ระบบแสดงผลข้อมูลบนแผนผังเมืองจำลอง
CityViz

จิตตินาท นารถนรกิจ
ณราวรรณ เวชประสิทธิกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบแสดงผลข้อมูลบนแผนผังเมืองจำลอง

CityViz

จิตตินาท นารถนรกิจ

ณราวรรณ เวชประสิทธิกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแสดงผลข้อมูลบนแผนผังเมืองจำลอง

CITYVIZ

ผู้จัดทำ

1. นางสาวจิตตินาท นารถนรกิจ รหัสนักศึกษา 57010176
2. นางสาวณราวรรณ เวชประสิทธิกุล รหัสนักศึกษา 57010379

_____อาจารย์ที่ปรึกษา

(ดร. อรทัย สังข์เพ็ชร)

_____อาจารย์ที่ปรึกษาร่วม

(ดร. อักฤทธิ์ สังข์เพ็ชร)

ระบบแสดงผลข้อมูลบนแผนผังเมืองจำลอง

นางสาวจิตตินาท	นารถนรกิจ	57010176
นางสาวณราวรรณ	เวชประสิทธิกุล	57010379
อาจารย์อรทัย	สังข์เพ็ชร	อาจารย์ที่ปรึกษา
อาจารย์อักฤทธิ์	สังข์เพ็ชร	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2560

บทคัดย่อ

เนื่องจากปัจจุบันพื้นที่ต่าง ๆในประเทศมีแผนการที่จะพัฒนาไปสู่การเป็นเมืองอัจฉริยะ สังเกตได้จากการติดตั้งเซ็นเซอร์เพื่อวัดคุณภาพของสิ่งแวดล้อม ระบบเมืองอัจฉริยะต่าง ๆที่ได้จัดทำขึ้นใหม่มากมาย และการนำข้อมูลจากส่วนต่าง ๆ มาใช้เพื่อพัฒนาคุณภาพชีวิตของประชากรในประเทศ หนึ่งในปัญหาที่เกิดขึ้นสำหรับผู้ที่ต้องการเห็นข้อมูลเหล่านี้คือการแสดงผลข้อมูล และเนื่องจากมีข้อมูลหลากหลายรูปแบบ (เช่น ข้อความ, ตัวเลข, รูปภาพ) หลากหลายโครงสร้าง และหลากหลายที่มา หากต้องการนำข้อมูลต่าง ๆ มาแสดงผลให้ได้อย่างถูกต้อง นักพัฒนาก็จะต้องวิเคราะห์และคัดกรองข้อมูลใหม่ในทุก ๆ ครั้ง รวมถึงการเลือกรูปแบบการแสดงผลให้เหมาะสมกับข้อมูลนั้นด้วยเช่นกัน และหากมีการติดตั้งเซ็นเซอร์ที่มีความหลากหลายมากขึ้น ก็จะต้องมีกระบวนการนี้เกิดขึ้นในทุก ๆ การนำเข้าข้อมูลใหม่ ส่งผลให้เกิดความล่าช้าในการแสดงข้อมูล ทางคณะผู้จัดทำจึงเห็นสมควรว่าควรมีการสร้างระบบแนะนำการแสดงผลที่สามารถรองรับข้อมูลหลากหลายชนิด หลากหลายโครงสร้าง และหลากหลายที่มาเพื่อลดความซ้ำซ้อนในการทำงานที่อาจเกิดขึ้นในอนาคตได้

โดยระบบแสดงผลข้อมูลที่ได้สร้างขึ้นนั้นจะทำการรับข้อมูลจากผู้ใช้งาน ก่อนนำข้อมูลนั้นมาวิเคราะห์หาโครงสร้างข้อมูลและแปลงให้เป็นชนิดข้อมูลของระบบแสดงผลที่ได้นิยามขึ้น ก่อนนำชนิดข้อมูลมาทำการจำแนกประเภทการแสดงผลที่เหมาะสมโดยอ้างอิงจากประเภทการแสดงผลที่ได้มีการแบ่งไว้ก่อนหน้า โดยอาจมีการแสดงผลบนระบบเมืองจำลองตามความเหมาะสมของรูปแบบการแสดงผลที่ได้รับมา ซึ่งระบบที่ได้สร้างขึ้นนี้จะลดระยะเวลาในการทำงานได้เมื่อเทียบกับกระบวนการที่ผู้แสดงผลต้องกระทำในขั้นตอนการแสดงผลข้อมูลแบบปกติ รวมถึงสามารถรองรับการนำเข้าข้อมูลในปริมาณมากที่ต้องการแสดงผลอย่างรวดเร็วได้เช่นกัน

CityViz

Ms.	Jittinat	Nartnorakij	57010176
Ms.	Narawan	Vejprasitthikul	57010379
Mrs.	Orathai	Sangpetch	Advisor
Mr.	Akkarit	Sangpetch	Co-Advisor

Academic Year 2016

ABSTRACT

Many Cities have plan to be a smart city in the near future. Currently, some cities (e.g., London, Amsterdam) have already deployed thousands of sensors to collect real-time data, such as environment quality, health sensor, power sensor, vehicle traffic and so on. In order to understand the data, municipalities and communities need to be able to visualize the data to serve particular purposes. In the visualization process, they need to analyze object structures and select the visualization model which is suitable to data type. If they get new data, they need to analyze and select the proper visualization model.

This project proposes a visualization system which helps users choose appropriate visualization models to suit different structures, formats, and sources of data. The proposed model will help people analyze and visualize data in the holistic view of city effectively. The visualization may be in 2D/3D perspective, depending on data. This project will reduce the steps in the visualization process and propose the visualization model namely CityViz, which helps users choose appropriate visualization models. CityViz works base on proposed visualization data model.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สามารถจัดทำขึ้นมาอย่างสำเร็จลุล่วงได้เป็นอย่างดี เนื่องด้วยการใช้ความช่วยเหลือและการให้คำปรึกษาจากหลาย ๆ ฝ่าย ซึ่งโครงงานฉบับนี้จะสำเร็จลุล่วงไม่ได้เลย หากปราศจากความช่วยเหลือจากบุคคลเหล่านี้

อาจารย์ที่ปรึกษาทั้งสองท่านคือ อาจารย์อรทัย สังข์เพ็ชร และ อาจารย์อักฤทธิ์ สังเพ็ชร ที่เป็นผู้ให้คำแนะนำ ปรึกษา และให้ความช่วยเหลือตลอดการทำโครงงาน ซึ่งทำให้การทำงานต่าง ๆ เป็นไปได้อย่างราบรื่น ต้องขอขอบคุณอาจารย์ที่ปรึกษาทั้งสองท่านสำหรับความช่วยเหลือเหล่านี้

นอกเหนือจากนี้ขอขอบคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้อบรมความรู้ และสั่งสอนคณะผู้จัดทำในด้านต่าง ๆ ซึ่งคณะผู้จัดทำได้นำความรู้ในหลาย ๆ ด้านมาประยุกต์ใช้ในการทำปริญญานิพนธ์ รวมถึงเอื้อเฟื้อสถานที่ในการทำวิจัยและพัฒนาโครงงาน

ในท้ายที่สุดนี้ขอขอบคุณบิดา มารดา และครอบครัวที่ได้เลี้ยงดูและสั่งสอน พร้อมทั้งให้โอกาสในการศึกษาและให้กำลังใจเสมอมา

จิตตินาท นารถนรกิจ
ณราวรรณ เวชประสิทธิกุล

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ IV

สารบัญตาราง IX

สารบัญรูป X

บทที่ 1 บทนำ 1

1.1 ที่มาของปัญหา 1

1.2 วัตถุประสงค์และเป้าหมายของโครงงาน 1

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตการดำเนินงาน 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 เมืองอัจฉริยะ (Smart City) 4

2.1.1 ลักษณะที่พึงมีของเมืองอัจฉริยะ 5

2.1.2 แบบแผนเทคโนโลยีในเมืองอัจฉริยะ 5

2.1.3 แบบแผนเกี่ยวกับทรัพยากรบุคคลในเมืองอัจฉริยะ 7

2.1.4 ตัวอย่างเมืองอัจฉริยะ 7

2.1.4.1 สิงค์โปร์ 7

2.1.4.2 บาร์เซโลน่า 7

2.2 ประเภทของข้อมูล 8

2.2.1 ประเภทของข้อมูลในแง่ของสถิติ 8

2.2.1.1 ข้อมูลประเภท Quantitative 8

2.2.1.1.1 Continuous Data 8

2.2.1.1.2 Discrete Data 8

2.2.1.2 ข้อมูลประเภท Qualitative 9

2.2.1.2.1 Nominal หรือ Attribute หรือ Categorical Data 9

สารบัญ (ต่อ)

หน้า

2.2.1.2.2 Binomial 9
2.2.1.2.3 Ordinal หรือ Ranked Data 9
2.2.2 ประเภทของข้อมูลในแง่ของการเขียนโปรแกรมคอมพิวเตอร์ 9
2.2.2.1 ข้อมูลประเภท Integers 9
2.2.2.2 ข้อมูลประเภท Booleans 10
2.2.2.3 ข้อมูลประเภท Characters 10
2.2.2.4 ข้อมูลประเภท Floating-Point Numbers 10
2.2.2.5 ข้อมูลประเภท String 10
2.3 ประเภทของกระบวนการประมวลผลข้อมูล 10
2.3.1 Batch Processing 10
2.3.2 Real Time Processing 10
2.3.3 Online Processing 10
2.4 ภาษาซีชาร์ป 10
2.4.1 ตัวอย่างการเขียนโปรแกรม ภาษา C# 11
2.4.2 จุดเด่นของภาษาC# 11
2.5 การสื่อสารข้อมูลผ่าน API 12
2.5.1 ประโยชน์ของAPI 12
2.5.2 ชนิดของAPI 13
2.5.2.1 Web Services API 13
2.5.2.2 Source Code API 13
2.5.2.3 Object Remoting API 13
2.5.2.4 Hardware API 13
2.5.3 ประเภทของ Web API 13
2.5.3.1 API แบบ SOAP 13
2.5.3.2 API แบบ REST 14
2.5.3.3 ความแตกต่างระหว่าง REST กับ SOAP 15

สารบัญ (ต่อ)

หน้า

2.6 Data Visualization 15
2.7 Machine Learning 16
2.7.1 ประเภทของ Machine Learning 16
2.7.1.1 Supervised Learning 16
2.7.1.2 Unsupervised Learning 16
2.7.1.3 Reinforcement Learning 16
2.8 ผลงานวิจัยที่เกี่ยวข้อง 17
2.8.1 A scalable middleware-based infrastructure for energy management and visualization in city districts 17
2.8.2 Integrated energy monitoring and visualization system for Smart Green City development 18

บทที่ 3 การวิเคราะห์และออกแบบระบบ 20
3.1 วิเคราะห์ข้อมูลที่เกี่ยวข้องกับระบบเมืองอัจฉริยะ 20
3.1.1 ศึกษาข้อมูล 20
3.1.2 วิเคราะห์ข้อมูล 30
3.2 ออกแบบระบบ 31
3.2.1 ออกแบบชนิดข้อมูลของ CityViz 31
3.2.2 ออกแบบระบบ 32
3.2.2.1 ส่วนเตรียมการข้อมูลสำหรับการแสดงผล (Data preparation) 32
3.2.2.2 ส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล (Key Attribute Selection and Visualization Recommendation) 33
3.2.2.2.1 ส่วนเลือกค่าหลักในการแสดงผล (Key Attribute Selection) 33
3.2.2.2.1.1 การเลือกค่าหลักในการแสดงผลตามหลักสถิติ (Statistical Approach) 34
3.2.2.2.1.2 การเลือกค่าหลักในการแสดงผลโดยใช้ Machine Learning 34

สารบัญ (ต่อ)

หน้า

3.2.2.2.2 ส่วนแนะนำรูปแบบในการแสดงผล (Visualization Recommendation) .. 35
3.2.2.3 ส่วนแสดงผลข้อมูล (Visualization) 37
3.3 วิธีการดำเนินงาน 38
3.3.1 โครงสร้างในการพัฒนาระบบ 38
3.3.2 การทำงานของระบบ CityViz 38
3.3.2.1 ส่วนเตรียมการข้อมูลสำหรับการแสดงผล 38
3.3.2.1.1 การวิเคราะห์โครงสร้างข้อมูล 38
3.3.2.1.2 การระบุชนิดข้อมูลของ CityViz 40
3.3.2.2 ส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล 40
3.3.2.2.1 ส่วนเลือกค่าหลักในการแสดงผล 40
3.3.2.2.1.1 การหาค่าหลักในการแสดงผลโดยใช้หลักสถิติ (Statistical Approach) 40
3.3.2.2.1.2 การหาค่าหลักในการแสดงผลโดยใช้ Machine Learning 41
3.3.2.2.2 การหาค่าพิกัดทางภูมิศาสตร์ให้กับค่าหลักในการแสดงผล 41
3.3.2.2.3 ส่วนแนะนำรูปแบบในการแสดงผล 42
3.3.2.2.4 ส่วนบันทึกข้อมูลผลลัพธ์ 42
3.3.2.3 ส่วนแสดงผลข้อมูล 42
3.4 ตัวอย่างการใช้งานระบบแสดงผล 43
3.4.1 การเปิดระบบ CityViz 43
3.4.2 หน้าการแสดงผล 43
3.4.2.1 หน้าหลักของระบบ 43
3.4.2.2 การเพิ่มข้อมูลเข้าสู่ระบบ 45
3.4.2.3 รูปแบบการแสดงผล 48

บทที่ 4 ผลการทดลอง 52
4.1 ผลการทดลองความแม่นยำของส่วน Key Attribute Prediction 52
4.1.1 ผลการทดลองความแม่นยำของระบบแบบ Rule-Based System 54

สารบัญ (ต่อ)

หน้า

4.1.2 ผลการทดลองความแม่นยำของระบบแบบ Decision Tree 55

4.1.3 ผลการทดลองความแม่นยำของระบบแบบ Naïve Bayes..... 56

4.1.4 ผลการทดลองความแม่นยำของระบบแบบ K-Nearest Neighbors..... 57

4.1.5 เปรียบเทียบผลการทำนายข้อมูลประเภท Key 58

4.1.6 เปรียบเทียบผลการทำนายข้อมูลประเภท Non – Key 60

4.1.7 ผลการเปรียบเทียบค่า Accuracy ของอัลกอริทึมทั้งหมด 62

4.2 ผลการทดลองความแม่นยำเมื่อปรับค่า k ใน K-Nearest Neighbors..... 63

4.2.1 ผลการทดลองความแม่นยำของK-Nearest Neighbors โดยมีค่า k เป็น 1 63

4.2.2 ผลการทดลองความแม่นยำของK-Nearest Neighbors โดยมีค่า k เป็น 3 64

4.2.3 ผลการทดลองความแม่นยำของK-Nearest Neighbors โดยมีค่า k เป็น $\sqrt{N/2}$ 65

4.2.4 เปรียบเทียบผลการทำนายข้อมูลประเภท Key โดยใช้ค่า k ใน K-Nearest Neighbors ที่ต่างกัน..... 66

4.2.5 เปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key โดยใช้ค่า k ใน K-Nearest Neighbors ที่ต่างกัน 68

4.2.6 ผลการเปรียบเทียบค่า Accuracy ของค่า k ทั้งหมด 70

4.3 วิเคราะห์ผลการทดลอง..... 71

บทที่ 5 สรุปผลผลการทดลองและแนวทางการพัฒนาต่อ 73

5.1 สรุปผลการทดลอง..... 73

5.2 แนวทางการพัฒนาต่อ 74

สารบัญตาราง

ตาราง **หน้า**

2.1 ข้อแตกต่างระหว่าง REST API และ SOAP API 15
3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา 20
3.2 ความสัมพันธ์ระหว่างชนิดข้อมูลของ CityViz กับชนิดข้อมูลทั่วไป 31
3.3 ชื่อตัวแปรและชนิดข้อมูลที่มีลักษณะเป็นค่าหลักในการแสดงผลตามสถิติ 34
3.4 ความสัมพันธ์ของข้อมูลที่ใช้ในการแสดงผลกับรูปแบบการแสดงผล 36
3.5 ลักษณะการแสดงผลตามรูปแบบการแสดงผล 36
4.1 ต้นแบบ Confusion Matrix 52
4.2 นิยามความหมายของข้อมูลที่ใช้ประกอบการหา Confusion Matrix ตามคลาสที่สนใจจะประเมินผลความแม่นยำ 53
4.3 Confusion Matrix ของ Rule-Based System 54
4.4 Confusion Matrix ของ Decision Tree 55
4.5 Confusion Matrix ของ Naïve Bayes 56
4.6 Confusion Matrix ของ K-Nearest Neighbors 57
4.7 Confusion Matrix ของ K-Nearest Neighbors โดย k = 1 63
4.8 Confusion Matrix ของ K-Nearest Neighbors โดย k = 3 64
4.9 Confusion Matrix ของ K-Nearest Neighbors โดย k = $\sqrt{N2}$ 65

สารบัญรูป

รูป หน้า

2.1 ภาพรวมของ Smart City 4
2.2 Technology of Smart City 6
2.3 ภาพรวมของการสื่อสารข้อมูลด้วย API 12
2.4 ภาพรวมของ Web API 14
2.5 โครงสร้างของงานวิจัย scalable middleware-based infrastructure 17
2.6 ลำดับการ Query ข้อมูลก่อนส่งกลับสู่ Web Application 18
2.7 โครงสร้างฐานข้อมูลของ Integrated energy monitoring and visualization system 19
3.1 การทำงานของระบบ 32
3.2 การทำงานของส่วนเตรียมการข้อมูลสำหรับการแสดงผล 32
3.3 การทำงานของส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล 33
3.4 การทำงานของส่วนแสดงผลข้อมูล 37
3.6 กระบวนการวิเคราะห์โครงสร้างข้อมูลของระบบ 39
3.6 โฟลเดอร์ที่เก็บไฟล์โปรแกรม 43
3.7 หน้าหลักของ CityViz 44
3.8 วิธีการเพิ่มข้อมูลเข้าสู่ระบบ 48
3.9 ตัวอย่างการแสดงผลตามรูปแบบการแสดงผล 51
4.1 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Key 58
4.2 กราฟแสดงผลเปรียบเทียบค่า Precision ของการทำนายข้อมูลประเภท Key 59
4.3 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key 60
4.4 กราฟแสดงผลเปรียบเทียบค่า Precision ของการทำนายข้อมูลประเภท Non-Key 61
4.5 กราฟผลการเปรียบเทียบค่า Accuracy ของอัลกอริทึมทั้งหมด 62
4.6 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Key ของ K-Nearest Neighbors 66
4.7 กราฟแสดงผลเปรียบเทียบค่าPrecision ของคลาส Key ของ K-Nearest Neighbors 67
4.8 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key ของ K-Nearest Neighbors 68
4.9 กราฟแสดงผลเปรียบเทียบค่าPrecision ของคลาส Non-Key ของ K-Nearest Neighbors 69
4.10 กราฟแสดงผลการเปรียบเทียบค่า Accuracy ของค่า k ทั้งหมด 70
4.11 ตัวอย่างลักษณะขอบเขตการตัดสินใจของ Decision Tree 71

สารบัญรูป (ต่อ)

รูป หน้า

4.12 ตัวอย่างลักษณะขอบเขตการตัดสินใจของ K-Nearest Neighbors เมื่อ k = 1 72

บทที่ 1

บทนำ

1.1 ที่มาของปัญหา

เนื่องจากปัจจุบันพื้นที่ต่าง ๆในประเทศมีแผนการที่จะพัฒนาไปสู่การเป็นเมืองอัจฉริยะ สังเกตได้จากการติดตั้งเซ็นเซอร์เพื่อวัดคุณภาพของสิ่งแวดล้อม ระบบอัจฉริยะต่าง ๆที่ได้จัดทำขึ้นใหม่มากมาย และการนำข้อมูลจากส่วนต่าง ๆ มาใช้เพื่อพัฒนาคุณภาพชีวิตของประชากรในประเทศ ทำให้ความหลากหลายของข้อมูลในแต่ละพื้นที่เริ่มมีมากขึ้นเรื่อย ๆ โดยการใช้ข้อมูลเพื่อให้ได้ประโยชน์สูงสุดนั้นผู้ที่ต้องการใช้งานจะต้องนำข้อมูลไปแสดงผลเพื่อวิเคราะห์เพื่อตอบสนองวัตถุประสงค์ที่ต้องการ ซึ่งในกระบวนการนำข้อมูลมาแสดงผลนั้น ผู้แสดงผลจะต้องวิเคราะห์โครงสร้างของข้อมูลที่ได้รับมาและหาชนิดข้อมูลทั้งหมดที่มีเพื่อจำแนกข้อมูลตามประเภทที่อาจแสดงผลได้ ก่อนทำการเลือกประเภทการแสดงผลข้อมูลแต่ละชนิดด้วยตนเอง รวมไปถึงขั้นตอนการนำข้อมูลมาแปลงให้อยู่ในรูปแบบที่สามารถแสดงผลในแพลตฟอร์มต่าง ๆได้อีกด้วย และหากผู้แสดงผลข้อมูลเริ่มได้รับข้อมูลที่มีความหลากหลายมากขึ้น ก็จะต้องทำกระบวนการเดิมซ้ำ ๆ ในแต่ละชนิดและประเภทของข้อมูลที่เข้ามาใหม่ ซึ่งจะส่งผลถึงความล่าช้าในการแสดงผลเนื่องจากต้องผ่านกระบวนการต่าง ๆ ในข้างต้น และปริมาณงานที่มากเกินพอดีในการแสดงผลข้อมูลที่มีทั้งหมด

ทางคณะผู้จัดทำจึงเห็นสมควรว่าควรมีระบบที่สามารถทำการวิเคราะห์รูปแบบของข้อมูล รวมถึงเลือกประเภทการแสดงผลที่เหมาะสมกับข้อมูลที่รับเข้ามาได้ รองรับการแสดงผลข้อมูลที่มีในปริมาณมาก หลากหลายชนิดและโครงสร้างข้อมูล โดยระบบจะทำการวิเคราะห์โครงสร้างของข้อมูลและนำมาแสดงผลตามประเภทการแสดงผลที่ได้มีการแบ่งไว้ เพื่อลดขั้นตอนการทำงานซ้ำซ้อนที่อาจเกิดขึ้นด้วย ซึ่งหากข้อมูลมีการแสดงผลที่ถูกต้องเหมาะสมและรวดเร็ว ก็จะทำให้ผู้ที่ต้องการใช้ข้อมูลสามารถนำข้อมูลไปวิเคราะห์ถึงปัญหาต่างๆ ที่อาจเกิดขึ้นได้อย่างทันท่วงที

1.2 วัตถุประสงค์และเป้าหมายของโครงงาน

1) เพื่อรองรับการแสดงผลข้อมูลได้จากข้อมูลที่หลากหลายประเภท หลากหลายชนิด และหลากหลายแหล่งที่มาของข้อมูล
2) เพื่อลดกระบวนการวิเคราะห์โครงสร้าง และชนิดของข้อมูลแต่ละข้อมูลที่ผู้แสดงผลต้องทำก่อนการแสดงผลข้อมูลใหม่

เพื่อรองรับการแสดงผลข้อมูลที่เข้ามาใหม่ในปริมาณมาก และต้องการแสดงผลอย่างรวดเร็วเพื่อนำไปวิเคราะห์และใช้งาน

3) เพื่อแสดงผลข้อมูลที่ได้รับมาให้เป็นไปอย่างเหมาะสมตามประเภทข้อมูลที่ระบบได้แบ่งไว้
4) เพื่อเข้าถึงผู้ใช้งานที่ต้องการแสดงผลข้อมูลแต่ไม่มีเวลาในการวิเคราะห์และสังเคราะห์ข้อมูล
5) เพื่อเข้าถึงผู้ใช้งานที่ต้องการเห็นภาพรวมระบบที่แสดงผลข้อมูลในแต่ละพื้นที่บนระบบผังเมืองจำลอง 3 มิติที่จะทำให้เห็นพื้นที่จริงได้ชัดเจนขึ้น

1.3 ประโยชน์ที่คาดว่าจะได้รับ

จากการที่คณะผู้จัดทำได้เล็งเห็นถึงปริมาณของข้อมูลที่เพิ่มมากขึ้น ทั้งจากหลากหลายประเภท หลากหลายที่มา รวมถึงหลากหลายโครงสร้างข้อมูล ทำให้เกิดปัญหาในกระบวนการแสดงผลข้อมูลที่ต้องใช้กระบวนการการวิเคราะห์โครงสร้างและเลือกประเภทการแสดงผลให้เหมาะสมกับข้อมูล รวมถึงผู้ที่ต้องการแสดงผลข้อมูลจะต้องเสียเวลาในกระบวนการข้างต้นซ้ำซ้อนในทุกๆ ข้อมูลที่เข้ามาใหม่ คณะผู้จัดทำจึงได้จัดทำระบบนี้ขึ้นเพื่อรองรับการแสดงผลข้อมูลที่มีความหลากหลายในด้านต่างๆ เพื่อลดปริมาณการทำงานซ้ำซ้อนที่อาจเกิดขึ้นและสามารถเลือกแสดงผลข้อมูลที่ผู้ใช้ต้องการได้โดยไม่เสียเวลาจากกระบวนการข้างต้น ซึ่งจะทำให้ผู้ใช้สามารถเห็นข้อมูลที่ตนเองต้องการและนำไปใช้ได้อย่างทันท่วงที

1.4 ขอบเขตการดำเนินงาน

1) ระบบสามารถรองรับข้อมูลต่าง ๆ ผ่านรูปแบบการส่งข้อมูลแบบ API เท่านั้น
2) ระบบสามารถวิเคราะห์โครงสร้างของข้อมูลได้จากข้อมูลในรูปแบบ JSON เท่านั้น
3) หากเป็นข้อมูลแบบ real-time ระบบจะทำการดึงข้อมูลจาก API นั้นๆใหม่ทุก 15
4) หากเป็นข้อมูลแบบ real-time หรือข้อมูลที่ต้องมีการอัพเดต รูปแบบและโครงสร้างข้อมูลที่ผู้ใช้ส่งมาเพื่อแสดงผลจะต้องลักษณะเหมือนกันในทุก ๆ ครั้ง
5) รูปแบบการแสดงผลที่ได้ทำการแบ่งประเภทไว้ อ้างอิงจากข้อมูลที่ได้นำมาทดสอบในเบื้องต้นซึ่งเป็นข้อมูลประเภทค่าที่ได้ได้รับจาก sensor ชนิดต่าง ๆ, ข้อมูลความหนาแน่น

ประชากร, ข้อมูลการเคลื่อนที่ของกลุ่มคน และข้อมูลสภาพอารมณ์ในแต่ละพื้นที่และข้อมูลจากโซเชี่ยลมีเดีย

6) ระบบจะทำการแสดงผลตามรูปแบบที่ได้แยกประเภทของการแสดงผลไว้ โดยจะมุ่งเน้นไปที่การแสดงผลค่าทางสถิติแบบ 2 มิติ ในระดับพื้นฐาน เช่น กล่องข้อความหรือกล่องข้อความพร้อมค่าสถิติทางคณิตศาสตร์(ค่าเฉลี่ย,ค่าสูงสุดและค่าน้อยสุด) และการแสดงผลแบบ 3 มิติ เช่น ข้อความลอยตัว (text pop-up) และไอคอน 3 มิติที่สื่อถึงข้อมูลเชิงปริมาณ เช่น จำนวนคนที่มีอยู่มาก หรือสภาพอารมณ์ของคนหมู่มากในพื้นที่นั้น
7) ระบบสามารถแสดงผลบนผังเมืองจำลอง 3 มิติได้เฉพาะข้อมูลที่อยู่ในพื้นที่ที่ได้จัดทำผังเมือง 3 มิติ (เช่น โซนใจกลางกรุงเทพ,โซนสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,บางแสนและหาดใหญ่) แล้วเท่านั้น
8) ระบบสามารถแสดงผลผ่านโปรแกรมบนระบบปฏิบัติการวินโดว์

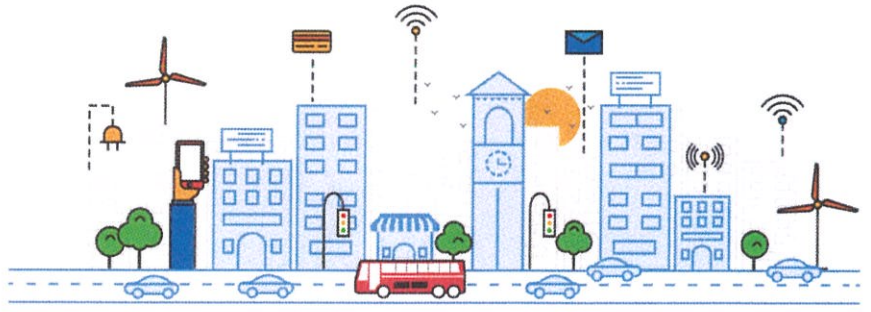
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 เมืองอัจฉริยะ (Smart City)

เมืองอัจฉริยะเป็นแนวคิดที่มีจุดประสงค์ในการใช้เทคโนโลยีสารสนเทศและการสื่อสารมาใช้สนับสนุนการทำงานของรัฐบาลเพื่อให้เข้าถึงความต้องการของประชาชนในพื้นที่ ตรวจสอบและเฝ้าระวังปัญหาเพื่อที่จะสามารถแก้ปัญหาได้อย่างทันท่วงที รวมไปถึงใช้สนับสนุนการยริหารทรัพยากรเพื่อให้ใช้ทรัพยากรในพื้นที่ให้เกิดประโยชน์สูงสุด เนื่องจากเมืองอัจฉริยะเป็นแนวคิดใหม่และเป็นแนวคิดที่ไม่มีคำนิยามแน่ชัด เพราะขึ้นอยู่กับว่าผู้ที่จะพัฒนาระบบเมืองอัจฉริยะให้กับเมืองของตนเองนั้นจะให้มุ่งเน้นไปที่ความอัจฉริยะทางใด จุดมุ่งหมายใหญ่ 4 ประการที่เมืองอัจฉริยะมักจะมี ได้แก่

1) เป็นระบบที่มีอุปกรณ์อิเล็กทรอนิกส์จำนวนมากและหลากหลายรูปแบบกระจายอยู่ทั่วเมืองเพื่อติดต่อสื่อสารแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์กับระบบเมืองอัจฉริยะ
2) มุ่งใช้เทคโนโลยีสารสนเทศและการสื่อสารในการพัฒนาคุณภาพชีวิต,ทรัพยากร และสิ่งแวดล้อม
3) เทคโนโลยีสารสนเทศและการสื่อสารเป็นส่วนหนึ่งของรัฐบาล
4) สนับสนุนให้ประชาชนเข้าถึงเทคโนโลยีสารสนเทศและการสื่อสารเพื่อเป็นส่วนกระตุ้นในการพัฒนาสิ่งใหม่และเสริมสร้างความรู้ของประชาชน



รูป 2.1 ภาพรวมของ Smart City

(รูปจาก https://en.wikipedia.org/wiki/Smart_city)

2.1.1 ลักษณะที่พึงมีของเมืองอัจฉริยะ

เมืองอัจฉริยะเพื่อให้บรรลุเป้าหมายและมีการทำงานอย่างมีประสิทธิภาพ เมืองอัจฉริยะนั้นควรที่จะ

1) ทำให้ทรัพยากรต่าง ๆ ในพื้นที่สามารถจับต้องได้ เช่น ถนน ตึก สะพาน ฯลฯ ถูกใช้ได้ อย่างมีประสิทธิภาพขึ้น ผ่านการใช้ปัญญาประดิษฐ์ (Artificial Intelligence) และเทคนิคการวิเคราะห์ข้อมูล (Data Analytics) เพื่อสนับสนุนให้เมืองเป็นเมืองที่แข็งแกร่งทั้งในด้านเศรษฐกิจ สังคมและวัฒนธรรม
2) พยายามสร้างปฏิสัมพันธ์ระหว่างประชนชนกับเจ้าหน้าที่ของรัฐในแต่ละพื้นที่ให้ได้มาที่สุด นำความต้องการและความเห็นของประชาชนมาเป็นเหตุผลหลักในการตัดสินใจสั่งการสร้างหรือพัฒนาสิ่งต่าง ๆ เน้นความมีสส่วนร่วมของแระชาชน และเปิดโอกาสให้ประชาชนแสดงความคิดเห็น
3) เรียนรู้และปรับตัวเข้าหาความต้องการของประชาชนอย่างว่องไวและสม่ำเสมอ เนื่องจากมีเทคโนโลยีมาสนับสนุนการตัดสินใจแล้ว

ทั้งนี้ทั้งนั้นการเป็นจะเป็นเมืองอัจฉริยะที่ดีได้จะต้องได้รับความร่วมมือจากประชนชน การเก็บข้อมูลที่มีประสิทธิภาพ รวมไปถึงมีปัญญาประดิษฐ์ที่ชาญฉลาดมาช่วยในการวิเคราะห์และตัดสินใจสิ่งที่เกิดขึ้นในเมือง

2.1.2 แบบแผนเทคโนโลยีในเมืองอัจฉริยะ

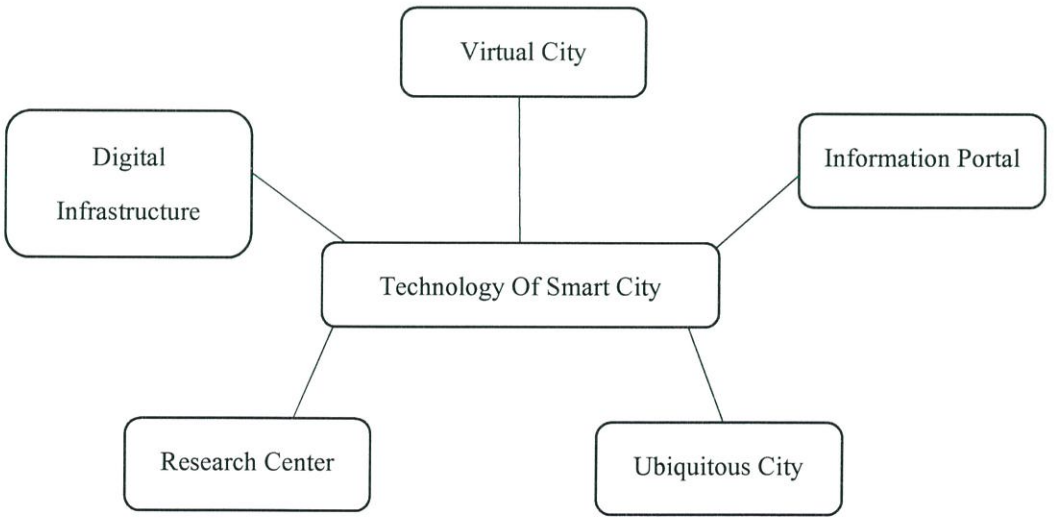
การสร้างเมืองอัจฉริยะประกอบไปด้วยการนำเทคโนโลยีหลากหลายประเภทมาใช้ร่วมกันเพื่อให้เมืองอัจฉริยะบรรลุเป้าหมายและเกิดความสะดวกทั้งฝั่งประชาชนและฝั่งของเจ้าหน้าที่รัฐ แบ่งเป็น 5 ส่วน ดังนี้

1) ส่วนของผังเมืองดิจิตอล คือเชื่อมต่อทรัพยากร สาธารณูปโภคต่าง ๆ ในพื้นที่ ด้วยอุปกรณ์อิเล็กทรอนิกส์ เช่น เซนเซอร์ตรวจจับ ฯลฯ ลงไปในทรัพยากรต่าง ๆของเมือง จากนั้นเชื่อมต่อสิ่งเหล่านั้นเข้าสู่ระบบเมืองอัจฉริยะส่วนกลาง เพื่อรับรู้ความเป็นไปของเมืองและแลกเปลี่ยนข้อมูลที่อาจจะเป็นประโยชน์
2) ส่วนของเมืองจำลองที่สร้างมาให้เหมือนเมืองของจริงโดยอยู่ในรูปแบบของข้อมูลอิเลกทรอนิกส์ เช่น โปรแกรมแสดงผลเมืองจำลอง เป็นต้น โดยส่วนนี้เป็นเมืองจำลองซึ่งสร้างมาจากข้อมูลที่เกิดขึ้นจริงที่มองเห็นได้ เช่นส่วนสูงของตึก ขนาด ระยะห่างของทรัพยากรต่าง ๆ ผนวกกับข้อมูลที่เกิดขึ้นจริงแต่เราไม่สังเกตได้ง่าย เช่น

ปริมาณการใช้ไฟฟ้า จำนวนคนในพื้นที่ ณ เวลาใด ๆ มาแสดงผลรวมกัน ทำให้มองเห็นความเป็นไปของเมืองและสนับสนุนการตัดสินใจให้เป็นไปอย่างมีประสิทธิภาพและฉับไวขึ้น

3) ส่วนศูนย์รวมข้อมูลของเมือง นำข้อมูลได้จากส่วนเมืองดิจิตอลและวิธีการเก็บข้อมูลอื่น ๆ มารวบรวมกันไว้เป็นศูนย์กลาง เพื่อให้เข้าถึงได้ง่ายทั้งโดยเจ้าหน้าที่ของรัฐและเปิดกล้างให้กับประชาชน
4) ส่วนวิจัยข้อมูล ทำการวิจัยข้อมูลที่ได้จากศูนย์รวมข้อมูลของเมือง มาค้นหาและศึกษา หาประโยชน์จากข้อมูลเหล่านั้น ว่ามีสิ่งใดมีผลกระทบต่อคุณภาพชีวิต สังคม หรือเศรษฐกิจของเมือง
5) ส่วนอำนวยความสะดวกด้านทรัพยากร ทำให้ทรัพยากรทุกอย่างเข้าถึงได้ง่ายผ่านเทคโนโลยีเช่นอินเทอร์เน็ต ยกตัวอย่างเช่น ระบบขอยืมจักรยาน ประชาชนสามารถส่งร้องขอยืมจักรยานจากที่สถานที่ใดก็ได้ในเมืองผ่านทางแอปพลิเคชันบนโทรศัพท์มือถือ

หากมีทุกส่วนประกอบรวมกันได้เมืองจะกลายเมืองที่มีโครงสร้างทางเทคโนโลยีที่รองรับการทำงานแบบเมืองอัจฉริยะได้อย่างเต็มประสิทธิภาพ



รูป 2.2 Technology of Smart City

2.1.3 แบบแผนเกี่ยวกับทรัพยากรบุคคลในเมืองอัจฉริยะ

ต่อให้มีแผนเทคโนโลยีที่ดีในการทำเมืองอัจฉริยะแต่คนในพื้นที่ไม่ให้ความร่วมมือเมืองอัจฉริยะอาจจะไม่สำเร็จได้โดยราบรื่น ดังนั้นคนจึงปัจจัยสำคัญปัจจัยหนึ่งในการทำเมืองอัจฉริยะ แนวทางปฏิบัติทางด้านบุคลากรในเมืองอัจฉริยะมีดังนี้

1) ประชาชนในเมืองควรจะมีความคิดสร้างสรรค์และสนใจสภาพบ้านเมือง พร้อมที่จะมีส่วนร่วมในการพัฒนาเมืองอยู่เสมอ
2) มีแผนการพัฒนาคนให้มีคุณภาพเพื่อที่จะให้คนเหล่านั้นมาเป็นกำลังสำคัญในสร้างสรรค์และคิดค้นนวัตกรรมใหม่ๆ
3) ยกระดับการศึกษาพื้นฐานส่งเสริมให้ทุกคนมีการศึกษาที่ดีและมากพอ เพื่อสร้างสภาพแวดล้อมที่ฉลาด และดึงดูดบุคคลกรที่มีความสามารถเข้าเมือง
4) พัฒนาเมืองด้วยความรู้และหลักการ ตัดสินใจอย่างมีเหตุผลโยมี ข้อมูลหรือทฤษฎีมาสนับสนุนเสมอ

