ASS1: Decision Tree Classifiers

6440126922 นิปุณ อังควิชัย

ข้อมูล Employee Attrition

จากการสร้างแบบโมเดล Decision Tree ทำนาย Employee Attrition ตามขั้นตอนใน Google Colab notebook decision_tree.ipynb โดยจากข้อมูลมี Label distribution ใน dataset ที่แบ่งไว้เทรนและทดสอบดังนี้

Dataset/Label	Positive label (1)	Negative label (0)
Training set	192	984
Test set	45	249

โมเดลและพารามิเตอร์

1. CART using GINI Criterion

Tree Depth	14
Min Samples Split	2

2. CART using GridSearchCV()

Max Depth	5
Min Samples Split	2
Min Samples Leaf	8
Max Feature	0.8

3. RandomForestClassifier()

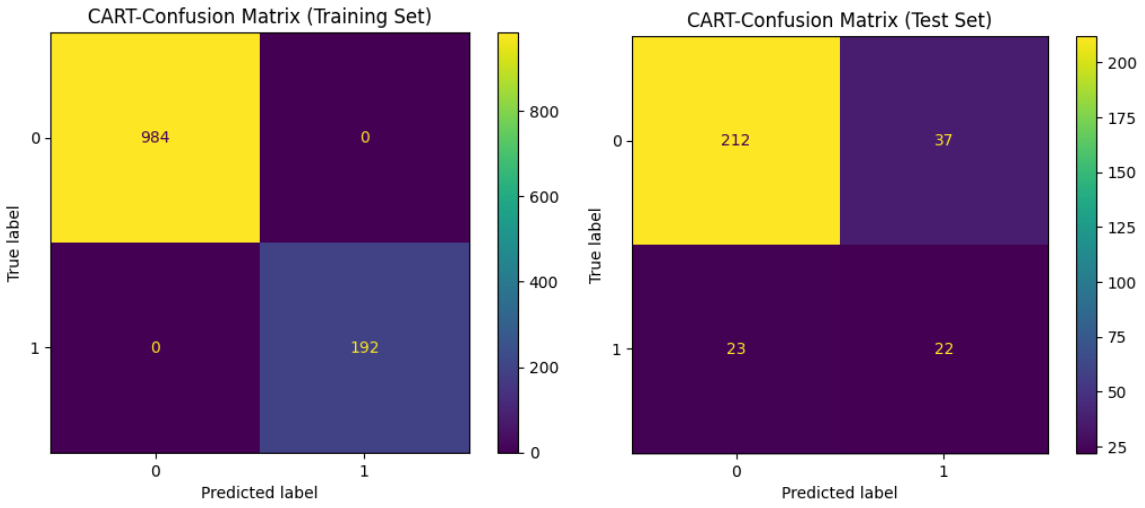
Max Depth	15
Min Samples Split	5
n_estimators	100
Max Feature	0.5

ผลลัพธ์ของโมเดล

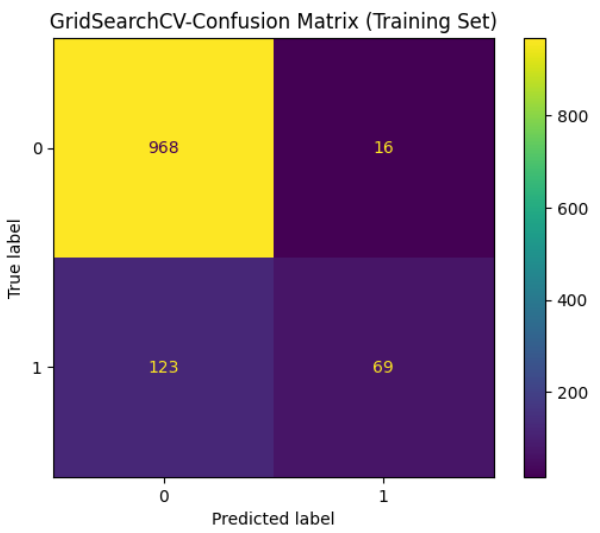
ประสิทธิภาพความแม่นยำของโมเดล Accuracy ต่อ Training set และ Test set

