



รายงาน

การประยุกต์ใช้ Machine Learning เพื่อสนับสนุนการตัดสินใจเลือกวิชาเลือก
(ระบบทำนายผลลัพธ์วิชาเลือก)

ผู้จัดทำ

นางสาว สุตารัตน์ สุขอยู่ รหัสนิสิต 66163297

อาจารย์ที่ปรึกษา

ผศ.ดร.กัมปนาท ปิยะธำรงชัย

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชาระบบภูมิสารสนเทศสนับสนุนการตัดสินใจ

ภาคเรียนที่ 1 ปีการศึกษา 2568

มหาวิทยาลัยนเรศวร

สารบัญ

	หน้า
ที่มาและความสำคัญของปัญหา	1
วัตถุประสงค์	1
ขั้นตอนและวิธีการดำเนินการ	2
วิธีการใช้งาน	6
- ตัวอย่างผลลัพธ์	7
เอกสารอ้างอิง	9

ที่มาและความสำคัญของปัญหา

บริบทของปัญหา

ในการศึกษาระดับมหาวิทยาลัย การเลือกวิชาเลือก (Elective Courses) เป็นการตัดสินใจที่มีความสำคัญอย่างยิ่งต่อนิสิต เนื่องจากส่งผลต่อ (1) ภาระงาน (Workload), (2) ระดับความยากง่าย (Difficulty), และ (3) ผลการเรียนโดยรวม (Overall GPA)

ความไม่แน่นอนในการตัดสินใจ

นิสิตมักเผชิญกับความท้าทายในการประเมินวิชาเลือก โดยส่วนใหญ่อาศัยเพียงคำบอกเล่าหรือข้อมูลรีวิวที่ไม่เป็นระบบ ทำให้เกิดความเสี่ยงในการลงทะเบียนวิชาที่อาจมีภาระงานสูงหรือยากเกินความสามารถ ซึ่งนำไปสู่โอกาสที่จะได้ผลการเรียนที่ไม่พึงพอใจ (เช่น 'Result' เป็น 0 ในข้อมูลของคุณ)

ความสำคัญของการวิเคราะห์ด้วย ML

โครงการนี้จึงมีความสำคัญในการนำเทคนิค Machine Learning (ML) มาวิเคราะห์ความสัมพันธ์ระหว่างปัจจัยต่างๆ ของวิชา (เช่น ความยาก, ภาระงาน, อาจารย์ผู้สอน) กับผลลัพธ์ที่ได้จริง

การสนับสนุนการตัดสินใจ (DSS)

การพัฒนาระบบสนับสนุนการตัดสินใจ (Decision Support System - DSS) ด้วย ML จะช่วยให้นิสิตสามารถป้อนข้อมูลวิชาที่สนใจ และได้รับ การทำนายผลลัพธ์และความน่าจะเป็นล่วงหน้า ซึ่งเป็นการช่วยลดความเสี่ยงและเพิ่มโอกาสในการเลือกวิชาที่เหมาะสมกับตนเองมากที่สุด (อาจมีการอ้างอิงถึงความสำคัญของ DSS ในยุค Data-Driven Society)

วัตถุประสงค์

1. เพื่อศึกษาและวิเคราะห์ปัจจัยที่มีผลต่อผลลัพธ์การลงทะเบียนวิชาเลือก (Result) โดยใช้ข้อมูลจริง
2. เพื่อสร้างและฝึกฝนโมเดล Machine Learning (เช่น Pipeline Model) ที่มีความแม่นยำในการทำนายผลลัพธ์การลงทะเบียนวิชาเลือก
3. เพื่อพัฒนาแอปพลิเคชันสนับสนุนการตัดสินใจ (DSS) โดยใช้ Streamlit เพื่อให้นิสิตสามารถทดลองทำนายผลลัพธ์ได้อย่างง่ายดาย

ขั้นตอนและวิธีการดำเนินการ

1. การรวบรวมข้อมูล (Data Collection)

