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.
- Classification techniques employed for model fit, analysis, interpretation, and visualization:
  1. Penalized Logistic Regression
     - LASSO
     - Elastic Net
  2. Decision trees
     - Random Forest
     - Bootstrapping
  3. Support Vector Machines
  4. KNN
- Libraries used:
  1. `glmnet`
  2. `tree`
  3. `randomForest`

Data-set

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

# observations
dim(heart)
```

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

```r
# 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

```r
set.seed(123)

X <- as.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
```

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

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

Classification Methods

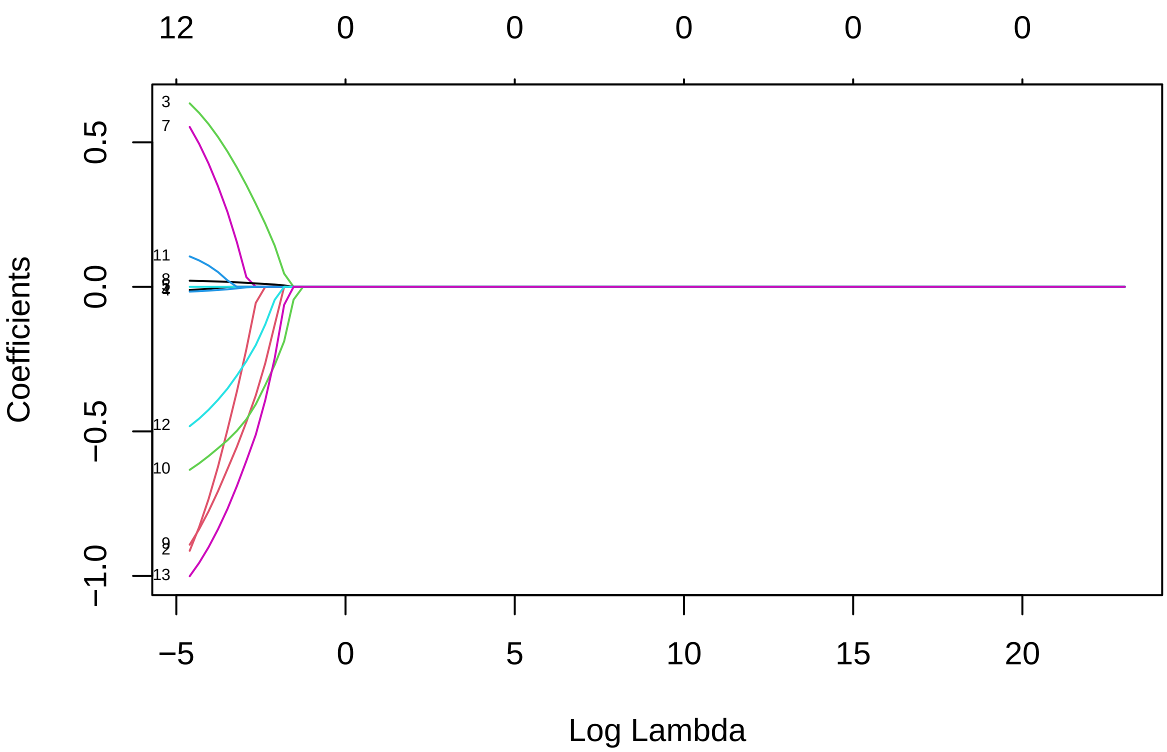
1. Penalized Logistic Regression

1.1 Lasso

```r
library(glmnet)

grid <- 10^seq(10, -2, length = 100)

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




```r
# cross-validation for lambda
cv.out <- cv.glmnet(X.train, as.factor(y.train), family = "binomial", alpha = 1,
                    type.measure = "class")
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.0323212
```

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

