ML-3

What’s covered

- ML, ML types
- examples
- Few jargon terms
- Unsupervised learning

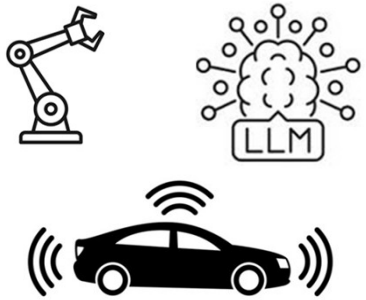
ML Types

Supervised

Unsupervised

Semi-supervised

Reinforced



Machine learning

Unsupervised

Dependent variable

Supervised

Not a part (Unlabeled)
(x1,x2,x3,x4,x5)

A part (labeled)
Y= f(x1,x2,x3,x4,x5)

Types

Clustering
Dim.Red
ARM

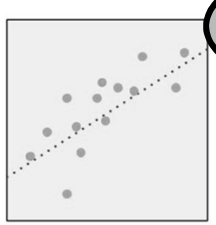
Algorithms

K-means
Clustering
PCA, FA, ANN
Apriori

Variable type

Type

Regression



Continuous

Algorithms

SLR, MLR, ANN,
R&LR, SVR

Type

Classification

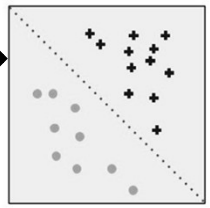
categorical

ordinal

Algorithms

Ordinal regression
Ordered logit
Ordered probit

Nominal



Algorithms

CART, C4.5, CHAID,
RF, LR, LDA, ANN
SVM, SGM, KNN

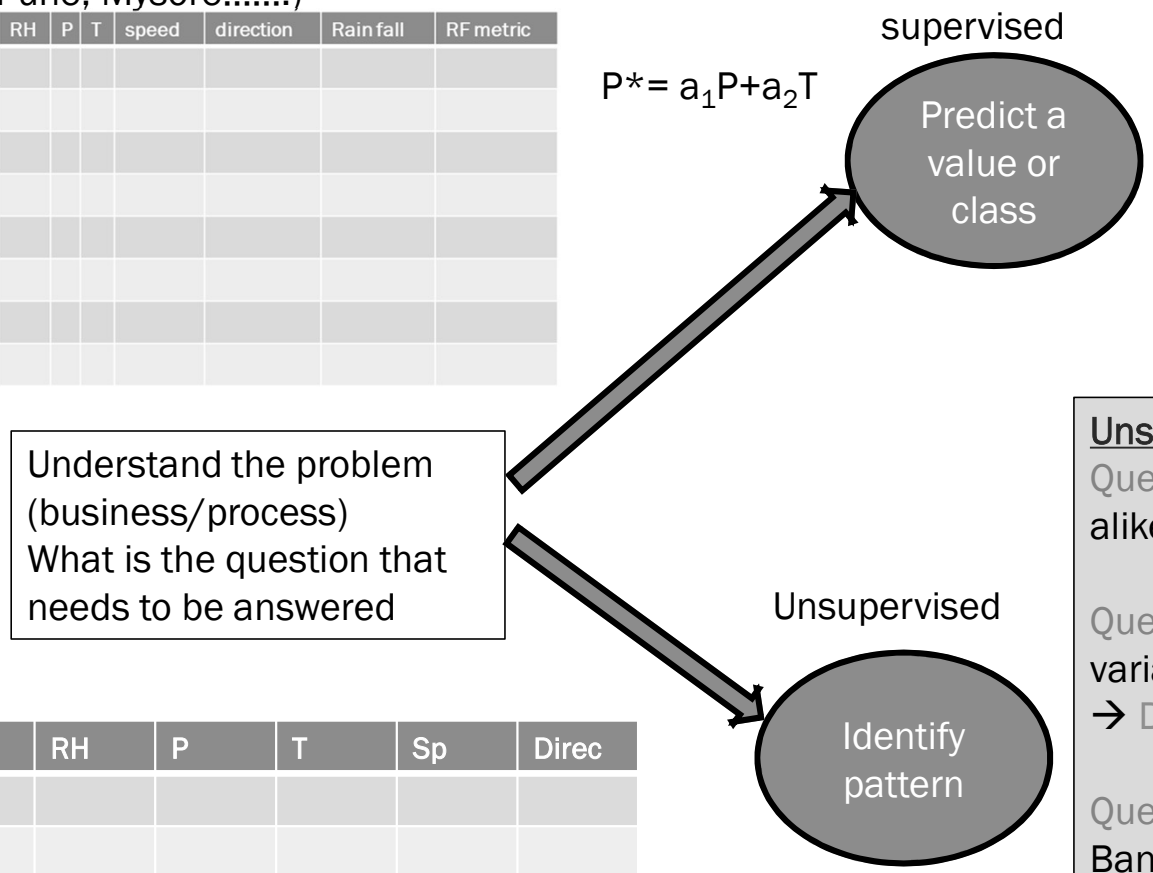
Continuous: can take any value in a given range
Range (0,1) values → 0.1, 0.01345, 0.6, 0.78

