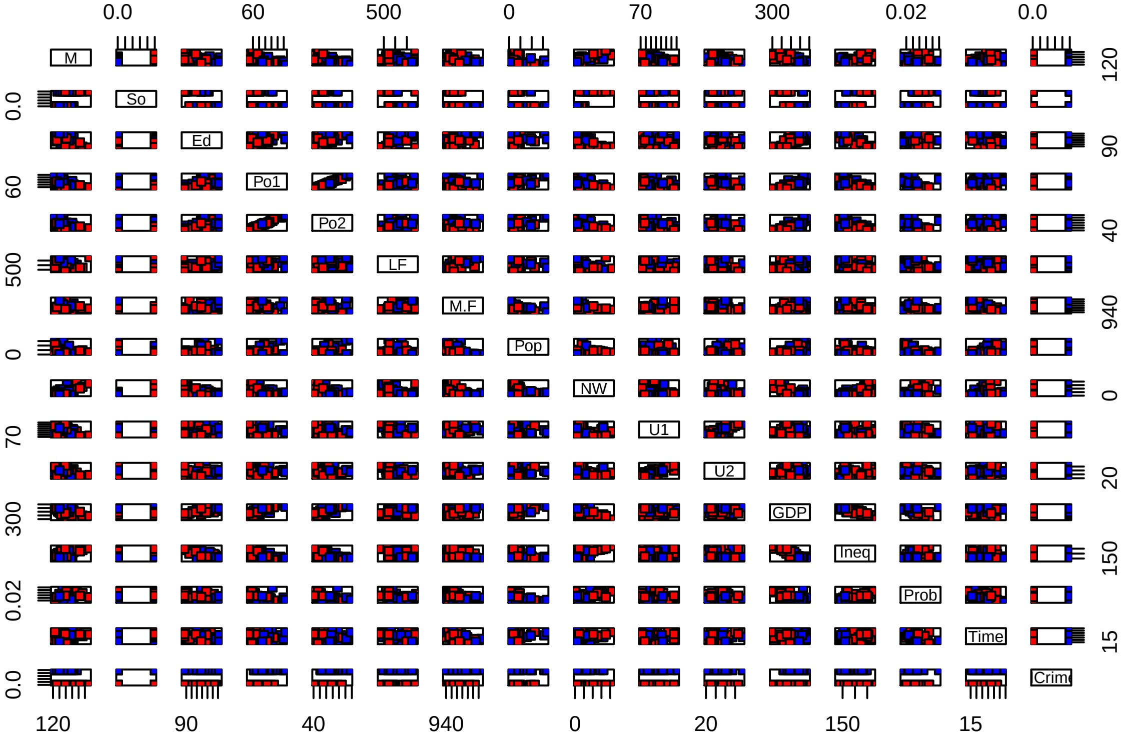
MRR_2019_TP3

Jingzhuo HUI, You ZUO

2019/10/27

logistic regression model

First of all, we plotted the data with a scatterplot where the target variable medCrimeBin is represented using the color red for 0 and blue for 1.



From the scatter plot above we can see that binomial variable medCrime has a distribution of given values 1 or 2, and some of the two variables have some kind of linear relation.

we computed a logistic regression model of all the given variables.

```
##
## Call:
## glm(formula = medCrimeBin ~ ., family = binomial, data = UScrime)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.82997  -0.46541  -0.03231   0.51667   2.01060
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -72.241606  35.027247  -2.062   0.0392 *
## M             0.036547   0.083705   0.437   0.6624
## So            1.691206   2.382900   0.710   0.4779
## Ed            0.178178   0.122473   1.455   0.1457
## Po1           0.125344   0.205847   0.609   0.5426  
## Po2          -0.036477   0.207977  -0.175   0.8608  
## LF            0.001646   0.019740   0.083   0.9335  
## M.F           0.005085   0.037851   0.134   0.8931  
## Pop          -0.028223   0.024505  -1.152   0.2494  
## NW            0.007698   0.009569   0.804   0.4212  
## U1            0.056896   0.074787   0.761   0.4468  
## U2           -0.044563   0.146422  -0.304   0.7609  
## GDP           0.026175   0.020549   1.274   0.2027  
## Ineq          0.085517   0.050786   1.684   0.0922 .
## Prob        -36.040211  34.448007  -1.046   0.2955  
## Time          0.065725   0.108316   0.607   0.5440  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 65.135  on 46  degrees of freedom
## Residual deviance: 32.278  on 31  degrees of freedom
## AIC: 64.278
## 
## Number of Fisher Scoring iterations: 6
```

```
##       esti_y
## real_y  0  1
##      0 20  4
##      1  3 20
```

```
## [1] "mapt = 0.5, performance=0.851064, error=0.148936, FP=0.166667, FN=0.833333"
```

From the results above we can see that, the global performence is satisfied in some kind, but the target is not that significant with the variables. And since we have already noticed the linearity of some of our variables, we decide to use the model selection approaches to pick up the best variables.

Statistical approach

The value of the criterian AIC of approach forward, backward and stepwise, and the coefficients selected for each model are:

```
## [1] 51.03008
```

```
## [1] 50.46298
```

```
## [1] 50.46298
```

```
## medCrimeBin ~ Po2 + M.F + So + Prob
```

```
## medCrimeBin ~ M + Ed + Po1 + U1 + GDP + Ineq
```

```
## medCrimeBin ~ M + Ed + Po1 + U1 + GDP + Ineq
```

From the three methods we would like to choose the one with the smallest AIC, which is got by backward or stepwise, with variables selected: M, Ed, Po1, U1, GDP and Ineq.

