DATA PREPROCESSING

Data mining

* Data mining is predecessor to data science.
* The most popular methodology used is CRoss Industry Standard Process for Data Mining (CRISP-DM).

CRISP DM has the following phases:

→ Business understanding: Business person (domain experts) formulate

08/02/2025

a business problem.

→ Data understanding: Technical analyst gets involved w/ domain experts to understand complicated data. Also called exploration phase.

→ Data preparation: Creation of training & test data sets. Also called pre-processing phase.

→ Modelling: Formulate a model to fit data and is usually associated w/ ML.

→ Evaluation: Evaluate the model & data to check if the business problem is solved.

→ Deployment: Setting up the system in a production environment.

* There is no one way of cleaning data, often involves manual work.
* Data can be of various types → grid of numbers, audio, text etc.
* For the purpose of visualising data, it is convenient to have numerical features.

FEATURE ENGINEERING

More art than science. Features can be of following types:
* Distributions: normal, binomial, poisson etc.
* Binaries: yes/no, +/-, true/false etc.
* Categorical: instance of continents, high/medium/low etc.
* Quantitative: temperature in degree, price in dollars etc.

It is the process of creating or improving features.

Features are created based on common sense, domain knowledge and prior experience.

Data can have missing values for some features. In such cases, the following happens:
→ the missing values are ignored.
→ the missing values are imputed w/ fixed values (can be arithmetic mean, median or mode).

Humans work w/ various types of values but ML algorithms

12/02/2025

work best w/ numerical values. Hence label encoding is done to convert labels into * numeric form.

One-hot-encoding or one-of-K

Refers to splitting the column which contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains '0' or '1' corresponding to which column it has been placed.

Example: Original data:

Fruit	Categorical value of fruit	Price
Apple	1	5
Mango	2	10
Apple	1	15
Orange	3	20

After one-hot encoding:

Apple	Mango	Orange	Price
1	0	0	5
0	1	0	10
1	0	0	15
0	0	1	20

19/02/2025

DATA SCIENCE

It is an interdisciplinary academic field that uses statistics, scientific computing, methods, processing & visualisations, algorithms and systems to extract or extrapolate knowledge from potentially noisy, structured or unstructured data.

Raw data		Data Science Process		Business insights
* Sales data	→		→	* Sales prediction
* Customer feedback				* Product optimisation
* Web logs				* User behaviour

DATA MINING

* Process of discovering valuable insights, patterns & information various data sets by employing from vast datasets by employing various techniques, algos and tools.
* Technology that blends traditional data analysis methods w/ sophisticated algos for processing large volumes of data.

Statistics: sampling, estimation, hypothesis testing

Machine Learning: search algo, modelling techniques, learning theories from AI, pattern recognition

Types of patterns:

(i) Association: Coffee buyers usually also buy sugar.

(ii) Clustering: Segments of customers requiring different promotional strategies

(iii) Classification: Determining if a bank customer who is applying for a loan will be a defaulter.

Association

Identifies relationships between items in a data set.

19/02/2025

Example: 1. Bread Butter Milk
2. Bread Butter Salt
3. Bread Butter Milk We can cluster 1 & 3 together.
4. Bread Butter Sugar

Bread $\rightarrow$ Butter 100%
(Bread, Butter) $\rightarrow$ Milk 50%
(Bread, Butter) $\rightarrow$ Salt 25%
(Bread, Butter) $\rightarrow$ Sugar 25%

~~Rules~~ Association rules:
~~These~~ There's always an antecedent & a consequent in an association rule.
No. of association rules possible $= \sum_{i=1}^{n} {}^{n}C_i \, (2^{n-i} - 1)$

