# Project I: SEMMA with Regularized Logistic Regression

Appiah Prince* University of Texas at El Paso (UTEP)

September 06, 2022

## Contents

*pappiah@miners.utep.edu

# 1 Bring in the data

```
diabetes <- read.csv("diabetes_data_upload.csv")
dim(diabetes)
```

```
## [1] 520  17
```

```
names(diabetes)
```

```
##  [1] "Age"                "Gender"             "Polyuria"
##  [4] "Polydipsia"         "sudden.weight.loss" "weakness"
##  [7] "Polyphagia"         "Genital.thrush"     "visual.blurring"
## [10] "Itching"            "Irritability"       "delayed.healing"
## [13] "partial.paresis"    "muscle.stiffness"   "Alopecia"
## [16] "Obesity"            "class"
```

```
head(diabetes)
```

```
##   Age Gender Polyuria Polydipsia sudden.weight.loss weakness Polyphagia
## 1  40   Male       No        Yes                 No      Yes         No
## 2  58   Male       No         No                 No      Yes         No
## 3  41   Male      Yes         No                 No      Yes        Yes
## 4  45   Male       No         No                Yes      Yes        Yes
## 5  60   Male      Yes        Yes                Yes      Yes        Yes
## 6  55   Male      Yes        Yes                 No      Yes        Yes
##   Genital.thrush visual.blurring Itching Irritability delayed.healing
## 1             No              No     Yes           No             Yes
## 2             No             Yes      No           No              No
## 3             No              No     Yes           No             Yes
## 4            Yes              No     Yes           No             Yes
## 5             No             Yes     Yes          Yes             Yes
## 6             No             Yes     Yes           No             Yes
##   partial.paresis muscle.stiffness Alopecia Obesity    class
## 1              No              Yes      Yes     Yes Positive
## 2             Yes               No      Yes      No Positive
## 3              No              Yes      Yes      No Positive
## 4              No               No       No      No Positive
## 5             Yes              Yes      Yes     Yes Positive
## 6              No              Yes      Yes     Yes Positive
```

REMARKS

- There are **520** observations and **17** variables.

## 2 Exploratory Data Analysis(EDA)

```
str(diabetes) # checking for variable types
```

```
## 'data.frame':    520 obs. of  17 variables:
##  $ Age               : int  40 58 41 45 60 55 57 66 67 70 ...
##  $ Gender            : chr  "Male" "Male" "Male" "Male" ...
##  $ Polyuria          : chr  "No" "No" "Yes" "No" ...
##  $ Polydipsia        : chr  "Yes" "No" "No" "No" ...
##  $ sudden.weight.loss: chr  "No" "No" "No" "Yes" ...
##  $ weakness          : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Polyphagia        : chr  "No" "No" "Yes" "Yes" ...
##  $ Genital.thrush    : chr  "No" "No" "No" "Yes" ...
##  $ visual.blurring   : chr  "No" "Yes" "No" "No" ...
##  $ Itching           : chr  "Yes" "No" "Yes" "Yes" ...
##  $ Irritability      : chr  "No" "No" "No" "No" ...
##  $ delayed.healing   : chr  "Yes" "No" "Yes" "Yes" ...
##  $ partial.paresis   : chr  "No" "Yes" "No" "No" ...
##  $ muscle.stiffness  : chr  "Yes" "No" "Yes" "No" ...
##  $ Alopecia          : chr  "Yes" "Yes" "Yes" "No" ...
##  $ Obesity           : chr  "Yes" "No" "No" "No" ...
##  $ class             : chr  "Positive" "Positive" "Positive" "Positive" ...
```

