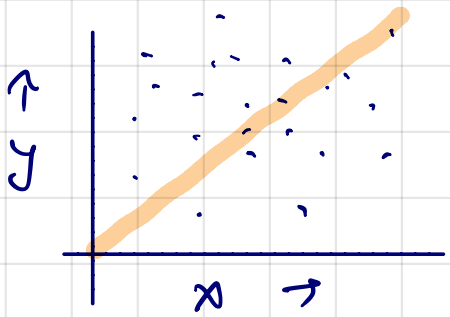
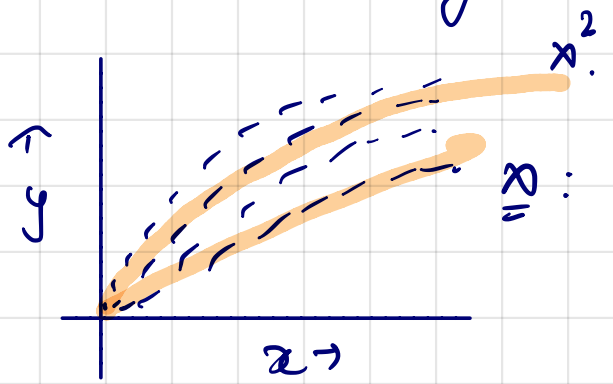
Regularization

Why Regularisation ?

→ Reduce Outliers
→ Shrinking values.
→ lambda with beta.
→ Scattered Points Train vs Test Error.
→ Train Error ; Testing Error.

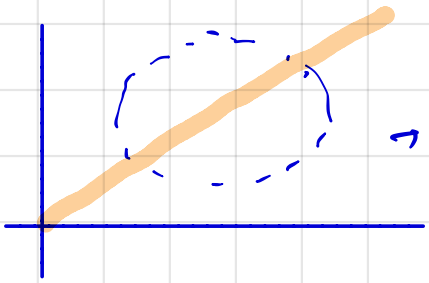
Test Error > Train Error : Overfitting.





x : underfits.

x^2 as power increases model overfits.



→ underfitting

Most instances, LR appears to be an underfit.

It means it is not able to learn correct pattern from data.

Coefficients : Some coefficients get a lot of weightage and hence are higher or lower than the rest.

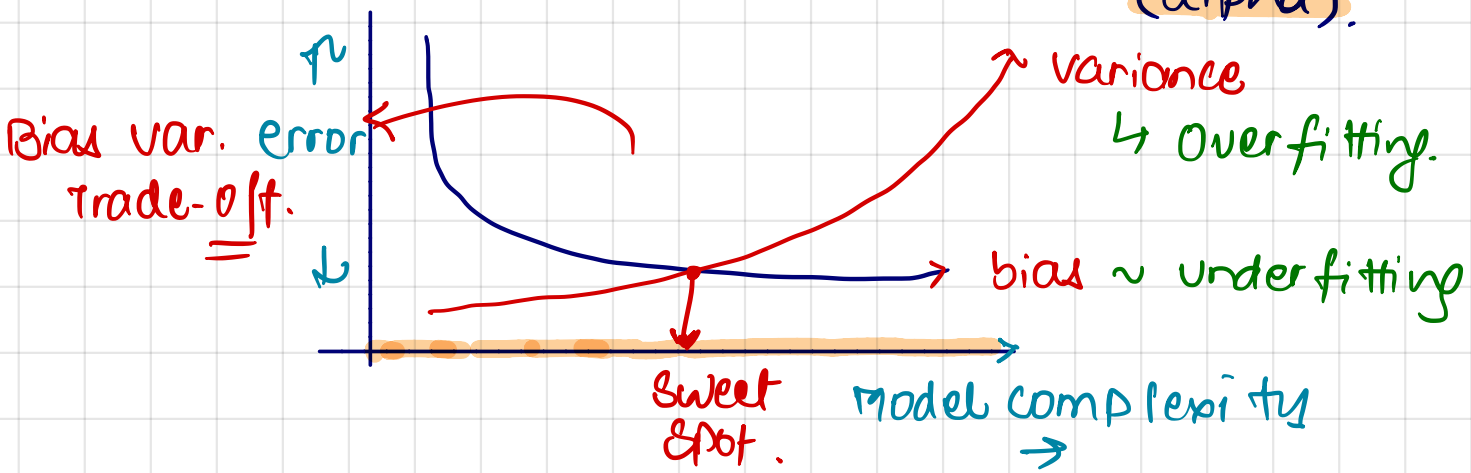
Regularization helps in making the model

→ consistent

→ avoid overfitting by introducing bias (λ).

