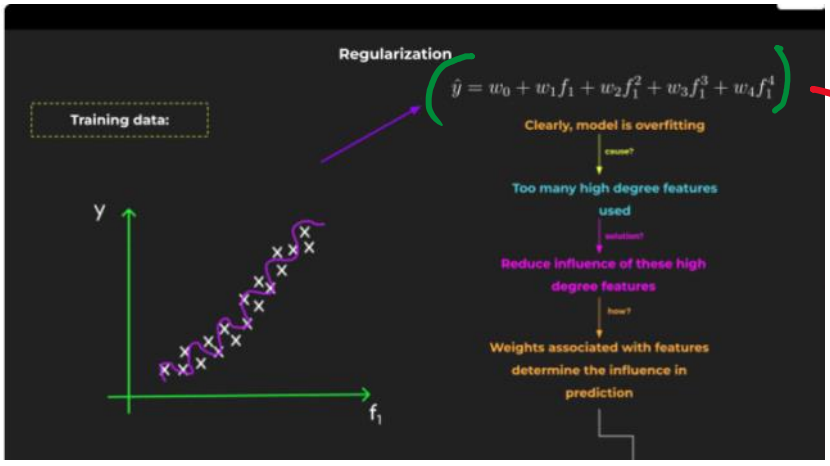
Regularization Intro

05 August 2025 21:32

Instead of training and testing Multiple Polynomial Models with different degrees, is there a better solution ?



→ Maybe f_1^2 is not required

$w_0 + w_1 F_1 + w_3 f_1^3 + w_4 f_1^4$ May be enough

Maybe a higher degree like f_1^4 May not be required

$w_0 + w_1 F_1 + w_2 f_1^2 + w_3 f_3^3$ May be enough

Is there a way we can reduce the influence or impact of degrees or features to avoid overfitting ?

For example :- $y = 2 + 3.1 x_1 + 4.2 x_1^2 + 2.2 x_1^3 + (0.01 x_1^4)$

Very low influence

observe and compare with other co-efficients

Any technique to help us achieve the above ?

{Regularization !} => as part of training process

Instead of Minimizing just the loss function using Min $\mathcal{L}$, add another term to the objective function => $\sum w$

$\mathcal{L} = SSE = \sum (\hat{y} - y_i)^2 + \sum w^2$ -> Co-efficients / impact

New objective :- minimize $\left\{ \mathcal{L} + \sum_{i=1}^{d} w_i^2 \right\}$

Why do we add $\sum_{i=1}^{d} w_i^2$?

Along with loss function, we also want to minimize impact of w (weights) as much as possible !

Outcome :- Influence of Unwanted weights significantly reduces !!

New Loss Function

05 August 2025 22:11

In Linear Regression, What do we want to Minimize ?

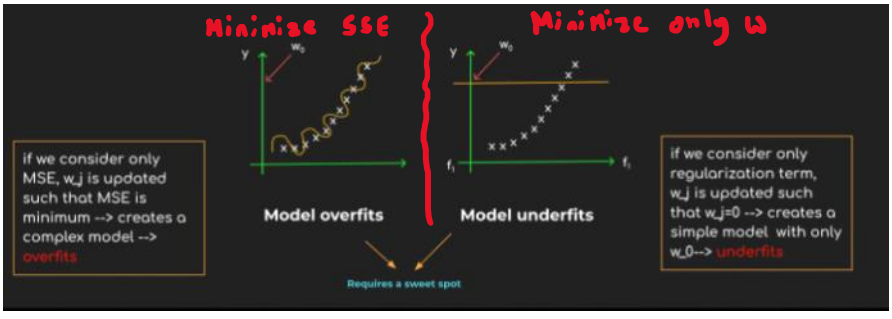
$$SSE = \sum_{i=1}^{i} (\hat{y}_i - y_i)^2$$

Downside of using this approach alone is overfitting

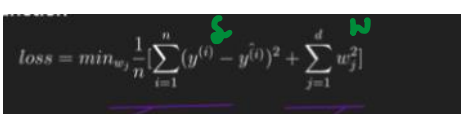
To reduce impact of a feature F_1 or F_1^2 or F_1^3, What do we need to minimize ?

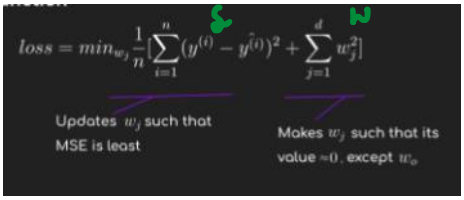
$\{w_i\}$

Downside of using this approach alone is Underfitting



Giving birth to,




$loss = min_{w_j} \frac{1}{n} [\sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + \sum_{j=1}^{d} w_j^2]$
Updates w_j such that MSE is least
Makes w_j such that its value ≈ 0, except w_o

The regularization parameter

05 August 2025 22:24

All okay but we want some control over how much impact regularization has right ?

