---

title: “Project I” author: “Isaiah Thompson Ocansey” date: “09/06/2022” output: pdf_document: latex_engine: xelatex

---

SEMMA WITH REGULARIZED LOGISTIC REGRESSION

1) Bring the data into R (or Python).

```
df<-read.csv("diabetes_data.csv")
head(df); dim(df)
```

```
  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
```

```
[1] 520  17
```

The diabetes data has 520 observations and 17 variables

2) (EDA) Explore the data with EDA (Exploratory Data Analysis) by inspecting the variable types, outlying and possibly wrong records, and other issues. In particular, • inspect the frequency distribution of the target variable class and see, e.g., whether we have an unbalanced classification problem. • Are there missing values? If so, handle them with an appropriate strategy such as listwise deletion or single/multiple imputation.

VARIABLE TYPES

```
str(df)
```

```
'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" ...
```

From the above output, we observe that the variable age is continuous whereas the other variables are categorical

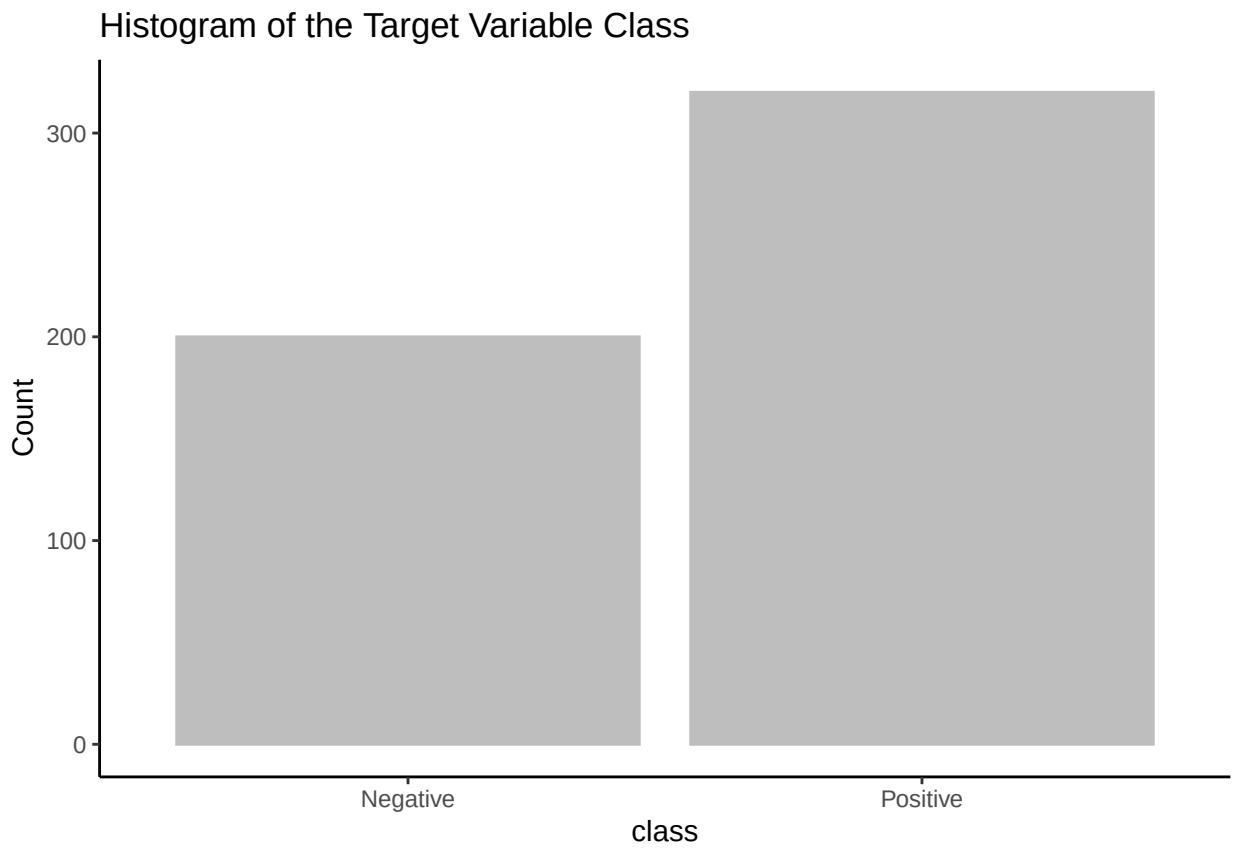
FREQUENCY DISTRIBUTION OF THE TARGET VARIABLE CLASS

```
library(questionr)
freq(df$class, total=T)
```

```
           n     %  val%
Negative 200  38.5  38.5
Positive 320  61.5  61.5
Total    520 100.0 100.0
```

