Machine Learning

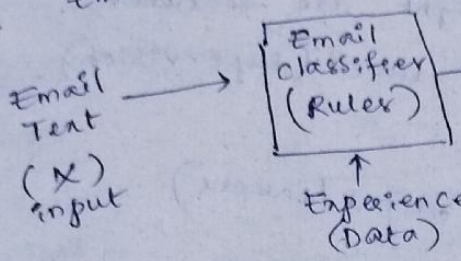
28/04/2025.

It has algorithms too. It is an algorithms ← tasks

- classification
- prediction
- Clustering
- dimensionality reduction

Example:

Email classification.



if
- domain email id is not repeated.
- contains words like discount, free, lottery, win.

then
Spam

else
not spam

Learning: learn from the experience (data)

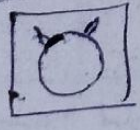
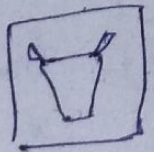
Data is of two types Supervised, Unsupervised.

a) Supervised (x and Y)

ex email

X: Email	Y: label
Meeting at 10:00 AM	not spam
Won $10000	Spam

ex: Recognition of different Animals.

Feature	label
	Cat
	Dog

example for supervised learning

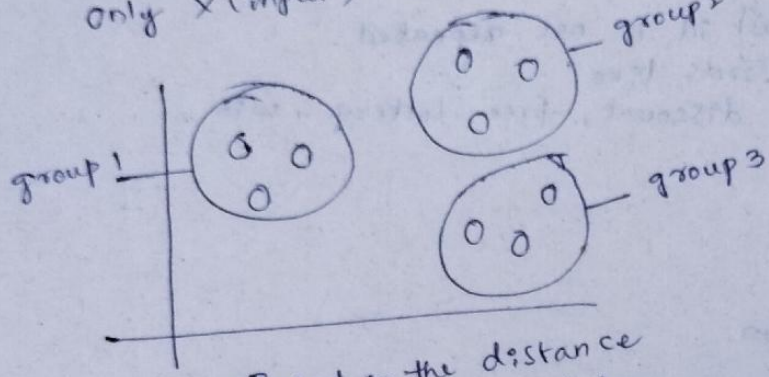
Tomorrow, will it rain ? ← Featurer → label (Y)

temp	humidity	wind	sunlight	Rain ?
				Yes
				NO

- temp, humidity, wind, sunlight are featurer
- will it rain (Rain ?) is label (supervision).

b) Unsupervised Learning (Y)

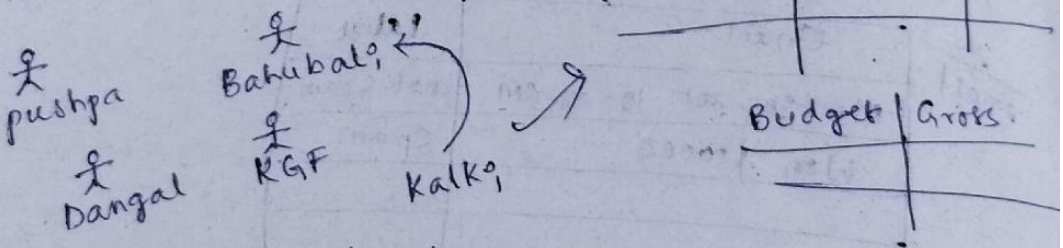
only X (input) and no Y (output/label)



Based on the distance between the data we are classified into groups.

Real world Example:

- Movie recommendation System



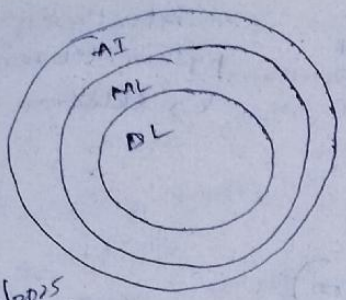
Kalki will be watched by the person who watcher bahubali.

c) Reinforcement learning (trial and Error) (DeepSeek R1)

learning by trying.

ex: Bicycling, Swimming

d) Semi- supervised learning.



Deeplearning - it uses Neural Network to solve the problem.

