CS5525 Project

Jennifer Appiah-Kubi, Rebecca DeSipio, Ajinkya Fotedar

11/30/2021

Contents

Introduction **1**
Data-set . . . 2
Attribute Information . . . 2
Splitting Into Train and Test . . . 2

Sparse Regression Methods **3**
Lasso . . . 3
Elastic Net . . . 4

Classification Methods **6**
Logistic Regression . . . 6
Decision Trees . . . 6
Support Vector Machines . . . 6

Analysis **6**

Conclusion **6**

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)
```

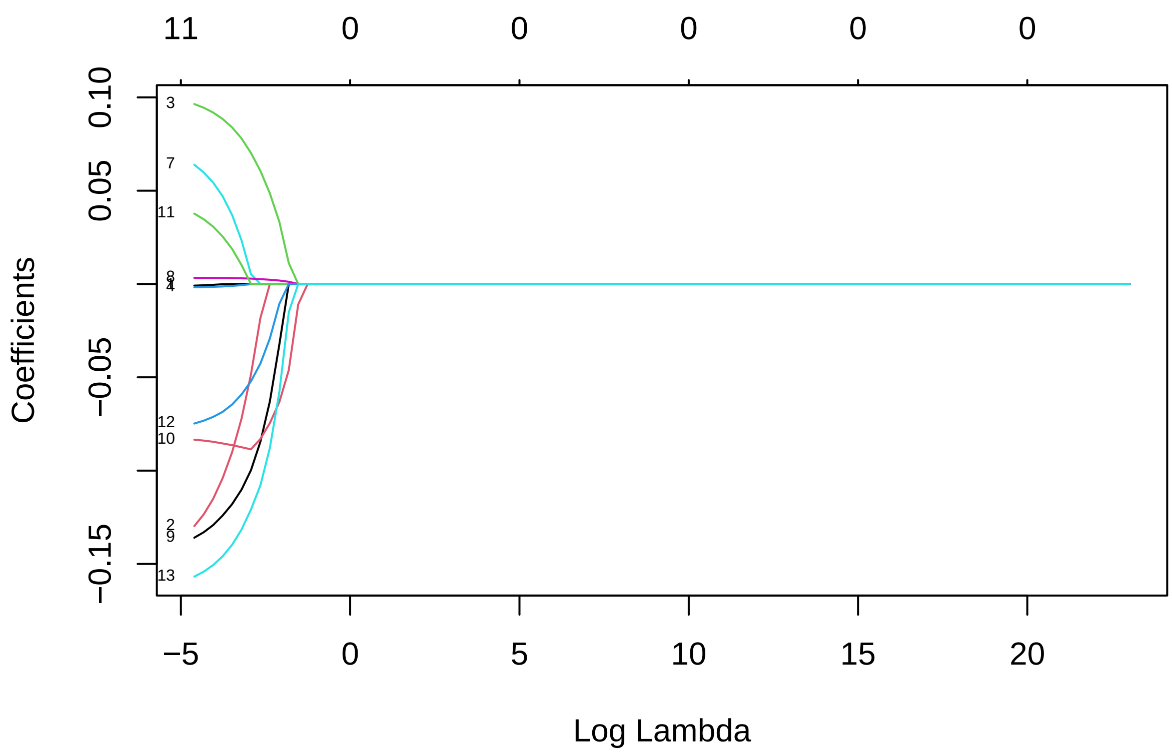
```
## [1] 91 13
```

Sparse Regression Methods

Lasso

```
library(glmnet)

# lasso model
lasso.mod <- glmnet(X.train, y.train, alpha = 1, lambda = grid)
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] 0.1186403
```

```
# non-zero coefficients
lasso.coef <- predict(lasso.mod, type = "coefficients", s = bestlam)
```