From the above frequency table, we can observe that there are a total of 200 Negative class and 320 positive class which is not very unbalanced

```
library(ggplot2)
ggplot(df, aes(class)) +
  geom_bar(color = "gray", fill = "gray") +
  labs(
    title = "Histogram of the Target Variable Class",

    x = "class",
    y = "Count"
  ) +
  theme_classic()
```



We confirm the fact that the classification is not very unbalanced by the bar plot above

HANDLING MISSING VALUES in X

```
cols <- 1:NCOL(df)
for (j in cols){
  x <- df[,j]
  print(sort(unique(x, incomparables=TRUE)))
  print(table(x, useNA="ifany"))
}
```

```
 [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] "Female" "Male"
x
Female   Male
   192    328
[1] "No"  "Yes"
x
 No Yes
```

```
262 258
[1] "No"  "Yes"
x
 No Yes
287 233
[1] "No"  "Yes"
x
 No Yes
303 217
[1] "No"  "Yes"
x
 No Yes
215 305
[1] "No"  "Yes"
x
 No Yes
283 237
[1] "No"  "Yes"
x
 No Yes
404 116
[1] "No"  "Yes"
x
 No Yes
287 233
[1] "No"  "Yes"
x
 No Yes
267 253
[1] "No"  "Yes"
x
 No Yes
394 126
[1] "No"  "Yes"
x
 No Yes
281 239
[1] "No"  "Yes"
x
 No Yes
296 224
[1] "No"  "Yes"
x
 No Yes
325 195
[1] "No"  "Yes"
x
 No Yes
341 179
[1] "No"  "Yes"
x
 No Yes
432  88
[1] "Negative" "Positive"
```

```
x
Negative Positive
     200      320
```

Before handling missing data, We first explore the distinct values of the variables. And from the above, we don't see any missing values. We will move ahead to check the missing percentages of each variable to further ascertain there are no missing values in the dataset

```
colMeans(is.na(df))
```

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

From the above output, there appear to be no missing values in the diabetes data set

3) (Variable Screening) Explore the marginal (bivariate) associations between class and each attribute/predictor. The involved tools depend on the type of the attribute: • For a continuous predictor, use the parametric two-sample t test or the nonparametric Wilcoxon rank-sum test. • For a categorical predictor, use the 2 test of independence or Fisher's exact test in case of small cell counts.

TWO SAMPLE T-TEST BETWEEN CLASS AND AGE

```
t.test(Age~class,data=df)
```

```
    Welch Two Sample t-test

data:  Age by class
t = -2.489, df = 422.96, p-value = 0.01319
alternative hypothesis: true difference in means between group Negative and group Positive is not equal
95 percent confidence interval:
 -4.8534571 -0.5702929
sample estimates:
mean in group Negative mean in group Positive
              46.36000               49.07187
```

The continuous variable is Age and after performing the two sample t-test for Age and class, we observe that the p-value is less than 0.05, thus, we conclude that there is a difference between the means of class and Age,

