ข้อเสนอโครงงานวิศวกรรมคอมพิวเตอร์

วิชา 01076014 การเตรียมโครงงานวิศวกรรมคอมพิวเตอร์

ภาคเรียนที่ 2 ปีการศึกษา 2563

1. ชื่อหัวข้อโครงงาน (ไทย) การแนะนำการใช้ยาต้านจุลชีพในสัตว์โดยใช้เทคนิคการเรียนรู้ของเครื่อง
2. ชื่อหัวข้อโครงงาน (อังกฤษ) Antimicrobial Recommendation In Pets Using Machine Learning
3. Keyword 3 คำ Antimicrobial, Classification, Machine Learning
4. ประเภทโครงงาน (✓)

   ☐ 1. HW+SW ☐ 2. SW_Dev ✓ 3. Research

5. รายชื่อผู้ทำโครงงาน

   5.1. นาย/นางสาว รวิสรา หนูเพ็ง รหัส 61010888

   5.2. นาย/นางสาว วัชรินทร์ กัณหา รหัส 61010960

6. อาจารย์ที่ปรึกษา

   6.1. อาจารย์ที่ปรึกษาหลัก ผศ.ดรชุติเมษฎ์ ศรีนิลทา

   6.2. อาจารย์ที่ปรึกษาร่วม

1. ที่มาและความสำคัญของปัญหา (Motivation)

ยาต้านจุลชีพเป็นยาที่ใช้ในการรักษาโรคติดเชื้อจุลินทรีย์ โดยออกฤทธิ์ฆ่าหรือยับยั้งการเจริญเติบโตของจุลินทรีย์ แต่ปัจจุบันนี้ได้มีปัญหาการดื้อยาต้านจุลชีพเพิ่มมากขึ้นเรื่อย ๆ ซึ่งเป็นภัยคุกคามอย่างมากต่อวงการแพทย์ คนและสัตว์ที่มีภาวะติดเชื้อดื้อยาจะมีอัตราการป่วยและเสียชีวิตในระดับที่สูงกว่าการติดเชื้อปกติ ทำให้การรักษาอาจไม่ตอบสนองต่อยาต้านจุลชีพพื้นฐานที่มีอยู่ในโรงพยาบาล สาเหตุของการดื้อยามีหลายประการ เช่น การเลือกใช้ยาต้านจุลชีพที่ไม่เหมาะสมต่อเชื้อ การใช้ยาต้านจุลชีพที่มากเกินความจำเป็น เป็นต้น ทำให้จุลินทรีย์มีวิวัฒนาการของการดื้อยาเพิ่มมากขึ้นเรื่อย ๆ ดังนั้นจึงควรเลือกใช้ยาต้านจุลชีพให้ถูกต้องและเหมาะสมเพื่อลดโอกาสการดื้อยาของเชื้อจุลินทรีย์

การเลือกใช้ยาต้านจุลชีพที่เหมาะสมนั้น สัตวแพทย์จะพิจารณาจากผลการทดสอบความไวต่อยาต้านจุลชีพ (Result of Antimicrobial Susceptibility Test : AST) ซึ่งเป็นการตรวจหายาต้านจุลชีพที่มีความไวต่อการตอบสนองต่อเชื้อจุลินทรีย์ที่ก่อให้เกิดโรค และตรวจหายาต้านจุลชีพที่ดื้อต่อการรักษา เพื่อช่วยเป็นแนวทางให้สัตวแพทย์เลือกใช้ยาต้านจุลชีพที่ถูกต้องและเหมาะสมกับสัตว์ป่วย

ดังนั้นทางผู้จัดทำจึงได้เสนอเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เพื่อทำนายการใช้ยาต้านจุลชีพที่เหมาะสมตามคำแนะนำของสัตวแพทย์ โดยใช้ชุดข้อมูลผลการทดสอบความไวต่อยาต้านจุลชีพซึ่งเป็นรายงานผลการตรวจห้องปฏิบัติการจุลชีววิทยา เพื่อเป็นการช่วยเหลือสัตวแพทย์ในการเลือกใช้ยาต้านจุลชีพที่เหมาะสม และรวดเร็วมากยิ่งขึ้น เพื่อผลสัมฤทธิ์ด้านการรักษา และลดโอกาสการดื้อยาของเชื้อจุลินทรีย์

2. วัตถุประสงค์ (Objectives)

2.1 เพื่อศึกษาและวิเคราะห์ปัจจัยที่มีผลต่อการเลือกใช้ยาต้านจุลชีพของสัตวแพทย์จากผลการทดสอบความไวต่อยาต้านจุลชีพ

2.2 เพื่อสร้างโมเดลทำนายการใช้ยาต้านจุลชีพที่เหมาะสมตามคำแนะนำของสัตวแพทย์

2.3 เพื่อศึกษาวิธีการแก้ปัญหาชุดข้อมูลไม่สมดุล (Imbalanced Dataset)

2.4 เพื่อเปรียบเทียบประสิทธิภาพของโมเดลที่ได้จาก Decision Tree, SVM, Random Forest และ Gradient Boosting

3. ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

3.1 การเรียนรู้ของเครื่อง (Machine Learning)

เป็นการเรียนรู้อัลกอริทึมของคอมพิวเตอร์โดยมีการปรับปรุงผ่านประสบการณ์และข้อมูล ซึ่งเป็นสาขาหนึ่งของปัญญาประดิษฐ์ อัลกอริทึมการเรียนรู้ของเครื่องจะสร้างโมเดลจากข้อมูลตัวอย่าง เพื่อทำการคาดการณ์หรือตัดสินใจผลลัพธ์โดยไม่ใช้การโปรแกรมแบบทั่วไป ซึ่งการเรียนรู้ของเครื่องนั้นแบ่งออกเป็น 3 ประเภท [1]

3.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

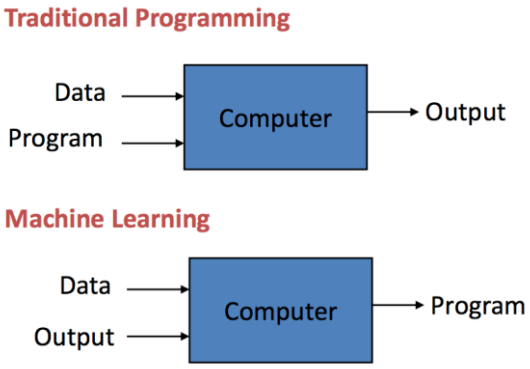
เป็นการสอนคอมพิวเตอร์โดยมีตัวอย่างข้อมูลขาเข้า (Input) และข้อมูลขาออก (Output) ที่ต้องการให้คอมพิวเตอร์ทำการสร้างฟังก์ชันที่เกี่ยวข้องระหว่างข้อมูลขาเข้าและข้อมูลขาออก ตัวอย่างการเรียนรู้แบบนี้ เช่น การถดถอย (Regression), ต้นไม้ตัดสินใจ (Decision Tree), ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) เป็นต้น [1]

