การคาดการณ์เวลาเดินทางบนท้องถนนระหว่างพิกัดสองจุดในกรุงเทพมหานคร ด้วยวิธีการเรียนรู้ของเครื่อง

นายปวริศ เวชวรรณกิจกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565

Road Travel Time Prediction Between Two Coordinates in Bangkok Using Machine Learning Approaches

Mr. Pawaris Wachwannakijkul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Industrial Engineering
Department of Industrial Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022

หัวข้อวิทยานิพนธ์	การคาดการณ์เวลาเดินทางบนท้องถนนระหว่างพิกัดสองจุดในกรุงเทพมหานคร ด้วยวิธีการเรียนรู้ของเครื่อง
โดย	นายปวริศ เวชวรรณกิจกุล
สาขาวิชา	วิศวกรรมอุตสาหการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.พิศิษฏ์ จารุมณีโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.ดาริชา สุธีวงศ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.พิศิษฏ์ จารุมณีโรจน์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นันทชัย กานตานันทะ)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ชูเวช ชาญสง่าเวช)

ปวริศ เวชวรรณกิจกุล : การคาดการณ์เวลาเดินทางบนท้องถนนระหว่างพิกัดสองจุดในกรุงเทพมหานคร ด้วยวิธีการเรียนรู้ของเครื่อง. (Road Travel Time Prediction Between Two Coordinates in Bangkok Using Machine Learning Approaches) อ.ที่ปรึกษาหลัก : รศ. ดร.พิศิษฎ์ จารุมณีโรจน์

ระยะเวลาเดินทางบนท้องถนนในกรุงเทพมหานครนั้นมีความไม่แน่นอน เนื่องจากความแออัดของการจราจร อย่างไรก็ดี ข้อมูลดังกล่าวกลับมีความสำคัญในการจัดเส้นทางการเดินรถเพื่อธุรกิจ ซึ่งส่วนใหญ่มักใช้ค่าประมาณการณ์ซึ่งอาจมีความคลาดเคลื่อนไปจากความเป็นจริง ส่งผลทำให้ประสิทธิภาพของแผนงานดังกล่าวลดต่ำลง ด้วยเหตุดังกล่าว ผู้วิจัยจึงได้ทำการพัฒนาตัวแบบที่สามารถการคาดการณ์ระยะเวลาเดินทางบนท้องถนนในกรุงเทพมหานครด้วยการเรียนรู้ของเครื่องโดยอ้างอิงจากข้อมูลที่ทุกคนสามารถเข้าถึงได้โดยไม่มีค่าใช้จ่าย ทำให้ผู้ใช้งานสามารถนำไปใช้พัฒนา หรือบูรณาการร่วมกับแผนงานเดิมได้ เริ่มต้น ผู้วิจัยได้ทำการเก็บข้อมูล Mobile Probe จาก iTIC foundation จากนั้นจึงแปลงข้อมูลดังกล่าวออกเป็น Origin-Destination Pairs แล้วคัดเลือกเฉพาะชุดข้อมูลที่มีพิกัดอยู่ภายในเขตกรุงเทพมหานครไปใช้สร้างต้นแบบการเรียนรู้ของเครื่องผ่านอัลกอริทึมแบบต่าง ๆ จนได้อัลกอริทึมที่สามารถสร้างต้นแบบที่มีประสิทธิภาพสูงที่สุดได้ ผู้วิจัยพบว่า จากอัลกอริทึมต่าง ๆ Random forest ถือเป็นอัลกอริทึมที่สามารถสร้างต้นแบบที่มีศักยภาพสูงที่สุด ในขณะที่ XGBoost และ CatBoost มีแนวโน้มที่ดีในการนำไปพัฒนาต่อเนื่องจากใช้ระยะเวลาในการสร้างต้นแบบน้อย

สาขาวิชา วิศวกรรมอุตสาหการ ลายมือชื่อนิสิต .....

ปีการศึกษา 2565 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6270157821 : MAJOR INDUSTRIAL ENGINEERING

KEYWORD: Travel time, Machine learning, Decision tree, Prediction, Ensemble technique

Pawaris Wachwannakijkul : Road Travel Time Prediction Between Two Coordinates in Bangkok Using Machine Learning Approaches. Advisor: Assoc. Prof. Pisit Jarumaneeroj, Ph.D.

While road travel time in Bangkok is uncertain, due largely to traffic congestion, accurate travel time is, however, important for both businesses and research – especially for vehicle route planning that normally adopts estimates that may be far different from their real values. To properly predict road travel time in Bangkok with less data restrictions, several machine learning approaches have been herein explored, based solely on publicly available data so that users can further extend or combine this prediction module with others for their own proposes at the least cost. In doing so, we first collect data from iTIC foundation and later transform such a data set into Origin-Destination Pairs, excluding those outside Bangkok area. Various machine learning approaches are then applied, where Random forest is found to be the most accurate algorithm providing the least MAPE. We also find that, among these many algorithms, XGBoost, CatBoost have good potential to be further investigated – as their computational times are relatively low, but with comparatively high efficiency.

Field of Study: Industrial Engineering Student's Signature .....

Academic Year: 2022 Advisor's Signature .....

กิตติกรรมประกาศ

การที่วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีข้าพเจ้าต้องขอขอบพระคุณ รองศาสตราจารย์ ดร.พิศิษฎ์ จารุมณีโรจน์ ผู้เป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้มอบความรู้คำแนะนำอันมีค่า อีกทั้งความช่วยเหลือต่าง ๆ ตลอดการจัดทำวิทยานิพนธ์ฉบับนี้ ขอขอบคุณ ว่าที่ ร.ต.พิพัฒน์ พิมพะนิตย์ นางสาวชญานี ประคอง นายวรกร เขาวงศ์ และนายกฤติธี ญาณพิสิฐกุล ที่คอยให้คำปรึกษาและความช่วยเหลือในด้านต่าง ๆ อันเป็นประโยชน์ต่อการทำวิทยานิพนธ์

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ผู้ให้กำเนิด รวมทั้งครอบครัวของข้าพเจ้าที่คอยเป็นกำลังใจและแรงสนับสนุนอันดีให้กับข้าพเจ้าตลอดมา

ปวริศ เวชวรรณกิจกุล

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

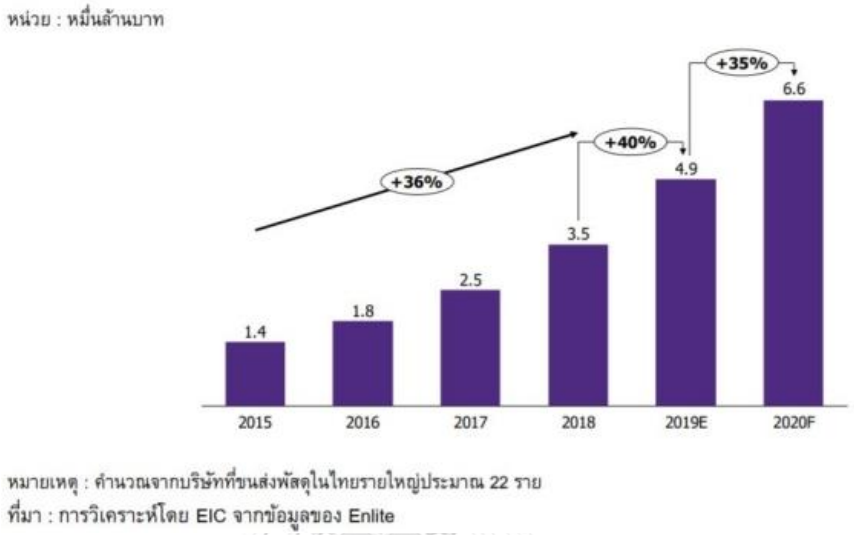
1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันธุรกิจการให้บริการขนส่งทางถนนในประเทศไทยที่มีลักษณะการให้บริการในรูปแบบของการรับส่งสินค้าไม่ว่าจะเป็น คน หรือพัสดุ ล้วนแล้วแต่มีผู้ให้บริการที่หลากหลาย ดังแสดงในภาพที่ 1.1 ซึ่งแสดงให้เห็นถึงผู้ให้บริการธุรกิจบริการขนส่งทางถนนบางส่วน ผู้ให้บริการแต่ละรายอาจประกอบธุรกิจขนส่งมากกว่าหนึ่งประเภท แบ่งตามประเภทของสินค้าที่ขนส่ง (คน, พัสดุ, อาหาร) หรือ ประเภทของลูกค้าที่รับสินค้า (B2B, B2C, C2C) ทำให้มีกระบวนการทำงานแตกต่างกันทั้งในระหว่างผู้ให้บริการ และ ภายในผู้ให้บริการเดียวกัน แต่ในทุกกระบวนการส่งสินค้าไม่ว่าประเภทใดก็ตาม จะมีกระบวนการที่เหมือนกัน คือ การขนส่งไปยังลูกค้าปลายทาง (Last mile delivery)

ภาพที่ 1.1 ตัวอย่างผู้ให้บริการขนส่งทางถนน

การขนส่งไปยังลูกค้าปลายทาง คือ การรับสินค้าจากสถานที่หนึ่งไปส่งยังอีกสถานที่หนึ่ง กระบวนการดังกล่าวเป็นหนึ่งในส่วนสำคัญของธุรกิจขนส่งทางถนน ซึ่งโดยทั่วไปแล้วมักมีประสิทธิภาพต่ำ และ มีต้นทุนสูง (Li et al., 2021) ทั้งนี้ในส่วนของต้นทุนการขนส่งไปยังลูกค้าปลายทางมีค่าระหว่างร้อยละ 13 ถึงร้อยละ 75 ของโซ่อุปทาน (Supply Chain) ทั้งหมด (Gevaers et al., 2009) ส่วนประสิทธิภาพของการขนส่งไปยังลูกค้าปลายทางนั้นขึ้นอยู่กับหลายปัจจัย เช่น ความหนาแน่นของผู้บริโภคและกรอบเวลา (Boyer et al., 2009) ความชำนาญเส้นทางของพนักงานขนส่ง (Cortes & Suzuki, 2021) การนำจ่ายพัสดุไม่สำเร็จ (Gevaers et al., 2009) เป็นต้น ขณะที่มูลค่าตลาดขนส่งพัสดุในประเทศไทย ตั้งแต่ปี ค.ศ.2015 ถึง ค.ศ.2020 เติบโตขึ้นทุกปี ดังแสดงในภาพที่ 1.2 การวางแผนเส้นทางเดินรถเพื่อขนส่งสินค้าจึงมีส่วนสำคัญในการเพิ่มประสิทธิภาพของการขนส่งไปยังลูกค้าปลายทาง แรกเริ่มในงานวิจัยส่วนใหญ่มักมุ่งเน้นไปที่การแก้ไขปัญหาวิถีสั้นสุด

(Shortest path) จากข้อมูลชุดเดิมเพียงชุดเดียว (Geng et al., 2020) ต่อมาจึงเริ่มมีการพัฒนาปัญหาการจัดเส้นทางในรูปแบบเดิม ให้มีความเสมือนจริงมากขึ้นเพื่อรับมือกับเงื่อนไขสภาพแวดล้อมที่เปลี่ยนแปลงไปอยู่ตลอดเวลา เช่น ความหนาแน่นของการจราจรในแต่ละช่วงเวลาและสถานที่ ที่ส่งผลต่อระยะเวลาในการเดินทาง (Frohner et al., 2021)



ภาพที่ 1.2 มูลค่าตลาดขนส่งพัสดุของไทย (วาที, 2563)

กรุงเทพมหานคร เป็นเมืองหลวงของประเทศไทย ซึ่งมีอันดับระดับความแออัดของสภาพการจราจรอยู่ในอันดับที่ 10 จาก 416 เมือง 57 ประเทศ จากการจัดอันดับสามารถกล่าวได้ว่ากรุงเทพมหานครเป็นเมืองที่มีสภาพการจราจรแออัดมาก และ TomTom (2021b) ได้มีการแสดงร้อยละความแออัดของสภาพการจราจรตามช่วงเวลาไว้ ดังแสดงในภาพที่ 1.3

WEEKLY TRAFFIC CONGESTION BY TIME OF DAY

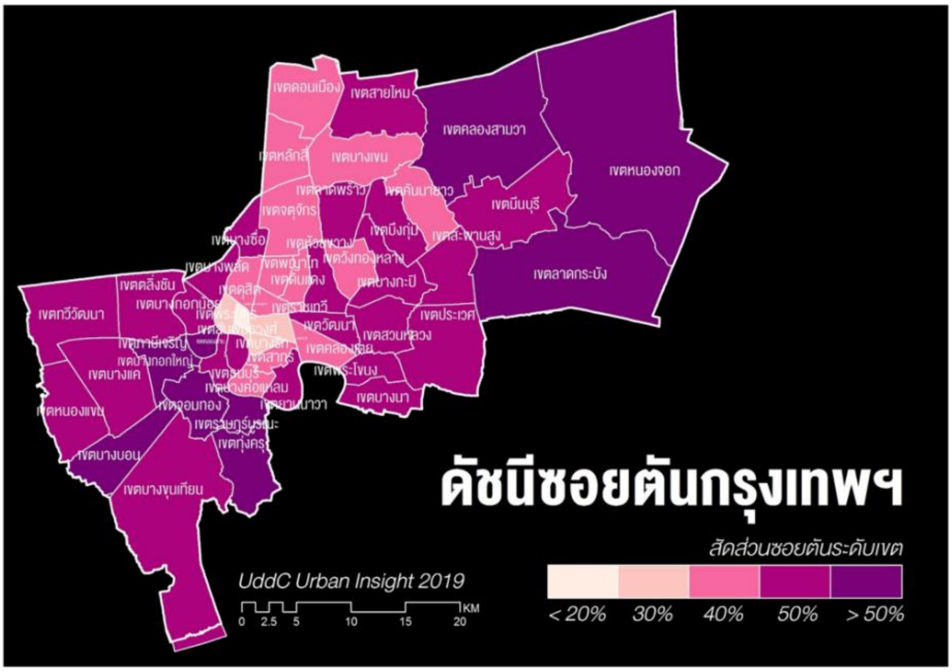
What time was rush hour in Bangkok?

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
12:00 AM	3%	0%	0%	1%	0%	1%	5%
	0%	0%	0%	0%	0%	0%	1%
02:00 AM	0%	0%	0%	0%	0%	0%	0%
	0%	0%	0%	0%	0%	0%	0%
04:00 AM	0%	0%	0%	0%	0%	0%	0%
	0%	9%	8%	7%	7%	6%	1%
06:00 AM	0%	43%	39%	38%	35%	33%	4%
	4%	72%	68%	66%	62%	57%	13%
08:00 AM	10%	67%	67%	64%	60%	55%	23%
	15%	52%	57%	55%	53%	50%	35%
10:00 AM	21%	43%	49%	48%	48%	48%	46%
	25%	37%	42%	42%	43%	46%	51%
12:00 PM	28%	34%	38%	39%	39%	44%	54%
	28%	32%	36%	37%	37%	44%	51%
02:00 PM	29%	36%	40%	41%	41%	50%	48%
	30%	40%	44%	46%	45%	57%	47%
04:00 PM	32%	48%	54%	55%	55%	69%	48%
	35%	72%	79%	81%	81%	95%	56%
06:00 PM	33%	79%	86%	87%	86%	102%	51%
	27%	54%	61%	62%	60%	77%	36%
08:00 PM	23%	30%	36%	37%	35%	52%	26%
	15%	16%	20%	20%	21%	34%	19%
10:00 PM	9%	9%	11%	12%	13%	23%	15%
	2%	3%	4%	5%	6%	12%	8%

ภาพที่ 1.3 ความแออัดของสภาพการจราจรในกรุงเทพมหานคร (TomTom, 2021a)

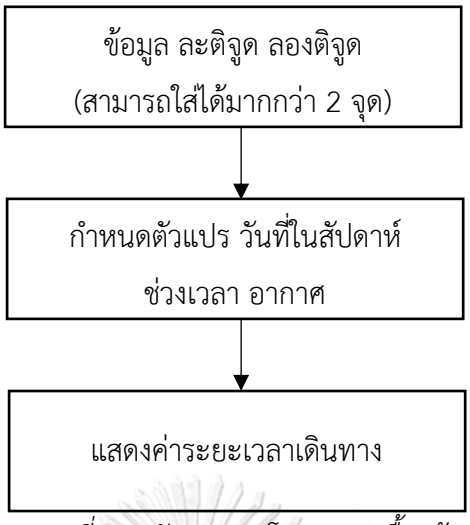
ในภาพที่ 1.3 แสดงให้เห็นถึงความแปรปรวนของความแออัดของสภาพการจราจรในกรุงเทพมหานคร ซึ่งมีความแตกต่างกันในแต่ละช่วงเวลาของแต่ละวันในสัปดาห์ ส่งผลทำให้ระยะเวลาในการเดินทางจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่ง อาจแตกต่างกันในแต่ละช่วงเวลา

กรุงเทพมหานครเป็นเมืองที่มีปัญหาทางด้านจราจรมาอย่างยาวนาน สาเหตุแรกเริ่มมาจากการสร้างโครงข่ายถนนของกรุงเทพมหานคร ซึ่งสามารถแสดงให้เห็นถึงปัญหาได้จากดัชนีซอยตันในแต่ละเขต ดังภาพที่ 1.4 (เพิ่มศักดิ์ พูลพรม, 2548) ทั้งนี้พื้นที่ถนนของกรุงเทพมหานครมีเพียงร้อยละ 9 เท่านั้น เมื่อเทียบกับในสัดส่วนถนนขั้นต่ำสากลที่เหมาะสมที่ร้อยละ 20 ของพื้นที่เมือง (จารึก ประพันธ์พจน์, 2533) นอกจากนี้กรุงเทพมหานครยังเป็นศูนย์กลางของความเจริญในประเทศไทยจากแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ (วรรธนา พันธ์สว่าง, 2539) กรุงเทพมหานครจึงมีประชากรย้ายที่อยู่อาศัยเข้ามาจำนวนมาก อัตราการเปลี่ยนแปลงของประชากรเพิ่มสูงขึ้นเฉลี่ยประมาณร้อยละ 4 ต่อปี (จารุวรรณ ลิมปเสนีย์, 2521) ทำให้รถโดยสารประจำทางไม่เพียงพอต่อความต้องการ ประชากรเริ่มใช้รถยนต์ส่วนตัวมากขึ้น (บุญเสริม อินทรตุล, 2517) ปัญหาโครงสร้างเหล่านี้ล้วนแล้วแต่ส่งผลให้ระดับความแออัดของสภาพการจราจรของกรุงเทพมหานครถูกจัดอยู่ในอันดับต้น ๆ ของโลก



ภาพที่ 1.4 ดัชนีซอยตันกรุงเทพมหานคร (อดิศักดิ์ กันทะเมืองลี้, 2562)

จากการค้นคว้างานวิจัยการจัดเส้นทางที่นำความไม่แน่นอนของเวลาเดินทางเข้ามาประกอบด้วย (Online or real-time problem) พบว่า มีหลายเทคนิคที่ถูกนำมาใช้ในส่วนของการหาระยะเวลาเดินทางที่เปลี่ยนแปลงไปในแต่ละช่วงเวลา เช่น การใช้แบบจำลองแบบสุ่ม (Stochastic Model) (Yu & Yang, 2019) การเรียกข้อมูลจากแอพพลิเคชั่น Google Map (Google distance Matrix API) (Parinya, 2019) การวิเคราะห์ข้อมูลที่รวบรวมได้จากสื่อโซเชี่ยล (Hadiyanto et al., 2019) ซึ่งงานวิจัยเหล่านี้ไม่ได้กล่าวถึงรายละเอียดของข้อมูลที่นำมาใช้หรือความแม่นยำของระยะเวลาเดินทางมากนัก ทำให้ไม่สามารถนำมาใช้เปรียบเทียบผลคำตอบกับงานวิจัยประเภทเดียวกันได้ นอกจากนี้แม้ว่าเราจะสามารถใช้การเรียกข้อมูลระยะเวลาเดินทาง จากแอพพลิเคชั่นผู้ให้บริการแผนที่ต่าง ๆ เช่น Google Map, TomTom, Apple Map, Longdo Traffic, Nostra Map การเรียกข้อมูลในปริมาณที่มากกว่าข้อจำกัด จำเป็นต้องเสียค่าใช้จ่าย (การเรียกข้อมูลจากแอพพลิเคชั่น MapQuest ซึ่งเป็นผู้ให้บริการแผนที่แบบไม่มีค่าใช้จ่าย ไม่มีข้อจำกัดในด้านปริมาณการเรียกข้อมูล แต่ก็มีข้อจำกัดในด้านของขนาดข้อมูลและการกำหนดตัวแปรต่าง ๆ) ผู้วิจัยจึงมีความสนใจในการสร้างต้นแบบการคาดการณ์ระยะเวลาเดินทางระหว่างพิกัดสองจุดด้วยวิธีการเรียนรู้ของเครื่อง (Machine learning) ในเขตพื้นที่ของจังหวัดกรุงเทพมหานคร โดยที่ผู้ใช้งานต้นแบบสามารถกำหนดตัวแปรสภาพแวดล้อมต่าง ๆ ในการคาดการณ์ได้ เช่น วันในสัปดาห์ สภาพอากาศ ช่วงเวลา โดยโปรแกรมจะแสดงค่ากลับมาเป็นระยะเวลาเดินทาง โดยไม่มีการจำกัดปริมาณการเรียกข้อมูล ผังงานของโปรแกรมเบื้องต้น (Flowchart) สามารถแสดงได้ดังภาพที่ 1.5



ภาพที่ 1.5 ผังงานของโปรแกรมเบื้องต้น

เราสามารถเปรียบเทียบข้อจำกัดในการเรียกข้อมูลระยะเวลาเดินทางของแอพพลิเคชั่นผู้ให้บริการแผนที่ต่าง ๆ แบบไม่มีค่าใช้จ่ายกับงานวิจัยนี้ได้ดังตารางที่ 1.1

ตารางที่ 1.1 เปรียบเทียบข้อจำกัดในการเรียกข้อมูลระยะเวลาเดินทางของผู้ให้บริการแผนที่

ผู้ให้บริการแผนที่	ข้อจำกัด		
	ปริมาณการเรียกข้อมูล	ขนาดของข้อมูล	ตัวแปรสภาพแวดล้อม
Google Map	✓	✓	✓
TomTom	✓	✓	✓
Apple Map	✓	✓	✓
Longdo Traffic	✓	✓	✓
Nostra Map	✓	✓	✓
Openrouteservice	✓	✓	✓
MapQuest	✗	✓	✓
Here	✓	✗	✓
งานวิจัย	✗	✗	✗

หมายเหตุ ✓ หมายถึง ผู้ให้บริการแผนที่มีข้อจำกัดนั้น

✗ หมายถึง ผู้ให้บริการแผนที่ไม่มีข้อจำกัดนั้น

จากตารางที่ 1.1 แสดงให้เห็นว่าการเรียกข้อมูลระยะเวลาเดินทางจากผู้ให้บริการแผนที่แบบไม่เสียค่าใช้จ่าย มักจะมีข้อจำกัดในการเรียกข้อมูลในด้าน ปริมาณการเรียกข้อมูล ขนาดของข้อมูล หรือการตั้งค่าตัวแปรสภาพแวดล้อม ซึ่งในงานวิจัยนี้จะทำให้สามารถเรียกข้อมูลระยะเวลาเดินทางได้อย่างไม่มีข้อจำกัดใด ๆ ด้วยการนำข้อมูลที่เกี่ยวข้องกับการจราจร เช่น ข้อมูลการติดตามพิกัดยานหนะ ข้อมูลปริมาณน้ำฝน ข้อมูลดัชนีชี้วัดรถติด เป็นต้น มาทำการแปลงให้เป็นตัวแปร เช่น พิกัดตอนที่รับผู้โดยสาร พิกัดตอนที่ส่งผู้โดยสาร ระยะทางในการเดินทาง ปริมาณน้ำฝน และ ตัวแปรอื่น ๆ นำตัวแปรเหล่านี้ไปสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่อง ซึ่งจะทำให้ได้ต้นแบบที่สามารถคาดการณ์ระยะเวลาในการเดินทาง โดยไม่เสียค่าใช้จ่ายใด ๆ

1.1 วัตถุประสงค์

- เพื่อสร้างต้นแบบการเรียนรู้ของเครื่องในการคาดการณ์ระยะเวลาเดินทางระหว่างพิกัดสองจุดในกรุงเทพมหานคร และพัฒนาต้นแบบให้ค่าความผิดพลาดระหว่างระยะเวลาเดินทางที่คาดการณ์กับระยะเวลาเดินทางจากข้อมูลจริงมีค่าต่ำ

1.2 ขอบเขตของการวิจัย

- ต้นแบบสามารถใช้ได้เฉพาะพิกัดในพื้นที่ 50 เขตของจังหวัดกรุงเทพมหานคร
- ข้อมูลที่นำมาใช้เป็นข้อมูลที่สามารถเข้าถึงได้แบบสาธารณะโดยไม่เสียค่าใช้จ่าย
- ประเภทของพาหนะในข้อมูลที่นำมาใช้ คือ แท็กซี่โดยสารที่ปิดไฟรับผู้โดยสาร ปริมาณ 12 ล้าน ตัวอย่าง
- ข้อมูลที่นำมาใช้อยู่ในช่วงเวลาระหว่าง 1 มกราคม ค.ศ.2020 ถึง 31 ธันวาคม ค.ศ.2020
- ข้อมูลพิกัดเริ่มต้น และ พิกัดสิ้นสุด (Origin-destination data) ไม่รวมถึงเส้นทางในการเดินทาง

1.3 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

- เผยแพร่ต้นแบบการเรียนรู้ของเครื่องเพื่อให้ผู้ที่สนใจสามารถนำไปใช้งาน ศึกษา หรือพัฒนาต่อได้
- สามารถจัดลำดับความสำคัญของปัจจัยที่มีผลกระทบต่อระยะเวลาเดินทางในจังหวัดกรุงเทพมหานคร

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การคาดการณ์เวลาเดินทาง เป็นหนึ่งในส่วนสำคัญของระบบ ITS (Intelligent transportation system) ในช่วง 30 ปีที่ผ่านมา นักวิจัยด้านการขนส่ง และ นักวิทยาศาสตร์ข้อมูลได้พัฒนาหลากหลายเทคนิคเพื่อเพิ่มความน่าเชื่อถือแก่วิธีการคาดการณ์เวลาเดินทางในอนาคต (Oh et al., 2015) โดยทั่วไปแล้วเทคนิคเหล่านั้นสามารถแบ่งออกได้เป็น 3 กลุ่ม ได้แก่ Naive methods, traffic theory-based method และ data-driven methods (Fan & Qiu, 2021) โดยเทคนิค data-driven นั้นจะใช้ชุดข้อมูลที่เกี่ยวข้องกับการเดินทาง และ ข้อมูลวิธีทางสถิติในการคาดการณ์เวลาเดินทาง ทำให้มีผู้ที่สามารถวิจัยหัวข้อนี้ได้มากขึ้น เพราะไม่จำเป็นต้องมีความเชี่ยวชาญในด้านทฤษฎีการจราจรโดยเฉพาะ เทคนิคนี้จำเป็นต้องใช้ข้อมูลปริมาณมากซึ่งอาจจะไม่ได้มีอยู่ในทุกที่ ต้นแบบที่สร้างจากเทคนิคนี้จะมีความแม่นยำมากขึ้นจากความพร้อมของข้อมูล (Lint, 2006)

ถึงแม้ว่าจะมีงานวิจัยมากมายที่เกี่ยวข้องกับการคาดการณ์ระยะเวลาเดินทางบนท้องถนน แต่มีไม่มากนักที่นำวิธีการเรียนรู้ของเครื่องมาใช้ร่วมกัน และงานวิจัยประเภทนี้ยังมีการใช้ชุดข้อมูลจากแหล่งข้อมูลที่แตกต่างกัน เช่น ข้อมูลสาธารณะของแท็กซี่ในนิวยอร์ก (NYC Yellow Cab trip record) (Huang & Xu, 2018) เว็บไซต์วางแผนการเดินทาง Uber Movement (Deb et al., 2019) เป็นต้น ข้อมูลเหล่านี้เป็นข้อมูลที่บันทึกจากแต่ละประเทศ ซึ่งมีความแตกต่างในโครงข่ายถนน พฤติกรรมการขับขี่ ความหนาแน่นของการจราจร พื้นที่ภูมิประเทศ ส่งผลทำให้การคาดการณ์ระยะเวลาเดินทางบนท้องถนนหนึ่ง ๆ ไม่สามารถนำไปประยุกต์ใช้กับการคาดการณ์ระยะเวลาอื่น ๆ ได้อย่างมีความแม่นยำ งานวิจัยประเภทเดียวกันจึงมีความแตกต่างกันอย่างเห็นได้ชัดในด้านของชุดข้อมูล และ วิธีการที่ใช้คาดการณ์ ซึ่งงานวิจัยที่ใช้วิธีการเรียนรู้ของเครื่องส่วนใหญ่จะหยิบยกอัลกอริทึมบางอย่างสำหรับวิเคราะห์การถดถอยมาใช้ เช่น Linear Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF) โดยการคัดเลือกอัลกอริทึมเหล่านี้มาใช้คาดการณ์ก็เป็นอีกหนึ่งความแตกต่างของงานวิจัยประเภทนี้

ในงานวิจัยบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับวิธีการเรียนรู้ของเครื่องซึ่งประกอบไปด้วย ความหมายของวิธีการเรียนรู้ของเครื่อง วิธีการคัดเลือกและประเมินผลของต้นแบบ การลดความผิดพลาดที่เกิดขึ้นจากการสร้างต้นแบบ และ อัลกอริทึมต่าง ๆ ของวิธีการเรียนรู้ของเครื่องที่นำมาใช้ในการสร้างต้นแบบ

2.1 วิธีการเรียนรู้ของเครื่อง

Mohri et al. (2018) กล่าวว่า เราสามารถให้ความหมายแบบกว้าง ๆ ของวิธีการเรียนรู้ของเครื่องว่า เป็นวิธีการคำนวณด้วยคอมพิวเตอร์โดยการใช้ประสบการณ์ในการเพิ่มประสิทธิภาพหรือทำให้การคาดการณ์แม่นยำขึ้น โดยประสบการณ์ในที่นี้หมายถึงข้อมูลสารสนเทศที่ผ่านมาในอดีต ซึ่งโดยปกติจะถูกจัดเก็บอยู่ในรูปแบบข้อมูลอิเล็กทรอนิกส์ ข้อมูลเหล่านี้จะถูกดัดแปลงให้อยู่ในรูปแบบของตัวเลขเชิงปริมาณและทำการจัดแบ่งประเภทของข้อมูลโดยมนุษย์เพื่อให้สามารถนำมาใช้ได้ในวิธีการเรียนรู้ของเครื่อง ทั้งนี้คุณภาพ และ ขนาดของข้อมูลมีความสำคัญในประสิทธิผลของการคาดการณ์ที่ถูกสร้างโดยวิธีการเรียนรู้ของเครื่อง

วิธีการเรียนรู้ของเครื่องนั้นถูกออกแบบมาเพื่อใช้กับงานหลากหลายรูปแบบ ในงานวิจัยนี้จะใช้วิธีการเรียนรู้ของเครื่องกับปัญหาการถดถอย (Regression) ซึ่งเป็นปัญหาในการคาดการณ์ค่าที่แท้จริงของสิ่งของแต่ละอย่าง ตัวอย่างเช่น การคาดการณ์มูลค่าของหุ้นหรือความผันแปรของตัวแปรทางเศรษฐกิจ โดยความผิดพลาดของงานการถดถอยนี้จะขึ้นอยู่กับขนาดของความแตกต่างระหว่างค่าที่เกิดขึ้นจริงและค่าที่เกิดจากการคาดการณ์

ประเภทของวิธีการเรียนรู้ของเครื่องจะถูกแบ่งโดยข้อมูลที่ใช้ ซึ่งในงานวิจัยนี้จะเป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) กล่าวคือ ผู้เรียนรู้จะได้รับข้อมูลที่มีคำตอบอยู่แล้ว และทำการคาดการณ์เพื่อเปรียบเทียบกับคำตอบนั้น

