Exploratory Data Analysis

EDA (Data Wrangling)

1. Read Data
2. Explore Data — pandas profiling, ydata profiling
3. Duplicated values, Missing values, outliers, Scaling, Categorical Handling
4. Correlation, Visualization → Histogram, Boxplot
   - Barplot, Countplot
   - Hue
5. Feature Engineering : mpg → L/Kg, pickup Lat, Long; Distination lat, Long → Distance → pricing
6. Communication → Conclusion / Insights
   - output

* Search for "Steps of EDA"

* Car price
* laptop price
* Insurance Cost
* ... other

* Data Analysis with python

- [x] Import Data / Export
- [x] Data wrangling
- [x] EDA → (self study) → Chi-Square Test
- [x] Modelling → Linear Regression
- [x] Evaluation & Tuning

(Today / Previously)

Next session (practical)

- [ ] Final project → Assignment

Business Model

Modelling → Relation

$y = f(x)$

dependent var.

independent var.

Bicycle

① $y = 5x$

② $y = 3x^2$

③ $y = e^x$

..... Many Many Relations

$$\text{Speed} = \frac{\text{Distance}}{\text{Time}}$$

Known Relation

R

$V = IR$

Data Modelling

$$y = f(x)$$

Target — Features

Target
Label
output

Features
Input

Linear — Model

Car Data

	Features					Target
City	Year	Brand	Km	mpg	Fuel type	price
–	–	–	–	–	–	–

price = B1 (100000) / B2 (150000) Brand − (20 → 30 : 20000 / 30 − 50 : 50000) Km + (Yes : 20000 / Electric : 50000) fuel type

= 20,000

Linear Model

one variable

$y = 5x$ ⟶ price = 50 × horse power

price → Target

horse power → one feature

$y = 50000 + 5x$ ⟶ price = 100000 + 25 × hp

100000 → Const

Salary = 6000 + ✓✓ = ✓✓

6000 → Const

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n$$

Find Coeff.

Multiple Features

x	y
- -	- -
- -	- -
- -	-

Historical Data

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + b_3 x_2 + b_4 x_1 x_2 + b_5 x_2^2 - - - - -$$

