Logistic Regression

Kittirak Moungmingsuk

Logistic Regression

- แม้ชื่อจะลงท้ายด้วย **Regression** แต่เป็นเครื่องมือที่สำคัญของ **Classification** (การพยากรณ์ว่าข้อมูลจะตกอยู่ในกลุ่มไหน)
- โมเดลจะพยากรณ์ ค่าความน่าจะเป็น **(P)** เช่น โอกาสที่ลูกค้าจะซื้อของ **= 0.85**
- ใช้ **Logit function** เพื่อบีบผลลัพธ์ให้อยู่ในช่วง **0 - 1** เสมอ
- เป็น **Baseline Model** ที่ดี
  - เวลาทำโปรเจกต์ Data Science เรามักจะเริ่มที่ Logistic Regression ก่อน เพื่อดูว่า "โจทย์นี้ยากแค่ไหน"
  - ถ้าโมเดลพื้นฐานอย่าง Logistic ทำแม่นยำได้ 80% แล้ว และโมเดลซับซ้อนทำได้ 82% เราอาจจะเลือกใช้ Logistic เพราะดูแลง่ายกว่าและอธิบายได้ดีกว่า

Pros/Cons

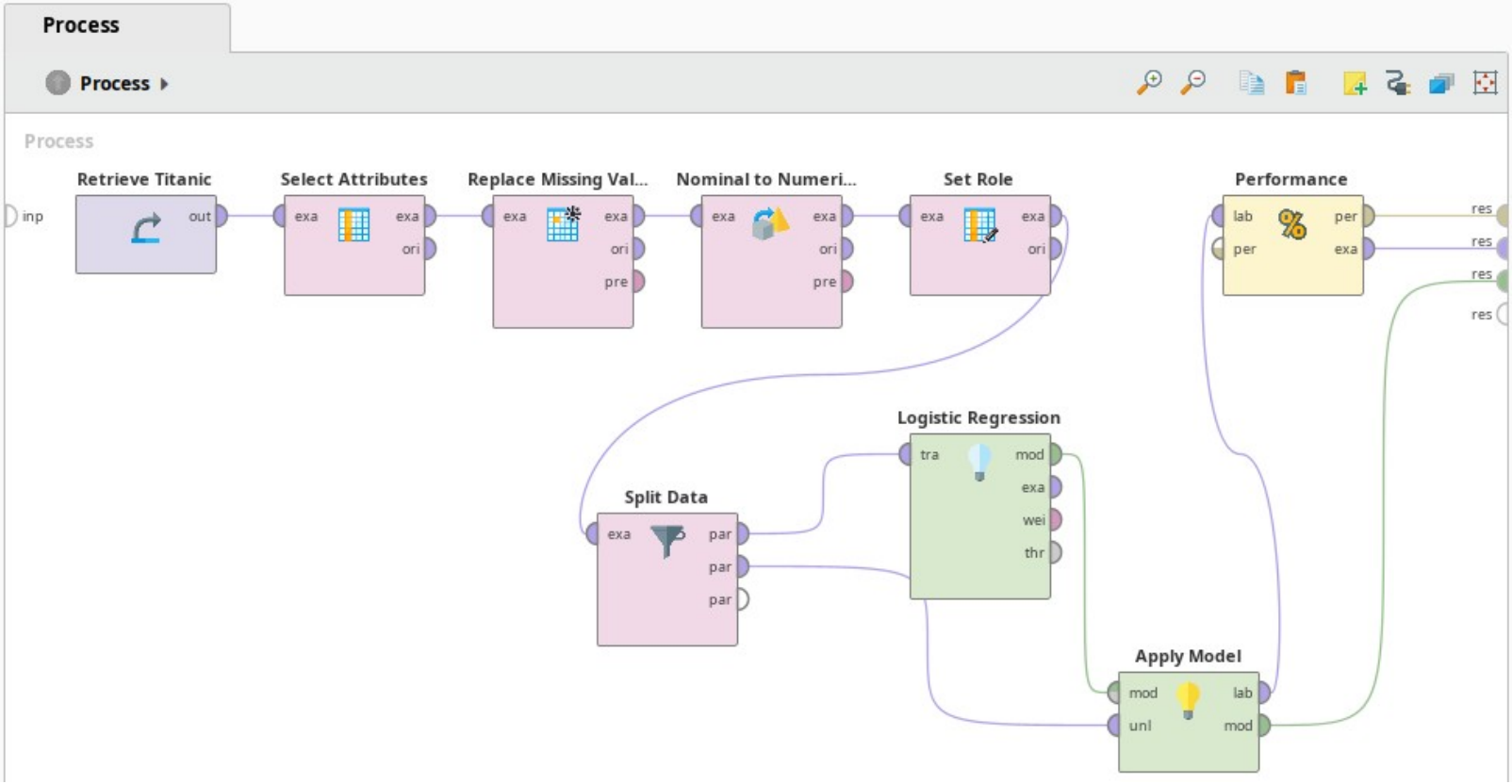
ข้อดี

- ให้ค่าความน่าจะเป็น **(Probability)** ทำให้ธุรกิจนำไปจัดลำดับความสำคัญได้ (เช่น โทรหาลูกค้าที่มีโอกาสซื้อสูงสุดก่อน)
- ความโปร่งใส **(Interpretability):** อธิบายได้ว่าปัจจัยแต่ละตัวส่งผลต่อการตัดสินใจอย่างไร