```r
coef(best.lasso.mod)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)  1.176776502
## age          .
## sex         -0.467569833
## cp           0.457343243
## trestbps    -0.007852149
## chol         .
## fbs          .
## restecg      0.239425545
## thalach      0.017066141
## exang       -0.614708498
## oldpeak     -0.524761483
## slope        0.015640883
## ca          -0.343435426
## thal        -0.752920629
```

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

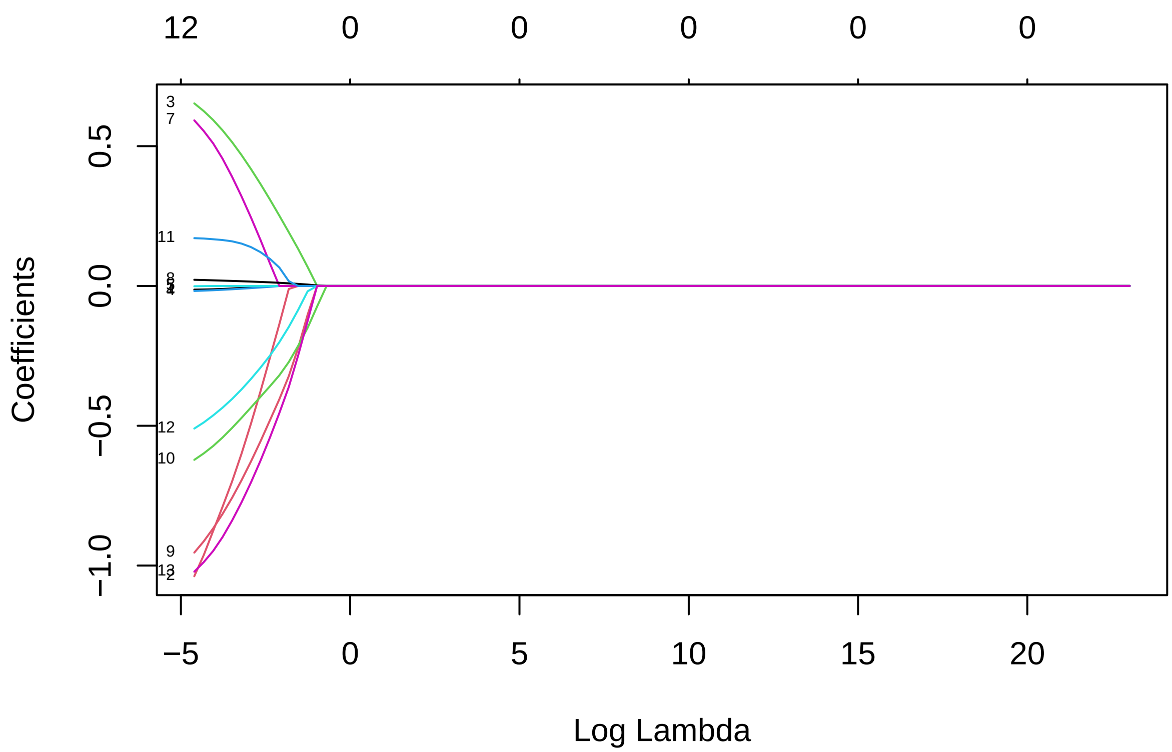
```
## [1] 2.099165
```

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

```
##  [1]  1.176776502 -0.467569833  0.457343243 -0.007852149  0.239425545
##  [6]  0.017066141 -0.614708498 -0.524761483  0.015640883 -0.343435426
## [11] -0.752920629
```

1.2 Elastic Net

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

# coefficients of the best model
best.en.mod <- glmnet(X.train, as.factor(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
```

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

```
## [1] 3.515688
```

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

```
##  [1]  2.49796172 -0.01066195 -0.78764834  0.55563347 -0.01360960  0.45365860
##  [7]  0.01909390 -0.81407451 -0.54119334  0.16415637 -0.43458297 -0.89679218
```

2. Decision Trees

3. Support Vector Machines

4. KNN

Analysis

Conclusion