Homework 8: Modeling Crime Rates with Stepwise, Lasso, and Elastic Net Regression

Georgia Institute of Technology, Business Analytics

Introduction to Analytics Modeling

Professor Joel Sokol

October 16, 2025

Files submitted: homework8_answers.pdf (this doc), homework8.R

Question 11.1

Using the crime data set `uscrime.txt` from Questions 8.2, 9.1, and 10.1, build a regression model using:

1. **Stepwise regression**
2. **Lasso**
3. **Elastic net**

For Parts 2 and 3, remember to scale the data first – otherwise, the regression coefficients will be on different scales and the constraint won't have the desired effect.

For Parts 2 and 3, use the `glmnet` function in R.

Notes on R:

- **For the elastic net model, what we called λ in the videos, `glmnet` calls "alpha"; you can get a range of results by varying alpha from 1 (lasso) to 0 (ridge regression) [and, of course, other values of alpha in between].**
- **In a function call like `glmnet(x,y,family="mgaussian",alpha=1)` the predictors x need to be in R's matrix format, rather than data frame format. You can convert a data frame to a matrix using `as.matrix` – for example, `x <- as.matrix(data[,1:n-1])`**
- **Rather than specifying a value of T, glmnet returns models for a variety of values of T.**

Methodology

This question required building three regression models using the `uscrime.txt` dataset:

1. **Stepwise Regression**

- **Tool Used**: `stepAIC()` from the `MASS` package in R.
- **Approach**: Started with a full linear model using all predictors. Stepwise selection was performed in both directions (forward and backward) using AIC as the criterion.
- **Evaluation**: Final model selected based on lowest AIC value.

2. **Lasso Regression**

- **Tool Used**: `glmnet()` with `alpha = 1`.
- **Preprocessing**: Predictors were standardized using `scale()` to ensure consistent penalization across features.
- **Evaluation**: Used `cv.glmnet()` to perform cross-validation and select the optimal λ (`lambda.min`). Coefficients were extracted at this value.

3. **Elastic Net Regression**

- **Tool Used**: `glmnet()` with `alpha = 0.5`.

- **Preprocessing**: Same standardized predictors as Lasso.
- **Evaluation**: Used `cv.glmnet()` to select optimal λ. Coefficients were extracted at `lambda.min`.

Results

Stepwise Regression Output:

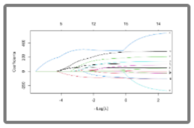
Final Model: `Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob`
Adjusted R^2: 0.7444 **Significant Predictors**:

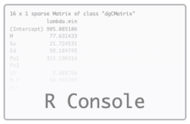
- Po1 ($p < 0.001$)
- Ed, Ineq, Prob, M, U2 ($p < 0.05$)
- U1 (marginal, $p \approx 0.076$)
- M.F (not significant)

```
Call:
lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
    data = uscrime)

Residuals:
    Min      1Q  Median      3Q     Max 
-444.70 -111.07    3.03  122.15  483.30 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
M              93.32      33.50   2.786  0.00828 ** 
Ed            180.12      52.75   3.414  0.00153 ** 
Po1           102.65      15.52   6.613 8.26e-08 ***
M.F            22.34      13.60   1.642  0.10874    
U1          -6086.63    3339.27  -1.823  0.07622 .  
U2            187.35      72.48   2.585  0.01371 *  
Ineq           61.33      13.96   4.394 8.63e-05 ***
Prob        -3796.03    1490.65  -2.547  0.01505 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 195.5 on 38 degrees of freedom
Multiple R-squared:  0.7888,	Adjusted R-squared:  0.7444 
F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

Figure 1: Summary output *of* stepwise_model showing coefficients, p-values, R^2, and AIC.

Lasso Regression Output:
Selected Predictors (non-zero coefficients at `lambda.min`):
- M, So, Ed, Po1, LF, M.F, NW, U2, Ineq, Prob **Zeroed Out**:
- Po2, Pop, U1, Wealth, Time





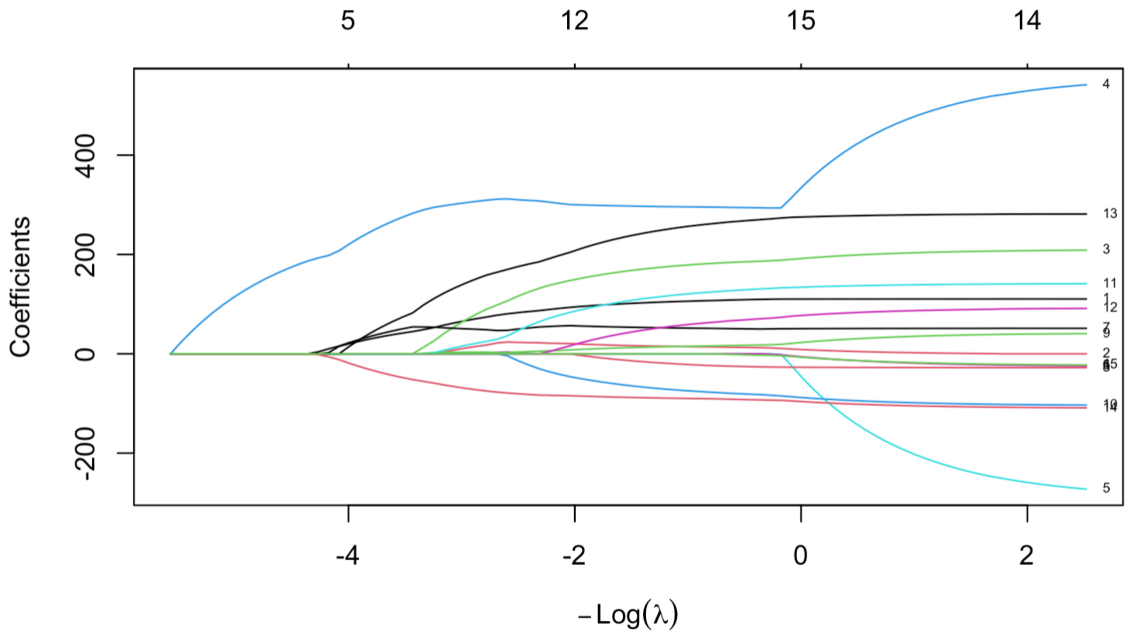


Figure 2: Lasso coefficient path plot — the one with colored lines and "-Log(λ)" on the x-axis.