Remarks

- Age is numeric variable while the remaining **16** variables are character variables.

```
# INSPECT THE DISTINCT VALUES OF EACH X
cols <- 1:NCOL(diabetes)
for (j in cols){
  x <- diabetes[,j]
  print(names(diabetes)[j])
  print(sort(unique(x, incomparables=TRUE)))
  print(table(x, useNA="ifany"))
}
```

```
## [1] "Age"
##  [1] 16 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
## [26] 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 72 79 85
## [51] 90
## x
## 16 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
##  1  2  1  6  9  1 25  3  5  4  6 30  8  7 20 16 24  4  9 25  7 18  8 21 28  7
## 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 72 79 85 90
## 18  5  4 20 16 22  8 15 18  4 15  8  7  3  5  6  9  8 10  5  5  9  1  2  2
## [1] "Gender"
## [1] "Female" "Male"
## x
## Female   Male
##    192    328
## [1] "Polyuria"
## [1] "No"  "Yes"
## x
##  No Yes
## 262 258
## [1] "Polydipsia"
## [1] "No"  "Yes"
## x
##  No Yes
## 287 233
## [1] "sudden.weight.loss"
## [1] "No"  "Yes"
## x
##  No Yes
## 303 217
## [1] "weakness"
## [1] "No"  "Yes"
## x
##  No Yes
## 215 305
## [1] "Polyphagia"
## [1] "No"  "Yes"
## x
##  No Yes
## 283 237
## [1] "Genital.thrush"
## [1] "No"  "Yes"
## x
##  No Yes
## 404 116
## [1] "visual.blurring"
## [1] "No"  "Yes"
## x
##  No Yes
## 287 233
## [1] "Itching"
## [1] "No"  "Yes"
## x
##  No Yes
## 267 253
## [1] "Irritability"
## [1] "No"  "Yes"
## x
##  No Yes
## 394 126
## [1] "delayed.healing"
## [1] "No"  "Yes"
## x
##  No Yes
## 281 239
## [1] "partial.paresis"
## [1] "No"  "Yes"
## x
##  No Yes
## 296 224
## [1] "muscle.stiffness"
## [1] "No"  "Yes"
## x
##  No Yes
## 325 195
## [1] "Alopecia"
## [1] "No"  "Yes"
## x
##  No Yes
## 341 179
## [1] "Obesity"
## [1] "No"  "Yes"
## x
##  No Yes
## 432  88
## [1] "class"
## [1] "Negative" "Positive"
## x
## Negative Positive
##      200      320
```

## 2.1 Frequency Distribution of the target variable class

```
t <- table(diabetes$class, useNA="ifany")
freq_dist <- as.data.frame(t)
colnames(freq_dist) <- c("class", "frequency")
freq_dist
```

```
##      class frequency
## 1 Negative       200
## 2 Positive       320
```

Remarks

- There are **320** patients that their diabetes diagnosis is positive while **200** patients are diagnose negative. So, there is an unequal distribution of the results of the diagnosis. Hence, we have a slightly unbalanced classification problem.

## 2.2 Missing Values

```
library(questionr)
freq.na(diabetes)
```

```
##                    missing %
## Age                      0 0
## Gender                   0 0
## Polyuria                 0 0
## Polydipsia               0 0
## sudden.weight.loss       0 0
## weakness                 0 0
## Polyphagia               0 0
## Genital.thrush           0 0
## visual.blurring          0 0
## Itching                  0 0
## Irritability             0 0
## delayed.healing          0 0
## partial.paresis          0 0
## muscle.stiffness         0 0
## Alopecia                 0 0
## Obesity                  0 0
## class                    0 0
```

Remarks

There are no missing values in the dataset.

```
# Assigning 0 for Negative class and 1 for Positive class
diabetes$class <- ifelse(diabetes$class=="Negative", 0,1)
```

# 3 Variable Screening

```
# Two sample t-test
cond.1 <- diabetes$class == 1
cond.2 <- as.vector(which(sapply(diabetes[,-c(17)], is.numeric), arr.ind = T))
print("Test of Normality of the numerical variables for patients diagnosed
      diabetes postive")
```

```
## [1] "Test of Normality of the numerical variables for patients diagnosed \n      diab
```

```
shapiro.test(diabetes[cond.1, cond.2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  diabetes[cond.1, cond.2]
## W = 0.9804, p-value = 0.0002325
```

```
print("Test of Normality of the numerical variables for patients diagnosed
      diabetes Negative")
```

```
## [1] "Test of Normality of the numerical variables for patients diagnosed \n      diab
```

```
shapiro.test(diabetes[!cond.1, cond.2])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  diabetes[!cond.1, cond.2]
## W = 0.96687, p-value = 0.0001182
```

Remarks

- For the numerical variables, we first use Shapiro-Wilk test to check the assumption of normality so as to know whether to use parametric or nonparametric approach for the two sample t-test.We see from the output of the Shapiro-Wilk normality test that the assumption of normality is violated since the p-values are less than 0.05 in each group.Thus, we use the Wilcoxon rank-sum test.

