# Content Generation Assignment

Department:- ICT
Sem:- 5
Subject: Machine Learning

Faculty Mentor:- Prof. C D Parmar
Prof. Nishith Kotak

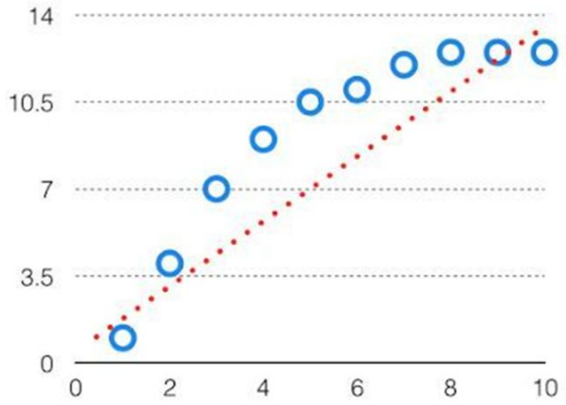
Created By :-
Ritesh Sanchala - 92200133001
Fenil Vadher - 92200133023
Aryan Langhnoja - 92200133030
Abhay Nathwani - 92310133007

# Topic:
# From Training Triumph to Testing Troubles: The Role of Regularization in Machine Learning

## Fundamentals of Machine Learning :- Bias and Variance

## What is Bias?

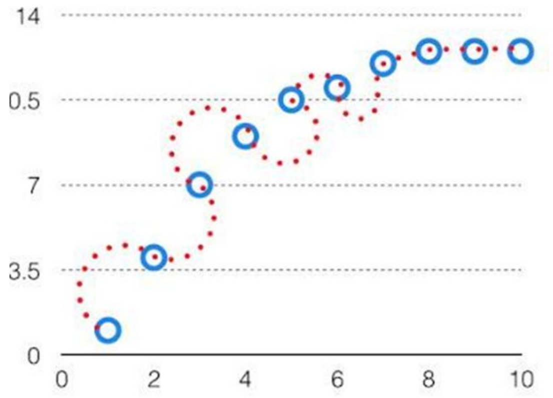
- Bias refers to the error due to overly simplistic assumptions in the learning algorithm. These assumptions make the model easier to comprehend and learn but might not capture the underlying complexities of the data. It is the error due to the model’s inability to represent the true relationship between input and output accurately. When a model has poor performance both on the training and testing data means high bias because of the simple model, indicating underfitting.
- In simple Words , the inability for a machine learning model to capture the true relationship is called bias.



## Variance

### What is Variance?

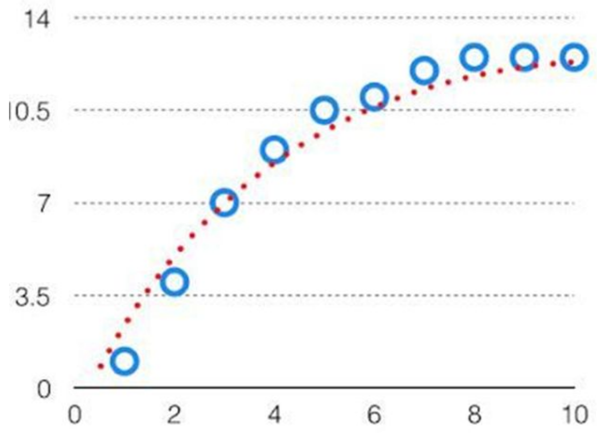
- Variance, on the other hand, is the error due to the model's sensitivity to fluctuations in the training data. It's the variability of the model's predictions for different instances of training data. High variance occurs when a model learns the training data's noise and random fluctuations rather than the underlying pattern. As a result, the model performs well on the training data but poorly on the testing data, indicating overfitting.
- In simple words, The model's sensitivity to small fluctuations in the training data, leads to high variability in predictions across different datasets.



## Bias Variance Tradeoff

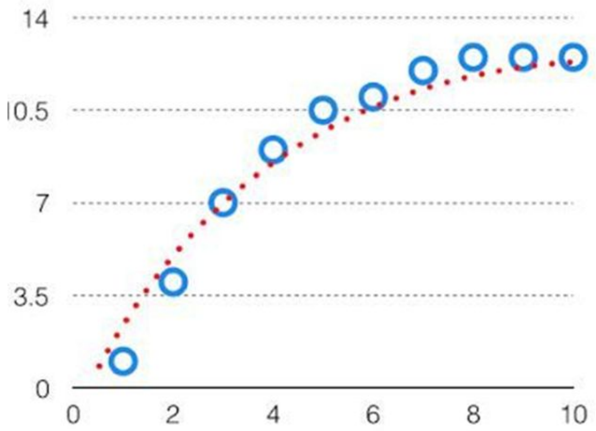
### What is Bias Variance Tradeoff ?