2.2 การเลือกอัลกอริทึมในการสร้างต้นแบบ

การเลือกอัลกอริทึมเพื่อสร้างต้นแบบจากปัญหาการถดถอยในงานวิจัยนี้จะอ้างอิงจากการแข่งขัน Kaggle (Kaggle Competition) ซึ่งเป็นการแข่งขันทำโจทย์ทางด้าน Data Science รวมถึงการใช้วิธีการเรียนรู้ของเครื่องเพื่อสร้างต้นแบบแล้วหาผู้ชนะที่สามารถสร้างต้นแบบให้มีค่าเมตริกความผิดพลาดน้อยที่สุด โดยเป็นการแข่งขันระดับโลกที่เปิดให้ทุกคนเข้าร่วม ซึ่งอัลกอริทึมที่สามารถสร้างความสำเร็จในหลาย ๆ การแข่งขันก็คือ Gradient Boosting หรืออัลกอริทึมที่มีการใช้วิธีร่วมกันตัดสินใจ (Ensemble) (KA-KA-shi, 2021) อัลกอริทึมที่มีการใช้ Gradient Boosting และวิธีร่วมกันตัดสินใจเช่น XGBoost, LightGBM, CatBoost หรือ Random forest (ที่ใช้เฉพาะวิธีร่วมกันตัดสินใจกับต้นไม้ตัดสินใจแบบดั้งเดิม) หลังจากนี้จะเรียกรวมอัลกอริทึมที่กล่าวมาว่าอัลกอริทึม Tree based

หากเทียบอัลกอริทึม Tree based กับอัลกอริทึมที่ทันสมัยกว่าอย่าง Neural Network ซึ่งมีความซับซ้อน และถูกเรียกว่าเป็นทางออกของทุกปัญหาในวิธีการเรียนรู้ของเครื่องแล้ว Tree based นั้นอาจดูมีความน่าสนใจน้อยกว่าเนื่องจากเป็นอัลกอริทึมที่เรียบง่าย (Andre Ye, 2020) แต่กลับเป็นอัลกอริทึมที่ได้รับความนิยมสูงสุดในด้านการแข่งขันประสิทธิภาพของต้นแบบ ทั้งสองอัลกอริทึมนี้ถูก

จัดให้อยู่ในประเภทเดียวกันที่มีการแยกโครงสร้างของโจทย์ และทำการวิเคราะห์ทีละส่วนแทนที่การใช้ขอบเขตเพียงอันเดียวที่สามารถครอบคลุมได้ทั้งชุดข้อมูลอย่าง Support Vector Machine หรือ Linear Regression

อัลกอริทึม Tree based นั้นมีประสิทธิภาพเหนือกว่าอัลกอริทึม Neural Network เมื่อใช้กับข้อมูลที่เป็นรูปแบบตาราง (Tabular data) (Grinsztajn et al., 2022) ซึ่งในงานวิจัยได้บอกถึงเหตุผลไว้ดังนี้

- Neural Network จะมี Biased และหาจุดที่เหมาะสมได้แย่กว่า Tree based เมื่อเป็น Non-smooth functions หรือ Decision boundaries
- เมื่อใช้ชุดข้อมูลขนาดใหญ่ ตัวแปรที่ไม่มีความสัมพันธ์กันจะทำให้ Neural Network มีประสิทธิภาพแย่ลง
- Neural Network ไม่แปรผันกับการ Rotation ข้อมูล

เนื่องจากงานวิจัยนี้ใช้ชุดข้อมูลที่เป็นรูปแบบตาราง ผู้วิจัยจึงเลือกใช้อัลกอริทึม Tree based ที่สามารถวิเคราะห์ข้อมูลแยกเป็นส่วน ๆ และเหมาะสมกับชุดข้อมูลนี้ในการสร้างต้นแบบ

2.3 การคัดเลือกและการประเมินผลของต้นแบบ

การสร้างต้นแบบ การคาดการณ์ระยะเวลาเดินทางบนท้องถนนระหว่างพิกัดสองจุดด้วยวิธีการเรียนรู้ของเครื่อง จำเป็นต้องมีการคัดเลือกต้นแบบที่ดีที่สุดไม่ว่าจะเป็นในการสร้างต้นแบบเดียวกันหรือระหว่างต้นแบบอื่น ๆ ซึ่งในปัญหาการถดถอย จะทำการสร้างต้นแบบเพื่อคาดการณ์ค่าที่ต้องการจากข้อมูล โดยสามารถเขียนสมการเพื่อคำนวณค่าความผิดพลาดของต้นแบบได้ดังนี้

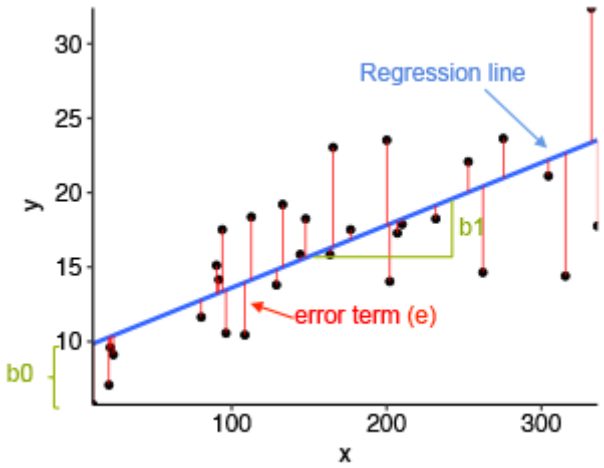
$$Error = prediction - actual \qquad (2.1)$$

โดยที่ $Error$ คือ ค่าความผิดพลาดของต้นแบบ

$prediction$ คือ ค่าที่คาดการณ์ได้จากต้นแบบ

actual คือ ค่าที่ใส่ให้กับผู้เรียนรู้

ในการเรียนรู้ของเครื่อง เรียกสมการนี้ว่า Loss function โดยเป้าหมายของการสร้างต้นแบบ คือการทำให้ตัวแปรนี้มีค่าน้อยที่สุด ยกตัวอย่างกราฟของต้นแบบการถดถอยเชิงเส้น ดังแสดงในภาพที่ 2.1



ภาพที่ 2.1 ตัวอย่างกราฟของต้นแบบการถดถอยเชิงเส้น (พรทิวา วิศิษฏ์สรอรรถ, 2562)

จากภาพที่ 2.1 แสดงถึงกราฟของต้นแบบการถดถอยเชิงเส้นที่คาดการณ์ค่า y ด้วยตัวแปร x จุดสีดำ คือ ค่าที่ใส่ให้กับผู้เรียนรู้ เส้นสีน้ำเงิน คือ ค่าคาดการณ์ที่ได้จากต้นแบบการถดถอยเชิงเส้น และเส้นสีแดง คือ ค่าผิดพลาดที่ต้นแบบต้องการทำให้มีค่าน้อยที่สุด

ในการคำนวณค่าความผิดพลาดจากต้นแบบ จากกราฟจะเห็นได้ว่าค่าผิดพลาดที่ได้จะมีทั้งค่าบวก และค่าลบ ซึ่งถ้าคำนวณโดยปกติอาจจะทำให้เกิดการหักล้างกันเอง จึงต้องมีการใช้เมตริกต่าง ๆ ในการช่วยคำนวณค่าผิดพลาดของต้นแบบ เช่น MAE (Mean Abosolute Error), MSE (Mean Squred Error), RMSE (Root Mean Square Error) และ R^2 (R-Squared) โดยที่ เมตริก 3 ตัวแรก จะมีค่าอยู่ที่ 0 ถึง ∞ และไม่สนใจทิศทางของค่าความผิดพลาด ยิ่งมีค่าต่ำหมายความว่าต้นแบบนั้นมีประสิทธิภาพที่ดี ส่วนตัวสุดท้ายจะมีค่า 0 ถึง 1 ยิ่งมีค่ามากหมายความว่าต้นแบบนั้นมีความเหมาะสมกับข้อมูล

2.3.1 Mean Absolute Error

เมตริก MAE (Mean Absolute Error) เป็นเมตริกที่ทำให้ค่าผิดพลาดเป็นบวกโดยทำให้เป็นค่าสัมบูรณ์ แล้วนำไปหาค่าเฉลี่ย เมตริก MAE จะได้รับผลกระทบจากค่าผิดปกติน้อยกว่าเมตริก MSE และ RMSE โดยสามารถเขียนสมการการคำนวณเมตริก MAE ได้ดังนี้

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Error| \qquad (2.2)$$

2.3.2 Mean Absolute Percentage Error

เมตริก MAPE (Mean Absolute Percentage Error) เป็นเมตริกที่คล้ายกับเมตริก MAE จะบ่งบอกถึงความแตกต่างระหว่างค่าเฉลี่ย และ ค่าที่คาดการณ์ได้ในรูปแบบร้อยละ โดยสามารถเขียนสมการคำนวณเมตริก MAPE ได้ดังนี้

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{|Error|}{|actual|}\right) * 100$$

2.3.3 Mean Squared Error

เมตริก MSE (Mean Squared Error) เป็นเมตริกที่ทำให้ค่าผิดพลาดเป็นบวกโดยการยกกำลังสอง แล้วนำไปหาค่าเฉลี่ย เมตริก MSE เหมาะสำหรับการคำนวณประสิทธิภาพต้นแบบที่ต้องการพิจารณาข้อมูลที่มีค่าผิดปกติ เนื่องจากการยกกำลังสองจะมีผลกับค่าผิดปกติมาก และเห็นได้ชัดกว่าเมตริก MAE โดยสามารถเขียนสมการการคำนวณเมตริก MSE ได้ดังนี้

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Error)^2 \qquad (2.3)$$

2.3.4 Root Mean Square Error

เมตริก RMSE (Root Mean Square Error) เป็นเมตริกที่เป็นรากที่สองของเมตริก MSE เพื่อให้ได้ค่าที่มีหน่วยเดียวกับข้อมูลที่ใส่ให้ผู้เรียนรู้ ทำให้สามารถตีความประสิทธิภาพจากเมตริกได้ง่ายขึ้น โดยสามารถเขียนสมการการคำนวณเมตริก RMSE ได้ดังนี้

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Error)^2} \qquad (2.4)$$

2.3.5 R-Squared

เมตริก R^2 (R-Squared) เป็นเมตริกที่บ่งบอกถึงค่าความแปรปรวนของต้นแบบที่สร้างขึ้นว่าสามารถอธิบายได้จากความแปรปรวนที่เกิดขึ้นทั้งหมดเป็นสัดส่วนเท่าใด โดยสามารถเขียนสมการการคำนวณเมตริก R^2 ได้ดังนี้

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n}(Error)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}\right) \qquad (2.5)$$

โดยที่ $y_1, y_2, \dots, y_n$ คือ ค่าที่ใส่ให้กับผู้เรียนรู้

$\bar{y}$ คือ ค่าเฉลี่ยของค่าที่ใส่ให้กับผู้เรียนรู้

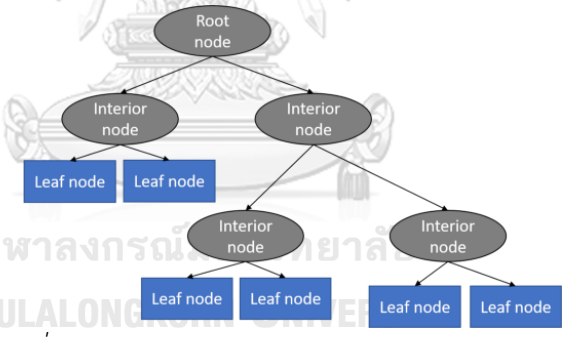
2.4 การลดความผิดพลาดที่เกิดขึ้นจากการสร้างต้นแบบ

ปัญหาที่เกิดขึ้นในการสร้างต้นแบบเป็นปัญหาที่เกิดจากโครงสร้างของวิธีการเรียนรู้ของเครื่อง ซึ่งหากเราสร้างต้นแบบจากข้อมูลชุดหนึ่งโดยตั้งจุดประสงค์ในการทำให้ค่าผิดพลาดมีค่าน้อยมาก ๆ เมื่อนำต้นแบบนี้ไปใช้กับข้อมูลชุดอีกชุดหนึ่งที่ผู้เรียนรู้ไม่เคยเห็นมาก่อนอาจมีความเสี่ยงที่ค่าผิดพลาดจะมากขึ้น เพราะตัวต้นแบบเกิดการจำลักษณะเฉพาะของข้อมูลชุดแรกมากเกินไป โดยเรียกปัญหานี้ว่า overfitting (Dietterich, 1995)

ในงานวิจัยนี้ ผู้วิจัยได้แก้ไขปัญหาการเกิด overfitting กับต้นแบบโดยใช้วิธี K-fold cross validation (Rodriguez et al., 2009) หลักการของวิธีนี้ คือ การแบ่งข้อมูลเป็น K ส่วน แบบสุ่ม เพื่อให้ข้อมูลมีการกระจายตัวทำการวัดประสิทธิภาพของต้นแบบโดยการใช้ข้อมูล 1 ส่วนเรียกส่วนนี้ว่า test set และส่วนที่เหลือจะนำมาใช้ในการสร้างต้นแบบเรียกส่วนนี้ว่า train set ดำเนินการซ้ำโดยสลับ test set จนครบ K ส่วนและหาต้นแบบที่มีประสิทธิภาพมากที่สุดวิธีนี้จะทำให้เกิดการสร้างและทดสอบต้นแบบหลายครั้งช่วยในแก้ไขปัญหาการเกิด overfitting

2.5 ต้นไม้ตัดสินใจ

ต้นไม้ตัดสินใจ (Decision tree) เป็นหนึ่งในอัลกอริทึมที่นิยมในการใช้งาน หรือ นำมาประยุกต์ใช้กับการสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่องแบบมีผู้สอน สามารถสร้างต้นแบบได้ทั้งปัญหาแบบถดถอย (Regression) และ ปัญหาแบบจำแนก (Classification) โครงสร้างของต้นไม้ตัดสินใจมีอยู่ 3 อย่าง ได้แก่ โหนดราก (Root node) คือ โหนดเริ่มต้นของตัวอย่างทั้งหมด โหนดภายใน (Interior node) คือ โหนดที่แสดงถึงปัจจัย (Feature) ของชุดข้อมูล สุดท้ายคือ โหนดใบ (Leaf node) แสดงถึงผลลัพธ์ ตัวอย่างต้นไม้ตัดสินใจดังแสดงในภาพที่ 2.2



ภาพที่ 2.2 ตัวอย่างต้นไม้ตัดสินใจ (Klein, 2017)

หลักการของต้นไม้ตัดสินใจ คือ การแบ่งข้อมูลออกทีละ 2 ส่วน (Recursive Binary split) จากโหนดรากจนถึงโหนดใบ และ ทำการคาดการณ์ค่าของเป้าหมายหรือคำตอบ (Target variable) (วิชญ์พงศ์ ดรุณธรรม, 2561) จุดในการแบ่งข้อมูลในแต่ละโหนดจะกำหนดด้วยค่า RSS (Residual sum of squares) ที่น้อยที่สุด สามารถเขียนสมการได้ดังนี้

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \qquad (2.6)$$

โดยที่ J คือ จำนวนกลุ่มของตัวอย่างที่ถูกแบ่งออกมา

R_j คือ แต่ละกลุ่มของตัวอย่างที่ถูกแบ่งออกมา

y_i คือ ค่าของเป้าหมาย

$\hat{y}_{R_j}$ คือ ค่าคาดการณ์ในแต่ละกลุ่ม

ค่าคาดการณ์ในแต่ละกลุ่มสามารถหาได้จากค่าเฉลี่ยของค่าเป้าหมายในกลุ่มนั้น ๆ สามารถเขียนสมการได้ดังนี้

$$\hat{y}_{R_j} = \frac{1}{n}\sum_{j\in R_j} y_j \qquad (2.7)$$

โดยที่ n คือ จำนวนของตัวอย่างในกลุ่ม R_j

2.6 การทำงานของอัลกอริทึม Gradient boosting decision trees

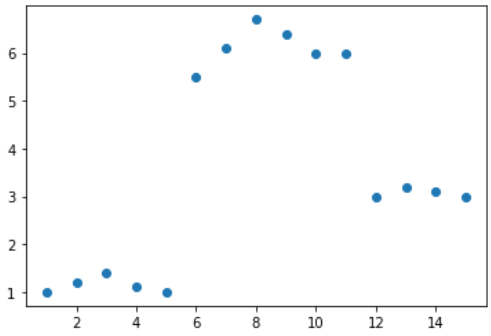
อัลกอริทึม Gradient boosting decision trees (GBDTs) คืออัลกอริทึมสำหรับการเรียนรู้ของเครื่องที่พัฒนามาจากอัลกอริทึมต้นไม้ตัดสินใจโดยเทคนิคร่วมกันตัดสินใจแบบ Boosting คือการสร้างต้นแบบเริ่มต้นด้วยต้นไม้ตัดสินใจกับชุดข้อมูล แล้วใช้ต้นแบบที่ได้คาดการณ์ชุดข้อมูลนั้น ทำการปรับปรุงต้นแบบโดยสร้างต้นแบบใหม่จากความผิดพลาดของต้นแบบก่อนหน้า ทำกระบวนการนี้ซ้ำเพื่อพัฒนาประสิทธิภาพของต้นแบบจนได้ต้นแบบสุดท้ายที่มีประสิทธิภาพมากที่สุด

เริ่มต้นด้วยการอธิบายอัลกอริทึมพื้นฐานของ GBDTs อย่างการสร้างต้นไม้ตัดสินใจแบบถดถอย ซึ่งเป็นต้นไม้ตัดสินใจที่มีไว้ใช้สำหรับปัญหาการถดถอยโดยเฉพาะ สามารถคาดการณ์ค่าที่เป็นค่าต่อเนื่อง แทนที่ค่าแบบจำนวนเต็ม โดยการสมมติชุดข้อมูลที่มี 2 ตัวแปรขึ้นมา ดังตารางที่ 2.1

ตารางที่ 2.1 ชุดข้อมูลทดลองในการสร้างต้นไม้ตัดสินใจแบบถดถอย (Prasad, 2021)

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Y	1	1.2	1.4	1.1	1	5.5	6.1	6.7	6.4	6	6	3	3.2	3.1

จากตารางที่ 2.1 เป็นชุดข้อมูลจำนวน 2 ตัวแปรโดยมีการกระจายตัวที่ไม่เป็นรูปแบบ สามารถสร้างกราฟแสดงรูปแบบการกระจายตัวของข้อมูล ดังภาพที่ 2.3

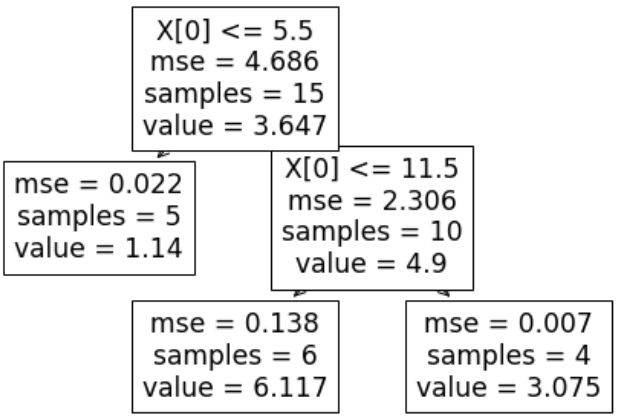


ภาพที่ 2.3 กราฟแสดงข้อมูลชุดทดลอง (Prasad, 2021)

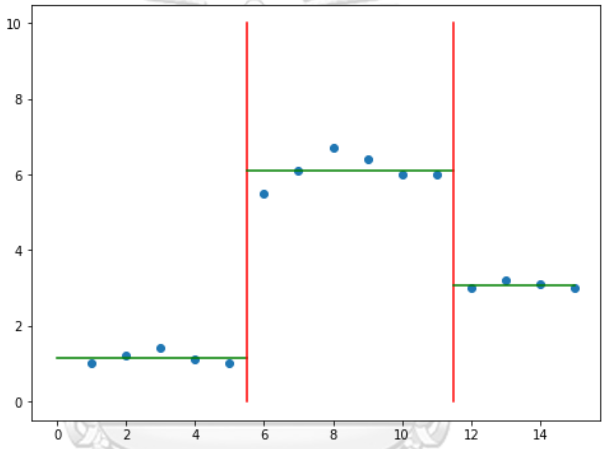
จากภาพที่ 2.3 จะนำชุดข้อมูลนี้สาธิตการสร้างต้นแบบด้วยวิธีต้นไม้ตัดสินใจที่ตัวแปร X และ Y เป็นค่าแบบต่อเนื่อง และต้นแบบจะทำการคาดการณ์ค่า Y จากค่า X โดยที่ n คือจำนวนแถวของชุดข้อมูล

- ขั้นแรกจะทำการเรียงข้อมูลโดยอ้างอิงจากค่า X (ในข้อมูลชุดนี้ได้ทำการจัดเรียงแล้ว) แล้วหาค่าเฉลี่ยจากค่า X 2 แถวแรก (ในที่นี้คือ 1.5 จากชุดข้อมูลข้างต้น) จากนั้นแบ่งชุดข้อมูลออกเป็น 2 ช่วง ได้แก่ช่วง A และช่วง B โดยเงื่อนไข $X<1.5$ และ $X\geq1.5$
- ตอนนี้ในช่วง A จะครอบคลุมข้อมูลจุดเดียว (1,1) และช่วง B จะครอบคลุมข้อมูลที่เหลือ แล้วหาค่าเฉลี่ยของค่า Y ในช่วง A และช่วง B 2 ค่าเฉลี่ยที่ได้มานี้คือค่าคาดการณ์ของต้นไม้ตัดสินใจที่ $X<1.5$ และ $X\geq1.5$ ตามลำดับทำการคำนวณค่า Loss จากค่าจริงกับค่าคาดการณ์ โดยทั่วไปแล้วต้นไม้ตัดสินใจแบบถดถอยจะใช้เมตริก MSE ในการคำนวณค่า Loss
- ขั้นต่อมาจะทำเหมือนขั้นตอนแรก แต่เปลี่ยนจากค่า X 2 แถวแรก (แถว 1 และ 2) เป็นค่า X 2 แถวถัดมา (แถว 2 และ 3) เมื่อทำตามขั้นตอนแรกจะได้ค่าเฉลี่ยของค่า X (ในที่นี้คือ 2.5) แล้วแบ่งชุดข้อมูลออกอีกครั้งโดยเงื่อนไข $X<2.5$ และ $X\geq2.5$ เป็นช่วง A และช่วง B ทำการคาดการณ์แล้วคำนวณค่า Loss ดำเนินการต่อทุกแถวจนสิ้นสุดชุดข้อมูล
- ตอนนี้เราจะได้ค่า Loss จำนวน n-1 ต่อมาทำการเลือกจุดหนึ่งจุดที่จะทำหน้าที่เป็นเงื่อนไขการกระจายตัว โดยเลือกจุดที่มีค่า Loss น้อยที่สุด ในที่นี้คือจุดที่ $X=5.5$ เพราะฉะนั้นต้นไม้จะแบ่งออกเป็น 2 ช่วงที่ $X<5.5$ และ $X\geq5.5$ นี่คือวิธีการเลือกโหนดราก และข้อมูลที่ถูกแบ่งออกเป็น 2 ช่วงก็จะกลายเป็นโหนดรากแล้วขยายตัวต่อไปด้วยวิธีการเดียวกัน

สรุปแล้วแนวแล้วของอัลกอริทึมต้นไม้ตัดสินใจแบบถดถอยคือการหาจุดที่ตัวแปรเป็นอิสระที่จะกระจายข้อมูลออกเป็น 2 ส่วน นั่นหมายถึงค่า Loss จะทำให้น้อยที่สุดในจุดนั้น จะสามารถแสดงผลลัพธ์ของต้นไม้ตัดสินใจแบบถดถอยจากข้อมูลข้างต้นดังภาพที่ 2.4 และแสดงภาพผลการคาดการณ์ของต้นไม้ตัดสินใจนี้ดังภาพที่ 2.5



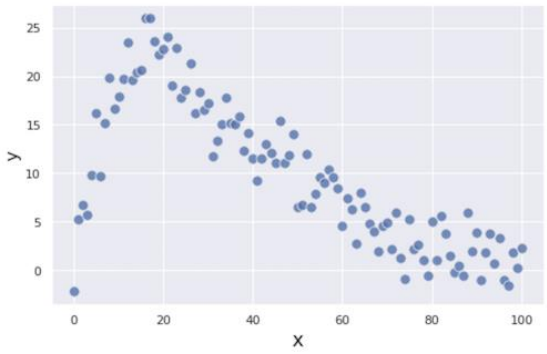
ภาพที่ 2.4 ผลลัพธ์ของต้นไม้ตัดสินใจแบบถดถอย (Prasad, 2021)



ภาพที่ 2.5 ผลลัพธ์การคาดการณ์ของต้นแบบที่สร้างโดยต้นไม้ตัดสินใจ (Prasad, 2021)

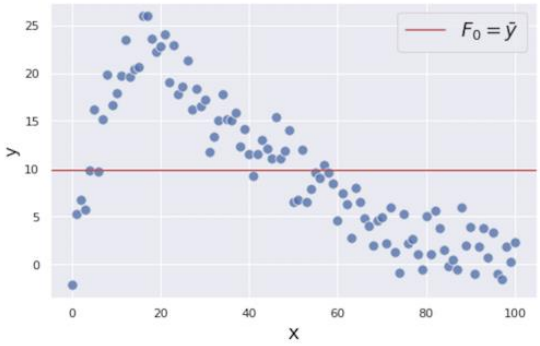
จากภาพที่ 2.4 และ 2.5 หากชุดข้อมูลนั้นมีตัวแปรอิสระ 3 ตัว ซึ่งเหมือนกับตัวแปรอิสระ X จากตารางที่ 2.1 ตัวแปรอิสระทุกตัวจะต้องทำกระบวนการเช่นเดียวกันกับตัวแปร X ข้อมูลจะถูกเรียงโดยตัวแปรอิสระทั้ง 3 ตัวแบบแยกกัน คำนวณค่า Loss จากตัวแปรทุกตัว และเลือกจุดที่มีค่า Loss น้อยที่สุดจากทั้งหมด

เมื่อทำการสร้างต้นแบบจากอัลกอริทึมต้นไม้ตัดสินใจแบบถดถอยแล้ว จึงเข้าสู่กระบวนการของ GBDTs โดยผู้วิจัยจะทำการยกตัวอย่าง อธิบายขั้นตอนของอัลกอริทึม GBDTs แบบสรุปให้เข้าใจง่ายก่อนที่จะแสดงขั้นตอนการทำงานด้วยตัวแบบคณิตศาสตร์ เริ่มจากการยกตัวอย่างชุดข้อมูลหนึ่งที่มีความสัมพันธ์แบบไม่เป็นเชิงเส้นระหว่างตัวแปร x และตัวแปร y ดังแสดงในภาพที่ 2.6



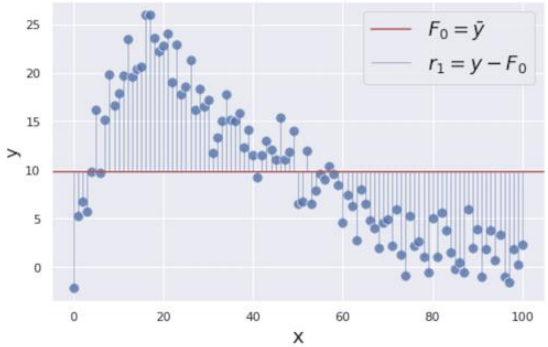
ภาพที่ 2.6 ตัวอย่างชุดข้อมูลสำหรับอัลกอริทึม GBDTs (Masui, 2022)

จากภาพที่ 2.6 ขั้นตอนแรกทำการสร้างต้นแบบเริ่มต้นในการคาดการณ์ค่า y คือ F_0 ซึ่งเป็นค่าเฉลี่ยของค่า y ทั้งหมดดังแสดงในภาพที่ 2.7



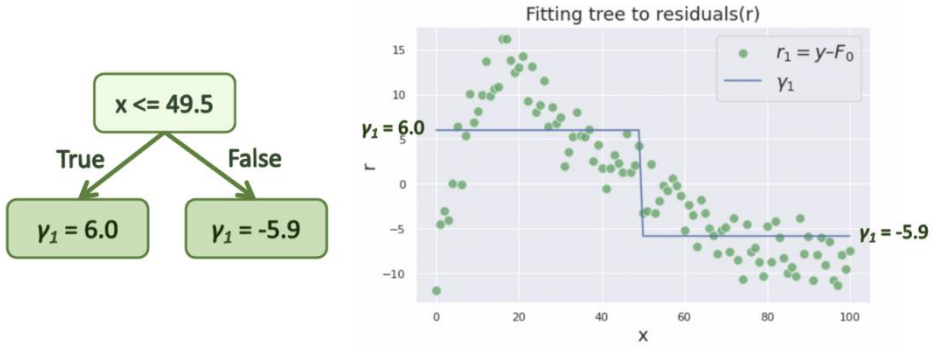
ภาพที่ 2.7 สร้างต้นแบบการคาดการณ์เริ่มต้น (Masui, 2022)

ในการพัฒนาต้นแบบการคาดการณ์ เราจะสนใจค่าที่ต้นแบบคาดการณ์ผิดพลาดจากขั้นตอนแรก เนื่องจากนั่นคือสิ่งที่เราต้องการทำให้มันลดลง เพื่อทำให้ต้นแบบสามารถคาดการณ์ได้ดีขึ้น ค่าที่ต้นแบบคาดการณ์ผิดพลาด r_1 เป็นเส้นแนวตั้งสีฟ้าดังแสดงในภาพที่ 2.8



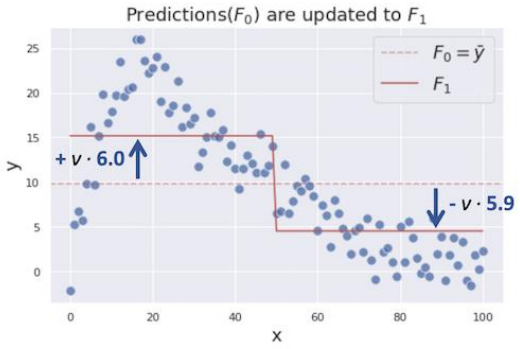
ภาพที่ 2.8 r_1 ค่าที่ต้นแบบคาดการณ์ผิดพลาดจากต้นแบบเริ่มต้น (Masui, 2022)

เราจะทำการสร้างต้นไม้ตัดสินใจแบบถดถอยเพื่อลดค่าผิดพลาดนี้ลงโดยที่ x คือ feature และ r_1 คือค่าที่คาดการณ์ ถ้าหากสามารถหารูปแบบระหว่าง x และ r_1 โดยการสร้างต้นแบบ จะสามารถใช้มันเพื่อลดค่าความผิดพลาดได้ เพื่อทำให้สาธิตได้ง่ายขึ้น เราจะสร้างต้นไม้ตัดสินใจที่เรียบง่ายโดยแต่ละต้นจะมีการกระจายตัวแค่ครั้งเดียวหรือ 2 โหนด (โดยทั่วไปแล้ว GBDTs มักมีตั้งแต่ 8 ถึง 32 โหนด) เมื่อสร้างต้นไม้ตัดสินใจแรกจะได้ต้นแบบคาดการณ์ 2 ค่าได้แก่ γ_1=(6.0, -5.9) (ใช้สัญลักษณ์แกมม่าในการแสดงค่าคาดการณ์) แสดงได้ดังภาพที่ 2.9

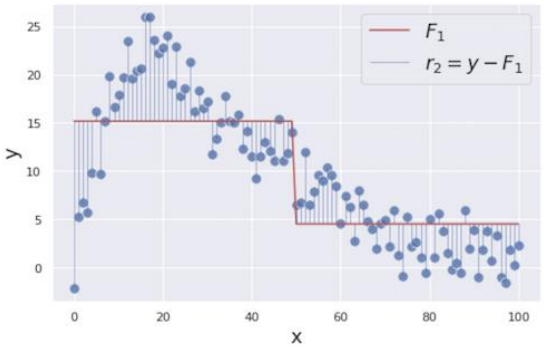


ภาพที่ 2.9 ค่าคาดการณ์จากต้นไม้ตัดสินใจแรก (Masui, 2022)

จากภาพที่ 2.9 ค่าคาดการณ์ γ_1 ถูกเพิ่มเข้าไปในค่าคาดการณ์เริ่มต้น F_0 เพื่อลดค่าความผิดพลาด ในทางการเรียนรู้ของเครื่องนั้นอัลกอริทึม GBDTs ไม่สามารถเพิ่มค่า γ เข้าไปโดยตรงได้ เพราะจะทำให้เกิดปัญหา overfitting ค่า γ จึงถูกลดขนาดลงด้วย Hyperparameter ที่ชื่อว่า learning rate ν ซึ่งมีค่าตั้งแต่ 0 ถึง 1 แล้วค่อยเพิ่มลงไปใน F ($F_1=F_0+\nu\gamma_1$) ในตัวอย่างนี้จะใช้ค่า learning rate ที่ 0.9 ทำให้สามารถเข้าใจตัวอย่างได้ง่ายขึ้น ซึ่งโดยปกติแล้ว learning rate มักจะมีค่าน้อยมาก ๆ เช่น 0.1 หลังจากทำการเพิ่มค่าคาดการณ์ลงไปจะได้ค่าคาดการณ์ใหม่ และค่าผิดพลาดใหม่ ดังแสดงในภาพที่ 2.10 และ 2.11

