Regularization

Sometimes when we train a model it will start to overfit. A way to avoid overfitting data (especially for models like linear regressions that are heavily affected by outliers) we can use regularization. This will lead to a more general model that is technically less accurate but generalizes to the data better.

1. Ridge / L2 Regression

(used to reduce overfitting)

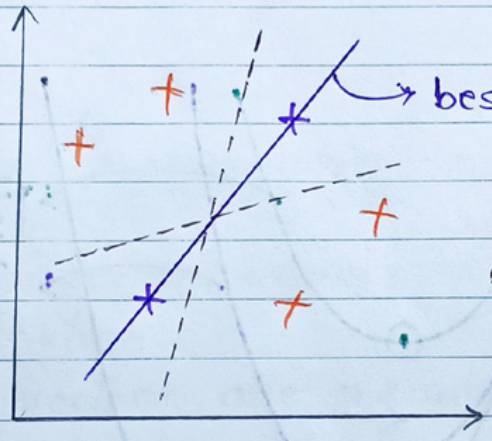
cost function = 0.

Training data: low bias

Testing data: low/high variance

- If the data (test data) is near to best fit line then performance will be good. (low variance)
- If the test data is far (away) to best fit line then performance will be bad. (high variance)

aim: To reduce overfitting



→ we create multiple lines to improve performance of test data.

cost Function:

$$\text{cost function} = \frac{1}{m}\sum_{i=1}^{m}\left\{h_\theta(x)^{(i)} - y^{(i)}\right\}^2 + \lambda(\text{slope})^2$$

λ = hyperparameter

Eg.

$h_\theta(x) = \theta_0 + \theta_1 x \longrightarrow$ slope $= \theta_1$

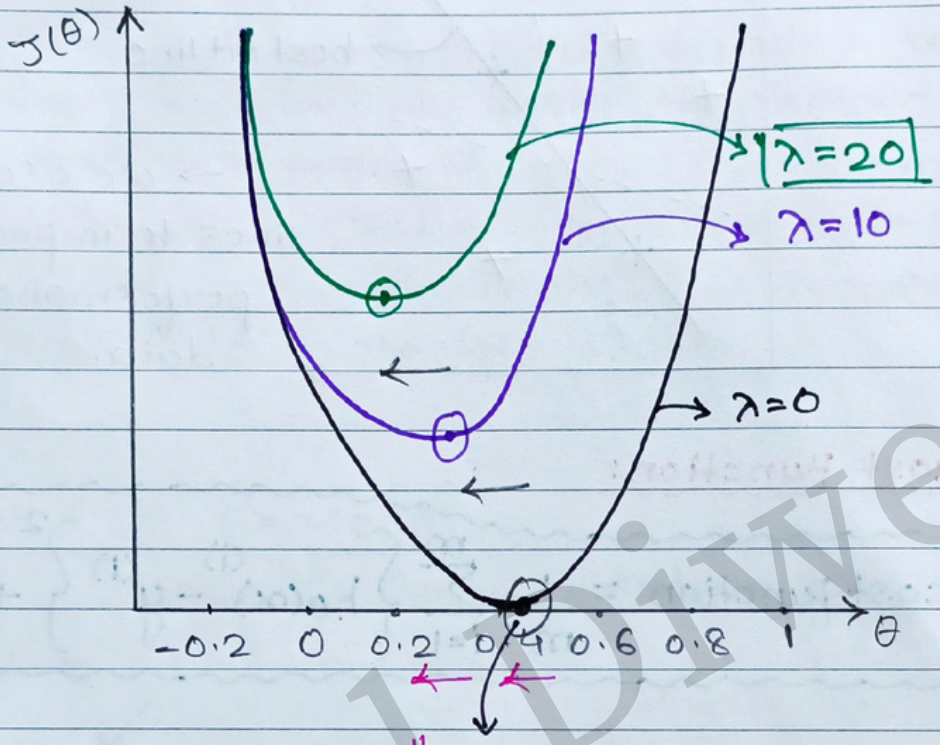
If multiple features are present, then

$$(\text{slope})^2 = \sum_{i=1}^{n}(\text{slope})^2$$

slope = slope of different lines.