- **แหล่งข้อมูล** ข้อมูลวิชาเลือกจาก (ระบุแหล่งที่มาของไฟล์ electives_data.csv เช่น การสำรวจนิสิต, ข้อมูลรีวิว)
- **ข้อมูลตั้งต้น** ไฟล์ electives_data.csv ประกอบด้วยคอลัมน์สำคัญ เช่น course_code, course_name, difficulty, workload, professor, และ result (ผลลัพธ์)

2. การเตรียมข้อมูลและปรับปรุงข้อมูล (Data Preprocessing)

- **การจัดการข้อมูลสูญหาย/ผิดปกติ** ใช้เทคนิคการทำความสะอาดข้อมูล (Data Cleaning)
- **การแปลงข้อมูล (Encoding)** ทำการแปลงข้อมูลเชิงหมวดหมู่ (Categorical Features) ให้เป็นตัวเลขที่โมเดลเข้าใจ
 - สำหรับตัวแปรเป้าหมาย (result) ใช้ **Label Encoder** (เช่น le_target ในโค้ด) เพื่อแปลงผลลัพธ์ (เช่น Pass/Fail หรือ 1/0)
 - สำหรับตัวแปรต้น (Features) ใช้ **One-Hot Encoding** สำหรับข้อมูลหมวดหมู่ เช่น course_name, professor ภายใน Scikit-Learn Pipeline
- **การปรับมาตรฐาน (Scaling/Normalization)** ใช้ StandardScaler สำหรับตัวแปรเชิงปริมาณ เช่น difficulty, workload เพื่อปรับช่วงของข้อมูล

3. การสร้างและฝึกฝนโมเดล (Model Building and Training)

- **เทคนิค ML** เลือกใช้โมเดล (ระบุโมเดลที่คุณใช้ เช่น Decision Tree, Random Forest, Logistic Regression)
- **Pipeline** สร้าง Scikit-learn **Pipeline** (best_pipe) เพื่อรวมขั้นตอน Preprocessing และ Model เข้าไว้ด้วยกัน (เช่น ColumnTransformer + Classifier) เพื่อความสะดวกและลดโอกาสเกิด Data Leakage
- **การฝึกฝน** นำ Pipeline มาฝึกฝนด้วยชุดข้อมูลฝึก (Training Data)

4. การประเมินผลและการปรับแต่งโมเดล (Evaluation and Tuning)

- ประเมินประสิทธิภาพของโมเดลโดยใช้เมตริกที่เหมาะสม (เช่น Accuracy, Precision, Recall, F1-Score)
- ใช้เทคนิค Cross-Validation หรือ Hyperparameter Tuning เพื่อหาโมเดลที่ดีที่สุด (best_pipe)

5. การนำไปใช้งาน (Deployment)

- การบันทึกโมเดล ใช้ `joblib.dump` เพื่อบันทึกโมเดล Pipeline, Label Encoder, และรายชื่อฟีเจอร์ลงในไฟล์ `best_model_joblib.pkl` ในรูปแบบ Dictionary เพื่อให้แอปพลิเคชันสามารถโหลดมาใช้งานได้อย่างครบถ้วน
- การสร้างแอปพลิเคชัน ใช้ **Streamlit** (ไฟล์ `app.py`) ในการสร้าง User Interface สำหรับรับข้อมูล Input และแสดงผลการทำนายแบบ Real-time
- อธิบายปัจจัยต่างๆ ที่นิสิตเลือก และอธิบายโค้ดที่นิสิตเขียนเป็นขั้นตอน

1. ปัจจัย (Features) ที่ใช้ในการวิเคราะห์

(อธิบายคอลัมน์ที่คุณเลือกจากไฟล์ `cleaned_data_for_report.csv` หรือ `electives_data.csv`)