What is Neural Network?

collect the signals and send forward. — It is inspired by human brain (Neurons)

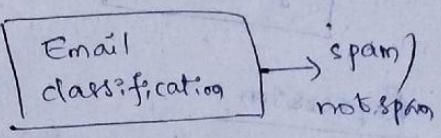
29/01/2025

Steps

1) Train
2) Test
3) Deploy

X	Y
—	NO
—	Yes

Given by human first



30/01/2025

Machine learning

Dataset

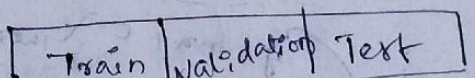
Labelled — Labelled data is used for supervised learning

unlabelled — Unlabelled data is used for unsupervised learning

Supervised learning

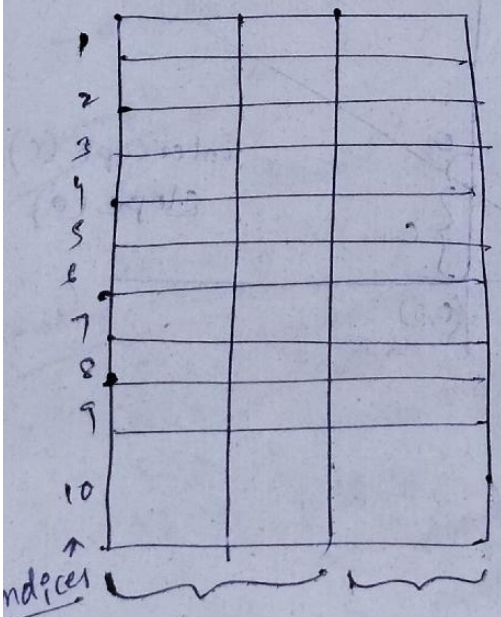
1) Train
2) Test
3) Deployment

Splitting the dataset



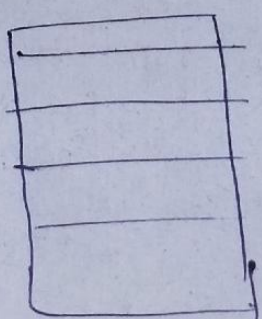
ex: Solved Example — Exercise Question — Exam Question

① Train-Set → 80% (1,7,2,3,5 ...)
② Validation Set / development Set → 10% (1,6)
③ Test Set → 10% (9,10)

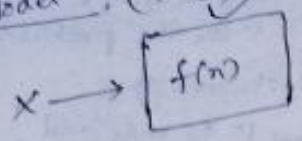


Cross-validation

- if dataset size is small dataset.



Model : (x, y) — True label



predicted label

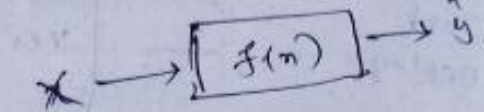
Intuitively python-libraries
↳ sklearn

ex: y = true age
$\hat{y}$ = predicted age

$\hat{y} = f(n)$ — Model (function)

Regression

↳ Simple Linear Regression.

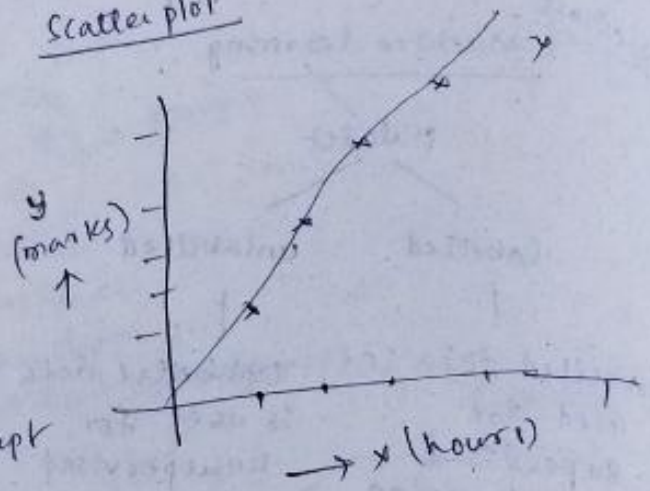


Hours	Marks
1	50
2	55
3	65
4	70
5	80

Train data.

