Data learning: Linear Model vs Classification Tree

Example 1: Single Outcome Variable and Two Covariates

Data source: Simulation by R or Python programming language

Data:

Study Hours	%Attendance	Outcome	Study Hours	%Attendance	Outcome	Study Hours	%Attendance	Outcome
7	90	Pass	8	89	Pass	1	46	Fail
4	46	Fail	8	97	Pass	6	83	Pass
8	60	Fail	3	48	Fail	9	47	Fail
5	48	Fail	6	65	Fail	1	86	Fail
7	78	Pass	5	92	Fail	3	74	Fail
3	57	Fail	2	41	Fail	7	53	Fail
7	43	Fail	8	59	Fail	4	56	Fail
8	64	Fail	6	67	Fail	9	75	Pass
5	99	Fail	2	86	Fail	3	89	Fail
4	53	Fail	5	99	Fail	5	79	Fail

Step 1: Formulate your research question based on this data set.

Are study hours and class attendance important factors classifying students' course achievements?

Step 2: Define data role and data type.

Variable name	Data role	Data type
Study hours	ตัวแปรอิสระ (Predictor)	เชิงปริมาณ (Continuous)
Class attendance	ตัวแปรอิสระ (Predictor)	เชิงปริมาณ (Continuous)
Course achievement outcome	ตัวแปรตาม (Response)	เชิงคุณภาพแบบสองกลุ่ม (Binary: Pass/Fail)

Step 3: Select an appropriate method to analyze the data e.g., statistical learning, machine learning.

Binary logistic regression is used to analyze the data because…

ใช้สำหรับการวิเคราะห์ตัวแปรตามที่มีค่าเป็นแบบสองกลุ่ม เช่น "ผ่าน" หรือ "ไม่ผ่าน"

Classification tree is used to analyze the data because…

ใช้สำหรับการสร้างกฎการตัดสินใจที่สามารถตีความได้ง่าย และเหมาะกับความสัมพันธ์เชิงไม่เชิงเส้นระหว่างตัวแปร

Step 4: Collect data.

A simulated data set of 30 students from R or Python programming language is used for this example as shown in above table.

Step 5: Explore your data using numerical summary and graphs

(Hint: explore data separated by outcome class: see Iris flower data as an example)

Descriptive statistics:

สถิติเชิงพรรณนา:

	Study Hours				% Attendance			
	mean	std	min	max	mean	std	min	max
Outcome								
Fail	4.75	2.288915	1	9	64.875000	18.601806	41	99
Pass	7.50	1.048809	6	9	85.333333	8.213810	75	97

กลุ่มที่ "Fail" (**สอบตก**)

1. Study Hours (**ชั่วโมงการเรียน**):

ค่าเฉลี่ย (**mean**) = 4.75 ชั่วโมง นักเรียนในกลุ่มนี้มีจำนวนชั่วโมงการเรียนเฉลี่ยต่ำกว่ากลุ่ม "Pass"

ส่วนเบี่ยงเบนมาตรฐาน (**std**) = 2.29 ชั่วโมงแสดงว่าชั่วโมงการเรียนในกลุ่มนี้มีการกระจายตัวมากกว่า