- ทำงานได้เร็วและใช้ทรัพยากรน้อย

ข้อจำกัด

- สันนิษฐานว่าความสัมพันธ์ระหว่างตัวแปรเป็นเส้นตรง (ในรูปแบบ **Logit)**
- ไม่เก่งในการจัดการกับข้อมูลที่มีความซับซ้อนสูง หรือมีความสัมพันธ์แบบ **Non-linear**

Logistic Regression (Titanic data) **ทำหน้านี้ ให้เวลาถึง 09.20



Confusion Matrix

accuracy: 79.64%

	true Yes	true No	class precision	precision
pred. Yes	80	10	88.89%	80/(80+10)
pred. No	70	233	76.90%	233/(70+233)
class recall	53.33%	95.88%		

Recall	80/(80+70)	233/(10+233)

Accuracy (80+233)/(80+10+70+233) = 79.64%

ExampleSet (Apply Model)

Row No.	Survived	prediction(Survived)	confidence(Yes)	confidence(No)
1	Yes	Yes	0.725	0.275
2	No	Yes	0.965	0.035
3	No	Yes	0.925	0.075
4	No	No	0.176	0.824
5	Yes	Yes	0.923	0.077
6	No	No	0.558	0.442
7	Yes	Yes	0.847	0.153
8	Yes	Yes	0.901	0.099
9	No	No	0.509	0.491
10	No	No	0.343	0.657
11	Yes	Yes	0.913	0.087
12	Yes	No	0.467	0.533