10

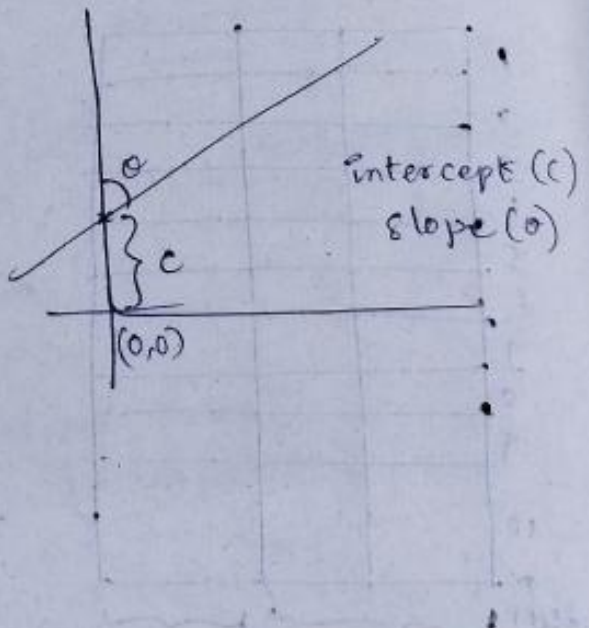
Scatter plot



$y = mx + c$ — slope, intercept

output

$y = \beta_1 x + \beta_0$



Data preprocessing

Categorical – Nominal
– Ordinal

1) Missing values
2) Encoding
3) feature scaling
4) Train val Test split

ex:

Colors	T-Shirt size
Red	L
Blue	M
Black	S
White	XL

↳ Nominal Value

↳ ordinal value.
S<M<L<XL

iloc

Weight	Weight (0-1)
50	0.5
60	0.6
70	0.7
80	0.8
100	1

↳ we get by dividing by 100

Missing value

pandas
- fillna()
  - ↳ mean
    median
    mode.
- dropna()

imputer class is imported from sklearn.preprocessing

imputer (missing_value, strategy)

↳ it takes two parameter

example code:

```
from sklearn.preprocessing import Imputer
imputer = Imputer(missing_value="NaN", strategy="mean")
```

encoding

S:1 Dictionary/map
M:2 – level encoder
L:3 – Onehotencoder
XL:4

(if it is categorical & nominal data is present equal importance)

Scaling

1) Standardization
2) Min_Max Scaling

bringing large number to a range

$z = \frac{x-\mu}{\sigma}$ → normal distribution

↳ standard normal distribution

x	x − μ
2	2−6 = −4
4	4−6 = −2
6	6−6 = 0
8	8−6 = 2
10	10−6 = 4

μ = 6

input		output
Salary	own house	Loan?
100,000	1	Y

Here the salary is very high & no. of own houses are very small to make them equal we require scaling

Standardisation

from SIC

Min-Max Scaling

$$X' \quad \frac{X - X_{min}}{X_{max} - X_{min}}$$

Conclusion

Data preprocessing

1) Missing value & handling it
   - fillna()
   - imputer()
2) Encoding
   - map()
   - LevelEncoder()
   - OnehotEncoder()
3) Scaling
   - Standard scaler
   - Min-max scale
4) Split
   ↳ You have to learn and find out the library

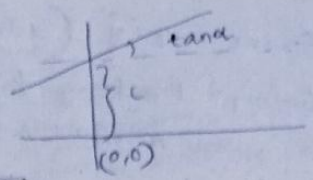
Model

Data

hours of study x	Marks Y
1	50
2	55
3	65
4	70
5	80

$\hat{y} = f(x)$

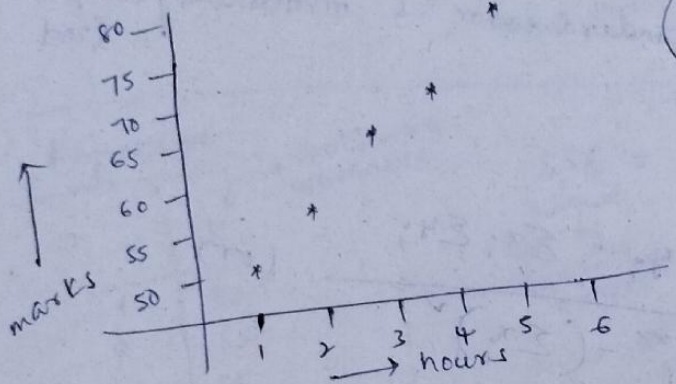
↳ prediction.



$x \rightarrow \boxed{\hat{y} = mx + c} \rightarrow \hat{y}$

m = slope
c = intercept
} parameter (2)

(ChatGpt has 175 Billion parameters)



Learning is always denoted with error.

* Error *

By the linear regression, we can solve

Let equation of be $\hat{y} = 10 + 5x$ then data will be straight line

x	y	$\hat{y} = 10+5x$ model 1 $\hat{y}$	ε_i $y - y_i$	$\hat{y} = 40+2x$ model 2	ε_i $-\hat{y}+y$
1	50	15	35	38	12
2	55	20	35	36	14
3	65	25	40	34	16
4	70	30	40	32	18
5	80	35	45	30	20
			195		80

Fill the table (✓)

Error = $y - \hat{y}$ (actual value - predicted value)

→ Training - Finding the parameter that will give the minimum error

$\hat{y} = \theta_0 + \theta_1 x$

parameters θ_0, θ_1

Error can be make as positive by.