3.1.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

เป็นการสอนคอมพิวเตอร์โดยไม่มีการกำหนดป้ายชื่อให้กับอัลกอริทึม คอมพิวเตอร์จะทำการเรียนรู้โครงสร้างของข้อมูลขาเข้า (Input) โดยการหารูปแบบของข้อมูลและทำการจัดกลุ่มของข้อมูลด้วยตัวเอง ตัวอย่างการเรียนรู้แบบนี้ เช่น K Mean, DBSCAN เป็นต้น [1]

3.1.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

เป็นการที่คอมพิวเตอร์โต้ตอบกับสภาพแวดล้อม ซึ่งจะต้องทำตามเป้าหมายโดยการลองผิดลองถูก ถ้าทำถูกจะได้รางวัล (Reward) ตัวอย่างการเรียนรู้แบบนี้ เช่น Markov decision process เป็นต้น [1]



รูปที่ 1 เปรียบเทียบ Traditional Programming กับ Machine Learning[1]

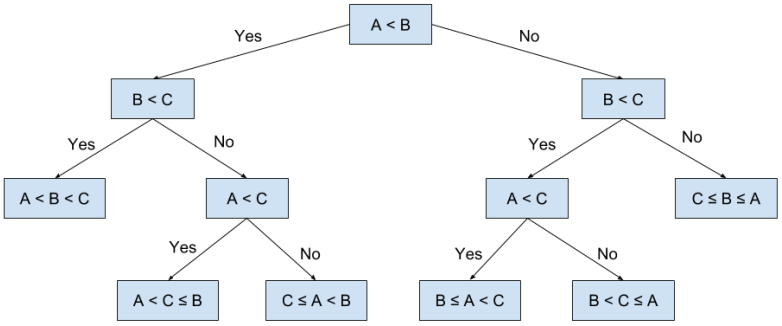
[1] ที่มาภาพ: https://i.pinimg.com/originals/3a/42/97/3a4297c9a348c65f6a37b58ab638ed3d.png

3.2 การจำแนกประเภท (Classification)

เป็นการจัดประเภทของข้อมูลใหม่ว่าอยู่ประเภทใดจากข้อมูลฝึกสอน (Training data) โดยที่ทราบอยู่แล้วว่ามีประเภทอะไรบ้าง ตัวอย่าง เช่น การระบุว่าอีเมลฉบับนี้เป็นสแปมหรือไม่ใช่สแปม ซึ่งการเรียนรู้ของเครื่องนั้น การจำแนกประเภทจะเป็นส่วนหนึ่งของการเรียนรู้แบบมีผู้สอน [2]

3.3 ต้นไม้ตัดสินใจ (Decision Tree)

เป็นเครื่องมือที่ช่วยสนับสนุนการตัดสินใจโดยใช้โมเดลแบบต้นไม้ในการแสดงทุกความเป็นไปได้ของการตัดสินใจรวมถึงโอกาสของเหตุการณ์ และอื่น ๆ โดยเป็นการแสดงอัลกอริทึมที่มีเฉพาะคำสั่งแบบมีเงื่อนไข [3]



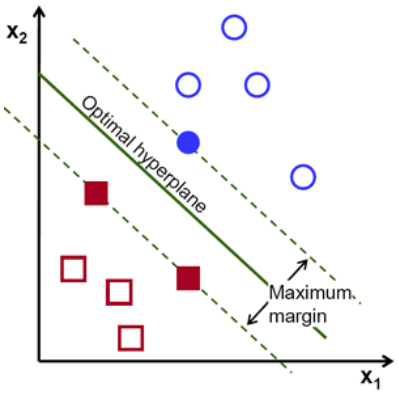
รูปที่ 2 ตัวอย่าง Decision Tree[2]

3.4 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine : SVM)

SVM เป็นการเรียนรู้แบบมีผู้สอนซึ่งเป็นอัลกอริทึมที่สามารถใช้ได้ทั้งการ จำแนกประเภทข้อมูลและการถดถอย แต่ส่วนใหญ่มักจะใช้กับปัญหาการจำแนกประเภทข้อมูล โดย SVM จะทำการพล็อตข้อมูลแต่ละรายการเป็นจุดในพื้นที่ n มิติ (n คือจำนวนฟีเจอร์) ซึ่งค่าของแต่ละฟีเจอร์จะเป็นค่าของพิกัดเฉพาะ จากนั้นจึงทำการจำแนกประเภทโดยการหาเส้นแบ่ง (Hyperplane) ที่จำแนกข้อมูลของทั้งสองคลาสได้อย่างชัดเจน นอกจากนี้ SVM ยังมีเทคนิคที่เรียกว่า “Kernel Trick” ซึ่ง SVM Kernel เป็นฟังก์ชันที่สามารถแปลงข้อมูลที่มีมิติที่ต่ำกว่าให้มีมิติที่สูงขึ้นได้ [4]

---

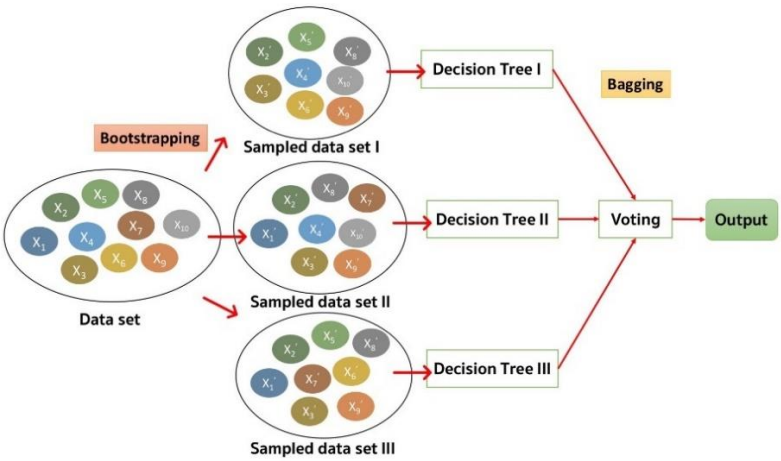
[2] ที่มาภาพ: https://elf11.github.io/images/decisionTree.png



รูปที่ 3 ตัวอย่าง Support Vector machine[3]

3.5 Random Forest

เป็นโมเดลที่ถูกพัฒนาขึ้นมาจาก Decision Tree หลักการคือการสร้างโมเดลจาก Decision Tree หลาย ๆ โมเดล โดยแต่ละโมเดลจะเลือกฟีเจอร์และส่วนของข้อมูลฝึกสอนที่ไม่เหมือนกัน แล้วคำนวณผลลัพธ์จากการโหวตของโมเดลว่าคลาสใดถูกเลือกมากที่สุด ทำให้มีประสิทธิภาพการทำงานสูงและมีความแม่นยำมากขึ้น [5]