## 3.1 Chisq test and Wilcoxon test

```
suppressPackageStartupMessages(library(car))
vars.nominal <- c("Gender","Polyuria","Polydipsia","sudden.weight.loss",
                  "weakness","Polyphagia","Genital.thrush","visual.blurring",
                  "Itching","Irritability","delayed.healing","partial.paresis",
                  "muscle.stiffness","Alopecia","Obesity")
cols.x <- 1:(NCOL(diabetes)-1)
xnames <- names(diabetes)[cols.x]
y <- diabetes$class
OUT <- NULL
for (j in 1:length(cols.x)){
  x <- diabetes[, cols.x[j]]
  xname <- xnames[j]
  if (is.element(xname, vars.nominal)){
    tbl <- table(x, y)
    pvalue <- chisq.test(tbl)$p.value
  } else {
    # WILCOXON TEST
    pvalue <- wilcox.test(x~y, alternative="two.sided")$p.value
  }
  OUT <- rbind(OUT, cbind(xname=xname, pvalue=pvalue))
}
OUT <- as.data.frame(OUT, stringsAsFactors =F)
colnames(OUT) <- c("name", "pvalue")
OUT
```

```
##                   name               pvalue
## 1                  Age     0.01240447825802
## 2               Gender 3.28970373055333e-24
## 3             Polyuria 1.74091178034421e-51
## 4           Polydipsia 6.18700964088628e-49
## 5   sudden.weight.loss 5.96916626254991e-23
## 6             weakness 4.86984344658554e-08
## 7           Polyphagia 1.16515843464091e-14
## 8       Genital.thrush   0.0160979029919381
## 9      visual.blurring 1.70150367532412e-08
## 10            Itching    0.829748395948501
## 11       Irritability 1.77148314939594e-11
## 12    delayed.healing    0.326659937714402
## 13    partial.paresis 1.56528907105633e-22
## 14   muscle.stiffness  0.00693909569792398
## 15           Alopecia 1.90927949636339e-09
## 16            Obesity    0.127107993198967
```

Remarks

- The predictors variables **itching**, **delayed.healing** and **obesity** have relatively higher p-values as compared to the other predictor variables.

## 3.2 Non Significant Variables

```
cond.3 <- as.numeric(OUT$pvalue) > 0.25
OUT[cond.3, ]
```

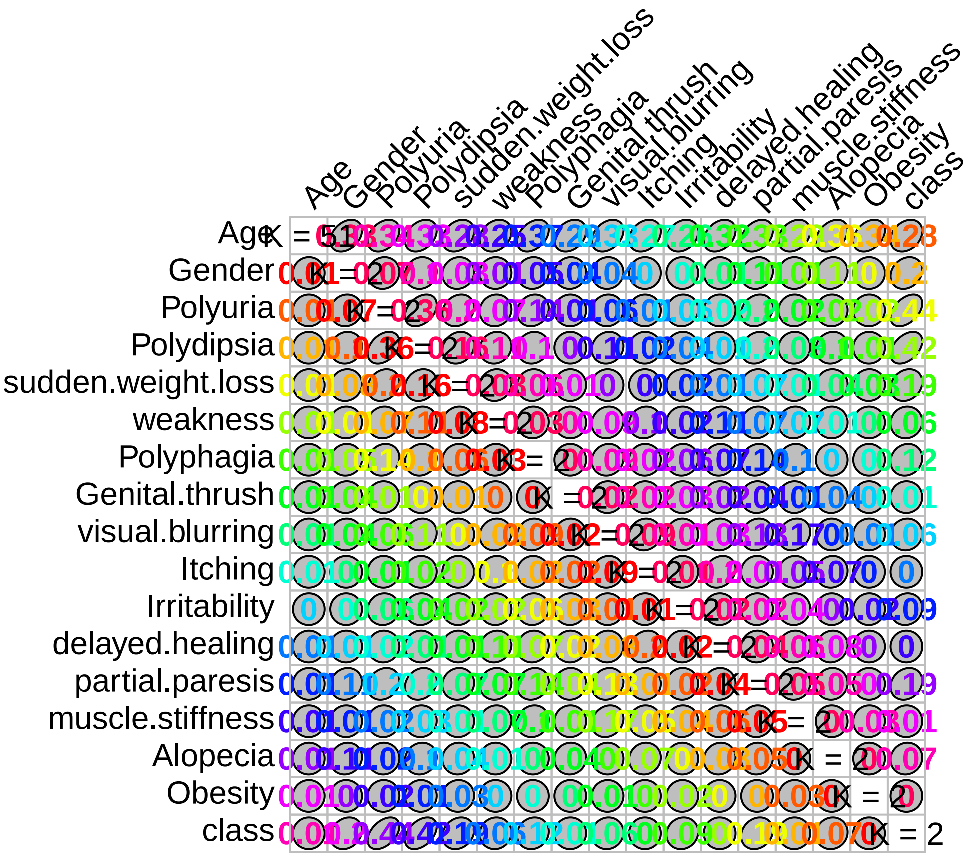
```
##               name            pvalue
## 10         Itching 0.829748395948501
## 12 delayed.healing 0.326659937714402
```

