01-11-2025

Agenda:

→ Naive Bayes Algo (2 methods)

→ k-means clustering

Naive Bayes Algorithm

Bayes theorem : $P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$

↓
probability

what if?

→ Data is categorical?

→ we want probabilistic reasoning

→ we want something that learns from frequency count.

→ we want fast & interpretable solution.

→ NB → classification → probabilistic model

Dataset: (play Tennis)

prob of Rainy & Hot ?

Outlook	Temperature	play
Sunny	Hot	No
Sunny	Mild	No
Cloudy	Hot	Yes
Rainy	Mild	Yes
Rainy	Cool	Yes
Cloudy	Cool	Yes

$P(Yes) = \frac{4}{6} \rightarrow 0.66$

$P(No) = \frac{2}{6} \rightarrow 0.33$

$P(Yes) \rightarrow 0.66$

outlook	O_calc	Temp	T_calc
Cloudy	2/4 → 0.5	Hot	1/4 → 0.25
Rainy	2/4 → 0.5	Mild	1/4 → 0.25
Sunny	0/4 → 0	Cool	2/4 → 0.5

$P(No) \rightarrow 0.33$

Cloudy	0/2 → 0	Hot	1/2 → 0.5
Rainy	0/2 → 0	Mild	1/2 → 0.5
Sunny	2/2 → 1	Cool	0/2 → 0

→ prob of Rainy & Hot

$$P(Yes \mid Rain, Hot) \propto P(Yes) \times P(Rain \mid Yes) \times P(Hot \mid Yes)$$

$$= 0.66 \times 0.5 \times 0.25$$

$$= 0.0825$$

$$P(\text{No} \mid \text{Rain}, \text{Hot}) \propto P(\text{No}) \times P(\text{Rain} \mid \text{No}) \times P(\text{Hot} \mid \text{No})$$

$$= 0.33 \times 0 \times 0.5$$

$$= 0$$

$$P(\text{Yes} \mid \text{Rain}, \text{Hot}) > P(\text{No} \mid \text{Rain}, \text{Hot}) \Rightarrow \text{Yes}$$

→ prob of Sunny & Cool

$$P(\text{Yes} \mid \text{Sunny}, \text{Cool}) \propto P(\text{Yes}) \times P(\text{Sunny} \mid \text{Yes}) \times P(\text{Cool} \mid \text{Yes})$$

$$= 0.66 \times 0 \times 0.5$$

$$= 0$$

$$P(\text{No} \mid \text{Sunny}, \text{Cool}) \propto P(\text{No}) \times P(\text{Sunny} \mid \text{No}) \times P(\text{Cool} \mid \text{No})$$

$$= 0.33 \times 1 \times 0$$

$$= 0$$

$P(\text{Yes} \mid \text{Sunny}, \text{Cool})$ & $P(\text{No} \mid \text{Sunny}, \text{Cool})$

Both are 0

↓

To solve this we need smoothing

we add 1 to every count & divide by # samples + # categories

Outlook	O	-Calc	Temp	T-Calc
Cloudy	2/4 →	2+1/4+3 → 0.429	HOT	1/4 → 1+1/4+3 → 0.286
Rainy	2/4 →	2+1/4+3 → 0.429	Mild	1/4 → 1+1/4+3 → 0.286
Sunny	0/4 →	0+1/4+3 → 0.143	Cool	2/4 → 2+1/4+3 → 0.429
Cloudy	0/2 →	0+1/2+3 → 0.2	HOT	1/2 → 1+1/2+3 → 0.4
Rainy	0/2 →	0+1/2+3 → 0.2	Mild	1/2 → 1+1/2+3 → 0.4
Sunny	2/2 →	2+1/2+3 → 0.6	Cool	0/2 → 0+1/2+3 → 0.2

$P(Yes) \rightarrow 4/6 \rightarrow 0.66$

$P(No) \rightarrow 2/6 \rightarrow 0.33$

→ prob of sunny & cool

$$P(Yes \mid Sunny, Cool) \propto P(Yes) \times P(Sunny \mid Yes) \times P(Cool \mid Yes)$$

$$= 0.66 \times 0.143 \times 0.429 = \underline{0.040}$$

$$P(No \mid Sunny, Cool) \propto P(No) \times P(Sunny \mid No) \times P(Cool \mid No)$$

$$= 0.33 \times 0.6 \times 0.2$$

$$= \underline{0.0396}$$

Normalise to get the final probabilities:

