↗ on google colab notebooks
↳ Numpy, Pandas, Data preprocessing, Matplotlib.

ML [MACHINE LEARNING]

Dated:

- Columns → features / attributes.
- h → hypothesis
- x_i → vector

  x_1

  x_2
- y_i → label of i^{th} row

  y_1

  y_2

  y_3

  ⋮

2D dataframe

1D series

Model :- mathematical formula with a number of parameters that needs to be learned from the data. Fitting model to the data is known as Model training.

- If two dimension $= x_{ij} = [x_{11}, x_{12}, x_{22}, x_{21}, ..]$

  (x_{ij}, y_i)
- training data = D.

ML Types :-

- supervised learning set of input datasets & its known outcomes.

  ↳ includes :- linear regression, logistic regression, neural networks, decision tree, SVM, random forest, naive Bayes & KNN.

Ex :- Feature Matrix

Patient Age	Tumor size	Clump Thickness	....	Malignant?
55	5	3	---	TRUE
70	4	7		TRUE
85	4	6		FALSE
35	2	1		FALSE
---	---	---	---	

↙ Target function / label

↳ Labeled dataset

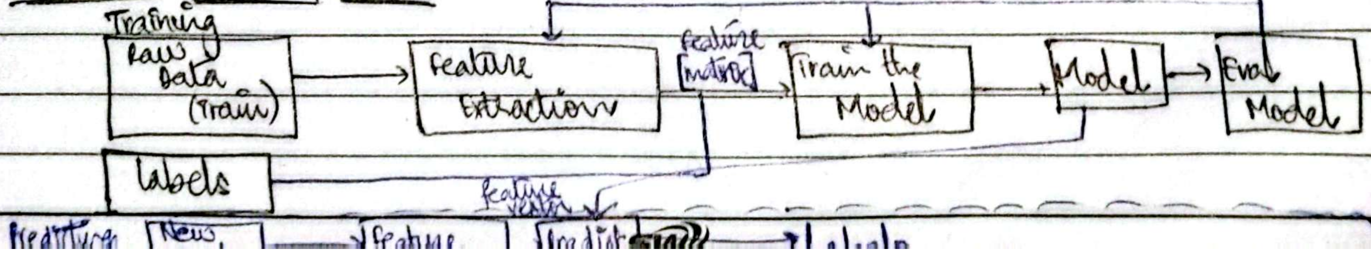
$f(55, 5, 3, ...)$

$f(\underbrace{72, 3, 3, ...}_{\text{Feature vector}})$

Patient age	Tumor size	Clump		Malignant
72	3	3	---	?
				?

Test Data

<u>SUPERVISED LEARNING WORK</u>



KNN, Decision trees, Random Forest, SVM, Neural Network.

Dated:

Classification :- discrete outcomes (Yes/No, True/False); outcomes can be tagged, categorized, separated into groups or classes → group A/B/C etc.

Ex - • Tumor is benign or malignant?
• To recognize letters & numbers in Handwriting.

Regression :- Quantitative Prediction on a continuous scale.
↳ Best fit line 

SUPERVISED LEARNING SETUP :-

• Inputs → features • Output → Label • training data - (input, output)
• loss, an objective or a cost function aimed to minimises error.
• Test Data - (input, output)
↑ Predict

Formulation :-

x_i → refers to the features of the i^{th} sample. that is :-

$x_i = [x_{i,1}, x_{i,2}, x_{i,3}, ----, x_{i,d}]$

y_i → label associated with i^{th} sample x_i, we formulate training data, in pairs as

$(x_i, y_i), i = 1, 2, ---, n.$

dataset

$D = \{(x_1, y_1), (x_2, y_2), ----, (x_n, y_n)\} \subseteq X^d \times Y$

X^d - d dimensional feature space & Y is the label space.

Regression :- $Y = R$ (prediction on continuous scale).

Classification :- $Y = \{0, 1\}$ or $Y = \{-1, 1\}$ or $Y = \{1, 2\}$ (Binary classification)
$Y = \{1, 2, ----, M\}$ (M-class classification).

