CS5525 Project

Jennifer Appiah-Kubi, Rebecca DeSipio, Ajinkya Fotedar

11/30/2021

Contents

Introduction

- In this project, we will be trying to predict the probability of having a heart attack using 14 variables available in the `hearts.csv` data-set.
- Techniques employed for model fit, analysis and interpretation, and visualization:
  - Classification
    1. Logistic Regression
    2. Decision trees
    3. Support Vector Machines
  - Sparse Regression
    1. LASSO
    2. Elastic Net
- Libraries used:
  1. `glmnet`
  2. `tree`
  3. `randomForest`

Data-set

```
# reading data
setwd("/Users/ajinkyafotedar/CS5525/Project/CS5525-Final-Project")
heart <- read.csv("heart.csv")

# observations
dim(heart)
```

```
## [1] 303  14
```

```
# attributes
names(heart)
```

```
##  [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"
```

Attribute Information

- age
- sex
- chest pain type (4 values)
- resting blood pressure
- serum cholesterol in mg/dl
- fasting blood sugar > 120 mg/dl
- resting electrocardiograph results (values 0, 1, 2)
- maximum heart rate achieved
- exercise induced angina
- old peak = ST depression induced by exercise relative to rest
- the slope of the peak exercise ST segment
- number of major vessels (0 - 3) colored by fluoroscope
- thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
- target: 0 = less chance of heart attack; 1 = more chance of heart attack

Splitting Into Train and Test

```
set.seed(123)
grid = 10^seq(10, -2, length = 100)

X <- data.matrix(heart[, c("age", "sex", "cp", "trestbps", "chol", "fbs",
                           "restecg", "thalach", "exang", "oldpeak", "slope",
                           "ca", "thal")
                       ]
                 )
y <- heart$target

n = nrow(X)
train_rows <- sample(1:n, n * 0.7)

X.train <- X[train_rows,]
X.test <- X[-train_rows,]
y.train <- y[train_rows]
y.test <- y[-train_rows]
```

```
dim(X.train)
```

```
## [1] 212  13
```

```
dim(X.test)
```

