Diabetes Risk prediction via SEMMA w/ Regularized Logistic Regression

Kevin Acosta[*]   M.S. in Data & Information Sciences (UTEP)

February 08, 2025

Contents

---

1 Data Import

```
data <- read.csv("/Users/kevinacosta/Desktop/Stats ML 2/Comp Proj 01/diabetes_data_upload.csv"
dim(data)
```

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

```
summary(data)
```

```
##       Age           Gender            Polyuria          Polydipsia
##  Min.   :16.00   Length:520         Length:520         Length:520
##  1st Qu.:39.00   Class :character   Class :character   Class :character
##  Median :47.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   :48.03
##  3rd Qu.:57.00
##  Max.   :90.00
##  sudden.weight.loss   weakness           Polyphagia         Genital.thrush
##  Length:520         Length:520         Length:520         Length:520
```

[*] kmacosta2@miners.utep.edu

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  visual.blurring      Itching          Irritability       delayed.healing
##  Length:520          Length:520         Length:520         Length:520
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  partial.paresis    muscle.stiffness     Alopecia           Obesity
##  Length:520          Length:520         Length:520         Length:520
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     class
##  Length:520
##  Class :character
##  Mode  :character
##
##
##
```

```
head(data)[c(1,2,3),]
```

```
##   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
##   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
##   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
```

2 Exploratory Data Analysis

```
table(data$class) # simple frequency table
```

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

```r
for (i in 1:(ncol(data)-1)) {
  cat("Column:", colnames(data)[i],"--> ")
  x <- data[,i]
  cat("Number of unique values:", length(unique(x, incomparables=TRUE)),"\n")
  # checking for missing values per column
  col_count <- sum(is.na(x))

  if(col_count>0) {
    cat("Missing Values:", col_count, "\n")
  }
  else {
    cat("\t\tNONE Missing for this column\n")
  }

}
```

```
## Column: Age --> Number of unique values: 51
##      NONE Missing for this column
## Column: Gender --> Number of unique values: 2
##      NONE Missing for this column
## Column: Polyuria --> Number of unique values: 2
##      NONE Missing for this column
## Column: Polydipsia --> Number of unique values: 2
##      NONE Missing for this column
## Column: sudden.weight.loss --> Number of unique values: 2
##      NONE Missing for this column
## Column: weakness --> Number of unique values: 2
##      NONE Missing for this column
## Column: Polyphagia --> Number of unique values: 2
##      NONE Missing for this column
## Column: Genital.thrush --> Number of unique values: 2
##      NONE Missing for this column
## Column: visual.blurring --> Number of unique values: 2
##      NONE Missing for this column
## Column: Itching --> Number of unique values: 2
##      NONE Missing for this column
## Column: Irritability --> Number of unique values: 2
##      NONE Missing for this column
## Column: delayed.healing --> Number of unique values: 2
##      NONE Missing for this column
## Column: partial.paresis --> Number of unique values: 2
##      NONE Missing for this column
## Column: muscle.stiffness --> Number of unique values: 2
##      NONE Missing for this column
## Column: Alopecia --> Number of unique values: 2
##      NONE Missing for this column
## Column: Obesity --> Number of unique values: 2
##      NONE Missing for this column
```

After seeing the first frequency table, we can confirm this does resemble an unbalanced classification problem favoring 'positive' labels with 200 negative cases for Diabetes and 320 positive cases. Using a for loop I checked for every column/feature excluding our response variable and confirmed that there are no missing values found.

3 Variable Screening

```r
#library(tidyverse)
response <- data$class
p_values <- numeric(ncol(data) - 1)

data2 <- data
data2[, 1:(ncol(data2) - 1)] <- lapply(data2[, 1:(ncol(data2) - 1)], as.factor)

for (i in 1:(ncol(data2) - 1)) {
  attrib <- data2[, i]

  if (is.factor(attrib)){ # categorical attribute --> chi-squared test
    cat_attr <- chisq.test(table(attrib, response))
    p_values[i] <-  cat_attr$p.value
  }
  else{ # continuous attribute --> t-test
    cont_attr <- t.test(attrib ~ response)
    p_values[i] <- cont_attr$p.value
  }
  cat("Attribute: ", colnames(data2)[i], "\n\t\tP-value: ", p_values[i], "\n")
}
```