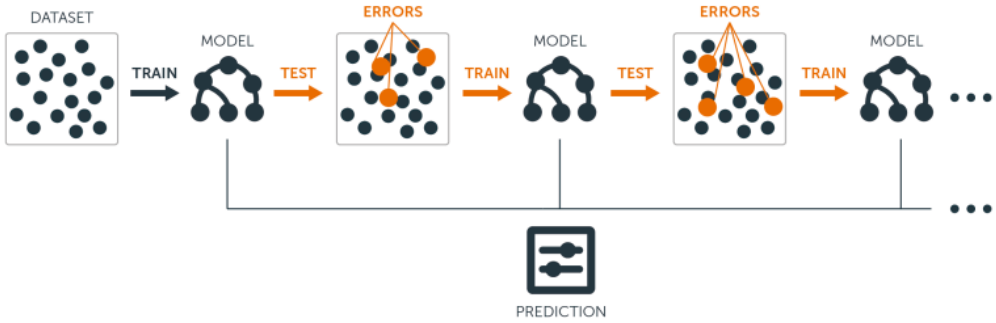
รูปที่ 4 หลักการของ Random Forest[4]

[3] ที่มาภาพ: https://ichi.pro/assets/images/max/640/1*LiTpHWC00p62TqXdckEXJw.png

[4] ที่มาภาพ: https://miro.medium.com/max/2400/1*iFgI9nTtiCupbcK8D9mCXg.jpeg

3.6 Gradient Boosting

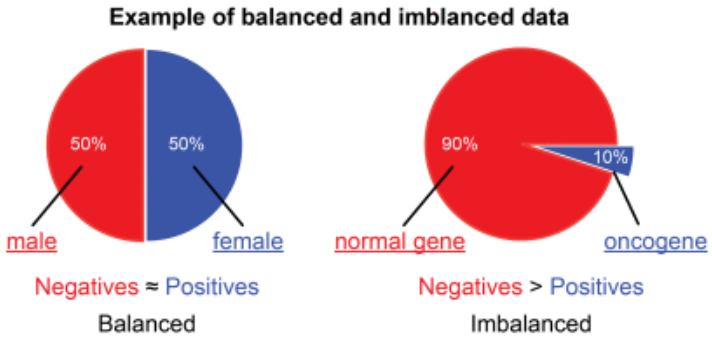
Gradient boosting เป็นเทคนิคการเรียนรู้ของเครื่องสำหรับปัญหาแบบการถดถอยและแบบแยกประเภท โดยสร้างโมเดลจาก Weak Learners ซึ่งเป็นโมเดลที่มีประสิทธิ์ภาพต่ำ และ Residuals มาช่วยปรับปรุงประสิทธิภาพต่อไปเรื่อย ๆ ให้กลายเป็น Strong Learner ซึ่งดีกว่า Random forest [6]



รูปที่ 5 หลักการของ Gradient Boosting[5]

3.7 ชุดข้อมูลไม่สมดุล (Imbalanced Dataset)

เป็นชุดข้อมูลที่มีสัดส่วนของคลาสใดคลาสหนึ่งเบ้ไปในทางใดทางหนึ่ง ซึ่งคลาสที่มีสัดส่วนมากของชุดข้อมูลจะเรียกว่าคลาสส่วนใหญ่ (Majority Classes) และคลาสที่มีสัดส่วนน้อยกว่าจะเรียกว่าคลาสส่วนน้อย (Minority Classes) [7]



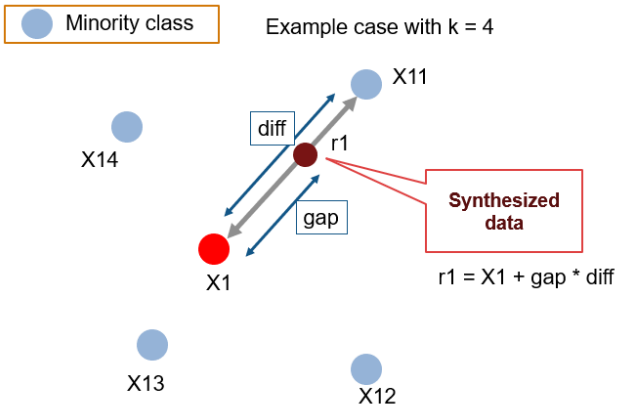
รูปที่ 6 ตัวอย่างข้อมูลสมดุลและไม่สมดุล[6]

[5] ที่มาภาพ: https://littleml.files.wordpress.com/2017/03/boosted-trees-process.png?w=810

[6] ที่มาภาพ: https://cdn-images-1.medium.com/max/450/1*zsyN08VVrgHbAEdvv27Pyw.png

3.8 Synthetic Minority Oversampling Technique: SMOTE

เป็นการแก้ปัญหาการแยกกลุ่มที่ไม่สมดุล (Imbalance Classification) ที่มีคลาสส่วนน้อยจำนวนน้อยเกินไปเพื่อให้โมเดลตัดสินใจได้อย่างมีประสิทธิภาพ โดยการ Oversample ตัวอย่างและการสังเคราะห์ (Synthesize) ข้อมูลจากคลาสส่วนน้อยเป็นข้อมูลใหม่ ซึ่ง SMOTE นั้นทำงานโดยการสุ่มเลือกข้อมูลที่เป็นคลาสส่วนน้อยมาแล้วทำการสุ่มข้อมูลที่เป็นเพื่อนบ้านใกล้เคียง (nearest neighbor) ของข้อมูลนั้นแล้วทำการเชื่อมกันด้วยเส้นในพื้นที่ฟีเจอร์ (feature space) และสุ่มค่าในระหว่างเส้นนั้นมาเป็นค่าของข้อมูลใหม่ในฟีเจอร์ (feature) นั้น [8]



รูปที่ 7 การ Oversampling สังเคราะห์ข้อมูลในคลาสส่วนน้อยเป็นข้อมูลใหม่[7]

4. งานวิจัยที่เกี่ยวข้อง (Related Works)

4.1 งานวิจัย “Using Machine Learning Techniques to Aid Empirical Antibiotic Therapy Decisions in the Intensive Care Unit of a General Hospital in Greece” [9]

งานวิจัยฉบับนี้เสนอการใช้เทคนิคการเรียนรู้ของเครื่องทำนายการดื้อยาต้านจุลชีพ เพื่อช่วยแพทย์ในการพิจารณาว่าแบคทีเรียมีภาวะดื้อยาหรือไม่ ซึ่งจะเป็นแนวทางให้แพทย์สามารถตัดสินใจเลือกใช้ยาต้านจุลชีพที่เหมาะสมได้ โดยทราบเพียงแค่การย้อมสีแกรมของตัวอย่าง บริเวณที่ติดเชื้อ และข้อมูลประชากร และทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการเรียนรู้ของเครื่อง 8 อัลกอริทึม ได้แก่ Linear SVC, SVM, SMO, KNN, J48, Random Forest, RIPPER และ MLP โดยชุดข้อมูลที่ใช้เป็นข้อมูลของห้องปฏิบัติการจุลชีววิทยาจากหออภิบาลผู้ป่วยวิกฤตในโรงพยาบาลของรัฐในกรีซในช่วงปี 2017 – 2018 ซึ่งชุดข้อมูล ประกอบไปด้วยแอตทริบิวต์ต่าง ๆ ได้แก่ เพศ (ไบนารี), อายุ (ตัวเลข), ประเภทของตัวอย่าง (หมวดหมู่), การย้อมสีแกรม (ไบนารี), ยา

