สรุปเนื้อหาสอบปลายภาค

CPE232: Data Models

Chapter 6: ML Classification

Chapter 7: ML Regression

Chapter 8: Model Parameter Tuning

Chapter 9: ML Clustering

Dr. Sansiri Tarnpradab

Department of Computer Engineering, KMUTT

สารบัญ (Table of Contents)

บทที่ 6: ML Classification ..... *หน้า 3*

บทที่ 7: ML Regression ..... *หน้า 7*

บทที่ 8: Model Parameter Tuning ..... *หน้า 11*

บทที่ 9: ML Clustering ..... *หน้า 15*

ตาราง Cheat Sheet สูตรทั้งหมด ..... *หน้า 19*

แนวข้อสอบ - บทที่ 6 ..... *หน้า 21*

แนวข้อสอบ - บทที่ 7 ..... *หน้า 24*

แนวข้อสอบ - บทที่ 8 ..... *หน้า 27*

แนวข้อสอบ - บทที่ 9 ..... *หน้า 30*

บทที่ 6: ML Classification (การจำแนกประเภทด้วย Machine Learning)

6.1 ภาพรวม (Overview)

Classification คือการทำนายว่าข้อมูลที่ป้อนเข้ามาอยู่ในกลุ่ม/ประเภทใด โดยโมเดลเรียนรู้จาก labeled data (Supervised Learning)

- Input: Feature vector (x_1, x_2, ..., x_n)
- Output: Class label (discrete category)
- เปรียบเทียบกับ Regression ที่ output เป็น continuous value

6.2 Decision Tree

ต้นไม้การตัดสินใจที่แบ่งข้อมูลตาม feature ที่ให้ข้อมูลมากที่สุด

Entropy (ความไม่แน่นอนของข้อมูล)

$$Entropy(t) = -\sum_j p(j|t) \cdot \log_2 p(j|t)$$

- p(j|t) = สัดส่วนของ class j ใน node t
- Entropy = 0 ⟶ node บริสุทธิ์ (pure node), Entropy สูงสุด = 1 (ข้อมูลปนกันเท่ากัน)

Information Gain (Gain_split)

$$Gain_split = Entropy(parent) - \sum_i (n_i/n) \cdot Entropy(child_i)$$

- n_i = จำนวน sample ใน child node i, n = จำนวน sample ทั้งหมด
- เลือก feature ที่ให้ Gain สูงสุดมาแบ่ง node

ตัวอย่าง Information Gain จากบทเรียน

Feature	Information Gain
age	0.246
income	0.028
student	0.131
credit_rating	0.047

⟶ เลือก age ก่อนเพราะ Gain สูงสุด (0.246)

Gini Index

$$Gini(t) = 1 - \sum_j [p(j|t)]^2$$

- ใช้แทน Entropy ได้ • Gini = 0 ⟶ pure node

6.3 K-Nearest Neighbors (KNN)

- จำแนก sample ใหม่โดยดูจาก K เพื่อนบ้านที่ใกล้ที่สุด
- K น้อยเกินไป ⟶ Overfitting, K มากเกินไป ⟶ Underfitting
- ต้องเลือก distance metric: Euclidean, Manhattan ฯลฯ

6.4 Naive Bayes

$$P(C|X) \propto P(C) \cdot \prod P(x_i|C)$$

- สมมติว่า features เป็นอิสระต่อกัน (naive assumption)
- ทำงานได้ดีกับข้อมูล text classification

6.5 Logistic Regression

$$P(y=1|x) = 1 / (1 + e^{(-(wx + b))})$$

- ใช้ Sigmoid function แปลงเป็นความน่าจะเป็น (0–1)
- Decision boundary: P ≥ 0.5 ⟶ class 1

6.6 Support Vector Machine (SVM)

- หา hyperplane ที่แบ่ง class โดยมี margin สูงสุด
- Support Vectors = จุดข้อมูลที่ใกล้ hyperplane มากที่สุด
- Kernel trick: ขยายมิติข้อมูลเพื่อแก้ปัญหา non-linear

6.7 Random Forest

- Ensemble method: รวม Decision Trees หลายต้นเข้าด้วยกัน
- ใช้ Bagging (Bootstrap Aggregating) ⟶ สุ่มข้อมูล + สุ่ม features
- ลด Overfitting ได้ดีกว่า Decision Tree เดี่ยว

6.8 การประเมินโมเดล Classification

Confusion Matrix

	Predicted: Yes	Predicted: No
Actual: Yes	TP (True Positive)	FN (Type II Error)
Actual: No	FP (Type I Error)	TN (True Negative)

เมตริกการประเมิน

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Recall (Sensitivity) = TP / (TP + FN)

Precision = TP / (TP + FP)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

- Accuracy: ความถูกต้องโดยรวม
- Recall: ครอบคลุม positive ได้มากแค่ไหน (สำคัญเมื่อ FN มีต้นทุนสูง เช่น โรคมะเร็ง)
- Precision: สิ่งที่ทำนายว่า positive นั้นถูกต้องแค่ไหน
- F1: ค่าเฉลี่ยแบบ harmonic mean ของ Precision และ Recall
- ROC Curve / AUC: วัดประสิทธิภาพโมเดล binary classification

บทที่ 7: ML Regression (การพยากรณ์ค่าต่อเนื่อง)

7.1 ภาพรวม

Regression ทำนาย numerical value (continuous output) จาก features

- สมการหลัก: Data = Model + Error
- Independent variables (x) = Predictors
- Dependent variable (y) = Outcome

7.2 Linear Regression

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

- w_0 = intercept (bias), w_i = coefficients (weights)
- หา w ที่ minimize loss function (MSE)

7.3 Polynomial Regression

$$\hat{y} = w_0 + w_1x + w_2x^2 + \dots + w_nx^n \quad (\text{degree } n > 1)$$

- ใช้เมื่อความสัมพันธ์ระหว่าง x และ y ไม่ใช่เส้นตรง
- degree สูงมาก ⟶ Overfitting

7.4 Loss Functions

Mean Squared Error (MSE)

$$MSE = (1/n) \Sigma_i (y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{[(1/n) \Sigma_i (y_i - \hat{y}_i)^2]}$$

