MSDS 7333: Quantifying the World

Week 3: Linear Regression Review with Regularization

Why Linear Regression

- Fundamental building block of advanced algorithms
- Introduced regularization
- Introduces loss functions
- Introduce variable transforms

Review of Linear Regression Basics

$$y = mx + b$$

$$y = m_i x_i + b$$

$$y = \sum_{i=1}^{n} m_i x_i + b$$

$$y = \sum_{i=0}^{n} m_i x_i \ , x_0 = 1$$

$$y_j = x_{ji} m_i$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_{1\alpha} x_{1\beta} x_{1\gamma} \\ x_{2\alpha} x_{1\beta} x_{1\gamma} \end{bmatrix} \begin{bmatrix} m_\alpha \\ m_\beta \\ m_\gamma \end{bmatrix}$$

So how do I solve this?

- YOU DON’T
- Seek a numerical solution
- Gradient Descent
  - “I was told there was to be no math”
  - You were lied to

Gradient Descent

- Instead of: $y_j = x_{ji} m_i$
- Make a guess: $g_j = x_{ji} m_i$ Don't forget—these are matrices
- How good is my guess? $\frac{1}{2n}\sum_{j=1}^{n}\left(g_j - y_j\right)^2$
- Let's minimize my guess! $\frac{1}{2n}\sum_{j=1}^{n}\left(x_{ji} m_i - y_j\right)^2$

Gradient Descent pt 2

I only have 1 thing to change to minimize this function—the coefficients m!

$$\frac{1}{2n}\sum_{j=1}^{n}\left(x_{ji}m_i - y_j\right)^2$$

$$\frac{\partial}{\partial m_i}\frac{1}{2n}\sum_{j=1}^{n}\left(x_{ji}m_i - y_j\right)^2$$

$$\frac{1}{n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)\,x_{ji}$$

Update rule:

$$m_i = m_i - \frac{\propto}{n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)x_{ji}$$

If I want to minimize, measure where I'm at and go "down" (the minus sign)

Wall of math crits you for 2000. You die

Lets fit a simple example: x=0, y=1 and x = 5, y=4

Easy to solve via algebra, but use our Gradient Descent to understand what is happening.
Make a dumb guess: m=0, I choose my learning rate of 0.1 (alpha)

$$m_i' = m_i - \frac{\alpha}{n}\sum_{j=1}^{n}\left(x_{ji}m_i - y_j\right)x_{ji}$$

$$m_i' = 0 - \frac{0.1}{2}\{[(0 * 0 - 1)0] + [(5 * 0) - 4]5\}$$

$$m_i' = 0 - \frac{0.1}{2}\{-[4]5\}$$

$$m_i' = 0 + \frac{0.1}{2}\{[4]5\}$$

$$m_i' = 1$$

Hey…Wait a sec

- Where did the intercept go…?

Make a guess: $m = 0$, $b = -1$ (aka m_0=b, m_1=m)

ReDo: Lets fit a simple example: x=0, y=1 and x = 5, y=4

Make a guess: m = 0, b = -1 (aka m_0=b, m_1=m)

$$m_i' = m_i - \frac{\alpha}{n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)x_{ji}$$

$$m_0' = m_0 - \frac{\alpha}{n}\sum_{j=1}^{n}(x_{j0}m_0 - y_j)x_{j0}$$

$$m_1' = m_1 - \frac{\alpha}{n}\sum_{j=1}^{n}(x_{j1}m_1 - y_j)x_{j1}$$

$$m_0' = -1 - \frac{0.1}{2}\{[(1 * -1 - 1)1] + [(1 * -1) - 4]1\}$$

Same as previous slide

$$m_0' = -1 - \frac{0.1}{2}\{[(-2)1] + [-5]1\}$$

$$m_0' = -1 + \frac{7}{20}$$

Sometimes toy examples help!!

$$\frac{1}{2n}\sum_{j=1}^{n}\left(x_{ji}m_i - y_j\right)^2$$

Take 3 points – (1,2), (2,5), (3,8)
Write out the loss and look at the equation:

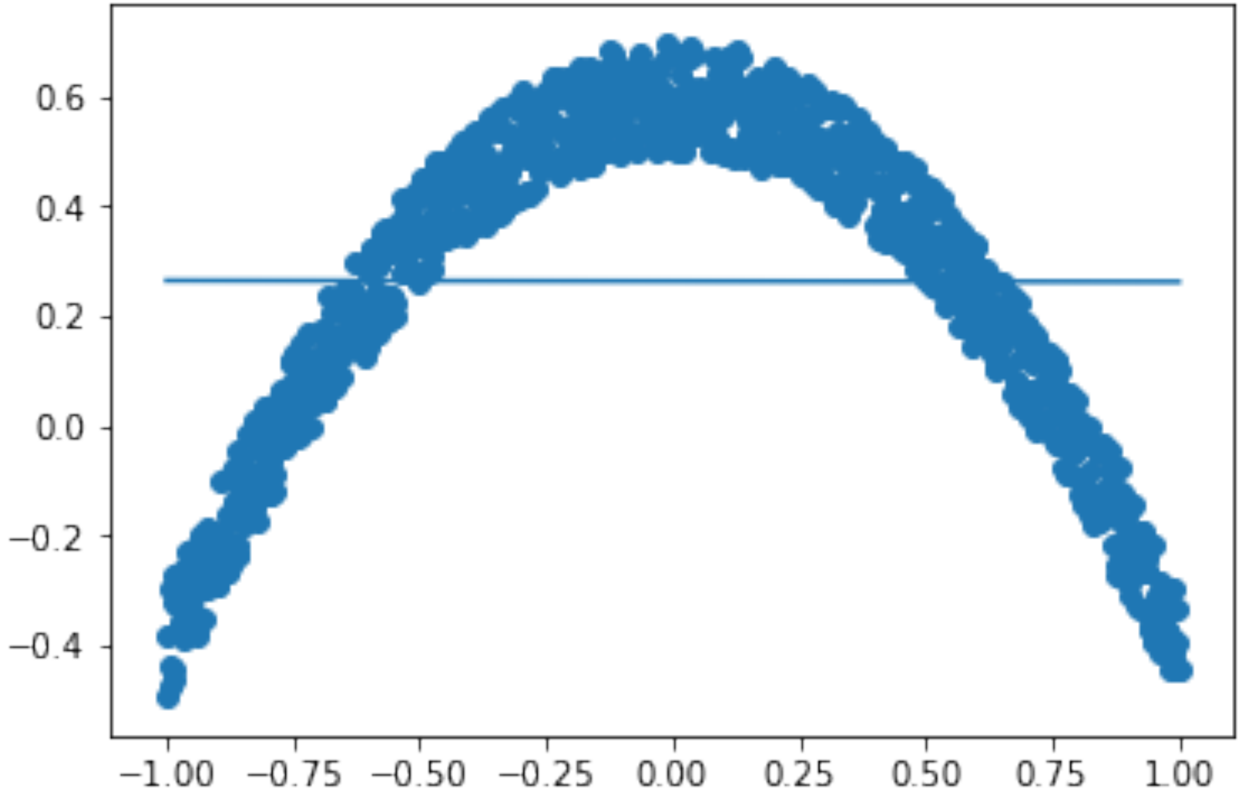
$$\frac{1}{6}\{(1m-2)^2+(2m-5)^2+(3m-8)^2\}$$

$$\frac{1}{6}\{m^2-2m+4+4m^2-20m+25+9m^2-48m+64\}$$

$$\frac{1}{6}\{14m^2-70m+93\}$$

Notice: even if I made one of the points an ‘error’ (slope is 3, intercept is -1 for this example), only the coefficients change. Error is still quadratic.

Minimized error ! = good model



What about categorical targets?

- Variable transform:

$$r(z) = \frac{1}{1 - e^{-z}}$$

z=mx

- Loss becomes:

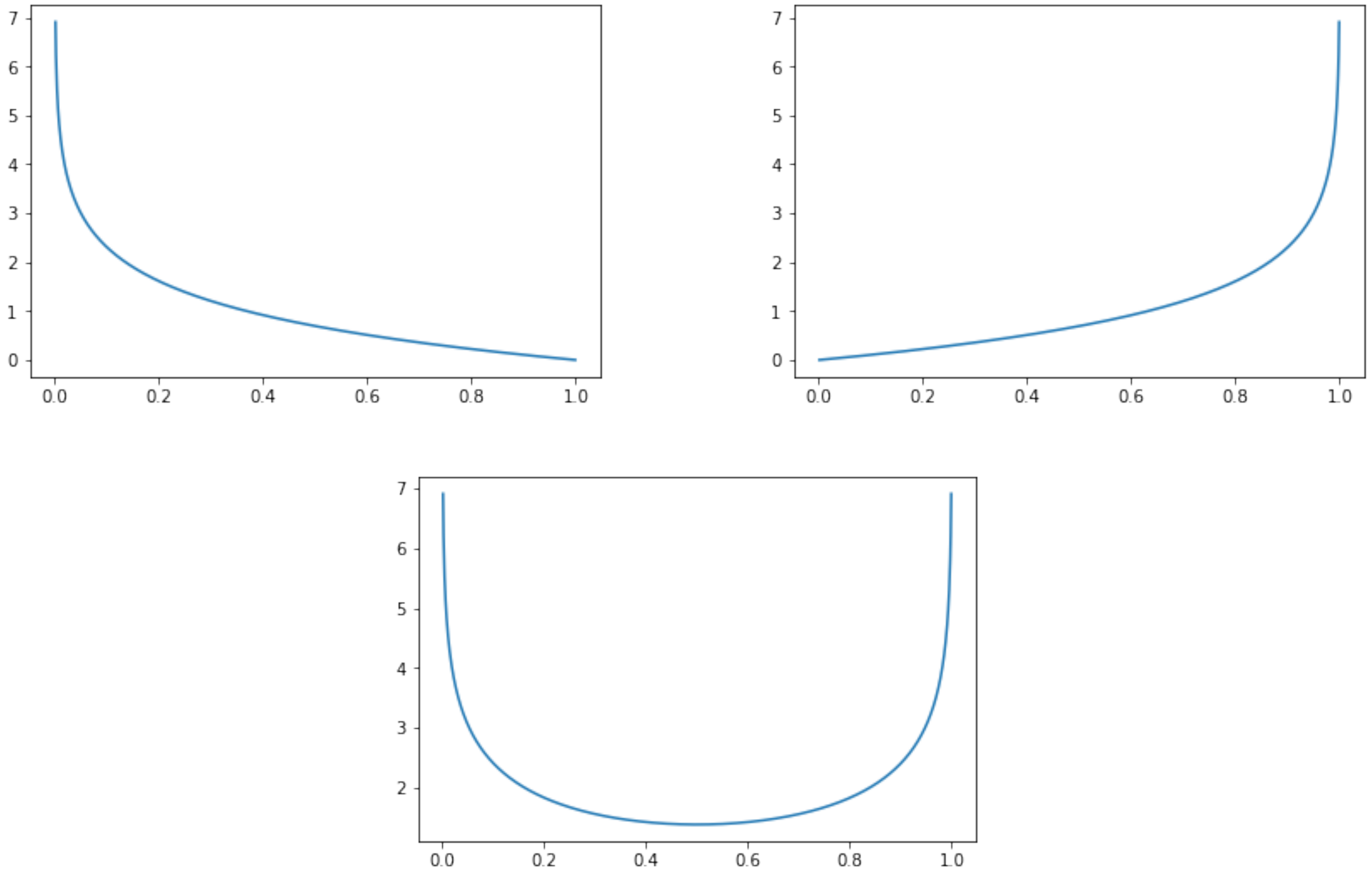
$$-y \ln(r) - (1 - y)\ln(1 - r)$$

- Update rule: (it’s the same!!)

$$m_i = m_i - \frac{\propto}{n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)x_{ji}$$