ค่าน้อยที่สุด (**min**) = 1 ชั่วโมง ค่าสูงสุด (**max**) = 9 ชั่วโมง

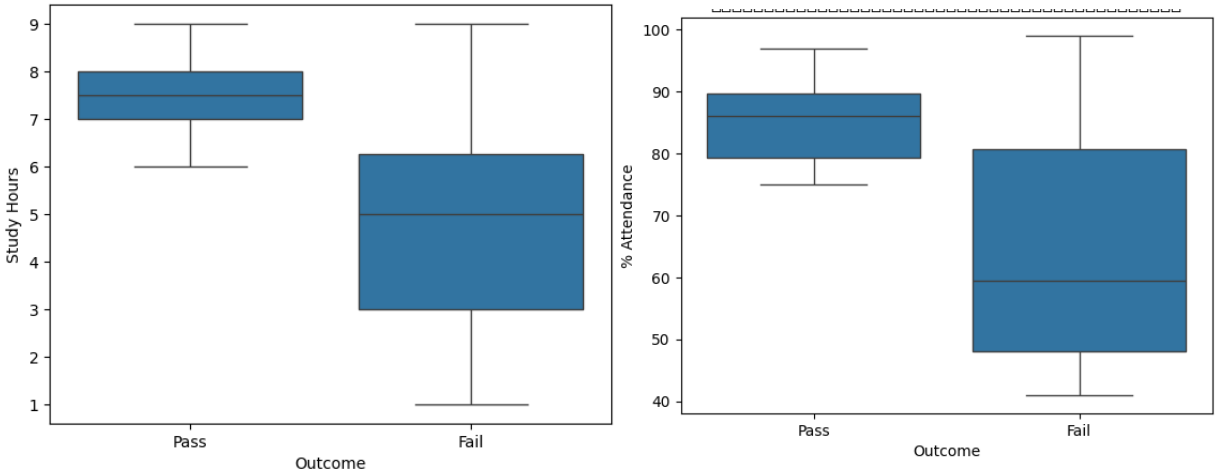
2. % Attendance (**เปอร์เซ็นต์การเข้าชั้นเรียน**):

ค่าเฉลี่ย (**mean**) = 64.88%แสดงว่ากลุ่มนี้มีการเข้าชั้นเรียนเฉลี่ยน้อยกว่ากลุ่ม "Pass"

ส่วนเบี่ยงเบนมาตรฐาน (**std**) = 18.60%แสดงว่าการกระจายตัวของเปอร์เซ็นต์การเข้าชั้นเรียนในกลุ่มนี้ค่อนข้างกว้าง

ค่าน้อยที่สุด (**min**) = 41% ค่าสูงสุด (**max**) = 99%

Show your graphs:



Interpret:

กลุ่มที่สอบผ่าน จะใช้เวลาเรียนในช่วง 7-8 ชั่วโมง กลุ่มที่สอบตก จะใช้เวลาเรียนในช่วง 3-6 ชั่วโมง

กลุ่มที่สอบผ่าน จะใช้เวลาเรียนในช่วง 7-8 ชั่วโมง กลุ่มที่สอบตก จะใช้เวลาเรียนในช่วง 3-6 ชั่วโมง

Binary logistic model

Step 6: Fit the model.

(Hint: t-test, chi-squared test, likelihood ratio test, logistic regression model)

```
T-test for Study Hours between Pass and Fail: TtestResult(statistic=-2.8401877872187726, pvalue=0.008308034560870627, df=28.0)
T-test for Attendance between Pass and Fail: TtestResult(statistic=-2.6039737607876763, pvalue=0.014579995455252586, df=28.0)
```

T-test ใช้ในการทดสอบว่าค่าเฉลี่ยของ **Study Hours** และ **% Attendance** แตกต่างกันระหว่างผู้ที่ "Pass" และ "Fail" หรือไม่

ผลลัพธ์ที่ได้จะให้ค่า **t-statistic** และ **p-value** ซึ่งถ้า p-value น้อยกว่า 0.05 แสดงว่าแตกต่างกันอย่างมีนัยสำคัญทางสถิติ

สรุปว่า ค่าเฉลี่ยของ **Study Hours** แตกต่างกันระหว่างผู้ที่ "Pass" และ "Fail"

ค่าเฉลี่ยของ **% Attendance** ไม่แตกต่างกันระหว่างผู้ที่ "Pass" และ "Fail"

```
Chi-squared test result:
Chi2: 8.958333333333334 p-value: 0.34582310682623596
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.000003
         Iterations: 35
```