Mean Absolute Error (MAE)

$$MAE = (1/n) \Sigma_i |y_i - \hat{y}_i|$$

Metric	สูตร	ลักษณะ
MSE	$(1/n)\Sigma(y-\hat{y})^2$	ให้น้ำหนัก error ใหญ่มาก (squared)
RMSE	√MSE	หน่วยเดียวกับ y, ตีความง่าย
MAE	$(1/n)\Sigma\lvert y-\hat{y}\rvert$	ทนทานต่อ outlier กว่า MSE

7.5 Gradient Descent

วิธีอัปเดต weight เพื่อลด loss function ทีละ step

$$\text{Cost function: } J(w, b) = (1/n)\ \Sigma_i\ (y_i - \hat{y}_i)^2$$

$$\text{Update rule: } w = w - \alpha \cdot (\partial J/\partial w)$$

$$\text{Update rule: } b = b - \alpha \cdot (\partial J/\partial b)$$

- α = Learning rate (ขนาดก้าวการเดิน)
- α เล็กเกินไป ⟶ เรียนช้า, α ใหญ่เกินไป ⟶ diverge

7.6 R^2 (Coefficient of Determination)

$$R^2 = 1 - (SSE / SSyy)$$

$$SSyy = \Sigma(y_i - \bar{y})^2 \quad \text{(Total variance)}$$

$$SSE = \Sigma(y_i - \hat{Y}_i)^2 \quad \text{(Unexplained variance)}$$

- R^2 ใกล้ 1 ⟶ โมเดลอธิบายความแปรปรวนได้ดี
- $R^2 = 0$ ⟶ โมเดลไม่ดีกว่าการใช้ค่าเฉลี่ย

7.7 Regularization (การป้องกัน Overfitting)

Ridge Regression (L2)

$$Loss_Ridge = \Sigma(y_i - \hat{y}_i)^2 + \lambda \cdot \Sigma\theta_j^2$$

- เพิ่ม penalty ของ weight² ⟶ ลด magnitude ของ weight แต่ไม่เป็น 0

Lasso Regression (L1)

$$Loss_Lasso = \Sigma(y_i - \hat{y}_i)^2 + \lambda \cdot \Sigma|\theta_j|$$

- เพิ่ม penalty ของ |weight| ⟶ บาง weight กลายเป็น 0 (feature selection)
- λ (lambda) = hyperparameter ควบคุมความแรงของ regularization

7.8 Decision Tree Regression

- ใช้ Decision Tree แบ่งพื้นที่ feature space
- ค่าพยากรณ์ = ค่าเฉลี่ยของ sample ใน leaf node
- ข้อดี: ไม่ต้องสมมติความสัมพันธ์เชิงเส้น, ข้อเสีย: Overfitting ง่าย

บทที่ 8: Model Parameter Tuning (การปรับแต่งพารามิเตอร์โมเดล)

8.1 Learnable Parameters vs Hyperparameters

ประเภท	คำอธิบาย	ตัวอย่าง
Learnable Parameters	ถูกเรียนรู้โดยโมเดลระหว่าง training โดยอัตโนมัติ (minimize loss)	weights, coefficients, biases
Hyperparameters	กำหนดโดยผู้ใช้ก่อน training (ไม่ได้เรียนรู้จากข้อมูล)	learning rate, max_depth, k ใน KNN

8.2 Hyperparameters ของ Decision Tree

Hyperparameter	ความหมาย
max_depth	ความลึกสูงสุดของต้นไม้ — ป้องกัน Overfitting
min_samples_split	จำนวน sample ขั้นต่ำในโหนดก่อนแยก
min_samples_leaf	จำนวน sample ขั้นต่ำที่ leaf node ต้องมี
max_features	จำนวน feature สูงสุดที่พิจารณาในการแยก
criterion	วิธีวัดคุณภาพการแบ่ง: Entropy หรือ Gini
splitter	กลยุทธ์การแบ่ง: Best split หรือ Random split

8.3 Overfitting และ Underfitting

Overfitting

- โมเดลเรียนรู้ noise ใน training data มากเกินไป
- ลักษณะ: Training accuracy สูง, Test accuracy ต่ำ
- สาเหตุ: โมเดลซับซ้อนเกินไป (deep trees, too many parameters)

วิธีป้องกัน Overfitting

- Simplify model (pruning trees, reduce parameters)
- Regularization (L1/L2)
- เพิ่มข้อมูล (More training data)
- Cross-Validation
- Dropout (Neural Networks)

Underfitting

- โมเดลเรียบง่ายเกินไป ไม่สามารถจับ pattern ของข้อมูลได้
- ลักษณะ: Training และ Test accuracy ต่ำทั้งคู่

วิธีป้องกัน Underfitting

- เพิ่มความซับซ้อนของโมเดล (Increase model complexity)

- Feature Engineering (เพิ่ม relevant features)
- ลด Regularization (Reduce regularization)
- เพิ่มข้อมูล (More data)

8.4 Cross-Validation

เทคนิคประเมินโมเดลโดยแบ่งข้อมูลเป็นหลายชุด training/validation

ประเภท	คำอธิบาย	เหมาะกับ
k-Fold CV	แบ่งข้อมูลเป็น k ส่วน วนทดสอบ k รอบ	ข้อมูลทั่วไป
Stratified k-Fold	แต่ละ fold มี class distribution เหมือนชุดข้อมูลต้นฉบับ	ข้อมูล imbalanced
Leave-One-Out (LOO-CV)	ใช้ 1 sample เป็น test ทีละครั้ง	ข้อมูลน้อย
Time Series CV	train บน past data, test บน future data	ข้อมูล time-series

Validation Set vs Test Set:

- Validation Set: ใช้ระหว่าง training เพื่อ tune hyperparameters
- Test Set: ใช้ครั้งเดียวหลัง training เสร็จ เพื่อประเมิน final performance

8.5 วิธีการ Hyperparameter Tuning

วิธี	คำอธิบาย	ข้อดี/ข้อเสีย
Manual	ลองด้วยประสบการณ์และความรู้	เข้าใจง่าย / ช้า ไม่ครอบคลุม
Grid Search	ค้นหา exhaustively ทุก combination	รับประกันว่าหา best ได้ / ใช้เวลามาก
Random Search	สุ่ม sample จาก parameter space	เร็วกว่า Grid Search / อาจพลาด best
Bayesian Opt.	ใช้ผลก่อนหน้าเป็น prior เพื่อเลือก next point	ฉลาด ประหยัด / ซับซ้อน