If $\lambda = 0$, cost function is same as linear regression's cost function.

Relationship between slope and λ



"shifting towards zero" global minima

- Global minima gets shifted towards left with increase in λ.

cost function = 0 + (slope)² · λ
= +ve ↓↓

- change θ value to create another best fit line.

$$\lambda \propto \frac{1}{\text{slope}}$$

* inversly proportional.

λ = It make sure that our line doesn't overfit.

$\lambda \geq 0$, is a complexity parameter that controls the amount of shrinkage:

the larger the value of λ, the greater the amount of shrinkage.
The coefficient are shrunk towards zero.

*θ value never becomes zero!

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + 0.95 x_1 + 0.82 x_2 + 0.10 x_3$$

It will get deleted.

∴ Ridge Regression is used to introduce bias to the data in order to generalize the data and increase bias.
This is useful if you don't have much training data.

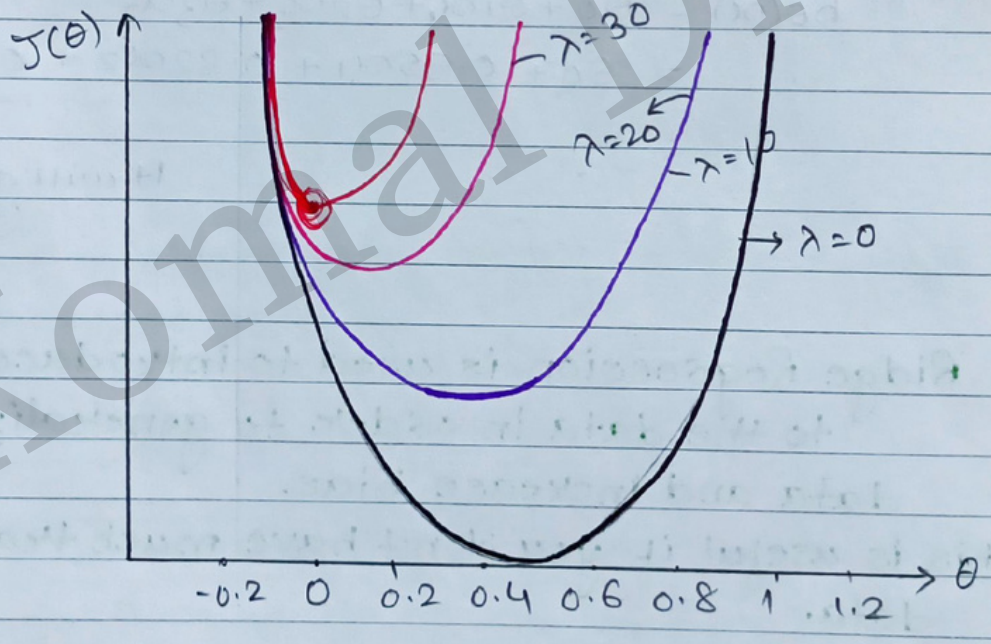
Lasso Regression

(L1 Regularization / L1 Norm)

- It is used to reduce the features. It helps in feature selection.

cost Function:

$$\text{cost function} = \frac{1}{m}\sum_{i=1}^{m}\left(h\theta(x)^{(i)} - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{m}|\text{slope}|$$



$$h\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + 0.54 x_1 + 0.23 x_2 + 0.10 x_3$$

least correlated.

- If data has outliers → use Ridge Regression.

Lasso = Least Absolute Shrinkage and Selection Operator Regression.

- Lasso regression tends to eliminate the weights of the least important features by setting their weights to zero.

