

การใช้วิศวกรรมข้อความร่วมกับปัญญาประดิษฐ์เชิงสร้างสรรค์เพื่อการวิเคราะห์ ข้อมูลภาพถ่ายเทียมเซนทิเนล-2 บนแพลตฟอร์มกูเกิลเอิร์ธเอนจิน Using Prompt Engineering with Generative AI for the Analysis of Sentinel-2 Satellite Imagery on Google Earth Engine Platform

ศุภกร เอี่ยมสุนทรกุล¹ ชาติชาย ไวยสุระสิงห์^{2*} และชุตินา ไวยสุระสิงห์³

Suphakorn Iamsuntornkul¹ Chattichai Waisurasingha^{2*} and Chutima Waisurasingha³

Received: September 29, 2025; Revised: December 9, 2025; Accepted: December 9, 2025

บทคัดย่อ

การศึกษานี้มีวัตถุประสงค์เพื่อประเมินศักยภาพของวิศวกรรมข้อความ (Prompt Engineering) ร่วมกับปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative AI) ในการแปลงภาษาธรรมชาติให้เป็นซอร์สโค้ดภาษาจาวาสคริปต์ (JavaScript) สำหรับใช้งานกูเกิลเอิร์ธเอนจิน (Google Earth Engine: GEE) เพื่อวิเคราะห์ข้อมูลภาพถ่ายเทียมเซนทิเนล-2 (Sentinel-2) สำหรับการจำแนกประเภทการใช้ที่ดินในพื้นที่อำเภอเมืองขอนแก่น โดยใช้ตัวจำแนกแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine: SVM) ซึ่งผลลัพธ์จากการศึกษาครั้งนี้พบว่า ซอร์สโค้ดที่ได้จากการทำงานร่วมกันระหว่างวิศวกรรมข้อความร่วมกับ Generative AI นั้นสามารถนำมาใช้งานประมวลผลข้อมูลภูมิสารสนเทศขนาดใหญ่บน GEE ได้เป็นอย่างดี โดยมีค่าความถูกต้องโดยรวม (Overall Accuracy) ร้อยละ 87 และค่าสัมประสิทธิ์แคปป่า (Kappa Coefficient) เป็น 0.84 จากการศึกษานี้พบว่า ประสิทธิภาพของผลลัพธ์จะออกมาดีได้นั้นส่วนหนึ่งขึ้นอยู่กับคุณภาพข้อมูลที่ใช้สำหรับใช้สอนตัวจำแนกแบบ SVM และอีกส่วนหนึ่งขึ้นอยู่กับคุณภาพของข้อความ (Prompt) ที่จัดเป็นภาษาธรรมชาติที่ใช้ในการทำงานร่วมกับจาก Generative AI ในการผลิตซอร์สโค้ดภาษา JavaScript และองค์ความรู้ของผู้ใช้ในด้านภูมิสารสนเทศซึ่งมีความสำคัญอย่างยิ่งในการออกแบบคำสั่ง การเลือกดัชนี การเลือกตัวอย่างซอร์สโค้ด และการตีความผลลัพธ์ ดังนั้น การวิจัยครั้งนี้แสดงศักยภาพของ Generative AI ร่วมกับเทคนิค Prompt Engineering เป็นเครื่องมือสนับสนุนการสร้างซอร์สโค้ดสำหรับวิเคราะห์ข้อมูลเชิงพื้นที่บน GEE ทำให้ลดภาระด้านเทคนิคและขยายโอกาสในการเข้าถึงเทคโนโลยีการวิเคราะห์ข้อมูลเชิงพื้นที่ให้แก่ผู้ใช้งานทั่วไปผ่านการใช้ภาษาธรรมชาติได้อย่างมีประสิทธิภาพ

คำสำคัญ : ภาษาธรรมชาติ; วิศวกรรมข้อความ; ฐานข้อมูลภูมิสารสนเทศขนาดใหญ่; กูเกิลเอิร์ธเอนจิน

¹ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

² ศูนย์วิจัยและพัฒนาโครงสร้างพื้นฐานอย่างยั่งยืน มหาวิทยาลัยขอนแก่น

³ คณะศิลปศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

¹ Faculty of Engineering, Khon Kaen University

² Sustainable Infrastructure Research and Development Center, Khon Kaen University

³ School of Liberal Arts, King Mongkut's Institute of Technology Ladkrabang

* Corresponding Author, Tel. 06 5145 1994, E - mail: fcecws@kku.ac.th

Abstract

This study aims to evaluate the potential of prompt engineering combined with generative AI to translate natural language into JavaScript code for Google Earth Engine (GEE), enabling land use classification of Sentinel-2 imagery in Mueang Khon Kaen District using a Support Vector Machine (SVM) classifier. The results demonstrated that code generated through prompt engineering and Generative AI could be effectively utilized for processing large-scale geospatial data on GEE, achieving 87 % overall accuracy and a Kappa coefficient of 0.84. The effectiveness of the results depended on two primary factors: (1) the quality of training data for the SVM classifier, and (2) the quality of natural language prompts used with Generative AI. Additionally, the user's geospatial expertise proved essential for optimizing command design, selecting appropriate indices, sampling code examples, and interpreting results. This research demonstrates that Generative AI combined with prompt engineering can serve as a valuable tool for supporting code generation for spatial data analysis on GEE, reducing technical barriers and expanding access to spatial data analysis technology for general users through effective natural language interaction.

Keywords: Natural Language; Prompt Engineering; Geospatial Big Data; Google Earth Engine

บทนำ

ยุคแห่งปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative Artificial Intelligence Era: Generative AI Era) เป็นช่วงเวลา ณ จุดเปลี่ยนสำคัญที่เปลี่ยนผ่านจากยุคอุตสาหกรรม 4.0 สู่ยุคอุตสาหกรรม 5.0 ซึ่งเป็นการเปลี่ยนผ่านจากยุคที่มีการนำปัญญาประดิษฐ์มาใช้เพื่อเพิ่มประสิทธิภาพในกระบวนการผลิตแบบอัตโนมัติ ไปสู่ยุคที่มีการนำ Generative AI มาทำงานร่วมกับมนุษย์ (Human-Centric) เพื่อสร้างนวัตกรรม เพิ่มความยืดหยุ่น และออกแบบแนวทางแก้ปัญหาที่ตอบโจทย์เฉพาะบุคคล โดยจุดเปลี่ยนนี้ได้เริ่มต้นชัดเจนขึ้นบริษัท OpenAI ได้เปิดให้คนทั่วโลกทดลองใช้ Generative AI อย่าง ChatGPT ที่พัฒนาแบบจำลองภาษาขนาดใหญ่ (Large Language Models: LLMs) ให้ทั่วโลกได้ใช้งานแล้วหลังจากนั้นไม่นานได้มีการเปิดตัวในเชิงพาณิชย์อย่างเต็มรูปแบบ ซึ่งได้มีการพัฒนาอย่างต่อเนื่องจนมีขีดความสามารถในการพัฒนาซอร์สโค้ด (Source Code) จากภาษาธรรมชาติ (Natural Language) ที่ให้ผลลัพธ์ที่มีโครงสร้างที่ดีและสามารถทำงานที่ซับซ้อนได้ในระดับหนึ่ง (Denny et al., 2024) ซึ่งขีดความสามารถนี้ได้มีการบรรจุไว้ใน Generative AI อื่น ๆ เช่น Claude AI, Gemini และ Copilot ด้วยเช่นกัน และด้วยขีดความสามารถดังกล่าว Generative AI จึงกลายเป็นเครื่องมือที่มีบทบาทต่อการทำงานของมนุษย์ส่วนใหญ่ในปัจจุบัน จึงเป็นที่น่าสนใจอย่างยิ่งที่จะนำมาเป็นตัวช่วยสำหรับการวิเคราะห์ฐานข้อมูลภูมิสารสนเทศขนาดใหญ่ (Geospatial Big Data) อย่าง Google Earth Engine (GEE) ซึ่งเป็นแพลตฟอร์มคลาวด์ที่กำลังได้รับความนิยมอย่างสูงในปัจจุบัน (Tamiminia et al., 2020)

Google Earth Engine (GEE) เป็นระบบฐานข้อมูลภูมิสารสนเทศขนาดใหญ่ที่เผยแพร่บนแพลตฟอร์มคลาวด์ โดยเป็นการรวมฐานข้อมูลขนาดใหญ่ของภาพถ่ายดาวเทียมและเครื่องมือสำหรับการวิเคราะห์ประมวลผลเข้าไว้ด้วยกัน จึงทำให้การวิเคราะห์ข้อมูลเชิงพื้นที่ ณ ตำแหน่งต่าง ๆ ในโลกนั้นสามารถทำได้อย่างมีประสิทธิภาพ ส่งผลให้ GEE ได้กลายเป็นเครื่องมือสำคัญในการศึกษาการเปลี่ยนแปลงสิ่งแวดล้อม การจัดการทรัพยากรธรรมชาติ และการวางแผนการใช้ประโยชน์ที่ดิน (Tamiminia et al., 2020) อีกทั้งงานในหลายด้านอย่างเช่นงานของ Sainarntip et al. (2021) ที่เป็นการศึกษาเกี่ยวกับข้อมูลจากดาวเทียมในช่วงคลื่นอินฟราเรดความร้อนมาประมาณค่าอุณหภูมิพื้นผิวก็สามารถนำเอา GEE มาประยุกต์ใช้งานได้เช่นกัน อย่างไรก็ตาม การเริ่มต้นใช้งาน GEE ต้องอาศัยความรู้ในการเขียนซอร์สโค้ดด้วยภาษา JavaScript ซึ่งนับเป็นอุปสรรคสำคัญสำหรับนักวิจัยและผู้เชี่ยวชาญในสาขาที่เกี่ยวข้องกับภูมิสารสนเทศ

