CPE 232: Data Models

Keyword Glossary — ML Lectures 6–9

ครอบคลุม: Classification · Regression · Parameter Tuning · Clustering

Lecture 6 — ML: Classification

Keyword	ความหมาย
Machine Learning	ศาสตร์ที่ทำให้คอมพิวเตอร์เรียนรู้จากข้อมูลโดยไม่ต้องเขียน rule ให้ครบทุกกรณีด้วยมือ
Supervised Learning	การเรียนรู้แบบมี "ครู" คือมี Labeled Data (input + output ที่รู้คำตอบแล้ว) ใช้ฝึก model เพื่อทำนายผลลัพธ์ใหม่
Unsupervised Learning	การเรียนรู้แบบไม่มี label — ให้ model หาโครงสร้างหรือ pattern ซ่อนเร้นในข้อมูลเอง เช่น Clustering
Reinforcement Learning	การเรียนรู้โดยการลองผิดลองถูก model ได้ Reward เมื่อทำถูก และ Punishment เมื่อทำผิด
Classification	ประเภทของ Supervised Learning ที่ output เป็น "ประเภท/class" เช่น ใช่/ไม่ใช่, ประเภทอีเมล
Regression	ประเภทของ Supervised Learning ที่ output เป็น "ตัวเลขต่อเนื่อง" เช่น ราคาบ้าน, อุณหภูมิ
Model	การจำลองความเป็นจริงแบบย่อส่วน สร้างขึ้นเพื่อวัตถุประสงค์เฉพาะ
Predictive Model	สูตรหรือ algorithm ที่ใช้ทำนายค่า/class ที่ยังไม่รู้จากข้อมูล input แบ่งเป็น Classification model และ Regression model
Labeled Data	ข้อมูลที่มีทั้ง input (features) และ output (คำตอบ) อยู่แล้ว ใช้สำหรับ Supervised Learning
Unlabeled Data	ข้อมูลที่มีแต่ input ไม่มี output ใช้ใน Unsupervised Learning
Training Data	ข้อมูลที่ใช้สอน model ให้เรียนรู้ pattern
Testing Data	ข้อมูลที่ไม่เคยให้ model เห็น ใช้ประเมินประสิทธิภาพ model หลัง train เสร็จ
Feature / Attribute	คุณลักษณะของข้อมูลแต่ละตัวที่ใช้เป็น input ให้ model เช่น อายุ, รายได้, สี
Target	ค่า output ที่ต้องการทำนาย (class to predict)

Records	แถวข้อมูลแต่ละแถวใน dataset แทนข้อมูลหนึ่งรายการ
Feature Extraction	การเลือก feature ที่มีความสำคัญต่อ task classification มากที่สุด
Data Splitting	การแบ่ง dataset ออกเป็น training data และ testing data
Model Training	การเลือก algorithm และ train มันด้วย training data
Hyperparameter Tuning	การปรับค่า hyperparameter เพื่อ optimize ประสิทธิภาพ model
Model Evaluation	การประเมินประสิทธิภาพ model บน testing data ด้วย evaluation metrics
Classification / Prediction	ขั้นตอนสุดท้าย — นำ model ที่ train แล้วไปทำนายข้อมูลใหม่ที่ยังไม่เคยเห็น
Decision Tree	Algorithm ML สำหรับ classification ที่ใช้โครงสร้างต้นไม้ในการตัดสินใจ โดยแบ่งข้อมูลตาม feature ที่ให้ข้อมูลมากที่สุด
Tree Induction	กระบวนการ training ของ Decision Tree — การสร้างต้นไม้จาก training data
Root Node	โหนดบนสุดของ Decision Tree ที่ใช้ feature ที่ให้ Information Gain สูงสุด
Leaf Node	โหนดปลายสุดของ Decision Tree ที่ระบุผลลัพธ์ (class) สุดท้าย
Best Split	การแบ่งข้อมูลที่ดีที่สุด ณ โหนดหนึ่งๆ โดยพิจารณาจาก Node Impurity หรือ Information Gain
Node Impurity	ความ "ปนเปื้อน" ของโหนด — โหนดที่ pure = มี class เดียว, โหนดที่ impure = มีหลาย class ปนกัน
Entropy	ค่าวัดความ "disorder" หรือความไม่บริสุทธิ์ของโหนด สูง = ปนเปื้อนมาก, ต่ำ = บริสุทธิ์ สูตร: $\text{Entropy}(t) = -\sum p(j\|t) \log_2 p(j\|t)$
Information Gain (IG)	ค่าวัดว่า feature นั้นๆ มีประโยชน์มากแค่ไหนในการแบ่ง — entropy ก่อน split ลบ entropy หลัง split สูตร: $\text{Gain} = \text{Entropy}(\text{parent}) - \sum(n_i/n) \times \text{Entropy}(\text{partition}_i)$
Ordinal Attributes	Attribute ที่เป็นประเภท/ลำดับ (categorical) เช่น CarType: Family, Sports, Luxury
Continuous Attributes	Attribute ที่เป็นตัวเลขต่อเนื่อง เช่น Income, Age
Parent Node	โหนดที่ถูก split แบ่งออกไปเป็น child nodes
Partition	กลุ่มย่อยของข้อมูลที่ได้จากการ split โหนด
Majority Class	Class ที่พบมากที่สุดในโหนด ใช้เป็น label ของ leaf node เมื่อถึง stop condition
Stop Condition	เงื่อนไขที่ทำให้หยุดสร้าง Decision Tree เพิ่ม เช่น ทุก record ใน node เป็น class เดียวกันแล้ว
Gini Index	อีกหนึ่งวิธีวัด Node Impurity นอกจาก Entropy ใช้เลือก best split เช่นกัน เป็น alternative ของ Entropy