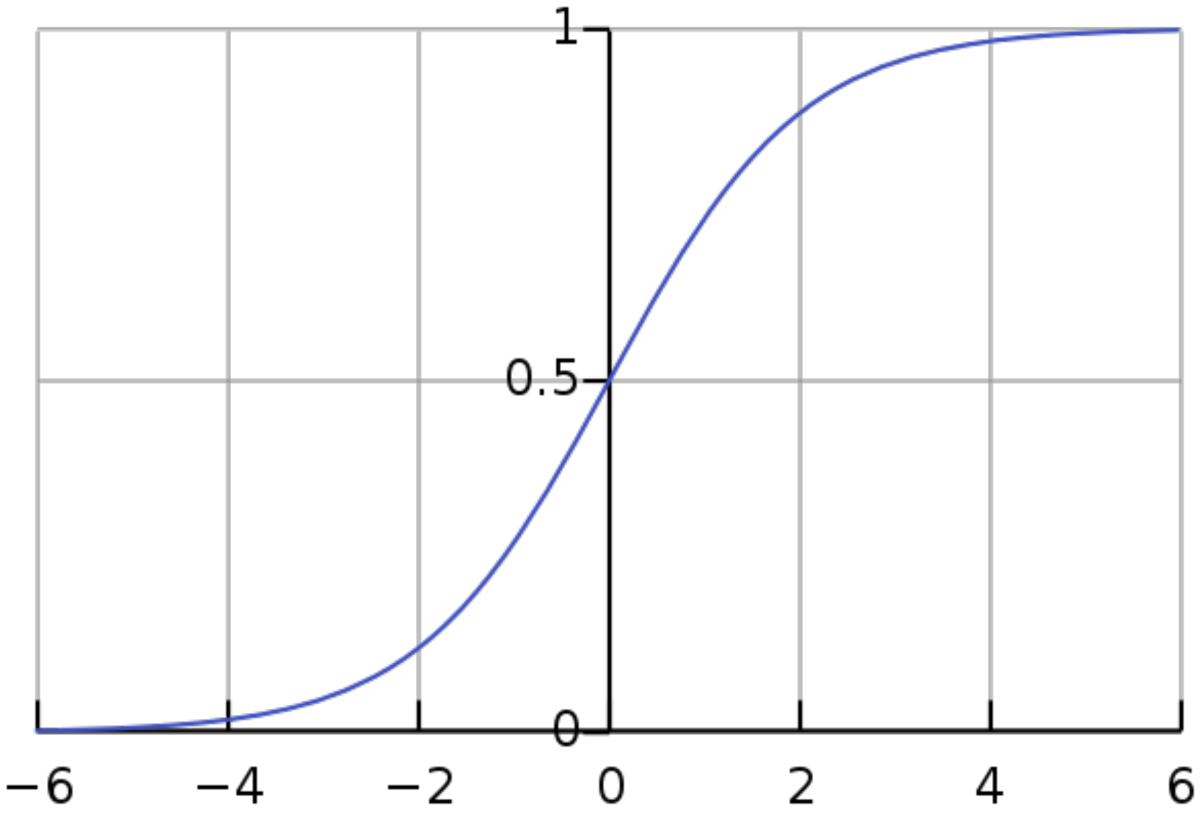
https://math.stackexchange.com/questions/78575/derivative-of-sigmoid-function-sigma-x-frac11e-x

Wow...hold up (loss)



Sigmoid

$$r(z) = \frac{1}{1 - e^{-z}}$$



Regularization

- Start with a loss function

$$m_i = m_i - \frac{\propto}{n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)x_{ji}$$

- Add a penalty

$$\frac{1}{2n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)^2 + \lambda\sum_{i=1}^{p}|m_i|$$

L1 Regularization

$$\frac{1}{2n}\sum_{j=1}^{n}(x_{ji}m_i - y_j)^2 + \lambda\sum_{i=1}^{p}m_i^2$$

L2 Regularization

Why do this?

- Consider last week—12,000 ‘features’
- What if you try feature creation?
  - 10 features
  - Create cross products
  - 100 new features
    - Some good, some bad—which ones to keep
    - Keep from overfitting
      - Helps generalize

