Ridge Regression

Ridge Regression

- Ridge Regression is a regularization technique that works by helping reduce the potential for overfitting to the training data.
- It does this by adding in a penalty term to the error that is based on the squared value of the coefficients.

Ridge Regression

- Ridge Regression is a regularization method for Linear Regression.
- Relevant Reading in ISLP:
  - Section 6.2.1
- Let's explore the main concepts behind how Ridge Regression works...

Ridge Regression

- Recall the general formula for the regression line:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

Ridge Regression

- These Beta coefficients were solved by minimizing the residual sum of squares (RSS).

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

Ridge Regression

- These Beta coefficients were solved by minimizing the residual sum of squares (RSS).

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Ridge Regression

- We could substitute our regression equation for ŷ:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Ridge Regression

- We could substitute our regression equation for **ŷ**:

$$
\begin{aligned}
\mathrm{RSS} &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\
&= \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2
\end{aligned}
$$

Ridge Regression

- We can then summarize RSS as:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Ridge Regression

- The goal of Ridge Regression is to help prevent overfitting by adding an additional penalty term.

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Ridge Regression

- Ridge Regression adds a **shrinkage penalty:**

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \boxed{+\lambda\sum_{j=1}^{p}\beta_j^2}$$

Ridge Regression

- **Shrinkage penalty** based off the squared coefficient:

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\boxed{\beta_j^2}$$

Ridge Regression

- **Shrinkage penalty** has a **tunable lambda parameter!**

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda}\sum_{j=1}^{p}\beta_j^2$$

$\boxed{\lambda}$ We shall talk about Lambda parameter sen

Ridge Regression

- Lambda determines how severe the penalty is.

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda}\sum_{j=1}^{p}\beta_j^2$$

Ridge Regression

- In theory it can be any value from 0 to positive infinity.

$$\text{Error} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \boxed{\lambda} \sum_{j=1}^{p} \beta_j^2$$

Ridge Regression

- If it is zero, then it is simply back to RSS.

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda}\sum_{j=1}^{p}\beta_j^2$$

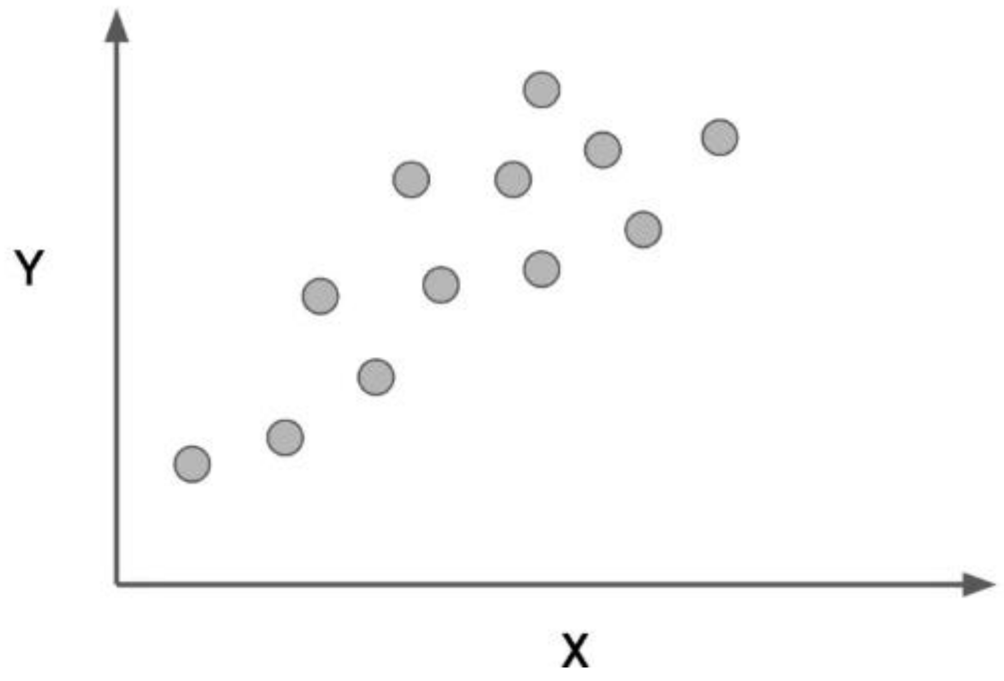
Ridge Regression

- Let's explore a simple thought experiment to get an intuition behind Ridge Regression...

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

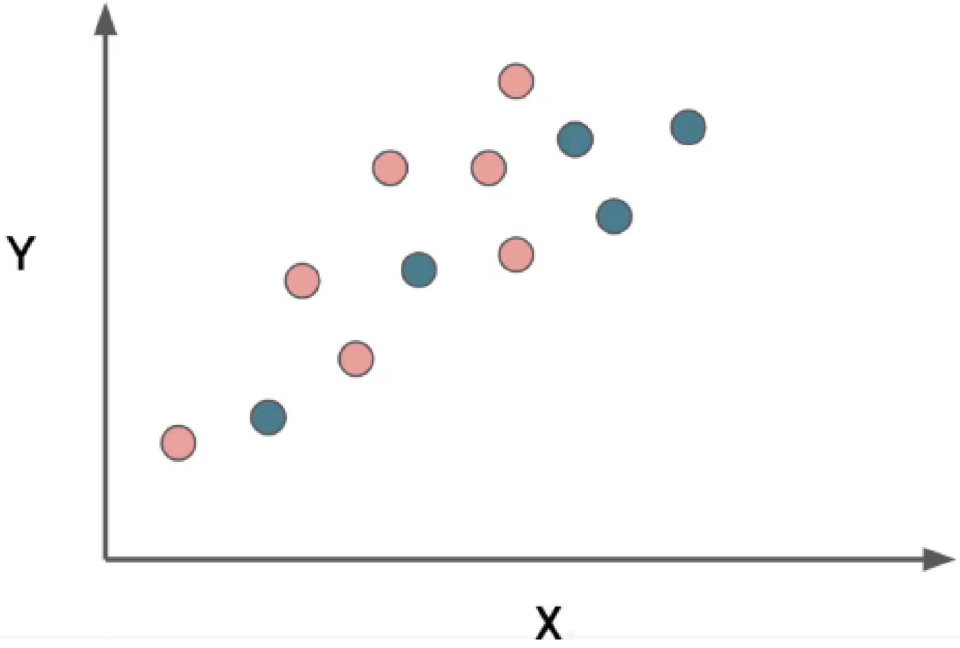
Ridge Regression – Simple one Feature

- Imagine the following data set.



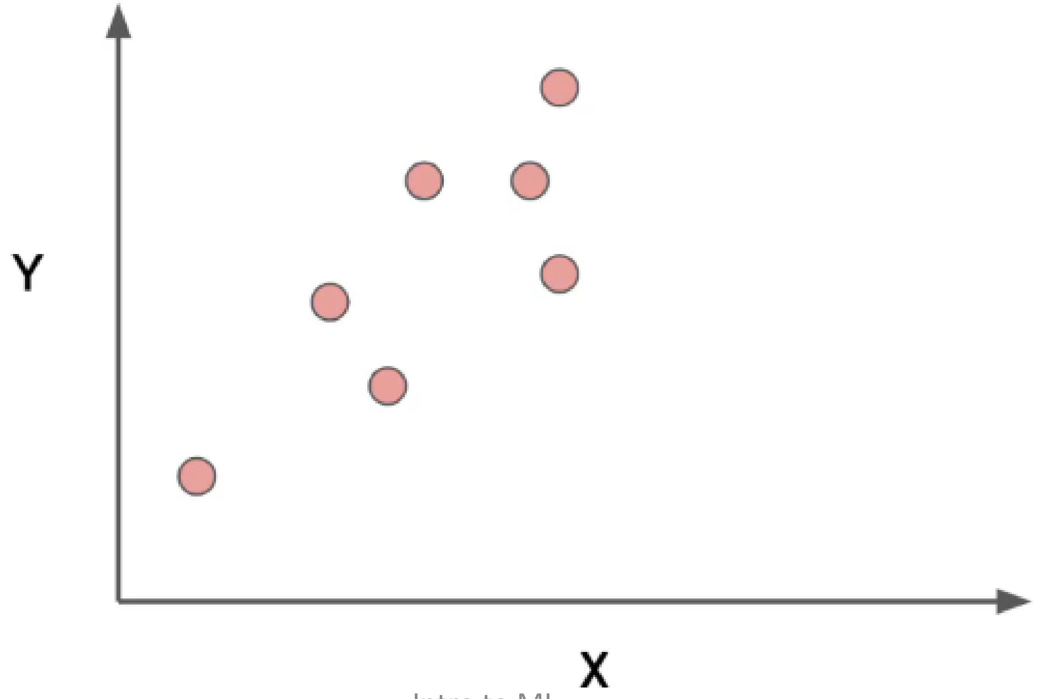
Ridge Regression – Simple one Feature

- We can split it into a training set and test set:



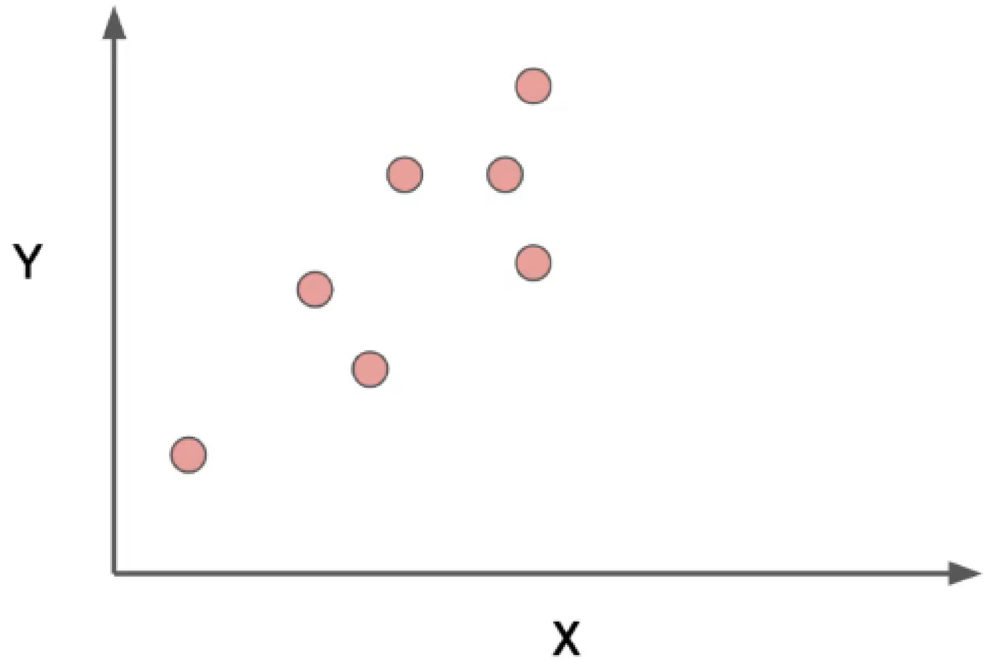
Ridge Regression – Simple one Feature

- Now we can fit on the training data to produce the line: $\hat{y} = \beta_1 x + \beta_0$



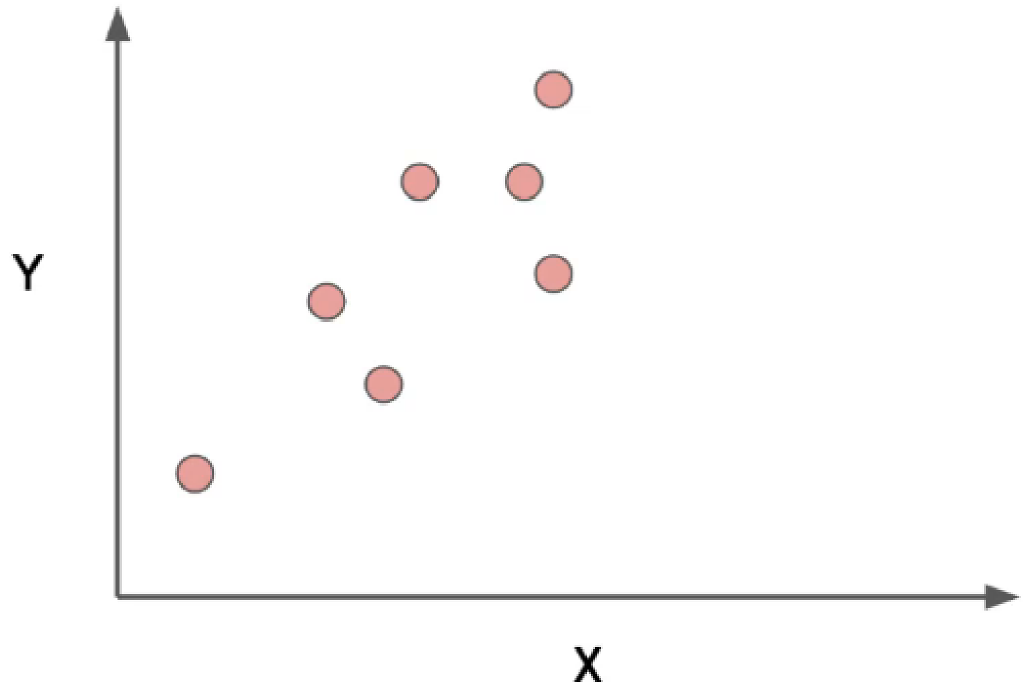
Ridge Regression – Simple one Feature

- Regardless of RSS or Ridge error, we're still trying to create a line: $\hat{y} = \beta_1 x + \beta_0$



Ridge Regression – Simple one Feature

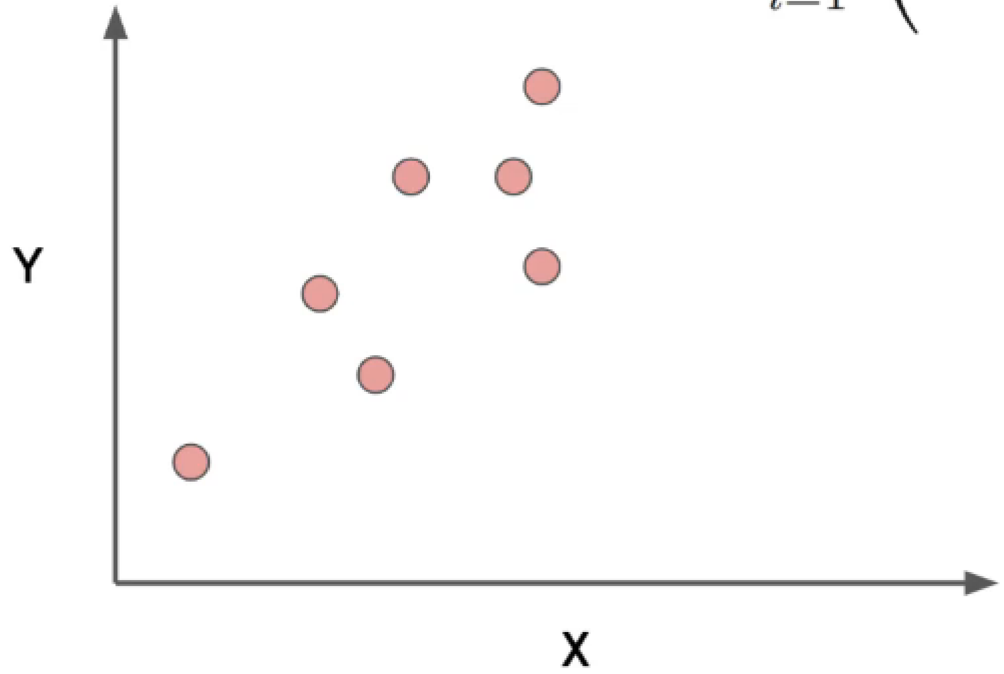
- The only difference would be the coefficients found.



Ridge Regression – Simple one Feature

- First let's fit using only RSS...

