สรุปการทดลอง Mini Project

ชุดข้อมูล : Cancer

จัดทำโดย

นายภูมิระพี เสริญวณิชกุล

รหัสนิสิต 6510405750 หมู่ 1

เสนอ

รองศาสตราจารย์ ดร. นวลวรรณ สุนทรภิษัช

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา

Artificial Intelligence (01418261)

ชุดข้อมูลที่ได้รับ: cancer.csv

ในไฟล์ cancer.csv จะเป็นข้อมูลผู้ป่วยที่รักษาโรคมะเร็งตับ โดยจะแบ่งออกเป็น 4 ระยะ (CA Level) ซึ่งจะมีข้อมูลแต่ละคอลัมน์ดังนี้ Sex, Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alk_Phos, SGOT, Trygliceride, Platelets, Prothrombin, CA level

Sex	Ascites	Hepatome	Spiders	Edema	Bilirubin	Cholester	Albumin	Copper	Alk_Phos	SGOT	Trygliceric	Platelets	Prothrom	CA level
F	Y	Y	Y	Y	14.5	261	2.6	156	1718	137.95	172	190	12.2	4
F	N	Y	Y	N	1.1	302	4.14	54	7394.8	113.52	88	221	10.6	3
M	N	N	N	S	1.4	176	3.48	210	516	96.1	55	151	12	4
F	N	Y	Y	S	1.8	244	2.54	64	6121.8	60.63	92	183	10.3	4
F	N	Y	Y	N	3.4	279	3.53	143	671	113.15	72	136	10.9	3
F	N	Y	N	N	0.8	248	3.98	50	944	93	63	200	11	3
F	N	Y	N	N	1	322	4.09	52	824	60.45	213	204	9.7	3

cancer.csv

1. Preprocessing

ในชุดข้อมูลที่ได้รับมีบางคอลัมน์ที่มีข้อมูลที่เป็นสตริง เราก็ต้องแปลงข้อมูลให้อยู่ในรูปแบบชนิดเลขจำนวนเต็มก่อนเริ่มทำการเทรน หลังจากนั้นเราก็เลือกคอลัมน์ Sex, Ascites, Hepatomegaly, Spiders, Edema, Bilirubin, Cholesterol, Albumin, Copper, Alk_Phos, SGOT, Trygliceride, Platelets, Prothrombin เป็นฟีเจอร์ และ CA level เป็นคลาส

2. Classification Model

2.1 Decision Tree

2.1.1 การแบ่งข้อมูล

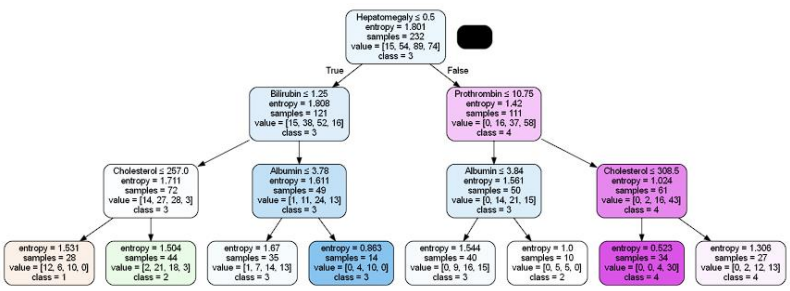
ทำการแบ่งข้อมูลออกเป็นข้อมูลที่ใช้สำหรับเทรน 75% และทดสอบ 25%

2.1.2 การสร้างโมเดล

ทำการสร้างโมเดลโดยใช้เกณฑ์ Entropy เพื่อวัดค่าความไม่แน่นอน (ค่าต่ำยิ่งดี) หลังจากนั้นกำหนดความลึกของต้นไม้คือ 3

2.1.3 ผลลัพธ์

ทดสอบได้ความแม่นยำประมาณ 51.28%



2.2 Support Vector Machine (SVM)

2.2.1 การแบ่งข้อมูล

ทำการแบ่งข้อมูลออกเป็นข้อมูลที่ใช้สำหรับเทรน 70% และทดสอบ 30%

2.2.2 การสร้างโมเดล

ใช้ GridSearchCV เพื่อเลือกปรับ Hyperparameter ให้มีความเหมาะสมที่สุด โดยกำหนดค่าใน Grid ดังนี้ C: [0.01, 0.1, 1, 10], kernel: ['linear', 'poly', 'rbf', 'sigmoid'], degree: [1, 3, 5, 7], gamma: [0.01, 1] จากนั้นทำ MultiClass-Classification โดยใช้ Linear Support Vector Classificationและใช้ OneVsOne เพื่อแบ่งคู่ในการจำแนก

