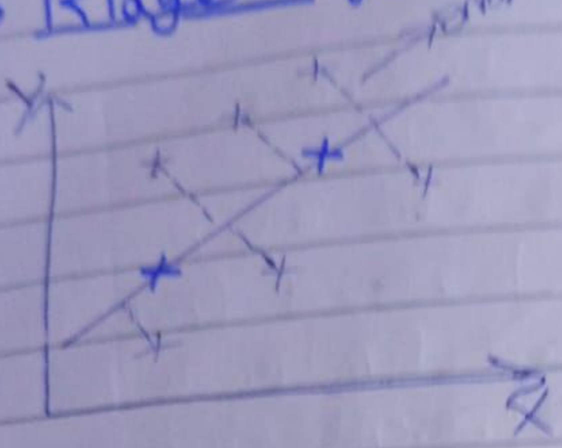
used to reduce overfitting

⇒ **Ridge Regression** (L2 Regularization)



Overfitting:

Train data ⇒ High Accuracy → Low Bias

Test data ⇒ Low Accuracy ⇒ High Variance

* We use Ridge Regression to reduce overfitting (we should never get accuracy of 100% on training data)
* Suppose by using Linear Regression we get a model which is overfitting to tackle it we will use **Ridge Regression**

It actually do **Hyperparameter tunning**

We know the cost function of Linear Regression

Cost function $= \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x)^{(i)} - y^{(i)})^2$

If the best fit line passes through all points of training data the cost function will become 0 to prevent that in Ridge Regression we add some parameters in it and it become

Cost function in Ridge Regression

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

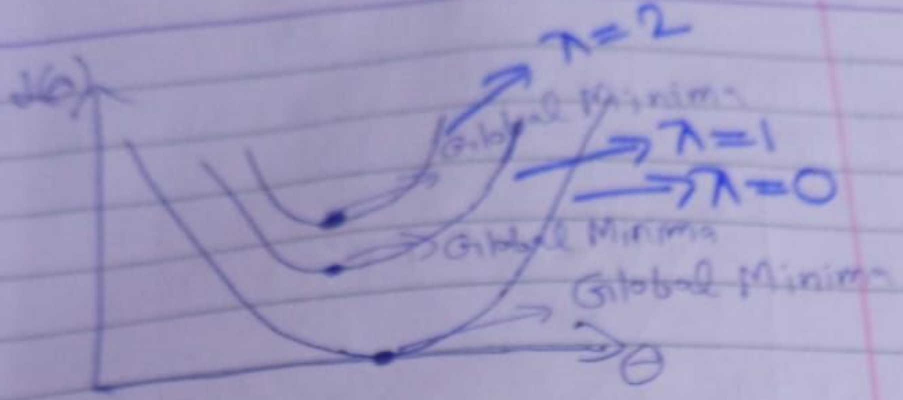
Hyperparameter

⇒ **Relationship b/w λ and $(\text{slope})^2$**

As λ increases the slope(θ_1) decreases. You can see graphically

but it will never become zero

②



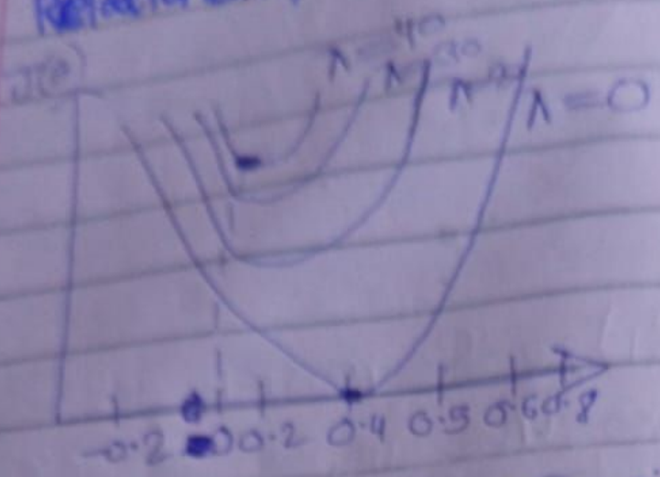
⇒ So Ridge Regression actually reduces the impact of loosely correlated feature on best fit line by reducing coefficients (slope) and prevent the cost function from becoming zero. In this way prevents overfitting.

