Feature Extraction: The process of selecting and extracting the most important features from raw data.

Dataset → 1000 Features
↓
Take most important Features
↓
Train them with ML algorithms

① **Feature scaling:** Here we scale down the data using standardization (zscore) or normalization.

Age (Y)	Heigh(cm)	Weight (Kg)	BMI
24	120	42	–
34	140	54	–
30	160	44	–
39	170	62	–

By using Z-score $\left(z = \frac{x_i - \bar{x}}{\sigma} \right)$ we can scale down the data like Age data or Height, Weight data to (−3, +3) Range.

This is called Feature scalling. It is used to optimize many mL Agorithms who counts the distance between two data points

After standardization, $\mu = 0$ and $\sigma = 1$ (will be)

Code has been uploaded to github

Normalization: It is another technique to scale down (Feature) data between [0,1] range. (Also known as min max scaling)

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

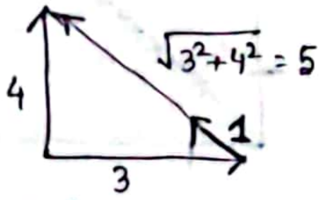
(ii) **Feature Selection:** Here, we just pick the most important feature to train our ML models.

We use 1) Filter method 2) Embedded method techniques for Feature selection.

(iii) **PCA (Principal Component Analysis):** It is also a type of Algorithm which helps us to extract features.

Unit Vector: The vector which has a magnitude of 1, that is unit vector.

Vector $\vec{x}$ at point (3,4)



$$\|\vec{x}\| = \sqrt{3^2+4^2} = 5$$

unit vector of $\vec{x}$ is $\hat{u} = \left(\frac{3}{\|\vec{x}\|}, \frac{4}{\|x\|}\right)$

$$= \left(\frac{3}{5}, \frac{4}{5}\right)$$

$$\|\hat{u}\| = \sqrt{\left(\frac{3}{5}\right)^2 + \left(\frac{4}{5}\right)^2}$$

$$= 1$$

So, we can scale down 2D vector with their magnitude and unit vector.

The code is uploaded to github.

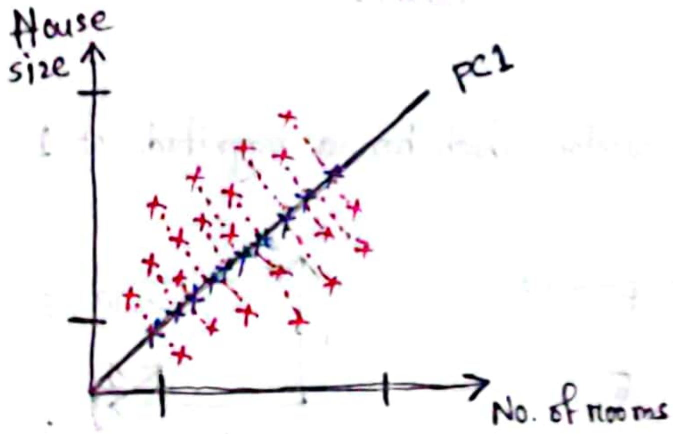
PCA (Principle Component Analysis): (Elaboration)

This is a type of algorithm which helps us to reduce dimensions.

Suppose in a dataset, we have 3 features.

1) No. of rooms 2) House size 3) Price

We can use PCA algorithm to make a single feature from the two (No. of rooms, House size) feature.



Here using the pc1 line, we are projecting the data of both of the feature and making it a single feature. It will eventually have data loss a bit but we can reduce features and make our model more efficient.

Data Encoding: The aim of data endoding is to convert a categorical feature to a suitable numerical feature in order to train our ML model.

Types of data encoding → ① Nominal or One Hot encoding
② Ordinal & Label encoding
③ Target guided ordinal encoding

One hot (Nominal) Encoding:

Suppose, we have a dataset →

No. Room	size	Location	Price
4	30	Banglore	-
3	40	Delhi	-
1	10	Delhi	-
6	15	Noida	-
5	25	Bangalore	-

Here, the Location column is categorical. We will take ~~in~~ unique column values and put them as new columns in the dataset and provide values 0 or 1

No. of rooms	size	Location	Price	Bangalore	Delhi	Noida
4	30	Banglore	–	1	0	0
3	40	Delhi	–	0	1	0
1	10	Delhi	–	0	1	0
6	15	Noida	–	0	0	1
5	25	Banglore	–	1	0	0

Disadvantage:

1) Suppose if we had 10 different locations, then we had to create 10 columns which would be a sparse matrix (lot of 0s and 1s) and lead to overfitting the model

2) If we had 1000 new unique locations, the one hot method would introduce 1000 new feature which would obviously decrease the model accuracy a lot.

The coding part has been uploaded in github.

Label & Ordinal encoding:

Label encoding is another way to assign numeric values to the categorical feature. It provides a unique number for each categorical value.

But the problem with label encoding is, it can mislead ML algorithms into assuming non existantial ordinal relationship. Which causes inaccuracy and bias.

Suppose if you label ~~order~~ encode colors → "red", "blue", "green" with 0, 1 and 2. ML model will think red < blue < green which make no sense.

That is where ~~ordinal encoding concept come~~. alternative encoding methods like one-hot-encoding can be used to avoid drawbacks.

Ordinal encoding is used when you want to give your categorical data a custom order according to your choice. Because you can set the order in this encoding method.

Both of their code has been uploaded to github.

Target guided Ordinal encoding:

Suppose you have a categorical data which has a relationship with any numerical feature. You can make that numerical feature your target feature. What you will do is, you group-by your categorical feature and for each group of categorical data, you will find the mean, and median in the target feature. That mean or median value will be used as an encoded value.

Example→ dataset:

City	Price
London	100
Paris	200
London	200
New york	200
Paris	100
London	200

→ Group by and find out the mean (if there is no outliers) else median

city	price
London	166.7
Paris	150
New york	200

Now map the mean prices with the cities

city	Price	city_encoded
London	100	166.67
Paris	200	150
London	200	166.67
Newyork	200	200
~~London~~		
Paris	100	150
London	200	166.67

This is how you encode categorical feature in target guided ordinal encoding technique.

Code has been uploaded to github.