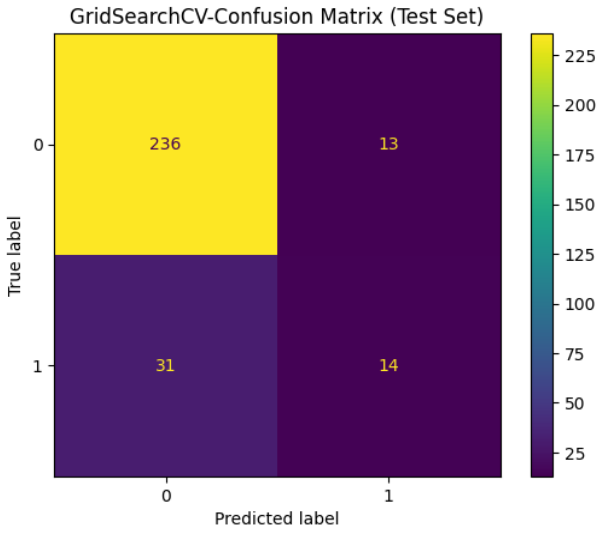
Dataset/Model	CART using GINI Criterion	CART using GridSearchCV()	RandomForestClassifier()
Training set	**1.00**	0.88	0.99
Test set	0.80	0.85	**0.89**

Confusion Matrix ของ CART

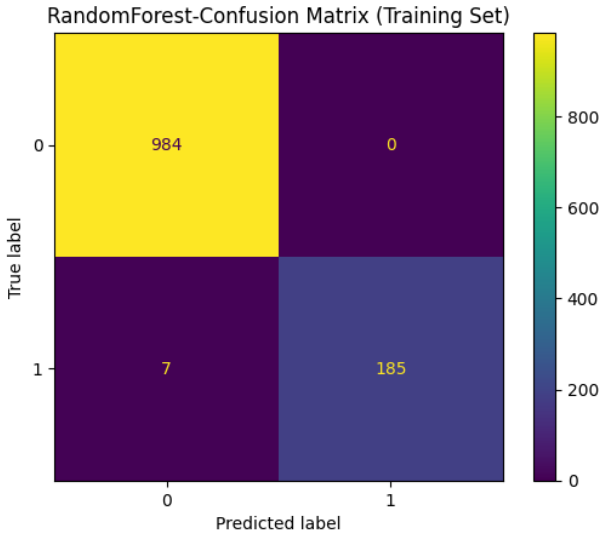


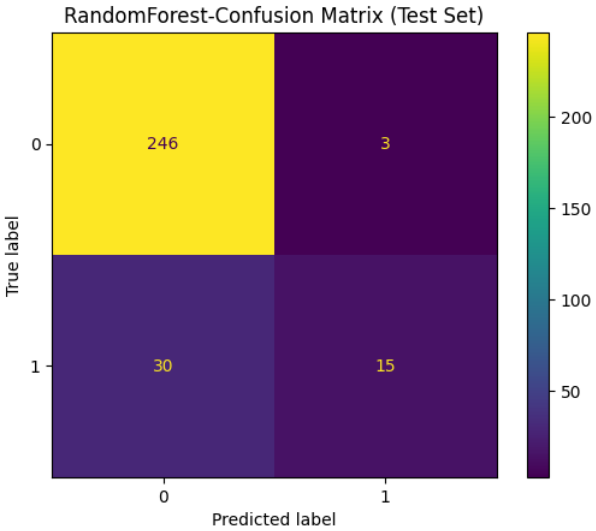
Confusion Matrix ของ CART using GridSearchCV()