② Lasso Regression (L1 Regularization) (used for feature selection)

Cost function:

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

Relationship b/w λ and |slope|



⇒ Means by increasing λ the slope decreases but difference is that in Ridge Regression it don't become 0 but in Lasso Regression it become 0 means that loosely correlated feature get removed.

③ **Elasticnet Regression:**

~~It~~

① Reduce overfitting
② Feature Selection

⇒ It is combination of both Ridge and Lasso

Regression
Cost Function:

$$\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right) + \underbrace{\lambda\sum_{i=1}^{n}(\text{slope})^2}_{\text{Performs reducing overfitting}} + \overbrace{\lambda\sum_{i=1}^{n}|\text{slope}|}^{\text{Perform Feature selection}}$$

* All these three Algorithms i.e Ridge, Lasso and Elasticnet are used for Hyperparameter tunning of Linear Regression.

⇒ Types of cross-validation:

Dataset

Training → train → used for training model

Training → validation → used for Hyperparameter Tunning

We do this stage using cross-validation

Test → To check the performance of my model

Changing the random states give different data

* We use C.V for Hyperparameter tuning
* We use C.V for checking the accuracy of model

① Leave One Out Cross Validation (LOOCV)

Training → 500 records

Experiment 01:

Validation → 1 | 499 → Train

Experiment 02:

Training → 1 | Validation → 2 | 498 → Training

⋮

Experiment 500:

499 → Training | 1 → Validation

* This technique is not efficient because the size of validation data is only 1 and it becomes inefficient as the size of dataset increases.

Disadvantage
It leads to overfitting as size of train data is very large as compared to validation data.

② Leave P Out Cross Validation (LPOCV)
Here instead of 1 you choose some p value and pick that much records from dataset out as validation in one experiment. Everything else is similar to Leave one out CV.

★ P value can be 10, 20, 30 ... or any selected as hyperparameter

③ K Fold CV:
Let $K=5$ and
n(size of dataset) $= 500$

Test size (validation size) $= \frac{n}{K} = \frac{500}{5}$

$\boxed{\text{Test size} = 100}$

Experiment 01:

100	400
Validation	Train

Experiment 02:

100	101-200	201-500
Train	Validation	Train

⋮

Experiment 05:

400	100
Train	Validation

* Then we will take Average of accuries of all experiments for the accuracy of our model.

Disadvantage:

As we are selecting a whole part as continuous block in classification problem there is possibility that we get only one type of data in whole validation dataset

④ Stratified K fold C.V:

Let $K=5$ and $n=500$

Test (validation) data size $= \frac{n}{K} = \frac{500}{5}$

$\boxed{\text{Test size} = 100}$

Now it choose 100 records from train dataset so that the number of outputs in validation dataset are almost equal like (60 0's 40 1's).

* Every other thing is similar to K Fold C.V

⑤ Time Series C.V:

• e.g

Product Sentiment Analysis → Reviews

⇒ Base on time

Let JAN → DEC

from JAN → April (Reviews are bad)

then become good

★ So, In time series cross validation happens based on days or time
Means we cannot pick randomly we pick days in sequence

Train

validation

Day1 Day2 Day3 ... Day n

⇒ Machine Learning Project Lifecycle:

1. Data Collection
   ↓
2. EDA
   ↓
3. Feature Engineering
   ↓
4. Feature Selection
   ↓
5. Model Training
   ↓ PKL of model
6. Webapp ← Flask
   ↓
7. Deployment ← AWS Cloud

AWS Deployment

Github Repo $\rightarrow$ Codebase

$\downarrow$ Push code for Deployment

Pipelines (Automatically deploy changes by a click button)

AWS

Elastic Beanstalk $\Rightarrow$ Linux Machine

$\Uparrow$

Some configurations required in our codebase