Classification Algorithm in various situations / cases

✧ learning how to apply the techniques to real world data.

(KNN , logistic regression, Decision Tree etc)

Imbalanced vs balanced data

2 class - classification problem:-

D_n → n_1 +ve pts

D_n → n_2 -ve pts

$n_1 + n_2 = n$

1) if $n_1 \approx n_2$

58% ← $n_1 \simeq 580$ $n_1 \pm n_2$

42% ← $n_2 = 420$

If $n_1 \approx n_2$ than it is called balanced data set

2) if $n_1 << n_2$ (or) $n_2 << n_1$ → Imbalanced dataset

$n_1 = 100$

$n_2 = 900$

or

$n_2 = 150$

$n_1 = 850$

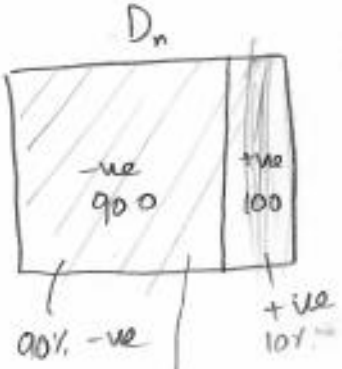
K-NN (imbalanced dataset)

$n_1 = 50$ (+ve) ; $n_2 = 950$ (-ve) (not always)

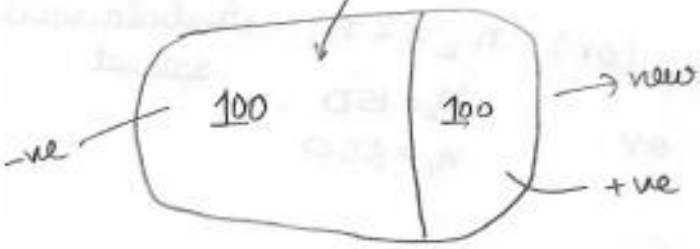
↑ majority class
Dominating class
could have some advantage

? How to work around imbalanced dataset problem / issue?

① Undersampling : ⓓ $n_1 = 100$ (+ve)
$n_2 = 900$ (-ve)



use sampling trick to create a new dataset D_n'

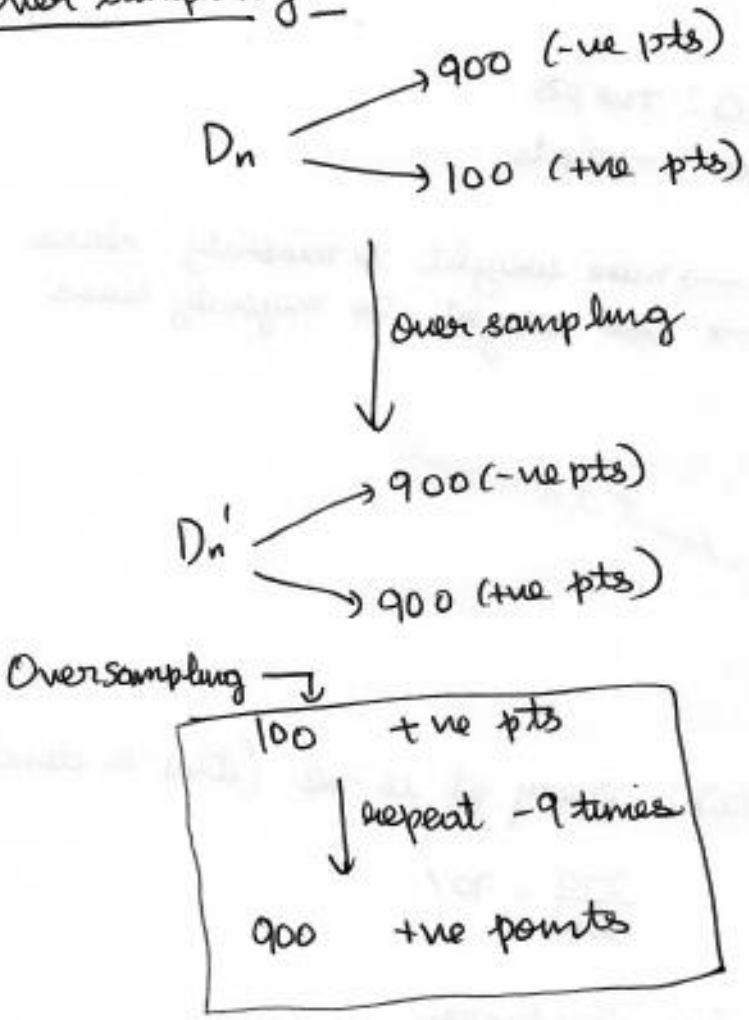


$|D_n'| < |D_n|$

model have smaller amount of data
it will not work accurately

? Problem with undersampling
Throwing away data is not good

② Over sampling:-



⊘ — +ve pts
× — -ve pts
• → oversampled (repeatation) +ve pts

☆ Artificial / Synthetic pts → (Extrapolation)
add +ve pts in particular neighbours

