Project-1

Chitra Karki[*] University of Texas at El Paso (UTEP)

September 03, 2022

Contents

---

1 Import the data

Bring the data into R (or Python).

```
dat = read.csv(file = "diabetes_data_upload.csv",header = T)
```

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.

[*] cbkarki@miners.utep.edu

- Are there missing values? If so, handle them with an appropriate strategy such as listwise deletion or single/multiple imputation.

```
# dimension of data frame,
dim(dat) # the dimension is 520  17
```

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

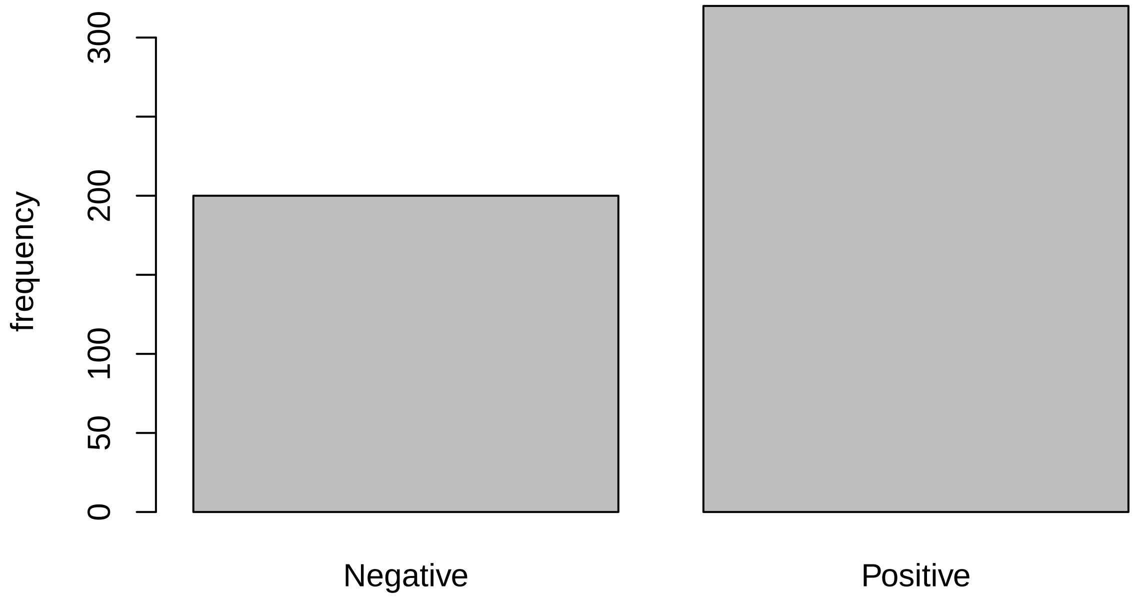
```
# variables types
str(dat) # age is of integer type and rest are of character type
```

```
## '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" ...
```

```
# checking the unique values in each columns
for (i in 1:ncol(dat)) {
  print(table(dat[,i]))

}
```

```
##
## 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
##
## Female   Male
##    192    328
##
##  No Yes
## 262 258
##
##  No Yes
## 287 233
##
##  No Yes
## 303 217
##
##  No Yes
## 215 305
##
##  No Yes
## 283 237
##
##  No Yes
## 404 116
##
##  No Yes
## 287 233
##
##  No Yes
## 267 253
##
##  No Yes
## 394 126
##
##  No Yes
## 281 239
##
##  No Yes
## 296 224
##
##  No Yes
## 325 195
##
##  No Yes
## 341 179
##
##  No Yes
## 432  88
##
## Negative Positive
##      200      320
```

```
# freq distbn of column class
plot(as.factor(dat$class),ylab = "frequency")
```



```
table(dat$class) # 200 negative and 320 positive cases
```

```
##
## Negative Positive
##      200      320
```

```
200/520; 320/520 # do not seem like unbalanced classification problem
```

```
## [1] 0.3846154
```

```
## [1] 0.6153846
```

```
# na values
sum(is.na(dat))  # no NA values
```

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

```
# library(VIM)
# aggr(dat, col=c('navyblue','yellow'), numbers=TRUE, sortVars=TRUE,
#     labels=names(dat), cex.axis=.7, gap=3, ylab=c("Missing%","Pattern"))
```

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 χ^2 test of independence or Fisher's exact test in case of small cell counts.

