Project I: SEMMA with Regularized Logistic Regression

Quaye, George Ekow

Contents

1 Bringing in the data **2**

2 Exploratory Data Analysis **2**

2.1 Variable Type 2

2.2 Frequency distribution of the target variable 3

2.3 Checking for Missing values 3

3 Variable Screening **5**

3.1 Chi-Square test of association 5

3.2 Deletion of non significant variables 6

3.3 Correlation plot among the variables 6

4 Data Partition **7**

5 Logistic Regression Modeling **7**

5.1 Fitting the model 7

5.2 Selecting best tuning parameter using validation data 8

5.3 Final best model fit 9

5.4 Checking for important predictors 9

6 Model Assestment / Deployment **9**

1 Bringing in the data

1. Bring the data into R (or Python)

```
#Reading the data
data <- read.table(file = "diabetes_data_upload.csv",sep=",", header = T, na.strings = c("NA", "", " ")
                   stringsAsFactors = T)
dim(data)
```

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

The data set has 520 observations and 17 columns.

2 Exploratory Data Analysis

2.1 Variable Type

```
colnames(data)
```

```
##  [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"
```

```
str(data)
```

```
## 'data.frame':    520 obs. of  17 variables:
##  $ Age               : int  40 58 41 45 60 55 57 66 67 70 ...
##  $ Gender            : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Polyuria          : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 2 2 2 1 ...
##  $ Polydipsia        : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 2 2 2 2 2 ...
##  $ sudden.weight.loss: Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 1 2 1 2 ...
##  $ weakness          : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Polyphagia        : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 2 2 1 2 2 ...
##  $ Genital.thrush    : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 1 2 1 ...
##  $ visual.blurring   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 2 1 2 1 2 ...
##  $ Itching           : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 1 2 2 2 ...
##  $ Irritability      : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 2 2 ...
##  $ delayed.healing   : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 1 1 1 ...
##  $ partial.paresis   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 2 2 2 1 ...
##  $ muscle.stiffness  : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 1 2 2 1 ...
##  $ Alopecia          : Factor w/ 2 levels "No","Yes": 2 2 2 1 2 2 1 1 1 2 ...
##  $ Obesity           : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 2 1 1 2 1 ...
##  $ class             : Factor w/ 2 levels "Negative","Positive": 2 2 2 2 2 2 2 2 2 2 ...
```

A data with 520 observations on 17 variables, 1 being numeric variables and 16 being nominal(categorical) variables.

2.2 Frequency distribution of the target variable

inspect the frequency distribution of the target variable class and see, e.g., whether we have an unbalanced classification problem.

```
#inspect the frequency distribution of the target variable class.
library(questionr)
freq(data$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 output the counts for negative are 200 and that for positive are 320. Thus in this scenario, we don't have a very unbalanced classification problem.

2.3 Checking for Missing values

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

```
# INSPECT THE DISTINCT VALUES OF EACH X
cols<- 2:NCOL(data)
for (j in cols){
 print(colnames(data)[j])
 print(table(table(data[,j]), useNA="ifany"))
}
```

```
## [1] "Gender"
##
## 192 328
##   1   1
## [1] "Polyuria"
##
## 258 262
##   1   1
## [1] "Polydipsia"
##
## 233 287
##   1   1
## [1] "sudden.weight.loss"
##
## 217 303
##   1   1
## [1] "weakness"
##
## 215 305
```

```
##   1   1
## [1] "Polyphagia"
##
## 237 283
##   1   1
## [1] "Genital.thrush"
##
## 116 404
##   1   1
## [1] "visual.blurring"
##
## 233 287
##   1   1
## [1] "Itching"
##
## 253 267
##   1   1
## [1] "Irritability"
##
## 126 394
##   1   1
## [1] "delayed.healing"
##
## 239 281
##   1   1
## [1] "partial.paresis"
##
## 224 296
##   1   1
## [1] "muscle.stiffness"
##
## 195 325
##   1   1
## [1] "Alopecia"
##
## 179 341
##   1   1
## [1] "Obesity"
##
##  88 432
##   1   1
## [1] "class"
##
## 200 320
##   1   1
```

```
# MISSING PERCENTAGES FOR ALL COLUMNS (OR VARIABLES)
colMeans(is.na(data))
```

```
##                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
```

There are no missing values in the data set.

3 Variable Screening

3.1 Chi-Square test of association

Explore the marginal (bivariate) associations between class and each attribute/predictor.