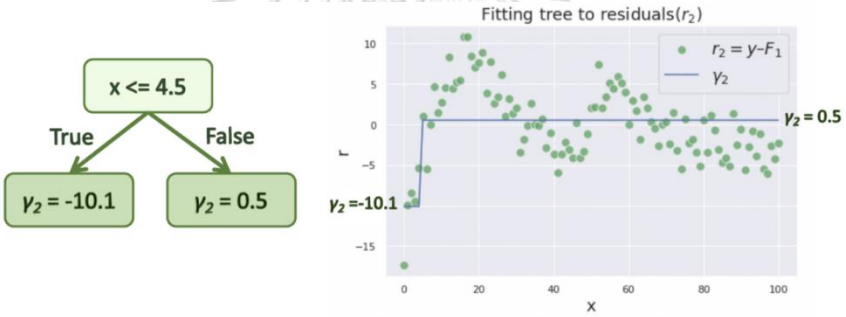


ภาพที่ 2.10 ค่าคาดการณ์ใหม่หลังจากสร้างต้นแบบต้นไม้ตัดสินใจ (Masui, 2022)



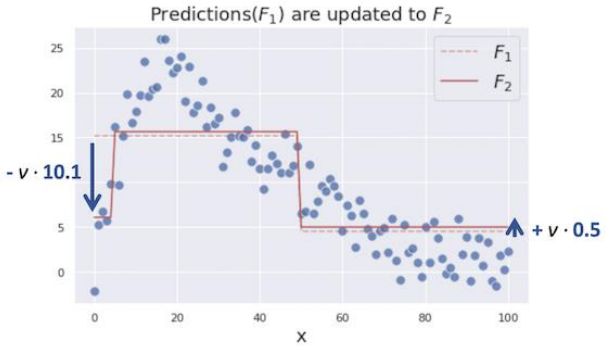
ภาพที่ 2.11 r_2 ค่าผิดพลาดใหม่หลังจากสร้างต้นแบบต้นไม้ตัดสินใจ (Masui, 2022)

ในขั้นตอนต่อไปจะทำการสร้างต้นไม้ตัดสินใจแบบถดถอยอีกครั้งโดยให้ x เป็น feature และ r_2 คือค่าที่คาดการณ์จะได้ผลลัพธ์ดังแสดงในภาพที่ 2.12



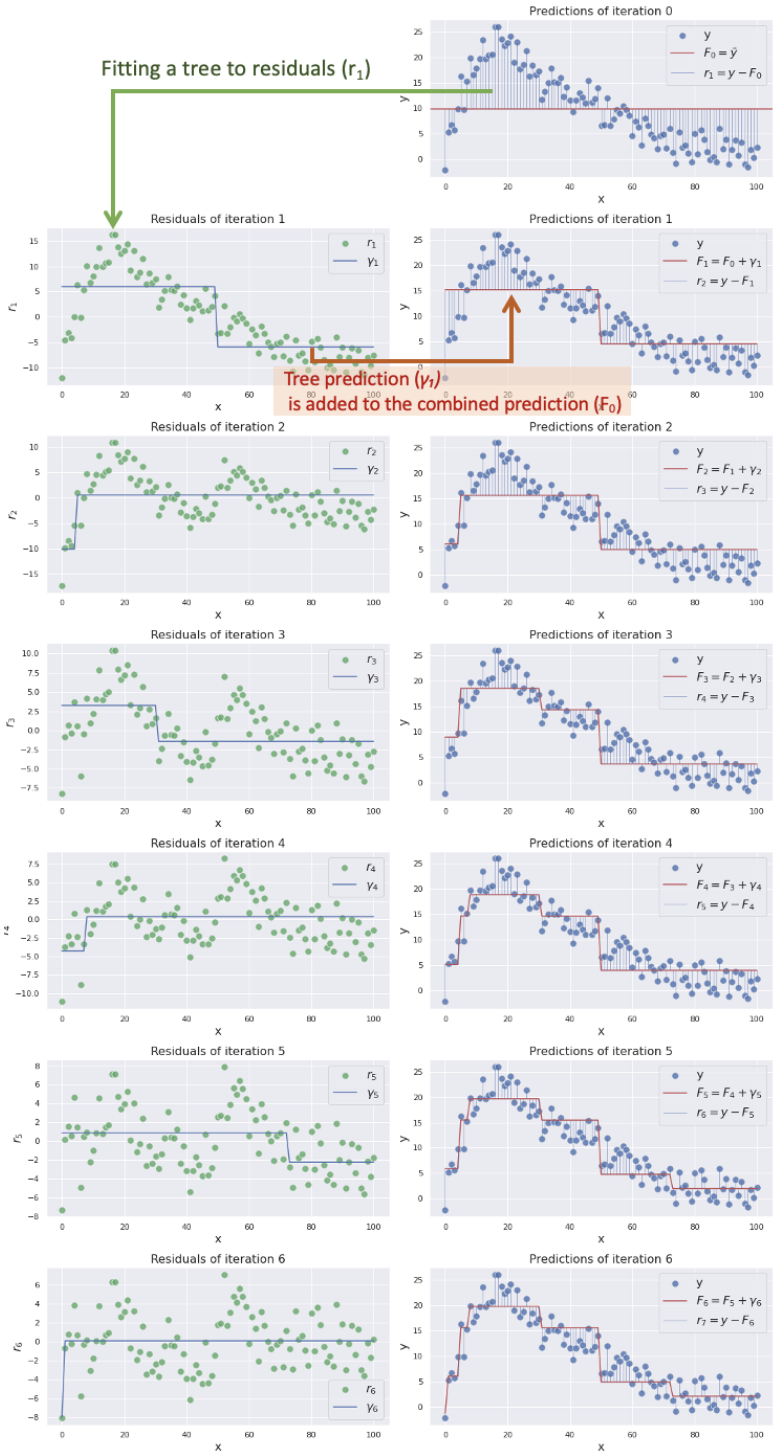
ภาพที่ 2.12 ค่าคาดการณ์จากต้นไม้ตัดสินใจที่สอง (Masui, 2022)

จากภาพที่ 2.12 เมื่อทำการเพิ่มค่าคาดการณ์ γ_2 ไปในค่าคาดการณ์แรก F_1 จะทำให้ต้นแบบเกิดการปรับปรุงจากต้นแบบก่อนหน้า ดังแสดงในภาพที่ 2.13



ภาพที่ 2.13 ค่าคาดการณ์ใหม่หลังจากสร้างต้นแบบต้นไม้ตัดสินใจครั้งที่สอง (Masui, 2022)

จากภาพที่ 2.13 เมื่อทำขั้นตอนข้างต้นซ้ำ ๆ จนต้นแบบหยุดการพัฒนา โดยขั้นตอนการเพิ่มประสิทธิภาพของข้อมูลตัวอย่างนี้จากการทำซ้ำครั้งที่ 0 ถึงครั้งที่ 6 ด้วยอัลกอริทึม GBDTs สามารถแสดงดังภาพที่ 2.14



ภาพที่ 2.14 ขั้นตอนการเพิ่มประสิทธิภาพของอัลกอริทึม GBDTs (Masui, 2022)

จากภาพที่ 2.14 จะเห็นได้ว่าค่าคาดการณ์ F นั้นจะเข้าใกล้กับค่า y มากขึ้นเรื่อย ๆ เมื่อเราให้ต้นแบบถัดมาได้เรียนรู้ข้อผิดพลาดของต้นแบบก่อนหน้า ซึ่งเป็นจุดเด่นของอัลกอริทึม GBDTs โดยสามารถสรุปกระบวนการของอัลกอริทึม GBDTs ด้วยตัวแบบคณิตศาสตร์ดังต่อไปนี้

- สร้างต้นแบบเริ่มต้นด้วยค่าคาดการณ์คงที่

$$F_0(x) = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(y_i, \gamma) \quad (2.8)$$

โดยที่ F_0 คือ ค่าคาดการณ์เริ่มต้น

L คือ ค่า Loss (สำหรับปัญหาถดถอย $L = (y_i - \gamma)^2$)

γ คือ ค่าคาดการณ์ของต้นแบบ

$\underset{\gamma}{argmin}$ คือ การค้นหาค่า γ ที่ทำให้ค่า Loss มีค่าน้อยที่สุด

- ขั้นตอนการสร้างต้นแบบโดยเรียนรู้จากความผิดพลาดของต้นแบบก่อนหน้าโดยที่ m คือจำนวนครั้งในการสร้างต้นแบบใหม่ (m=1 ถึง M)
  - คำนวณค่าความผิดพลาด (residuals)

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad for\ i = 1, \dots, n \ (2.9)$$

  - สร้างต้นแบบด้วยอัลกอริทึมต้นไม้ตัดสินใจแบบถดถอยด้วย feature x กับ r แล้วจะได้โหนดใบที่บ่งบอกถึงค่าคาดการณ์จากต้นแบบ

$$R_{jm} \quad for \quad j = 1, \dots, J_m$$

โดยที่ R คือ โหนดใบ

j คือ ลำดับของโหนดใบ

m คือ ลำดับของต้นไม้ตัดสินใจ (จำนวนครั้งที่พัฒนา)

J คือ จำนวนโหนดใบ

  - ขั้นตอนการคำนวณหาค่าคาดการณ์ของต้นแบบที่ทำให้เกิดค่า Loss น้อยที่สุดเพื่อนำมาปรับปรุงต้นแบบต่อไป

$$\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \epsilon R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \ for\ j = 1, \dots, J_m \quad (2.10)$$

โดยที่ γ_{jm} คือ ค่าคาดการณ์ต้นแบบที่ทำให้เกิดค่าผิดพลาดน้อยที่สุด

- การสร้างต้นแบบใหม่เพื่อพัฒนาประสิทธิภาพของต้นแบบ

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} 1(x \epsilon R_{jm}) \quad (2.11)$$

โดยที่ ν *คือ Hyperparameter learning rate*

จากตัวแบบคณิตศาสตร์ข้างต้นแสดงให้เห็นถึงความยืดหยุ่น และสะดวกสบายของอัลกอริทึม GBDTs ซึ่งสามารถรองรับปัญหาได้หลายแบบเช่น หากปัญหานั้นต้องการลดค่า MAE แทนค่า MSE ที่แสดงในตัวอย่างก่อนหน้า ก็สามารถแทนที่ Loss function ด้วยสูตรของเมตริก MAE จึงเป็นสาเหตุหนึ่งที่ทำให้อัลกอริทึม GBDTs เป็นที่นิยม

2.7 Hyperparameters ของอัลกอริทึม LightGBM, XGBoost และ CatBoost

Hyperparameters คือ พารามิเตอร์ที่ผู้ใช้งานสามารถกำหนดก่อนสร้างต้นแบบเช่น Learning rate ซึ่งการปรับปรุงค่า Hyperparameters จะส่งผลไปยังความแม่นยำของต้นแบบ ความรวดเร็วในการสร้างต้นแบบ รวมไปถึงการควบคุมปัญหา Overfitting

สำหรับอัลกอริทึมวิธีการเรียนรู้ของเครื่องที่ใช้ต้นไม้ตัดสินใจเป็นพื้นฐานไม่ว่าจะเป็น Random forest, LightGBM, XGBoost หรือ CatBoost ก็มักจะพบปัญหา Overfitting แม้ว่าเทคนิคการร่วมกันตัดสินใจจะทำการคาดการณ์จากต้นแบบหลาย ๆ ต้นแบบ แต่ก็ยังมีความเป็นไปได้ที่จะเกิดปัญหา Overfitting ซึ่งนอกจากการแบ่งชุดข้อมูลด้วยวิธี K-fold cross validation แล้วยังมีการเปรียบเทียบกันระหว่างอัลกอริทึม และการปรับ Hyperparameters ที่จะสามารถช่วยลดปัญหา Overfitting ได้

โดยการควบคุมความซับซ้อนของต้นแบบของอัลกอริทึม XGBoost จะใช้ Hyperparameters max_depth (สำหรับการต่อตัวของต้นไม้ตัดสินใจแบบ Level-wise) และอัลกอริทึม LightGBM จะใช้ Hyperparameters num_leaves (สำหรับการต่อตัวของต้นไม้ตัดสินใจแบบ Leaf-wise) (Kay Jan Wong, 2022) สามารถแสดง Hyperparameters ที่สำคัญในการสร้างต้นแบบ ตามประโยชน์การใช้งานดังตารางที่ 2.2

ตารางที่ 2.2 Hyperparameters สำคัญของอัลกอริทึม LightGBM, XGBoost และ CatBoost

	XGBoost	LightGBM	CatBoost
Hyperparameters สำหรับการปรับแต่ง	n_estimators : จำนวนของต้นไม้ max_depth : ความลึกของต้นไม้ min_child_weight : ควบคุมความลึกต้นไม้	num_leaves : ควรมีค่าน้อยกว่า 2^{max_depth} min_data_in_leaf : ควบคุมความลึกต้นไม้ max_depth : ความลึกของต้นไม้	iterations : จำนวนของต้นไม้ depth : ความลึกของต้นไม้ min_data_in_leaf : ควบคุมความลึกต้นไม้
Hyperparameters สำหรับพัฒนาความแม่นยำ		max_bin : จำนวน feature ที่มากที่สุดที่จะถูกบรรจุใน num_leaves	
Hyperparameters สำหรับพัฒนาความเร็วในการสร้างต้นแบบ	colsample_bytree : ร้อยละของจำนวน feature ที่ใช้ในต้นไม้ subsample : ร้อยละของจำนวนข้อมูลที่ใช้ในต้นไม้ n_estimators	feature_fraction : ร้อยละของจำนวน feature ที่ใช้ในต้นไม้ bagging_fraction : ร้อยละของจำนวนข้อมูลที่ใช้ในต้นไม้ bagging_freq : ความถี่ในการเปลี่ยนตัวอย่างข้อมูลต้นไม้ max_bin	rsm : ร้อยละของจำนวน feature ที่ใช้ในต้นไม้ subsample : ร้อยละของจำนวนข้อมูลที่ใช้ในต้นไม้ iterations sampling_frequency: ความถี่ของน้ำหนักและสิ่งของของข้อมูลตัวอย่างเมื่อสร้างต้นไม้

ตารางที่ 2.2 Hyperparameters สำคัญของอัลกอริทึม LightGBM, XGBoost และ CatBoost (ต่อ)

	XGBoost	LightGBM	CatBoost
Hyperparameters สำหรับควบคุมปัญหา Overfitting	learning_rate gamma : พารามิเตอร์สำหรับการregularization max_depth min_child_weight subsample	max_bin num_leaves max_depth bagging_freq feature_fraction lambda_l1/lambda_l2 /min_gain_to_split : พารามิเตอร์สำหรับการregularization	early_stopping_rounds : หยุดสร้างต้นแบบหลังจากจำนวนที่ใส่เข้าไป od_type : เครื่องมือช่วยตรวจสอบ overfitting learning_rate depth l2_leaf_reg : พารามิเตอร์สำหรับการregularization

จากตารางที่ 2.2 ค่า Hyperparameters ของทั้งอัลกอริทึมทั้ง 3 ตัวจะมีความคล้ายคลึงกัน (เพียงแค่ระบุชื่อตัวแปรแตกต่างกัน) เนื่องจากเป็นอัลกอริทึมที่มีพื้นฐานมาจาก GBDTs เช่นเดียวกัน สำหรับ Hyperparameters ที่แต่ละอัลกอริทึมมีเฉพาะตัวนั้นเกิดจากการปรับปรุงรูปแบบการสร้างต้นแบบของแต่ละผู้พัฒนาซึ่งส่งผลให้อัลกอริทึมมีความแตกต่างกันในบางส่วน แต่จุดประสงค์ในการปรับปรุงล้วนเป็นการพัฒนาประสิทธิภาพให้อัลกอริทึมสามารถทำงานได้ดียิ่งขึ้น

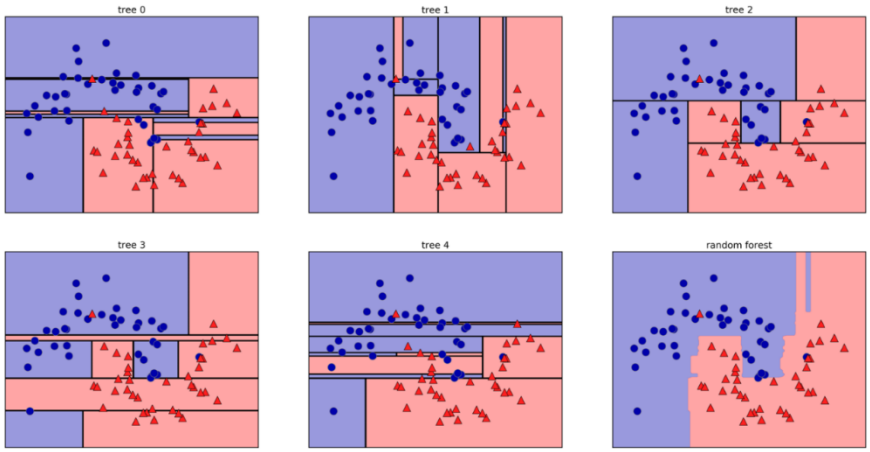
2.8 การทำงานของอัลกอริทึม Random forest

อัลกอริทึม Random forest คืออัลกอริทึมสำหรับการเรียนรู้ของเครื่องที่พัฒนามาจากอัลกอริทึมต้นไม้ตัดสินใจโดยเทคนิคร่วมกันตัดสินใจแบบ Bagging คือการรวมการคาดการณ์จากหลาย ๆ ต้นแบบเข้าด้วยกันเพื่อทำให้มีความแม่นยำมากยิ่งขึ้น สำหรับอัลกอริทึม Random forest คือการนำต้นไม้ตัดสินใจแบบถดถอยหลายต้นมารวมกัน เป็นอัลกอริทึมที่นิยมใช้ในการแข่งขัน หรือการศึกษาเช่นเดียวกับพวกอัลกอริทึม GBDTs

ขั้นตอนของอัลกอริทึม Random forest เริ่มจากการที่เรามีชุดข้อมูล D แล้วต้องการสร้างต้นไม้ตัดสินใจ K ต้นเพื่อใช้เทคนิค Bagging จะทำการสร้างต้นไม้ตัดสินใจจนกว่าจะมีจำนวน N ตัวอย่างในแต่ละโหนด (ปกติแล้วในปัญหาถดถอยนี้ N จะมีค่าเท่ากับ 5) ให้ F เป็นจำนวนของ Feature ซึ่งจะถูกสุ่มเลือกขึ้นมาในแต่ละโหนดของต้นไม้ตัดสินใจ Feature ที่สุ่มมานี้จะเป็นเงื่อนไขในการกระจายตัวของโหนด ซึ่งนี่เป็นจุดเด่นของอัลกอริทึม Random forest ทำให้นอกจากจะมี

ต้นไม้ตัดสินใจหลายต้นแล้ว ยังทำให้ Feature ของแต่ละต้นนั้นไม่เหมือนกันอีกด้วย ส่งผลให้ต้นไม้ตัดสินใจแต่ละต้นมีความหลากหลายและอิสระต่อกันมากขึ้น

จากนั้นทำการสุ่มสร้างสับเซ็ตของชุดข้อมูล D เป็นจำนวน K สับเซ็ต (หากมีตัวอย่างที่ไม่ปรากฏอยู่ในสับเซ็ตเหล่านี้จะถูกเรียกว่า out of bag) แล้วทำการสร้างต้นไม้ตัดสินใจขึ้นมาจากสับเซ็ตหนึ่งแล้วทำซ้ำจนได้ต้นไม้ตัดสินใจ K ต้น แล้วใช้สับเซ็ตอื่นในการวัดประสิทธิภาพของต้นแบบจากต้นไม้ตัดสินใจแต่ละต้น ผลลัพธ์สุดท้ายคือการเฉลี่ยการคาดการณ์ของต้นไม้ตัดสินใจทั้งหมด ตัวอย่างของการสร้างต้นแบบด้วยอัลกอริทึม Random forest ดังแสดงในภาพที่ 2.15



ภาพที่ 2.15 ตัวอย่างการสร้างต้นแบบด้วยอัลกอริทึม Random forest

จากภาพที่ 2.15 ต้นแบบสุดท้ายจะเกิดจากต้นแบบหลาย ๆ ตัวเฉลี่ยกัน โดยอัลกอริทึม Random forest มีข้อดีคือสามารถลดปัญหา Overfitting โดยการเพิ่มจำนวนของต้นไม้ตัดสินใจ ไม่อ่อนไหวต่อค่าผิดปกติ แต่การสร้างต้นไม้ตัดสินใจจำนวนมาก ๆ พร้อมกันจำเป็นต้องใช้หน่วยความจำของคอมพิวเตอร์ที่สูงตามจำนวนของต้นไม้จึงเป็นข้อเสียของอัลกอริทึม Random forest

สำหรับ Hyperparameters ของอัลกอริทึม Random forest บางตัวจะมีความคล้ายคลึงกับ Hyperparameters ของอัลกอริทึม LightGBM, XGBoost และ CatBoost เนื่องจากพื้นฐานของอัลกอริทึมนั้นมาจากต้นไม้ตัดสินใจแบบถดถอย สามารถแสดง Hyperparameters ที่สำคัญของอัลกอริทึม Random forest ได้ดังตารางที่ 2.3

ตารางที่ 2.3 Hyperparameters สำคัญของอัลกอริทึม Random forest

การใช้งาน	Hyperparameters	คำอธิบาย
พัฒนาความแม่นยำในการคาดการณ์	n_estimators	จำนวนของต้นไม้ตัดสินใจ
	max_features	จำนวน feature สูงสุดที่จำกัดในการกระจายโหนด
	mini_sample_leaf	จำนวนขั้นต่ำที่ต้องกระจายโหนดใบ
พัฒนาความเร็วในการสร้างต้นแบบ	n_jobs	จำนวนที่ใช้หน่วยประมวลผลคอมพิวเตอร์ที่ใช้งาน
	random_state	ควบคุมการสุ่มของตัวอย่าง ถ้าเป็นจำนวน definite ต้นแบบจะให้ผลลัพธ์ที่เหมือนเดิมตลอด
	oob_score	out of bag เป็นการทำ cross validation ของอัลกอริทึม Random forest ช่วยแบ่งข้อมูลออก 1 ใน 3 ไม่ใช้ในการสร้างต้นแบบ แต่ใช้ข้อมูลนี้ในการทดสอบประสิทธิภาพลดปัญหา Overfitting

จากตารางที่ 2.3 เนื่องจากอัลกอริทึม Random forest นั้นถูกออกแบบมาเพื่อจัดการกับปัญหา Overfitting นอกจากการทำ K-fold cross validation แล้ว จึงไม่จำเป็นต้องมุ่งเน้นการปรับ Hyperparameters เพื่อจัดการกับปัญหา Overfitting มากนัก

2.9 การปรับปรุง Hyperparameters ด้วยวิธี RandomizedSearchCV

การปรับปรุง Hyperparameters นั้นทำเพื่อพัฒนาประสิทธิภาพของต้นแบบ โดยการเปลี่ยนแปลง Hyperparameters ของต้นแบบแล้วทำการสร้างต้นแบบขึ้นใหม่จนกว่าจะได้ต้นแบบที่มีประสิทธิภาพแบบที่เราต้องการ การปรับปรุง Hyperparameters สามารถทำได้หลายวิธี แต่ในงานวิจัยนี้จะใช้เครื่องมือที่ชื่อว่า RandomizedSearchCV ซึ่งเครื่องมือนี้เป็นเครื่องมือสำเร็จรูปจากไลบารี่ scikit-learn โดยความสามารถของเครื่องมือนี้คือการ Random search hyperparameter มีประโยชน์เมื่อมีพารามิเตอร์ที่ต้องการทดลองเป็นจำนวนมาก และขอบเขตของพารามิเตอร์นั้นค่อนข้างกว้าง จะประหยัดเวลามากกว่าการทำ Brute force ด้วยชุดข้อมูลที่มีขนาดใหญ่

RandomizedSearchCV จะทำการสุ่มค่า Hyperparameters แล้วคำนวณคะแนนออกมา โดยคะแนนในที่นี้คือค่าเมตริกที่เราต้องการพัฒนา โดยเราสามารถตั้งค่าพารามิเตอร์ของเครื่องมือนี้เพื่อวัตถุประสงค์ต่าง ๆ ได้ดังนี้

- estimator คือค่าเมตริกที่เราต้องการให้เครื่องมือนี้พัฒนา
- cv คือค่าที่เป็นเลขจำนวนเต็มสำหรับการทำ cross validation (ปกติเป็น 5)

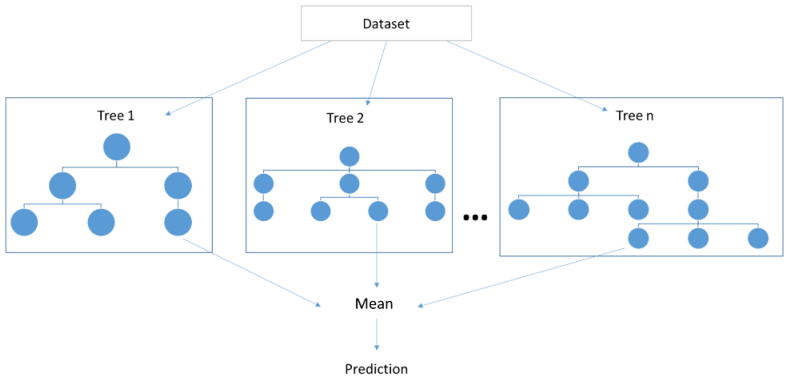
- n_iter คือจำนวนที่ต้องการค้นหา Hyperparameters ใหม่
- n_jobs คือจำนวนที่ใช้หน่วยประมวลผลคอมพิวเตอร์ที่ใช้งาน
- param_distributions คือขอบเขตของค่า Hyperparameters ในกรณีที่เราต้องการกำหนดขอบเขตในการใช้เครื่องมือ RandomizedSearchCV

RandomizedSearchCV เป็นเครื่องมือที่สะดวก และใช้งานง่ายเหมาะสำหรับผู้ที่เริ่มต้นสร้างต้นแบบ เครื่องมือนี้แม้จะมีความรวดเร็วในการใช้งานแต่ไม่ได้ยืนยันว่าเราจะได้รับ Hyperparameters ที่ดีที่สุดเนื่องจากไม่ได้มีการทดสอบทุกความเป็นไปได้ เครื่องมือนี้จึงนิยมใช้กับชุดข้อมูลที่มีขนาดใหญ่ และ Hyperparameters มีจำนวนมาก

2.10 งานวิจัยที่เกี่ยวข้อง

2.10.1 Random Forest Algorithm

Random Forest (Breiman, 2001) เป็นหนึ่งในอัลกอริทึมที่พัฒนามาจากต้นไม้ตัดสินใจ (Safavian & Landgrebe, 1991) ด้วยเทคนิคการใช้วิธีร่วมกันตัดสินใจ (Ensemble) เป็นการนำแนวคิดทางสถิติที่เมื่อมีการเก็บข้อมูลจากจำนวนตัวอย่างประชากรมากขึ้นจะยิ่งทำให้สถิติเข้าใกล้ความเป็นจริงมากขึ้นมาใช้สร้างต้นแบบวิธีการเรียนรู้ของเครื่อง หนึ่งในเทคนิคการใช้วิธีร่วมกันตัดสินใจที่นำมาใช้ในต้นแบบนี้ คือ Bagging (Bootstrap Aggregation) โดยการสร้างต้นแบบจากต้นไม้ตัดสินใจหลาย ๆ ต้นแบบแต่ละต้นแบบจะได้รับข้อมูลที่ไม่เหมือนกัน ประสิทธิภาพของต้นแบบต้นไม้ตัดสินใจแต่ละต้นอาจจะน้อยมาก ๆ แต่เมื่อนำต้นไม้ตัดสินใจหลาย ๆ ต้นแบบมาทำการคาดการณ์ร่วมกันก็จะได้ต้นแบบที่มีประสิทธิภาพสูง ตัวอย่างการสร้างต้นแบบ Random Forest ดังแสดงในภาพที่ 2.16

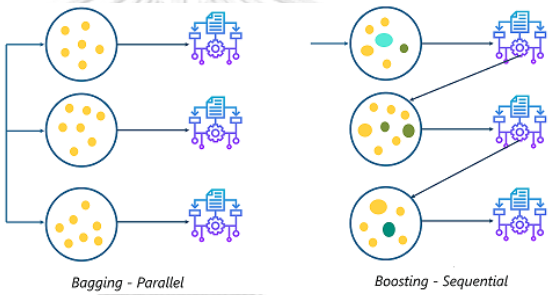


ภาพที่ 2.16 ตัวอย่างการสร้างต้นแบบ Random Forest (Kumar, 2018)

ข้อดีของต้นแบบ Random Forest คือ สามารถใช้ได้ในกับปัญหาการจัดหมวดหมู่และการถดถอย ใช้ได้กับข้อมูลที่มีลักษณะเป็นตารางหรือรูปภาพก็ได้ ข้อมูลที่ใช้ในการสร้างต้นแบบไม่จำเป็นต้องมีการกระจายตัวแบบปกติ

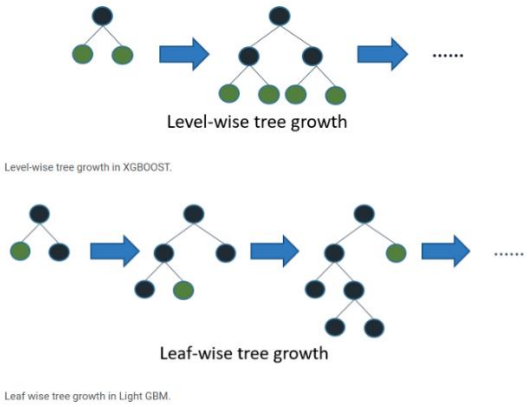
2.10.2 Light Gradient Boosting Machine Algorithm

LightGBM (Light Gradient Boosting Machine) (Ke et al., 2017) เป็นหนึ่งในอัลกอริทึมที่พัฒนามาจากต้นไม้ตัดสินใจ ด้วยเทคนิคการใช้วิธีร่วมกันตัดสินใจอีกเทคนิค คือ Boosting เป็นการนำต้นแบบต้นไม้ตัดสินใจที่มีประสิทธิภาพต่ำมาต่อกันโดยที่ต้นแบบต้นไม้ตัดสินใจที่มาต่อก็จะช่วยแก้ไขค่าผิดพลาดของต้นแบบก่อนหน้าจนได้ต้นแบบที่ดีที่สุด เรียกอัลกอริทึมที่มีลักษณะการทำงานแบบนี้ว่า Gradient boosting decision trees (GBDTs) ความแตกต่างระหว่าง Bagging และ Boosting ดังแสดงในภาพที่ 2.17



ภาพที่ 2.17 ความแตกต่างระหว่างเทคนิค Bagging และ Boosting (Lateef, 2019)

สิ่งที่ทำให้ LightGBM มีประสิทธิภาพในการคำนวณที่รวดเร็วกว่าต้นแบบที่ใช้เทคนิค Boosting อื่น ๆ คือ การต่อตัวของต้นแบบต้นไม้ตัดสินใจ ที่มีลักษณะในแนวตั้งในขณะที่ต้นแบบอื่นมีลักษณะในแนวนอน ดังแสดงในภาพที่ 2.18



ภาพที่ 2.18 เปรียบเทียบลักษณะการทำงานของ XGBoost กับ LightGBM (Khandelwal, 2017)

ข้อดีของต้นแบบ LightGBM คือ สามารถสร้างต้นแบบที่มีการใช้ข้อมูลขนาดใหญ่อย่างรวดเร็วและใช้ทรัพยากรหน่วยความจำที่น้อย มีการพัฒนาให้ใช้งานได้อย่างสะดวกโดยมีวัตถุประสงค์เพื่อให้ต้นแบบที่สร้างขึ้นมามีประสิทธิภาพสูงสุด