Error = $|y_i - \hat{y}_i|$ (or) — make all error positive

Error = $(y_i - \hat{y}_i)^2$ — Amplifier the error.

finding θ_0 and θ_1 standard error is minimum / ways to find

1) using the formula.

$\theta_0 = \bar{y} - \theta_1 \bar{x}$

$\theta_1 = \dfrac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$ (or)

$\dfrac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$ → it is little bit more calculative.

Solve

$y = 10 + 5x$.

x	y	$x_i y_i$	$(x_i)^2$
1	50	50	1
2	55	110	4
3	65	195	9
4	70	280	16
5	80	400	25
15	320	1035	55

$\theta_1 = \dfrac{5(1035) - 15 \cdot 320}{5(55) - 225}$

$= \dfrac{375}{50} = 7.5$

$\theta_0 = \bar{y} - \theta_1 \bar{x} = \dfrac{320}{5} - 7.5(3)$

$= 64 - 22.5 = 41.5$

$\therefore \theta_0 = 41.5$

$\theta_1 = 7.5$

$\therefore$ prediction

$\hat{y} = \theta_0 + \theta_1 x_i$

$\hat{y} = 41.5 + 7.5 x_i$

Then the table will be

x	Y	prediction $\hat{y} = \theta_0 + \theta_1 x$	$\varepsilon_i = y_i - \hat{y}_i$
1	50	49	1
2	55	56.5	1.5
3	65	64	1
4	70	71.5	1.5
5	80	79	1

Total Error = 6

Conclusion in Machine learning

1) model contains L sparameters have some Errors

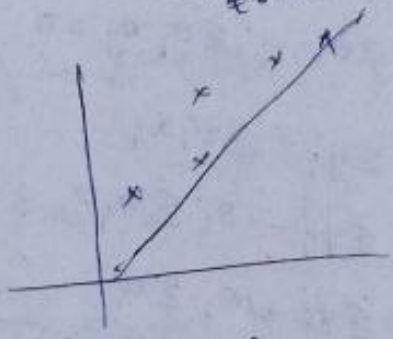
2) Train - to reduce the standard error or to get minimum error by finding the parameters.

Regression. → continuous variable.

X	Y
1	50
2	55
3	65
4	70
5	80

$SSE = \sum_{i=1}^{n} (y_i - \hat{y})^2$ 06/08/2025

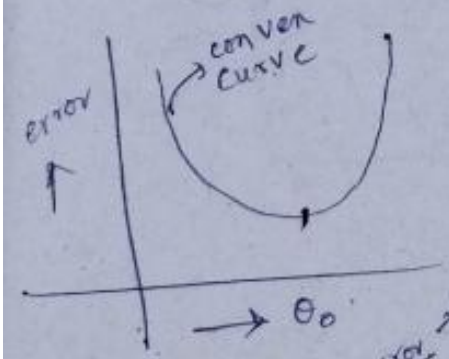
Sum of Square of Errors



$\theta_0 = \bar{y} - \theta_1 \bar{x}$

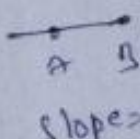
$\theta_1 = \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2}$

Why these attributes will give minimum error ? and how



θ_0 ← Changing, θ_1 Constant

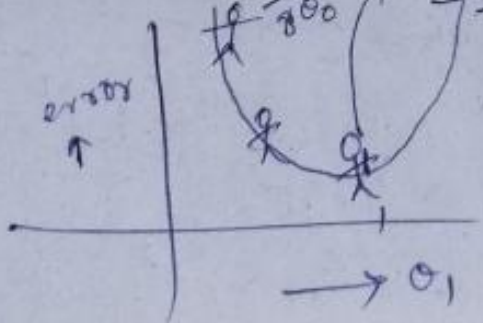
Constant Changing



-ve slope

+ve slope

slope = 0



negative slope

$\text{Slope} = \frac{dy}{dx}$ ✓ derivative

↳ import concept for neural network

Find θ_0 and θ_1 such that. (intuition) ∂ - partial derivative.