AR: $X \rightarrow Y$

$\{x_1, x_2, \ldots, x_n\} \rightarrow \{y_1, y_2, \ldots, y_m\}$

* Itemset(AR) = list of all items in the antecedent & consequent $= X \cup Y$
* Support (AR) $= P(X \cap Y)$
$$= \frac{\#\text{ transactions containing } X \cap Y}{\#\text{ transactions in database}}$$
* Confidence (AR) $= P(Y \mid X)$
$$= \frac{\#\text{ transactions containing } X \cap Y}{\#\text{ transactions in } X}$$
* Lift (AR) $= \frac{P(Y \mid X)}{P(Y)}$
$$= \frac{\frac{\#\text{ transactions containing } X \cap Y}{\#\text{ transactions in } X}}{\frac{\#\text{ transactions in } Y}{\#\text{ transactions in database}}}$$

If Lift < 1, then you can ignore it.

19/02/2025

Applications: Market basket analysis, recommendation system, fraud detection, healthcare & medical

Example:

Transaction ID	Items
1	tomato, potato, onion
2	tomato, potato, brinjal, pumpkin
3	tomato, potato, onion, chilli
4	tamarind, lemons.

AR: (tomato, potato) → onion
⋮
etc.

Itemset = {tomato, potato, onion, brinjal, pumpkin, chilli, tamarind, lemons}

Apriori algorithm
Invented by Rakesh Agrawal & Ramakant Srikanth (1994)
Apriori: acknowledges the prior knowledge
→ If any itemset is not frequent, then its superset cannot be frequent
→ An itemset is frequent only if all its subsets are frequent.

Step 0: Create 1-size frequent itemsets list that meet threshold support k=1.
Step 1: Expand the itemsets list by combining overlapping sets from k-sized itemsets to k+1-sized itemsets list.
Step 2: Prune the expanded itemsets list using apriori property, k=k+1
Step 3: Remove infrequent item sets from the list!
Repeat steps 1, 2, 3 till no ~~more~~ further expansion is possible

Example:

1	1,2,3,4	5	2,3
2	1,2,4	6	3,4
3	1,2	7	2,4
4	2,3,4		

k=1   {1}   {2}   {3}   {4}
      3   6   4   5

k=2   {1,2}   ~~{1,3}~~   ~~{1,4}~~   {2,3}   {2,4}   {3,4}
      3   1   2   3   4   3

k=3   ~~{1,2,3}~~   ~~{1,2,4}~~   ~~{1,3,4}~~   ~~{2,3,4}~~
      1   2   1   2

k=4   ~~{1,2,3,4}~~
      1

Frequent itemsets : {1}, {2}, {3}, {4}, {1,2}, {2,3}, {2,4}, {3,4}

Pseudocode :

```
Apriori (T, ε) {
    C = All '1'-sized itemsets
    k = 1
    while (|C| > 0) {
        find frequency of all itemsets in C
        L = sets in C w/ frequency/count ≥ ε

        C = Apriori Generate (L, k)
        k++
    }
}

Apriori Generate (L, k) {
    ∀ x, y ∈ L
        if |X ∩ Y| = k-1 :
            Z = X ∪ Y
        if any (k-1)-sized subset of Z not in L :
            do not add Z to L
        else :
            add Z to L
    return L
}
```

22/02/2025

* Different types of association rules (categorical, hierarchical, cyclic).
* Eclat algorithm
  Equivalent Class Transformation: a DFS strategy
* FP growth
  Frequent Pattern growth: a compact data structure called the frequent Pattern tree is used to compress the dataset.

Classification

We collect different requirements / facts / data about certain features

$X = (x_1, x_2, \ldots, x_d) \subseteq \mathbb{R}^d$ $\longrightarrow$ data set

$Y = \{1, 2, \ldots, K\} = [K]$ $\longrightarrow$ class set (label space)

$K=1 \longrightarrow$ no need for classification

$K=2 \longrightarrow$ binary classification

$K>2 \longrightarrow$ multi-class

What is a classifier?