Elastic Net

→ combination of L1 and L2 Regularization.

$$\text{cost function} = \frac{1}{m}\sum_{i=1}^{m}\left\{h_\theta(x)^i - y^{(i)}\right\}^2 + \underbrace{\lambda(\text{slope})^2}_{L2} + \lambda|\text{slope}|$$

$\lambda|\text{slope}|$ → L1

can be changed to MAE, RMSE, MSE.

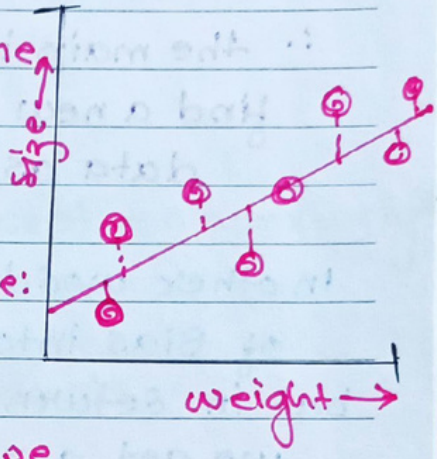
Notes taken from John Starmer of Stat Quest

Youtube Videos

Komal Diwe

Regularization: Ridge (L2) Regression.

we find the line that results in the minimum sum of squared residuals.

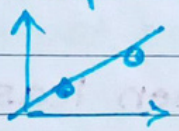


∴ we end up with the eqn of the line:

size = 0.95 + 0.75 × weight

0.95 → y-intercept, 0.75 → slope

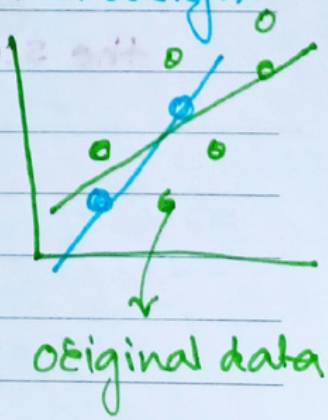
when we have of lot of measurements, we can be fairly confident that least squares line accurately reflects the relationship between size and weight.



But what if we only have two measurements? we fit new line. since the new line overlaps the two data points, the minimum sum of squared residuals = 0.

∴ New line eqn: size = 0.4 + 1.3 × weight

sum of the squared residuals for testing data is large which means the new line has high variance.



In M2, new line (blue) is overfit to training data.

∴ the main idea behind Ridge Regression is to find a new line that doesn't fit the training data as well.

In other words, we introduce a small amount of Bias into how the new line is fit to the data but in return for that small amount of bias, we get a significant drop in variance.

∴ Ridge Regression can provide better long term predictions.

when least squares determines values for the parameters in this equation

↓

size = y-axis intercept + slope × weight

it minimizes....
the sum of the squared residuals.

In contrast,

when Ridge Regression determines values for the parameters in this equation....

↓

size = y-axis intercept + slope x weight

.... it minimizes

the sum of the squared residuals

+

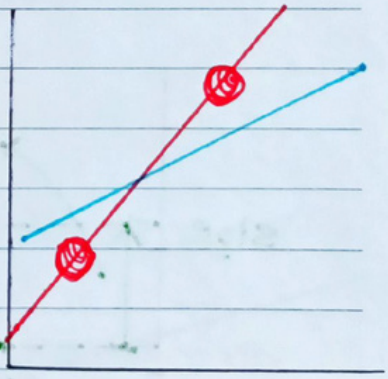
λ x the slope^2

↳ lambda.

this part adds a penalty to the traditional least square method.

and lambda(λ) determines how severe that penalty is.

→ the sum of squared residuals for the ~~residuals~~ least square fit is 0 (because the line overlaps the data points). and the slope is 1.3.