Remarks

- The predictor variables **Itching** and **delayed.healing** are unimportant predictors given the liberal threshold significance level of **0.25**. Therefore, we remove the predictor variables **Itching** and **delayed.healing** from the data.

## 3.3 Correlation plot among the variables

```
library(GoodmanKruskal)
data <- GKtauDataframe(diabetes)
plot(data, corColors = "magenta")
```



Remarks

- We observe that there is no high correlation among the variables that is no high multicollinearity.

## 3.4 Removing non significant variables

```
diabetes <- diabetes[, -c(10, 12)]
names(diabetes)
```

```
##  [1] "Age"                "Gender"             "Polyuria"
##  [4] "Polydipsia"         "sudden.weight.loss" "weakness"
##  [7] "Polyphagia"         "Genital.thrush"     "visual.blurring"
## [10] "Irritability"       "partial.paresis"    "muscle.stiffness"
## [13] "Alopecia"           "Obesity"            "class"
```

# 4 Data Partition

```
set.seed(123)
n <- NROW(diabetes)
ratio <- 2/3
id.training <- sample(1:n, size=n*ratio, replace=FALSE)
D1 <- diabetes[id.training, ]  # training data
D2 <- diabetes[-id.training, ]  # test data
dim(D1)
```

```
## [1] 346  15
```

```
dim(D2)
```

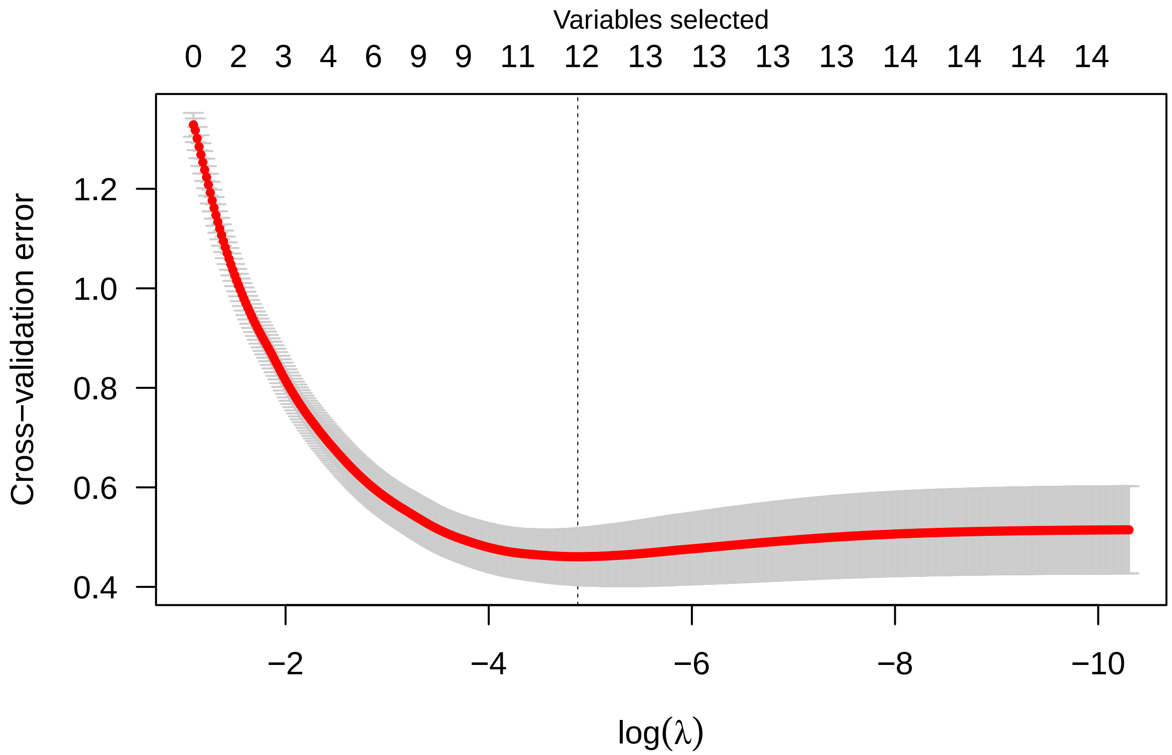
```
## [1] 174  15
```

Remarks

- The training data has 346 observations and 15 variables
- The test data has 174 obseervations and 15 variables

