Sectional Project 1

Group 3

2/11/2021

Introduction to the Boston Housing Dataset

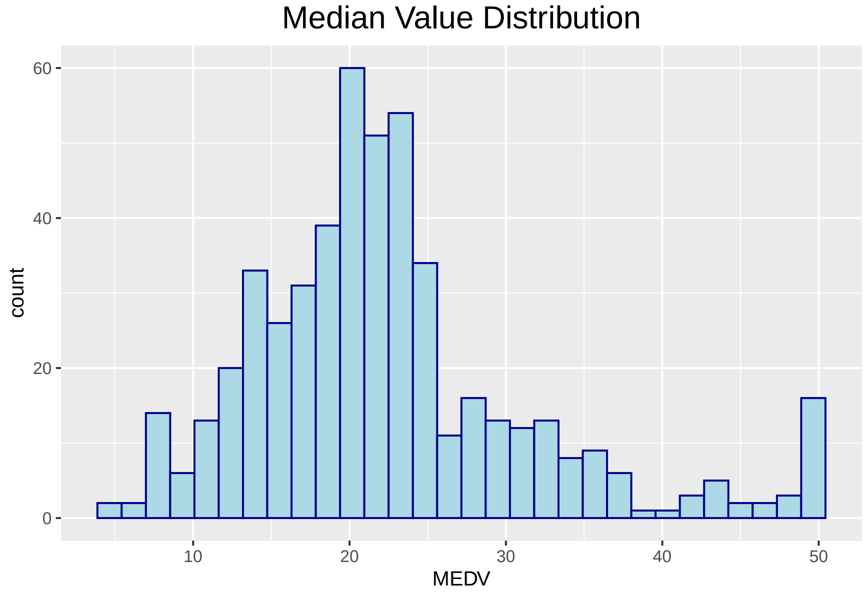
The Boston Housing dataset considers housing values and their associated properties in suburbs of Boston, Massachusetts. The dataset contains 506 observations and 14 attributes. We acquired the dataset from the Machine Learning Database (MLDB), found here. In particular, we are interested in constructing a model through regression techniques to gain insight on housing values. As such, we will use the 13 features to model 'MEDV', the median value of owner occupied homes (in $1,000s).

The data is displayed as follows.

```
##      CRIM ZN INDUS CHAS   NOX    RM  AGE    DIS RAD TAX PTRATIO      B LSTAT
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   MEDV
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

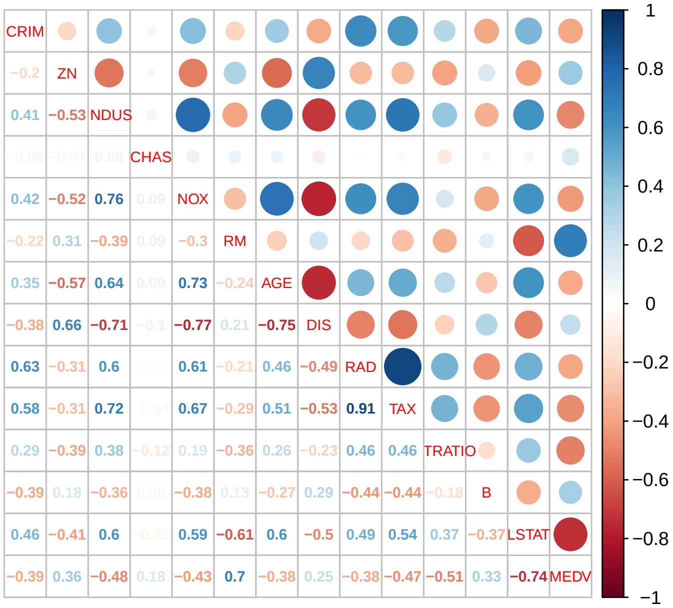
Exploratory Data Analysis

MEDV Distribution



The histogram demonstrates the values are not uniformly distributed. Rather, they follow a mostly normal distributions, with some outliers at the tail.

Correlation Matrix



While we produce many correlation values, we are firstly interested in how each attribute correlates to MEDV. This is represented by the bottom row or last column. We can immediately see the binary CHAS attribute does not correlate strongly with MEDV. However, it can be seen that RM (0.7) and LSTAT (-0.74) correlate with MEDV stronger than other attributes. Furthermore, the correlation between RM and LSTAT is -0.61. Since they do not correlate very strongly with one another, we can select both as predictor attributes without too much concern of collinearity for their case. The greatest correlation is between RAD and TAX of 0.91. Including both of these may raise some concerns regarding the minimal collinearity assumption of linear regression.

Example EDA subsection title 3

Modelling and Regression

MedV takes the value for Y, along 13 feature attributes of the dataset, in the form of $Y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$.

Multiple Linear Regression

This is very subject to change. I chose to simply split the data in a 80/20 split. If you think we should consider another split, or more, please do so. Furthermore, this is simply a start, and have yet to articulate everything. Please feel free to make changes.

Naively consider most attributes at onset.