$\lambda = 1$

$\therefore \quad 0 + \lambda \times (1.3)^2 = 0 + 1 \times (1.3)^2 = 1.69$

for blue line → slope = 0.8

$(0.3)^2 + (0.1)^2 + (0.8)^2$ x 0.1 altogether

0.74

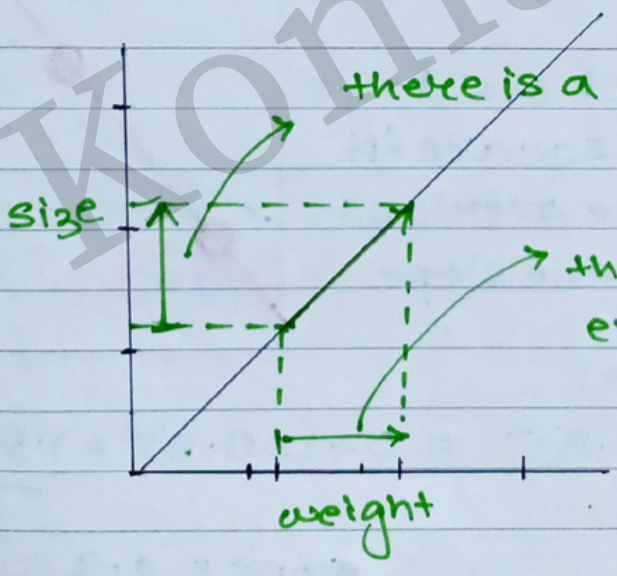
∴ Ridge Regression line :-

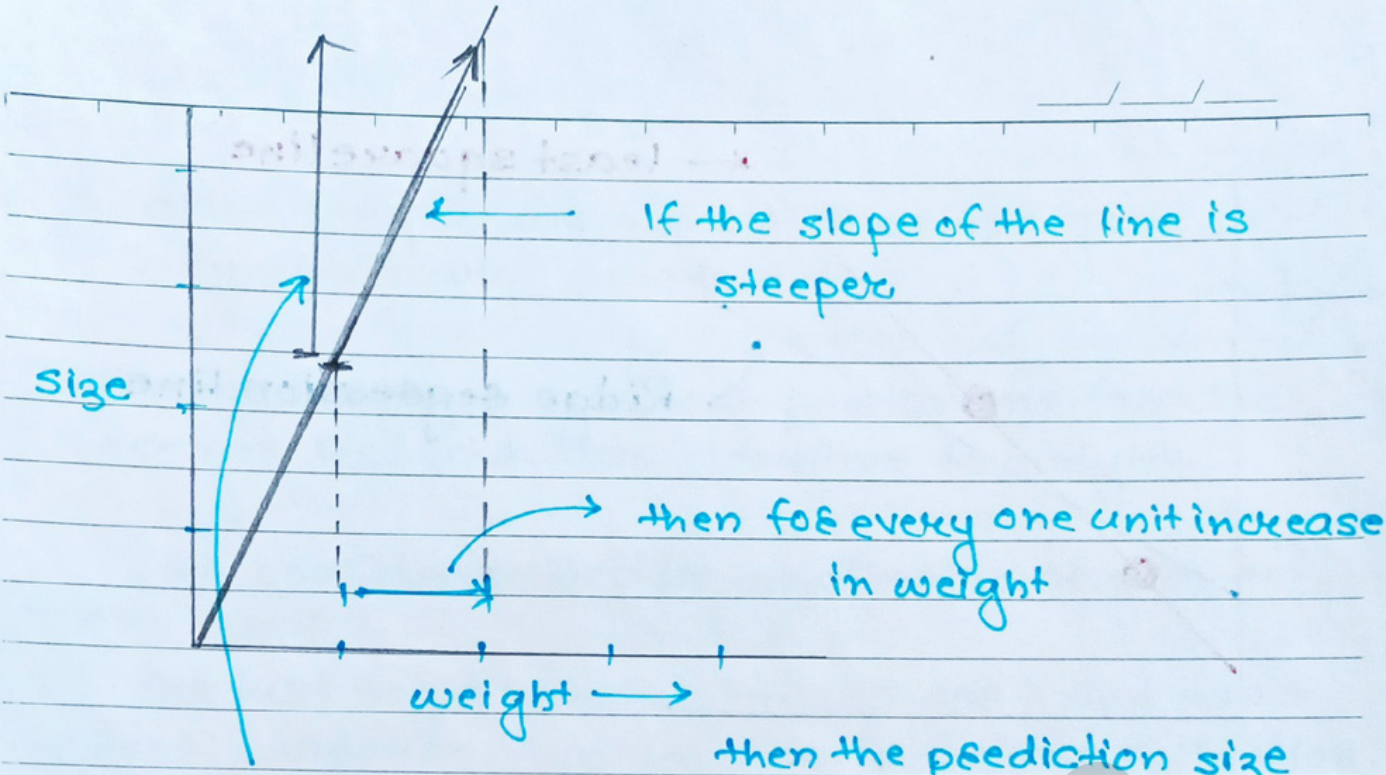
Red = 1.69 , Blue = 0.74

Thus, if we wanted to minimize the sum of the squared residuals plus the Ridge Regression penalty, we would choose the Ridge Regression Line over the least square line.