[7] ที่มาภาพ: https://editor.analyticsvidhya.com/uploads/77417image1.png

ต้านจุลชีพ (หมวดหมู่) และคลาสแอตทริบิวต์ซึ่งก็คือความไวต่อยาต้านจุลชีพ (ไบนารี) ในงานวิจัยนี้ได้เลือกใช้ซอฟต์แวร์ WEKA ซึ่งเป็นหนึ่งในเครื่องมือการเรียนรู้ของเครื่องที่ได้รับความนิยมมากที่สุด อีกทั้งยังมีอัลกอริทึมที่หลากหลาย โดยทดสอบประสิทธิภาพของโมเดลด้วย 10-fold cross validation และวัดประสิทธิภาพของโมเดลจาก TP Rate, FP Rate, Precision, Recall, F-Measure, MMC, ROC Area และ PRC Area

4.2 งานวิจัย “Machine Learning Techniques to Identify Antimicrobial Resistance in the Intensive Care Unit” [10]

งานวิจัยฉบับนี้เสนอการใช้เทคนิคการเรียนรู้ของเครื่องเพื่อตรวจสอบว่าแบคทีเรียมีภาวะดื้อต่อยาต้านจุลชีพในกลุ่มต่าง ๆ หรือไม่ โดยพิจารณาจากข้อมูลทางคลินิกและข้อมูลประชากรของผู้ป่วย ตลอดจนข้อมูลการเพาะเลี้ยงเชื้อ และบันทึกการดื้อยาต้านจุลชีพของแบคทีเรีย (Antibiogram) นอกจากนี้ได้แสดงให้เห็นถึงความสัมพันธ์ระหว่างแบคทีเรียชนิดต่าง ๆ และกลุ่มของยาต้านจุลชีพโดยทำการวิเคราะห์การสมนัย (Correspondence Analysis) ผลลัพธ์ของการใช้เทคนิคการเรียนรู้ของเครื่องแสดงให้เห็นถึงความสัมพันธ์ที่ไม่เป็นเชิงเส้นซึ่งจะช่วยระบุการดื้อยาต้านจุลชีพในหออภิบาลผู้ป่วยวิกฤต โดยประสิทธิภาพจะขึ้นอยู่กับกลุ่มของยาต้านจุลชีพ

4.3 งานวิจัย “SMOTE: Synthetic Minority Oversampling Technique” [11]

ในงานวิจัยฉบับนี้เสนอเทคนิคการแก้ปัญหาชุดข้อมูลไม่สมดุล โดยใช้วิธีการ Over-sampling ซึ่งเป็นการนำคลาสส่วนน้อย (Minority Class) มาทำการ Over-sampling โดยการสร้างตัวอย่างสังเคราะห์ (Synthetic examples) เรียกเทคนิคนี้ว่า Synthetic Minority Oversampling Technique (SMOTE) ซึ่งให้ประสิทธิภาพดีกว่าการทำ Over-sampling โดยการสุ่มตัวอย่างแบบใส่คืน (Over-sampling with replacement) จากผลการศึกษาแสดงให้เห็นว่าวิธีการ SMOTE สามารถช่วยปรับปรุงความถูกต้องของตัวจำแนกประเภท (Classifier) สำหรับคลาสส่วนน้อยได้ (Minority Class) โดย SMOTE นั้นได้เป็นแนวทางใหม่ในการทำ Over-sampling และยังแสดงให้เห็นว่าการใช้ SMOTE กับ Under-sampling ร่วมกันจะให้ประสิทธิภาพที่ดีกว่าการทำ Under-sampling แบบธรรมดา

เหตุผลที่ SMOTE มีประสิทธิภาพมากกว่าการทำ Over-sampling โดยการสุ่มตัวอย่างแบบใส่คืน โดยพิจารณาจากผลกระทบต่อขอบเขตการตัดสินใจใน Feature space การสุ่มตัวอย่างแบบใส่คืนนั้นจะส่งผลให้การตัดสินใจจำแนกคลาสสำหรับคลาสส่วนน้อยอาจมีขนาดเล็กลงและมีความเฉพาะเจาะจงมากขึ้น เนื่องจากตัวอย่าง

ของคลาสส่วนน้อยถูกสุ่มซ้ำ ส่วนการสร้างตัวอย่างสังเคราะห์ (SMOTE) จะทำให้ตัวจำแนกคลาสสร้างขอบเขตการตัดสินใจที่ใหญ่ขึ้นซึ่งเป็นจุดที่อยู่ใกล้เคียงกับคลาสส่วนน้อย ทำให้ได้ประสิทธิภาพที่ดีกว่า

5. ขอบเขตของโครงงาน (Scope)

5.1 ใช้ข้อมูลกลุ่มตัวอย่างจากรายงานผลตรวจความไวต่อยาต้านจุลชีพของห้องปฏิบัติการจุลชีววิทยา (Results of Antimicrobial Susceptibility Testing : AST) ปี 2016 – 2020 ของโรงพยาบาลสัตว์เล็ก คณะสัตวแพทย์ศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

5.2 ใช้เทคนิค Classification โดยเลือกใช้อัลกอริทึม Decision Tree, SVM, Random Forest และ Gradient Boosting

5.3 ใช้เทคนิค SMOTE ในการแก้ปัญหาชุดข้อมูลไม่สมดุล

6. การพัฒนาโครงงาน (Project Development)

6.1 ขั้นตอนการพัฒนา (Methodology)

6.1.1 ศึกษา วิเคราะห์ และทำความเข้าใจปัญหา

6.1.2 การศึกษาข้อมูลที่เกี่ยวข้อง

- ศึกษาเพื่อค้นหาอัลกอริทึมที่จะนำมาใช้ในการสร้างโมเดล
- ศึกษาวิธีการแก้ปัญหาชุดข้อมูลไม่สมดุล
- ค้นคว้างานวิจัยที่เกี่ยวข้อง
- ศึกษาการใช้เครื่องมือต่าง ๆ

6.1.3 การจัดเตรียมข้อมูล (Data Preprocessing)

6.1.3.1 เตรียมชุดข้อมูล (Dataset)

6.1.3.2 ทำความสะอาดข้อมูล (Data Cleansing)

6.1.3.3 การวิเคราะห์ข้อมูลเชิงสำรวจ (Exploratory Data Analysis)

6.1.3.4 กระบวนการแปลงข้อมูล (Feature Engineering)