2.1.4 ตัวอย่างเมืองอัจฉริยะ

2.1.4.1 สิงค์โปร์

สาธารณรัฐสิงค์โปร์ประกาศตัวว่าจะเป็นเมืองอัจฉริยะที่แรกของโลก โดยเริ่มโครงการในปลายปี 2014 ภายใต้วาระการทำงานของนายกรัฐมนตรี Lee Hsien Loong เริ่มติดเซนเซอร์และกล้องจำนวนมากทั่วทั้งเกาะสิงค์โปร์ เพื่อติดตามสถานการณ์ต่าง ๆภายในเกาะตั้งแต่เรื่องของความสะอาดไปจนถึงสภาพจราจร จากนั้นเริ่มพัฒนาระบบตรวจจับคนสูบบุหรี่ในพื้นที่ห้ามสูบ,ระบบตรวจจับหากมีคนทิ้งขยะลงมาจากตึกสูง เป็นต้น นอกจากนี้ยังมีการพัฒนาซอฟต์แวร์ที่มีชื่อว่า “Virtual Singapore” ที่มีจุดประสงค์ในการทดลองต่าง ๆที่สามารถทำได้ยากในสถานที่จริงเช่น ทดลองการอพยพผู้คนหากเกิดฉุกเฉิน เพื่อหาเส้นทางที่ดีที่สุดไว้ใช้งานหากเกิดเหตุการณ์ขึ้นจริง ระบบอินเทอร์เน็ตของสิงค์โปร์มีความเร็วถึง 10-Gbps โดยให้บริการผ่านไฟเบอร์ออฟติก มีการวางระบบการคมนาคม ติดเซนเซอร์และกล้องลงบนถนนหนทางเพื่อพัฒนาสภาพการจราจรทำให้ลดความหนาแน่นของรถบนท้องถนน สิงค์โปร์พยายามทำให้ข้อมูลที่เก็บมาได้เป็นนิรนามเท่าที่จะเป็นไปได้และพยายามเปิดเผยข้อมูลแก่ประชาชนมากเท่าที่สามารถจะเปิดเผยเพื่อให้ประชาชนมีส่วนร่วมในการตัดสินใจพัฒนาประเทศ

2.1.4.2 บาร์เซโลน่า

บาร์เซโลน่าทำการติดตั้งเซนเซอร์เพื่อตรวจสอบและจัดการการจราจรของเมือง โดยมุ่งหวังที่จะจัดเส้นทางการเดินทางของรถใหม่ให้ดียิ่งขึ้นและกระจายความหนาแน่นของรถบนท้องถนน โดยจะลดความหนาแน่นของรถลงจากเดิมถึง 21% แต่แท้จริงแล้วบาร์เซโลน่าทำมากกว่านั้น โดยมี ทั้งระบบที่จอดรถอัจฉริยะ,เซนเซอร์วัดคุณภาพน้ำและอากาศภายในเมือง ,ขยายเครือข่าย Free Wi-Fi ในพื้นที่สาธารณะ บาร์เซโลน่ามุ่งเน้นไปที่การทำให้พลังงานและทรัพยากรในเมืองนั้นยั่งยืน เนื่องจากเมืองเคยมีปัญหาภัยแล้ง เมืองจึงใช้ข้อมูลจากเซนเซอร์ที่ติดตั้งเพื่อวัดปริมาณน้ำฝน มาวิเคราะห์วิธีการกักเก็บน้ำและทำนายปริมาณน้ำฝนที่จะเกิดขึ้น

2.2 ประเภทของข้อมูล

ข้อมูลในโลกนี้มีหลากหลายแบบเพื่อการเป็นระเบียบและนำไปใช้งานได้ง่ายดายขึ้น เนื่องจากบนโลกใบนี้มีงานที่ใช้ข้อมูลในการทำงานที่หลากหลาย จึงทำให้เกิดการแบ่งประเภทข้อมูลตามหลักการหรือกฎเกณฑ์ที่ต่างกันไป ดังนี้

2.2.1 ประเภทของข้อมูลในแง่ของสถิติ

ประเภทของข้อมูลในแง่ของสถิติแบ่งออกเป็น 2 ใหญ่ ได้แก่

2.2.1.1 ข้อมูลประเภท Quantitative

ข้อมูลประเภท Quantitative หมายถึงข้อมูลใดก็ตามที่สามารถวัดค่าออกมาเป็นเลขได้ โดยสามารถแบ่งออกเป็น 2 แบบย่อย ได้แก่

2.2.1.1.1 Continuous Data

Continuous Data คือข้อมูลตัวเลขที่สามารถปรับค่าให้ละเอียดขึ้นได้ โดยข้อมูลยังคงสื่อความหมายได้ถูกต้อง เช่น ส่วนสูงที่สามารถแปลงจากหน่วยเซนติเมตรเป็นมิลลิเมตรได้ หรือแปลงเป็นนาโนเมตรได้ โดยยังคงเป็นข้อมูลตัวเดิมแค่มีความละเอียดที่มากขึ้น

วิธีที่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าฐานนิยม (mode), การหาเปอร์เซ็นต์ (percentile),การหาค่าเฉลี่ย(mean),การหาค่ามัธยฐาน (median)

2.2.1.1.2 Discrete Data

Discrete Data คือข้อมูลที่วัดแล้วไม่สามารถแปลงให้มีความละเอียดหรือแม่นยำไปกว่านี้ไม่ได้แล้ว เช่นการนับจำนวนคน หรือจำนวนรถ คนก็มีเป็นได้แค่จำนวนเต็ม เป็น 2.5คนไม่ได้ รถก็มี 3.5คันไม่ได้

วิธีที่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าฐานนิยม (mode), การหาเปอร์เซ็นต์ (percentile),การหาค่าเฉลี่ย(mean),การหาค่ามัธยฐาน (median)

2.2.1.2 ข้อมูลประเภท Qualitative

ข้อมูลประเภท Qualitative คือข้อมูลที่บ่งบอกถึงลักษณะเฉพาะหรือรายละเอียดของสิ่งๆหนึ่ง โดยวัดเป็นเลขได้ยาก แต่สังเกตได้ง่าย เช่น ขนาด ลักษณะพื้นผิว สี แบ่งออกเป็นประเภทย่อยได้อีก 3 แบบ ได้แก่

2.2.1.2.1 Nominal หรือ Attribute หรือ Categorical Data

คือข้อมูลที่บ่งบอกถึงกลุ่มหรือลักษณะสิ่งต่าง ๆได้แต่ไม่ได้มีความหมายเชิงลึกในเรื่องของลำดับก่อนหลัง คือแบ่งกลุ่มได้ แต่ จัดลำดับไม่ได้เช่น เพศ ชนิดของยา ศาสนา ประเทศ วิธีที่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าฐานนิยม, การหาเปอร์เซ็นต์

วิธีที่ไม่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าเฉลี่ย,การหาส่วนเบี่ยงเบนมตราฐาน

2.2.1.2.2 Binomial

คือข้อมูลNominal ที่มีค่าที่เป็นไปได้เพียง 2 ค่า เช่น Good/Bad, True/False, Right/Wrong, Accept/Reject วิธีที่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าฐานนิยม, การหาเปอร์เซ็นต์ วิธีที่ไม่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าเฉลี่ย,การหาส่วนเบี่ยงเบนมาตราฐาน

2.2.1.2.3 Ordinal หรือ Ranked Data

คือข้อมูลที่ค่าต่าง ๆนั้นบ่งบอกถึงลำดับได้ เช่น เล็ก/ปานกลาง/ใหญ่ สว่างมาก/สว่าง/มืด เตี้ย/ปานกลาง/สูง พอใจ/เฉยๆ/ไม่ชอบ วิธีที่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าฐานนิยม, การหาเปอร์เซ็นต์ วิธีที่ไม่ควรใช้สรุปความของข้อมูลประเภทนี้ คือ การหาค่าเฉลี่ย,ส่วนเบี่ยงเบนมตราฐาน

2.2.2 ประเภทของข้อมูลในแง่ของการเขียนโปรแกรมคอมพิวเตอร์

ในการเขียนโปรแกรมคอมพิวเตอร์มีการเก็บข้อมูลที่ต่างจากแบบสถิติเนื่องจากจะส่งผลถึงพื้นที่และวิธีการประมวลและจัดเก็บข้อมูลเหล่านั้นบนคอมพิวเตอร์ โดยเราสามารถแบ่งประเภทข้อมูลเป็นประเภทหลักๆได้ 5 แบบ ได้แก่

2.2.2.1 ข้อมูลประเภท Integers

Integers คือข้อมูลตัวเลขที่เป็นจำนวนเต็มเท่านั้น

2.2.2.2 ข้อมูลประเภท Booleans

Booleans คือข้อมูลที่บ่งบอกค่าความจริง เช่น True/False

2.2.2.3 ข้อมูลประเภท Characters

Characters คือข้อมูลที่เป็นตัวอักษรและไม่ใช่เลข

2.2.2.4 ข้อมูลประเภท Floating-Point Numbers

Floating-Point Numbers คือข้อมูลที่เป็นตัวเลขทศนิยม

2.2.2.5 ข้อมูลประเภท String

String คือข้อมูลที่เป็นตัวอักษรมากหว่าหนึ่งตัวขึ้นไปรวมถึงสามารถประกอบไปด้วยเลขและอักขระพิเศษเช่น _, * เป็นต้น

2.3 ประเภทของกระบวนการประมวลผลข้อมูล

เมื่อทำการเก็บข้อมูลมาได้แล้วเราจะต้องนำข้อมูลนั้นมาประมวลต่อ ลักษณะการประมวลผลสามารถแบ่งได้เป็น 3 แบบ ได้แก่

2.3.1 Batch Processing

เป็นการประมวลผลโดยแบ่งเป็นชุด ๆแล้วส่งเข้าไปประมวลตามลำดับ เมื่อได้มาอาจไม่ทำการประมวลผลเลยอาจจะพักไว้ก่อนจนถึงจำนวนที่กำหนดไว้หรือจนถึงเวลาที่กำหนดไว้จากนั้นค่อยส่งไปประมวลผล เช่น ข้อมูลจากระบบคิดเงินเดือน,ระบบคำนวณบิล ที่ได้รับข้อมูลมาอาจจะพักไว้ก่อนและนำมาประมวลผลก็ต่อเมื่อถึงรอบเบิก

2.3.2 Real Time Processing

เป็นการประมวลทันทีที่ได้รับข้อมูลเข้าสู่ระบบ มักจะเป็นงานรูปแบบเดิมคงที่ เช่น ระบบธนาคาร,การจองตั๋วหนัง,จองตั๋วเครื่องบิน,จองที่พักหรือจองตั๋วรถไฟ เป็นต้น

2.3.3 Online Processing

เป็นการประมวลผลเมื่อมีคำสั่งเข้ามา โดยคำสั่งนั้นเป็นคำสั่งที่นาน ๆเกิดขึ้นที หรือมีความเฉพาะเจาะจง ไม่ใช่รูปแบบเดิมซ้ำ ๆ

2.4 ภาษาซีชาร์ป

ภาษาซีชาร์ป (C#,C Sharp) ก่อตั้งโดยบริษัทไมโครซอฟต์ เป็นภาษาที่อยู่ในตระกูลเดียวกับภาษา C และ C++ โดยเป็นภาษาที่มีการพัฒนาให้ก้าวหน้ากว่า ภาษา C# ภาษา C# ปรากฏครั้งแรกในปี 2000 มีจุดประสงค์คือเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework การสร้างและพัฒนาภาษาC๒ ได้รับอิทธิผลมาจากภาษาที่เกิดก่อนหน้า เช่นภาษา C++ หรือ JAVA โดยพยายามสืบทอดข้อดีและลบข้อบกพร่องของภาษาก่อนหน้าและเพิ่มความเป็น object oriented programming (OOP) ลงไป จนกลายมาเป็นภาษา C# ในตอนเริ่มแรกภาษา C#ถูกสร้างและกำหนดมาตรฐานโดยบริษัทไมโครซอฟต์เอง ในภายหลังได้ถูกรับรองจากหน่วยงาน ECMA และ ISO จวบจนถึงปัจจุบันภาษาC#ยังคงถูกพัฒนาโดยไมโครซอฟต์มาอย่างต่อเนื่อง ปัจจุบันเวอร์ชั่นล่าสุดของภาษาC#คือ 7.2

2.4.1 ตัวอย่างการเขียนโปรแกรม ภาษา C#

โปรแกรม 2.1 Hello World

```
public class example
{  public static void Main()
     {
          System.Console.WriteLine("Hello, World!");
     }
}
```

2.4.2 จุดเด่นของภาษาC#

1) ด้วยความเป็น OOP ทำให้สะดวกในการพัฒนางาน

2) มีประสิทธิภาพสูง เพราะถูกพัฒนาขึ้นภายใต้ .NET Framework ทำให้สามารถใช้ความสามารถของเทคโนโลยีของ .NET ออกมาใช้ได้

3) ทำงานร่วมกับภาษา XML ได้อย่างราบรื่นเพราะเทคโนโลยี .NET Framework

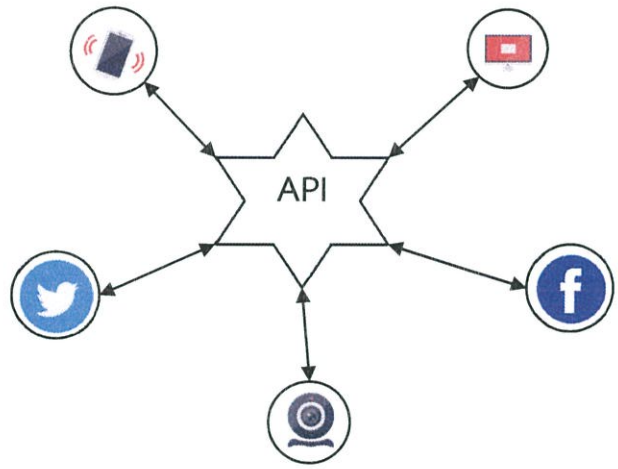
4) มี Generic Type ทำให้มีความยืดหยุ่นในการพัฒนางาน

5) มีเทคโนโลยี LINQ ทำในทำงานกับฐานข้อมูลได้อย่างมีประสิทธิภาพ

6) ใน IDE ต่าง ๆมีกลไกที่ช่วย debug ในภาษา C# ที่สะดวก

2.5 การสื่อสารข้อมูลผ่าน API

API ย่อมาจาก Application Programing Interface คือช่องทางการติดต่อระหว่างเว็บไซต์,ระบบ ,ซอฟต์แวร์ หรือฮาร์ดแวร์โดยมีการกำหนดการรูปแบบของคำร้องขอ (Request) และผลตอบรับ (Response) ไว้อย่างชัดเจน เพื่อให้การสื่อสารเป็นไปอย่างราบรื่นและเป็นระเบียบ หากต้องการจะใช้งานAPIใด ผู้ที่ต้องการใช้งานAPI จะต้องทำการศึกษารูปแบบการใช้งานของAPIจากผู้ให้บริการ APIนั้นก่อน ทั้งรูปแบบของการส่งคำร้องขอและผลตอบรับ เมื่อทราบแล้วจึงสามารถทำการส่งคำร้องขอได้ โดยอาจมีการใส่ค่าข้อมูลที่ผู้ให้บริการต้องการ (Parameter) ส่งไปหาผู้ให้บริการอย่างถูกต้อง เมื่อได้รับคำร้องขอที่ถูกต้อง ผู้ให้บริการจะส่งผลตอบรับกลับมาในรูปแบบที่ผู้ใช้งานทราบอยู่ล่วงหน้าว่าจะได้รับมาในรูปแบบใด การใช้API เปรียบเสมือนการเรียกใช้ฟังก์ชันหรือบริการของผู้อื่น โดยต้องกรอกแบบฟอร์มการขอใช้บริการให้ถูกต้อง ฟังก์ชันหรือบริการต่าง ๆที่มีให้เลือกใช้เป็นสิทธิ์ของผู้ให้บริการว่าจะมีบริการใดให้ใช้งานบ้าง



รูป 2.3 ภาพรวมของการสื่อสารข้อมูลด้วย API

2.5.1 ประโยชน์ของAPI

1) เป็นการเพิ่มมีประสิทธิภาพให้ซอฟต์แวร์, ฮาร์ดแวร์ หรือ ระบบ การทำ API Service เปรียบเสมือนการแบ่งปันแบ่งปันบริการหรือฟังก์ชันการทำงานให้ผู้อื่นใช้ได้ ทำให้การพัฒนางานเกิดขึ้นอย่างมีประสิทธิภาพ ลดระยะเวลาการพัฒนาลง เพราะหากมี API ที่สามารถให้บริการในฟังก์ชันหรือการประมวลผลที่เราต้องการได้แล้ว ก็ไม่จำเป็นจะต้องพัฒนาฟังก์ชันหรือบริการแบบนั้นขึ้นมาใหม่

2) API เป็นวิธีการสื่อสารที่เข้าถึงได้ง่าย API Service ทำให้การแลกเปลี่ยนสื่อสารข้อมูลและบริการเกิดขึ้นได้ง่ายขึ้นเนื่องจากเป็นช่องทางการสื่อสารที่เสียค่าใช้จ่ายน้อย ไม่ได้ต้องการอุปกรณ์ใดเพิ่มเติม เว้นเสียแต่การเขียนโปรแกรมมาเพื่อรองรับการใช้งาน

3) เพิ่มโอกาสการทำงานร่วมกันระหว่างบริษัทหรือหน่วยงานที่ต่างกันสามารถทำงาน ร่วมกันได้โดยไม่มีการล่วงล้ำข้อมูลภายในของกันและกันโดยสื่อสารแลกเปลี่ยนข้อมูลผ่าน API Service

2.5.2 ชนิดของAPI

2.5.2.1 Web Services API

Web Service API คือ API ที่ให้บริการผ่านทาง World Wide Web โดยประกอบไปด้วย REST (Representational State Transfer), SOAP (Simple Object Access Protocol), XML_RPC และ JSON_RPC

2.5.2.2 Source Code API

แบ่งออกเป็น Library Based และ Class Based

Library Based จะอ้างอิงจากการเขียนโปรแกรมใน Library และใช้ฟังก์ชันหรือกิจกรรม (Routine)ในLibraryนั้นในการทำงานหรือแลกเปลี่ยนข้อมูล

Class Based อ้างอิงจากคลาสที่ประกาศใน Object Oriented Languages

กล่าวคือทั้งสองแบบจะต้องมีการติดตั้งส่วนของการเขียนโปรแกรม (Code) ที่ใช้อ้างอิงในการเรียกใช้ API ลงในซอฟต์แวร์หรือระบบของตนเองเพื่อให้ใช้งานได้

2.5.2.3 Object Remoting API

ใช้ Remoting Protocol,COBRA (Common Object Request Broker Architecture) มาร่วมในการเรียกใช้งาน API

2.5.2.4 Hardware API

ใช้ในการเข้าถึงการทำงานในระดับฮาร์ดแวร์ เช่นการอ่านค่าอุณหภูมิ เข้าถึงข้อมูลของฮาร์ดแวร์ เช่น Model Number, Serial Number เป็นต้น

2.5.3 ประเภทของ Web API

API ที่เลือกใช้ในโครงงานคือ Web API แบบ REST-JSON โดย Web API แบ่งเป็น 2 โปรโตคอล คือ REST และ SOAP

2.5.3.1 API แบบ SOAP

SOAP ย่อมาจาก Simple Object Access Protocol สร้างโดยบริษัทไมโครซอฟต์ พัฒนามาเพื่อมาแทนที่ Distributed Component Object Model (DCOM) และ COBRA ซึ่งมีข้อเสีย เพราะต้องใช้คู่กับรูปแบบข้อความแบบไบนารีเท่านั้น จึงพัฒนา SOAP มาใช้คู่กับ XML ซึ่งเป็นรูปแบบข้อความมนุษย์อ่านได้ หากต้องการใช้ SOAP มักจะต้องใช้คู่กับ XML เสมอ การใช้ API แบบนี้อาจจะช้ากว่าแบบอื่นเนื่องจากส่วนการส่งคำร้องขอ (Request) มีความซับซ้อนและไม่ทนต่อความผิดพลาด (Error) แต่สามารถทำให้ใช้งานง่ายขึ้นได้ หากใช้ร่วมกับ .Net Framework โดยเลือกโปรโตคอลในการส่งข้อมูลตามหลัก Data Communication ได้ 2 แบบ คือ HTTP และ SMTP

2.5.3.2 API แบบ REST

REST ย่อมาจาก Representational State Transfer API แบบนี้ใช้งานได้ง่ายกว่า การเขียนโปรแกรมเพื่อส่งคำร้องขอ (Request) สั้นกว่าแบบSOAP ใช้เพียงการเข้าถึง World Wide Web ผ่านทาง URL (Uniform Resource Locator) แต่บางครั้งอาจต้องเพิ่มพารามิเตอร์อื่น เช่น ชื่อผู้ใช้หรือรหัสผ่านในการเข้าใช้บริการ API Service โดยส่งข้อมูลสื่อสารกันผ่านโปรโตคอล HTTP การใช้ API แบบ REST สามารถใช้งานได้ผ่าน 4 คำสั่งหลัก คือ GET,POST,PUT และ DELETE

GET ใช้สำหรับการดึงข้อมูลภายใน URL

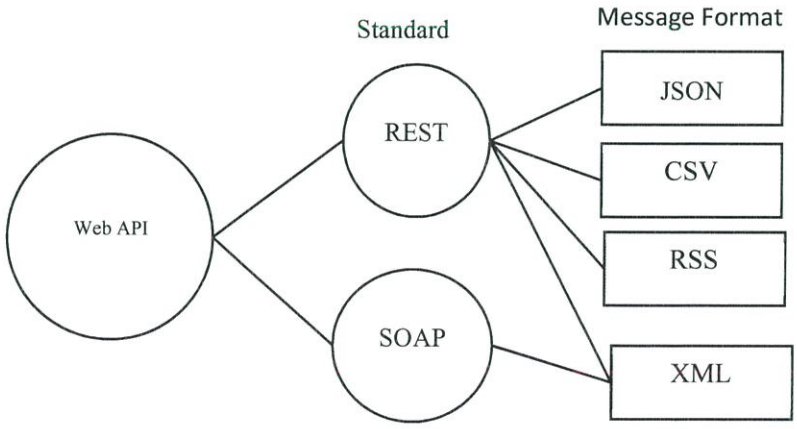
POST ใช้สำหรับสำหรับสร้างข้อมูลใหม่ภายใน URL

PUT ใช้สำหรับแก้ไขข้อมูลภายใน URL

DELETE ใช้สำหรับลบข้อมูลภายใน URL

รูปแบบข้อความที่ใช่คู่กับ REST มีหลายประเภทให้เลือกเช่น .csv, .json, .rss ทำให้สามารถเลือกใช้รูปแบบที่เหมาะสมกับงานที่ทำอยู่ได้

ตัวอย่างการเข้าถึงAPIผ่าน URL : http://APIkmitl.ac.th/getAPIexample/example.json



รูป 2.4 ภาพรวมของ Web API

2.5.3.3 ความแตกต่างระหว่าง REST กับ SOAP

กล่าวได้ว่า API แบ่งตามจุดประสงค์ที่ใช้ติดต่อสื่อสารได้หลายประเภท ในแต่ละประเภทย่อยนั้นก็แบ่งตามมาตรฐานที่ใช้ได้อีก โครงงานของเราเลือกใช้ Web API โดย Web API แบ่งตามมาตรฐานที่ใช้ได้เป็น REST และ SOAP และรูปแบบของข้อความที่ใช้คู่กับโปรโตคอล 2 โปรโตคอลนี้มีหลายรูปแบบ เช่น JSON หรือ XML เป็นต้น โดยหากเลือกใช้โปรโตคอล SOAP จะต้องใช้รูปแบบข้อความแค่แบบ XML เท่านั้น แต่ถ้าแบบ REST จะมีตัวเลือกให้เลือกได้มากกว่า

***ตาราง* 2.1 *ข้อแตกต่างระหว่าง* REST API *และ* SOAP API**

SOAP	REST
- สามารถเลือกโปรโตคอลในการส่งข้อมูลได้ระหว่าง HTTP และ SMTP - ใช้งานได้ดีในงานแบบ Distributed Enterprise Environment - มีกลไกการรองรับความผิดพลาดในตัวเอง (Built-In Error Handling) - ดีต่องานระยะยาวที่มีการเปลี่ยนแปลงน้อย	- ไม่สามารถเลือกโปรโตคอลในการส่งข้อมูลได้ ใช้ได้เพียงแบบเดียวคือแบบ HTTP - ใช้งานง่าย - มีภาระ (overload) น้อย - ใช้เวลาในการเรียนรู้เพื่อที่จะใช้งานสั้น - เลือกรูปแบบข้อความได้หลากหลายกว่า - เทคโนโลยีที่ใช้มีความทันสมัยเข้าใกล้เว็บเทคโนโลยีมากกว่า

2.6 Data Visualization

เมื่อเรามีข้อมูลเชิงปริมาณ (Quantitative Data) จำนวนมากและมาจากหลากหลายแหล่งข้อมูล การบริโภคข้อมูลเข้าไปอาจจะทำได้ยากขึ้น เพราะมีเลขจำนวนมากที่ต้องอ่าน มองเห็นภาพรวมหรือความเกี่ยวข้องของข้อมูลแต่ละอย่างได้อย่างลำบาก Data Visualization เป็นอีกทางเลือกหนึ่งที่ทำให้การบริโภคและการมองหาภาพรวมและการมองหาความสัมพันธ์ระหว่างข้อมูลเป็นไปได้อย่างง่ายดายขึ้น โดยนำข้อมูลเชิงปริมาณมาแสดงผลผ่านภาพ เช่น แผนภูมิ กราฟ รูปภาพ แผนผัง ฯลฯ เพื่อให้ข้อมูลดูน่าสนใจ เข้าใจได้ง่าย มองเห็นภาพรวมได้ ง่ายต่อการนำเสนอแก่ผู้อื่น

2.7 Machine Learning

Machine Learning คือการประยุต์การใช้งานปัญญาประดิษฐ์ (Artificial Intelligence) ทำให้คอมพิวเตอร์มีความสามารถในการเรียนรู้ด้วยตนเอง เมื่อมีข้อมูลเข้ามาให้ตัดสินใจ, เลือกหรือทำนาย คอมพิวเตอร์ก็จะสารมารถทำได้เอง

2.7.1 ประเภทของ Machine Learning

ประเภทของ Machine Learning สามารถจำแนกตามหน้าที่และขอบเขตได้หลักๆ 3 ประเภทดังนี้

2.7.1.1 Supervised Learning

การเรียนรู้ของ Machine Learning เกิดขึ้นโดยผ่านการเทรนจากข้อมูลที่มีการทำนายหรือแยกประเภทไว้อยู่แล้ว Machine Learning จะทำการเรียนรู้จากข้อมูลเหล่านั้นเพื่อที่สามารถทำนายหรือแยกประเภท(Classification) ข้อมูลที่จะถูกส่งเข้ามาใหม่ในอนาคต

2.7.1.2 Unsupervised Learning

การเรียนรู้ของ Machine Learning แบบ Unsupervised Learning ตรงข้ามจากแบบ Supervised Learning คือ ข้อมูลที่ถูกนำมาสอนให้แก่ Machineจะไม่มีการระบุประเภทหรือแบ่งกลุ่มมาก่อน การแบ่งกลุ่มนี้เรียกว่า Clustering ซึ่งต่างจาก Classification เพราะการแบ่งแบบ Classification จะรู้แน่ชัดว่ามีกี่ประเภทตามจำนวนประเภทตามที่เราได้ใส่ลงไปในข้อมูลที่ใช้สอน แต่ Clustering เป็นการจัดกลุ่มโดยไม่ทราบขอบเขตล่วงหน้าว่าจะมีกี่กลุ่ม

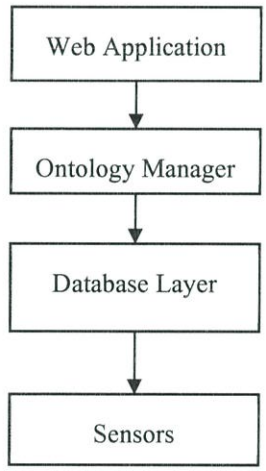
2.7.1.3 Reinforcement Learning

การเรียนรู้แบบ Reinforcement Learning หรือการเรียนรู้แบบเสริมกำลัง เป็นการสอน Machine ให้เรียนรู้แบบ9yตัดสินใจกระทำจากนั้นให้รางวัลหรือลงโทษโดยสนใจสภาวะแวดล้อมหรือสภาวะขณะนั้นเป็นพิเศษ โดยรางวัลอาจจะได้เลยหลังจากจัดสินใจกระทำหรืออาจจะได้ในภายหลังก็ได้ ตัวอย่างการใช้งาน เช่น AlphaGo ของ Google

2.8 ผลงานวิจัยที่เกี่ยวข้อง

2.8.1 A scalable middleware-based infrastructure for energy management and visualization in city districts

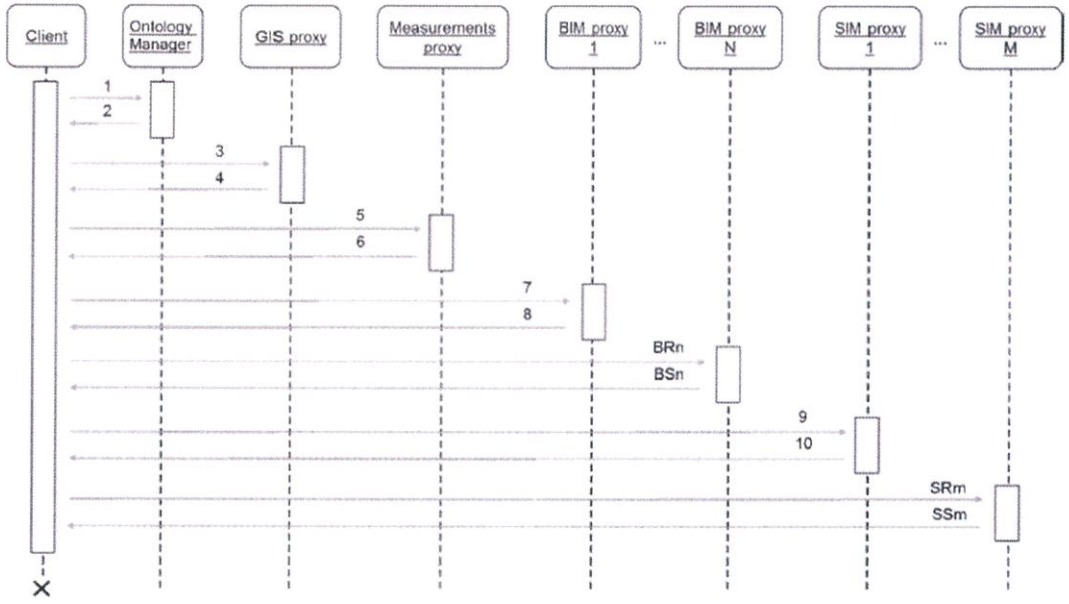
เนื่องด้วยการทำเมืองอัจฉริยะในประเทศ Italy ต้องการจัดการและวิเคราะห์อัตราการใช้พลังงานและปัจจัยที่ส่งผลต่ออัตราการใช้พลังงานในระดับเขต แต่เนื่องด้วยการติดตั้งเซนเซอร์มีหลากหลายชนิดเช่น วัดคุณภาพน้ำ,อากาศ,การใช้ไฟฟ้า และหลากหลายยี่ห้อ ทำให้ยากในการเข้าถึงหรือมองภาพรวมของข้อมูล อีกทั้งหากมีการติดตั้งเซนเซอร์แบบใหม่เข้ามาอาจต้องแก้ไขการเขียนระบบเกิดขึ้น ซึ่งจะส่งผลกระทบกับผู้ที่กำลังใช้งานระบบต้องหยุดชะงักได้ ทางผู้จัดทำการวิจัยนี้จึงมีจุดประสงค์ที่จะสร้างระบบส่วนกลาง (middleware) ที่รองรับการเพิ่มของเซนเซอร์โดยที่ไม่จำเป็นต้องปิดปรับปรุงระบบทุกครั้งที่มีการเพิ่มเซนเซอร์ โดยมีวิธีการทำงานดังนี้



รูป 2.5 โครงสร้างของงานวิจัย scalable middleware-based infrastructure

1) ส่วนWeb Application ที่มีหน้าที่ติดต่อกับผู้ใช้งาน โยผู้ใช้งานสามารถเลือกพื้นที่ที่ต้องการให้แสดงผลข้อมูลต่าง ๆ ได้ โดยเมื่อผู้ใช้งานเลือกเสร็จจะ REST HTTP query ต่อไปที่ Ontology Manager
2) Ontology Manager เป็นตรวจประสานงานเรียกข้อมูลจากเซนเซอร์ที่มีทั้งหมดในระบบ โดยจะทำการส่ง SPARQL query เข้าไปใน Database Layer เช่นผู้ใช้งานต้องการข้อมูลของแขวง A ในเขต Turin ของประเทศItaly

3) Database layer เป็นการรวบรวมข้อมูลจากเซนเซอร์ทั้งหมดในเขตมาไว้ในที่เดียว โดยอาจมีการเก็บใน schema ที่ต่างกัน
4) เมื่อ Ontology ค้นหาข้อมูลจากเซนเซอร์หรือฐานข้อมูลที่ใช้ในการจำลองตึกที่เกี่ยวข้องเพียงพอแล้วจะส่งข้อมูลกลับไปที่ Web Application จากนั้น Web Application ก็จะทำการแสดงผลข้อมูลตามผลข้อมูลที่ได้รับ



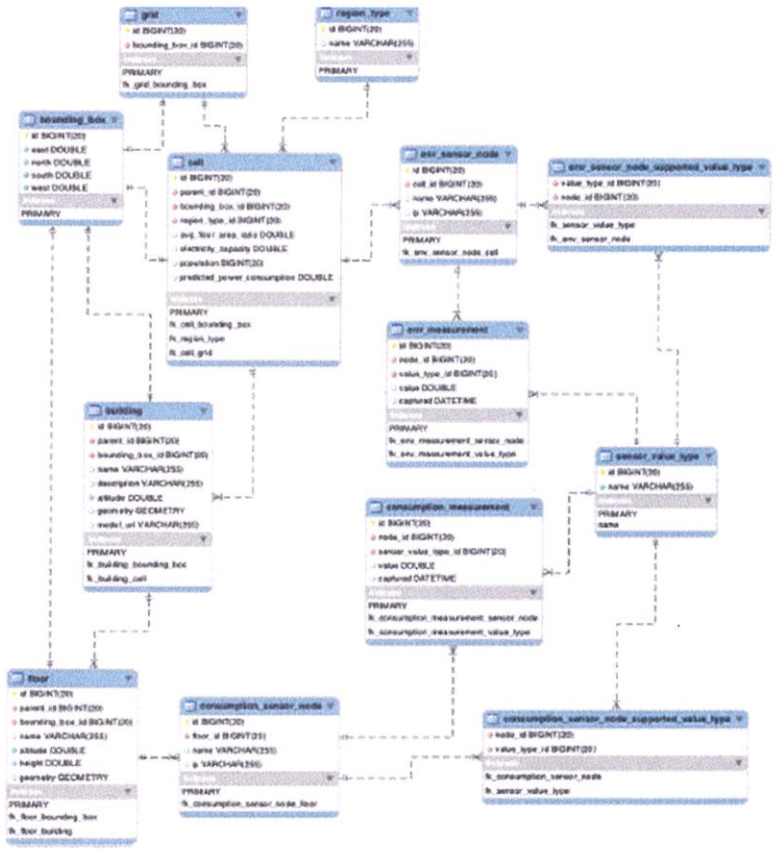
รูป 2.6 ลำดับการ Query ข้อมูลก่อนส่งกลับสู่ Web Application

กล่าวได้ว่าระบบพยายามสร้างตัวกลางในการเตรียมการข้อมูลจากเซนเซอร์หลายประเภทที่ถูกเก็บอยู่ที่ศูนย์กลางเดียวเพื่อต้องการให้ระบบรองรับการเพิ่มจำนวนและการเพิ่มรูปแบบของเซนเซอร์โดยที่ไม่ต้องทำการแก้ไขส่วนWeb Application ซึ่งมีหน้าที่แสดงผลข้อมูลและติดต่อกับผู้ใช้ โดยใช้กลไกการจัดหาข้อมูลในขอบเขตพื้นที่ที่ผู้ใช้งานต้องการจากนั้นส่งข้อมูลที่หามาได้จากฐานข้อมูลของระบบกลับไปที่ Web application เพื่อแสดงผลการ Visualization

2.8.2 Integrated energy monitoring and visualization system for Smart Green City development

รัฐบาลเกาหลีพยายามที่จะสร้างระบบการแสดงผลเพื่อการพัฒนาเมือง โดยแสดงผลจากเซนเซอร์หลากหลายชนิดและต้องการแสดงผลให้ตรงกับสถานที่ติดตั้งจริง ในส่วนของการแสดงผลด้านพิกัดสถานที่ โครงงานวิจัยชิ้นนี้ได้นำ Google Earth and Google Maps plug-ins เข้ามา

มีส่วนร่วม เนื่องจากเซนเซอร์ที่ต่างกันมีการจัดการการเข้าถึงข้อมูลที่ต่างกัน เพื่อแก้ปัญหานี้ โครงงานวิจัยชิ้นนี้จึงมีจุดมุ่งหมายที่จะพัฒนาโครงสร้างฐานข้อมูลส่วนกลางที่เอื้อต่อการมีเซนเซอร์ที่หลากหลายชนิดและรองรับข้อมูลจำนวนมาก รวมถึงยังต้องการให้การออกแบบนี้มีส่วนช่วยในการเพิ่มประสิทธิภาพ Visualization ในแสดงผลได้อย่างราบรื่นบน Web browser โดยยึดหลักการ LOD (Levels of details)



รูป 2.7 โครงสร้างฐานข้อมูลของ Integrated energy monitoring and visualization system

จะพบได้ว่าปัญญาหลักๆที่ทั้งสองโครงงานวิจัยนี้เผชิญคือความหลากหลายของเซนเซอร์ที่สร้างความยุ่งยากและลดประสิทธิภาพการทำงานของระบบการแสดงผล โดยทั้งสองโครงงานวิจัยใช้วิธีการแก้ไขปัญหาที่แตกต่างกัน ทั้งนี้ทั้งนั้นการแก้ปัญหาทั้งสองยังคงต้องการมนุษย์เข้ามาช่วยเหลือในการตั้งค่าหรือแก้ไขระบบซึ่งอาจทำให้เกิดความล่าช้าในการแสดงผลข้อมูลใหม่ๆ ไม่มีระบบแนะนำการแสดงผลทำให้ผู้ใช้งานต้องเลือกสิ่งที่อยากให้แสดงผลเอง ซึ่งอาจสร้างความยุ่งยากในกรณีที่มีข้อมูลหลายแบบมากให้เลือก ซึ่งต่างจากโครงงานของทางคณะผู้จัดทำที่มีพยายามจะแก้ปัญหาเดียวกันโดยลดการเข้ามาเกี่ยวข้องข้องมนุษย์และลดความซับซ้อนในการเลือกแสดงผลข้อมูลด้วยระบบแนะนำการแสดงผล

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 วิเคราะห์ข้อมูลที่เกี่ยวข้องกับระบบเมืองอัจฉริยะ

