Data Mining 第二次作業

第一大題

(一)

資料設定

有一組二維資料點，還有一個新資料點 x=(1.4,1.6)x = (1.4, 1.6)x=(1.4,1.6) 作為查詢點 (query)。

題目

點	向量 `(A₁, A₂)`
q	(1.4, 1.6)
x_1	(1.5, 1.7)
x_2	(2.0, 1.9)
x_3	(1.6, 1.8)
x_4	(1.2, 1.5)
x_5	(1.5, 1.0)

(a) 計算歐式距離 $d(\mathbf{x}, \mathbf{q})$

$$d(\mathbf{x}, \mathbf{q}) = \sqrt{(A_1^x - A_1^q)^2 + (A_2^x - A_2^q)^2}.$$

點	ΔA_1	ΔA_2	$(\Delta A_1)^2$	$(\Delta A_2)^2$	$(\Delta A_1)^2 + (\Delta A_2)^2$	d
x_1	$1.5 - 1.4 = 0.1$	$1.7 - 1.6 = 0.1$	0.01	0.01	0.02	$\sqrt{0.02} = \mathbf{0.1414}$
x_2	$2.0 - 1.4 = 0.6$	$1.9 - 1.6 = 0.3$	0.36	0.09	0.45	$\sqrt{0.45} = \mathbf{0.6708}$
x_3	$1.6 - 1.4 = 0.2$	$1.8 - 1.6 = 0.2$	0.04	0.04	0.08	$\sqrt{0.08} = \mathbf{0.2828}$
x_4	$1.2 - 1.4 = -0.2$	$1.5 - 1.6 = -0.1$	0.04	0.01	0.05	$\sqrt{0.05} = \mathbf{0.2236}$
x_5	$1.5 - 1.4 = 0.1$	$1.0 - 1.6 = -0.6$	0.01	0.36	0.37	$\sqrt{0.37} = \mathbf{0.6083}$

排序（由近到遠）

$$x_1 \;<\; x_4 \;<\; x_3 \;<\; x_5 \;<\; x_2.$$

(b) 餘弦相似度

$$\cos(\mathbf{x}, \mathbf{q}) = \frac{\mathbf{x} \cdot \mathbf{q}}{\|\mathbf{x}\|\,\|\mathbf{q}\|}, \qquad \mathbf{x} \cdot \mathbf{q} = A_1^x A_1^q + A_2^x A_2^q.$$

算查詢點的長度(下方)：

$$\|\mathbf{q}\| = \sqrt{1.4^2 + 1.6^2} = \sqrt{1.96 + 2.56} = \sqrt{4.52} = 2.1260.$$

點	內積 $\mathbf{x}\cdot\mathbf{q}$	$\|\mathbf{x}\|$	分母 $\|\mathbf{x}\|\,\|\mathbf{q}\|$	cos
x_1	$1.5 \cdot 1.4 + 1.7 \cdot 1.6 = 2.1 + 2.72 = 4.82$	$\sqrt{1.5^2 + 1.7^2} = \sqrt{2.25 + 2.89} = 2.2689$	$2.2689 \times 2.1260 = 4.8220$	$\frac{4.82}{4.8220} = \mathbf{0.9990}$
x_2	$2.0 \cdot 1.4 + 1.9 \cdot 1.6 = 2.8 + 3.04 = 5.84$	$\sqrt{2.0^2 + 1.9^2} = \sqrt{4 + 3.61} = 2.7580$	$2.7580 \times 2.1260 = 5.8670$	$\frac{5.84}{5.8670} = \mathbf{0.9954}$
x_3	$1.6 \cdot 1.4 + 1.8 \cdot 1.6 = 2.24 + 2.88 = 5.12$	$\sqrt{1.6^2 + 1.8^2} = \sqrt{2.56 + 3.24} = 2.4083$	$2.4083 \times 2.1260 = 5.1260$	$\frac{5.12}{5.1260} = \mathbf{0.9988}$
x_4	$1.2 \cdot 1.4 + 1.5 \cdot 1.6 = 1.68 + 2.40 = 4.08$	$\sqrt{1.2^2 + 1.5^2} = \sqrt{1.44 + 2.25} = 1.9203$	$1.9203 \times 2.1260 = 4.0790$	$\frac{4.08}{4.0790} = \mathbf{0.9990}$
x_5	$1.5 \cdot 1.4 + 1.0 \cdot 1.6 = 2.10 + 1.60 = 3.70$	$\sqrt{1.5^2 + 1.0^2} = \sqrt{2.25 + 1.00} = 1.8028$	$1.8028 \times 2.1260 = 3.8330$	$\frac{3.70}{3.8330} = \mathbf{0.9654}$