Tools: scikit-optimize, Hyperopt, Optuna

8.6 Gradient Descent Variants

ประเภท	Data ที่ใช้ต่อ update	Update ต่อ Epoch	ลักษณะ
Batch GD	ทั้งชุดข้อมูล	1	เสถียร แต่ช้า
Stochastic GD	1 sample	n (= n samples)	เร็ว แต่ noisy
Mini-batch GD	batch ขนาดเล็ก (32/64)	n/batch_size	สมดุลระหว่าง 2 แบบ

8.7 Challenges: Local Minima

- Local Minima: gradient descent หยุดที่จุดต่ำสุดใน local แต่ไม่ใช่ global minimum
- วิธีแก้: SGD (เพิ่ม noise), Learning Rate Scheduling, Adaptive Optimizers (Adam, RMSProp)

บทที่ 9: ML Clustering (การจัดกลุ่มข้อมูล)

9.1 Unsupervised Learning

- ไม่มี label ⟶ โมเดลหา pattern เองจากข้อมูล
- Clustering: จัดกลุ่ม data points ที่คล้ายกันเข้าด้วยกัน
- คุณภาพ clustering ที่ดี: High intra-class similarity + Low inter-class similarity

9.2 Distance Functions

Distance	สูตร	หมายเหตุ
Euclidean	$d(X,Y) = \sqrt{\Sigma(x_i - y_i)^2}$	ระยะทางตรง (straight line)
Manhattan	$d(X,Y) = \Sigma\lvert x_i - y_i\rvert$	ระยะทางแบบ city block
Cosine	$d(X,Y) = (X \cdot Y) / (\lVert X\rVert \times \lVert Y\rVert)$	วัดมุมระหว่างเวกเตอร์ (text)
Hamming	จำนวน position ที่แตกต่างกัน	ใช้กับข้อมูล categorical/binary

9.3 Clustering Approaches

1) Partitioning-based: K-Means

แบ่งข้อมูลเป็น k cluster โดยใช้ centroid (จุดศูนย์กลาง)

K-Means Algorithm Steps:

- ขั้นที่ 1: กำหนด k และสุ่ม initial centroids
- ขั้นที่ 2: คำนวณ centroid ของแต่ละ cluster
- ขั้นที่ 3: กำหนด sample แต่ละตัวให้ cluster ที่ centroid ใกล้ที่สุด
- ขั้นที่ 4: วนกลับขั้นที่ 2 จนกว่าไม่มีการเปลี่ยน assignment

Objective: $\hat{C} = \text{argmin}\{SSE(C)\}$

$$SSE = \Sigma_clusters\ \Sigma_points\ dist(point, centroid)^2$$

การเลือก k

- Elbow Method: Plot WCSS vs k ⟶ หาจุด 'elbow'

$$WCSS = \Sigma\ squared\ distances\ from\ all\ points\ to\ their\ cluster\ centroid$$

- Silhouette Score: วัดว่า sample อยู่ใน cluster ที่เหมาะสมแค่ไหน
- +1 ⟶ จัดกลุ่มถูกต้องมาก, 0 ⟶ อยู่บน boundary, -1 ⟶ น่าจะอยู่ cluster ผิด

2) Hierarchical Clustering

- สร้าง nested clusters เป็น tree structure (dendrogram)

- Agglomerative (bottom-up): เริ่มจากแต่ละ sample เป็น cluster ⟶ merge ทีละคู่
- Divisive (top-down): เริ่มจาก cluster เดียว ⟶ แบ่งออก

Agglomerative Algorithm:

- 1. คำนวณ proximity matrix
- 2. ให้ทุก data point เป็น cluster ของตัวเอง
- 3. Merge คู่ที่ใกล้ที่สุด
- 4. Update proximity matrix
- 5. ทำซ้ำจนเหลือ 1 cluster

Complete Linkage: MAX((P3,P6),P1) = MAX(P3,P1), (P6,P1)) — ใช้ distance ไกลสุด

3) Density-based: DBSCAN

- จัดกลุ่มตามความหนาแน่นของข้อมูล (density-based)
- รับมือกับ cluster รูปทรงแปลก (non-convex shapes) ได้ดี
- จัดการ noise/outliers ได้โดยอัตโนมัติ (ไม่ต้องกำหนดจำนวน cluster)
- Parameters: **ε** (epsilon = radius), MinPts (จำนวน sample ขั้นต่ำ)

Algorithm	กำหนด k?	จัดการ outlier?	รูปทรง cluster
K-Means	ใช่	ไม่	Convex (กลม)
Hierarchical	ไม่	บางส่วน	ยืดหยุ่น
DBSCAN	ไม่	ใช่	ทุกรูปทรง

ตาราง Cheat Sheet สูตรทั้งหมด (Formula Cheat Sheet)

บทที่ 6 — Classification Formulas

สูตร/Concept	นิยาม
Entropy(t)	$-\Sigma_j\ p(j\|t) \cdot \log_2 p(j\|t)$
Information Gain	Entropy(parent) - Σ_i (n_i/n)·Entropy(child$_i$)
Gini Index	$1 - \Sigma_j\ [p(j\|t)]^2$
Accuracy	(TP+TN) / (TP+TN+FP+FN)
Recall	TP / (TP+FN)
Precision	TP / (TP+FP)
F1-Score	2×(P×R) / (P+R)
Logistic (Sigmoid)	1 / (1 + e^(-z))

บทที่ 7 — Regression Formulas

สูตร/Concept	นิยาม
Linear Regression	$\hat{y} = w_0 + w_1x_1 + \ldots + w_nx_n$
MSE	$(1/n)\ \Sigma(y_i - \hat{y}_i)^2$
RMSE	$\sqrt{[(1/n)\ \Sigma(y_i - \hat{y}_i)^2]}$
MAE	$(1/n)\ \Sigma\|y_i - \hat{y}_i\|$
Cost Function J	$(1/n)\ \Sigma(y_i - \hat{y}_i)^2$
Gradient Descent w	$w \longleftarrow w - \alpha \cdot (\partial J/\partial w)$
Gradient Descent b	$b \longleftarrow b - \alpha \cdot (\partial J/\partial b)$
R^2	$1 - SSE/SSyy$ where $SSyy=\Sigma(y-\bar{y})^2$, $SSE=\Sigma(y-\hat{y})^2$
Ridge Loss (L2)	$\Sigma(y-\hat{y})^2 + \lambda\Sigma\beta_j^2$
Lasso Loss (L1)	$\Sigma(y-\hat{y})^2 + \lambda\Sigma\|\beta_j\|$