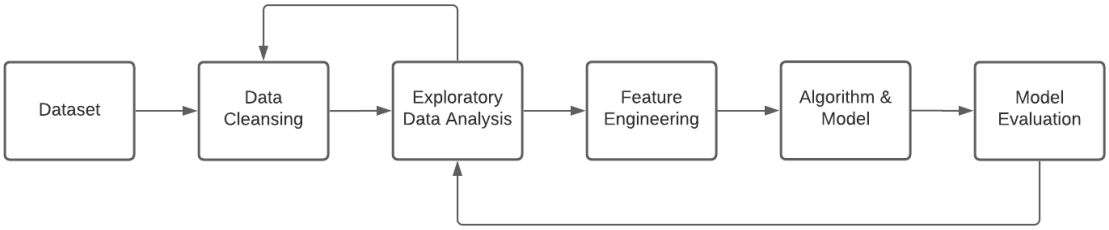
6.1.4 การเลือกอัลกอริทึมและสร้างโมเดล (Algorithm & Model)

6.1.5 การวัดประสิทธิภาพของโมเดล (Model Evaluation)

6.1.6 การปรับปรุงประสิทธิภาพของโมเดล

6.2 การออกแบบ (Design)

6.2.1 การออกแบบขั้นตอนการพัฒนา



รูปที่ 8 ขั้นตอนการพัฒนา

1) เตรียมชุดข้อมูล (Dataset)

ชุดข้อมูลได้มาจากรายงานผลตรวจความไวต่อยาต้านจุลชีพของห้องปฏิบัติการจุลชีววิทยา (Results of Antimicrobial Susceptibility Testing : AST) ปี 2016 – 2020 ของโรงพยาบาลสัตว์เล็ก คณะสัตวแพทย์ศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โดยทำการแบ่งชุดข้อมูลออกเป็น 2 ชุด คือ ชุดข้อมูลของแบคทีเรียแกรมบวก และชุดข้อมูลของแบคทีเรียแกรมลบ

2) ทำความสะอาดข้อมูล (Data Cleansing)

ตรวจสอบแก้ไขข้อมูลที่มีความผิดพลาดให้ถูกต้อง แก้ไขข้อมูลให้มีความสอดคล้องกัน เติมข้อมูลที่ขาดหายให้สมบูรณ์ กำจัดข้อมูลซ้ำซ้อน รวมทั้งกำจัด Noisy data

3) การวิเคราะห์ข้อมูลเชิงสำรวจ (Exploratory Data Analysis)

วิเคราะห์ตรวจสอบข้อมูลเบื้องต้นโดยใช้เทคนิค Data Visualization เช่น Bar Chart เป็นต้น เพื่อทำความเข้าใจชุดข้อมูล (Dataset) และวิเคราะห์ปัจจัยที่มีผลต่อการเลือกใช้ยาต้านจุลชีพ

4) กระบวนการแปลงข้อมูล (Feature Engineering)

แปลงข้อมูลโดยใช้เทคนิค Binning Methods, Label Encoding, One Hot Encoding รวมทั้งการตัด Feature ที่ไม่จำเป็นออก เพื่อช่วยทำให้อัลกอริทึมเรียนรู้ได้ดีขึ้น

5) การเลือกอัลกอริทึมและสร้างโมเดล (Algorithm & Model)

- อัลกอริทึมที่เลือกมาใช้ ได้แก่ Decision tree, SVM , Random Forest และ Gradient Boosting

- สร้างโมเดลของยาต้านจุลชีพชนิดละ 1 โมเดล โดยออกแบบให้แต่ละโมเดลมี 2 คลาส คือ True (แนะนำให้ใช้ยาต้านจุลชีพชนิดนี้) และ False (ไม่แนะนำให้ใช้ยาต้านจุลชีพชนิดนี้)

- แก้ปัญหาชุดข้อมูลไม่สมดุลโดยใช้เทคนิค SMOTE

6) การวัดประสิทธิภาพของโมเดล (Model Evaluation)

ใช้ค่าวัดประสิทธิภาพ ได้แก่ Accuracy, Precision, Recall, F-measure, ROC and AUC และ Confusion Matrix

6.2.2 การออกแบบการทดลอง (Experimental design)

การทดลองที่ 1 การเปรียบเทียบประสิทธิภาพของโมเดลที่ได้จาก Decision Tree, SVM, Random Forest และ Gradient Boosting

จุดประสงค์การทดลอง

1) เพื่อหาอัลกอริทึมที่เหมาะสมที่สุดกับชุดข้อมูล
2) เพื่อวัดประสิทธิภาพของโมเดลที่ได้จากแต่ละอัลกอริทึม

สมมติฐานการทดลอง

Gradient Boosting เป็นอัลกอริทึมที่เหมาะสมกับชุดข้อมูลและมีประสิทธิภาพดีที่สุด เพราะเป็น Algorithm ที่นำเอาข้อผิดพลาดของโมเดลก่อนหน้านี้มาปรับปรุง

วิธีการทดลอง

1) เลือกยาต้านจุลชีพที่จะนำมาสร้างโมเดลทำนายผล
2) สร้างโมเดลจากทั้ง 4 อัลกอริทึม
3) วัดประสิทธิภาพของโมเดล
4) เปรียบเทียบประสิทธิภาพของโมเดล
5) สรุปผลการทดลอง

การทดลองที่ 2 การแก้ปัญหาชุดข้อมูลไม่สมดุลโดยใช้เทคนิค SMOTE

จุดประสงค์การทดลอง

1) เพื่อศึกษาการแก้ปัญหาชุดข้อมูลไม่สมดุล
2) เพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลโดยใช้เทคนิค SMOTE
3) เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลระหว่างก่อนและหลังใช้เทคนิค SMOTE

สมมติฐานการทดลอง

1) การใช้เทคนิค SMOTE สามารถแก้ปัญหาชุดข้อมูลไม่สมดุลได้
2) การใช้เทคนิค SMOTE สามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลได้
3) หลังใช้เทคนิค SMOTE โมเดลมีประสิทธิภาพในการจำแนกข้อมูลมากกว่าก่อนใช้เทคนิค SMOTE

วิธีการทดลอง

1) เลือกยาต้านจุลชีพที่จะนำมาสร้างโมเดลทำนายผล
2) สร้างโมเดลจากทั้ง 4 อัลกอริทึม ร่วมกับการใช้เทคนิค SMOTE
3) วัดประสิทธิภาพของโมเดล
4) เปรียบเทียบประสิทธิภาพของโมเดลก่อนและหลังใช้เทคนิค SMOTE
5) สรุปผลการทดลอง

6.3 แนวทางการทดสอบและการวัดประสิทธิภาพ (Test and Performance Evaluation Approaches)

โครงงานนี้มีแนวทางการทดสอบโมเดลด้วยวิธี K-Fold Cross Validation ซึ่งจะแบ่งข้อมูลออกเป็น K ชุด เท่า ๆ กัน แล้วใช้ 1 ชุดมาเป็นชุดข้อมูลทดสอบ (Test Set) ส่วนที่เหลือ K-1 ชุด นำมาใช้เป็นชุดข้อมูลฝึกสอน (Training Set) แล้วทำวนไป K รอบ โดยเปลี่ยนชุดข้อมูลทดสอบไปเรื่อย ๆ จนครบ และใช้ค่าวัดประสิทธิภาพของโมเดล ดังนี้