排名（相似度大→小）：

$$x_1 \gtrsim x_4 \gtrsim x_3 > x_2 > x_5.$$

(二) 做向量正規化

> 單位化公式
>
> $$\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}, \qquad \hat{\mathbf{q}} = \frac{\mathbf{q}}{\|\mathbf{q}\|}.$$

(a) 求單位向量

點	$\|\mathbf{x}\|$	$\hat{\mathbf{x}} = (A_1/\|\mathbf{x}\|,\ A_2/\|\mathbf{x}\|)$
x_1	2.2689	(0.6610, 0.7504)
x_2	2.7580	(0.7251, 0.6887)
x_3	2.4083	(0.6644, 0.7474)
x_4	1.9203	(0.6245, 0.7809)
x_5	1.8028	(0.8321, 0.5546)

查詢點

$$\hat{\mathbf{q}} = \big(1.4/2.1260,\ 1.6/2.1260\big) = (0.6587,\ 0.7524).$$

(b) 歐氏距離於單位空間

點	ΔA_1	ΔA_2	$(\Delta A_1)^2$	$(\Delta A_2)^2$	距離 $d(\hat{\mathbf{x}}, \hat{\mathbf{q}})$
x_1	0.0023	–0.0020	5.3×10^{-6}	4.0×10^{-6}	**0.0041**
x_3	0.0057	–0.0050	3.3×10^{-5}	2.5×10^{-5}	**0.0078**
x_4	–0.0342	0.0285	1.17×10^{-3}	8.12×10^{-4}	**0.0441**
x_2	0.0664	–0.0637	4.41×10^{-3}	4.06×10^{-3}	**0.0922**
x_5	0.1734	–0.1978	3.01×10^{-2}	3.91×10^{-2}	**0.2632**