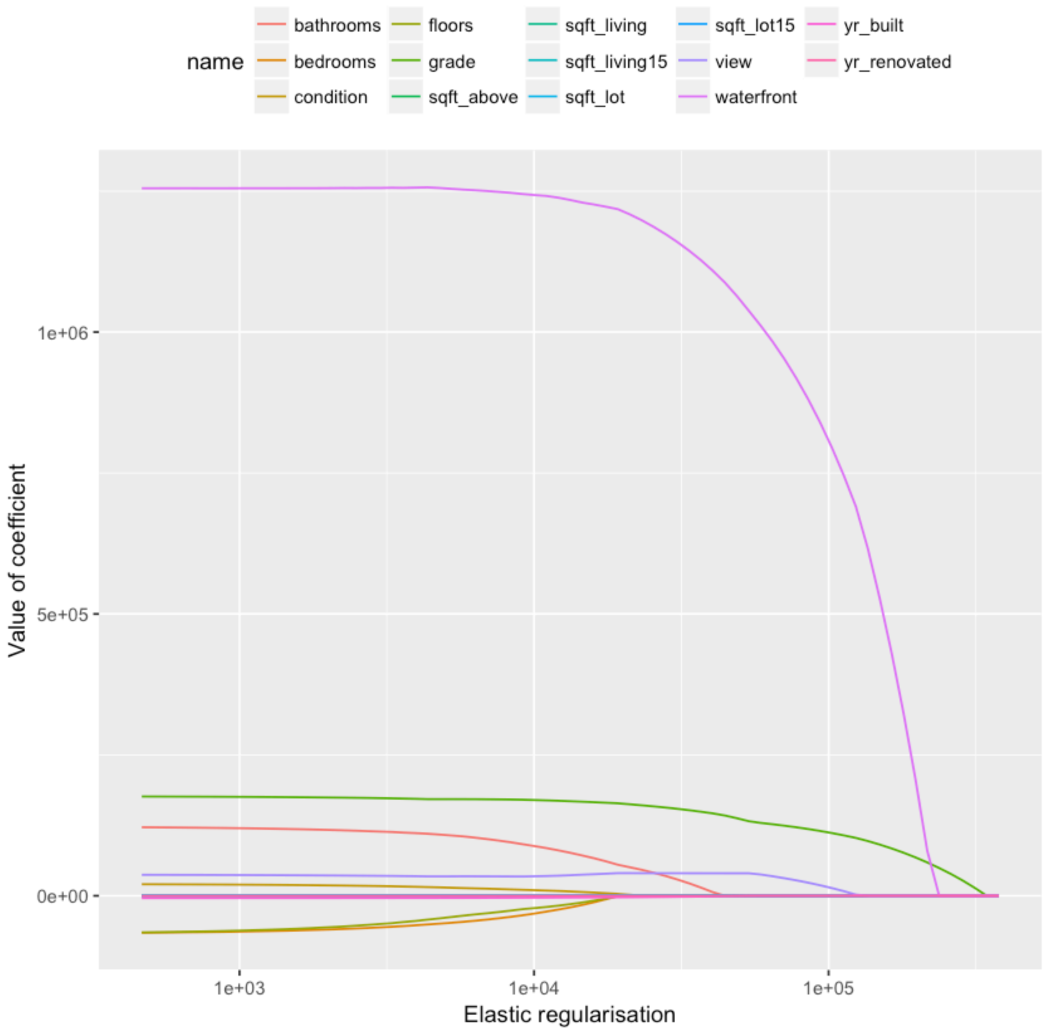
Which to use

- L1 works great for feature selection
  - Induces ‘Sparsity’ or A lot of zero coefficients
- L2 prevents overfitting
- Use both: Elastic Net