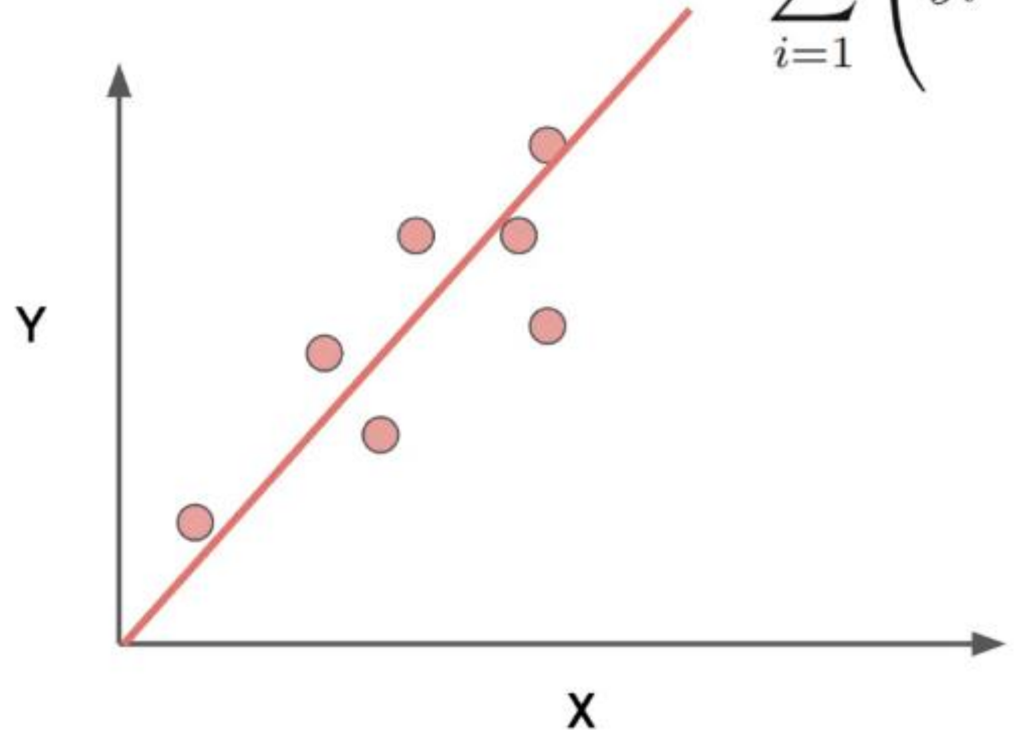
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$



Ridge Regression – Simple one Feature

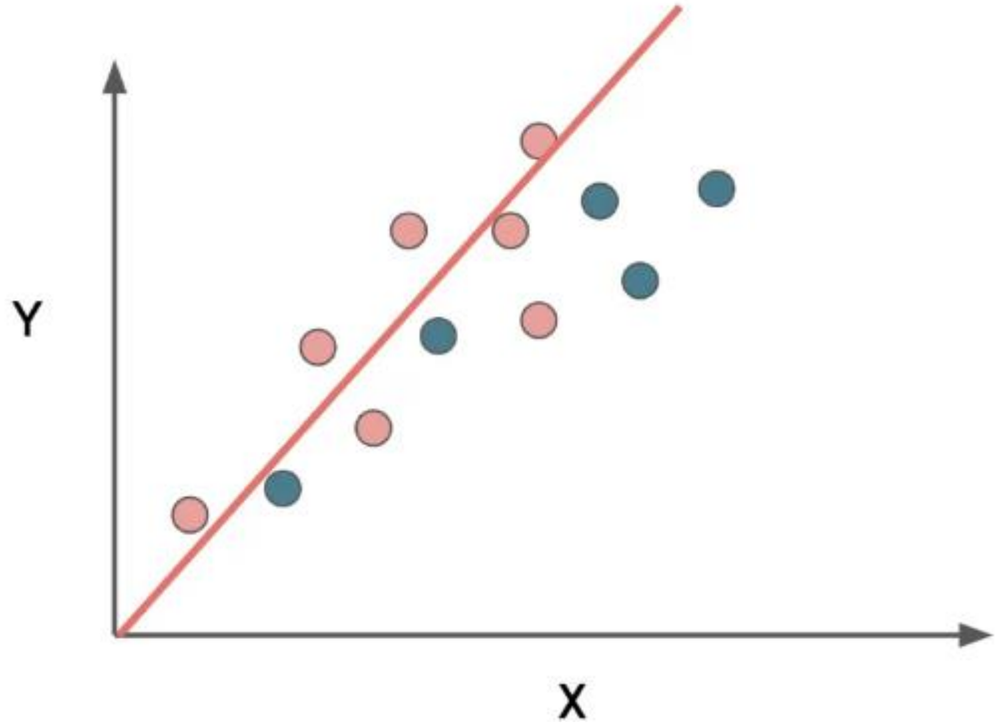
- Our fitted $\hat{y} = \beta_1 x + \beta_0$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$



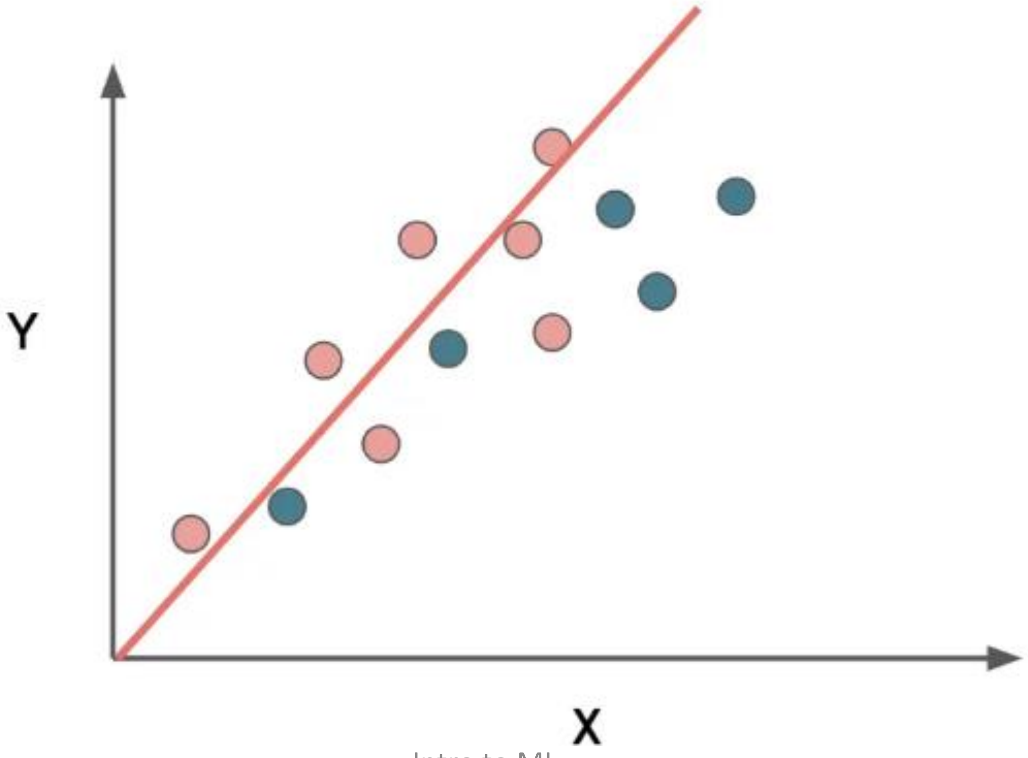
Ridge Regression – Simple one Feature

- Appears to have over fit to training data.