ปัจจัย (Feature)	ประเภทข้อมูล	คำอธิบายความหมายในการวิเคราะห์
difficulty	ตัวเลข (Ordinal/Scale)	ระดับความยากของเนื้อหาวิชา (เช่น 1 = ง่ายมาก, 5 = ยากมาก)
workload	ตัวเลข (Ordinal/Scale)	ภาระงานที่ต้องทำในวิชา เช่น ปริมาณการบ้าน, รายงาน, งานกลุ่ม (เช่น 1 = น้อยมาก, 5 = มากที่สุด)
professor	ตัวเลข (Ordinal/Scale)	อาจารย์ผู้สอนวิชานั้นๆ เป็นปัจจัยสำคัญต่อสไตล์การสอนและการวัดผล (เช่น สอนดีมาก = 5, สอนไม่ดี = 1)
course_name	หมวดหมู่ (Categorical)	ชื่อวิชา (บ่งชี้ถึงเนื้อหาเฉพาะทางของวิชานั้นๆ)
ตัวแปรเป้าหมาย (result)	หมวดหมู่ (Binary)	ผลลัพธ์การลงทะเบียน (1 = ลงทะเบียนสำเร็จ/ผ่าน, 0 = ไม่สำเร็จ/ไม่ผ่าน/ถอน)

2. คำอธิบายโค้ด (ไฟล์ app.py)

(อธิบายโค้ดตามหน้าที่หลักที่เขียนไว้)

ขั้นตอน	โค้ดที่สำคัญ	คำอธิบายหน้าที่ของโค้ด
A. การตั้งค่าและโหลดโมเดล	import streamlit as st; import joblib	นำเข้าไลบรารีที่จำเป็นและตั้งค่าหน้าเว็บ
	model_data = joblib.load(PICKLE)	โหลดไฟล์โมเดล best_model_joblib.pkl ที่ บรรจุโมเดล (pipe), Label Encoder (le_target), และ Features
	การจัดการ Error	ตรวจสอบว่า model_data เป็น Dictionary และมีคีย์ 'pipeline' เพื่อป้องกันข้อผิดพลาด KeyError ที่เคยเกิดขึ้น
B. การรับ Input จาก ผู้ใช้	st.markdown("### กรอกข้อมูลตัวแปร ...")	สร้างหัวข้อสำหรับรับข้อมูล
	for feat in features:	วนลูปตามรายชื่อ Features ที่โหลดมาจาก ไฟล์โมเดล
	st.number_input(...)	สร้างช่อง Input สำหรับตัวแปรที่เป็นตัวเลข (เช่น difficulty, workload)
	st.text_input(...)	สร้างช่อง Input สำหรับตัวแปรที่เป็นข้อความ (เช่น professor, course_name)
C. การทำนายและ แสดงผล	if st.button("Submit / Predict"):	ตรวจสอบเมื่อผู้ใช้กดปุ่มทำนาย
	input_df = pd.DataFrame(...)	แปลงข้อมูล Input ที่ได้รับจากผู้ใช้ให้เป็น DataFrame ที่ถูกต้องตามรูปแบบที่โมเดล ต้องการ
	pred = pipe.predict(input_df)	การเรียกใช้โมเดล: ทำการทำนายผลลัพธ์ด้วย โมเดล Pipeline
	label = le_target.inverse_transform(pred)[0]	แปลงผลลัพธ์ตัวเลข (Class ID) กลับไปเป็น Label ที่มีความหมาย (เช่น 0 เป็น "ไม่ สำเร็จ", 1 เป็น "สำเร็จ")
	st.success(f" ผลการทำนาย: ** {label}**")	แสดงผลลัพธ์การทำนายให้ผู้ใช้เห็น

• แสดงผลลัพธ์ที่ได้โดยละเอียด และการนำไปประยุกต์ใช้

1. ผลลัพธ์จากการประเมินโมเดล (Model Evaluation Results)