```
#Bivariate association of the response with the categorical predictors.
m<-data[,-c(1)]

library(car);
cols.x <- 1:(NCOL(m)-1)
xnames <- names(m)[cols.x]
y <- m$class
OUT <- NULL
for (j in 1:length(cols.x)){
x <- data[, cols.x[j]]
xname <- xnames[j]
tbl <- table(y, x)
pvalue <- chisq.test(tbl)$p.value
OUT <- rbind(OUT, cbind(xname=xname, pvalue=pvalue)) }
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
OUT <- as.data.frame(OUT)
colnames(OUT) <- c("name", "pvalue")
OUT
```

```
##                   name               pvalue
## 1               Gender 2.65468477115389e-11
## 2              Polyuria 3.28970373055333e-24
## 3            Polydipsia 1.74091178034421e-51
## 4  sudden.weight.loss 6.18700964088628e-49
## 5             weakness 5.96916626254991e-23
## 6            Polyphagia 4.86984344658554e-08
## 7       Genital.thrush 1.16515843464091e-14
## 8      visual.blurring   0.0160979029919381
## 9              Itching 1.70150367532412e-08
## 10       Irritability    0.829748395948501
## 11    delayed.healing 1.77148314939594e-11
## 12    partial.paresis    0.326659937714402
## 13   muscle.stiffness 1.56528907105633e-22
## 14            Alopecia  0.00693909569792398
## 15             Obesity 1.90927949636339e-09
```

3.2 Deletion of non significant variables

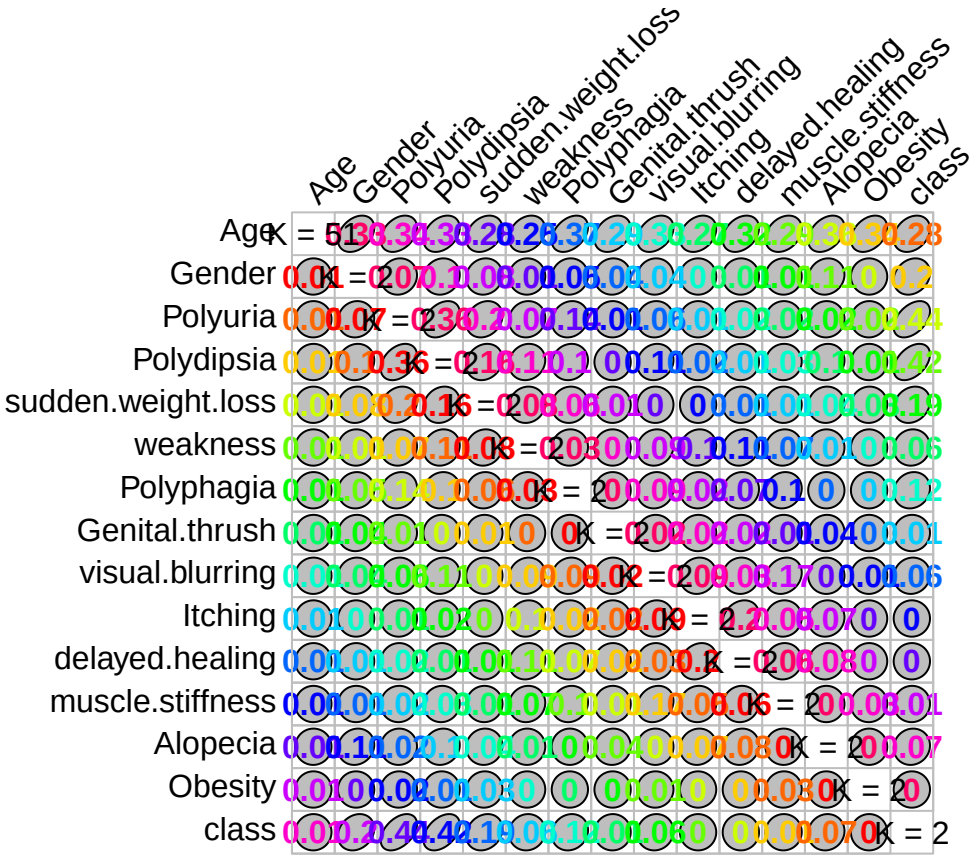
```
#Taking the non significant variables out
data<-data[,-c(11,13)]
data$class<- ifelse(data$class=="Negative", 0,1)
colnames(data)
```

```
##  [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, all the predictors are significant except partial.paresis and irritability given the threshold probability of 0.25. Thus there is significant evidence there exist an association between Class and all significant attributes.

3.3 Correlation plot among the variables

```
library(GoodmanKruskal)
data1<- data[,-c(17)]
dat<- GKtauDataframe(data1)
plot(dat, corColors = "magenta")
```



There appear to be no correlation between the predictor variables

4 Data Partition

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

```
#Partitioning data
set.seed(125)
sampleData <- sample(nrow(data), (2.0/3.0)*nrow(data), replace = FALSE) # training set
D1 <- data[sampleData, ]
# test set
D2 <- data[-sampleData, ]
dim(D1)
```

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

Given the train data there are 346 observations with 15 variables.

```
dim(D2)
```

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

Given the test data there are 174 observations with 15 variables.