Confusion Matrix ของ RandomForest





สรุปผล

จากผลลัพธ์ความแม่นยำของโมเดล Accuracy ใน Test set จะพบว่าโมเดลที่ได้ Accuracy เยอะที่สุดก็คือ RandomForestClassifier() รองลงมาเป็น CART using GridSearchCV() และ CART using GINI Criterion ตามลำดับ ในส่วนของ Training set ก็พบว่าโมเดลที่ได้ Accuracy เยอะที่สุดก็คือ CART using GINI Criterion รองลงมาเป็น CART using GridSearchCV() และ CART using GINI Criterion

จะเห็นได้ว่าโมเดล CART using GINI Criterion ที่ไม่มีการปรับจูนพารามิเตอร์ แม้ว่าจะได้ Accuracy ใน Training set สูงสุด 1.00 แต่กลับได้ Accuracy ใน Test set ต่ำที่สุด 0.80 เนื่องมาจากการที่โมเดล Overfitting ทำได้ดีแต่แค่ข้อมูลที่ใช้เรียนรู้ เมื่อพารามิเตอร์จะพบว่ามี Tree Dept เท่ากับ 14 ซึ่งเป็นความลึกของต้นไม้ ยิ่งเยอะก็ยิ่งมีแนวโน้มให้โมเดล Overfitting เรียนรู้จากข้อมูลที่ใช้เรียนเยอะจนเกินไป เมื่อเจอชุดข้อมูลใหม่จาก Test set จึงทำไม่ได้ดี พอลองเทรน CART using GridSearchCV() ที่มีการ finetuning hyperparameter หลังจากใช้ GridSearch เพื่อหาพารามิเตอร์ที่ดีที่สุด ทำให้ได้พารามิเตอร์ดังนี้

```
param_grid = {
    'max_depth': [3, 5, 7, 10, 15, None],
    'min_samples_split': [2, 5, 10, 20, 50],
    'min_samples_leaf': [1, 2, 4, 8, 16],
    'max_features': [None, 'sqrt', 'log2', 0.5, 0.8],
}
```

Max Depth	5
Min Samples Split	2
Min Samples Leaf	8
Max Feature	0.8

จะเห็นว่าได้ Max Depth หรือความลึกของต้นไม้ที่ลดลง และได้จำนวนตัวอย่างข้อมูลใน Leaf มากขึ้น โมเดลจะได้สร้าง Predictor จากข้อมูลที่มากขึ้น

จากพารามิเตอร์ดังกล่าวทำให้ Accuracy ใน Training set ลดลงเหลือ 0.88 แต่กลับได้ Accuracy ใน Test set เพิ่มขึ้นถึง 0.85 (เพิ่มขึ้น 0.05 จากเดิม)

เมื่อดูจาก Confusion matrix ของ Test set พบว่าโมเดล CART ที่ไม่ได้มีการปรับจูนพารามิเตอร์มีความสามารถในการทำนาย Positive label น้อย ทายผิดเกือบครึ่งหนึ่งของ Positive label ทั้งหมด กล่าวคือมี False negative ที่เยอะ แม้ว่าใน Training set จะทำนายถูก 100% นอกจากนี้ก็มี False Positive อยู่จำนวนหนึ่ง

ส่วน Confusion matrix ของโมเดล CART ที่มีการปรับจูนพารามิเตอร์ด้วย GridSearchCV ใน Test set นั้นมี False negative ที่เพิ่มขึ้นจากโมเดลเดิม แต่ว่ามี False positive ที่ลดลง ด้วยความที่ Dataset ของเราไม่สมดุล กล่าวคือมี Negative label ที่เยอะกว่า Positive label จำนวนมาก ด้วยเหตุนี้จึงอาจทำให้โมเดลนี้ได้ Accuracy ใน Test set เพิ่มขึ้น ส่วนใน Training set จะเห็นว่ามี False negative ที่เพิ่มขึ้นเกินครึ่งถึง 123 และพอมี False positive บ้าง จึงทำให้โมเดลนี้ได้ Accuracy ใน Training set ลดลงเหลือ 0.88