- If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



## What is Bias Variance Tradeoff ?

- If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



## Regression Basics :-

## Linear Regression : The Basics

### What is Linear Regression ?

Linear regression is one of the simplest and most widely used algorithms in machine learning. It establishes a linear relationship between a dependent variable (target) and one or more independent variables (features). The equation for linear regression is:

$$\mathbf{y = \theta_0 + \theta_1 x + \varepsilon}$$

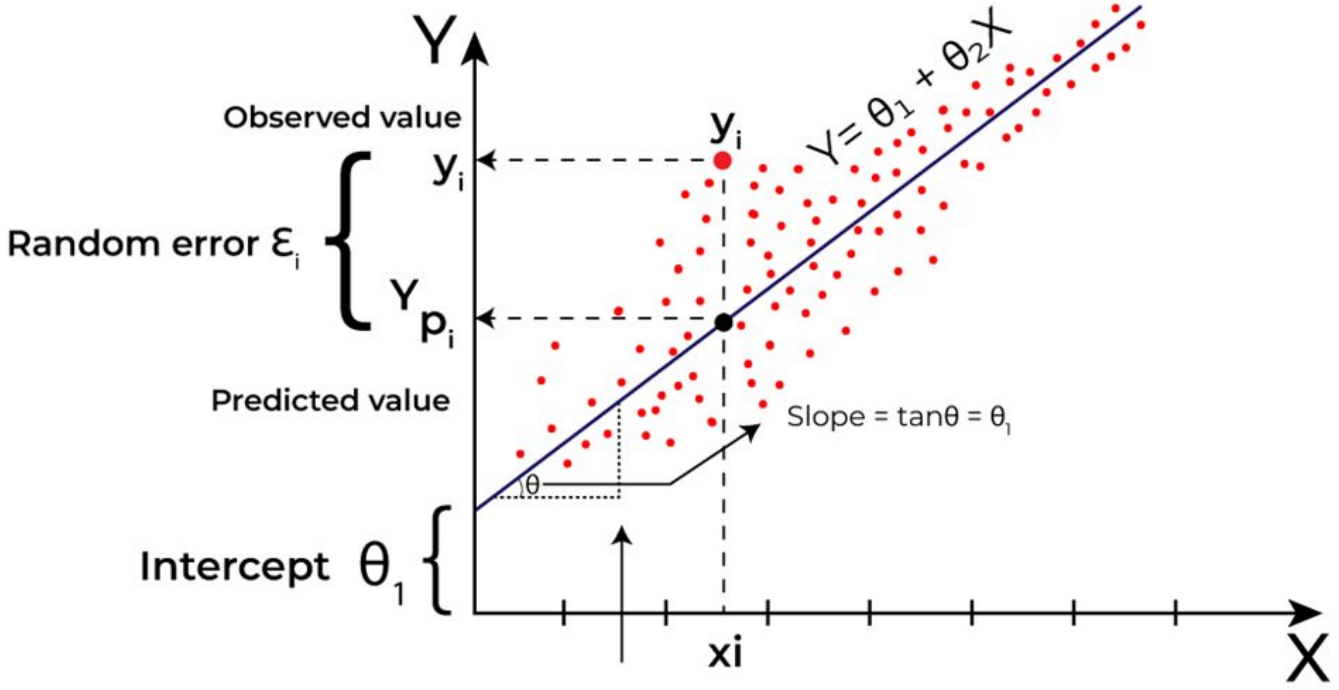
Where:

- $y$ is the predicted value,
- $x$ is the independent variable,
- $\theta_0$ and $\theta_1$ are the intercept and slope of the line,
- $\varepsilon$ represents the error term.

**Example:**

Consider predicting house prices based on the size of the house. The size (x) is the independent variable, and the price (y) is the dependent variable. Linear regression will calculate the best-fitting line to predict the price of a house based on its size.

## Linear Regression : The Basics


Y
Observed value
$y_i$
$y_i$
Random error $\varepsilon_i$
$Y_{p_i}$
Predicted value
$Y = \theta_1 + \theta_2 X$
$\theta$
Slope = $\tan\theta = \theta_1$
Intercept $\theta_1$
xi
X

## Multivariable Linear Regression : Expanding to More Features

While simple linear regression works well with one feature, many real-world problems involve multiple features. This is where **multivariable linear regression** comes in. The equation is expanded as:

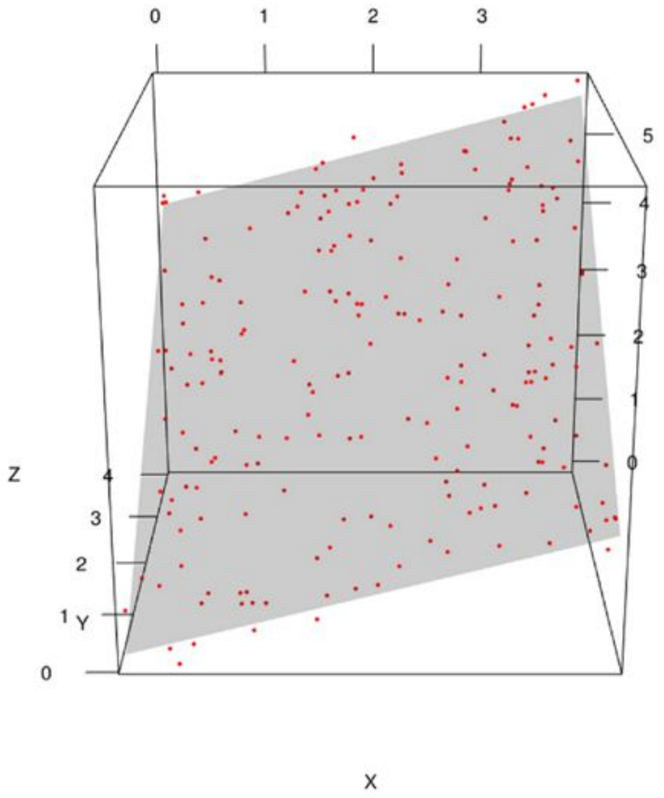
$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_{\square} x_{\square} + \varepsilon$$

**Example:**

When predicting house prices, multiple features such as house size, location, and the number of rooms may play a role. In this case, the model will learn the relationship between all these features and the house price, allowing for a more accurate prediction.