Chi-squared test ใช้ทดสอบว่า **Study Hours** มีความสัมพันธ์กับ **Outcome** หรือไม่

ถ้า p-value มากกว่า 0.05 แสดงว่า **Study Hours** ไม่มีความสัมพันธ์กัน **Outcome**

```
Likelihood Ratio Test Summary:
                           Logit Regression Results
==============================================================================
Dep. Variable:                Outcome   No. Observations:                   21
Model:                          Logit   Df Residuals:                       18
Method:                           MLE   Df Model:                            2
Date:                Thu, 19 Dec 2024   Pseudo R-squ.:                   1.000
Time:                        17:42:29   Log-Likelihood:            -6.6674e-05
converged:                      False   LL-Null:                       -10.225
Covariance Type:            nonrobust   LLR p-value:                 3.625e-05
==============================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       -8851.2739   1.02e+05     -0.087      0.931   -2.09e+05    1.91e+05
Study Hours   594.8801   6858.725      0.087      0.931   -1.28e+04     1.4e+04
% Attendance   63.7627    734.841      0.087      0.931   -1376.499    1504.025
==============================================================================
```

p-value น้อยกว่า 0.05 แสดงว่าโมเดลที่มีตัวแปรอิสระดีกว่าโมเดลที่ไม่มีตัวแปร

Step 8: Evaluate model accuracy.

เปรียบเทียบค่าความแม่นยำ (Accuracy), Precision, Recall, และ F1 Score ระหว่างสองโมเดล

```
Confusion Matrix:
 [[5 2]
 [1 1]]
```

True Negatives (TN): 5 (จำนวนที่ทำนาย "Fail" และจริงๆ เป็น "Fail")

False Positives (FP): 2 (จำนวนที่ทำนาย "Pass" แต่จริงๆ เป็น "Fail")

False Negatives (FN): 1 (จำนวนที่ทำนาย "Fail" แต่จริงๆ เป็น "Pass")

True Positives (TP): 1 (จำนวนที่ทำนาย "Pass" และจริงๆ เป็น "Pass")

1. **Precision**:

Precision (0): 0.83 หมายถึง จากทั้งหมดที่ทำนายว่าเป็น "Fail" 83% เป็นจริง "Fail"

Precision (1): 0.33 หมายถึง จากทั้งหมดที่ทำนายว่าเป็น "Pass" 33% เป็นจริง "Pass"

2. **Recall**:

Recall (0): 0.71 หมายถึง จากทั้งหมดที่จริงๆ เป็น "Fail" 71% ถูกทำนายถูกต้องว่าเป็น "Fail"

Recall (1): 0.50 หมายถึง จากทั้งหมดที่จริงๆ เป็น "Pass" 50% ถูกทำนายถูกต้องว่าเป็น "Pass"

3. **F1-Score**:

F1-Score (0): 0.77 คือค่ารวมระหว่าง Precision และ Recall สำหรับ "Fail"

F1-Score (1): 0.40 คือค่ารวมระหว่าง Precision และ Recall สำหรับ "Pass"

4. **Accuracy**: 0.67 หมายถึง โมเดลนี้ทำนายถูกต้องประมาณ 67% ของกรณีทั้งหมด

5. **Macro Average**:

Macro avg: คำนวณค่าเฉลี่ยของ Precision, Recall, และ F1-Score โดยไม่พิจารณาขนาดของแต่ละคลาส

Precision: 0.58

Recall: 0.61

F1-Score: 0.58

6. **Weighted Average**:

Weighted avg: คำนวณค่าเฉลี่ยของ Precision, Recall, และ F1-Score โดยมีการนำขนาดของแต่ละคลาสมาพิจารณา

Precision: 0.72

Recall: 0.67

F1-Score: 0.69

Step 9: Interpret the results.