บรรทัดที่ 6 ค่า confidence(Yes) สูงกว่า แต่ทำไมทำนาย No

Threshold

- ลากเส้น **threshold** จากโมเดลไปที่ **result** เพื่อดูผล

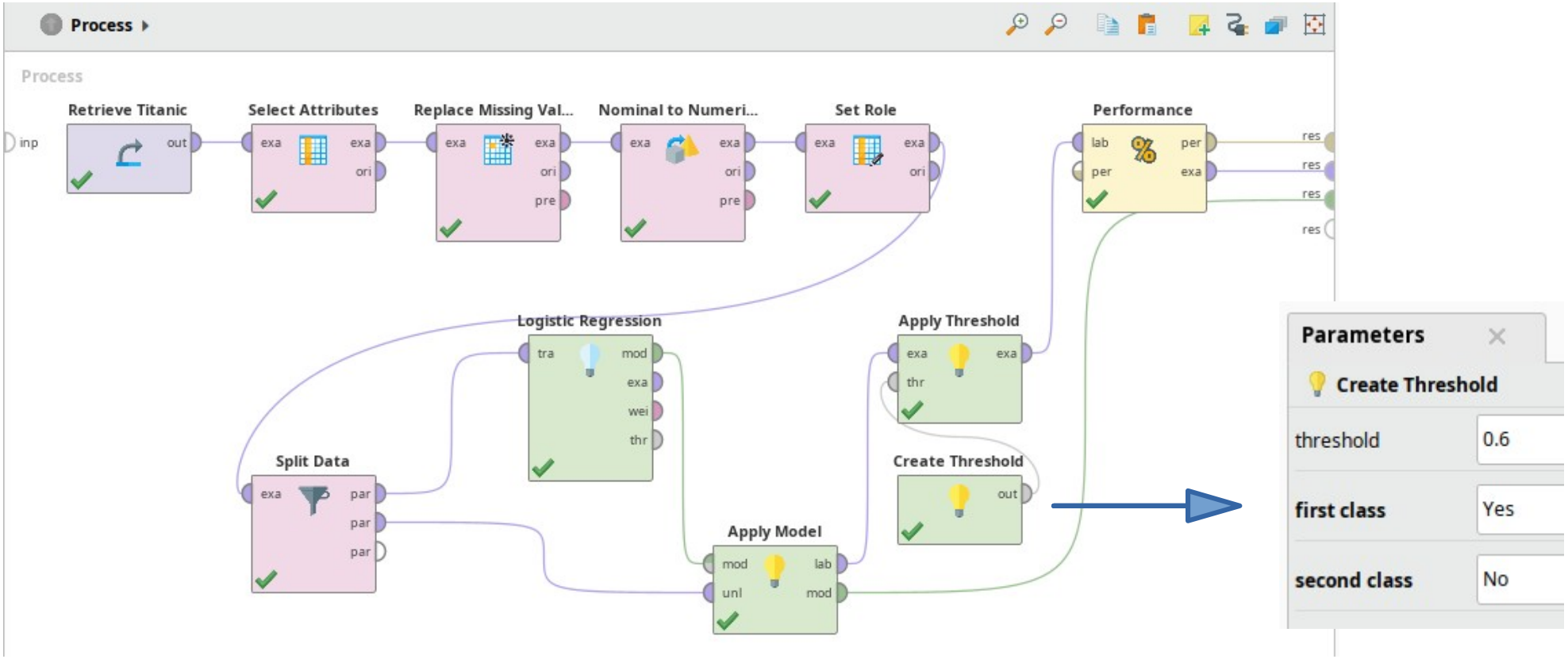
```
                    Threshold
Threshold: 0.35272673883795336
first class: Yes
second class: No
if confidence(No) > 0.35272673883795336 then No
else Yes
```

- ถ้า **confidence(No)** มากกว่า **0.352** จะทำนายว่า **No (**เกณฑ์คือจุดนี้**)**

ทำอย่างไรให้ **model** แม่นยำขึ้น

- ก่อนรันโมเดล
  - Data cleansing
  - Select attribute
- **Cross validation**
- **Threshold optimization**
- เปลี่ยนโมเดล
- ใช้หลายโมเดลผสมกัน **(ensemble model)**

ปรับ Threshold (เพิ่ม Create Threshold and Apply Threshold)



