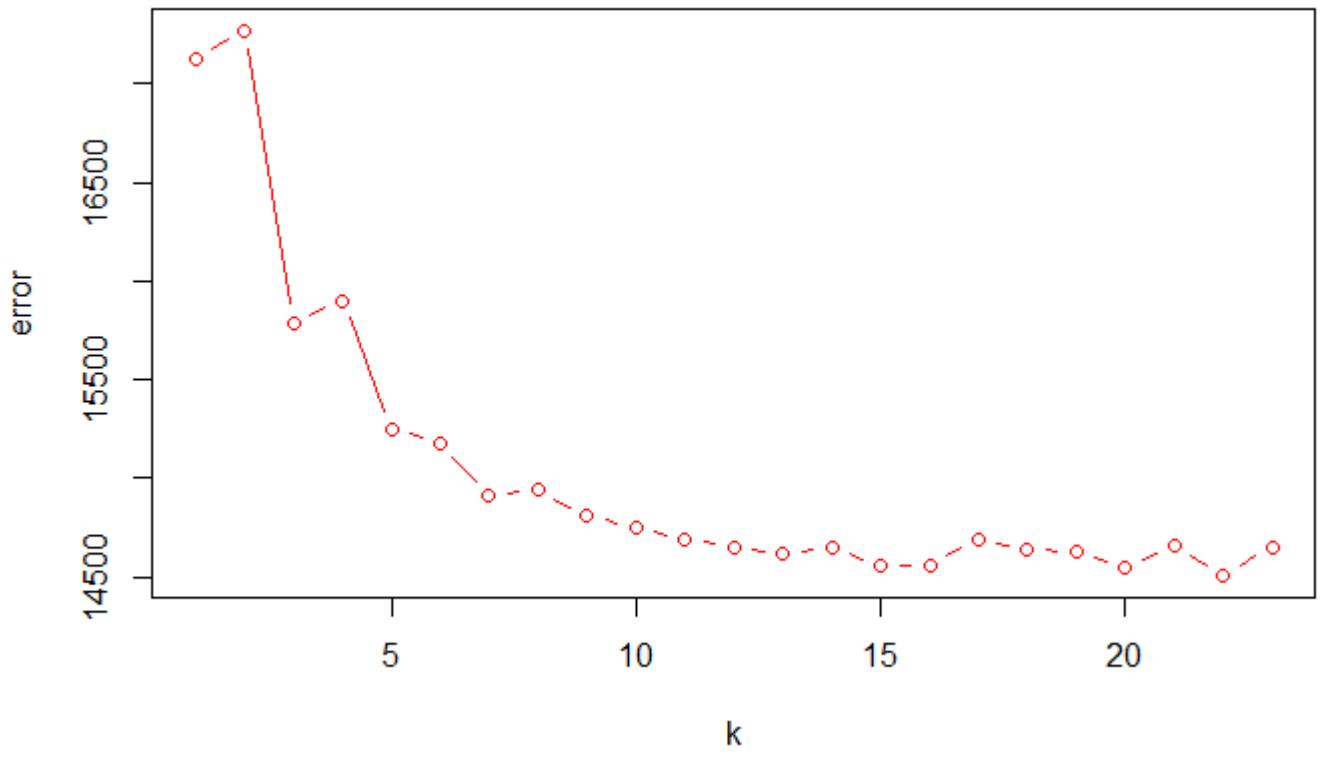
K - Nearest Neighbors on original data

Hide

```
RES = c()
smp_size_1 <- floor(0.1 * nrow(train))
smp_size_2 <- floor(0.1 * nrow(validation))
for (i in 1:23) {
  # set.seed(123)
  train_ind <- sample(seq_len(nrow(train)), size = smp_size_1)
  val_ind <- sample(seq_len(nrow(validation)), size = smp_size_2)
  knn.original.train <- train[train_ind,]
  knn.original.val <- validation[val_ind,]
  result = knn(train = knn.original.train[,2:6], test = knn.original.val[,2:6], cl = knn.origina
l.train[,7], k=i)
  err = sum(abs(as.numeric(unlist(result)) - knn.original.val[,7]), na.rm = TRUE)
  RES = c(RES,err)
}
```

Hide

```
plot(RES, type = 'b', col = 'red', xlab = "k", ylab = "error")
```



Hide

```
knn.original.result = knn(train = train[,2:6], test = validation[,2:6], cl = train[,7], k=5, pro
b = TRUE)
knn.original.result.prob <- attributes(knn.original.result)$prob
```

Hide

```
knn.original.result.prob.treated = knn.original.result.prob
for (r in 1:173040) {
  t = knn.original.result.prob[r]
  if (knn.original.result[r] == 1) {
    knn.original.result.prob.treated[r] = 1 - t
  } else {
    knn.original.result.prob.treated[r] = t
  }
}
```