# 5 Logistic Regression Modeling

```
set.seed(123)
library(ncvreg);
y <- D1$class
formula0<- class ~.
X <- model.matrix(as.formula(formula0), data=D1)
cvfit.lasso <- cv.ncvreg(X=X,y=y, nfolds=5, family="binomial",
            penalty="lasso",lambda.min=.0001, nlambda=500,eps=.01,
            max.iter=1000)
plot(cvfit.lasso)
```



Remarks

- The graph shows that 12 variables must be selected as important predictor variables.

## 5.1 Selecting the best tuning parameter

```
cvfit.lasso$lambda.min
```

```
## [1] 0.007622463
```

Remarks

- We used the minimum cross-validation error as a criteria for selecting best tuning parameter.

## 5.2 Important Predictor Variables

```
result.lasso <- cvfit.lasso$fit
beta.hat <- as.vector(result.lasso$beta[-1, cvfit.lasso$min])
cutoff <- 0
terms <- colnames(X)[abs(beta.hat) > cutoff]
terms
```

```
##  [1] "Age"                 "GenderMale"          "PolyuriaYes"
##  [4] "PolydipsiaYes"       "sudden.weight.lossYes" "PolyphagiaYes"
##  [7] "Genital.thrushYes"   "IrritabilityYes"     "partial.paresisYes"
## [10] "muscle.stiffnessYes" "AlopeciaYes"         "ObesityYes"
```

## 5.3 Final Best Model Fit

```
formula01 <- class ~ Age + Gender + Polyuria + Polydipsia +
          sudden.weight.loss + Polyphagia + Genital.thrush + Irritability +

formula.lasso <- as.formula(formula01)
fit.lasso <- glm(formula.lasso, data = D1, family="binomial")
summary(fit.lasso)
```

```
##
## Call:
## glm(formula = formula.lasso, family = "binomial", data = D1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.63128  -0.23995   0.01083   0.07846   3.00708
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.69441    1.13030   1.499  0.13385
## Age                  -0.04201    0.02727  -1.540  0.12345
## GenderMale           -3.63158    0.66055  -5.498 3.85e-08 ***
## PolyuriaYes           3.30744    0.66914   4.943 7.70e-07 ***
## PolydipsiaYes         3.38383    0.81488   4.153 3.29e-05 ***
## sudden.weight.lossYes  0.89416    0.56036   1.596  0.11056
## PolyphagiaYes         1.49313    0.64072   2.330  0.01979 *
## Genital.thrushYes     1.64468    0.61436   2.677  0.00743 **
## IrritabilityYes       2.86832    0.71811   3.994 6.49e-05 ***
## partial.paresisYes    1.41855    0.58791   2.413  0.01583 *
## muscle.stiffnessYes  -0.50038    0.64662  -0.774  0.43902
## AlopeciaYes          -1.01894    0.61856  -1.647  0.09950 .
## ObesityYes            -0.93395    0.63763  -1.465  0.14300
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 461.92  on 345  degrees of freedom
## Residual deviance: 119.18  on 333  degrees of freedom
## AIC: 145.18
##
## Number of Fisher Scoring iterations: 8
```

Remarks

- The AIC for the final model is somehow smaller which is good
- Most of the predictor variables are statistically significant considering their p-values.

