* Feature extraction → process of selecting and extracting the relevant features from the raw data.

1000 features → ✗ All the features you cannot use model training.

* Curse of dimensionality

↳ with increase in no of features:-

① model training becomes computationally expensive.

② Model interpretation becomes complex.

f_1 f_2 f_3 - - - - f_{1000}

① creating new feature

ex.1

distance	time	Speed = Distance/time
50	2	50/2
100	3.5	100/3.5
—	—	—
—	—	—

ex.2

temp	3 months moving avg temp.
21°	- NaN
22°	NaN
23°	→ (21+22+23)/3 = 22°
24°	
25°	
:	
:	

No of Rooms	Area of 1 room	effective carpet area
2	100	200
3	200	600
4	100	
5		

② Modifying existing feature

ⓐ Changing the data type

Age	Modified age
15 yrs	15
20 yrs	20
25 yrs	25

eg

install	Modified - install
15+	15
20+	20
25+	25

eg Date

	Day	Month	Year
15-2-2022	15	2	2022
16-2-2022			

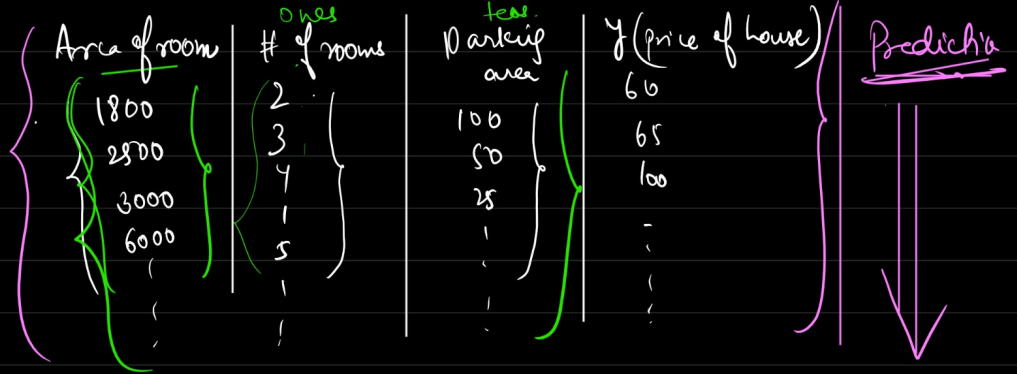
* Feature Scaling (optional) / transformation.

→ To bring all the features on same scale.

why?

Scaling / not do scaling

↓ no affect



ML → Mathematical relationship

⇓

Computation

$2 \times 3 = 6$

$5 \times 2 = 10$

$3 \times 5 = 15$

$250 \times 328 = ?$

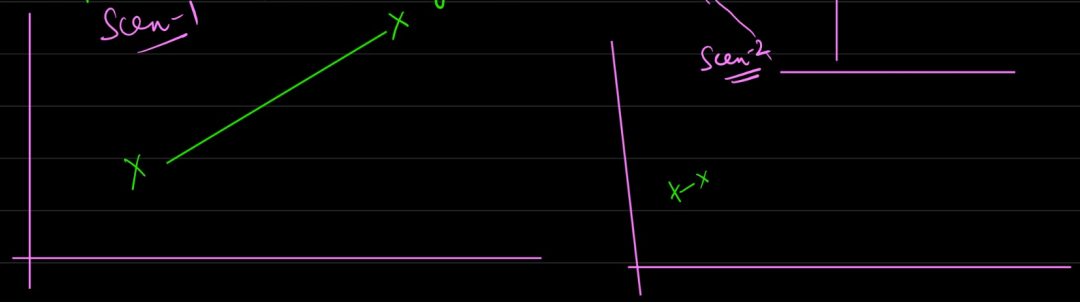
Why Scaling ?

→ Computation is less expensive.

→ Gradient descent

↳ Optimisation becomes faster

⇒ Interpretation becomes easy.



* Type of Scaling

① Standardisation (ML algorithms)

$$Z_{score} = \frac{x_i - \bar{x}}{\sigma}$$

SND ⇒ $\mu = 0, \sigma = 1$

Age	Age_scaled
25	$\frac{25 - 24.6}{1.03}$
26	$\frac{26 - 24.6}{1.03}$
23	
24	
25	

$\bar{x} = 24.6$

$\sigma = 1.03$

② Normalisation (min-max Scaler) → DL Algorithms → PCA

⇓

[0, 1]

