Tutorial session I: Binary Classification and Model Evaluation Metrics (การจำแนกประเภทแบบทวิภาคและการประเมินผลแบบจำลอง)

วัตถุประสงค์:

1. ฝึกการสร้างแบบจำลองการจำแนกประเภทแบบทวิภาค
2. ทำความเข้าใจและคำนวณ TP, TN, FP, FN
3. คำนวณค่า Accuracy, Precision, Recall, และ F1-Score

คำแนะนำในการทำแลบ:

1. เปิด Google Colab.
2. คัดลอกและวางโค้ดด้านล่างนี้ลงในโน้ตบุ๊กใหม่
3. รันแต่ละเซลล์เพื่อดูผลลัพธ์และทำตามคำอธิบายที่ให้ไว้

Code
import numpy as np import pandas as pd from sklearn.model_selection import train_test_split from sklearn.linear_model import LogisticRegression from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score from sklearn.datasets import load_breast_cancer data = load_breast_cancer() X = data.data y = data.target # Binary target: 0 = malignant, 1 = benign X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42) model = LogisticRegression(max_iter=10000) model.fit(X_train, y_train) y_pred = model.predict(X_test)

Code

```
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)

TN, FP, FN, TP = conf_matrix.ravel()

print("\nMetrics Breakdown:")
print("True Positives (TP):", TP)
print("True Negatives (TN):", TN)
print("False Positives (FP):", FP)
print("False Negatives (FN):", FN)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("\nEvaluation Metrics:")
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-Score:", f1)
```

4. ให้นักศึกษาเพิ่มโค้ดต่อไปนี้ในเซลล์ใหม่ใน Google Colab หลังจากโค้ดของ Logistic Regression เพื่อดูการทำงานของโมเดล Decision Tree ในการจำแนกประเภท

Code
from sklearn.tree import DecisionTreeClassifier print("\nUsing Decision Tree Classifier:") dt_model = DecisionTreeClassifier(random_state=42) dt_model.fit(X_train, y_train) y_pred_dt = dt_model.predict(X_test) conf_matrix_dt = confusion_matrix(y_test, y_pred_dt) accuracy_dt = accuracy_score(y_test, y_pred_dt) precision_dt = precision_score(y_test, y_pred_dt) recall_dt = recall_score(y_test, y_pred_dt) f1_dt = f1_score(y_test, y_pred_dt) print("Confusion Matrix (Decision Tree):\n", conf_matrix_dt) print("Accuracy (Decision Tree):", accuracy_dt) print("Precision (Decision Tree):", precision_dt) print("Recall (Decision Tree):", recall_dt) print("F1-Score (Decision Tree):", f1_dt)

5. ให้นักศึกษาเพิ่มโค้ดต่อไปนี้ในเซลล์ใหม่ใน Google Colab หลังจากโค้ดของ Logistic Regression เพื่อดูการทำงานของโมเดล K-Nearest Neighbors (KNN) ในการจำแนกประเภท

Code

```
from sklearn.neighbors import KNeighborsClassifier


print("\nUsing K-Nearest Neighbors Classifier:")


knn_model = KNeighborsClassifier(n_neighbors=5)  # กำหนดจำนวน neighbors เป็น 5
knn_model.fit(X_train, y_train)


y_pred_knn = knn_model.predict(X_test)


conf_matrix_knn = confusion_matrix(y_test, y_pred_knn)
accuracy_knn = accuracy_score(y_test, y_pred_knn)
precision_knn = precision_score(y_test, y_pred_knn)
recall_knn = recall_score(y_test, y_pred_knn)
f1_knn = f1_score(y_test, y_pred_knn)


print("Confusion Matrix (K-Nearest Neighbors):\n", conf_matrix_knn)
print("Accuracy (K-Nearest Neighbors):", accuracy_knn)
print("Precision (K-Nearest Neighbors):", precision_knn)
print("Recall (K-Nearest Neighbors):", recall_knn)
print("F1-Score (K-Nearest Neighbors):", f1_knn)
```