However, with more features comes the risk of **overfitting**. As more variables are introduced, the model may start to capture not just the underlying relationships but also the noise in the data. This is where regularization techniques come into play.

## Multivariable Linear Regression : Expanding to More Features



## Logistic Regression : For Classification Problems

While linear regression is used for predicting continuous values, **logistic regression** is employed for classification tasks, particularly binary classification. It uses a logistic function (sigmoid) to predict probabilities that a given input belongs to a particular class.

The equation for logistic regression is:

$$P(y=1|x) = 1 / (1 + e^{-(\theta_0 + \theta_1 x)})$$

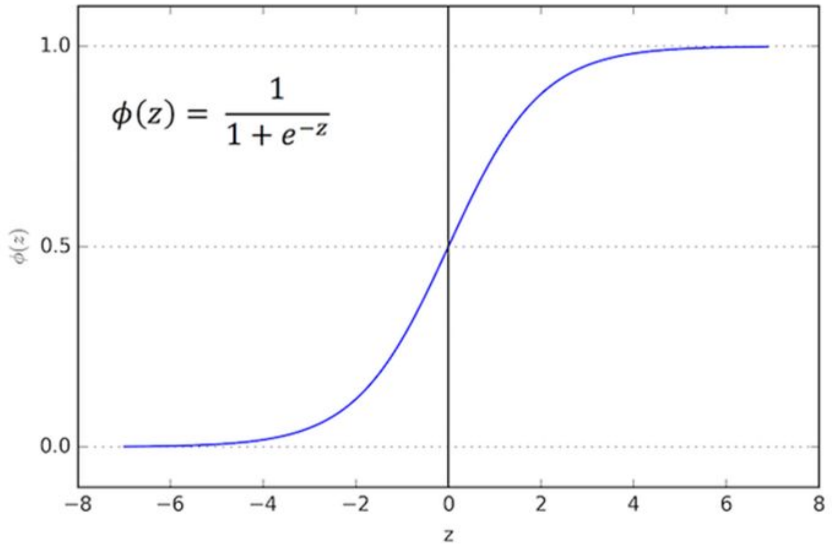
Where:

- $P(y=1|x)$ is the probability that the target variable y is 1 (e.g., spam) given the input features x.

**Example:**

Consider the problem of classifying emails as spam or not spam. Logistic regression will output the probability of an email being spam, and by setting a threshold (typically 0.5), we can classify the email as spam (1) or not spam (0).

## Logistic Regression : For Classification Problems



Similar to linear regression, logistic regression can also suffer from overfitting, particularly when the number of features is large. Regularization is used to address this by penalizing overly complex models and reducing the likelihood of overfitting.

## Underfitting and Overfitting

## Underfitting

### What is Underfitting ?

- A statistical model or a machine learning algorithm is said to have underfitting when a model is too simple to capture data complexities. It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data. In simple terms, an underfit model's are inaccurate, especially when applied to new, unseen examples. It mainly happens when we uses very simple model with overly simplified assumptions. To address underfitting problem of the model, we need to use more complex models, with enhanced feature representation, and less regularization.

- **The underfitting model has High bias and low variance.**

## Underfitting

**Reasons for Underfitting**

1. The model is too simple, So it may be not capable to represent the complexities in the data.
2. The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.
3. The size of the training dataset used is not enough.
4. Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.
5. Features are not scaled.

## Underfitting

**Techniques to Reduce Underfitting**

1. Increase model complexity.
2. Increase the number of features, performing feature engineering.
3. Remove noise from the data.
4. Increase the number of epochs or increase the duration of training to get better results.

## Overfitting

### What is **Overfitting** ?

- A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

- In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

- **The overfitting model has Low bias and High variance.**

## Overfitting

**Reasons for Overfitting**

1. High variance and low bias.
2. The model is too complex.
3. The size of the training data.

## Techniques to Reduce Overfitting

1. Improving the quality of training data reduces overfitting by focusing on meaningful patterns, mitigate the risk of fitting the noise or irrelevant features.
2. Increase the training data can improve the model's ability to generalize to unseen data and reduce the likelihood of overfitting.
3. Reduce model complexity.
4. Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
5. Ridge Regularization and Lasso Regularization.
6. Use dropout for neural networks to tackle overfitting.

## Fundamentals of Lasso(L1) And Ridge(L2) Regression:-

## Ridge Regression (L2 Regularization)

**What is Ridge Regression?**

A regularization technique that adds a penalty to the loss function proportional to the sum of the squared coefficients (L2).