Hide

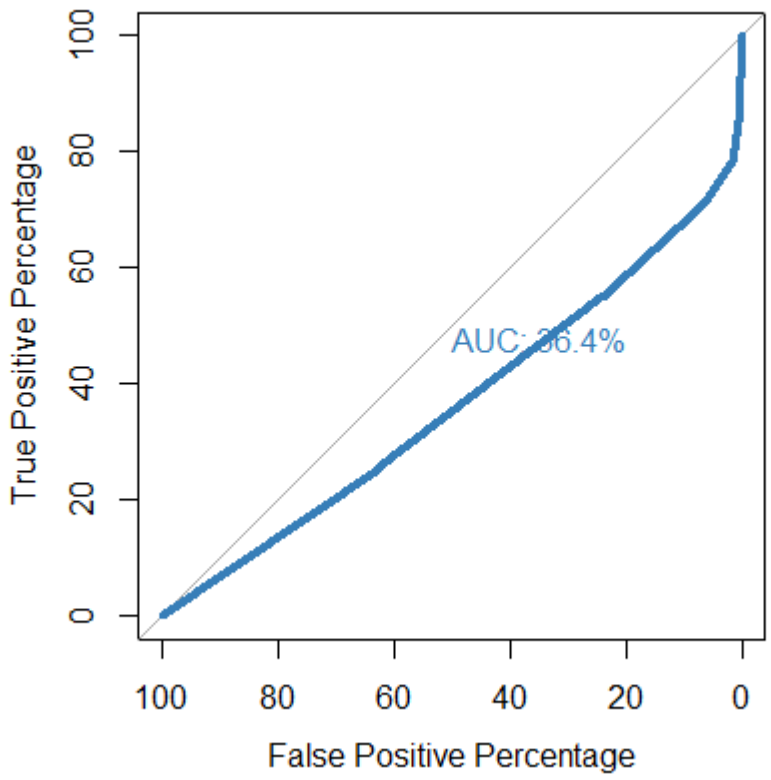
```
par(pty = 's')
roc(factor(validation$train_indessa.loan_status), knn.original.result.prob.treated, plot = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```


Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = knn.original.result.prob.treated, percent = TRUE,     plot = TRUE, xlab = "False Positive Percentage", ylab = "True Positive Percentage",     col = "#377eb8", lwd = 4, print.auc = TRUE)

Data: knn.original.result.prob.treated in 132036 controls (factor(validation$train_indessa.loan_status) 0) < 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 36.36%
```



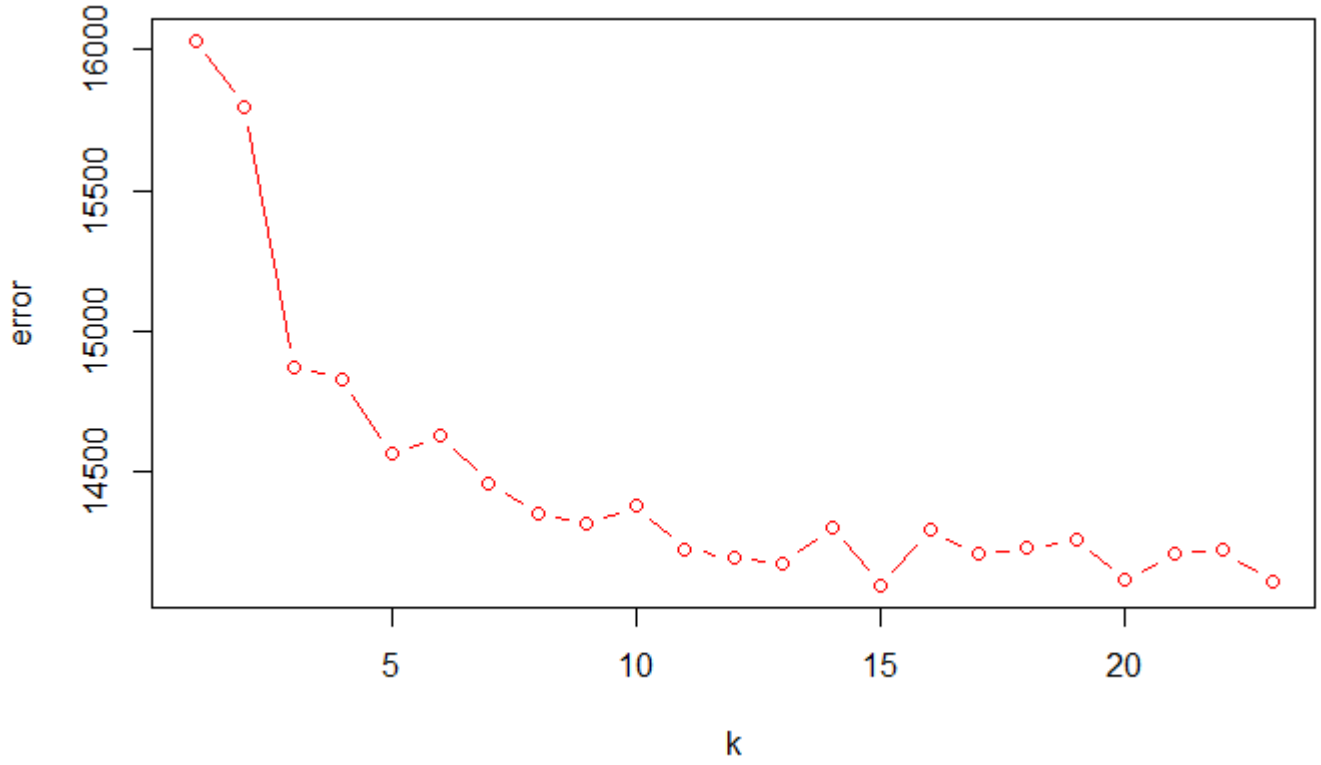
K - Nearest Neighbors on normalized data