$h(x) = y$ or $h(x) \approx y$

what we are predicting from model for test data 'x' should be = or approximately equal actual label

Dated:

Hypothesis Class :-

H = all candidate ML models

h = { individual candidate ML model }

h belongs to H

h^* belongs to H

h^* = { Best Model }

Ex:- H = { linear regression, KNN, logistic regression, SVM, Decision tree, }

h = { KNN, SVM }

h^* = { KNN }

$h^*(x) = y$

CamScanner

h, P :-

Ex:- $y = x + 2 - 3$ ← Model

$f(x) = y_p$

$h(x) = y_p$

Dated:

$x+2-3 = h(x)$

$y = y$

$(x=6, y=?)$ new data point

P= pair drawn from random probability distribution to get the random value.

$P(x=6, y=?) \dashrightarrow h(x) = x+2-3 = y, \quad y=?$

$6+2-3 = 8-3 = 5 = y$

$P(x=6, y=5)$ with some minor error with higher chances.

Hypothesis class (H)

H= candidate ML models.

h= { Individual candidate ML model }

h^* = { Best Model }

$h \subseteq H, \quad h^* \subseteq H$

Ex: H= { Linear Reg., KNN, Logistic, SVM, Decision tree. }.

h = { KNN, Logistic Reg., SVM }

h^* = { KNN }.

Loss Function :- $\mathcal{L}$

0/1 Loss Function :-

$$\delta_{h(x_i)-y_i}$$

$$\delta_{\hat{y}_i - y_i} \qquad \delta_K = \begin{cases} 1, & K=0 \\ 0 & \text{otherwise} \end{cases}$$

$$\boxed{\mathcal{L}_{0/1}(h) = \frac{1}{n} \sum_{i=1}^{n} 1 - \delta_{h(x_i)-y_i}}$$

- Normalize loss $1/n \to$ total no. of training samples so that output can be interpreted as average loss per sample.
- Not differentiable (jumps directly from 0 to 1), making it unsuitable for gradient-based optimization methods.
- Not continuous.
- Loss function counts the # of mistakes made by hypothesis function formula, can also be manipulated :-

$$\mathcal{L}_{0/1}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{0/1}(h(x_i), y_i) \qquad \text{if } \mathcal{L}_{0/1}(h(x_i), y_i) = \begin{cases} 0 & \text{if } h(x_i) = y_i \\ 1 & \text{if } h(x_i) \neq y_i \end{cases}$$

or

$$\delta_K = \delta_{h(x_i - y_i)}$$

Dated:

Mean - Squared Error (MSE) :-

$$L_{sq}(h) = \frac{1}{n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

for normalization

- But squaring amplifies / makes the error look big.
- Loss grows quadratically with absolute error amount in each sample.

Root Mean Squared Error (RMSE) :-

$$L_{rms}(h) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (h(x_i) - y_i)^2}$$

Absolute Loss Function :-

$$L_{abs}(h) = \frac{1}{n} \sum_{i=1}^{n} |h(x_i) - y_i|$$ ← All losses will be non-negative.

- Loss grows linearly with the absolute of the error in each prediction.
- Used in regression & suited for noisy data.

Loss is the "training objective". Lower loss → better fit to data.

$$h(x) = \begin{cases} y_i, & \exists (x_i, y_i) \in D, \ x_i = x, \\ 0, & \text{otherwise} \end{cases}$$

0% loss error on training data → Model is fit to every data point in D.

Q) How can we ensure that hypothesis h will give low loss on the input not in D? Train/Test Split.

Dated:

Mathematical Formulation of K-NN Algorithm :-

Algorithm :-

$D = \{ (x_1, y_1), (x_2, y_2) \; ---- \; (x_n, y_n) \}$

$D = \{ x_i, y_i \; \_\_\_\_ \; x, y_n \} \rightarrow$ generalization

1-D / 2-D, 3-D ____ N-D

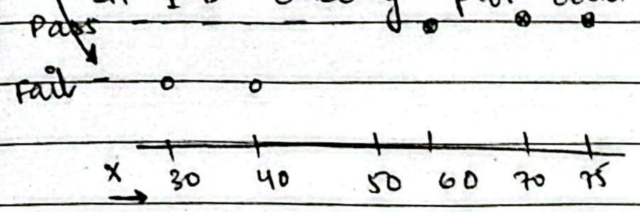
x	y
60	Pass
70	pass
40	Fail
30	Fail
75	pass

Height	weight	Group
65	25	group A
75	25	group B
60	85	group A
40	55	

$S_x = \{$ No. of $\underbrace{\text{nearest neighbors}}$ and its data point $\}$

To know nearest neighbors we use distance formula.