max_depth	Hyperparameter ของ Decision Tree ที่กำหนดความลึกสูงสุดของต้นไม้
min_samples_split	Hyperparameter ที่กำหนดจำนวน sample ขั้นต่ำที่ node ต้องมีถึงจะ split ได้
min_samples_leaf	Hyperparameter ที่กำหนดจำนวน sample ขั้นต่ำที่ leaf node ต้องมี
Accuracy Score	สัดส่วนของการทำนายที่ถูกต้องทั้งหมด ใช้ประเมิน model หลัง training
Logistic Regression	Algorithm ที่ใช้ linear model แต่ output เป็น class (categorical) ใช้สำหรับ classification
Random Forest	Ensemble method ที่รวม Decision Tree หลายต้นเข้าด้วยกัน ลด overfitting ได้ดี
Gradient Boosting Machines (GBM)	Ensemble method ที่สร้าง model อ่อน (weak learners) ทีละตัวแบบต่อเนื่อง เน้นแก้ข้อผิดพลาดของ model ก่อนหน้า
XGBoost	Gradient Boosting ที่มีประสิทธิภาพสูง ใช้ได้ทั้ง classification และ regression
Ensemble Method	วิธีการรวม model หลายๆ ตัวเข้าด้วยกันเพื่อให้ผลลัพธ์ดีขึ้น
CRISP-DM	กรอบขั้นตอนการทำ data science: Business Understanding ⟶ Data Understanding ⟶ Data Preparation ⟶ Modeling ⟶ Evaluation ⟶ Deployment

Lecture 7 — ML: Regression

Keyword	ความหมาย
Regression	การทำนายค่า output ที่เป็นตัวเลขต่อเนื่อง (continuous value) เช่น ราคา, อุณหภูมิ, ยอดขาย
Linear Regression	Regression ที่สมมติว่าความสัมพันธ์ระหว่าง input กับ output เป็นเส้นตรง สูตร: $y = wx + b$
Polynomial Regression	Regression ที่ใช้สมการพหุนาม (x^2, x^3...) เพื่อจับความสัมพันธ์ที่โค้งงอได้
Ridge Regression	Regression ที่เพิ่ม L2 Regularization — ลด overfitting โดยลด weight ให้เล็กลงแต่ไม่เป็น 0
Lasso Regression	Regression ที่เพิ่ม L1 Regularization — สามารถทำให้ weight บางตัวกลายเป็น 0 (feature selection อัตโนมัติ)
Decision Tree Regression	Decision Tree ที่ใช้ทำนายค่า continuous แทนการจำแนก class
Slope (w) / Weight	ค่าสัมประสิทธิ์ที่บอกความชันของเส้น regression ว่า output