```
##
## Call:
## lm(formula = MEDV ~ CRIM + ZN + INDUS + NOX + RM + AGE + DIS +
##     RAD + TAX + PTRATIO + B + LSTAT, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7529  -2.3580  -0.3601   1.4662  25.0880
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.607269   6.933545   5.857 1.28e-08 ***
## CRIM         -0.117088   0.036419  -3.215  0.00145 **
## ZN            0.046371   0.016641   2.787  0.00568 **
## INDUS         0.098641   0.071828   1.373  0.17072
## NOX         -16.550316   5.088089  -3.253  0.00128 **
## RM            3.280962   0.577424   5.682 3.24e-08 ***
## AGE           0.013777   0.016774   0.821  0.41214
## DIS          -1.216793   0.250349  -4.860 1.92e-06 ***
## RAD           0.494834   0.078440   6.308 1.05e-09 ***
## TAX          -0.020556   0.004329  -4.749 3.22e-06 ***
## PTRATIO      -0.948615   0.160907  -5.895 1.04e-08 ***
## B             0.005826   0.003561   1.636  0.10289
## LSTAT        -0.635492   0.068081  -9.334  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.596 on 290 degrees of freedom
## Multiple R-squared:  0.7209, Adjusted R-squared:  0.7094
## F-statistic: 62.42 on 12 and 290 DF,  p-value: < 2.2e-16
```

Some initial insight is that LSTAT and RM indeed were strong predictors. Removing INDUS and AGE, every attribute becomes a significant predictor with the possible exception of B, depending on alpha. Let’s consider what happens if we remove RAD, which varies strongly with TAX.

```
##
## Call:
## lm(formula = MEDV ~ CRIM + ZN + NOX + RM + DIS + TAX + PTRATIO +
##     B + LSTAT, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5122  -2.7493  -0.4671   1.5476  26.8383
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.081015   6.965984   4.031 7.08e-05 ***
## CRIM         -0.059411   0.037077  -1.602   0.1102
## ZN            0.043133   0.017523   2.462   0.0144 *
## NOX         -10.990050   4.943774  -2.223   0.0270 *
## RM            3.931161   0.584748   6.723 9.31e-11 ***
## DIS          -1.234525   0.243923  -5.061 7.37e-07 ***
## TAX           0.001092   0.002723   0.401   0.6887
## PTRATIO      -0.794315   0.166071  -4.783 2.74e-06 ***
## B             0.005448   0.003775   1.443   0.1500
## LSTAT        -0.604175   0.065251  -9.259  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.877 on 293 degrees of freedom
## Multiple R-squared:  0.6824, Adjusted R-squared:  0.6727
## F-statistic: 69.96 on 9 and 293 DF,  p-value: < 2.2e-16
```

We can see that without RAD, TAX is no longer a strong predictor. As such, TAX adds predictive value in relation to RAD. The next model removes TAX and adds RAD back in.

```
##
## Call:
## lm(formula = MEDV ~ CRIM + ZN + NOX + RM + DIS + RAD + PTRATIO +
##     B + LSTAT, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9377  -2.7857  -0.4053   1.3582  25.2534
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.513394   6.993327   5.078 6.79e-07 ***
## CRIM         -0.112026   0.037494  -2.988 0.003047 **
## ZN            0.039376   0.017040   2.311 0.021541 *
## NOX         -18.251075   4.714168  -3.872 0.000133 ***
## RM            3.674324   0.572928   6.413 5.70e-10 ***
## DIS          -1.303422   0.238088  -5.475 9.43e-08 ***
## RAD           0.198842   0.049375   4.027 7.20e-05 ***
## PTRATIO      -0.969830   0.162569  -5.966 7.01e-09 ***
## B             0.007091   0.003669   1.933 0.054248 .
## LSTAT        -0.609417   0.063499  -9.597  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 4.749 on 293 degrees of freedom
## Multiple R-squared:  0.6989, Adjusted R-squared:  0.6897
## F-statistic: 75.57 on 9 and 293 DF,  p-value: < 2.2e-16
```