2.10.3 Extreme Gradient Boosting Algorithm

XGBoost (Extreme Gradient Boosting) (Chen & Guestrin, 2016) เป็นอัลกอริทึมที่คล้ายคลึงกับ LightGBM พัฒนามาจากต้นไม้ตัดสินใจ ด้วยเทคนิคการใช้วิธีร่วมกันตัดสินใจแบบ Boosting หรือเรียกว่า GBDTs เป็นอัลกอริทึมที่พัฒนาขึ้นมาเพื่อรองรับกับชุดข้อมูลขนาดใหญ่ โดยใช้ทรัพยากรหน่วยความจำอย่างมีประสิทธิภาพเช่นเดียวกัน สิ่งที่แตกต่างกันกับ LightGBM อย่างชัดเจน คือ การต่อตัวของต้นไม้ตัดสินใจที่เป็นแบบ Level-wise tree growth ซึ่งทำให้การสร้างต้นแบบนั้นใช้เวลามากกว่า แต่ในด้านประสิทธิภาพยังคงมีความแตกต่างกันอย่างเล็กน้อยในแต่ละชุดข้อมูล หากต้องการสร้างต้นแบบที่มีค่าความผิดพลาดน้อยแล้ว XGBoost และ LightGBM เป็น 2 อัลกอริทึมที่เหมาะสมกับการทดสอบเป็นอย่างมาก

2.10.4 Category Boosting Algorithm

CatBoost (Category Boosting model) (Dorogush et al., 2018) เป็นอัลกอริทึม GBDTs เช่นเดียวกับ LightGBM และ XGBoost แต่แบ่งแยกตัวเองจากสองอัลกอริทึมนี้อย่างชัดเจนโดยการมุ่งเน้นไปในด้านของการเพิ่มประสิทธิภาพต้นไม้ตัดสินใจสำหรับตัวแปรที่เป็นหมวดหมู่ที่ซึ่งแต่ละตัวแปรอาจจะไม่มีความสัมพันธ์กันในทางตัวเลข ยกตัวอย่างเช่นการจำแนกส้ม และแอปเปิล สำหรับ XGBoost จะทำการแบ่งตัวแปรออกเป็นสองตัวแปรเพื่อบ่งบอกว่านี่คือส้ม และนี่คือแอปเปิล แต่สำหรับ CatBoost จะสามารถบ่งบอกความแตกต่างของตัวแปรที่เป็นหมวดหมู่โดยอัตโนมัติไม่จำเป็นต้องทำกระบวนการใด ๆ ก่อน สำหรับ LightGBM มีการสนับสนุนตัวแปรที่เป็นหมวดหมู่เช่นกันแต่มีข้อจำกัดมากกว่า CatBoost

2.10.5 OSMnx

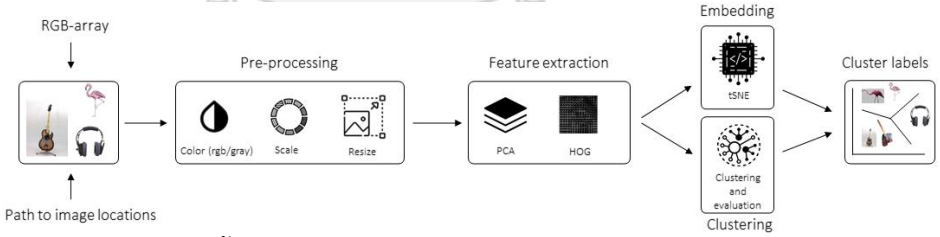
OSMnx (Open Street Map network) (Boeing, 2017) คือ เครื่องมือสำหรับวิเคราะห์โครงข่ายถนน โดยการนำข้อมูลแผนที่จาก Open Street Map ซึ่งเป็นโครงการความร่วมมือเพื่อสร้างแผนที่ฟรี ทำให้สามารถใช้ข้อมูลได้โดยไม่เสียค่าใช้จ่าย นำข้อมูลโครงข่ายถนนจากแผนที่มาสร้างเป็นรูปแบบกราฟ เพื่อวิเคราะห์ระยะทางและแสดงเส้นทางระหว่างจุดสองจุด ดังแสดงในภาพที่ 2.19

ภาพที่ 2.19 เส้นทางระหว่างจุดสองจุดบนแผนที่ในรูปแบบกราฟ

เนื่องจากข้อมูลโครงข่ายถนนที่ได้จาก Open Street Map นั้นอาจมีสภาพที่ไม่ตรงกับความเป็นจริงทั้งหมด และกรุงเทพมหานครนั้นมีข้อบังคับจราจรบนถนนแต่ละเส้น เช่น การห้ามเลี้ยวขวา การห้ามกลับรถ ส่งผลให้ระยะทางที่ได้จากกราฟนั้นมีค่าน้อยกว่าความเป็นจริง (วรพร ปุณยกนก, 2562)

2.10.6 Clustimage

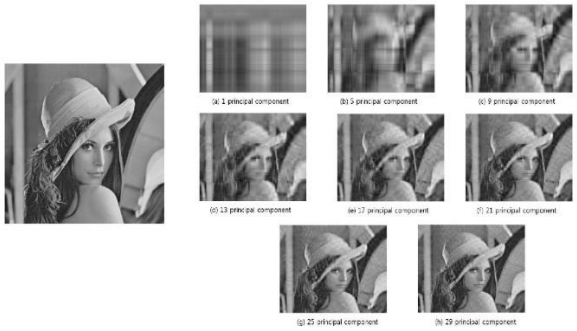
Library clustimage คือ เครื่องมือสำหรับการจัดกลุ่มรูปภาพด้วยวิธีการเรียนรู้ของเครื่อง โดยมีเป้าหมายในการตรวจจับธรรมชาติของกลุ่มของรูปภาพ แบบไม่จำเป็นต้องมีคำตอบให้ผู้เรียนรู้ มีขั้นตอนในการดำเนินการดังแสดงในภาพที่ 2.20



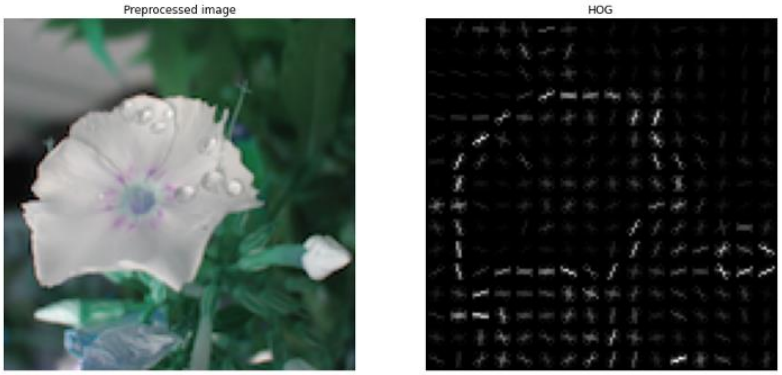
ภาพที่ 2.20 ขั้นตอนการดำเนินการของ Clustimage (Taskesen, 2021)

จากภาพที่ 2.20 ขั้นตอนแรกเป็นการ Pre-processing แปลงรูปภาพให้เป็นระดับสีเทา ปรับค่าพิกเซลระหว่างค่า [0, 255] และปรับขนาดของทุกรูปภาพให้เท่ากัน หลังจากนั้นจึงทำการ Feature extraction จากรูปภาพโดยใช้ข้อมูลของพิกเซลใน Library นี้ จะมีให้เลือกใช้ 2 วิธีได้แก่ Principal component analysis (PCA) เป็นการลดมิติของรูปภาพลงโดยดึงส่วนที่ค่าพิกเซลมีความแปรปรวนอย่างเห็นได้ชัดออกมา และ Histogram

of oriented gradients (HOG) เป็นวิธีที่สามารถแยกส่วนของรูปจากทิศทาง และขอบมุม ตัวอย่างของรูปภาพที่ใช้วิธี PCA และ HOG ดังแสดงในภาพที่ 2.21 และ 2.22



ภาพที่ 2.21 ตัวอย่างรูปภาพที่ใช้วิธี PCA (Burns, 2019)



ภาพที่ 2.22 ตัวอย่างรูปภาพที่ใช้วิธี HOG (Taskesen, 2021)

จากภาพที่ 2.21 และ 2.22 จะเห็นได้ว่า HOG สามารถงานได้ดีเมื่อเป็นรูปภาพที่มีรูปทรงชัดเจนจากขอบ แต่ PCA จะสามารถแยกจุดเด่นภายในได้ชัดเจนมากกว่าเช่น ใบหน้า หลังจาก Feature extraction เสร็จแล้ว จึงทำการแยกกลุ่ม และประเมินผล

2.11 สรุป

หัวข้อนี้จะกล่าวถึงวิธีการนำเครื่องมือในบทที่ 2 มาใช้ในงานวิจัยโดยเริ่มจากการใช้เครื่องมือในการแปลงข้อมูลให้เป็นตัวแปรต่าง ๆ ด้วยการใช้ OSMnx ในการค้นหาระยะทางระหว่างจุด เพื่อสร้างตัวแปรระยะทางระหว่างจุด และ การใช้ Clustimage ในการจัดกลุ่มรูปกราฟจากข้อมูลดัชนีรถติดออกเป็นกลุ่ม เพื่อสร้างตัวแปรกลุ่มของรูปแบบดัชนีรถติดที่คล้ายกัน เมื่อนำตัวแปรทั้งหมดที่ได้จากการแปลงข้อมูลมาแล้วรวบรวม นำข้อมูลทั้งหมดตรวจสอบด้วยวิธีการทางสถิติต่าง ๆ เพื่อแก้ไข

ปัญหาค่าผิดปกติ หรือปัญหาการกระจายตัว นำข้อมูลสุดท้ายมาแบ่งส่วนด้วยวิธี K-fold cross validation เพื่อลดการเกิดปัญหา Overfitting นำข้อมูลที่แบ่งส่วน train set มาสร้างต้นแบบ ด้วยอัลกอริทึมวิธีการเรียนรู้ของเครื่อง Random forest, LightGBM, XGBoost และ CatBoost แล้วทำการทดสอบประสิทธิภาพกับข้อมูลที่แบ่งส่วน test set ซึ่งเป็นข้อมูลที่ผู้เรียนรู้ไม่เคยเห็นมาก่อน โดยเมื่อใช้วิธี K-fold cross validation ข้อมูลจะถูกแบ่งเป็น 5 เซ็ต เมื่อทำการสร้างต้นแบบ และทดสอบประสิทธิภาพทั้ง 5 เซ็ตแล้ว จึงนำผลลัพธ์ทั้งหมดมาหาค่าเฉลี่ยเป็นประสิทธิภาพของต้นแบบสุดท้าย ทำการปรับปรุง Hyperparameters ด้วยการสร้างต้นแบบแล้วทดสอบประสิทธิภาพซ้ำ ๆ จึงได้ผลลัพธ์สุดท้ายของการทดลอง

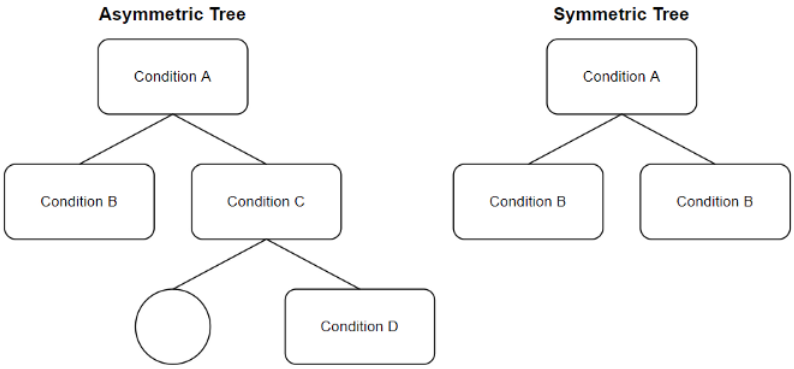
แม้ว่าในปัจจุบัน สำหรับการแข่งขันการเขียนโปรแกรมเพื่อคาดการณ์ผลลัพธ์ การใช้โครงข่ายประสาทเทียม (Neural network) จะเป็นที่นิยม แต่ต้นแบบที่สร้างโดยอัลกอริทึมประเภท GBDTs นั้นยังคงมีบทบาทสำคัญกับข้อมูลที่มีจำกัด มีความโดดเด่นในด้านประสิทธิภาพ ใช้เวลาในการสร้างต้นแบบไม่มาก และไม่ต้องอาศัยความเชี่ยวชาญที่เฉพาะเจาะจงในการปรับปรุง Hyperparameters

สำหรับอัลกอริทึม Random forest จะเป็นอัลกอริทึมตัวเดียวที่ใช้เทคนิคร่วมกันตัดสินใจแบบ Bagging ส่วนอัลกอริทึม LightGBM, XGBoost และ CatBoost เป็นอัลกอริทึมที่ใช้เทคนิคร่วมกันตัดสินใจแบบ Boosting และมีพื้นฐานมาจากอัลกอริทึมต้นไม้ตัดสินใจ สามารถแสดงความแตกต่างของอัลกอริทึมแบบ Boosting ทั้ง 3 อัลกอริทึมได้ดังตารางที่ 2.4

ตารางที่ 2.4 ความแตกต่างของอัลกอริทึม LightGBM, XGBoost และ CatBoost

	XGBoost	LightGBM	CatBoost
ผู้พัฒนา	DMLC	Microsoft	Yandex
ปี	2014	2016	2017
ลักษณะความสมมาตรของต้นไม้ตัดสินใจ	ไม่สมมาตร Level-wise tree growth	ไม่สมมาตร Leaf-wise tree growth	สมมาตร
วิธีการกระจายตัวของต้นไม้ตัดสินใจ	Pre-sorted and histogram-based	Gradient-based One-Side Sampling	Greedy method
ลักษณะ Boosting	-	-	Ordered

จากตารางที่ 2.4 ความแตกต่างประการแรกในด้านลักษณะความสมมาตรของต้นไม้ตัดสินใจ สำหรับ CatBoost ลักษณะต้นไม้สมมาตร หรือต้นไม้สมดุลหมายถึงเงื่อนไขการกระจายตัวมีความสม่ำเสมอในทุกโหนดในระดับความลึกเดียวกันของต้นไม้ตัดสินใจ ในส่วนของ LightGBM และ XGBoost เงื่อนไขการกระจายตัวนั้นอาจจะแตกต่างกันในแต่ละโหนด ที่ระดับความลึกของต้นไม้เดียวกัน ตัวอย่างลักษณะความสมมาตรของต้นไม้ตัดสินใจดังแสดงในภาพที่ 2.23



ภาพที่ 2.23 ตัวอย่างลักษณะความสมมาตรของต้นไม้ตัดสินใจ (Kay Jan Wong, 2022)

จากภาพที่ 2.23 สำหรับลักษณะต้นไม้แบบสมมาตร จะส่งผลให้เงื่อนไขการกระจายตัวนั้นมีค่า Loss น้อยที่สุดในทุก ๆ โหนดที่ระดับความลึกของต้นไม้เท่ากัน ประโยชน์ของต้นไม้แบบสมมาตรรวมถึงการประมวลผลที่รวดเร็วกว่าต้นไม้แบบไม่สมมาตร และยังช่วยในการควบคุมการเกิดปัญหา Overfitting อีกด้วย

ในส่วนของวิธีการกระจายตัวของต้นไม้ตัดสินใจ หมายถึงวิธีการค้นหาเงื่อนไขของการกระจายตัว สำหรับ CatBoost นั้น Greedy method จะสร้างความเป็นไปได้ทั้งหมดของการจับคู่ของทุก Feature แล้วเลือกการจับคู่ที่ให้ค่า Loss น้อยที่สุด สำหรับ LightGBM นั้น Gradient-based One-Side Sampling (GOSS) จะเก็บข้อมูลตัวอย่างทั้งหมดกับ Gradient ขนาดใหญ่ แล้วค่อยดำเนินการสุ่มตัวอย่างจากข้อมูลตัวอย่างกับ Gradient ขนาดเล็ก ซึ่ง Gradient ในที่นี้หมายถึงความชันของเส้นสัมผัสของ Loss function ตัวอย่างเช่น มีชุดข้อมูลอยู่ 500,000 แถว ซึ่ง 10,000 แถวมี Gradient ที่ใหญ่กว่า อัลกอริทึมจะเลือกข้อมูล 10,000 แถวนั้นรวมกับข้อมูลที่เหลืออีก 490,000 แถวจำนวนหนึ่งแบบสุ่มมาทำการสร้างต้นแบบ ซึ่งจุดข้อมูลกับ Gradient ที่ใหญ่กว่าจะมีค่าผิดพลาดสูง และจะมีความสำคัญในการหาจุดกระจายตัวที่เหมาะสมที่สุด วิธี GOSS จะเลือกใช้ข้อมูลในการสร้างต้นแบบน้อยกว่า เพราะฉะนั้นจึงใช้เวลาในการสร้างต้นแบบน้อยกว่าอีกด้วย สำหรับ XGBoost นั้น อัลกอริทึม Pre-sorted จะพิจารณา Feature ทั้งหมดและเรียงลำดับโดยค่าของ Feature หลังจากนั้นการตรวจสอบแบบเชิงเส้นจะตัดสินจะตัดสินการกระจายตัวที่ดีที่สุดสำหรับ

Feature และค่าของ Feature ที่ได้รับข้อมูลมากที่สุด ส่วนอัลกอริทึม Histogram-based ทำงานเช่นเดียวกับอัลกอริทึม Pre-sorted แต่แทนที่จะพิจารณา Feature ทั้งหมดมันจะจับกลุ่มค่าของ Feature เป็นกลุ่มของข้อมูลที่ต่อเป็นจำนวนเต็ม (เช่น รหัสไปรษณีย์ , หมวดหมู่(1,2,3,4,5,6,7) หรือการนับจำนวนคน) แล้วหาจุดกระจายตัวอิงตามกลุ่มของข้อมูลนั้นแทน ซึ่งมีประสิทธิภาพมากกว่าอัลกอริทึม Pre-sorted แต่ยังช้ากว่า GOSS

ส่วนของลักษณะ Boosting ในการเลือกข้อมูลเพื่อนำไปสร้างต้นแบบจะมีรูปแบบต่าง ๆ ซึ่งการ Boosting แบบ Ordered จะทำการสร้างต้นแบบด้วยข้อมูลชุดหนึ่ง แล้วประมวลผลด้วยข้อมูลอีกชุดหนึ่ง มีประโยชน์ในการทำให้สามารถนำต้นแบบไปใช้กับข้อมูลที่ไม่เคยเห็นได้ดีขึ้น

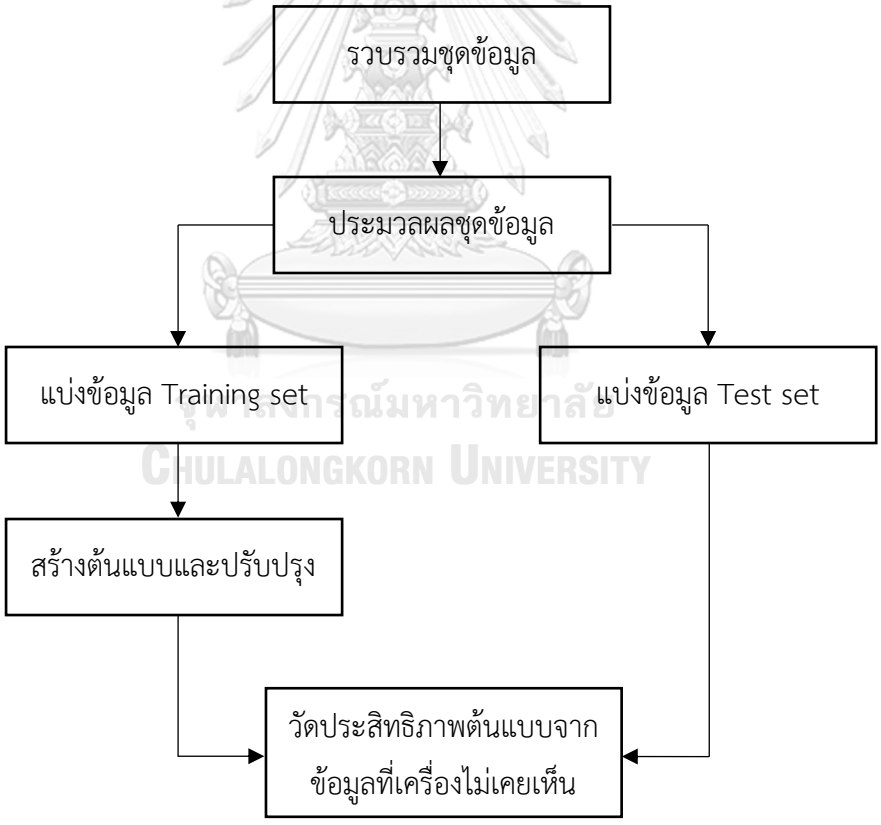
อัลกอริทึม LightGBM, XGBoost และ CatBoost ล้วนเป็นอัลกอริทึมที่มีประสิทธิภาพสูง แม้จะมีพื้นฐานเดียวกันคือ GBDTs แต่ในแต่ละอัลกอริทึมนั้นก็ถูกพัฒนาด้วยเทคนิคที่แตกต่างกัน ทำให้เมื่อใช้อัลกอริทึมในชุดข้อมูลที่แตกต่างกัน ก็อาจจะทำให้อันดับของประสิทธิภาพ รวมไปถึงเวลาในการสร้างต้นแบบของอัลกอริทึมเหล่านี้เปลี่ยนแปลงไป จึงยากที่จะสรุปได้อย่างชัดเจนว่าอัลกอริทึมใดเหมาะสมที่สุดกับชุดข้อมูลของงานวิจัยนี้ หากไม่ได้ทำการทดสอบ

บทที่ 3
วิธีดำเนินงานวิจัย

ในงานวิจัยบทนี้จะอธิบายถึงขั้นตอนของการสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่อง การรวบรวมข้อมูลที่นำมาใช้ในงานวิจัยซึ่งเป็นปัจจัยที่เกี่ยวข้องกับระยะเวลาเดินทาง และ การประมวลผลของแต่ละชุดข้อมูลที่รวบรวมมาให้อยู่ในรูปแบบของตัวแปรต่าง ๆ ก่อนที่จะนำไปสร้างต้นแบบ โดยข้อมูลทั้งหมดเป็นข้อมูลที่สามารถเข้าถึงได้โดยสาธารณะ

3.1 ขั้นตอนการสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่อง

ขั้นตอนของการสร้างต้นแบบการคาดการณ์ระยะเวลาเดินทางระหว่างพิกัดสองจุดในหลากหลายสภาพแวดล้อมบนท้องถนน ด้วยวิธีการเรียนรู้ของเครื่อง สามารถแสดงได้ดังภาพที่ 3.1



ภาพที่ 3.1 ผังงานขั้นตอนการสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่อง

จากภาพที่ 3.1 ขั้นตอนแรกของการสร้างต้นแบบ คือ การรวบรวมข้อมูลจากหลากหลายแหล่งที่มา แล้วทำให้รูปแบบของข้อมูลอยู่ในรูปแบบเดียวกัน (Data frame) เพื่อให้ง่ายต่อการจัดการด้วยโปรแกรม

ขั้นตอนที่สอง คือ การประมวลผลชุดข้อมูล โดยการกำจัดข้อมูลที่ไม่จำเป็นในการสร้างต้นแบบออก ข้อมูลที่ผิดปกติ (Outlier) ข้อมูลที่ขาดหายไปบางส่วน หรือ ข้อมูลที่มีการใส่ค่าผิด หลังจากนั้นอาจทำการสร้างข้อมูลใหม่ขึ้นมาจากข้อมูลที่มีอยู่ เช่น การสร้างข้อมูลระยะทาง จากข้อมูลพาหนะและโทรศัพท์มือถือ แล้วทำการรวมข้อมูลทั้งหมดให้เป็นชุดเดียว ตัวรูปแบบของชุดข้อมูลหลังจากการประมวลผล ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างรูปแบบชุดข้อมูลที่ผ่านการประมวลผล

X_1	X_2	X_3	X_4	X_5	Y
13.74835	100.65628	13.81607	100.64795	9400	1080
13.81607	100.64795	13.82811	100.63003	3600	420
13.82811	100.63003	13.83277	100.57139	7600	600
13.83277	100.57139	13.74802	100.54831	12100	720

จากตารางที่ 3.1 หลัก X คือ ปัจจัย ตัวแปรอิสระ หรือ ตัวแปรอินพุต หลัก Y คือ คำตอบของตัวอย่าง ตัวแปรตาม หรือ ตัวแปรเอาต์พุต สำหรับการเรียนรู้แบบมีผู้สอน และ จำนวนของแถวเท่ากับจำนวนของข้อมูลตัวอย่าง

ขั้นตอนที่สาม คือ การแบ่งข้อมูลเป็น 2 ส่วนเป็น training set ร้อยละแปดสิบของข้อมูล และ test set ร้อยละยี่สิบของข้อมูล

ขั้นตอนที่สี่ คือ การสร้างต้นแบบจากข้อมูล training set ด้วยอัลกอริทึมต่าง ๆ และ วัดประสิทธิภาพของต้นแบบที่สร้างขึ้นโดยเปรียบเทียบระหว่าง ค่า Y ในชุดข้อมูล และ ค่า Y ที่ต้นแบบคาดการณ์ได้ ปรับปรุงต้นแบบโดยการเปลี่ยนแปลงพารามิเตอร์ต่าง ๆ เพื่อให้ต้นแบบมีประสิทธิภาพมากขึ้น

ขั้นตอนสุดท้าย คือ การนำต้นแบบที่ปรับปรุงแล้ว มาวัดประสิทธิภาพกับข้อมูล test set ซึ่งไม่ได้ถูกใช้ในการสร้างต้นแบบ และ ผู้เรียนรู้ไม่เคยเห็นมาก่อน ทำการวัดประสิทธิภาพ และ สรุปผล

3.2 ชุดข้อมูลในการสร้างต้นแบบ (Dataset)

ข้อมูลที่นำมาใช้ในการสร้างต้นแบบนั้น ถูกรวบรวมจากหลายแหล่งที่มาทำให้มีรูปแบบที่แตกต่างกันทั้งนามสกุลไฟล์ แถวหรือตอน ผู้วิจัยจึงต้องจัดการกับข้อมูลจากแต่ละแหล่งที่มาให้แสดงผลอยู่ในรูปแบบเดียวกันเพื่อให้สะดวกต่อการอ่านงานวิจัยและสามารถนำข้อมูลทั้งหมดมาใช้ร่วมกันในการสร้างต้นแบบ โดยข้อมูลทั้งหมดที่ถูกนำเสนอจะเป็นข้อมูลของปี ค.ศ.2020

3.2.1 ข้อมูลจากพาหนะและโทรศัพท์มือถือ (Vehicles and Mobile Probe Data)

ข้อมูลจากพาหนะและโทรศัพท์มือถือหรือเครื่องตรวจวัดเคลื่อนที่จาก iTIC Foundation (มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย, 2564) คือ ข้อมูลการใช้งานของพาหนะที่ถูกรวบรวมจากระบบของพาหนะเอง และจากแอพพลิเคชั่นโทรศัพท์มือถือโดยข้อดีของการใช้ข้อมูลแบบนี้เมื่อเทียบกับการใช้เซ็นเซอร์ตรวจจับแบบเก่า (Fixed Sensors) จะครอบคลุมพื้นที่มากกว่าและมีค่าใช้จ่ายที่ถูกกว่า ข้อมูลที่ได้มีความเที่ยงตรงมากกว่าการประมาณเวลาเดินทางจากความเร็วที่จับได้ขณะผ่านเซ็นเซอร์ (มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย) ข้อมูลจากพาหนะและโทรศัพท์มือถือมี 8 ข้อมูล ได้แก่

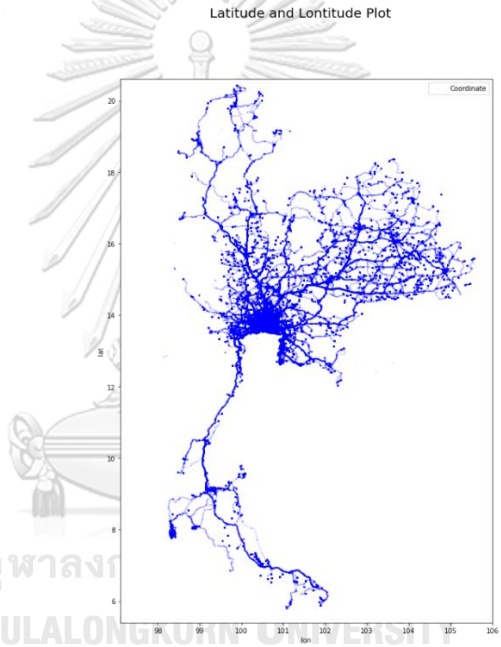
- VehicleID คือ ข้อมูลที่ใช้ระบุพาหนะแต่ละคันโดยพาหนะแต่ละคันใช้ VehicleID เดียวกันเสมอ ไม่มีการเปลี่ยนแปลง
- gpsvalid คือ ค่าที่บ่งบอกถึงสถานะของระบบตรวจจับข้อมูลพาหนะว่ามีสัญญาณหรือไม่ในขณะที่เก็บข้อมูล
- lat และ lon คือ ค่าละติจูดและลองติจูดของพาหนะในขณะที่เก็บข้อมูล
- timestamp คือ เวลาในขณะที่เก็บข้อมูล
- speed คือ ความเร็วของพาหนะในขณะที่เก็บข้อมูล
- for_hire_light คือ ค่าสำหรับพาหนะประเภทแท็กซี่ที่ใช้ในการบ่งบอกถึงสถานะไฟรับผู้โดยสารโดยจะมีค่าเป็น 0 และ 1 หมายถึงปิดไฟและเปิดไฟตามลำดับ
- engine_acc คือ ค่าที่บ่งบอกถึงสถานะเครื่องยนต์ของพาหนะโดยจะมีค่าเป็น 0 และ 1 หมายถึงเครื่องยนต์ไม่ทำงานและทำงานตามลำดับ

ตัวอย่างข้อมูลจากพาหนะและโทรศัพท์มือถือ ดังแสดงในภาพที่ 3.2

	VehicleID	gpsvalid	lat	lon	timestamp	speed	heading	for_hire_light	engine_acc
0	RsiQgWE3MJCt3jAqnvHLILgS5L0	1	13.74835	100.65628	2019-12-31 23:59:25	0	10	0	1
1	UXuWkrJvC5j9FYgMX7AxdpEn/O8	1	13.81607	100.64795	2019-12-31 23:58:46	0	33	0	0
2	RdZ+aLuit7HgakstnAm8wICD3II	1	13.82811	100.63003	2020-01-01 00:00:12	30	188	1	1
3	aGm4w40GOjA/JUkobOgRwQWu/7s	1	13.83277	100.57139	2019-12-31 23:59:58	47	30	1	1
4	Z/EHO1TlpBQhRPkBWuppPrKXXwg	1	13.74802	100.54831	2020-01-01 00:00:09	0	12	1	1

ภาพที่ 3.2 ตัวอย่างข้อมูลจากพาหนะและโทรศัพท์มือถือ ปี ค.ศ.2020

จากภาพที่ 3.2 ข้อมูลจากพาหนะและโทรศัพท์มือถือนั้นถูกเก็บจากทั่วประเทศไทย สามารถแสดงการกระจายตัวของพาพนะด้วยการแสดงพิกัดละติจูดและลองติจูดในรูปแบบกราฟโดยแกน x คือ lon และ แกน y คือ lat ดังแสดงในภาพที่ 3.3



ภาพที่ 3.3 กราฟการกระจายตัวของพาหนะของวันที่ 1 มกราคม ปี ค.ศ.2020

จากภาพที่ 3.3 สังเกตได้ว่านอกจากพื้นที่ตรงกลางแล้ว (บริเวณกรุงเทพมหานคร) การกระจายตัวของพาหนะนั้นไม่ครอบคลุมพื้นที่ทั้งหมด และเนื่องจากงานวิจัยนี้มุ่งเน้นไปที่พาหนะประเภทแท็กซี่ทำให้ข้อมูลน้อยลงกว่านี้ ผู้วิจัยจึงเลือกใช้ข้อมูลบริเวณกรุงเทพมหานครในการสร้างต้นแบบเท่านั้น

3.2.2 ดัชนีรถติด (Traffic Index)

ข้อมูลดัชนีรถติดจาก Longdo Traffic (ลองดู Traffic, 2564) คือ ตัวชี้วัดสภาพความติดขัดของท้องถนนในภาพรวมของกรุงเทพมหานครและปริมณฑล ณ เวลาหนึ่ง ๆ โดยใช้ตัวเลข 0-10 ในการแสดงผล (ค่าที่มาก หมายถึง ติดขัดมาก, 0 = รถไม่ติดเลย, 10 = รถ

ติดทุกถนน) โดยค่าดัชนีนี้จะถูกคำนวณทุก ๆ 5 นาที จากการเก็บข้อมูลสภาพการจราจรบนถนนในโครงข่าย Location Table (มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย, 2561) โดยจะใช้เฉพาะถนนที่มีปริมาณข้อมูลอย่างน้อย 2 ใน 3 ของถนนทั้งสาย ข้อมูลดัชนีรถติดมี 2 ข้อมูล ได้แก่