☆ Class-weight :- 100 : +ve pts
900 : -ve pts

$w_+ = 9$ → more weight to minority class
$w_- = 1$ → less weight to majority class

High accuracy $f(x)$: every pt is -ve (that is dumb model)

$$\frac{270}{300} = 90\%.$$

Multi class classification

Binary classifier : $y_i \in \{0, 1\}$
Multi class classifier : $y_i \in \{0, 1, 2, 3, \cdots, 9\}$ → MNIST.
(10 classes)

K-NN

$$D_n = \{(x_i, y_i) \mid x_i \in \mathbb{R}^d,\ y_i \in \{1, 2, 3, \ldots, C_n\}\}$$

7-NN

x_q : | 0 | 6 | 1 | --- | 0 |
1 2 3 --- C

Majority vote y_q = class 2

$y_q \neq 2$

probabilistic classifier

$$\begin{cases} P(y_q = 2) = \frac{6}{7} \\ P(y_q = 3) = \frac{1}{7} \\ P(y_q = 1) = 0 = P(y_q = 4) = \cdots\cdots P(y_q = C) \end{cases}$$

NOTE : logistic Regression → They are fundamentally binary classifier & cannot do multiclass - classifications easily as KNN

Q Given a multi class classification problem; can we convert it into a binary classifier prob

$f(x) \longrightarrow 0, 1$ } binary

$f(x) \longrightarrow 0, 1, 2, \ldots, C_n$

$y_i \in \{1, 2, 3, \ldots, C\} \rightarrow C$ class $\rightarrow$ {C binary classifier}

D_n

① $D_n \rightarrow \{(x_i, y_i) \mid y_i = 1\} \rightarrow$ +ve pts

$D_n \rightarrow \{(x_i, y_i) \mid y_i \neq 1\} \rightarrow$ -ve pts

(Binary classifier)$_{1 \text{ or not}}$

② Again D_n

$D_n \rightarrow y_i = 2$, $y_i \neq 2$ $\} \rightarrow$ using this (Binary classifier)$_{2 \text{ or not}}$

③ $D_n \rightarrow y_i = 3$, $y_i \neq 3$ $\}$ (Binary classifier)$_{3 \text{ or not}}$

Multi - class classification problem.

↓

C binary classification problem

↓

$f_1(x) \rightarrow$ class 1 or not

$f_2(x) \rightarrow$ class 2 or not

⋮

$f_C(x) \rightarrow$ class C or not

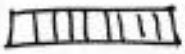
✧ KNN, given a distance measure

Classification :- $x_i \in \mathbb{R}^d$

↳ vector is not easy

$y_i \leftarrow x_i$ — Chemical compound in Pharma → numeric vector may not be easy.

y_i → useful for treatment

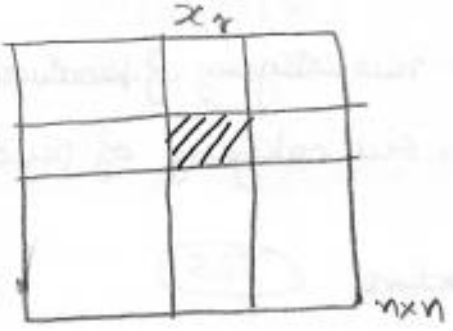
x_i → Paracetamol → 

↓

it is difficult to write chemical formula cannot converted into numerical vector

$sim(x_i, x_j)$ → So similarity (distance is calculated)

$sim(x_i, x_j)$ = n datapoints

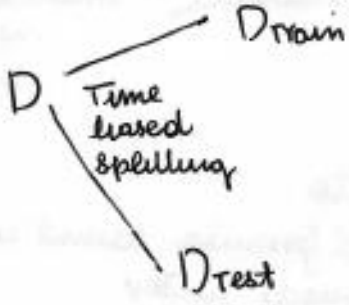


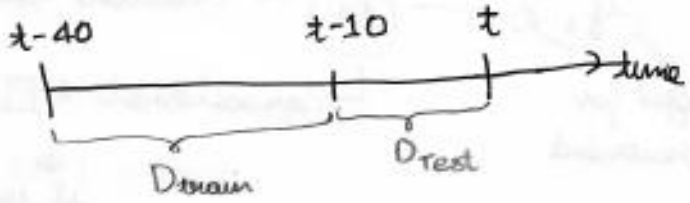
$S_{ij} = sim(x_i, x_j)$

Distance : $d_{ij} = \frac{1}{S_{ij}}$

✧ Instead of each x_i as paracetamol we are given similarity matrix or distance matrix

Train and test set differences





✧ Amazon Food reviews

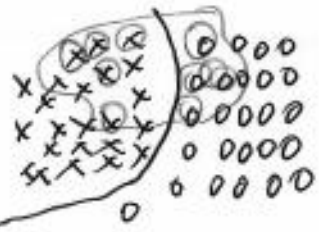
D_{Train} & D_{test} could be very different

Food reviews

- t-10 to t : - new category of products (wine)
- t-40 to t-10 - old category of products

→ data changes overtime (TBS)

X: Train dataset −ve
+ve
O: Train.
Ⓧ: Test −ve
Ⓞ: Test +ve



KNN decision boundary using Dtrain

D_{train} and D_{test} are fundamentally different

distribution of -ve data has changed from D_{train} to D_{test}.