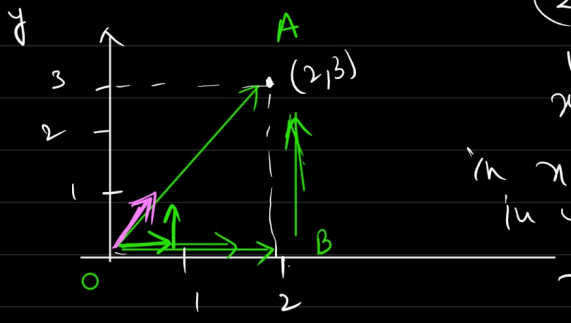
Age	Age Scaled
25	$\frac{25-23}{26-23} = \frac{2}{3}$
26	$\frac{26-23}{26-23} = 1$
23	$\frac{23-23}{26-23} = 0$
24	
25	

min = 23 → 0.

max = 26 → 1

$$x_{scale} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

③ Unit Vector



$(2,3) \rightarrow 2\hat{i} + 3\hat{j}$

x y

in x dir — 2 units

in y dir — 3 unit

$$OA^2 = OB^2 + AB^2$$

$$\vec{OA} = \sqrt{OB^2 + AB^2}$$

$$\|\vec{OA}\| = \sqrt{4+9} = \sqrt{13}$$

$$\hat{A} = \left(\frac{2}{\|OA\|}, \frac{3}{\|OA\|}\right)$$

$$\sqrt{\left(\frac{2}{\sqrt{13}}\right)^2 + \left(\frac{3}{\sqrt{13}}\right)^2} = \sqrt{\frac{4}{13} + \frac{9}{13}} = \sqrt{\frac{13}{13}} = 1$$

Selecting the right feature

1000 — feature → Top 10 feature

1. Filter method
2. Embeded method
3. Wrapper method.

① Filter method

→ Correlation

(f1, y) (f2, y)

→ Multicollinearity (VIF)

f1	f2	Price of house (y)
# of house	# of parking	

→ correlation among feature themselves.

$X_1 \approx (X_2\ X_3\ X_4)$

$X_1 = X_2$

$7X_1 + 3X_2$

$3X_1 + 7X_2$

$10X_1$

$10X_2$

x_1	x_2	x_3	x_4	y

say $X_1 = X_2$

$10x_1 + 3x_2 + 5x_3 + 4x_4 = y$

② Wrapper method → Recursive feature elimination.

- Forward select
- Backward selection

③ Embeded method

→ Lasso Regression, Elastic Net. (L1)

→ Decision tree → XGBoost, LightGBM

Data Encoding

ML Algorithms ⇓ numerical values

1. Nominal / OHE
2. Label and ordinal
3. Target Guided ordinal encoding.

1. Nominal / OHE | Dummy variable creation.

→ Categorical data to numerical data

→ No order in the data.

Status	dummy S/M	dum-ma	dummied
Single	1	0	0
married	0	1	0
in a relationship	0	0	1
Single	1	0	0
married	0	1	0
Single	1	0	1
in a relat.	0		
=	0		

Single = 0

Married = 1

in a rel^{n} = 2

	Single	married	in a relⁿ
Single	1	0	0
Married	0	1	0
in a relⁿ	0	0	1

	DS	DE	BA	DA
DS	1	0	0	0
DE	0	1	0	0
BA	0	0	1	0
DA	0	0	0	1

No of Dummy Variables create = No of Unique Categories in the columns

	S	M	R
S	1	0	0
M	0	1	0
R	0	0	1

for n unique groups = n−1 variable will be enough to explain the original features

* disadvantage
↳ A column has many categories

f1
St
D
T
B
⋮
1000

– 5 each ⋮ u → 999 → new columns
⇓
Curse of dimensionality

* Solⁿ ⇒ Select of top 5 or top 10 unique categories in the feature and make that no dummy variables.

→
→
→

f1.value_counts()

Dum DS	Dum DE	Dum DA
1	0	1
0	0	0
0	1	0
0	0	0
0	0	0
0	0	0

DS	− 102
DE	− 90
DA	− 10

② label and Ordinal encoding

* Label encoding — assign numerical label to each category.

Red − 1
Green − 2
Yellow − 3

1
2
3
1
1
2
3

DS − 1
DE − 2
DA − 3

1
2
3
1
2
3
1

Adv → * No problem of curse of dimensionality.

disadvantage :- ML can assume that there is some inherent order due to numerical ranking.

* Ordinal encoding

High School — 1
College — 2
Post Graduate — 3
PhD — 4

Grade
A — 4
B — 3
C — 2
D — 1

③ target guided ordinal encoding (also useful for Nominal data with higher no of categories)

→ based on their relationship with target variable
→ When we have large no of Unique Categories in Categorical value.
→ Categorical grouping with mean/ median of corresponding target variable

Ex.

tips

time	total_bill
lunch	150
lunch	120
lunch	100
dinn	—
br	—
di	—
b	—

105
105
105
150
150
90
90

groupby('time')['total_bill'].mean()

lunch — 105
dinner = 150
br = 90