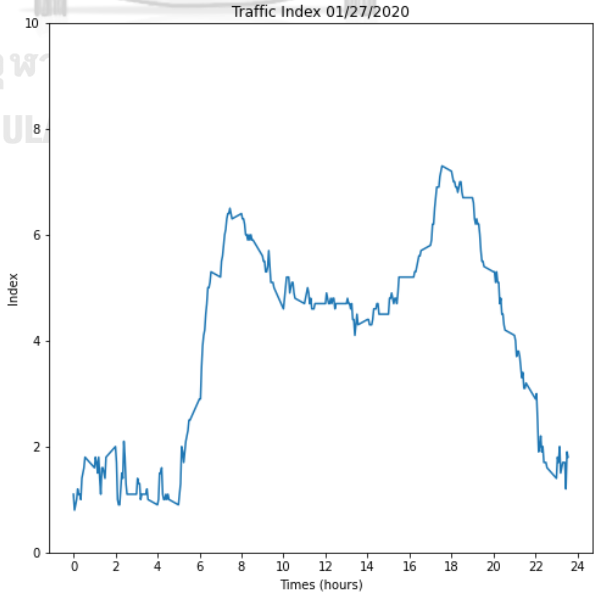
- datetime คือ เวลาที่คำนวณดัชนีรถติดออกมาในขณะนั้น
- index คือ ค่าของดัชนีรถติด

ตัวอย่างข้อมูลดัชนีรถติด ดังแสดงในภาพที่ 3.4

	timestamp	datetime	index
0	1577811600	2020-01-01T00:00	1.0
1	1577811900	2020-01-01T00:05	1.0
2	1577812200	2020-01-01T00:10	1.1
3	1577812500	2020-01-01T00:15	1.2
4	1577812800	2020-01-01T00:20	1.2

ภาพที่ 3.4 ตัวอย่างข้อมูลดัชนีรถติด ปี ค.ศ.2020

จากภาพที่ 3.4 สามารถนำข้อมูลมาแสดงในรูปแบบกราฟโดยแกน x = datetime และแกน y = index เพื่อแสดงให้เห็นถึงระดับความรุนแรงของรถติดในแต่ละช่วงเวลาของวัน ดังแสดงในภาพที่ 3.5



ภาพที่ 3.5 กราฟดัชนีรถติดในทุก 5 นาทีของวันที่ 1 มกราคม ปี ค.ศ.2020

3.2.3 พิกัดกรุงเทพมหานคร (Spatial File)

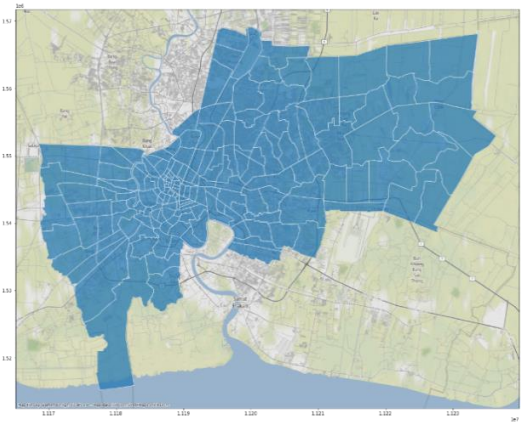
ข้อมูลพิกัดกรุงเทพมหานครจาก BangkokGIS (ศูนย์เทคโนโลยีสารสนเทศภูมิศาสตร์กรุงเทพมหานคร, 2564) คือ ข้อมูลที่บอกว่าพิกัดของแต่ละเขตและแขวงในกรุงเทพมหานครนั้นอยู่ที่ใด ในรูปแบบของพื้นที่ (Polygon) ข้อมูลพิกัดกรุงเทพมหานครมี 4 ข้อมูล ได้แก่

- AREA_CAL คือ พื้นที่ของแขวงในหน่วยตารางกิโลเมตร
- DISTRICT_I กับ SUBDISTRIC คือ ตัวเลขของเขตและแขวงที่ใช้แทนชื่อในการสร้างต้นแบบ (ไม่สามารถใช้ตัวอักษรในการสร้างต้นแบบได้)
- geometry คือ ข้อมูลพื้นที่ของแต่ละแขวงมีชนิดของข้อมูลเป็น geometry ตัวอย่างข้อมูลพิกัดกรุงเทพมหานคร ดังแสดงในภาพที่ 3.6

	OBJECTID	AREA_CAL	AREA_BMA	PERIMETER	ADMIN_ID	SUBDISTRIC	SUBDISTR_1	DISTRICT_I	DISTRICT_N	CHANGWAT_I	CHANGWAT_N	Shape_Leng	Shape_Area	geometry
0	1	15.799	16.461	21537.211388	2	100608	หัวหมาก	1006	บางกะปิ	10	กรุงเทพมหานคร	21534.199039	1.579931e+07	POLYGON ((680135.886 1523155.086, 680143.555 1...
1	2	11.777	12.062	18260.517332	3	100601	คลองจั่น	1006	บางกะปิ	10	กรุงเทพมหานคร	18389.635298	1.177654e+07	POLYGON ((674583.456 1526164.484, 674595.734 1...
2	3	15.830	14.150	17831.192204	2	104003	บางไผ่	1040	บางแค	10	กรุงเทพมหานคร	17823.010749	1.583048e+07	POLYGON ((646244.790 1520711.234, 646245.318 1...
3	4	18.046	18.406	19142.466103	2	100502	อนุสาวรีย์	1005	บางเขน	10	กรุงเทพมหานคร	19100.682438	1.804615e+07	POLYGON ((671453.903 1532668.990, 671455.395 1...
4	5	22.746	23.717	24066.164118	3	100508	ท่าแร้ง	1005	บางเขน	10	กรุงเทพมหานคร	24034.908679	2.275052e+07	POLYGON ((677540.717 1535669.999, 677541.690 1...

ภาพที่ 3.6 ตัวอย่างข้อมูลพิกัดกรุงเทพมหานคร

จากภาพที่ 3.6 เมื่อนำข้อมูลพิกัดกรุงเทพมหานครมาสร้างเป็นภาพ ทับซ้อนกับแผนที่ของกรุงเทพมหานครจะมีลักษณะเป็นรูปทรงสีน้ำเงินตามแขวงในเขตกรุงเทพมหานคร ดังแสดงในภาพที่ 3.7



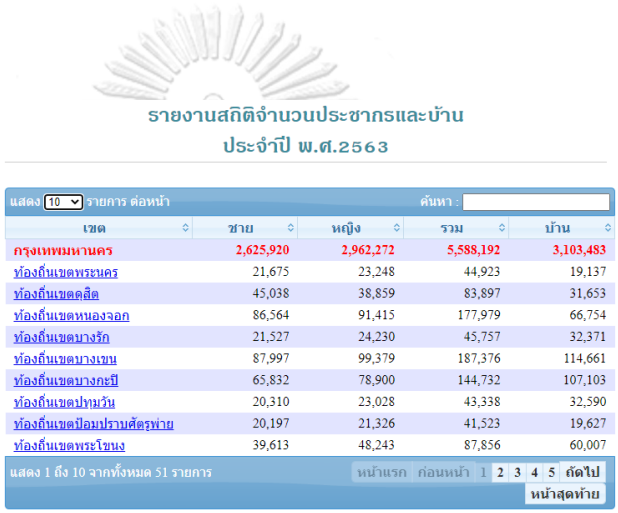
ภาพที่ 3.7 ภาพลักษณะการแบ่งพื้นที่แขวงด้วยข้อมูลพิกัดกรุงเทพมหานคร

3.2.4 สถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร

ข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานครจาก ระบบสถิติทางการทะเบียน (ส่วนบริหารและพัฒนาเทคโนโลยีการทะเบียน & สำนักบริหารการทะเบียน, 2564) คือ ข้อมูลที่บอกจำนวนประชากรและจำนวนบ้านในพื้นที่แต่ละเขตและแขวงในกรุงเทพมหานครจากข้อมูลทะเบียนบ้าน ข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานครมี 2 ข้อมูล ได้แก่

- รวมและบ้าน คือ จำนวนประชากรและจำนวนบ้านในแต่ละแขวง

ตัวอย่างของข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร ดังแสดงในภาพที่ 3.7 และภาพที่ 3.8



รายงานสถิติจำนวนประชากรและบ้าน
ประจำปี พ.ศ.2563

แสดง 10 รายการ ต่อหน้า ค้นหา :

เขต	ชาย	หญิง	รวม	บ้าน
กรุงเทพมหานคร	2,625,920	2,962,272	5,588,192	3,103,483
ท้องถิ่นเขตพระนคร	21,675	23,248	44,923	19,137
ท้องถิ่นเขตดุสิต	45,038	38,859	83,897	31,653
ท้องถิ่นเขตหนองจอก	86,564	91,415	177,979	66,754
ท้องถิ่นเขตบางรัก	21,527	24,230	45,757	32,371
ท้องถิ่นเขตบางเขน	87,997	99,379	187,376	114,661
ท้องถิ่นเขตบางกะปิ	65,832	78,900	144,732	107,103
ท้องถิ่นเขตปทุมวัน	20,310	23,028	43,338	32,590
ท้องถิ่นเขตป้อมปราบศัตรูพ่าย	20,197	21,326	41,523	19,627
ท้องถิ่นเขตพระโขนง	39,613	48,243	87,856	60,007

แสดง 1 ถึง 10 จากทั้งหมด 51 รายการ หน้าแรก ก่อนหน้า 1 2 3 4 5 ถัดไป หน้าสุดท้าย

ภาพที่ 3.8 ตัวอย่างสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร แบ่งตามเขตการปกครอง

รายงานสถิติจำนวนประชากรและบ้าน
ประจำปี พ.ศ.2563

แสดง 25 รายการ ต่อหน้า ค้นหา :

อำเภอ	ชาย	หญิง	รวม	บ้าน
ท้องถิ่นเขตพระนคร	21,675	23,248	44,923	19,137
แขวงพระบรมมหาราชวัง	1,977	1,369	3,346	1,207
แขวงวังบูรพาภิรมย์	4,880	4,770	9,650	5,363
แขวงวัดราชบพิธ	1,391	1,680	3,071	979
แขวงสำราญราษฎร์	1,509	1,431	2,940	1,127
แขวงศาลเจ้าพ่อเสือ	1,256	1,703	2,959	999
แขวงเสาชิงช้า	948	1,101	2,049	698
แขวงบวรนิเวศ	1,834	2,249	4,083	1,573
แขวงตลาดยอด	992	1,142	2,134	1,190
แขวงชนะสงคราม	860	916	1,776	830
แขวงบ้านพานถม	2,688	3,404	6,092	2,017
แขวงบางขุนพรหม	2,033	2,030	4,063	2,183
แขวงวัดสามพระยา	1,307	1,453	2,760	971

แสดง 1 ถึง 13 จากทั้งหมด 13 รายการ หน้าแรก ก่อนหน้า 1 ถัดไป หน้าสุดท้าย

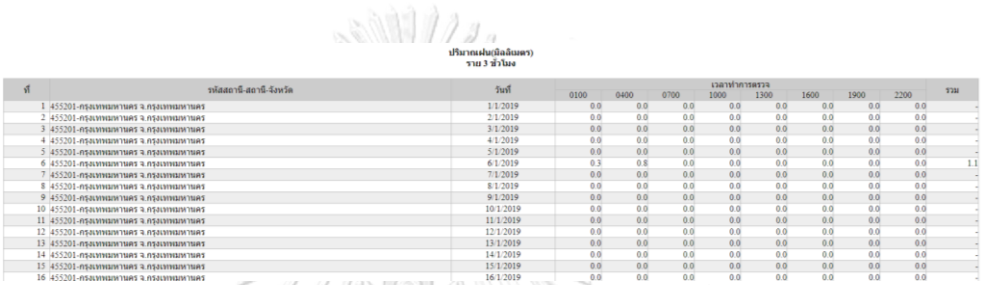
ภาพที่ 3.9 ตัวอย่างสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร แบ่งตามแขวงการปกครอง

3.2.5 สภาพอากาศ

ข้อมูลสภาพอากาศกรุงเทพมหานคร จากกรมอุตุนิยมวิทยา คือ ข้อมูลที่บ่งบอกปริมาณฝน ความชื้นสัมพัทธ์ อุณหภูมิและความเร็วลม มี 4 ข้อมูล ได้แก่

- ปริมาณฝนหน่วยมิลลิเมตร
- ความชื้นสัมพัทธ์หน่วยเปอร์เซ็นต์
- อุณหภูมิหน่วยฟาเรนไฮต์
- ความเร็วลมหน่วยไมล์ต่อชั่วโมง

ตัวอย่างข้อมูลปริมาณฝนและความชื้นสัมพัทธ์ ดังแสดงในภาพที่ 3.10 และภาพที่ 3.11



ปริมาณฝน(มิลลิเมตร)
ราย 3 ชั่วโมง

ที่	รหัสสถานี-สถานี-จังหวัด	วันที่	เวลาทำการตรวจ								รวม
			0100	0400	0700	1000	1300	1600	1900	2200	
1	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	1/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
2	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	2/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
3	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	3/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
4	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	4/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
5	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	5/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
6	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	6/1/2019	0.3	0.8	0.0	0.0	0.0	0.0	0.0	0.0	1.1
7	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	7/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
8	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	8/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
9	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	9/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
10	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	10/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
11	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	11/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
12	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	12/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
13	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	13/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
14	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	14/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
15	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	15/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
16	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	16/1/2019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-

ภาพที่ 3.10 ตัวอย่างปริมาณฝนรายสามชั่วโมง

ความชื้นสัมพัทธ์(เปอร์เซ็นต์)
ราย 3 ชั่วโมง

ที่	รหัสสถานี-สถานี-จังหวัด	วันที่	เวลาทำการตรวจ								เฉลี่ย
			0100	0400	0700	1000	1300	1600	1900	2200	
1	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	1/1/2019	72	76	85	57	52	58	70	69	67
2	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	2/1/2019	69	77	68	60	51	50	60	66	63
3	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	3/1/2019	76	74	70	60	57	61	69	65	67
4	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	4/1/2019	69	71	66	63	60	65	70	76	68
5	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	5/1/2019	68	68	71	75	70	70	77	80	72
6	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	6/1/2019	86	87	93	87	60	60	71	81	78
7	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	7/1/2019	82	83	88	75	60	60	73	80	75
8	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	8/1/2019	85	90	92	85	69	71	79	87	82
9	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	9/1/2019	90	91	90	83	75	78	76	84	83
10	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	10/1/2019	93	92	92	81	67	57	74	81	80
11	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	11/1/2019	86	90	90	67	55	56	76	77	75
12	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	12/1/2019	91	93	90	70	54	55	70	72	74
13	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	13/1/2019	86	90	90	65	61	56	65	73	73
14	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	14/1/2019	88	92	90	65	56	60	78	76	76
15	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	15/1/2019	84	88	90	67	72	60	69	85	77
16	455201-กรุงเทพมหานคร จ.กรุงเทพมหานคร	16/1/2019	87	89	95	72	52	47	57	67	71

ภาพที่ 3.11 ตัวอย่างความชื้นสัมพัทธ์รายสามชั่วโมง

จากภาพที่ 3.10 ข้อมูลปริมาณฝนรายสามชั่วโมงที่จำเป็นต้องใช้ประกอบไปด้วยปริมาณฝนหน่วยมิลลิเมตร มีชนิดของข้อมูลเป็น float64 และภาพที่ 3.11 ข้อมูลความชื้นสัมพัทธ์รายสามชั่วโมงที่จำเป็นต้องใช้ประกอบไปด้วยความชื้นสัมพัทธ์หน่วยเปอร์เซ็นต์ มีชนิดของข้อมูลเป็น int64

3.3 การประมวลผลชุดข้อมูลเริ่มต้น

การประมวลผลชุดข้อมูลเริ่มต้น เพื่อเปลี่ยนแปลงข้อมูลที่มีอยู่ให้เป็นตัวแปรที่เราต้องการ ทำการวิเคราะห์ และปรับปรุงข้อมูลด้วยวิธีทางสถิติ ก่อนรวบรวมตัวแปรทั้งหมดเพื่อไปสู่ขั้นตอนต่อไปของการสร้างต้นแบบ งานวิจัยนี้จะมุ่งเน้นไปที่พาหนะประเภทแท็กซี่ที่มีผู้โดยสารอยู่บนรถ ซึ่งสามารถอนุมานได้ว่าขณะนั้นรถแท็กซี่เคลื่อนที่ด้วยความเร็วปกติเช่นเดียวกับรถส่งสินค้า โดยผู้วิจัยสามารถสรุปข้อมูลตัวแปรทั้งหมดที่นำไปใช้ในการสร้างต้นแบบ แสดงไว้ดังตารางที่ 3.2

ตารางที่ 3.2 สรุปข้อมูลตัวแปร

ลำดับ	ตัวแปร	รายละเอียด	หน่วย	ชุดข้อมูลเดิม
1	PickupLat	พิกัดละติจูดที่รับผู้โดยสาร	-	ข้อมูลจากพาหนะและโทรศัพท์มือถือ (Vehicles and Mobile Probe Data)
2	PickupLon	พิกัดลองติจูดที่รับผู้โดยสาร	-	
3	DropoffLat	พิกัดละติจูดที่ส่งผู้โดยสาร	-	
4	DropoffLon	พิกัดลองติจูดที่ส่งผู้โดยสาร	-	
5	AverageSpeed	ความเร็วเฉลี่ยของพาหนะ	กิโลเมตรต่อชั่วโมง	
6	PickupSecofDay	เวลาที่รับผู้โดยสารนับตั้งแต่เวลาเริ่มต้นวัน	วินาที	
7	DayofWeek	วันในสัปดาห์ที่รับผู้โดยสาร	-	
8	DayofMonth	วันที่ที่รับผู้โดยสาร	-	
9	Month	เดือนที่รับผู้โดยสาร	-	
10	Hour	ชั่วโมงที่รับผู้โดยสาร	ชั่วโมง	
11	Distance	ระยะทางระหว่างจุดรับและส่งผู้โดยสาร	เมตร	
12	TravelTime	เวลาระหว่างรับและส่งผู้โดยสาร	วินาที	
13	Displacement	ระยะการกระจัดระหว่างจุดรับและส่งผู้โดยสาร	เมตร	
14	Direction	ทิศทางระหว่างจุด	องศา	
15	PickupDistr	เขตที่รับผู้โดยสาร	-	พิกัดกรุงเทพมหานคร (Spatial File)
16	DropoffDistr	เขตที่ส่งผู้โดยสาร	-	
17	PickupSubDistr	แขวงที่รับผู้โดยสาร	-	

ตารางที่ 3.2 สรุปข้อมูลตัวแปร (ต่อ)

18	DropoffSubDistr	แขวงที่ส่งผู้โดยสาร	-	พิกัดกรุงเทพมหานคร (Spatial File)
19	PickupArea	พื้นที่แขวงที่รับผู้โดยสาร	ตารางกิโลเมตร	
20	DropoffArea	พื้นที่แขวงที่ส่งผู้โดยสาร	ตารางกิโลเมตร	
21	Cluster	การแบ่งกลุ่มวันตามรูปแบบดัชนีรถติดที่คล้ายกัน	-	ดัชนีรถติด (Traffic Index)
22	PickPplStat	จำนวนประชากรในแขวงที่รับผู้โดยสาร	คน	สถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร
23	PickBuildStat	จำนวนบ้านในแขวงที่รับผู้โดยสาร	หลังคาเรือน	
24	DropPplStat	จำนวนประชากรในแขวงที่ส่งผู้โดยสาร	คน	
25	DropBuildStat	จำนวนบ้านในแขวงที่ส่งผู้โดยสาร	หลังคาเรือน	
26	Temperature	อุณหภูมิ	ฟาเรนไฮต์	สภาพอากาศ
27	Humidity	ความชื้น	เปอร์เซ็นต์	
28	Wind Speed	ความเร็วลม	ไมล์ต่อชั่วโมง	
29	Rain	ปริมาณน้ำฝน	มิลลิเมตร	

จากตารางที่ 3.2 ในการประมวลผลข้อมูลเพื่อเปลี่ยนเป็นตัวแปรต่าง ๆ มีขั้นตอน และ การใช้เครื่องมือที่แตกต่างกัน ซึ่งจะมีการอธิบายในหัวข้อย่อยถัดไป สำหรับตัวแปรที่กล่าวถึง แต่ไม่ได้อยู่ในตาราง คือ ตัวแปรที่ทำการลบออกก่อนนำไปสร้างต้นแบบ

3.3.1 การประมวลผลชุดข้อมูลพาหนะและโทรศัพท์มือถือ

ชุดข้อมูลพาหนะและโทรศัพท์มือถือ เป็นข้อมูลที่สร้างตัวแปรที่สำคัญที่สุด ของงานวิจัยนี้ ซึ่งเมื่อประมวลผลชุดข้อมูลพาหนะและโทรศัพท์มือถือ จะได้ชุดข้อมูลใหม่ที่เรียกว่า Origin-Destination pair คือ ชุดข้อมูลที่มีตัวแปร [1-4] สำหรับบอกพิกัดละติจูดลองติจูด เวลารับ และ เวลาส่งผู้โดยสาร โดยจะเรียกข้อมูลเหล่านี้ในแต่ละแถวว่า ทริป

ขั้นตอนแรก ทำการตรวจสอบตัวแปร engine_acc และลบแถวที่มีค่าเป็น 0 ทั้งหมดเพื่อคัดช่วงที่รถหยุดการทำงานเป็นเวลาเกิน 3 นาทีออก เพราะในการสร้างทริป ไม่จำเป็นต้องใช้ข้อมูลในช่วงที่รถหยุดนิ่งเป็นเวลานาน ๆ

ขั้นตอนที่สอง ในการแยก VehicleID ที่เป็นแท็กซี่ จะทำการตรวจสอบทีละ VehicleID ถ้าในข้อมูล ค่าของตัวแปร for_hire_light มีการเปลี่ยนแปลงจาก 1 เป็น 0 หรือ 0 เป็น 1 ได้ หมายความว่าไฟรับคนของรถแท็กซี่ของ VehicleID นั้นสามารถเปิด และปิดได้ หมายความว่า VehicleID นั้นเป็นรถแท็กซี่ ซึ่งพาหนะอื่นไม่สามารถเปลี่ยนแปลงตัวแปรนี้ได้ วิธีนี้รวมถึงการคัดแท็กซี่ที่ไม่มีการเปลี่ยนแปลงไฟรับคนออกด้วย เนื่องจากไม่สามารถระบุได้ว่าแท็กซี่กำลังมีผู้โดยสารหรือไม่ รวมแล้วจะได้ VehicleID แท็กซี่ที่มีการเปิด ปิดไฟรับผู้โดยสาร 2,944 คัน หลังจากนั้นทำการตัดแถวที่ตัวแปร for_hire_light เป็น 1 ออก เพราะการสร้างทริปจะใช้เฉพาะตอนที่แท็กซี่ปิดไฟรับคน (มีผู้โดยสาร)

ขั้นตอนที่สาม หลักการแปลงข้อมูลพาหนะและโทรศัพท์มือถือ เป็น Origin-Destination pair โดยพิจารณาแต่ละ VehicleID นำตัวแปร lat lon และ timestamp จากแถวเริ่มต้นไปเป็นตัวแปร [1] PickupLat [2] PickupLon และ PickupTime (ตัวแปรบ่งบอกเวลารับผู้โดยสาร) ตามลำดับ ทำการตรวจสอบแถวถัดไปโดยสังเกตที่ตัวแปร timestamp หากมีความต่างกับแถวก่อนหน้าไม่เกิน 3 นาที (เวลาที่ตัวแปรการทำงานของรถเปลี่ยนเป็น 0 หรือ แถวที่ตัดช่วงเวลาที่ไฟรับคนเปิดออกไปจะเป็นช่วงว่างระยะเวลาหนึ่ง) ให้ทำการตรวจสอบแถวถัดไปเรื่อย ๆ จนพบแถวที่ไม่เข้าเงื่อนไข ให้นำตัวแปร lat lon และ timestamp ของแถวก่อนหน้าแถวนั้นเปลี่ยนเป็นตัวแปร [3] DropoffLat [4] DropoffLon และ DropoffTime หลังจากนั้นนำตัวแปร timestamp ของแถวก่อนหน้ามาลบกับแถวเริ่มต้นจะได้ตัวแปร [12] TravelTime ทำการคิดความเร็วเฉลี่ยจากตัวแปร speed จะได้ตัวแปร [5] AverageSpeed ตัวแปรทั้งหมดในขั้นตอนที่สามนี้ หมายถึง 1 ทริป

ขั้นตอนที่สี่ เปลี่ยนแถวที่ไม่เข้าเงื่อนไขให้เป็นแถวเริ่มต้น แล้วทำขั้นตอนที่สามต่อไปจนสิ้นสุดแถวข้อมูล จะได้ข้อมูล Origin-Destination pair ออกมาดังภาพที่ 3.12

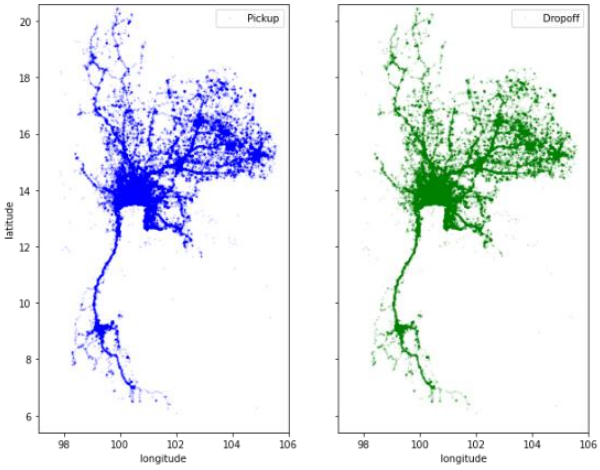
	VehicleID	Pickuplat	Pickuplon	Dropofflat	Dropofflon	Pickuptime	Dropofftime	AverageSpeed	TravelTime
0	t/qUUjC9QcHmOcUkBbqCMQlCwqg	13.76747	100.40770	13.63370	100.36327	2019-12-31 23:59:30	2020-01-01 00:18:29	54.733333	1139
1	t/qUUjC9QcHmOcUkBbqCMQlCwqg	13.63363	100.36286	13.58800	101.01082	2020-01-01 01:24:45	2020-01-01 03:06:20	61.600000	6095
2	t/qUUjC9QcHmOcUkBbqCMQlCwqg	13.58811	101.01086	12.94305	100.89512	2020-01-01 03:28:05	2020-01-01 04:56:34	58.184615	5309
3	t/qUUjC9QcHmOcUkBbqCMQlCwqg	12.93836	100.89349	12.89822	100.86746	2020-01-01 04:59:34	2020-01-01 05:31:34	18.708333	1920
4	t/qUUjC9QcHmOcUkBbqCMQlCwqg	12.89708	100.86851	12.89682	100.86874	2020-01-01 06:08:05	2020-01-01 07:17:05	0.230769	4140
...	...	...	...	...	...	...	...	...	...
23625623	biEZKZtF3oyJLVRd03007CKoAoc	13.82461	100.52951	13.81436	100.51960	2020-12-31 22:18:32	2020-12-31 22:23:59	37.250000	327
23625624	biEZKZtF3oyJLVRd03007CKoAoc	13.75994	100.49893	13.75960	100.50233	2020-12-31 22:41:54	2020-12-31 22:42:59	10.000000	65
23625625	biEZKZtF3oyJLVRd03007CKoAoc	13.75745	100.49723	13.79242	100.50484	2020-12-31 22:48:17	2020-12-31 22:56:59	39.571429	522
23625626	biEZKZtF3oyJLVRd03007CKoAoc	13.79061	100.48856	13.79984	100.50862	2020-12-31 23:18:18	2020-12-31 23:27:59	17.875000	581
23625627	biEZKZtF3oyJLVRd03007CKoAoc	13.73111	100.46631	13.74864	100.46990	2020-12-31 23:46:59	2020-12-31 23:51:59	31.800000	300

ภาพที่ 3.12 ข้อมูล Origin-Destination pair

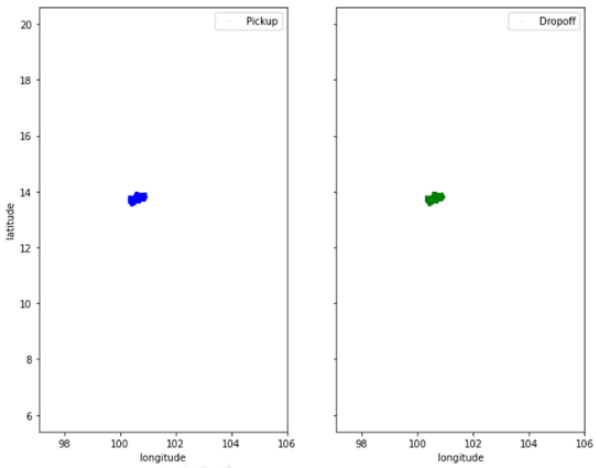
จากภาพที่ 3.12 ข้อมูล Origin-Destination pair มีจำนวน 23,625,628 แถว 9 หลัก ประกอบไปด้วยตัวแปรที่นำไปสร้างต้นแบบ 6 ตัว ได้แก่ [1] PickupLat, [2] PickupLon, [3] DropoffLat, [4] DropoffLon [5] AverageSpeed, [12] TravelTime และตัวแปร PickupTime, DropoffTime สำหรับ VehicleID ทำการตัดออกเนื่องจากไม่สามารถใช้ตัวอักษรในการสร้างต้นแบบได้

3.3.2 การประมวลผลชุดข้อมูลพิกัดกรุงเทพมหานคร

ชุดข้อมูลพิกัดกรุงเทพมหานคร คือ ข้อมูลเกี่ยวกับพื้นที่เขต และแขวงในกรุงเทพมหานคร ซึ่งจะจัดการแบ่งพื้นที่ตามพิกัด โดยตัดทริปที่มีพิกัดนอกเหนือจากกรุงเทพมหานครออกไป ด้วยเครื่องมือ GeoPandas ซึ่งเป็นเครื่องมือที่ช่วยในการจัดการกับข้อมูลทางภูมิศาสตร์ด้วยภาษา Python กราฟแสดงการตัดทริปที่ไม่เกี่ยวข้องออกดังภาพที่ 3.13 และ 3.14



ภาพที่ 3.13 กราฟพิกัดทริปก่อนลบทริปที่อยู่นอกพิกัดกรุงเทพมหานคร



ภาพที่ 3.14 กราฟพิกัดทริปหลังลบทริปที่อยู่นอกพิกัดกรุงเทพมหานคร

จากภาพที่ 3.13 และ 3.14 เมื่อทำการตัดทริปอื่น จนข้อมูลเหลือเฉพาะทริปพิกัดที่อยู่ในกรุงเทพมหานครแล้ว จึงทำการเพิ่มตัวแปรต่าง ๆ จากชุดข้อมูลเข้าไป ดังแสดงในภาพที่ 3.15

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	Pickuptime	Dropofftime	AverageSpeed	TravelTime	PickupArea	PickupSubDistr	PickupDistr	DropoffArea	DropoffSubDistr	DropoffDistr
0	13.76747	100.40770	13.63370	100.36327	2019-12-31 23:59:30	2020-01-01 00:18:29	54.733333	1139	8.539	101905	1019	34.745	105002	1050
1	13.77576	100.40927	13.67465	100.40660	2020-01-02 09:01:31	2020-01-02 09:13:30	52.333333	719	8.539	101905	1019	34.745	105002	1050
2	13.77469	100.41561	13.67527	100.40664	2020-01-02 08:31:57	2020-01-02 08:46:57	44.250000	900	8.539	101905	1019	34.745	105002	1050
3	13.76175	100.41558	13.67498	100.40664	2020-01-04 08:23:48	2020-01-04 08:40:48	37.857143	1020	8.539	101905	1019	34.745	105002	1050
4	13.77525	100.42672	13.66921	100.40621	2020-01-13 11:47:26	2020-01-13 12:06:26	46.533333	1140	8.539	101905	1019	34.745	105002	1050
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15608779	13.89062	100.86656	13.88920	100.86365	2020-12-30 13:19:18	2020-12-30 13:22:18	16.333333	180	38.867	100304	1003	38.867	100304	1003
15608780	13.90717	100.86414	13.91175	100.86362	2020-12-30 09:24:24	2020-12-30 09:25:24	29.500000	60	38.867	100304	1003	38.867	100304	1003
15608781	13.89330	100.86513	13.89083	100.86367	2020-12-31 09:44:34	2020-12-31 09:47:34	23.666667	180	38.867	100304	1003	38.867	100304	1003
15608782	13.90718	100.86463	13.90369	100.86305	2020-12-31 17:39:16	2020-12-31 17:41:55	17.666667	159	38.867	100304	1003	38.867	100304	1003
15608783	13.90204	100.86260	13.89549	100.86299	2020-12-31 20:15:36	2020-12-31 20:17:01	22.666667	85	38.867	100304	1003	38.867	100304	1003