**Formula:**

$$Loss = MSE + \lambda \sum_{i=1}^{n} (\text{slope})^2$$

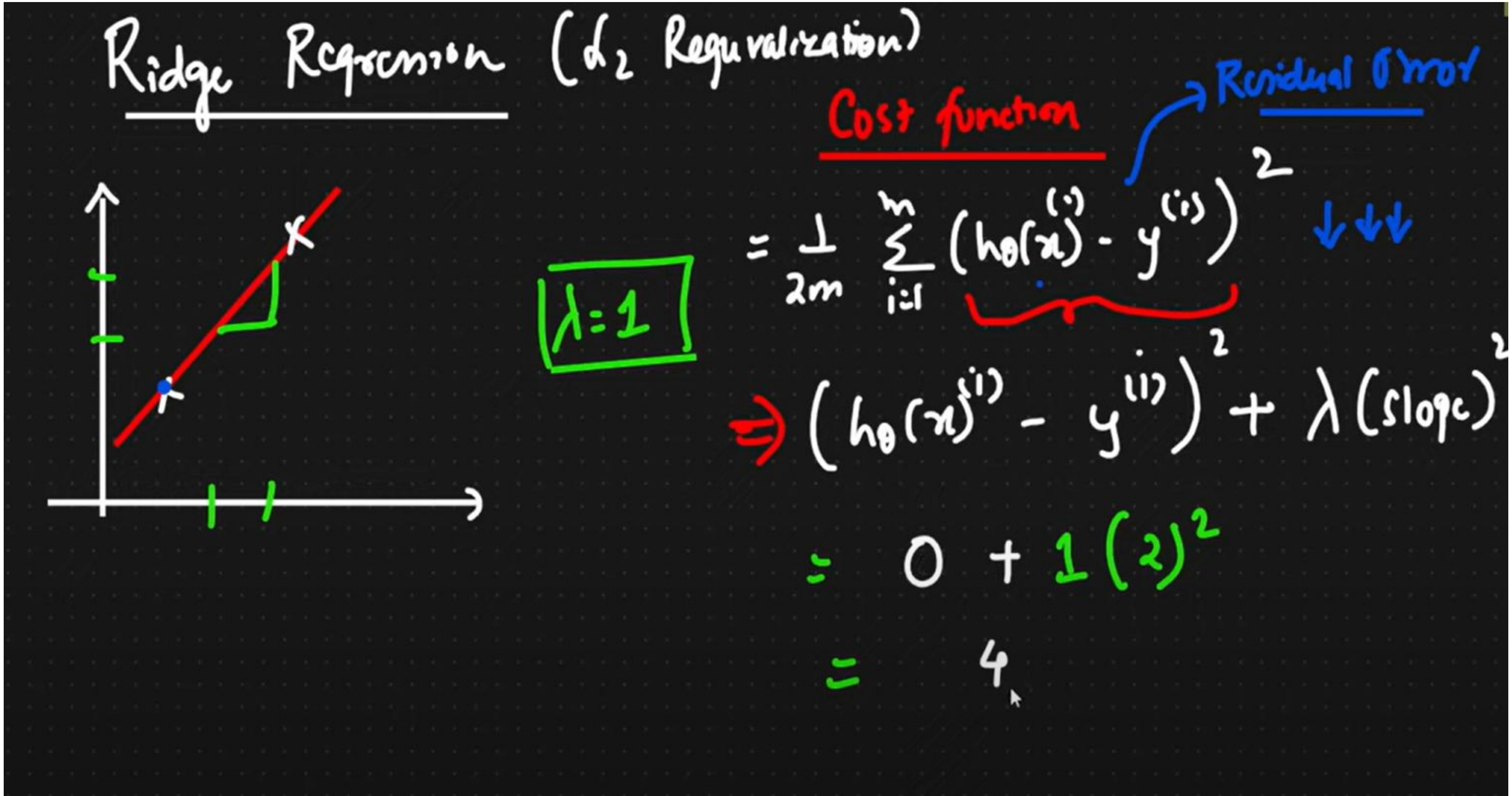
Slope:- 1 unit increase in (x) leads how many unit increase in (y)

**How Ridge Helps Overcome Overfitting:**

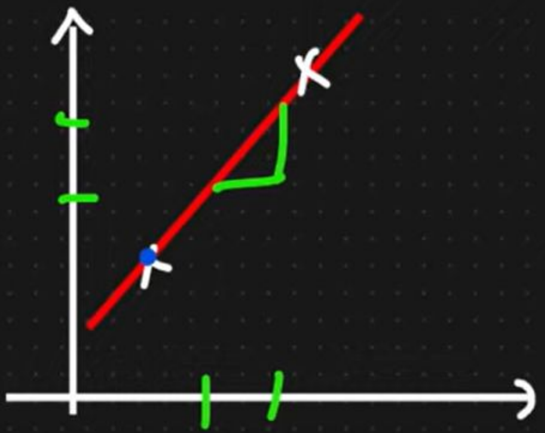
1. Shrinks large coefficients closer to zero, reducing model complexity.
2. Balances bias and variance for better generalization.