- **ค่าความถูกต้อง (Accuracy)**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **ค่าความแม่นยำ (Precision)**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **ค่าความระลึก (Recall)**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **ค่าความถ่วงดุล (F-Measure)** เป็นค่าเฉลี่ยแบบ Harmonic ระหว่าง Precision และ Recall

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

- **Receiver Operating Characteristics (ROC) Curve And Area Under The Curve (AUC)**

ROC Curve เป็นกราฟที่แสดงความสัมพันธ์ระหว่าง True Positive Rate (TPR) และ False Positive Rate (FPR) โดยที่ AUC จะเป็นพื้นที่ใต้ ROC Curve มีค่าอยู่ระหว่าง 0 - 1

$$\text{TRP} = \frac{TP}{TP + FN} \qquad \text{FRP} = \frac{FP}{FP + TN}$$

AUC Value	Test Quality
0.90 – 1.00	Excellent
0.80 – 0.90	Very good
0.70 – 0.80	Good
0.60 – 0.70	Satisfactory
0.50 – 0.60	Unsatisfactory

- **Confusion Matrix**

Actual	Predicted	
	Positive	**Negative**
Positive	TP	FN
Negative	FP	TN

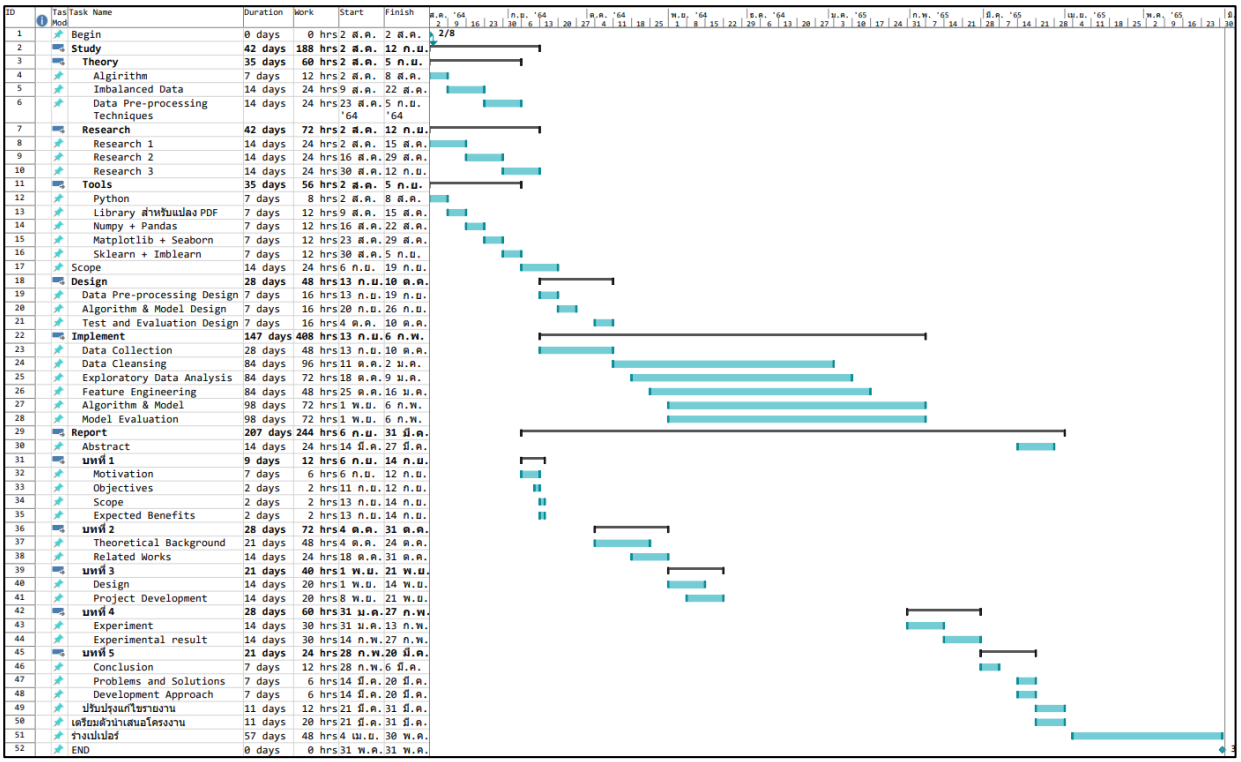
True Positive (TP) คือ โมเดลทำนายว่าให้ใช้ยาต้านจุลชีพชนิดนี้ และแพทย์แนะนำว่าให้ใช้ยาต้านจุลชีพชนิดนี้

False Positive (FP) คือ โมเดลทำนายว่าให้ใช้ยาต้านจุลชีพชนิดนี้ แต่แพทย์ไม่ได้แนะนำว่าให้ใช้ยาต้านจุลชีพชนิดนี้

True Negative (TN) คือ โมเดลทำนายว่าไม่ให้ใช้ยาต้านจุลชีพชนิดนี้ และแพทย์ไม่ได้แนะนำว่าให้ใช้ยาต้านจุลชีพชนิดนี้

False Negative (FN) คือ โมเดลทำนายว่าไม่ให้ใช้ยาต้านจุลชีพชนิดนี้ แต่แพทย์แนะนำว่าให้ใช้ยาต้านจุลชีพชนิดนี้

7. แผนการดำเนินโครงงาน (Gantt Chart)