และการรับรู้ระยะไกล (Remote Sensing) ที่มีพื้นฐานด้านการเขียนซอร์สโค้ดภาษา JavaScript ไม่แข็งแกร่งด้วยข้อจำกัดนี้ ผู้สนใจจำนวนหนึ่งที่มีแนวโน้มจะมีความเชี่ยวชาญทางด้านภูมิสารสนเทศและการรับรู้ระยะไกลไม่สามารถเข้าถึงศักยภาพอันยิ่งใหญ่ของ GEE ได้ เพื่อแก้ไขข้อจำกัดดังกล่าวในยุคที่การพัฒนา Generative AI มีความก้าวหน้าอย่างรวดเร็วและได้เข้ามามีบทบาทสำคัญในการสร้างซอร์สโค้ดจากภาษาธรรมชาติ (Natural Language to Code: NL2C) ซึ่งช่วยลดข้อจำกัดในการเข้าถึงระบบวิเคราะห์ข้อมูลภูมิสารสนเทศขนาดใหญ่อย่าง GEE (Jhonnerie et al., 2024; Osco et al., 2023; Tao and Xu, 2023) โดยเหตุนี้ การศึกษาถึงการประยุกต์ใช้วิศวกรรมข้อความ (Prompt Engineering) กับ Generative AI โดยการแปลงภาษาธรรมชาติให้เป็นการสร้างซอร์สโค้ดที่สามารถทำงานได้จริงบน GEE ในบริบทของการวิเคราะห์การใช้ประโยชน์ที่ดิน (Land Use Classification) จึงเป็นประเด็นที่น่าสนใจ

วัตถุประสงค์การวิจัย

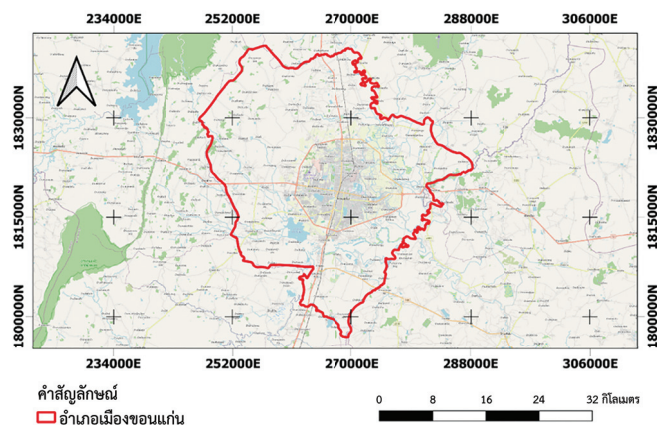
เพื่อประยุกต์ใช้ Prompt Engineering กับ Generative AI ในการแปลงภาษาธรรมชาติเป็นซอร์สโค้ดภาษา JavaScript สำหรับ GEE Code Editor ที่สามารถทำการวิเคราะห์และจำแนกประเภทการใช้ที่ดิน โดยไม่มีการปรับแก้ซอร์สโค้ดเพิ่มเติมภายหลัง และเลือกพื้นที่อำเภอเมืองขอนแก่นเป็นกรณีศึกษา

พื้นที่ในการศึกษา

พื้นที่ศึกษาอยู่ในเขตอำเภอเมืองขอนแก่น ซึ่งเป็นเขตเมืองหลักของจังหวัดและมีความหลากหลายด้านการใช้ประโยชน์ที่ดิน ทั้งพื้นที่ชุมชน พื้นที่แหล่งน้ำ พื้นที่ป่าไม้ พาดวิทยกรรม เกษตรกรรม และพื้นที่ว่างเปล่า โดยตั้งอยู่ในภูมิประเทศที่เป็นที่ราบสูงตอนกลางของภาคตะวันออกเฉียงเหนือและอยู่ห่างจากกรุงเทพมหานคร ประมาณ 445 กิโลเมตร มีพื้นที่ประมาณ 960 ตารางกิโลเมตร โดยมีขอบเขตพื้นที่ใช้ในการศึกษาดังรูปที่ 1

การแปลงคำสั่งจากภาษาธรรมชาติไปสู่ซอร์สโค้ด (NL2C)

การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) เป็นฐานคิดสำคัญของ LLMs ในการแปลงคำสั่งจากภาษาธรรมชาติไปสู่ซอร์สโค้ด (NL2C) โดยการประยุกต์ใช้ Prompt Engineering ที่เป็นการออกแบบชุดข้อความ (Prompt) อย่างมีระบบนั้นสามารถช่วยเพิ่มความแม่นยำของ NL2C ได้อย่างมีนัยสำคัญ กล่าวคือ การให้ตัวอย่างการทำงานหรือซอร์สโค้ดเพียง 2 - 5 ตัวอย่าง หรือที่เรียกว่า Few-Shot Prompting จะมีส่วนช่วยในการปรับปรุงคุณภาพของผลลัพธ์ให้มีความแม่นยำและตรงตามความต้องการ ทั้งนี้หลักการดังกล่าวมีรากฐานจากแนวคิด Meta-Learning ที่เน้นการให้โมเดลเรียนรู้จากตัวอย่างเพียงเล็กน้อย โดยอาศัยความรู้จากการฝึกก่อนหน้า (Pre-Training) ในกรณีของ LLMs ความสามารถนี้เกิดขึ้นผ่านกลไก In-Context Learning ซึ่งเทคนิคนี้สามารถปรับใช้ในการปรับแต่งโมเดลให้เหมาะสมกับข้อมูลเชิงภูมิสารสนเทศได้เป็นอย่างดี (Brown et al., 2020; Li and Ning, 2023) ทั้งนี้การออกแบบคำสั่งที่มีโครงสร้างและบริบทชัดเจนจะช่วยให้การสร้างซอร์สโค้ดจาก NL2C เป็นไปอย่างถูกต้อง (Garg et al., 2025) หลักการ NL2C สามารถนำมาใช้ในการสร้างซอร์สโค้ดสำหรับ GEE ดังเช่นงานของ Jhonnerie et al. (2024) และ Hou et al. (2025) อย่างไรก็ตาม งานวิจัยเหล่านี้เป็นการใช้ LLMs ที่พัฒนาขึ้นมาเฉพาะกิจสำหรับการวิเคราะห์ข้อมูลภูมิสารสนเทศเท่านั้นซึ่งยังไม่มีเผยแพร่อย่างกว้างขวาง ดังนั้นงานวิจัยนี้จึงมุ่งเน้นไปที่การใช้งาน LLMs ที่กำลังเป็นที่นิยมอย่าง Claude AI ทั้งนี้เพราะงานศึกษาวิจัยในอดีตของ Hochmair et al. (2024) ได้มีการศึกษาเปรียบเทียบ Generative AI พบว่า Claude AI มีขีดความสามารถสูงกว่า LLMs ตัวอื่น ๆ ในการสร้างซอร์สโค้ดภาษา Python สำหรับการวิเคราะห์ข้อมูลภูมิสารสนเทศ โดยเหตุนี้ การศึกษาวิจัยนี้จึงได้นำเอา Claude AI มาใช้เป็นเครื่องมือ LLMs ในการแปลงภาษาธรรมชาติให้เป็นซอร์สโค้ดภาษา JavaScript สำหรับ GEE



รูปที่ 1 พื้นที่ศึกษาวิจัยตั้งอยู่ในกรอบสีแดง ตั้งอยู่ ณ เขตอำเภอเมือง จังหวัดขอนแก่น

เครื่องมือและข้อมูลที่ใช้ในการวิเคราะห์

เครื่องมือหลักสำหรับในการศึกษานี้ประกอบด้วย 2 ส่วนหลักคือ (1) แบบจำลอง LLMs ซึ่งในที่นี้ใช้ Claude AI รุ่น Sonnet 4.0 สำหรับใช้ในการแปลงภาษาธรรมชาติให้เป็นซอร์สโค้ดภาษา JavaScript สำหรับในงาน (2) แพลตฟอร์ม GEE Code Editor ซึ่งเป็นแพลตฟอร์มระบบภูมิสารสนเทศขนาดใหญ่ที่มีขีดความสามารถในการวิเคราะห์สภาพการใช้ที่ดินและสิ่งปกคลุมดิน (Land Use/Cover) โดยใช้ข้อมูลภาพถ่ายดาวเทียม Sentinel-2 ในพื้นที่เขตอำเภอเมือง จังหวัดขอนแก่น มาทำการวิเคราะห์ด้วยวิธีการจำแนกประเภทการใช้ที่ดินแบบควบคุม (Supervised Classification) โดยการแปลตีความจากข้อมูลภาพถ่ายดาวเทียมความละเอียดสูง (High-Resolution Satellite Image) ของปีเดียวกัน ที่มีใน Google Earth เพื่อกำหนดจุดตัวอย่าง (Sample Data) ในแต่ละคลาสรายละเอียดดังตารางที่ 1 สำหรับใช้งานใน GEE เพื่อใช้เป็นข้อมูลฝึกสอน (Training) และตรวจสอบผล (Validation)