Output the resultant p-value for each predictor. Select a few interesting findings to present. Apply a liberal threshold significance level, say, $\alpha = 0.25$, to remove a few unimportant predictors. Note that variable screening helps reduce the dimension of predictors. Applying a liberal threshold for statistical significance here helps prevent removing predictors that are apparently not associated with the target variable without considering other predictors but may become important in a model when adding (or adjusting for) other predictors.
– Optionally, you may also compute and visualize the correlation matrix. Nevertheless, be careful with the choice of the correlation measure.

```r
# two sample t-test
test.t = t.test(dat$Age~dat$class)  # p-value = 0.01319
test.t$p.value < 0.25
```

```
## [1] TRUE
```

```r
# alternative hypothesis is true, so we will keep the age variable in model

# wilcoxon test
wilcox.test(dat$Age~dat$class) # p-value = 0.0124
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dat$Age by dat$class
## W = 27834, p-value = 0.0124
## alternative hypothesis: true location shift is not equal to 0
```

```r
# chi-squared test for Independence
p.values = NULL
for (i in 2:16) {
  test = chisq.test(dat[,i],dat[,17])
  p.values[i-1] = test$p.value
}

#creating table of p values wrt output variable class
```

```
p.val.tab = data.frame(Variables = colnames(dat)[1:16], p.value = c(test.t$p.value,p.values))

# selection of variables with  0.25 level of significance, pick = reject ho, drop= accept ho
p.val.tab$decission = ifelse(p.val.tab$p.value < 0.25,"pick","drop")
p.val.tab
```

```
##             Variables      p.value decission
## 1                 Age 1.319328e-02      pick
## 2              Gender 3.289704e-24      pick
## 3            Polyuria 1.740912e-51      pick
## 4          Polydipsia 6.187010e-49      pick
## 5  sudden.weight.loss 5.969166e-23      pick
## 6            weakness 4.869843e-08      pick
## 7          Polyphagia 1.165158e-14      pick
## 8      Genital.thrush 1.609790e-02      pick
## 9     visual.blurring 1.701504e-08      pick
## 10            Itching 8.297484e-01      drop
## 11       Irritability 1.771483e-11      pick
## 12    delayed.healing 3.266599e-01      drop
## 13    partial.paresis 1.565289e-22      pick
## 14   muscle.stiffness 6.939096e-03      pick
## 15           Alopecia 1.909279e-09      pick
## 16            Obesity 1.271080e-01      pick
```

```
# variables to drop
p.val.tab[which(p.val.tab[,3]=="drop"),1]
```

```
## [1] "Itching"         "delayed.healing"
```

```
# dropping the variables

dat = dat[,-c(which(p.val.tab[,3]=="drop"))]
```

4 Data Partition

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

```
set.seed(123)
n.row = nrow(dat)
indices = sample(1:n.row,size = round((2/3)*n.row))

# converting the class to lebels
dat$class = ifelse(dat$class=="Negative",0,1)
```

```
# training data
d1 = dat[indices,]

# test data
d2 = dat[-indices,]

y.obs = d2$class
x.test = d2[,-17]
```

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 L1 regularization or LASSO is suggested here, you may use other penalty functions of your choice.
Select the best tuning parameter λ using a validation method such as v-fold cross validation. Specify the criterion that you use for the selection.

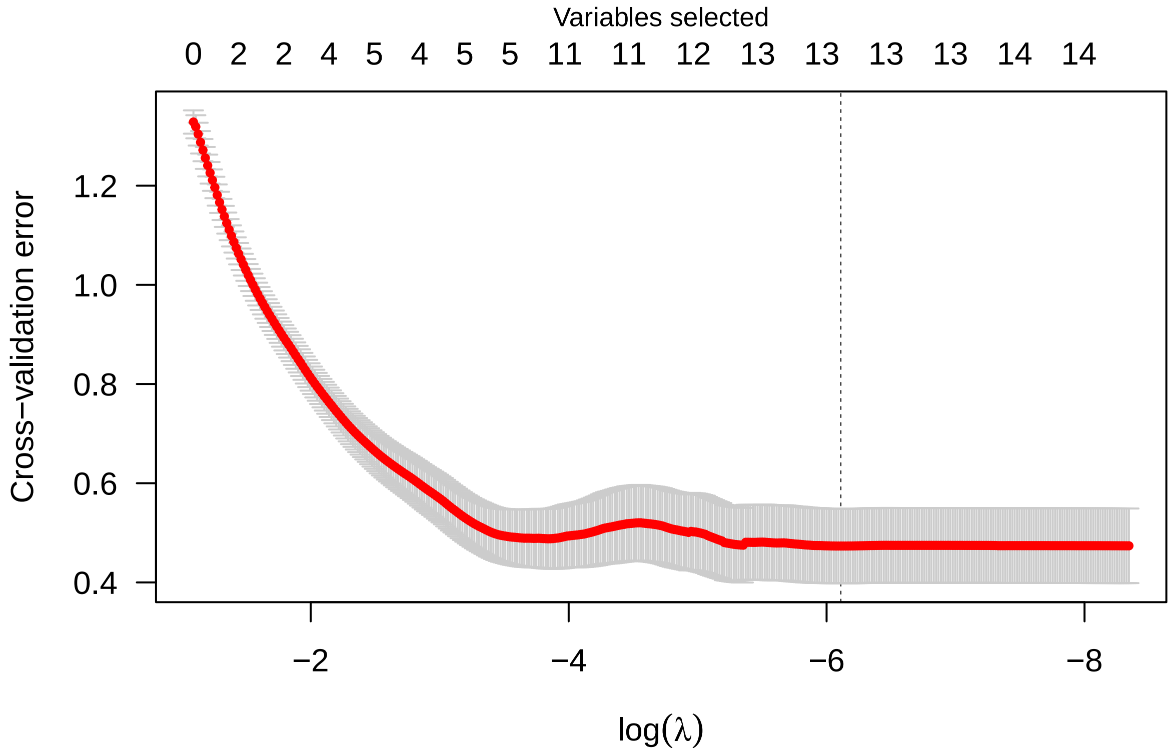
– Optionally, you may also consider including first-order interaction terms.