ID	Task Mode	Task Name	Duration	Work	Start	Finish
1		Begin	0 days	0 hrs	2 ส.ค. '64	2 ส.ค. '64
2		**Study**	**42 days**	**188 hrs**	**2 ส.ค. '64**	**12 ก.ย. '64**
3		**Theory**	**35 days**	**60 hrs**	**2 ส.ค. '64**	**5 ก.ย. '64**
4		Algirithm	7 days	12 hrs	2 ส.ค. '64	8 ส.ค. '64
5		Imbalanced Data	14 days	24 hrs	9 ส.ค. '64	22 ส.ค. '64
6		Data Pre-processing Techniques	14 days	24 hrs	23 ส.ค. '64	5 ก.ย. '64
7		**Research**	**42 days**	**72 hrs**	**2 ส.ค. '64**	**12 ก.ย. '64**
8		Research 1	14 days	24 hrs	2 ส.ค. '64	15 ส.ค. '64
9		Research 2	14 days	24 hrs	16 ส.ค. '64	29 ส.ค. '64
10		Research 3	14 days	24 hrs	30 ส.ค. '64	12 ก.ย. '64
11		**Tools**	**35 days**	**56 hrs**	**2 ส.ค. '64**	**5 ก.ย. '64**
12		Python	7 days	8 hrs	2 ส.ค. '64	8 ส.ค. '64
13		Library สำหรับแปลง PDF	7 days	12 hrs	9 ส.ค. '64	15 ส.ค. '64
14		Numpy + Pandas	7 days	12 hrs	16 ส.ค. '64	22 ส.ค. '64
15		Matplotlib + Seaborn	7 days	12 hrs	23 ส.ค. '64	29 ส.ค. '64
16		Sklearn + Imblearn	7 days	12 hrs	30 ส.ค. '64	5 ก.ย. '64
17		Scope	14 days	24 hrs	6 ก.ย. '64	19 ก.ย. '64
18		**Design**	**28 days**	**48 hrs**	**13 ก.ย. '64**	**10 ต.ค. '64**
19		Data Pre-processing Design	7 days	16 hrs	13 ก.ย. '64	19 ก.ย. '64
20		Algorithm & Model Design	7 days	16 hrs	20 ก.ย. '64	26 ก.ย. '64
21		Test and Evaluation Design	7 days	16 hrs	4 ต.ค. '64	10 ต.ค. '64
22		**Implement**	**147 days**	**408 hrs**	**13 ก.ย. '64**	**6 ก.พ. '65**
23		Data Collection	28 days	48 hrs	13 ก.ย. '64	10 ต.ค. '64
24		Data Cleansing	84 days	96 hrs	11 ต.ค. '64	2 ม.ค. '65
25		Exploratory Data Analysis	84 days	72 hrs	18 ต.ค. '64	9 ม.ค. '65
26		Feature Engineering	84 days	48 hrs	25 ต.ค. '64	16 ม.ค. '65
27		Algorithm & Model	98 days	72 hrs	1 พ.ย. '64	6 ก.พ. '65
28		Model Evaluation	98 days	72 hrs	1 พ.ย. '64	6 ก.พ. '65
29		**Report**	**207 days**	**244 hrs**	**6 ก.ย. '64**	**31 มี.ค. '65**
30		Abstract	14 days	24 hrs	14 มี.ค. '65	27 มี.ค. '65
31		**บทที่ 1**	**9 days**	**12 hrs**	**6 ก.ย. '64**	**14 ก.ย. '64**
32		Motivation	7 days	6 hrs	6 ก.ย. '64	12 ก.ย. '64
33		Objectives	2 days	2 hrs	11 ก.ย. '64	12 ก.ย. '64
34		Scope	2 days	2 hrs	13 ก.ย. '64	14 ก.ย. '64
35		Expected Benefits	2 days	2 hrs	13 ก.ย. '64	14 ก.ย. '64
36		**บทที่ 2**	**28 days**	**72 hrs**	**4 ต.ค. '64**	**31 ต.ค. '64**
37		Theoretical Background	21 days	48 hrs	4 ต.ค. '64	24 ต.ค. '64
38		Related Works	14 days	24 hrs	18 ต.ค. '64	31 ต.ค. '64
39		**บทที่ 3**	**21 days**	**40 hrs**	**1 พ.ย. '64**	**21 พ.ย. '64**
40		Design	14 days	20 hrs	1 พ.ย. '64	14 พ.ย. '64
41		Project Development	14 days	20 hrs	8 พ.ย. '64	21 พ.ย. '64
42		**บทที่ 4**	**28 days**	**60 hrs**	**31 ม.ค. '65**	**27 ก.พ. '65**
43		Experiment	14 days	30 hrs	31 ม.ค. '65	13 ก.พ. '65
44		Experimental result	14 days	30 hrs	14 ก.พ. '65	27 ก.พ. '65
45		**บทที่ 5**	**21 days**	**24 hrs**	**28 ก.พ. '65**	**20 มี.ค. '65**
46		Conclusion	7 days	12 hrs	28 ก.พ. '65	6 มี.ค. '65
47		Problems and Solutions	7 days	6 hrs	14 มี.ค. '65	20 มี.ค. '65
48		Development Approach	7 days	6 hrs	14 มี.ค. '65	20 มี.ค. '65
49		ปรับปรุงแก้ไขรายงาน	11 days	12 hrs	21 มี.ค. '65	31 มี.ค. '65
50		เตรียมตัวนำเสนอโครงงาน	11 days	20 hrs	21 มี.ค. '65	31 มี.ค. '65
51		ร่างเปเปอร์	57 days	48 hrs	4 เม.ย. '65	30 พ.ค. '65
52		END	0 days	0 hrs	31 พ.ค. '65	31 พ.ค. '65

8. ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits)

8.1 ได้ทราบปัจจัยที่มีผลต่อการเลือกใช้ยาต้านจุลชีพของสัตวแพทย์จากผลการทดสอบความไวต่อยาต้านจุลชีพ

8.2 สามารถทำนายการใช้ยาต้านจุลชีพที่เหมาะสมได้ตรงตามคำแนะนำของสัตวแพทย์ได้

8.3 สามารถแก้ไขปัญหาชุดข้อมูลไม่สมดุลได้ (Imbalanced Dataset)

8.4 ทำการเปรียบเทียบประสิทธิภาพโมเดลของแต่ละอัลกอริทึม เพื่อเป็นแนวทางในการเลือกใช้อัลกอริทึมที่เหมาะสม

9. ผลการศึกษาเทคโนโลยีที่ใช้พัฒนา

9.1 Python

รูปที่ 8 Python[8]

Python เป็นเครื่องมือที่ใช้สำหรับ Programming เพื่อจัดการข้อมูลต่าง ๆ และสร้างโมเดลสำหรับทำนายผล โดยเราได้ทำการศึกษา Syntax ของภาษา python เช่น If-else, for-loop, while-loop, list, tuple, dictionary, function เป็นต้น

```python
def matrix_multiplication(A, B):
    if len(A) != len(B[0]):
        return [[]]
    M = []
    for i in range(len(A)):
        M.append([])
        for j in range(len(B[i])):
            M[i].append(0)
            for k in range(len(A)):
                M[i][-1] += A[i][k] * B[k][j]
    return M

print(matrix_multiplication([[1, 2, 3], [4, 5, 6], [
      7, 8, 9]], [[5, 7, 9], [2, 4, 6], [1, 5, 3]]))
```

รูปที่ 9 การสร้าง function ในภาษา Python

9.2 Numpy

รูปที่ 10 Numpy[9]

Numpy เป็นเครื่องมือที่ใช้สำหรับจัดการข้อมูลจำนานมากแบบชนิดเดียวกัน โดยเราได้ทำการศึกษาการใช้ numpy เช่น การสร้าง Array, การเปลี่ยนมิติของ Array, การดำเนินทางคณิตศาสตร์การต่าง ๆ ของ Array เช่น บวก, ลบ, คูณ, หาร, Transport และการหาค่าทางสถิติของ Array เช่น Mean, Median, Min-Max, Standard Deviation เป็นต้น

---

[8] ที่มาภาพ: https://logodownload.org/wp-content/uploads/2019/10/python-logo-3.png

[9] ที่มาภาพ: https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/NumPy_logo_2020.svg/1024px-NumPy_logo_2020.svg.png

```python
import numpy as np

A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = np.array([[5, 7, 9], [2, 4, 6], [1, 5, 3]])

print(np.dot(A, B))
```

รูปที่ 11 การสร้าง Array โดยใช้ Numpy

9.3 Pandas

รูปที่ 12 Pandas[10]

