Statlog Heart Disease Prediction using Logistic Regression, Lasso and Elastic-Net Regularized Generalized Linear Models, and Principal Component

Wei Chieh Chen

2025-05-02

Goal

Use the Statlog (Heart) dataset. Fitting three models: GLM, GLMNET, and PCA (first two components). Compute ROC curves for all three models using base R.

Step 1: Load the heart.dat file from the repository

```r
# Read the CSV file from the raw GitHub URL
heart_data <- read.table("https://raw.githubusercontent.com/JavoNazarov/Statlog-Heart-Disease-Predictio

# heart_data <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/hea

# Rename columns based on the provided table
colnames(heart_data) <- c("age", "sex", "chest_pain", "blood_press", "serum_chol",
                          "blood_sugar", "electrocard", "max_heart_rate",
                          "induc_ang", "oldpeak", "peak_st_seg", "major_ves",
                          "thal", "presence")

# View the first few rows of the data
head(heart_data)
```

```
##   age sex chest_pain blood_press serum_chol blood_sugar electrocard
## 1  70   1          4         130        322           0           2
## 2  67   0          3         115        564           0           2
## 3  57   1          2         124        261           0           0
## 4  64   1          4         128        263           0           0
## 5  74   0          2         120        269           0           2
## 6  65   1          4         120        177           0           0
##   max_heart_rate induc_ang oldpeak peak_st_seg major_ves thal presence
## 1            109         0     2.4           2         3    3        2
## 2            160         0     1.6           2         0    7        1
## 3            141         0     0.3           1         0    7        2
## 4            105         1     0.2           2         1    7        1
## 5            121         1     0.2           1         1    3        1
## 6            140         0     0.4           1         0    7        1
```

Step 2: Verify the Data

From the variable table, the dataset should have 270 rows and 14 columns, as per the Statlog (Heart) dataset description. There are no missing values, as indicated in your table. The column types should match the descriptions: - Continuous: age, rest-bp, serum-chol, max-heart-rate, oldpeak, major-vessels - Binary: sex, fasting-blood-sugar, angina - Categorical: chest-pain, electrocardiographic, thal - Integer: slope, heart-disease (target)

In this Project, the dataset is at the given URL and has 14 columns: 13 predictors and 1 target (heart-disease). heart-disease is coded as 1 (absence) and 2 (presence); convert it to 0 (absence) and 1 (presence) for binary logistic regression. Convert binary and categorical variables (chest-pain, electrocardiographic, thal) to factors for glm compatibility, as glmnet will handle them via model.matrix.

```r
str(heart_data) # Data Structure
```

```
## 'data.frame':    270 obs. of  14 variables:
##  $ age          : num  70 67 57 64 74 65 56 59 60 63 ...
##  $ sex          : num  1 0 1 1 0 1 1 1 1 0 ...
##  $ chest_pain   : num  4 3 2 4 2 4 3 4 4 4 ...
##  $ blood_press  : num  130 115 124 128 120 120 130 110 140 150 ...
##  $ serum_chol   : num  322 564 261 263 269 177 256 239 293 407 ...
##  $ blood_sugar  : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ electrocard  : num  2 2 0 0 2 0 2 2 2 2 ...
##  $ max_heart_rate: num  109 160 141 105 121 140 142 142 170 154 ...
##  $ induc_ang    : num  0 0 0 1 1 0 1 1 0 0 ...
##  $ oldpeak      : num  2.4 1.6 0.3 0.2 0.2 0.4 0.6 1.2 1.2 4 ...
##  $ peak_st_seg  : num  2 2 1 2 1 1 2 2 2 2 ...
##  $ major_ves    : num  3 0 0 1 1 0 1 1 2 3 ...
##  $ thal         : num  3 7 7 7 3 7 6 7 7 7 ...
##  $ presence     : int  2 1 2 1 1 1 2 2 2 2 ...
```

```r
# Convert binary and categorical variables to factors
heart_data$sex <- as.factor(heart_data$sex)
heart_data$chest_pain <- as.factor(heart_data$chest_pain)
heart_data$blood_sugar <- as.factor(heart_data$blood_sugar)
heart_data$electrocard <- as.factor(heart_data$electrocard)
heart_data$induc_ang <- as.factor(heart_data$induc_ang)
heart_data$thal <- as.factor(heart_data$thal)

# Convert heart-disease to binary (0 = absence, 1 = presence)
heart_data$presence <- ifelse(heart_data$presence == 2, 1, 0)

summary(heart_data)
```

```
##       age         sex     chest_pain  blood_press      serum_chol    blood_sugar
##  Min.   :29.00   0: 87   1: 20      Min.   : 94.0   Min.   :126.0   0:230
##  1st Qu.:48.00   1:183   2: 42      1st Qu.:120.0   1st Qu.:213.0   1: 40
##  Median :55.00           3: 79      Median :130.0   Median :245.0
##  Mean   :54.43           4:129      Mean   :131.3   Mean   :249.7
##  3rd Qu.:61.00                      3rd Qu.:140.0   3rd Qu.:280.0
##  Max.   :77.00                      Max.   :200.0   Max.   :564.0
##  electrocard max_heart_rate  induc_ang    oldpeak      peak_st_seg
##  0:131       Min.   : 71.0   0:181     Min.   :0.00   Min.   :1.000
##  1:  2       1st Qu.:133.0   1: 89     1st Qu.:0.00   1st Qu.:1.000
##  2:137       Median :153.5             Median :0.80   Median :2.000
##              Mean   :149.7             Mean   :1.05   Mean   :1.585
##               3rd Qu.:166.0             3rd Qu.:1.60   3rd Qu.:2.000
##               Max.   :202.0             Max.   :6.20   Max.   :3.000
##    major_ves       thal       presence
##  Min.   :0.0000   3:152   Min.   :0.0000
##  1st Qu.:0.0000   6: 14   1st Qu.:0.0000
##  Median :0.0000   7:104   Median :0.0000
##  Mean   :0.6704           Mean   :0.4444
##  3rd Qu.:1.0000           3rd Qu.:1.0000
##  Max.   :3.0000           Max.   :1.0000
```