```
#This model is using any value that is abs(Intercept) > 1 for the ridge
multiModel4 <-lm(MEDV ~NOX+RM+DIS+PTRATIO+LSTAT, data=training)
summary(multiModel4)
```

```
##
## Call:
## lm(formula = MEDV ~ NOX + RM + DIS + PTRATIO + LSTAT, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.324  -2.610  -0.629   1.810  27.131
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.34097    6.37104   5.076 6.80e-07 ***
## NOX         -12.00327    4.32985  -2.772  0.00592 **
## RM            3.95741    0.57522   6.880 3.55e-11 ***
## DIS          -0.97773    0.21913  -4.462 1.15e-05 ***
## PTRATIO      -0.88185    0.15806  -5.579 5.44e-08 ***
## LSTAT        -0.65159    0.06227 -10.463  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.938 on 297 degrees of freedom
## Multiple R-squared:   0.67,  Adjusted R-squared:  0.6645
## F-statistic: 120.6 on 5 and 297 DF,  p-value: < 2.2e-16
```

MSE for model 1, 2, and 3.

```
## [1] 22.55438
```

```
## [1] 21.78997
```

```
## [1] 22.13214
```

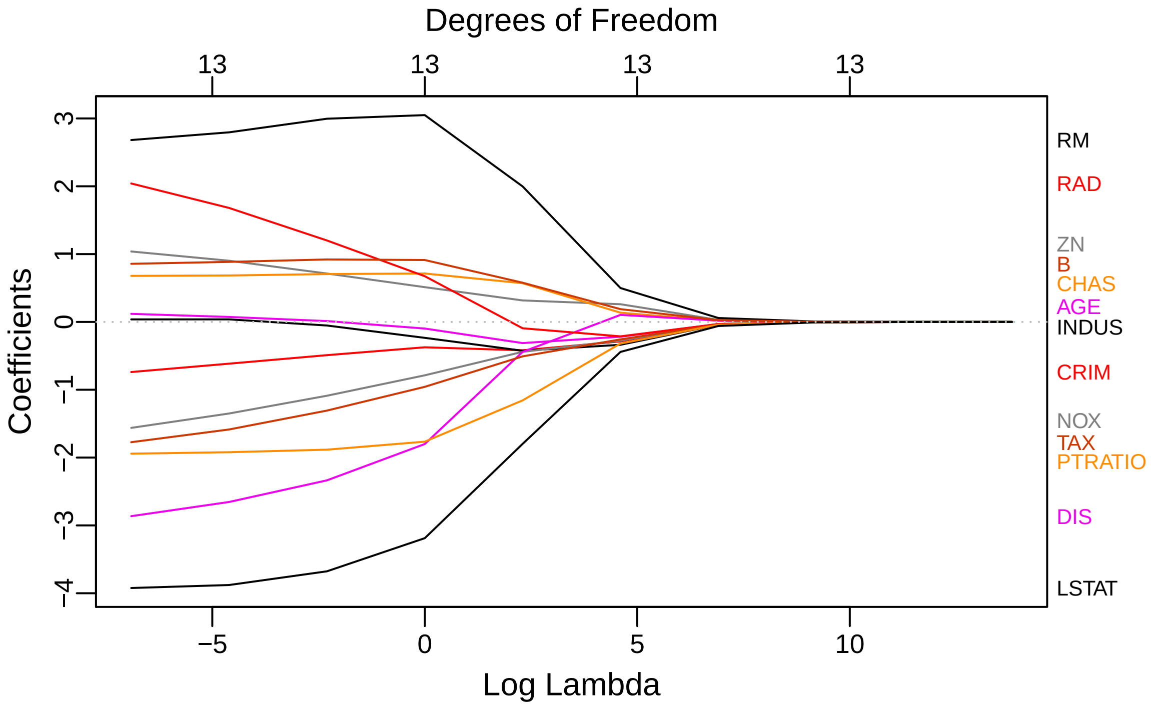
```
## [1] 23.23179
```

Lasso with parameter tuning will give us further insight into paramater selection. This section can/should be expanded/refined.

Ridge Regression

Constructing a ridge model.