polynomial

$y = f(x)$

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression

$$y = b_0 + b_1 x$$

$$y = b_0 + \sum_{i=1}^{n} b_i x_i$$

$b_0 x_0$ ← ①

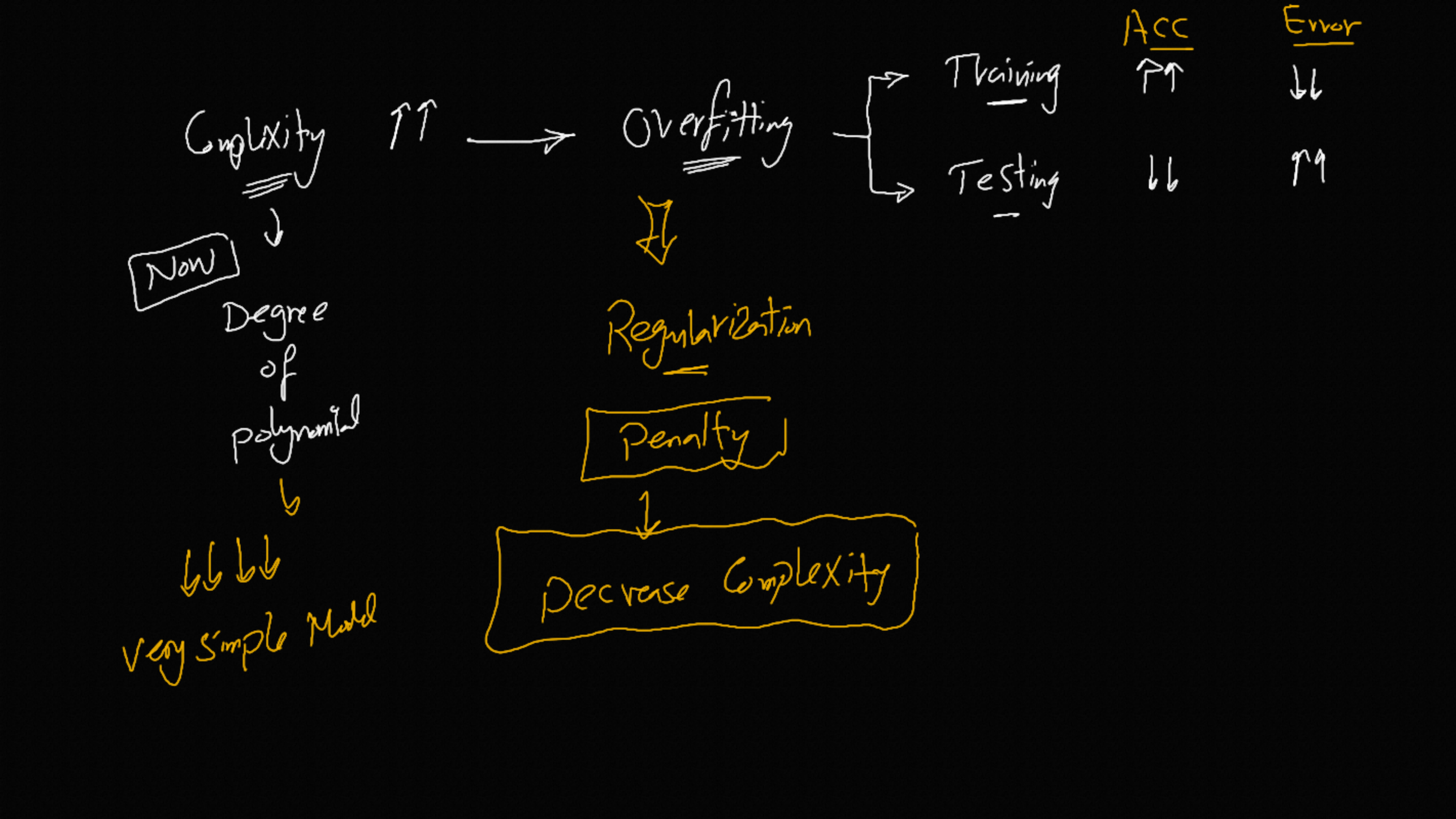
$$y = \sum_{i=0}^{n} b_i x_i$$

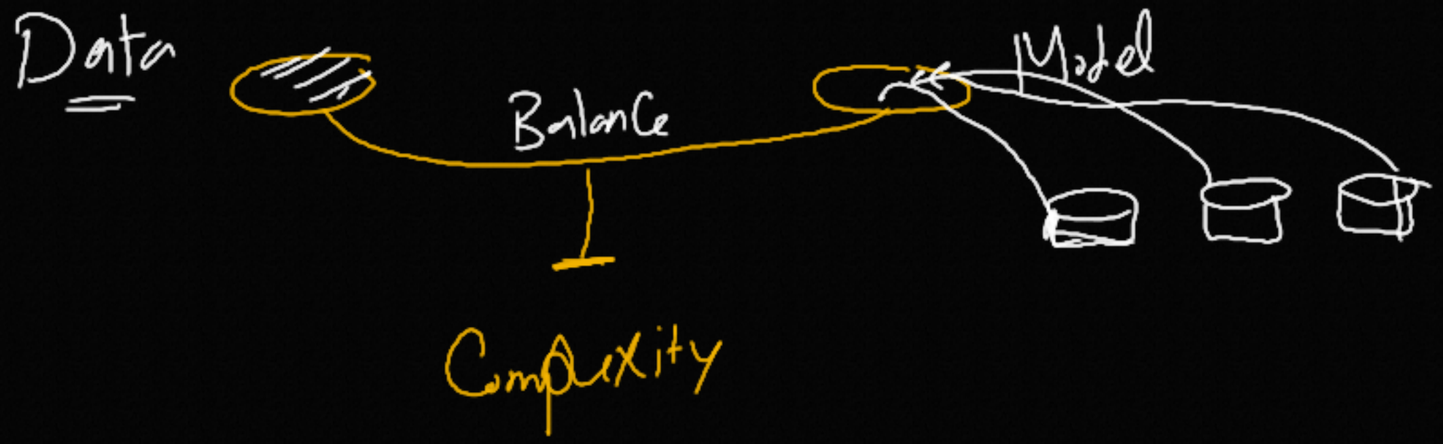
Degree ↑

Complexity ↑

$x_1 x_2^2$

$x_1^2 x_2$


Acc
Error
Complexity ↑↑
Overfitting
Training ↑↑ ↓↓
Testing ↓↓ ↑↑
Now
Degree of polynomial
↓↓ ↓↓
very simple Model
Regularization
Penalty
Decrease Complexity



Complexity → Model >> Data → overfitting

Model << Data → underfitting

