**ระบบคาดการณ์ผลลัพธ์การผลิตบัณฑิตของหลักสูตร จากข้อมูลผลการเรียนของนักศึกษา**

**Curriculum output prediction from student academic data**

**ณิชกานต์ สุขุมจิตพิทโยทัย**

**นรวิชญ์ อยู่บัว**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2565**

ปริญญานิพนธ์ปี การศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบคาดการณ์ผลลัพธ์การผลิตบัณฑิตของหลักสูตร จากข้อมูลผลการเรียนของนักศึกษา

Curriculum output prediction from student academic data

ผู้จัดทำ

1. นางสาวณิชกานต์ สุขุมจิตพิทโยทัย รหัสนักศึกษา 62010299

2. นายนรวิชญ์ อยู่บัว รหัสนักศึกษา 62010465

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_อาจารย์ที่ปรึกษา

(ผศ. ดร. ธนัญชัย ตรีภาค)

**ระบบคาดการณ์ผลลัพธ์การผลิตบัณฑิตของหลักสูตร จากข้อมูลผลการเรียนของนักศึกษา**

นางสาวณิชกานต์ สุขุมจิตพิทโยทัย 62010299

นายนรวิชญ์ อยู่บัว 62010465

ผศ. ดร. ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา

ปีการศึกษา 2565

# บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อ พัฒนาระบบประมวลผลข้อมูลผลการเรียนของนักศึกษาในอดีต ข้อมูลของรายวิชาต่างๆ และข้อมูลจากแบบสำรวจการมีงานทำของบัณฑิต เพื่อนำเสนอข้อมูลสถิติต่างๆ วิเคราะห์ข้อมูลผลการผลิตบัณฑิตเพื่อให้ได้ผลลัพธ์ว่าที่ผ่านมาหลักสูตรสามารถผลิตบัณฑิตกลุ่มใดได้บ้าง มีจำนวนมากน้อยเพียงใด สามารถพยากรณ์ว่าในอนาคตหลักสูตรสามารถผลิตบัณฑิตกลุ่มใดได้เป็นจำนวนเท่าใด เพื่อเป็นประโยชน์และอำนวยความสะดวกให้กรรมการหลักสูตรในการวางแผนการบริหารหลักสูตรในอนาคต และแสดงเป็นแผนภาพกราฟิกในการอำนวยความสะดวกให้หน่วยงานภายนอกได้รับทราบว่าหลักสูตรปัจจุบันของสถาบันสามารถผลิตบุคลากรที่มีความชํานาญด้านใดได้บ้าง

**Curriculum output prediction from student academic data**

Ms. Nichakan Sukhumjitpitayotai 62010299

Mr. Narawich Youbua 62010465

Mr. Thanunchai Threepak Advisor

Academic Year 2022

# Abstract

The project was created to develop a data processing system that will measure the process of processing academic performance data from former students, data from various courses, and data from graduate employment surveys. We can present statistics and analyze graduate data to figure out which graduate groups have already been generated. It can forecast how many graduate groups that program will generate in the future based on historical data. This can help and facilitate course directors in future course administration planning, and it can be displayed as a graphical diagram in course management. The purpose of showing as a graphical representation is to let external agencies understand how the institution's existing curriculum can generate employees with expertise in any field.

# กิตติกรรมประกาศ

โครงงานในภาคการศึกษานี้สำเร็จลุล่วงได้ด้วยดีจากความช่วยเหลือจากหลากหลายบุคคล โครงงานในภาคการศึกษานี้จะผ่านไปไม่ได้หากปราศจากความช่วยเหลือจากบุคคลเหล่านี้ขอขอบคุณอาจารย์ที่ปรึกษา ผศ. ดร. ธนัญชัย ตรีภาค ที่ให้ความช่วยเหลือในเรื่องต่าง ๆ ไม่ว่าจะเป็นการให้คำแนะนำถึงแนวทางการทำงานที่ดี การให้คำปรึกษาเพื่อหาทางออกเมื่อพบเจอกับปัญหา รวมถึงให้ความรู้เกี่ยวกับตัวงานทำให้งานต่าง ๆ เมื่อเจอปัญหาก็สามารถผ่านไปได้ด้วยดี

ขอขอบคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ประสาทวิชาการความรู้มาตลอด 4 ปี ซึ่งความรู้หลาย ๆ แขนงก็ถูกใช้เป็นพื้นฐาน และเป็นส่วนหนึ่งของโครงงานนี้