```r
lasso.coef <- lasso.coef[which(lasso.coef != 0)]
lasso.coef
```

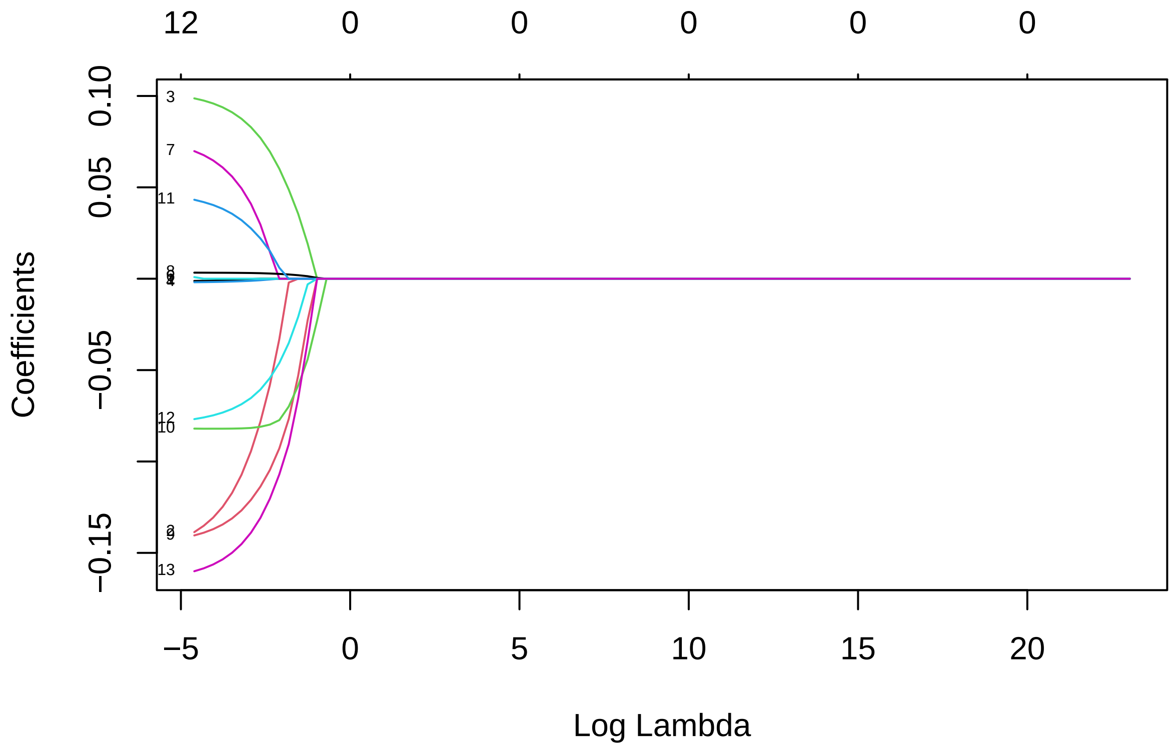
```
##  [1]  0.7987175429 -0.0008647686 -0.1297485211  0.0964194286 -0.0017732604
##  [6]  0.0639164913  0.0033041257 -0.1359563250 -0.0834206234  0.0377118027
## [11] -0.0747732294 -0.1568133651
```

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

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  0.8152081113
## age         -0.0009834503
## sex         -0.1336888936
## cp           0.0976758534
## trestbps    -0.0018418380
## chol         .
## fbs          .
## restecg      0.0664503348
## thalach      0.0033126533
## exang       -0.1377144029
## oldpeak     -0.0831180261
## slope        0.0395829330
## ca          -0.0757097260
## thal        -0.1584609241
```

Elastic Net

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




```r
# 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] 0.118814
```

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

```
##  [1]  0.7848392017 -0.0008879291 -0.1247170099  0.0937548204 -0.0016822065
##  [6]  0.0608248014  0.0032543766 -0.1344398231 -0.0820534870  0.0381919105
## [11] -0.0732034128 -0.1535118232
```

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

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  0.7848040647
## age         -0.0008891907
## sex         -0.1247444716
## cp           0.0937854581
## trestbps    -0.0016826766
## chol          .
```

```
## fbs          .
## restecg     0.0608100211
## thalach     0.0032551311
## exang      -0.1343918929
## oldpeak    -0.0820439608
## slope       0.0381966321
## ca         -0.0731938545
## thal       -0.1535077270
```

Classification Methods

Logistic Regression

```r
# 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
```

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

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

Decision Trees

Support Vector Machines

Analysis

Conclusion