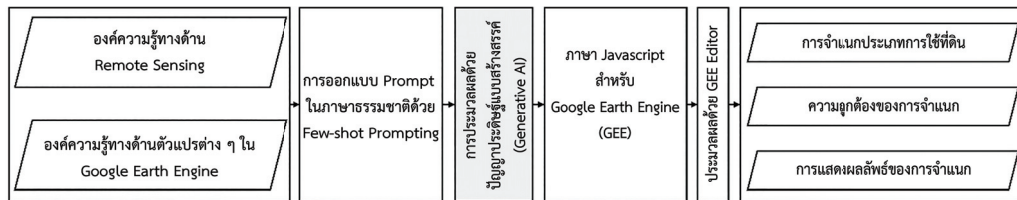
ตารางที่ 1 รายละเอียดข้อมูลและเครื่องมือที่ใช้ในการวิเคราะห์

ประเภทข้อมูล	รายละเอียด
ภาพถ่ายดาวเทียม	Sentinel-2
ช่วงเวลา	มกราคม - เมษายน 2567
ขอบเขตพื้นที่ศึกษา	อำเภอเมือง จังหวัดขอนแก่น
เครื่องมือวิเคราะห์	Claude AI (Sonnet 4.0), Google Earth Engine Code Editor

วิธีการดำเนินงานวิจัย

การศึกษานี้มีกรอบแนวคิดในการทำงานที่เริ่มต้นจากการสกัดองค์ความรู้ทางด้านการสำรวจระยะไกล (Remote Sensing) และความรู้เกี่ยวกับตัวแปรต่าง ๆ ของ GEE ที่ต้องใช้ในการวิเคราะห์ข้อมูลภาพถ่ายดาวเทียม Sentinel-2 บน GEE แล้วนำมาเป็นฐานคิดในการออกแบบภาษาธรรมชาติในรูปแบบของ Prompt Engineering บนกรอบการทำงานแบบ Few-Shot Prompting ที่มีการแสดงตัวอย่างซอร์สโค้ด JavaScript สำหรับ GEE ที่เกี่ยวข้องกับลักษณะประโยชน์ ๆ เท่าที่จำเป็น เพื่อสอนให้ Generative AI เข้าใจถึงระบบการทำงานของ GEE เช่น การจัดเก็บข้อมูลใน GEE รูปแบบของชั้นข้อมูล และรูปแบบการเขียนโค้ดต่าง ๆ ของผลลัพธ์ ทั้งนี้เพื่อสร้างซอร์สโค้ดที่ตรงกับการทำงานของ GEE รวมถึงเพิ่มความสามารถของ Generative AI ในการแปลงคำสั่งจากภาษาธรรมชาติที่มีโครงสร้างเป็นซอร์สโค้ดสำหรับการวิเคราะห์เชิงพื้นที่ได้อย่างครบถ้วน โดยมีชุดข้อความ (Prompt) ที่ใช้ประกอบด้วย ชุดข้อความที่ 1: การเตรียมข้อมูล (Data Preparation) และชุดข้อความที่ 2: การประมวลผลข้อมูลและการประเมินความถูกต้อง (Data Processing

and Accuracy Assessment) ซึ่งมีลำดับการทำงานดังรูปที่ 2 โดยหลังจากได้ซอร์สโค้ดจากการแปลงภาษาธรรมชาติแล้วจะนำซอร์สโค้ดที่ได้ไปใช้งานใน GEE เพื่อตรวจสอบศักยภาพของ NL2C ที่มีโครงสร้างเป็นซอร์สโค้ดว่าได้ผลลัพธ์ตามที่ต้องการ อย่างการจำแนกที่ดินด้วยโดยใช้ตัวจำแนกแบบ Support Vector Machine (SVM) พร้อมด้วยการประเมินผลลัพธ์ด้วย Accuracy และ Kappa และการแสดงผลในส่วนต่อประสานกับผู้ใช้ (User Interface)



รูปที่ 2 กรอบแนวคิดในการทำงานวิจัย

ผลการศึกษาและอภิปรายผล

เนื่องจากการศึกษานี้มีวัตถุประสงค์เพื่อประยุกต์ใช้ Prompt Engineering กับ Generative AI ในการแปลงภาษาธรรมชาติเป็นซอร์สโค้ดภาษา JavaScript สำหรับ GEE Code Editor โดยไม่มีการปรับแก้ซอร์สโค้ดเพิ่มเติมภายหลัง อันจะแสดงให้เห็นถึงศักยภาพของ Generative AI ในการสร้างซอร์สโค้ดวิเคราะห์เชิงพื้นที่บนแพลตฟอร์ม GEE ผ่านการสื่อสารด้วยภาษาธรรมชาติแบบมีโครงสร้างในการสร้างชุดข้อความที่เป็นภาษาธรรมชาติที่เหมาะสม แล้วนำเข้าสู่ Generative AI ในการสร้างคำสั่งเชิงโปรแกรมที่ถูกต้อง ครบคลุม และสามารถรันได้จริงในสภาพแวดล้อมจริงบน GEE โดยแบ่งผลลัพธ์ที่ได้จากการประมวลผลภาษาธรรมชาติเป็นซอร์สโค้ดภาษา JavaScript สำหรับ GEE Code Editor ออกเป็น 2 ส่วนหลัก คือ ผลลัพธ์ส่วนที่ 1 ซอร์สโค้ดการเตรียมข้อมูล และผลลัพธ์ส่วนที่ 2 ซอร์สโค้ดการประมวลผลข้อมูลและการประเมินความถูกต้อง ซึ่งได้จากชุดข้อความที่ 1 และ 2 มีรายละเอียดดังนี้

ผลลัพธ์ส่วนที่ 1 ซอร์สโค้ดการเตรียมข้อมูล (Data Preparation)

ชุดข้อความและซอร์สโค้ดในส่วนนี้จัดเป็นพื้นฐานที่จำเป็นต่อการสร้างชุดข้อความในรูปแบบของภาษาธรรมชาติในการเตรียมข้อมูลพื้นฐานที่จำเป็นต่อการวิเคราะห์ข้อมูล ซึ่งจัดเป็นขั้นตอนสำคัญในการนำมาวิเคราะห์การใช้ประโยชน์ที่ดินอำเภอขอนแก่นด้วย GEE ผ่านการสร้างชุดข้อความตามหลักการ Prompt Engineering ด้วยเทคนิค Few-Shot Prompting ผ่านการให้ตัวอย่างที่ชัดเจนและเฉพาะเจาะจง โครงสร้างของชุดข้อความดังตารางที่ 2 สามารถแยกการอธิบายออกเป็น 3 ส่วนสำคัญ ดังนี้

ส่วนที่ 1 การกำหนดพื้นที่ศึกษา ชุดข้อความและซอร์สโค้ดในส่วนนี้ มีข้อมูลพื้นฐานที่จำเป็นต่อการวิเคราะห์คือ ข้อมูลขอบเขตการปกครอง (Administrative Boundary Dataset) ซึ่งชุดข้อมูลที่ได้จากองค์กร FAO เพื่อแสดงขอบเขตการปกครอง ทั้งในระดับจังหวัดและอำเภอ

ส่วนที่ 2 การเตรียมข้อมูลภาพถ่ายดาวเทียม ชุดข้อความและซอร์สโค้ดที่กำหนดให้คัดเอาชุดข้อมูล (Dataset) ของข้อมูลภาพถ่ายดาวเทียม Sentinel-2 ที่กวาดข้อมูลระหว่าง 1 มกราคม - 30 เมษายน พ.ศ. 2567 กำหนดให้มีเมฆปกคลุมไม่เกิน 20 % และมีการสร้างภาพแบบ Composite ด้วยวิธี Median ซึ่งเป็นการเลือกค่าของจุดภาพของแต่ละจุดภาพด้วยการ Median ค่าจากชุดภาพหลาย ๆ ภาพ ณ ตำแหน่งเดียวกัน ซึ่งมีจุดเด่นในการกำจัดเมฆและเงาเมฆได้ออกจากข้อมูล

ส่วนที่ 3 การคำนวณดัชนีต่าง ๆ ชุดข้อความและซอร์สโค้ดนี้เป็นการคำนวณดัชนีสำหรับใช้ประกอบการวิเคราะห์การใช้ประโยชน์ที่ดิน โดยในการศึกษานี้ได้เลือกใช้ค่าดัชนีต่าง ๆ อันประกอบด้วย ค่าดัชนี Normalized Difference Vegetation Index (NDVI) คือ ดัชนีวัดความหนาแน่นและสุขภาพของพืชพรรณค่าดัชนี ค่าดัชนี Modified Normalized Difference Water Index (MNDWI) คือ ดัชนีตรวจจับและติดตามการเปลี่ยนแปลงของแหล่งน้ำ ค่าดัชนี Normalized Difference Built-up Index (NDBI) คือ ดัชนีวิเคราะห์การขยายตัวของเขตเมือง Urban Index (UI) คือ ดัชนีประเมินระดับการพัฒนาเมือง และค่าดัชนี Bare Soil Index (BSI) คือ ดัชนีระบุพื้นที่ดินเปลือยเปล่า (Bare Land)