ขอขอบคุณเพื่อน ๆ วิศวกรรมคอมพิวเตอร์ที่ให้คำปรึกษา และแลกเปลี่ยนความรู้ซึ่งกันและกันรวมถึงการรับฟังปัญหา

สุดท้ายนี้ขอขอบคุณบิดา มารดาและครอบครัว ที่เลี้ยงดูอบรมสั่งสอนและให้ความรู้คุณธรรมจริยธรรม และให้การสนับสนุนด้านการศึกษาจนได้มีโอกาสมาทำโครงงานนี้

ณิชกานต์ สุขุมจิตพิทโยทัย

นรวิชญ์ อยู่บัว

# สารบัญ

**หน้า**

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ประโยชน์ของโครงงาน 2

1.4 ข้อจำกัดของโครงงาน 2

1.5 แผนการดำเนินงาน 3

บทที่ 2 ทฤษฏีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฏีที่เกี่ยวข้อง 4

2.2 เครื่องมือที่เกี่ยวข้อง 12

2.3 งานวิจัยที่เกี่ยวข้อง 16

บทที่ 3 การออกแบบ

3.1 โครงสร้างและการทำงานของระบบ 19

3.2 การทำงานของระบบ 19

3.3 Use Case Diagram 22

3.4 Sequence Diagram 30

# สารบัญ(ต่อ)

**หน้า**

3.5 การออกแบบฐานข้อมูล 36

บทที่ 4 ผลการดำเนินงาน

4.1 การสร้าง Django Framework เชื่อมต่อ MariaDB 39

4.2 การเตรียม Server ที่ใช้สำหรับ Deploy Application 42

4.3 การใช้งานฐานข้อมูลร่วมกับ Application 43

4.4 การพัฒนา Similarity Model 46

4.5 การนำชุดข้อมูลวิชาที่จัดกลุ่มไปใช้กับ Similarity Model 48

4.6 การพัฒนา Job Classification Model 52

4.7 การพัฒนาหน้าต่างผู้ใช้งานของ Application 53

4.8 การนำ Model และการเรียกใช้ลงบน Django Application 57

บทที่ 5 สรุป

5.1 บทสรุป 59

5.2 ปัญหาและอุปสรรคที่พบ 60

5.3 แนวทางในการพัฒนาต่อ 60

5.4 แนวทางในการพัฒนาต่อ 60

เอกสารอ้างอิง 61

ภาคผนวก ก 63

# สารบัญตาราง

**ตาราง** **หน้า**

[ตาราง 3.1 Use Case ดูผลลัพธ์สายงานของบัณฑิตย้อนหลัง 23](#_Toc88743161)

[ตาราง 3.2 Use Case คาดการณ์ผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต](#_Toc88743162) 23

[ตาราง 3.3 Use Case กรอกแบบฟอร์มเกรดสำหรับการวิเคราะห์ข้อมูล](#_Toc88743163) 24

[ตาราง 3.4 Use Case แสดงค่าความถนัดของนักศึกษา](#_Toc88743164) 24

[ตาราง 3.5 Use Case คาดการณ์เกรดแต่ละวิชาของนักศึกษา](#_Toc88743165) 25

[ตาราง 3.6 Use Case แนะนำวิชาเลือกภาคให้แก่นักศึกษา](#_Toc88743166) 25

[ตาราง 3.7 Use Case ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร](#_Toc88743167) 26

[ตาราง 3.8 Use Case อัปเดตข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร](#_Toc88743168) 26

[ตาราง 3.9 Use Case ลบข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร](#_Toc88743169) 27