CHI SQUARE TEST FOR ALL CATEGORICAL VARIABLES WITH TARGET VARIABLE CLASS

```
library(broom)
library(data.table)
CHIS <- lapply(df[,-c(1,17)], function(x) chisq.test(df[,1], x))
rbindlist(lapply(CHIS, tidy), idcol=TRUE)
```

```
                  .id statistic      p.value parameter
 1:             Gender  173.6745 1.449643e-15        50
 2:           Polyuria  178.6154 2.378889e-16        50
 3:         Polydipsia  171.2204 3.532858e-15        50
 4: sudden.weight.loss  146.1385 2.356206e-11        50
 5:           weakness  127.9319 9.294788e-09        50
 6:         Polyphagia  193.1639 1.051858e-18        50
 7:     Genital.thrush  153.0678 2.194750e-12        50
 8:    visual.blurring  173.4857 1.552730e-15        50
 9:            Itching  141.9530 9.641724e-11        50
10:       Irritability  133.0056 1.831105e-09        50
11:    delayed.healing  166.6576 1.827467e-14        50
12:    partial.paresis  170.7882 4.130763e-15        50
13:   muscle.stiffness  149.0867 8.634681e-12        50
14:           Alopecia  186.9223 1.095458e-17        50
15:            Obesity  176.8068 4.619429e-16        50
                        method
 1: Pearson's Chi-squared test
 2: Pearson's Chi-squared test
 3: Pearson's Chi-squared test
 4: Pearson's Chi-squared test
 5: Pearson's Chi-squared test
 6: Pearson's Chi-squared test
 7: Pearson's Chi-squared test
 8: Pearson's Chi-squared test
 9: Pearson's Chi-squared test
10: Pearson's Chi-squared test
11: Pearson's Chi-squared test
12: Pearson's Chi-squared test
13: Pearson's Chi-squared test
14: Pearson's Chi-squared test
15: Pearson's Chi-squared test
```

The above table shows the p-values of the various categorical variables

DELETING IRRELEVANT VARIABLES

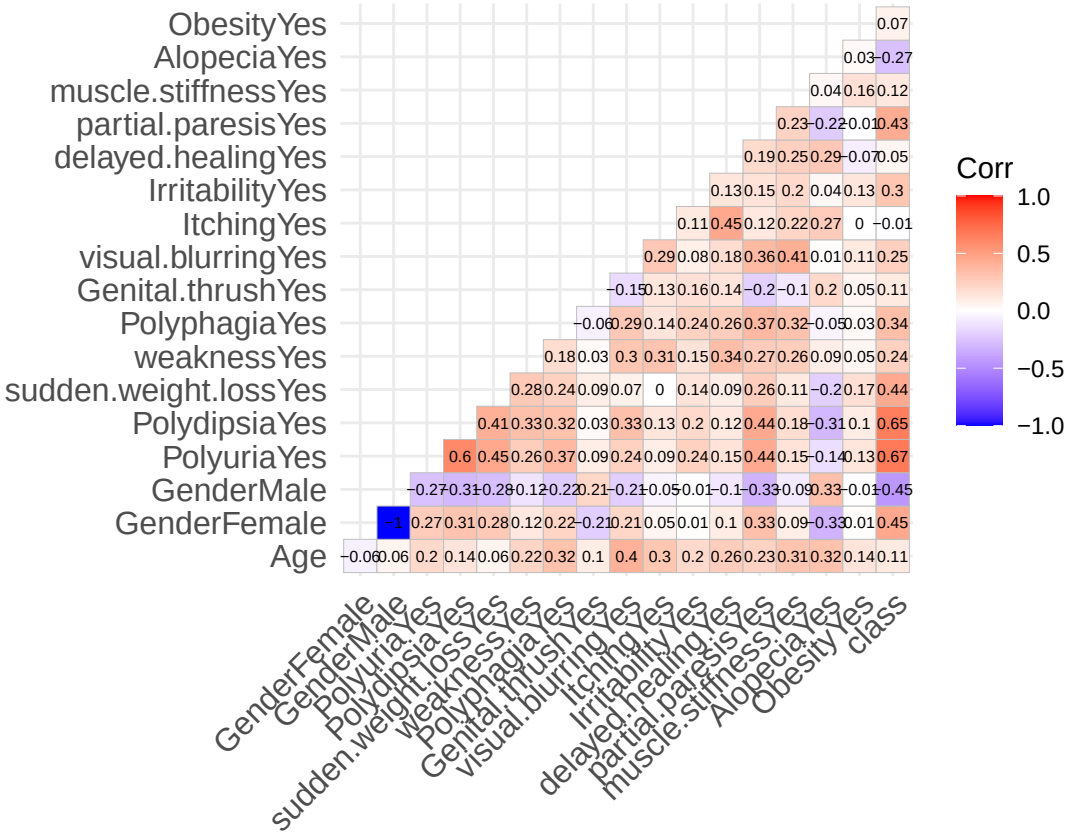
```
dat<-df[,-c(11,13)]
df$class<- ifelse(dat$class=="Negative", 0,1)
colnames(dat)
```