Introducing λ (lambda), also called strength

$$loss = min_{w_j} \frac{1}{n}[\sum_{i=1}^{n}(y^{(i)} - \hat{y^{(i)}})^2 + \sum_{j=1}^{d} w_j^2]$$

becomes

$$loss = min_{w_j} \frac{1}{n}[\sum_{i=1}^{n}(y^{(i)} - \hat{y^{(i)}})^2 + \boxed{\lambda}\sum_{j=1}^{d} w_j^2]$$

$\lambda \geq 0$ — Acts as a tuner

if $\lambda = 0$, no regularization, overfitting

if $\lambda = 1000000$, weights will heavily reduce ⇒ Underfitting

But be careful !

Too high λ causes all weights to become 0,

which causes Underfitting

Too low λ causes all weights to stay as they are, causing overfitting

$\lambda \sum_{j=0}^{M} W_j^2$ → together, this is known as

{penalty.}

Interview → Regularization is the process of adding a penalty term to your loss Function to avoid overfitting

Types of Penalty

05 August 2025 22:39

$\lambda \sum_{j=0}^{M} W_j^2$ → Goal is to Minimize Weights of Features that cause overfitting.

W^2

This means, some of the weights become close to 0 but never 0!

$$y = 3.9x_1 + 2.1x_1^2 + 0.01x_1^3 + 1.1x_2 + 0.001x_2^2$$

This type of regularization is called L2 or Ridge regularization / Ridge Regression

$\lambda \sum_{j=0}^{M} |W_j|$ → Goal is to completely turn off Weights of Features that cause overfitting.

↓

observe the change to $|W_j|$ from W_j^2

This means, some of the weights will be exactly 0

ex:- $y = 3.9x_1 + 2.1x_1^2 + 0x_1^3 + 1.1x_2 + 0x_2^2$

This type of Regularization is called L1 or Lasso Regularization

L1 Regularization / Lasso

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization / Ridge

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2$$

Loss function — Regularization Term

So When to use Which ?

Interesting property of L1 and L2 :

⇒ L1 :

1. Since L1 makes wts of useless features = 0 Hence used in feature selection } => Feature Selection
2. As L1 creates sparse weight vector, it reduces the number of non-zero parameter leading to Efficient memory usage and faster computation in high dimensional datasets

⇒ L2 : } -> Minimize impact of high value features

1. The square penalty term w_j^2 encourages smoother parameter values → beneficial in case of noisy data
2. L2 tends to Not zero out weight value of less useful features ⇒ helpful when we want model to aim for a balanced use of features

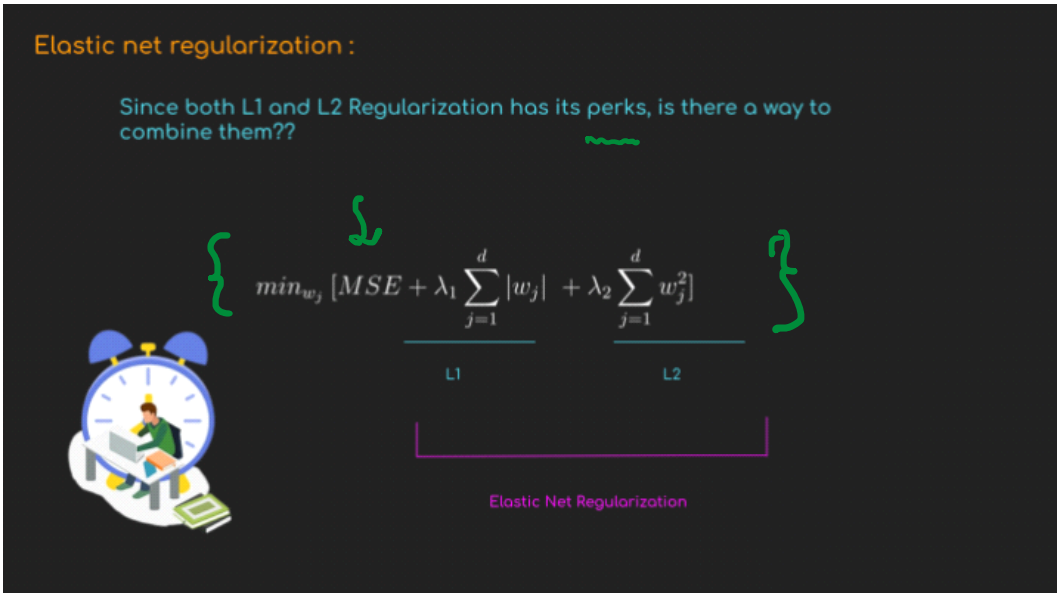
Elastic Net

05 August 2025 23:02

What if I'm not sure which to use?

What if I want the best of both worlds? Some weights completely turned off and some very low but not really 0.

Solution: Elastic net regression



At what temperature to bake a cake?

What brand of shirt to purchase for yourself?