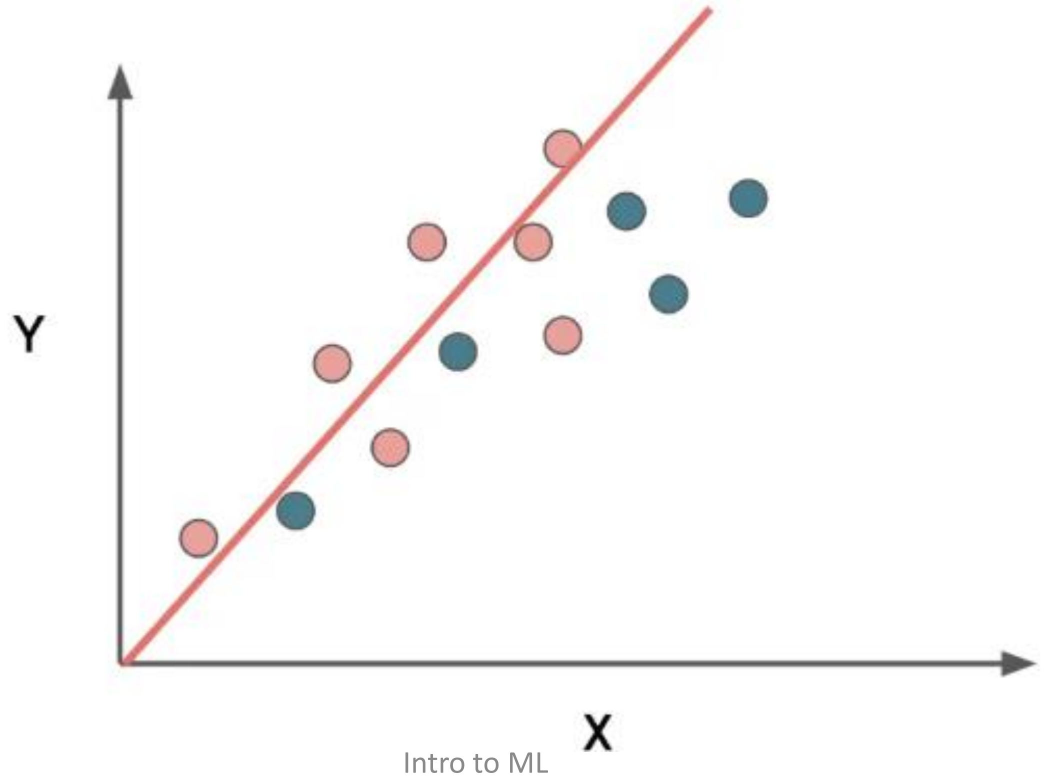
Ridge Regression – Simple one Feature

- This means we have high **variance.**



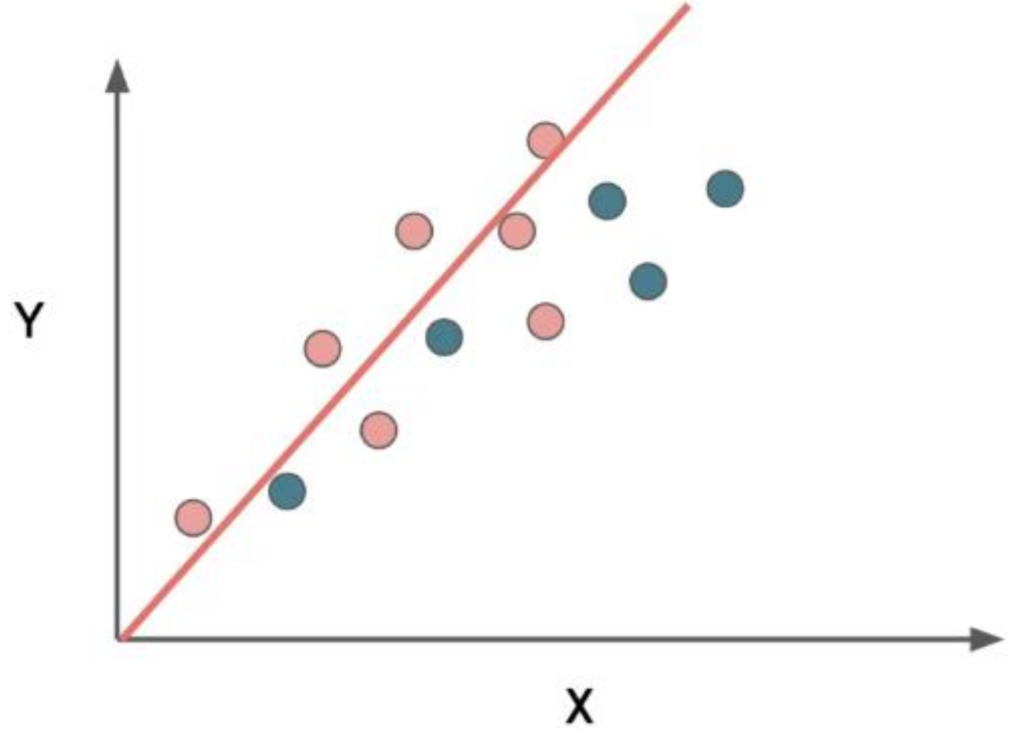
Ridge Regression – Simple one Feature

- We know there is a **bias-variance** trade-off.



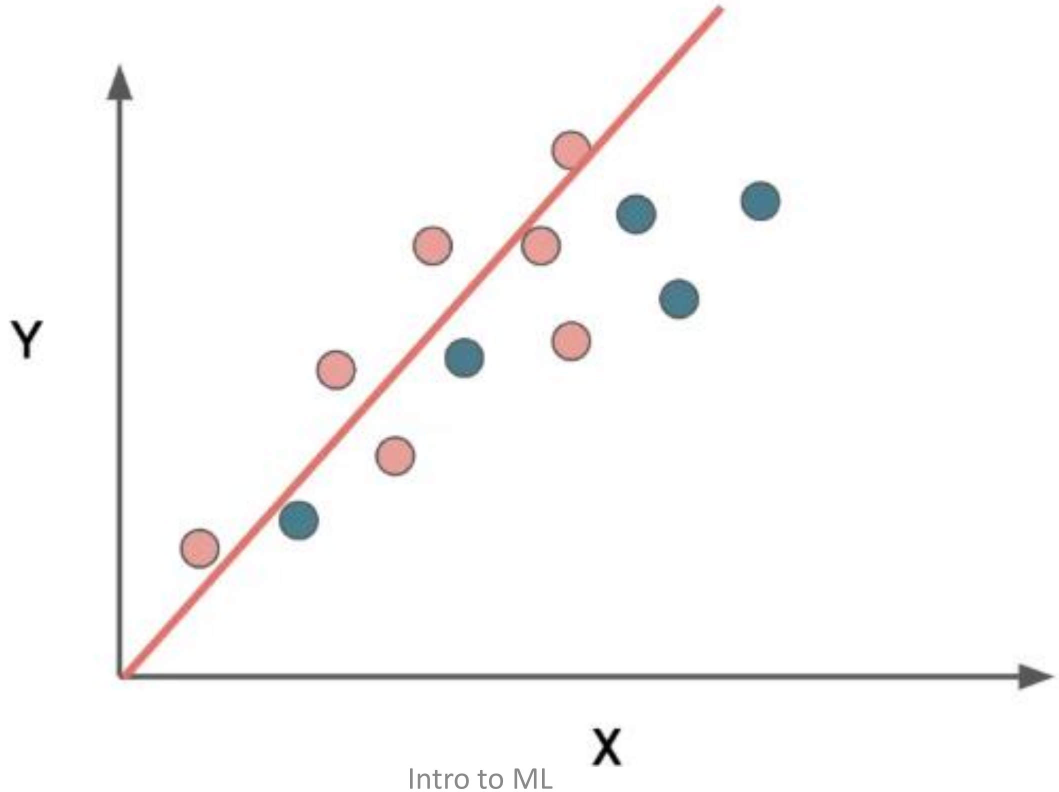
Ridge Regression – Simple one Feature

- But could we introduce a little more **bias** to significantly **reduce** variance?