Hide

```
RES = c()
smp_size_1 <- floor(0.1 * nrow(train))
smp_size_2 <- floor(0.1 * nrow(validation))
for (i in 1:23) {
  # set.seed(123)
  train_ind <- sample(seq_len(nrow(train.normalized)), size = smp_size_1)
  val_ind <- sample(seq_len(nrow(validation.normalized)), size = smp_size_2)
  knn.normalized.train <- train.normalized[train_ind,]
  knn.normalized.val <- validation.normalized[val_ind,]
  result = knn(train = knn.normalized.train[,2:6], test = knn.normalized.val[,2:6], cl = knn.nor
malized.train[,7], k=i)
  err = sum(abs(as.numeric(unlist(result)) - knn.normalized.val[,7]), na.rm = TRUE)
  RES = c(RES,err)
}
```

Hide

```
plot(RES, type = 'b', col = 'red', xlab = "k", ylab = "error")
```



Hide

```
knn.normalized.result = knn(train = train.normalized[,2:6], test = validation.normalized[,2:6],
 cl = train.normalized[,7], k=6, prob = TRUE)
knn.normalized.result.prob <- attributes(knn.normalized.result)$prob
```

Hide

```
knn.normalized.result.prob.treated = knn.normalized.result.prob
for (r in 1:173040) {
  t = knn.normalized.result.prob[r]
  if (knn.normalized.result[r] == 1) {
    knn.normalized.result.prob.treated[r] = 1 - t
  } else {
    knn.normalized.result.prob.treated[r] = t
  }
}
```

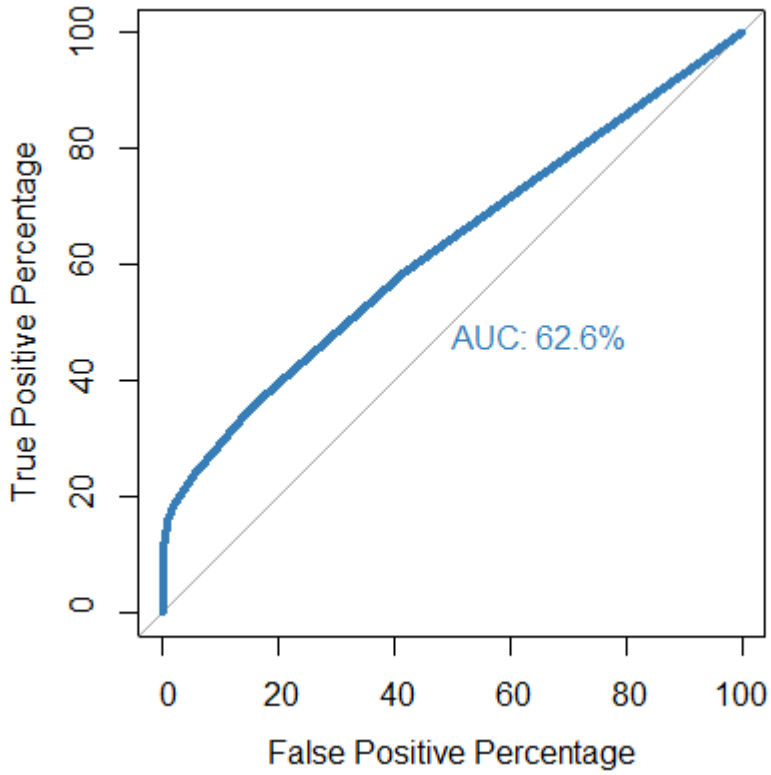
Hide

```
par(pty = 's')
roc(factor(validation.normalized$train_indessa.loan_status), knn.normalized.result.prob.treated,
plot = TRUE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "Tr
ue Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```
Call:
roc.default(response = factor(validation.normalized$train_indessa.loan_status),     predictor =
knn.normalized.result.prob.treated, percent = TRUE,     plot = TRUE, legacy.axes = TRUE, xlab =
"False Positive Percentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,
print.auc = TRUE)

Data: knn.normalized.result.prob.treated in 132036 controls (factor(validation.normalized$train_
indessa.loan_status) 0) > 41004 cases (factor(validation.normalized$train_indessa.loan_status)
1).
Area under the curve: 62.56%
```