5 Logistic Regression Modeling

5.1 Fitting the model

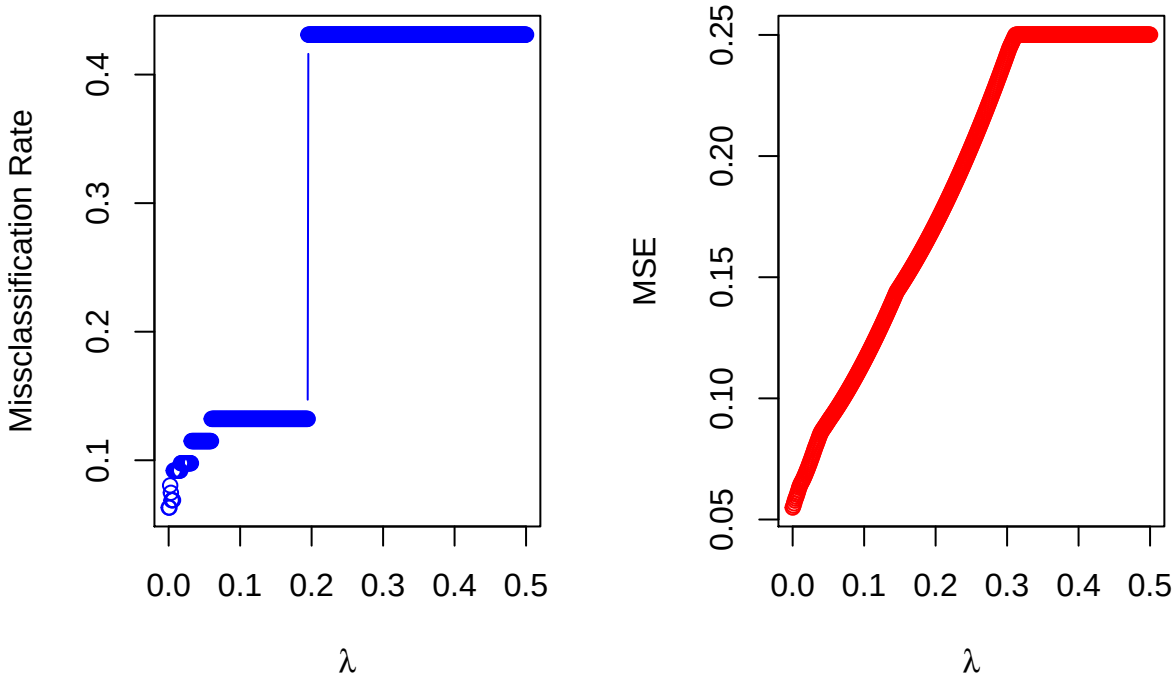
5)a We now build a logistic regression model for this medical diagnosis task.

```
#Fitting a regularized logistic regression model
library(ncvreg)
library(glmnet)
formula0 <- class~Age + Gender + Polyuria + Polydipsia + sudden.weight.loss + weakness + Polyphagia + G

X <- model.matrix(as.formula(formula0), data=D1)
y <- D1$class
XTest <- model.matrix(as.formula(formula0), data=D2)
ytest <- D2$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 0.06321839 0.05493920 0.9820875
## [2,] 0.001101804 0.06321839 0.05571836 0.9806061
## [3,] 0.002103607 0.08045977 0.05669266 0.9808754
## [4,] 0.003105411 0.07471264 0.05779785 0.9811448
## [5,] 0.004107214 0.06896552 0.05860718 0.9797980
## [6,] 0.005109018 0.06896552 0.05939389 0.9796633
```

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



Given the plot of missclassifcation rate it is seen that as λ increases, the classification rate also increases. However when $\lambda >= 0.2$, the missclassification rate remains constant. Also given the plot of MSE it is noticed that as λ increases, the MSE also increases. However when $\lambda >= 0.3$, the MSE remains constant.

5.2 Selecting best tuning parameter using validation data

```
#Selection of tuning parameter using the validation data D2
lambda.best <- OUT[which.min(OUT[,3]), 1]; lambda.best
```

```
## [1] 1e-04
```

The criteria used to select the tuning parameter is the mean square error for the predicted probabilities.