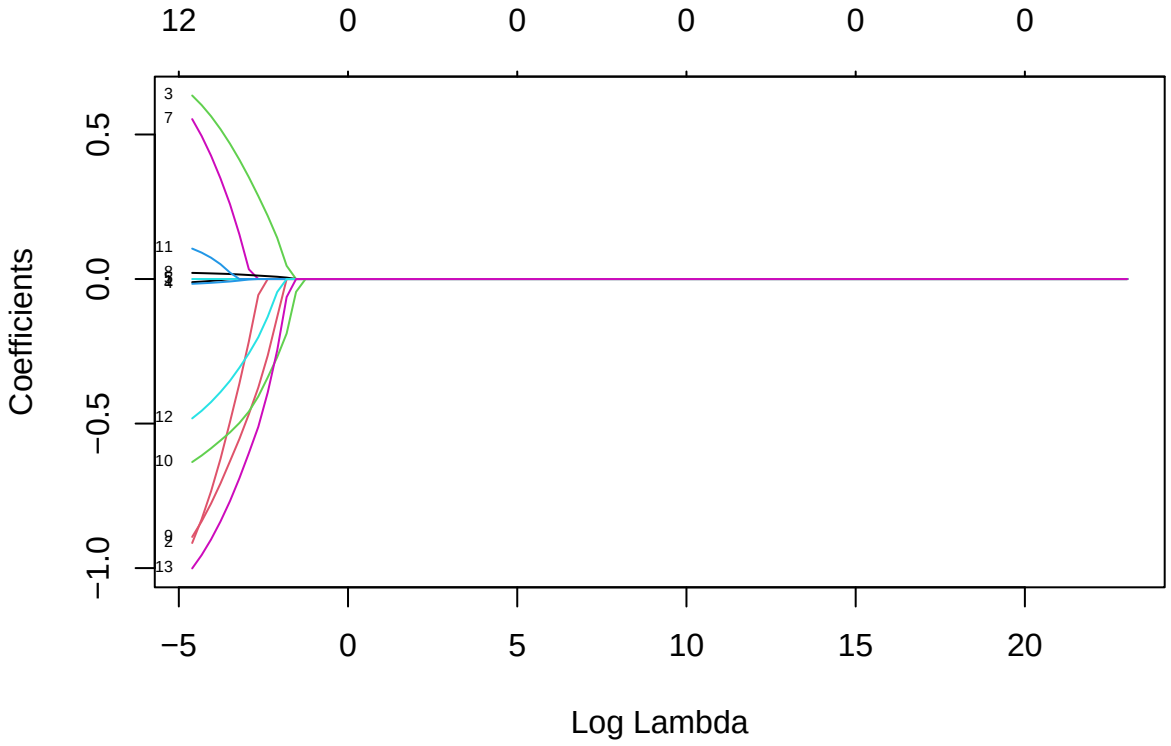
```
## [1] 91 13
```

Sparse Regression Methods

Lasso

```
library(glmnet)

# lasso model
lasso.mod <- glmnet(X.train, y.train, alpha = 1, lambda = grid,
                    family = "binomial")
plot(lasso.mod, xvar = "lambda", label = T)
```




```
# cross-validation for lambda
cv.out <- cv.glmnet(X.train, y.train, alpha = 1)
bestlam <- cv.out$lambda.min

# test error
lasso.pred <- predict(lasso.mod, s = bestlam, newx = X.test)
lasso.mse <- mean((lasso.pred - y.test)^2)
lasso.mse
```

```
## [1] 4.55179
```

```r
# non-zero coefficients
lasso.coef <- predict(lasso.mod, type = "coefficients", s = bestlam)
lasso.coef <- lasso.coef[which(lasso.coef != 0)]
lasso.coef
```

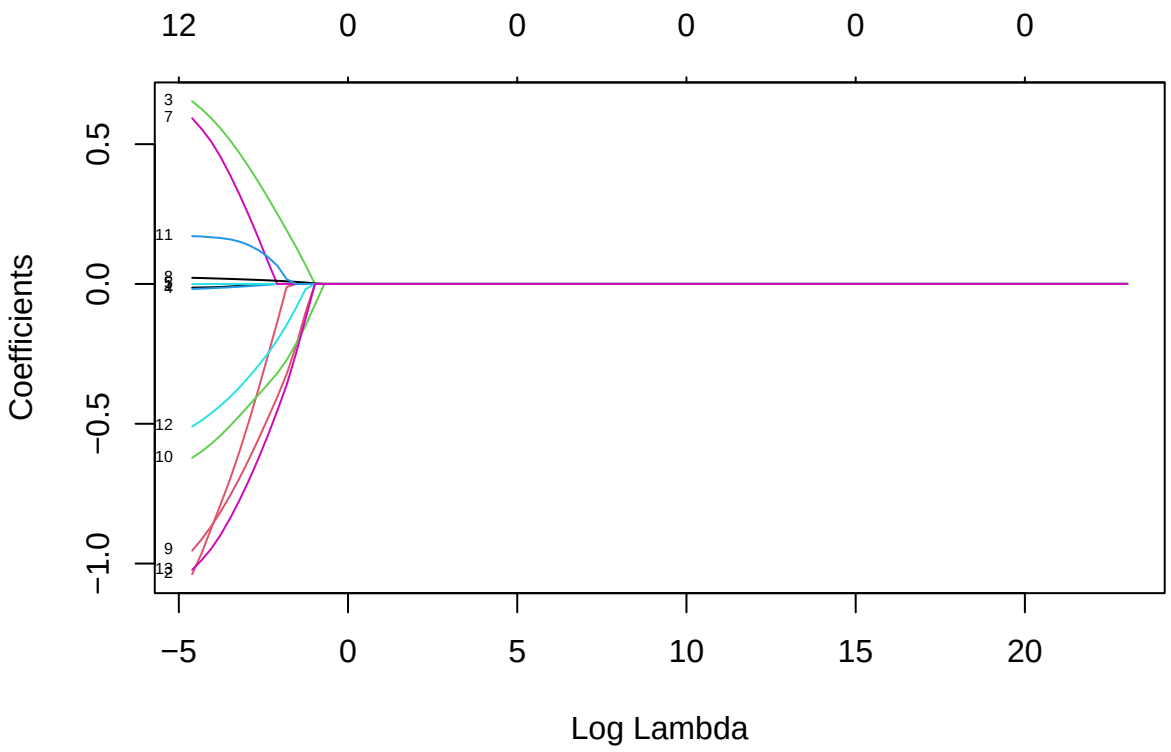
```
##  [1]  2.994438e+00 -1.090841e-02 -9.131960e-01  6.348623e-01 -1.647880e-02
##  [6] -1.051699e-05  5.532416e-01  2.124596e-02 -8.921111e-01 -6.330783e-01
## [11]  1.052623e-01 -4.821241e-01 -1.001051e+00
```

```r
# coefficients of the best model
best.lasso.mod <- glmnet(X.train, y.train, alpha = 1, lambda = bestlam,
                         family = "binomial")
coef(best.lasso.mod)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)  3.2596709630
## age         -0.0116090143
## sex         -0.9868012630
## cp           0.6565847650
## trestbps    -0.0175641434
## chol        -0.0004366527
## fbs          .
## restecg      0.5856803928
## thalach      0.0220138928
## exang       -0.9295042006
## oldpeak     -0.6468163818
## slope        0.1167544074
## ca          -0.5013939176
## thal        -1.0271678499
```