$$x_1 < x_3 < x_4 < x_2 < x_5.$$

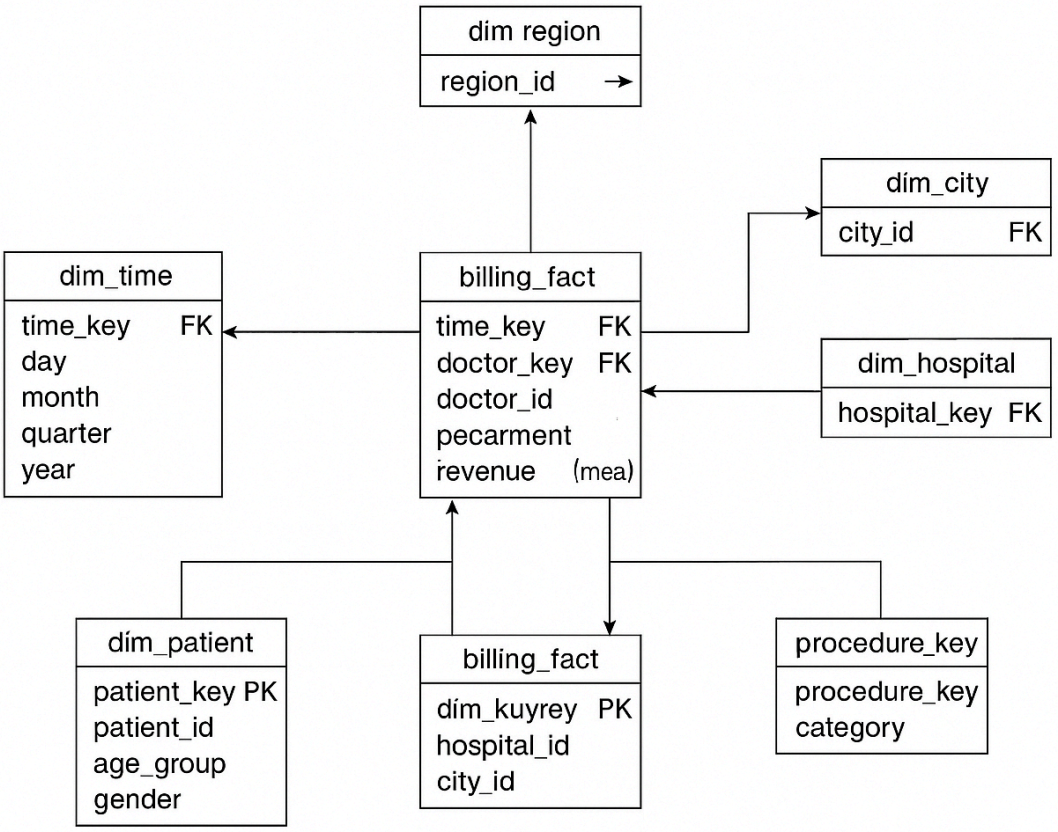
第二大題

(1) 三種常見 OLAP Operation 與醫院倉儲

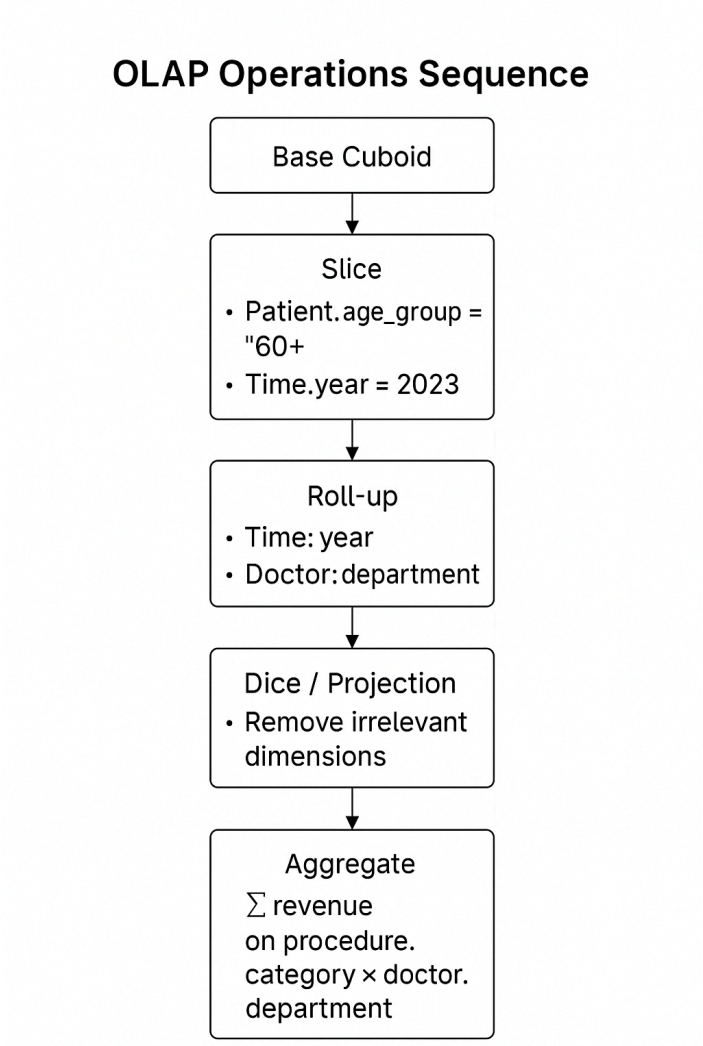
有Roll-up Drill-down Slice/Dice

操作	說明	醫院倉儲
Roll-up	由細節層級 ⟶ 較高層級聚合	先以 `Time=day` 彙總，再 **roll-up** 到 `month` → 查看 *2023 各月* 總 revenue
Drill-down	與 roll-up 相反，向下查看更細節	已在 `year` 粒度檢視成本後，**drill-down** 到 `quarter`，進一步比較 Q1～Q4
Slice	固定單一維度的一個成員，抽出子立方體	**slice** `Patient.age_group = '60+'` → 只保留高齡病患的 cost / revenue