ตารางที่ 2 ชุดข้อความภาษาธรรมชาติสำหรับสร้างพอร์สโค้ดสำหรับขั้นตอนการเตรียมข้อมูล

ชุดข้อความภาษาธรรมชาติ	พอร์สโค้ดภาษา JavaScript สำหรับ GEE ที่สร้างขึ้นจาก Prompt Engineering และ Generative AI
ส่วนที่ 1 การกำหนดพื้นที่ศึกษา	
พรมท 1: การตั้งค่าโครงการและการเตรียมข้อมูลสำหรับการวิเคราะห์การใช้ประโยชน์ที่ดิน โจทย์: การวิเคราะห์การใช้ประโยชน์ที่ดินอำเภอขอนแก่นด้วย Google Earth Engine 1. การกำหนดพื้นที่ศึกษา (Study Area) - ชุดข้อมูลที่ใช้: FAO/GAUL/2015/level2 (Global Administrative Unit Layers) - การเลือกพื้นที่: อำเภอเมืองขอนแก่น (ระดับ ADM2, ADM2_NAME = "Muang Khon Kaen") ต้องใช้ฟิลเตอร์แบบ AND เพื่อกรองทั้ง ADM1_NAME และ ADM2_NAME - การตรวจสอบข้อมูล: แสดงข้อมูลของทั้งอำเภอในคอนโซลด้วย print(), แสดงพื้นที่ (ตร.กม.) ของทั้งอำเภอในคอนโซล, จดวางตำแหน่งแผนที่ให้แสดงอำเภอขอนแก่นด้วย Map.centerObject()	<pre>// การวิเคราะห์การใช้ประโยชน์ที่ดินอำเภอเมืองขอนแก่นด้วย Google Earth Engine // 1. การกำหนดพื้นที่ศึกษา (Study Area) // โหลดชุดข้อมูลขอบเขตการปกครองจาก FAO/GAUL var level2 = ee.FeatureCollection("FAO/GAUL/2015/level2"); // เลือกอำเภอเมืองขอนแก่น (ระดับ ADM2) โดยใช้ฟิลเตอร์แบบ AND var muangKhonkaen = level2.filter(ee.Filter.and(ee.Filter.eq('ADM1_NAME', 'Khon Kaen'), ee.Filter.eq('ADM2_NAME', 'Muang Khon Kaen'))); // แสดงข้อมูลในคอนโซล print('ข้อมูลอำเภอเมืองขอนแก่น:', muangKhonkaen); // คำนวณและแสดงพื้นที่ (ตารางกิโลเมตร) var districtArea = muangKhonkaen.geometry().area().divide(1000000); print('พื้นที่อำเภอเมืองขอนแก่น (ตร.กม.):', districtArea); // จดวางตำแหน่งแผนที่ Map.centerObject(muangKhonkaen, 11);</pre>
ส่วนที่ 2 การเตรียมข้อมูล Sentinel-2	
2. การเตรียมข้อมูลภาพถ่ายดาวเทียม Sentinel-2 - ชุดข้อมูลที่ใช้: COPERNICUS/S2_SR_HARMONIZED - การกรองข้อมูล: ช่วงเวลา: 1 มกราคม - 30 เมษายน 2567 (ใช้ filterDate('2024-01-01', '2024-04-30')), พื้นที่: อำเภอขอนแก่น (ใช้ filterBounds()), เมฆปกคลุม: น้อยกว่าหรือเท่ากับ 20% (ใช้ filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))) - การประมวลผลข้อมูล: สร้างฟังก์ชัน scaleS2 สำหรับปรับค่าการสะท้อนแสงโดยหารด้วย 10,000, รักษาคุณสมบัติ 'system:time_start' ด้วย copyProperties, สร้างภาพรวม (composite) ด้วยวิธี median() - การตรวจสอบข้อมูล: แสดงจำนวนภาพใน collection (ใช้ size()) แสดงข้อมูลเมตาเดต้าและแบนด์ของภาพแรก (ใช้ first())	<pre>// 2. การเตรียมข้อมูลภาพถ่ายดาวเทียม Sentinel-2 // ฟังก์ชันสำหรับปรับค่าการสะท้อนแสง Sentinel-2 function scaleS2(image) { return image.multiply(0.0001).copyProperties(image, ['system:time_start']); } // หารด้วย 10,000 เหมือนกับการคูณด้วย 0.0001 // โหลดและกรองข้อมูล Sentinel-2 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED') .filterDate('2024-01-01', '2024-04-30').filterBounds(muangKhonkaen).filter (ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20)).map(scaleS2); // เมฆปกคลุมน้อยกว่า 20% // ช่วงเวลา 1 ม.ค. - 30 เม.ย. 2024 // พื้นที่อำเภอเมืองขอนแก่น // ปรับค่าการสะท้อนแสง // สร้างภาพรวมด้วยวิธี median var s2Composite = sentinel2.median(); // แสดงข้อมูลในคอนโซล print('จำนวนภาพ Sentinel-2:', sentinel2.size()); print('ข้อมูลเมตาเดต้าภาพแรก:', sentinel2.first()); print('ข้อมูลแบนด์ของภาพ composite:', s2Composite.bandNames());</pre>

ตารางที่ 2 ชุดข้อความภาษาธรรมชาติสำหรับสร้างซอร์สโค้ดสำหรับขั้นตอนการเตรียมข้อมูล (ต่อ)

ส่วนที่ 3 การคำนวณดัชนี	
3. การคำนวณดัชนีพืชพรรณและดัชนีพื้นที่เมือง - ดัชนีที่ต้องคำนวณ: NDVI (ดัชนีพืชพรรณ): $(B8 - B4)/(B8 + B4)$ ใช้ฟังก์ชัน <code>normalizedDifference(['B8', 'B4'])</code> ตั้งชื่อเป็น 'NDVI', MNDWI (ดัชนีน้ำ): $(B3 - B11)/(B3 + B11)$ ใช้ฟังก์ชัน <code>normalizedDifference(['B3', 'B11'])</code> ตั้งชื่อเป็น 'MNDWI', NDBI (ดัชนีพื้นที่สิ่งปลูกสร้าง): $(B11 - B8)/(B11 + B8)$ ใช้ฟังก์ชัน <code>normalizedDifference(['B11', 'B8'])</code> ตั้งชื่อเป็น 'NDBI', BSI (ดัชนีพื้นที่ว่างเปล่า): $((B11+B4)-(B8+B2))/((B11+B4)+(B8+B2))$ ใช้ฟังก์ชัน <code>expression()</code> กับสูตรที่กำหนด ตั้งชื่อเป็น 'BSI', Urban Index (UI): $(B11 - B8)/(B11 + B8)$ เหมือนกับ NDBI แต่เปลี่ยนชื่อเป็น 'UI' โปรดเขียนโค้ด JavaScript สำหรับ Google Earth Engine ตามขั้นตอนข้างต้น โดยจัดโครงสร้างให้เป็นระเบียบและมีคำอธิบายประกอบในแต่ละส่วน	<pre>// 3. การคำนวณดัชนีพืชพรรณและดัชนีพื้นที่เมือง // คำนวณ NDVI (ดัชนีพืชพรรณ) var ndvi = s2Composite.normalizedDifference(['B8', 'B4']).rename('NDVI'); // คำนวณ MNDWI (ดัชนีน้ำ) var mndwi = s2Composite.normalizedDifference(['B3', 'B11']).rename('MNDWI'); // คำนวณ NDBI (ดัชนีพื้นที่สิ่งปลูกสร้าง) var ndbi = s2Composite.normalizedDifference(['B11', 'B8']).rename('NDBI'); // คำนวณ BSI (ดัชนีพื้นที่ว่างเปล่า) var bsi = s2Composite.expression('((B11 + B4) - (B8 + B2)) / ((B11 + B4) + (B8 + B2))', { 'B11':s2Composite.select('B11'), 'B8': s2Composite.select('B8'), 'B4': s2Composite.select('B4'), 'B2':s2Composite.select('B2') }).rename('BSI'); // คำนวณ Urban Index (UI) var UI = s2Composite.normalizedDifference(['B11', 'B8']).rename('UI'); // รวมดัชนีทั้งหมดเข้ากับภาพหลัก var indices = s2Composite.addBands([ndvi, mndwi, ndbi, bsi, UI]); print('ข้อมูลดัชนีทั้งหมด:', indices.bandNames());</pre>

ซอร์สโค้ดการประมวลผลข้อมูลและการประเมินความถูกต้อง (Data Processing and Accuracy Assessment)

ชุดข้อความและซอร์สโค้ดที่ได้ในส่วนนี้ใช้สำหรับประมวลผลข้อมูลภาพดาวเทียม Sentinel-2 ควบคู่กับการวิเคราะห์ดัชนีต่าง ๆ และตามด้วยการประเมินความถูกต้อง โดยแบ่งโครงสร้างของชุดข้อความเป็น 9 ส่วนสำคัญ ดังตารางที่ 3 ซึ่งมีรายละเอียดดังนี้