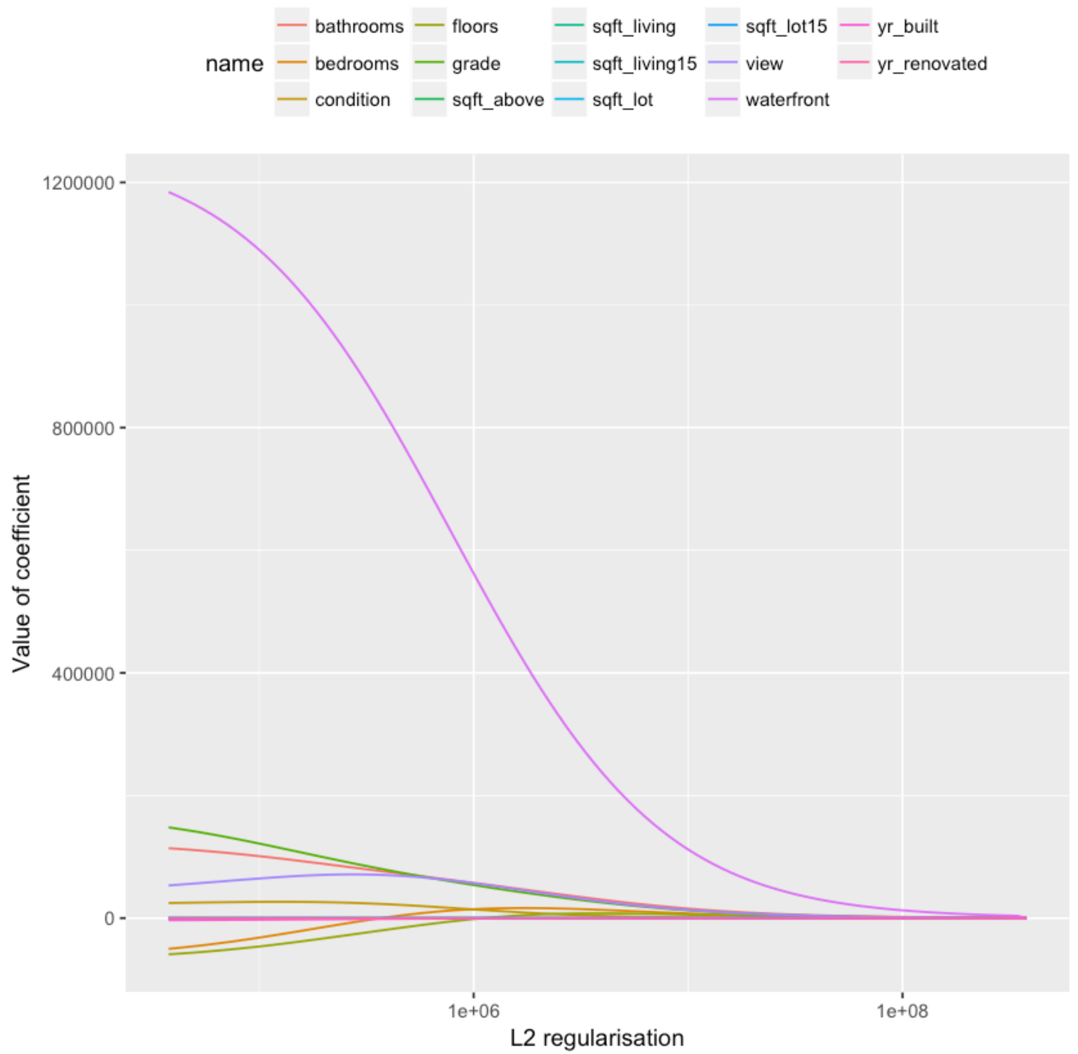
$$\alpha|\lambda_1| + \frac{1-\alpha}{2}(\lambda_2)^2$$

$$\lambda_1 = \lambda_2$$

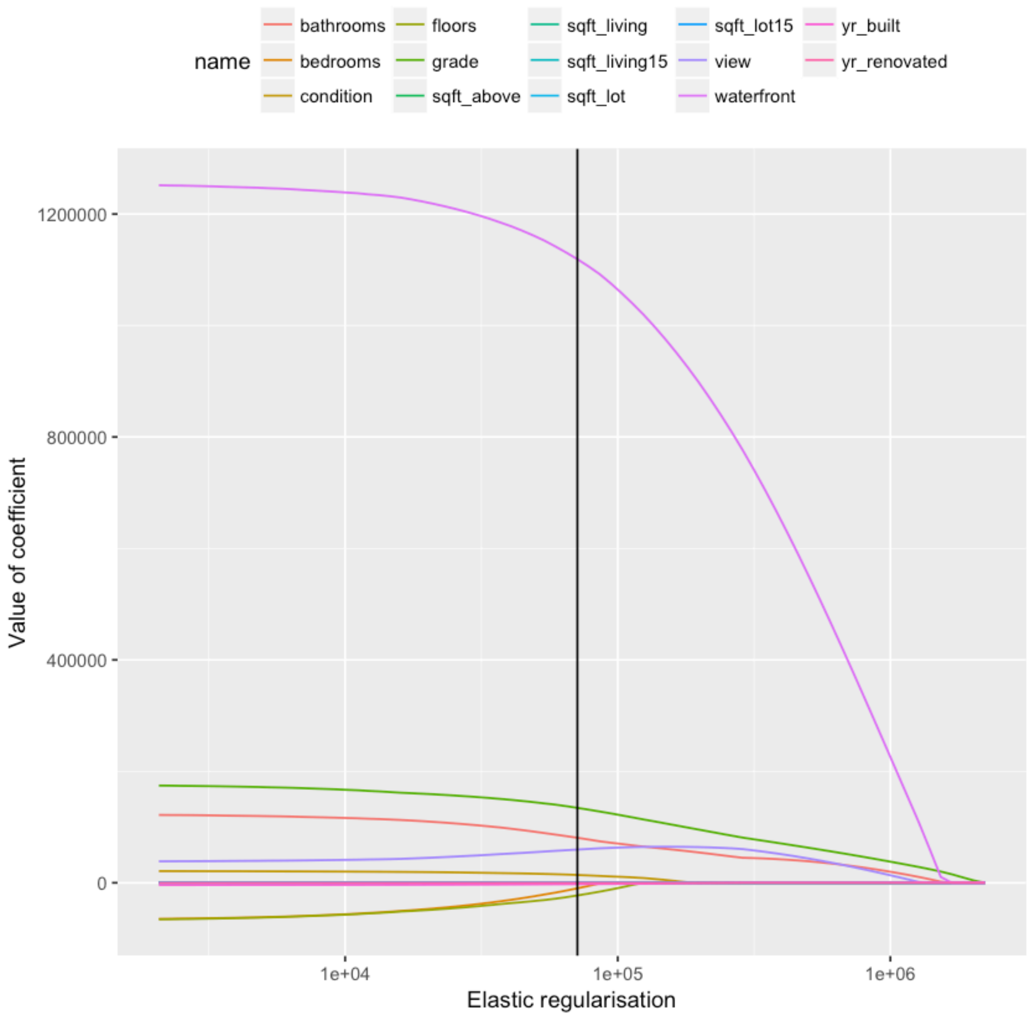
L1 Regularization (alpha = 1)