5.3 Final best model fit

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

```
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=3000)
```

5.4 Checking for important predictors

```
#Checking for important predictors.
fit.best$beta
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)            .
## Age                   -0.02513673
## GenderMale            -3.92851513
## PolyuriaYes            4.84053722
## PolydipsiaYes          4.94410002
## sudden.weight.lossYes  0.12703286
## weaknessYes            1.17885155
## PolyphagiaYes          1.23775289
## Genital.thrushYes      1.78626578
## visual.blurringYes     1.23766139
## ItchingYes            -2.84209239
## delayed.healingYes    -0.46436377
## muscle.stiffnessYes   -0.79666928
## AlopeciaYes           -0.32192622
## ObesityYes             0.09849529
```

From the output, the coefficients with non zero values are the important predictors.

6 Model Assestment / Deployment

6) Apply the final logistic model to the test data $D2$. Present the ROC curve and the area under the curve, i.e., the C-index.

```
# Final model to test data
FinalPred <- predict(fit.best, newx=XTest, s=lambda.best, type="response")
```

```r
# ROC Curve and AUC
library(cvAUC)
AUC <- ci.cvAUC(predictions=FinalPred, labels=ytest, folds=1:NROW(D2), confidence=0.95); AUC
```

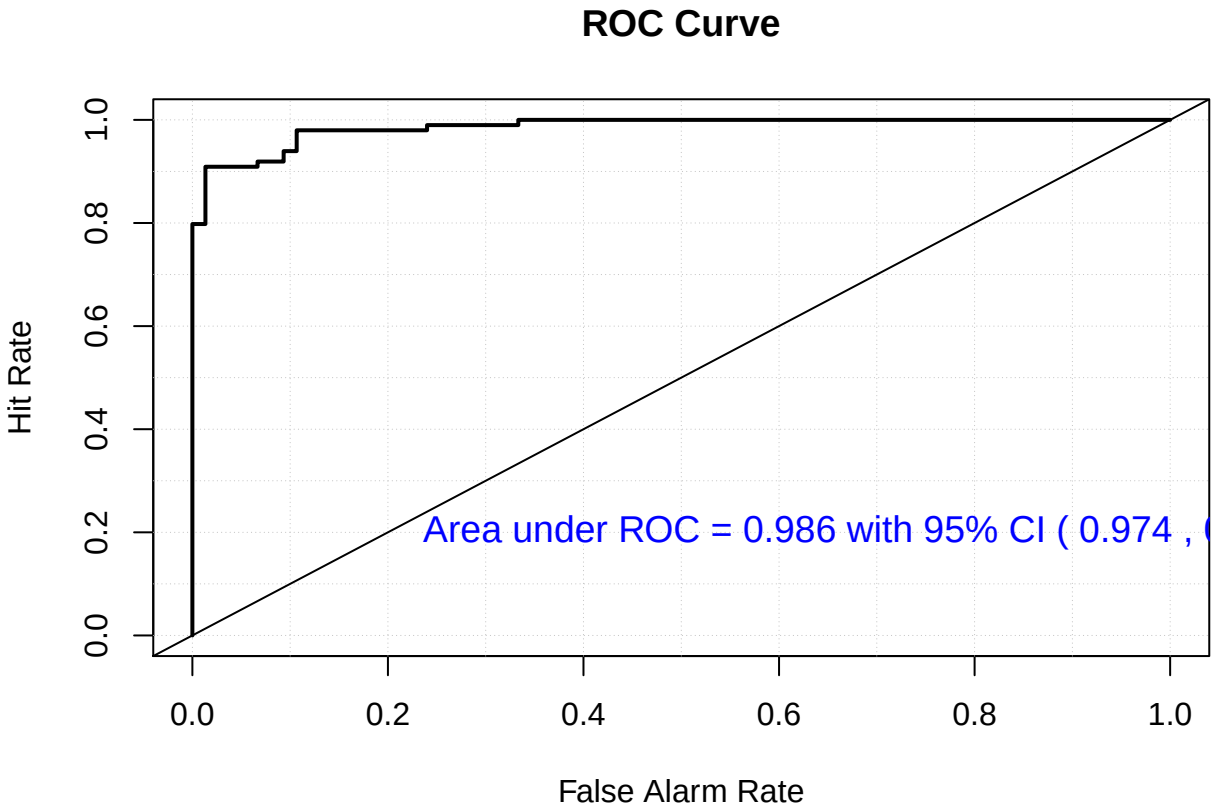
```
## Warning in if (class(predictions) == "list" | class(labels) == "list") {: the
## condition has length > 1 and only the first element will be used
```

```
## $cvAUC
## [1] 0.9858586
##
## $se
## [1] 0.006046578
##
## $ci
## [1] 0.9740075 0.9977097
##
## $confidence
## [1] 0.95
```

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

mod.glm <- verify(obs=ytest, pred=FinalPred)
```

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

```r
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)
```



The Area under ROC curve is obtained as 0.986 and the confidence interval for the area under ROC curve is also shown on the plot with 95% confidence level.