27-09-2025

Agenda: ML - 11

ML intro, AI, Parts of ML, Types of learning, Data split

completed

Data point 1
Data point 2
Data point 3
Data point 4
Data point 5
Data point 6
Data point 7

→

Data point 1
Data point 2
Data point 3
Data point 4
Data point 5
Data point 6
Data point 7

→ Train set

→ Test set

Train → ML algo → Evaluate

x_1 x_2 (y)
2, 3, (10) → ML algo → (y) (3)

difference → we require a method to tell ML algo about the error, so it can ajust itself.

Evaluation Metric

whether our model is learning or not?

whether our model is good or bad?

y (real values)	$\hat{y}$ (ML model prediction)	$(y - \hat{y})$
10	5	5
15	7	8
20	19	1
40	50	-10

Error calculation → each data point → loss function

→ set of data (Train, val) → Cost function → (4)

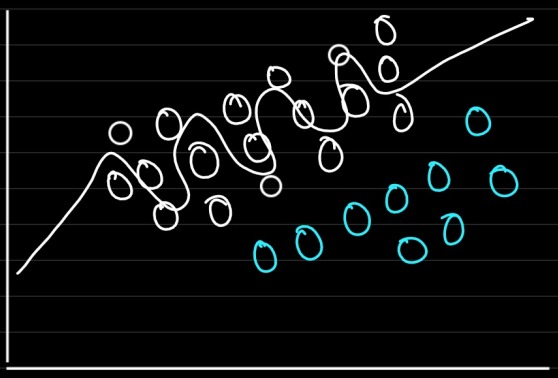
Train data → cost function → 180

Test data → 400

Loss function : Measure the error for a single example

Cost function : Sum of the loss functions over the dataset (Train/Val/Test)

If Training Loss <<<<< Test/validation Loss, means this is a overfitting case



→ Train set

→ Test set

Q1, Q2, Q3, Q4 , Q5, Q6, Q7

→ 100% → Train set

→ 30% → Test set

Train → Error → Cost function → number
(formola)

data[i] → Error → loss function → number
(formula)

If Training Loss === Test/validation Loss, means this is a "Ideal case"

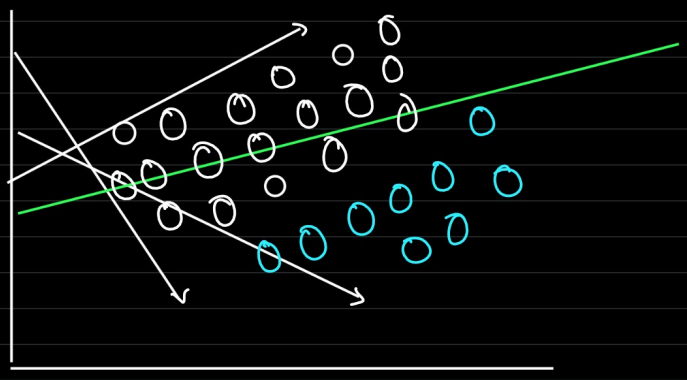
Train loss → 30
Test loss → 32

Q1, Q2, Q3, Q4, Q5

92% ⟷ 87%

↑ train ↑ test

If Training Loss and Test/validation Loss is bad, means this is a "underfitting case"



Train → 100 → 30%

Test → 5 → 50%

sample is small

Overfitting → y $\hat{y}$

10 →	10	train → 100%.
20 →	20	
30 →	30	
100	30	test
200	40	

Under fitting → y $\hat{y}$

10	1	train
20	2	
30	3	
100	10	test
200	20	

failed to learn any pattern

good fitting → y $\hat{y}$

10	9	train
20	16	
30	28	
100	92	test
200	196	

Evaluation Metrics: (Regression)

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

(3) MSE → $\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

Mean Square Error

y → height	$\hat{y}$ → height		
10	8	2^2 → 4	
20	14	6^2 → 36	
30	20	10 → 100	

English	Maths	Science
↓	↓	↓
90%	10%	70%
	↓	
	more attention	

$(-1)^2 \rightarrow 1$, $(1)^2 \rightarrow 1$ } making error positive

	y	$\hat{y}$	$y-\hat{y}$
(1)	1 →	2 →	-1
(2)	3	4 →	-1
(3)	5	6 →	-1

-1 -1 -1 = -3

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

(4) MAE $\rightarrow \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$

Mean Absolute Error

(5) RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$

(6) Modified MSE

$\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ — derivative → $\frac{2}{}\cdot(y_i-\hat{y}_i)\cdot\frac{\partial}{\partial\theta}(y_i-\hat{y}_i)$