สรุป:

โมเดลมีความแม่นยำ (accuracy) รวมที่ 67% ซึ่งแสดงว่าโมเดลสามารถทำนายได้ถูกต้องในหลายกรณี แต่ยังมีความผิดพลาดในบางส่วน โดยเฉพาะในกรณีของการทำนาย "Pass" ซึ่งมี **Precision** และ **Recall** ค่อนข้างต่ำ

ค่า **F1-Score** สำหรับ "Pass" (0.40) แสดงให้เห็นว่าโมเดลอาจต้องการการปรับปรุงในการทำนายผลลัพธ์ในกลุ่มนี้

ผลลัพธ์ของ **Macro avg** และ **Weighted avg** แสดงให้เห็นว่าแม้ว่าความแม่นยำจะไม่สูง แต่ก็ยังมีการทำนายที่ดีในบางส่วน (เช่น "Fail")

****

Show your R or Python programming language for binary logistic regression.

https://colab.research.google.com/drive/1QIpWoLsdWkltsiOPEFTGVPOOLGuInlIa?usp=sharing

Classification tree

Step 6: Fit the model.

To understand the mathematical fundamentals behind classification trees, we'll break it down into the following key components:

At each split, the Gini Index and Entropy help measure the impurity of the resulting groups.

A lower Gini Index or Entropy indicates purer groups.

Both measures are used to determine the best split by comparing the weighted impurity before and after the split.

Splitting stops when all nodes are pure or meet stopping criteria (same as in regression tree).

Gini Index Formula

For a node containing observations from k classes, the Gini Index is defined as:

$$G = 1 - \sum_{i=1}^{k} p_i^2$$

Where:

- p_i is the proportion of observations in class i within the node.
- k is the total number of classes.

Entropy Formula

For a node containing observations from k classes, the Entropy is defined as:

$$H = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

Where:

- p_i is the proportion of observations in class i within the node.
- k is the total number of classes.
- $\log_2$ is the logarithm base 2.

Weighted Impurity Formula

To calculate the overall impurity of a split, we compute the **weighted impurity** for all child nodes. The formula applies to both Gini Index and Entropy.

$$\text{Weighted Impurity} = \sum_{j=1}^{m} \frac{n_j}{n} \cdot I_j$$

Where:

- m: Number of child nodes (typically 2 for binary splits).
- n_j: Number of observations in child node j.
- n: Total number of observations in the parent node.
- I_j: Impurity measure (Gini Index or Entropy) for child node j.

Show your calculation:

$$Gini = 1 - (0.6^2 + 0.4^2) = 1 - (0.36 + 0.16) = 1 - 0.52 = 0.48$$

$$Entropy = -(0.6 \log_2 0.6 + 0.4 \log_2 0.4) = -(0.6 \times (-0.737) + 0.4 \times (-1.322)) = 0.971$$

$$\text{Weighted Impurity} = \sum \left(\frac{\textbf{จำนวนตัวอย่างในกลุ่ม}}{\textbf{จำนวนข้อมูลทั้งหมด}} \times \text{Impurity } \textbf{ของกลุ่ม} \right)$$

คำนวณ:

1. **Pass:** $\frac{18}{30} \times 0.44 = 0.264$
2. **Fail:** $\frac{12}{30} \times 0.5 = 0.2$