เมตริก (Metric)	ค่าที่ได้ (Value)	การตีความ
Accuracy Score	(ระบุค่าที่ได้จากการทดสอบ)	ความแม่นยำโดยรวมของโมเดลในการทำนายที่ถูกต้อง
Precision (Class 0/Fail)	(ระบุค่าที่ได้จากการทดสอบ)	สัดส่วนของการทำนาย "ไม่สำเร็จ" ที่ถูกต้องเมื่อเทียบกับการทำนาย "ไม่สำเร็จ" ทั้งหมด
Recall (Class 1/Success)	(ระบุค่าที่ได้จากการทดสอบ)	สัดส่วนของการทำนาย "สำเร็จ" ที่ถูกต้องเมื่อเทียบกับกรณี "สำเร็จ" จริงทั้งหมด
Confusion Matrix	(อธิบายเป็นตาราง 2x2 หรือรูปภาพ)	แสดงจำนวน True Positives, True Negatives, False Positives, และ False Negatives
ปัจจัยที่มีอิทธิพล	(ระบุ Feature Importance 3 อันดับแรก)	ปัจจัยใดที่มีผลต่อผลลัพธ์การลงทะเบียนมากที่สุด (เช่น difficulty, professor)

2. การนำไปประยุกต์ใช้ (Application and Impact)

- การวางแผนการลงทะเบียนส่วนบุคคล นิสิตสามารถใช้แอปพลิเคชันนี้เพื่อทดสอบชุดวิชาที่ตนเองสนใจ เพื่อประเมินความเสี่ยงล่วงหน้าและปรับเปลี่ยนแผนการลงทะเบียนเพื่อเพิ่มโอกาสในการประสบความสำเร็จ
- เครื่องมือช่วยแนะนำ สามารถขยายผลเป็นระบบแนะนำ (Recommendation System) ที่แนะนำชุดวิชาเลือกที่มีความเสี่ยงต่ำที่สุดตามโปรไฟล์ของนิสิตแต่ละคน
- การปรับปรุงหลักสูตร ข้อมูลและปัจจัยที่โมเดลระบุว่ามียุทธูปสรรคสูง (เช่น ภาระงานที่ไม่สมเหตุผล) สามารถเป็นข้อมูลเชิงลึกให้กับคณะ/ภาควิชา เพื่อใช้ในการปรับปรุงหรือออกแบบวิชาเลือกให้มีคุณภาพและสมดุลมากขึ้น

วิธีการใช้งาน

แอปพลิเคชันนี้จะทำงานตามข้อมูลที่คุณป้อนเข้าไป แล้วใช้โมเดล `best_model_joblib.pkl` ที่คุณสร้างไว้มาประมวลผล

1. ส่วน "กรอกข้อมูลตัวแปร (features)"

ส่วนนี้คือส่วนที่คุณจะต้องป้อนค่าสำหรับฟีเจอร์ (Features) ที่โมเดลของคุณใช้ในการทำนาย ซึ่งอ้างอิงจากรายการคอลัมน์ในไฟล์โมเดลของคุณ (เช่น `difficulty`, `workload`, `professor`, `course_name`)

- **ฟีเจอร์ที่เป็นตัวเลข** เช่น `difficulty` หรือ `workload` (ถ้ามีค่าเป็นตัวเลข) คุณจะเห็นช่องให้กรอกเป็นตัวเลข โดยมีค่าเริ่มต้น (default value) คือ 1 - 5
- **ฟีเจอร์ที่ไม่ใช่ตัวเลข** เช่น `professor` หรือ `course_name` คุณจะเห็นช่องให้กรอกเป็นข้อความ (Text Input)

สิ่งที่คุณต้องทำ

- **ป้อนข้อมูล** กรอกค่าที่ต้องการทำนายลงในช่องว่างแต่ละช่องให้ครบถ้วน
 - `difficulty` ความยาก/ง่ายของวิชานั้นๆ กรอกเป็นตัวเลข โดยค่าเริ่มต้นคือ 1 – 5 (น้อยไปมาก)
 - `workload` งานที่อาจารย์มอบหมาย กรอกเป็นตัวเลข โดยค่าเริ่มต้นคือ 1 – 5 (น้อยไปมาก)
 - `professor` ความน่าสนใจ/สไตล์วิธีการสอนของอาจารย์วิชานั้นๆ กรอกเป็นตัวเลข โดยค่าเริ่มต้นคือ 1 – 5 (น้อยไปมาก)