3.1.1 ศึกษาข้อมูล

ในขั้นต้นทางคณะผู้จัดทำได้ทำการศึกษาข้อมูลจากระบบเมืองอัจฉริยะ เช่น ข้อมูลดิบที่ได้รับจากเซ็นเซอร์ชนิดต่าง ๆ ข้อมูลที่ผ่านการวิเคราะห์ผลจากค่าเซ็นเซอร์ ข้อมูลที่ได้จากสื่อสังคมออนไลน์ รวมไปถึงข้อมูลด้านอื่น ๆ ที่เปิดให้บุคคลทั่วไปใช้งานได้ เพื่อเป็นตัวอย่างข้อมูลที่จะนำมาใช้ในการทดสอบระบบ และหาวิธีการในการออกแบบระบบได้อย่างถูกต้องและแม่นยำที่สุด โดยข้อมูลส่วนใหญ่ที่นำมาใช้งานนั้น มุ่งเน้นไปยังข้อมูลที่เกี่ยวข้องกับระบบเมืองอัจฉริยะ เพื่อให้ผลจากการวิเคราะห์ข้อมูลตรงกับกลุ่มข้อมูลเป้าหมายที่ต้องการนำมาใช้งานมากที่สุด

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
1. Geoname.org		
1.1. Postal Code Search	http://www.geonames.org/export/web-services.html	ภาคผนวก ตัวอย่างข้อมูล ก.1
2. Wunderground.com		
2.1. A severe alert	https://www.wunderground.com/weather/API/d/docs?d=data/alerts&MR=1	ภาคผนวก ตัวอย่างข้อมูล ก.2
2.2. Reporting stations	https://www.wunderground.com/weather/API/d/docs?d=data/almanac&MR=1	ภาคผนวก ตัวอย่างข้อมูล ก.3
2.3. Astronomy	https://www.wunderground.com/weather/API/d/docs?d=data/astronomy	ภาคผนวก ตัวอย่างข้อมูล ก.4
3. Travelpayouts.com		

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
3.1. Hotels name	https://support.travelpayouts.com/hc/en-us/articles/115000343268-Hotels-data-API	ภาคผนวก ตัวอย่างข้อมูล ก.5
3.2. Hotels location	https://support.travelpayouts.com/hc/en-us/articles/115000343268-Hotels-data-API	ภาคผนวก ตัวอย่างข้อมูล ก.6
3.3. Hotels cost	https://support.travelpayouts.com/hc/en-us/articles/115000343268-Hotels-data-API	ภาคผนวก ตัวอย่างข้อมูล ก.7
4. Wefeelfine.org		
4.1. Sentimental	http://wefeelfine.org/API.html	ภาคผนวก ตัวอย่างข้อมูล ก.8
5. Walkscore.com		
5.1. Near bus stops	https://www.walkscore.com/professional/public-transit-API.php#search_stops	ภาคผนวก ตัวอย่างข้อมูล ก.9
5.2. Routes and stops	https://www.walkscore.com/professional/public-transit-API.php#search_network	ภาคผนวก ตัวอย่างข้อมูล ก.10
6. Ura.gov.sg		
6.1. Car park availability	https://www.ura.gov.sg/maps/ura/ura_dataservice/samples/Car_Park_Availability.txt	ภาคผนวก ตัวอย่างข้อมูล ก.11
6.2. Car park locations and rates	https://www.ura.gov.sg/maps/ura/ura_dataservice/samples/Car_Park_Details.txt	ภาคผนวก ตัวอย่างข้อมูล ก.12
6.3. Car park locations and rates for application	https://www.ura.gov.sg/maps/ura/ura_dataservice/samples/Season_Car_Park_Details.txt	ภาคผนวก ตัวอย่างข้อมูล ก.13
6.4. Past 3 years of transaction records	https://www.ura.gov.sg/maps/ura/ura_dataservice/samples/PMI_Resi_Transaction.txt	ภาคผนวก ตัวอย่างข้อมูล ก.14
7. Aqicn.org		
7.1. Air quality	http://aqicn.org/json-API/doc/#API-City_Feed-GetCityFeed	ภาคผนวก ตัวอย่างข้อมูล ก.15
8. Freegeoip.net		

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
8.1. Search location from IP	https://freegeoip.net/?q=49.49.242.221	ภาคผนวก ตัวอย่างข้อมูล ก.16
9. Citybik.es		
9.1. City bikes	http://API.citybik.es/v2/	ภาคผนวก ตัวอย่างข้อมูล ก.17
10. Aerisweather.com		
10.1.Currently active alerts	https://www.aerisweather.com/support/docs/API/reference/endpoints/advisories/	ภาคผนวก ตัวอย่างข้อมูล ก.18
10.2.Core details of cAPItals	https://www.aerisweather.com/support/docs/API/reference/endpoints/countries/	ภาคผนวก ตัวอย่างข้อมูล ก.19
10.3.Forecasts data	https://www.aerisweather.com/support/docs/API/reference/endpoints/forecasts/	ภาคผนวก ตัวอย่างข้อมูล ก.20
10.4.Core details of places	https://www.aerisweather.com/support/docs/API/reference/endpoints/places/	ภาคผนวก ตัวอย่างข้อมูล ก.21
10.5.Core details of airports	https://www.aerisweather.com/support/docs/API/reference/endpoints/places-airports/	ภาคผนวก ตัวอย่างข้อมูล ก.22
11. Openweathermap.org		
11.1.Weather data	https://openweathermap.org/current	ภาคผนวก ตัวอย่างข้อมูล ก.23
11.2.Forecast 5 day	https://openweathermap.org/forecast5	ภาคผนวก ตัวอย่างข้อมูล ก.24
11.3.Forecast 16 day	https://openweathermap.org/forecast16	ภาคผนวก ตัวอย่างข้อมูล ก.25
11.4.Ultraviolet index	https://openweathermap.org/API/uvi	ภาคผนวก ตัวอย่างข้อมูล ก.26
11.5.Weather stations	https://openweathermap.org/stations	ภาคผนวก ตัวอย่างข้อมูล ก.27

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
11.6.Weather Alerts	https://openweathermap.org/triggers	ภาคผนวก ตัวอย่างข้อมูล ก.28
11.7.Carbon Monoxide	https://openweathermap.org/API/pollution/co	ภาคผนวก ตัวอย่างข้อมูล ก.29
12. Weather.mg		
12.1.Air quality	https://API.weather.mg/API-detail-pages/air-quality-parameter.html	ภาคผนวก ตัวอย่างข้อมูล ก.30
12.2.Pollen forecast	https://API.weather.mg/API-detail-pages/air-quality-parameter.html	ภาคผนวก ตัวอย่างข้อมูล ก.31
13. 511ny.org		
13.1.Link definitions	https://511ny.org/developers/help/API/get-API-getmainlinelinkdefinitions_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.32
13.2.Events	https://511ny.org/developers/help/API/get-API-getevents_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.33
13.3.Cameras	https://511ny.org/developers/help/API/get-API-getcameras_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.34
13.4.Message Signs	https://511ny.org/developers/help/API/get-API-getmessagesigns_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.35
13.5.Alerts	https://511ny.org/developers/help/API/get-API-getalerts_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.36
13.6.Winter Road	https://511ny.org/developers/help/API/get-API-getwinterroadconditions_Key_format	ภาคผนวก ตัวอย่างข้อมูล ก.37
14. TransportAPI.com		
14.1.Places detail	https://developer.transportAPI.com/docs?raml=https://transportAPI.com/v3/raml/transportAPI.raml#/uk_car_journey_from_from_to_to_json#uk_places_json	ภาคผนวก ตัวอย่างข้อมูล ก.38

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
14.2.Tweet	https://developer.transportAPI.com/docs?raml=https://transportAPI.com/v3/raml/transportAPI.raml#/uk_car_journey_from_from_to_to_json#uk_buzz_tweets_json	ภาคผนวก ตัวอย่างข้อมูล ก.39
14.3.Journey plan	https://developer.transportAPI.com/docs?raml=https://transportAPI.com/v3/raml/transportAPI.raml#/uk_car_journey_from_from_to_to_json#uk_cycle_journey_from_from_to_to_json	ภาคผนวก ตัวอย่างข้อมูล ก.40
14.4.Station box	https://developer.transportAPI.com/docs?raml=https://transportAPI.com/v3/raml/transportAPI.raml#/uk_car_journey_from_from_to_to_json#uk_tube_stations_bbox_json	ภาคผนวก ตัวอย่างข้อมูล ก.41
15. Ip2location.com		
15.1.Location from IP	https://www.ip2location.com/web-service	ภาคผนวก ตัวอย่างข้อมูล ก.42
16. Readthedocs.io		
16.1.Alert	http://fiware-datamodels.readthedocs.io/en/latest/Alert/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.43
16.2.311 Service	http://fiware-datamodels.readthedocs.io/en/latest/IssueTracking/Open311_ServiceType/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.44
16.3.311 Request	http://fiware-datamodels.readthedocs.io/en/latest/IssueTracking/Open311_ServiceRequest/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.45

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
16.4.Device	http://fiware-datamodels.readthedocs.io/en/latest/Device/Device/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.46
16.5.Air quality	http://fiware-datamodels.readthedocs.io/en/latest/Environment/AirQualityObserved/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.47
16.6.Water quality	http://fiware-datamodels.readthedocs.io/en/latest/Environment/WaterQualityObserved/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.48
16.7.Noise level	http://fiware-datamodels.readthedocs.io/en/latest/Environment/NoiseLevelObserved/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.49
16.8.Off street parking	http://fiware-datamodels.readthedocs.io/en/latest/Parking/OffStreetParking/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.50
16.9.Parking spot	http://fiware-datamodels.readthedocs.io/en/latest/Parking/ParkingSpot/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.51
16.10. Garden	http://fiware-datamodels.readthedocs.io/en/latest/ParksAndGardens/Garden/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.52
16.11. Greenspace	http://fiware-datamodels.readthedocs.io/en/latest/ParksAndGardens/GreenspaceRecord/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.53
16.12. Flower bed	http://fiware-datamodels.readthedocs.io/en/latest/ParksAndGardens/FlowerBed/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.54

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
16.13. Street light	http://fiware-datamodels.readthedocs.io/en/latest/StreetLighting/Streetlight/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.55
16.14. Waste container	http://fiware-datamodels.readthedocs.io/en/latest/WasteManagement/WasteContainer/doc/spec/index.html	ภาคผนวก ตัวอย่างข้อมูล ก.56
17. Noaa.gov		
17.1.Noaa stations	https://www.ncdc.noaa.gov/cdo-web/webservices/v2#stations	ภาคผนวก ตัวอย่างข้อมูล ก.57
18. Yahoo.com		
18.1.Forecast	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.58
18.2.Wind	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.59
18.3.Conditions	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.60
18.4.Phase condition	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.61
18.5.Sunset time	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.62
18.6.Weather	https://developer.yahoo.com/weather/	ภาคผนวก ตัวอย่างข้อมูล ก.63
19. Breezometer.com		
19.1.Air quality	https://breezometer.com/air-quality-API/	ภาคผนวก ตัวอย่างข้อมูล ก.64
20. Data.gov.sg		
20.1.Car park	https://data.gov.sg/dataset/carpark-availability	ภาคผนวก ตัวอย่างข้อมูล ก.65

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
20.2.Ultra-violet	https://data.gov.sg/dataset/ultraviolet-index-uvi	ภาคผนวก ตัวอย่างข้อมูล ก.66
20.3.Weather	https://data.gov.sg/dataset/realtime-weather-readings	ภาคผนวก ตัวอย่างข้อมูล ก.67
20.4.Traffic	https://data.gov.sg/dataset/traffic-images	ภาคผนวก ตัวอย่างข้อมูล ก.68
20.5.PM 2.5	https://data.gov.sg/dataset/pm2-5	ภาคผนวก ตัวอย่างข้อมูล ก.69
20.6.Taxi	https://data.gov.sg/dataset/taxi-availability	ภาคผนวก ตัวอย่างข้อมูล ก.70
20.7.Forecast 24 hour	https://data.gov.sg/dataset/weather-forecast?resource_id=9a8bd97e-0e38-46b7-bc39-9a2cb4a53a62	ภาคผนวก ตัวอย่างข้อมูล ก.71
20.8.Pollutant	https://data.gov.sg/dataset/psi	ภาคผนวก ตัวอย่างข้อมูล ก.72
21. Data.cityofnewyork.us		
21.1.311 requests	https://data.cityofnewyork.us/dataset/311-Service-Requests-From-2013/hybb-af8n	ภาคผนวก ตัวอย่างข้อมูล ก.73
21.2.Alert	https://dev.socrata.com/foundry/data.cityofnewyork.us/6b9w-uyrq	ภาคผนวก ตัวอย่างข้อมูล ก.74
21.3.Recycling Bins	https://dev.socrata.com/foundry/data.cityofnewyork.us/ggvk-gyea	ภาคผนวก ตัวอย่างข้อมูล ก.75
21.4.Farmers markets	https://dev.socrata.com/foundry/data.cityofnewyork.us/cw3p-q2v6	ภาคผนวก ตัวอย่างข้อมูล ก.76
22. Transport.opendata.com		
22.1.Locations	https://transport.opendata.ch/docs.html	ภาคผนวก ตัวอย่างข้อมูล ก.77

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
22.2.Connections	https://transport.opendata.ch/docs.html	ภาคผนวก ตัวอย่างข้อมูล ก.78
22.3.Station board	https://transport.opendata.ch/docs.html	ภาคผนวก ตัวอย่างข้อมูล ก.79
23. Data.cityofchicago.org		
23.1.Crimes	https://dev.socrata.com/foundry/data.cityofchicago.org/6zsd-86xi	ภาคผนวก ตัวอย่างข้อมูล ก.80
23.2.Police stations	https://dev.socrata.com/foundry/data.cityofchicago.org/6zsd-86xi	ภาคผนวก ตัวอย่างข้อมูล ก.81
23.3.CDPH Notification	https://dev.socrata.com/foundry/data.cityofchicago.org/gr5j-f3dh	ภาคผนวก ตัวอย่างข้อมูล ก.82
23.4.Energy	https://dev.socrata.com/foundry/data.cityofchicago.org/pvq4-8f4k	ภาคผนวก ตัวอย่างข้อมูล ก.83
23.5.Green Infrastructure	https://dev.socrata.com/foundry/data.cityofchicago.org/ggws-77ih	ภาคผนวก ตัวอย่างข้อมูล ก.84
24. Data.cityofnewyork.us		
24.1.Harbor Water quality	https://dev.socrata.com/foundry/data.cityofnewyork.us/peb4-ivfn	ภาคผนวก ตัวอย่างข้อมูล ก.85
24.2.Drinking Water quality	https://dev.socrata.com/foundry/data.cityofnewyork.us/xehj-c7r8	ภาคผนวก ตัวอย่างข้อมูล ก.86
24.3.LCR Monitoring	https://dev.socrata.com/foundry/data.cityofnewyork.us/5amj-qj2m	ภาคผนวก ตัวอย่างข้อมูล ก.87
25. Dallasopendata.com		
25.1.Garbage	https://dev.socrata.com/foundry/www.dallasopendata.com/eqhe-99hc	ภาคผนวก ตัวอย่างข้อมูล ก.88
25.2.Police call	https://dev.socrata.com/foundry/www.dallasopendata.com/are8-xahz	ภาคผนวก ตัวอย่างข้อมูล ก.89

ตาราง 3.1 ตัวอย่างข้อมูลที่ได้ทำการเก็บรวบรวมมาเพื่อศึกษา (ต่อ)

ชื่อ	แหล่งอ้างอิง	ตัวอย่าง
25.3.Issued construction	https://dev.socrata.com/foundry/www.dallasopendata.com/are8-xahz	ภาคผนวก ตัวอย่างข้อมูล ก.90
25.4.Water quality	https://dev.socrata.com/foundry/data.austintexas.gov/shdq-7zry	ภาคผนวก ตัวอย่างข้อมูล ก.91
25.5.Recycle drop off	https://dev.socrata.com/foundry/data.austintexas.gov/thy5-qknh	ภาคผนวก ตัวอย่างข้อมูล ก.92
26. Propellerhealth.com		
26.1.Forecast	https://www.propellerhealth.com/air-by-propeller/	ภาคผนวก ตัวอย่างข้อมูล ก.93
27. OpenDataPhilly.org		
27.1.Rain Gauges	https://www.opendataphilly.org/dataset/rain-gauges	ภาคผนวก ตัวอย่างข้อมูล ก.94
28. Smartcitystructure.com		
28.1.Carbon sensor	https://www.smartcitystructure.com/API/v1/environment/things/58b5a4d1f4d5fd84057b23e3/telemetries?access_token=lDzl1qr0n4KUcUlR6sC4fKcj07lMM0KvN4GniUkl&sort=id%2C-createdAt&page=0&perpage=1	ภาคผนวก ตัวอย่างข้อมูล ก.95
28.2.Water sensor	https://www.smartcitystructure.com/API/v1/environment/things/58b5a1f0f4d5fdb8057b23cc/telemetries?access_token=lDzl1qr0n4KUcUlR6sC4fKcj07lMM0KvN4GniUkl	ภาคผนวก ตัวอย่างข้อมูล ก.96
29. Smartcity.kmitl.io		
29.1.Sentimental	https://API.smartcity.kmitl.io/API/v1/sentimental/predicted	ภาคผนวก ตัวอย่างข้อมูล ก.97

3.1.2 วิเคราะห์ข้อมูล

จากการศึกษาตัวอย่างข้อมูลในตาราง 3.1 ทางคณะผู้จัดทำสามารถสรุปคุณลักษณะของข้อมูลได้ดังนี้

1) ชุดข้อมูลตัวอย่างสามารถแบ่งประเภทได้ดังนี้
   1.1) ข้อมูลที่ดิบ (Raw data) 79 ชุดข้อมูล
   1.2) ข้อมูลที่เก็บได้จากการวิเคราะห์ผลค่าข้อมูลดิบ (Processed data) 18 ชุดข้อมูล
2) ชุดข้อมูลตัวอย่างที่เก็บมาสามารถแบ่งหมวดหมู่ของตัวอย่างข้อมูลได้ดังนี้
   2.1) ข้อมูลด้านสภาพอากาศและสิ่งแวดล้อม 35 ชุดข้อมูล
   2.2) ข้อมูลด้านอสังหาริมทรัพย์ 19 ชุดข้อมูล
   2.3) ข้อมูลด้านการคมนาคม 18 ชุดข้อมูล
   2.4) ข้อมูลด้านรายละเอียดสถานที่ 16 ชุดข้อมูล
   2.5) ข้อมูลด้าน IOT 4 ชุดข้อมูล
   2.6) ข้อมูลด้านวิเคราะห์สื่อสงคมออนไลน์ 3 ชุดข้อมูล
   2.7) ข้อมูลด้านสุขภาพ 1 ชุดข้อมูล
   2.8) ข้อมูลด้านพลังงาน 1 ชุดข้อมูล
3) ชนิดของข้อมูลที่ได้จากการสังเกตชุดข้อมูลทั้งหมดมีดังนี้
   3.1) ข้อมูลชนิดตัวอักษร (String) 97 ชุดข้อมูล
   3.2) ข้อมูลชนิดจำนวนทศนิยม (Float) 62 ชุดข้อมูล
   3.3) ข้อมูลชนิดจำนวนเต็ม (Integer) 55 ชุดข้อมูล
   3.4) ข้อมูลชนิดค่าความจริง (Boolean) 9 ชุดข้อมูล
4) มีชุดข้อมูลที่มีค่าพิกัดทางภูมิศาสตร์ส่งมาด้วยเพื่อระบุสถานที่มีทั้งหมด 63 ชุดข้อมูล
5) ชุดข้อมูลที่มีค่าข้อมูลเวลา 54 ชุดข้อมูล โดยเป็นการประทับเวลาเพื่อบอกช่วงเวลาในการเก็บข้อมูล
6) ชุดข้อมูลตัวอย่างส่วนใหญ่จะมีค่าหลักที่ใช้เพื่อการแสดงผล โดยข้อมูลที่สามารถใช้ในการแสดงผลเพื่อให้ผู้ใช้งานเข้าถึงได้ง่ายนั้นไม่ใช่ค่าทั้งหมดที่ส่งมาในชุดรูปแบบข้อมูล

3.2 ออกแบบระบบ

3.2.1 ออกแบบชนิดข้อมูลของ CityViz

จากการศึกษาข้อมูลในหัวข้อ 3.1 ทำให้ทางคณะผู้จัดทำได้ออกแบบชนิดข้อมูลของ CityViz ได้ดังนี้

ตาราง 3.2 ความสัมพันธ์ระหว่างชนิดข้อมูลของ CityViz กับชนิดข้อมูลทั่วไป

ชนิดข้อมูลของ CityViz	ชนิดข้อมูล
ข้อมูลตำแหน่งทางภูมิศาสตร์	ข้อมูลชนิดจำนวนเต็ม หรือ ข้อมูลชนิดจำนวนทศนิยม
ข้อมูลเวลา	ข้อมูลชนิดตัวอักษร
ข้อมูลตัวเลข	ข้อมูลชนิดจำนวนเต็ม หรือ ข้อมูลชนิดจำนวนทศนิยม
ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ	ข้อมูลชนิดตัวอักษร หรือ ข้อมูลชนิดค่าความจริง
ข้อมูลตัวอักษรทั่วไป	ข้อมูลชนิดตัวอักษร

1) ข้อมูลตำแหน่งทางภูมิศาสตร์

คือ ข้อมูลที่แสดงตำแหน่งพิกัดทางภูมิศาสตร์ในรูปแบบละติจูดและลองจิจูด เพื่อบอกให้ทราบถึงตำแหน่งบนผิวโลก เช่น “latitude : 13.7458, longitude: 100.5343”, “coordinates: 13.7458,100.5254” เป็นต้น

2) ข้อมูลเวลา

คือ ข้อมูลที่บ่งบอกถึงค่า วัน เดือน ปี หรือเวลาในรูปแบบสากล เช่น "05/01/2009 14:57:32.8", "1 May 2008 2:57:32.8 PM" เป็นต้น

3) ข้อมูลตัวเลข

คือ ข้อมูลที่เป็นปริมาณทางคณิตศาสตร์ซึ่งแสดงความมากน้อยของค่าใดๆ เช่น 120, 34.445 เป็นต้น

4) ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ

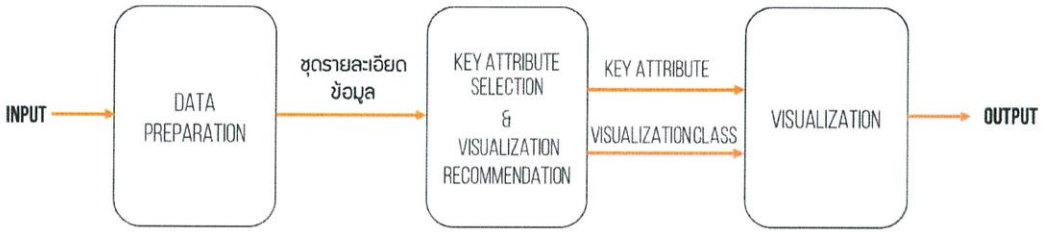
คือ ข้อมูลที่บ่งบอกถึงปริมาณทางคุณภาพได้ เช่น good, bad, low, medium, high เป็นต้น

5) ข้อมูลตัวอักษรทั่วไป

คือ ข้อมูลตัวอักษรที่อาจะมีหรือไม่มีความหมาย นอกเหนือจากข้อมูลตัวอักษรที่แสดงถึงการอันดับ เช่น “I’ll go to Bekki tomorrow”, “วันนี้จะไปเที่ยวสยาม” เป็นต้น

3.2.2 ออกแบบระบบ

ในการแสดงผลข้อมูลที่ไม่สามารถจำกัดลักษณะโครงสร้างข้อมูล ชนิดข้อมูล และที่มาของข้อมูลได้อย่างเฉพาะเจาะจงนั้น จะทำให้กระบวนการแสดงผลข้อมูลมีขั้นตอนภายในหลากหลายขั้นตอน ไม่ว่าจะเป็นการวิเคราะห์หาโครงสร้างข้อมูล การวิเคราะห์หาชนิดของข้อมูล การจัดการข้อมูลให้อยู่ในรูปแบบที่สามารถดึงมาใช้งานได้ และการเลือกรูปแบบการแสดงผลให้ข้อมูลแต่ละชนิด ซึ่งขั้นตอนเหล่านี้เป็นกระบวนการที่ผู้ที่ต้องการแสดงผลข้อมูลต้องทำซ้ำ ๆ ในทุก ๆ ข้อมูลที่เข้ามาใหม่ ระบบที่ทางคณะผู้จัดทำต้องการออกแบบนั้นคือการลดกระบวนการทำงานซ้ำซ้อนในข้างต้นโดยการทำกระบวนการดังที่ได้กล่าวไว้แทนผู้ที่ต้องการแสดงผล

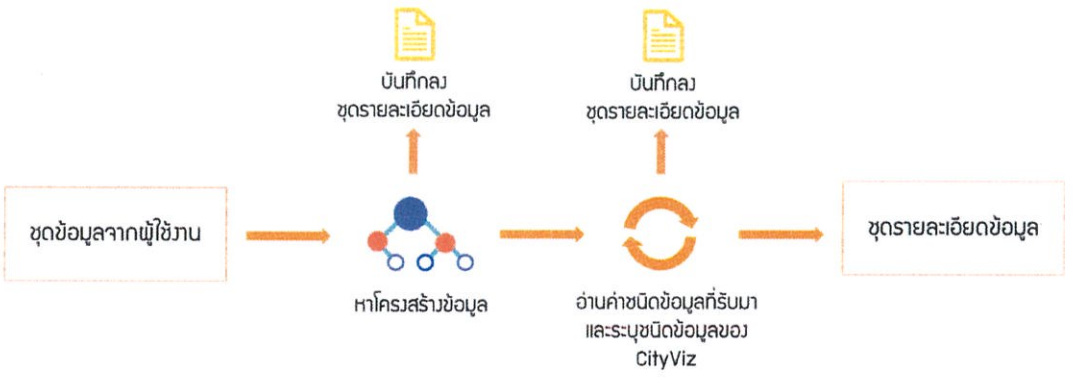


รูป 3.1 การทำงานของระบบ

ซึ่งระบบแสดงผลที่คณะผู้จัดทำได้ออกแบบไว้นั้น สามารถแบ่งการทำงานได้เป็น 3 ส่วน ดังนี้

3.2.2.1 ส่วนเตรียมการข้อมูลสำหรับการแสดงผล (Data preparation)

ชุดข้อมูลที่ผู้ใช้งานป้อนเข้าสู่ระบบนั้น เป็นชุดข้อมูลที่ทางคณะผู้จัดทำไม่สามารถคาดเดาลักษณะของชุดข้อมูล โครงสร้างของชุดข้อมูล หรือวิธีการเข้าถึงข้อมูลแต่ละส่วนได้ ทำให้ทางคณะผู้จัดทำได้ออกแบบส่วนเตรียมการข้อมูลสำหรับการแสดงผลขึ้น เพื่อจัดรูปแบบข้อมูลให้อยู่ในรูปแบบที่ระบบ CityViz สามารถดึงไปใช้งานได้



รูป 3.2 การทำงานของส่วนเตรียมการข้อมูลสำหรับการแสดงผล

โดยการทำงานของระบบส่วนนี้มีขั้นตอนดังนี้

1) รับชุดข้อมูลของผู้ใช้งานเข้าสู่ระบบ

2) หาโครงสร้างข้อมูลโดยหลักการวิเคราะห์คำ (parse)

3) บันทึกเส้นทางในการเข้าถึงข้อมูล ชื่อตัวแปร ชนิดข้อมูล และตัวอย่างข้อมูลลงในชุดรายละเอียดข้อมูล

4) หาชนิดข้อมูลของ CityViz และระบุลงในชุดรายละเอียดข้อมูล

5) ได้ชุดรายละเอียดข้อมูลที่บันทึกไว้สำหรับทำขั้นตอนถัดไป

ชุดรายละเอียดข้อมูลที่ส่วนเตรียมการข้อมูลสำหรับการแสดงผลได้สร้างขึ้นนั้น จะทำให้ขั้นตอนในการหาค่าหลักในการแสดงผลและการหารูปแบบการแสดงผลทำได้ง่ายขึ้น รวมถึงช่วยให้การเรียกใช้งานข้อมูลในส่วนแสดงผลลดกระบวนการทำงานและระยะเวลาลงไปได้เป็นอย่างมาก

3.2.2.2 ส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล (Key Attribute Selection and Visualization Recommendation)

ชุดข้อมูลที่ผู้ใช้งานป้อนเข้าสู่ระบบนั้น มีข้อมูลหลายค่า การเลือกข้อมูลที่เป็นค่าหลักในการแสดงผลจะช่วยผู้ใช้งานในการเลือกค่าหลักในการแสดงผลในกรณีที่ผู้ใช้งานไม่ทราบว่าค่าหลักในการแสดงผลที่ตนเองต้องการนั้นคือค่าใดในชุดข้อมูลที่ส่งมา ซึ่งหากระบบเลือกค่าหลักในการแสดงผลให้ผู้ใช้งานผิดพลาดผู้ใช้งานก็สามารถแก้ไขค่าหลักในการแสดงผลที่ระบบเลือกให้ได้ ซึ่งค่าหลัดในการแสดงผลที่ผู้ใช้งานได้เลือกนั้น จะเป็นตัวบ่งชี้รูปแบบการแสดงผลของระบบว่าสามารถแสดงผลรูปแบบใดเช่นกัน

รูป 3.3 การทำงานของส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล

3.2.2.2.1 ส่วนเลือกค่าหลักในการแสดงผล (Key Attribute Selection)

การแสดงผลข้อมูลบนระบบแสดงผลนั้นจำเป็นที่จะต้องมีข้อมูลที่เป็นค่าหลักในการแสดงผลเพื่อที่จะแสดงข้อมูลดังกล่าวให้กับผู้ใช้งานได้เห็นถึงค่าของข้อมูล ทางคณะ

ผู้จัดทำจึงได้มีระบบในการเลือกข้อมูลที่เป็นค่าหลักในการแสดงผลซึ่งจะมีรูปแบบการเลือกข้อมูล 2 รูปแบบดังนี้

3.2.2.2.1.1 การเลือกค่าหลักในการแสดงผลตามหลักสถิติ (Statistical Approach)

การเลือกข้อมูลที่เป็นค่าหลักในการแสดงผลนั้น พิจารณาผลจากการเก็บสถิติข้อมูลที่ได้จากการสำรวจในเบื้อต้น โดยจะทำการสังเกตจากรูปแบบความสัมพันธ์ของชื่อตัวแปรในการแสดงผลข้อมูล ชนิดข้อมูล และชนิดข้อมูลของระบบแสดงผล เพื่อหาว่ารูปแบบข้อมูลแบบใดที่มีลักษณะเป็นข้อมูลที่เป็นค่าหลักในการแสดงผล โดยผลลัพธ์ที่ได้จากการหาความสัมพันธ์ของข้อมูลนั้นทำให้สรุปลักษณะข้อมูลที่เป็นค่าหลักในการแสดงผลตามค่าสถิติได้ดังตาราง 3.3

ตาราง 3.3 ชื่อตัวแปรและชนิดข้อมูลที่มีลักษณะเป็นค่าหลักในการแสดงผลตามสถิติ

ลำดับความสำคัญ	ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ	ข้อมูลตัวเลข	ข้อมูลตัวอักษรทั่วไป
1	status	index	description หรือ descriptions
2	condition หรือ conditions	main	details
3	level	value	text
4	results หรือ result	total	alerts
5	label	average หรือ avg	messages หรือ message
6	-	normal	-
7	-	speed	-

3.2.2.2.1.2 การเลือกค่าหลักในการแสดงผลโดยใช้ Machine Learning

การเลือกข้อมูลที่เป็นค่าหลักในการแสดงผลโดยใช้ Machine Learning ในการช่วยเลือกนั้น จะทำโดยการนำชุดข้อมูลที่คณะผู้จัดทำได้เก็บค่าไว้มาทำการระบุค่าต่าง ๆที่คาดว่ามีผลต่อการเลือก เช่น ชนิดของข้อมูล ชนิดข้อมูลของระบบแสดงผล จำนวนการซ้ำของ

ข้อมูล และความสามารถในการเป็นค่าหลักในการแสดงผล จากนั้นจึงนำชุดข้อมูลที่ได้ทำการระบุไว้ไปหาความสัมพันธ์โดยใช้ Machine Learning โดยเลือกทดสอบตามหลักอัลกอริทึมต่าง ๆ ที่เป็นอัลกอริทึมในด้านของการแบ่งประเภทข้อมูล แล้วหาค่าความแม่นยำของแต่ละอัลกอริทึมที่ได้เลือกมาเพื่อทดสอบระบบว่าอัลกอริทึมชนิดไหนมีค่าความแม่นยำมากที่สุด

โดยอัลกอริทึมที่ผู้ใช้งานเลือกใช้มาเพื่อทดสอบนั้นเลือกจากความเหมาะสมของอัลกอริทึมต่อชนิดและปริมาณข้อมูล ทางคณะผู้จัดทำเลือกทดสอบด้วยอัลกอริทึมทั้งสิ้น 3 อัลกอริทึมซึ่งเป็นอัลกอริทึมสำหรับทำการแบ่งประเภทข้อมูล (Classification) ทั้งหมด ดังนี้

1) Decision Tree

เป็นการนำ Attributes มาสร้างเป็น node จากนั้นพิจารณาค่าของ node นั้นแล้วแตกกิ่งต่อไปเรื่อยๆ จนสามารถไปถึง Class Attribute ได้ โดย Decision Tree เหมาะกับชุดข้อมูลที่ Attribute และ Class มีจำนวนจำกัด และค่าของAttribute ไม่เป็นค่าต่อเนื่อง

2) K-Nearest Neighbors

เป็นการแบ่งประเภทข้อมูลโดยดูจากเพื่อนบ้านข้างเคียงจำนวน k ตัว รอบข้อมูลที่ต้องทำการทำนาย โดยในเพื่อนบ้าน k ตัวหากมีข้อมูลประเภทใดมากกว่าจะจัดข้อมูลที่ทำนายอยู่เป็นประเภทนั้น

3) Naïve-Bayes

เป็นการแบ่งประเภทข้อมูลโดยใช้ค่าความน่าจะเป็นเข้ามาช่วย

โดยทางคณะผู้จัดทำได้ทำการใช้ชุดข้อมูลเบื้องต้นจากข้อมูลที่เก็บมาเพื่อเป็นโมเดลในการหาคำตอบของชุดข้อมูลใหม่สำหรับชุดข้อมูลที่ผู้ใช้งานป้อนเข้ามาเพื่อหาค่าหลักในการแสดงผล หลังจากนั้นหากมีการยืนยันจากผู้ใช้งานว่าค่าหลักแสดงผลที่แท้จริงเป็นค่าใด ระบบจะทำการเพิ่มข้อมูลไปยังชุดทดสอบเพื่อใช้ในการทำโมเดลในการหาคำตอบให้ผู้ใช้งานในครั้งถัดไป เพื่อเพิ่มความแม่นยำในการคาดเดาค่าหลักในการแสดงผลของ Machine Learning ที่ได้สร้างขึ้นนี้

หลังจากที่ระบบได้ค่าหลักในการแสดงผลแล้วระบบจะทำการหาค่าพิกัดทางภูมิศาสตร์ที่เป็นไปได้กับค่าหลักในการแสดงผลที่ผู้ใช้งานได้เลือกไว้ ซึ่งค่าหลักในการแสดงผลที่ได้และค่าพิกัดทางภูมิศาสตร์ที่ระบบได้หาไว้นั้น จะเป็นข้อมูลที่ใช้ในการจำแนกประเภทการแสดงผลในขั้นตอนถัดไป

3.2.2.2.2 ส่วนแนะนำรูปแบบในการแสดงผล (Visualization Recommendation)

การแบ่งประเภทการแสดงผลในระบบแสดงผลนั้น จะทำการแบ่งประเภทจากรูปแบบข้อมูลที่รวมถึงความสามารถในการแสดงผลบนระบบแสดงผล ซึ่งจะวิเคราะห์จากชนิดข้อมูลของระบบแสดงผลเท่านั้น

สามารถแบ่งกลุ่มได้ดังตาราง 3.4

ตาราง 3.4 ความสัมพันธ์ของข้อมูลที่ใช้ในการแสดงผลกับรูปแบบการแสดงผล

ข้อมูลตำแหน่งทางภูมิศาสตร์	ข้อมูลที่เป็นค่าหลักในการแสดงผล	รูปแบบการแสดงผล
มี	ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ	รูปแบบที่ 1
มี	ข้อมูลตัวเลข	รูปแบบที่ 2
มี	ข้อมูลตัวอักษรทั่วไป	รูปแบบที่ 3
มี	ไม่มี	รูปแบบที่ 4
ไม่มี	ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ	รูปแบบที่ 5
ไม่มี	ข้อมูลตัวเลข	รูปแบบที่ 6
ไม่มี	ข้อมูลตัวอักษรทั่วไป	รูปแบบที่ 7
ไม่มี	ไม่มี	รูปแบบที่ 8

การแสดงผลข้อมูลในรูปแบบต่าง ๆ ที่ได้มีการกำหนดไว้นั้น จะแบ่งประเภทการแสดงผลหลักจากตำแหน่งพิกัดทางภูมิศาสตร์ ซึ่งจะบ่งบอกว่าข้อมูลสามารถแสดงผลบนแผนที่ 3 มิติ ที่ได้มีการจัดทำไว้ได้หรือไม่ หากไม่มีตำแหน่งพิกัดทางภูมิศาสตร์ก็จะไม่สามารถแสดงผลบนแผนที่ 3 มิติได้ แต่หากมีตำแหน่งพิกัดภูมิศาสตร์ก็จะนำค่าไปแสดงผลบนแผนที่ 3 มิติได้ โดยรูปแบบการแสดงผลย่อยนั้นจะพิจารณาจากข้อมูลที่เป็นค่าหลักในการแสดงผล โดยรูปแบบการแสดงผลก็จะแตกต่างตามรูปแบบข้อมูลที่เป็นค่าหลักในการแสดงผล

ตาราง 3.5 ลักษณะการแสดงผลตามรูปแบบการแสดงผล

รูปแบบการแสดงผล	การแสดงผลบนแผนที่ 3 มิติ	การแสดงผลบนแถบหน้าจอ 2 มิติ
รูปแบบที่ 1	แสดงสัญลักษณ์ตามข้อมูล	แสดงความถี่ข้อมูล
รูปแบบที่ 2	แสดงข้อมูลตัวเลข	แสดงค่าเฉลี่ย ค่ามากสุด น้อยสุด
รูปแบบที่ 3	แสดงข้อมูลตัวอักษร	แสดงข้อมูล
รูปแบบที่ 4	แสดงพิน	ไม่มี

ตาราง 3.5 ลักษณะการแสดงผลตามรูปแบบการแสดงผล (ต่อ)

รูปแบบการแสดงผล	การแสดงผลบนแผนที่ 3 มิติ	การแสดงผลบนแถบหน้าจอ 2 มิติ
รูปแบบที่ 5	ไม่มี	แสดงความถี่ข้อมูล
รูปแบบที่ 6	ไม่มี	แสดงค่าเฉลี่ย ค่ามากสุด น้อยสุด
รูปแบบที่ 7	ไม่มี	แสดงข้อมูล
รูปแบบที่ 8	ไม่มี	ไม่มี