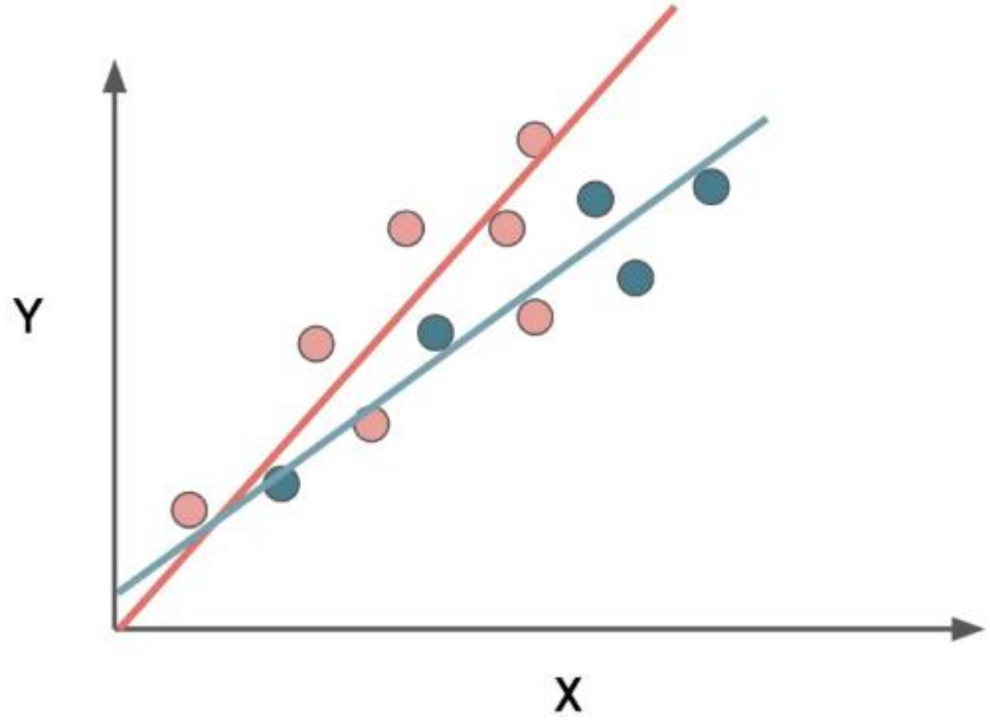
Ridge Regression – Simple one Feature

- Would adding the penalty term help generalize with more **bias?**



Ridge Regression – Simple one Feature

- Adding bias can help generalize $\hat{y} = \beta_1 x + \beta_0$

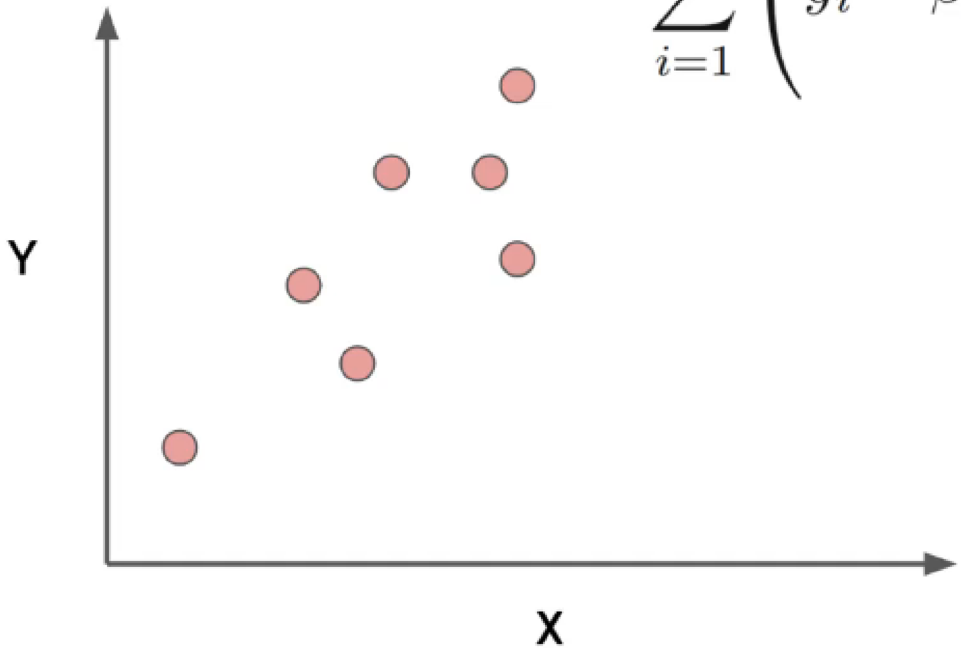


How to fix this Mathimaticlly?

Ridge Regression – Simple one Feature

- Let's imagine trying to reduce the Ridge Regression error term:

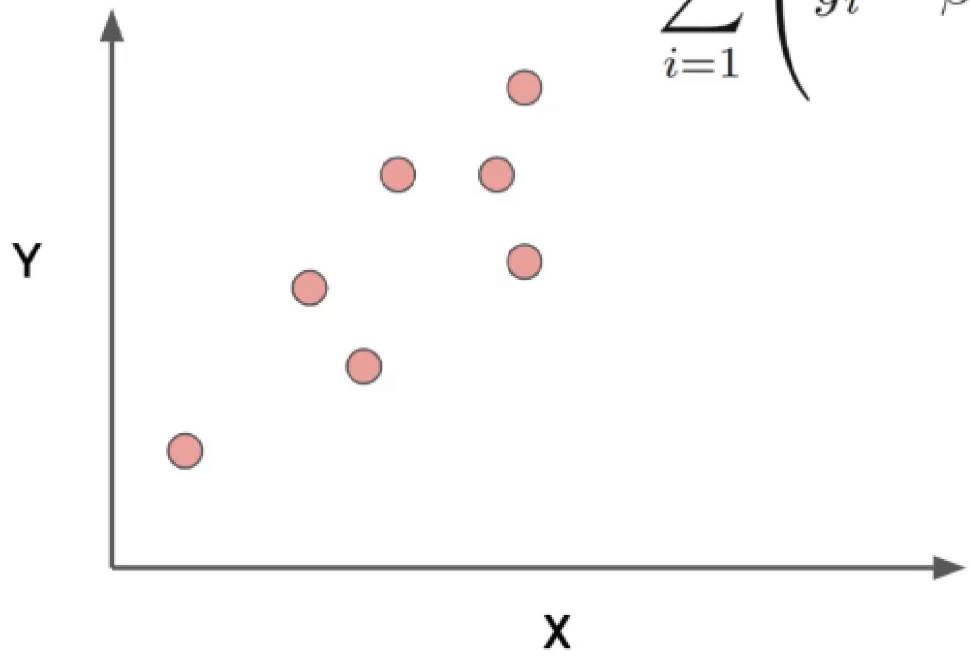
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$



Ridge Regression – Simple one Feature

- There is λ and the squared slope coefficient.

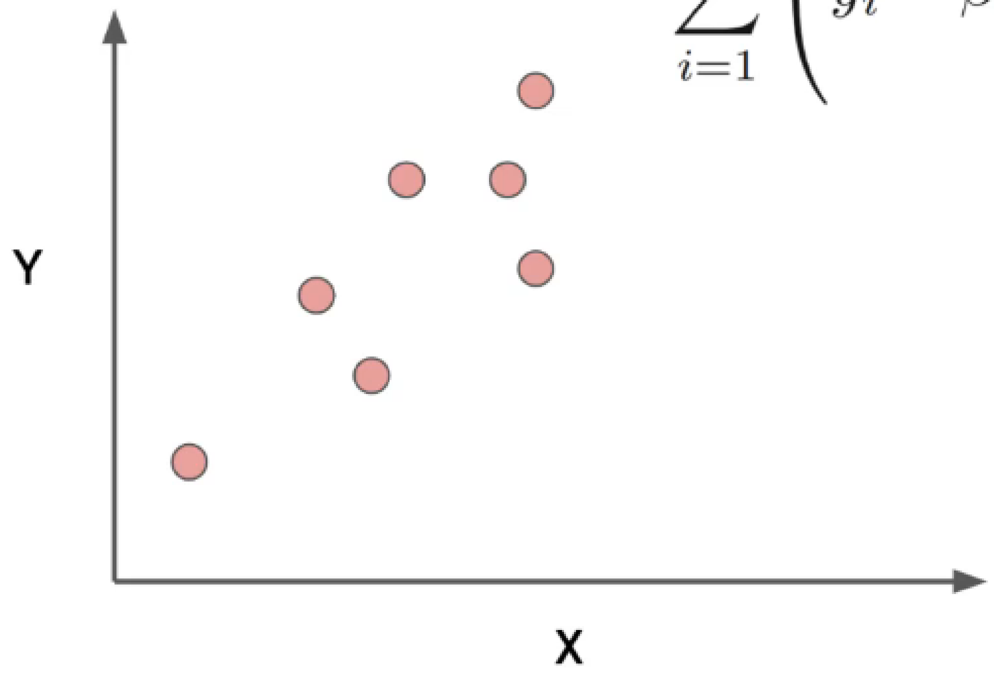
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda\sum_{j=1}^{p}\beta_j^2}$$