Model ~~ Data → Good fit

⇒ High Correlation

⇒ Linear Model

↳ Best fit Model

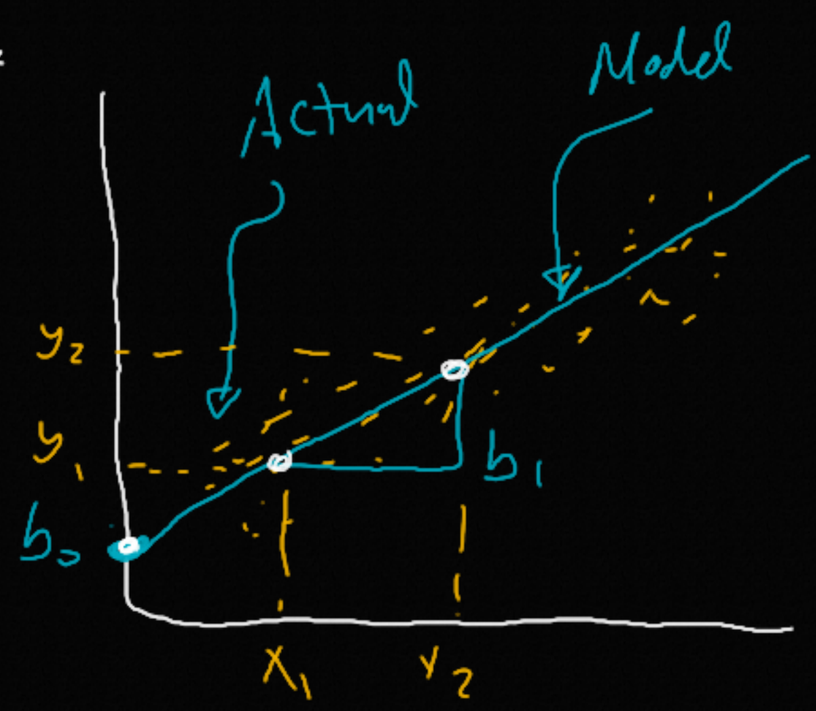
⇒ Actual & Model (prediction)

⇒ Measure Error

→ Min. Error

Every Line → b_0, b_1

↳ Best fit line



Historical Data

horse power

$price = b_0 + b_1 \, hp$

intercept (Bias)

Coeff (Slope)

$\frac{y_2 - y_1}{x_2 - x_1}$

Get Parameters b_0, b_1

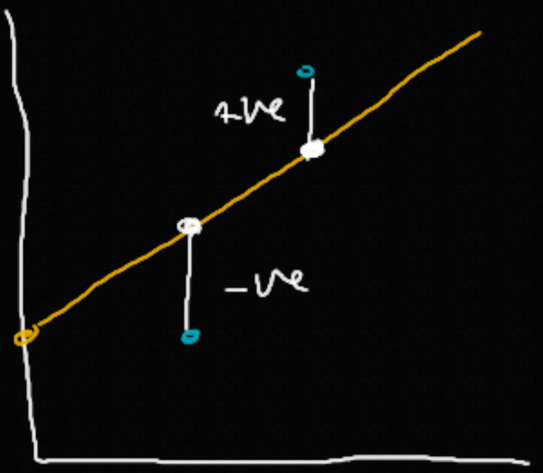
→ Min. Error — Cost fn

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_{i\,pred} - y_{i\,actual}\right)^2$$