(b) Present your final 'best' model. Which variables are important predictors? Interpret the results.

```
# (a)

# =============================================
# LOGISTIC REGRESSION WITH REGULARIZATION
# =============================================

# install.packages("ncvreg")# none concave regression
suppressPackageStartupMessages(library(ncvreg))

y = d1$class
x = model.matrix(~-1+ Age + Gender + Polyuria + Polydipsia + sudden.weight.loss +
               weakness + Polyphagia + Genital.thrush + visual.blurring +
                Irritability + partial.paresis +
               muscle.stiffness + Alopecia + Obesity,
               data = d1[,-17])

# tuning lambda
cvfit.SCAD = cv.ncvreg(X=x,y=y, nfolds=5, family="binomial", penalty="SCAD",
    lambda.min=.0001, nlambda=500, eps=.001, max.iter=1000,seed = 123)
```

```
## Warning in ncvreg(X = X, y = y, ...): Maximum number of iterations reached
```

```
# USING THE ARGUMENT penalty="MCP" TO CHOOSE AMONG DIFFERENT PENALTY FUNCTIONS
# glmnet is better it uses 2 se rule

plot(cvfit.SCAD)
```



```
# survived betas
result.SCAD <- cvfit.SCAD$fit
beta.hat <- as.vector(result.SCAD$beta[-1, cvfit.SCAD$min])
cutoff <- 0

# predictors included in final model
terms <- colnames(x)[abs(beta.hat) > cutoff]; terms # x is the model matrix
```

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

```
formula.SCAD <- as.formula(paste(c("class ~ 1+ Gender+ Age",
                          paste(lapply(terms[3:length(terms)], function(x) substr(x,1,nchar(x)-
```

```r
# (b)
#final model
fit.SCAD <- glm(formula.SCAD , data = d1, family="binomial")

summary(fit.SCAD)
```

```
##
## Call:
## glm(formula = formula.SCAD, family = "binomial", data = d1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.53273  -0.23644   0.01142   0.08781   3.06759
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.77610    1.16670   1.522  0.12793
## GenderMale            -3.54607    0.66803  -5.308 1.11e-07 ***
## Age                   -0.04719    0.02870  -1.644  0.10010
## PolyuriaYes            3.33396    0.68116   4.895 9.85e-07 ***
## PolydipsiaYes          3.28530    0.82498   3.982 6.83e-05 ***
## sudden.weight.lossYes  0.96234    0.56699   1.697  0.08964 .
## PolyphagiaYes          1.44280    0.66028   2.185  0.02888 *
## Genital.thrushYes      1.77394    0.64057   2.769  0.00562 **
## visual.blurringYes     0.65173    0.75731   0.861  0.38947
## IrritabilityYes        2.91041    0.71951   4.045 5.23e-05 ***
## partial.paresisYes     1.26851    0.61833   2.051  0.04022 *
## muscle.stiffnessYes   -0.66353    0.68958  -0.962  0.33594
## AlopeciaYes           -1.17990    0.65946  -1.789  0.07358 .
## ObesityYes            -0.94621    0.64406  -1.469  0.14180
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 463.81  on 346  degrees of freedom
## Residual deviance: 118.48  on 333  degrees of freedom
## AIC: 146.48
##
## Number of Fisher Scoring iterations: 8
```

The summary of the final fitted model shows the predictors in final model which were obtained form cross validation method corresponding to the minimum cross validation error. According to final model, predictors Age, Polyphagia ,visual.blurring,muscle.stiffness Obesity are not significant at any level for prediction the class variable.