Step 3: Create ROC Dataframe

The lecture notes loops over thresholds (0.01 to 0.99) to compute TPR and FPR for each threshold. Therefore, we try to mimic and create a reusable function compute_roc_df to apply this logic to all three models. We pre-allocate the ROC_DF data frame with 981 rows (for 981 thresholds). Handling edge cases where the confusion matrix might not be 2x2 (e.g., if all predictions are 0 or 1 at a threshold).

Step 4: Fit the GLM Model

- Identify Variable Types: From the variable description table provided earlier:
  - Continuous: age, rest-bp (resting blood pressure), serum-chol (serum cholesterol), max-heart-rate (maximum heart rate), oldpeak, major-vessels.
  - Binary: sex, fasting-blood-sugar, angina.
  - Categorical: chest-pain, electrocardiographic, thal.
  - Integer: slope, heart-disease (target, already converted to 0/1).
- Health Record Variables: The problem specifies age, blood pressure (rest-bp), cholesterol (serum-chol), and maximum heart rate (max-heart-rate).
- Model 1 (Continuous Only): Use only continuous variables: age, rest-bp, serum-chol, max-heart-rate, oldpeak, major-vessels. Fit a glm model with family = "binomial", predict on the same dataset (no train-test split), and compute the ROC curve.
- Model 2 (Health Record Variables): Use age, rest-bp, serum-chol, max-heart-rate. Fit a glm model, predict, and compute the ROC curve.
- Model 3 (All Variables): Use all predictors (heart.disease ~ .), as in the original Step 5. Fit a glm model, predict, and compute the ROC curve. ROC Computation: Use the compute_roc_df function (defined in Step 3) to compute ROC curves for each model.

```
### Fit Three GLM Models
# 1. GLM Model (Continuous Variables Only)
glm_continuous <- glm(presence ~ age + blood_press + serum_chol + max_heart_rate + oldpeak + major_ves,
glm_continuous_pred_prob <- predict(glm_continuous, newdata = heart_data, type = "response")
glm_continuous_roc_df <- compute_roc_df(glm_continuous_pred_prob, heart_data$presence)

# Summary of GLM (Continuous) Model
cat("Summary of GLM Model (Continuous Variables):\n")
```

```
## Summary of GLM Model (Continuous Variables):
```

```
print(summary(glm_continuous))
```

```
##
## Call:
## glm(formula = presence ~ age + blood_press + serum_chol + max_heart_rate +
##     oldpeak + major_ves, family = "binomial", data = heart_data)
```

```
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.703616   1.963147   1.377   0.1685
## age             -0.042482   0.020735  -2.049   0.0405 *
## blood_press      0.017591   0.009460   1.860   0.0630 .
## serum_chol       0.004892   0.003138   1.559   0.1190
## max_heart_rate  -0.037075   0.008365  -4.432 9.34e-06 ***
## oldpeak          0.648066   0.163814   3.956 7.62e-05 ***
## major_ves        1.191850   0.218632   5.451 5.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 370.96  on 269  degrees of freedom
## Residual deviance: 248.85  on 263  degrees of freedom
## AIC: 262.85
##
## Number of Fisher Scoring iterations: 5
```

```r
cat("\n")
```

```r
# 2. GLM Model (Health Record Variables: age, blood_press, serum_chol, max_heart_rate)
glm_health <- glm(presence ~ age + blood_press + serum_chol + max_heart_rate,
                  data = heart_data, family = "binomial")
glm_health_pred_prob <- predict(glm_health, newdata = heart_data, type = "response")
glm_health_roc_df <- compute_roc_df(glm_health_pred_prob, heart_data$presence)

# Summary of GLM (Health Record) Model
cat("Summary of GLM Model (Health Record Variables):\n")
```

```
## Summary of GLM Model (Health Record Variables):
```

```r
print(summary(glm_health))
```

```
##
## Call:
## glm(formula = presence ~ age + blood_press + serum_chol + max_heart_rate,
##     family = "binomial", data = heart_data)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.040158   1.742851   1.744   0.0811 .
## age             -0.001264   0.017410  -0.073   0.9421
## blood_press      0.016924   0.008185   2.068   0.0387 *
## serum_chol       0.004319   0.002669   1.618   0.1056
## max_heart_rate  -0.043573   0.007365  -5.916  3.3e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 370.96  on 269  degrees of freedom
## Residual deviance: 311.32  on 265  degrees of freedom
## AIC: 321.32
```

```
##
## Number of Fisher Scoring iterations: 3
```

```r
cat("\n")
```

```r
# 3. GLM Model (All Variables)
glm_all <- glm(presence ~ ., data = heart_data, family = "binomial")
glm_all_pred_prob <- predict(glm_all, newdata = heart_data, type = "response")
glm_all_roc_df <- compute_roc_df(glm_all_pred_prob, heart_data$presence)

# Summary of GLM (All Variables) Model
cat("Summary of GLM Model (All Variables):\n")
```

```
## Summary of GLM Model (All Variables):
```

```r
print(summary(glm_all))
```

```
##
## Call:
## glm(formula = presence ~ ., family = "binomial", data = heart_data)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.776474   3.180414  -2.131  0.03311 *
## age            -0.015861   0.026397  -0.601  0.54793
## sex1            1.693145   0.570608   2.967  0.00300 **
## chest_pain2     1.237497   0.877396   1.410  0.15842
## chest_pain3     0.495449   0.734142   0.675  0.49976
## chest_pain4     2.310858   0.745210   3.101  0.00193 **
## blood_press     0.024754   0.011960   2.070  0.03848 *
## serum_chol      0.007250   0.004203   1.725  0.08451 .
## blood_sugar1   -0.386603   0.619939  -0.624  0.53288
## electrocard1    0.783375   3.222164   0.243  0.80791
## electrocard2    0.627080   0.408292   1.536  0.12457
## max_heart_rate -0.022179   0.011277  -1.967  0.04920 *
## induc_ang1      0.652215   0.455012   1.433  0.15174
## oldpeak         0.446954   0.246947   1.810  0.07031 .
## peak_st_seg     0.510467   0.402651   1.268  0.20488
## major_ves       1.170873   0.274261   4.269 1.96e-05 ***
## thal6          -0.035841   0.841203  -0.043  0.96602
## thal7           1.452834   0.443952   3.273  0.00107 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 370.96  on 269  degrees of freedom
## Residual deviance: 171.40  on 252  degrees of freedom
## AIC: 207.4
##
## Number of Fisher Scoring iterations: 6
```

```r
cat("\n")
```