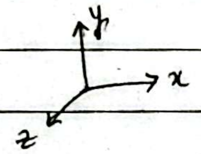
In 1-D directly plot data



Now plot $\underbrace{\text{Pass, Fail}}$
label Encoding / one-Hot Encoding

Label Encoding
Pass – 1
Fail – 0

o- In 2-D data, graph will be 3-D



one-Hot Encoding

	Pass	
	1	0
Fail	0	1

o- Metric to define nearest neighbor :-

distance Formula $\rightarrow$ Euclidean distance

$dist(x, y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (Atleast 2-D).

For 1-D $= (x_2 - x_1)$

Dated:

- K :- positive integer
- K should be odd.

For e.x :-

classification problem

x	y	dist
60	Pass	$\lvert 62-60 \rvert = 2$ ✓
70	Pass	$\lvert 62-70 \rvert = 8$ ✓
40	fail	$\lvert 62-40 \rvert = 22$
30	fail	$\lvert 62-30 \rvert$ 32
75	Pass	$\lvert 62-75 \rvert = 13$
65	Pass	$\lvert 62-65 \rvert = 3$ ✓

Predict for 62

if $K = 3$

$$S_x = \{60, 70, 65\}$$

$$\underbrace{\text{Pass} \quad \text{Pass} \quad \text{Pass}}$$

↓ majority voting

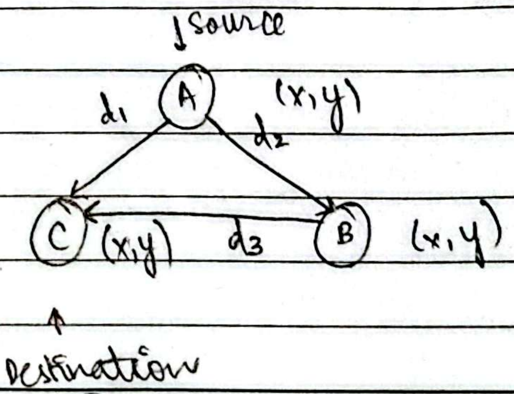
62 ⟶ Pass

- KNN is not parametric.
- K ki value is not used in formula, we are assuming K. though we can find optimal value of K.
- K is a hyperparameter.

distance → Euclidean distance
↳ Chebyshev distance
↳ Manhattan

Triangular Inequality :-
- Assuming Weighted Directional Graph.

$$d_1 \le (d_2 + d_3)$$



Dated:

KNN for Categorical Data :-

- convert into numbers.

Label Encoding   One-Hot Encoding

- WHAT IF DATA IS CATEGORICAL?

Hamming Distance :- koi bhi 2 words kitne qareeb hain.

- First, it compares string length, s1, s2.
- Now, find difference between each character.

$$dist(x, x') = \sum_{i=1}^{d} 1 - \delta_{x_i - x_i'}$$

$\delta_{x_i - x_i'}$: mapping function
if same toe write 1, else 0.

For ex:- cat   car

string i   string j

x_i x_j

x_i x_i'

$|\lambda| = 3$

$$dist(cat, car) = \sum_{i=1}^{3} 1 - \delta_{x_i - x_i'}$$

$= (1-1) + (1-1) + (1-0)$

$= 0 + 0 + 1$

$= 1$

↑ r & t are different.

Follow any.