Decision Trees on original data

Hide

```
dt.original.fit = rpart(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv + te
rm_num + train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim, data = train, method = "cla
ss", control = rpart.control(cp = 0.1))
```

Hide

```
dt.original.result = predict(dt.original.fit, validation[,2:6], type = "prob")
```

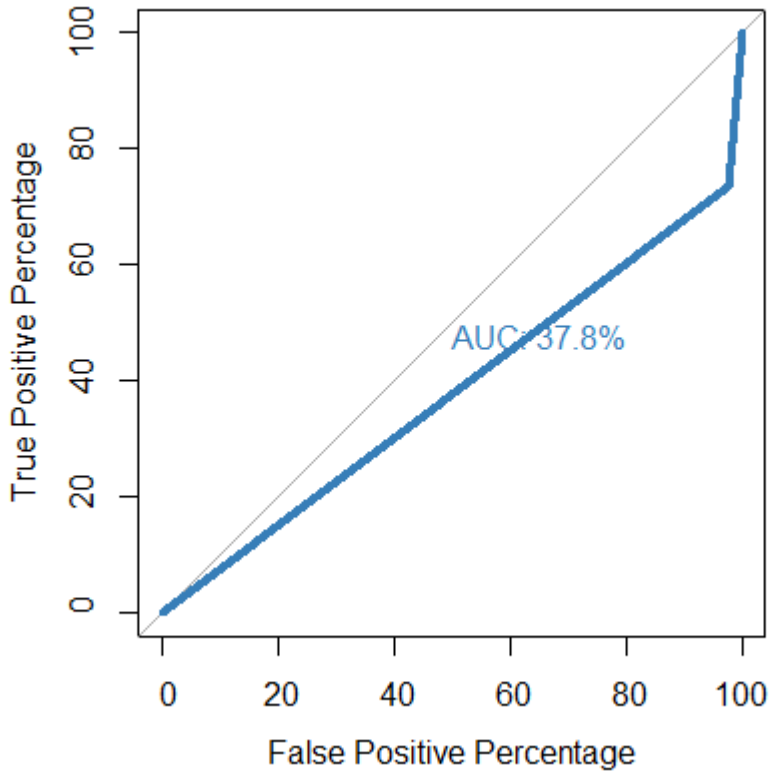
Hide

```
par(pty = 's')
roc(factor(validation$train_indessa.loan_status), dt.original.result[,1], plot = TRUE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = dt.original.result[, 1], percent = TRUE, plot = TRUE,     legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True Positive Percentage",     col = "#377eb8", lwd = 4, print.auc = TRUE)

Data: dt.original.result[, 1] in 132036 controls (factor(validation$train_indessa.loan_status) 0) < 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 37.85%
```



Decision Trees on normalized data

Hide

```
dt.normalized.fit = rpart(formula = train_indessa.loan_status ~ train_indessa.funded_amnt_inv + term_num + train_indessa.dti + l_w_p + train_indessa.total_rev_hi_lim, data = train.normalized, method = "class", control = rpart.control(cp = 0.1))
```

Hide

```
dt.normalized.result = predict(dt.normalized.fit, validation.normalized[,2:6], type = "prob")
```

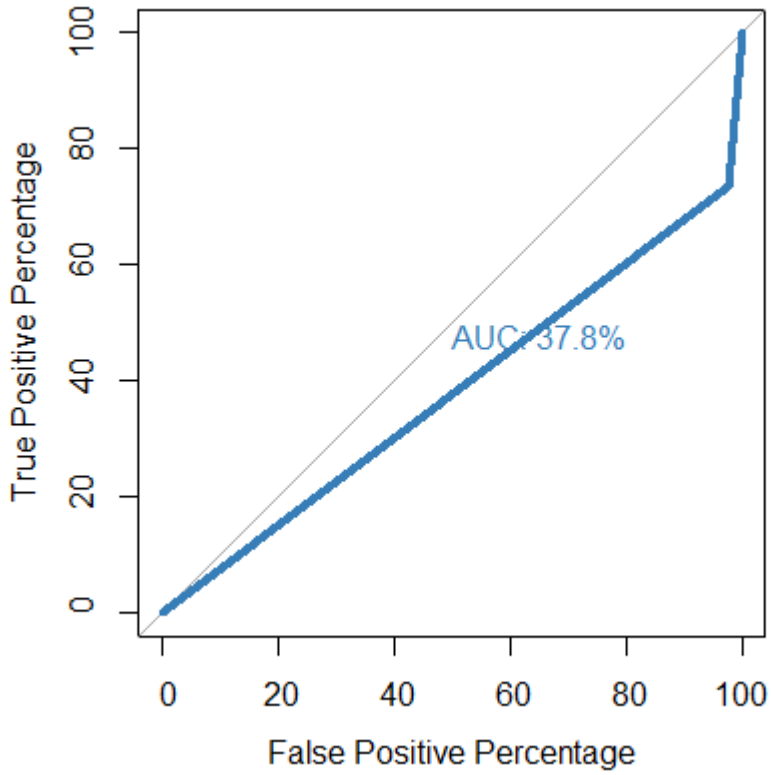
Hide

```
par(pty = 's')
roc(factor(validation.normalized$train_indessa.loan_status), dt.normalized.result[,1], plot = TR
UE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positi
ve Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
Call:
roc.default(response = factor(validation.normalized$train_indessa.loan_status),     predictor =
dt.normalized.result[, 1], percent = TRUE, plot = TRUE,     legacy.axes = TRUE, xlab = "False Po
sitive Percentage", ylab = "True Positive Percentage",     col = "#377eb8", lwd = 4, print.auc =
TRUE)

Data: dt.normalized.result[, 1] in 132036 controls (factor(validation.normalized$train_indessa.l
oan_status) 0) < 41004 cases (factor(validation.normalized$train_indessa.loan_status) 1).
Area under the curve: 37.85%
```