(1) On D_{Train} we performing well but for D_{test} error is high.

We need to check that D_{Train} & D_{test} cannot do well.

Q how to determine if data is changing over time?

or

D_{train} & D_{test} have different distribution?

Sol

D_n → TBS → $D_{train}(x_i, y_i)$ → true or -ve → (x_i', y_i')

D_n → TBS → $D_{Test}(x_i, y_i)$ → two or -ve → (x_i', y_i')

To solve problem we create new dataset

$D_n' \doteq$ D_{Train} $\doteq$ $y_i' = 1$; $x_i' = (x_i, y_i)$ in D_{Train}.

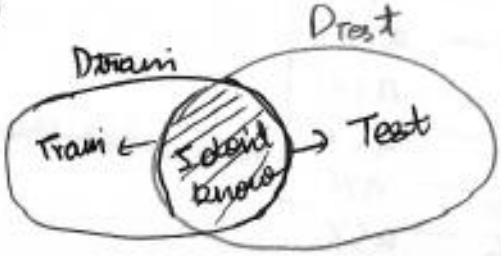
D_{test} $\doteq$ $y_i' = 0$; $x_i' = \text{concat}(x_i, y_i)$ in D_{test}

Build a binary classifier on D_n'

$f(x) \longrightarrow +1$ or 0

KNN

Case I

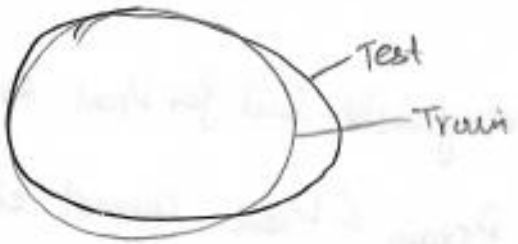


Binary classifier has accuracy of 70%.

↓

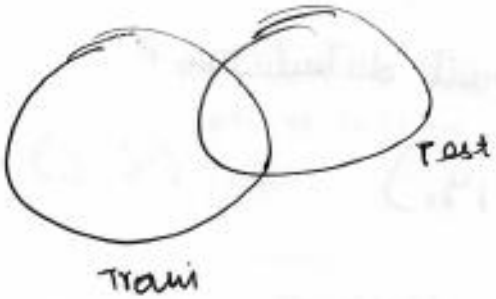
Train & Test data can be separated partially distributions are different

Case II



→ almost overlapping
→ binary classification (KNN)
↳ accuracy is low
→ distributions are very similar

Case III :



→ overlap v. low
→ acc. is high
→ disb are very different

D_{train} & D_{test} ⟶ same distribution
↳ not from same distribution
↳ This means features are changing with time

Impact of outliers

For KNN, when K is small outliers can easily impact your model

10 fold CV

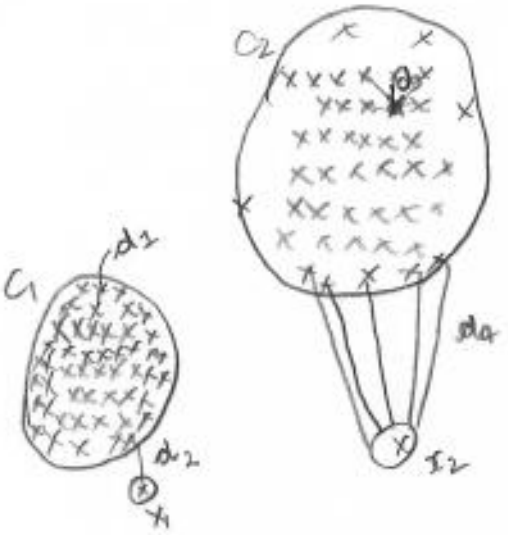
- K=1 — 97%
- K=2 — 97%
- K=3 — 97%
- K=4 — 97%
- K=5 — 97%

K=5 is less prone to outliers than k=1

- K=6 — 95%
- K=7 — 92%

Local outlier Factor (LOF)

Detect outliers in data inspired by K-NN.



→ C_1: very dense cluster
C_2: sparser cluster.

both x_1 and x_2 are outliers

→ simple solution
x_i : K-nearest neighbours
mean distance from x_i to its K-nearest neighbours

for 5 nearest neighbour.

d_1, d_2, d_3, d_4, d_5 are average distance from point

d_4 is the longest value
mean distance from outlier is large consider it as an outlier