without the small amount of Bias that the penalty creates, the least squares fit has a large amount of Variance.

In contrast, the Ridge Regression line, which has the small amount of Bias due to the penalty, has less variance.

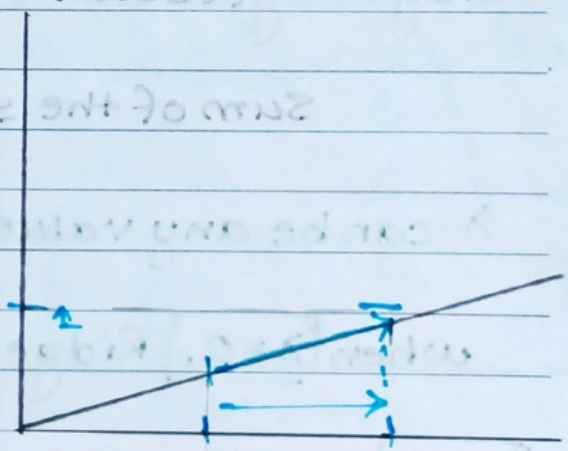




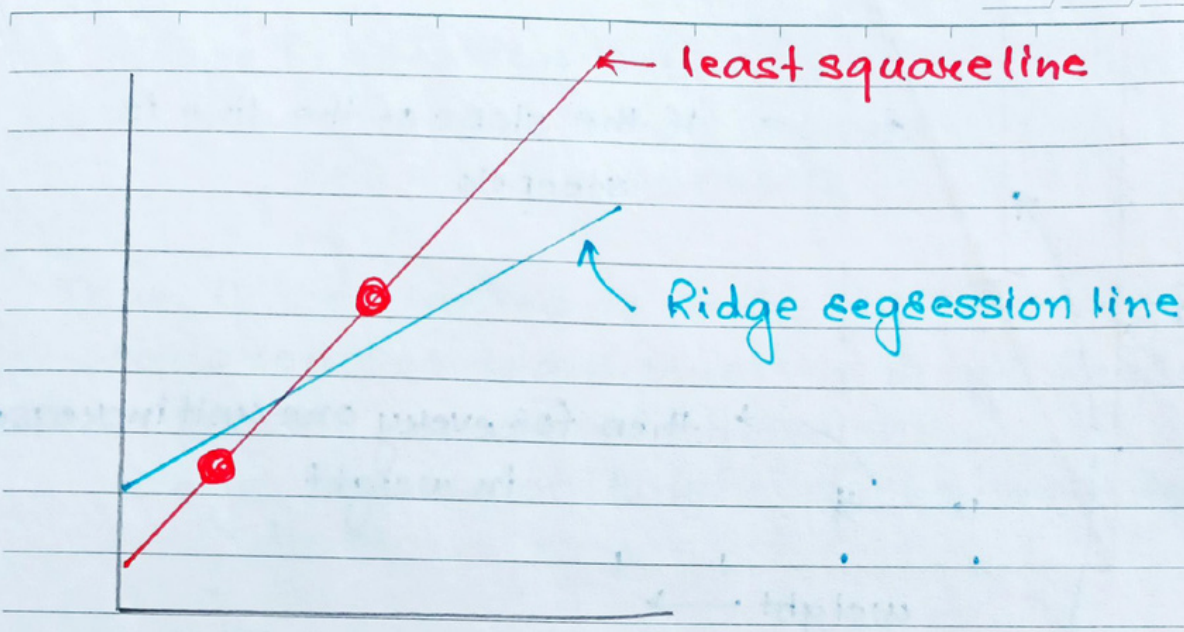
In other words, when the slope of the line is steep, then the prediction for size is very ~~sensitivity~~ sensitive to relatively small changes in weight.