As we can see above, the best results from 1. Gaussian Naive Bayes 2. Logistic Regression 3. K - Nearest Neighbors 4. Decision Trees

on 1. Original 2. Normalized

data; are manifested during, 1. Gaussian Naive Bayes on Original data, AUC: 68.5 2. K - Nearest Neighbors on Original data, k = 5, AUC: 63.6 3. K - Nearest Neighbors on Normalized data, k = 6, AUC: 62.6

Next we will try to optimize for the results in 2 and 3, by changing the k values a little bit.

K - Nearest Neighbors on original data for k = 4

Hide

```
for (r in 1:173040) {
  t = knn.original.result.optimization_1.prob[r]
  if (knn.original.optimization_1.result[r] == 1) {
    knn.original.result.optimization_1.prob.treated[r] = 1 - t
  } else {
    knn.original.result.optimization_1.prob.treated[r] = t
  }
}
```

```
Error in knn.original.optimization_1.result :
  object 'knn.original.optimization_1.result' not found
```

Hide

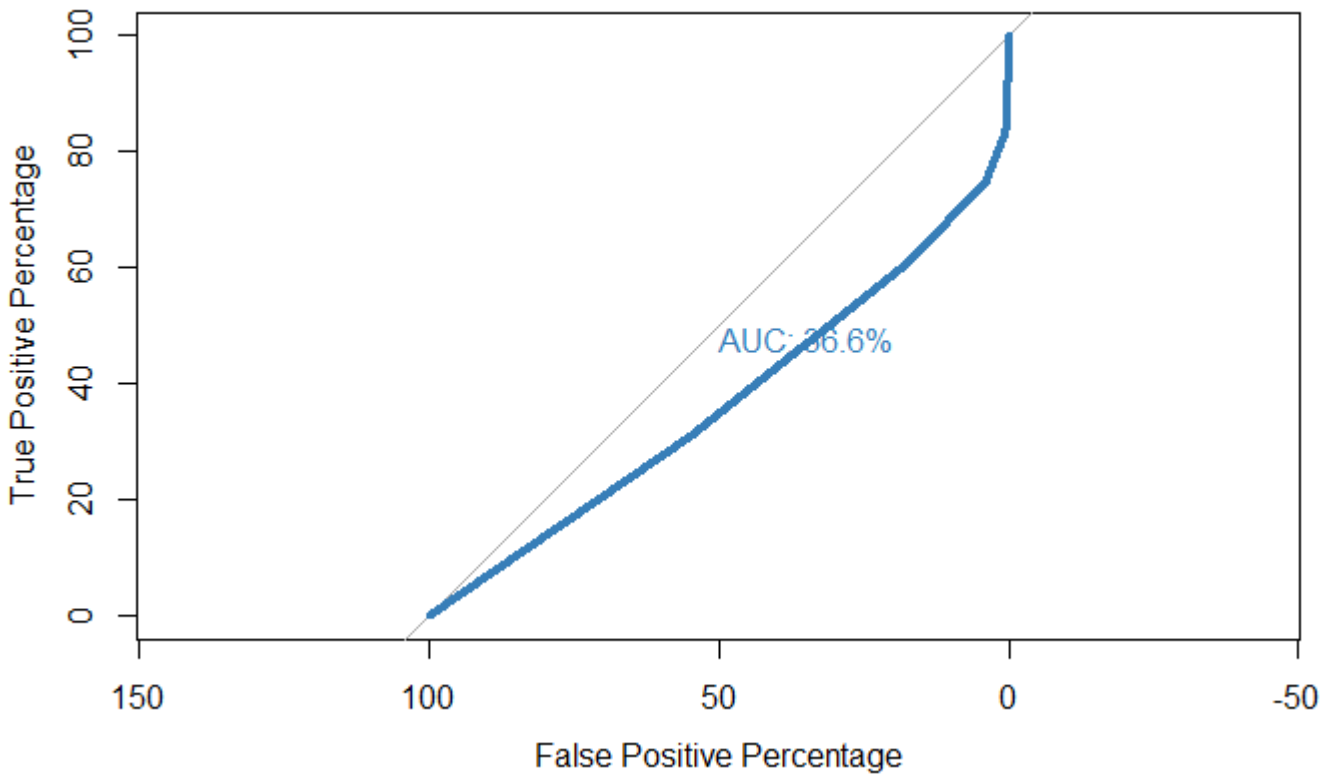
```
roc(factor(validation$train_indessa.loan_status), knn.original.result.optimization_1.prob.treate
d, plot = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Perce
ntage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```


Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = knn.origina
l.result.optimization_1.prob.treated,     percent = TRUE, plot = TRUE, xlab = "False Positive Pe
rcentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     print.auc = TRU
E)

Data: knn.original.result.optimization_1.prob.treated in 132036 controls (factor(validation$trai
n_indessa.loan_status) 0) < 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 36.59%
```