$$\frac{\partial \text{Error}}{\partial \theta_0} = 0$$

and $$\frac{\partial \text{Error}}{\partial \theta_1} = 0$$

we know, $SSE = \sum (y_i - \hat{y})^2$

$$SSE = \sum (y_i - \theta_0 - \theta_1 x_i)^2 \quad \left[\because \hat{y} = \theta_0 + \theta_1 x\right]$$

$$\frac{\partial \text{Error}}{\partial \theta_0} = \frac{\partial}{\partial \theta} \sum (y_i - \theta_0 - \theta_1 x_i)^2 = 0$$

error

$$= 2 \sum (y_i - \theta_0 - \theta_1 x_i)(-1)$$

chain rule.

$$= -2 \sum (y_i - \theta_0 - \theta_1 x_i)$$

$$\Rightarrow \sum (y_i - \theta_0 - \theta_1 x_i) = 0$$

$$\sum y_i - \sum \theta_0 - \sum \theta_1 x_i = 0$$

$$\Rightarrow \sum \theta_0 = \sum y_i - \sum \theta_1 x_i$$

$$n\theta_0 = \sum y_i - \theta_1 \sum x_i$$

$$\theta_0 = \frac{\sum y_i - \theta_1 \sum x_i}{n} \longrightarrow \text{eq (2)}$$

$$\theta_0 = \frac{\sum y_i}{n} - \theta_1 \frac{\sum x_i}{n}$$

$$\boxed{\theta_0 = \bar{y} - \theta_1 \bar{x}}$$

for θ_1.

$$\frac{\partial \text{Error}}{\partial \theta_1} = 0$$

$$\Rightarrow \frac{\partial}{\partial \theta_1} \Sigma (y_i - \theta_0 - \theta_1 x_i)^2 = 0 \quad \leftarrow \text{this } x_i \text{ is within the sigma } (\Sigma)$$

$$2 \Sigma (y_i - \theta_0 - \theta_1 x_i)(-x_i) = 0$$

$$-2 \Sigma x_i (y_i - \theta_0 - \theta_1 x_i) = 0$$

$$\Rightarrow \Sigma x_i (y_i - \theta_0 - \theta_1 x_i) = 0.$$

$$\Rightarrow \Sigma x_i y_i - \theta_0 \Sigma x_i - \theta_1 \Sigma x_i^2 = 0$$

$$\Rightarrow \Sigma x_i y_i - \left(\frac{\Sigma y_i - \theta_1 \Sigma x_i}{n} \right) \Sigma x_i - \theta_1 \Sigma x_i^2 = 0 \qquad (\text{from eq}^n 2)$$

$$\Rightarrow \frac{n \Sigma x_i y_i - \Sigma x_i \Sigma y_i + \theta_1 (\Sigma x_i)^2 - n \theta_1 \Sigma x_i^2}{n} = 0$$

$$\Rightarrow n \Sigma x_i y_i - \Sigma x_i \Sigma y_i + \theta_1 (\Sigma x_i)^2 - n \theta_1 \Sigma x_i^2 = 0$$

$$\Rightarrow n \Sigma x_i y_i - \Sigma x_i \Sigma y_i = n \theta_1 \Sigma x_i^2 - \theta_1 (\Sigma x_i)^2$$

$$\Rightarrow \theta_1 \left(n \Sigma x_i^2 - (\Sigma x_i)^2 \right) = n \Sigma x_i y_i - \Sigma x_i \Sigma y_i$$

$$\Rightarrow \boxed{\theta_1 = \frac{n \Sigma x_i y_i - \Sigma x_i \Sigma y_i}{n \Sigma x_i^2 - (\Sigma x_i)^2}}$$

Regression

$\hat{y} = \theta_0 + \theta_1 x$

$\theta_0 = \bar{y} - \theta_1 \bar{x}$

$\theta_1 = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2}$

$= \frac{\Sigma (x_i - \bar{x})(y_i - \bar{y})}{\Sigma (x_i - \bar{x})^2}$

GPU.
Graphics processing unit
Colab
Machine learning requires GPU

GPU-T

independent dependent

x	y
1	50
2	55
3	65
4	70
5	80

Normal form Equation:

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ $X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}_{5\times 2}$

$y = \begin{bmatrix} 50 \\ 55 \\ 65 \\ 70 \\ 80 \end{bmatrix}$

numpy

$\theta = (X^T X)^{-1} X^T Y$ → normal equation / normal form.

numpy:

```
import numpy as np
x = np.array ( [ [1,1], [1,2], [1,3], [1,4], [1,5] ] );
Y = np.array ([50,55,65,70,80])
theta = np.linalg.im ( X.T @ X) @ X.T @ Y
print (theta)
```

→ $\begin{bmatrix} 41.5 \\ 7.5 \end{bmatrix}$

library → sklearn

above can also be learnt from the sklearn library.