when the slope is small, then for every one unit increase in weight

the prediction for size barely increases.



In other words, when the slope of the line is small, then predictions for size are much less sensitive to changes in weight.



The Ridge Regression penalty resulted in a line that has a smaller slope ...
which means that predictions made with the Ridge Regression line are less sensitive to weight than the least square line.

Ridge Regression (RR)

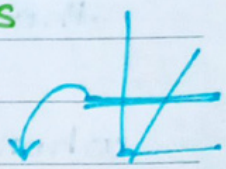
Sum of the squared residuals + $\lambda \times (\text{slope})^2$

λ can be any value from 0 to positive infinity.

when $\boxed{\lambda = 0,}$ Ridge regression line = least square line

$\boxed{\lambda = 1}$ RR ended up with a smaller slope than the least square line.

and the larger we make λ, the slope gets asymptotically close to 0.



So, the larger λ gets, our predictions for size becomes less and less sensitive to weight.

→ so how do we decide what value to give λ?

we just try a bunch of values for λ and use cross-validation, typically 10-fold cross validation, to determine which one results in the lowest variance.

.... uptill now RR was for continuous variable.

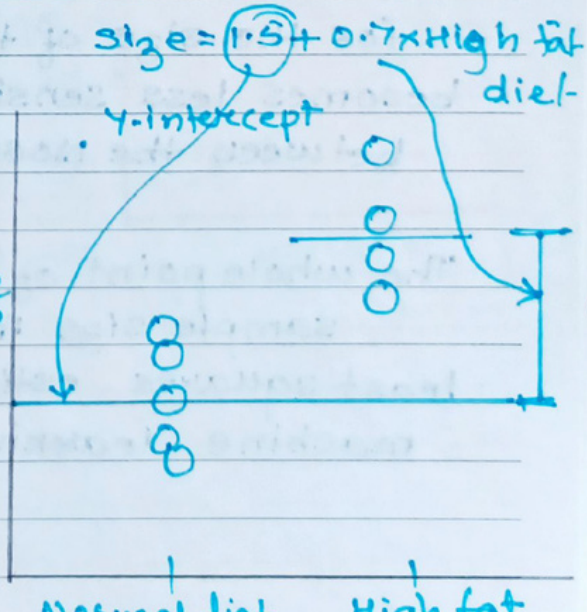
However, RR also works when we use discrete variable.

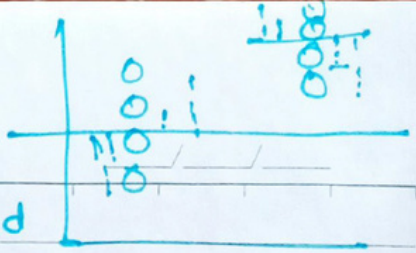
Discrete variable:

y-intercept → corresponds to the average size of the mice on the Normal diet.