K - Nearest Neighbors on original data for k = 6

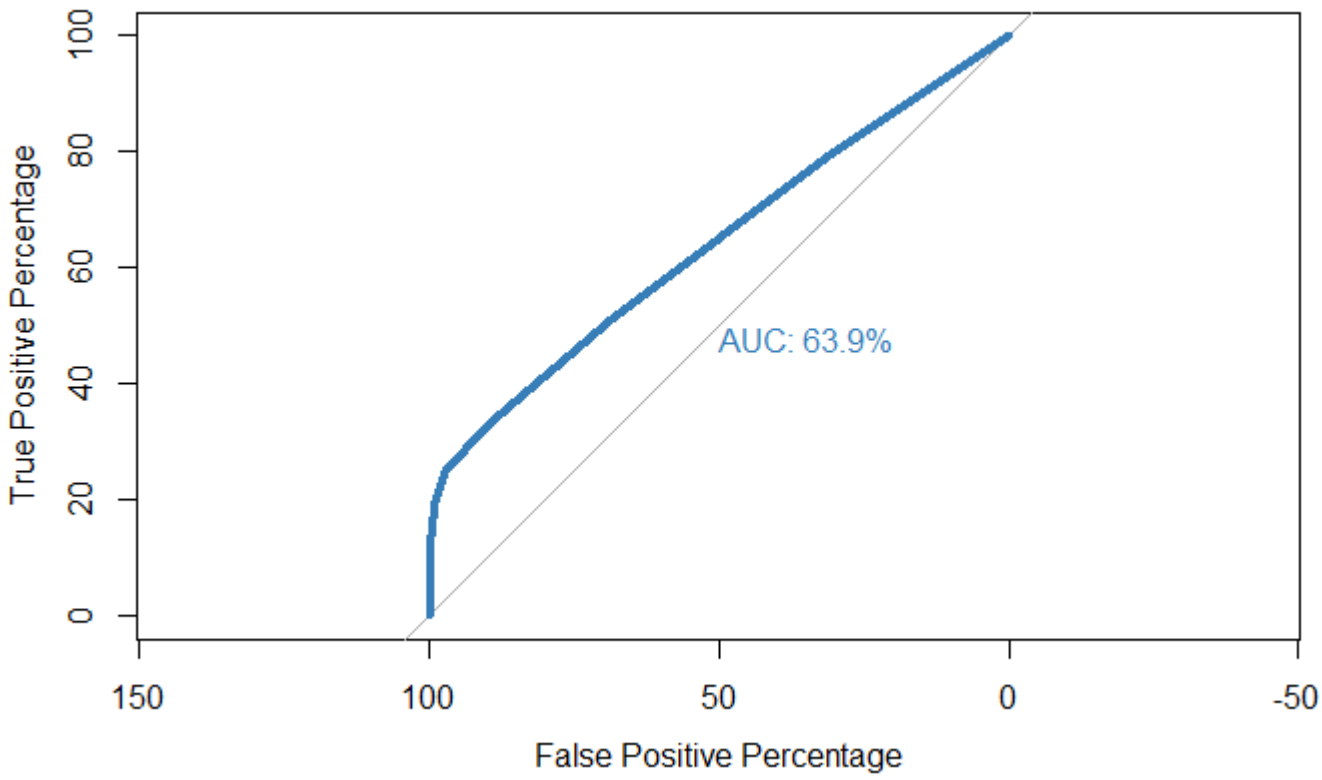
Hide

```
roc(factor(validation$train_indessa.loan_status), knn.original.result.optimization_2.prob.treate
d, plot = TRUE, percent = TRUE, xlab  = "False Positive Percentage", ylab = "True Positive Perce
ntage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```
Call:
roc.default(response = factor(validation$train_indessa.loan_status),     predictor = knn.origina
l.result.optimization_2.prob.treated,     percent = TRUE, plot = TRUE, xlab = "False Positive Pe
rcentage",     ylab = "True Positive Percentage", col = "#377eb8", lwd = 4,     print.auc = TRU
E)

Data: knn.original.result.optimization_2.prob.treated in 132036 controls (factor(validation$trai
n_indessa.loan_status) 0) > 41004 cases (factor(validation$train_indessa.loan_status) 1).
Area under the curve: 63.92%
```