**Key Benefits:**

1. Works well when all features are relevant but correlated.
2. Reduces the influence of multicollinearity.



Ridge Regression ($L_2$ Regularization)



Cost function

$\boxed{\lambda = 1}$

Residual error

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

$\downarrow\downarrow\downarrow$

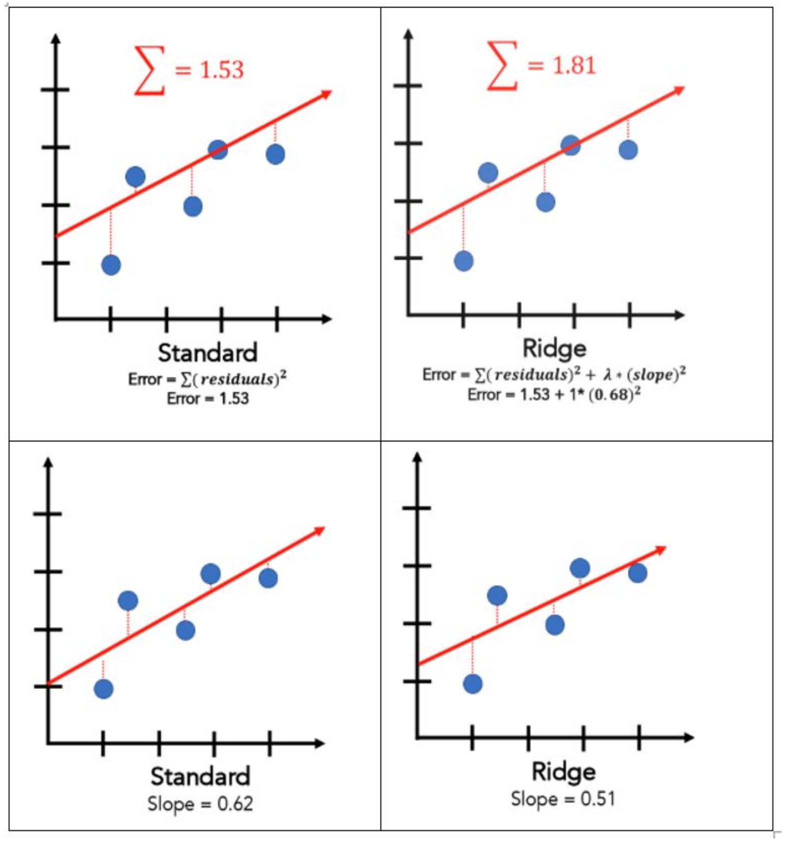
$$\Rightarrow \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\,(\text{slope})^2$$

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

$$= 4$$

## Ridge Regression (L2 Regularization)

### How Ridge Regression helps to overcome the overfitting?



## Lasso Regression (L1 Regularization)

### What is Lasso Regression?

A regularization technique that adds a penalty proportional to the absolute value of coefficients.

**Formula:**

$$Loss = MSE + \lambda \sum_{i=1}^{n} |\text{slope}|$$

Slope:- 1 unit increase in (x) leads how many unit increase in (y)

### How Lasso Helps Overcome Overfitting:

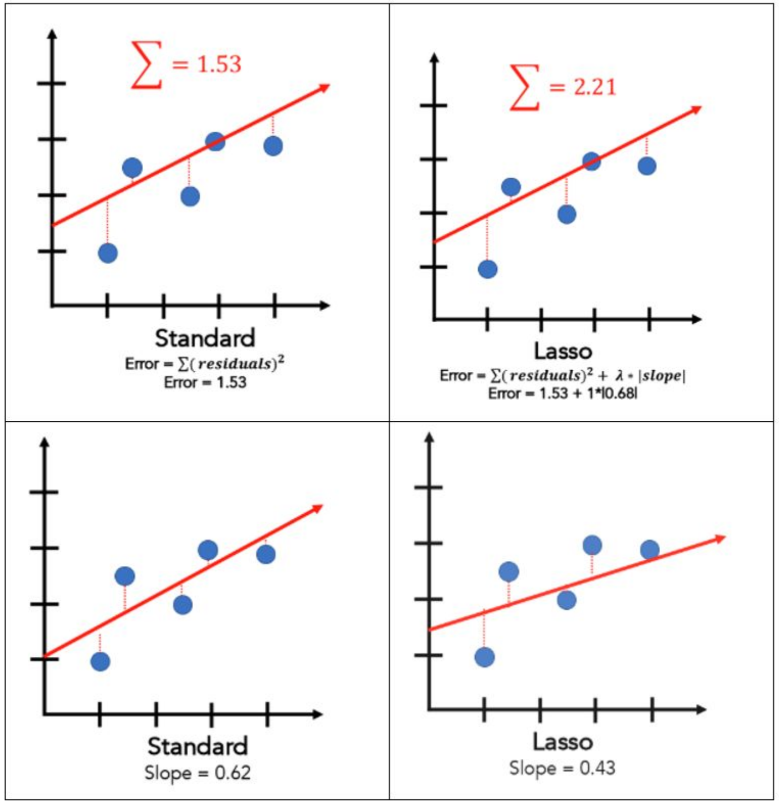
1. Forces some coefficients to become exactly zero.
2. Simplifies the model by selecting only the most important features.