After create threshold

- Accuracy เพิ่มขึ้น

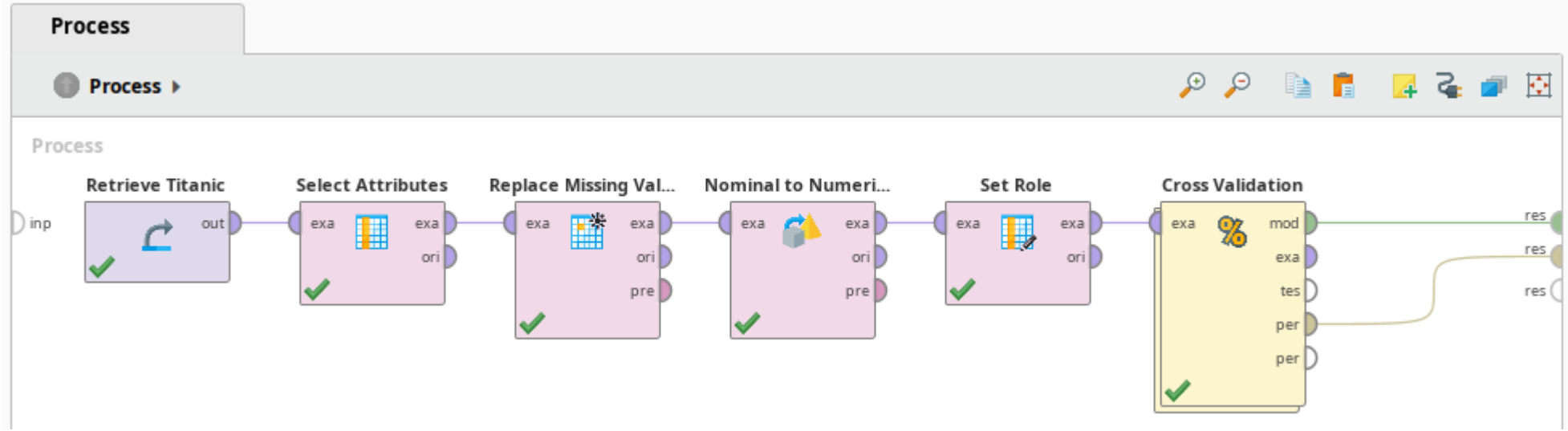
accuracy: 80.41%

	true Yes	true No	class precision
pred. Yes	120	47	71.86%
pred. No	30	196	86.73%
class recall	80.00%	80.66%	