ภาพที่ 3.15 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลพิกัดกรุงเทพมหานคร

จากภาพที่ 3.15 ข้อมูลหลังจากตัดทริปนอกพิกัดกรุงเทพมหานคร และเพิ่มตัวแปร มีจำนวน 15,608,784 แถว 14 หลัก มีตัวแปรที่นำไปสร้างต้นแบบเพิ่มขึ้น 6 ตัว ได้แก่ [15] PickupDistr, [16] DropoffDistr, [17] PickupSubDistr, [18] DropoffSubDistr, [19] PickupArea, [20] DropoffArea

3.3.3 การประมวลผลชุดข้อมูลดัชนีรถติด

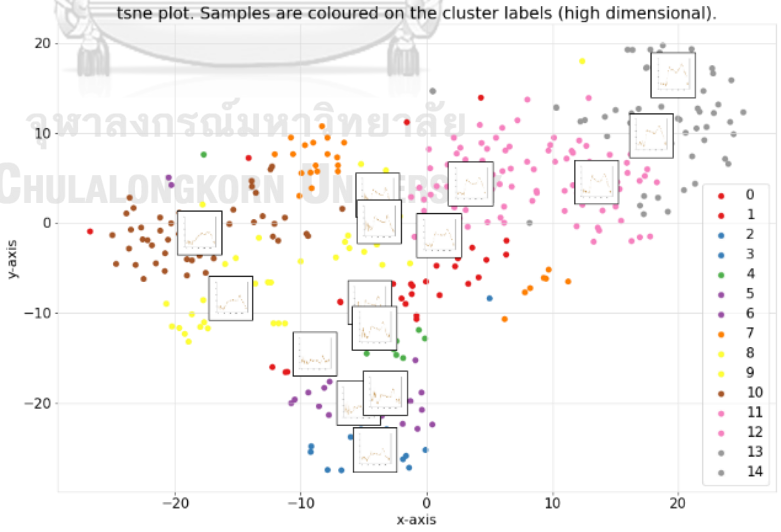
การประมวลผลชุดข้อมูลดัชนีรถติด มีหลักการ คือ การนำกราฟดัชนีรถติดมาจัดกลุ่มวันที่มีแนวโน้มดัชนีรถติดที่คล้ายคลึงกัน ด้วยการใช้เครื่องมือวิธีการเรียนรู้ของเครื่องที่

เรียกว่า clustimage เมื่อผ่านขั้นตอนการประมวลผลสำเร็จ รูปภาพแต่ละรูปจะถูกจัดกลุ่มโดยการใช้ label ดังแสดงในภาพที่ 3.16

```
array([ 8,  8,  8, 10, 10, 11, 12, 12, 12, 14,  7, 10, 11, 12, 12, 11, 12,
        7, 10, 11, 12, 12, 14, 11, 10, 10, 11, 12, 11, 11, 14,  7, 10, 11,
       11, 12, 11, 14, 10,  8, 10, 11, 11, 11, 14,  7, 10, 11, 11, 11, 11,
       14,  7, 10, 11, 11, 11, 11, 14,  7, 10, 11, 11, 11, 11, 14, 10, 10,
        9,  9, 11, 11, 14, 10, 10,  9,  9,  9,  9,  9, 10,  2,  8,  8,  8,
        2,  2,  2,  2,  3,  3,  3,  2,  2,  2,  3,  3,  6,  2,  2,  6,  2,
        3,  6,  2,  2,  6,  6,  5,  3,  6,  6,  6,  6,  6,  5,  2,  6,  6,
        6,  4,  5,  5,  6,  6,  4,  5,  4,  4,  6,  6,  6,  4,  4,  4,  4,
        1,  5,  5,  4,  4,  4,  1,  1,  5,  1,  8,  1,  1,  0,  1,  8,  1,
        0,  1,  1,  0,  1,  5,  1,  1,  1,  1,  0,  1,  5,  5, 14,  0,  0,
        0,  7,  8,  9,  9,  0,  0,  0,  7, 10,  0,  0, 14, 12, 14,  7,  8,
        8,  9, 12, 11, 14,  7, 10, 11, 12, 12, 14, 14,  7, 10, 11, 11, 11,
       12, 14,  7,  8,  8,  8, 12, 12, 14,  7, 10, 14, 13, 12, 12, 14,  7,
       10, 11, 14, 10, 14, 14,  7, 10, 14, 12, 12, 12, 14,  7,  0, 12, 12,
       12, 12, 14,  7, 10, 14, 14, 12, 14, 10,  8,  8,  8, 12, 12, 12, 14,
        7, 10, 11, 12, 12, 11, 14,  9, 10, 11, 12, 14, 12, 14,  7, 10, 14,
       14, 12, 14, 14,  7, 10, 13, 12, 12, 12, 13,  7, 10, 11, 10, 12, 14,
       13,  9, 10, 13, 13, 13, 13, 10, 10, 10, 13, 13, 13, 13, 13,  9, 10,
       13, 13, 13, 13, 13,  7, 10, 11, 11, 11, 11, 14,  7, 10, 11, 11, 11,
       10, 10, 10,  8, 11, 11, 11, 11, 14,  7, 10, 11, 11, 12, 12, 14, 10,
       10, 10, 12, 14, 10, 10, 10, 10, 11, 12, 12, 12, 14,  7, 10, 11, 11,
        9,  9, 11, 10,  8,  8,  8,  8,  8], dtype=int32)
```

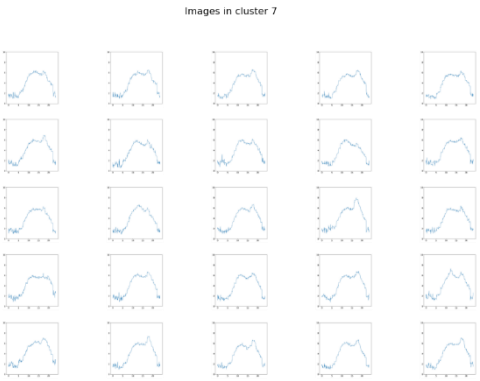
ภาพที่ 3.16 การกำหนด label ของรูปภาพ

จากภาพที่ 3.16 อีกทั้งเรายังสามารถแสดงตัวอย่างผลลัพธ์การจัดกลุ่มทั้งหมดของรูปภาพ ในรูปของการพล็อตกราฟ tsne (t-distributed stochastic neighbor embedding) ซึ่งเป็นวิธีการทางสถิติสำหรับการแสดงข้อมูลที่มีมิติสูง ดังภาพที่ 3.17

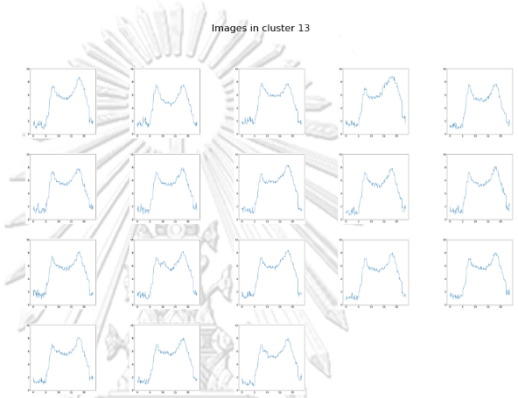


ภาพที่ 3.17 กราฟ tsne แสดงตัวอย่างของผลลัพธ์จากการจัดกลุ่มรูปภาพ

จากภาพที่ 3.17 กราฟแสดงให้เห็นถึงการจัดกลุ่มของรูปภาพทั้งหมด 15 กลุ่ม โดยจะยกตัวอย่างของกลุ่มในที่ 7 และกลุ่มที่ 13 ดังภาพที่ 3.18 และ 3.19



ภาพที่ 3.18 รูปภาพที่ถูกจัดอยู่ในกลุ่มที่ 7



ภาพที่ 3.19 รูปภาพที่ถูกจัดอยู่ในกลุ่มที่ 13

จากภาพที่ 3.18 และ 3.19 แสดงให้เห็นว่ารูปภาพที่มีความคล้ายคลึงกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน เมื่อทำการเพิ่มตัวแปรเข้าไปในข้อมูลจะได้ดังภาพที่ 3.20

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	Pickuptime	Dropofftime	AverageSpeed	TravelTime	Cluster	PickupArea	PickupSubDistr	PickupDistr	DropoffArea	DropoffSubDistr	DropoffDistr
0	13.76747	100.40770	13.63370	100.36327	2019-12-31 23:59:30	2020-01-01 00:18:29	54.733333	1139	8	8.539	101905	1019	34.745	105002	1050
1	13.77576	100.40927	13.67465	100.40660	2020-01-02 09:01:31	2020-01-02 09:13:30	52.333333	719	8	8.539	101905	1019	34.745	105002	1050
2	13.77469	100.41561	13.67527	100.40664	2020-01-02 08:31:57	2020-01-02 08:46:57	44.250000	900	8	8.539	101905	1019	34.745	105002	1050
3	13.76175	100.41558	13.67498	100.40664	2020-01-04 08:23:48	2020-01-04 08:40:48	37.857143	1020	10	8.539	101905	1019	34.745	105002	1050
4	13.77525	100.42672	13.66921	100.40621	2020-01-13 11:47:26	2020-01-13 12:06:26	46.533333	1140	11	8.539	101905	1019	34.745	105002	1050
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15608779	13.89062	100.86656	13.88920	100.86365	2020-12-30 13:19:18	2020-12-30 13:22:18	16.333333	180	8	38.867	100304	1003	38.867	100304	1003
15608780	13.90717	100.86414	13.91175	100.86362	2020-12-30 09:24:24	2020-12-30 09:25:24	29.500000	60	8	38.867	100304	1003	38.867	100304	1003
15608781	13.89330	100.86513	13.89083	100.86367	2020-12-31 09:44:34	2020-12-31 09:47:34	23.666667	180	8	38.867	100304	1003	38.867	100304	1003
15608782	13.90718	100.86463	13.90369	100.86305	2020-12-31 17:39:16	2020-12-31 17:41:55	17.666667	159	8	38.867	100304	1003	38.867	100304	1003
15608783	13.90204	100.86260	13.89549	100.86299	2020-12-31 20:15:36	2020-12-31 20:17:01	22.666667	85	8	38.867	100304	1003	38.867	100304	1003

ภาพที่ 3.20 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลดัชนีรถติด

จากภาพที่ 3.20 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลดัชนีรถติด มีจำนวน 15,608,784 แถว 15 หลัก มีตัวแปรที่นำไปสร้างต้นแบบเพิ่มขึ้น 1 ตัว ได้แก่ [21] Cluster

3.3.4 การประมวลผลชุดข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร

ชุดข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร เป็นข้อมูลที่บันทึกจำนวนประชากร และบ้านในแต่ละเขต แขวง ทำการเพิ่มตัวแปรจำนวนประชากร และบ้านในข้อมูล โดยอ้างอิงจากตัวแปร [17] PickupSubDistr และ [18] DropoffSubDistr ทำให้ได้ข้อมูลดังแสดงในภาพที่ 3.21

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	Pickuptime	Dropofftime	AverageSpeed	TravelTime	Cluster	PickupArea	PickupSubDistr	PickupDistr	PickPplStat	PickBuildStat	DropoffArea	DropoffSubDistr	DropoffDistr	DropPplStat	DropBuildStat
0	13.76747	100.40770	13.63370	100.36327	2019-12-31 23:59:30	2020-01-01 00:18:29	54.733333	1139	8	8.539	101905	1019	19783	8032	34.745	105002	1050	22619	10234
1	13.77576	100.40927	13.67465	100.40660	2020-01-02 09:01:31	2020-01-02 09:13:30	52.333333	719	8	8.539	101905	1019	19783	8032	34.745	105002	1050	22619	10234
2	13.77469	100.41561	13.67527	100.40664	2020-01-02 08:31:57	2020-01-02 08:46:57	44.250000	900	8	8.539	101905	1019	19783	8032	34.745	105002	1050	22619	10234
3	13.76175	100.41558	13.67498	100.40664	2020-01-04 08:23:48	2020-01-04 08:40:48	37.857143	1020	10	8.539	101905	1019	19783	8032	34.745	105002	1050	22619	10234
4	13.77525	100.42672	13.66921	100.40621	2020-01-13 11:47:26	2020-01-13 12:06:26	46.533333	1140	11	8.539	101905	1019	19783	8032	34.745	105002	1050	22619	10234
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15608779	13.89062	100.86656	13.88920	100.86365	2020-12-30 13:19:18	2020-12-30 13:22:18	16.333333	180	8	38.867	100304	1003	11858	4369	38.867	100304	1003	11858	4369
15608780	13.90717	100.86414	13.91175	100.86362	2020-12-30 09:24:24	2020-12-30 09:25:24	29.500000	60	8	38.867	100304	1003	11858	4369	38.867	100304	1003	11858	4369
15608781	13.89330	100.86513	13.89083	100.86367	2020-12-31 09:44:34	2020-12-31 09:47:34	23.666667	180	8	38.867	100304	1003	11858	4369	38.867	100304	1003	11858	4369
15608782	13.90718	100.86463	13.90369	100.86305	2020-12-31 17:39:16	2020-12-31 17:41:55	17.666667	159	8	38.867	100304	1003	11858	4369	38.867	100304	1003	11858	4369
15608783	13.90204	100.86260	13.89549	100.86299	2020-12-31 20:15:36	2020-12-31 20:17:01	22.666667	85	8	38.867	100304	1003	11858	4369	38.867	100304	1003	11858	4369

ภาพที่ 3.21 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลสถิติจำนวนประชากรและบ้าน

จากภาพที่ 3.21 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลสถิติจำนวนประชากรและบ้านในกรุงเทพมหานคร มีจำนวน 15,608,784 แถว 19 หลัก มีตัวแปรที่นำไปสร้างต้นแบบเพิ่มขึ้น 4 ตัว ได้แก่ [22] PickPplStat, [23] PickBuildStat, [24] DropPplStat และ [25] DropBuildStat

3.3.5 การประมวลผลชุดข้อมูลสภาพอากาศ

ชุดข้อมูลสภาพอากาศ จะทำการเพิ่มตัวแปรโดยอ้างอิงจากเวลาที่รับผู้โดยสาร จึงทำการ แยกตัวแปร PickupTime ออกมาเป็นตัวแปร [6] PickupSecofDay, [7] DayofWeek, [8] DayofMonth, [9] Month และ [10] Hour เพื่อเป็นตัวแปรอ้างอิงในการเพิ่มตัวแปรที่ได้จากชุดข้อมูลสภาพอากาศ ทำการลบตัวแปร PickupTime และ DropoffTime ออก แล้วลบทริปที่มีค่าตัวแปรซ้ำกันทั้งหมดจะได้ข้อมูลดังแสดงในภาพที่ 3.22

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	AverageSpeed	TravelTime	Cluster	PickupArea	PickupSubDistr	...	DropBuildStat	PickupDayofWeek	PickupDayofMonth	PickupSecofDay	PickupMonth	Hour	Temperature	Humidity	Wind Speed	Rain
0	13.76747	100.40770	13.63370	100.36327	54.733333	1139	8	8.539	101905	...	10234	1	31	86370	12	23	72	69	3	0.0
1	13.77576	100.40927	13.67465	100.40660	52.333333	719	8	8.539	101905	...	10234	3	2	32491	1	9	81	70	6	0.0
2	13.77469	100.41561	13.67527	100.40664	44.250000	900	8	8.539	101905	...	10234	3	2	30717	1	8	77	74	7	0.0
3	13.76175	100.41558	13.67498	100.40664	37.857143	1020	10	8.539	101905	...	10234	5	4	30228	1	8	79	74	0	0.0
4	13.77525	100.42672	13.66921	100.40621	46.533333	1140	11	8.539	101905	...	10234	0	13	42446	1	11	84	79	7	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15598360	13.89062	100.86656	13.88920	100.86365	16.333333	180	8	38.867	100304	...	4369	2	30	47958	12	13	90	46	8	0.0
15598361	13.90717	100.86414	13.91175	100.86362	29.500000	60	8	38.867	100304	...	4369	2	30	33864	12	9	84	55	1	0.0
15598362	13.89330	100.86513	13.89083	100.86367	23.666667	180	8	38.867	100304	...	4369	3	31	35074	12	9	75	61	8	0.0
15598363	13.90718	100.86463	13.90369	100.86305	17.666667	159	8	38.867	100304	...	4369	3	31	63556	12	17	81	51	1	0.0
15598364	13.90204	100.86260	13.89549	100.86299	22.666667	85	8	38.867	100304	...	4369	3	31	72936	12	20	77	54	7	0.0

ภาพที่ 3.22 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลสภาพอากาศ

จากภาพที่ 3.22 ข้อมูลหลังจากเพิ่มตัวแปรจากชุดข้อมูลสภาพอากาศ มีจำนวน 15,598,365 แถว 26 หลัก มีตัวแปรที่นำไปสร้างต้นแบบเพิ่มขึ้น 9 ตัว ได้แก่ [6] PickupSecofDay, [7] DayofWeek, [8] DayofMonth, [9] Month, [10] Hour, [25] Temperature, [26] Humidity, [27] Wind Speed และ [28] Rain

3.3.6 การประมวลผลชุดข้อมูลสำหรับตัวแปรระยะทาง และ ตัวแปรทิศทาง

ในข้อมูลนี้จะทำการสร้างตัวแปรระยะทาง 2 ตัว และ ตัวแปรทิศทาง 1 ตัวโดยตัวแปรระยะทางแรก [11] Distance คือ ระยะทางที่ได้จากการใช้เครื่องมือ OSMnx เป็นการสร้างกราฟจากแอพพลิเคชันแผนที่ OpenStreetMap แล้วพล็อตจุดพิกัดละติจูด ลองติจูด รับ และส่งผู้โดยสารให้อยู่ใกล้กับโหนดที่มีในแผนที่ หาระยะทางด้วยวิธี Shortest path จะได้ระยะทางระหว่างพิกัด 2 จุดแบบจำลองเส้นทางการวิ่งที่ใกล้ที่สุดของรถยนต์มา

ตัวแปรระยะทางที่สอง [13] Displacement คือ ระยะการกระจัดระหว่างพิกัด 2 จุด คำนวณด้วยวิธี Haversine ที่เป็นการคำนวณระยะการกระจัดบนทรงกลม (ทรงกลมในที่นี้หมายถึงโลก) สามารถเขียนสมการได้ดังนี้

$$d = 2r \bullet \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + \cos\varphi_1 \bullet \cos\varphi_2 \bullet \sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \quad (3.1)$$

โดยที่ d คือ ระยะการกระจัด

r คือ รัศมีของทรงกลม

φ_1, φ_2 คือ ละติจูดที่รับผู้โดยสาร และ ละติจูดที่ส่งผู้โดยสาร

λ_1, λ_2 คือ ลองติจูดที่รับผู้โดยสาร และ ลองติจูดที่ส่งผู้โดยสาร

ตัวแปรทิศทาง [14] Direction คือ ทิศทาง หรือมุมแบริงส์ ระหว่างพิกัด 2 จุด สามารถเขียนสมการได้ดังนี้

$$\beta = arctan2(X, Y) \quad (3.2)$$

$$X = \cos\theta b \bullet \sin\Delta L \quad (3.3)$$

$$Y = \cos\theta a \bullet \sin\theta b - \sin\theta a \bullet \cos\theta b \bullet \cos\Delta L \quad (3.4)$$

โดยที่ β คือ แบริงส์

L คือ ลองติจูด

θ คือ ละติจูด

a, b คือ จุดรับผู้โดยสาร และ จุดส่งผู้โดยสาร

เมื่อเพิ่มตัวแปรแล้ว จะได้ข้อมูลสุดท้ายที่มีตัวแปรครบทั้งหมด ก่อนไปขั้นตอนถัดไป ดังแสดงในภาพที่ 3.23

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	AverageSpeed	TravelTime	Distance	Cluster	PickupArea	PickupSubDistr	...	PickupDayofMonth	PickupSecofDay	PickupMonth	Hour	Temperature	Humidity	Wind Speed	Rain	Displacement	Direction
0	13.76747	100.40770	13.63370	100.36327	54.733333	1139	18288.903	8	8.539	101905	...	31	86370	12	23	72	69	3	0.0	15.619978	-162.110526
1	13.77576	100.40927	13.67465	100.40660	52.333333	719	11599.373	8	8.539	101905	...	2	32491	1	9	81	70	6	0.0	11.239557	-178.530206
2	13.77469	100.41561	13.67527	100.40664	44.250000	900	12986.211	8	8.539	101905	...	2	30717	1	8	77	74	7	0.0	11.090413	-174.989930
3	13.76175	100.41558	13.67498	100.40664	37.857143	1020	11219.486	10	8.539	101905	...	4	30228	1	8	79	74	0	0.0	9.690506	-174.283136
4	13.77525	100.42672	13.66921	100.40621	46.533333	1140	14303.327	11	8.539	101905	...	13	42446	1	11	84	79	7	0.0	11.989916	-169.355944
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15598360	13.89062	100.86656	13.88920	100.86365	16.333333	180	638.202	8	38.867	100304	...	30	47958	12	13	90	46	8	0.0	0.351347	-116.687011
15598361	13.90717	100.86414	13.91175	100.86362	29.500000	60	566.160	8	38.867	100304	...	30	33864	12	9	84	55	1	0.0	0.512035	-6.289002
15598362	13.89330	100.86513	13.89083	100.86367	23.666667	180	556.365	8	38.867	100304	...	31	35074	12	9	75	61	8	0.0	0.316455	-150.152470
15598363	13.90718	100.86463	13.90369	100.86305	17.666667	159	398.122	8	38.867	100304	...	31	63556	12	17	81	51	1	0.0	0.423623	-156.276509
15598364	13.90204	100.86260	13.89549	100.86299	22.666667	85	624.614	8	38.867	100304	...	31	72936	12	20	77	54	7	0.0	0.729084	176.692012

ภาพที่ 3.23 ข้อมูลหลังจากเพิ่มตัวแปรจากการประมวลผลครั้งสุดท้าย

จากภาพที่ 3.23 ข้อมูลสุดท้ายก่อนนำไปสู่ขั้นตอนการสร้างต้นแบบ มีจำนวน 15,598,365 แถว 29 หลัก มีตัวแปรที่นำไปสร้างต้นแบบเพิ่มขึ้น 3 ตัว ได้แก่ [11] Distance , [13] Displacement และ [14] Direction

เมื่อได้ข้อมูลสุดท้ายที่ตัวแปรที่ต้องการครบทุกตัวแล้ว ขั้นตอนต่อไปคือการประมวลผลชุดข้อมูลสุดท้ายด้วยวิธีทางสถิติก่อนนำไปสร้างต้นแบบซึ่งจะอธิบายในหัวข้อถัดไป

3.4 การประมวลผลชุดข้อมูลสุดท้าย

ขั้นตอนการประมวลผลชุดข้อมูลสุดท้าย คือ การจัดการกับข้อมูลที่มีอยู่ให้สามารถนำไปสร้างต้นแบบได้ ด้วยวิธีการจัดการรูปแบบของตารางในโปรแกรมเขียนภาษา Python และการจัดการกับข้อมูลที่มีอยู่ให้สามารถสร้างต้นแบบที่มีประสิทธิภาพได้มากขึ้น ด้วยวิธีทางสถิติต่าง ๆ

เริ่มต้นการประมวลผลข้อมูลสุดท้าย ด้วยการทำความสะอาดข้อมูล การทำความสะอาดข้อมูลเป็นขั้นตอนการตรวจสอบความผิดปกติของข้อมูลว่ามีข้อมูลส่วนไหนที่ไม่สมบูรณ์ หรือมีความผิดพลาดหรือไม่ ก่อนที่จะนำข้อมูลไปทำการสร้างต้นแบบ ขั้นตอนแรก ทำการตรวจสอบว่ามีแถวที่ข้อมูลซ้ำกัน และแถวที่ข้อมูลนั้นว่างเปล่าหรือไม่ ดังภาพที่ 3.24

```
df.duplicated().sum()
```

```
0
```

```
df.isna().sum()
```

```
Pickuplat           0
Pickuplon           0
Dropofflat          0
Dropofflon          0
AverageSpeed        0
TravelTime          0
Distance            0
Cluster             0
PickupArea          0
PickupSubDistr      0
PickupDistr         0
PickPplStat         0
PickBuildStat       0
DropoffArea         0
DropoffSubDistr     0
DropoffDistr        0
DropPplStat         0
DropBuildStat       0
PickupDayofWeek     0
PickupDayofMonth    0
PickupSecofDay      0
PickupMonth         0
Hour                0
Temperature         0
Humidity            0
Wind Speed          0
Rain                0
Displacement        0
Direction           0
```

ภาพที่ 3.24 การตรวจสอบข้อมูลที่ซ้ำกัน และข้อมูลที่ว่างเปล่า

จากภาพที่ 3.24 ในส่วนแรกแสดงถึงการตรวจสอบแถวที่ซ้ำกันในข้อมูลพบว่าไม่มีการซ้ำกันในแต่ละแถว และในส่วนที่สองแสดงถึงการตรวจสอบข้อมูลที่ว่างเปล่าโดยบอกเป็นจำนวนข้อมูลที่ว่างเปล่าของแต่ละตัวแปรพบว่าไม่มีข้อมูลที่ว่างเปล่า

ขั้นต่อมา คือ การตรวจสอบข้อมูลค่าผิดปกติ (Outlier) คือ ข้อมูลที่มีค่าสูงหรือต่ำผิดจากกลุ่มโดยดูจากค่าสถิติต่าง ๆ ของแต่ละตัวแปร ดังแสดงในภาพที่ 3.25

	Pickuplat	Pickuplon	Dropofflat	Dropofflon	AverageSpeed	TravelTime
count	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000
mean	13.762949	100.555161	13.763079	100.554677	20.944510	622.746862
std	0.063870	0.090762	0.063040	0.089276	13.034597	733.342353
min	13.507240	100.329500	13.507340	100.329330	0.005291	1.000000
25%	13.717640	100.494210	13.718260	100.494370	11.333333	180.000000
50%	13.757620	100.552910	13.757800	100.552670	19.000000	419.000000
75%	13.803380	100.612760	13.803590	100.611280	28.222222	780.000000
max	13.954960	100.937310	13.955050	100.935410	248.000000	57642.000000

ภาพที่ 3.25 ค่าสถิติของแต่ละตัวแปรในข้อมูล

	Distance	Cluster	PickupArea	PickupSubDistr	PickupDistr	PickPplStat
count	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000
mean	3193.498690	9.502839	13.170949	102559.111155	1025.560476	42157.194086
std	3259.236459	3.662492	11.945647	1347.747800	13.483434	27417.271151
min	0.000000	0.000000	0.144000	100101.000000	1001.000000	1776.000000
25%	981.141000	8.000000	3.485000	101502.000000	1015.000000	22006.000000
50%	2196.896000	10.000000	9.595000	102701.000000	1027.000000	36313.000000
75%	4282.615000	12.000000	19.306000	103701.000000	1037.000000	55590.000000
max	60590.153000	14.000000	84.712000	105005.000000	1050.000000	125133.000000

	PickBuildStat	DropoffArea	DropoffSubDistr	DropoffDistr	DropPplStat	DropBuildStat
count	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000
mean	24717.869569	12.983758	102557.402358	1025.543487	41891.302927	24693.083149
std	15229.261894	11.830042	1343.069121	13.436418	27464.168554	15275.234798
min	698.000000	0.144000	100101.000000	1001.000000	1776.000000	698.000000
25%	13209.000000	3.375000	101502.000000	1015.000000	22006.000000	13209.000000
50%	23630.000000	9.595000	102704.000000	1027.000000	35838.000000	23630.000000
75%	33896.000000	19.306000	103701.000000	1037.000000	54659.000000	33418.000000
max	60746.000000	84.712000	105005.000000	1050.000000	125133.000000	60746.000000

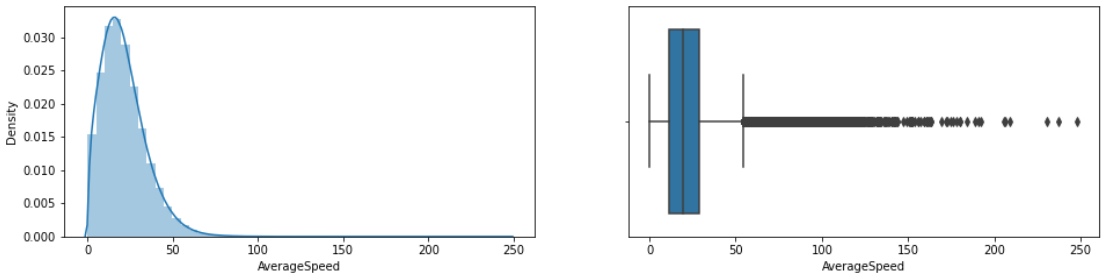
	PickupDayofWeek	PickupDayofMonth	PickupSecofDay	PickupMonth	Hour	Temperature
count	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000
mean	2.978503	15.658885	47774.424758	6.610111	12.772159	85.547431
std	1.970065	8.671423	20602.831981	3.598950	5.721812	6.935580
min	0.000000	1.000000	0.000000	1.000000	0.000000	10.000000
25%	1.000000	8.000000	33532.000000	3.000000	9.000000	82.000000
50%	3.000000	16.000000	48383.000000	7.000000	13.000000	86.000000
75%	5.000000	23.000000	64069.000000	10.000000	17.000000	90.000000
max	6.000000	31.000000	86399.000000	12.000000	23.000000	99.000000

	Humidity	Wind Speed	Rain	Displacement	Direction
count	15598365.000000	15598365.000000	15598365.000000	15598365.000000	15598365.000000
mean	65.995850	4.568534	0.144731	2137.697741	-6.434395
std	16.861157	2.948209	1.602401	2405.310219	103.421202
min	10.000000	0.000000	0.000000	1.078895	-179.994195
25%	55.000000	1.000000	0.000000	592.709288	-94.991663
50%	66.000000	5.000000	0.000000	1357.228859	-3.428017
75%	79.000000	7.000000	0.000000	2788.084788	82.969769
max	94.000000	9.000000	63.000000	52689.027744	180.000000

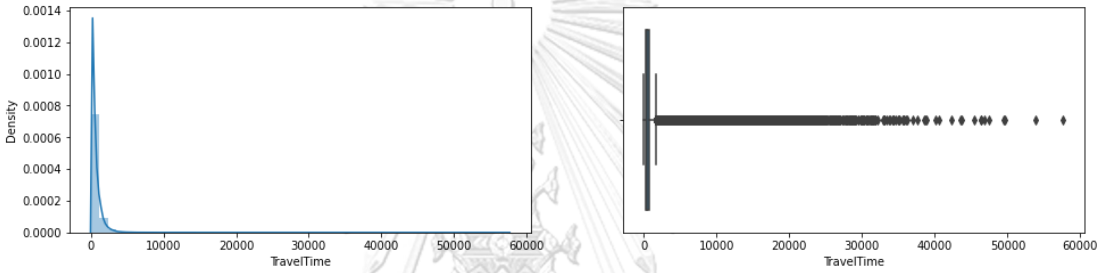
ภาพที่ 3.25 ค่าสถิติของแต่ละตัวแปรในข้อมูล (ต่อ)

จากภาพที่ 3.25 เมื่อสังเกตข้อมูลทางสถิติของแต่ละตัวแปรแล้ว หากดูที่ค่าเฉลี่ยเทียบกับค่าสูงสุด จะเห็นได้ว่าตัวแปร Average Speed, TravelTime, Distance และ Displacement มี

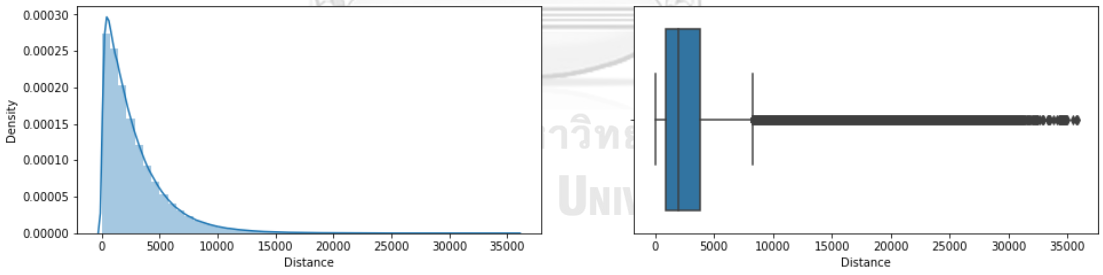
ค่าสูงสุดที่มากกว่าอย่างเห็นได้ชัด ทำการพล็อตกราฟการกระจายตัว และแผนภูมิกล่องแต่ละตัวแปร เพื่อให้สามารถสังเกตข้อมูลที่ผิดปกติได้มากขึ้น ดังแสดงในภาพที่ 3.26, 3.27, 3.28 และ 3.29