	เปลี่ยนไปเท่าไรเมื่อ input เปลี่ยน 1 หน่วย
Intercept (b) / Bias / y-intercept	ค่า output เมื่อ input = 0 คือจุดที่เส้นตัดแกน y
Predicted y ($\hat{y}$)	ค่าที่ model ทำนายออกมาจาก input
Actual y	ค่าจริงในข้อมูล (ground truth)
Error / Residual	ความต่างระหว่าง Predicted y กับ Actual y
Loss Function / Cost Function	ฟังก์ชันที่วัดว่า model ทำนายผิดพลาดมากแค่ไหน ใช้เป็นเป้าหมายในการ minimize ระหว่าง training
SSE (Sum of Squared Errors)	ผลรวมของค่า error ยกกำลังสองทุก data point — Loss function พื้นฐานที่สุด
MSE (Mean Squared Error)	ค่าเฉลี่ยของ SSE หารด้วยจำนวน n สูตร: MSE = SSE/n
RMSE (Root Mean Squared Error)	รากที่สองของ MSE — มีหน่วยเดียวกับ output ตีความง่ายกว่า MSE
MAE (Mean Absolute Error)	ค่าเฉลี่ยของ error สัมบูรณ์ (ไม่ยกกำลัง) — robust ต่อ outlier มากกว่า MSE
Gradient Descent	Algorithm ที่ใช้ optimize model โดยค่อยๆ ปรับ weight ไปในทิศทางที่ลด loss function ได้มากที่สุด
Learning Rate (α)	Hyperparameter ที่กำหนดขนาดของก้าวแต่ละก้าวใน gradient descent - ใหญ่เกินไป = ไม่ converge - เล็กเกินไป = ช้ามาก
Convergence	ภาวะที่ gradient descent หยุดปรับ weight เพราะ loss ลดลงน้อยมากหรือไม่ลดแล้ว
Local Minima	จุดต่ำสุดเฉพาะที่ในพื้นผิว loss function ที่ gradient descent อาจติดอยู่ ไม่ใช่จุดต่ำที่สุดจริง
Global Minimum	จุดต่ำที่สุดจริงๆ ของ loss function — เป้าหมายของ optimization
Batch Gradient Descent	Gradient Descent ที่ใช้ข้อมูลทั้งหมดในการคำนวณ gradient ทุก iteration — แม่นยำแต่ช้าและใช้ memory มาก
Stochastic Gradient Descent (SGD)	Gradient Descent ที่ใช้ data point ทีละ 1 รายการต่อ iteration — เร็วแต่ noisy
Mini-Batch Gradient Descent	Gradient Descent ที่ใช้ข้อมูลย่อย (batch) ต่อ iteration — สมดุลระหว่าง Batch GD และ SGD
Overfitting	Model จำ training data ดีเกินไปจนทำงานได้ไม่ดีกับข้อมูลใหม่ (เส้น regression คดเคี้ยวตาม noise)
Underfitting	Model ง่ายเกินไป ไม่สามารถ capture pattern ในข้อมูลได้ (เส้น regression ตรงเกินไป)
Right Fit	Model ที่ generalize ได้ดี ทำงานได้ดีทั้งกับ training data และ data ใหม่

Regularization	เทคนิคลด overfitting โดยเพิ่ม penalty term เข้าไปใน loss function เพื่อลดขนาด weight
L1 Regularization	เพิ่ม penalty เป็น |w| — สามารถทำให้ weight = 0 ได้ (sparse model) ใช้ใน Lasso
L2 Regularization	เพิ่ม penalty เป็น w^2 — ทำให้ weight เล็กลงแต่ไม่เป็น 0 ใช้ใน Ridge
Coefficient of Determination (R^2)	ค่าวัดว่า model อธิบาย variance ของข้อมูลได้ดีแค่ไหน ค่าอยู่ 0–1 ยิ่งใกล้ 1 ยิ่งดี สูตร: $R^2 = 1 - SSE/SSyy$
SSyy	ผลรวมของค่าความแตกต่างระหว่าง actual y กับ mean ของ y ทั้งหมด (total variance)