Step 5: Fit the GLMNET Model

Follow the lecture note's approach: create a predictor matrix (MatrixPred) and response vector (VecResponse). Using model.matrix to convert categorical variables to dummy variables, as glmnet requires numeric inputs. Fitting a glmnet model with family = "binomial", predicting with s = 0.1. Then compute the ROC curve using compute_roc_df.

```r
## Step 5: Fit the GLMNET Model

# Load the required library
library(glmnet)

MatrixPred <- model.matrix(presence ~ ., data = heart_data)[, -1]
VecResponse <- heart_data$presence
ObjGlmNet1 <- glmnet(MatrixPred, VecResponse, family = "binomial")
glmnet_pred_prob <- predict(ObjGlmNet1, newx = MatrixPred, type = "response", s = 0.1)
glmnet_roc_df <- compute_roc_df(glmnet_pred_prob, heart_data$presence)

# Summary of GLMNET Model
cat("Summary of GLMNET Model:\n")
```

```
## Summary of GLMNET Model:
```

```r
cat("Coefficients at lambda = 0.1:\n")
```

```
## Coefficients at lambda = 0.1:
```

```r
print(coef(ObjGlmNet1, s = 0.1))
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)    -0.558127078
## age             .
## sex1            .
## chest_pain2     .
## chest_pain3     .
## chest_pain4     0.730684378
## blood_press     .
## serum_chol      .
## blood_sugar1    .
## electrocard1    .
## electrocard2    .
## max_heart_rate -0.005024868
## induc_ang1      0.164517210
## oldpeak         0.128724588
## peak_st_seg     .
## major_ves       0.328273695
## thal6           .
## thal7           0.798036364
```

```r
cat("\n")
```

Step 6: Fit the PCA Model

Extract predictors and create a numeric matrix with model.matrix. Perform PCA on the entire dataset using prcomp (base R), scaling the data for consistency. Using the first two principal components as predictors. Fit a glm model on these components (similar to the lecture's focus on logistic regression). Predict on the same data and compute the ROC curve.

```r