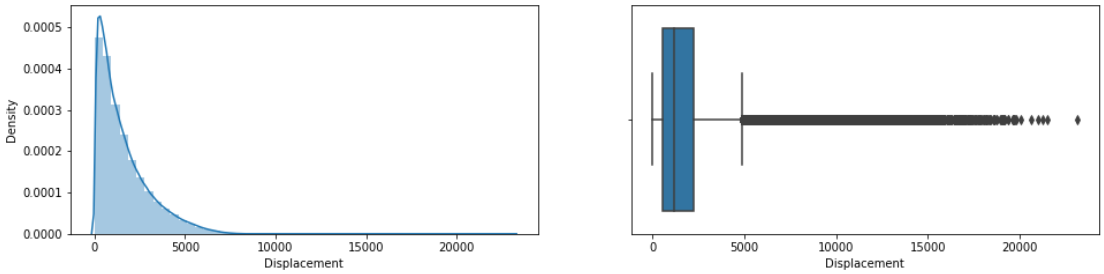
ภาพที่ 3.26 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Average Speed



ภาพที่ 3.27 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร TravelTime



ภาพที่ 3.28 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Distance



ภาพที่ 3.29 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Displacement

จากภาพที่ 3.26-3.29 แสดงให้เห็นว่าแต่ละตัวแปรมีค่าผิดปกอยู่จำนวนหนึ่ง ซึ่งเป็นค่าที่ไม่สามารถหาค้นหาเหตุผลที่ยอมรับได้ จึงทำการใช้วิธีทางสถิติในการกำจัดค่าผิดปกนี้ออก โดยสามารถเขียนสมการได้ดังนี้

$$Upper\ bound = Q3 + 1.5 \bullet IQR \quad (3.5)$$

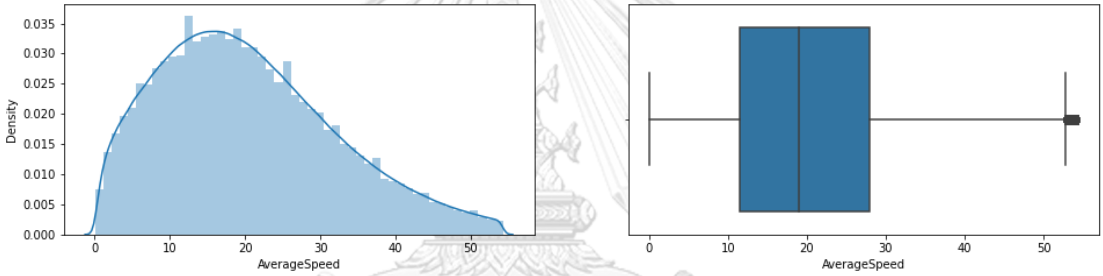
$$Lower\ bound = Q1 - 1.5 \bullet IQR \quad (3.6)$$

$$IQR = Q3 - Q1 \quad (3.7)$$

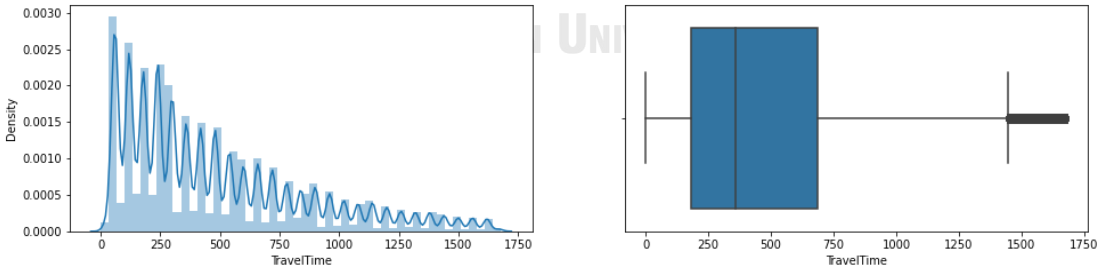
โดยที่ $Q1$ คือ ข้อมูลตำแหน่งควอไทล์ที่ 1

$Q3$ คือ ข้อมูลตำแหน่งควอไทล์ที่ 3

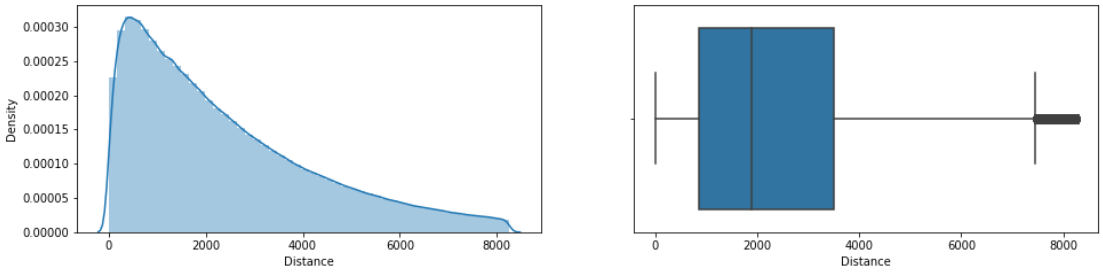
เมื่อทำการลบข้อมูลที่มีค่ามากกว่า Upper bound ในแต่ละตัวแปรออก จะสามารถสร้างกราฟการกระจายตัว และ แผนภูมิกล่องใหม่อีกครั้งดังแสดงในภาพที่ 3.30, 3.31, 3.32 และ 3.33



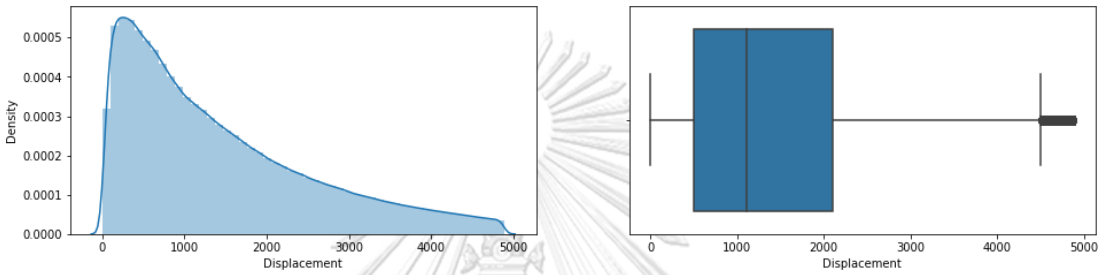
ภาพที่ 3.30 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Average Speed *หลังลบค่าผิดปกติ*



ภาพที่ 3.31 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Travel Time *หลังลบค่าผิดปกติ*



ภาพที่ 3.32 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Distance *หลังลบค่าผิดปกติ*



ภาพที่ 3.33 กราฟการกระจายตัวและแผนภูมิกล่องของตัวแปร Displacement *หลังลบค่าผิดปกติ*

จากภาพที่ 3.30-3.33 ภาพแผนภูมิกล่องแสดงให้เห็นว่าตัวแปรแต่ละตัวมีค่าผิดปกติที่ลดลงอย่างมาก หากเทียบกับก่อนลบค่าผิดปกติ สามารถแสดงค่าสถิติของข้อมูลหลังลบค่าผิดปกติได้ดังภาพที่ 3.34

	TravelTime	Average Speed	Distance	Displacement
count	13050623.000000	13050623.000000	13050623.000000	13050623.000000
mean	438.332973	19.435932	2233.583844	1438.705517
std	354.035178	11.294609	1729.074091	1158.706078
min	1.000000	0.045455	0.000000	1.078895
25%	180.000000	10.800000	836.883000	507.459906
50%	343.000000	18.000000	1808.140000	1114.229363
75%	600.000000	26.500000	3268.309000	2104.519683
max	1679.000000	54.266667	8263.553000	4887.860566

ภาพที่ 3.34 ค่าสถิติของตัวแปรหลังลบค่าผิดปกติ

จากภาพที่ 3.34 เมื่อแก้ไขค่าผิดปกติแล้ว ต่อมาจึงดูที่การกระจายตัวของตัวแปรแต่ละตัว แต่อัลกอริทึม Tree based นั้นเป็น Non Parametric method ลักษณะการกระจายตัวของข้อมูล

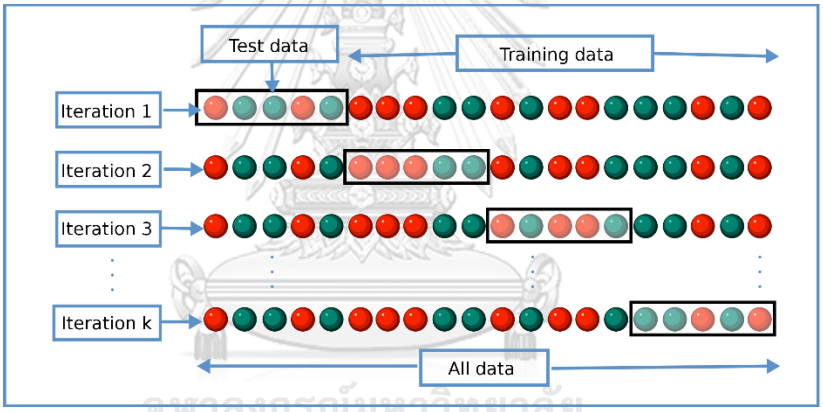
นั้น ไม่มีผลกับหลักการในการแบ่งโหนดของต้นไม้ตัดสินใจ ในงานวิจัยนี้ที่ใช้อัลกอริทึม Tree based ในการสร้างต้นแบบ จึงไม่ได้ทำการเปลี่ยนแปลงข้อมูลใด ๆ เพื่อปรับลักษณะการกระจายตัวของตัวแปร

เมื่อทำการประมวลผลข้อมูลชุดสุดท้ายแล้ว ทำให้ข้อมูลสามารถนำไปสร้างต้นแบบได้โดย งานวิจัยนี้จะใช้อัลกอริทึม 3 แบบในการสร้างต้นแบบได้แก่ LightGBM, XGBoost และ Random forest และใช้วิธี K-Fold cross validation ในการลดปัญหา Overfitting

บทที่ 4
สรุปผลการทดลอง

4.1 ผลลัพธ์การสร้างต้นแบบ

การสร้างต้นแบบจากอัลกอริทึม LightGBM, XGBoost, CatBoost และ Random forest ด้วยวิธีแบ่งข้อมูลแบบ K-Fold cross validation จะทำการแบ่งข้อมูลเป็นข้อมูลสำหรับสร้างต้นแบบ (train set) 80 ส่วน และข้อมูลสำหรับทดสอบประสิทธิภาพของต้นแบบ (test set) 20 ส่วน โดยใช้ค่า K เท่ากับ 5 คือการทำการสร้าง และทดสอบต้นแบบ 5 ครั้ง โดยทำการสลับข้อมูลสำหรับสร้างต้นแบบ และข้อมูลสำหรับทดสอบประสิทธิภาพไปแบบไม่ซ้ำกัน ตัวอย่างการแบ่งข้อมูลดังแสดงในภาพที่ 4.1



ภาพที่ 4.1 ตัวอย่างการแบ่งข้อมูลด้วยวิธี K-Fold cross validation (Gufosowa, 2017)

จากภาพที่ 4.1 เมื่อทำการสร้างต้นแบบ และวัดประสิทธิภาพต้นแบบจนครบ 5 ครั้งแล้ว นำผลจากการวัดประสิทธิภาพทั้งหมดมาเฉลี่ยกัน จะได้ประสิทธิภาพของต้นแบบนั้น ๆ ออกมา

ในการสร้างต้นแบบด้วยวิธีการเรียนรู้ของเครื่อง ผู้วิจัยได้ใช้คอมพิวเตอร์ที่มีหน่วยประมวลผล AMD Ryzen7 3700X 8-Core Processor และ Memory 64 GB โดยใช้โปรแกรม Jupyter Notebook เขียนโค้ดด้วยภาษา Python

การวิเคราะห์ผลลัพธ์ของการทดลองในบทนี้ จะแบ่งเป็นผลการทดลองแต่ละอัลกอริทึมโดยเรียงจาก LightGBM, XGBoost, CatBoost และ Random forest ตามลำดับ และการสรุปผลการทดลองเปรียบเทียบทั้ง 4 อัลกอริทึมในหัวข้อถัดไป

4.1.1 ผลลัพธ์การสร้างต้นแบบของอัลกอริทึม LightGBM

การทดสอบประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม LightGBM ครั้งแรกและหลังจากทำการปรับปรุง Hyperparameters ด้วยเครื่องมือ RandomizedSearchCV จำนวน 100 ครั้ง สุดท้ายนำต้นแบบที่มีประสิทธิภาพมากที่สุดมาทดสอบประสิทธิภาพด้วยการคาดการณ์ชุดข้อมูลทั้งหมด จะได้ผลลัพธ์ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม LightGBM

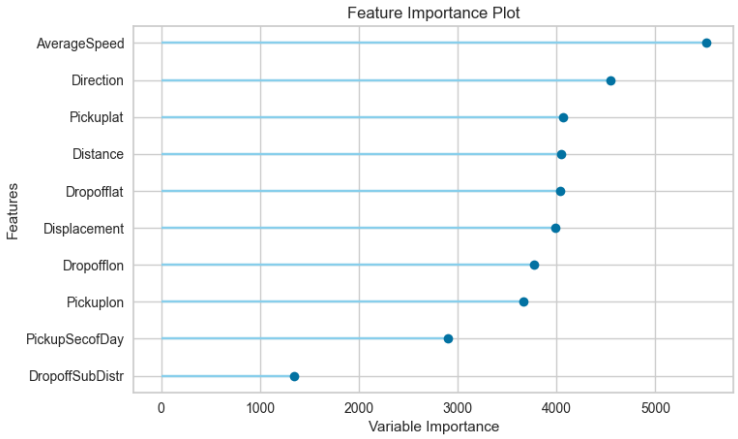
สถานะ	Validation	MAE	RMSE	R^2	MAPE	Time (sec)
ต้นแบบเริ่มต้น	0	150.6844	222.5604	62.95%	53.44%	107.61
	1	150.8411	222.3919	62.98%	53.56%	
	2	150.6316	222.0524	63.07%	53.42%	
	3	150.4431	221.8319	63.13%	53.5%	
	4	150.8734	222.472	63%	53.54%	
	Mean	150.6947	222.2617	63.03%	53.49%	
	SD	0.1554	0.2753	0.0007	0.0006	
ต้นแบบหลังจากปรับปรุง	0	136.5278	206.747	68.03%	43.35%	15292.28
	1	136.3644	206.3675	68.12%	43.23%	
	2	136.6494	206.5997	68.03%	43.55%	
	3	136.4314	206.2784	68.12%	43.5%	
	4	136.7972	206.8567	68.02%	43.46%	
	Mean	136.5541	206.5699	68.06%	43.42%	
	SD	0.1549	0.2193	0.0005	0.0011	
ทดสอบประสิทธิภาพ		135.6372	205.3272	68.45%	43.1%	428.97

จากตารางที่ 4.1 ประสิทธิภาพของต้นแบบจากอัลกอริทึม LightGBM มีค่าเฉลี่ยของความผิดพลาดสัมบูรณ์ (MAE) เท่ากับ 135.6372 ค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) เท่ากับ 43.1% และค่า R^2 เท่ากับ 70.11% สามารถแสดงข้อมูลการปรับปรุง Hyperparameters ได้ดังตารางที่ 4.2

ตารางที่ 4.2 ข้อมูลการปรับปรุง Hyperparameters ของอัลกอริทึม LightGBM

Hyperparameters	ขอบเขต	ผลลัพธ์ที่ดีที่สุด
reg_lambda	[1.00E-07, 1.00E+01]	1.00E+00
reg_alpha	[1.00E-07, 1.00E+01]	7.00E-01
num_leaves	[2, 256]	200
n_estimators	[10, 300]	250
min_split_gain	[0, 0.9]	0.1
min_child_samples	[1, 100]	26
learning_rate	[1.00E-07, 5.00E-1]	5.00E-01
feature_fraction	[0.4, 1]	0.9
bagging_freq	[0, 7]	4
bagging_fraction	[0.4, 1]	0.8

จากตารางที่ 4.2 สามารถแสดงอันดับตัวแปรสำคัญ 10 ลำดับที่ส่งผลต่อต้นแบบที่สร้างจากอัลกอริทึม LightGBM ดังภาพที่ 4.2



ภาพที่ 4.2 อันดับตัวแปรสำคัญที่ส่งผลต่อต้นแบบ LightGBM

จากภาพที่ 4.2 อันดับตัวแปร 10 ลำดับที่ส่งผลต่อต้นแบบ LightGBM ได้แก่ Average Speed, Direction, Pickuplat, Distance, Dropofflat, Displacement, Dropofflon, Pickuplon, PickupSecofDay และ DropoffSubDistr ตามลำดับ

4.1.2 ผลลัพธ์การสร้างต้นแบบของอัลกอริทึม XGBoost

การทดสอบประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม XGBoost ครั้งแรกและหลังจากทำการปรับปรุง Hyperparameters ด้วยเครื่องมือ RandomizedSearchCV จำนวน 100 ครั้ง สุดท้ายนำต้นแบบที่มีประสิทธิภาพมากที่สุดมาทดสอบประสิทธิภาพด้วยการคาดการณ์ชุดข้อมูลทั้งหมด จะได้ผลลัพธ์ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม XGBoost

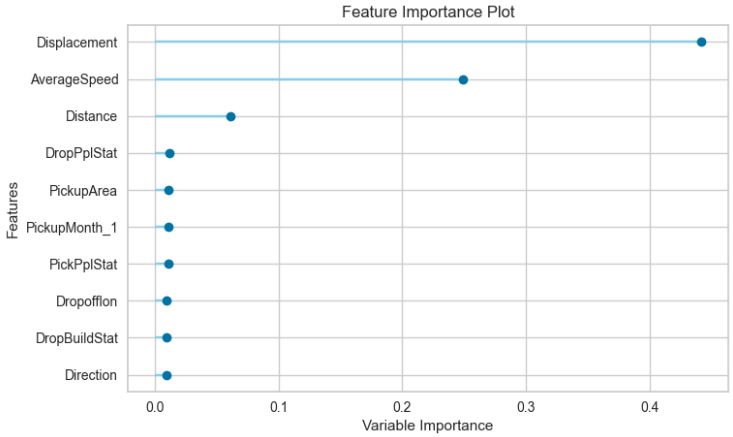
สถานะ	Validation	MAE	RMSE	R^2	MAPE	Time (sec)
ต้นแบบเริ่มต้น	0	145.7466	217.1616	64.73%	49.61%	122.65
	1	145.5842	216.6098	64.88%	49.58%	
	2	145.9796	216.862	64.78%	50.03%	
	3	145.4597	216.4251	64.91%	49.74%	
	4	145.7261	216.7915	64.87%	49.65%	
	Mean	145.6992	216.77	64.83%	49.72%	
	SD	0.1744	0.2477	0.0007	0.0017	
ต้นแบบหลังจากปรับปรุง	0	135.994	205.5312	68.4%	43.44%	10219.99
	1	136.161	205.3464	68.44%	43.41%	
	2	135.8945	204.9653	68.54%	43.39%	
	3	135.8934	205.0492	68.5%	43.57%	
	4	136.1886	205.293	68.5%	43.48%	
	Mean	136.0263	205.237	68.48%	43.46%	
	SD	0.1269	0.2053	0.0005	0.0006	
ทดสอบประสิทธิภาพ		135.3859	204.5078	68.7%	43.15%	224.84

จากตารางที่ 4.3 ประสิทธิภาพของต้นแบบจากอัลกอริทึม XGBoost มีค่าเฉลี่ยของความผิดพลาดสัมบูรณ์ (MAE) เท่ากับ 135.3859 ค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) เท่ากับ 43.15% และค่า R^2 เท่ากับ 68.7% สามารถแสดงข้อมูลการปรับปรุง Hyperparameters ได้ดังตารางที่ 4.4

ตารางที่ 4.4 ข้อมูลการปรับปรุง Hyperparameters ของอัลกอริทึม XGBoost

Hyperparameters	ขอบเขต	ผลลัพธ์ที่ดีที่สุด
subsample	[0.2, 1]	1
scale_pos_weight	[0.3, 48.2]	23.5
reg_lambda	[1.00E-07, 1.00E+01]	5.00E+00
reg_alpha	[1.00E-07, 1.00E+01]	4.00E-01
n_estimators	[10, 300]	270
min_child_weight	[1, 4]	1
max_depth	[1, 11]	8
learning_rate	[1.00E-07, 5.00E-01]	5.00E-01
colsample_bytree	[0.5, 1]	0.7

จากตารางที่ 4.4 สามารถแสดงอันดับตัวแปรสำคัญ 10 ลำดับที่ส่งผลต่อต้นแบบที่สร้างจากอัลกอริทึม XGBoost ดังภาพที่ 4.3



ภาพที่ 4.3 อันดับตัวแปรสำคัญที่ส่งผลต่อต้นแบบ XGBoost

จากภาพที่ 4.3 อันดับตัวแปร 10 ลำดับที่ส่งผลต่อต้นแบบ XGBoost ได้แก่ Displacement, Average Speed, Distance, DropPplStat, PickupArea, PickupMonth_1, PickupPplStat, Dropofflon, DropoffBuildStat และ Direction ตามลำดับ

4.1.3 ผลลัพธ์การสร้างต้นแบบของอัลกอริทึม CatBoost

การทดสอบประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม CatBoost ครั้งแรกและหลังจากทำการปรับปรุง Hyperparameters ด้วยเครื่องมือ RandomizedSearchCV จำนวน 100 ครั้ง สุดท้ายนำต้นแบบที่มีประสิทธิภาพมากที่สุดมาทดสอบประสิทธิภาพด้วยการคาดการณ์ชุดข้อมูลทั้งหมด จะได้ผลลัพธ์ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 ประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม CatBoost

สถานะ	Validation	MAE	RMSE	R^2	MAPE	Time (sec)
ต้นแบบเริ่มต้น	0	148.5119	220.2678	63.71%	52.62%	229.02
	1	148.4452	219.8279	63.83%	52.54%	
	2	148.358	219.5742	63.89%	52.54%	
	3	148.2264	219.5103	63.9%	52.64%	
	4	148.5621	219.9332	63.84%	52.59%	
	Mean	148.4207	219.8227	63.84%	52.58%	
	SD	0.1188	0.2719	0.0007	0.0004	
ต้นแบบหลังจากปรับปรุง	0	148.7824	220.5636	63.61%	52.7%	2982.72
	1	148.5941	220.0161	63.77%	52.55%	
	2	148.5134	219.8054	63.82%	52.56%	
	3	148.5454	219.8464	63.79%	52.71%	
	4	148.7867	220.1947	63.76%	52.63%	
	Mean	148.6444	220.0852	63.75%	52.63%	
	SD	0.1173	0.276	0.0007	0.0007	
ทดสอบประสิทธิภาพ		148.412	219.8107	63.85%	52.55%	90.72

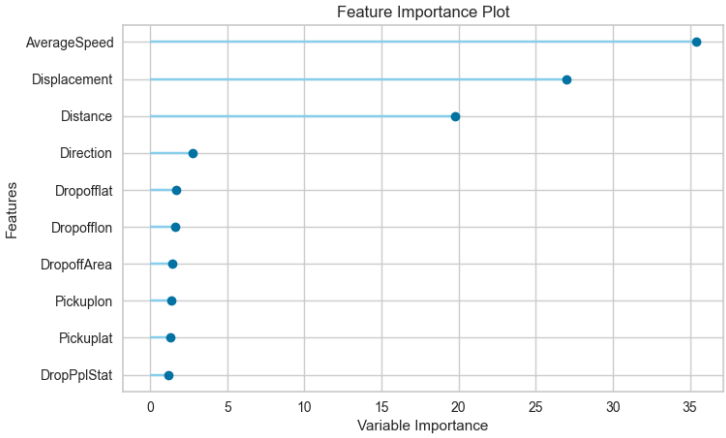
จากตารางที่ 4.5 เนื่องจากหลังปรับปรุง Hyperparameters แล้วประสิทธิภาพของต้นแบบไม่สามารถพัฒนาให้ดีกว่าเดิมได้จึงใช้ต้นแบบเริ่มต้นในการทดสอบประสิทธิภาพ โดยประสิทธิภาพของต้นแบบจากอัลกอริทึม CatBoost มีค่าเฉลี่ยของความผิดพลาดสัมบูรณ์ (MAE) เท่ากับ 148.412 ค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) เท่ากับ

52.55% และค่า R^2 เท่ากับ 63.85% สามารถแสดงข้อมูลการปรับปรุง Hyperparameters ได้ดังตารางที่ 4.6

ตารางที่ 4.6 ข้อมูลการปรับปรุง Hyperparameters ของอัลกอริทึม CatBoost

Hyperparameters	ขอบเขต	ผลลัพธ์ที่ดีที่สุด (เริ่มต้น)
random_strength	[0, 0.8]	1
n_estimators	[10, 300]	100
l2_leaf_reg	[1, 200]	3
eta	[1.00E-07, 5.00E-01]	4.00E-01
depth	[1, 8]	6

จากตารางที่ 4.6 ในขอบเขตแสดงถึงข้อมูลในขั้นตอนการปรับปรุง Hyperparameters แต่เนื่องจากไม่สามารถพัฒนาต้นแบบได้ ในผลลัพธ์ที่ดีที่สุดจะใส่เป็นค่าของต้นแบบเริ่มต้นที่มีประสิทธิภาพมากที่สุด สามารถแสดงอันดับตัวแปรสำคัญ 10 ลำดับที่ส่งผลต่อต้นแบบที่สร้างจากอัลกอริทึม CatBoost ดังภาพที่ 4.4



ภาพที่ 4.4 อันดับตัวแปรสำคัญที่ส่งผลต่อต้นแบบ CatBoost

จากภาพที่ 4.4 อันดับตัวแปร 10 ลำดับที่ส่งผลต่อต้นแบบ CatBoost ได้แก่ Average Speed, Displacement, Distance, Direction, Dropofflat, Dropofflon, DropoffArea, Pickuplon, Pickuplat, DropPplStat ตามลำดับ

4.1.4 ผลลัพธ์การสร้างต้นแบบของอัลกอริทึม Random forest

สำหรับต้นแบบ Random forest นั้นหากทำการสร้างต้นแบบด้วย Hyperparameters เริ่มต้นจะใช้เวลาในการสร้างต้นแบบนานถึง 18 ชั่วโมง และปรับปรุง Hyperparameters มากกว่า 3 วัน (อุปกรณ์ของผู้วิจัยไม่สามารถหาคำตอบได้) ซึ่งหากเปรียบเทียบเวลากับต้นแบบอื่นแล้วใช้เวลานานกว่าเกินไป ผู้วิจัยจึงทำการปรับปรุง Hyperparameters n_estimators ให้มีค่าเท่ากับ 10 ในต้นแบบเริ่มต้นเพื่อลดเวลาในการสร้างต้นแบบ แต่อุปกรณ์ก็ยังไม่สามารถทำขั้นตอนปรับปรุง Hyperparameters ได้ (มีอาการค้างแม้ปรับ n_jobs ให้น้อยลงแล้ว) จึงทำให้ต้นแบบ Random forest สามารถสร้างได้เพียงต้นแบบเริ่มต้น และทดสอบประสิทธิภาพกับชุดข้อมูลทั้งหมดเท่านั้น ผลลัพธ์ที่ได้ดังแสดงในตารางที่ 4.7

ตารางที่ 4.7 ประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม Random forest

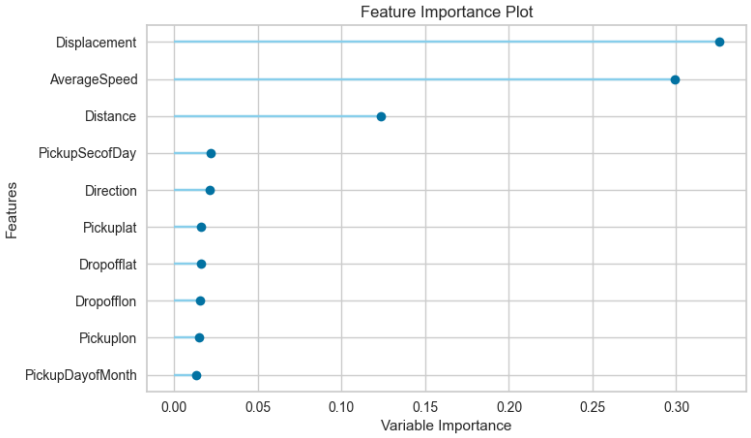
สถานะ	Validation	MAE	RMSE	R^2	MAPE	Time (sec)
ต้นแบบเริ่มต้น	0	137.5031	207.2725	0.6786	0.4406	2885.02
	1	137.7402	207.1536	0.6788	0.4422	
	2	137.6426	206.8789	0.6795	0.442	
	3	137.3218	206.7648	0.6797	0.4415	
	4	138.0455	207.3684	0.6786	0.4447	
	Mean	137.6506	207.0876	0.679	0.4422	
	SD	0.2425	0.2303	0.0005	0.0014	
ทดสอบประสิทธิภาพ		134.7196	203.7001	0.6895	0.4306	4591.16

จากตารางที่ 4.7 ประสิทธิภาพของต้นแบบจากอัลกอริทึม Random forest มีค่าเฉลี่ยของความผิดพลาดสัมบูรณ์ (MAE) เท่ากับ 134.7196 ค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) เท่ากับ 43.06% และค่า R^2 เท่ากับ 68.95% สามารถแสดงค่าของ Hyperparameters ของต้นแบบเริ่มต้นได้ดังตารางที่ 4.8

ตารางที่ 4.8 ข้อมูล Hyperparameters ของอัลกอริทึม Random forest

Hyperparameters	ผลลัพธ์ที่ดีที่สุด
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	10
random_state	123

จากตารางที่ 4.8 สามารถแสดงอันดับตัวแปรสำคัญ 10 ลำดับที่ส่งผลต่อต้นแบบที่สร้างจากอัลกอริทึม Random forest ดังภาพที่ 4.5



ภาพที่ 4.5 อันดับตัวแปรสำคัญที่ส่งผลต่อต้นแบบ Random forest

จากภาพที่ 4.5 อันดับตัวแปร 10 ลำดับที่ส่งผลต่อต้นแบบ Random forest ได้แก่ Displacement, Average Speed, Distance, PickupSecofDay, Direction, Pickuplat, Dropofflat, Dropofflon, Pickuplon และ PickupDayofMonth ตามลำดับ

4.2 สรุปผลการทดลอง

ในงานวิจัยนี้ การทดสอบประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึมต่าง ๆ มีปัจจัยในการประเมินคือ ประสิทธิภาพของต้นแบบ เวลาที่ใช้ในการสร้างต้นแบบทั้งต้นแบบเริ่มต้นและการปรับปรุง Hyperparameters สุดท้ายคือการทดสอบประสิทธิภาพต้นแบบกับข้อมูลที่ต้นแบบไม่เคยเห็นมาก่อนเพื่อรับรอง Robustness ให้กับต้นแบบ

เริ่มต้นจากประสิทธิภาพของต้นแบบซึ่งจะพิจารณาจากเมตริก MAE ที่บ่งบอกถึงค่าเฉลี่ยของค่าความผิดพลาดระหว่างค่าที่คาดการณ์จากต้นแบบกับค่าจริงของข้อมูล และ เมตริก MAPE ที่บ่งบอกถึงค่าเฉลี่ยของค่าความผิดพลาดในรูปแบบร้อยละ ยิ่งเมตริกทั้ง 2 ตัวมีค่าน้อย หมายความว่าต้นแบบนั้นมีประสิทธิภาพดี เปรียบเทียบเมตริกโดยใช้ผลลัพธ์จากการทดสอบครั้งสุดท้ายด้วยชุดข้อมูลทั้งหมด การเปรียบเทียบประสิทธิภาพของต้นแบบดังแสดงในตารางที่ 4.9

ตารางที่ 4.9 การเปรียบเทียบประสิทธิภาพของต้นแบบ

ต้นแบบ	MAE	MAPE
LightGBM	135.6372	43.1%
XGBoost	135.3859	43.15%
CatBoost	148.412	52.55%
Random forest	134.7196	43.06%