```r
# Prepare data
x <- as.matrix(uscrime[, -which(names(uscrime) == "Crime")])  # predictors
y <- uscrime$Crime                                             # response

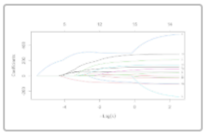
# Standardize predictors
x_scaled <- scale(x)

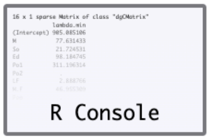
# Fit Lasso model
lasso_model <- glmnet(x_scaled, y, alpha = 1)

# Plot coefficient paths
plot(lasso_model, xvar = "lambda", label = TRUE)

# Cross-validation to choose lambda
cv_lasso <- cv.glmnet(x_scaled, y, alpha = 1)
best_lambda_lasso <- cv_lasso$lambda.min

# Coefficients at best lambda
coef(cv_lasso, s = "lambda.min")
```






```
16 x 1 sparse Matrix of class "dgCMatrix"
             lambda.min
(Intercept) 905.085106
M            77.631433
So           21.724531
Ed           98.184745
Po1         311.196314
Po2           .
LF            2.888766
M.F          46.955309
Pop           .
NW            2.653843
U1            .
U2           29.301947
Wealth        .
Ineq        164.141688
Prob        -77.051716
Time          .
```

Figure 3: Output of coef(cv_lasso, s = "lambda.min") — showing which variables were retained.

Elastic Net Regression Output:

Selected Predictors:

- M, So, Ed, Po1, Po2, LF, M.F, Pop, NW, U1, U2, Wealth, Ineq, Prob

Zeroed Out:

- Time

```
```{r}
# Try alpha = 0.5 for a balance between Lasso and Ridge
elastic_net_model <- glmnet(x_scaled, y, alpha = 0.5)

# Cross-validation
cv_enet <- cv.glmnet(x_scaled, y, alpha = 0.5)
best_lambda_enet <- cv_enet$lambda.min

# Coefficients at best lambda
coef(cv_enet, s = "lambda.min")

```
```

```
16 x 1 sparse Matrix of class "dgCMatrix"
              lambda.min
(Intercept) 905.08510638
M            91.96372970
So           22.64067510
Ed          143.78265810
Po1         246.25293575
Po2          38.57732124
LF            0.81136213
M.F          63.04456652
Pop          -0.08049603
NW           16.44317612
U1          -54.78094946
U2           91.04598541
Wealth       29.33322887
Ineq        198.25493057
Prob        -86.92790851
Time          .
```

Figure 4: Output of coef(cv_enet, s = "lambda.min") — showing coefficients at optimal λ.

Discussion of Results

Insights:

- **Po1**, **Ed**, and **Ineq** consistently emerged as strong predictors across all models, suggesting a robust relationship with crime rates.
- **Stepwise regression** offered a clean, interpretable model with high explanatory power.
- **Lasso** effectively performed variable selection, shrinking less informative coefficients to zero.
- **Elastic Net** retained more variables, useful when predictors are correlated (e.g., Po1 and Po2).

Surprises:

- **U1** was dropped in Lasso but retained in Stepwise and Elastic Net, indicating sensitivity to penalization.
- **Time** was consistently dropped, suggesting limited predictive value in this dataset.

Improvements:

- Future models could explore interaction terms or nonlinear relationships.
- Consider using external socioeconomic data to enrich the feature set.

Ethical Considerations:

- Predictive models on crime data must be interpreted cautiously to avoid reinforcing biases.
- Variables like race, income, and education should be contextualized to avoid misrepresentation.

REFERENCES

https://ieeexplore.ieee.org/document/10937580