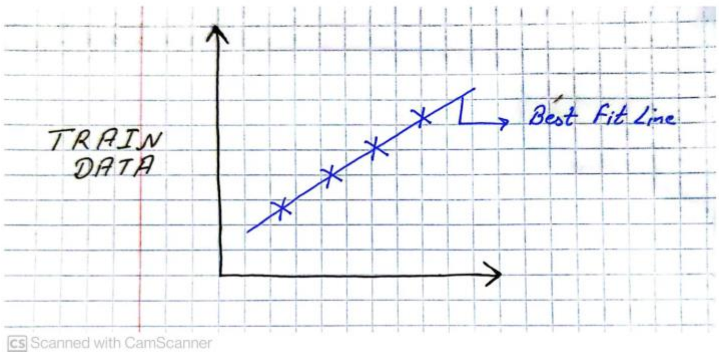
Lasso & Ridge Regression

Ridge and Lasso Regression are two of the most popular algorithms used in the field of machine learning. Both algorithms are used to reduce the errors of a **linear regression** model, but there are key differences between the two which make them suitable for different types of problems.

Ridge Regression
Ridge regression is a type of regularized regression model. This means it is a variation of the standard linear regression model that includes a regularized term in the cost function. The purpose of this is to prevent Overfitting. Ridge Regression adds an L2 regularization term to the linear equation. That's why it is also known as L2 Regularization or L2 Norm.

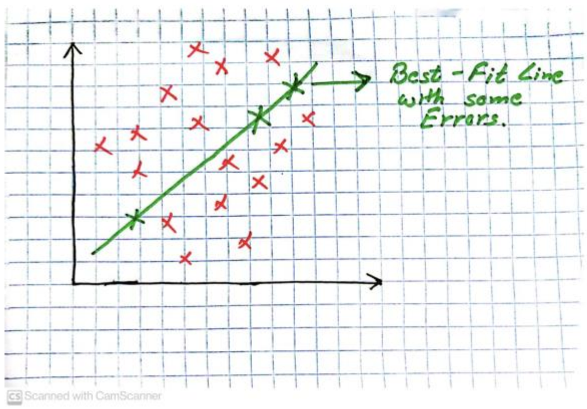
The main aim of Ridge Regression is to reduce Overfitting.

Graphical Intuition of Ridge Regression
Let's say we built a linear regression model with the best-fit line passing through all the train data points.



As the best-fit line is passing through all the train data points this means Cost Function = 0. This means that the train data will have very good accuracy but the test data will have very bad accuracy. This means it will cause **overfitting**.

In order to avoid this problem or overfitting, we need to get a best-fit line with some errors or residuals. For this, we can use **Ridge Regression.**



How does Ridge Regression work?

Ridge Regression works by adding a penalty term to the cost function of a linear regression model, called the regularization term. This regularization term prevents the model from overfitting by penalizing the large coefficients. The regularization parameter determines how much the model should be penalized. By increasing the regularization parameter, the coefficients are shrunk toward zero.

Cost Function (Ridge Regression)

First, let's check the Cost Function of a linear regression model.

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h\theta(x)^i - y^i)^2$$

$h\theta(x)^i$ → Predicted Points

y^i → Actual Points

$\sum_{i=1}^{m} (h\theta(x)^i - y^i)^2$ → Mean Squared Error

As mentioned above ridge regression adds a parameter to the linear regression cost function to avoid overfitting so the cost function of ridge regression looks like this:

$$Cost\ Function = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda\,(slope)^2$$

$$\lambda = Hyperparameter$$

Cost Function Ridge Regression

- f λ = 0 then the cost function of Ridge Regression and the cost function of linear regression is the same.
- If even adding the hyperparameter to the original cost function the training accuracy won't improve then it will keep on changing the value of λ and try to find the best λ parameter.
- As the cost function is changing after adding a hyperparameter, the model will also change the best-fit line.

Lasso Regression

Lasso Regression is also a type of regularization linear model. It also adds a penalty term to the cost function but it adds L1 regularization instead of L2 regularization and hence is also known as **L1 Regularization** or **L1 Norm**. This term is the sum of the absolute values of the coefficients, multiplied by a constant lambda.

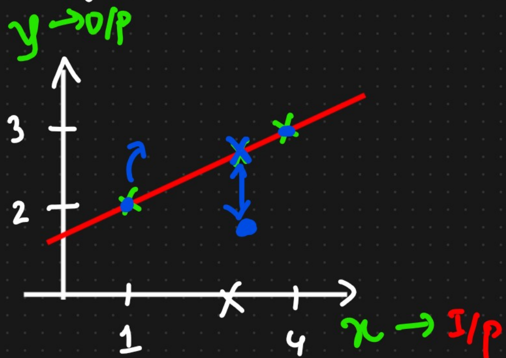
The main aim of Lasso Regression is to reduce the features and hence can be used for Feature Selection.

$$Cost\ Function\ = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda \sum_{i=1}^{n}|slope|$$

$$\lambda = Hyperparameter$$

Cost Function Lasso Regression

Ridge and Lasso Regression Math Intuition



Training Set

x	y
1	2
4	3

⇒ Linear Regression

$x \rightarrow \boxed{\quad} \rightarrow y$

↓

Model

↓

Overfitting

Train Accuracy = 90%

Test Accuracy = 70% ↓↓

Low Bias
High Variance

Underfitting

Train Accuracy = 60%
Test Data = 62%

High Bias
High Variance

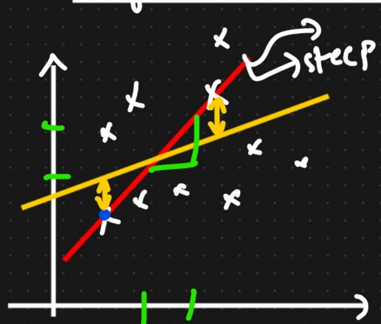
→ **Generalized Model**

Train Acc = 90%

Test Acc = 89%

Low Bias &
Low Variance

Ridge Regression (L_2 Regularization)



$\boxed{\lambda = 1}$

Cost function

Residual Error

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 \quad \downarrow\downarrow\downarrow$$

penalizing

$$\Rightarrow \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 + \underline{\underline{\lambda (\text{slope})^2}} =$$

$$= 0 + 1(2)^2$$

$$= 4 \downarrow\downarrow\downarrow$$

① Overfitting is removed }

$y = mx + c$

$$\Rightarrow \{\text{small value}\} + 1(1.3)^2 \qquad \lambda (m^2)$$

$$\Rightarrow \approx \underline{\underline{2.05}} \downarrow\downarrow\downarrow$$

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

small

Lasso Regression (L_1 Regularization)

↓↓↓↓

$$\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 + \lambda \,|\text{slope}| \qquad \lambda |m|$$

$$\hookrightarrow \lambda |m_1 + m_2 + m_3 + \cdots m_n|$$

1. Overfitting prevent
2. Feature selection