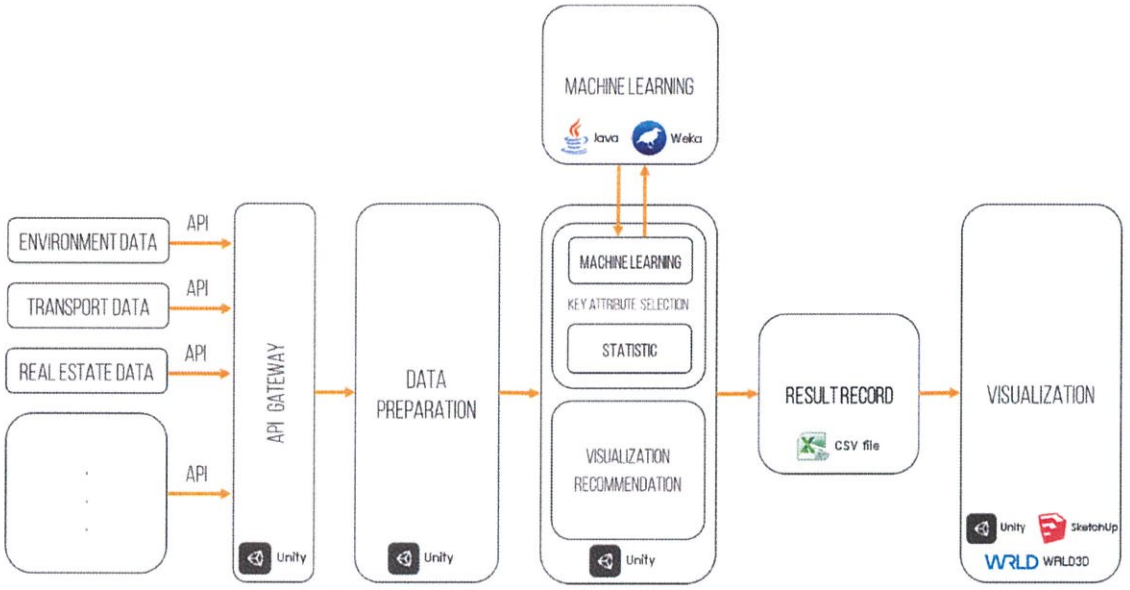
3.2.2.3 ส่วนแสดงผลข้อมูล (Visualization)

ส่วนแสดงผลข้อมูลจะแสดงผลข้อมูลที่ถูกจัดเก็บอยู่ในค่าความจำของระบบ โดยจะทำการเรียกอ่านข้อมูลในค่าความจำทั้งหมด ซึ่งจะมีการเก็บ ที่อยู่ของข้อมูล วิธีการเข้าถึงข้อมูลที่เป็นค่าหลักในการแสดงผลและค่าหลักทางภูมิศาสตร์ และวิธีการแสดงผลข้อมูล หลังจากนั้นจะทำการเรียกใช้งานวิธีการแสดงผลให้ตรงกับค่าที่ได้อ่านมาจากค่าความจำของระบบ ซึ่งส่วนแสดงผลข้อมูลนี้จะทำการอ่านข้อมูลใหม่ในทุก ๆ 15 วินาที เพื่อเป็นการอัพเดทข้อมูลให้ผู้ใช้งานสามารถเห็นข้อมูลในการแสดงผลได้อย่างทันท่วงทีและตรงกับข้อมูลจริง โดยส่วนแสดงผลข้อมูลนี้จะมีวิธีการแสดงผลข้อมูลทั้งสิ้น 2 รูปแบบใหญ่ ๆ คือการแสดงผลข้อมูลบนผังเมือง 3 มิติและการแสดงผลข้อมูลบนแผนผัง 2 มิติ โดยมีรูปแบบการแสดงผลย่อยตามค่าหลักในการแสดงผลอีก 4 ชนิด จึงสามารถแบ่งประเภทการแสดงผลข้อมูลได้ทั้งสิ้น 8 ประเภท ดังที่ได้อธิบายไว้ในส่วนก่อนหน้า

รูป 3.4 การทำงานของส่วนแสดงผลข้อมูล

3.3 วิธีการดำเนินงาน

3.3.1 โครงสร้างในการพัฒนาระบบ



รูป 3.5 โครงสร้างการติดต่อภายในระบบ

3.3.2 การทำงานของระบบ CityViz

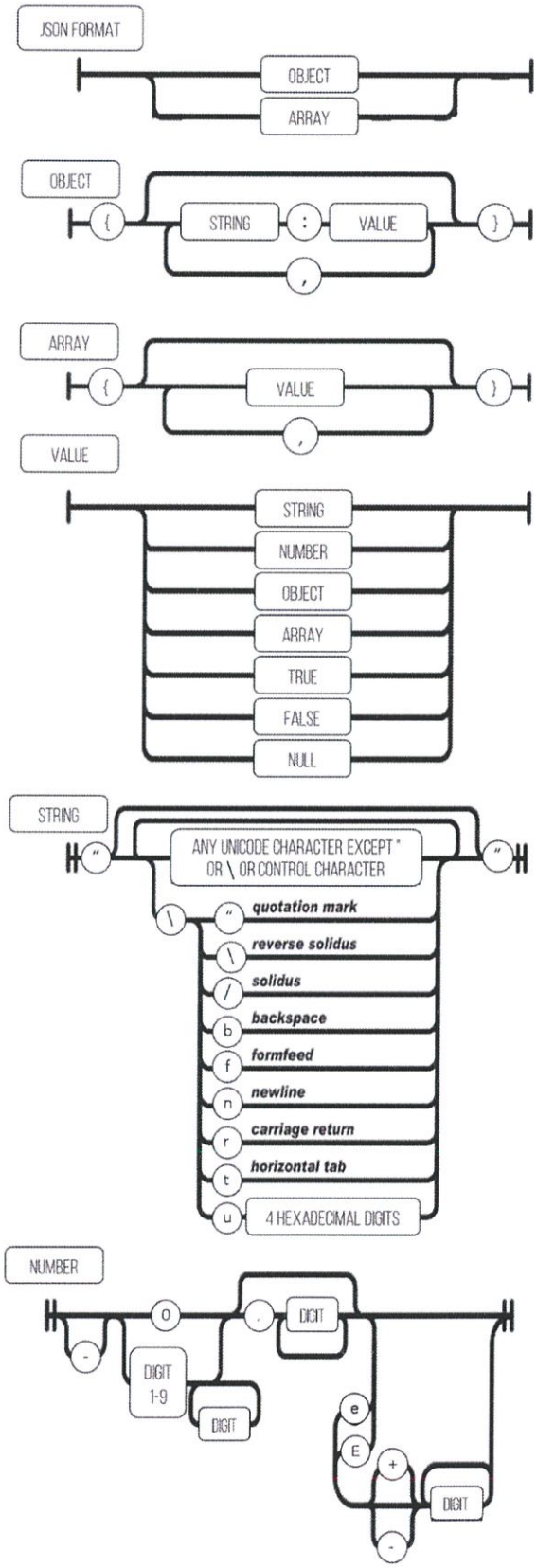
ระบบถูกสร้างขึ้นโดยอ้างอิงการทำงานมาจากการออกแบบระบบในหัวข้อ 3.2 ซึ่งหลักการทำงานจะถูกแบ่งออกเป็น 3 ส่วนดังนี้

3.3.2.1 ส่วนเตรียมการข้อมูลสำหรับการแสดงผล

การทำงานของส่วนเตรียมการข้อมูลสำหรับการแสดงผลในด้านเทคนิคนั้นมีด้วยกัน 2 ส่วน คือ การวิเคราะห์โครงสร้างข้อมูล และการระบุชนิดข้อมูลของ CityViz

3.3.2.1.1 การวิเคราะห์โครงสร้างข้อมูล

เนื่องจากข้อมูลที่ผู้ใช้งานป้อนเข้าสู่ระบบนั้นมีรูปแบบเป็น JSON format ทำให้ทางคณะผู้จัดทำได้ทำการเขียนหลักการวิเคราะห์โครงสร้างข้อมูลสำหรับ JSON format เท่านั้น โดยหลักการวิเคราะห์โครงสร้างข้อมูลมีวิธีการดังรูป 3.6 ซึ่งได้ศึกษาวิธีจากรูปแบบโครงสร้างของข้อมูลในรูปแบบ JSON ส่วนใหญ่ที่มีการสำรวจ



รูป 3.6 กระบวนการวิเคราะห์โครงสร้างข้อมูลของระบบ

3.3.2.1.2 การระบุชนิดข้อมูลของ CityViz

การระบุชนิดข้อมูลของ CityViz นั้นถูกทำขึ้นโดยอ้างอิงจากหัวข้อ 3.2.1 ที่ได้ทำการออกแบบรูปแบบชนิดข้อมูลของ CityViz ไว้ โดยหลักการระบุชนิดข้อมูลมีดังนี้

1) ข้อมูลตำแหน่งทางภูมิศาสตร์ คือ ข้อมูลที่มีลักษณะเป็นข้อมูลชนิดทศนิยมและจำนวนเต็ม ซึ่งมีชื่อตัวแปรในการเก็บข้อมูลที่สื่อถึงข้อมูลทางภูมิศาสตร์ เช่น geolocation, geo-location, coordinates, coord, latitude, lat, longitude, lng, long หรือคำที่ใกล้เคียง
2) ข้อมูลตัวเลข คือ ข้อมูลชนิดจำนวนเต็มหรือข้อมูลชนิดทศนิยมทั้งหมด ยกเว้น ข้อมูลที่ถูกคัดกรองให้เป็นข้อมูลตำแหน่งทางภูมิศาสตร์
3) ข้อมูลเวลา คือ ข้อมูลชนิดตัวอักษรที่สามารถแปลงเป็นข้อมูลวันเวลาได้จากการเรียกใช้งานฟังก์ชันการแปลงข้อมูลตัวอักษรเป็นข้อมูลวันเวลา
4) ข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ คือ ข้อมูลค่าความจริงทั้งหมดและข้อมูลชนิดตัวอักษรที่มีค่าข้อมูลตรงกับข้อมูลที่แสดงถึงการจัดอันดับที่พบได้จากรูปแบบข้อมูลที่ได้จากการศึกษาในขั้นต้น เช่น good, bad, low, medium, high, moderate, normal เป็นต้น
5) ข้อมูลตัวอักษรทั่วไป คือ ข้อมูลชนิดตัวอักษรทั้งหมด ยกเว้น ข้อมูลที่ถูกคัดกรองให้เป็นข้อมูลเวลา และข้อมูลที่ถูกคัดกรองให้เป็นข้อมูลตัวอักษรที่แสดงถึงการจัดอันดับ

3.3.2.2 ส่วนเลือกค่าหลักในการแสดงผลและแนะนำรูปแบบในการแสดงผล

3.3.2.2.1 ส่วนเลือกค่าหลักในการแสดงผล

3.3.2.2.1.1 การหาค่าหลักในการแสดงผลโดยใช้หลักสถิติ (Statistical Approach)

การหาค่าหลักในการแสดงผลนั้นอ้างอิงจากการเก็บค่าทางสถิติที่ทางคณะผู้จัดทำได้ศึกษาชุดตัวอย่างข้อมูลและสรุปผลไปในหัวข้อ 3.2.2.1.1 โดยมีวิธีดำเนินงานดังนี้

1) ลำดับความสำคัญในการตรวจสอบข้อมูลเรียงตาม ข้อมูลชนิดตัวอักษรแบบจัดอันดับ ข้อมูลตัวเลข และข้อมูลตัวอักษรทั่วไป
2) ตรวจสอบค่าในข้อมูลทั้งหมดโดยเริ่มตรวจสอบจากข้อมูลที่มีโอกาสเป็นค่าหลักในการแสดงผลที่มีความสำคัญเป็นอันดับที่ 1 ของทุกประเภท หากเจอมากกว่า 1 ประเภทจะอ้างอิงตามลำดับความสำคัญของชนิดข้อมูล
3) ตรวจสอบทุกๆ ค่าในชุดข้อมูล หากไม่เจอค่าที่ต้องการ ก็จะตรวจสอบในค่าความสำคัญลำดับถัดไป

4) เมื่อเช็คทุกๆ ข้อมูลที่มีในชุดข้อมูลแล้ว หากไม่เจอค่าที่ต้องการ ระบบก็จะเลือกข้อมูลใดๆ ที่ไม่ใช่ข้อมูลพิกัดทางภูมิศาสตร์ และข้อมูลเวลามาใช้เป็นค่าหลักในการแสดงผล

3.3.2.2.1.2 การหาค่าหลักในการแสดงผลโดยใช้ Machine Learning

การหาค่าหลักในการแสดงผลโดยใช้ Machine Learning นั้น ในขั้นตอนการดำเนินงานทางคณะผู้จัดทำจะทำการเตรียมการชุดข้อมูลก่อนส่งไปยัง Machine Learning โดยขั้นตอนการเตรียมการชุดข้อมูลมีดังนี้

1) สร้างโมเดลในการทดสอบชุดข้อมูลโดยเลือกรับค่าข้อมูลตามที่ระบบได้ออกแบบไว้และแสดงผลลัพธ์ในรูปแบบที่ระบบสามารถนำมาใช้งานได้
2) สร้างชุดข้อมูลโดยจัดเก็บชื่อข้อมูล ชนิดข้อมูล ชนิดข้อมูลของระบบ CityViz และจำนวนการซ้ำของเส้นทางในการเข้าถึงข้อมูล เพื่อใช้เป็นชุดข้อมูลสำหรับทดสอบใน Machine Learning
3) ส่งชุดข้อมูลไปยัง Machine Learning ที่ได้มีการจัดเตรียมไว้
4) เลือกการทดสอบข้อมูลโดยใช้โมเดลการทดสอบจาก Machine Learning ที่มีค่าความแม่นยำสูงสุด
5) อ่านผลการทดสอบข้อมูลแล้วเลือกค่าที่ถูกแนะนำให้เป็นค่าหลักในการแสดงผลเข้าสู่ระบบ CityViz
6) หลังจากที่ผู้ใช้งานยืนยันความถูกต้องของการแนะนำแล้ว ระบบจะทำการเพิ่มค่าที่ถูกต้องในการคาดเดาลงไปในชุดข้อมูลที่ใช้สำหรับสร้างโมเดลทดสอบ

3.3.2.2.2 การหาค่าพิกัดทางภูมิศาสตร์ให้กับค่าหลักในการแสดงผล

หลังจากที่ผู้ใช้งานได้ตรวจสอบค่าหลักในการแสดงผลแล้วระบบจะทำการหาค่าพิกัดทางภูมิศาสตร์ที่สามารถใช้งานกับค่าหลักในการแสดงผลที่ผู้ใช้งานได้เลือกขึ้นมา โดยค่าพิกัดทางภูมิศาสตร์ที่ระบบได้เลือกมานั้นมีหลักการในการเลือกดังนี้

1) ค่าพิกัดทางภูมิศาสตร์จะต้องเป็นค่าชนิดค่าพิกัดทางภูมิศาสตร์ของระบบ CityViz
2) หากข้อมูลที่เป็นค่าหลักในการแสดงผลนั้นอยู่ใน array ค่าพิกัดทางภูมิศาสตร์ที่ระบบได้เลือกจะต้องอยู่ใน array ระดับเดียวกัน
3) เส้นทางในการเข้าถึงค่าพิกัดทางภูมิศาสตร์ชนิดละติจูดและลองจิจูดจะต้องมีเส้นทางการเข้าถึงที่ระดับก่อนหน้าเส้นทางเดียวกัน

4) หากระบบไม่สามารถหาค่าพิกัดทางภูมิศาสตร์ที่เหมาะสมกับค่าหลักในการแสดงผลที่ผู้ใช้งานได้เลือกขึ้นมา ระบบจะถือว่าค่านั้นไม่มีค่าพิกัดทางภูมิศาสตร์ที่สามารถใช้งานได้

3.3.2.2.3 ส่วนแนะนำรูปแบบในการแสดงผล

การแนะนำรูปแบบในการแสดงผลอ้างอิงแบบในการแสดงผลจากหัวข้อ 3.2.2.2.2 และ ตาราง 3.4 ซึ่งสามารถแบ่งรูปแบบในการแสดงผลตามค่าพิกัดทางภูมิศาสตร์และค่าหลักในการแสดงผลโดยใช้หลักการสังเกตประเภทของค่าหลักในการแสดงผลและสำรวจการมีอยู่ของค่าพิกัดทางภูมิศาสตร์

3.3.2.2.4 ส่วนบันทึกข้อมูลผลลัพธ์

หลังจากที่ระบบได้รูปแบบในการแสดงผลข้อมูลแล้วระบบจะนำ URL ข้อมูล เส้นทางในการเข้าถึงค่าหลักในการแสดงผลข้อมูล เส้นทางในการเข้าถึงค่าพิกัดทางภูมิศาสตร์ และรูปแบบในการแสดงผล บันทึกลงในไฟล์ผลลัพธ์เพื่อใช้ในการเรียกแสดงผลข้อมูลในขั้นตอนถัดไป

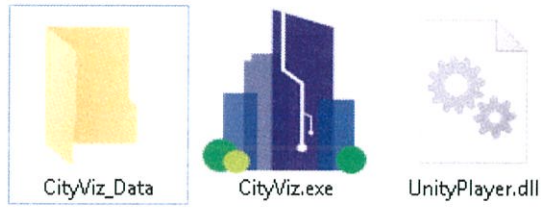
3.3.2.3 ส่วนแสดงผลข้อมูล

การแสดงผลข้อมูลนั้นจะทำการอ่านค่าจากไฟล์ผลลัพธ์ที่ได้มีการบันทึกไว้โดยส่วนแสดงผลข้อมูลจะทำการเรียกใช้งานชุดข้อมูลผ่าน URL ที่และเข้าถึงข้อมูลตามเส้นทางการเข้าถึงข้อมูล เพื่อเรียกใช้งานค่าข้อมูลนำมาแสดงผลตามรูปแบบที่ระบบได้จัดเตรียมไว้ โดยระบบจะแสดงผลข้อมูลลงบนผังเมือง 3 มิติและแผนผัง 2 มิติ ตามรายละเอียยดการบันทึกข้อมูลในขั้นตอนที่ 3.3.2.1

โดยผู้ใช้งานสามารถดูรายละเอียดข้อมูลที่แสดงผลได้จากแถบตั้งค่าที่ทางคณะผู้จัดทำได้จัดเตรียมไว้ให้ ซึ่งมีฟังก์ชันเพิ่มข้อมูลใหม่ที่จำทำการเพิ่มข้อมูลเข้าสู่ระบบ ฟังก์ชั่นไม่แสดงผลชั่วคราวสำหรับข้อมูลที่ไม่ต้องการเห็น ณ เวลานั้นๆ ซึ่งสามารถเรียกแสดงผลใหม่ได้ หรือฟังก์ชันลบข้อมูลที่จะทำการลบข้อมูลที่ไม่ต้องการใช้งานแล้วได้

3.4 ตัวอย่างการใช้งานระบบแสดงผล

3.4.1 การเปิดระบบ CityViz



รูป 3.6 โฟลเดอร์ที่เก็บไฟล์โปรแกรม

การเปิดเข้าใช้งานระบบนั้นผู้ใช้งานจะต้องคลิกที่ไฟล์ CityViz.exe เพื่อเปิดเข้าใช้งานระบบ โดยไฟล์ CityViz.exe จะต้องอยู่ในโฟลเดอร์เดียวกับไฟล์ UnityPlayer.dll และโฟลเดอร์ CityViz_Data

3.4.2 หน้าการแสดงผล

3.4.2.1 หน้าหลักของระบบ

มีส่วนแสดงผลหลักทั้งสิ้น 3 ส่วน โดยจะแบ่งเป็น

1) แผนผังเมือง 3 มิติ (รูป 3.7 ก)) จะปรากฏอัตโนมัติเมื่อเปิดโปรแกรม โดยจะแสดงชื่อของเมืองในแถบซ้ายบน และแสดงปุ่มควบคุมการทำงานในแถบซ้ายล่าง โดยปุ่มควบคุมการทำงานนั้น จะมีทั้งสิ้น 4 ปุ่มคือ ปุ่มกลับสู่พิกัดเริ่มต้น ปุ่มตั้งค่า ปุ่ม Side Bar และปุ่มออกจากโปรแกรม ตามลำดับจากซ้ายไปขวา
2) แถบตั้งค่า (รูป 3.7 ข)) จะปรากฏเมื่อมีการกดปุ่ตั้งค่า โดยจะปรากฏแถบการตั้งค่าในด้านขวาของหน้าจอ และสามารถปิดแถบนี้ได้จากการกดปุ่มตั้งค่าซ้ำอีกครั้ง โดยแถบการตั้งค่าจะประกอบไปด้วยรายละเอียดของข้อมูลที่มีการบันทึกเข้าสู่ระบบในปัจจุบัน รวมถึงสถานะของข้อมูล ว่าได้ถูกตั้งค่าให้มองเห็นหรือไม่ โดยผู้ใช้งานสามารถเลือกลบข้อมูล ปิด/เปิดการมองเห็นข้อมูลที่ได้จัดเก็บไว้ หรือเพิ่มข้อมูลใหม่ได้ในแถบนี้
3) แถบแสดงผลข้อมูล 2 มิติ (รูป 3.7 ค)) จะปรากฏเมื่อมีการกดปุ่ม Side Bar ซึ่งจะมีการแสดงรายละเอียดข้อมูลที่มีการอัพเดตใหม่ตามที่ได้ตั้งค่าไว้ โดยจะแสดงให้เห็นถึงชื่อของข้อมูลและรายละเอียดข้อมูลตามรูปแบบชนิดข้อมูล

ก.)

ข)

ค)

รูป 3.7 หน้าหลักของ CityViz

ก) หน้าหลักของระบบเมื่อไม่กดปุ่มใดๆ

ข) หน้าหลักของระบบเมื่อกดปุ่มตั้งค่า

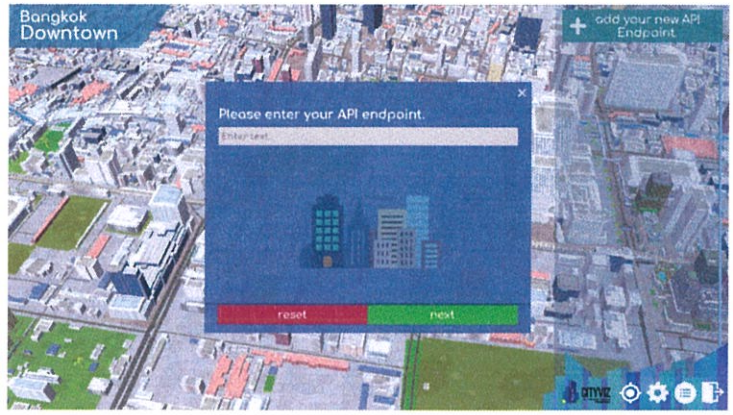
ค) หน้าหลักของระบบเมื่อกดปุ่ม Sidebar

3.4.2.2 การเพิ่มข้อมูลเข้าสู่ระบบ

วิธีการเพิ่มข้อมูลใหม่เข้าสู่ระบบนั้นมีขั้นตอนดังนี้

1) กดปุ่ม add your new API Endpoint ที่ปรากฏในรูป 3.8 ก)
2) กรอก API Endpoint ของข้อมูลที่ต้องการเพิ่มลงในช่องว่างในรูป 3.8 ข) ก่อนกดปุ่ม next โดยสามารถกดปุ่ม reset เพื่อลบข้อมูลที่กรอกผิดพลาดได้
3) ตรวจสอบ Key Attribute ที่ระบบเลือกให้ดังปรากฏในรูป 3.8 ค) หากต้องหารแก้ไขให้กดปุ่ม Select Key และหากพอใจแล้วให้กดปุ่ม next
4) หากต้องการแก้ไข Key Attribute จะปรากฏรูป 3.8 ง) โดยให้ผู้ใช้เลือก Key ที่ต้องการ ซึ่งจะนำให้ตรวจสอบในขั้นตอนที่ 3 ซ้ำอีกครั้ง
5) ระบบจะแสดงรูปแบบการแสดงผลที่ตรงกับข้อมูลดังภาพที่ 3.8 จ) เมื่อผู้ใช้รับทราบให้กดปุ่ม next
6) ผู้ใช้งานต้องกรอกชื่อข้อมูลที่ป้อนเข้ามาในช่องว่างดังรูป 3.8 ฉ) ก่อนกดปุ่ม next
7) เมื่อระบบเพิ่มข้อมูลเรียบร้อยจะแสดงผลลัพธ์ในหน้า 3.8 ช) และข้อมูลใหม่จะปรากฏในแถบตั้งค่าดังภาพ 3.8 ซ)

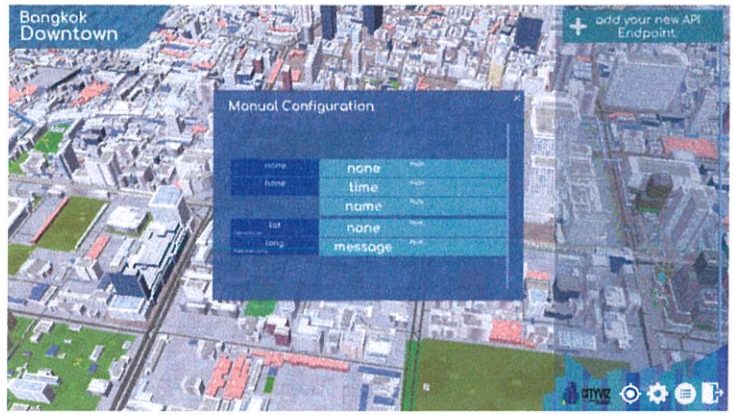
ก)


Bangkok
Downtown
add your new API Endpoint
Please enter your API endpoint.
reset
next

ข)

Bangkok
Downtown
add your new API Endpoint
Key and Location Attribute Prediction
name
message
Key Attribute
name
lat , long
Location Attribute
select key
next

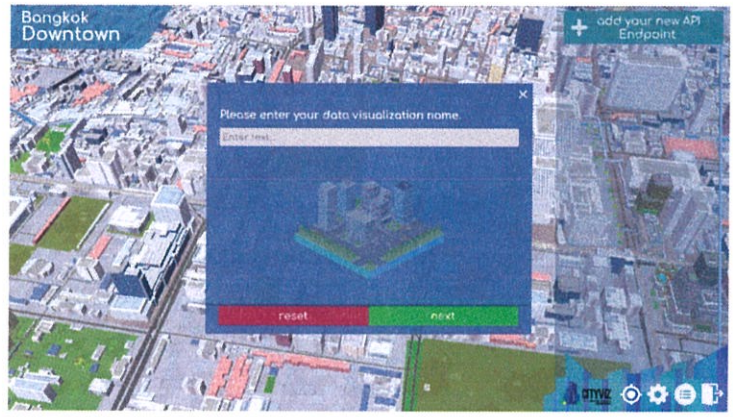
ค)


Bangkok
Downtown
add your new API Endpoint
Manual Configuration
none
time
name
none
message

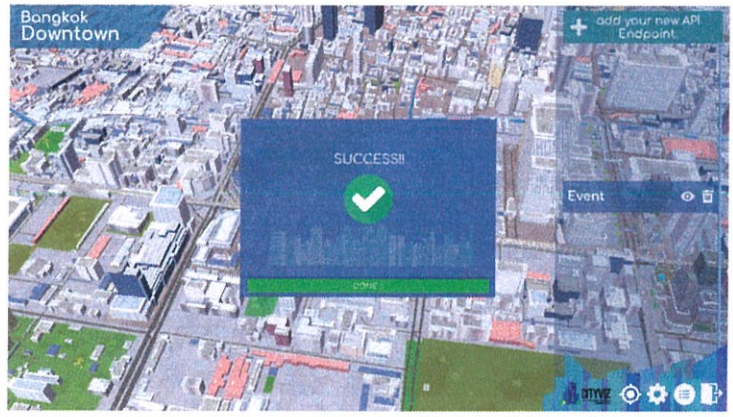
ง)

Bangkok
Downtown
add your new API Endpoint
Data Visualization Recommendation
Map and Sidebar
next

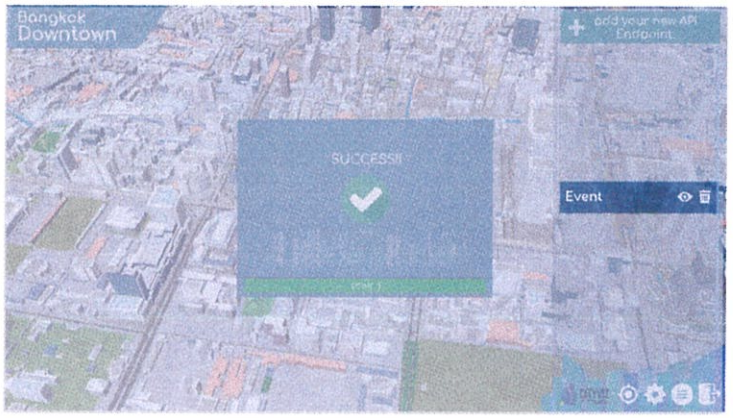
จ)


Bangkok
Downtown
add your new API Endpoint
Please enter your data visualization name.
reset
next

ฉ)


Bangkok
Downtown
add your new API Endpoint
SUCCESS!!
Event

ช)



ซ)

รูป 3.8 วิธีการเพิ่มข้อมูลเข้าสู่ระบบ

ก) หน้าแสดงปุ่มเพิ่มข้อมูล

ข) หน้าแสดงแถบกรอก API ข้อมูล

ค) หน้าแสดงค่าหลักในการแสดงผลที่ระบบแนะนำ

ง) หน้าแสดงการเลือกค่าหลักในการแสดงผลด้วยตนเอง

จ) หน้าแสดงรูปแบบการแสดงผล

ฉ) หน้าแสดงการใส่ชื่อข้อมูล

ช) หน้าแสดงผลการเพิ่มข้อมูลเข้าสู่ระบบ

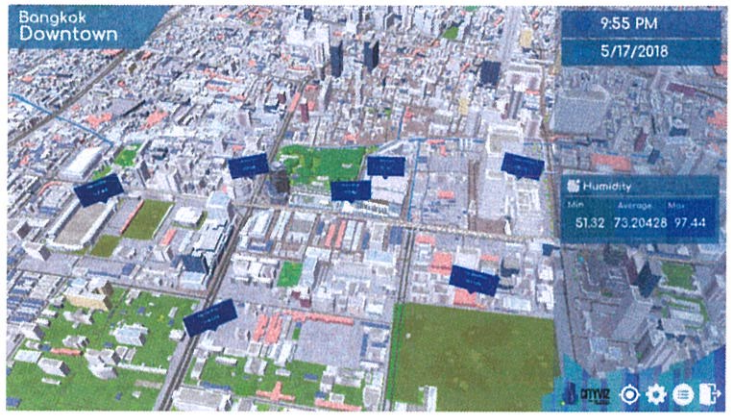
ซ) หน้าแสดงแถบข้อมูลที่เพิ่มเข้ามาใหม่

3.4.2.3 รูปแบบการแสดงผล

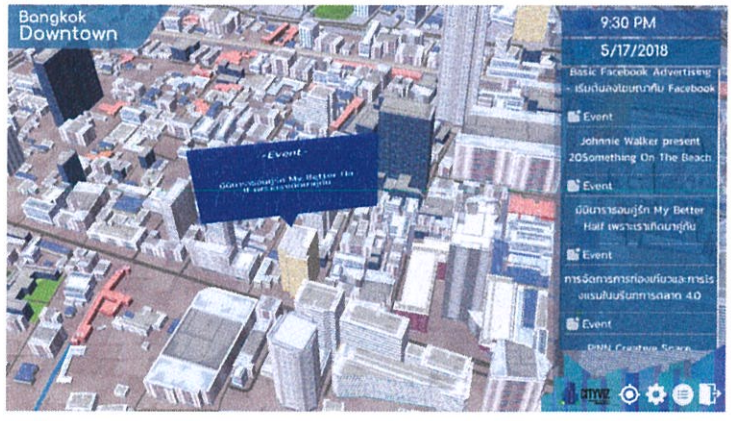
รูปแบบการแสดงผลข้อมูลที่เป็นไปได้ตามตาราง 3.5 ซึ่งจะมีทั้งสิ้น 8 รูปแบบการแสดงผล

Bangkok
Downtown
9:59 PM
5/17/2018
Density
high : 2 low : 3 medium :
2

ก)


Bangkok
Downtown
9:55 PM
5/17/2018
Humidity
Min Average Max
51.32 73.20428 97.44

ข)


Bangkok
Downtown
9:30 PM
5/17/2018
Basic Facebook Advertising
- เริ่มต้นลงโฆษณากับ Facebook
Event
Johnnie Walker present
20Something On The Beach
Event
มีนิทรารอมรู้รัก My Better
Half เพราะเรามีคนเหมือนกัน
Event
การจัดการการท่องเที่ยวและการเร
งแรงในยุคการตลาด 4.0
Event

ค)

Bangkok
Downtown
9:51 PM
5/17/2018

ง)

Bangkok
Downtown
10:13 PM
5/17/2018
Water level
high 3 medium 4

จ)

Bangkok
Downtown
10:15 PM
5/17/2018
Air pollution
Min Average Max
51.32 73.20428 97.44

ฉ)

ช)

ซ)

รูป 3.9 ตัวอย่างการแสดงผลตามรูปแบบการแสดงผล

ก) รูปแบบที่ 1

ข) รูปแบบที่ 2

ค) รูปแบบที่ 3

ง) รูปแบบที่ 4

จ) รูปแบบที่ 5

ฉ) รูปแบบที่ 6

ช) รูปแบบที่ 7

ซ) รูปแบบที่ 8

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองความแม่นยำของส่วน Key Attribute Prediction

ทางผู้จัดทำได้ทำการทดลองเพื่อประเมินค่าความแม่นยำของส่วน Key Attribute Prediction ของ CityViz โดยระบบในส่วนนี้ได้ถูกทำการทดลองเพื่อให้ได้อัลกอริทึมและกลไกการทำงานภายในที่เหมาะสม โดยแบ่งเป็น 2 ประเภท ได้แก่ แบบ Rule-Based System ซึ่งเป็นการเขียนโปรแกรมแบบ Rule-Based ที่เกิดขึ้นจากการเก็บข้อมูลและการหาสถิติและวิเคราะห์ตัดสินใจโดยคณะผู้จัดทำ และแบบ Machine Learning เลือกการเทรนแบบ Supervised Learning เพื่อทำการแบ่งประเภทข้อมูล (Classification) เป็น 2 กลุ่มคือกลุ่มที่เป็น Key Attribute (Key) และกลุ่มที่ไม่เหมาะเป็น Key Attribute (Non-Key) แต่อัลกอริทึมที่ใช้ในการแบ่งประเภทข้อมูล (Classifier) นั้นมีหลายประเภท ทางผู้จัดทำจึงต้องทำการทดลองเพื่อเปรียบเทียบว่าแบบใดมีความแม่นยำและเหมาะสมกับงานของระบบมากที่สุด ในการใช้ Machine Learning จะทำการเปรียบเทียบความแม่นยำทั้งหมด 3 Classifier คือ Decision Tree, K-Nearest Neighbors และ Naïve Bayes จากนั้นจะทำการเปรียบเทียบผลความแม่นยำของอัลกอริทึมทั้งหมดทั้งประเภท Rule-Based System และ Machine Learning

วิธีการประเมินค่าประกอบไปด้วย Confusion Matrix, Precision & Recall, F-Measure และ Accuracy

Confusion Matrix คือตารางประเมินผลลัพธ์การทำนายโดยเทียบกับข้อมูลที่เป็นจริง ประกอบด้วยข้อมูลดังตาราง 4.1 ซึ่งถูกนิยามความหมายตามคลาสที่ประเมินความแม่นยำได้ดังตาราง 4.2

ตาราง 4.1 ต้นแบบ Confusion Matrix

Predicted/Actual	Positive Class	Negative Class
Positive Class	True Positive	False Positive
Negative Class	False Negative	True Negative

ตาราง 4.2 นิยามความหมายของข้อมูลที่ใช้ประกอบการหา Confusion Matrix ตามคลาสที่สนใจจะประเมินผลความแม่นยำ

คลาสที่ประเมินความแม่นยำ / คำเฉพาะ	Key	Non-Key
Positive Class	Key	Non-Key
Negative Class	Non-Key	Key
True Positive	จำนวนข้อมูลที่ระบบทำนายถูกว่าเป็น Key	จำนวนข้อมูลที่ระบบทำนายถูกว่าเป็น Non-Key
True Negative	จำนวนข้อมูลที่ระบบทำนายถูกว่าเป็น Non-Key	จำนวนข้อมูลที่ระบบทำนายถูกว่าเป็น Key
False Positive	จำนวนข้อมูลที่เป็น Key แต่ระบบทำนายผิดว่าเป็น Non-Key	จำนวนข้อมูลที่เป็น Non-Key แต่ระบบทำนายผิดว่าเป็น Key
False Negative	จำนวนข้อมูลที่เป็น Non-Key แต่ระบบทำนายผิดว่าเป็น Key	จำนวนข้อมูลที่เป็น Key แต่ระบบทำนายผิดว่าเป็น Non-Key

$$\text{จำนวนที่ทำนายว่าเป็นคลาสที่เราสนใจจะประเมินผลความแม่นยำ} = \text{True Positive} + \text{False Negative} \qquad (4.1)$$

Precision คือ จำนวนข้อมูลที่ระบบทำนายได้ถูกต้องของคลาสนั้นๆเทียบกับจำนวนทั้งหมดของข้อมูลจริงคลาสนั้นๆ โดยสามารถคำนวณได้จาก สูตร

$$\text{Precision ของ Class ใดๆ} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \qquad (4.2)$$

Recall คือ จำนวนข้อมูลที่ระบบทำนายได้ถูกต้องของคลาสนั้นๆจากจำนวนทั้งหมดที่ถูกเดาว่าเป็นคลาสนั้น โดยสามารถคำนวณได้จาก สูตร

$$\text{Recall ของ Class ใดๆ} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \quad (4.3)$$

F-Measure คือ ค่าเฉลี่ยของ Recall และ Precision โดยสามารถคำนวณได้จาก สูตร

$$\text{F-Measure} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \quad (4.4)$$

Accuracy คือ จำนวนข้อมูลที่ระบบทำนายถูกต้องทั้งหมดในทุกคลาสเทียบกับจำนวนของข้อมูลทั้งหมด โดยสามารถคำนวณได้จาก สูตร

$$\text{Accuracy} = \frac{\text{True Positive+True Negative}}{\text{True Positive+True Negative+False Positive+False Negative}} \quad (4.5)$$

ในการทดสอบครั้งนี้ทางผู้จัดทำได้ทำการทดสอบระบบด้วย ตัวอย่างของ Attribute จากทั้งหมด 97 API รวมทั้งสิ้น 1,479 Attributes ประกอบไปด้วย Attribute ที่มีลักษณะเป็น Key ทั้งหมด 233 ตัว และ Attribute ที่ไม่มีลักษณะเป็น Key ทั้งหมด 1,246 ตัว ในอัลกอริทึมแบบ Machine Learning