ส่วนที่ 1 การนำเข้าและตรวจสอบข้อมูลฝึกสอน ชุดข้อความและซอร์สโค้ดสำหรับการนำเข้าข้อมูลฝึกสอนจากแหล่งข้อมูลส่วนตัวที่ระบุตำแหน่งเก็บข้อมูลบน GEE Platform สำหรับการใช้งาน GEE กับข้อมูลฝึกสอนสำหรับการกำหนดการจำแนกประเภทการใช้ประโยชน์ที่ดินแต่ละประเภท ซึ่งในการศึกษาวิจัยนี้ได้แบ่งออกเป็น 5 ประเภทหลักได้แก่

- ก. พื้นที่แหล่งน้ำ (Water: WA) คือ บริเวณที่เป็นน้ำ เช่น แม่น้ำ อ่างเก็บน้ำ หนอง บึง หรือน้ำท่วมขัง
- ข. พื้นที่พืชพรรณ (Forest: FA) คือ บริเวณที่มีการกระจายของต้นไม้ เช่น ป่าดิบ ป่าโปร่ง และป่าละเมาะ
- ค. พื้นที่การเกษตร (Agriculture: AG) คือ บริเวณที่เพาะปลูกการเกษตร เช่น ข้าว อ้อย และมันสำปะหลัง
- ง. พื้นที่ว่างเปล่า (Barren: BA) คือ บริเวณพื้นที่โล่งหรือพื้นที่ว่างเปล่าที่อาจมีหญ้าปกคลุมบางส่วน
- จ. พื้นที่แหล่งชุมชน (Urban: UR) คือ บริเวณที่มนุษย์สร้างขึ้น เช่น ที่พักอาศัย อาคารและสิ่งปลูกสร้าง

ส่วนที่ 2 การกำหนดแบนด์สำหรับการจำแนกประเภทการใช้ประโยชน์ที่ดิน ชุดข้อความและซอร์สโค้ดสำหรับการเลือกแบนด์ที่เหมาะสมสำหรับการจำแนกประเภทการใช้ประโยชน์ที่ดิน โดยผสมผสานระหว่างแบนด์พื้นฐานของ Sentinel-2 และดัชนีเชิงคลื่นที่คำนวณไว้ในพรอมต์แรก การเลือกแบนด์ครอบคลุมช่วงคลื่นที่สำคัญทั้งในช่วงแสงที่ตามองเห็นและอินฟราเรดใกล้ ประกอบด้วย B2 B3 B4 B8 B11 และ B12 และการรวมดัชนีที่ใช้ในการวิเคราะห์ (NDVI MNDWI NDBI BSI และ UI) เข้าด้วยกัน เพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทการใช้ที่ดิน

ส่วนที่ 3 การสร้างข้อมูลฝึกสอน ชุดข้อความและซอร์สโค้ดสำหรับการสร้างชุดข้อมูลฝึกสอนที่มีประสิทธิภาพสำหรับการเรียนรู้ของเครื่อง (Machine Learning)

ส่วนที่ 4 การแบ่งข้อมูล Train-Test Split ชุดข้อความและซอร์สโค้ดสำหรับแบ่งชุดข้อมูลฝึกสอนออกเป็น 2 ส่วนคือ ส่วนสำหรับฝึกสอน (Training Dataset) และส่วนสำหรับการทดสอบเพื่อยืนยันผล (Testing Dataset) โดยจุดตัวอย่างฝึกสอนสำหรับการใช้ที่ดินทั้งหมด 5 ประเภท (UR AG FA BA WA) อย่างละ 53 จุด รวม 265 จุด และจุดตรวจสอบอย่างละ 53 จุด รวม 265 จุด

ส่วนที่ 5 การฝึกสอนโมเดล ชุดข้อความและซอร์สโค้ดสำหรับการเลือกและกำหนดค่าพารามิเตอร์ของตัวจำแนก Support Vector Machine (SVM) โดยในที่นี้ได้เลือกใช้ Kernel แบบ RBF (Radial Basis Function) สำหรับการจำแนกประเภทการใช้ประโยชน์ที่ดินเนื่องจากสามารถจัดการกับข้อมูลที่มีความซับซ้อนและไม่เป็นเชิงเส้น

ส่วนที่ 6 การจำแนกพื้นที่ ชุดข้อความและซอร์สโค้ดสำหรับการนำโมเดลที่ฝึกสอนแล้วมาใช้งานจริงในขั้นตอนการจำแนกประเภทการใช้ประโยชน์ที่ดินออกเป็น 5 ประเภทหลักดังที่กล่าวไว้ข้างต้น

ส่วนที่ 7 การประเมินประสิทธิภาพ ชุดข้อความและซอร์สโค้ดสำหรับการทดสอบที่แยกไว้แล้วในส่วนที่ 4 ซึ่งจะทำการวิเคราะห์ Error Matrix, Overall Accuracy และ Kappa Coefficient Confusion Matrix

ส่วนที่ 8 การแสดงบนแผนที่ ชุดข้อความและซอร์สโค้ดสำหรับการแสดงผลข้อมูลการจำแนกประเภทอย่างมีประสิทธิภาพและเหมาะสม โดยการกำหนดสีสำหรับการแสดงผลที่สื่อความหมายชัดเจนตามธรรมชาติของแต่ละประเภทการใช้ประโยชน์ที่ดิน

ส่วนที่ 9 คำอธิบายสัญลักษณ์ ชุดข้อความและซอร์สโค้ดสำหรับการออกแบบคำสัญลักษณ์ (Legend) ที่มีประสิทธิภาพสำหรับการแสดงผลข้อมูลใน GEE

ตารางที่ 3 ชุดข้อความภาษาธรรมชาติสำหรับสร้างซอร์สโค้ดเพื่อการจำแนกการใช้ที่ดินและการตรวจสอบ

ชุดข้อความภาษาธรรมชาติ	ซอร์สโค้ดภาษา JavaScript สำหรับ GEE ที่สร้างขึ้นจาก Prompt Engineering และ Generative AI
ส่วนที่ 1 การนำเข้าและตรวจสอบข้อมูลฝึกสอน	
พรมท 2: การฝึกสอนโมเดลจำแนกประเภทการใช้ประโยชน์ที่ดิน โจทย์: การสร้างและฝึกสอนโมเดลจำแนกประเภทการใช้ประโยชน์ที่ดินอำเภอขอนแก่น สร้างโค้ด Google Earth Engine JavaScript ที่ช่วยฝึกสอนโมเดลจำแนกประเภทการใช้ประโยชน์ที่ดิน โดยเริ่มบทใหม่ต่อจากพรมท 1 โดยมีขั้นตอนดังนี้ 1. การนำเข้าชุดข้อมูลฝึกสอน (Training Data) - ข้อมูลที่ต้องนำเข้า: ข้อมูลจุดตัวอย่างของอำเภอเมืองขอนแก่น: 'projects/ee-gramsm32/assets/PointMuangKK24' -การตรวจสอบข้อมูล: แสดงจำนวนฟีเจอร์ในแต่ละชุดข้อมูลด้วยฟังก์ชัน size() เก็บตัวแปรเป็น districtPoints - ข้อมูลคลาสในชุดข้อมูล: คลาส 0: พื้นที่แหล่งน้ำ (Water), คลาส 1: พื้นที่ป่าไม้ (Forest), คลาส 2: พื้นที่เกษตรกรรม (Agriculture), คลาส 3: พื้นที่ว่างเปล่า (Barren), คลาส 4: พื้นที่เมือง (Urban)	<pre>// การฝึกสอนโมเดลจำแนกประเภทการใช้ประโยชน์ที่ดินอำเภอเมืองขอนแก่น // หมายเหตุ: ต้องรันโค้ดจากพรมท 1 ก่อนเพื่อให้มีตัวแปรพื้นฐาน // 1. การนำเข้าชุดข้อมูลฝึกสอน (Training Data) // นำเข้าข้อมูลจุดตัวอย่าง var districtPoints = ee.FeatureCollection('projects/ee-gramsm32/assets/PointMuangKK24'); // ตรวจสอบจำนวนฟีเจอร์ในแต่ละชุดข้อมูล print('จำนวนจุดตัวอย่างในอำเภอเมืองขอนแก่น:', districtPoints.size()); // แสดงข้อมูลตัวอย่างของฟีเจอร์แรกในแต่ละชุดข้อมูล print('ตัวอย่างข้อมูลจุดตัวอย่างอำเภอ:', districtPoints.first()); // ข้อมูลคลาสที่มีในชุดข้อมูล: // คลาส 0: Water // คลาส 1: Forest // คลาส 2: Agriculture // คลาส 3: Barren // คลาส 4: Urban</pre>

ตารางที่ 3 ชุดข้อความภาษาธรรมชาติสำหรับสร้างซอร์สโค้ดเพื่อการจำแนกการใช้ที่ดินและการตรวจสอบ (ต่อ)