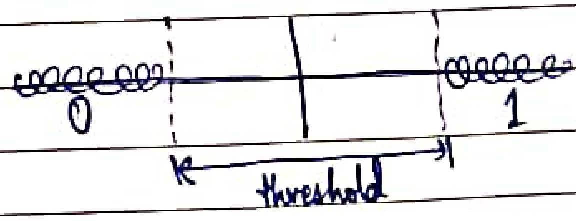
x_{new} $\quad$ y_{new} $\quad$ $\hat{y}_{new}$

$y_{new} \downarrow$ True label $\quad$ $\hat{y}_{new} \downarrow$ Predicted label

Classifier is a function

$$f: X \longrightarrow Y$$

Example: $f(x) = 1$; $x \geq$ threshold
$ 0$; otherwise



Goodness of a classifier: $P(\hat{y}_{new} = y_{new})$

$$\text{Accuracy} = \frac{\#\, x_i \text{ correctly classified}}{L}$$

22/02/2025

$y_i \backslash \hat{y}_i$	0	1
0	TN	FP
1	FN	TP

$P = FN + TP$

$N = TN + FP$

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

$$\text{Recall (also called sensitivity)} = \frac{TP}{P} = \frac{TP}{FN + TP}$$

→ also called True Positive Rate (TPR)

$$\text{Precision} = \frac{TP}{TP + FP} \longrightarrow \text{Positive Predictive Value (PPV)}$$

$FNR = \frac{FN}{P}$ $\quad TNR = \frac{TN}{N}$ $\quad FPR = \frac{FP}{N}$

$TNR + FPR = 1$ $\quad FNR + TPR = 1$

F_1 score = harmonic mean of recall & precision

$$= \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = \frac{2}{\frac{FN+TP}{TP} + \frac{FP+TP}{TP}} = \frac{2TP}{2TP + FN + FP}$$

Applications :

* Text application
  - → Classify emails into spam/non-spam
  - → NLP problems

    Tagging : classifying words into verbs, nouns etc.

* Risk management, fraud detection, computer intrusion detection
  - → given the properties of a transaction (items purchased amt, location, customer profile etc.)
  - → determine if it is a fraud.

* Machine learning / pattern recognition applications
  - → Vision, speech recognition etc.

* All of science & knowledge is abt predicting future in terms of past
  - → So classification is a very fundamental problem w/ ultra-wide scope of applications

~~NP~~ collect

Bayes Classifier

Suppose $Y \in \{1, 2, \dots, K\}$

$\mathbb{P}(Y=k) \rightarrow$ Probability that one observes a data point from class k (Prior)

$\mathbb{P}(Y=k \mid X=x) \rightarrow$ Conditional prob (the label of the data point) $= k$ (Posterior)

$\mathbb{P}(X=x \mid Y=k) \rightarrow$ Prob dist on X given that the data point is in class k (Likelihood)

$k \in \operatorname{argmax}_{j \in [K]} \mathbb{P}(Y=j \mid X=x)$

$\hookrightarrow$ one can argue it is optimal classifier

$\hookrightarrow$ Bayes error rate is minimal

$$\mathbb{P}(Y=k \mid X=x) = \frac{\mathbb{P}(Y=k) \times \mathbb{P}(X=x \mid Y=k)}{\mathbb{P}(X=x)}$$

$$\Rightarrow \text{Posterior} = \frac{\text{Prior} \times \text{Likelihood}}{\text{Evidence}}$$

ML - model $\rightarrow$ d-dimensional

$\mathbb{P}(X = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_d) \mid Y=k)$

$\cancel{= \mathbb{P}(X_1 = \vec{x}_1) \cdot \mathbb{P}(X_2}$

$= \mathbb{P}(X_1 = \vec{x}_1 \mid Y=k) \cdot \mathbb{P}(X_2 = \vec{x}_2 \mid Y=k) \dots \mathbb{P}(X_d = \vec{x}_d \mid Y=k)$

$= \prod_{i=1}^{d} \mathbb{P}(X_i = x_i \mid Y=k)$

$$k^* \in \operatorname*{argmax}_{k \in [K]} \mathbb{P}(Y=k) \times \prod_{i=1}^{d} \mathbb{P}(X_i = x_i \mid Y=k)$$