SSE : $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, $\hat{y} = X\theta$

$$\hat{Y} = \begin{bmatrix} 49.5 \\ 56.5 \\ 62. \\ 71.5 \\ 75 \end{bmatrix}$$

$$= (Y - \hat{y})^T (Y - \hat{Y})$$

$$= (Y - X\theta)^T \cdot (Y - X\theta)$$

$$= Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X\theta$$

$Y^T X\theta = \theta^T X^T Y$

$1 \times 5.5 \times 2 . 2 \times 1$ | $1 \times 2 . 2 \times 5 . 5 \times 1$

1×1 | 1×1

$$= Y^T Y - 2 . \theta^T X^T Y + \theta^T X^T X\theta$$

$$\frac{\partial SSE}{\partial \theta} = 0 - 2 . X^T Y + 2 X^T X . \theta = 0$$

$$\Rightarrow 2 . X^T X\theta = 2 X^T Y$$

$$\Rightarrow X^T X\theta = X^T Y$$

$$\theta = \frac{X^T Y}{X^T X}$$

$$\boxed{\theta = (X^T X)^{-1} X^T Y}$$

Multiple Linear Regression

$$\theta = (X^T X)^{-1} \cdot X^T Y$$

size, bedrooms, age (independent) → [f(x)] → price (Dependent)

$$\hat{y} = \theta_0 * 1 + \theta_1 * \text{size} + \theta_2 * \text{bedrooms} + \theta_3 * \text{age}$$

parameter need to find are $\{\theta_0, \theta_1, \theta_2, \theta_3\}$

i.e $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 \theta_3 + \cdots$

Example:

independent variables (size, Bedrooms, Age) and Dependent variable (price)

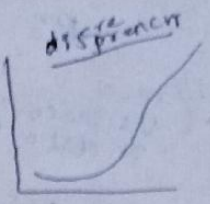
Size (sq ft)	Bedrooms	Age (years)	price (Y)
1500	3	15	400,000
1800	4	20	460,000
2400	3	10	56000
3000	4	8	60000
3500	5	5	2000

$$X = \begin{bmatrix} 1 & 1500 & 3 & 15 \\ 1 & 1800 & 4 & 20 \\ 1 & 2400 & 3 & 10 \\ 1 & 3000 & 4 & 8 \\ 1 & 3500 & 5 & 5 \end{bmatrix}_{5\times 4} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}_{4\times 1} \quad Y = \begin{bmatrix} 400000 \\ 460000 \\ 56000 \\ 60000 \\ 2000 \end{bmatrix}$$

- polynomial (linear) Regression.
- Evaluating Model performance.

PLR: $\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n$

↳ polynomial feature.



("overfitting")

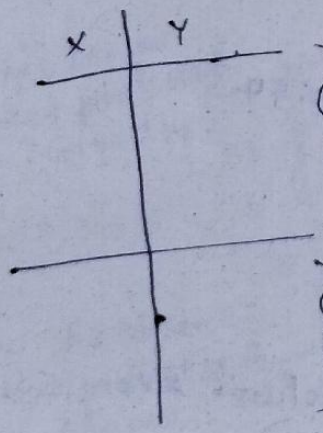
hours (x)	$(\text{hours})^2$ x^2	Y (marks)
1	1	
2	4	
3	9	
4	16	
5	25	

more error → less learn
less error → more learn.

Evaluation:

↳ Test
↳ Error
|
→ Mean Absolute Error
→ MSE
→ RMSE
→ R^2

Evaluation Metrics



Train
↳ Error
↳ $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
→ $MSE = \frac{SSE}{n}$
→ $RMSE = \sqrt{MSE}$

Test
↳ Evaluation
↳ Evaluation Metrics
↳ R^2
→ Adjusted R^2

Every ML algorithm will have any one of these error based on problem.

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

→ variance is explained by model.

Variance in data ←

variance: how the data is distributed around the mean.

if var(m) is less, most of the data near to mean.
else if var(m) is more, far from mean.

eg:

x	y	$\hat{y}$
1	50	49
2	55	56.5
3	65	64
4	70	71.5
5	80	79.