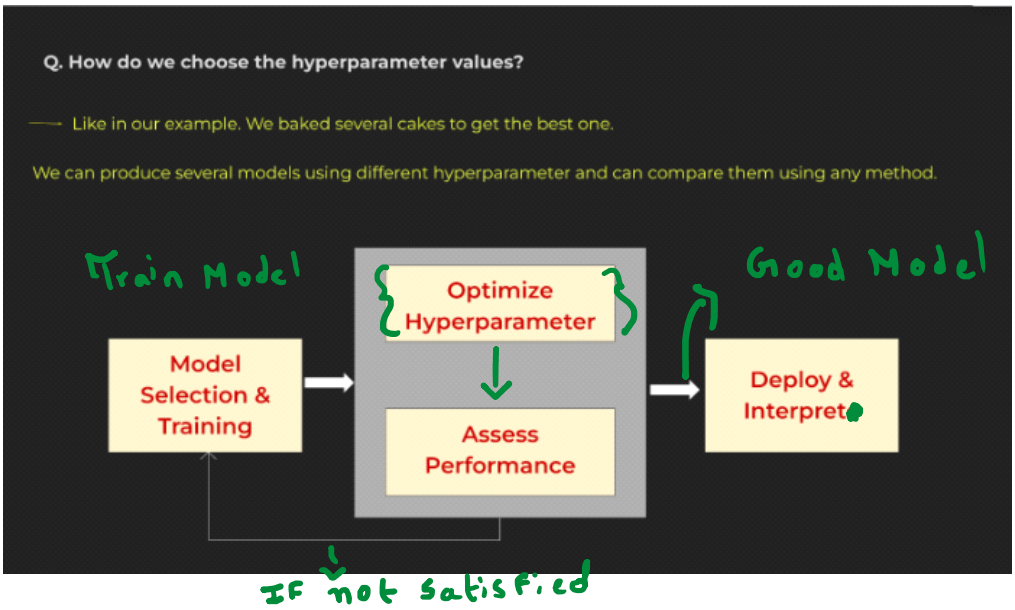
Which size of new pair of shoes to purchase?

These are all hyperparameters in ML.

You try different variations to find the best option. This is called Hyperparameter Tuning

In polynomial Regression, degree (highest power) you want to fit your model with.

In Regularization λ (strength of regularization)


Q. How do we choose the hyperparameter values?
Like in our example. We baked several cakes to get the best one.
We can produce several models using different hyperparameter and can compare them using any method.
Train Model
Model Selection & Training
Optimize Hyperparameter
Assess Performance
Good Model
Deploy & Interpret
IF not satisfied

Cross Validation

05 August 2025 23:19

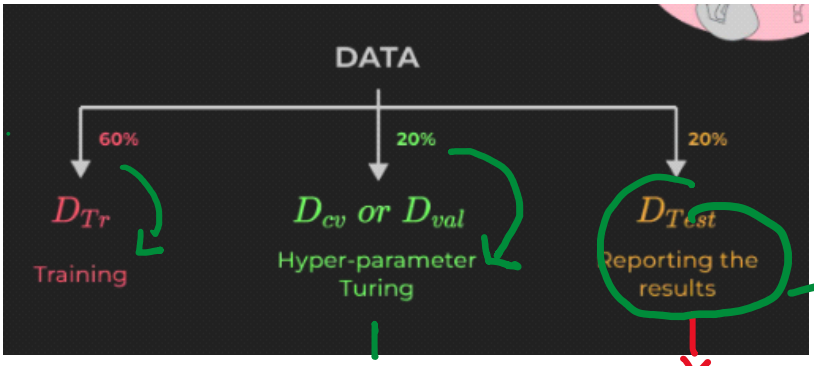
We may take 80 % of data to train and 20 % to test.

We perform hyperparameter tuning on Model by and validate results on test data

by looking at test Rsqrd

Alternative approach :-

Cross Validation



Final test RSq = 0.7

{Purely for this purpose} ← Back and Forth

{Purely for this purpose} => only once


Training Data
Validation Data
Test Data

Cross validation heavily reduces number of training samples. Only 60 % becomes training

Solution → {K-Fold Cross Validation}

Divide entire data into K chunks. Lets say 4 in this case.

Train on 75 %			Unseen validate on 25 %	Metric
Chunk 1	Chunk 2	Chunk 3	Chunk 4	Rsq 4
Chunk 1	Chunk 2	Chunk 4	Chunk 3	Rsq 3
Chunk 1	Chunk 3	Chunk 4	Chunk 2	Rsq 2
Chunk 2	Chunk 3	Chunk 4	Chunk 1	Rsq 1

as test Rsq Metrics

Final Metric → Avg Rsq = $\frac{(Rsq\ 4 + Rsq\ 3 + Rsq\ 2 + Rsq\ 1)}{4}$

↓

{Test Metric to report}

Purpose of Validation Set
to validate different hyperparameters } Satisfied

Report Model results on unseen data } Satisfied

↓

In a way, we made
sure all data was unseen
at some point or another!