Ridge Regression – Simple one Feature

- Let's assume λ = 1

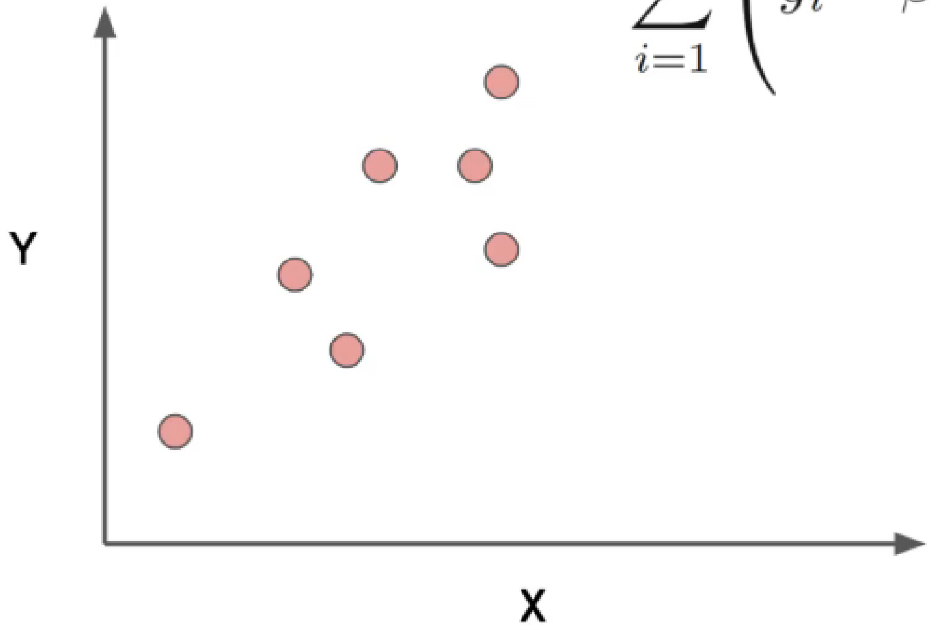
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$



Ridge Regression – Simple one Feature

- This punishes a large slope for $\hat{y} = \boxed{\beta_1} x + \beta_0$

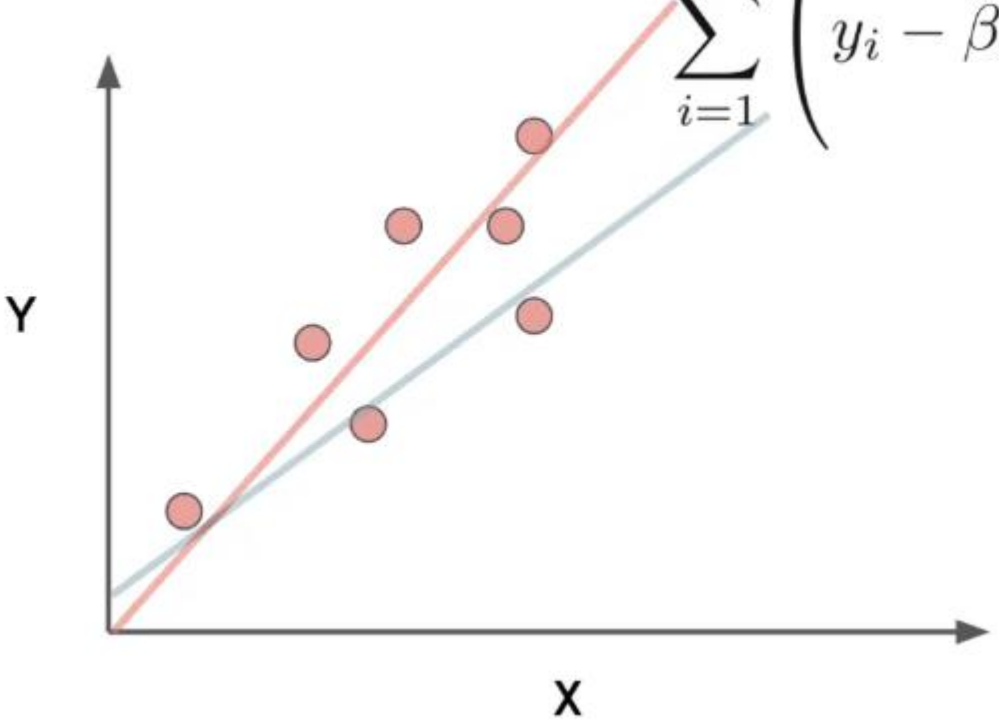
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda\sum_{j=1}^{p}\beta_j^2}$$



Ridge Regression – Simple one Feature

- For single feature this lowers slope

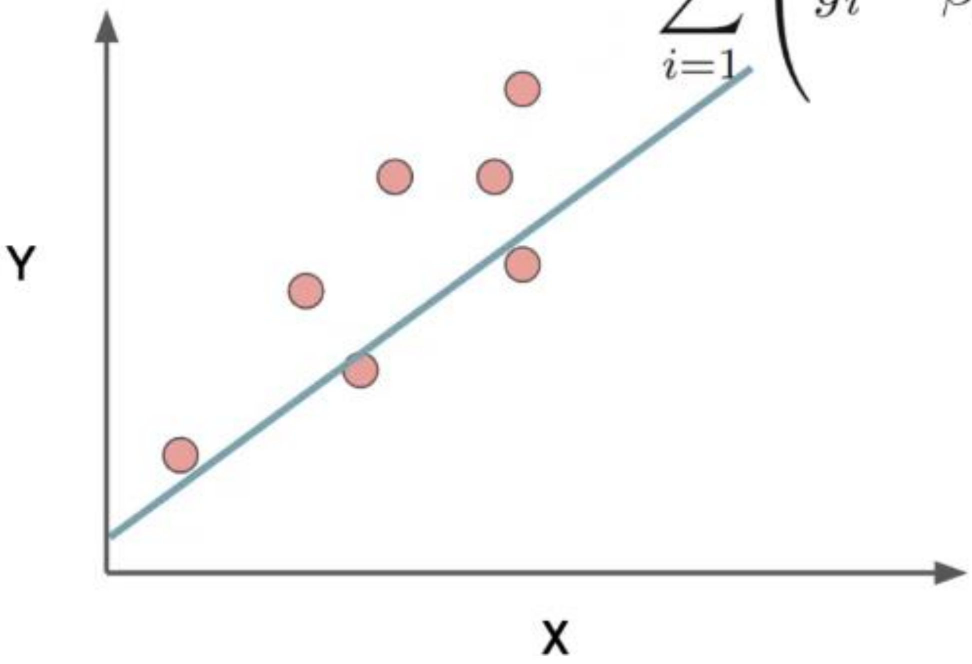
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda\sum_{j=1}^{p}\beta_j^2}$$



Ridge Regression – Simple one Feature

- At the cost of some additional bias (error in training set)

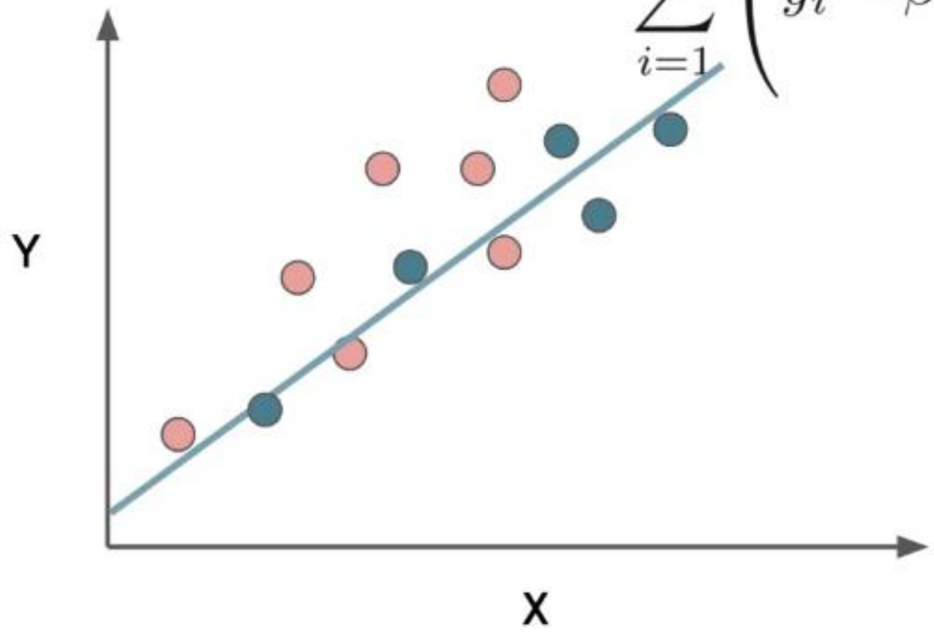
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\boxed{\lambda\sum_{j=1}^{p}\beta_j^2}$$