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

From the output, we observe that all the predictors are significant except irritability and partial.paresis given the threshold probability of 0.25. Therefore, there is an evidence that there is an association between Class and all the significant variables

CORRELATION AMONG VARIABLES

```
library(magrittr)
library(dplyr)
library(ggcorrplot)
model.matrix(~0+., data=df) %>%
  cor(use="pairwise.complete.obs") %>%
  ggcorrplot(show.diag = F, type="lower", lab=TRUE, lab_size=2)
```



From the above, we observe that there are no significant correlation among the variables

4) (Data Partition) Partition the data into two parts, the training data D1 and the test data D2, with a ratio of 2:1.

DATA PARTITION IN THE RATIO 1:2

```
n <- NROW(df); ratio <- 2/3
set.seed(123)
id.training <- sample(1:n, size=trunc(n*ratio), replace=FALSE)
D1 <- df[id.training, ]
D2 <- df[-id.training, ]
yobs <- D2$class
```

DIMENSION OF TEST AND TRAINING DATA

```
dim(D1); dim(D2)
```

```
[1] 346  17
```

```
[1] 174  17
```

After partitioning the data into training and test set in the ratio 2:1, the training data, D1 has 346 observations and 17 variables whiles the test data, D2 has 174 observations and 17 variables

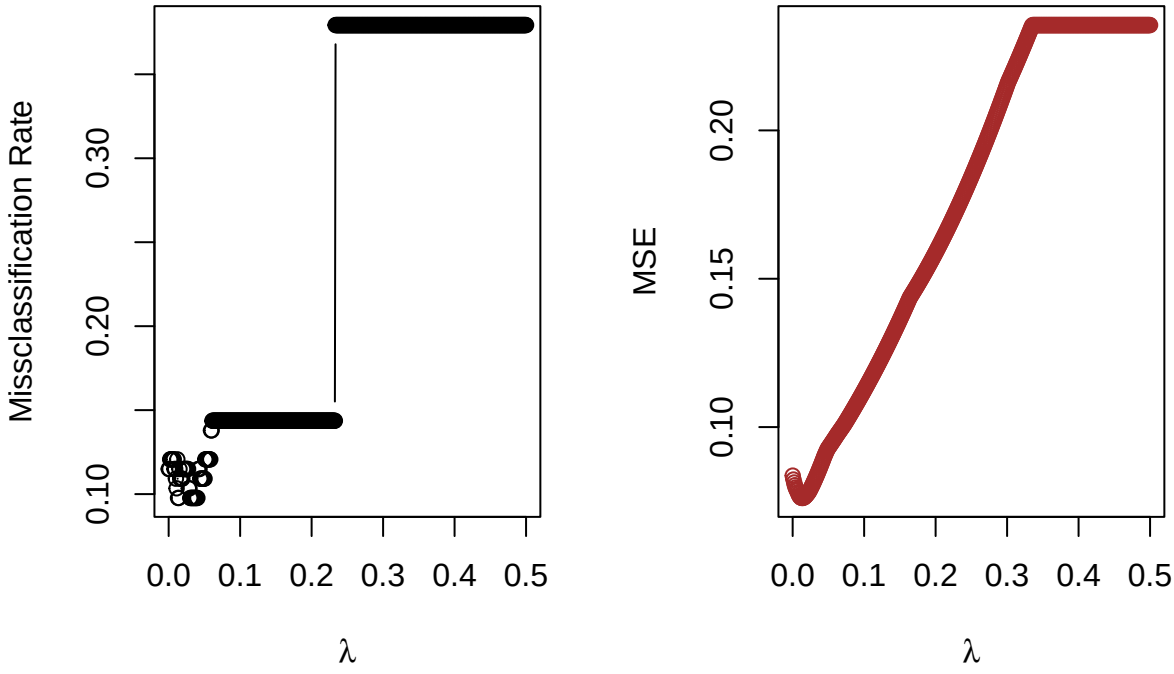
5) (Logistic Regression Modeling) We now build a logistic regression model for this medical diagnosis task.