### Key Benefits:

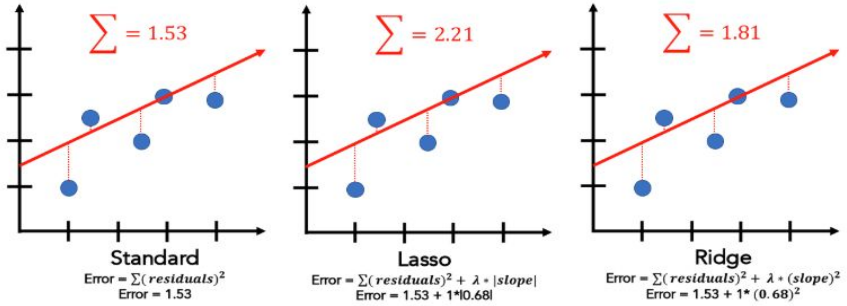
1. Performs feature selection by eliminating irrelevant or redundant variables.
2. Creates sparse models, improving interpretability.

## Lasso Regression (L1 Regularization)

### How Lasso Regression helps to overcome the overfitting?

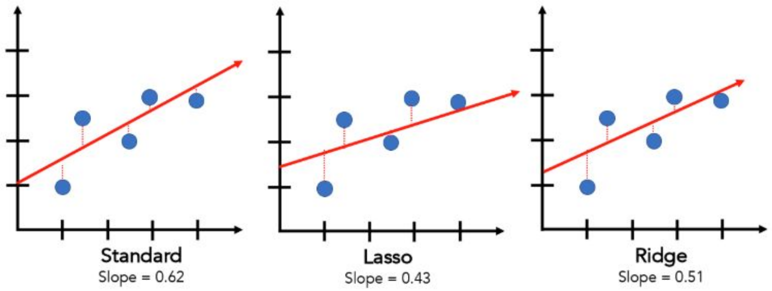


## Residual Error:-



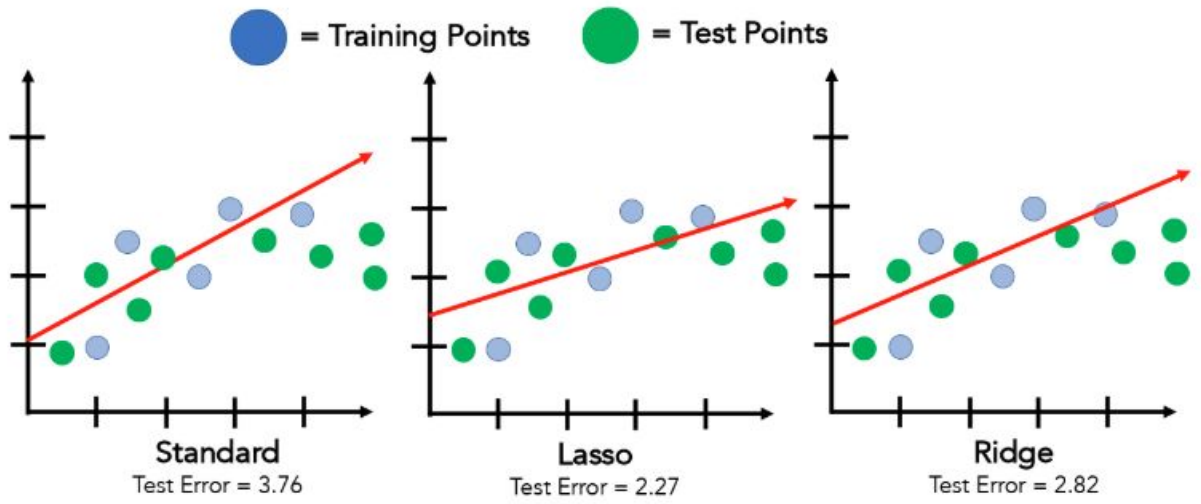
$* \boldsymbol{Note}: \boldsymbol{\lambda = 1}$

Best Fit Lines



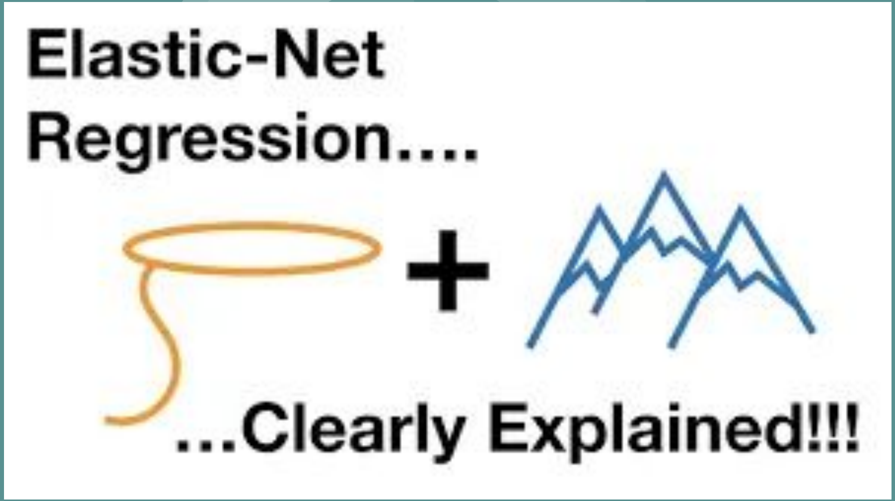
**Best Fit Lines with small slope, that will hopefully fit our test data better.**