*** if your classification tree is overfitted, then prune it. ***

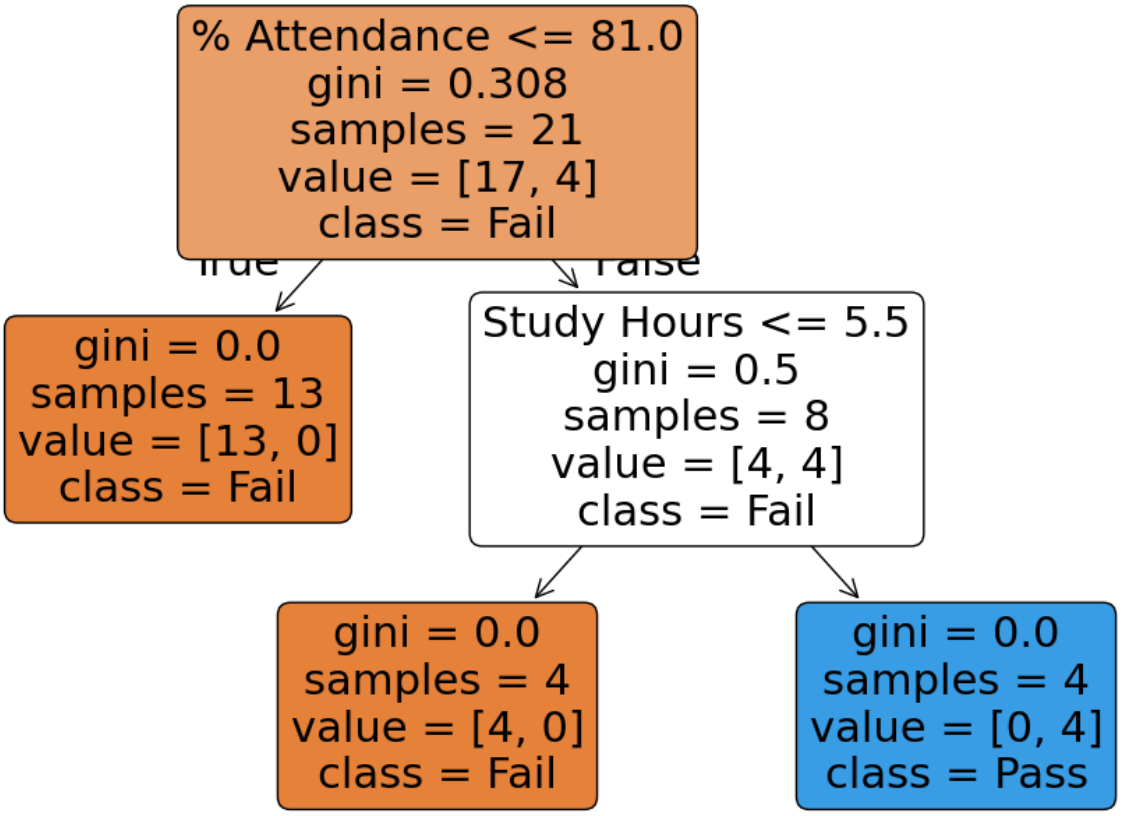
Step 7: Interpret the results.

ค่า Gini Index = 0.48 แสดงว่าชุดข้อมูลในตอนนี้มีความไม่บริสุทธิ์ระดับกลาง ๆ ไม่ได้เป็นกลุ่มที่บริสุทธิ์และก็ไม่ถึงกับกระจายอย่างเต็มที่

Entropy = 0.971 แสดงว่าข้อมูลในชุดนี้มีความไม่แน่นอนหรือความหลากหลายสูง ซึ่งหมายความว่าคลาสต่าง ๆ ในชุดข้อมูลมีการกระจายตัวอย่างใกล้เคียงกัน หรือมีความไม่แน่นอนในการทำนายผล (ไม่สามารถแยกข้อมูลได้ดี)

Weighted Impurity ของชุดข้อมูลนี้คือ **0.464** ซึ่งแสดงถึงค่าความไม่บริสุทธิ์ของข้อมูลหลังจากแบ่งเป็นกลุ่ม "Pass" และ "Fail"

Draw your classification tree:



Interpret the classification results based on decision rules (explain your tree structure):

Classification Report:

Precision: เป็นอัตราส่วนของการทำนายเป็นบวก (positive) ที่ถูกต้อง (True Positives / (True Positives + False Positives)) โดยคำนวณแยกตามแต่ละคลาส:

Precision for Class 0 (Fail) = 0.78: หมายความว่า 78% ของการทำนายว่า "Fail" นั้นถูกต้องจริง