Categorical: can take specific values in a given range.
Range (0,1) values → 0, 1
(Yes, no), (pos, neg)

Supervised Vs. Unsupervised

Suppose you are given the variables/ features (Cloud density, Relative Humidity, Pressure, Temperature, wind speed and direction, rain fall, rain fall metric) that indicate the weather in different cities (Bangalore, Hyderabad, Chennai, Pune, Mysore.....)

City	D	RH	P	T	speed	direction	Rain fall	RF metric
Bangalore								
Mumbai								
Delhi								
Hyderabad								
Mysore								
Trivandrum								
Pune								
Noida								



C	D	RH	P	T	Sp	Direc
Bang						
Mumbai						

Supervised

Question: will it rain? → classification (given other variables)

Question: how much will it rain? Regression

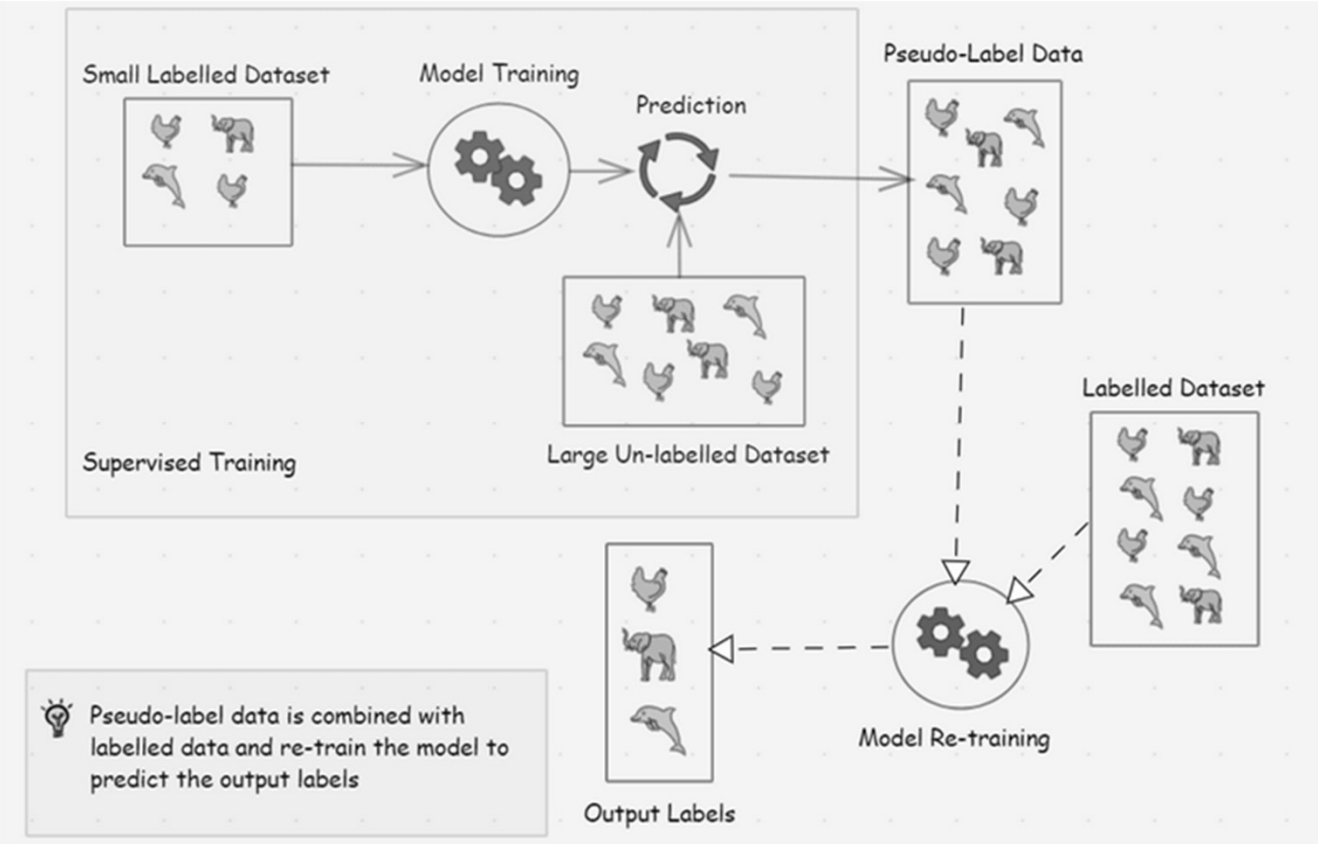
Unsupervised

Question: Which cities are more alike→ clustering

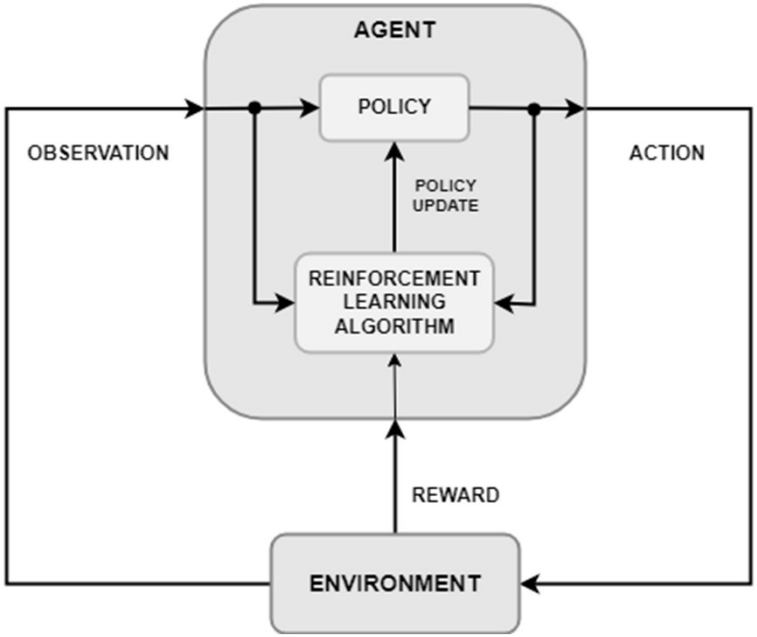
Question: do I need all the six variables to understand the weather → Dimensionality reduction

Question: When the RH is high in Bangalore, is the RH high is Hyderabad too?→ Association

Semi-supervised work-flow



Reinforcement learning



Telecommunication company

- Scenarios:
  - A telecommunication company approaches you and asks you to help them to win their customer loyalty.

Telecommunication company

- Scenarios:
  - A telecommunication company approaches you and asks you to help them to win their customer loyalty.
  - Identify different types of services offered (Data plan, international calls, messaging)
  - Identify the customers using similar services to a large extent (cluster)
  - Sell custom made services to the target group

Real world applications of clustering

- Document clustering (Ex: Googles news: clustering of news articles based on content)

- location clustering (Ex: Uber: clustering based on popular location)

What's clustering

- Group data such that within group variance < between group variance

- Group such that objects in the same group are more similar to each other in some sense than to objects of different groups.

- These groups are known as clusters and each cluster gets distinct label called cluster ID and the centroid of cluster.