Elastic Net

```r
# elastic net model
en.mod <- glmnet(X.train, y.train, alpha = 0.5, lambda = grid,
                 family = "binomial")
plot(en.mod, xvar = "lambda", label = T)
```




```
# cross-validation for lambda (with a fixed alpha)
cv.out <- cv.glmnet(X.train, y.train, alpha = 0.5)
bestlam <- cv.out$lambda.min

# test error
en.pred <- predict(en.mod, s = bestlam, newx = X.test)
en.mse <- mean((en.pred - y.test)^2)
en.mse
```

```
## [1] 3.516643
```

```
# non-zero coefficients
en.coef <- predict(en.mod, type = "coefficients", s = bestlam)
en.coef <- en.coef[which(en.coef != 0)]
en.coef
```

```
##  [1]  2.49856200 -0.01066523 -0.78777204  0.55570522 -0.01361220  0.45376731
##  [7]  0.01909560 -0.81416735 -0.54126897  0.16413223 -0.43462839 -0.89688306
```

```
# coefficients of the best model
best.en.mod <- glmnet(X.train, y.train, alpha = 0.5, lambda = bestlam,
                      family = "binomial")
coef(best.en.mod)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)  2.49796172
## age         -0.01066195
## sex         -0.78764834
## cp           0.55563347
## trestbps    -0.01360960
## chol         .
```

```
## fbs          .
## restecg      0.45365860
## thalach      0.01909390
## exang       -0.81407451
## oldpeak     -0.54119334
## slope        0.16415637
## ca          -0.43458297
## thal        -0.89679218
```

Classification Methods

Logistic Regression

```
# splitting data
target <- as.factor(heart$target)
train <- sample(1:nrow(heart), 0.75 * nrow(heart))

heart.train <- heart[train, ]
heart.test <- heart[-train, ]

# training data
dim(heart.train)
```

```
## [1] 227  14
```

```
# testing data
dim(heart.test)
```

```
## [1] 76 14
```

Decision Trees

Support Vector Machines

Analysis

Conclusion