(a) Fit the regularized logistic regression using the training data D1. While `1 regularization or LASSO is suggested here, you may use other penalty functions of your choice. Select the best tuning parameter using a validation method such as v-fold cross validation. Specify the criterion that you use for the selection. - Optionally, you may also consider including first-order interaction terms.`

```
library(ncvreg)
library(glmnet)


X <- model.matrix(as.formula(class~Age + Gender + Polyuria + Polydipsia + sudden.weight.loss + weakness
y <- D1$class
XTest <- model.matrix(as.formula(class~Age + Gender + Polyuria + Polydipsia + sudden.weight.loss + weakn
ytest <- D2$class

library(verification)
Lambda <- seq(0.0001, 0.5, length.out = 500)
L <- length(Lambda)
OUT <- matrix(0, L, 4)
for (i in 1:L){
    fit.lasso <- glmnet(x=X, y=y, family="binomial", alpha=1, # LASSO
        lambda = Lambda[i], standardize=T, thresh = 1e-07,
        maxit=3000)
    pred <- predict(fit.lasso, newx=XTest, s=Lambda[i], type="response")
    missRate <- mean(ytest != (pred > 0.5))
    mse <- mean((ytest-pred)^2)
    AUC <- roc.area(obs=ytest, pred=pred)$A
    OUT[i, ] <- c(Lambda[i], missRate, mse, AUC)
}
head(OUT)
```

```
            [,1]      [,2]       [,3]      [,4]
[1,] 0.000100000 0.1149425 0.08364297 0.9504770
[2,] 0.001101804 0.1149425 0.08232237 0.9494949
[3,] 0.002103607 0.1206897 0.08132035 0.9501964
[4,] 0.003105411 0.1206897 0.08040677 0.9534231
[5,] 0.004107214 0.1206897 0.07966144 0.9544052
[6,] 0.005109018 0.1206897 0.07906638 0.9544052
```

```
par(mfrow = c(1,2))
plot(OUT[,1], OUT[,2], type = "b", col="black",ylab = "Missclassification Rate", xlab = expression(lambd
plot(OUT[,1], OUT[,3], type = "b", col="brown",ylab = "MSE", xlab = expression(lambda))
```



From the missclassification rate plot, it can be observed that as lamda increases, the classification rate also increase but when lambda is greater than 0.15, the missclassification rate remains constant. Also, from the MSE plot above, it can be observed that as lamda increases, the MSE also increases and remains constant when MSE is greater than 0.25

SELECTING TUNING PARAMETER USING THE TEST DATA D2

```
lambda.best <- OUT[which.min(OUT[,3]), 1]
lambda.best
```

```
[1] 0.01312345
```

The best lamda is approximately 0.013 and the criteria used to select the tuning parameter is the mean square error for the predicted probabilities.

THE FINAL MODEL

```
Xnew <- rbind(X, XTest)
ynew <- c(y, ytest)
fit.best <- glmnet(x=Xnew, y=ynew, family="binomial", alpha=1,
        lambda = lambda.best, standardize=T, thresh = 1e-07,maxit=1000)
```

*SIGNIFICANT PREDICTORS

```
fit.best$beta
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                              s0
(Intercept)            .
Age                    .
GenderMale            -2.53958298
PolyuriaYes            2.82536671
PolydipsiaYes          3.00092334
sudden.weight.lossYes  0.29179601
weaknessYes            0.18544356
PolyphagiaYes          0.43884557
Genital.thrushYes      0.67657779
visual.blurringYes     .
ItchingYes            -1.07357262
delayed.healingYes    -0.08008223
muscle.stiffnessYes    .
AlopeciaYes           -0.10421821
ObesityYes             .
```

From the above output, the non-zero predictors are the significant predictors

6) (Model Assessment/Deployment) Apply the final logistic model to the test data D2. Present the ROC curve and report the area under the curve, i.e., the C-index or C-statistic.