$c = c = 1$	$c = c = 0$
$a = a = 1$	$a = a = 0$
$t \neq r = 0$	$t \neq r = 1$

agar ye use karahe toe we'll direct do sum

$$dist(x, x') = \sum_{i=1}^{d} \delta_{x_i - x_i'}$$

- WHAT DATA IS BINARY?

s_1 = 101011

s_2 = 11110

Hamming distance = 3

Dated:

Ex:- Fruit	Color	Shape	Label
Sample 1	Red	Round	Apple
Sample 2	Green	Round	Apple
Sample 3	Yellow	Long	Banana
Sample 4	Green	Long	Banana
Sample 5	Red	Long	Banana

X = (Green, Round) = ?

x_i x_i'

dist (Green, Round, Red, Round) = 1 ✓ s_1

dist (Green, Round, Green, Round) = 0 ✓ s_2

dist (Green, Round, Yellow, long) = 2

dist (Green, Round, Green, Long) = 1 ✓ s_4

dist (Green, Round, Red, long) = 2

if K=3 pickup three nearest.

s_1 s_2 s_4

Label :- Apple Apple Banana

So

Majority voting → Apple.

Ex:-

Red	Green	yellow	x_1	x_2	x_3	y → can do label encoding on this column.
1	0	0	2	3	Red	True → 1
0	1	0	4	5	Green	False → 0
0	0	1	7	8	yellow	True → 1
1	0	0	10	11	Red	False → 0
0	1	0	12	13	Green	True → 1

One-Hot Encoding

DLP → BERT encoding.

Dated:

Label Encoding :- imposes order

Red - 0
Green - 1
Yellow - 2

here it donot has any order.

so label encoding should only be used when there is order.

Red/Green/Yellow

Round →	1	0
Long	0	1

Ex After doing one-hot encoding of all features.

Fruit	Color	Shape	Vector-notation (Red Green Yellow Round long)	label
sample 1	Red	Round	(1, 0, 0, 1, 0)	Apple
sample 2	Green	Round	(0, 1, 0, 1, 0)	Apple
Sample 3	Yellow	Long	(0, 0, 1, 0, 1)	Banana
Sample 4	Green	long	(0, 1, 0, 0, 1)	Banana
Sample 5	Red	long	(1, 0, 0, 0, 1)	Banana

New : (Green, Round)

Encoded vector is (0,1,0,1,0)

Vector notation	Hamming Distance
(1,0,0,1,0)	2 ✓
(0,1,0,1,0)	0 ✓
(0,0,1,0,1)	4
(0,1,0,0,1)	2 ✓
(1,0,0,0,1)	4

if k = 3

(Red, Round) (Green, Round), (Green, Long)

By majority voting. (Green, Round)

Dated:

- In case of sentences, we take keywords only and remove irrelevant words such as :- The, is, a, an, punctuation marks.

Ex:- My name is Kinza.

↳ Keywords :- name, Kinza.

- Document Similarity :-

Point	tea	me	two
doc 1	2	0	2
doc 2	2	1	0
doc 3	0	2	0
doc 4	5	0	7

- Euclidean distance → is not suitable.

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

squaring increases the distance.

So, even if documents are similar, euclidean distance increases distance showing that documents are not similar.

Another ex:-

This is called TFV [Term Frequency Vector].

	cat	dog	
Doc 1	2	1	"cat cat dog"
Doc 2	1	1	"cat dog"
Doc 3	0	4	"dog dog dog dog"
Doc 4	10	10 = ?	

Euclidean Distance :-

$$d_E(A,B) = \sqrt{\sum(A_i-B_i)^2}$$

Distance (Doc 1, Doc 2) :-

$$\sqrt{(2-1)^2+(1-1)^2} = \sqrt{1} = 1$$

Distance (Doc 1, Doc 3) :-

$$\sqrt{(2-0)^2+(1-4)^2} = \sqrt{4+9} \approx 3.6$$

Now, this works fine but if we take a point (10, 10).

Ex:- Doc 2 = (1, 1), Doc 4 = (10, 10)

Distance (Doc 2, Doc 4) $= \sqrt{(10-1)^2+(10-1)^2} = 12.727$

distance is greater/large but they / pattern is similar

COSINE SIMILARITY:- $\cos\theta = 0 \rightarrow$ Not similar

$\cos\theta = 1 \rightarrow$ Totally similar.

Dated:

So, that's why euclidean distance is not suitable to use for finding Document similarity.