จากตารางที่ 4.9 ต้นแบบจากอัลกอริทึม CatBoost มีประสิทธิภาพน้อยอย่างเห็นได้ชัด ซึ่งหากเทียบกับประสิทธิภาพของต้นแบบจากอัลกอริทึม LightGBM, XGBoost และ Random forest นั้นมีค่า MAPE ใกล้เคียงร้อยละ 40 ถ้าตัดสินด้วยประสิทธิภาพของต้นแบบ Random forest จะเป็นต้นแบบที่มีประสิทธิภาพสูงสุด มีค่าเฉลี่ยของค่าความผิดพลาดอยู่ที่ 134.7196 วินาที หรือคิดเป็นร้อยละ 43.06

ต่อไปเป็นการเปรียบเทียบในด้านเวลาซึ่งเป็นอีกหนึ่งตัวแปรสำคัญของการสร้างต้นแบบ แม้ว่าในการแข่งขันปกติแล้วจะพิจารณาเพียงแค่ประสิทธิภาพของต้นแบบเท่านั้น แต่สำหรับงานวิจัยนี้ ใช้ชุดข้อมูลที่ค่อนข้างใหญ่ทำให้การประมวลผลใช้เวลานานกว่าปกติ เพราะฉะนั้นเวลาในการสร้างต้นแบบจึงมีผลเป็นอย่างมากในการค้นหาความเป็นไปได้ต่าง ๆ จากข้อมูลปริมาณมาก สามารถแสดงถึงเวลาในการสร้างต้นแบบเริ่มต้น และเวลาที่ใช้ในการปรับปรุง Hyperparameters 100 ครั้ง ในหน่วยวินาที ดังตารางที่ 4.10

ตารางที่ 4.10 การเปรียบเทียบด้านเวลาของต้นแบบ

ต้นแบบ	สร้างต้นแบบเริ่มต้น	ปรับปรุง Hyperparameters
LightGBM	107.61	15292.28
XGBoost	122.65	10219.99
CatBoost	229.02	2982.72
Random forest	2885.02	-

จากตารางที่ 4.10 ในด้านเวลาของการสร้างต้นแบบ หากเทียบกันระหว่างอัลกอริทึมที่มีประสิทธิภาพเป็นอันดับสอง และอันดับสามอย่าง LightGBM และ XGBoost นั้น LightGBM ที่มีค่า MAPE มากกว่าเพียงร้อยละ 0.41 แต่ใช้เวลาในการสร้างรวมปรับปรุง Hyperparameters มากกว่าถึง 5072.29 วินาทีซึ่งในส่วนต่างนี้ XGBoost อาจจะสามารถทำการปรับปรุง Hyperparameters ได้ถึงอีก 50 ครั้งเพราะฉะนั้นหากพิจารณาทั้งด้านประสิทธิภาพ และเวลาแล้ว XGBoost จะเป็นทางเลือกที่ดีกว่า LightGBM หากต้องการพัฒนาประสิทธิภาพของต้นแบบต่อเมื่อมีทรัพยากรคอมพิวเตอร์ที่จำกัด ในส่วนของ Random forest นั้น แม้ว่าจะใช้เวลาสร้างต้นแบบเริ่มต้นนานกว่าอัลกอริทึมอื่นหลายเท่าตัวมากแต่กลับมีประสิทธิภาพที่สูงที่สุด หากเทียบกับเวลาทั้งหมดที่ XGBoost ใช้ Random forest นั้นใช้เวลาในการสร้างต้นแบบที่มีประสิทธิภาพได้ใกล้เคียงกันน้อยถึงกว่า 3.5 เท่าโดยที่ไม่ต้องปรับปรุง Hyperparameters เลย นั่นอาจหมายถึงชุดข้อมูลนี้มีความเหมาะสมกับอัลกอริทึม Random forest มากที่สุด

สำหรับ CatBoost นั้นแม้ว่าจะใช้เวลาในการสร้างต้นแบบมาก แล้วยังมีประสิทธิภาพต่ำที่สุดในทุก ๆ อัลกอริทึม GBDTs แต่หากสังเกตเวลาในการปรับปรุง Hyperparameters ที่น้อยกว่า XGBoost หรือ LightGBM มาก ทำให้สามารถปรับปรุง Hyperparameters ในจำนวนครั้งมาก ๆ ได้เทียบกับอัลกอริทึม Random forest ที่ไม่สามารถปรับปรุงได้ อาจจะมีความเป็นไปได้ที่จะมีการพัฒนาประสิทธิภาพมากกว่า

สุดท้ายเพื่อการทดสอบประสิทธิภาพของต้นแบบอย่างเหมาะสม ผู้วิจัยได้นำข้อมูลการเดินทางบนท้องถนนในกรุงเทพมหานครที่ไม่ได้อยู่ใน train set หรือ test set เป็นข้อมูลที่ผู้เรียนรู้ไม่เคยเห็นมาก่อน จำนวน 1.2 ล้านแถว มาทำการทดสอบต้นแบบจากอัลกอริทึมทั้งหมด โดยได้ผลลัพธ์ดังแสดงในตารางที่ 4.11

ตารางที่ 4.11 ประสิทธิภาพของต้นแบบเมื่อลองใช้กับข้อมูลที่ผู้เรียนรู้ไม่เคยเห็น

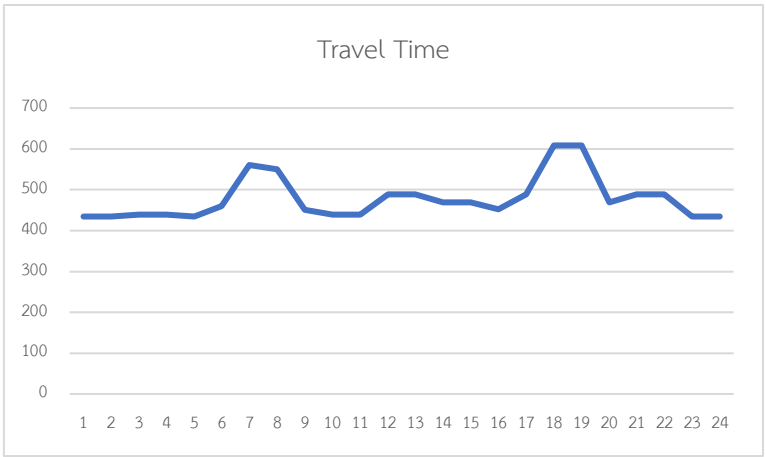
ต้นแบบ	MAE	RMSE	R^2	MAPE
LightGBM	134.7616	203.7133	68.89%	42.67%
XGBoost	135.0723	203.6289	68.91%	43.05%
CatBoost	148.421	219.4036	63.91%	52.6%
Random forest	132.7027	201.0857	69.68%	42.21%

จากตารางที่ 4.11 แสดงให้เห็นว่าประสิทธิภาพของต้นแบบนั้นไม่ได้มีการเปลี่ยนแปลงจากการทดสอบครั้งสุดท้ายไปมาก หมายความว่าต้นแบบจากอัลกอริทึมเหล่านี้มีความ Robustness และสามารถนำไปใช้กับข้อมูลที่ผู้เรียนรู้ไม่เคยเห็นได้นั่นเอง

ในงานวิจัยนี้แสดงให้เห็นถึงความเป็นไปได้ในการพัฒนาต้นแบบการคาดการณ์ระยะเวลาเดินทางบนท้องถนนด้วยวิธีการเรียนรู้ของเครื่อง โดยชุดข้อมูลที่สามารถเข้าถึงได้แบบสาธารณะ สรุปได้ว่าอัลกอริทึม Random forest สามารถสร้างต้นแบบจากชุดข้อมูลนี้ได้มีประสิทธิภาพมากที่สุด

4.3 การใช้งานต้นแบบ

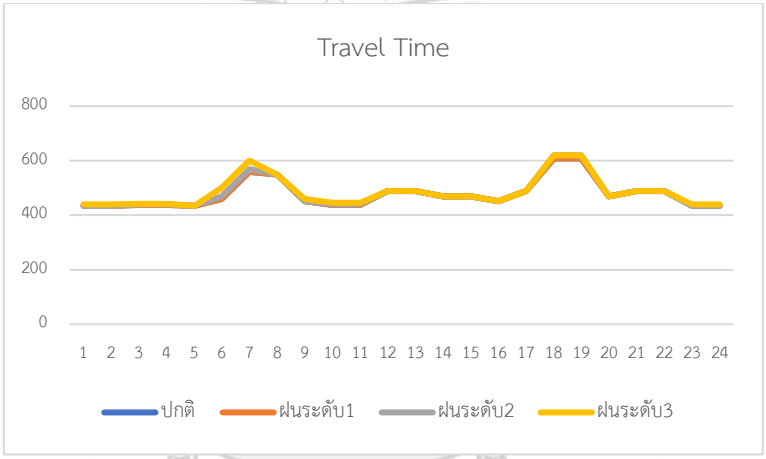
หัวข้อนี้จะทำการนำต้นแบบที่สร้างขึ้นในงานวิจัยมาทดสอบการใช้งานในรูปแบบต่าง ๆ โดยใช้ต้นแบบที่มีประสิทธิภาพสูงสุดจากอัลกอริทึม Random forest ซึ่งสถานที่จุดเริ่มต้นจากห้างสรรพสินค้าสามย่านมิตรทาวน์ไปจุดสิ้นสุดที่ห้างสรรพสินค้าสยามพารากอนเป็นระยะทาง 2.5 กิโลเมตร เริ่มจากการทดสอบความแตกต่างของระยะเวลาในการเดินทางในช่วงเวลาระหว่างวันที่แตกต่างกันออกไป ดังแสดงในภาพที่ 4.6



ภาพที่ 4.6 ทดสอบความแตกต่างของระยะเวลาเดินทางในแต่ละช่วงเวลา

จากภาพที่ 4.6 เป็นกราฟระยะเวลาเดินทางในแต่ละช่วงเวลาของวันจันทร์ที่ 9 ในเดือนพฤศจิกายน สภาพอากาศปกติ จะสังเกตได้ว่าระยะเวลาเดินทางในช่วงเช้า และช่วงเย็นใช้เวลาในการเดินทางมากกว่าช่วงเวลาอื่น ๆ อาจเนื่องมาจากเป็นเวลาเข้างาน เข้าเรียน หรือเลิกงาน เลิกเรียน จึงทำให้การจราจรในช่วงเวลานี้หนาแน่นกว่าปกติ

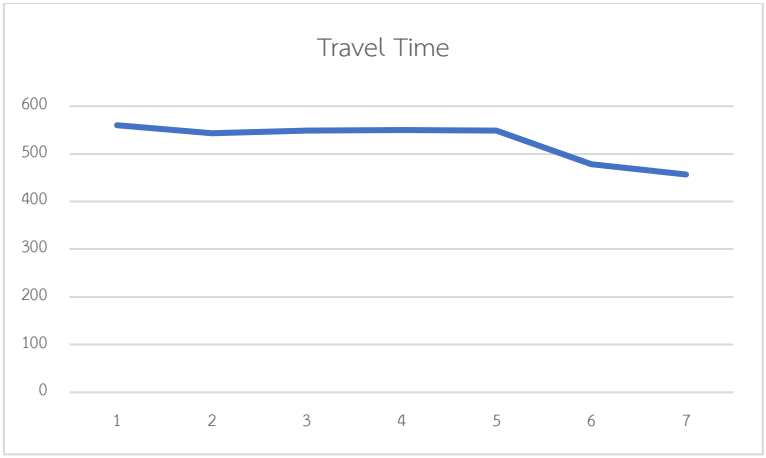
เมื่อทดสอบต่อด้วยการเพิ่มปริมาณน้ำฝนซึ่งเป็นการบ่งบอกถึงสภาพอากาศฝนตก แบ่งระดับของน้ำฝนออกเป็น 3 ระดับโดยใช้เกณฑ์อากาศเดียวกับกรมอุตุนิยมวิทยาของประเทศไทยแบ่งตามปริมาณน้ำฝน ในระดับที่ 1 ฝนเล็กน้อย 10 มิลลิเมตร ระดับที่ 2 ฝนปานกลาง 35 มิลลิเมตร และระดับที่ 3 ฝนหนัก 90 มิลลิเมตร จะได้กราฟความแตกต่างของระยะเวลาเดินทางดังแสดงในภาพที่ 4.7



ภาพที่ 4.7 ทดสอบความแตกต่างของระยะเวลาเดินทางเมื่อฝนตก

จากภาพที่ 4.7 แสดงให้เห็นว่าตัวแปรปริมาณน้ำฝนนั้นมีผลน้อยมาก ในฝนตกระดับที่ 1 และระดับที่ 2 นั้นมีความใกล้เคียงหรือไม่แตกต่างเลยกับตอนที่ฝนไม่ตก และฝนตกระดับที่ 3 นั้นสร้างความแตกต่างในระยะเวลาเดินทางเพียงเล็กน้อย อาจเนื่องมาจากระยะทางที่ไม่มาก และตัวแปรปริมาณน้ำฝนนั้นส่งผลกับต้นแบบที่สร้างจากอัลกอริทึม Random forest น้อยมาก

ทำการทดสอบด้วยการเปลี่ยนวันในสัปดาห์ ตั้งแต่วันที่จันทร์ที่ 9 ถึงวันอาทิตย์ที่ 15 พฤศจิกายน ในเวลา 7 นาฬิกา จะได้ผลลัพธ์ดังแสดงในภาพที่ 4.8



ภาพที่ 4.8 ทดสอบความแตกต่างของระยะเวลาเดินทางในสัปดาห์

จากภาพที่ 4.8 จะสามารถสังเกตได้ถึงความแตกต่างของระยะเวลาที่ชัดเจนระหว่างวันธรรมดากับวันเสาร์-อาทิตย์ แสดงให้เห็นว่าในช่วงเช้าของวันหยุดนั้น การจราจรจะแออัดน้อยกว่าในช่วงเช้าของวันธรรมดา

จากตัวอย่างข้างต้นเป็นการทดลองใช้งานต้นแบบที่สร้างจากอัลกอริทึม Random forest ที่เมื่อผู้ใช้เปลี่ยนแปลงตัวแปรตามการใช้งาน ก็จะแสดงผลลัพธ์ระยะเวลาเดินทางที่แตกต่างกันออกไปด้วยค่าเฉลี่ยความผิดพลาดประมาณ 2 นาที แสดงให้เห็นถึงความสามารถของการใช้งานสำหรับการศึกษาการจัดเส้นทางเดินทางของผู้ให้บริการทางท้องถนนของกรุงเทพมหานครที่มีสภาพการจราจรที่แออัดในแต่ละช่วงเวลาแตกต่างกัน

บทที่ 5
สรุปผลการดำเนินงานวิจัย

ในบทนี้ ผู้วิจัยจะกล่าวถึงบทสรุปของงานวิจัย ประกอบด้วยบทสรุป และข้อเสนอแนะของงานวิจัย การคาดการณ์เวลาเดินทางบนท้องถนนระหว่างพิกัดสองจุดในกรุงเทพมหานคร ด้วยวิธีการเรียนรู้ของเครื่อง

5.1 สรุปผลการดำเนินงานวิจัย

ในงานวิจัยนี้ได้นำเสนอการสร้างต้นแบบในการคาดการณ์เวลาเดินทางบนท้องถนนของกรุงเทพมหานคร เพื่อลดข้อจำกัดในการเรียกเมตริกระยะเวลาเดินทางที่เกิดขึ้นจากแอพพลิเคชั่นผู้ให้บริการแผนที่ต่าง ๆ ได้แก่ ปริมาณการเรียกข้อมูล ขนาดของเมตริก และ ตัวแปรสภาพแวดล้อม เพื่อนำไปใช้ในการศึกษาการจัดเส้นทางของผู้ให้บริการบนท้องถนน หรือผู้ที่สนใจเรื่องระยะเวลาเดินทางบนท้องถนน โดยใช้เทคนิค data-driven methods ซึ่งเป็นหนึ่งในเทคนิคการเพิ่มความน่าเชื่อถือให้กับการคาดการณ์ระยะเวลาเดินทางบนท้องถนน ซึ่งหากรวบรวมข้อมูลได้ในปริมาณมากจะทำให้เทคนิคนี้มีประสิทธิภาพมากขึ้น

การสร้างต้นแบบจะใช้ข้อมูลต่าง ๆ ที่เกี่ยวกับข้อมูลการจราจรของประเทศไทย และข้อมูลที่คาดว่าจะส่งผลต่อระยะเวลาเดินทางบนท้องถนน โดยมีข้อมูลสำคัญคือข้อมูลพาหนะและโทรศัพท์มือถือจาก iTIC foundation ที่แปลงเป็นข้อมูล Origin-Destination pair และข้อมูลอื่น ๆ โดยสรุปแล้วมีตัวแปรทั้งหมด 29 ตัวในข้อมูลสุดท้าย ได้แก่ พิกัดละติจูดที่รับผู้โดยสาร, พิกัดลองติจูดที่รับผู้โดยสารม พิกัดละติจูดที่ส่งผู้โดยสาร, พิกัดลองติจูดที่ส่งผู้โดยสาร, ความเร็วเฉลี่ยของพาหนะ, เวลาที่รับผู้โดยสารนับตั้งแต่, เวลาเริ่มต้นวัน, วันในสัปดาห์ที่รับผู้โดยสาร, วันที่ที่รับผู้โดยสาร, เดือนที่รับผู้โดยสาร, ชั่วโมงที่รับผู้โดยสาร, ระยะทางระหว่างจุดรับและส่งผู้โดยสาร, เวลาระหว่างรับและส่งผู้โดยสาร, ระยะการกระจัดระหว่างจุดรับและส่งผู้โดยสาร, ทิศทางระหว่างจุดรับและส่งผู้โดยสาร, เขตที่รับผู้โดยสาร, เขตที่ส่งผู้โดยสาร, แขวงที่รับผู้โดยสาร, แขวงที่ส่งผู้โดยสาร, พื้นที่แขวงที่รับผู้โดยสาร, พื้นที่แขวงที่ส่งผู้โดยสาร, การแบ่งกลุ่มวันตามรูปแบบดัชนีรถติดที่คล้ายกัน, จำนวนประชากรในแขวงที่รับผู้โดยสาร, จำนวนบ้านในแขวงที่รับผู้โดยสาร, จำนวนประชากรในแขวงที่ส่งผู้โดยสาร, จำนวนบ้านในแขวงที่ส่งผู้โดยสาร, อุณหภูมิ, ความชื้น, ความเร็วลม และ ปริมาณน้ำฝน

การตรวจสอบข้อมูลในขั้นสุดท้ายเพื่อที่จะสามารถสร้างต้นแบบได้อย่างมีประสิทธิภาพ พบว่าไม่มีข้อมูลที่ไม่สมบูรณ์หรือว่างเปล่า แต่มีตัวแปรที่มีค่าผิดปกติได้แก่ ความเร็วเฉลี่ยของพาหนะ, ระยะทางระหว่างจุดรับและส่งผู้โดยสาร, เวลาระหว่างรับและส่งผู้โดยสาร และ ระยะการกระจัด

ระหว่างจุดรับและส่งผู้โดยสาร ทำการแก้ไขด้วยวิธีการหา IQR ทางสถิติ แล้วจะได้ข้อมูลสุดท้ายก่อนที่จะนำไปสร้างต้นแบบ

เมื่อนำข้อมูลไปสร้างต้นแบบขั้นแรกจะเข้าสู่ขั้นตอนการแบ่งข้อมูลด้วย K-fold cross validation ในงานวิจัยนี้กำหนดให้ K เท่ากับ 5 แล้วทำการสร้างต้นแบบ 5 ครั้งตามที่แบ่งข้อมูลไว้ แล้วนำผลลัพธ์ที่ได้มาเฉลี่ยกันเพื่อลดการเกิดปัญหา overfitting จะได้ต้นแบบเริ่มต้นออกมา จากนั้นทำการปรับ Hyperparameters ด้วยวิธี RandomizedSearchCV จำนวน 100 ครั้ง เพื่อเพิ่มประสิทธิภาพของต้นแบบจึงได้ต้นแบบสุดท้าย โดยสามารถเรียงลำดับอัลกอริทึมที่สามารถสร้างต้นแบบสุดท้ายที่มีประสิทธิภาพมากที่สุดจากมากไปน้อยได้ดังนี้ Random forest, XGBoost, LightGBM และ CatBoost โดยมีค่า MAE อยู่ที่ 134.7196, 135.6372, 135.3859 และ 148.412 ตามลำดับ และ ค่า MAPE อยู่ที่ 43.06%, 43.1%, 43.15%และ 52.55% ตามลำดับ สำหรับ LightGBM และ XGBoost ใช้เวลาในการสร้างต้นแบบ และปรับ Hyperparameters ที่ประมาณ 3 ชั่วโมงและ 4 ชั่วโมงตามลำดับ ในส่วนของ Random forest นั้นใช้ไม่สามารถปรับปรุงได้ และ CatBoost นั้นใช้เวลาเพียงไม่ถึงชั่วโมง ดังนั้นอัลกอริทึมที่พัฒนาจากต้นไม้ตัดสินใจโดยเทคนิคการใช้วิธีร่วมกันตัดสินใจ Bagging (Random forest) จะใช้ทรัพยากรหน่วยความจำของคอมพิวเตอร์ที่มากกว่า และประมวลผลนานกว่าอัลกอริทึมที่พัฒนามาจากเทคนิค Boosting (LightGBM, XGBoost และ CatBoost) ทั้งนี้ในด้านประสิทธิภาพ หากใช้คอมพิวเตอร์ที่มีทรัพยากรมากขึ้นอาจจะส่งผลต่อประสิทธิภาพของต้นแบบที่สร้างจากอัลกอริทึม Random forest ได้ และหากใช้เวลาที่นานขึ้นอาจจะมีการเปลี่ยนแปลงอันดับของประสิทธิภาพของ LightGBM, XGBoost และ CatBoost ได้

5.2 ข้อเสนอแนะ

- ข้อจำกัดในด้านข้อมูล อาจส่งผลกับงานวิจัยนี้ เนื่องจากข้อมูลจาก iTIC มีการติดตามพาหนะประเภทแท็กซี่จำนวน 4,634 คันเท่านั้น
- ข้อจำกัดจากทรัพยากรคอมพิวเตอร์ ที่หน่วยความจำสามารถใช้ข้อมูลสร้างต้นแบบได้เพียง 1 ปี
- ปี 2020 ที่นำข้อมูลมาใช้ในงานวิจัยนี้อาจได้รับผลกระทบที่ไม่ปกติจากการระบาดของโรคติดเชื้อไวรัสโคโรน่า ส่งผลให้การจราจรอยู่ในสภาพไม่ปกติ จากการ Lock Down, Work from Home, Curfew ส่งผลกระทบกับข้อมูลปกติ สังเกตได้จากรูปแบบของดัชนีรถติดที่เปลี่ยนแปลงไปในช่วงเวลาเหล่านี้ และยังส่งผลกระทบไปยังการสร้างต้นแบบโดยตรง
- การใช้วิธี RandomizedSearchCV เพียง 100 ครั้งในการหาค่า Hyperparameters อาจจะไม่ได้ผลลัพธ์ที่ใกล้เคียงกับค่า Optimal

บรรณานุกรม

Andre Ye. (2020). *When and Why Tree-Based Models (Often) Outperform Neural Networks*. https://towardsdatascience.com/when-and-why-tree-based-models-often-outperform-neural-networks-ceba9ecd0fd8

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65, 126-139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004

Boyer, K. K., Prud'homme, A. M., & Chung, W. (2009). The last mile challenge: evaluating the effects of customer density and delivery window patterns. *Journal of business logistics*, 30(1), 185-201.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Burns, N. (2019). *Pattern Recognition via Principal Components Analysis*. https://www.sqlservercentral.com/articles/pattern-recognition-via-principal-components-analysis

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,

Cortes, J. D., & Suzuki, Y. (2021). Last-mile delivery efficiency: <italic>en route transloading</italic> in the parcel delivery industry [Article]. *International Journal of Production Research*, 1-18. https://doi.org/10.1080/00207543.2021.1907628

Deb, B., Khan, S. R., Hasan, K. T., Khan, A. H., & Alam, M. A. (2019, 29-31 March 2019). Travel Time Prediction using Machine Learning and Weather Impact on Traffic Conditions. 2019 IEEE 5th International Conference for Convergence in Technology (I2CT),

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3), 326-327.

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with

categorical features support. *arXiv preprint arXiv:*1810.11363.

Fan, W., & Qiu, B. (2021). Travel Time Forecasting on a Freeway Corridor: a Dynamic Information Fusion Model based on the Random Forests Approach [Tech Report]. https://rosap.ntl.bts.gov/view/dot/58260

Frohner, N., Horn, M., & Raidl, G. R. (2021). Route Duration Prediction in a Stochastic and Dynamic Vehicle Routing Problem with Short Delivery Deadlines [Article]. *Procedia Computer Science*, 180, 366-370. https://doi.org/10.1016/j.procs.2021.01.175

Geng, Y., Liu, E., Wang, R., & Liu, Y. (2020). Deep Reinforcement Learning Based Dynamic Route Planning for Minimizing Travel Time. *arXiv preprint arXiv:*2011.01771.

Gevaers, R., Van de Voorde, E., & Vanelslander, T. (2009). Characteristics of innovations in last-mile logistics-using best practices, case studies and making the link with green and sustainable logistics. *Association for European Transport and contributors*, 1-21.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:*2207.08815.

Gufosowa. (2017). *Cross-validation (statistics)*. https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Hadiyanto, Budi, W., Maryono, Subowo, E., Sediyono, E., & Farikhin. (2019). Ant Colony Algorithm for Determining Dynamic Travel Routes Based on Traffic Information from Twitter [Article]. *E3S Web of Conferences*, 125, 1-13. https://doi.org/10.1051/e3sconf/201912523012

Huang, L., & Xu, L. (2018, 19-21 July 2018). Research on Taxi Travel Time Prediction Based on GBDT Machine Learning Method. 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC),

KA-KA-shi. (2021). *What machine learning approaches have won most Kaggle competitions?* https://www.kaggle.com/general/248068

Kay Jan Wong. (2022). *CatBoost vs. LightGBM vs. XGBoost*. https://towardsdatascience.com/catboost-vs-lightgbm-vs-xgboost-c80f40662924

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017).

Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 3146-3154.

Khandelwal, P. (2017). *Which algorithm takes the crown: Light GBM vs XGBOOST?* https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

Klein, B. (2017). *What are Decision Trees?* https://python-course.eu/Decision_Trees.php

Kumar, V. (2018). *Random forests and decision trees from scratch in python*. https://towardsdatascience.com/random-forests-and-decision-trees-from-scratch-in-python-3e4fa5ae4249

Lateef, Z. (2019). *A Comprehensive Guide To Boosting Machine Learning Algorithms*. https://www.edureka.co/blog/boosting-machine-learning/

Li, F., Fan, Z.-P., Cao, B.-B., & Li, X. (2021). Logistics Service Mode Selection for Last Mile Delivery: An Analysis Method Considering Customer Utility and Delivery Service Cost. *Sustainability*, 13(1), 284. https://www.mdpi.com/2071-1050/13/1/284

Lint, J. W. C. (2006). Reliable Real-Time Framework for Short-Term Freeway Travel Time Prediction. *Journal of Transportation Engineering*, 132. https://doi.org/10.1061/(ASCE)0733-947X(2006)132:12(921)

Masui, T. (2022). *All You Need to Know about Gradient Boosting Algorithm – Part* 1. *Regression*. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Oh, S., Byon, Y.-J., Jang, K., & Yeo, H. (2015). Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach. *Transport Reviews*, 35(1), 4-32. https://doi.org/10.1080/01441647.2014.992496

Parinya, S. (2019). Developing Applications for Vehicle Routing Problems with Real Time Data Acquisition [Article]. *International Journal of Simulation -- Systems, Science & Technology*, 20(2), 1-6. https://doi.org/10.5013/IJSSST.a.20.02.09

Prasad, A. (2021). *Regression Trees | Decision Tree for Regression | Machine Learning*. https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning-e4d7525d8047

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3), 569-575.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3), 660-674.

Taskesen, E. (2021). *A step-by-step guide for clustering images*. https://towardsdatascience.com/a-step-by-step-guide-for-clustering-images-4b45f9906128

TomTom. (2021a). *Bangkok traffic*. https://www.tomtom.com/en_gb/traffic-index/bangkok-traffic/

TomTom. (2021b). *TomTom Traffic Index* 2020

Yu, G., & Yang, Y. (2019). Dynamic routing with real-time traffic information [Article]. *Operational Research*, 19(4), 1033-1058. https://doi.org/10.1007/s12351-017-0314-9

จารึก ประพันธ์พจน์. (2533). การศึกษาสภาพการเดินทางของนักเรียน เพื่อเป็นแนวทางประกอบการแก้ไขปัญหาการจราจรของกรุงเทพมหานคร : กรณีศึกษาเขตชั้นในฝั่งพระนคร จุฬาลงกรณ์มหาวิทยาลัย.

จารุวรรณ ลิมปเสนีย์. (2521). ที่ตั้งโรงเรียนกับการลดปัญหาจราจรในเขตบางรักและยานนาวา จุฬาลงกรณ์มหาวิทยาลัย.

บุญเสริม อินทรตุล. (2517). ปัญหาเกี่ยวกับการปรับปรุงค่าโดยสารรถประจำทางในเขตกรุงเทพมหานคร จุฬาลงกรณ์มหาวิทยาลัย.

พรทิวา วิศิษฏ์สรอรรถ. (2562). *Metrics* พื้นฐานสำหรับวัดประสิทธิภาพของโมเดล *Machine Learning*. https://shorturl.asia/nrKXT

เพิ่มศักดิ์ พูลพรม. (2548). ความสัมพันธ์ระหว่างปัญหาจราจรกับโครงข่ายถนนของพื้นที่ปิดล้อมขนาดใหญ่ของกรุงเทพมหานคร จุฬาลงกรณ์มหาวิทยาลัย.

มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย. (2561). *Thailand Location Table data*. https://itic.longdo.com/opendata/location-table/

มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย. (2563). ทำไมต้อง *iTIC?* https://www.iticfoundation.org/th/node/208

มูลนิธิศูนย์ข้อมูลจราจรอัจฉริยะไทย. (2564). *Thailand Vehicles and Mobile Probe Data*. https://itic.longdo.com/opendata/probe-data/

ลองดู Traffic. (2564). ดัชนีรถติด *Longdo (Longdo Traffic Index)*. https://traffic.longdo.com/trafficindex

วรพร ปุณยกนก. (2562). การใช้ *OSMnx* วิเคราะห์โครงข่ายถนน. http://www.urbanwhy.com/2019/05/28/osmnx/

วรรธนา พันธ์สว่าง. (2539). พฤติกรรมของคนขับรถจักรยานยนต์รับจ้างกับปัญหาจราจรในกรุงเทพมหานคร จุฬาลงกรณ์มหาวิทยาลัย.

วาที. (2563). 3 กลยุทธ์ ‘ธุรกิจขนส่งพัสดุ’ รับมือสงครามราคา ปริมาณส่ง 4 ล้านชิ้นต่อวัน มูลค่า 6.6 หมื่นล้าน. https://www.marketingoops.com/reports/industry-insight/transport-and-logistics-strategy-2020/

วิชญ์พงศ์ ดรุณธรรม. (2561). รู้จัก *Decision Tree, Random Forest,* และ *XGBoost*. https://shorturl.asia/7IhHy

ศูนย์เทคโนโลยีสารสนเทศภูมิศาสตร์กรุงเทพมหานคร. (2564). *Bangkok District Spatial File*. http://www.bangkokgis.com/modules.php?m=download_shapefile

ส่วนบริหารและพัฒนาเทคโนโลยีการทะเบียน, & สำนักบริหารการทะเบียน. (2564). รายงานสถิติจำนวนประชากรและบ้าน ประจำปี พ.ศ.2563. https://stat.bora.dopa.go.th/stat/statnew/statTDD/views/showDistrictData.php?rcode=10&statType=1&year=63

อดิศักดิ์ กันทะเมืองลี้. (2562). พื้นที่นอกบ้าน กรุงเทพฯ กับ 22% แห่งโอกาส. https://theurbanis.com/public-realm/20/12/2019/185

ประวัติผู้เขียน

ชื่อ-สกุล	นายปวริศ เวชวรรณกิจกุล
วัน เดือน ปี เกิด	9 กรกฎาคม 2540
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ที่อยู่ปัจจุบัน	2/17 ถนนโพแก้ว ซอย 3 แยก 19 หมู่บ้าน Private town แขวงคลองจั่น เขตบางกะปิ กรุงเทพมหานคร 10240