บทที่ 8 — Parameter Tuning Concepts

Concept	สรุป
Overfitting	High train acc, Low test acc ⟶ โมเดลซับซ้อนเกิน
Underfitting	Low train & test acc ⟶ โมเดลเรียบง่ายเกิน
k-Fold CV	แบ่ง k ส่วน วนทดสอบ k รอบ ⟶ ค่าเฉลี่ย accuracy
Grid Search	ค้นหา exhaustively ทุก combination
Random Search	สุ่ม sample จาก hyperparameter space
Batch GD update/epoch	1 update
SGD update/epoch	n updates (n = samples)
Mini-batch update	n/batch_size updates
Acc (CM)	(TP+TN)/(TP+TN+FP+FN)
F1 (CM)	2PR/(P+R)

บทที่ 9 — Clustering Formulas

สูตร/Concept	นิยาม
Euclidean distance	$d(X,Y) = \sqrt{\Sigma(x_i-y_i)^2}$
Manhattan distance	$d(X,Y) = \Sigma\lvert x_i-y_i\rvert$
Cosine similarity	$sim(X,Y) = (X\cdot Y)/(\lVert X\rVert\cdot\lVert Y\rVert)$
WCSS (Inertia)	Σ_clusters Σ_points dist(p, centroid)^2
K-Means objective	$\hat{C}$ = argmin_C {SSE(C)}
Silhouette Score	s(i) = (b(i)-a(i)) / max(a(i),b(i)) range: [-1,+1]

แนวข้อสอบ — บทที่ 6: ML Classification

ส่วนที่ 1: ข้อสอบปรนัย (Multiple Choice Questions)

ข้อ 1. Entropy ของ node ที่มีข้อมูล 2 class เท่ากัน (50:50) มีค่าเท่าใด?

- a) 0
- b) 0.5
- c) 1.0
- d) 2.0

✓ เฉลย: c) 1.0 *(Entropy = $-(0.5 \cdot \log_2 0.5 + 0.5 \cdot \log_2 0.5) = -(-1) = 1$)*

ข้อ 2. จาก Confusion Matrix: TP=80, TN=70, FP=10, FN=20 Accuracy มีค่าเท่าใด?

- a) 80%
- b) 83.3%
- c) 90%
- d) 75%

✓ เฉลย: b) 83.3% *((80+70)/(80+70+10+20) = 150/180 = 0.833)*

ข้อ 3. อัลกอริทึมใดที่ใช้ 'จำนวนเพื่อนบ้านที่ใกล้ที่สุด' เป็น hyperparameter หลัก?

- a) Naive Bayes
- b) Decision Tree
- c) K-Nearest Neighbors
- d) SVM

✓ เฉลย: c) K-Nearest Neighbors

ข้อ 4. Random Forest แตกต่างจาก Decision Tree อย่างไร?

- a) ใช้ Entropy แทน Gini
- b) รวม Decision Trees หลายต้นด้วย Bagging
- c) ไม่มี leaf node
- d) ใช้ Linear Regression ภายใน

✓ เฉลย: b) รวม Decision Trees หลายต้นด้วย Bagging

ข้อ 5. Recall = TP/(TP+FN) วัดอะไร?

- a) ความแม่นยำของ prediction ที่เป็น positive
- b) สัดส่วนของ positive จริงที่โมเดลตรวจพบได้
- c) ความถูกต้องโดยรวม
- d) ความสามารถในการแยก class

✓ เฉลย: b) สัดส่วนของ positive จริงที่โมเดลตรวจพบได้

ข้อ 6. Gini Index = 0 หมายความว่าอะไร?

- a) Node มีความไม่แน่นอนสูงสุด
- b) Node เป็น pure (มีแค่ 1 class)
- c) ต้องแบ่ง node ต่อ
- d) ข้อมูลมี noise มาก

✓ เฉลย: b) Node เป็น pure (มีแค่ 1 class)

ข้อ 7. SVM มีเป้าหมายหลักคืออะไร?

- a) Minimize entropy
- b) หา hyperplane ที่แบ่ง class โดยมี margin สูงสุด
- c) สร้าง decision tree ลึกที่สุด
- d) คำนวณ probability ของแต่ละ class

✓ เฉลย: b) หา hyperplane ที่แบ่ง class โดยมี margin สูงสุด

ข้อ 8. Information Gain ของ feature ที่ดีที่สุดใน Decision Tree ควรมีค่า...

- a) ต่ำสุด
- b) เท่ากับ 0
- c) สูงสุด
- d) เท่ากับ 1

✓ เฉลย: c) สูงสุด *(Gain สูงหมายถึง feature นั้นช่วยแบ่งข้อมูลได้ดีที่สุด)*

ข้อ 9. F1-Score เหมาะใช้เมื่อใด?

- a) ข้อมูล balanced
- b) ข้อมูล imbalanced และต้องการ balance precision กับ recall
- c) ต้องการ maximize TN
- d) ข้อมูลเป็น regression

✓ เฉลย: b) ข้อมูล imbalanced และต้องการ balance precision กับ recall

ข้อ 10. Logistic Regression ใช้ function ใดแปลงผลลัพธ์เป็นค่าความน่าจะเป็น?