ส่วนที่ 2 การกำหนดแบนด์สำหรับการจำแนก	
2. การกำหนดแบนด์สำหรับการจำแนกประเภท (Band Selection) - แบนด์ที่ต้องใช้:แบนด์พื้นฐาน: 'B2', 'B3', 'B4', 'B8', 'B11', 'B12', ดัชนีที่คำนวณไว้ในพรมแดน 1: 'NDVI', 'MNDWI', 'NDBI', 'BSI', 'UrbanIndex' - การกำหนดตัวแปร:สร้างตัวแปร bands เป็น array เพื่อเก็บรายชื่อแบนด์ที่ต้องการใช้ทั้งหมด, สร้างภาพรวม classificationImage	<pre>// 2. การกำหนดแบนด์สำหรับการจำแนกประเภท (Band Selection) // กำหนดแบนด์ที่ต้องการใช้ var bands = ['B2', 'B3', 'B4', 'B8', 'B11', 'B12', 'NDVI', 'MNDWI', 'NDBI', 'BSI', 'UI']; // สร้างภาพรวมที่ประกอบด้วยแบนด์ทั้งหมดที่ต้องการใช้ (จากพรมแดน 1) var classificationImage = indices.select(bands); print('แบนด์ที่ใช้สำหรับการจำแนก:', classificationImage.bandNames());</pre>
ส่วนที่ 3 การสร้างข้อมูลฝึกสอน	
3. การสร้างชุดข้อมูลฝึกสอน (Training Sample Generation) - การสุ่มตัวอย่างจากภาพดาวเทียม: ใช้ฟังก์ชัน sampleRegions() เพื่อสุ่มตัวอย่างข้อมูลจากภาพในตำแหน่งของจุดตัวอย่าง กำหนดพารามิเตอร์ ดังนี้: collection: ชุดข้อมูลจุดตัวอย่าง , properties: ['class'] (คอลัมน์ที่เก็บค่าคลาส), scale: 30 (ใช้ scale ที่ใหญ่พอเพื่อลดการใช้หน่วยความจำ), tileSize: 4 (เพิ่มประสิทธิภาพการประมวลผลเมื่อมีข้อมูลจำนวนมาก), geometries: false , เก็บผลลัพธ์ในตัวแปร sampledDistrict	<pre>// 3. การสร้างชุดข้อมูลฝึกสอน (Training Sample Generation) // สุ่มตัวอย่างข้อมูลจากภาพในระดับอำเภอ var sampledDistrict = classificationImage.sampleRegions({ collection: districtPoints, properties: ['class'], scale: 30, tileSize: 4, geometries: false}); print('จำนวนตัวอย่างที่สุ่มได้ (อำเภอ):', sampledDistrict.size());</pre>
ส่วนที่ 4 การแบ่งข้อมูล Train-Test Split	
4. การแบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ (Train-Test Split) - แบ่งข้อมูลในระดับอำเภอ:สร้างคอลัมน์สุ่มด้วยฟังก์ชัน randomColumn(): seed 42, เรียงลำดับข้อมูลตามค่าสุ่มด้วย sort('random'), แปลงเป็น List และรับจำนวนข้อมูลด้วย toList() และ size().getInfo(), คำนวณจำนวนครึ่งหนึ่งของข้อมูลด้วย Math.floor(count / 2), แบ่งข้อมูลออกเป็นสองส่วนเท่า ๆ กันด้วย slice() แปลงกลับเป็น FeatureCollection ด้วย ee.FeatureCollection(), เก็บในตัวแปร trainingDistrict และ testingDistrict - ตรวจสอบผลลัพธ์: แสดงจำนวนข้อมูลในแต่ละชุดด้วย print() และ size()	<pre>// 4. การแบ่งข้อมูลเป็นชุดฝึกสอนและชุดทดสอบ (Train-Test Split) // แบ่งข้อมูลในระดับอำเภอ var sampledDistrictWithRandom = sampledDistrict.randomColumn('random', 42); var sortedDistrict = sampledDistrictWithRandom.sort('random'); var districtList = sortedDistrict.toList(sortedDistrict.size()); var districtCount = districtList.size().getInfo(); var halfDistrictCount = Math.floor(districtCount / 2); var trainingDistrict = ee.FeatureCollection(districtList.slice(0, halfDistrictCount)); var testingDistrict = ee.FeatureCollection(districtList.slice(halfDistrictCount)); // ตรวจสอบผลลัพธ์การแบ่งข้อมูล print('การแบ่งข้อมูลฝึกสอนและทดสอบ'); print('ข้อมูลฝึกสอนอำเภอ:', trainingDistrict.size()); print('ข้อมูลทดสอบอำเภอ:', testingDistrict.size());</pre>

ตารางที่ 3 ชุดข้อความภาษาธรรมชาติสำหรับสร้างซอร์สโค้ดเพื่อการจำแนกการใช้ที่ดินและการตรวจสอบ (ต่อ)

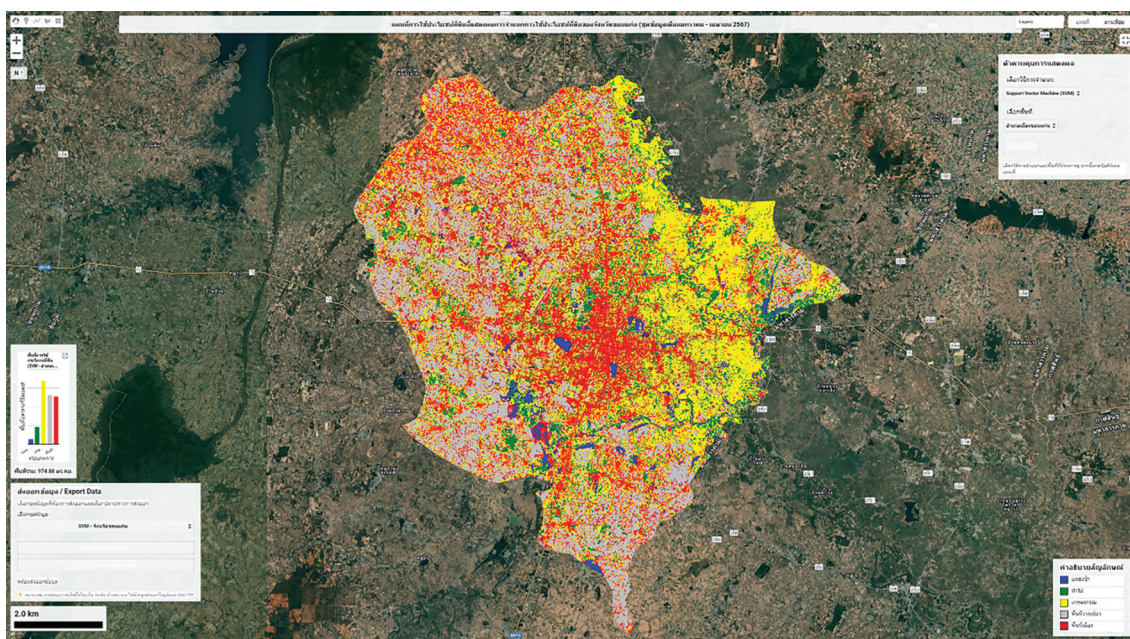
ส่วนที่ 5 การฝึกสอนโมเดล	
5. การฝึกสอนโมเดล (Model Training) - โมเดล SVM แบบ RBF: ใช้ ee.Classifier.libsvm() พร้อมกำหนด พารามิเตอร์: kernelType: 'RBF', gamma: 0.5, cost: 10 เก็บใน svmClassifierDistrict	// 5. การฝึกสอนโมเดล (Model Training) // โมเดล SVM แบบ RBF สำหรับอำเภอ var svmClassifierDistrict = ee.Classifier.libsvm({ kernelType: 'RBF', gamma: 0.5, cost: 10 }).train({ features: trainingDistrict, classProperty: 'class', inputProperties: bands}); print('การฝึกสอนโมเดลเสร็จสิ้น'); print('- SVM RBF (อำเภอ)');
ส่วนที่ 6 การจำแนกพื้นที่	
6. การจำแนกพื้นที่ (Area Classification) - จำแนกพื้นที่แต่ละโมเดล: ทำการจำแนกโดยใช้ classify() กับ classificationImage ,ใช้ clip() เพื่อตัดภาพตามขอบเขตพื้นที่ (อำเภอ), เก็บ ผลลัพธ์ในตัวแปรระดับพื้นที่ เช่น: svmClassificationDistrict เป็นต้น	// 6. การจำแนกพื้นที่ (Area Classification) // การจำแนกพื้นที่ ระดับอำเภอ var svmClassificationDistrict = classificationImage.classify(svmClassifierDistrict) .clip(muangKhonkaen); print('การจำแนกพื้นที่เสร็จสิ้น');
ส่วนที่ 7 การประเมินประสิทธิภาพ	
7. การประเมินประสิทธิภาพของโมเดล (Model Evaluation) - คำนวณค่าสถิติสำคัญ: ใช้ชุดข้อมูลทดสอบที่ แยกไว้, จำแนกชุดข้อมูลทดสอบด้วยโมเดลที่ ฝึกสอนแล้ว, สร้าง Confusion Matrix ด้วย errorMatrix(), คำนวณ Overall Accuracy ด้วย accuracy(), คำนวณ Kappa Coefficient ด้วย kappa() - แสดงผลการประเมิน: แสดงผลลัพธ์ทั้งหมดใน คอนโซลด้วย print(), แยกการประเมินสำหรับ แต่ละโมเดลและแต่ละระดับพื้นที่	// 7. การประเมินประสิทธิภาพของโมเดล (Model Evaluation) // ฟังก์ชันประเมินโมเดล function evaluateModel(classifier, testData, modelName, region) { var validated = testData.classify(classifier); var matrix = validated.errorMatrix('class', 'classification'); var accuracy = matrix.accuracy(); var kappa = matrix.kappa(); print('การ ประเมิน ' + modelName + ' (' + region + ')'); print('Overall Accuracy:', accuracy); print('Kappa Coefficient:', kappa); print('Confusion Matrix:', matrix); return { accuracy: accuracy, kappa: kappa, matrix: matrix }; // ประเมินโมเดล ทั้งหมด var svmDistrictEval = evaluateModel(svmClassifierDistrict, testingDistrict, 'SVM RBF', 'อำเภอ');
ส่วนที่ 8 การแสดงผลบนแผนที่	
8. การแสดงผลบนแผนที่ (Map Visualization) - กำหนดพารามิเตอร์การแสดงผล: กำหนด min: 0 และ max: 4 ตามจำนวนคลาส, กำหนด พาเลตสีดังนี้: คลาส 0 (Water): '#0000FF' (สีน้ำ เงิน), คลาส 1 (Forest): '#008000' (สีเขียว), คลาส 2 (Agriculture): '#FFFF00' (สีเหลือง), คลาส 3 (Barren): '#C0C0C0' (สีเทา), คลาส 4 (Urban): '#FF0000' (สีแดง) - แสดงผลการจำแนก: ใช้ Map.addLayer() เพื่อ แสดงผลลัพธ์ของแต่ละโมเดล, ตั้งค่าให้ซ่อนเริ่มต้น ด้วย visible: true	// 8. การแสดงผลบนแผนที่ (Map Visualization) // กำหนด พารามิเตอร์การแสดงผล var classificationVis = { min: 0, max: 4, palette: ['#0000FF', '#008000', '#FFFF00', '#C0C0C0', '#FF0000'] }; // แสดงผลการจำแนกระดับอำเภอ Map.addLayer(svmClassificationDistrict, classificationVis, 'SVM Classification (อำเภอ)', true);