Precision for Class 1 (Pass) = 0.00: ไม่มีการทำนายที่ถูกต้องสำหรับคลาส "Pass" ซึ่งมีค่าเป็น 0%

Recall: เป็นอัตราส่วนของข้อมูลจริงที่ทำนายได้ถูกต้อง (True Positives / (True Positives + False Negatives)) โดยคำนวณแยกตามแต่ละคลาส:

Recall for Class 0 (Fail) = 1.00: 100% ของข้อมูลที่แท้จริงว่า "Fail" ถูกทำนายได้ถูกต้อง

Recall for Class 1 (Pass) = 0.00: ไม่มีข้อมูลที่แท้จริงว่า "Pass" ถูกทำนายถูกต้อง

F1-Score: เป็นค่าเฉลี่ยของ Precision และ Recall (2 * Precision * Recall) / (Precision + Recall) โดยคำนวณแยกตามแต่ละคลาส:

F1-Score for Class 0 (Fail) = 0.88: ค่า F1-Score สูงแสดงว่าโมเดลทำนาย "Fail" ได้ดี

F1-Score for Class 1 (Pass) = 0.00: ค่า F1-Score เป็น 0 เพราะไม่มีการทำนายที่ถูกต้องสำหรับ "Pass"

Accuracy: เป็นอัตราส่วนของการทำนายทั้งหมดที่ถูกต้อง (True Positives + True Negatives) / จำนวนข้อมูลทั้งหมด:

Accuracy = 0.78: หมายความว่าโมเดลทำนายได้ถูกต้อง 78% ของข้อมูลทั้งหมด

Macro Average: เป็นค่าเฉลี่ยของ Precision, Recall, และ F1-Score สำหรับทุกคลาส โดยไม่สนใจสัดส่วนของข้อมูลในแต่ละคลาส:

Macro avg Precision = 0.39

Macro avg Recall = 0.50

Macro avg F1-Score = 0.44

Weighted Average: ค่าเฉลี่ยที่นำสัดส่วนของข้อมูลในแต่ละคลาสมาคำนวณ:

Weighted avg Precision = 0.60

Weighted avg Recall = 0.78

Weighted avg F1-Score = 0.68

Confusion Matrix:

True Negatives (TN) = 7: จำนวนตัวอย่างที่เป็นคลาส "Fail" และทำนายว่าเป็น "Fail" ถูกต้อง

False Positives (FP) = 0: จำนวนตัวอย่างที่เป็น "Fail" แต่ทำนายว่าเป็น "Pass" (ไม่มีในกรณีนี้)

False Negatives (FN) = 2: จำนวนตัวอย่างที่เป็น "Pass" แต่ทำนายว่าเป็น "Fail"

True Positives (TP) = 0: จำนวนตัวอย่างที่เป็น "Pass" และทำนายว่าเป็น "Pass" (ไม่มีในกรณีนี้)

Model selection/comparison: Logistic Regression VS Classification Tree

by comparing overall accuracy, precision, recall, and F1 metrics.

Create a confusion matrix for Logistic Regression

	Observation	
Prediction	Positive	Negative
Positive	5	2
Negative	1	1

Create a confusion matrix for Classification Tree

Prediction	Observation	
	Positive	Negative
Positive	7	0
Negative	2	0

Performance metric	Model	
	Logistic Regression	Classification Tree
Overall accuracy	0.78	0.78
Precision	0.78	0
Recall	1	0
F1	0.88	0

Which model would you select for classification?

เลือก Logistic Regression เป็นโมเดลที่ดีกว่าในกรณีนี้เนื่องจากมีความแม่นยำและความสามารถในการตรวจจับ **Positive** ที่ดีกว่า **Classification Tree**.

****

Show your R or Python programming language for classification tree.

https://colab.research.google.com/drive/1QIpWoLsdWkltsiOPEFTGVPOOLGuInlIa?usp=sharing