Lecture 8 — ML: Parameter Tuning

Keyword	ความหมาย
Parameters	ค่าที่ model เรียนรู้จากข้อมูลเองระหว่าง training เช่น weight (w) และ bias (b) ใน linear regression
Hyperparameters	ค่าที่ผู้ใช้กำหนดก่อน training เช่น learning rate, จำนวน epochs, max_depth — ไม่ได้เรียนรู้จากข้อมูล
Learnable Parameters	อีกชื่อของ Parameters — ค่าที่ model ปรับเองระหว่าง training
Hyperparameter Tuning / HPO	กระบวนการหาค่า hyperparameter ที่ดีที่สุดเพื่อให้ model มี performance สูงสุด (HPO = Hyperparameter Optimization)
Manual Tuning	การปรับ hyperparameter ด้วยมือ อาศัยประสบการณ์และการลองผิดลองถูก
Grid Search	วิธีหา hyperparameter ที่ดีที่สุดโดยทดลองทุก combination ที่กำหนดไว้ใน "grid" — ละเอียดแต่ช้า
Random Search	วิธีหา hyperparameter โดยสุ่ม combination มาทดลอง — เร็วกว่า Grid Search และมักได้ผลดีพอกัน
Automated Tuning	การใช้ library อย่าง Optuna, Hyperopt, scikit-optimize ช่วย tune hyperparameter อัตโนมัติ
Search Space	ช่วงค่าที่กำหนดให้ tuning algorithm ค้นหา hyperparameter ที่เหมาะสม
Suboptimal Settings	ค่า hyperparameter ที่ไม่ดีที่สุด ทำให้ model มี performance ต่ำกว่าที่ควรเป็น
Validation Set	ส่วนของข้อมูลที่แยกออกจาก training set ใช้ประเมิน model ระหว่าง

	tuning (ไม่ใช่ test set)
Test Set	ข้อมูลที่เก็บไว้ประเมินผล model สุดท้าย หลัง tuning เสร็จแล้ว ใช้เพียงครั้งเดียว
Cross-validation	วิธีประเมิน model ที่แบ่งข้อมูลออกเป็นหลาย fold สลับกันเป็น train/validation เพื่อให้การประเมินแม่นยำขึ้น
k-fold Cross-validation	Cross-validation ที่แบ่ง dataset ออกเป็น k fold ใช้ k-1 fold train และ 1 fold validate สลับกัน k รอบ
Learning Rate Scheduling	การปรับ learning rate ให้ลดลงตามเวลา เพื่อให้ gradient descent converge ได้ดีขึ้น
Adaptive Optimizers	Optimizer ที่ปรับ learning rate อัตโนมัติต่างกันในแต่ละ parameter ตัวอย่าง: Adam, RMSProp, Adagrad
Adam	Adaptive optimizer ที่รวม Momentum และ RMSProp เข้าด้วยกัน — นิยมใช้มากที่สุดในปัจจุบัน
RMSProp	Adaptive optimizer ที่ปรับ learning rate ตาม gradient ที่ผ่านมาในแต่ละ parameter
Adagrad	Adaptive optimizer ที่ใช้ learning rate เล็กกับ parameter ที่ update บ่อย และใหญ่กับ parameter ที่ update น้อย
Confusion Matrix	ตารางสรุปผลการจำแนก class — แสดง TP, TN, FP, FN เพื่อประเมินประสิทธิภาพ classifier อย่างละเอียด
TP (True Positive)	ทำนายว่าเป็น Positive และถูกต้อง (positive จริงๆ)
TN (True Negative)	ทำนายว่าเป็น Negative และถูกต้อง (negative จริงๆ)
FP (False Positive) / Type I Error	ทำนายว่าเป็น Positive แต่ผิด (จริงๆ เป็น Negative) — "แจ้งเตือนเท็จ"
FN (False Negative) / Type II Error	ทำนายว่าเป็น Negative แต่ผิด (จริงๆ เป็น Positive) — "พลาดการตรวจจับ"
Accuracy (Acc)	สัดส่วน prediction ที่ถูกต้องทั้งหมด สูตร: (TP+TN) / (TP+TN+FP+FN)
Precision (P)	จากที่ทำนายว่าเป็น Positive ถูกต้องกี่สัดส่วน สูตร: TP / (TP+FP)
Recall (R) / Sensitivity	จาก Positive จริงทั้งหมด model ตรวจจับได้กี่สัดส่วน สูตร: TP / (TP+FN)
F-measure / F1 Score	ค่าเฉลี่ย harmonic mean ของ Precision และ Recall สูตร: 2×P×R / (P+R) — ใช้เมื่อ class ไม่สมดุล
Optuna	Library Python สำหรับ automated hyperparameter optimization — ใช้ง่ายและมีประสิทธิภาพสูง
Hyperopt	Library Python สำหรับ automated hyperparameter optimization —

	ใช้ Bayesian optimization
scikit-optimize	Library Python สำหรับ automated hyperparameter optimization แบบ sequential model-based