Pandas เป็นเครื่องมือที่ใช้สำหรับจัดการข้อมูลในรูปแบบของตารางซึ่งสามารถใช้งานร่วมกับ Numpy ได้ โดยเราได้ทำการศึกษาการใช้ Pandas เช่น การ Import และ Export file, การสร้าง series และ Dataframe, การเข้าถึงค่า, การแก้ไขค่า, การเพิ่ม-ลบ Row และ Column, การเปลี่ยนชื่อ Column, การตรวจสอบค่าว่างและกำจัดค่าว่าง, การดำเนินทางคณิตศาสตร์และสถิติ และการจับกลุ่มของข้อมูล

```python
import pandas as pd

data = [{"name" : "John" , "age" : 21 , "sex" : "male"},
{"name" : "Smith" , "age" : 23 , "sex" : "male"},
{"name" : "Maple" , "age" : 19 , "sex" : "female"}]

df = pd.DataFrame(data);

print(df.head())
```

รูปที่ 13 การสร้าง Dataframe โดยใช้ Pandas

9.4 Matplotlib

รูปที่ 14 Matplotlib[11]

[10] ที่มาภาพ: https://upload.wikimedia.org/wikipedia/commons/thumb/e/ed/Pandas_logo.svg/1200px-Pandas_logo.svg.png

[11] ที่มาภาพ: https://matplotlib.org/_static/logo2_compressed.svg

Matplotlib เป็นเครื่องมือที่ใช้สำหรับสร้างกราฟ โดยเราได้ทำการศึกษาการใช้ Matplotlib เช่น การ Plot ต่าง ๆ ได้แก่ Bar Chart, Pie Chart, Scatter Plot, Histogram เป็นต้น และการตกแต่งตัวกราฟ ได้แก่ การใส่ชื่อแกน, ชื่อกราฟ, การเปลี่ยนสีของกราฟ และการเปลี่ยนสัญลักษณ์ของจุด

```python
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 50)
y = x * .5 + (np.sin(x) * np.random.rand(*x.shape))

plt.scatter(x, y)
plt.plot(x, x * .5)
plt.show()
```

รูปที่ 15 การสร้าง Scatter Plot โดยใช้ Matplotlib

9.5 Seaborn

รูปที่ 16 Seaborn[12]

Seaborn เป็นเครื่องมือที่ใช้สำหรับสร้างกราฟเหมือนกับ matplotlib แต่สามารถสร้างกราฟได้สะดวกกว่าและใช้งานร่วมกับ Dataframe ได้ดีกว่า matplotlib โดยเราได้ทำการศึกษาการใช้ seaborn เช่น การ Plot ต่าง ๆ ได้แก่ Count Plot, Bar Plot, Heat Map เป็นต้น และการตกแต่งตัวกราฟ

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.DataFrame()

sex = np.array(['male', 'male', 'female', 'female', 'male',
                'female', 'female', 'female', 'female', 'male',
                'male', 'male', 'female', 'female', 'male'])
grade = np.array(['C', 'C', 'B', 'B', 'A',
                  'B', 'B', 'B', 'C', 'A',
                  'C', 'B', 'B', 'A', 'C'])
df["sex"] = sex
df["grade"] = grade

sns.countplot(data=df, x="grade", hue="sex")
plt.show()

```

รูปที่ 17 การสร้าง Count Plot โดยใช้ Seaborn

[12] ที่มาภาพ: https://seaborn.pydata.org/_static/logo-wide-lightbg.svg

9.6 Scikit-learn

รูปที่ 18 Scikit-learn[13]

Scikit-learn เป็นเครื่องมือที่ไว้สำหรับทำงาน Machine Learning โดยเราได้ทำการศึกษาการใช้ Scikit-Learn เช่น การแบ่งข้อมูลไว้สำหรับ Train Validation และ การสร้างโมเดลต่าง ๆ ได้แก่ Decision Tree, SVM, Random Forest เป็นต้น และการประเมินผลโมเดล ได้แก่ Accuracy, Precision, Recall, F1-Score เป็นต้น

```
1  import numpy as np
2  from sklearn import tree
3  from sklearn import datasets
4
5  iris = datasets.load_iris()
6  clf = tree.DecisionTreeClassifier()
7  clf = clf.fit(iris.data, iris.target)
8  answer = clf.predict([[4.5, 2.9, 3.1, 1.8]])
9  print(iris.target_names[answer])
10
```

รูปที่ 19 การสร้างโมเดลด้วย Decision Tree

9.7 Imbalanced-learn

รูปที่ 20 Imbalanced-learn[14]

Imbalanced-learn เป็นเครื่องมือที่เอาไว้ช่วยจัดการข้อมูลสำหรับข้อมูลที่เป็น imbalanced classes โดยเราทำการศึกษาการใช้ Imbalanced-learn เช่น การทำ Over-sampling โดยใช้ SMOTE, การทำ Under-Sampling เป็นต้น

---

[13] ที่มาภาพ: https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Scikit_learn_logo_small.svg/1200px-Scikit_learn_logo_small.svg.png

[14] ที่มาภาพ: https://imbalanced-learn.org/stable/_static/logo.png

```python
from collections import Counter
from imblearn.datasets import fetch_datasets
from imblearn.over_sampling import SMOTE

data_set = fetch_datasets(filter_data=(1,))['ecoli']
print('before', Counter(data_set.target))
sm = SMOTE()
new_data_set = sm.fit_resample(data_set.data, data_set.target)
print('after', Counter(new_data_set[1]))
```

รูปที่ 21 การทำ Over-sampling โดยใช้ SMOTE

10. เอกสารอ้างอิง (Reference)

[1] Wikipedia. **Machine learning.** [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning

[2] Wikipedia. **Statistical classification**. [Online]. Available: https://en.wikipedia.org/wiki/Statistical_classification

[3] Wikipedia. **Decision tree**. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree

[4] Sunil Ray. 2017. **Understanding Support Vector Machine (SVM) algorithm from examples (along with code)**. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[5] ชิตพงษ์ กิตติราดร. 2020. **Random Forest**. [Online]. Available: https://guopai.github.io/ml-blog10.html

[6] Wikipedia. **Gradient boosting**. [Online]. Available: https://en.wikipedia.org/wiki/Gradient_boosting

[7] Google Developers. **Imbalanced Data**. [Online]. Available: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

[8] Jason Brownlee. 2021. **SMOTE for Imbalanced Classification with Python**. [Online]. Available: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[9] Feretzakis G, Loupelis E, Sakagianni A, et al. 2020. **Using Machine Learning Techniques to Aid Empirical Antibiotic Therapy Decisions in the Intensive Care Unit of a General Hospital in Greece**, Antibiotics 9(2). [Online]. Available: https://www.mdpi.com/2079-6382/9/2/50

[10] Martínez-Agüero S, Mora-Jiménez I et al. 2019. **Machine Learning Techniques to Identify Antimicrobial Resistance in the Intensive Care Unit**, Entropy 21. [Online]. Available: https://www.mdpi.com/1099-4300/21/6/603

[11] Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. 2002. SMOTE: **Synthetic Minority Over-Sampling Technique**, Journal of Artificial Intelligence Research, vol. 16, 321-357., [Online]. Available: https://arxiv.org/pdf/1106.1813.pdf