## Test Error Difference:-



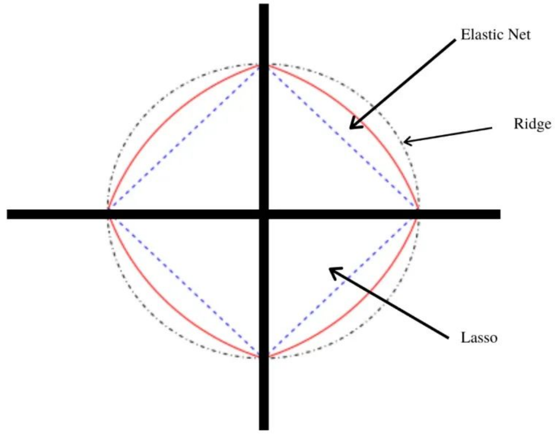
## Elastic Net

## Regularization



## Elastic Net Regularization

- Elastic Net is a **hybrid regularization technique** that combines the strengths of both **L1 (Lasso)** and **L2 (Ridge)** regularization. It adds penalties for both the absolute values (L1) and squared values (L2) of the model coefficients, providing a balance between **sparsity** and **stability**.
- Elastic Net Regression is a powerful linear regression technique that combines the penalties of both Lasso and Ridge regression. It is particularly useful in scenarios where traditional linear regression might struggle with multicollinearity, i.e., when predictor variables are highly correlated. In this blog post, we'll delve into the concept of Elastic Net Regression, its mathematical formulation, and its practical significance in machine learning.



## Ridge Regression (L2 Regularization)

### Loss Function for Elastic Net

The loss function for Elastic Net regularization is given by:

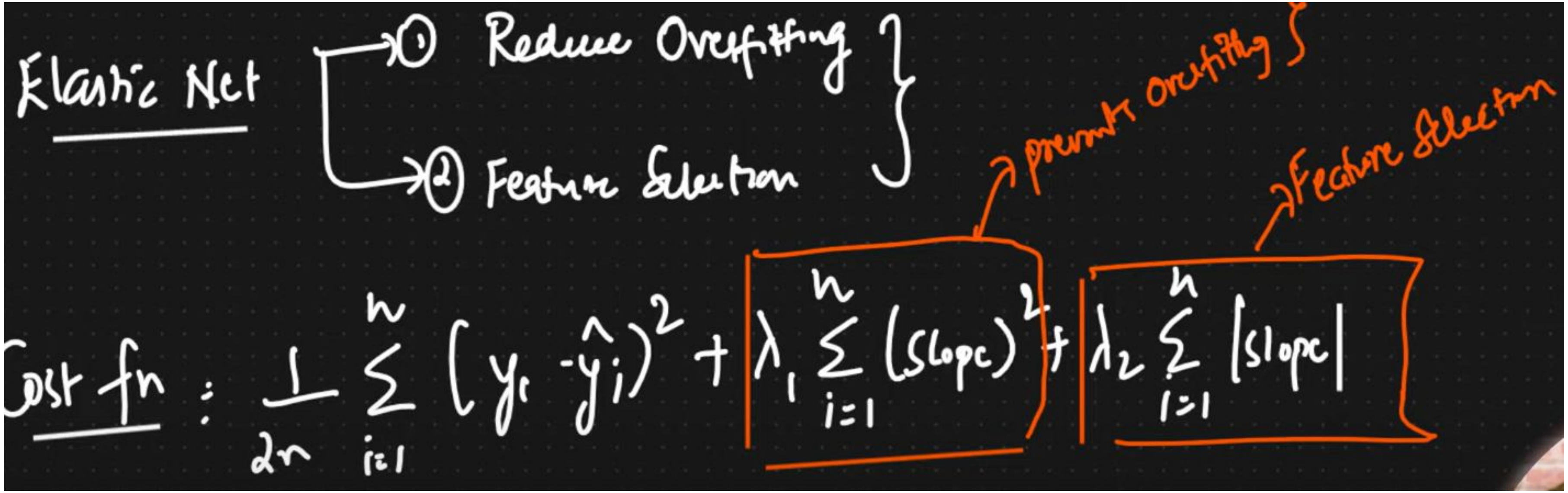
**Formula:**

$$\text{Loss} = \text{Original Loss} + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{j=1}^{p} \theta_j^2$$

Here:

- λ1: Regularization strength for the L1 penalty (Lasso).
- λ2: Regularization strength for the L2 penalty (Ridge).
- θj: Coefficients of the model.

## Ridge Regression (L2 Regularization)



**Elastic Net**

$$\begin{tikzcd} & \text{① Reduce Overfitting} \\ \text{Elastic Net} \arrow[ur] \arrow[dr] & \\ & \text{② Feature Selection} \end{tikzcd}$$

$$\text{Cost fn} : \frac{1}{2n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 + \overset{\text{prevents overfitting}}{\boxed{\lambda_1 \sum_{i=1}^{n}(\text{slope})^2}} + \overset{\text{Feature Selection}}{\boxed{\lambda_2 \sum_{i=1}^{n}|\text{slope}|}}$$

## Key Characteristics of Elastic Net

1. **Combines L1 and L2 Penalties**:
   - The L1 penalty encourages sparsity by shrinking some coefficients to zero, performing feature selection.
   - The L2 penalty shrinks all coefficients, stabilizing the model and handling multicollinearity.
2. **Overcomes Limitations of L1**:
   - L1 regularization struggles with highly correlated features, selecting only one feature from a group.
   - Elastic Net selects groups of correlated features, which is useful in many real-world datasets.