(2)設計 Snowflake Schema



(3)OLAP 操作序列(針對本題為例)



> 起點：**Base Cuboid** (= 全部維度最細粒度)

操作序列

1. **Slice**
   - `Patient.age_group = '60+'`
   - `Time.year = 2023`
2. **Roll-up** 時間維 → 不需要 month/quarter/day，保留 `year` (已 slice)。
3. **Roll-up** 醫師維 → 由 `doctor_id` 聚合到 `department`。
4. **Roll-up** 手術維 → 由 `procedure_type` 聚合到 `category`。
5. **Dice / Projection** 移除與查詢無關的維度 (`Hospital`, 其餘時間層級等)。
6. **Aggregate** (sum) → 計算 `Σ revenue` 於二維切面 `(procedure.category, doctor.department)`。

(4)SQL查詢寫法

```sql
SELECT
    p.category            AS procedure_category,
    d.department          AS doctor_department,
    SUM(f.revenue)        AS total_revenue
FROM
    billing_fact      f
    JOIN dim_time      t ON f.time_key      = t.time_key
    JOIN dim_patient   pa ON f.patient_key  = pa.patient_key
    JOIN dim_doctor    d ON f.doctor_key    = d.doctor_key
    JOIN dim_procedure p ON f.procedure_key = p.procedure_key
WHERE
    t.year = 2023          -- 時間 Slice
    AND pa.age_group = '60+'    -- 病患 Slice
GROUP BY
    p.category,
    d.department
ORDER BY
    total_revenue DESC;
```

第三大題

department	status	age range	salary range	count
sales	senior	31…35	46K…50K	30
sales	junior	26…30	26K…30K	40
sales	junior	31…35	31K…35K	40
systems	junior	21…25	46K…50K	20
systems	senior	31…35	66K…70K	5
systems	junior	26…30	46K…50K	3
systems	senior	41…45	66K…70K	3
marketing	senior	36…40	46K…50K	10
marketing	junior	31…35	41K…45K	4
secretary	senior	46…50	36K…40K	4
secretary	junior	26…30	26K…30K	6

總筆數:

$$N = \sum count = 165, \qquad N_{\text{senior}} = 52,\ N_{\text{junior}} = 113$$

(1) count 參與決策樹演算法

> 以 **ID3 / C4.5** 為例，只需把每筆樣本的「詞頻」改成「加權詞頻」。

1. **節點熵（或基尼）**

$$H(S) = -\sum_{c \in \{\text{senior},\text{junior}\}} \frac{\sum_{i \in c} count_i}{\sum_i count_i} \log_2 \left(\frac{\sum_{i \in c} count_i}{\sum_i count_i} \right)$$

2. **分割後的加權熵**

$$H_{\text{split}} = \sum_{v \in \text{values}(A)} \frac{\sum_{i \in v} count_i}{N} H(S_v)$$

3. **資訊增益 / 增益率 / 基尼減少量**

與原演算法公式相同，只是所有頻數都以 **count 欄** 相加。

4. 其餘流程（選最佳屬性、遞迴建樹、剪枝）全部不變。

主要是把**一列代表中的一個樣本 ➤一列代表 *count* 個樣本**。

(2)

(a)選根節點
計算三個屬性的資訊增益（以熵為例）：

屬性	加權熵 H_{split}	信息增益 $= H(\text{root}) - H_{\text{split}}$
salary	0.362	**0.536**
age	0.473	0.425
department	0.851	0.047

> 最大增益 → **salary** 為根。

(b)依照 `salary` 列表去做劃分

除 `46K···50K` 外，其餘葉節點皆純（熵 = 0）：

salary range	筆數	senior	junior	標籤
26K⋯30K	46	0	46	**junior**
31K⋯35K	40	0	40	**junior**
36K⋯40K	4	4	0	**senior**
41K⋯45K	4	0	4	**junior**
66K⋯70K	8	8	0	**senior**
46K⋯50K	63	40	23	⟶ **需再分裂**

(c) 分裂 `46K···50K`

在該subset中，`department` 或 `age` 任一皆能產生純葉。

department	senior	junior	標籤
sales	30	0	**senior**
systems	0	23	**junior**
marketing	10	0	**senior**