$\rightarrow \frac{1}{2n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ $\quad \frac{1}{2n}\cdot 2\cdot(y_i-\hat{y}_i)\cdot\frac{\partial}{\partial\theta}(y_i-\hat{y}_i)$

Dog	0
cat	1
Elephant	2

	y	$\hat{y}$
(1)	0	2
(2)	1	1
(3)	2	0
(4)	2	1

Evaluation Metrics: (classification)

TP, TN, FP, FN (confusion matrix)

Accuracy: $(TP+TN)/(TP+TN+FP+FN)$

All true case / All cases

: overall correctness. $= 0.96 \rightarrow 96\%$

Precision: $(TP)/(TP+FP) \rightarrow 0.857 \rightarrow 85.7\%$

: How often we are correct

:

Email → ML → spam / not spam

Actual \ Predicted	spam	Non-spam
spam	600 TP	300 FN
Non-spam	100 FP	9000 TN

Recall : TP / (TP + FN)

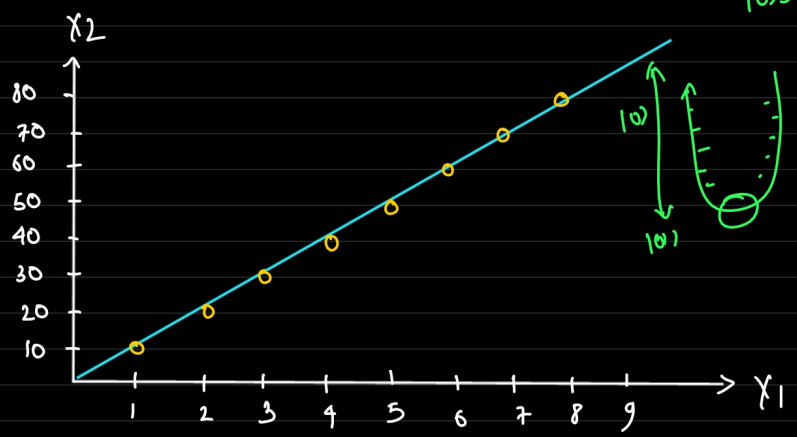
: 0.6667 → 66.67%.

: How many actual spam caught.

F1 score : Harmonic mean → balance precision & Recall

: $\frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$

Linear Regression



x_1	x_2
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
x_1	y

$y \longleftrightarrow x$

$y = 10x$

m

linear regression :

$y = mx + c$ → reduce complexity

$y = mx$

add or subtract parameter → model → linear model

→ $y = mx + c$ → element not multiplied
→ not raised to a power

→ $y = m_1x1 + m_2x2 + m_3x3 + c$

→ $y = m_0 + m_1x1 + m_2(x_1^2) \to x_2$

Non-linear :

$y = e^{m_0 + m_1x1}$, $y = \frac{m_0 + m_1x1}{1 + m_2x_2}$, $m_1(x1)\,m_2(x_2) + c$

→ linear regression

→ polynomial regression ← linear features

x_1	x_2	y
1	4	120
2	8	240

$120 \cdot x_1 \rightarrow y$

$60x_1 + 2x_2 \rightarrow y$ } linear

$m_1 = 60$ $\quad$ $m_2 = 2$

Data → EDA → FE → Final Data

Final Data → Train / Test

Train → Formula: $y = mx + c$ → pred → error → gradient

train data

$m = 0$ $\quad$ $c = 0$

x_1	x_2
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80

x_1 $\quad$ y

$y \rightarrow 10$, $x = 1$

$m, c \rightarrow 0$

$\hat{y} = mx + c \rightarrow 0$

error → $y - \hat{y} = 10$

x_1	x_2	y
1	2	
2	3	
4	5	
6	7	
8	9	

m_1, m_2

$x_1 m_1 + x_2 m_2 = \hat{y}$

$m_1 \rightarrow \infty$ $\quad$ $m_2 \rightarrow \infty$

error $= \hat{y} - y$

final loss → m_1, m_2

Dataset

Year_of_exp	Completed_project	Salary_k
1	0	39.764
2	1	48.400
3	1	56.978
4	2	68.240
5	3	77.867
6	4	85.022