## Key Characteristics of Elastic Net

1. **Combines L1 and L2 Penalties**:
   - The L1 penalty encourages sparsity by shrinking some coefficients to zero, performing feature selection.
   - The L2 penalty shrinks all coefficients, stabilizing the model and handling multicollinearity.
2. **Overcomes Limitations of L1**:
   - L1 regularization struggles with highly correlated features, selecting only one feature from a group.
   - Elastic Net selects groups of correlated features, which is useful in many real-world datasets.

## Ridge Regression (L2 Regularization)

**3. Weighted Balance**:

- The hyperparameter α\alphaα controls the balance between L1 and L2 regularization $A \in [0,1]$,

where:

- $\alpha=1$: Pure L1 regularization.
- $\alpha=0$: Pure L2 regularization.
- Intermediate values create a mix of both penalties.

## When to Use Elastic Net:

- **High-Dimensional Data**: Suitable for datasets where the number of features (p) is much greater than the number of observations (n).
- **Correlated Features**: Ideal for handling groups of correlated variables, unlike Lasso which may exclude some relevant features.
- **Sparse and Stable Models**: Useful when you need a model that is both sparse (via L1) and robust to multicollinearity (via L2).

## Ridge Regression (L2 Regularization)

**Hyperparameter Tuning:**

Elastic Net has two key hyperparameters:

1. **α**: The mixing ratio between L1 and L2.
2. **λ**: The overall regularization strength.

These are typically chosen through cross-validation to optimize model performance.

## Ridge Regression (L2 Regularization)

**Hyperparameter Tuning:**

Elastic Net has two key hyperparameters:

1. **α**: The mixing ratio between L1 and L2.
2. **λ**: The overall regularization strength.

These are typically chosen through cross-validation to optimize model performance.

## Ridge Regression (L2 Regularization)

**Advantages of Elastic Net:**

1. Combines the strengths of L1 and L2 regularization.
2. Suitable for high-dimensional, highly correlated datasets.
3. Provides flexibility in tuning the balance between sparsity and stability.

Elastic Net is a versatile and powerful regularization method, particularly in complex datasets where neither L1 nor L2 alone can handle the challenges effectively.

## *1. Dropout Penalty*

**Definition**: Randomly drops a proportion of neurons during training to penalize reliance on specific features.

**Effect**:

- Prevents overfitting in deep learning by forcing the model to generalize.
- Reduces co-dependence of neurons on each other.

## ***2. Group Lasso***

- **Definition**: Extends L1 regularization by applying penalties to groups of features rather than individual features.

**Loss Function**: Where g represents groups of features.

$$\text{Loss} = \text{Original Loss} + \lambda \sum_{g=1}^{G} \sqrt{\sum_{j \in g} \theta_j^2}$$

**Effect**:

- Promotes sparsity at the group level, i.e., entire groups of features are either included or excluded.

**Application**:

- Useful when features are naturally grouped (e.g., polynomial features or categorical encodings).

## ***3. Max-Norm Regularization***

**Definition**: Constrains the maximum norm of weight vectors in neural networks.

**Effect**:

- Prevents weights from growing too large, ensuring stability and better generalization.
- Helps in avoiding exploding gradients, especially in deep networks.

## Types of Penalties in Regularization

**Application**:

- Common in deep learning, particularly in convolutional neural networks (CNNs).

## *4. Data Augmentation as Implicit Regularization*

While not a direct penalty, techniques like **data augmentation** act as regularizers by increasing the diversity of the training data, thereby reducing overfitting.

**Effect**:

- Forces the model to generalize better by training on augmented variations of data.

## Types of Penalties in Regularization

**Application**:

- Widely used in image processing and natural language processing (NLP).

## *5. Jacobian Norm Regularization*

**Definition**: Penalizes the norm of the Jacobian matrix of model predictions with respect to inputs.

**Effect**:

- Encourages smoother changes in predictions for small variations in inputs.

**Application**:

- Adversarial robustness, preventing models from being overly sensitive to input perturbations.

## *6. Label Smoothing*

**Definition**: Modifies the target labels by softening the one-hot encoding during training.

**Effect**:

- Prevents the model from becoming overly confident in its predictions.

## Types of Penalties in Regularization

**Application**:

- Widely used in classification tasks, especially in neural networks.

## Types of Penalties in Regularization

### *7. Total Variation Regularization*

- **Definition**: Penalizes abrupt changes between neighboring feature coefficients, encouraging smoothness.

**Loss Function**:

$$\text{Loss} = \text{Original Loss} + \lambda \sum_{j=1}^{p-1} |\theta_{j+1} - \theta_j|$$

## Types of Penalties in Regularization

**Effect**:

- Ensures smooth transitions in coefficients, useful in signal processing or image denoising.

**Application**:

- Used in image restoration or feature selection for sequential data.

## Conclusion

- Regularization is a crucial technique in machine learning that helps improve model generalization by preventing overfitting. By adding penalties to the model's loss function, regularization methods like **L1 (Lasso)**, **L2 (Ridge)**, **Elastic Net**, and others control the complexity of the model, ensuring that it performs well on unseen data.
- Ultimately, regularization is an essential tool in the machine learning practitioner's toolkit, helping to create models that generalize well, are less prone to overfitting, and offer greater reliability and performance in practical use cases.