- a) ReLU
- b) Tanh
- c) Sigmoid
- d) Softmax

✓ เฉลย: c) Sigmoid *(P(y=1|x) = 1/(1+e^(-z)))*

ส่วนที่ 2: คำนวณ (Calculation Questions)

ข้อ 1. จงคำนวณ Entropy ของ node ที่มีข้อมูล: Class A = 4 ตัวอย่าง, Class B = 6 ตัวอย่าง

วิธีทำ:

- p(A) = 4/10 = 0.4, p(B) = 6/10 = 0.6
- Entropy = $-(0.4 \cdot \log_2 0.4) - (0.6 \cdot \log_2 0.6)$
- = -(0.4·(-1.322)) - (0.6·(-0.737))
- = 0.529 + 0.442
- = 0.971

ข้อ 2. Confusion Matrix: TP=90, TN=85, FP=15, FN=10 จงคำนวณ Accuracy, Precision, Recall, F1

วิธีทำ:

- Accuracy = (90+85)/(90+85+15+10) = 175/200 = 0.875 = 87.5%
- Precision = 90/(90+15) = 90/105 = 0.857 = 85.7%
- Recall = 90/(90+10) = 90/100 = 0.900 = 90.0%
- F1 = 2×(0.857×0.900)/(0.857+0.900) = 2×0.771/1.757 = 0.878 = 87.8%

ข้อ 3. คำนวณ Information Gain ของ feature 'student' จากข้อมูลต่อไปนี้: Parent Entropy = 0.94, Child yes (7 samples, Entropy=0.985), Child no (7 samples, Entropy=0.592)

วิธีทำ:

- Gain = 0.94 - [(7/14)×0.985 + (7/14)×0.592]
- = 0.94 - [0.5×0.985 + 0.5×0.592]
- = 0.94 - [0.4925 + 0.296]
- = 0.94 - 0.7885
- = 0.152

ส่วนที่ 3: วิเคราะห์ (Analysis Questions)

ข้อ 1. อธิบายความแตกต่างระหว่าง Precision กับ Recall และยกตัวอย่างสถานการณ์ที่ควรให้ความสำคัญกับ Recall มากกว่า Precision

แนวคำตอบ:

- Precision = TP/(TP+FP) — วัดว่า prediction ที่บอกว่า positive นั้น ถูกต้องกี่%
- Recall = TP/(TP+FN) — วัดว่า positive จริงๆ ถูกตรวจพบได้กี่%
- Recall สำคัญกว่าเมื่อ FN มีต้นทุนสูง เช่น การตรวจโรคมะเร็ง (พลาดผู้ป่วยจริง = อันตราย)
- Precision สำคัญกว่าเมื่อ FP มีต้นทุนสูง เช่น spam filter (email สำคัญถูก block = ปัญหา)

ข้อ 2. เปรียบเทียบ Decision Tree กับ Random Forest ในด้านการ Overfitting

แนวคำตอบ:

- Decision Tree: มีแนวโน้ม Overfit สูงหากไม่กำหนด max_depth เพราะต้นไม้เติบโตจนจำ noise
- Random Forest: ลด Overfitting ด้วย Bagging — สร้างหลาย trees จาก random subset ของข้อมูลและ features
- ผลลัพธ์ใน RF มาจากการ vote ของหลาย trees ⟶ ลด variance, เพิ่ม generalization
- RF ทนทานต่อ noise กว่า แต่ interpretability ต่ำกว่า Decision Tree เดี่ยว

แนวข้อสอบ — บทที่ 7: ML Regression

ส่วนที่ 1: ข้อสอบปรนัย (Multiple Choice Questions)

ข้อ 1. MSE แตกต่างจาก MAE อย่างไร?

- a) MSE ใช้ค่าสัมบูรณ์, MAE ใช้กำลังสอง
- b) MSE กำลังสอง error จึงให้น้ำหนัก outlier มากกว่า MAE
- c) MSE ใช้เมื่อข้อมูล imbalanced เท่านั้น
- d) ไม่มีความแตกต่าง

✓ เฉลย: b) MSE กำลังสอง error จึงให้น้ำหนัก outlier มากกว่า MAE

ข้อ 2. ใน Gradient Descent: $w = w - \alpha \cdot (\partial J/\partial w)$ ตัวแปร α หมายถึงอะไร?

- a) Loss function
- b) Regularization term
- c) Learning rate
- d) Number of epochs

✓ เฉลย: c) Learning rate

ข้อ 3. Ridge Regression แตกต่างจาก Lasso Regression อย่างไร?

- a) Ridge ใช้ $|\beta|$, Lasso ใช้ β^2
- b) Ridge ใช้ β^2 (L2), Lasso ใช้ $|\beta|$ (L1) — Lasso อาจทำให้ weight = 0
- c) Lasso ใช้ลด Underfitting
- d) Ridge เหมาะกับ classification เท่านั้น

✓ เฉลย: b) Ridge ใช้ β^2 (L2), Lasso ใช้ $|\beta|$ (L1) — Lasso อาจทำให้ weight = 0

ข้อ 4. $R^2 = 0.85$ หมายความว่าอะไร?

- a) โมเดล error 85%
- b) โมเดลอธิบายความแปรปรวนของ y ได้ 85%
- c) Accuracy = 85%
- d) MSE = 0.15

✓ เฉลย: b) โมเดลอธิบายความแปรปรวนของ y ได้ 85%

ข้อ 5. Polynomial Regression degree สูงมากๆ ก่อให้เกิดปัญหาใด?

- a) Underfitting
- b) Overfitting
- c) ลด MSE เสมอ
- d) เพิ่ม bias

✓ เฉลย: b) Overfitting

ข้อ 6. RMSE มีข้อดีอะไรเหนือ MSE?

- a) คำนวณเร็วกว่า
- b) มีหน่วยเดียวกับตัวแปร y ทำให้ตีความง่ายกว่า
- c) ไม่ sensitive กับ outlier
- d) ค่าน้อยกว่า MSE เสมอ

✓ เฉลย: b) มีหน่วยเดียวกับตัวแปร y ทำให้ตีความง่ายกว่า

ข้อ 7. Decision Tree Regression ต่างจาก Linear Regression อย่างไร?

- a) ไม่ต้องสมมติความสัมพันธ์เชิงเส้น
- b) ใช้ Gradient Descent
- c) ต้องการข้อมูลมากกว่า
- d) ทำงานได้เฉพาะ classification

✓ เฉลย: a) ไม่ต้องสมมติความสัมพันธ์เชิงเส้น

ข้อ 8. หาก learning rate สูงเกินไป Gradient Descent จะเกิดอะไร?

- a) เรียนช้ามาก
- b) Converge เร็ว
- c) Diverge (ไม่ลู่เข้า)
- d) Underfitting

✓ เฉลย: c) Diverge (ไม่ลู่เข้า)

ข้อ 9. R^2 สูงสุดที่เป็นไปได้คือเท่าใด?