FINAL MODEL PREDICTION USING THE TEST DATA D2

```
FinalPred1 <- predict(fit.best, newx=XTest, s=lambda.best, type="response")
```

ROC CURVE and AREA UNDER THE CURVE

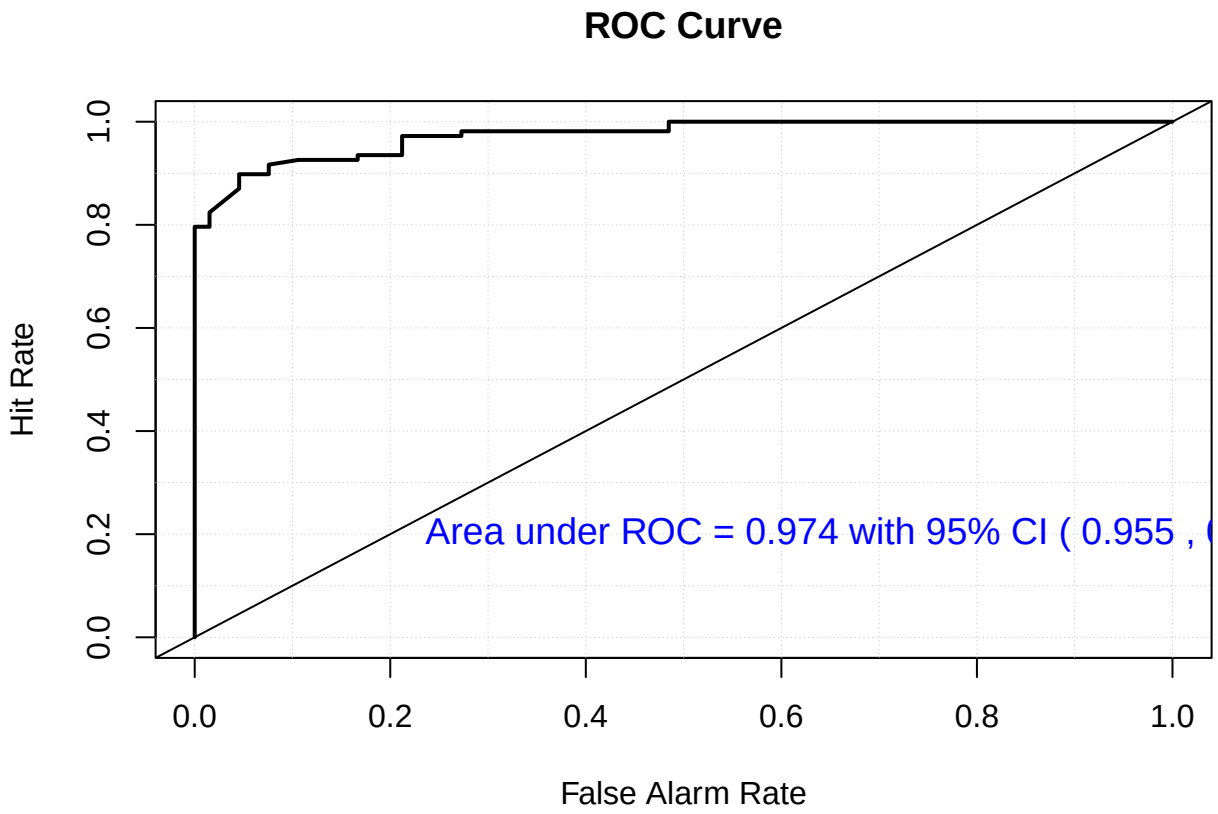
```
library(cvAUC)
AUC <- ci.cvAUC(predictions=FinalPred1, labels=ytest, folds=1:NROW(D2), confidence=0.95); AUC
```

```
Warning in if (class(predictions) == "list" | class(labels) == "list") {: the
condition has length > 1 and only the first element will be used

$cvAUC
[1] 0.9737654

$se
[1] 0.009564159

$ci
[1] 0.9550200 0.9925108

$confidence
[1] 0.95
```

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

fit.glm <- verify(obs=ytest, pred=FinalPred1)
```

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

```
roc.plot(fit.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)
```



"' *From the above ROC curve,the Area under the Curve(AUC) is 97.4% and the 95% confidence interval for the area under ROC curve is shown above*