↓

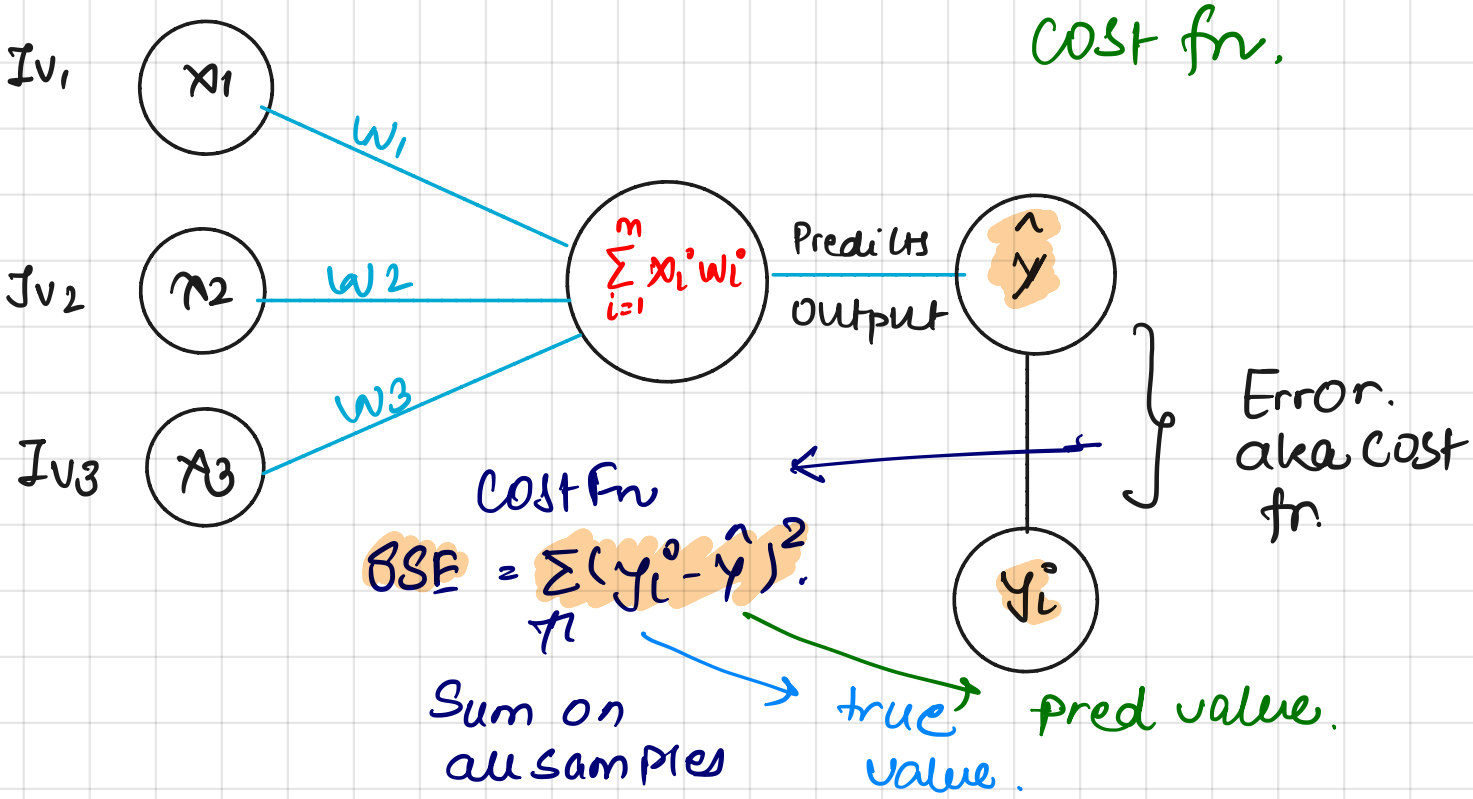
(alpha).



" Reg is a technique that reduces error by fitting a fn (Model) appropriately on the training set in order to minimize the overfit effect. "

Regularization → L1 : lasso Regression
→ L2 : Ridge Regression.
→ Drop out → NNets.
Regularization

Cost Function. : objective is to minimise the cost fn.



$$Cf = \min \left(\sum_{i=1}^{m} (y_i^o - \hat{y})^2 \right)$$

$$\left[\text{Cost Function} = \frac{1}{m} \sum_{i=1}^{m} (y_i^o - \hat{y})^2 \right]$$

optimization problem.