2. ส่วน "Submit / Predict"

เมื่อคุณกรอกข้อมูลครบแล้ว ให้กดปุ่ม "Submit / Predict"

ผลลัพธ์ที่ได้

1. ผลการทำนาย (Prediction) ระบบจะแสดงผลที่โมเดลทำนายออกมา

- ถ้าโฟลโมเดลมี Label Encoder (le_target) ที่สมบูรณ์ มันจะแสดงเป็นชื่อคลาส/ผลลัพธ์ที่อ่านง่าย เช่น "Pass" หรือ "Fail"
- ถ้าโฟลโมเดลไม่สมบูรณ์ (ขาด Label Encoder) มันจะแสดงเป็นรหัสตัวเลขแทน เช่น "Class ID: 0" หรือ "Class ID: 1"
- ผลลัพธ์คือ 1 แสดงว่า น่าเรียน/น่าสนใจ ควรลงทะเบียน
- ผลลัพธ์คือ 0 แสดงว่า ไม่น่าสนใจ ควรทำการถอน

2. ความน่าจะเป็น (Probability) โมเดลจะแสดงตารางที่แสดงว่าผลลัพธ์แต่ละคลาสมีความน่าจะเป็นเท่าไร (เป็นเปอร์เซ็นต์)

ตัวอย่างผลลัพธ์ คือ 0 แสดงว่า ไม่น่าสนใจ ควรทำการถอน

miniproject2568.pdf x DSS ML Mini Project x DSS ML Mini Project x DSS ML Mini Project x +

localhost:8501

Import favorites Gmail YouTube Maps McAfee Security

Deploy

ระบบทำนายผลลัพธ์วิชาเลือก

กรอกข้อมูลตัวแปร (features)

difficulty
3 - +

workload
4 - +

professor
3

Submit / Predict

Predicted: 0

Probabilities:

- 0: 1.000
- 1: 0.000

ตัวอย่างผลลัพธ์ คือ 1 แสดงว่า น่าเรียน/น่าสนใจ ควรลงทะเบียน

miniproject2568.pdf x DSS ML Mini Project x DSS ML Mini Project x DSS ML Mini Project x +

localhost:8501

Import favorites Gmail YouTube Maps McAfee Security

Deploy

ระบบทำนายผลลัพธ์วิชาเลือก

กรอกข้อมูลตัวแปร (features)

difficulty

2 - +

workload

3 - +

professor

5

Submit / Predict

Predicted: 1

Probabilities:

- 0: 0.000
- 1: 1.000

เอกสารอ้างอิง

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, V., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). **Scikit-learn: Machine Learning in Python**. *Journal of Machine Learning Research*, 12, 2825–2830. (สำหรับอ้างอิงการใช้ไลบรารี Scikit-Learn)
2. Streamlit Inc. (Latest version). **Streamlit: The fastest way to build data apps**. [Online]. Available: <https://streamlit.io> (Accessed: 9 November 2025). (สำหรับอ้างอิงการใช้งาน Streamlit ในการพัฒนาแอปพลิเคชัน DSS)
3. Trivedi, S., & Sharma, S. (2020). **Predicting Student Academic Performance using Decision Support System: A Machine Learning Approach**. *International Journal of Computer Applications (0975 – 8887)*, 175(13), 32-38. (สำหรับอ้างอิงเอกสารวิชาการที่เกี่ยวข้องกับการประยุกต์ใช้ DSS และ ML ในบริบทการศึกษา)
4. ผู้จัดทำโครงการ. (2568). **ข้อมูลวิชาเลือก (electives_data.csv) ที่ใช้ในการวิเคราะห์**. (ข้อมูลรวบรวมจากการสำรวจนิสิต, หรือข้อมูลจำลองที่สร้างขึ้นเพื่อวัตถุประสงค์ในการทำโครงการ). (สำหรับอ้างอิงแหล่งที่มาของชุดข้อมูลหลักที่ใช้ในโครงการ)