Decision Tree

Eg 1 Recall cricket ball v/s tennis ball example

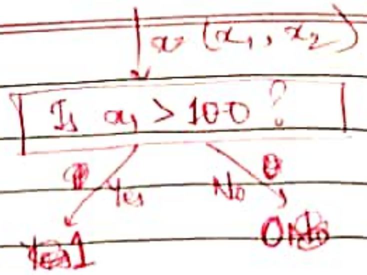
$x = (x_1, x_2) \in \mathbb{R}^2$

$y = \{0, 1\}$ 1 is 'cricket', 0 is 'tennis'

If weight of a ball is 100 g, ball is cricket ball else tennis ball

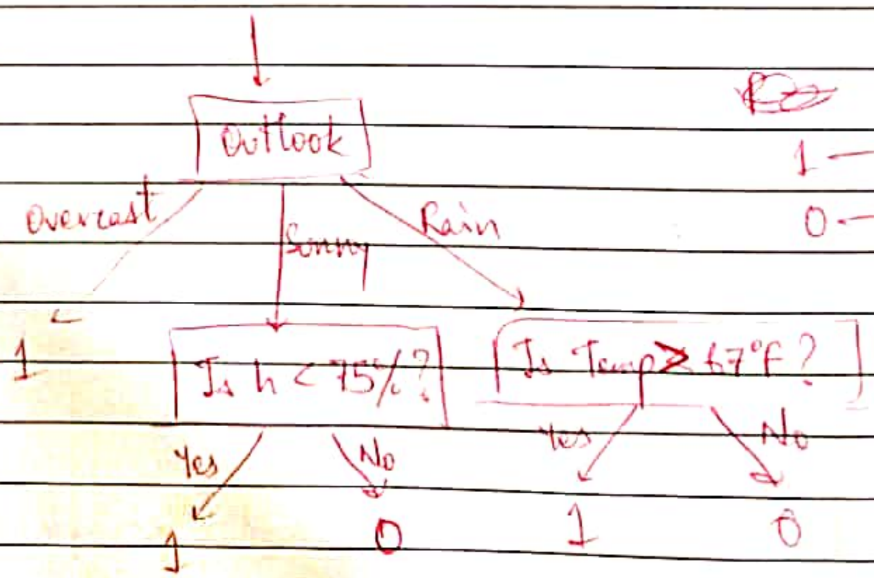
Date
Page

$f(x) = 1$ if $x_1 > 100$
$= 0$; ~~oth~~ o/w



g2

Outlook	Temp (°F)	Humidity (%)	Windy?	Class
→ sunny	75	70	T	play
→ sunny	80	90	T	don't play
"	85	85	F	DP
"	72	95	F	DP
"	69	70	F	P
overcast	72	90	T	P
"	83	78	F	P
"	64	65	T	P
"	81	75	F	P
rain	71	80	T	DP
"	65	70	T	DP
"	75	80	F	P
"	68	80	F	P
"	70	96	F	P



1 → Play
0 → Don't play

Date
Page

ID3
* Iterative Dichotomizer Ross Quinlan in 1980s
* Feature that should be used to make decision:
→ The one which will seperate all classes well
→ Information theoretically, the one from which we have highest info gain
→ How to measure Info Gain?

Entropy
* I need to send you one of the two obs:
→ It is raining today → It is not raining today
Only one bit is enough to send you a message
0 → raining today 1 → not raining today

Similarly if there are 4 messages : 2 bits.