- Clustering types:

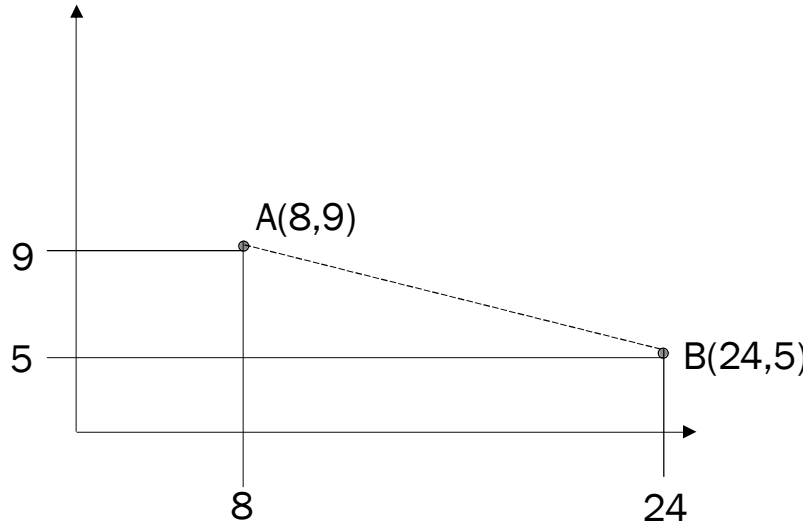
  - **Centroid based(ex: K-Means)**

  - **Hierarchical clustering (ex: Agglomerative)**

  - **Density based (ex: DBSCAN)**

Kmeans: Distance formulations

- In KNN method, we calculate the gap/ distance between the new record and the existing class points
- Decide on the number of neighbors to compare with
- Depending on the class of the dominant closest neighbors, decide on the class of the new record
- Distance/ gap calculation
  - Euclidian(p=2)
  - Manhattan(p=1)



The Minkowski distance of order *p* between two points

$$X = (x_1, x_2, \ldots, x_n) \text{ and } Y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$$

is defined as:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

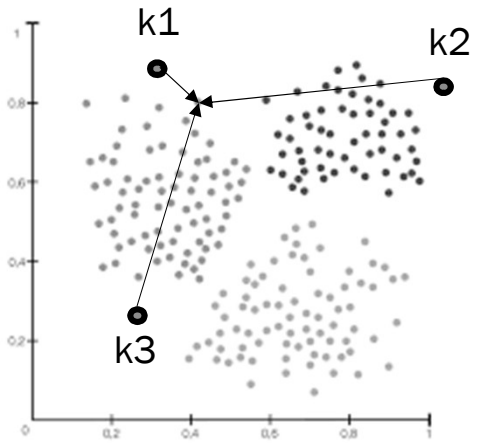
E→ D=sqrt ((24-8)^2+(5-9)^2)

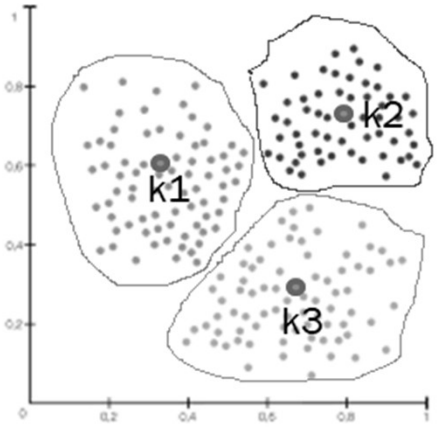
M→ D= (9-5)+(24-8)

C→ D= max (|9-5|,|24-8|)

K-means

- need to specify the number of clusters-K
- Step-1: create arbitrarily ‘k’ number of centroids, assign labels k1,k2,…
- Step-2: calculate the distance of each of the ‘n’ points from each of the ‘k’ centroids
- Step-3: assign each point to nearest centroid (based on distance)
- Step-4: calculate the new centroids based on the records belonging to k*i*
- *Step-5:* check did the cluster index of the points change or iterations less than max: yes→ go to step-2. no→ stop





Different distance metrics used

Data Type	Metric	Formula	Best Use Case
Numerical	Euclidean	$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$	Dense numerical; K-means
Numerical	Manhattan (L1)	$d(p,q) = \sum_{i=1}^{n} \|p_i - q_i\|$	Sparse/outlier-sensitive
Numerical	Minkowski	$d(p,q) = \left(\sum_{i=1}^{n} \|p_i - q_i\|^r\right)^{1/r}$	Tunable (r=1 Manhattan, r=2 Euclidean)
Categorical	Hamming	$d(p,q) = \sum_{i=1}^{n} \mathbb{I}(p_i \neq q_i)$	Count mismatches
Mixed	Gower	$d(p,q) = \frac{1}{n}\sum_{i=1}^{n} \delta_{ij} d_{ij}$	Numerical + categorical