↳ Mean Squared Error

↑ Minimize

Squared Error



	MSE
L_1	200
L_2	50
L_3	13 ✓

Get paramaters (weights)

Analytical

Closed loop

Normal Equation (vectorized)

$\Rightarrow y = f(w, x)$

$\rightarrow \underline{w} = f(y, x)$

(known)

Numerical

open loop

Gradient Descent

Iterations

$\rightarrow$ Try & Error (with steps)

$\rightarrow$ Set weights random

$\rightarrow$ Error

$\rightarrow$ change weights $\begin{cases} \rightarrow + \\ \hookrightarrow - \end{cases}$

$\rightarrow$ Error

$\rightarrow$ change weights $\rightarrow$ (W)

Gradient Descent

(#) set Random weights

$w_0 + w_1 x$

new weight →

① $J = \frac{1}{m} \sum_{i=1}^{m} (\hat{y} - y)^2$

$f(w)$

② $\frac{\partial J}{\partial w}$ → gradient

↳ Contribution of weight to the Error

③ $w^{new} = w^{old} - \alpha \frac{\partial J}{\partial w}$

defined by user ← learning rate

$\frac{\partial J}{\partial w} = 0$ (tolerance)

Normal Equation

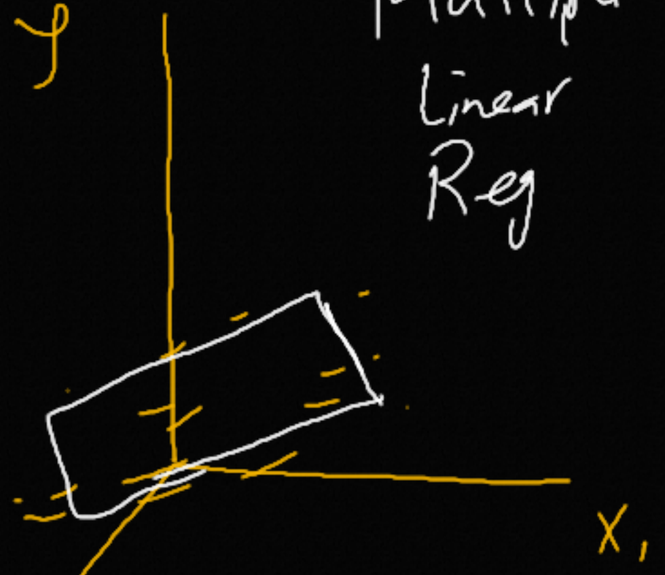
$y = w^T x$

$w = (X^T X)^{-1} X^T y$

Simple Linear Reg.