Every system of discrete symbols has some 'info' and we need those many bits to represent the system.

$$\text{Entropy} = \sum_i P_i \cdot \log\left(\frac{1}{P_i}\right) = -\sum_i P_i \log P_i$$

$$\text{Entropy}, H(S) = \sum_{i \in [n]} \frac{S_i}{S} \times \log \frac{S}{S_i}$$

Info Gain due to A, $IG(S,A) = H(S) - H(S|A)$

$$= H(S) - \sum_t \frac{S_t}{S} H(S_t)$$

where $t \in T$ is the diff values present in S

Eg: https://www.kaggle.com/datasets/tareqjoy/trainplaytennis

$|S| = 14$ $|S_{yes}| = 9$ $|S_{No}| = 5$

$$H(S) = \frac{9}{14} \log_2 \frac{14}{9} + \frac{5}{14} \log_2 \frac{14}{5} = 0.940$$

$A = \text{Outlook} \qquad T = \{S, O, R\}$

$t = \text{Sunny}$:

$|S_t| = 5 \qquad |S_t\{\text{Yes}\}| = 2 \qquad |S_t\{\text{No}\}| = 3$

$H(S_t) = \frac{2}{5}\log\frac{5}{2} + \frac{3}{5}\log\frac{5}{3} = 0.971$

$t = \text{Rain}$

$|S_t| = 5 \qquad |S_t\{\text{Yes}\}| = 3 \qquad |S_t\{\text{No}\}| = 2$

$H(S_t) = 0.971$

$t = \text{Overcast}$

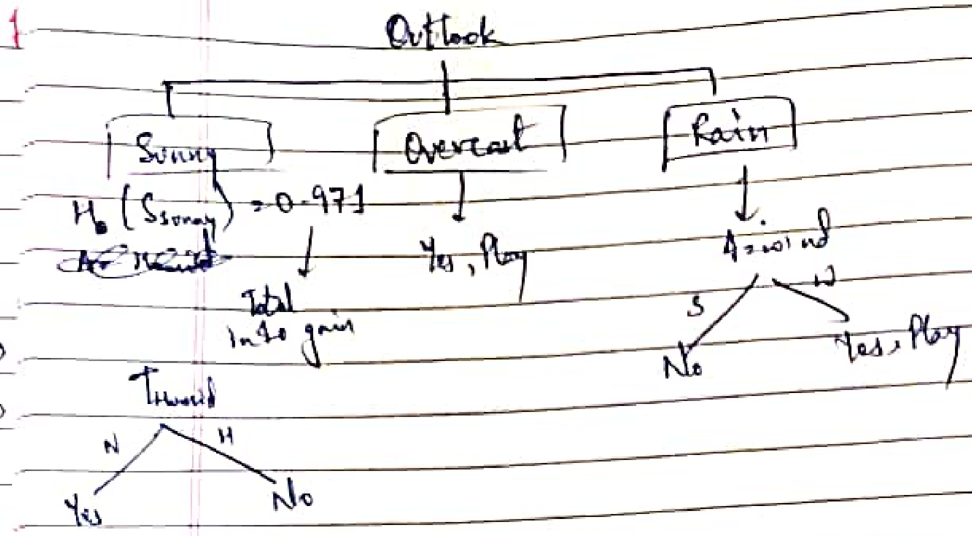
$|S_t| = 4 \qquad |S_t\{\text{Yes}\}| = 4 \qquad |S_t\{\text{No}\}| = 0$

$H(S_t) = \frac{4}{4}\log\frac{4}{4} \cancel{+ \frac{0}{4}\log} = 0$

$$H(S|A) = \frac{5}{14} \times 0.971 + \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0$$

$$= \frac{5}{7} \times 0.971$$

$$= 0.693$$



CART

Classification & Regression Tree

Key idea:

* Only 2 children
* Some goodness criterion to split (typically minimise Gini Index index based)

$$\sum p_t (1 - p_t)$$

* Pruning to avoid overfitting (typically info gain)

Decision trees ⟷ good for explaining decisions

↳ Very ~~sen~~ sensitive to noise in the data, small change in the data can lead to a very diff decision tree