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.

```
# prediction for test data d2 ie x.test
yhat <- predict(fit.SCAD, newdata=x.test, type="response")

# ==================
# ROC CURVE AND AUC
# ==================

# install.packages("verification")
suppressPackageStartupMessages(library(verification))
a.ROC <- roc.area(obs=y.obs, pred=yhat)$A
print(a.ROC)
```

```
## [1] 0.9559829
```

```
# USING PACKAGE cvAUC
# install.packages("cvAUC")
suppressPackageStartupMessages(library(cvAUC))
AUC <- ci.cvAUC(predictions=yhat, labels=y.obs, folds=1:NROW(x.test), confidence=0.95); AUC
```

```
## $cvAUC
## [1] 0.9559829
##
## $se
## [1] 0.01517276
##
## $ci
## [1] 0.9262448 0.9857210
##
## $confidence
## [1] 0.95
```

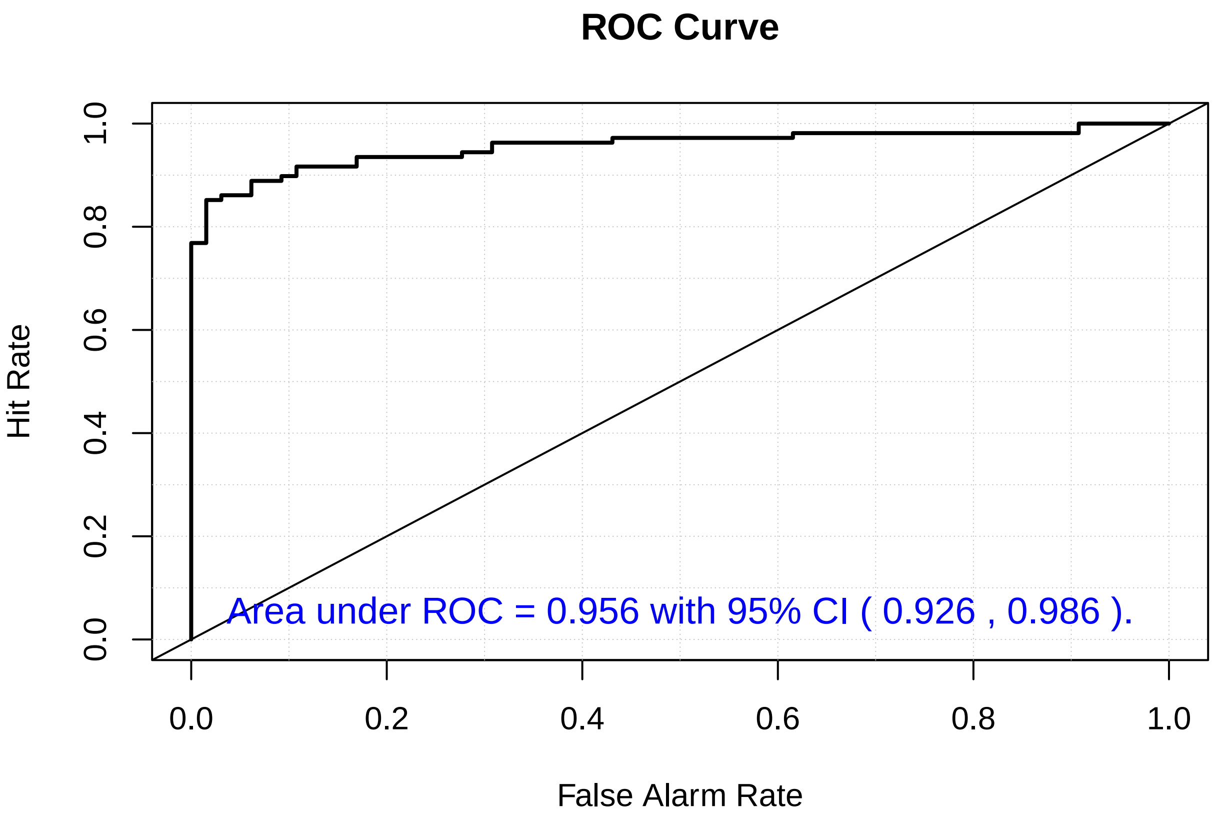
```
auc.ci <- round(AUC$ci, digits=3);auc.ci
```

```
## [1] 0.926 0.986
```

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

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

```
roc.plot(mod.glm, plot.thres = NULL)
text(x=0.5, y=0.05, paste("Area under ROC =", round(AUC$cvAUC, digits=3),
    "with 95% CI (", auc.ci[1], ",", auc.ci[2], ").",
    sep=" "), col="blue", cex=1.2)
```



The area under the ROC curve is 0.956. Which is close to one, so the model should be a good one.