```r
#lambda grid
grid <- 10^seq(6, -3, length=10)
#ridgeModel
ridge.mod <- glmnet(scale(x), y, alpha = 0, lambda = grid, thresh = 1e-2, standardize = TRUE)
plot_glmnet(ridge.mod, xvar = "lambda", label = 13)
```

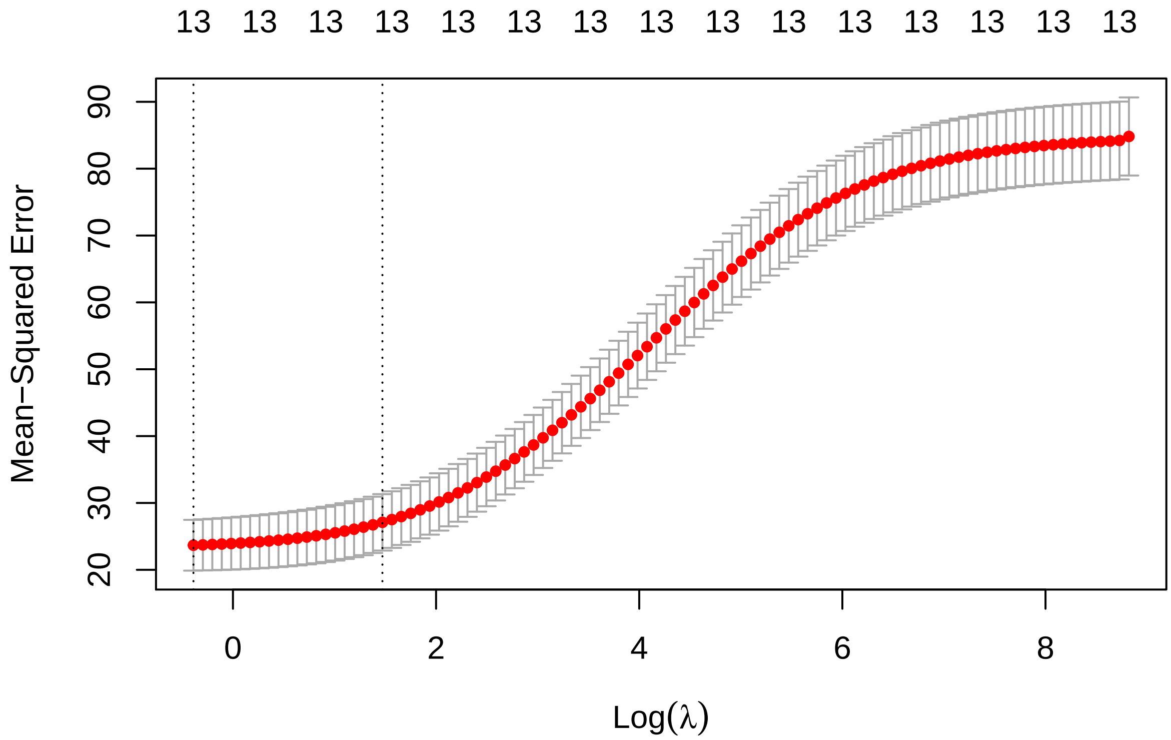


Considering a best lambda for the model and hyperparameter tuning.

```r
cv.out <- cv.glmnet(x, y, alpha=0, nfolds = 10)
cv.out
```

```
##
## Call:  cv.glmnet(x = x, y = y, nfolds = 10, alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min  0.678   100   23.68 3.797      13
## 1se  4.357    80   27.10 4.220      13
```

```r
plot(cv.out)
```



```
best.lambda <- cv.out$lambda.min
best.lambda
```

```
## [1] 0.6777654
```

Considering coefficients for ridge.

```
ridge.final <- glmnet(scale(x), y, alpha = 0, lambda = best.lambda, thresh=1e-2, standardsize = TRUE)
predict(ridge.final, type="coefficients", s=best.lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 22.5328063
## CRIM        -0.4433313
## ZN           0.5934338
## INDUS       -0.7647167
## CHAS         0.7494329
## NOX         -1.4139968
## RM           3.2123657
## AGE         -0.3130941
## DIS         -2.5539636
## RAD          0.8424452
## TAX         -0.7510226
## PTRATIO     -1.8099040
## B            0.8974522
## LSTAT       -2.8927874
```

Considering MSE and RMSE for ridge using all coefficients.

```
## [1] 42046.5
```

```
## [1] 205.0524
```

Lasso Regression

K-Fold Cross Validation

A simple quick table for us to store into for now:

	MSE	RMSE
Multiple	0	0
Multiple 2	0	0
Ridge	0	0
Lasso	0	0

Citations