Bayes v/s	Naive Bayes v/s	Decision tree
* Optimal ↳ ultimate goal	* Easy to build	* Easy to build
* practically difficult to learn Bayes classifier directly from the data	* Not all features ind	* Explainability is good
* Curse of dimensionality	* explainability is poor	* Overfitting
		* Sensitivity

KNN

k ~~near~~ nearest neighbours

Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

Given a point x, find out k nearest point from D to x

Eg: say $|x_{\sigma(1)} - x| \leq |x_{\sigma(2)} - x| \leq \ldots \leq |x_{\sigma(n)} - x|$

Collect labels of $x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)}$

$\hat{y}$ = most frequently occuring label in $x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(k)}$

(Copy from Saloni)

Running time, linear in $n \times d$.

As $n \to \infty$, 1-NN error rate is at most 2× Bayes error rate

How to measure nearness?

1. If features are conti → Euclidian metric
2. Discrete features → Hamming distance

$d : X \times X \to \mathbb{R}_{\geq 0}$

1. $d(x, y) \geq 0 \quad \forall x, y \in X$
2. $d(x, x) = 0$
3. $d(x, y) = d(y, x) \quad \forall x, y \in X$
4. $d(x, y) + d(y, z) \geq d(x, z)$

Drawback

→ Frequent classes dominate

Combining Classifiers

We get k classifiers

Bagging: Take a majority vote for the best class for each new record

Boosting: Each classifier's vote has a weight proportional to its accuracy on training data

like a patient taking multiple opinions from several doctors

CLUSTERING

Applications

* Targetting similar ppl or objects
  - → Student tutorial grps
  - → Hobby grps
  - → Health support grps
  - → Customer grps for marketing
  - → Organising e-mail
* Spatial clustering
  - → exam centres
  - → locations for a business chain
  - → planning a political strategy

* Two types of algo:
  - → Hard clustering: each point is assigned to some cluster
  - → Soft clustering: each point is assigned probabilities of being assigned to each cluster

Hierarchical clustering → type of Hard clustering

(I) Connectivity-based clustering

1. Agglomerative (eg: AGNES)

Step1 Start: Each point in seperate cluster

Step2 Merge 2 closest clusters

Step3 Repeat until all records are in a single cluster

* Computationally expensive
* Better at handling outliers

2. Divisive (eg: DIANA)

Step1 Start: All points in 1 cluster

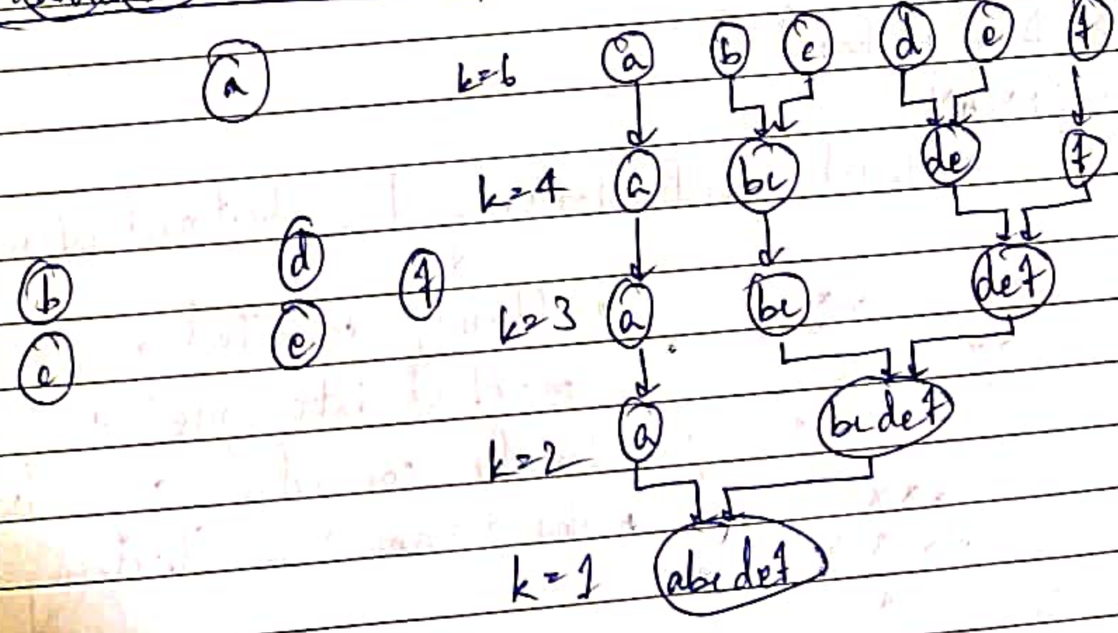
Step2 Find most extreme points in each cluster