ตารางที่ 3 ชุดข้อความภาษาธรรมชาติสำหรับสร้างซอร์สโค้ดเพื่อการจำแนกการใช้ที่ดินและการตรวจสอบ (ต่อ)

ส่วนที่ 9 คำอธิบายสัญลักษณ์	
9. สร้างคำอธิบายสัญลักษณ์ (Legend): สร้างด้วยui.Panel และกำหนดตำแหน่งที่มุมขวาล่าง, เพิ่มหัวข้อและรายการสำหรับแต่ละคลาสแสดงทั้งสีและคำอธิบายพร้อมค่าคลาส (0-4)	<pre>// สร้างคำอธิบายสัญลักษณ์ (Legend) // สร้างแผงคำอธิบาย สัญลักษณ์ var legend = ui.Panel({ style: { position: 'bottom-right', padding: '8px 15px' }}); // หัวข้อคำอธิบาย var legendTitle = ui.Label({ value: 'การใช้ประโยชน์ที่ดิน', style: {fontWeight: 'bold', fontSize: '18px', margin: '0 0 4px 0', padding: '0' }}); legend.add(legendTitle); // ข้อมูลคลาสและสี var classNames = ['Water (น้ำ)', 'Forest (ป่า)', 'Agriculture (เกษตร)', 'Barren (ว่างเปล่า)', 'Urban (เมือง)']; var classColors = ['#0000FF', '#008000', '#FFFF00', '#C0C0C0', '#FF0000']; // สร้างรายการสำหรับแต่ละคลาส for (var i = 0; i < classNames.length; i++) {var colorBox = ui.Label({style: {backgroundColor: classColors[i], padding: '8px', margin: '0 0 4px 0'}}); var className = ui.Label({value: i + ' : ' + classNames[i], style:{margin: '0 0 4px 6px' }}); var legendRow = ui.Panel({widgets: [colorBox, className], layout: ui.Panel.Layout.Flow('horizontal')}); legend.add(legendRow); // เพิ่มคำอธิบายลงในแผนที่ Map.add(legend);</pre>

อภิปรายผลการจำแนกประเภทการใช้ที่ดิน (Land Use Classification)

จากผลลัพธ์ดังรูปที่ 3 ได้อยู่ในรูปแบบ User Interface ที่ได้จากซอร์สโค้ดภาษา JavaScript ทำงานบน GEE ที่พัฒนาขึ้นจากภาษาธรรมชาติด้วยเทคนิค Prompt Engineering แบบ Few-Shot Prompting ซึ่งสามารถบันทึกผลลัพธ์ในรูปแบบของ Earth Engine Assets ที่สามารถส่งออกเป็นไฟล์ผลลัพธ์ให้อยู่ในระบบจัดเก็บข้อมูลอย่าง Google Drive ได้



รูปที่ 3 ผลลัพธ์การจำแนกประเภทการใช้ประโยชน์ที่ดินด้วยตัวจำแนกแบบ SVM ระดับอำเภอเมืองขอนแก่น

ทั้งนี้ผลลัพธ์ที่ได้จากการจำแนกการใช้ที่ดินของอำเภอเมืองขอนแก่นด้วยตัวจำแนกแบบ SVM โดยให้ค่า Overall Accuracy และ Kappa Coefficient (Van Ninh and Waisurasinha, 2018) ที่ 87 % และ 0.84 ดังตารางที่ 4 ซึ่งสูงกว่าเกณฑ์มาตรฐานทางวิชาการ (มากกว่า 80 %) และเทียบเท่าหรือดีกว่างานวิจัยก่อนหน้าในพื้นที่เดียวกัน อย่างเช่นผลงานของ Van Ninh and Waisurasinha (2018) ที่ได้ Overall Accuracy เพียง 82 % และงานของ Biswas et al. (2023) ที่ได้ 85 % ในพื้นที่ที่มีลักษณะคล้ายคลึงกัน ทั้งนี้เนื่องจากความครบถ้วนของชุดข้อมูลฝึกสอนที่มีอยู่กระจายอยู่ทั่วทั้งพื้นที่ที่มีคุณภาพสูง นอกจากนี้เมื่อพิจารณาในเชิงลึกยังพบว่า ตัวจำแนก SVM ให้ค่าความถูกต้องในการจำแนกการใช้ที่ดินแต่ละประเภทนั้นมีแทบทุกประเภทมีค่า Producer's Accuracy สูงกว่า 85 % ยกเว้นการจำแนกการใช้ที่ดินประเภทป่าไม้ที่มีค่าความถูกต้องประมาณ 70 % และเมื่อพิจารณาในรายละเอียดพบว่า มีพื้นที่จำนวนหนึ่งที่ถูกจำแนกเป็นการใช้ที่ดินเป็นแบบเกษตรกรรม ในขณะที่ User's Accuracy ของการใช้ที่ดินแทบทุกประเภทมีค่าสูงกว่า 85 % ยกเว้นการจำแนกการใช้ที่ดินในพื้นที่เกษตรกรรมที่มีความถูกต้องในระดับไม่สูงมากเช่นกันโดยมีค่าประมาณ 65 % โดยมีพื้นที่จำนวนหนึ่งที่ถูกจำแนกเป็นการใช้ที่ดินเป็นแบบป่าไม้ ที่เป็นดังนี้เพราะความคล้ายคลึงของลายเซ็นเชิงคลื่นระหว่างพื้นที่เกษตรกรรมกับพื้นที่ป่าไม้โดยต่างเป็นพื้นที่พืชพรรณที่มีการสะท้อนของคลอโรฟิลล์ทั้งคู่ อย่างไรก็ตามการเปลี่ยนแปลงฤดูกาลที่อาจมีผลต่อภาพ Sentinel-2 นั้นหมายถึงการศึกษาต่อยอดเรื่องการขยายตัวของเมืองขอนแก่นในอนาคตควรเลือกช่วงเวลาเดียวกันกับที่ใช้ในการศึกษานี้เพื่อที่จะสามารถเปรียบเทียบหรืออ้างอิงการขยายตัวกับการศึกษานี้ จากที่กล่าวมาทั้งหมดนี้จะเห็นว่า Generative AI นั้นมีขีดความสามารถในการสร้างซอร์สโค้ดที่ให้ผลลัพธ์ออกมาอย่างเป็นรูปธรรมและสามารถทำงานได้อย่างเต็มประสิทธิภาพ โดยจะเห็นว่าเทคนิค Prompt Engineering แบบ Few-Shot Prompting นั้นเป็นสะพานเชื่อมระหว่างผู้ใช้งานระบบ GEE ในการจำแนกภาพถ่ายดาวเทียมและซอร์สโค้ดภาษา JavaScript เพื่อการเขียนโปรแกรมจากเดิมที่ต้องจดจำวิธีการเขียนโปรแกรมมาเป็นการสร้างข้อความในภาษาธรรมชาติที่มีการผสมผสานเอาเฉพาะคำสำคัญที่เกี่ยวข้อง เช่น การแบ่งขอบเขตการปกครอง การจัดเก็บข้อมูล และการกำหนดสี เป็นต้น ทั้งนี้เพื่อป้องกันการเกิดการหลอนของ AI (AI Hallucination) ที่อาจจะทำให้เกิดการสร้างซอร์สโค้ดที่ไม่สามารถทำงานได้บน GEE ซึ่งเป็นข้อจำกัดของ Generative AI