4.1.1 ผลการทดลองความแม่นยำของระบบแบบ Rule-Based System

ตาราง 4.3 Confusion Matrix ของ Rule-Based System

Predicted/Actual	Key	Non-Key
Key	46	187
Non-Key	51	1,195

จำนวนที่ทำนายว่าเป็น Key = 46+51 = 97 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 187+1,195 = 1,382 ตัว

$$\text{Precision ของ Key} = \frac{46}{46+187} \times 100 = 19.7425\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,195}{1,195+51} \times 100 = 95.9069\ \%$$

$$\text{Recall ของ Key} = \frac{46}{46+51} \times 100 = 47.4227\ \%$$

$$\text{Recall ของ Non-Key} = \frac{1,195}{1,195+187} \times 100 = 86.4689\ \%$$

$$\text{F-Measure ของ Key} = \frac{2 \times 19.7425 \times 47.4227}{19.7425+47.4227} = 27.8788$$

$$\text{F-Measure ของ Non-Key} = \frac{2 \times 95.9069 \times 86.4689}{95.9069+86.4689} = 90.9437$$

$$\text{Accuracy} = \frac{46+1,195}{46+1,195+51+187} \times 100 = 83.9080\ \%$$

4.1.2 ผลการทดลองความแม่นยำของระบบแบบ Decision Tree

ตาราง 4.4 Confusion Matrix ของ Decision Tree

Predicted/Actual	Key	Non-Key
Key	26	207
Non-Key	2	1,244

จำนวนที่ทำนายว่าเป็น Key =26+2 = 28 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 207+1,244 = 1,451 ตัว

$$\text{Precision ของ Key} = \frac{26}{26+207} \times 100 = 11.1588\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,244}{1,244+2} \times 100 = 99.6795\ \%$$

$$\text{Recall ของ Key} = \frac{26}{26+2} \times 100 = 92.8571\ \%$$

$$\text{Recall ของ Non-Key} = \frac{1,244}{1,244+207} \times 100 = 85.7340\ \%$$

$$\text{F-Measure ของ Key} = \frac{2\times11.1588\times54.1667}{11.1588+54.1667} = 18.5053$$

$$\text{F-Measure ของ Non-Key} = \frac{2\times99.6795\times85.7340}{99.6795+85.7340} = 95.1823$$

$$\text{Accuracy} = \frac{26+1,244}{26+1,244+207+2}\times100 = 85.8688\ \%$$

4.1.3 ผลการทดลองความแม่นยำของระบบแบบ Naïve Bayes

ตาราง 4.5 Confusion Matrix ของ Naïve Bayes

Predicted/Actual	Key	Non-Key
Key	25	208
Non-Key	11	1,235

จำนวนที่ทำนายว่าเป็น Key =25+11 = 36 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 208+1,235 = 1,443 ตัว

$$\text{Precision ของ Key} = \frac{25}{25+208}\times100 = 10.7296\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,235}{1,235+11}\times100 = 99.1171\ \%$$

$$\text{Recall ของ Key} = \frac{25}{25+11}\times100 = 69.4444\ \%$$

$$\text{Recall ของ Non-Key} = \frac{1,235}{1,235+208}\times100 = 85.5856\ \%$$

$$\text{F-Measure ของ Key} = \frac{2\times10.7296\times69.4444}{10.7296+69.4444} = 18.5873\ \%$$

$$\text{F-Measure ของ Non-Key} = \frac{2\times99.1171\times85.5856}{99.1171+85.5856} = 91.8557\ \%$$

$$\text{Accuracy} = \frac{25+1,235}{25+1,235+208+11}\times100 = 85.1927\ \%$$

4.1.4 ผลการทดลองความแม่นยำของระบบแบบ K-Nearest Neighbors

ตาราง 4.6 Confusion Matrix ของ K-Nearest Neighbors

Predicted/Actual	Key	Non-Key
Key	79	154
Non-Key	36	1,210

จำนวนที่ทำนายว่าเป็น Key = 79+36 = 115 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 154+1,210 = 1,364 ตัว

$$\text{Precision ของ Key} = \frac{79}{79+154} \times 100 = 33.9056\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,210}{1,210+36} \times 100 = 97.1108\ \%$$

$$\text{Recall ของ Key} = \frac{79}{79+36} \times 100 = 68.6957\ \%$$

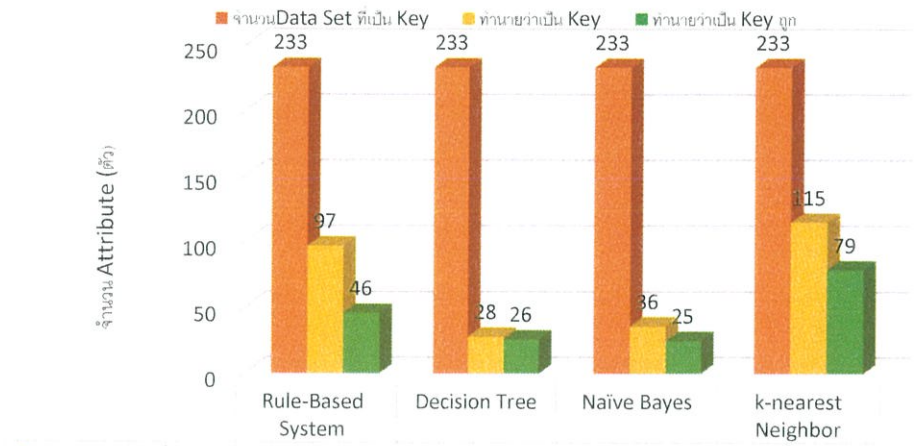
$$\text{Recall ของ Non-Key} = \frac{1,210}{1,210+154} \times 100 = 88.7097\ \%$$

$$\text{F-Measure ของ Key} = \frac{2 \times 33.9056 \times 68.6957}{33.9056+68.6957} = 45.4023$$

$$\text{F-Measure ของ Non-Key} = \frac{2 \times 97.1108 \times 88.7097}{97.1108+88.7097} = 92.7203$$

$$\text{Accuracy} = \frac{79+1,210}{79+1,210+36+154} \times 100 = 87.1535\ \%$$

4.1.5 เปรียบเทียบผลการทำนายข้อมูลประเภท Key



	Rule-Based System	Decision Tree	Naïve Bayes	k-nearest Neighbor
จำนวนData Set ที่เป็น Key	233	233	233	233
ทำนายว่าเป็น Key	97	28	36	115
ทำนายว่าเป็น Key ถูก	46	26	25	79

รูป 4.1 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Key

จากการนำ Data Set เข้าไปทดสอบความแม่นยำของระบบส่วน Key Attribute Prediction โดยใช้อัลกอริทึมที่แตกต่างกันทั้งหมด 4 แบบ ได้แก่ Rule-Based System, Decision Tree, Naïve Bayes และ K-Nearest Neighbors พบว่า แบบ Rule-Based System ทำการทำนายว่าเป็น Key Attribute ทั้งหมด 97 ตัว และทำนายถูกต้อง เป็นจำนวน 46 ตัว

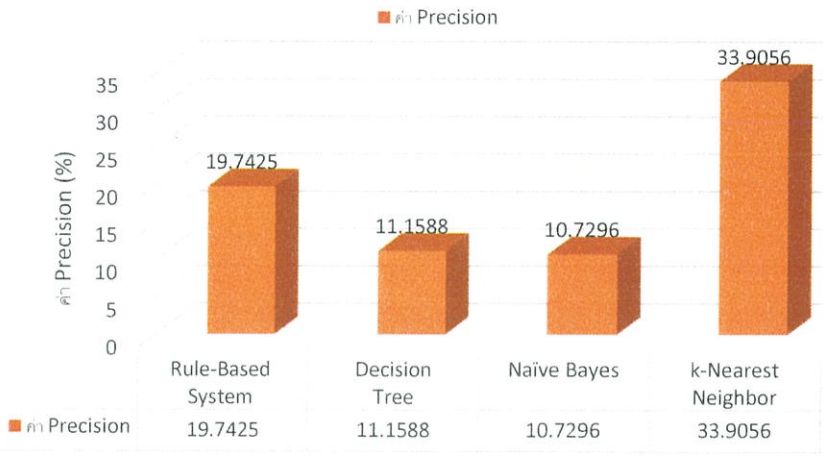
แบบ Decision Tree ทำการทำนายว่าเป็น Key Attribute ทั้งหมด 28 ตัวและทำนายถูกต้องเป็นจำนวน 26 ตัว

แบบ Naïve Bayes ทำการทำนายว่าเป็น Key Attribute ทั้งหมด 36 ตัว และทำนายถูกต้องเป็นจำนวน 25 ตัว

และแบบ K-Nearest Neighbors ทำการทำนายว่าเป็น Key Attribute ทั้งหมด 115 ตัว และทำนายถูกต้องเป็นจำนวน 79 ตัว

โดยทุกอัลกอริทึมมีการทำนายว่าเป็น Key Attribute น้อยกว่าจำนวนข้อมูลจริงที่ถูกจัดประเภทเป็น Key Attribute โดย K-Nearest Neighbors มีการจำนวนการทำนายว่าเป็น Key Attribute มากที่สุด ตามด้วย Rule-Based System, Naïve Bayes และ Decision Tree ตามลำดับ ส่วนอัลกอริทึมที่มีจำนวนการทำนายเข้าใกล้จำนวนข้อมูลจริงประเภท Key Attribute มากที่สุด ได้แก่ K-Nearest Neighbors ทำนายขาดไป 118 ตัวจาก 223, Rule-Based System ทำนายขาดไป 136 ตัวจาก

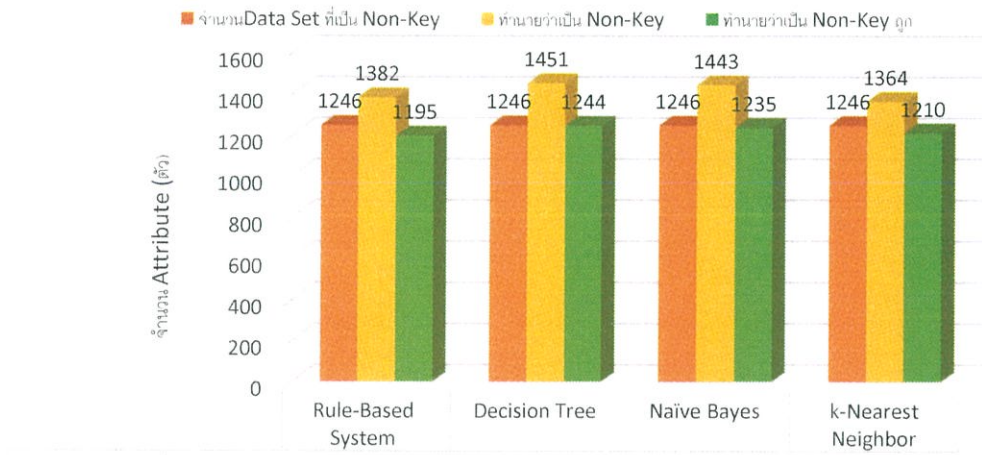
223,Naïve Bayes ทำนายขาดไป 197 ตัวจาก 223 และ Decision Tree ทำนายขาดไป 205 ตัวจาก 223 ตามลำดับ



รูป 4.2 กราฟแสดงผลเปรียบเทียบค่า Precision ของการทำนายข้อมูลประเภท Key

จากจำนวนการทำนาย Key Attribute ที่ถูกต้องเมื่อนำมาเทียบกับจำนวนข้อมูลจริงที่เป็นประเภท Key ทั้งหมด พบว่า K-Nearest Neighbors มีค่า Precision มากที่สุดอยู่ที่ 33.9056 เนื่องจากมีการทำนายว่าเป็น Key เป็นจำนวนที่เยอะกว่าอัลกอริทึมแบบอื่นๆ ตามมาด้วย Rule-Based System ที่มีค่า Precision เท่ากับ 19.7425 เนื่องจากมีจำนวนการทายว่าเป็น Key Attribute เยอะเป็นอันดับที่สอง ในอันดับที่ 3 คืออัลกอริทึมแบบ Decision Tree มีค่า Precision เท่ากับ 11.1588 แต่จำนวนที่ทายว่าเป็น Key ไม่ได้อยู่ในอันดับ 3 แต่มีค่า Precision อยู่อันดับ 3 เนื่องจากมีจำนวนที่ทายถูกมากกว่าแบบ Naïve Bayes และอันดับสุดท้ายคือแบบ Naïve Bayes ค่า Precision เท่ากับ 10.7269 ถึงแม้จะมีการทำนายว่าเป็น Key มากกว่าแบบ Decision Tree แต่จำนวนที่ทายถูกนั้นมีจำนวนน้อยกว่า

4.1.6 เปรียบเทียบผลการทำนายข้อมูลประเภท Non – Key



	Rule-Based System	Decision Tree	Naïve Bayes	k-Nearest Neighbor
จำนวนData Set ที่เป็น Non-Key	1246	1246	1246	1246
ทำนายว่าเป็น Non-Key	1382	1451	1443	1364
ทำนายว่าเป็น Non-Key ถูก	1195	1244	1235	1210

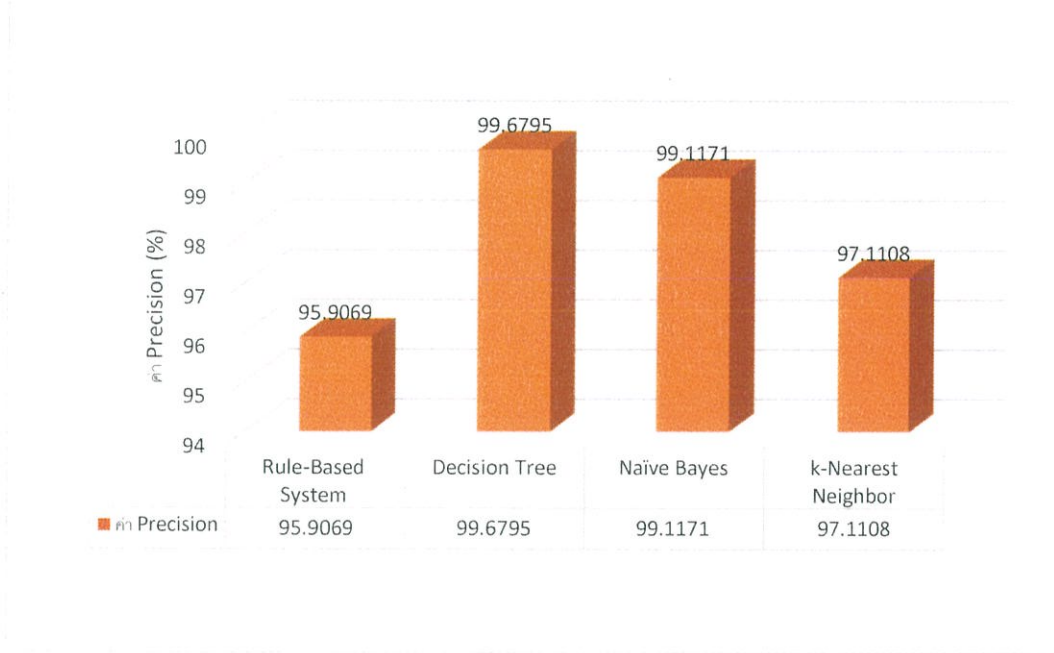
รูป 4.3 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key

จากการนำ Data Set เข้าไปทดสอบความแม่นยำของระบบส่วน Key Attribute Prediction โดยใช้อัลกอริทึมที่แตกต่างกันทั้งหมด 4 แบบ ได้แก่ Rule-Based System, Decision Tree, Naïve Bayes และ K-Nearest Neighbors พบว่า แบบ Rule-Based System ทำการทำนายว่าเป็น Non-Key Attribute ทั้งหมด 1,382 ตัว และทำนายถูกต้อง เป็นจำนวน 1,195 ตัว

แบบ Decision Tree ทำการทำนายว่าเป็น Non-Key Attribute ทั้งหมด 1,451 ตัวและทำนายถูกต้องเป็นจำนวน 1,244 ตัว

แบบ Naïve Bayes ทำการทำนายว่าเป็น Non-Key Attribute ทั้งหมด 1,443 ตัว และทำนายถูกต้องเป็นจำนวน 1,235 ตัว

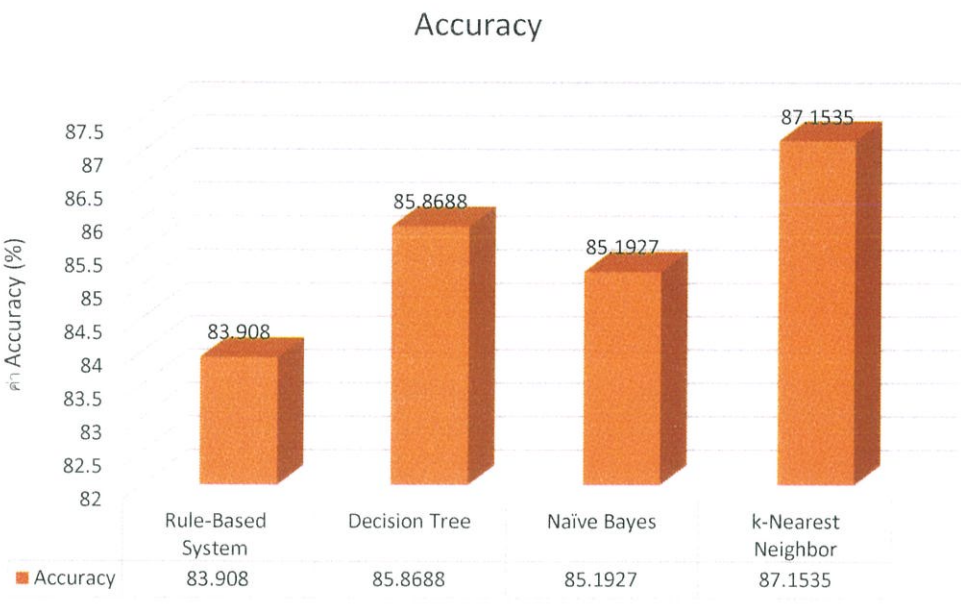
และแบบ K-Nearest Neighbors ทำการทำนายว่าเป็น Non-Key Attribute ทั้งหมด 1,364 ตัว และทำนายถูกต้องเป็นจำนวน 1,210 ตัว โดยทุกอัลกอริทึมมีการทำนายว่าเป็น Non-Key Attribute มากกว่าจำนวนข้อมูลจริงที่ถูกจัดประเภทเป็น Non-Key Attribute โดย Decision Tree มีการทำนายมากที่สุด ตามด้วย Naïve Bayes, Rule-Based System และ K-Nearest Neighbors ส่วนอัลกอริทึมที่มีจำนวนการทายเข้าใกล้จำนวนข้อมูลจริงประเภท Non-Key Attribute มากที่สุด ได้แก่ K-Nearest Neighbors เดาเกินไป 118 ตัว จาก 1,246 ตัว, Rule-Based System เดาเกินไป 136 ตัว จาก 1,246 ตัว, Naïve Bayes เดาเกินไป 197 ตัว จาก 1,246 ตัว และ Decision Tree เดาเกินไป 205 ตัว จาก 1,246 ตัว ตามลำดับ



รูป 4.4 กราฟแสดงผลเปรียบเทียบค่า Precision ของการทำนายข้อมูลประเภท Non-Key

จากจำนวนการทำนาย Non-Key Attribute ที่ถูกต้องเมื่อนำมาเทียบกับจำนวนข้อมูลจริงที่เป็นประเภท Non-Key ทั้งหมด พบว่า Decision Tree มีค่า Precision มากที่สุดอยู่ที่ 99.6795 เนื่องจากมีการทำนายว่าเป็น Key เป็นจำนวนที่เยอะกว่าอัลกอริทึมแบบอื่นๆ ตามมาด้วย Naïve Bayes ที่มีค่า Precision เท่ากับ 99.1171 เนื่องจากมีจำนวนการทายว่าเป็น Non-Key Attribute เยอะเป็นอันดับที่สอง อัลกอริทึมแบบ K-Nearest Neighbors มีค่า Precision เป็นอันดับที่ 3 เท่ากับ 97.1108 แต่จำนวนที่ทายว่าเป็น Non-Key ไม่ได้อยู่ในอันดับ 3 แต่มีค่า Precision อยู่อันดับ 3 เนื่องจากมีจำนวนที่ทายถูกมากกว่าแบบ Rule-Based System และอันดับสุดท้ายคือแบบ Rule-Based System ค่า Precision เท่ากับ 95.9069 ถึงแม้จะมีการทำนายว่าเป็น Non-Key มากกว่าแบบ K-Nearest Neighbors แต่จำนวนที่ทายถูกนั้นมีจำนวนน้อยกว่า

4.1.7 ผลการเปรียบเทียบค่า Accuracy ของอัลกอริทึมทั้งหมด



รูป 4.5 กราฟผลการเปรียบเทียบค่า Accuracy ของอัลกอริทึมทั้งหมด

จากการคำนวณหาค่า Accuracy คือจำนวนข้อมูลที่ทำนายถูกทั้งหมดทั้งประเภท Key และ Non-Key เทียบกับจำนวนข้อมูลจริงทั้งหมด พบว่าเมื่อเทียบแบบ Rule-Based System กับอัลกอริทึมประเภท Machine Learning ทั้งหมด ค่า Accuracy ของแบบ Machine Learning มีค่าสูงกว่า โดย K-Nearest Neighbors มีค่า Accuracy มากที่สุดเท่ากับ 87.1535 % ตามมาด้วย Decision Tree มีค่า Accuracy เท่ากับ 85.8688 % แบบ Naïve Bayes มีค่าAccuracy เท่ากับ 85.1927 % และอันดับสุดท้ายคือแบบ Rule-Based System มีค่า Accuracy อยู่ที่ 83.908 % ดังนั้นเพื่อทำให้ระบบส่วน Key Attribute มีความแม่นยำที่ดีจึงควรเลือกอัลกอริทึมแบบ K-Nearest Neighbors

4.2 ผลการทดลองความแม่นยำเมื่อปรับค่า k ใน K-Nearest Neighbors

หลังจากที่ได้ Classifier ที่เหมาะสม ทางผู้จัดทำได้จัดทำการทดลองปรับการตั้งค่าของ Machine Learning เพื่อให้โมเดลมีประสิทธิภาพที่ดีเหมาะสมแก่ข้อมูลที่ใช้ โดยข้อมูลที่ใช้ประกอบไปด้วย 2 คลาส คือ Key และ Non-Key

ในขณะทำการทดลองมีข้อมูลที่ใช้ทั้งหมด 1,479 ตัว ในการทำงานของ K-Nearest Neighbors จะมีการใช้ตัวรอบข้างจำนวณ k ตัว เพื่อโหวตดูโดยประสิทธิภาพการทำงานของ K-Nearest Neighbors จะเปลี่ยนไป หากทำการปรับค่า k นี้ ทางผู้จัดทำจึงได้ทำการทดลองวัดค่าแม่นยำเมื่อเปลี่ยนค่า k เป็น 1,3 และ $\frac{\sqrt{N}}{2}$ เมื่อ N เท่ากับจำนวน Dataset ที่ใช้เทรน โดยเลือกเป็นเลขคี่เสมอเพื่อลดปัญหาผลโหวตไม่เป็นเอกฉันท์ และใช้หลักการวัดแบบเดียวกับ หัวข้อ 4.1

4.2.1 ผลการทดลองความแม่นยำของ K-Nearest Neighbors โดยมีค่า k เป็น 1

ตาราง 4.7 Confusion Matrix ของ K-Nearest Neighbors โดย k = 1

Predicted/Actual	Key	Non-Key
Key	218	3
Non-Key	15	1,243

จำนวนที่ทำนายว่าเป็น Key = 218+3 = 221 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 15+1,243 = 1,258 ตัว

$$\text{Precision ของ Key} = \frac{218}{218+15} \times 100 = 93.5622\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,243}{1,243+3} \times 100 = 99.7592\ \%$$

$$\text{Recall ของ Key} = \frac{218}{218+3} \times 100 = 98.6425\ \%$$

$$\text{Recall ของ Non-Key} = \frac{1,243}{1,243+15} \times 100 = 98.8076\ \%$$

$$\text{F-Measure ของ Key} = \frac{2 \times 93.5622 \times 98.6425}{93.5622+98.6425} = 96.0352$$

$$\text{F-Measure ของ Non-Key} = \frac{2\times99.7592\times98.8076}{99.7592+98.8076} = 99.2841$$

$$\text{Accuracy} = \frac{218+1,243}{218+1,243+3+15}\times100 = 98.7830\ \%$$

4.2.2 ผลการทดลองความแม่นยำของK-Nearest Neighbors โดยมีค่า k เป็น 3

ตาราง 4.8 Confusion Matrix ของ K-Nearest Neighbors โดย k = 3

Predicted/Actual	Key	Non-Key
Key	75	7
Non-Key	158	1,239

จำนวนที่ทำนายว่าเป็น Key = 75+7 = 82 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 158+1,2369 = 1,397 ตัว

$$\text{Precision ของ Key} = \frac{75}{75+158}\times100 = 32.1888\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,239}{1,239+7}\times100 = 99.4382\ \%$$

$$\text{Recall ของ Key} = \frac{75}{75+7}\times100 = 91.4634\ \%$$

$$\text{Recall ของ Non-Key} = \frac{1,239}{1,239+158}\times100 = 88.6900\ \%$$

$$\text{F-Measure ของ Key} = \frac{2\times32.1888\times91.4634}{32.1888+91.4634} = 47.6190$$

$$\text{F-Measure ของ Non-Key} = \frac{2\times99.4382\times88.6900}{99.4382+88.6900} = 93.7571$$

$$\text{Accuracy} = \frac{75+1,239}{75+1,239+7+158}\times100 = 88.8438\ \%$$

4.2.3 ผลการทดลองความแม่นยำของ K-Nearest Neighbors โดยมีค่า k เป็น $\frac{\sqrt{N}}{2}$

(ในที่นี้คือ ($\sqrt{1479}$)/2 = 19.228884523 ปัดลงเท่ากับ 19)

ตาราง 4.9 Confusion Matrix ของ K-Nearest Neighbors โดย k = $\frac{\sqrt{N}}{2}$

Predicted/Actual	Key	Non-Key
Key	5	0
Non-Key	228	1,246

จำนวนที่ทำนายว่าเป็น Key = 5+0 = 5 ตัว

จำนวนที่ทำนายว่าเป็น Non-Key = 228+1,246 = 1,474 ตัว

$$\text{Precision ของ Key} = \frac{5}{5+228} \times 100 = 2.1459\ \%$$

$$\text{Precision ของ Non-Key} = \frac{1,246}{1,246+0} \times 100 = 100\ \%$$

$$\text{Recall ของ Key} = \frac{5}{5+0} \times 100 = 100\ \%$$

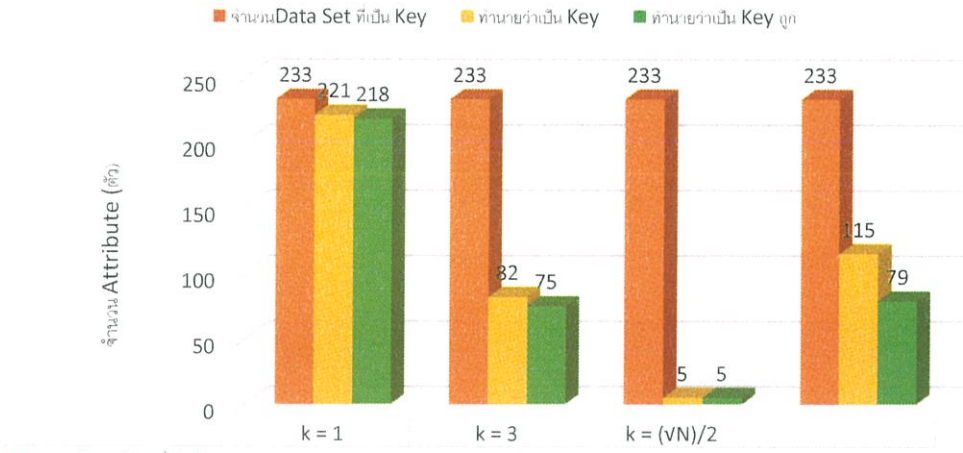
$$\text{Recall ของ Non-Key} = \frac{1,246}{1,246+228} \times 100 = 84.5318\ \%$$

$$\text{F-Measure ของ Key} = \frac{2\times 2.1459\times 100}{2.1459+100} = 4.2016$$

$$\text{F-Measure ของ Non-Key} = \frac{2\times 100\times 84.5318}{100+84.5318} = 91.6176$$

$$\text{Accuracy} = \frac{5+1,246}{5+1,246+0+228} \times 100 = 84.5842\ \%$$

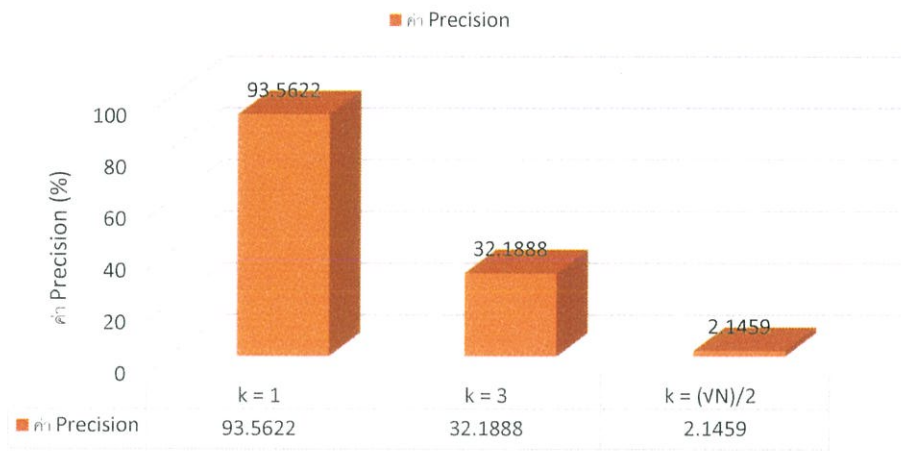
4.2.4 เปรียบเทียบผลการทำนายข้อมูลประเภท Key โดยใช้ค่า k ใน K-Nearest Neighbors ที่ต่างกัน



	k = 1	k = 3	k = (√N)/2	
จำนวนData Set ที่เป็น Key	233	233	233	233
ทำนายว่าเป็น Key	221	82	5	115
ทำนายว่าเป็น Key ถูก	218	75	5	79

รูป 4.6 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Key ของ K-Nearest Neighbors

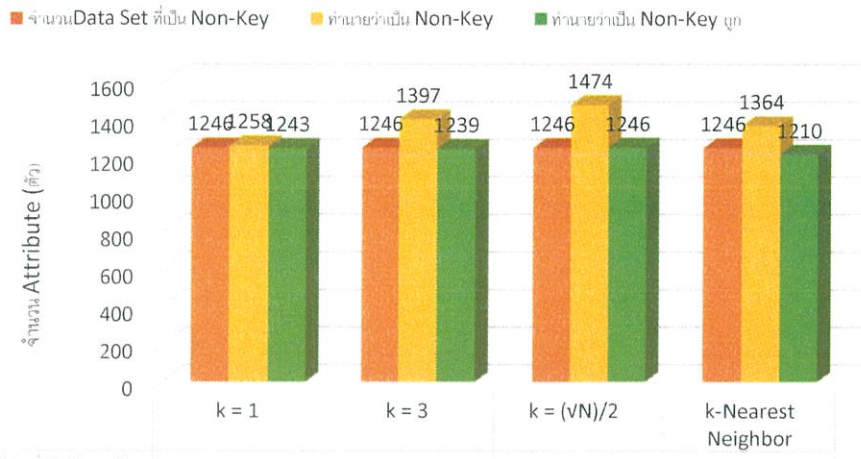
จากการทดลองพบว่าเมื่อค่า k = 1 มีการทำนายว่าเป็น Key ทั้งหมด 221 ตัวและทำนายถูก 218 ตัว เมื่อเพิ่มค่า k เป็น k = 3 จำนวนการทำนายว่าเป็น Key ลดลงเหลือ 82 ตัวและทำนายถูก 75 ตัว เมื่อใช้ k ตามสูตร $k = \frac{\sqrt{N}}{2}$ คือ k = 19 จำนวนทำนายว่าเป็น Key ลดลงเป็นอย่างมาก เหลือเพียง 5 ตัวแต่ทำนายถูกทั้งหมด โดยค่า k = 1 ทำนายว่าเป็น Key เป็นจำนวนมากที่สุด ตามด้วย k = 3 และ $k = \frac{\sqrt{N}}{2}$ หรือในที่นี้คือ 19



รูป 4.7 กราฟแสดงผลเปรียบเทียบค่าPrecision ของคลาส Key ของ K-Nearest Neighbors

จากการคำนวณค่า Precision ในการทำนายว่าเป็น Key ได้อันดับดังนี้ คือ k = 1 มีค่าPrecision มากที่สุด เท่ากับ 93.5622 % ตามมาด้วย k = 3 มีค่า Precision เท่ากับ 32.1888% และค่าPrecision น้อยที่สุดคือ 2.1459 % เมื่อ k = $\frac{\sqrt{N}}{2}$ หรือในที่นี้คือ 19 เพราะที่ค่า k นี้เลือกเดาค่าเป็น Non-Key เป็นจำนวนมาก

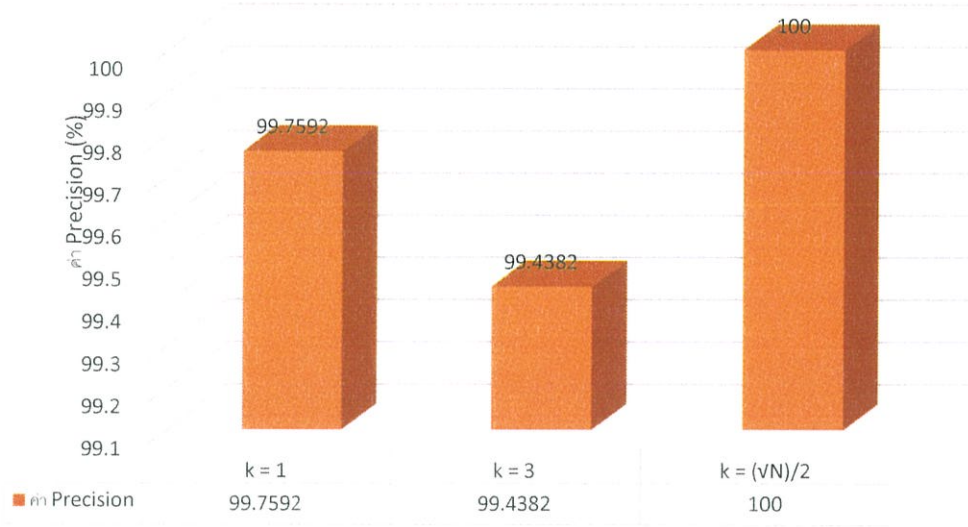
4.2.5 เปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key โดยใช้ค่า k ใน K-Nearest Neighbors ที่ต่างกัน



	k = 1	k = 3	k = (√N)/2	k-Nearest Neighbor
จำนวนData Set ที่เป็น Non-Key	1246	1246	1246	1246
ทำนายว่าเป็น Non-Key	1258	1397	1474	1364
ทำนายว่าเป็น Non-Key ถูก	1243	1239	1246	1210

รูป 4.8 กราฟแสดงผลเปรียบเทียบผลการทำนายข้อมูลประเภท Non-Key ของ K-Nearest Neighbors

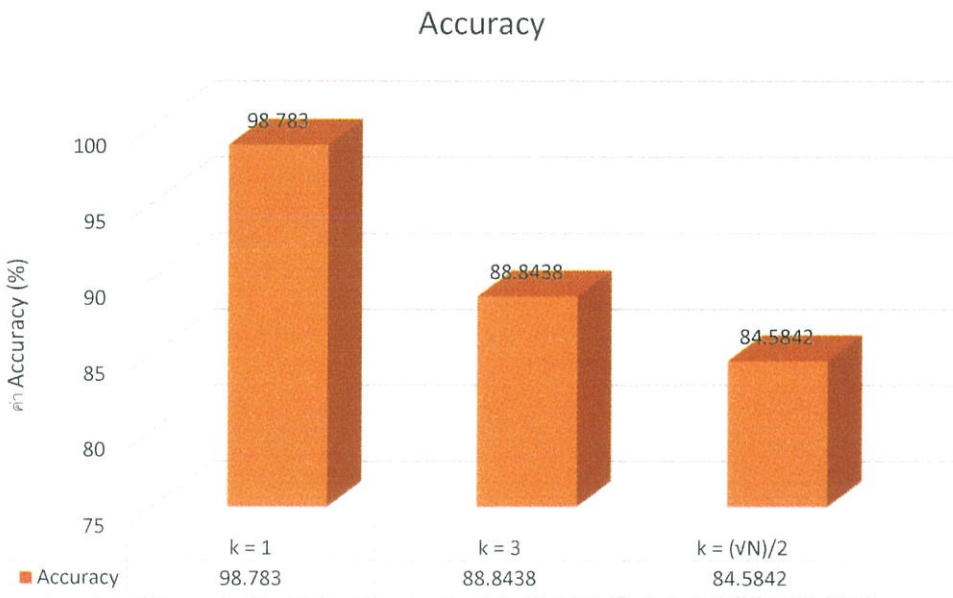
จากการทดลองพบว่าเมื่อค่า k = 1 มีการทำนายว่าเป็น Key ทั้งหมด 1,258 ตัวและทำนายถูก 1,243 ตัว เมื่อเพิ่มค่า k เป็น k = 3 จำนวนการทำนายว่าเป็น Non-Key ทั้งหมด 1,397 ตัวและทำนายถูก 1,239 ตัว เมื่อใช้ k ตามสูตร $k = \frac{\sqrt{N}}{2}$ คือ k = 19 จำนวนทำนายว่าเป็น Non-Key ทั้งหมด 1,474 ตัวและทำนายถูก 1,246 ตัว โดยค่า $k = \frac{\sqrt{N}}{2}$ หรือในที่นี้คือ 19 ทำนายว่าเป็น Non-Key เป็นจำนวนมากที่สุด ตามด้วย k = 3 และ k = 1



รูป 4.9 กราฟแสดงผลเปรียบเทียบค่า Precision ของคลาส Non-Key ของ K-Nearest Neighbors

จากการคำนวณค่า Precision ในการทำนายว่าเป็น Non-Key ได้อันดับดังนี้ คือ $k = \frac{\sqrt{N}}{2}$ หรือในที่นี้คือ 19 มีค่าPrecision มากที่สุด เท่ากับ 100 % เพราะมีการทำนายว่าเป็นNon-Key เป็นจำนวนมากจนครอบคลุมข้อมูลที่เป็น Non-Key ทั้งหมด ตามมาด้วย k = 1 มีค่า Precision เท่ากับ 99.7592% และค่าPrecision น้อยที่สุดคือ 99.4382 % เมื่อ k = 3 เนื่องจากข้อมูลที่นำมาเทรนมีจำนวนข้อมูล Non-Key มาก จึงทำให้โมเดลมีความสามารถในการแบ่งประเภทข้อมูลแบบ Non-Key ได้อย่างมีประสิทธิภาพสูง

4.2.6 ผลการเปรียบเทียบค่า Accuracy ของค่า k ทั้งหมด



รูป 4.10 กราฟแสดงผลการเปรียบเทียบค่า Accuracy ของค่า k ทั้งหมด

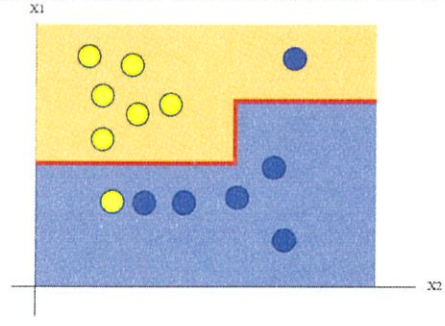
จากการทดลองพบว่าที่ k = 1 มีค่า Accuracy สูงที่สุด คือ 98.7830 % และอันดับรองลงมาคือ k = 3 เท่ากับ 88.8438 % และท้ายสุดคือ k = $\frac{\sqrt{N}}{2}$ หรือในที่นี้คือ 19 มีค่า Accuracy เท่ากับ 84.5842 % ยิ่งค่า k มีค่ามากยิ่งพบว่ามีการทำนายว่าเป็น Non-Key เพิ่มมากขึ้นเนื่องจากอัตราส่วนระหว่างข้อมูลแบบ Non-Key และ Key ที่ใช้เทรนมีอัตราส่วนไม่เท่ากัน โดยแบบ Non-Key มีมากกว่า เมื่อเพิ่มค่า k จึงมีโอกาสที่เจอเพื่อนบ้าน (Neighbor) ที่เป็น Non-Key มากกว่า Key ผลการทำนายจึงออกเป็น Non-Key เยอะมากขึ้น ค่า k = 1 จึงเหมาะสมสำหรับขนาดและอัตราส่วนของข้อมูลที่ใช้อยู่