Step3 Regroup points based on closest extreme point

Step4 Repeat until each record is in its own cluster

* Outliers may disrupt the cluster.

Measurement of similarity



To find k-clusters,

optimisation problem is :

$$\min \sum_{i=1}^{k} \sum_{x_j \in C_i} \underbrace{\|x_j - \mu_i\|^2}_{\text{Variance } C_i} \qquad \mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

① fix k, general d] NP-Hard d-features
② general k, fix d] also NP-hard
③ $O(n^{dk+1})$

(II) Centroid-based clustering
① Randomly sample k points
$\mu_1, \dots, \mu_k$ $C_i = \{\mu_i\}$
② $\forall x_j \in D$
$i^* = \arg\min \|x_j - \mu_i\|$
$C_{i^*} = C_{i^*} \cup \{x_j\}$
③ for $i \to k$
$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_j} x_j$
④ Continue to step ② till $\exists \, |\mu_i - \mu_i^{prev}| \neq 0$

1. k-means
Step 1 Randomly select k points as centers
Step 2 Assign each point to the cluster closest to these centers
Step 3
Step 4

(III) Density-based
2. DBSCAN
Density-based spatial clustering of applications w/ noise

* Closely connected points are marked into one cluster
* Loosely connected are called noise
* Non-parametric Clustering Algo

Optimisation
$C = \{c_1, c_2, \dots, c_k\}$ be the clusters
$\min k \quad std(p,q) \leq \epsilon \;\forall p,q \in c_j \;\forall c_j$

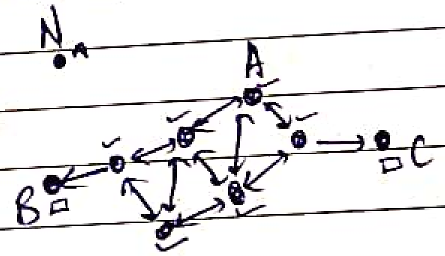
* A point p is a core point if at least minPts points w/ ϵ distance
* A point is directly reachable from a core point if it is within ϵ distance from a core point
  → we talk abt reachability only from core point
* A point q is reachable from p if there is a path $\{p, p_1, p_2 \cdots p_n\}$ such that p_{i+1} is directly reachable from p_i

Step1 Find the points in the ε(eps) neighbourhood of every point and identify the core points w/ more than minPts neighbours.

Step2. Find the connected components of core points on the neighbour graph ignoring all non-core points.

Step3 Assign each non-core point to a nearby neighbour otherwise assign it to noise.

Eg: minPts 4
A & other points are in the core
B & C are not in the core



Which k? Which algo?
* Project on 2D/3D & determine if they are visually nicely seperated
* Silhouette score
  $S(i)$ = Based on avg dist w/ pts in the same cluster $a(i)$ v/s smallest dist to the points not in the cluster $b(i)$

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Avg $S(i) \in [-1, 1]$ → Higher the better

Page

* Davies - Bouldin Index

Captures compactness of a cluster & its seperation across clusters

$$DB = \frac{1}{k} \sum_i \max_j \frac{\Delta X_i + \Delta X_j}{\delta(X_i, X_j)}$$

$\Delta \longrightarrow$ Intra cluster dist

$\delta \longrightarrow$ Intercluster dist