## Step 6: PCA Model (First 2 PCs, All Variables)
# Prepare data for PCA (exclude the target variable)
features <- heart_data[, -which(names(heart_data) == "presence")]
features_matrix <- model.matrix(~ . - 1, data = features)

# Perform PCA on all variables using princomp
pca_all <- princomp(features_matrix, cor = TRUE)

# Extract the first two principal components
pca_scores <- pca_all$scores[, 1:2]
pca_data <- data.frame(PC1 = pca_scores[, 1], PC2 = pca_scores[, 2], presence = heart_data$presence)

# Fit GLM on the first two PCs
pca_glm <- glm(presence ~ PC1 + PC2, data = pca_data, family = "binomial")
pca_pred_prob <- predict(pca_glm, newdata = pca_data, type = "response")
pca_roc_df <- compute_roc_df(pca_pred_prob, pca_data$presence)

# Summary of PCA and GLM on PCs
cat("Summary of PCA Model:\n")
```

```
## Summary of PCA Model:
```

```r
cat("Component variances:\n")
```

```
## Component variances:
```

```r
print(pca_all$sdev^2)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 3.4849426 2.2638673 1.5433522 1.3576679 1.2558319 1.0974274 1.0613599 0.9678690
##    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14   Comp.15   Comp.16
## 0.8741922 0.7848741 0.7642903 0.6567903 0.5785908 0.4722699 0.4023954 0.3303459
##   Comp.17   Comp.18
## 0.1039329 0.0000000
```

```r
cat("\nSummary of GLM Model on First Two PCs:\n")
```

```
##
## Summary of GLM Model on First Two PCs:
```

```r
print(summary(pca_glm))
```

```
##
## Call:
## glm(formula = presence ~ PC1 + PC2, family = "binomial", data = pca_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.3724     0.1810  -2.057   0.0397 *
## PC1           1.3340     0.1538   8.674   <2e-16 ***
## PC2          -0.1053     0.1165  -0.904   0.3659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

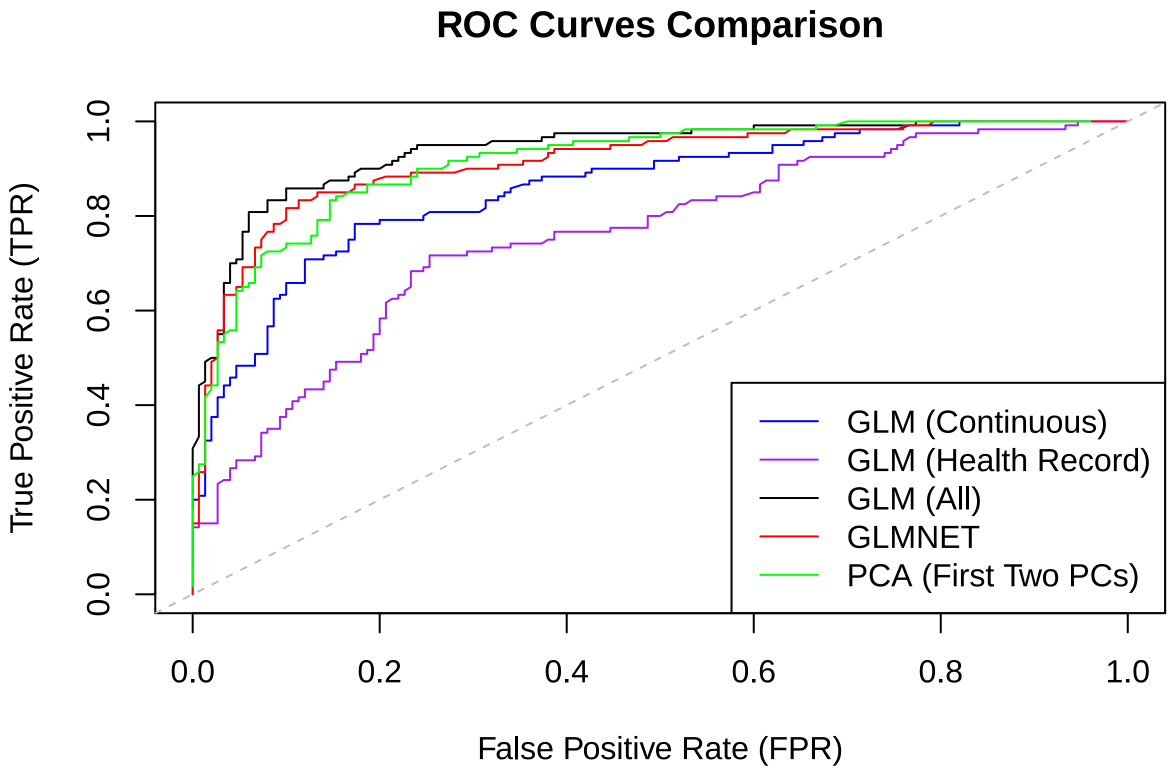
```
##     Null deviance: 370.96  on 269  degrees of freedom
## Residual deviance: 198.91  on 267  degrees of freedom
## AIC: 204.91
##
## Number of Fisher Scoring iterations: 5
```