# 6 Model Assessment/Deployment

## 6.1 Applying the final logistic model to the test data D2

```
yobs <- D2$class
phat <- predict(fit.lasso, newdata=D2, type="response")
cutoff <- 0.5
yhat <- (phat <= cutoff) + 0
table(yobs, yhat)
```

```
##     yhat
## yobs  0  1
##    0  7 59
##    1 98 10
```

## 6.2 ROC CURVE AND AUC

```
suppressPackageStartupMessages(library(verification))
a.ROC <- roc.area(obs=yobs, pred=phat)$A
print(a.ROC)
```

```
## [1] 0.9588945
```

```
suppressPackageStartupMessages(library(cvAUC))
AUC <- ci.cvAUC(predictions=phat, labels=yobs, folds=1:NROW(D2), confidence=0.95); AUC
```

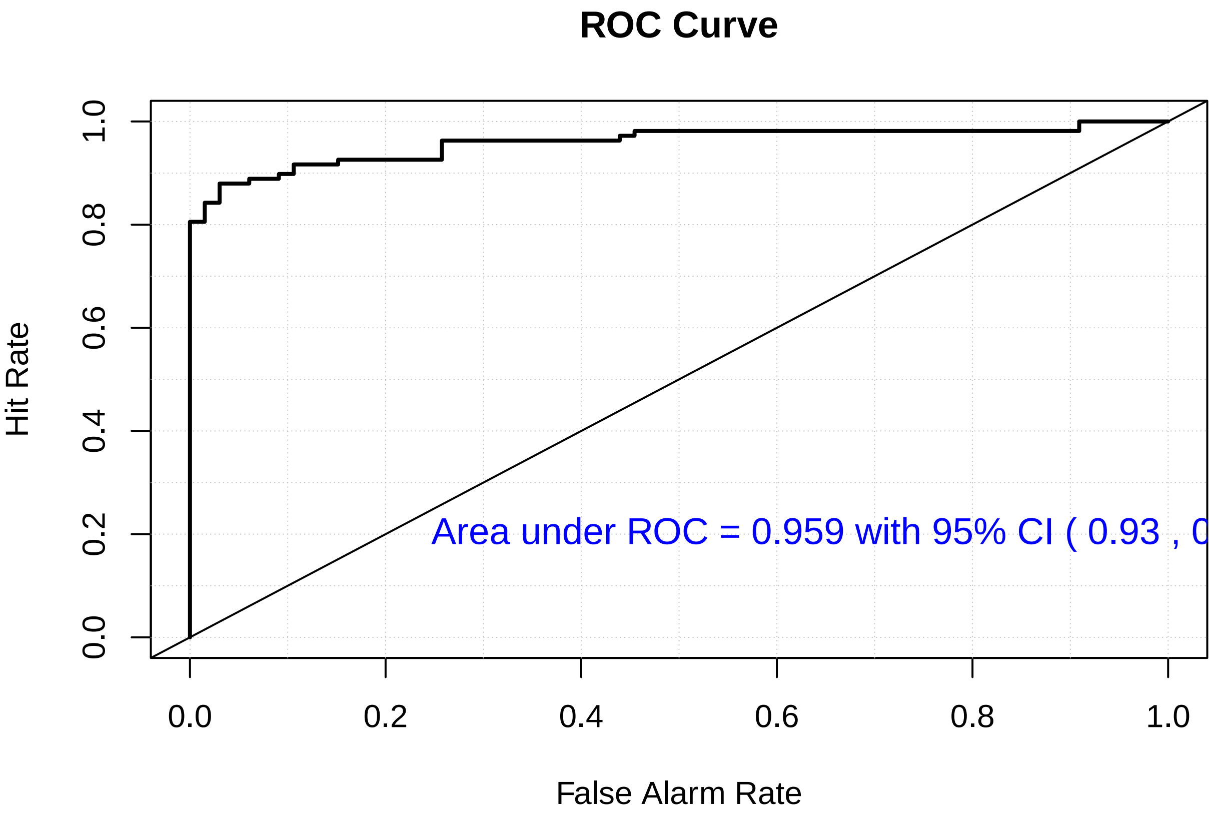
```
## $cvAUC
## [1] 0.9588945
##
## $se
## [1] 0.01455277
##
## $ci
## [1] 0.9303716 0.9874174
##
## $confidence
## [1] 0.95
```

```
auc.ci <- round(AUC$ci, digits=3)

suppressPackageStartupMessages(library(verification))
mod.glm <- verify(obs=yobs, pred=phat)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs
```

```
roc.plot(mod.glm, plot.thres = NULL)
text(x=0.7, y=0.2, paste("Area under ROC =", round(AUC$cvAUC, digits=3),
    "with 95% CI (", auc.ci[1], ",", auc.ci[2], ").",
    sep=" "), col="blue", cex=1.2)
```



Remarks

- The area under the ROC curve is **0.959** and its confidence interval is (0.930, 0.987)