$\bar{y} = 64$

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = 0.986$$

$$R^2 = \frac{(50-49)^2 + (55-56.5)^2 + (65-64)^2 + (70-71.5)^2 + (80-79)^2}{(50-64)^2 + (55-64)^2 + (65-64)^2 + (70-64)^2 + (80-64)^2}$$

$$= \frac{1 + (-1.5)^2 + 1 + (-1.5)^2 + 1}{(-14)^2 + (-9)^2 + 1 + (6)^2 + (16)^2} = \frac{3 + 2(2.25)}{196 + 81 + 1 + 36 + 256}$$

$$= \frac{7.5}{570} = 0.013$$

Note: If $R^2 = 1$ → Good Model / perfect Fit

$R^2 = 0$ → Bad Model / not fitting.

$$\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(n-1)}{n-k-1} = 0.94$$

where n = no. of data points = 5

k = no. of features = 1

→ Adjusted R^2 will decrease when unnecessary colums are added to the dataset (ID, B>0065)

↳ It does not required to predict the marks

R^2 value remains same but adjusted R^2 value decreases

Classification

Evaluation Metrics:

Binary Classification

Email → f(x) → $Y = \{ \underset{1}{\text{spam}}, \underset{0}{\text{not spam}} \}$

discreted

	x	y	$\hat{y}$	
TN		0	0	✓
TP		1	1	✓
FN		1	0	✗
FP		0	1	✗
TN		0	0	✓
TP		1	1	✓
FP		0	1	✗
FN		1	0	✗
TN		0	0	✓

$$\text{Accuracy} = \frac{5}{9} \times 100$$

Confusion Matrix

actual \ predicted	1	0
1	TP 2	FN 2
0	FP 2	TN 3

spam & marked as spam

Spam but not marked as not spam

not spam and not marked as spam

Not spam but marked as spam

TP - True possible
TN - True Negative
FN - False Negative

Example:

Features (dimensions) → Cancer prediction → $\{ \underset{1}{\text{yes}}, \underset{0}{\text{NO}} \}$

	1	0
1	TP	FN
0	FP	TN

has Cancer and predicted correctly.

has cancer but not predicted (more dangerous)

no Cancer and predicted no Cancer

no Cancer but predicted Cancer

Classification

eg:

X	Y	$\hat{y}$
	0	0
	0	0
	1	0
	0	1
	1	1
	1	1
	0	1
	1	0
	0	0
	1	1

→ Binary classification
↳ Two labels
{1, 0}
+ve (1) — −ve (0)
Spam / not spam
Cancer / not cancer.

- Accuracy
- precision
- Recall
- F1 score

} Near to 1 = Good model
Near to 0 = Bad model

Confusion Matrix

actual \ prediction	1	0
1	TP: 3	FN: 2 (Type 2 error)
0	FP: 2 (Type 1 Error)	TN: 3

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

$$= \frac{6}{10} = 0.6 = 60\%$$

$$P(\text{precision}) = \frac{\text{No. of Correct positive predictions}}{\text{No. of positive prediction.}} = \frac{TP}{TP+FP}$$

$$= \frac{3}{5} = 0.6$$

$$R(\text{Recall}) = \frac{\text{No. of Correct positive Production}}{\text{No. of actual positive cases}} = \frac{TP}{TP+FN}$$

$$= \frac{3}{5} = 0.6$$

$$F1 = \frac{2*P*R}{P+R} = \frac{2*0.6*0.6}{0.6+0.6} = 0.6$$

To understand the complete picture, we need precision, recall and F1 along with accuracy.

eg:

x	y	$\hat{y}$
	0	0
	0	0
	0	0
	1	0
	0	0
	0	0
	0	0
	1	0
	0	0
	0	0

actual

	1	0
1	TP 0	FN 2
0	FP 0	TN 8

Accuracy = $\frac{8}{10}$ = 80%.

P = 0

R = 0

F1 = not defined.

↳ Unbalanced data.

eg:

prediction

email Classification actual	1	0
1	TP 1	FN 2
0	FP 3	TN 14

Accuracy = $\frac{15}{1+2+3+14} = \frac{15}{20} = \frac{3}{4} = 0.75$

precision = $\frac{1}{1+3} = \frac{1}{4} = 0.25$.