(d) 最後得出以下Decision Tree：

```
salary?
├── 26 ... 30K → junior
├── 31 ... 35K → junior
├── 36 ... 40K → senior
├── 41 ... 45K → junior
├── 46 ... 50K
|     ├── department = sales → senior
```

```
8  |      ├── department = systems → junior
9  |      └── department = marketing → senior
10 └── 66 ... 70K → senior
```

(3) naïve Bayesian

> **欲分類樣本**
> $(X = department = systems, age = 26 \cdot \cdot \cdot 30, salary = 46K \cdot \cdot \cdot 50K)$

(a)先驗機率

$$P(\text{senior}) = \frac{52}{165} = 0.315, \quad P(\text{junior}) = \frac{113}{165} = 0.685$$

(b)條件機率

條件	senior	junior
$(P(\text{dept=systems})$	$\frac{8}{52} = 0.154$	$\frac{23}{113} = 0.204$
$(P(\text{age } 26 \cdot \cdot \cdot 30)$	$\frac{0}{52} = 0$	$\frac{49}{113} = 0.434$
$(P(\text{salary } 46 \cdot \cdot \cdot 50\text{K})$	$\frac{40}{52} = 0.769$	$\frac{23}{113} = 0.204$

> 若不加平滑，**senior** 類因年齡條件為 0 而直接歸零，反之，也遠低於零。

(c)後驗比例

$$\Pr(\text{senior} \mid X) \propto 0.315 \times 0.154 \times 0 \times 0.769 = 0,$$
$$\Pr(\text{junior} \mid X) \propto 0.685 \times 0.204 \times 0.434 \times 0.204 \approx 1.236 \times 10^{-2}.$$

(d)預測結果

$$\boxed{\text{status} = \text{junior}}$$

基本上與上一節決策樹結論一樣。即使使用了 Laplace 平滑，$\Pr(\text{senior} \mid X)$ 仍低於 $\Pr(\text{junior} \mid X)$，且十分顯著。

第四大題

(a) 二元分類混淆矩陣符號

	實際 Positive (P)	**實際 Negative (N)**	**合計**
預測 Positive	True Positive (TP)	False Positive (FP)	TP+FP
預測 Negative	False Negative (FN)	True Negative (TN)	FN+TN
合計	TP+FN	FP+TN	N

- **靈敏度 (Sensitivity, Recall)**

$$\text{Se} = \frac{TP}{TP + FN}.$$

- **特異度 (Specificity)**

$$\text{Sp} = \frac{TN}{TN + FP}.$$

- **整體準確率 (Accuracy)**

$$\text{Acc} = \frac{TP + TN}{N}, \qquad N = TP + FP + TN + FN.$$

(b) 以 Sensitivity、Specificity 表達 Accuracy

1. 記 **患病率 (Prevalence)**

$$\pi = \frac{TP + FN}{N}.$$

2. 把 TP、TN 改寫成 $(\mathrm{Se}, \mathrm{Sp}, \pi)$ 之函數

   - $$TP = \mathrm{Se}\,(TP + FN) = \mathrm{Se}\,\pi N.$$

   - $$TN = \mathrm{Sp}\,(TN + FP) = \mathrm{Sp}\,(1 - \pi) N.$$

3. 代回 Accuracy 定義

$$\begin{aligned} \mathrm{Acc} &= \frac{TP + TN}{N} \\ &= \frac{\mathrm{Se}\,\pi N + \mathrm{Sp}\,(1-\pi) N}{N} \\ &= \boxed{\pi\,\mathrm{Se} \;+\; (1-\pi)\,\mathrm{Sp}}. \end{aligned}$$

(c) 結論

- 在某一固定的 **患病率** π（或稱 *positive class prevalence*）下，
  準確率 **Acc** 僅由 **Sensitivity** 與 **Specificity** 線性決定。
- 可寫成以下方式：

$$\mathrm{Acc} = (1 - \pi)\,\mathrm{Sp} + \pi\,\mathrm{Se}.$$

因此 **Accuracy 是 $(\mathrm{Se}, \mathrm{Sp})$ 的函數**（以 π 為固定參數）。