$y = b_0 + b_1 x$

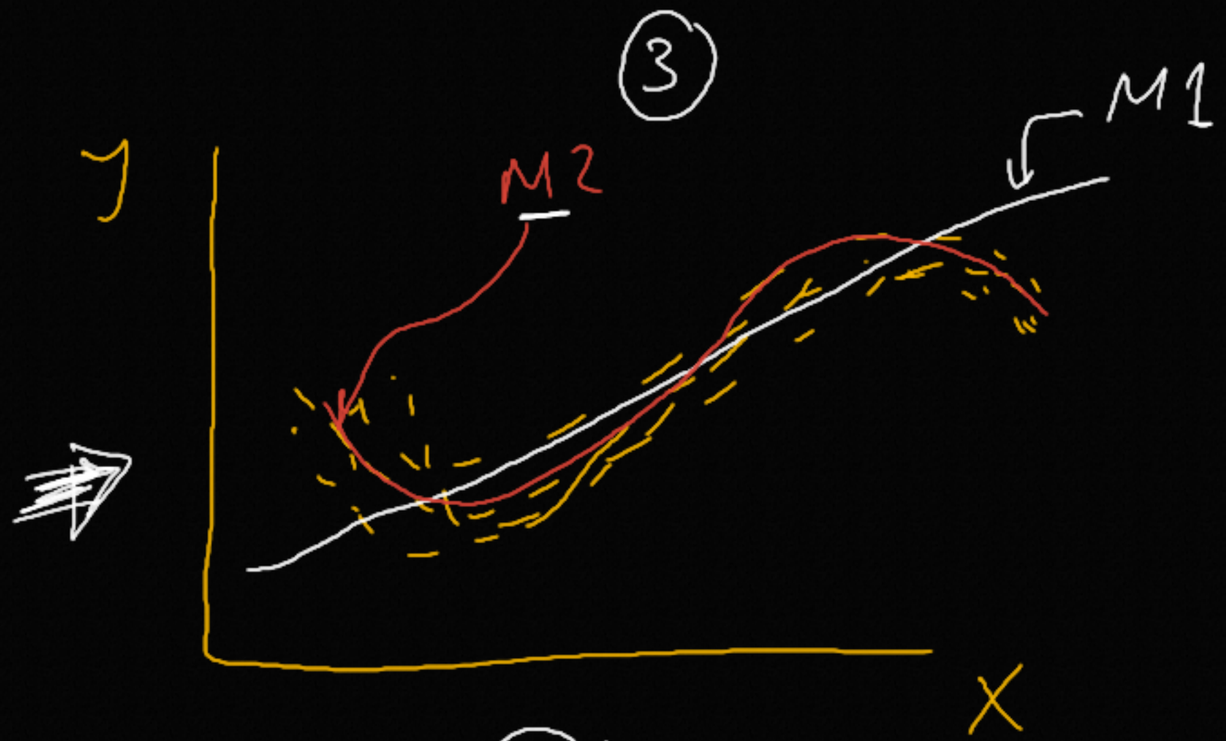
get weights (Best fit line), Min Error



Multiple Linear Reg

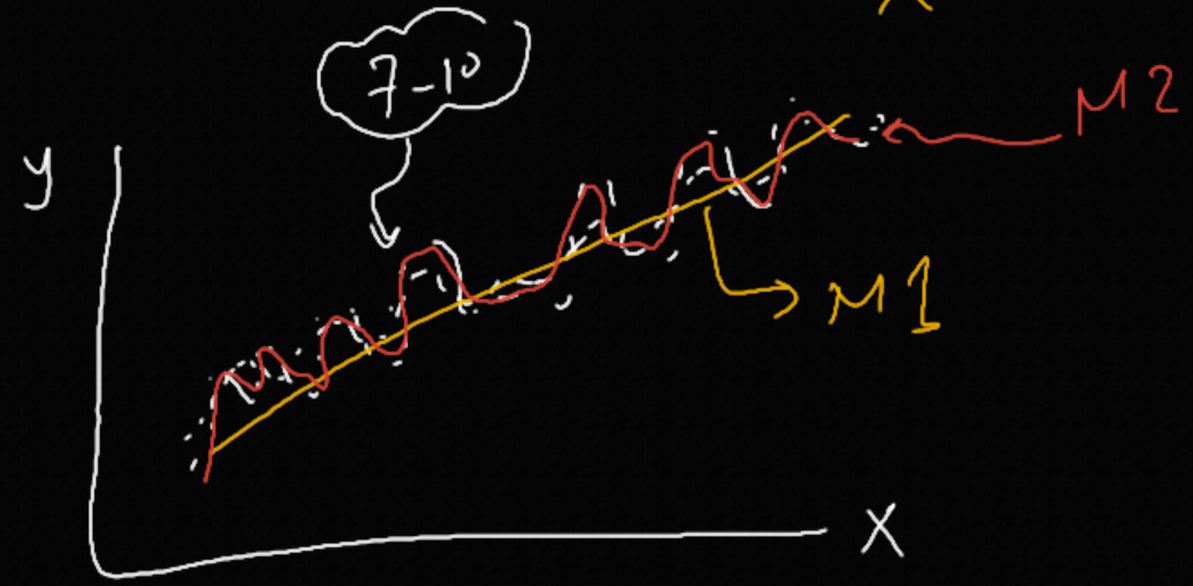
$y = b_0 + b_1 x_1 + b_2 x_2$

get weight



Error

M1 Error >> M2 Error



Error

M1 Error >> M2 error

overfitting

Non Linear Data ⟶ Non Linear Model