Recall = $\frac{1}{1+2} = \frac{1}{3} = 0.33$

F1 = $\frac{2 \times 0.25 \times 0.33}{0.25+0.33} = 0.27$

Multiclass Classification

↳

y	$\hat{y}$
0	0
0	0
0	2
0	0
0	2
1	1
1	1
1	1
1	0
1	1
2	1
2	1
2	2
2	2
2	2

0 - Regular Email
1 - Social Email
2 - Spam Email

predicted

actual

	0	1	2
0	3	0	2
1	1	4	0
2	0	2	3

Accuracy: $= \frac{10}{15} = 0.66$

class 0: TP = 3, TN = 9, FP = 1, FN = 2

P = 0.75 R = 0.6 F1 =

class 1: TP = 4, TN = 8, FP = 2, FN = 1

P = R = F1 =

class 2: TP = 3, TN = 8, FP = 2 FN = 2

P = R = , F1 =

Macro P = $\frac{\quad}{3}$

Macro R = $\frac{\quad}{3}$

Macro F1 = $\frac{\quad}{3}$

Eg:

y	$\hat{y}$
0	2
0	2
0	0
0	0
0	1
1	1
1	1
1	2
1	0
1	1
2	0
2	0
2	0
2	2
2	2

Evaluation Metrics

- Classification
  - Accuracy
  - precision
  - Recall
  - F1 score

Classification Algorithms

① Decision Tree

{ X, Y }
↳ discrete

{ spam, not spam }

Cloudy, weather prediction, raining? → Will you take Umbrella? → Y/N

X			Y
Cloudy	weather prediction	raining	Umbrella ?
N	N	N	N
N	N	N	N
Y	N	Y	Y
Y	Y	Y	Y
Y			

Accuracy = 100%

Sample code:

```
if raining
    then yes
elif cloudy:
    if rain prediction
        then yes
    else
        NO
else
    NO:
```

Decision Tree → Raining (Feature)
- Y → Yes (unique feature value)
- N → Cloudy
  - Y → Rain prediction
    - Y → Yes
    - N → NO
  - N → NO

predicted label / decision node

1. Uncertainity and Entropy (TENET movie)

basically entropy measures uncertainity

Example: will the students wear uniform tomorrow ?

C1	C2	C3	C4
Y	Y	Y	Y
Y	Y	N	Y
Y	Y	N	N
Y	N	N	N
Y			

C1: $P(U) = \frac{4}{4} = 1$, $P(\overline{U}) = \frac{0}{4} = 0$
↑ More certainity (Entropy is less)

C2: $P(U) = 3/4$, $P(NU) = 1/4$

C3: $P(U) = 1/4$, $P(NU) = 3/4$

C4: $P(U) = 2/4 = 1/2$, $P(\overline{U}) = 2/4 = 1/2$
↑ less certainity (Entropy is more)

My calculate $P(U)$, $P(\overline{U})$ for C2 & C3

$\log 1 = 0$
$\log 0$ = undefined

Entropy $= -P_1 \log_2 P_1 - P_2 \log_2 P_2$

$\{P_1, P_2\} = \{U, NU\}$ (U: uniform, NU: not uniform)

$E(C_1) = -1 \log_2 1 - 0 \log_2 0 = 0$

$E(C_4) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = -\log_2 \frac{1}{2} = 1$

$E(C_2) = -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} = 0.81$

$E(C_3) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.81$

Note: Uncertainity is more $\Rightarrow$ Entropy is more

i.e Uncertainity $\propto$ Entropy

Entropy(1) = High uncertainity / entropy

Entropy(0) = low uncertainity / entropy

Example: watch Movie ?

Movie	Genre	Time	Watch?
M1	Action	Even	Yes
M2	comedy	Morn	No
M3	Action	Morn	Yes
M4	Drama	Even	No
M5	Comedy	Even	No

$p(Yes) = 2/5$, $p(No) = 3/5$, $E = 0.97$ → reducing with the uncertainity / entropy

Genre: Action, comedy, Drama

- Action: $\{m_1, m_3\}$ Yes, Yes — $p(Yes) = 1$, $p(No) = 0$, $E = 0$
- comedy: $\{m_2, m_5\}$ No No — $p(Yes) = 0$, $p(No) = 1$, $E = 0$
- Drama: $\{m_4\}$ No — $p(Yes) = 0$, $p(No) = 1$, $E = 0$

$p(Yes) = 2/5$, $p(No) = 3/5$

Time: Morning, Evening

- Morning: $\{m_2, m_3\}$ No, Yes — $p(Yes) = 1/2$, $p(NO) = 1/2$, $E = 1$
- Evening: $\{m_1, m_4, m_5\}$ Yes, No, Yes — $p(Yes) = 2/3$, $p(No) = 1/3$, $E =$

Select the attribute which help you take the decision.

Here, genere is suitable attribute (feature) which decreases the uncertainity (entropy) more than the attribute time.