4.3 วิเคราะห์ผลการทดลอง

จากการทดลองในการหาค่าหลักในการแสดงผลที่ได้มีการทดสอบนั้น พบว่าอัลกอริทึม K-Nearest Neighbors ให้ผลลัพธ์ที่มีประสิทธิภาพดีที่สุดที่ความแม่นยำ 87% เมื่อเปรียบเทียบกับอัลกอริทึม Decision Tree และอัลกอริทึม Naïve Bayes ของระบบที่จัดทำในรูปแบบ Machine Learning

ในส่วนของอัลกอริทึม Naïve Bayes ใช้หลักการความน่าจะเป็นที่จะเกิดขึ้นของความสัมพันธ์ของค่าในAttribute ต่างๆกับ Class ที่หาได้จาก Training Set เช่น โอกาสที่มีคำว่า Normal และอยู่ในคลาส Key เป็นต้นโดยมีเกณฑ์การตัดสินใจคือหากClass ใดมีความน่าจะเป็นที่จะเกิดขึ้นมากกว่าให้ถือว่าข้อมูลที่ต้องการทำนายเป็นคลาสนั้น แต่เนื่องจาก Training Set ที่ใช้ทำการทดลองมีความโน้มเอียงไปทาง Class Non-Key มากกว่า จึงทำให้ความน่าจะเป็นของการเป็น class Key น้อยกว่า Class Non-Key อย่างเห็นได้ชัดจึงทำให้มักจะทำนายออกมาเป็น Non-Key เสียส่วนใหญ่

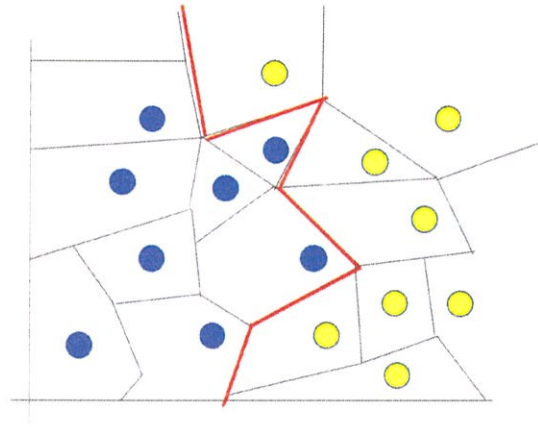
และเนื่องด้วยการลักษณะการทำงานของอัลกอริทึม Decision Tree มีการแบ่งขอบเขตการตัดสินใจ (Dicision Boundary) ที่เข้มงวดคือเส้นขอบเขตจะเป็นเส้นตรงที่ขนาดกับแกนเท่านั้น ไม่สามารถเอียงได้ แต่ลักษณะของ Training Set ที่ใช้ การเกาะกลุ่มของข้อมูลแต่ละคลาสไม่ได้เกาะกลุ่มแยกตัว ซ้ายขาวหรือบนล่างอย่างชัดเจน จึงทำให้ประสิทธิภาพการทำนายค่าหลักลดน้อยลงไป



รูป 4.11 ตัวอย่างลักษณะขอบเขตการตัดสินใจของ Decision Tree

ส่วนอัลกอริทึมที่มีความแม่นยำสุดในโจทย์ปัญหานี้ คือ K-Nearest Neighbors อัลกอริทึมชนิดนี้มีการทำงาน 2 รูปแบบคือ แบบเมื่อ k เท่ากับ 1 และเมื่อ k มากกว่า ทางผู้จัดทำได้ทำการทดลองเลือกค่า k ที่จะนำมาใช้และสรุปจากผลลัพธ์เลือกใช้ค่า k เท่ากับ 1 ที่ให้ค่าความแม่นยำสูงสุดเนื่องจากมีคลาสที่ต้องการให้แบ่งเพียง 2 คลาส (2 Class Problem) และขนาดของ Training Set มีจำนวนไม่มากนัก หลักการทำงานของ K-Nearest Neighbors ทีทำให้มีความแม่นยำเหนือกว่าอัลกอริทึมอื่นเพราะว่าลักษณะของขอบเขตการตัดสินใจเหมาะกับรูปแบบการกระจายตัวของ

Training Set ที่ไม่เป็นเป็นกลุ่มซ้ายขาวหรือบนล่างอย่างชัดเจน โดยลักษณะของขอบเขตการตัดสินใจของ K-Nearest Neighbors เกิดจากการสร้างขอบเขตการตัดสินใจย่อยที่ไม่ใช่ขอบเขตที่เกิดจากเส้นตรงขนานแกนจากข้อมูลทุกแถวใน Training Set ขอบเขตที่เกิดขึ้นออกมามีลักษณะคล้ายแผนภาพโวโรนอย จากนั้นจึงสรุปออกมาเป็นขอบเขตการตัดสินใจที่ใช้จริงของแต่ละคลาส ทำให้ขอบเขตของการตัดสินใจของอัลกอริทึมชนิดนี้มีความยืดหยุ่นมากกว่าเมื่อเทียบกับ Decision Tree จึงส่งผลให้สามารถทำนายข้อมูลได้แม่นยำกว่า



รูป 4.12 ตัวอย่างลักษณะขอบเขตการตัดสินใจของ K-Nearest Neighbors เมื่อ k = 1

สุดท้ายคือ Statistical Approach ที่ถูกสร้างขึ้นจากกระบวนการเก็บค่าทางสถิติที่ทางคณะผู้จัดทำได้สร้างขึ้นนั้นมีประสิทธิภาพที่แย่ที่สุด เนื่องจากกฎที่ได้สร้างขึ้นเพื่อรองรับผลทางสถิตินั้นเกิดจากชุดข้อมูลตัวอย่างที่ทางคณะผู้จัดทำได้จัดเก็บขึ้นมาจากชุดข้อมูลที่เปิดให้ทดลองใช้งานอินเตอร์เน็ตเท่านั้น ซึ่งทำให้ความสามารถในการเรียนรู้ข้อมูลหยุดนิ่งที่ชุดข้อมูลตัวอย่าง เมื่อระบบเจอข้อมูลรูปแบบใหม่ที่มีโครงสร้างและลักษณะของค่าหลักในการแสดงผลที่ไม่เหมือนชุดข้อมูลตัวอย่าง ระบบจะไม่มีความสามารถในการตัดสินใจเลือกค่าที่ถูกต้องได้ ซึ่งแตกต่างจากระบบที่ถูกสร้างขึ้นจาก Machine Learning ที่จะมีการเรียนรู้ใหม่ตลอดเวลาเมื่อมีการตัดสินใจที่ผิดพลาดขึ้นทำให้มีโอกาสในการเข้าถึงข้อมูลรูปแบบใหม่และโครงสร้างของข้อมูลที่อาจไม่เคยเห็นได้

บทที่ 5

สรุปผลผลการทดลองและแนวทางการพัฒนาต่อ

5.1 สรุปผลการทดลอง

การแสดงผลข้อมูลนั้นเป็นส่วนสำคัญที่ช่วยให้ผู้ใช้งานเข้าถึงข้อมูลได้ง่ายขึ้นและสามารถเข้าใจในภาพรวมของสถานการณ์ที่เกิดขึ้นจากข้อมูลชนิดต่าง ๆ ที่ได้มีการเก็บบันทึกในปัจจุบัน ผู้ใช้งานจะต้องทำการสร้างระบบแสดงผลด้วยตนเองทุกขั้นตอนไม่ว่าจะเป็นการวิเคราะห์ข้อมูล เลือกการแสดงผล รวมถึงกระบวนการด้านเทคนิคซึ่งอาจสร้างความลำบากให้แก่ผู้ใช้งาน

ในโครงงานนี้ได้นำเสนอระบบแสดงผลข้อมูลจำลองบนแผนผังเมือง 3 มิตินั้นจะทำการสร้างการแสดงผลให้ผู้ใช้งานได้แบบอัตโนมัติ เพื่อให้ผู้ใช้งานลดระยะเวลาในการสร้างการแสดงผล รวมถึงไม่จำเป็นต้องอาศัยทักษะด้านเทคนิคและกราฟฟิกในการทำงานอีกด้วย เพียงแค่ผู้ใช้งานนั้นให้ข้อมูลในรูปแบบ API แก่ระบบแสดงผล ระบบจะทำการวิเคราะห์ข้อมูล หาค่าหลักที่ใช้ในการแสดงผลบนผังเมือง 3 มิติ เลือกรูปแบบการแสดงผล และแสดงผลข้อมูลขึ้นสู่ระบบโดยอัตโนมัติ

โดยการหาค่าหลักที่ใช้ในการแสดงผลที่ได้กล่าวถึงในขั้นต้นนั้น มีการใช้กระบวนการ Machine Learning อัลกอริทึม K-Nearest Neighbors เป็นหลักในการทำงาน โดยเลือกใช้อัลกอริทึมจากผลการทดสอบที่แม่นยำที่สุดตามที่ได้แสดงค่าไปในบทที่ 4 และผู้ใช้งานสามารถปรับปรุงแก้ไขค่าที่ระบบเลือกให้ได้ในกรณีที่ค่าหลักที่ใช้ในการแสดงผลที่ระบบเลือกมีความผิดพลาดหรือไม่ตรงกับความต้องการ โดยระบบจะทำการนำค่าที่ถูกต้องที่ผู้ใช้ได้เลือกไว้ไปใช้ในการปรับปรุง Training Set ที่จะใช้กับ Machine Learning ของระบบต่อไป

จากการทำการทดลองเพื่อทำโครงงานชิ้นนี้ทำให้ทราบว่าบริการข้อมูลของหน่วยงานต่างๆ ผ่านทาง API นั้นยังไม่ค่อยมีให้ใช้อย่างแพร่หลายในประเทศไทย จึงทำให้การเก็บข้อมูลส่วนใหญ่มาจากแหล่งข้อมูลต่างประเทศและมีแหล่งข้อมูลที่จำกัด ทำให้สังเกตได้ว่าการออกแบบ API ต่าง ๆ ไม่มีหลักการตายตัวและจุดประสงค์ในการจัดทำซึ่งอาจมีจุดประสงค์เดียวต่อ 1 ชุดข้อมูล หรือรวมหลาย ๆ จุดประสงค์ต่อ 1 ชุดข้อมูล ด้วยลักษณะเหล่านี้ทำให้ข้อมูล Training Set มีการกระจายปะปนกันไม่ชัดเจนในแต่ละคลาสและแนวโน้มของข้อมูลที่เก็บได้เอียงไปทางฝั่ง Non-Key จึงทำให้ K-Nearest Neighbors สามารถทำนายค่าหลักได้อย่างแม่นยำที่สุด เพราะมีขอบเขตในการตัดสินใจที่ยืดหยุ่นเหมาะกับข้อมูลที่ซับซ้อน ซึ่งหากจำนวน Training Set มีจำนวนมากขึ้น และชุดข้อมูลไม่มีความโน้มเอียง อาจจะทำให้ผลการเลือกอัลกอริทึมที่เหมาะสมเปลี่ยนไป แต่ด้วยข้อมูลที่มีในปัจจุบัน K-Nearest Neighbors จึงเป็นอัลกอริทึมที่เหมาะสมที่สุด

5.2 แนวทางการพัฒนาต่อ

1) พัฒนาให้ระบบสามารถแยกแยะหมวดหมู่ของข้อมูลได้ เช่น พยากรณ์อากาศ, การคมนาคม, ข้อมูลจากเซนเซอร์ ฯลฯ

2) พัฒนากลไก Machine Learning ให้สามารถพัฒนาความสามารถขึ้นได้เองโดยใช้ข้อมูลที่ได้รับจากผู้ใช้

3) เพิ่มรูปแบบการแสดงผลให้หลากหลายและตอบโจทย์ผู้ใช้งานมากขึ้น

4) เพิ่มส่วนปรับแต่งรูปแบบการแสดงผลเพื่อให้ผู้ใช้งานสามารถปรับแต่งได้เองตามใจ

บรรณานุกรม

พิทยารัตน์ แซ่โง้ว และศกลวรรณ นาวาเจริญ. 2559.
"ระบบการทำนายความเคลื่อนไหวของกลุ่มคนจากสื่อสังคมออนไลน์." วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Francesco G. Brundu1, Edoardo Patti1, Matteo Del Giudice, Anna Osello, Michela Ramassotto, Fabrizio Massara, Francesca Marchi, Alberto Musetti, Enrico Macii1 and Andrea Acquaviva. 2016. "A scalable middleware-based infrastructure for energy management and visualization in city districts". EAI Endorsed Transactions on Cloud Systems : 12 2016 - 06 2017 | Volume 3 | Issue 9 | e1

Sung Ah Kim,Dongyoun Shin,Yoon Choe,Thomas Seibert and Steffen P. Walz. 2011. "Integrated energy monitoring and visualization system for Smart Green City development". Automation in Construction 22 (2012) 51–59

Biglobster. 2558. **ภาษาซีชาร์ป.** [Online].Available : https://th.wikipedia.org/wiki/ภาษาซีชาร์ป

พงศธร สุทธิ และสราวุธ ดอนคำชัยมงคล. **Language C# In Loei .**
[Online].Available : http://tonkung.ueuo.com/index.html

mindphp.com. 2560. **API คืออะไร.**
[Online].Available : www.mindphp.com/คู่มือ/73-คืออะไร/2038-API-คืออะไร.html

Justin Baker. 2017. **What is an API?.** [Online].Available : https://www.quora.com/What-is-an-API-4

JSON.org.. **Introducing JSON.** [Online].Available : https://www.json.org/

vijay kumar rana. 2016. **Understanding API and their types.**
[Online].Available : http://www.techulator.com/resources/15115-Understanding-API-and-their-types.aspx

Swati Dhingra. 2016. **REST vs. SOAP: Choosing the best web service.**
[Online].Available : http://searchmicroservices.techtarget.com/tip/REST-vs-SOAP-Choosing-the-best-web-service

BattyBot. 2017. **Smart city.** [Online].Available : https://en.wikipedia.org/wiki/Smart_city

Margaret Rouse. 2017. **Definition smart city.**
[Online].Available : http://internetofthingsagenda.techtarget.com/definition/smart-city

The University of Vermont. . **Types of Data or Data Classification.**
[Online].Available : http://www.uvm.edu/~jleonard/AGRI85/fall2003/DataClassification.html

Eston Martz. 2017. **Understanding Qualitative, Quantitative, Attribute, Discrete, and Continuous Data Types.**
[Online].Available : http://blog.minitab.com/blog/understanding-statistics/understanding-qualitative-quantitative-attribute-discrete-and-continuous-data-types

Optakeover. 2017. **Data type.** [Online].Available : https://en.wikipedia.org/wiki/Data_type

Motion Info Studio. 2013. **เป็นมากกว่าข้อมูลด้วย Data Visualization.**
[Online].Available : http://www.motioninfostudio.com/เป็นมากกว่าข้อมูลด้วย-data-visual/#.WhevZzcxXIV

Environment Agency. . **Environment Agency Real Time flood-monitoring API.**
[Online].Available : https://environment.data.gov.uk/flood-monitoring/doc/reference

OpenWeatherMap. . **Weather API.** [Online].Available : https://openweathermap.org/API

ภาคผนวก

ภาคผนวก ก.

ตัวอย่างข้อมูล ก.1 API 1.1.Postal Code Search


```
{
    "postalCodes": [
        {
            "adminCode2": "07",
            "adminCode1": "DI",
            "adminName2": "Ettelbruck",
            "lng": 6.0917001729141855,
            "countryCode": "LU",
            "postalCode": "L-9011",
            "adminName1": "Diekirch",
            "ISO3166-2": "DI",
            "placeName": "Ettelbruck",
            "lat": 49.847514035922494
        },
        {
            "adminCode2": "1721",
            "adminCode3": "3203",
            "adminName3": "St. Gallen",
            "adminCode1": "SG",
            "adminName2": "Wahlkreis St. Gallen",
            "lng": 9.374771118164062,
            "countryCode": "CH",
            "postalCode": "9011",
            "adminName1": "Kanton St. Gallen",
            "ISO3166-2": "SG",
            "placeName": "St. Gallen",
            "lat": 47.42390636154281
        }
    ]
}
```

ตัวอย่างข้อมูล ก.2 API 2.1. A severe alert


```
{
	"response": {
		"version": "0.1",
		"termsofService": "http://www.wunderground.com/weather/API/d/terms.html",
		"features": {
			"alerts": 1
		}
	},
	"alerts": [
		{
			"type": "HEA",
			"wtype_meteoalarm": "5",
			"wtype_meteoalarm_name": "Extreme high temperature",
			"level_meteoalarm": "4",
			"level_meteoalarm_name": "Red",
			"level_meteoalarm_description": "The weather is very dangerous. Exceptionally intense meteorological phenomena have been forecast. Major damage and accidents are likely, in many cases with threat to life and limb, over a wide area. Keep frequently informed about detailed expected meteorological conditions and risks. Follow orders and any advice given by your authorities under all circumstances, be prepared for extraordinary measures.",
			"description": "Extreme high temperature",
			"date": "2012-08-21 04:00:00 GMT",
			"date_epoch": "NA",
			"expires": "2012-08-22 16:00:00 GMT",
			"expires_epoch": "NA",
			"message": "Extreme high temperature)",
			"phenomena": "NA",
			"significance": "NA",
			"attribution": "Information provided by, EUMETNET - MeteoAlarm Note: Time delays between this website and Meteoalarm.eu are possible. For the most up-to-date information about alert levels as published by the participating National Meteorological Services, please visit Meteoalarm. For terms of use of this information, and copyright information, see Meteoalarm Terms of Use."
		}
	]
}
```


ตัวอย่างข้อมูล ก.3 API 2.2. Reporting stations


```
{
	"response": {
		"version": "0.1",
```

```
                "termsofService": "http://www.wunderground.com/weather/API/d/terms.html",
                "features": {
                        "almanac": 1
                }
        },
        "almanac": {
                "airport_code": "KSFO",
                "temp_high": {
                        "normal": {
                                "F": "71",
                                "C": "22"
                        },
                        "record": {
                                "F": "89",
                                "C": "31"
                        },
                        "recordyear": "1970"
                },
                "temp_low": {
                        "normal": {
                                "F": "54",
                                "C": "12"
                        },
                        "record": {
                                "F": "48",
                                "C": "8"
                        },
                        "recordyear": "1953"
                }
        }
}
```


ตัวอย่างข้อมูล ก.4 API 2.3 Astronomy


```
{
        "response": {
                "version": "0.1",
                "termsofService": "http://www.wunderground.com/weather/API/d/terms.html",
                "features": {
                        "astronomy": 1
                }
```

```
        },
        "moon_phase": {
                "percentIlluminated": "81",
                "ageOfMoon": "10",
                "current_time": {
                        "hour": "9",
                        "minute": "56"
                },
                "sunrise": {
                        "hour": "7",
                        "minute": "01"
                },
                "sunset": {
                        "hour": "16",
                        "minute": "56"
                }
        }
}
```


ตัวอย่างข้อมูล ก.5 API 3.1. Hotels name


```
{
  "results": {
    "locations": [
      {
        "cityName": "Москва",
        "fullName": "Москва, Россия",
        "countryCode": "RU",
        "countryName": "Россия",
        "iata": [
          "BKA",
          "DME",
          "SVO",
          "VKO",
          "MOW"
        ],
        "id": "12153",
        "hotelsCount": "930",
        "location": {
          "lat": "55.752222200",
          "lon": "37.615555600"
```

```
            },
            "_score": 3.2695
          }
        ],
        "hotels": [
          {
            "label": "Отель Novotel Moscow City",
            "locationName": "Москва, Россия",
            "locationId": "12153",
            "id": "1074388",
            "fullName": "Отель Novotel Moscow City, Москва, Россия",
            "location": {
              "lat": "55.747160",
              "lon": "37.539302"
            }
          }
        ]
      },
      "status": "ok"
}
```


ตัวอย่างข้อมูล ก.6 API 3.2. Hotels location


```
{
  "results": {
    "locations": [
      {
        "id": "12167",
        "type": "City",
        "countryIso": "RU",
        "name": "Новосибирск",
        "state": "",
        "fullname": "Новосибирск, Россия",
        "geo": {
          "lat": "55.028705000",
          "lon": "82.906898000"
        }
      }
    ],
    "hotels": [
```

```
        {
          "id": "380923",
          "name": "AZIMUT Отель Сибирь",
          "locationId": "12167",
          "location": {
            "lat": "55.029140",
            "lon": "82.905990"
          }
        }
      ]
    },
    "status": "ok"
}
```


ตัวอย่างข้อมูล ก.7 API 3.3. Hotels cost


```
{
  "location": {
    "country": "Russia",
    "geo": {
      "lon": 37.617508,
      "lat": 55.752041
    },
    "state": null,
    "name": "Moscow"
  },
  "priceAvg": 60897.74,
  "pricePercentile": {
    "3": 28863.56,
    "10": 28863.56,
    "35": 47805.27,
    "50": 59531.09,
    "75": 65435,
    "99": 120128.17
  },
  "hotelName": "Mercure Arbat Moscow",
  "stars": 4,
  "locationId": 12153,
  "hotelId": 333561,
  "priceFrom": 28863.56
}
```

ตัวอย่างข้อมูล ก.8 API 4.1. Sentimental

Base URL: http://API.wefeelfine.org:8080/ShowFeelings?

returnfields – comma delimited list of all the data you want to get back. Options are:

imageid (used to get images)
feeling (sad, depressed, happy, etc. – full list of valid feelings is here)
sentence (the full sentence that went with the feeling)
posttime (epoch timestamp of the time the post was scraped)
postdate (YYYY-MM-DD of the day the post was scraped)
posturl (the url of the post)
gender (1 (male) or 0 (female))
born (YYYY from when the user was born)
country (country of the user)
state (state of the user)
city (city of the user)
lat (latitude of the user's city)
lon (longitude of the user's city)
conditions (the weather when the post was written in that city)
1 = sunny
2 = rainy
3 = snowy
4 = cloudy

ตัวอย่างข้อมูล ก.9 API 5.1. Near bus stops


```
[
  {
    "name": "1st Ave & Pine St",
    "summary_text": "1st Ave & Pine St (0.1 mi),15, 18, 21, 22, 56, 81",
    "summary_html":
      "walkscore"
        "stop": "1st Ave & Pine St",
        "distance": "0.1 mi",
       "Bus": 15, 18, 21, 22, 56, 81,
   "route_summary": [
      {
        "category": "Bus",
        "id": "r437",
        "name": "15"
```

```
        },
        {
          "category": "Bus",
          "id": "r448",
          "name": "18"
        },
        ...
      ],
      "lon": -122.341072,
      "lat": 47.609840400000003,
      "distance": 0.1,
      "id": "s16679"
    },
    ...
]
```


ตัวอย่างข้อมูล ก.10 API 5.2. Routes and stops


```
{
  "routes": {
    "r446": {
      "category": "Bus",
      "agency": "Metro Transit",
      "name": "10",
      "agency_url": "http://metro.kingcounty.gov",
      "stop_ids": [
        "s10644",
        "s14748",
        ...
      ],
      "id": "r446"
    },
    "r462": {
      "category": "Bus",
      "agency": "Metro Transit",
      "name": "81",
      "agency_url": "http://metro.kingcounty.gov",
      "stop_ids": [
        "s13828",
        "s11155",
        ...
```

```
        ],
        "id": "r462"
      },
      ...
    },
    "stops": {
      "s17737": {
        "lat": 47.610702500000002,
        "route_ids": [
          "r415",
          "r282",
          ...
        ],
        "lon": -122.340332,
        "id": "s17737",
        "name": "2nd Ave & Stewart St"
      },
      "s11544": {
        "lat": 47.609794600000001,
        "route_ids": [
          "r478",
          "r254",
          ...
        ],
        "lon": -122.340782,
        "id": "s11544",
        "name": "1st Ave & Pine St"
      },
      ...
    }
}
```


ตัวอย่างข้อมูล ก.11 API 6.1. Car park availability


```
{
 "Status": "Success",
 "Result": [
  {
   "geometries": [
    {
     "coordinates": "27582.895,31962.0601"
```

```
      }
     ],
     "lotsAvailable": "76",
     "lotType": "C",
     "carparkNo": "A0007"
    },
    {
     "geometries": [
      {
       "coordinates": "27582.895,31962.0601"
      }
     ],
     "lotsAvailable": "17",
     "lotType": "M",
     "carparkNo": "A0007"
    }
  ]
}
```


ตัวอย่างข้อมูล ก.12 API 6.2. Car park locations and rates


```
{
 "Status": "Success",
 "Result": [
  {
   "weekdayMin": "30 mins",
   "geometries": [
    {
     "coordinates": "31045.6165,31694.0055"
    },
    {
     "coordinates": "31126.0755,31564.9876"
    }
   ],
   "ppName": "ALIWAL STREET ",
   "endTime": "05.00 PM",
   "weekdayRate": "$0.50",
   "startTime": "08.30 AM",
   "ppCode": "A0004",
   "sunPHRate": "$0.50",
   "satdayMin": "30 mins",
```

```
    "sunPHMin": "30 mins",
    "parkingSystem": "C",
    "parkCapacity": 69,
    "vehCat": "Car",
    "satdayRate": "$0.50"
   },
   {
    "weekdayMin": "30 mins",
    "geometries": [
     {
      "coordinates": "31045.6165,31694.0055"
     },
     {
      "coordinates": "31126.0755,31564.9876"
     }
    ],
    "ppName": "ALIWAL STREET ",
    "endTime": "10.00 PM",
    "weekdayRate": "$0.50",
    "startTime": "05.00 PM",
    "ppCode": "A0004",
    "sunPHRate": "$0.50",
    "satdayMin": "30 mins",
    "sunPHMin": "30 mins",
    "parkingSystem": "C",
    "parkCapacity": 69,
    "vehCat": "Car",
    "satdayRate": "$0.50"
   }
  ]
}
```


ตัวอย่างข้อมูล ก.13 API 6.3. Car park locations and rates for application


```
{
  "Status": "Success",
  "Result": [
    {
      "monthlyRate": "130",
      "parkingHrs": "Everyday;7am to 7am (24 hrs)",
      "ticketType": "Commercial",
```

```
        "ppCode": "S0151",
        "ppName": "SEMBAWANG DRIVE HEAVY VEHICLE PARK",
        "vehCat": "Heavy Vehicle",
        "geometries": [
          {
            "coordinates": "25955.98820736,49175.13265519"
          }
        ]
      },
      {
        "monthlyRate": "80",
        "parkingHrs": "Mon-Sat;8.30am to 5pm",
        "ticketType": "Commercial",
        "ppCode": "GS005",
        "ppName": "SEMBAWANG HILL ESTATE ",
        "vehCat": "Car",
        "geometries": [
          {
            "coordinates": "27500.17301387,39367.14352029"
          },
          {
            "coordinates": "27531.73388722,39298.99834641"
          },
          {
            "coordinates": "27552.16283837,39473.62167334"
          }
        ]
      }
    ],
    "Message": ""
}
```


ตัวอย่างข้อมูล ก.14 API 6.4. Past 3 years of transaction records


```
{
  "Result": [
    {
      "project": "TURQUOISE",
      "marketSegment": "CCR",
      "transaction": [
        {
```

```
        "contractDate": "0715",
        "area": "203",
        "price": "2900000",
        "propertyType": "Condominium",
        "typeOfArea": "Strata",
        "tenure": "99 yrs lease commencing from 2007",
        "floorRange": "01-05",
        "typeOfSale": "3",
        "district": "04",
        "noOfUnits": "1"
      },
      {
        "contractDate": "0116",
        "area": "200",
        "price": "3014200",
        "propertyType": "Condominium",
        "typeOfArea": "Strata",
        "tenure": "99 yrs lease commencing from 2007",
        "floorRange": "01-05",
        "typeOfSale": "3",
        "district": "04",
        "noOfUnits": "1"
    },
  ],
  "street": "COVE DRIVE",
  "y": "24997.821719180001",
  "x": "28392.530515570001"
}
  ],
  "Status": "Success"
}
```


ตัวอย่างข้อมูล ก.15 API 7.1. Air quality


```
HTTP/1.1 200 OK
{
  "status": "ok",
  "data": {
      "idx":7397,
      "aqi":71,
      "time":{
```

```
        "v":1481396400,
        "s":"2016-12-10 19:00:00",
        "tz":"-06:00"
    },
    "city":{
        "name":"Chi_sp, Illinois",
        "url":"https://aqicn.org/city/usa/illinois/chi_sp/",
        "geo":["41.913600","-87.723900"]
    },
    "iaqi":{
        "pm25":{
            "v":71
        }
    }
}
```


ตัวอย่างข้อมูล ก.16 API 8.1. Search location from IP


```
{
        "ip": "192.30.253.112",
        "country_code": "US",
        "country_name": "United States",
        "region_code": "CA",
        "region_name": "California",
        "city": "San Francisco",
        "zip_code": "94107",
        "time_zone": "America/Los_Angeles",
        "latitude": 37.7697,
        "longitude": -122.3933,
        "metro_code": 807
}
```


ตัวอย่างข้อมูล ก.17 API 9.1. City bikes


```
{
    "networks": [
        {
            "company": [
                "Bike U Sp. z o.o."
```

```
                ],
                "href": "/v2/networks/bbbike",
                "id": "bbbike",
                "location": {
                        "city": "Bielsko-Biała",
                        "country": "PL",
                        "latitude": 49.8225,
                        "longitude": 19.044444
                },
                "name": "BBBike"
        },
        {
                "company": [
                        "PBSC"
                ],
                "href": "/v2/networks/bixi-montreal",
                "id": "bixi-montreal",
                "location": {
                        "city": "Montreal, QC",
                        "country": "CA",
                        "latitude": 45.5086699,
                        "longitude": -73.55399249999999
                },
                "name": "Bixi"
        },

    ]
}
```


ตัวอย่างข้อมูล ก.18 API 10.1. Currently active alerts


```
{
  "id":"4f040eedfd58153800598f59c39c44f9",
  "loc":{
    "long":-84.76328,
    "lat":33.35346
  },
  "details":{
    "type":"FF.A",
    "name":"FLASH FLOOD WATCH",
    "loc":"GAZ053",
```

"emergency":false,

"color":"32CD32",

"cat":"flood",

"body":"...FLASH FLOOD WATCH IN EFFECT THROUGH WEDNESDAY MORNING...\n\nThe National Weather Service in Peachtree City has issued a\n\n* Flash Flood Watch for portions of central Georgia...east \ncentral Georgia...north central Georgia...northeast Georgia... \nnorthwest Georgia and west central Georgia...including the \nfollowing areas...in central Georgia...Baldwin...Butts... \nJasper...Jones...Monroe and Putnam. In east central Georgia... \nGreene...Hancock...Taliaferro...Warren and Wilkes. In north \ncentral Georgia...Barrow...Clayton...Cobb...DeKalb...Douglas...\nFayette...Gwinnett...Henry...Morgan...Newton...North Fulton...\nRockdale...South Fulton and Walton. In northeast Georgia... \nClarke...Jackson...Madison...Oconee and Oglethorpe. In \nnorthwest Georgia...Carroll...Haralson and Paulding. In west \ncentral Georgia...Coweta...Harris...Heard...Lamar... \nMeriwether...Pike...Spalding...Talbot...Troup and Upson. \n\n* Through Wednesday morning\n\n* A weak frontal boundary will remain stationary across west\ncentral Georgia through the Atlanta Metro to near Athens through\ntonight. Heavy rainfall occurred along the Interstate 20 \ncorridor Monday night\/early Tuesday morning with widespread \nvalues between one and three inches, with some spotty 4 to 6 \ninches. An additional one to three inches with locally higher \namounts is possible through the day Tuesday and into the \novernight hours. \n\n* Flooding in low lying and flood prone areas will be possible,\nespecially in urban areas along the Interstate 20 corridor. In \naddition, creeks, streams and rivers may overflow their banks.\n\nA Flash Flood Watch means that conditions may develop that lead\nto flash flooding. Flash flooding is a very dangerous situation.\n\nYou should monitor later forecasts and be prepared to take action\nshould Flash Flood Warnings be issued.",

"bodyFull":"WGUS62 KFFC 200719\nFFAFFC\n\nURGENT - IMMEDIATE BROADCAST REQUESTED\nFlood Watch\nNational Weather Service Peachtree City GA\n319 AM EDT Tue Jun 20 2017\n\n\n\nGAZ025-027-031>039-041>062-066>073-078-079-201530-\n\/O.NEW.KFFC.FF.A.0006.170620T0719Z-170621T1200Z\/\n\/00000.0.ER.000000T0000Z.000000T0000Z.000000T0000Z.OO\/\nJackson-Madison-Paulding-Cobb-North Fulton-Gwinnett-Barrow-Clarke-\nOconee-Oglethorpe-Wilkes-Haralson-Carroll-Douglas-South Fulton-\nDeKalb-Rockdale-Walton-Newton-Morgan-Greene-Taliaferro-Heard-\nCoweta-Fayette-Clayton-Spalding-Henry-Butts-Jasper-Putnam-Hancock-\nWarren-Troup-Meriwether-Pike-Upson-Lamar-Monroe-Jones-Baldwin-\nHarris-Talbot-\nIncluding the cities of Marietta, Atlanta, Lawrenceville, Athens,\nCarrollton, Douglasville, East Point, Decatur, Conyers, \nCovington, Newnan, Peachtree City, Griffin, and Milledgeville\n319 AM EDT Tue Jun 20 2017\n\n\n\n...FLASH FLOOD WATCH IN EFFECT THROUGH WEDNESDAY MORNING...\n\nThe National Weather Service in Peachtree City has issued a\n\n* Flash Flood Watch for portions of central Georgia...east \ncentral Georgia...north central Georgia...northeast Georgia... \nnorthwest Georgia and west central Georgia...including the \nfollowing areas...in central Georgia...Baldwin...Butts... \nJasper...Jones...Monroe and Putnam. In east central Georgia... \nGreene...Hancock...Taliaferro...Warren and Wilkes. In north \ncentral Georgia...Barrow...Clayton...Cobb...DeKalb...Douglas...\nFayette...Gwinnett...Henry...Morgan...Newton...North Fulton...\nRockdale...South Fulton and Walton. In northeast Georgia... \nClarke...Jackson...Madison...Oconee and Oglethorpe. In \nnorthwest Georgia...Carroll...Haralson and Paulding. In west \ncentral Georgia...Coweta...Harris...Heard...Lamar... \nMeriwether...Pike...Spalding...Talbot...Troup and Upson. \n\n* Through Wednesday morning\n\n* A weak frontal boundary will remain stationary across west\ncentral Georgia through the Atlanta Metro to near Athens through\ntonight. Heavy rainfall occurred along the Interstate 20 \ncorridor Monday night\/early Tuesday morning with widespread \nvalues between one and three inches, with some spotty 4 to 6 \ninches. An additional one to three inches with locally higher \namounts is possible through the day Tuesday and into the \novernight hours. \n\n* Flooding in low lying and flood prone areas will be possible,\nespecially in urban areas along the Interstate 20 corridor. In \naddition, creeks, streams and rivers may overflow their banks.\n\nPRECAUTIONARY\/PREPAREDNESS ACTIONS...\n\nA Flash Flood Watch means that conditions may develop

that lead\nto flash flooding. Flash flooding is a very dangerous situation.\n\nYou should monitor later forecasts and be prepared to take action\nshould Flash Flood Warnings be issued.\n\n&&"

},

"timestamps":{

"issued":1497943140,

"issuedISO":"2017-06-20T03:19:00-04:00",

"begins":1497943140,

"beginsISO":"2017-06-20T03:19:00-04:00",

"expires":1498046400,

"expiresISO":"2017-06-21T08:00:00-04:00",

"added":1497943172,

"addedISO":"2017-06-20T03:19:32-04:00"

},

"poly":"",

"geoPoly":null,

"includes":{

"counties":[

"GAC009",

"GAC013",

"GAC035",

"GAC045",

"GAC121",

"GAC131",

"GAC293",

"GAC301",

"GAC317"

],

"fips":[

"13009",

"13013",

"13035",

"13285",

"13293",

"13297",

"13301",

"13317"

],

"wxzones":[

"GAZ025",

"GAZ027",

"GAZ031",

"GAZ071",

```
      "GAZ072",
      "GAZ073",
      "GAZ078",
      "GAZ079"
    ],
    "zipcodes":[
      30002,
      30003,
      30004,
      30005,
      30006,
      30007,
      30008,
      31830,
      31831,
      31833,
      31836,
      39901
    ]
  },
  "place":{
    "name":"coweta",
    "state":"ga",
    "country":"us"
  },
  "profile":{
    "tz":"America\/New_York"
  },
  "active":true
}
```


ตัวอย่างข้อมูล ก.19 API 10.2. Core details of cAPItals


```
{
  "place":{
    "name":"united states",
    "iso":"us",
    "iso3":"usa",
    "isoNum":"840",
    "fips":"us",
```

```
    "continent":"na"
  },
  "profile":{
    "cAPItal":"washington",
    "areaKM":9629091,
    "areaMI":5983239.75,
    "pop":310232863,
    "neighbors":["ca","mx","cu"]
  }
}
```


ตัวอย่างข้อมูล ก.20 API 10.3. Forecasts data


```
{
  "loc": {
    "long": -93.264,
    "lat": 44.98
  },
  "interval": "day",
  "periods": [{
    "timestamp": 1318417200,
    "validTime": "2011-10-12T06:00:00-05:00",
    "dateTimeISO": "2011-10-12T06:00:00-05:00",
    "maxTempC": 21,
    "maxTempF": 70,
    "minTempC": 11,
    "minTempF": 52,
    "avgTempC": 18,
    "avgTempF": 64,
    "tempC": null,
    "tempF": null,
    "pop": 30,
    "precipMM": 4.06,
    "precipIN": 0.16,
    "iceaccum": 0,
    "maxHumidity": 80,
    "minHumidity": 62,
    "humidity": 71,
    "uvi": 9,
```

"pressureMB": 1025,
"pressureIN": 30.27,
"sky": 75,
"snowCM": 0,
"snowIN": 0,
"feelslikeC": 15,
"feelslikeF": 59,
"minFeelslikeC": 15,
"minFeelslikeF": 59,
"maxFeelslikeC": 20,
"maxFeelslikeF": 68,
"avgFeelslikeC": 17,
"avgFeelslikeF": 63,
"dewpointC": 12,
"dewpointF": 54,
"maxDewpointC": 14,
"maxDewpointF": 58,
"minDewpointC": 12,
"minDewpointF": 54,
"avgDewpointC": 13,
"avgDewpointF": 56,
"windDirDEG": 210,
"windDir": "SSW",
"windDirMaxDEG": 140,
"windDirMax": "SE",
"windDirMinDEG": 140,
"windDirMin": "SE",
"windGustKTS": 6,
"windGustKPH": 11,
"windGustMPH": 7,
"windSpeedKTS": 6,
"windSpeedKPH": 11,
"windSpeedMPH": 7,
"windSpeedMaxKTS": 6,
"windSpeedMaxKPH": 11,
"windSpeedMaxMPH": 7,
"windSpeedMinKTS": 4,
"windSpeedMinKPH": 8,
"windSpeedMinMPH": 5,
"windDir80mDEG": 0,
"windDir80m": "ESE",
"windDirMax80mDEG": 140,

```
      "windDirMax80m": "SE",
      "windDirMin80mDEG": 120,
      "windDirMin80m": "ESE",
      "windGust80mKTS": 14,
      "windGust80mKPH": 25,
      "windGst80mMPH": 16,
      "windSpeed80mKTS": 13,
      "windSpeed80mKPH": 24,
      "windSpeed80mMPH": 15,
      "windSpeedMax80mKTS": 14,
      "windSpeedMax80mKPH": 25,
      "windSpeedMax80mMPH": 16,
      "windSpeedMin80mKTS": 12,
    "windSpeedMin80mKPH": 21,
    "windSpeedMin80mMPH": 13,
    "sunriseISO": "2015-06-01T05:29:15-05:00",
    "sunsetISO": "2015-06-01T20:52:40-05:00"
      "weather": "Mostly Cloudy with Isolated Storms",
      "weatherCoded": [{
        "timestamp": 1318417200,
        "wx": "L:L:RW,IS::T"
      },
      {
        "timestamp": 1318456800,
        "wx": "CHC:L:RW"
      }],
      "weatherPrimary": "Isolated Storms",
      "weatherPrimaryCoded": "IS::T",
      "cloudsCoded": "BK",
      "icon": "mcloudyt.png",
      "isDay": true,
      "sunrise": 1318422338,
      "sunset": 1318462389
    },
    ...
    ],
    "profile": {
      "tz": "America/Chicago"
    }
}
}
```