Cosine Distance :- focuses on angle (direction), not length.

Cosine Similarity ? :- Every word, sentence, or document can be represented as a vector (e.g.:- using TF-IDF, Word2Vec, ...)

- Cosine similarity measures how similar two vectors are by computing the cosine of the angle between them.

$$\text{Cosine similarity}(A,B) = \frac{A \cdot B}{\|A\|\,\|B\|}$$

Values range between :-

- 1 → vectors point in the same direction (high similarity)
- 0 → vectors are orthogonal (no similarity)
- -1 → vectors are opposite (in some contexts, dissimilar).

Cosine Distance :- Cosine distance(A, B) = 1 - Cosine Similarity(A, B)

- Range :- 0 (identical) → 2 (opposite).
- Smaller distance = more similar.

Ex:- If d_1 and d_2 are two document vectors, then $\cos(d_1,d_2) = \frac{d_1 \cdot d_2}{\|d_1\|\|d_2\|}$

Example

d_1 = 3 2 0 5 0 0 0 2 0 0

d_2 = 1 0 1 0 0 0 0 1 0 2

$d_1 \cdot d_2 = 3*1 + 2*0 + 0*1 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

$\|d_1\| = \sqrt{3*3 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$

$\|d_2\| = \sqrt{1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2} = 2.44$

$\cos(d_1, d_2) = 0.356$,

distance $= 1 - \cos(d_1, d_2)$.

DISTANCE METRICS :-

(1) EUCLIDEAN DISTANCE :- $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \_\_\_ + |x_{ip} - x_{jp}|^2}$$

Dated:

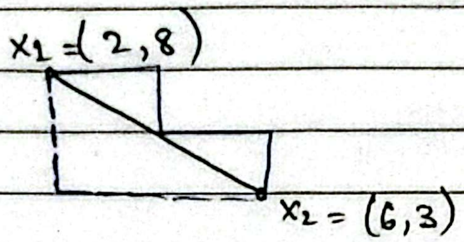
MANHATTAN DISTANCE :-

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

MINKOWSKI DISTANCE :-

$$d(i,j) = \sqrt[q]{\underbrace{|x_{i1} - x_{j1}|^q}_{\text{1st dimension}} + \underbrace{|x_{i2} - x_{j2}|^q}_{\text{2nd dimension}} + \dots + \underbrace{|x_{ip} - x_{jp}|^q}_{\text{pth dimension}}}$$

Ex:-



Euclidean distance :-

$$d(x_1, x_2) = \sqrt{(6-2)^2 + (3-8)^2} = \sqrt{41}$$

Manhattan distance :-

$$d(x_1, x_2) = |6-2| + |3-8| = 9$$

• - CHEBYSHEV DISTANCE :-

In case of $q \to \infty$, the distance equals to the maximum difference of the attributes. Useful if the worst case must be avoided:-

$$d_\infty(X, Y) = \lim_{q \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^q \right)^{1/q}$$

$$= \max(|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|).$$

Ex:- $d_\infty((2,8),(6,3)) = \max(|2-6|, |8-3|) = \max(4,5) = 5$

PROPERTIES OF DISTANCE METRICS :-

- Non-negative, $dist(x, x') \geq 0$
- Symmetric, $dist(x, x') = dist(x', x)$
- $dist(x, x') = 0 \iff x = x'$.

Triangular inequality :- $dist(x, x') \leq dist(x', x'') + dist(x'', x)$

↳ will be discussed in detail later.

Date:

HAMMING DISTANCE EXAMPLE.

$$dist(x, x') = \sum_{i=1}^{d} (1 - \delta_{x_i = x_i'})$$

$\delta_{x_i = x_i'} = 1$ if values are equal, otherwise 0

So, Hamming Distance = number of mismatches.

or $dist(x, x') = \sum_{i=1}^{d} (\delta_{x_i = x_i'})$

where $\delta_{x_i = x_i'} = 0$ if values are equal, 1 otherwise.