```
## Warning in chisq.test(table(attrib, response)): Chi-squared approximation may
## be incorrect
```

```
## Attribute:  Age
##      P-value:  2.654685e-11
## Attribute:  Gender
##      P-value:  3.289704e-24
## Attribute:  Polyuria
##      P-value:  1.740912e-51
## Attribute:  Polydipsia
##      P-value:  6.18701e-49
## Attribute:  sudden.weight.loss
##      P-value:  5.969166e-23
## Attribute:  weakness
##      P-value:  4.869843e-08
## Attribute:  Polyphagia
##      P-value:  1.165158e-14
## Attribute:  Genital.thrush
##      P-value:  0.0160979
## Attribute:  visual.blurring
##      P-value:  1.701504e-08
## Attribute:  Itching
##      P-value:  0.8297484
## Attribute:  Irritability
##      P-value:  1.771483e-11
## Attribute:  delayed.healing
##      P-value:  0.3266599
## Attribute:  partial.paresis
##      P-value:  1.565289e-22
## Attribute:  muscle.stiffness
##      P-value:  0.006939096
## Attribute:  Alopecia
##      P-value:  1.909279e-09
## Attribute:  Obesity
##      P-value:  0.127108
```

```r
# identifying the statistically significant attributes
significant_attrs <- which(p_values < 0.25 & !is.na(p_values))
significant_names <- colnames(data2)[significant_attrs]

print("Significant attributes based on level alpha 0.25:\n")
```

```
## [1] "Significant attributes based on level alpha 0.25:\n"
```

```r
print(significant_names)
```

```
##  [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"
```

4 Data Partition

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

```r
data$Itching <- NULL
data$delayed.healing <- NULL

n_samples <- nrow(data)
# 2/3 for train set below
n_train <- round(n_samples * 0.67) - 1
train <- sample(n_samples, n_train)
train_data <- data[train,]
test_data <- data[-train,]
```

```
dim(train_data)
```

```
## [1] 347  15
```

```
dim(test_data)
```

```
## [1] 173  15
```

5 Logistic Regression Modeling

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
y <- as.factor(train_data$class)
X <- model.matrix(class ~ ., data=train_data)[,-1] #
# fitting a lasso logistic regression with cv to help select the best lambda value.
cv_lasso_fit <- cv.glmnet(x=X,y=y, nfolds=10, family="binomial", alpha=1)
best_lambda <- cv_lasso_fit$lambda.min
#plot(cv_lasso_fit)

# here we're simply fitting again but to include the best lambda & afterwards
# inspect the coefficients
best_lasso <- glmnet(x=X, y=y, family="binomial", alpha=1, lambda = best_lambda)
cat("Best lambda:", best_lambda)
```

```
## Best lambda: 0.007410841
```

```
print("Final lasso model coefficients:")
```

```
## [1] "Final lasso model coefficients:"
```

```
print(coef(best_lasso))
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)            1.07933564
## Age                   -0.01889829
## GenderMale            -3.03847973
## PolyuriaYes            2.90296382
## PolydipsiaYes          2.87067604
## sudden.weight.lossYes  0.37104319
## weaknessYes            0.15711049
## PolyphagiaYes          0.72006422
## Genital.thrushYes      0.24066259
## visual.blurringYes     .
## IrritabilityYes        1.69408390
## partial.paresisYes     0.99441119
## muscle.stiffnessYes   -0.65258949
## AlopeciaYes           -0.37576883
## ObesityYes            -0.39621067
```

The model coefficient results presented above can vary from run to run, but as long as they have a value not zero, we would list them as significant to the model.

In general and on average, the most relevant coefficients are: ‘PolydipsiaYes’- (a lot of thirst is indicated with the highest positive coefficient), ‘PolyuriaYes’- (excessive urination is also indicated by a large coefficient, so its another strong predictor of diabetes) ‘GenderMale’- (so just being male significantly decreases the odds of getting a diabetes diagnosis because the coefficient is negative!).