ตัวอย่างข้อมูล ก.21 API 10.4. Core details of places


```
{
  "loc":{
    "lat":36.01562,
    "long":-86.58194
  },
  "place":{
    "name":"La Vergne",
    "state":"TN",
    "stateFull":"Tennessee",
    "country":"US",
    "countryFull":"United States",
    "region":"usec",
    "regionFull":"East Central",
    "continent":"nam",
    "continentFull":"North America"
  },
  "profile":{
    "elevM":177,
    "elevFT":581,
    "pop":34794,
    "tz":"America\/Chicago",
    "tzname":"CDT",
    "tzoffset":-18000,
    "isDST":true,
    "wxzone":[
      "TNZ062"
    ],
    "firezone":[
      "TNZ501"
    ],
    "fips":[
      "47149"
    ],
    "countyid":[
      "TNC149"
    ]
  }
}
```

```
}

{
  "loc":{
    "lat":36.01562,
    "long":-86.58194
  },
  "place":{
    "name":"La Vergne",
    "state":"TN",
    "stateFull":"Tennessee",
    "country":"US",
    "countryFull":"United States",
    "region":"usec",
    "regionFull":"East Central",
    "continent":"nam",
    "continentFull":"North America"
  },
  "profile":{
    "elevM":177,
    "elevFT":581,
    "pop":34794,
    "tz":"America\/Chicago",
    "tzname":"CDT",
    "tzoffset":-18000,
    "isDST":true,
    "wxzone":[
      "TNZ062"
    ],
    "firezone":[
      "TNZ501"
    ],
    "fips":[
      "47149"
    ],
    "countyid":[
      "TNC149"
    ]
  }
}
```

ตัวอย่างข้อมูล ก.22 API 10.5. Core details of airports


```
{
  "id":"kmsp",
  "loc":{
    "lat":44.8819999695,
    "long":-93.2218017578
  },
  "place":{
    "name":"Minneapolis-St Paul International\/Wold-Chamberlain Airport",
    "city":"Minneapolis",
    "state":"MN",
    "stateFull":"Minnesota",
    "country":"US",
    "countryFull":"United States",
    "region":"usnc",
    "regionFull":"North Central",
    "continent":"nam",
    "continentFull":"North America"
  },
  "profile":{
    "id":"kmsp",
    "iata":"msp",
    "local":"msp",
    "type":"la",
    "typeENG":"large airport",
    "elevFT":841,
    "elevM":256.34,
    "pop":null,
    "tz":"America\/Chicago",
    "tzname":"CDT",
    "tzoffset":-18000,
    "isDST":true,
    "wxzone":[
      "MNZ060"
    ],
    "firezone":[
      "MNZ060"
    ],
    "fips":[
```

```
      "27053"
    ],
    "countyid":[
      "MNC053"
    ]
  },
  "relativeTo":{
    "lat":44.9778,
    "long":-93.265,
    "bearing":162,
    "bearingENG":"SSE",
    "distanceKM":11.182,
    "distanceMI":6.948
  }
}
```


ตัวอย่างข้อมูล ก.23 API 11.1. Weather data


```
{
        "coord": {
                "lon": 139.01,
                "lat": 35.02
        },
        "weather": [
                {
                        "id": 800,
                        "main": "Clear",
                        "description": "clear sky",
                        "icon": "01n"
                }
        ],
        "base": "stations",
        "main": {
                "temp": 285.514,
                "pressure": 1013.75,
                "humidity": 100,
                "temp_min": 285.514,
                "temp_max": 285.514,
                "sea_level": 1023.22,
                "grnd_level": 1013.75
```

```
        },
        "wind": {
                "speed": 5.52,
                "deg": 311
        },
        "clouds": {
                "all": 0
        },
        "dt": 1485792967,
        "sys": {
                "message": 0.0025,
                "country": "JP",
                "sunrise": 1485726240,
                "sunset": 1485763863
        },
        "id": 1907296,
        "name": "Tawarano",
        "cod": 200
}
```


ตัวอย่างข้อมูล ก.24 API 11.2. Forecast 5 day


```
{
        "cod": "200",
        "message": 0.0082,
        "cnt": 40,
        "list": [
                {
                        "dt": 1485799200,
                        "main": {
                                "temp": 283.76,
                                "temp_min": 283.76,
                                "temp_max": 283.761,
                                "pressure": 1017.24,
                                "sea_level": 1026.83,
                                "grnd_level": 1017.24,
                                "humidity": 100,
                                "temp_kf": 0
                        },
                        "weather": [
                                {
```

```
                                        "id": 800,
                                        "main": "Clear",
                                        "description": "clear sky",
                                        "icon": "01n"
                                }
                        ],
                        "clouds": {
                                "all": 0
                        },
                        "wind": {
                                "speed": 7.27,
                                "deg": 15.0048
                        },
                        "rain": {},
                        "sys": {
                                "pod": "n"
                        },
                        "dt_txt": "2017-01-30 18:00:00"


        ],
        "city": {
                "id": 1907296,
                "name": "Tawarano",
                "coord": {
                        "lat": 35.0164,
                        "lon": 139.0077
                },
                "country": "none"
        }
}
```


ตัวอย่างข้อมูล ก.25 API 11.3. Forecast 16 day


```
{
        "city": {
                "id": 6940463,
                "name": "Altstadt",
                "coord": {
                        "lon": 11.5752,
                        "lat": 48.137
```

```
        },
        "country": "DE",
        "population": 0
    },
    "cod": "200",
    "message": 0.1094425,
    "cnt": 7,
    "list": [
        {
            "dt": 1487242800,
            "temp": {
                "day": 286.67,
                "min": 272.78,
                "max": 286.67,
                "night": 273.34,
                "eve": 277.05,
                "morn": 281.56
            },
            "pressure": 972.73,
            "humidity": 75,
            "weather": [
                {
                    "id": 800,
                    "main": "Clear",
                    "description": "sky is clear",
                    "icon": "01d"
                }
            ],
            "speed": 1.81,
            "deg": 248,
            "clouds": 0


,
        {
            "dt": 1487761200,
            "temp": {
                "day": 281.76,
                "min": 273.21,
                "max": 281.76,
                "night": 278.88,
                "eve": 279.22,
                "morn": 273.21
```

```
                },
                "pressure": 945.21,
                "humidity": 0,
                "weather": [
                        {
                                "id": 500,
                                "main": "Rain",
                                "description": "light rain",
                                "icon": "10d"
                        }
                ],
                "speed": 2.98,
                "deg": 272,
                "clouds": 65,
                "rain": 1.48
        }
    ]
}
```


ตัวอย่างข้อมูล ก.26 API 11.4. Ultraviolet index


```
{
        "lat": 37.75,
        "lon": -122.37,
        "date_iso": "2017-06-26T12:00:00Z",
        "date": 1498478400,
        "value": 10.06
}
```


ตัวอย่างข้อมูล ก.27 API 11.5. Weather stations


```
[
  {
    "type": "m",
    "date": 1480508280,
    "station_id": "583d7bbbebb3ff000123f1ce",
    "temp": {
      "max": 22.7,
      "min": 21.7,
```

```
        "average": 22.03,
        "weight": 3
      },
      "humidity": {
        "average": 76.33,
        "weight": 3
      },
      "wind": {
        "deg": 160,
        "speed": 3.3
      },
      "pressure": {
        "min": 1031,
        "max": 1033,
        "average": 1031.67,
        "weight": 3
      },
      "precipitation": {}
    },
    {
      "type": "m",
      "date": 1480599660,
      "station_id": "583d7bbbebb3ff000123f1ce",
      "temp": {
        "max": 21.7,
        "min": 21.7,
        "average": 21.7,
        "weight": 1
      },
      "humidity": {
        "average": 77,
        "weight": 1
      },
      "wind": {},
      "pressure": {
        "min": 1031,
        "max": 1031,
        "average": 1031,
        "weight": 1
      },
      "precipitation": {}
    }
```

```
]
```


ตัวอย่างข้อมูล ก.28 API 11.6. Weather Alerts


```
[
  {
    "_id":"585268b72516b90010058184",
    "__v":0,
    "alerts":{

    },
    "area":[
      {
        "type":"Point",
        "_id":"585268b72516b90010058185",
        "coordinates":[
          53,
          37
        ]
      }
    ],
    "conditions":[
      {
        "name":"temp",
        "expression":"$gt",
        "amount":299,
        "_id":"585268b72516b90010058186"
      }
    ],
    "time_period":{
      "end":{
        "expression":"after",
        "amount":432000000
      },
      "start":{
        "expression":"after",
        "amount":132000000
      }
    }

]
```

ตัวอย่างข้อมูล ก.29 API 11.7. Carbon Monoxide


```
{
        "time": "2017-01-26T01:04:15Z",
        "location": {
                "latitude": 0,
                "longitude": 9.9905
        },
        "data": [
                {
                        "precision": -4.999999987376214e-7,
                        "pressure": 1000,
                        "value": 9.30106835994593e-8
                },
                {
                        "precision": -4.999999987376214e-7,
                        "pressure": 681.2920532226562,
                        "value": 1.0675700679030342e-7
                }

        ]
}
```


ตัวอย่างข้อมูล ก.30 API 12.1. Air quality


```
    {
     "forecasts": [
      {
       "period": "PT24H",
       "validFrom": "2017-05-10T01:00:00+01:00",
       "validUntil": "2017-05-11T01:00:00+01:00",
       "forecastLocation": [ -0.1248, 51.5081 ],
       "airPollutionIndex": 3,
       "publishedAt": "2017-05-10T07:00:00+01:00"
      },
      {
       "period": "PT24H",
```

```
        "validFrom": "2017-05-11T01:00:00+01:00",
        "validUntil": "2017-05-12T01:00:00+01:00",
        "forecastLocation": [ -0.1248, 51.5081 ],
        "airPollutionIndex": 5,
        "publishedAt": "2017-05-10T07:00:00+01:00"
      }

    ]
  }
```


ตัวอย่างข้อมูล ก.31 API 12.2. Pollen forecast


```
{
    "forecasts": [
      {
        "period": "PT24H",
        "validFrom": "2017-05-10T01:00:00+01:00",
        "validUntil": "2017-05-11T01:00:00+01:00",
        "pollenForecastRegion": "SE",
        "pollenForecastRegionName": "London & South East England",
        "pollenIndex": "Moderate",
        "issuedAt": "2017-05-10T06:45:57+01:00"
      }
    ]
  }
```


ตัวอย่างข้อมูล ก.32 API 13.1. Link definitions


```
[

  {

    "EndLatitude": 1.1,

    "EndLongitude": 2.1,

    "StartLatitude": 3.1,
```

```
  "StartLongitude": 4.1,
  "ID": "sample string 1",
  "DirectionOfTravel": "sample string 2",
  "RoadwayName": "sample string 3"
 },
 {
  "EndLatitude": 1.1,
  "EndLongitude": 2.1,
  "StartLatitude": 3.1,
  "StartLongitude": 4.1,
  "ID": "sample string 1",
  "DirectionOfTravel": "sample string 2",
  "RoadwayName": "sample string 3"
 }
]
```


ตัวอย่างข้อมูล ก.33 API 13.2. Events


```
[
 {
  "LastUpdated": "sample string 1",
  "Latitude": 2.1,
  "Longitude": 3.1,
```

```
  "PlannedEndDate": "sample string 4",

  "Reported": "sample string 5",

  "StartDate": "sample string 6",

  "ID": "sample string 1",

  "RegionName": "sample string 2",

  "CountyName": "sample string 3",

  "Severity": "sample string 4",

  "RoadwayName": "sample string 5",

  "DirectionOfTravel": "sample string 6",

  "Description": "sample string 7",

  "Location": "sample string 8",

  "LanesAffected": "sample string 9",

  "LanesStatus": "sample string 10",

  "LcsEntries": [

    {},

    {}

  ],

  "NavteqLinkId": "sample string 11",

  "PrimaryLocation": "sample string 12",

  "SecondaryLocation": "sample string 13",

  "FirstArticleCity": "sample string 14",
```

```
    "SecondCity": "sample string 15",
    "EventType": "sample string 16",
    "EventSubType": "sample string 17",
    "MapEncodedPolyline": "sample string 18"
  }
]
```


ตัวอย่างข้อมูล ก.34 API 13.3. Cameras


```
[
  {
    "Latitude": 1.1,
    "Longitude": 2.1,
    "ID": "sample string 1",
    "Name": "sample string 2",
    "DirectionOfTravel": "sample string 3",
    "RoadwayName": "sample string 4",
    "Url": "sample string 5",
    "Disabled": true,
    "Blocked": true
  },
  {
```

```
    "Latitude": 1.1,
    "Longitude": 2.1,
    "ID": "sample string 1",
    "Name": "sample string 2",
    "DirectionOfTravel": "sample string 3",
    "RoadwayName": "sample string 4",
    "Url": "sample string 5",
    "Disabled": true,
    "Blocked": true
  }
]
```


ตัวอย่างข้อมูล ก.35 API 13.4. Message Signs


```
[
  {
    "Latitude": 1.1,
    "Longitude": 2.1,
    "ID": "sample string 1",
    "Name": "sample string 2",
    "Roadway": "sample string 3",
    "DirectionOfTravel": "sample string 4",
    "Messages": [
```

```
    "sample string 1",
    "sample string 2"
   ]
  },
  {
   "Latitude": 1.1,
   "Longitude": 2.1,
   "ID": "sample string 1",
   "Name": "sample string 2",
   "Roadway": "sample string 3",
   "DirectionOfTravel": "sample string 4",
   "Messages": [
    "sample string 1",
    "sample string 2"
   ]
  }
]
```


ตัวอย่างข้อมูล ก.36 API 13.5. Alerts


```
[
 {
```

```
  "Id": "sample string 1",
  "Message": "sample string 2",
  "Notes": "sample string 3",
  "AreaNames": [
   "sample string 1",
   "sample string 2"
  ]
 },
 {
  "Id": "sample string 1",
  "Message": "sample string 2",
  "Notes": "sample string 3",
  "AreaNames": [
   "sample string 1",
   "sample string 2"
  ]
 }
]
```


ตัวอย่างข้อมูล ก.37 API 13.6. Winter Road


```
{
 "Condition": "sample string 1",
```

```
  "AreaName": "sample string 2",

  "LocationDescription": "sample string 3",

  "RoadwayName": "sample string 4"

}
```


ตัวอย่างข้อมูล ก.38 API 14.1. Places detail


```
{
 "request_time": "2016-04-28T14:50:44+01:00",
 "source": "NaPTAN",
 "acknowledgements": "Contains DfT NaPTAN bus stops data, Contains DfT NaPTAN tube station data",
 "member": [
  {
   "type": "bus_stop",
   "name": "Wembley High Road (W-bound)",
   "description": "Wembley",
   "latitude": 51.55351,
   "longitude": -0.2918,
   "accuracy": 20,
   "atcocode": "490014378M",
   "distance": 880
  },
  {
   "type": "bus_stop",
   "name": "Wembley Central (W-bound)",
   "description": "Wembley",
   "latitude": 51.55212,
   "longitude": -0.29763,
   "accuracy": 20,
   "atcocode": "490000256H",
   "distance": 1311
  }
 ]
}
```

ตัวอย่างข้อมูล ก.39 API 14.2. Tweet


```
[
  {
   "_id": "54e0bd6a1f43211ac9003ca5",
   "anew": {
    "valence": {
     "score": null,
     "n_matched_words": 0
    },
    "valence_negation": {
     "score": null,
     "n_matched_words": 0
    }
   },
   "coordinates": {
    "coordinates": [
     -0.163341,
     51.522707
    ]
   },
   "created_at": null,
   "emoticon_sentiment": {
    "happy": false,
    "sad": false
   },
   "mode": "train",
   "offensive": false,
   "retweet_count": 0,
   "stored_at": "2015-02-15T15:38:18Z",
   "text": " are trains going from Euston to Birmingham today (Sunday 15 feb please)",
   "updated_at": null,
   "user": {
    "screen_name": "afloralstory",
    "followers_count": 470,
    "profile_image_url":
"http:\/\/pbs.twimg.com\/profile_images\/2857076405\/acbbd32de9c881d7f697ad080820c766_normal.png"
   }
  }

 }
]
```

ตัวอย่างข้อมูล ก.40 API 14.3. Journey plan


```
{
 "request_time": "2016-08-02T09:07:21+01:00",
 "routes": [
  {
   "mode": "cycle",
   "from_point_name": "Carver Road",
   "to_point_name": "Eversholt Street, A4200",
   "duration": "00:50:38",
   "distance": 11291,
   "distance_desc": "7.0 miles",
   "instructions": [
    {
     "text": "Head NW onto Short un-named link",
     "distance": "43",
     "duration": "00:00:07",
     "distance_desc": "50 yards"
    },
    {
     "text": "Turn left onto Carver Road",
     "distance": "37",
     "duration": "00:00:07",
     "distance_desc": "40 yards"
    }


   ],
   "coordinates": [
    [
     -0.100251,
     51.453222
    ],
    [
     -0.104208,
     51.4665
    ]

   ]
  }
```

```
 ],
 "source": "cyclestreets API: http:\/\/www.cyclestreets.net\/API",
 "acknowledgements": "Routing Engine: cyclestreets API. Route data from OpenStreetMap"
}
```


ตัวอย่างข้อมูล ก.41 API 14.4. Station box


```
{
 "minlon": -0.10958,
 "minlat": 51.51988,
 "maxlon": -0.10475,
 "maxlat": 51.5231,
 "searchlon": -0.107165,
 "searchlat": 51.52149,
 "page": 1,
 "rpp": 25,
 "total": 1,
 "request_time": "2015-02-19T09:58:32+00:00",
 "stations": [
  {
   "station_code": "FAR",
   "atcocode": "9400ZZLUFCN",
   "name": "Farringdon",
   "mode": "tube",
   "longitude": -0.10516,
   "latitude": 51.52043,
   "lines": [
    "circle",
    "hammersmith",
    "metropolitan"
   ],
   "distance": 293
  }
 ]
}
```


ตัวอย่างข้อมูล ก.42 API 15.1. Location from IP


```
{
"country_code":"US",
"country_name":"United States",
"region_name":"California",
```

```
"city_name":"Mountain View",
"latitude":"37.405992",
"longitude":"-122.078515",
"zip_code":"94043",
"time_zone":"-08:00",
"isp":"Google Inc.",
"domain":"google.com",
"net_speed":"T1",
"idd_code":"1",
"area_code":"650",
"weather_station_code":"USCA0746",
"weather_station_name":"Mountain View",
"mcc":"-",
"mnc":"-",
"mobile_brand":"-",
"elevation":"31",
"usage_type":"SES"
}
```


ตัวอย่างข้อมูล ก.43 API 16.1. Alert


```
{
 "id": "Alert:1",
 "type": "Alert",
 "category": "traffic",
 "subCategory": "trafficJam",
 "severity": "high",
 "location": {
  "type": "Point",
   "coordinates": [
    -3.712247222222222,
    40.423852777777775
   ]
 },
 "dateObserved": "2017-01-02T09:25:55.00Z",
 "description": "The road is completely blocked for 3kms",
 "alertSource": "https://account.lab.fiware.org/users/8"
}
```

ตัวอย่างข้อมูล ก.44 API 16.2. 311 Service


```
{
 "id": "o311:servicetype-guadalajara-sidewalks",
 "type": "Open311:ServiceType",
 "dateCreated": "2007-01-01",
 "jurisdiction_id": "www.smartguadalajara.com",
 "open311:type": "realtime",
 "service_code": 234,
 "service_name": "Aceras",
 "description": "When a sidewalk is broken or dirty allows citizens to request a fix",
 "Keywords": "street,sidewalk, cleaning, repair",
 "group": "street",
 "attributes": [
  {
   "variable": true,
   "code": "ISSUE_TYPE",
   "datatype": "singlevaluelist",
   "required": true,
   "datatype_description": null,
   "order": 1,
   "description": "What is the identified problem at the sidewalk?",
   "values": [
    {
     "Key": 123,
     "name": "Bump"
    },
    {
     "Key": 124,
     "name":"Dirty"
    }
   ]
  }
 ]
}
```


ตัวอย่างข้อมูล ก.45 API 16.3. 311 Request


```
{
 "id": "service-request:638344",
 "type": "Open311:ServiceRequest",
```

```
 "service_request_id": 638344,
 "status": "closed",
 "status_notes": "Duplicate request.",
 "service_name": "Aceras",
 "service_code": 234,
 "description": "Acera en mal estado con bordillo partido en dos",
 "agency_responsible": "Ayuntamiento de Ciudad",
 "requested_datetime": "2010-04-14T06:37:38-08:00",
 "updated_datetime": "2010-04-14T06:37:38-08:00",
 "expected_datetime": "2010-04-15T06:37:38-08:00",
 "address_string": "Calle San Juan Bautista, 2",
 "attributes": {
   "ISSUE_TYPE": ["Bordillo"]
 },
 "location": {
  "type": "Point",
  "coordinates": [ -3.164485591715449, 40.62785133667262 ]
 },
 "media_url":"http://exaple.org/media/638344.jpg"
}
```


ตัวอย่างข้อมูล ก.46 API 16.4. Device


```
{
 "id": "device-9845A",
 "type": "Device",
 "category": ["sensor"],
 "controlledProperty": ["fillingLevel","temperature"],
 "controlledAsset": ["wastecontainer-Osuna-100"],
 "ipAddress": ["192.14.56.78"],
 "mcc": "214",
 "mnc": "07",
 "batteryLevel": 0.75,
 "serialNumber": "9845A",
 "refDeviceModel": "myDevice-wastecontainer-sensor-345",
 "value": "l=0.22;t=21.2",
 "deviceState": "ok",
 "dateFirstUsed": "2014-09-11T11:00:00Z",
 "owner": ["http://person.org/leon"]
}
```

ตัวอย่างข้อมูล ก.47 API 16.5. Air quality


```
{
 "id": "Madrid-AmbientObserved-28079004-2016-03-15T11:00:00",
 "type": "AirQualityObserved",
 "address": {
  "addressCountry": "ES",
  "addressLocality": "Madrid",
  "streetAddress": "Plaza de España"
 },
 "dateObserved": "2016-03-15T11:00:00/2016-03-15T12:00:00",
 "location": {
  "type": "Point",
  "coordinates": [-3.712247222222222, 40.423852777777775]
 },
 "source": "http://datos.madrid.es",
 "precipitation": 0,
 "relativeHumidity": 0.54,
 "temperature": 12.2,
 "windDirection": 186,
 "windSpeed": 0.64,
 "airQualityLevel": "moderate",
 "reliability": 0.9,
 "CO": 500,
 "NO": 45,
 "NO2": 69,
 "NOx": 139,
 "SO2": 11,
 "CO_Level": "good",
 "NO_Level": "moderate",
 "refPointOfInterest": "28079004-Pza. de España"
}
```


ตัวอย่างข้อมูล ก.48 API 16.6. Water quality


```
{
  "id": "waterqualityobserved:Sevilla:D1",
  "type": "WaterQualityObserved",
```

```
  "dateObserved": "2017-01-31T06:45:00Z",
  "measurand": [
   "NO3, 0.01, M1, Concentration of Nitrates"
  ],
  "location": {
   "type": "Point",
   "coordinates": [ -5.993307, 37.362882 ]
  },
  "temperature" : 24.4,
  "conductivity": 0.005,
  "pH": 7.4,
  "NO3": 0.01,
}
```


ตัวอย่างข้อมูล ก.49 API 16.7. Noise level


```
{
    "id": "Vitoria-NoiseLevelObserved-2016-12-28T11:00:00_2016-12-28T12:00:00",
    "type": "NoiseLevelObserved",
    "location": {
      "type": "Point",
      "coordinates": [-2.6980, 42.8491]
    },
    "dateObserved": "2016-12-28T11:00:00/2016-12-28T12:00:00",
    "measurand": [
      "LAeq | 67.8 | A-weighted, equivalent, sound level",
      "LAmax | 94.5 | A-weighted, maximum, sound level",
    ],
    "LAeq": 67.8,
    "LAmax": 94.5,
    "sonometerClass": "2"
  }
```


ตัวอย่างข้อมูล ก.50 API 16.8. Off street parking


```
{
 "id": "porto-ParkingLot-23889",
 "type": "OffStreetParking",
 "name": "Parque de estacionamento Trindade",
```

```
 "category": ["underground", "public", "feeCharged", "mediumTerm", "barrierAccess"],
 "chargeType": ["temporaryPrice"],
 "requiredPermit": [],
 "layout": ["multiLevel"],
 "maximumParkingDuration": "PT8H",
 "location": {
  "coordinates": [-8.60961198807, 41.150691773],
  "type": "Point"
 },
 "allowedVehicleType": ["car"],
 "totalSpotNumber": 414,
 "availableSpotNumber": 132,
 "address": {
  "streetAddress": "Rua de Fernandes Tomás",
  "addressLocality": "Porto",
  "addressCountry": "Portugal"
 },
 "description": "Municipal car park located near the Trindade metro station and the Town Hall",
 "dateModified": "2016-06-02T09:25:55.00Z"
}
```


ตัวอย่างข้อมูล ก.51 API 16.9. Parking spot


```
{
 "id": "santander:daoiz_velarde_1_5:3",
 "type": "ParkingSpot",
 "name": "A-13",
 "location": {
  "type": "Point",
  "coordinates": [-3.80356167695194, 43.46296641666926]
 },
 "status": "free",
 "category: ["onstreet"],
 "refParkingSite": "santander:daoiz_velarde_1_5"
}
```


ตัวอย่างข้อมูล ก.52 API 16.10. Garden


```
{
```

```
    "id": "Santander-Garden-Piquio",
    "type": "Garden",
    "name": "Jardines de Piquio",
    "description": "Jardines de Piquio. Zona El Sardinero",
    "location": {
      "type": "Point",
      "coordinates": [-3.7836974, 43.4741091]
    },
    "address": {
      "streetAddress": "Avenida Castañeda",
      "addressLocality": "Santander",
      "postalCode": "39005"
    },
    "openingHours": "Mo-Su",
    "style": "french",
    "category": ["public"],
    "areaServed": "El Sardinero",
    "dateLastWatering": "2017-03-31T:08:00",
    "refRecord": ["Santander-Garden-Piquio-Record-1"]
  }
```


ตัวอย่างข้อมูล ก.53 API 16.11. Greenspace


```
{
    "id": "Santander-Garden-Piquio-Record-1",
    "type": "GreenspaceRecord",
    "location": {
      "type": "Point",
      "coordinates": [-3.7836974, 43.4741091]
    },
    "temperature": 17,
    "relativeHumidity": 0.87,
    "soilTemperature": 13,
    "refGreenspace": "Santander-Garden-Piquio"
  }
```


ตัวอย่างข้อมูล ก.54 API 16.12. Flower bed


```
{
```

```
 "id": "FlowerBed-345",
 "type": "FlowerBed",
 "category": ["urbanTreeSpot"],
 "dateLastWatering": "2017-03-31T08:00",
 "address": {
  "streetAddress": "Paseo Zorrilla, 122",
  "adressLocality": "Valladolid",
  "addressCountry": "Spain"
 },
 "soilTemperature": 17,
 "soilMoistureVwc": 0.85
}
```


ตัวอย่างข้อมูล ก.55 API 16.13. Street light


```
{
 "id": "streetlight:guadalajara:4567",
 "type": "Streetlight",
 "location": {
  "type": "Point",
  "coordinates": [ -3.164485591715449, 40.62785133667262 ]
 },
 "areaServed": "Roundabouts city entrance",
 "status": "ok",
 "refStreetlightGroup": "streetlightgroup:G345",
 "refStreetlightModel": "streetlightmodel:STEEL_Tubular_10m",
 "circuit": "C-456-A467",
 "lanternHeight": 10,
 "locationCategory" : "centralIsland",
 "powerState": "off",
 "controllingMethod": "individual",
 "dateLastLampChange": "2016-07-08T08:02:21.753Z"
}
```


ตัวอย่างข้อมูล ก.56 API 16.14. Waste container


```
{
 "id": "wastecontainer:Fleming:12a",
```

```
  "type": "WasteContainer",
  "refWasteContainerModel": "wastecontainermodel:c1",
  "refWasteContainerIsle":  "wastecontainerisle:Fleming:12",
  "serialNumber": "ab56kjl",
  "location": {
    "type": "Point",
    "coordinates": [  -3.164485591715449, 40.62785133667262 ]
  },
  "fillingLevel" : 0.4,
  "dateLastEmptying": "2016-06-21T15:05:59.408Z",
  "nextActuationDeadline": "2016-06-28T15:05:59.408Z",
  "status": "ok",
  "category": ["underground"],
  "refDevice": ["device-Fleming:12a:1"]
}
```


ตัวอย่างข้อมูล ก.57 API 17.1. Noaa stations


```
{
 "metadata": {
  "resultset": {
    "offset": 1,
    "count": 129764,
    "limit": 25
  }
 },
 "results": [
  {
    "elevation": 139,
    "mindate": "1948-01-01",
    "maxdate": "2014-01-01",
    "latitude": 31.5702,
    "name": "ABBEVILLE, AL US",
    "datacoverage": 0.8813,
    "id": "COOP:010008",
    "elevationUnit": "METERS",
    "longitude": -85.2482
  }

 ]
}
```

ตัวอย่างข้อมูล ก.58 API 18.1. Forecast


```
{
 "query": {
  "count": 1,
  "created": "2018-03-13T18:20:40Z",
  "lang": "en-US",
  "results": {
   "channel": {
    "units": {
     "distance": "mi",
     "pressure": "in",
     "speed": "mph",
     "temperature": "F"
    },
    "title": "Yahoo! Weather - Nome, AK, US",
    "link":      "http://us.rd.yahoo.com/dailynews/rss/weather/Country__Country/*https://weather.yahoo.com/country/state/city-
2460286/",
    "description": "Yahoo! Weather for Nome, AK, US",
    "language": "en-us",
    "lastBuildDate": "Tue, 13 Mar 2018 10:20 AM AKDT",
    "ttl": "60",
    "location": {
     "city": "Nome",
     "country": "United States",
     "region": " AK"
    },
    "wind": {
     "chill": "-6",
     "direction": "23",
     "speed": "14"
    },
    "atmosphere": {
     "humidity": "80",
     "pressure": "1004.0",
     "rising": "0",
     "visibility": "16.1"
    },
    "astronomy": {
     "sunrise": "9:27 am",
```

```
  "sunset": "8:56 pm"
 },
 "image": {
  "title": "Yahoo! Weather",
  "width": "142",
  "height": "18",
  "link": "http://weather.yahoo.com",
  "url": "http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif"
 },
 "item": {
  "title": "Conditions for Nome, AK, US at 09:00 AM AKDT",
  "lat": "64.499474",
  "long": "-165.405792",
  "link": "http://us.rd.yahoo.com/dailynews/rss/weather/Country__Country/*https://weather.yahoo.com/country/state/city-2460286/",
  "pubDate": "Tue, 13 Mar 2018 09:00 AM AKDT",
  "condition": {
   "code": "26",
   "date": "Tue, 13 Mar 2018 09:00 AM AKDT",
   "temp": "8",
   "text": "Cloudy"
  },
  "forecast": [
   {
    "code": "26",
    "date": "13 Mar 2018",
    "day": "Tue",
    "high": "15",
    "low": "6",
    "text": "Cloudy"
   },
   {
    "code": "32",
    "date": "14 Mar 2018",
    "day": "Wed",
    "high": "7",
    "low": "1",
    "text": "Sunny"
   }
   ],
  "description": "<![CDATA[<img src=\"http://l.yimg.com/a/i/us/we/52/26.gif\"/>\n<BR />\n<b>Current Conditions:</b>\n<BR />Cloudy\n<BR />\n<BR />\n<b>Forecast:</b>\n<BR /> Tue - Cloudy. High: 15Low: 6\n<BR /> Wed
```

```
- Sunny. High: 7Low: 1\n<BR /> Thu - Sunny. High: 13Low: 0\n<BR /> Fri - Sunny. High: 10Low: -1\n<BR /> Sat - Breezy. High: 22Low: 1\n<BR />\n<BR />\n<a href=\"http://us.rd.yahoo.com/dailynews/rss/weather/Country__Country/*https://weather.yahoo.com/country/state/city-2460286/\">Full Forecast at Yahoo! Weather</a>\n<BR />\n<BR />\n<BR />\n]]>",
    "guid": {
     "isPermaLink": "false"
    }
   }
  }
 }
}
```


ตัวอย่างข้อมูล ก.59 API 18.2. Wind


```
{
 "query": {
  "count": 1,
  "created": "2018-03-13T18:42:08Z",
  "lang": "en-US",
  "results": {
   "channel": {
    "wind": {
     "chill": "19",
     "direction": "5",
     "speed": "32"
    }
   }
  }
 }
}
```


ตัวอย่างข้อมูล ก.60 API 18.3. Conditions


```
{
 "query": {
  "count": 1,
  "created": "2018-03-13T18:47:01Z",
  "lang": "en-US",
```

```
  "results": {
   "channel": {
    "item": {
     "condition": {
      "code": "26",
      "date": "Tue, 13 Mar 2018 11:00 AM PDT",
      "temp": "62",
      "text": "Cloudy"
     }
    }
   }
  }
 }
}
```


ตัวอย่างข้อมูล ก.61 API 18.4. Phase condition


```
{
 "query": {
  "count": 1,
  "created": "2018-03-13T18:51:28Z",
  "lang": "en-US",
  "results": {
   "channel": {
    "item": {
     "condition": {
      "text": "Sunny"
     }
    }
   }
  }
 }
}
```


ตัวอย่างข้อมูล ก.62 API 18.5. Sunset time


```
{
 "query": {
  "count": 1,
```

```
 "created": "2018-03-13T18:55:03Z",
 "lang": "en-US",
 "results": {
  "channel": {
   "astronomy": {
    "sunset": "6:34 pm"
   }
  }
 }
}
```


ตัวอย่างข้อมูล ก.63 API 18.6. Weather


```
{
 "query": {
  "count": 0,
  "created": "2018-03-13T18:59:17Z",
  "lang": "en-US",
  "results": null
 }
}
```


ตัวอย่างข้อมูล ก.64 API 19.1. Air quality


```
{
   "datetime": "2016-06-21T11:36:00",
   "country_name": "United States",
   "breezometer_aqi": 68,
   "breezometer_color": "#90D32D",
   "breezometer_description": "Fair Air Quality",
   "country_aqi": 39,
   "country_aqi_prefix": "",
   "country_color": "#00E400",
   "country_description": "Good air quality",
   "data_valid": true,
   "Key_valid": true,
   "random_recommendations": {
      "children": "Have fun with the kids outside, but stay alert for our notifications",
```

```
    "sport": "You can exercise outdoors - but make sure to stay alert to our notifications",
    "health": "There is no real danger for people with health sensitivities. Just keep an eye out for changes in air quality for the next few hours",
    "inside": "The amount of pollutants in the air is noticeable, but still there is no danger to your health - It is recommended to continue monitoring changes in the coming hours",
    "outside": "It's still OK to go out and enjoy a stroll, just pay attention for changes in air quality"
  },
  "dominant_pollutant_canonical_name": "pm2.5",
  "dominant_pollutant_description": "Fine particulate matter (<2.5μm)",
  "dominant_pollutant_text": {
    "main": "The dominant pollutant is fine particulate matter (PM2.5).",
    "effects": "Particles enter the lungs and cause local and systemic inflammation in the respiratory system & heart, thus cause cardiovascular and respiratory diseases such as asthma and bronchitis.",
    "causes": "Main sources are fuel burning processes in industry, transportation and indoor heating."
  },
  "pollutants": {
    "co": {
      "pollutant_description": "Carbon monoxide",
      "units": "ppb",
      "concentration": 101.24
    },
    "no2": {
      "pollutant_description": "Nitrogen dioxide",
      "units": "ppb",
      "concentration": 9.49
    },
    "o3": {
      "pollutant_description": "Ozone",
      "units": "ppb",
      "concentration": 31.4
    },
    "pm10": {
      "pollutant_description": "Inhalable particulate matter (<10μm)",
      "units": "ug/m3",
      "concentration": 3
    },
    "pm25": {
      "pollutant_description": "Fine particulate matter (<2.5μm)",
      "units": "ug/m3",
      "concentration": 5.22
    },
    "so2": {
```

```
        "pollutant_description": "Sulfur dioxide",
        "units": "ppb",
        "concentration": null
      }
    }
}
```


ตัวอย่างข้อมูล ก.65 API 20.1. Car park


```
{
 "API_info": {
  "status": "healthy"
 },
 "items": [
  {
   "timestamp": "2018-03-13T20:34:32.498Z",
   "carpark_data": [
    {
     "total_lots": "string",
     "lot_type": "string",
     "lots_available": "string"
    }
   ]
  }
 ]
}
```


ตัวอย่างข้อมูล ก.66 API 20.2. Ultra-violet


```
{
 "API_info": {
  "status": "healthy"
 },
 "items": [
  {
   "update_timestamp": "2018-03-13T20:35:15.084Z",
   "timestamp": "2018-03-13T20:35:15.084Z",
   "index": [
    {
```

```
        "timestamp": "2018-03-13T20:35:15.084Z",
        "value": 0
       }
     ]
    }
  ]
}
```


ตัวอย่างข้อมูล ก.67 API 20.3. Weather


```
{
 "API_info": {
  "status": "healthy"
 },
 "metadata": {
  "stations": [
   {
    "id": "string",
    "device_id": "string",
    "name": "string",
    "location": {
     "longitude": 0,
     "latitude": 0
    }
   }
  ],
  "reading_type": "string",
  "reading_unit": "string"
 },
 "items": [
  {
   "timestamp": "2018-03-13T20:35:13.381Z",
   "readings": [
    {
     "station_id": "string",
     "value": 0
    }
   ]
  }
 ]
}
```

ตัวอย่างข้อมูล ก.68 API 20.4. Traffic


```
{
 "API_info": {
  "status": "healthy"
 },
 "items": [
  {
   "timestamp": "2018-03-13T20:50:31.945Z",
   "cameras": [
    {
     "timestamp": "2018-03-13T20:50:31.945Z",
     "camera_id": 0,
     "image_id": 0,
     "image": "string",
     "image_metadata": {
      "height": 0,
      "width": 0,
      "md5": "string"
     }
    }
   ]
  }
 ]
}
```