Ex:- Word 1 : KAROL
Word 2 : KAREL

	K	A	R	O	L	E
K	1	0	0	0	0	0
A	0	1	0	0	0	0
R	0	0	1	0	0	0
O	0	0	0	1	0	0
L	0	0	0	0	1	0
E	0	0	0	0	0	1

Position	Word 1	Word 2	Same/Different
1	K	K	Same.
2	A	A	Same
3	R	R	Same
4	O [000100]	E [000001]	Different, dist = 1
5	L	L	Same

COSINE SIMILARITY FORMULA.

$$\cos\theta = \frac{x \cdot x'}{\|x\|\|x'\|}$$

$x \cdot x'$ → Dot product

$\|x\|\|x'\|$ → magnitude

$\cos\theta \to 1$ documents are identical in direction.

$\to 0 \to$ documents are orthogonal, completely different.

Date:

Cosine Distance $= 1 - \cos\theta$.

Vocabulary (3 terms): [tea, time, two]

- Doc 1 $= (2, 0, 0)$ → label = Tea
- Doc 2 $= (2, 2, 0)$ → label = Tea
- Doc 3 $= (0, 0, 3)$ → label = Two
- Doc 4 $= (0, 1, 1)$ → label = Two
- NewDoc $= (1, 1, 0)$

Magnitude / Norms :-

$\|NewDoc\| = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2} = 1.414$

$\|Doc\ 1\| = \sqrt{2^2 + 0^2 + 0^2} = 2.00$

$\|Doc\ 2\| = \sqrt{2^2 + 2^2 + 0^2} = \sqrt{8} = 2.828$

$\|Doc\ 3\| = \sqrt{0^2 + 0^2 + 3^2} = \sqrt{9} = 3.00$

$\|Doc\ 4\| = \sqrt{0^2 + 1^2 + 1^2} = \sqrt{2} = 1.414$

Dot products :-

New · Doc 1 $= (1)(2) + (1)(0) + (0)(0) = 2$

New · Doc 2 $= (1)(2) + (1)(2) + (0)(0) = 4$

New · Doc 3 $= (1)(0) + (1)(0) + (0)(3) = 0$

New · Doc 4 $= (1)(0) + (1)(1) + (0)(1) = 0 + 1 + 0 = 1$

$\cos\theta = 0.707$ New, Doc 1

$\cos\theta = 1.000$ New, Doc 2

$\cos\theta = 0$ New, Doc 3

$\cos\theta = 0.5$ New, Doc 4

Cosine Distances $= 1 - \cos\theta$

New, Doc 1 $= 0.29$

New, Doc 2 $= 0$

New, Doc 4 $= 0.5$

New, Doc 3 $= 1$.

Date:

smaller distance = more similar.

k = 3 :-

0 , 0.29 , 0.5
Doc2 , Doc1 , Doc4
→ Tea Tea Two

Majority voting "Tea".

KNN ERROR RATE :-

$$\text{Error Rate} = \frac{\text{Number of Incorrect Predictions}}{\text{Total Predictions}}$$

Accuracy = 1 − Error Rate

Factors Affecting KNN Error Rate :-

- Value of K.
  - Small k → low bias, high variance Overfitting
    very sensitive to noise
  - Large k → High bias, lower variance Underfitting
- Distance → Euclidean, Manhattan, Cosine, Hamming distance
- Feature scaling
  ↳ If features are not normalized
    ↳ higher error
- Data quality
  Noisy, Imbalanced increase error.

Ex :- total = 20
correct = 16 , Wrong = 4

$$\text{Error} = \frac{4}{20} = 0.20$$

Accuracy = 1 − error rate
= 1 − 0.20 = 80%

tie breaker → weight $= \frac{1}{\text{distance}}$

KNN Regression

Error metrics :-

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

$$RMSE = \sqrt{MSE}$$

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

Confusion Matrix

Predicted ↓ / Actual	+	−	
+	True Positive	False positive	Precision $= TP / (TP + FP)$
−	False Negative	True Negative	

↑ when false positive are costly.

$$\text{Recall} = \frac{TP}{TP + FN}$$

↳ when false negative are costly

F1 score Harmonic mean

$$F1 = \frac{2 \cdot \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Error Rate = 1 - Accuracy.