K - Nearest Neighbors on normalized data for k = 5

Hide

```
knn.normalized.result.opimization_1 = knn(train = train.normalized[,2:6], test = validation.norm
alized[,2:6], cl = train.normalized[,7], k=6, prob = TRUE)
```

```
NA
```

```
NA
```

```
NA
```

```
NANANANANANANANANANANANANANANANANANANANANA
```

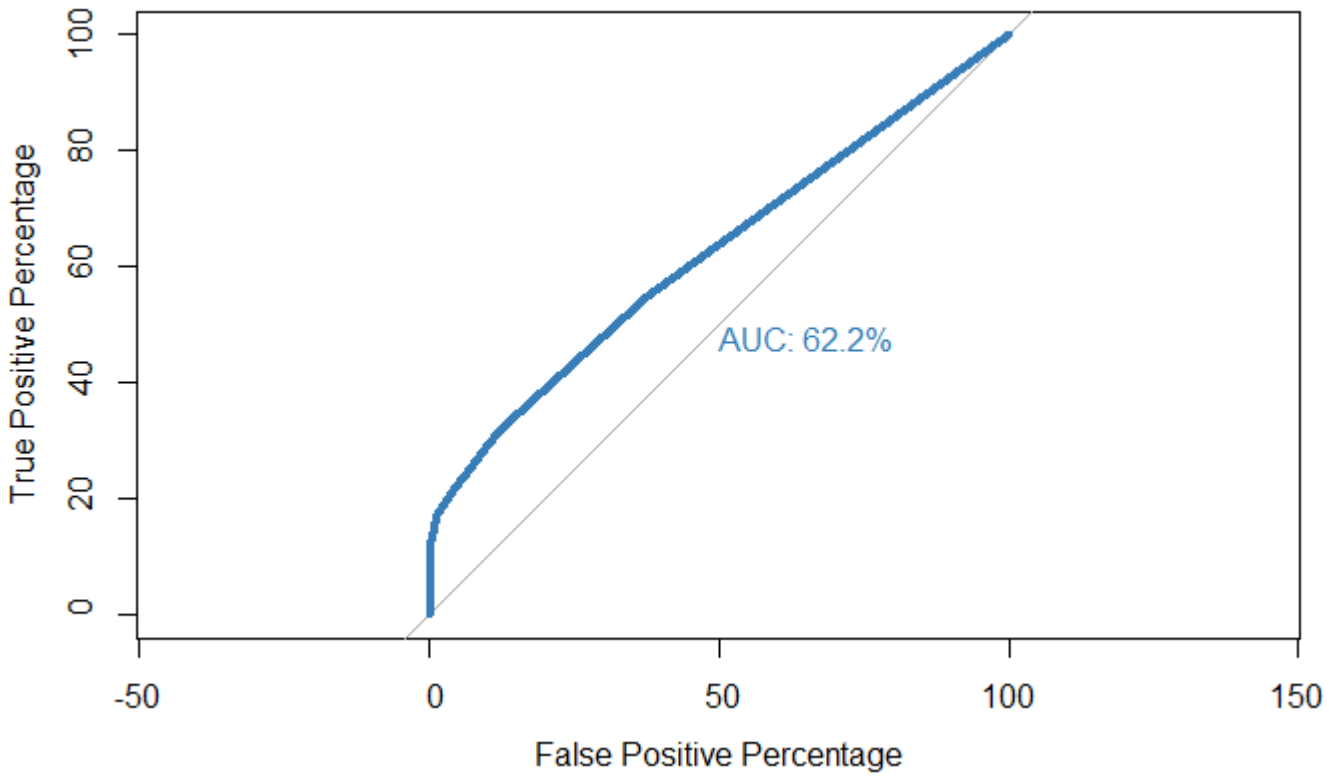
Hide

```
roc(factor(validation.normalized$train_indessa.loan_status), knn.normalized.result.opimization_
1.prob.treated, plot = TRUE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percent
age", ylab = "True Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```
Call:
roc.default(response = factor(validation.normalized$train_indessa.loan_status),     predictor =
knn.normalized.result.opimization_1.prob.treated,     percent = TRUE, plot = TRUE, legacy.axes =
TRUE, xlab = "False Positive Percentage",     ylab = "True Positive Percentage", col = "#377eb
8", lwd = 4,     print.auc = TRUE)

Data: knn.normalized.result.opimization_1.prob.treated in 132036 controls (factor(validation.nor
malized$train_indessa.loan_status) 0) > 41004 cases (factor(validation.normalized$train_indessa.
loan_status) 1).
Area under the curve: 62.22%
```