Lecture 9 — ML: Clustering

Keyword	ความหมาย
Clustering	การจัดกลุ่มข้อมูลที่คล้ายกันเข้าด้วยกัน โดยไม่มี label กำหนดล่วงหน้า (Unsupervised Learning)
Intra-class Similarity	ความคล้ายคลึงกันของข้อมูลภายใน cluster เดียวกัน — ยิ่งสูงยิ่งดี (cluster compact)
Inter-class Similarity	ความคล้ายคลึงกันของข้อมูลระหว่าง cluster ต่างกัน — ยิ่งต่ำยิ่งดี (cluster แยกชัด)
Intra-cluster Distance	ระยะห่างระหว่างข้อมูลภายใน cluster เดียวกัน — ยิ่งน้อยยิ่งดี
Inter-cluster Distance	ระยะห่างระหว่าง cluster ต่างๆ — ยิ่งมากยิ่งดี
Partitioning Approach	วิธี clustering ที่แบ่งข้อมูลออกเป็น k กลุ่มโดยตรง ต้องระบุ k ล่วงหน้า เช่น K-means
Hierarchical Approach	วิธี clustering ที่สร้างโครงสร้างลำดับชั้น (nested clusters) ไม่ต้องระบุ k ล่วงหน้า
Density-based Approach	วิธี clustering ที่กำหนด cluster จากความหนาแน่นของข้อมูล จัดการ noise/outlier ได้ดี เช่น DBSCAN
K-means Algorithm	Algorithm clustering ที่แบ่งข้อมูลออกเป็น k กลุ่ม โดยวนซ้ำปรับ centroid จนข้อมูลไม่เปลี่ยน cluster
Centroid	จุดศูนย์กลาง (ค่าเฉลี่ย) ของแต่ละ cluster ใน K-means
Seed Points	จุดเริ่มต้นของ centroid ใน K-means ที่สุ่มหรือกำหนดขึ้นก่อน — ส่งผลต่อผลลัพธ์ที่ได้
Brute-force Clustering	วิธีหา cluster ที่ดีที่สุดโดยลองทุก combination — ไม่สามารถทำได้จริงเพราะ compute สูงมาก
SSE (Sum of Squared Errors)	ในบริบท clustering = ผลรวมระยะทางยกกำลังสองจากแต่ละ data point ถึง centroid ของ cluster ตัวเอง
WCSS (Within-Cluster Sum of Squares)	ผลรวมระยะทาง squared ของทุก point ถึง centroid ใน cluster ของตัวเอง — ใช้วัดคุณภาพ clustering (เท่ากับ SSE)

Elbow Method	วิธีหาจำนวน k ที่เหมาะสมโดยพลอต WCSS vs k แล้วหาจุดที่กราฟ "งอ" คล้ายข้อศอก
Elbow Point	จุดที่กราฟ Elbow Method เริ่มลดลงช้าลงอย่างชัดเจน — บ่งบอก k ที่เหมาะสมที่สุด
Silhouette Score	ค่าวัดคุณภาพ clustering ตั้งแต่ -1 ถึง 1 บอกว่า data point อยู่ใน cluster ที่เหมาะสมมากแค่ไหน ยิ่งใกล้ 1 ยิ่งดี
k-medoids	คล้าย K-means แต่ใช้ data point จริงเป็น "center" (medoid) แทน centroid ที่คำนวณ — robust ต่อ outlier มากกว่า
DBSCAN	Density-Based Spatial Clustering of Applications with Noise — clustering โดยหากลุ่มจากความหนาแน่น จัดการ noise/outlier ได้ ไม่ต้องระบุ k ล่วงหน้า
Agglomerative Clustering	Hierarchical clustering แบบ bottom-up: เริ่มจาก data point แต่ละตัวเป็น cluster ของตัวเอง แล้วค่อยๆ merge รวมกัน
Divisive Clustering	Hierarchical clustering แบบ top-down: เริ่มจาก cluster เดียวแล้วค่อยๆ แบ่งออก
Dendrogram	แผนภาพต้นไม้ที่แสดงผลของ Hierarchical Clustering — ใช้ตัดสินว่าจะแบ่งกี่ cluster
Nested Clusters	โครงสร้าง cluster ที่ซ้อนกันเป็นลำดับชั้น ได้จาก Hierarchical Clustering
Proximity Matrix	ตารางที่เก็บระยะห่าง (distance) ระหว่าง data points ทุกคู่ ใช้ใน Hierarchical Clustering