6 Model Assessment/Deployment

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
y_test <- as.factor(test_data$class)
x_test <- model.matrix(class ~ ., data=test_data)[,-1]

# predict probability for a positive classification/diagnosis
y_pred <- predict(best_lasso, newx=x_test, type="response")

# generate ROC curve and get AUC
roc_curve <- roc(y_test, y_pred)
```

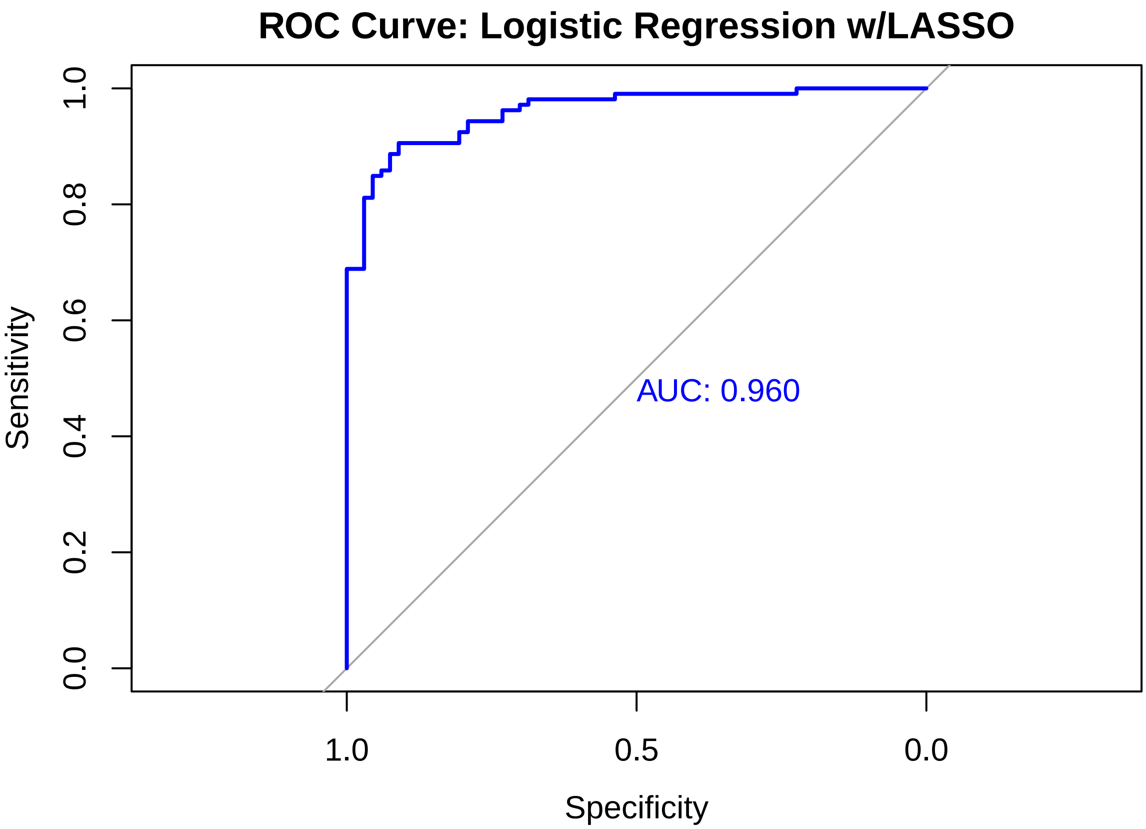
```
## Setting levels: control = Negative, case = Positive
```

```
## Warning in roc.default(y_test, y_pred): Deprecated use a matrix as predictor.
## Unexpected results may be produced, please pass a numeric vector.
```

```
## Setting direction: controls < cases
```

```r
auc_value <- auc(roc_curve)

plot(roc_curve, col = "blue", main = "ROC Curve: Logistic Regression w/LASSO", print.auc = TRU
```



```
print(paste("AUC/C-Statistic:", round(auc_value, 4)))
```

```
## [1] "AUC/C-Statistic: 0.9602"
```

Overall the Logistic Regression Classifier's performance is very good at predicting what is considered the true positives in classifying patients with diabetes when they do in fact have it.