Lab Questions

คำสั่ง: อ่านและทำความเข้าใจโค้ดทั้งหมด จากนั้นอธิบายและอภิปรายในประเด็นต่าง ๆ ตามข้อกำหนดต่อไปนี้:

1. อธิบายโค้ดการนำเข้าลิไบรารีและการเตรียมข้อมูล

- **คำถาม**:

    1. ลิไบรารีที่นำเข้า (numpy, pandas, train_test_split, LogisticRegression, confusion_matrix, ฯลฯ) มีหน้าที่อะไรในโค้ดนี้?
    2. ชุดข้อมูล **Breast Cancer** จาก sklearn.datasets ประกอบด้วยอะไรบ้าง
    3. การแบ่งข้อมูลเป็นชุดฝึก (X_train, y_train) และชุดทดสอบ (X_test, y_test) ด้วย train_test_split มีความสำคัญอย่างไร?

2. การฝึกและประเมินผลโมเดล Logistic Regression

- **คำถาม**:

    1. Logistic Regression คืออะไร และเพราะเหตุใดจึงเหมาะสำหรับปัญหาการจำแนกประเภทแบบทวิภาค?
    2. อธิบายขั้นตอนการฝึกโมเดล Logistic Regression ด้วยคำสั่ง .fit() และวิธีการทำนายผลด้วยคำสั่ง .predict()
    3. การใช้ confusion_matrix มีบทบาทอย่างไรในโค้ด และค่าต่าง ๆ (TP, TN, FP, FN) บอกอะไรเกี่ยวกับความแม่นยำของโมเดล?
    4. เมตริกการประเมิน เช่น Accuracy, Precision, Recall, และ F1-Score คืออะไร และทำไมถึงจำเป็นต้องคำนวณ?

3. การฝึกและประเมินผลโมเดล Decision Tree

- **คำถาม**:

    1. Decision Tree ทำงานอย่างไร และต่างจาก Logistic Regression อย่างไร?
    2. การใช้ DecisionTreeClassifier ช่วยให้เราสามารถมองเห็นการแบ่งประเภทข้อมูลได้อย่างไรบ้าง?
    3. การเปรียบเทียบค่าเมตริกการประเมินของ Logistic Regression และ Decision Tree มีความแตกต่างกันอย่างไร?

4. การฝึกและประเมินผลโมเดล K-Nearest Neighbors (KNN)

- **คำถาม**:

    1. อธิบายหลักการของ K-Nearest Neighbors และวิธีการทำงานของ KNeighborsClassifier
    2. ทำไมเราต้องกำหนด n_neighbors=5 ในโมเดล KNN และการปรับค่านี้จะส่งผลต่อผลลัพธ์อย่างไร?
    3. การเปรียบเทียบเมตริกการประเมินของ KNN กับโมเดลอื่น ๆ ที่ใช้อยู่ก่อนหน้า มีข้อแตกต่างกันอย่างไรบ้าง?

5. การอภิปรายผลลัพธ์และการเลือกโมเดลที่ดีที่สุด

- **คำถาม**:

    1. เมตริกใดในบรรดา Accuracy, Precision, Recall, และ F1-Score ที่มีความสำคัญที่สุดในบริบทของการจำแนกประเภทในชุดข้อมูลนี้? เพราะเหตุใด?
    2. การเลือกโมเดลที่เหมาะสม (Logistic Regression, Decision Tree, KNN) ควรพิจารณาจากอะไรบ้าง เช่น ข้อมูลที่มี, ความเร็วในการประมวลผล, หรือความซับซ้อนของปัญหา?
    3. หากต้องปรับปรุงความแม่นยำของโมเดลเพิ่มเติม คิดว่าจะปรับแต่งโมเดลอย่างไร เช่น การใช้วิธีการ Cross-Validation หรือการปรับ Hyperparameters