Sum = $0.04 + 0.0396 = 0.0796$

$$P(\text{Yes} \mid \text{Sunny}, \text{Cool}) = \frac{0.040}{0.0796} \approx 0.5025$$

sum ≈ 1

$$P(\text{No} \mid \text{Sunny}, \text{Cool}) = \frac{0.0396}{0.0796} \approx 0.4974$$

$$P(\text{Yes} \mid \text{Sunny}, \text{Cool}) > P(\text{No} \mid \text{Sunny}, \text{Cool})$$

$$P(\text{Yes}) = P(\text{No})$$

→ features → categorical
→ target → categorical
→ Categorical Naive Bayes

Next topic:

Dataset → numeric features → categorical NB?
→ categorical target column

→ Gaussian NB
↓
(normal)

we compute:

$$P(y \mid x) \propto P(y) \prod N(x \mid \mu_y, \sigma_y^2)$$

$$N(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

exp (2)
↓
e^2
↓
2.71^2

Data set:

id	x_1	x_2	y
A	1.0	2.0	Yes
B	2.0	1.0	Yes
C	1.5	1.8.	Yes
D	3.0	3.0	NO
E	3.5	2.8	NO
F	2.9	3.2	NO

Step 1: Calculate MLE:

Maximum likelihood estimate: (MLE) → uses n instead of n-1 which is used in unbiased sample variance

mean, variance, std

$\mu(x_1 = Yes) = 1+2+1.5/3 \rightarrow \mu_{Yes,1} = 1.5$
$= 1.5$

$\mu(x_2 = Yes) = 2+1+1.8/3 \rightarrow \mu_{Yes,2} = 1.6$
$= 1.6$

$\mu(x_1 = NO) = 3+3.5+2.9/3 \rightarrow \mu_{NO,1} = 3.133$
$= 3.133$

$\mu(x_2 = NO) = 3+2.8+3.2/3 \rightarrow \mu_{NO,2} = 3.0$
$= 3.0$

variance X1:

$\sigma^2 = \sum (x-\mu)^2 / N$

(Yes) : $(1-1.5)^2, (2-1.5)^2, (1.5-1.5)^2$

: $(-0.5)^2, (0.5)^2, 0^2$

Sum : $0.25 + 0.25 + 0 \rightarrow 0.5/3 \rightarrow 0.166$

variance x_2:

(Yes) : $(0.4)^2 , (-0.6)^2 , (0.2)^2 \rightarrow 0.56$

: $0.56/3 \rightarrow 0.186$

variance x_1: (NO) sum $\rightarrow 0.2066 \rightarrow 0.2066/3 \rightarrow 0.0688$

variance x_2: (NO) sum $\rightarrow 0.08 \rightarrow 0.08/3 \rightarrow 0.0266$

x_1 x_2

$x = (2.0, 2.0)$

Step 2: calculate Gaussian prob per feature:

For class = Yes

→ x_1:

→ $\sigma = \sqrt{0.166} \approx 0.408$

→ coeff = $1/(\sqrt{2\pi\sigma^2}) \approx 0.9772$

→ exponent exp: $\left(-\frac{(2-1.5)^2}{2 \times 0.166}\right) \approx \exp(-0.75) \approx 0.472$

→ $P(x_1 | \text{Yes}) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$\approx 0.9772 \times 0.472$

≈ 0.4616

$P(x_1 \mid Yes)$ ✓ $P(x_2 \mid Yes)$ ✓

$P(x_1 \mid No)$ $P(x_2 \mid No)$

x_2 :

→ $\sigma = \sqrt{0.1866} \approx 0.4320$

→ coeff ≈ 0.923

→ expon exp : $\exp\left(-\frac{(2-1.6)^2}{2 \times 0.1866}\right) = \exp(-0.428) \approx 0.6510$

$P(x_2 \mid Yes) \approx 0.4320 \times 0.6510$

≈ 0.6009

For class = No :

$P(x_1 \mid No) \approx 1.36 \times 10^{-4}$

$P(x_2 \mid No) \approx 1.68 \times 10^{-8}$

$P(x_1 \mid Yes) \approx 0.4816$ $P(x_2 \mid Yes) \approx 0.6009$

Joint prob : $P(x \mid Yes) = 0.4816 \times 0.6009$

≈ 0.2774

$P(x_1 \mid No) \approx 1.36 \times 10^{-4}$ $P(x_2 \mid No) \approx 1.68 \times 10^{-8}$

Joint prob : $P(x \mid No) \approx 2.29 \times 10^{-12}$

$p(x \mid \text{Yes}) \approx 0.2774$ $\quad p(x \mid \text{No}) \approx 2.29 \times 10^{-12}$

↓

unnormalized prob

$P(\text{Yes} \mid x) : p(\text{Yes}) \cdot p(x \mid \text{Yes})$

$: 0.5 \times 0.2774$

$\approx : 0.1387$

$P(\text{No} \mid x) : p(\text{No}) \cdot p(x \mid \text{No})$

$: 0.5 \times 2.29 \times 10^{-12}$

$: 1.145 \times 10^{-12}$

Step 3: normalize to get (real) probability.