①

Polynomial Regression

Polynomial features ①

Linear Regression ②

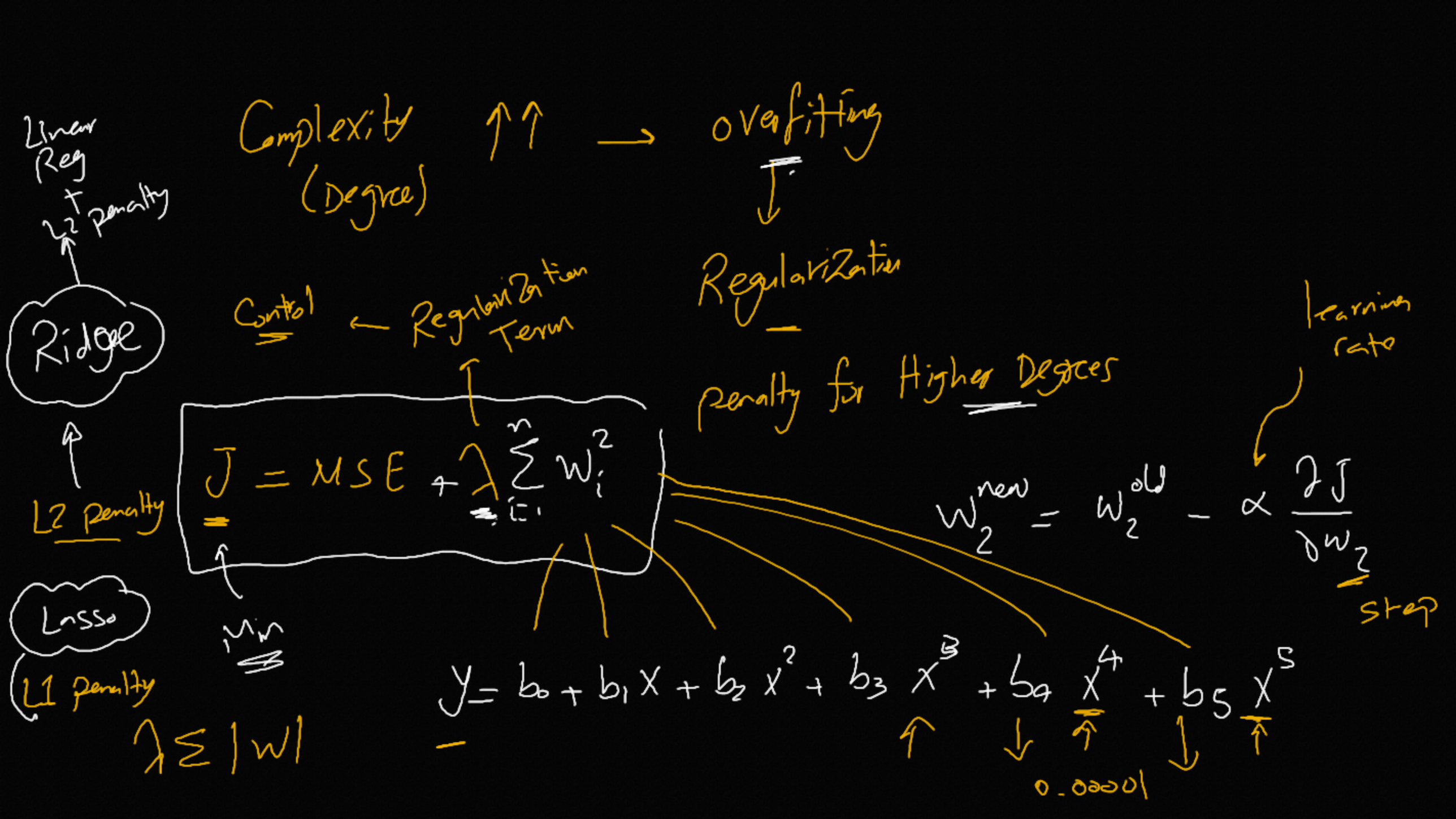
df['X₂'] = df['X'] ** 2

②

multiple Linear Regression

$$y = b_0 + b_1 x + b_2 x_2$$

x^2


Complexity (Degree) ↑↑ → overfitting
Regularization
Control ← Regularization Term
penalty for Higher Degrees
Linear Reg + L2 penalty
Ridge
L2 penalty
Lasso
L1 penalty
$\lambda \Sigma |w|$
$J = MSE + \lambda \sum_{i=1}^{n} w_i^2$
min
$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5$
$w_2^{new} = w_2^{old} - \alpha \frac{\partial J}{\partial w_2}$
learning rate
step
0.0001