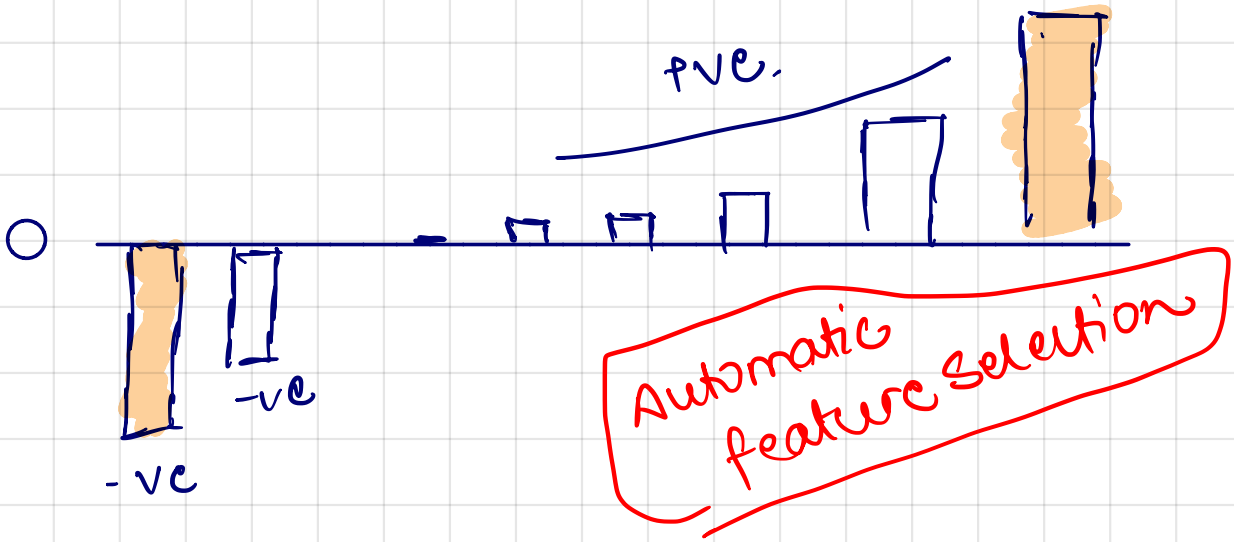
$$= \frac{1}{2m} \sum_{i=1}^{m} (h(\theta) - x_i^o)^2$$

A cost fn is measure of error b/w what model predicts & what actual value is.

Lasso Regression.

→ Lasso adds L1 penalty.

→ Shrinkage parameter that reduces the coefficients to 0 of those predictors which don't contribute to model.



→ Lasso will make some coeff. → 0. The model will eventually become simple, overfitting is reduced

→ At the same, underfitting is introduced

The model will loose the predictive power.

Reducing coefficient → 0 ⇒ less predictors ↓ Model becomes simple ← Loose prediction Power.

Cost Fn = $\min\left(\sum_{i=1}^{n}(y_i - \hat{y})^2\right)$

⇓

Penalty. → $2x$ $2(y_i - \hat{y})$.

Cost Fn(lasso) = $\min\left(\sum_{i=1}^{n}(y_i - \hat{y})^2\right) + \lambda\|\theta\|$

Here θ = Sum of absolute weights.

if $\lambda = 0$ ⇒ Cost fn → LR.

$\frac{d}{dx}(C) = 0$. $\theta = 3091$

$\frac{d}{dx}(3091) = 0$.

L2 : Ridge Regression. $\frac{d}{dx}(|\theta_1|^2) = 2\theta_1$

In case of lasso:

$\|W\| = |W_1| + |W_2| + \cdots |W_n|$.

Ridge: SQ. Magnitude of coefficients.

$\|\theta\| = (|\theta_1|^2 + |\theta_2|^2 + \cdots |\theta_n|^2)$

$CF = \sum_{i=1}^{n}(y_i - \hat{y})^2 + \lambda\|\theta\|^2$

when to use L1 & L2

→ L1 is good when considering a Cat. variable with many levels.

eg ItemType in bigmart.

→ L1 is preferred when we are interested in fitting a linear model with fewer var.

→ L1 converges to 0, helps in feature selection and reduces overfitting where as L2 does not encourage convergence towards zero but it gets close to zero and prevents Overfitting

→ Larger the features → L2 is better. Also, if predictors are highly collinear ⇒ Ridge.

→ In lasso, it will drop one collinear var but in Ridge, it will take both the variables and jointly Reduce the weight.

→ L2 regularization is called "weight Decay" as it forces the weights to decay towards 0.

Ridge	Lasso
→ does not reduce the coeff. to 0.	→ Reduces the coeff. to 0.
Ridge CF = $\Sigma(y_i - \hat{y})^2 + \lambda \cdot \Sigma \|\theta\|^2$	lasso CF = $\Sigma(y_i - \hat{y})^2 + \lambda \cdot \Sigma \|\theta\|$

Elastic Net

Enet CF = $\left[\Sigma(y_i - \tilde{y})^2 + \lambda \cdot \Sigma \|\theta\|^2 + \lambda \cdot \Sigma \|\theta\|\right]$

Gives better prediction by combining penalty of Ridge & lasso.

→ ($\alpha = 0$) & ($\alpha = 1$).