ตารางที่ 4 ผลการประเมินความถูกต้องของการจำแนกประเภทการใช้ที่ดินของอำเภอเมือง จังหวัดขอนแก่น

ประเภทการใช้ที่ดิน	WA	FA	AG	BA	UR	Total	User's Accuracy (%)
WA	54	0	0	0	1	55	98
FA	0	44	2	0	0	46	96
AG	0	19	42	1	3	65	65
BA	0	0	0	43	1	44	98
UR	0	0	1	6	48	55	87
Producer's Accuracy (%)	100	70	93	86	91	265	

หมายเหตุ : WA = Water (พื้นที่แหล่งน้ำ), FA = Forest (พื้นที่พืชพรรณ), AG = Agriculture (พื้นที่การเกษตร), BA = Barren (พื้นที่ว่างเปล่า), UR = Urban (พื้นที่แหล่งชุมชน)

สรุปผลการวิจัย

การศึกษานี้ได้แสดงให้เห็นถึงศักยภาพของการประยุกต์ใช้ปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative AI) ร่วมกับเทคนิควิศวกรรมข้อความ (Prompt Engineering) บนกรอบการทำงานแบบ Few-Shot Prompting ที่แสดงตัวอย่างซอร์สโค้ดเฉพาะส่วนที่จำเป็นสำหรับ Google Earth Engine (GEE) ผสมผสานกับองค์ความรู้ด้านภูมิสารสนเทศเพื่อใช้ประกอบกันในการสร้างข้อความที่เป็นภาษาธรรมชาติ เพื่อสั่งให้ Generative AI ทำการสร้างซอร์สโค้ดภาษา JavaScript สำหรับ GEE ในการประมวลผลข้อมูลภาพถ่ายดาวเทียมร่วมกับข้อมูลฝึกสอน (Training Samples) ให้กับ

ตัวจำแนกแบบ Support Vector Machine (SVM) ซึ่งให้ผลลัพธ์ที่มีความถูกต้องของการจำแนกการใช้ที่ดินในภาพรวม (Overall Accuracy) ในระดับที่ค่อนข้างสูง โดยผลลัพธ์ของการศึกษานี้ได้แสดงให้เห็นว่า ซอร์สโค้ดภาษา JavaScript สำหรับ GEE ที่สร้างขึ้นจากภาษาธรรมชาติด้วย Generative AI นี้ สามารถนำมาใช้งานได้จริง และสามารถที่จะเพิ่มเติมช่องว่างให้กับผู้สนใจในการใช้งาน GEE สำหรับการวิเคราะห์ข้อมูลภาพถ่ายดาวเทียมบนฐานข้อมูลภูมิสารสนเทศขนาดใหญ่ แต่ยังคงขาดความเชี่ยวชาญในการเขียนโปรแกรมภาษา JavaScript สำหรับ GEE สิ่งนี้นับเป็นการลดอุปสรรคทางเทคนิค และเพิ่มความเร็วในการพัฒนาโปรแกรม อย่างไรก็ตามยังต้องอาศัยความรู้พื้นฐานของ JavaScript สำหรับ GEE ในการออกแบบคำสั่งในภาษาธรรมชาติอยู่บ้าง โดยเป็นระดับ Low-Code Programming ทั้งนี้เพื่อป้องกันการเกิดปรากฏการณ์ที่ Generative AI ที่พัฒนามาบนฐาน LLMs แล้วเกิดการสร้างข้อมูลที่ไม่ถูกต้องหรือที่เราเรียกว่า การหลอนของ Generative AI (AI Hallucination) โดยการป้องกันนี้จะช่วยให้ Generative AI ทำการประมวลผลภาษาธรรมชาติให้ออกมาเป็นซอร์สโค้ด JavaScript สำหรับ GEE ที่สมบูรณ์และใช้งานได้จริง และการสร้างซอร์สโค้ดนี้สามารถทำได้อย่างครอบคลุมตั้งแต่การเตรียมข้อมูล การคำนวณดัชนีเชิงคลื่นต่าง ๆ การฝึกสอนการเรียนรู้ของเครื่อง ไปจนถึงการประเมินประสิทธิภาพและการสร้างส่วนต่อประสานกับผู้ใช้ (User Interface) ซึ่งผลการวิจัยนี้นับเป็นตัวอย่างสำคัญที่แสดงให้เห็นถึงการเปิดโอกาสให้เทคโนโลยีการวิเคราะห์ข้อมูลเชิงพื้นที่ด้วย GEE เข้าถึงได้ง่ายขึ้น ซึ่งจะนำไปสู่การเพิ่มขึ้นของงานวิจัยด้านการจัดการทรัพยากรธรรมชาติและการติดตามสิ่งแวดล้อมต่อไปในอนาคต อนึ่งการศึกษาต่อยอดในอนาคตควรเพิ่มเปรียบเทียบกับเทคนิคอื่นเพิ่มเติมโดยการเปรียบเทียบ SVM กับวิธีอื่น เช่น Random Forest และ CART เป็นต้น ทั้งนี้เพื่อยืนยันความสามารถในการแปลงภาษาธรรมชาติเป็นซอร์สโค้ดภาษา JavaScript สำหรับ GEE Code Editor ที่สามารถทำการวิเคราะห์และจำแนกประเภทการใช้ที่ดิน โดยไม่มีการปรับแก้ซอร์สโค้ดเพิ่มเติมภายหลัง

กิตติกรรมประกาศ

ขอขอบคุณทุนอุดหนุนงานวิทยานิพนธ์จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น และขอกราบขอบพระคุณทุนอุดหนุนงานวิจัยจากศูนย์วิจัยพัฒนามูลฐานอย่างยั่งยืน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

References

- Biswas, J., Jobaer, M.A., Haque, S.F., Islam Shozib, M.S. and Limon, Z.A. (2023). Mapping and Monitoring Land use Land Cover Dynamics Employing Google Earth Engine and Machine Learning Algorithms on Chattogram, Bangladesh. *Heliyon*, 9(11), 1-23. <https://doi.org/10.1016/j.heliyon.2023.e21245>
- Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language Models are Few-Shot Learners* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2005.14165>
- Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B.A. and Reeves, B. N. (2024). *Prompt Problems: A New Programming Exercise for the Generative AI Era*. In Ben Stephenson & Jeffrey A. (Eds.), *The 55th ACM Technical Symposium on Computer Science Education* (pp. 296-302). <https://doi.org/10.1145/3626252.3630909>
- Garg, A., Nisumba Soodhani, K. and Rajendran, R. (2025). Enhancing Data Analysis and Programming Skills Through Structured Prompt Training: The Impact of Generative AI in Engineering Education. *Computers and Education: Artificial Intelligence*, 8. <https://doi.org/10.1016/j.caeai.2025.100380>
- Hochmair, H.H., Juhász, L. and Kemp, T. (2024). Correctness Comparison of ChatGPT-4, Gemini, Claude-3, and Copilot for Spatial Tasks. *Transactions in GIS*, 28(7), 2219-2231. <https://doi.org/10.1111/tgis.13233>

- Hou, S., Shen, Z., Zhao, A., Liang, J., Gui, Z., Guan, X., Li, R. and Wu, H. (2025). GeoCode-GPT: A Large Language Model for Geospatial Code Generation. *International Journal of Applied Earth Observation and Geoinformation*, 138. <https://doi.org/10.1016/j.jag.2025.104456>
- Jhonnerie, R., Oktorini, Y., Volcherina Darlis, V., Pebriandi, P., Prianto, E., Fatmawati, R., Nofrizal, N., Ramses, R., Miswadi, M. and Rahmatdillah, R. (2024). Integration of Generative Artificial Intelligence and Google Earth Engine for Mangrove Land Cover Mapping. *BIO Web of Conferences*, 136. <https://doi.org/10.1051/bioconf/202413603002>
- Li, Z. and Ning, H. (2023). Autonomous GIS: The Next-Generation AI-Powered GIS. *International Journal of Digital Earth*, 16(2), 4668-4686. <https://doi.org/10.1080/17538947.2023.2278895>
- Oscó, L.P., Lemos, E.L., Gonçalves, W.N., Ramos, A.P.M. and Junior, J.M. (2023). The Potential of Visual ChatGPT for Remote Sensing. *Remote Sensing*, 15(13). <https://doi.org/10.3390/rs15133232>
- Sainarmtip, R., Wetchayont, P. and Chenrai, P. (2021). Development of Geothermal Potential Resources Map using Land Surface Temperature from MODIS in Northern Thailand. *RMUTI Journal of Science and Technology*, 14(1), 68-82. <https://ph01.tci-thaijo.org/index.php/rmutijo/article/view/241847/166073> (in Thai)
- Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S. and Brisco, B. (2020). Google Earth Engine for Geospatial Big Data Applications: A Meta-Analysis and Systematic Review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 164, 152-170. <https://doi.org/10.1016/j.isprsjprs.2020.04.001>
- Tao, R. and Xu, J. (2023). Mapping with ChatGPT. *ISPRS International Journal of Geo-Information*, 12(7). <https://doi.org/10.3390/ijgi12070284>
- Van Ninh, T. and Waisurasingha, C. (2018). Land Use/Cover Change and Landscape Fragmentation Analyses in Khon Kaen City, Northeastern Thailand. *International Journal of GEOMATE*, 15(47), 201-208. <https://doi.org/10.21660/2018.47.SGI174>