L1, L2 penalty → Elastic Net

Linear Reg + L1, L2 penalty

- Linear Regression
- Evaluation & Tuning

100,000 Normal Eqn

(α) Gradient Desc

Evaluation on Regression Models

y → Continuous

10 15 12.5 → 10000 15000 12500

5 6 10

Yact Ypred

Error

Relative

Min Error

MAE

MSE

SE

Kg

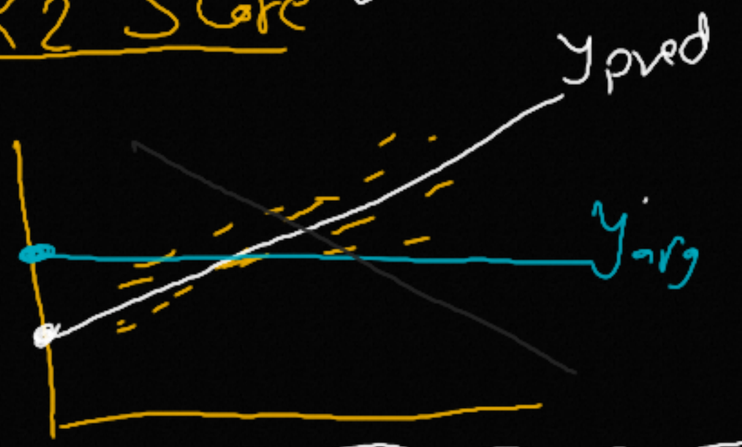
pound

Accuracy

Percent

0 → 1

R2 Score



$$R_1 = 1 - \frac{\sum (y_{pred} - y_{act})^2}{\sum (y_{avg} - y_{act})^2}$$

→ Model Error

M1 M2 M3

$MAE_{(1)}$ $MAE_{(2)}$ $MAE_{(3)}$

$MSE_{(1)}$ $MSE_{(2)}$ $MSE_{(3)}$

Tuning

learning Rate

$LR_{gd} = f(\alpha)$

Same Model

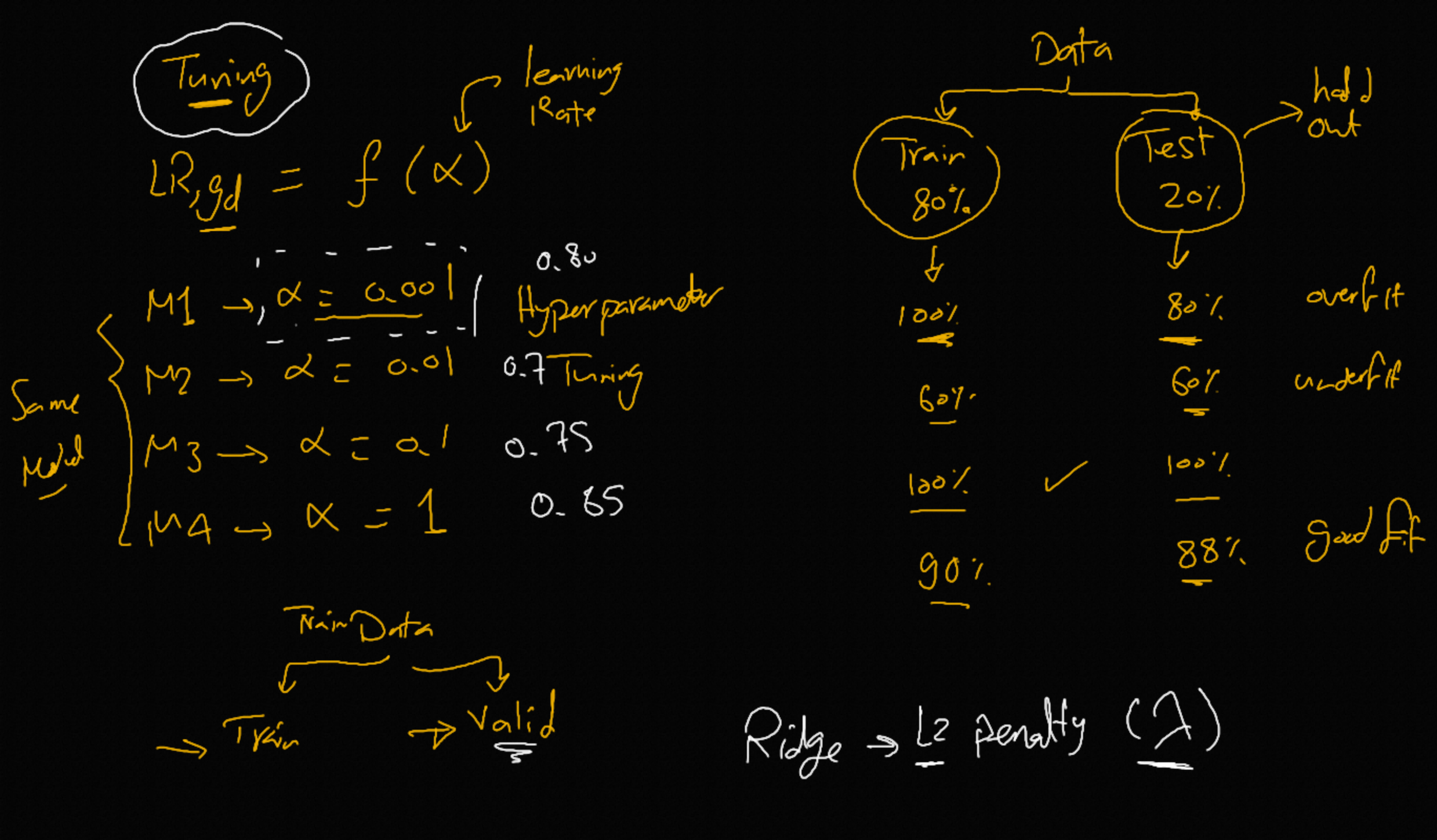
M1 → $\alpha = 0.001$ | 0.80

M2 → $\alpha = 0.01$ | 0.7

M3 → $\alpha = 0.1$ | 0.75

M4 → $\alpha = 1$ | 0.65

Hyperparameter Tuning



Train Data

→ Train → Valid

Ridge → L2 penalty (λ)