Size = 1.5 + 0.7 × High fat diease

sum of these two is the prediction for the size of the mice on the High fat diet.





these distance between the data and the means are minimized.

when RR determines value for the parameters in the equation...

... it minimizes →

the sum of the squared residuals

+

$\lambda \times (\text{diet difference})^2$

$\lambda = 0$, least squared-error = RR line

$\lambda = 1$, then only way to minimize the whole eqn is to shrink diet distance down.

In other words, as λ gets larger, our prediction for the size of the mice on the high fat diet becomes less sensitive to the difference between the Normal diet and High-fat diet.

The whole point of doing RR is because small sample size like these can lead to poor least squares estimates that result in terrible machine learning predictions.

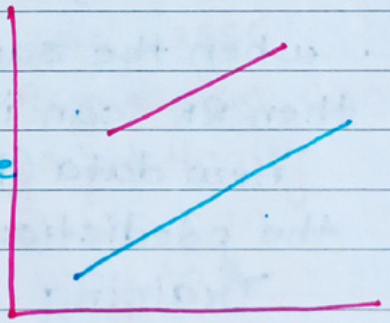
Ridge Regression can also be applied to Logistic Regression.

= the sum of the likelihoods + $\lambda(\text{slope})^2$

Note:- when applied to Logistic Regression, Ridge Regression optimizes the sum of the likelihoods instead of the squared residuals because Logistic Regression is solved using maximum likelihood.

" Ridge Regression helps reduce variance by shrinking parameters and making our predictions less sensitive to them

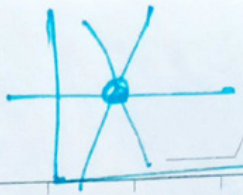
In general, RR penalty contains all of the parameters except for the y-intercept.



the sum of the squared residuals

+

$\lambda(\text{slop}^2 + \text{diet distance}^2)$.



least squares can't find a single optimal solution, since any line line that goes through the dot will minimize the sum of the squared residuals.

but RR can find a solution with cross validation and the RR ~~policy~~ penalty that favours smaller parameter values.

↓

sum of the squared residuals
+
$\lambda * (\text{slope})^2$

Summary:

- when the sample sizes are relatively small, then RR can improve predictions made from new data (i.e. reduce variance) by making the predictions less sensitive to the Training data.

RR penalty itself is λ times the sum of all squared parameters, except for the y-intercept. and λ is determined using cross validation.

Lasso Regression: (L1)

Ridge Regression Penalty = $\lambda \times (\text{slope})^2$

" Lasso Regression →

sum of all the squared residuals
+ $\lambda \times |\text{slope}|$

Lasso Regression Penalty contains all of the estimated parameters except for the y-intercept.

∵ Ridge and Lasso Regression shrink parameters, they don't have to shrink them all equally.

Big difference between Ridge and Lasso Regression is that Ridge Regression can only shrink the slope asymptotically close to 0 while Lasso Regression can shrink the slope all the way to 0.

LR can exclude useless variables from equations, better than RR at reducing the variance in models that contain a lot of useless variables.

Elastic Net Regression:

Elastic-Net Regression starts with least-squares then combines the Lasso Regression penalty with the Ridge Regression penalty.

$$\text{sum of the squared residuals} + \lambda_1 \times |\text{variable}_1| + \ldots\ldots + |\text{variable}_n| + \lambda_2 \times (\text{variable}_1)^2 + \ldots\ldots + (\text{variable}_n)^2$$

Note: LR and RR penalty get their own λs.

The hybrid Elastic Net Regression is especially good at dealing with situations when there are correlations between parameters.

This is because on it's own, Lasso Regression tends to pick just one of the correlated terms and eliminates the others whereas RR tends to shrink all of the parameters for the correlated variables together.

By combining LR and RR,
Elastic-Net Regression groups and shrinks the parameters associated with the correlated variables and leaves them in eqn or removes them all at once.