ตัวอย่างข้อมูล ก.69 API 20.5. PM 2.5


```
{
 "API_info": {
  "status": "healthy"
 },
 "region_metadata": [
  {
   "name": "string",
   "label_location": {
    "longitude": 0,
    "latitude": 0
```

```
    }
   }
  ],
  "items": [
   {
    "update_timestamp": "2018-03-13T20:50:15.772Z",
    "timestamp": "2018-03-13T20:50:15.772Z",
    "readings": {
     "pm25_one_hourly": {
      "national": 0,
      "north": 0,
      "south": 0,
      "east": 0,
      "west": 0,
      "central": 0
     }
    }
   }
  ]
}
```


ตัวอย่างข้อมูล ก.70 API 20.6. Taxi


```
{
 "type": "string",
 "crs": {},
 "features": [
  {
   "geometry": {
    "type": "MultiPoint",
    "coordinates": [
     [
      0
     ]
    ]
   },
   "properties": {
    "timestamp": "2018-03-13T20:50:22.438Z",
    "taxi_count": 0
   }
  }
```

```
 ]
}
```


ตัวอย่างข้อมูล ก.71 API 20.7. Forecast 24 hour


```
{
 "API_info": {
  "status": "healthy"
 },
 "area_metadata": [
  {
   "name": "string",
   "label_location": {
    "longitude": 0,
    "latitude": 0
   }
  }
 ],
 "items": [
  {
   "update_timestamp": "2018-03-13T21:04:39.019Z",
   "timestamp": "2018-03-13T21:04:39.019Z",
   "valid_period": {
    "start": "2018-03-13T21:04:39.020Z",
    "end": "2018-03-13T21:04:39.020Z"
   },
   "general": {
    "forecast": "string",
    "relative_humidity": {
     "low": 0,
     "high": 0
    },
    "temperature": {
     "low": 0,
     "high": 0
    },
    "wind": {
     "speed": {
      "low": 0,
      "high": 0
     },
```

```
        "direction": "string"
      }
     }
    }
  ]
}
```


ตัวอย่างข้อมูล ก.72 API 20.8. Pollutant


```
{
 "API_info": {
  "status": "healthy"
 },
 "region_metadata": [
  {
   "name": "string",
   "label_location": {
    "longitude": 0,
    "latitude": 0
   }
  }
 ],
 "items": [
  {
   "update_timestamp": "2018-03-13T20:50:27.101Z",
   "timestamp": "2018-03-13T20:50:27.101Z",
   "readings": {
    "psi_twenty_four_hourly": {
     "national": 0,
     "north": 0,
     "south": 0,
     "east": 0,
     "west": 0,
     "central": 0
    },
    "psi_three_hourly": {
     "national": 0,
     "north": 0,
     "south": 0,
     "east": 0,
     "west": 0,
```

```
  "central": 0
 },
 "pm10_sub_index": {
  "national": 0,
  "north": 0,
  "south": 0,
  "east": 0,
  "west": 0,
  "central": 0
 },
 "pm25_sub_index": {
  "national": 0,
  "north": 0,
  "south": 0,
  "east": 0,
  "west": 0,
  "central": 0
 },
 "so2_sub_index": {
  "national": 0,
  "north": 0,
  "south": 0,
  "east": 0,
  "west": 0,
  "central": 0
 },
 "o3_sub_index": {
  "national": 0,
  "north": 0,
  "south": 0,
  "east": 0,
  "west": 0,
  "central": 0
 },
 "co_sub_index": {
  "national": 0,
  "north": 0,
  "south": 0,
  "east": 0,
  "west": 0,
  "central": 0
 },
```

```
"pm10_twenty_four_hourly": {
 "national": 0,
 "north": 0,
 "south": 0,
 "east": 0,
 "west": 0,
 "central": 0
},
"pm25_twenty_four_hourly": {
 "national": 0,
 "north": 0,
 "south": 0,
 "east": 0,
 "west": 0,
 "central": 0
},
"no2_one_hour_max": {
 "national": 0,
 "north": 0,
 "south": 0,
 "east": 0,
 "west": 0,
 "central": 0
},
"so2_twenty_four_hourly": {
 "national": 0,
 "north": 0,
 "south": 0,
 "east": 0,
 "west": 0,
 "central": 0
},
"co_eight_hour_max": {
 "national": 0,
 "north": 0,
 "south": 0,
 "east": 0,
 "west": 0,
 "central": 0
},
"o3_eight_hour_max": {
 "national": 0,
```

```
        "north": 0,
        "south": 0,
        "east": 0,
        "west": 0,
        "central": 0
      }
    }
  }
 ]
}
```


ตัวอย่างข้อมูล ก.73 API 21.1. 311 requests


```
[
  {
    ":@computed_region_92fq_4b7q": "42",
    ":@computed_region_f5dn_yrer": "50",
    ":@computed_region_yeji_bk3q": "5",
    "address_type": "ADDRESS",
    "agency": "NYPD",
    "agency_name": "New York City Police Department",
    "borough": "BRONX",
    "city": "BRONX",
    "community_board": "04 BRONX",
    "complaint_type": "Noise - Residential",
    "created_date": "2018-03-12T01:15:07.000",
    "cross_street_1": "EAST 171 STREET",
    "cross_street_2": "EAST 172 STREET",
    "descriptor": "Banging/Pounding",
    "due_date": "2018-03-12T09:15:07.000",
    "facility_type": "Precinct",
    "incident_address": "1487 TELLER AVENUE",
    "incident_zip": "10457",
    "latitude": "40.83944257304853",
    "location": {
      "type": "Point",
      "coordinates": [
        -73.908944972112,
        40.839442573049
      ]
    },
```

```
    "location_type": "Residential Building/House",
    "longitude": "-73.9089449721118",
    "park_borough": "BRONX",
    "park_facility_name": "Unspecified",
    "resolution_description": "Your complaint has been forwarded to the New York Police Department for a non-emergency response. 311 will have additional information in 8 hours. Please note your service request number for future reference.",
    "school_address": "Unspecified",
    "school_city": "Unspecified",
    "school_code": "Unspecified",
    "school_name": "Unspecified",
    "school_not_found": "N",
    "school_number": "Unspecified",
    "school_phone_number": "Unspecified",
    "school_region": "Unspecified",
    "school_state": "Unspecified",
    "school_zip": "Unspecified",
    "status": "Open",
    "street_name": "TELLER AVENUE",
    "unique_Key": "38665203",
    "x_coordinate_state_plane": "1009445",
    "y_coordinate_state_plane": "245122"
  }


]
```


ตัวอย่างข้อมูล ก.74 API 21.2. Alert


```
[
  {
    "alert_date": "2016-11-01T00:00:00.000",
    "dataset_name": "ACRIS Real Property References",
    "dataset_url": "https://data.cityofnewyork.us/City-Government/ACRIS-Real-Property-References/qpy9-5hqe",
    "details": "Several records in the dataset are corrupt.",
    "resolution": "Datasets were recreated and reloaded.",
    "resolution_date": "2017-02-16T00:00:00.000",
    "status": "Resolved"
  },
  {
    "alert_date": "2016-11-01T00:00:00.000",
    "dataset_name": "ACRIS Personal Property Parties",
```

```
    "dataset_url": "https://data.cityofnewyork.us/City-Government/ACRIS-Personal-Property-Parties/gq4f-jwz9",
    "details": "Several records in the dataset are corrupt.",
    "resolution": "Datasets were recreated and reloaded.",
    "resolution_date": "2017-02-16T00:00:00.000",
    "status": "Resolved"
  }


]
```


ตัวอย่างข้อมูล ก.75 API 21.3. Recycling Bins


```
[
  {
    "address": "E 227 St/Bronx River Pkway",
    "borough": "Bronx",
    "latitude": "40.890848989",
    "longitude": "-73.864223918",
    "park_site_name": "227th St. Plgd",
    "site_type": "Subproperty"
  },
  {
    "address": "Allerton Ave & Moshulu Pkway",
    "borough": "Bronx",
    "latitude": "40.8488907878",
    "longitude": "-73.8771283938",
    "park_site_name": "Allerton Ballfields",
    "site_type": "Subproperty"
  }
  ]
```


ตัวอย่างข้อมูล ก.76 API 21.4. Farmers markets


```
[
  {
    "facilityaddinfo": "Spend $5 in EBT and get a $2 Health Buck coupon.<br><br><b>Program Website:</b> <a
href="http://www.grownyc.org/youthmarket" target="_blank">http://www.grownyc.org/youthmarket</a>",
    "facilitycity": "Bronx",
    "facilityname": "Riverdale Youthmarket",
```

```
    "facilitystate": "NY",
    "facilitystreetname": "256th St & Mosholu Ave",
    "facilityzipcode": "10471",
    "latitude": "41",
    "longitude": "-74"
  },
  {
    "facilityaddinfo": "Spend $5 in EBT and get a $2 Health Buck coupon.<br><br><b>Program Website:</b> <a
href="http://www.grownyc.org." target="_blank">http://www.grownyc.org.</a>",
    "facilitycity": "Queens",
    "facilityname": "Astoria Greenmarket",
    "facilitystate": "NY",
    "facilitystreetname": "30th Dr",
    "facilityzipcode": "11106",
    "latitude": "41",
    "longitude": "-74"
  }
  ]
```


ตัวอย่างข้อมูล ก.77 API 22.1. Locations


```
{
        "stations": [
                {
                        "id": "8500010",
                        "name": "Basel SBB",
                        "score": null,
                        "coordinate": {
                                "type": "WGS84",
                                "x": 47.547403,
                                "y": 7.589564
                        },
                        "distance": null
                },
                {
                        "id": "8588780",
                        "name": "Basel, Schifflände",
                        "score": null,
                        "coordinate": {
                                "type": "WGS84",
                                "x": 47.559197,
```

```
                    "y": 7.587166
                },
                "distance": null
            }

        ]
}
```


ตัวอย่างข้อมูล ก.78 API 22.2. Connections


```
{
    "connections": [
        {
            "from": {
                "station": {
                    "id": "8501120",
                    "name": "Lausanne",
                    "score": null,
                    "coordinate": {
                        "type": "WGS84",
                        "x": 46.516786,
                        "y": 6.629087
                    },
                    "distance": null
                },
                "arrival": null,
                "arrivalTimestamp": null,
                "departure": "2018-03-16T13:51:00+0100",
                "departureTimestamp": 1521204660,
                "delay": 0,
                "platform": "9",
                "prognosis": {
                    "platform": null,
                    "arrival": null,
                    "departure": "2018-03-16T13:51:00+0100",
                    "capacity1st": null,
                    "capacity2nd": null
                },
                "realtimeAvailability": null,
                "location": {
                    "id": "8501120",
```

```
                "name": "Lausanne",
                "score": null,
                "coordinate": {
                    "type": "WGS84",
                    "x": 46.516786,
                    "y": 6.629087
                },
                "distance": null
            }
        },
        "to": {
            "station": {
                "id": "8501008",
                "name": "Genève",
                "score": null,
                "coordinate": {
                    "type": "WGS84",
                    "x": 46.210219,
                    "y": 6.142435
                },
                "distance": null
            },
            "arrival": "2018-03-16T14:40:00+0100",
            "arrivalTimestamp": 1521207600,
            "departure": null,
            "departureTimestamp": null,
            "delay": null,
            "platform": "9",
            "prognosis": {
                "platform": null,
                "arrival": null,
                "departure": null,
                "capacity1st": null,
                "capacity2nd": null
            },
            "realtimeAvailability": null,
            "location": {
                "id": "8501008",
                "name": "Genève",
                "score": null,
                "coordinate": {
                    "type": "WGS84",
```

```
                                        "x": 46.210219,
                                        "y": 6.142435
                                    },
                                    "distance": null
                                }
                            },
                            "duration": "00d00:49:00",
                            "transfers": 0,
                            "service": null,
                            "products": [
                                "RE"
                            ],
                            "capacity1st": null,
                            "capacity2nd": null,
                            "sections": [
                                {
                                    "journey": {
                                        "name": "RE 3218",
                                        "category": "RE",
                                        "subcategory": null,
                                        "categoryCode": null,
                                        "number": "RE 3218",
                                        "operator": "SBB",
                                        "to": "Genève",
                                        "passList": [
                                            {
                                                "station": {
                                                    "id": "8501120",
                                                    "name": "Lausanne",
                                                    "score": null,
                                                    "coordinate": {
                                                        "type":
"WGS84",
                                                        "x":
46.516786,
                                                        "y": 6.629087
                                                    },
                                                    "distance": null
                                                },
                                                "arrival": null,
                                                "arrivalTimestamp": null,
```

```
                                                                "departure": "2018-03-
16T13:51:00+0100",
                                                                "departureTimestamp":
1521204660,
                                                                "delay": 0,
                                                                "platform": "9",
                                                                "prognosis": {
                                                                        "platform": null,
                                                                        "arrival": null,
                                                                        "departure": "2018-03-
16T13:51:00+0100",
                                                                        "capacity1st": null,
                                                                        "capacity2nd": null
                                                                },
                                                                "realtimeAvailability": null,
                                                                "location": {
                                                                        "id": "8501120",
                                                                        "name": "Lausanne",
                                                                        "score": null,
                                                                        "coordinate": {
                                                                                "type":
"WGS84",
                                                                                "x":
46.516786,
                                                                                "y": 6.629087
                                                                        },
                                                                        "distance": null
                                                                }
                                                        }
                                                }
                                        }
                                ]
                        }

                        ]
                }
}
```

ตัวอย่างข้อมูล ก.79 API 22.3. Station board


```
{
    "station": {
        "id": "8502113",
        "name": "Aarau",
        "score": null,
        "coordinate": {
            "type": "WGS84",
            "x": 47.391352,
            "y": 8.051257
        },
        "distance": null
    },
    "stationboard": [
        {   "stop": {
                "station": {
                    "id": "8502113",
                    "name": "Aarau",
                    "score": null,
                    "coordinate": {
                        "type": "WGS84",
                        "x": 47.391352,
                        "y": 8.051257
                    },
                    "distance": null
                },
                "arrival": null,
                "arrivalTimestamp": null,
                "departure": "2018-03-17T09:13:00+0100",
                "departureTimestamp": 1521274380,
                "delay": null,
                "platform": "5",
                "prognosis": {
                    "platform": null,
                    "arrival": null,
                    "departure": null,
                    "capacity1st": null,
                    "capacity2nd": null
                },
                "realtimeAvailability": null,
                "location": {
```

```
                "id": "8500218",
                "name": null,
                "score": null,
                "coordinate": {
                    "type": "WGS84",
                    "x": null,
                    "y": null
                },
                "distance": null
            }
        },
        "name": "RE 4762",
        "category": "RE",
        "subcategory": null,
        "categoryCode": null,
        "number": "RE 4762",
        "operator": "SBB",
        "to": "Olten",
        "passList": [
            {
                "station": {
                    "id": "8500218",
                    "name": null,
                    "score": null,
                    "coordinate": {
                        "type": "WGS84",
                        "x": null,
                        "y": null
                    },
                    "distance": null
                },
                "arrival": null,
                "arrivalTimestamp": null,
                "departure": "2018-03-17T09:13:00+0100",
                "departureTimestamp": 1521274380,
                "delay": null,
                "platform": "5",
                "prognosis": {
                    "platform": null,
                    "arrival": null,
                    "departure": null,
                    "capacity1st": null,
```

```
                                        "capacity2nd": null
                                    },
                                    "realtimeAvailability": null,
                                    "location": {
                                        "id": "8500218",
                                        "name": null,
                                        "score": null,
                                        "coordinate": {
                                            "type": "WGS84",
                                            "x": null,
                                            "y": null
                                        },
                                        "distance": null
                                    }
                                },
                            ]
                            "capacity1st": null,
                            "capacity2nd": null
                        }
                ]
}
```


ตัวอย่างข้อมูล ก.80 API 23.1. Crimes


```
[
  {
    "arrest": false,
    "beat": "1113",
    "block": "003XX N CICERO AVE",
    "case_number": "JB160804",
    "community_area": "25",
    "date": "2018-02-21T23:55:00.000",
    "description": "DOMESTIC BATTERY SIMPLE",
    "district": "011",
    "domestic": false,
    "fbi_code": "08B",
    "id": "11237136",
    "iucr": "0486",
    "latitude": "41.885550891",
    "location": {
      "type": "Point",
```

```
      "coordinates": [
        -87.745459209,
        41.885550891
      ]
    },
    "location_description": "SIDEWALK",
    "longitude": "-87.745459209",
    "primary_type": "BATTERY",
    "updated_on": "2018-02-28T16:16:24.000",
    "ward": "28",
    "x_coordinate": "1144328",
    "y_coordinate": "1901433",
    "year": "2018"
  },
  {
    "arrest": false,
    "beat": "2422",
    "block": "077XX N HASKINS AVE",
    "case_number": "JB160852",
    "community_area": "1",
    "date": "2018-02-21T23:50:00.000",
    "description": "TO CITY OF CHICAGO PROPERTY",
    "district": "024",
    "domestic": false,
    "fbi_code": "14",
    "id": "11237171",
    "iucr": "1345",
    "latitude": "42.022062685",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.67484904,
        42.022062685
      ]
    },
    "location_description": "CTA GARAGE / OTHER PROPERTY",
    "longitude": "-87.67484904",
    "primary_type": "CRIMINAL DAMAGE",
    "updated_on": "2018-02-28T16:16:24.000",
    "ward": "49",
    "x_coordinate": "1163173",
    "y_coordinate": "1951318",
```

```
    "year": "2018"
  },
  {
    ":@computed_region_43wa_7qmu": "35",
    ":@computed_region_6mkv_f3dw": "21546",
    ":@computed_region_awaf_s7ux": "9",
    ":@computed_region_bdys_3d7i": "490",
    ":@computed_region_d3ds_rm58": "240",
    ":@computed_region_d9mm_jgwp": "19",
    ":@computed_region_rpca_8um6": "61",
    ":@computed_region_vrxf_vc4k": "44",
    "arrest": false,
    "beat": "0412",
    "block": "088XX S DANTE AVE",
    "case_number": "JB159571",
    "community_area": "48",
    "date": "2018-02-20T23:56:00.000",
    "description": "DOMESTIC BATTERY SIMPLE",
    "district": "004",
    "domestic": true,
    "fbi_code": "08B",
    "id": "11236113",
    "iucr": "0486",
    "latitude": "41.73419593",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.589288462,
        41.73419593
      ]
    },
    "location_description": "APARTMENT",
    "longitude": "-87.589288462",
    "primary_type": "BATTERY",
    "updated_on": "2018-02-27T15:59:52.000",
    "ward": "8",
    "x_coordinate": "1187331",
    "y_coordinate": "1846610",
    "year": "2018"
  },
]
```

ตัวอย่างข้อมูล ก.81 API 23.2. Police stations


```
[
  {
    ":@computed_region_43wa_7qmu": "9",
    ":@computed_region_6mkv_f3dw": "21194",
    ":@computed_region_awaf_s7ux": "12",
    ":@computed_region_bdys_3d7i": "330",
    ":@computed_region_vrxf_vc4k": "34",
    "address": "1718 S State St",
    "city": "Chicago",
    "district": "1",
    "district_name": "Central",
    "fax": "312-745-3694",
    "latitude": "41.85837259",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.627356,
        41.858373
      ]
    },
    "longitude": "-87.62735617",
    "phone": "312-745-4290",
    "state": "IL",
    "tty": "312-745-3693",
    "website": "http://home.chicagopolice.org/community/districts/1st-district-central/",
    "x_coordinate": "1176569.052",
    "y_coordinate": "1891771.704",
    "zip": "60616"
  }
]
```


ตัวอย่างข้อมูล ก.82 API 23.3. CDPH Notification


```
[
  {
    ":@computed_region_43wa_7qmu": "47",
    ":@computed_region_6mkv_f3dw": "21202",
    ":@computed_region_awaf_s7ux": "47",
    ":@computed_region_bdys_3d7i": "602",
```

```
    ":@computed_region_vrxf_vc4k": "49",
    "address": {
      "type": "Point",
      "coordinates": [
        -87.535292,
        41.703874
      ]
    },
    "address_address": "10522 S EWING AVE",
    "address_city": "CHICAGO",
    "address_state": "IL",
    "comment": "CITY OF CHICAGO  DEMOLITION AND REMOVAL OF TWO STORY BRICK BUILDING.",
    "data_source": "DEPT. OF PUBLIC HEALTH",
    "direction": "S",
    "end_date": "2018-03-23T00:00:00.000",
    "notification_from": "KATIE MCCORMICK",
    "notification_type": "NOI DEMO/RENOVATION",
    "start_date": "2018-03-13T00:00:00.000",
    "street_name": "EWING",
    "street_number": "10522",
    "street_type": "AVE"
  },
  {
    ":@computed_region_43wa_7qmu": "23",
    ":@computed_region_6mkv_f3dw": "14920",
    ":@computed_region_awaf_s7ux": "48",
    ":@computed_region_bdys_3d7i": "96",
    ":@computed_region_vrxf_vc4k": "29",
    "address": {
      "type": "Point",
      "coordinates": [
        -87.661404,
        41.860921
      ]
    },
    "address_address": "1528 S LOOMIS ST",
    "address_city": "CHICAGO",
    "address_state": "IL",
    "comment": "1528  S. LOOMIS  REMOVAL OF PIPE AND FLUE INSULATION",
    "data_source": "DEPT. OF PUBLIC HEALTH",
    "direction": "S",
    "end_date": "2018-03-27T00:00:00.000",
```

```
    "notification_from": "CLADINE PARRA",
    "notification_type": "NOI NESHAP",
    "start_date": "2018-03-13T00:00:00.000",
    "street_name": "LOOMIS",
    "street_number": "1528",
    "street_type": "ST"
  }
]
```


ตัวอย่างข้อมูล ก.83 API 23.4. Energy


```
[
  {
    ":@computed_region_43wa_7qmu": "46",
    ":@computed_region_6mkv_f3dw": "21184",
    ":@computed_region_awaf_s7ux": "48",
    ":@computed_region_bdys_3d7i": "90",
    ":@computed_region_vrxf_vc4k": "29",
    "address": "1900 West Van Buren Street",
    "community_area": "NEAR WEST SIDE",
    "data_year": "2015",
    "electricity_use_kbtu": "25403152",
    "ghg_intensity_kg_co2e_sq_ft": "12.3",
    "gross_floor_area_buildings_sq_ft": "541575",
    "id": "250009",
    "latitude": "41.876251000000003",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.674127,
        41.876251
      ]
    },
    "longitude": "-87.674126999999999",
    "natural_gas_use_kbtu": "29511031",
    "of_buildings": "1",
    "primary_property_type": "College/University",
    "property_name": "Malcolm X College",
    "site_eui_kbtu_sq_ft": "101.4",
    "source_eui_kbtu_sq_ft": "204.5",
    "total_ghg_emissions_metric_tons_co2e": "6671",
```

```
    "weather_normalized_site_eui_kbtu_sq_ft": "105.5",
    "weather_normalized_source_eui_kbtu_sq_ft": "208.1",
    "year_built": "1971",
    "zip_code": "60612"
  }
  ]
```


ตัวอย่างข้อมูล ก.84 API 23.5. Green Infrastructure


```
[
  {
    "data_stream_id": "33264",
    "latitude": "41.90715",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.653996,
        41.90715
      ]
    },
    "longitude": "-87.653996",
    "measurement_id": "2518808831999999999",
    "measurement_medium": "NA",
    "measurement_period_type": "Instantaneous",
    "measurement_time": "2018-03-18T19:00:00.000",
    "measurement_title": "UI Labs Bioswale NWS Probability of Precipitation",
    "measurement_type": "TimeWindowBoundary",
    "measurement_value": "0.0",
    "record_id": "332642518808831999999999",
    "resource_id": "65153",
    "units": "universal coordinated time",
    "units_abbreviation": "UTC"
  },
  {
    "data_stream_id": "33265",
    "latitude": "41.90715",
    "location": {
      "type": "Point",
      "coordinates": [
        -87.653996,
        41.90715
```

```
      ]
    },
    "longitude": "-87.653996",
    "measurement_id": "2518809047999999999",
    "measurement_medium": "NA",
    "measurement_period_type": "Instantaneous",
    "measurement_time": "2018-03-18T13:00:00.000",
    "measurement_title": "UI Labs Bioswale NWS Quantitative Precipitation Forecast",
    "measurement_type": "TimeWindowBoundary",
    "measurement_value": "0.0",
    "record_id": "332652518809047999999999",
    "resource_id": "65148",
    "units": "universal coordinated time",
    "units_abbreviation": "UTC"
  }
]
```


ตัวอย่างข้อมูล ก.85 API 24.1. Harbor Water quality


```
[
  {
    "ctd_conductivity_temperature_depth_profiler_top_dissolved_oxygen_mg_l": "4.91",
    "current_direction_current_direction": "0",
    "current_speed_knot": "HW SLK",
    "lat": "40.813667",
    "light_trans_transparency_top_sample": "61.56",
    "long": "-73.871333",
    "oxidation_reduction_potential_mv_top_sample": "193.099",
    "percentage_o2_saturation_top_sample": "68.4",
    "sample_date": "2015-08-17T00:00:00.000",
    "sample_time": "1:07 PM",
    "sampling_location": "BR5",
    "sea_state": "1",
    "secchi_depth_ft": "2",
    "site_actual_depth_ft": "13",
    "top_active_chlorophyll_a_g_l": "8.96",
    "top_ammonium_mg_l": "0.03",
    "top_conductivity_s_m": "3.98",
    "top_dissolved_organic_carbon_mg_l": "2.39",
    "top_enterococci_bacteria_cells_100ml": "3",
    "top_fecal_coliform_bacteria_cells_100ml": "10",
```

```
    "top_fluorometer_mg_m3": "3.9804",
    "top_nitrate_nitrite_mg_l": "0.2",
    "top_ortho_phosphorus_mg_l": "0.124",
    "top_ph": "7.32",
    "top_salinity_psu": "25.58",
    "top_sample_depth_ft": "3",
    "top_sample_par_ue_s_m2": "3791.3",
    "top_sample_temperature_c": "24.68",
    "top_sigma_t_kg_m3": "16.33",
    "top_silica_mg_l": "2.61",
    "top_total_kjeldhal_nitrogen_mg_l": "0.435",
    "top_total_suspended_solid_mg_l": "16",
    "total_phosphorus_mg_l": "0.188",
    "type": "Permanent",
    "weather_condition_dry_or_wet": "Dry",
    "wind_direction_wind_direction": "S",
    "wind_speed_mph": "5",
    "winkler_method_top_dissolved_oxygen_mg_l": "5.17"
  }
]
```


ตัวอย่างข้อมูล ก.86 API 24.2. Drinking Water quality


```
[
  {
    "coliform_quanti_tray_mpn_100ml": "<1 ",
    "e_coli_quanti_tray_mpn_100ml": "<1 ",
    "fluoride_mg_l": "0.79",
    "location": "SS - Shaft 7 of City Tunnel No. 1 - W/S Sedgwick Ave OPP W 167th St (Tun 1)",
    "residual_free_chlorine_mg_l": "0.58",
    "sample_class": "Operational",
    "sample_date": "2015-01-01T00:00:00.000",
    "sample_site": " 1S07",
    "sample_time": "12:19",
    "turbidity_ntu": "0.96"
  }


]
```

ตัวอย่างข้อมูล ก.87 API 24.3. LCR Monitoring


```
[
  {
    "ortho_phosphate_mg_l": "1.99",
    "ph_unit": "7.24",
    "sample_date": "2014-07-01T00:00:00.000",
    "sample_site": " 1S04",
    "specific_conductance_s_cm": "85",
    "temperature_f": "62"
  },
  {
    "ortho_phosphate_mg_l": "1.99",
    "ph_unit": "7.34",
    "sample_date": "2014-07-02T00:00:00.000",
    "sample_site": " 1S04",
    "specific_conductance_s_cm": "86",
    "temperature_f": "63"
  }
]
```


ตัวอย่างข้อมูล ก.88 API 25.1. Garbage


```
[
  {
    "date": "2012-10-01T00:00:00.000",
    "day_of_week": "MON",
    "month": "OCT",
    "sanitation_district": "1",
    "tons": "4.86"
  }


]
```


ตัวอย่างข้อมูล ก.89 API 25.2. Police call


```
[
  {
    "beat": "543",
```

```
    "block": "2500",
    "date_time": "2018-03-17T08:21:00.000",
    "division": "Northwest",
    "incident_number": "18-0467078",
    "location": "Hondo Ave",
    "nature_of_call": "6X - Major Dist (Violence)",
    "priority": "2",
    "reporting_area": "3114",
    "status": "At Scene",
    "unit_number": "D560"
  }
  ]
```


ตัวอย่างข้อมูล ก.90 API 25.3. Issued construction


```
[
  {
    ":@computed_region_8spj_utxs": "4",
    ":@computed_region_a3it_2a2z": "3642",
    ":@computed_region_q9nd_rr82": "9",
    ":@computed_region_qwte_z96m": "3439",
    "applieddate": "2007-03-26T00:00:00.000",
    "calendar_year_issued": "2007",
    "condominium": "No",
    "contractor_address2": "406  Spring Grove",
    "contractor_city": "Liberty Hill",
    "contractor_company_name": "A-1 Plumbing",
    "contractor_full_name": "James Spencer",
    "contractor_phone": "5122947368",
    "contractor_trade": "Plumbing Contractor",
    "contractor_zip": "78642",
    "council_district": "4",
    "day_issued": "FRIDAY",
    "description": "addition and whole house remodel See permit 2008055186",
    "expiresdate": "2008-01-08T00:00:00.000",
    "fiscal_year_issued": "2007",
    "housing_units": "1",
    "issue_date": "2007-05-11T00:00:00.000",
    "issue_method": "Permit Center",
    "issued_in_last_30_days": "No",
    "jurisdiction": "AUSTIN FULL PURPOSE",
```

```
    "latitude": "30.35908015",
    "legal_description": "LOT 7 BLK I GEORGIAN ACRES",
    "link": "https://www.austintexas.gov/devreview/b_showpublicpermitfolderdetails.jsp?FolderRSN=10021720",
    "location": {
      "type": "Point",
      "coordinates": [
        -97.69149203,
        30.35908015
      ]
    },
    "longitude": "-97.69149203",
    "masterpermitnum": "10021717",
    "number_of_floors": "1",
    "original_address1": "314 HACKBERRY LN",
    "original_city": "AUSTIN",
    "original_state": "TX",
    "original_zip": "78753",
    "permit_class": "R- 434 Addition & Alterations",
    "permit_class_mapped": "Residential",
    "permit_location": "314 HACKBERRY LN",
    "permit_number": "2007-112847 PP",
    "permit_type_desc": "Plumbing Permit",
    "permittype": "PP",
    "plumbing_valuation_remodel": "1200",
    "project_id": "10021720",
    "status_current": "VOID",
    "statusdate": "2008-07-21T11:21:43.000",
    "tcad_id": "0239190211",
    "total_existing_bldg_sqft": "960",
    "total_new_add_sqft": "270",
    "work_class": "Addition and Remodel"
  }
]
```


ตัวอย่างข้อมูล ก.91 API 25.4. Water quality


```
[
  {
    ":@computed_region_a3it_2a2z": "3249",
    "ali_site": "F",
    "eii_site": "F",
```

```
    "first_sampledate": "1994-01-06T00:00:00.000",
    "location": {
      "type": "Point",
      "coordinates": [
        -97.876470494075,
        30.299199198562
      ]
    },
    "most_recent_sampledate": "2001-05-01T00:00:00.000",
    "number_of_data_points": "138",
    "number_of_visits": "11",
    "sample_site_no": "100",
    "site_name": "Hebbingston Hollow Trib @ Barton Creek (HHT)",
    "site_type": "Stream",
    "watershed": "Barton Creek",
    "watershed_reach": "BAR3"
  }
]
```


ตัวอย่างข้อมูล ก.92 API 25.5. Recycle drop off


```
[
  {
    ":@computed_region_8spj_utxs": "2",
    ":@computed_region_a3it_2a2z": "3258",
    "address": {
      "type": "Point",
      "coordinates": [
        -97.72995848699969,
        30.21197822400046
      ]
    },
    "address_address": "4906 Burleson Rd",
    "address_city": "Austin",
    "address_state": "Texas",
    "address_zip": "78744",
    "batteries": "Yes",
    "business_name": "Tex-Con Oil Company",
    "latitude": "30.21197822400046",
    "longitude": "-97.72995848699969",
    "oil": "Yes",
```

```
    "oil_filter": "Yes",
    "phone": "512-444-1941",
    "status": "1",
    "zip_code": "78744",
    "zone": "Southeast"
  }
  ]
```


ตัวอย่างข้อมูล ก.93 API 26.1. Forecast


```
{
 "geometry": {
  "coordinates": [
   0,
   0
  ],
  "type": "Point"
 },
 "properties": {
  "code": "high",
  "value": 0
 },
 "type": "Feature"
}
```


ตัวอย่างข้อมูล ก.94 API 27.1. Rain Gauges


```
{
        "type": "FeatureCollection",
        "features": [
                {
                        "type": "Feature",
                        "properties": {
                                "OBJECTID": 1,
                                "SYSTEM": "STORM",
                                "OWNER": "PWD OWNED",
                                "X_COORD": -75.22519478,
                                "Y_COORD": 39.8895714,
                                "FACILITYNAME": "RG_23 - SWWPC Plant",
```

```
                "FACILITYID": "40d3a5b2-3ac4-4af9-8866-29e5216e766d"
            },
            "geometry": {
                "type": "Point",
                "coordinates": [
                    -75.2251965630538,
                    39.88957944864953
                ]
            }
        },
        {
            "type": "Feature",
            "properties": {
                "OBJECTID": 2,
                "SYSTEM": "STORM",
                "OWNER": "PWD OWNED",
                "X_COORD": -75.14584654,
                "Y_COORD": 39.90230134,
                "FACILITYNAME": "RG_12 - SEWPC Plant",
                "FACILITYID": "15b2f05f-6f3b-48e9-938b-c34422a86d25"
            },
            "geometry": {
                "type": "Point",
                "coordinates": [
                    -75.14584829846066,
                    39.902309397743615
                ]
            }
        }


    ]
}
```


ตัวอย่างข้อมูล ก.95 API 28.1. Carbon sensor


```
[
    {
```

```
                "id": "5aa4b0520274e294487b23f4",
                "type": "CarbonMonoxideSensor",
                "status": "moderate",
                "label": "moderate",
                "representStatus": "Good",
                "telemetries": [
                        {
                                "name": "co",
                                "value": 5.404,
                                "timestamp": 1520742480
                        }
                ],
                "timestamp": 1520742481
        }
]
```


ตัวอย่างข้อมูล ก.96 API 28.2. Water sensor


```
[
        {
                "id": "5aa4b04b0274e287497b23cf",
                "type": "WaterLevelSensor",
                "status": "safe",
                "label": "safe",
                "representStatus": "Safe",
                "telemetries": [
                        {
                                "name": "level",
                                "value": 2211,
                                "timestamp": 1520742420
                        }
                ],
                "timestamp": 1520742473
        },
        {
                "id": "5aa4af1d0274e294487b23df",
                "type": "WaterLevelSensor",
                "status": "safe",
                "label": "safe",
                "representStatus": "Safe",
                "telemetries": [
```

```
                {
                    "name": "level",
                    "value": 2222,
                    "timestamp": 1520742120
                }
            ],
            "timestamp": 1520742173
        }
        ]
```


ตัวอย่างข้อมูล ก.97 API 29.1. Sentimental


```
{
 "predicted": {
  "id": "39e3105f-cf3f-4115-904e-6ff77a44dd1a",
  "predicted": [
   {
    "acceptance": "0.00",
    "anger": "0.00",
    "anticipation": "99.00",
    "disgust": "0.00",
    "fear": "0.00",
    "joy": "0.00",
    "latitude": 13.746118609,
    "longitude": 100.533124438,
    "max_emo": 7,
    "name": "Siam Center (\u0e2a\u0e22\u0e32\u0e21\u0e40\u0e0b\u0e47\u0e19\u0e40\u0e15\u0e2d\u0e23\u0e4c)",
    "name_id": "siamcenter",
    "place_id": "63b96234-80f9-4135-9526-43ec98f3681d",
    "predicted_texts": [
     "\ud83c\udf51 \u0e27\u0e31\u0e19\u0e19\u0e35\u0e49 @tarrtarr_thailand
\u0e23\u0e49\u0e32\u0e19\u0e17\u0e32\u0e23\u0e4c\u0e15\u0e0a\u0e37\u0e48\u0e2d\u0e14\u0e31\u0e07\u0e02\u0e2d\u0e0e
07\u0e40\u0e01\u0e32\u0e2b\u0e25\u0e35\u0e44\u0e14\u0e49
\u0e40\u0e1b\u0e34\u0e14\u0e2a\u0e32\u0e02\u0e32\u0e17\u0e35\u0e48 2
\u0e41\u0e25\u0e49\u0e27\u0e19\u0e30\u0e04\u0e30 \u0e17\u0e35\u0e48 Siam Center \u0e0a\u0e31\u0e49\u0e19 G
\u0e2a\u0e32\u0e02\u0e32\u0e19\u0e35\u0e49\u0e21\u0e35\u2026 https://t.co/PTaI5y9ObF",
     "\u0e21\u0e32\u0e23\u0e49\u0e32\u0e19 @tarrtarr_tart \ud83d\udc3b \u0e17\u0e35\u0e48\u0e0a\u0e31\u0e49\u0e19 G
\u0e2a\u0e22\u0e32\u0e21\u0e40\u0e0b\u0e47\u0e19\u0e40\u0e15\u0e2d\u0e23\u0e4c
\u0e40\u0e1b\u0e47\u0e19\u0e17\u0e32\u0e23\u0e4c\u0e15\u0e2a\u0e31\u0e0d\u0e0a\u0e32\u0e15\u0e34\u0e40\u0e01\u0e0e
32\u0e2b\u0e25\u0e35\u0e17\u0e35\u0e48\u0e19\u0e48\u0e32\u0e23\u0e31\u0e01\u0e01\u0e23\u0e30\u0e08\u0e38\u0e4b\u
0e21\u0e01\u0e23\u0e30\u0e08\u0e34\u0e4b\u0e21\u0e21\u0e32\u0e01
```

```
\u0e23\u0e32\u0e04\u0e32\u0e01\u0e47\u0e44\u0e21\u0e48\u0e44\u0e14\u0e49\u0e41\u0e23\u0e07\u0e21\u0e32\u0e01\u20
26 https://t.co/l8EOHmGyFS",
        "\u0e01\u0e27\u0e48\u0e32        30        \u0e41\u0e1a\u0e23\u0e19\u0e14\u0e4c        \u0e23\u0e27\u0e21        11
\u0e42\u0e0a\u0e27\u0e4c
\u0e17\u0e31\u0e49\u0e07\u0e41\u0e1a\u0e23\u0e19\u0e14\u0e4c\u0e44\u0e17\u0e22\u0e41\u0e25\u0e30\u0e2d\u0e34\u0e
19\u0e40\u0e15\u0e2d\u0e23\u0e4c\u0e0a\u0e31\u0e49\u0e19\u0e19\u0e33
\u0e1e\u0e23\u0e49\u0e2d\u0e21\u0e1c\u0e19\u0e36\u0e01\u0e01\u0e33\u0e25\u0e31\u0e07\u0e2a\u0e22\u0e32\u0e21      3
\u0e28\u0e39\u0e19\u0e22\u0e4c\u0e2f        \u0e2a\u0e22\u0e32\u0e21\u0e1e\u0e32\u0e23\u0e32\u0e01\u0e2d\u0e19
\u0e2a\u0e22\u0e32\u0e21\u0e40\u0e0b\u0e47\u0e19\u0e40\u0e15\u0e2d\u0e23\u0e4c        \u0e41\u0e25\u0e30\u2026
https://t.co/SCqhUGndrN"
      ],
      "sadness": "0.00",
      "surprise": "0.00",
      "zone": "downtown"
    }
    ]
  }
}
```