- a) 100
- b) 1.0
- c) ∞
- d) 0.999

✓ เฉลย: b) 1.0 *($R^2=1$ ⟶ โมเดล predict ได้สมบูรณ์, $R^2<0$ เป็นไปได้หากโมเดลแย่กว่า baseline)*

ข้อ 10. Lasso มีข้อดีพิเศษอะไรเหนือ Ridge?

- a) ลด MSE มากกว่า
- b) ทำ feature selection โดยทำให้ weight บาง feature = 0
- c) Training เร็วกว่า
- d) ไม่มี hyperparameter

✓ เฉลย: b) ทำ feature selection โดยทำให้ weight บาง feature = 0

ส่วนที่ 2: คำนวณ (Calculation Questions)

ข้อ 1. กำหนดข้อมูล: y = [3, 5, 4, 7, 6], $\hat{y}$ = [2.5, 5.5, 4.0, 6.5, 5.5] จงคำนวณ MSE และ MAE

วิธีทำ:

- Errors: [0.5, -0.5, 0, 0.5, 0.5]
- MSE = $(0.5^2 + 0.5^2 + 0^2 + 0.5^2 + 0.5^2) / 5$ = (0.25+0.25+0+0.25+0.25)/5 = 1.0/5 = 0.20
- MAE = (|0.5|+|0.5|+|0|+|0.5|+|0.5|) / 5 = 2.0/5 = 0.40
- RMSE = √0.20 = 0.447

ข้อ 2. กำหนด SSyy = 200, SSE = 30 จงคำนวณ R^2

วิธีทำ:

- R^2 = 1 - SSE/SSyy
- = 1 - 30/200
- = 1 - 0.15
- = 0.85
- ⟶ โมเดลอธิบายความแปรปรวนได้ 85%

ข้อ 3. โมเดลมี w=2.5, b=1.0, α=0.1, $\partial J/\partial w$=0.4, $\partial J/\partial b$=0.2 จงคำนวณ w และ b ใหม่หลัง 1 step

วิธีทำ:

- w_new = w - $\alpha \cdot (\partial J/\partial w)$ = 2.5 - 0.1×0.4 = 2.5 - 0.04 = 2.46
- b_new = b - $\alpha \cdot (\partial J/\partial b)$ = 1.0 - 0.1×0.2 = 1.0 - 0.02 = 0.98

ส่วนที่ 3: วิเคราะห์ (Analysis Questions)

ข้อ 1. อธิบายปัญหา Overfitting ใน Regression และวิธีใช้ Regularization แก้ปัญหา

แนวคำตอบ:

- Overfitting ใน Regression: โมเดลที่ degree สูงมากจะผ่านทุก data point แต่ predict ข้อมูลใหม่ได้ไม่ดี

- Ridge (L2): เพิ่ม $\lambda\Sigma\beta_j^2$ ใน loss ⟶ บังคับให้ weight มีค่าน้อยลง แต่ไม่เป็น 0
- Lasso (L1): เพิ่ม $\lambda\Sigma|\beta_j|$ ใน loss ⟶ บาง feature มี weight = 0 (feature selection)
- λ ใหญ่ = regularization แรง (ลด Overfitting แต่เสี่ยง Underfitting)
- ควรใช้ Cross-Validation เพื่อหาค่า λ ที่เหมาะสม

ข้อ 2. เปรียบเทียบ Batch Gradient Descent, Stochastic GD และ Mini-batch GD

แนวคำตอบ:

- Batch GD: ใช้ทั้ง dataset คำนวณ gradient ⟶ เสถียร แต่ช้าและใช้ memory มาก
- Stochastic GD: ใช้ 1 sample ⟶ เร็ว แต่ noisy อาจไม่ converge
- Mini-batch GD: ใช้ batch 32-64 samples ⟶ สมดุลระหว่าง 2 แบบ (ใช้ใน Deep Learning)
- ในทางปฏิบัติ Mini-batch GD นิยมใช้มากที่สุด

แนวข้อสอบ — บทที่ 8: Model Parameter Tuning

ส่วนที่ 1: ข้อสอบปรนัย (Multiple Choice Questions)

ข้อ 1. Hyperparameter ต่างจาก Learnable Parameter อย่างไร?

- a) Hyperparameter ถูกเรียนรู้จาก data โดยตรง
- b) Hyperparameter กำหนดโดยผู้ใช้ก่อน training (ไม่ได้ optimize ระหว่าง training)
- c) Hyperparameter ใช้เฉพาะกับ Neural Networks
- d) Learnable Parameter ต้อง tune ด้วย Grid Search

✓ เฉลย: b) Hyperparameter กำหนดโดยผู้ใช้ก่อน training (ไม่ได้ optimize ระหว่าง training)

ข้อ 2. max_depth ใน Decision Tree มีผลอย่างไร?

- a) กำหนดจำนวน feature ที่ใช้
- b) ควบคุมความลึกสูงสุด — ค่าน้อย = ป้องกัน Overfitting
- c) กำหนด learning rate
- d) เลือก Gini หรือ Entropy

✓ เฉลย: b) ควบคุมความลึกสูงสุด — ค่าน้อย = ป้องกัน Overfitting

ข้อ 3. Stratified k-Fold Cross-Validation เหมาะกับสถานการณ์ใด?

- a) ข้อมูล time-series
- b) ข้อมูล balanced
- c) ข้อมูล imbalanced (class distribution ไม่เท่ากัน)
- d) ข้อมูลขนาดใหญ่มาก

✓ เฉลย: c) ข้อมูล imbalanced (class distribution ไม่เท่ากัน)

ข้อ 4. Grid Search ต่างจาก Random Search อย่างไร?

- a) Grid Search สุ่ม, Random Search ค้นหาทุก combination
- b) Grid Search ค้นหาทุก combination, Random Search สุ่ม sample
- c) Grid Search เร็วกว่า
- d) Random Search รับประกัน global optimal

✓ เฉลย: b) Grid Search ค้นหาทุก combination, Random Search สุ่ม sample

ข้อ 5. Validation Set ต่างจาก Test Set อย่างไร?