ในส่วนของ RandomForest ที่มีการปรับจูนเพื่อหาพารามิเตอร์ที่ดีที่สุดดังนี้

```
param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 5, 10, 15],
    'min_samples_split': [2, 5, 10],
    'max_features': [None, 'sqrt', 'log2', 0.5, 0.8],

}
```

Max Depth	15
Min Samples Split	5
n_estimators	100
Max Feature	0.5

เนื่องจากเป็นโมเดลที่ประกอบด้วยหลาย decision tree เพื่อใช้ ensemble learning ในการรวมผลลัพธ์การทำนายจากหลายโมเดลในการพัฒนาประสิทธิภาพของโมเดล ทำให้โมเดลมีความหลากหลายในการเรียนรู้จากข้อมูลมากขึ้น โดยมีพารามิเตอร์ที่ช่วยในการเพิ่มความหลากหลายของโมเดลก็คือ n_estimators หรือจำนวนต้นไม้ที่จะใช้ และ Max feature หรือจำนวน feature ที่ต้นไม้แต่ละต้นจะใช้ตอน split node จากผลลัพธ์จะเห็นได้ว่าโมเดลมีความสามารถในการทำนายสูงทั้ง Training set ได้ Accuracy 0.99 และ Test set ได้ Accuracy 0.89

จาก Confusion matrix ของ RandomForest กับข้อมูล Training set จะเห็นได้ว่าที่โมเดลทายผิดพลาดมีเพียงแค่ False negative 7 ครั้งเท่านั้น ในส่วนข้อมูล Test set ก็จะพบว่ายังคงมี False negative ที่เยอะไม่ต่างจาก Confustion matrix ของ CART using GridSearch แต่ว่าโมเดลพัฒนาขึ้น เนื่องจากมี False positive ที่ต่ำมากเพียงแค่ 3 ครั้ง

ข้อมูล Lab 2 practice

จากการสร้างแบบโมเดล Decision Tree ทำนายประเภทของธุรกรรมทางการเงิน ตามขั้นตอนใน Google Colab notebook lab2-decision-tree-dm.ipynb เพื่อเปรียบเทียบประสิทธิภาพกับโมเดล Logistic regression

โมเดลและพารามิเตอร์

จากการใช้ GridSearch เพื่อหาพารามิเตอร์ที่ดีที่สุด ได้ผลลัพธ์ของพารามิเตอร์ดังนี้

1) CART using GridSearchCV()

CART using GridSearchCV

```
[ ] param_grid = {
        'max_depth': [3, 5, 7, 10, 15, None],
        'min_samples_split': [2, 5, 10, 20, 50],
        'min_samples_leaf': [1, 2, 4, 8, 16],
        'max_features': [None, 'sqrt', 'log2', 0.5, 0.8],
    }
```

Max Depth	None
Min Samples Split	20
Min Samples Leaf	1
Max Feature	0.8

2) RandomForestClassifier()

▾ RandomForest

```
[ ]  param_grid_forest = {
         'n_estimators': [50, 100, 300, 500],  # Define different hyperparameters for tuning
         'max_depth': [2, 5, 10, 15, 20],
         'min_samples_split': [2, 5, 10, 15, 20],
         'max_features': ['sqrt']
     }
```

Max Depth	20
Min Samples Split	2
n_estimators	100
Max Feature	sqrt

ผลลัพธ์ของโมเดล

Dataset/Model	Logistic regression	CART using GridSearchCV()	RandomForestClassifier()
Test set	0.84	**0.87**	0.82

สรุปผล

จากการทดลองใช้ decision tree model มาเปรียบกับ Logistic regression model พบว่าจากการปรับจูนพารามิเตอร์ด้วย GridSearch ดังที่กล่าวไปข้างต้น โมเดลที่มีความแม่นยำสูงสุดคือ CART using GridSearchCV() 0.87 รองลงมาเป็น RandomForest และ Logistic regression ตามลำดับ