```
cat("\n")
```

Step 7: Plot the ROC Curves

Plot the GLM model's ROC curve first, then add GLMNET and PCA curves using lines(). Add a diagonal line (abline) to represent a random classifier, as is standard in ROC plots. Include a legend to distinguish the three models, aligning with the template's visualization approach. Ensure xlim and ylim are [0, 1] for a proper ROC plot.

```
## Step 7: Plot ROC Curves
# Plot ROC curves using base R
plot(glm_continuous_roc_df$FPR, glm_continuous_roc_df$TPR, type = "l", col = "blue",
     xlab = "False Positive Rate (FPR)", ylab = "True Positive Rate (TPR)",
     main = "ROC Curves Comparison", xlim = c(0, 1), ylim = c(0, 1))
lines(glm_health_roc_df$FPR, glm_health_roc_df$TPR, col = "purple")
lines(glm_all_roc_df$FPR, glm_all_roc_df$TPR, col = "black")
lines(glmnet_roc_df$FPR, glmnet_roc_df$TPR, col = "red")
lines(pca_roc_df$FPR, pca_roc_df$TPR, col = "green")
abline(a = 0, b = 1, lty = 2, col = "gray")
legend("bottomright", legend = c("GLM (Continuous)", "GLM (Health Record)", "GLM (All)",
                                 "GLMNET", "PCA (First Two PCs)"),
       col = c("blue", "purple", "black", "red", "green"), lty = 1)
```


ROC Curves Comparison
True Positive Rate (TPR)
False Positive Rate (FPR)
GLM (Continuous)
GLM (Health Record)
GLM (All)
GLMNET
PCA (First Two PCs)