Sum S : $0.1387 + 1.145 \times 10^{-12} \approx 0.1387$

so normalized!

$$P(\text{Yes} \mid x) \approx \frac{0.1387}{0.1387} \approx 0.9999\ldots \approx 1$$

$$P(\text{No} \mid x) \approx \frac{1.145 \times 10^{-12}}{0.1387} \approx 8.26 \times 10^{-12}$$

Decision: predict → Yes (very strongly)

↓

$$\log p(y \mid x) = \log p(y) + \sum \log \mathcal{N}(x \mid \mu, \sigma^2)$$

↑

python implement

when to use Guassian NB:

→ features are continuous

→ you want a very fast, interpretable model.

→ we are estimating that features are normally distributed.

Email → spam → $P(s) = 0.2$

Not spam

$P(ns) = 0.8$

s : please click on the link below

$P(\text{plea} \mid s) = x_1$

$P(\text{click} \mid s) = x_2$

ns : $P(\quad) = A_1$

$= A_2$

P_1

→ Tommy is holiday

↓

enjoy

$P(\text{Yes} \mid x) = P(\text{spam}) \times P(\text{Tom} \mid \text{Yes}) \times P($

$= 0.6$

$P(\text{No} \mid x) = 0.4$

New Algo:

k-means clustering !

→ unsupervised learning → No target column

Data → k-means clustering → group data based on similarity (clustering)

↑
how many groups you want?

↑
K

→ k → no of centroids
→ k = 2

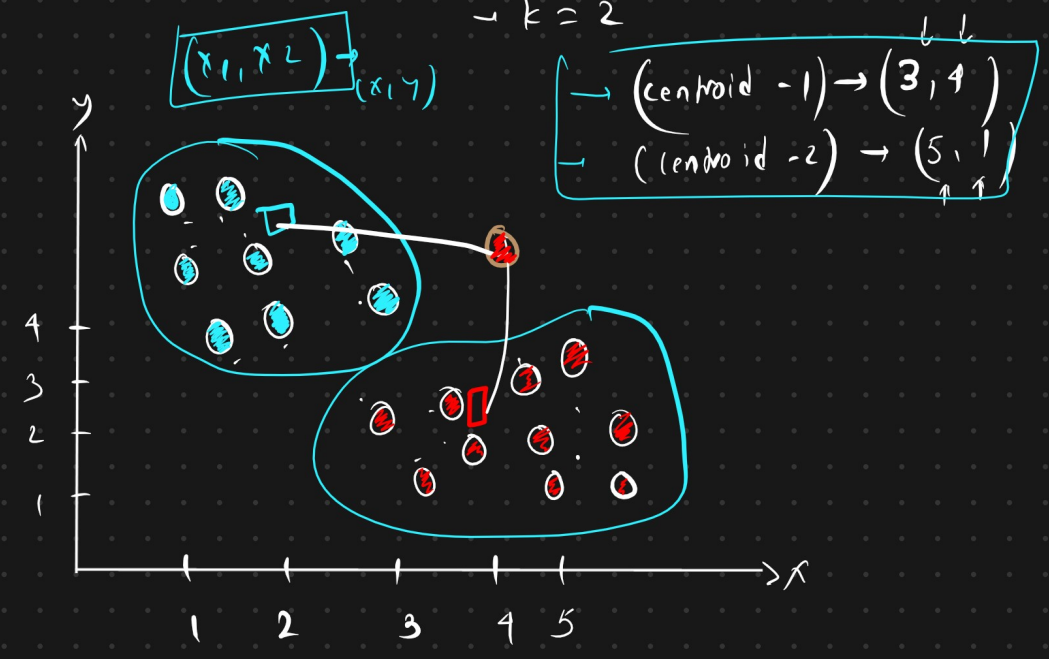
Intitution !

