

## Environment Setup

## Load and Prepare Data

```
# Load datasets: BreastCancer for classification and BostonHousing for regression
data("BreastCancer", package = "mlbench")
data("BostonHousing", package = "mlbench")

# Remove any rows with missing values and unnecessary ID column in BreastCancer data
BreastCancer <- na.omit(BreastCancer)[,-c(1)]
```

## Split BreastCancer Data into Train and Test Sets

```
set.seed(1)

# Split data: 70% for training, 30% for testing
index <- sample(1:nrow(BreastCancer), 0.7 * nrow(BreastCancer), replace = FALSE)
train <- BreastCancer[index,]
test <- BreastCancer[-index,]

# Prepare matrices for ridge logistic regression
X.train <- model.matrix(Class ~ ., data = train)[,-1] # Training predictors
y.train <- as.numeric(train$Class) - 1 # Convert Class to binary numeric (0/1)

X.test <- model.matrix(Class ~ ., data = test)[,-1] # Test predictors
y.test <- as.numeric(test$Class) - 1 # Convert Class to binary numeric (0/1)
```

## Ridge Logistic Regression with Cross-Validation on BreastCancer Data

```
# Perform cross-validated ridge regression (alpha = 0) with logistic family
cvfit.ridge <- cv.glmnet(x = X.train, y = y.train, alpha = 0, family = 'binomial')

# Display lambda that minimizes cross-validation error and corresponding coefficients
cvfit.ridge$lambda.min
```

```
## [1] 0.04004231
```

```
coef(cvfit.ridge, s = "lambda.min")
```

```
## 81 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept)    0.2722211132
## Cl.thickness.L 1.6011443145
## Cl.thickness.Q 0.2835974728
## Cl.thickness.C 0.0254030388
## Cl.thickness^4 -0.1774049377
## Cl.thickness^5 0.4849037481
## Cl.thickness^6 0.2079733872
## Cl.thickness^7 0.2237798858
## Cl.thickness^8 -0.2476795547
## Cl.thickness^9 -0.0002640986
## Cell.size.L    0.9914608106
## Cell.size.Q    -0.6582327958
## Cell.size.C    0.5552385247
## Cell.size^4    0.3490934124
## Cell.size^5    0.6981130142
## Cell.size^6    -0.2584391281
## Cell.size^7    0.3635266206
## Cell.size^8    0.6186132180
## Cell.size^9    -0.0964206595
## Cell.shape.L   1.2043049506
## Cell.shape.Q   -0.8782329562
## Cell.shape.C   0.6225655203
## Cell.shape^4   0.2358175477
## Cell.shape^5   -0.3857572229
## Cell.shape^6   0.0789281267
## Cell.shape^7   -0.0642125747
## Cell.shape^8   0.5483016220
## Cell.shape^9   0.4704234869
## Marg.adhesion.L 0.9935910817
## Marg.adhesion.Q -0.4660589395
## Marg.adhesion.C 0.4315413843
## Marg.adhesion^4 -0.0919270861
## Marg.adhesion^5 0.3324943672
## Marg.adhesion^6 0.1189093114
## Marg.adhesion^7 0.1816851817
## Marg.adhesion^8 0.1573494739
## Marg.adhesion^9 0.3997666956
## Epith.c.size.L 0.7275234626
## Epith.c.size.Q -0.4460424175
## Epith.c.size.C 0.6169381888
## Epith.c.size^4 0.6562179241
## Epith.c.size^5 -0.7921824653
## Epith.c.size^6 0.4487549708
## Epith.c.size^7 -0.2052564122
## Epith.c.size^8 0.3942738933
## Epith.c.size^9 0.4135122517
## Bare.nuclei2   -0.0859184136
## Bare.nuclei3   -0.0618340947
## Bare.nuclei4    0.2612703955
## Bare.nuclei5    0.5898049125
## Bare.nuclei6    2.2402599475
## Bare.nuclei7    1.7116280104
```

```
## Bare.nuclei8      0.4349369985
## Bare.nuclei9      1.7930596960
## Bare.nuclei10     1.0171633643
## Bl.cromatin2      -0.2418442248
## Bl.cromatin3      -0.1062174370
## Bl.cromatin4       0.4671598073
## Bl.cromatin5       0.8931431140
## Bl.cromatin6       0.7487184487
## Bl.cromatin7       0.5166567977
## Bl.cromatin8       0.6937902829
## Bl.cromatin9       0.6550058542
## Bl.cromatin10      0.6323031530
## Normal.nucleoli2  -0.7321689741
## Normal.nucleoli3   0.6339642625
## Normal.nucleoli4   0.5847179661
## Normal.nucleoli5   0.3765097383
## Normal.nucleoli6   0.3293308202
## Normal.nucleoli7  -0.1863298490
## Normal.nucleoli8   0.2984631157
## Normal.nucleoli9   1.2597848206
## Normal.nucleoli10  0.8129453169
## Mitoses2           0.7835081161
## Mitoses3           1.0350598932
## Mitoses4           0.5372892004
## Mitoses5           2.3723353980
## Mitoses6           0.1175690865
## Mitoses7           0.7791007420
## Mitoses8           0.3436351993
## Mitoses10          0.4721734269
```

## Prepare BostonHousing Data for Lasso Regression

```
# Set up design matrix for BostonHousing regression (exclude intercept) and response vector
X <- model.matrix(medv ~ ., data = BostonHousing)[,-1]
y <- BostonHousing$medv
```

## Lasso Regression with Cross-Validation on BostonHousing Data

```
set.seed(202211)

# Perform cross-validated lasso regression (alpha = 1)
cvfit.house.lasso <- cv.glmnet(x = X, y = y, alpha = 1)
# Display lambda that minimizes cross-validation error
cvfit.house.lasso$lambda.min
```

```
## [1] 0.009170489
```

```
# Display the coefficients associated with lambda.min  
coef(cvfit.house.lasso, s = "lambda.min")
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"  
##              s1  
## (Intercept) 35.707318035  
## crim        -0.104719398  
## zn           0.044527873  
## indus        0.007458193  
## chas1        2.698073139  
## nox          -17.125645162  
## rm           3.829379107  
## age          .  
## dis          -1.454866978  
## rad           0.285276396  
## tax          -0.011295472  
## ptratio      -0.942664365  
## b             0.009211899  
## lstat        -0.523075319
```