K - Nearest Neighbors on normalized data for k = 7

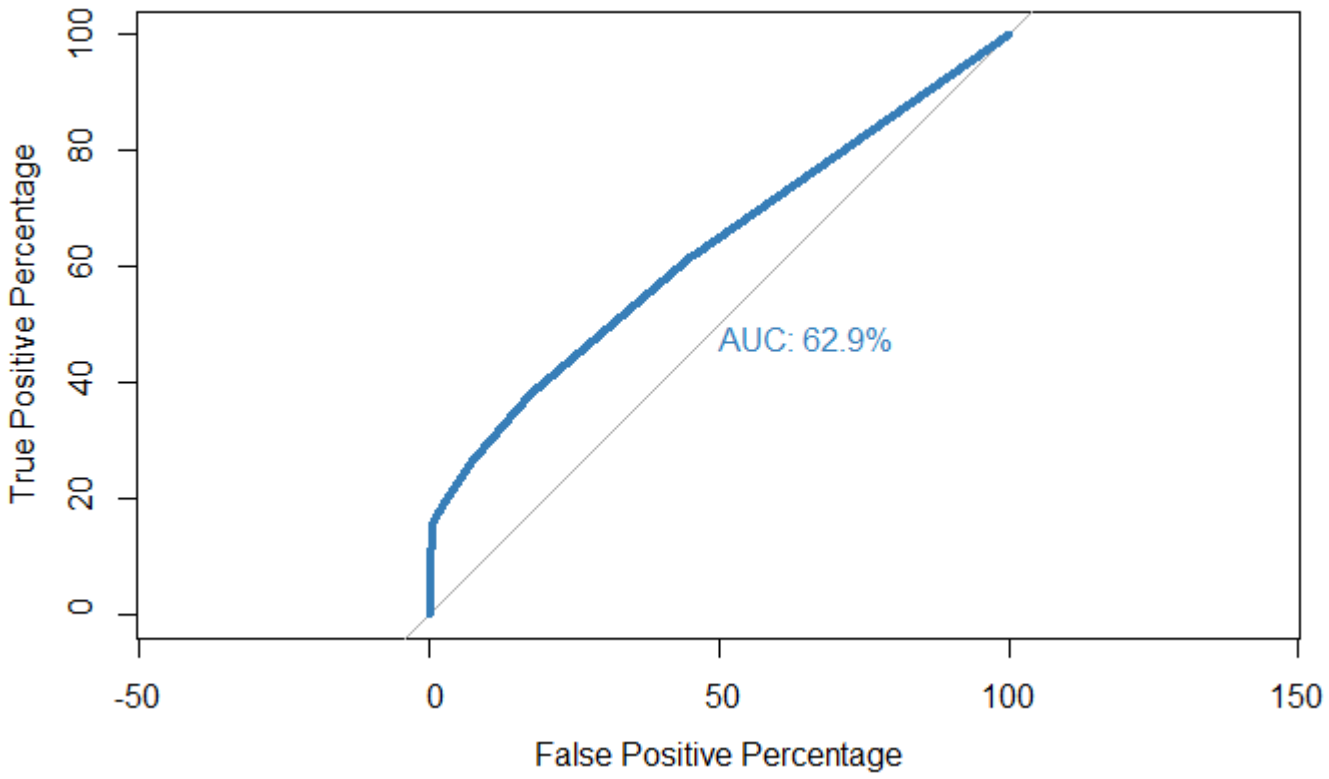
Hide

```
roc(factor(validation.normalized$train_indessa.loan_status), knn.normalized.result.opimization_
2.prob.treated, plot = TRUE, legacy.axes = TRUE, percent = TRUE, xlab  = "False Positive Percent
age", ylab = "True Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls > cases
```

```
Call:
roc.default(response = factor(validation.normalized$train_indessa.loan_status),     predictor =
knn.normalized.result.opimization_2.prob.treated,     percent = TRUE, plot = TRUE, legacy.axes =
TRUE, xlab = "False Positive Percentage",     ylab = "True Positive Percentage", col = "#377eb
8", lwd = 4,     print.auc = TRUE)

Data: knn.normalized.result.opimization_2.prob.treated in 132036 controls (factor(validation.nor
malized$train_indessa.loan_status) 0) > 41004 cases (factor(validation.normalized$train_indessa.
loan_status) 1).
Area under the curve: 62.94%
```



The winner is still Gaussian Naive Bayes on original data, AUC: 68.5.