Step train

B → 8
R → 10

Iter -3 :

B → 8
R → 10



→ (centroid -1) → (3, 4)
→ (centroid -2) → (5, 1)

Dataset:

A = (1,2)
B = (1.5, 1.8)
C = (5,8)
D = (8,8)
E = (1, 0.6)
F = (9,11)

k = 2

$c_1 = A = (1,2)$
$c_2 = C = (5,8)$

Euclidean distance:

$(x_1, x_2), (A_1, A_2)$

$$\sqrt{(x_1 - A_1)^2 + (x_2 - A_2)^2}$$

Iteration 1 : (Step 1)

1. point A (1,2)
   - to c_1: $\sqrt{(1-1)^2 + (2-2)^2} = 0$
   - to c_2: $\sqrt{(1-5)^2 + (2-8)^2} = 7.211$
   - → assign to cluster c1
2. point B (1.5, 1.8)
   - to c_1: $\sqrt{(1.5-1)^2 + (1.8-2)^2} \approx 0.5385$
   - to c_2: $\sqrt{(1.5-5)^2 + (1.8-8)^2} \approx 7.119$
   - → cluster 1
3. point C → cluster 2
4. point D → cluster 2
5. point E → cluster 1
6. point F → cluster 2

cluster 1 : { A, B, E }
cluster 2 : { C, D, F }

Step 2: update centroids

New c_1: mean of A(1,2), B(1.5, 1.8), E(1, 0.6)
- c_1_x : (1 + 1.5 + 1)/3 = 1.1667
- c_1_y : (2 + 1.8 + 0.6)/3 = 1.4667

New c_2:
mean of C(5,8), D(8,8), F(9,11)
- c_2_x : (5 + 8 + 9)/3 ≈ 7.33
- c_2_y : (8 + 8 + 11)/3 ≈ 9

c_1 : (1.1667, 1.4667)
c_2 : (7.33, 9)

How to compute loss → WCSS

→ we calculate WCSS after every iteration of centroid updates.

→ WCSS → within cluster summation square

→ Sum over clusters & point of squared distance to their centroid.

Iteration: 2

→ A(1,2) ≈ 0.5587
→ C1

→ B(1.5, 1.8) ≈ 0.4714
→ C1

→ C(5,8) ≈ 2.539
→ C2

→ D(8,8) ≈ 1.2019
→ C2

→ F(9,11) ≈ 2.602
→ C2

new_C1 (1.1667, 1.4667)
new_C2 (7.3333, 9)

WCSS:

C1 → A, B, E

$(0.5587)^2 + (0.4714)^2 + (0.1833)^2$

= 1.3163

C2 → C, D, F

→ $(2.539)^2 + (1.2019)^2 + (2.602)^2$

→ 14.666

WCSS → WCSS(C1) + WCSS(C2)

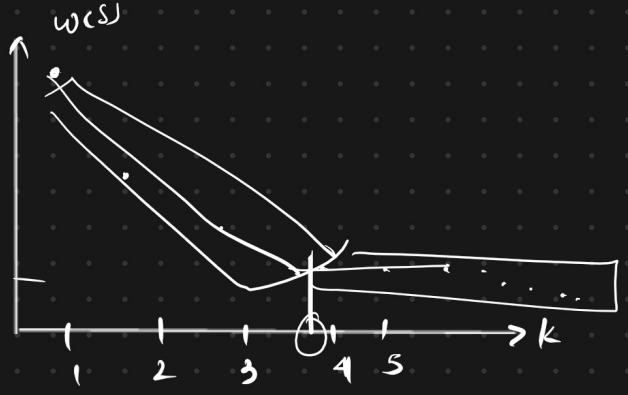
→ 15.992

18.683

→ To get optimal value k, plot elbow graph

x axis → k

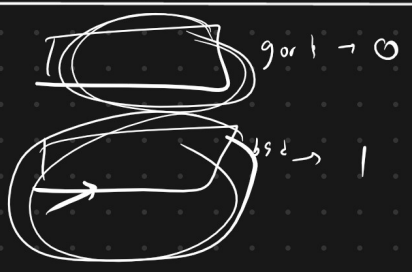
y-axis → WCSS



increase in k, decrease WCSS

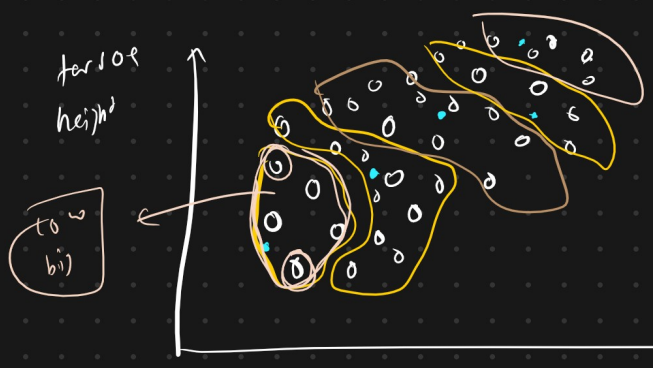
---

dataset → 1000 rows → kmean (2) →



review

bad, good k=2



→ S, M, L, XL, XXL

↓

5