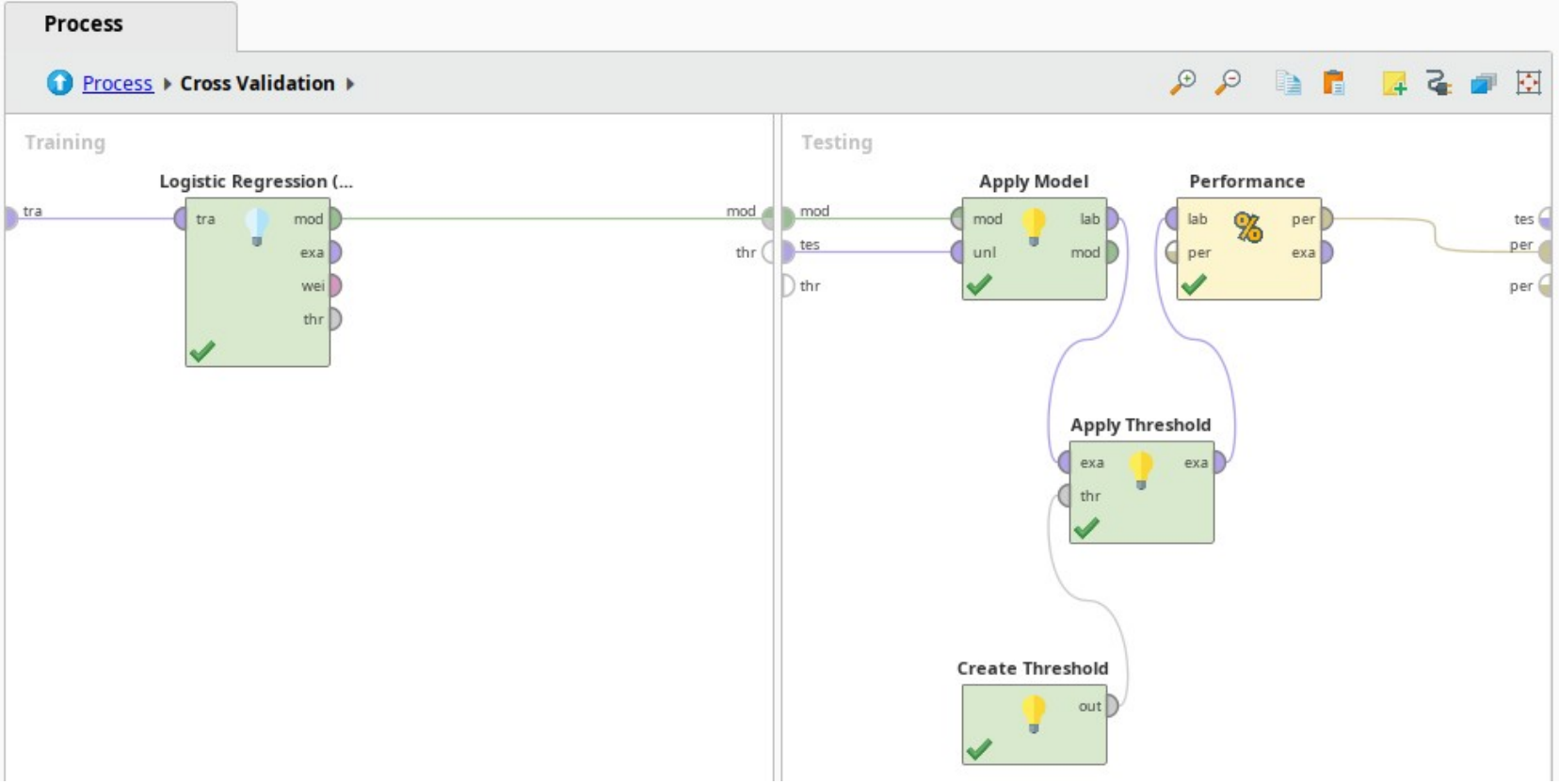
K-Fold Cross Validation

Image from: https://www.researchgate.net/figure/Repeated-K-Fold-Cross-Validation-A-repeated-10-fold-CV-was-applied-The-10-fold-CV-works_fig1_328798891

Cross Validation



In Cross Validation



No create threshold Vs. threshold = 0.6

accuracy: 78.00% +/- 3.28% (micro average: 78.00%)

No create threshold

	true Yes	true No	class precision
pred. Yes	266	54	83.12%
pred. No	234	755	76.34%
class recall	53.20%	93.33%	

accuracy: 76.93% +/- 4.73% (micro average: 76.93%)

Create threshold = 0.6

	true Yes	true No	class precision
pred. Yes	380	182	67.62%
pred. No	120	627	83.94%
class recall	76.00%	77.50%	

อ่านค่า

accuracy: 78.00% +/- 3.28% (micro average: 78.00%)

	true Yes	true No	class precision
pred. Yes	266	54	83.12%
pred. No	234	755	76.34%
class recall	53.20%	93.33%	

- **78.00% +/- 3.28%** ที่ได้จาก **Cross Validation** คือ "ค่าเฉลี่ยความจริง"
- ระบบแบ่งข้อมูลเป็น **10** ส่วน แล้ววนทำ **10** รอบ
- บางรอบอาจได้ **81%** บางรอบอาจได้ **75%**
- ค่า **+/- 3.28% (Standard Deviation)** บอกเราว่าโมเดลนี้มีความแกว่งอยู่บ้าง ไม่ได้นิ่งสนิท แต่เป็นค่าที่แม่นยำกว่าเมื่อนำไปใช้จริงกับข้อมูลใหม่

ส่งท้าย

- การเตรียมข้อมูล **(Data Preparation)**
- การเลือก **attribute** เช่นการอ่านค่า **Coefficients**
- การปรับเกณฑ์การตัดสินใจ **(Threshold Optimization)**
- ความน่าเชื่อถือของโมเดล **(Evaluation)**
- **Challenge :** ทดลองใช้ **Decision tree** กับข้อมูลชุดนี้ แล้วดูว่าใครแม่นกว่ากัน