Hamming distance

ID	Color	Size	Shape
1	Red	Small	Circle
2	Red	Small	Square
3	Blue	Small	Circle
4	Blue	Large	Circle
5	Blue	Large	Square

C1→(Red, Small, Circle)
C2→ (Blue, Large, Square)

We choose:

- **k = 2 clusters**
- Distance = **Hamming distance**

Distance Table

ID	Point	$D(C_1)$	$D(C_2)$	Assigned
1	(Red, Small, Circle)	0	3	C_1
2	(Red, Small, Square)	1	2	C_1
3	(Blue, Small, Circle)	1	2	C_1
4	(Blue, Large, Circle)	2	1	C_2
5	(Blue, Large, Square)	3	0	C_2

Gower distance

Age	Color	HasLoan
25	Red	Yes
40	Blue	No

reference

Feature ranges:

- Age: min=20, max=60

$$|25 - 40|/(60 - 20) = 15/40 = 0.375$$

Color

Red ≠ Blue → 1

HasLoan

Yes ≠ No → 1

Gower Distance

$$d = (0.375 + 1 + 1)/3 = 0.792$$

Different centroid based algorithms

Algorithm	Best For	Centroid Type	Distance Metric	Outlier Robustness	Computational Speed
K-means	Pure numerical data	Arithmetic mean	Euclidean (squared)	Low	Fastest
K-medoids	Numerical, noisy data, custom metrics	Actual data point (medoid)	Any (e.g., Manhattan, Gower)	High	Slow ($O(Kn^2)$)
K-modes	Pure categorical data	Mode (most frequent value)	Simple/Hamming matching	Medium	Fast
K-prototypes	Mixed numerical + categorical	Hybrid (mean + mode)	Hybrid (Euclidean + matching)	Medium	Fast

`sklearn_extra.cluster`.KMedoids

class **`sklearn_extra.cluster.KMedoids`**(*n_clusters=8, metric='euclidean', method='alternate', init='heuristic', max_iter=300, random_state=None*) [source]

k-medoids clustering.

Read more in the User Guide.

Parameters:

n_clusters : int, optional, default: 8

The number of clusters to form as well as the number of medoids to generate.

metric : string, or callable, optional, default: 'euclidean'

What distance metric to use. See :func:metrics.pairwise_distances metric can be 'precomputed', the user must then feed the fit method with a precomputed kernel matrix and not the design matrix X.

method : {'alternate', 'pam'}, default: 'alternate'

Which algorithm to use. 'alternate' is faster while 'pam' is more accurate.

init : {'random', 'heuristic', 'k-medoids++', 'build'}, or array-like of shape

K prototypes

```
from kmodes.kprototypes import KPrototypes
import numpy as np

# Example data (replace with your actual data)
# Data should be a numpy array for input
data = np.array([
    [1, 'A', 2, 'X'],
    [1, 'B', 2, 'Y'],
    [2, 'B', 3, 'X'],
    [2, 'A', 3, 'Y'],
    [3, 'A', 4, 'X'],
    [3, 'B', 4, 'Y'],
])

# Specify the indices of the categorical columns
# In this example, columns at indices 1 and 3 are categorical
categorical_indices = [1, 3]

# Create a KPrototypes instance
# n_clusters is the number of clusters to form
kproto = KPrototypes(n_clusters=2, init='Cao', verbose=0)
```

k-prototypes minimizes a **joint distortion function**:

$$J = \sum_{i=1}^{n} \sum_{l=1}^{k} z_{il} \left[\underbrace{\|\mathbf{x}_i^{(num)} - \boldsymbol{\mu}_l\|^2}_{\text{numeric (k-means)}} + \gamma \underbrace{\sum_{j=1}^{q} \delta(x_{ij}^{(cat)}, m_{lj})}_{\text{categorical (k-modes)}} \right]$$

Where:

- $z_{il} \in \{0, 1\}$: assignment indicator
- $\boldsymbol{\mu}_l$: numeric centroid of cluster l
- m_{lj}: mode of categorical feature j in cluster l
- $\delta(a, b) = 0$ if $a = b$, else 1
- γ: **trade-off parameter**

K modes initialization

Huang Initialization → combinations created based on frequency

Cao→ Density based → should be the default choice

Problem with K-means

- Convex shape
- Isotropic variance
- outliers