- a) ไม่ต่างกัน
- b) Validation ใช้ระหว่าง training เพื่อ tune HP; Test ใช้ครั้งเดียวหลัง training
- c) Test ใช้สำหรับ tune hyperparameters
- d) Validation เป็นส่วนหนึ่งของ Test Set

✓ เฉลย: b) Validation ใช้ระหว่าง training เพื่อ tune HP; Test ใช้ครั้งเดียวหลัง training

ข้อ 6. Leave-One-Out Cross-Validation (LOO-CV) มีข้อเสียใด?

- a) ไม่สามารถใช้กับ imbalanced data
- b) Computationally expensive เพราะต้อง train n รอบ
- c) ใช้ได้เฉพาะ regression
- d) ให้ผล biased estimate

✓ เฉลย: b) Computationally expensive เพราะต้อง train n รอบ

ข้อ 7. Dropout ใช้แก้ปัญหาใด?

- a) Underfitting ใน Decision Tree
- b) Overfitting ใน Neural Networks โดย random drop connections
- c) Local Minima ใน Gradient Descent
- d) Class imbalance ใน Classification

✓ เฉลย: b) Overfitting ใน Neural Networks โดย random drop connections

ข้อ 8. min_samples_split = 10 หมายความว่า...

- a) ต้นไม้มีได้ไม่เกิน 10 leaf nodes
- b) node ที่มีน้อยกว่า 10 samples จะไม่ถูกแบ่งต่อ
- c) แต่ละ leaf ต้องมีอย่างน้อย 10 samples
- d) จำนวน features สูงสุดที่พิจารณาคือ 10

✓ เฉลย: b) node ที่มีน้อยกว่า 10 samples จะไม่ถูกแบ่งต่อ

ข้อ 9. Local Minima ในการ optimize hyperparameter คืออะไร?

- a) จุดที่ loss สูงสุด
- b) จุดที่ loss ต่ำกว่าบริเวณใกล้เคียงแต่ไม่ใช่ต่ำสุดทั่วโลก (global minimum)
- c) จุดที่ learning rate = 0
- d) จุดที่ model เริ่มต้น training

✓ เฉลย: b) จุดที่ loss ต่ำกว่าบริเวณใกล้เคียงแต่ไม่ใช่ต่ำสุดทั่วโลก (global minimum)

ข้อ 10. k-Fold Cross-Validation ที่ k=5 หมายความว่า...

- a) train 5 models แยกกัน
- b) แบ่งข้อมูล 5 ส่วน วนทดสอบ 5 รอบ แต่ละรอบใช้ 1 ส่วนเป็น test
- c) ใช้ข้อมูล 5% เป็น test
- d) train 5 epochs

✓ เฉลย: b) แบ่งข้อมูล 5 ส่วน วนทดสอบ 5 รอบ แต่ละรอบใช้ 1 ส่วนเป็น test

ส่วนที่ 2: คำนวณ (Calculation Questions)

ข้อ 1. Grid Search สำหรับ Decision Tree: max_depth={2,3,4}, criterion={gini,entropy}, min_samples_split={2,5,10} มีจำนวน combination ทั้งหมดเท่าใด?

วิธีทำ:

- จำนวน max_depth = 3 ค่า
- จำนวน criterion = 2 ค่า
- จำนวน min_samples_split = 3 ค่า
- Total combinations = 3 × 2 × 3 = 18 combinations

ข้อ 2. 5-Fold CV ให้ accuracy: [0.82, 0.85, 0.80, 0.88, 0.83] จงหา mean accuracy และ std deviation

วิธีทำ:

- Mean = (0.82+0.85+0.80+0.88+0.83)/5 = 4.18/5 = 0.836 = 83.6%
- Variance = $\Sigma(x_i-\mu)^2/n$ = $[(0.82-0.836)^2+(0.85-0.836)^2+(0.80-0.836)^2+(0.88-0.836)^2+(0.83-0.836)^2]/5$
- = [0.000256+0.000196+0.001296+0.001936+0.000036]/5 = 0.00372/5 = 0.000744
- Std = $\sqrt{0.000744} \approx 0.027$ = 2.7%

ส่วนที่ 3: วิเคราะห์ (Analysis Questions)

ข้อ 1. อธิบาย Bias-Variance Tradeoff และความสัมพันธ์กับ Overfitting/Underfitting

แนวคำตอบ:

- Bias = ความผิดพลาดจากการสมมติที่ผิด (โมเดลง่ายเกินไป) ⟶ Underfitting
- Variance = ความผิดพลาดจากความ sensitive ต่อ noise (โมเดลซับซ้อนเกิน) ⟶ Overfitting
- Trade-off: ลด Bias มักเพิ่ม Variance และกลับกัน
- Cross-Validation ช่วยหา sweet spot ระหว่าง bias และ variance
- Regularization ช่วย balance โดยเพิ่ม constraint บน model complexity

แนวข้อสอบ — บทที่ 9: ML Clustering

ส่วนที่ 1: ข้อสอบปรนัย (Multiple Choice Questions)

ข้อ 1. K-Means Clustering กำหนดค่า k หมายถึงอะไร?

- a) จำนวน iterations
- b) จำนวน clusters ที่ต้องการ
- c) จำนวน features
- d) จำนวน data points

✓ เฉลย: b) จำนวน clusters ที่ต้องการ

ข้อ 2. Elbow Method ใช้วิเคราะห์อะไร?

- a) Silhouette Score vs k
- b) WCSS vs จำนวน clusters เพื่อหา optimal k
- c) Distance function ที่เหมาะสม
- d) จำนวน iterations ที่ต้องการ

✓ เฉลย: b) WCSS vs จำนวน clusters เพื่อหา optimal k

ข้อ 3. Silhouette Score = +1 หมายความว่าอะไร?

- a) sample อยู่บน boundary ระหว่าง 2 clusters
- b) sample อยู่ใน cluster ที่เหมาะสมมาก (ห่างจาก cluster อื่นๆ)
- c) sample น่าจะอยู่ผิด cluster
- d) ข้อมูลเป็น outlier

✓ เฉลย: b) sample อยู่ใน cluster ที่เหมาะสมมาก (ห่างจาก cluster อื่นๆ)

ข้อ 4. Euclidean distance ระหว่าง A(1,2) และ B(4,6) มีค่าเท่าใด?