2.2.3 ผลลัพธ์

ทดสอบได้ความแม่นยำประมาณ 54.84%

2.3 Naïve Bayes

2.3.1 การแบ่งข้อมูล

ทำการแบ่งข้อมูลออกเป็นข้อมูลที่ใช้สำหรับเทรน 70% และทดสอบ 30%

2.3.2 การสร้างโมเดล

สร้างโมเดลโดยใช้ Multinomial Naïve Nayes โดยมีการกำหนดพารามิเตอร์ดังนี้

alpha=1.5, class_prior=None, fir_prior=True

alpha: สำหรับหลีกเลี่ยง zero probability

class_prior: ความน่าจะเป็นของคลาส (เราไม่ได้กำหนดก็จะทำการคำนวณจากข้อมูล)

fit_prior: กำหนดว่าให้มีการเรียนรู้ class prior probabilities

2.3.3 ผลลัพธ์

ทดสอบได้ความแม่นยำประมาณ 41.94%

2.4 K-nearest neighbors

2.3.1 การแบ่งข้อมูล

ทำการแบ่งข้อมูลออกเป็นข้อมูลที่ใช้สำหรับเทรน 75% และทดสอบ 25%

2.3.2 การสร้างโมเดล

ทำการปรับข้อมูลให้เป็นมาตรฐานมากขึ้นด้วยฟังก์ชัน StandardScaler() หลังจากนั้นทำการสร้างโมเดล KNN โดยทำการระบุพารามิเตอร์ n_neighbors=7

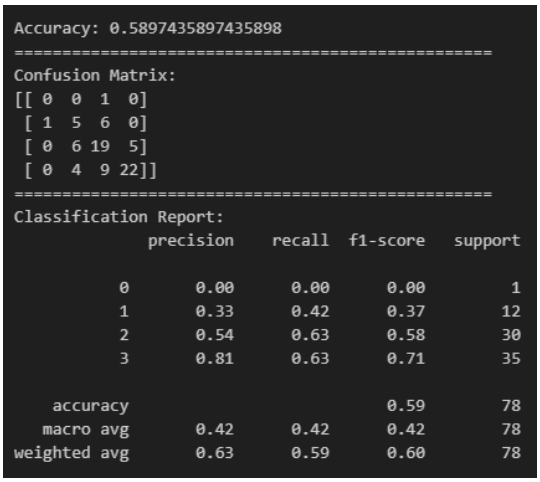
ซึ่งหมายความว่าตัวโมเดลจะใช้ 7 ตัวอย่างที่ใกล้ที่สุดเพื่อใช้ในการทำนาย โดยเลือก n_neighbors จะขึ้นอยู่กับโมเดลและประสิทธิภาพของผลลัพธ์ที่เกิดจากตัวโมเดล

2.3.3 ผลลัพธ์

ทดสอบได้ความแม่นยำประมาณ 58.97%

3. Conclusion

จากการวิเคราะห์ทั้ง 4 โมเดล จากผลลัพธ์ที่ได้จากการทดสอบและประเมินประสิทธิภาพของแต่ละโมเดลพบว่า K-nearest neighbors (KNN) เป็นตัวแบบที่ดีที่สุดสำหรับชุดข้อมูลนี้



```
Accuracy: 0.5897435897435898
================================================
Confusion Matrix:
[[ 0  0  1  0]
 [ 1  5  6  0]
 [ 0  6 19  5]
 [ 0  4  9 22]]
================================================
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.33      0.42      0.37        12
           2       0.54      0.63      0.58        30
           3       0.81      0.63      0.71        35

    accuracy                           0.59        78
   macro avg       0.42      0.42      0.42        78
weighted avg       0.63      0.59      0.60        78
```

4. How Different?

I. Decision Tree

- เป็นการแบ่งข้อมูลโดยเลือกตัวแปรที่สามารถแยกข้อมูลได้มากที่สุดมาเป็น Root และแยกข้อมูลไปเรื่อยๆ แสดงแผนภาพที่เข้าใจ ง่ายต่อการวิเคราะห์ แต่มีโอกาสเกิด Overfitting

II. SVM

- เป็นการหาเส้นแบ่งระหว่างคลาสในข้อมูลโดยสามารถจัดการกับข้อมูลที่เป็น non-linear ได้ และเหมาะกับข้อมูลที่มีมิติ แต่ใช้เวลาในการทำนายค่อยข้างนานและยังอาจเกิด Overfitting ได้ในบางกรณี

III. Naïve Bayes

- เป็นการจัดหมวดหมู่โดยอ้างอิงจากความน่าจะเป็น มีความรวดเร็วต่าการทำนาย เหมาะกับข้อมูลที่เป็นข้อความในบางกรณี แต่ถ้าข้อมูลไม่มีความเป็นอิสระก็อาจทำให้ผลลัพธ์การทำนายไม่ถูกต้อง

IV. K-Nearest Neighbors

- เป็นการจัดกลุ่มข้อมูลโดยเลือกตัวข้อมูลตัวที่ใกล้ที่สุดกับข้อมูลที่เราสนใจ โดยสามารถทำนายโดยที่ไม่ต้องฝึก เหมาะกับข้อมูลขนาดเล็กไปจนถึงปานกลาง แต่อาจทำให้ประสิทธิภาพลดลงถ้าเป็นข้อมูลที่มีมิติซับซ้อนเกินไป ไม่เหมาะกับข้อมูลขนาดใหญ่