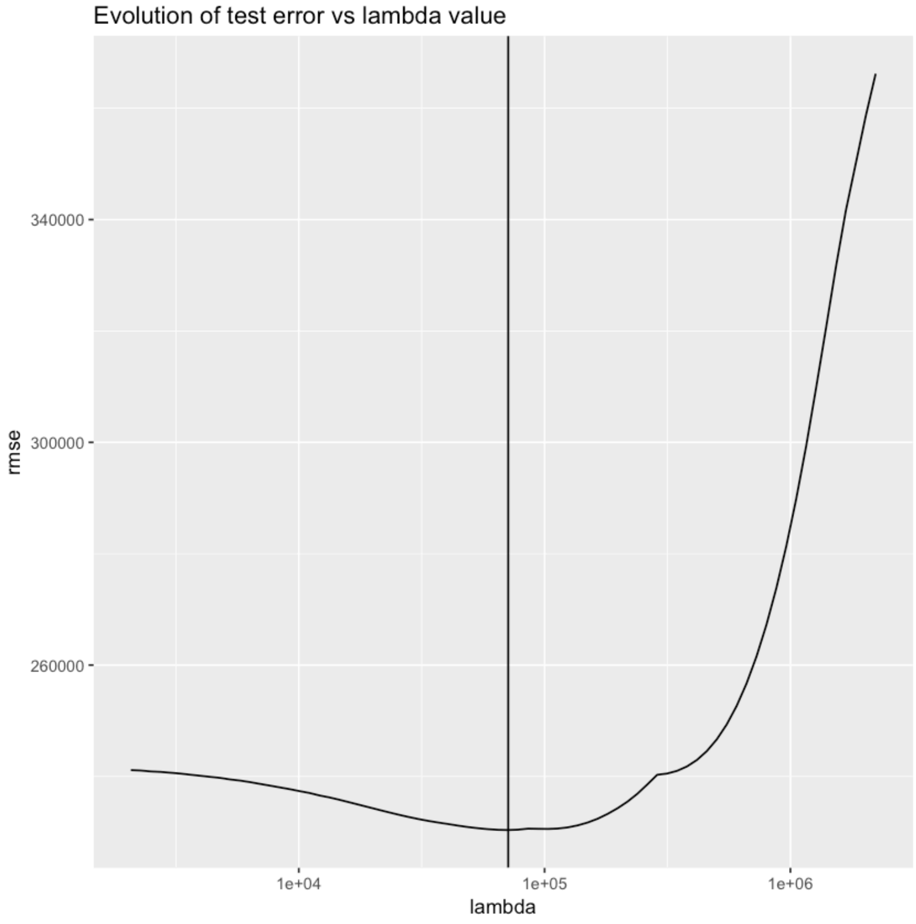


L2 Regularization (alpha = 0)



Elastic Net , alpha = 0.17





Examples

- Use R Package
- Add features and use regularization
- Get toy datasets

DataScience@SMU