Ridge Regression – Simple one Feature

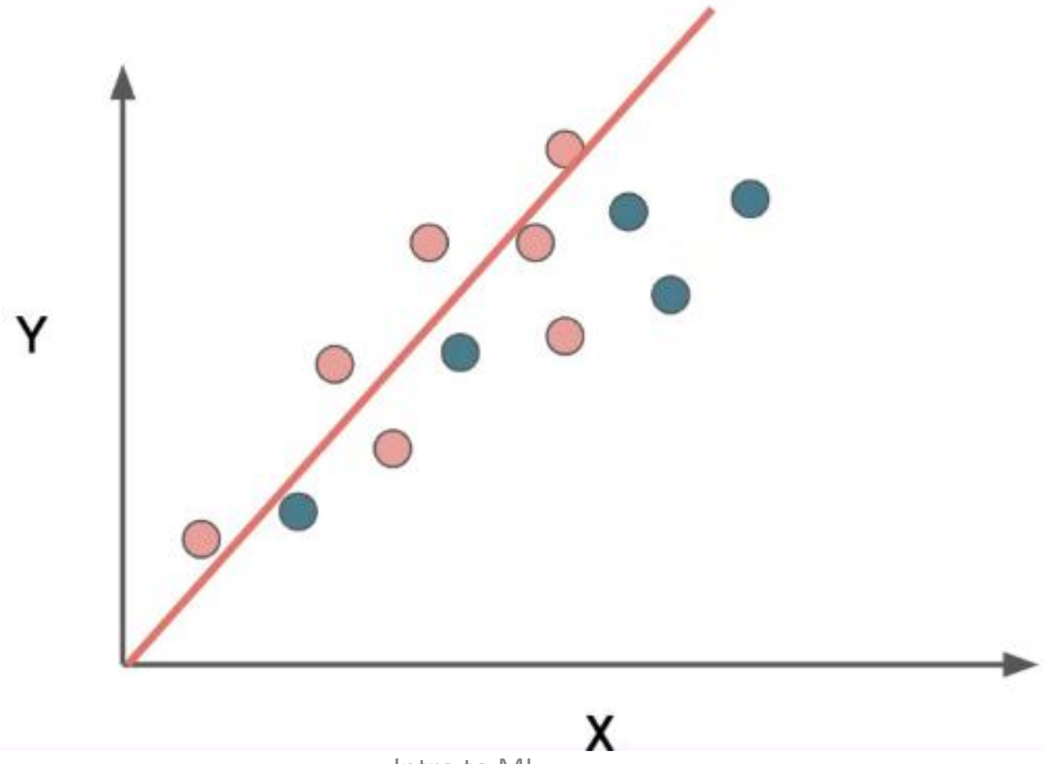
- We generalize better to unseen data

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\boxed{\lambda\sum_{j=1}^{p}\beta_j^2}$$



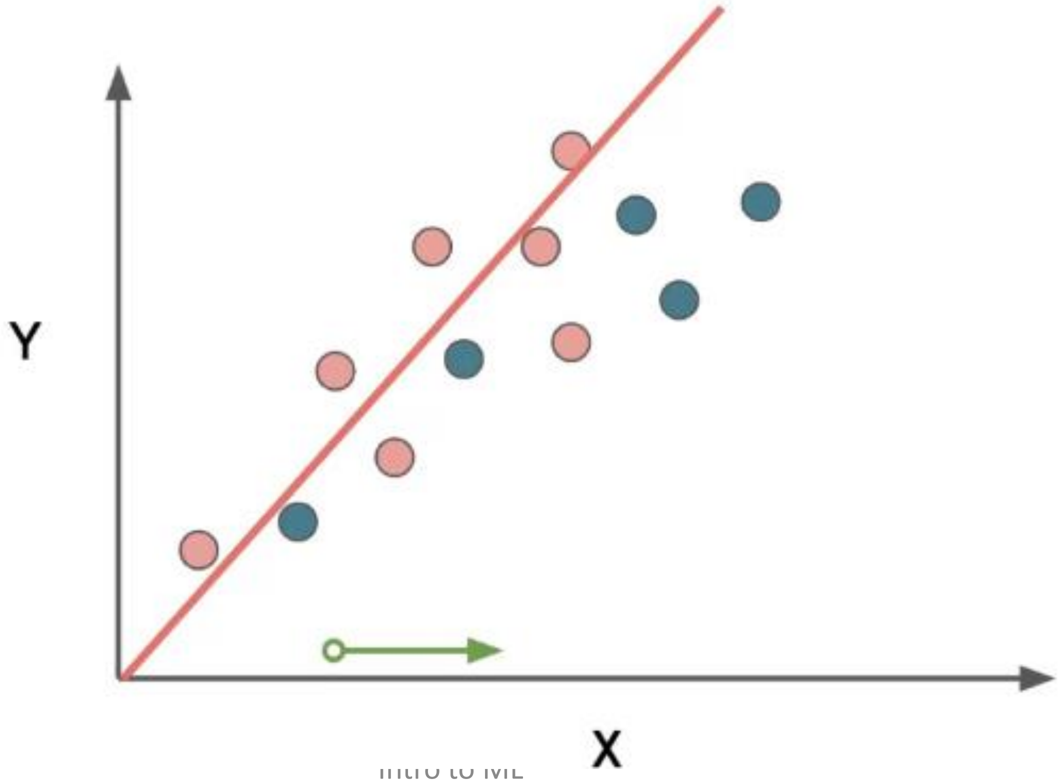
Ridge Regression – How we got better with it

- Consider overfitting to training set:



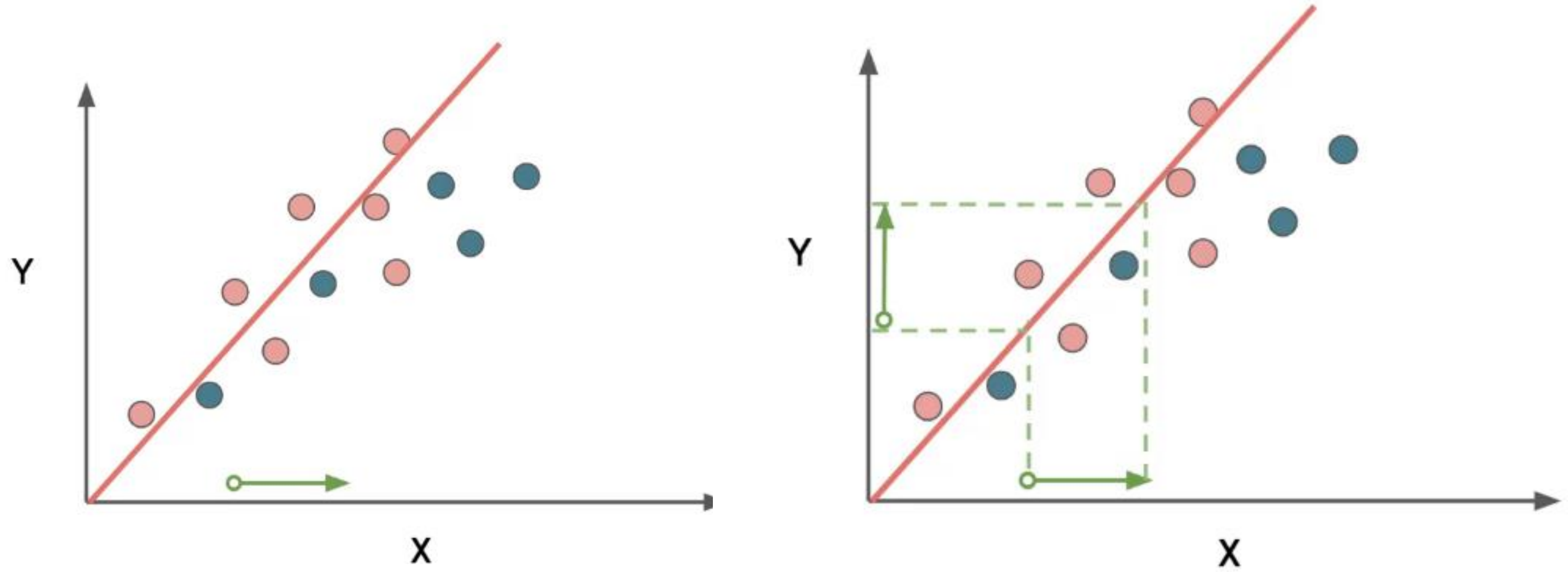
Ridge Regression – How we got better with it

- An increase in X results in a greater y response:



Ridge Regression – How we got better with it

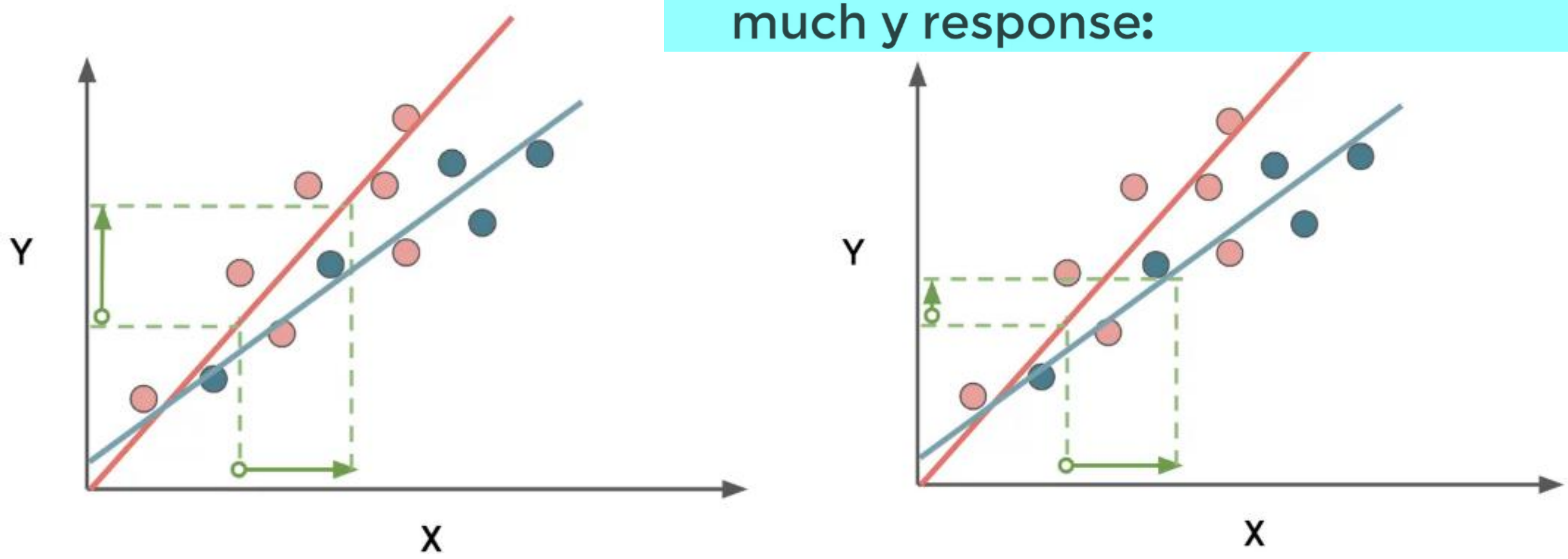
- An increase in X results in a greater y response:



Ridge Regression – How we got better with it

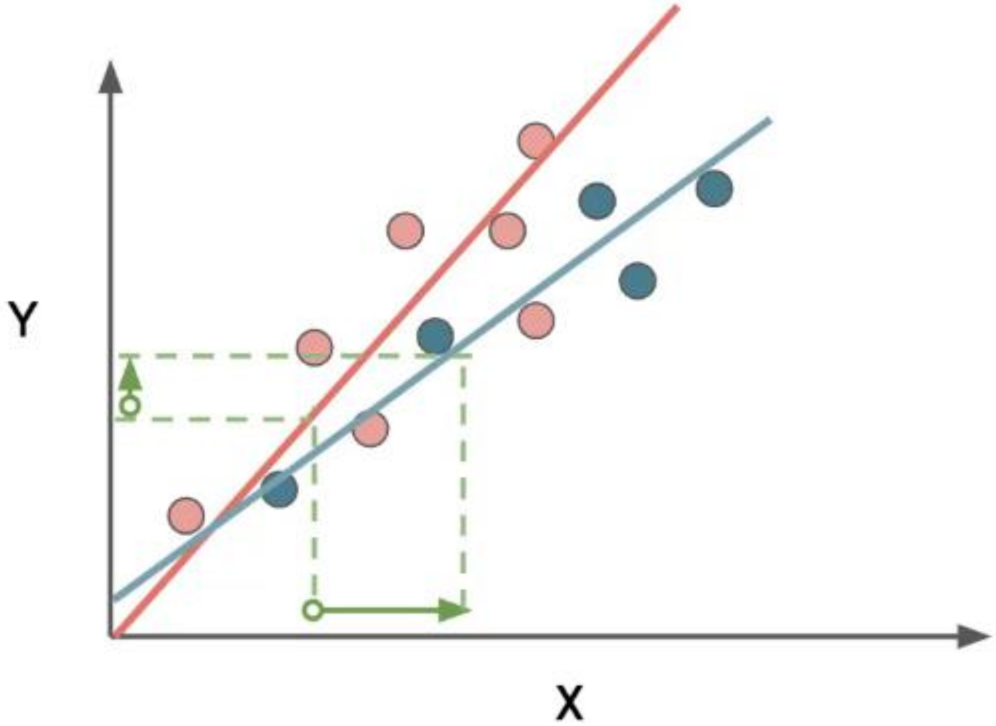
- Compare to a more generalized model that used Ridge Regression:

- Same feature change does not produce as much y response:



Ridge Regression – How we got better with it

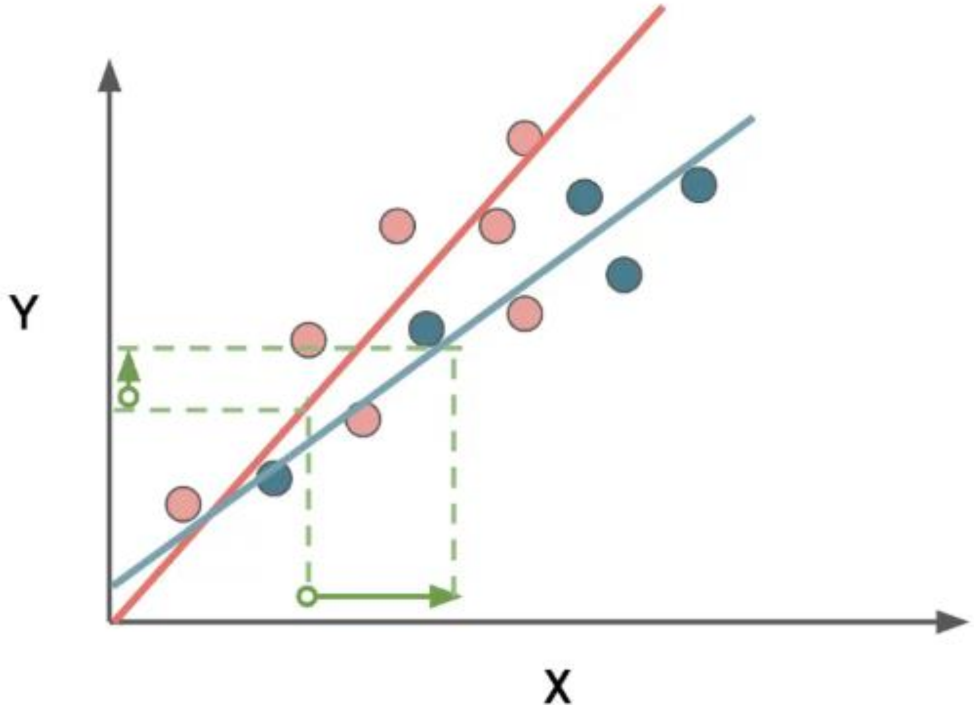
- Trying to minimize a squared Beta term leads us to punish larger coefficients.



$$\lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge Regression – How about Lambda ?

- What about the lambda term? How much should we punish these larger coefficients?



$$\lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge Regression – How about Lambda ?

- We simply use cross-validation to explore multiple lambda options and then choose the best one!

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Best LAMBDA : This what the Algorithm find to us, once giving a range of lambda values.

Ridge Regression – in Python - SKLEARN

- Important Note!
  - Sklearn refers to lambda as alpha within the class call!

$$\text{Error} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \boxed{\lambda}\sum_{j=1}^{p}\beta_j^2$$

Ridge Regression – in Python - SKLEARN

- Important Note!
  - For cross validation metrics, sklearn uses a “scorer object”.
  - All scorer objects follow the convention that **higher** return values are **better** than lower return values.

Ridge Regression – in Python - SKLEARN

- Important Note!
  - For cross validation metrics, sklearn uses a "scorer object".
  - All scorer objects follow the convention that **higher** return values are **better** than lower return values.