Let’s see the capacity our this model:

```
## 
## Call:
## glm(formula = formula(resback), data = UScrime)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.64647  -0.33768  -0.07895   0.34551   0.72314
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.641027   2.125799  -3.124  0.00331 **
## M            0.009814   0.006858   1.431  0.16020
## Ed           0.016498   0.008851   1.864  0.06968 .
## Po1          0.008442   0.003504   2.409  0.02069 *
## U1           0.004372   0.003556   1.229  0.22608
## GDP          0.002538   0.001858   1.366  0.17967
## Ineq         0.008037   0.003603   2.231  0.03138 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1722509)
##
##     Null deviance: 11.745  on 46  degrees of freedom
## Residual deviance:  6.890  on 40  degrees of freedom
## AIC: 59.137
##
## Number of Fisher Scoring iterations: 2
```

The significance has been improved, we are going to evaluate the predective power of this model.

```
##       esti_y
## real_y  0  1
##      0 22  2
##      1  8 15
```

```
## [1] "mapt = 0.5, performance=0.787234, error=0.212766, FP=0.083333, FN=0.916667"
```

According to the results above we can see that even though we have reduced the number of variables, but the performance and the error are worse than before. Remarking the FN is great higher than FP, which means we have a tendency to predict a relatively high level of crime rate place as a low crime rate one, and that could be very unrational to get this model into practice. So we could perhaps adjust the MAPT a little bit smaller to improve it.

```
##       esti_y
## real_y  0  1
##      0 20  4
##      1  4 19
```

```
## [1] "mapt = 0.45, performance=0.829787, error=0.170213, FP=0.166667, FN=0.833333"
```

After adjusting MAQT equal to 0.45, we have a better performance than before.

Logistic regression with l2 and l1 penalization

Ridge

Instead of using the statistical approach, we are going to apply firstly $\updownarrow_{\in}$ penalization and then $\updownarrow_{\infty}$ to regularize our model to constrain the variance of estimator and improve the prediction performance.

```
## Loading required package: Matrix
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-18
```

First we selected a λ through 10-fold cross-validation with the minimum rule, for which the coefficients of the model are:

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -2.9939239270
## M           -0.0026052869
## So           0.0863879501
## Ed           0.0061540386
## Po1          0.0059508818
## Po2          0.0058617978
## LF           0.0001321462
## M.F          0.0006720311
## Pop          0.0014764830
## NW           0.0004971618
## U1          -0.0007258219
## U2           0.0027035716
## GDP          0.0011950445
## Ineq        -0.0004041009
## Prob        -3.6121278354
## Time         0.0109000093
```

and then with the λ by “1 standard error” rule(the most penalized model with a 1 std distance from the model with the least error), for which the coefficients of its model are:

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -3.851443e-01
## M           -8.322927e-04
## So          -1.805066e-03
## Ed           1.222019e-03
## Po1          1.031324e-03
## Po2          1.037582e-03
## LF           2.374465e-05
## M.F         -9.505650e-05
## Pop          3.826205e-04
## NW           1.999666e-05
## U1          -1.286980e-04
## U2           5.825889e-04
## GDP          2.318449e-04
## Ineq        -2.533914e-04
## Prob        -7.072446e-01
## Time         1.821150e-03
```

Using the testing data set to evaluate the two models with different value of λ:

```
##       esti_y
## real_y 0 1
##      0 5 0
##      1 3 4
```

```
## [1] "For mapt = 0.5, the global performance is 0.750000"
```

```
##       esti_y
## real_y 0 1
##      0 5 0
##      1 6 1
```

```
## [1] "For mapt = 0.5, the global performance is 0.500000"
```

Lasso

And then the same methods for $\updownarrow_\infty$ penalization procedure, the difference is that lasso can select the variables and eliminate those less "necessary" variables.

```
## Warning: from glmnet Fortran code (error code -90); Convergence for 90th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned
```

```
## Warning: from glmnet Fortran code (error code -86); Convergence for 86th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned

## Warning: from glmnet Fortran code (error code -86); Convergence for 86th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned
```

First we selected a λ through 10-fold cross-validation with the minimum rule, for which the coefficients of the model are:

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept) -1.87747704
## M            .
## So           .
## Ed           .
## Po1          0.02031023
## Po2          .
## LF           .
## M.F          .
## Pop          .
## NW           .
## U1           .
## U2           .
## GDP          .
## Ineq         .
## Prob         .
## Time         .
```

and then with the λ by "1 standard error" rule(the most penalized model with a 1 std distance from the model with the least error), for which the coefficients of its model are:

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -0.691407792
## M            .
## So           .
## Ed           .
## Po1          0.006175808
## Po2          .
## LF           .
## M.F          .
## Pop          .
## NW           .
## U1           .
## U2           .
## GDP          .
## Ineq         .
## Prob         .
## Time         .
```

Using the testing data set to evaluate the two models with different value of λ:

```
##       esti_y
## real_y 0 1
##      0 4 1
##      1 3 4
```

```
## [1] "For mapt = 0.5, the global performance is 0.666667"
```

```
##       esti_y
## real_y 0 1
##      0 5 0
##      1 6 1
```

```
## [1] "For mapt = 0.5, the global performance is 0.500000"
```

Since we have different partition of traing and testing data set each time, the values of confusion matrix are not stable. As far as we concerned, it would be more accurate if we had larger quantity of data set to make the evaluation.