- a) 3
- b) 4
- c) 5
- d) 7

✓ เฉลย: c) 5 *($d = \sqrt{(4-1)^2+(6-2)^2} = \sqrt{9+16} = \sqrt{25} = 5$)*

ข้อ 5. DBSCAN เหนือกว่า K-Means ในด้านใด?

- a) เร็วกว่า
- b) จัดการ outlier และ cluster รูปทรงแปลกได้ดีกว่า
- c) ไม่ต้องกำหนด parameter ใดเลย
- d) ใช้ Euclidean distance เสมอ

✓ เฉลย: b) จัดการ outlier และ cluster รูปทรงแปลกได้ดีกว่า

ข้อ 6. Agglomerative Hierarchical Clustering เริ่มต้นอย่างไร?

- a) เริ่มด้วย cluster เดียวแล้วแบ่งออก
- b) เริ่มด้วยแต่ละ data point เป็น cluster แล้ว merge
- c) สุ่ม centroids แล้ว assign
- d) สร้าง dendrogram จากบนลงล่าง

✓ เฉลย: b) เริ่มด้วยแต่ละ data point เป็น cluster แล้ว merge

ข้อ 7. Manhattan distance ระหว่าง A(1,3) และ B(4,7) คือเท่าใด?

- a) 5
- b) 7
- c) 3
- d) 4

✓ เฉลย: b) 7 *(d = |4-1| + |7-3| = 3 + 4 = 7)*

ข้อ 8. Clustering เป็น Unsupervised Learning เพราะ...

- a) ใช้ labeled data
- b) ไม่มี ground truth labels — โมเดลหา pattern เองจากข้อมูล
- c) ใช้ loss function เหมือน Regression
- d) ต้องการ validation set

✓ เฉลย: b) ไม่มี ground truth labels — โมเดลหา pattern เองจากข้อมูล

ข้อ 9. Dendrogram ใช้กับ clustering approach ใด?

- a) K-Means
- b) DBSCAN
- c) Hierarchical Clustering
- d) K-Medoids

✓ เฉลย: c) Hierarchical Clustering

ข้อ 10. Cosine Similarity เหมาะกับการวัด distance ของข้อมูลประเภทใดที่สุด?

- a) ข้อมูลตัวเลขต่อเนื่อง
- b) ข้อมูล text/document vectors
- c) ข้อมูล binary categorical
- d) ข้อมูล time-series

✓ เฉลย: b) ข้อมูล text/document vectors

ส่วนที่ 2: คำนวณ (Calculation Questions)

ข้อ 1. กำหนดข้อมูล 2 มิติ: A(1,2), B(1,4), C(1,0), D(10,2), E(10,4), F(10,0) สมมติ k=2 Cluster1={A,B,C}, Cluster2={D,E,F} จงคำนวณ WCSS

วิธีทำ:

- Centroid1 = ((1+1+1)/3, (2+4+0)/3) = (1, 2)
- Centroid2 = ((10+10+10)/3, (2+4+0)/3) = (10, 2)
- $WCSS_1$: A⟶0, B⟶$(0^2+2^2)=4$, C⟶$(0^2+2^2)=4$ ⟶ $WCSS_1 = 8$
- $WCSS_2$: D⟶0, E⟶$(0^2+2^2)=4$, F⟶$(0^2+2^2)=4$ ⟶ $WCSS_2 = 8$
- WCSS_Total = 8 + 8 = 16

ข้อ 2. คำนวณ Euclidean distance ระหว่าง P1(0.4,0.53) กับ P3(0.35,0.32)

วิธีทำ:

- $d(P1,P3) = \sqrt{(0.4-0.35)^2 + (0.53-0.32)^2}$
- $= \sqrt{(0.05)^2 + (0.21)^2}$
- $= \sqrt{0.0025 + 0.0441}$
- $= \sqrt{0.0466}$
- ≈ 0.216

ข้อ 3. Agglomerative Clustering: จาก proximity matrix ค้นหาคู่ที่ใกล้ที่สุดใน {P1,P2,P3,P4,P5,P6} กำหนด distance matrix บางส่วน: d(P3,P6)=0.10, d(P2,P3)=0.14

วิธีทำ:

- เปรียบเทียบ distance ทุกคู่จาก proximity matrix
- คู่ที่ใกล้สุดคือ (P3, P6) = 0.10
- Merge P3 และ P6 เป็น cluster {P3,P6} ก่อน
- อัปเดต proximity matrix: d({P3,P6},P1) = MAX(d(P3,P1), d(P6,P1)) = MAX(0.22, 0.24) = 0.24

ส่วนที่ 3: วิเคราะห์ (Analysis Questions)

ข้อ 1. เปรียบเทียบ K-Means, Hierarchical Clustering และ DBSCAN ในด้านข้อดีข้อเสีย

แนวคำตอบ:

- K-Means: ข้อดี=เร็ว, เหมาะกับ convex clusters; ข้อเสีย=ต้องกำหนด k, sensitive ต่อ outlier
- Hierarchical: ข้อดี=ไม่ต้องกำหนด k ล่วงหน้า, สร้าง dendrogram; ข้อเสีย=ช้า $O(n^2)$
- DBSCAN: ข้อดี=จัดการ outlier, cluster รูปทรงใดก็ได้; ข้อเสีย=sensitive ต่อ **ε** และ MinPts
- เลือก K-Means เมื่อรู้จำนวน cluster และ cluster กลมๆ
- เลือก DBSCAN เมื่อมี outlier หรือ cluster รูปทรงแปลก

ข้อ 2. อธิบายปัญหาที่อาจเกิดขึ้นใน K-Means Clustering และวิธีแก้

แนวคำตอบ:

- ปัญหา 1: Sensitive ต่อ initial centroids ⟶ แก้โดยใช้ K-Means++ initialization
- ปัญหา 2: ต้องกำหนด k ⟶ แก้โดยใช้ Elbow Method หรือ Silhouette Score
- ปัญหา 3: ไม่รับมือ outlier ได้ดี ⟶ แก้โดยใช้ K-Medoids แทน
- ปัญหา 4: เหมาะกับ convex cluster เท่านั้น ⟶ แก้โดยใช้ DBSCAN สำหรับ non-convex
- ปัญหา 5: WCSS ลดเสมอเมื่อ k เพิ่ม ⟶ ใช้ Elbow Point หาจุดที่ลดอย่างมีนัย