[ตาราง 3.10 Use Case ล็อกอินเข้าใช้งาน](#_Toc88743170) 27

[ตาราง 3.11 Use Case ทำนายพยากรณ์ผลลัพธ์และคาดการณ์](#_Toc88743171) 28

[ตาราง 3.12 Use Case จัดเตรียมข้อมูล](#_Toc88743172) 28

[ตาราง 3.13 Use Case กรอกข้อมูลอาชีพ](#_Toc88743172) 29

# สารบัญรูป

**รูป** **หน้า**

รูป 1.1 แผนการดำเนินงาน 3

รูป 2.1 ประเภทของ Machine Learning 6

รูป 2.2 รูปแบบของ Content-based Filtering 10

รูป 2.3 รูปแบบของ Memory-based 11

รูป 2.4 การทำงานของแอปพลิเคชันต่าง ๆ บน Docker Engine 13

รูป 2.5 ประเภทของ Machine Learning 17

รูป 3.1 โครงสร้างการทำงานของระบบ 19

รูป 3.2 Use Case Diagram 22

รูป 3.3 Sequence diagram สำหรับ UC-01 30

รูป 3.4 Sequence diagram สำหรับ UC-02 30

รูป 3.5 Sequence diagram สำหรับ UC-03 31

รูป 3.6 Sequence diagram สำหรับ UC-04 31

รูป 3.7 Sequence diagram สำหรับ UC-05 32

รูป 3.8 Sequence diagram สำหรับ UC-06 32

รูป 3.9 Sequence diagram สำหรับ UC-07 33

รูป 3.10 Sequence diagram สำหรับ UC-08 33

รูป 3.11 Sequence diagram สำหรับ UC-09 34

รูป 3.12 Sequence diagram สำหรับ UC-10 34

รูป 3.13 Sequence diagram สำหรับ UC-11 35

รูป 3.14 Sequence diagram สำหรับ UC-12 35

รูป 3.15 Sequence diagram สำหรับ UC-13 36

รูป 3.16 Data Schema 36

รูป 4.1 File Requirements 39

รูป 4.2 Docker file 40

รูป 4.3 File Docker - compose.yml 41

# สารบัญรูป(ต่อ)

**รูป** **หน้า**

รูป 4.4 ภาพรวมของการตรวจสอบการทำงานของระบบ 41

รูป 4.5 Directory ที่เก็บ Django Application 42

รูป 4.6 ผลลัพธ์การสร้าง Docker Compose บน Server 43

รูป 4.7 ผลลัพธ์การค้นหาข้อมูลผ่าน API 44

รูป 4.8 ผลลัพธ์การแก้ไขข้อมูลผ่าน API 44

รูป 4.9 ผลลัพธ์การเพิ่มข้อมูลผ่าน API 45

รูป 4.10 ผลลัพธ์การลบข้อมูลผ่าน API 45

รูป 4.11 ชุดข้อมูลในการทดลอง 46

รูป 4.12 ชุดข้อมูลหลังจากผ่านการเตรียมข้อมูล 47

รูป 4.13 ตัวอย่างข้อมูล Data Subject 48

รูป 4.14 ข้อมูล Data Subject หลังจากตัดรหัสวิชาที่ซ้ำออก 49

รูป 4.15 ข้อมูล Data Subjectหลังจากจัดแต่งบทคัดย่อ 49

รูป 4.16 Coding การเทียบค่า Similarity 50

รูป 4.17 ชุดข้อมูลหลังจากผ่านการเตรียมข้อมูลผลลัพธ์จาก NLP 51

รูป 4.18 ผลลัพธ์การทดลองร่วมกับ NLP 51

รูป 4.19 ชุดข้อมูลแบบสอบถามการมีงานทำของบัณฑิต 52

รูป 4.20 ชุดข้อมูลที่ผ่านการ Transpose 53

รูป 4.21 Dockerfile React Project 54

รูป 4.22 docker-compose.yml React Project 54

รูป 4.23 ภาพรวมของการตรวจสอบการทำงานของ React Project 55

รูป 4.24 ภาพรวมของ React Project และ Django Application ที่เรียกดูผ่าน Web Browser 56

รูป 4.25 ทดสอบการติดต่อระหว่าง React Project และ Django Application 56

รูป 4.26 React Project ที่นำขึ้นไปยัง Server ที่เตรียมไว้ 57

รูป ก1 แผนภาพการทำงานโดยรวมของระบบ 64

**บทที่ 1**

**บทนำ**

* 1. **ความเป็นมาของปัญหา**

Data Analytics เป็นการวิเคราะห์ข้อมูลที่มีอยู่ตั้งแต่อดีตจนถึงปัจจุบัน ในกรณีที่ข้อมูลเพียงพอและเหมาะสมจะสามารถนำมาคาดการณ์แนวโน้ม ทำนายอนาคตที่เป็นประโยชน์ พยากรณ์สิ่งที่กำลังจะเกิดขึ้นหรือน่าจะเกิดขึ้นโดยใช้ข้อมูลในอดีตกับแบบจำลองทางสถิติรวมถึงการให้คำแนะนำทางเลือกต่าง ๆ และผลของแต่ละทางเลือก

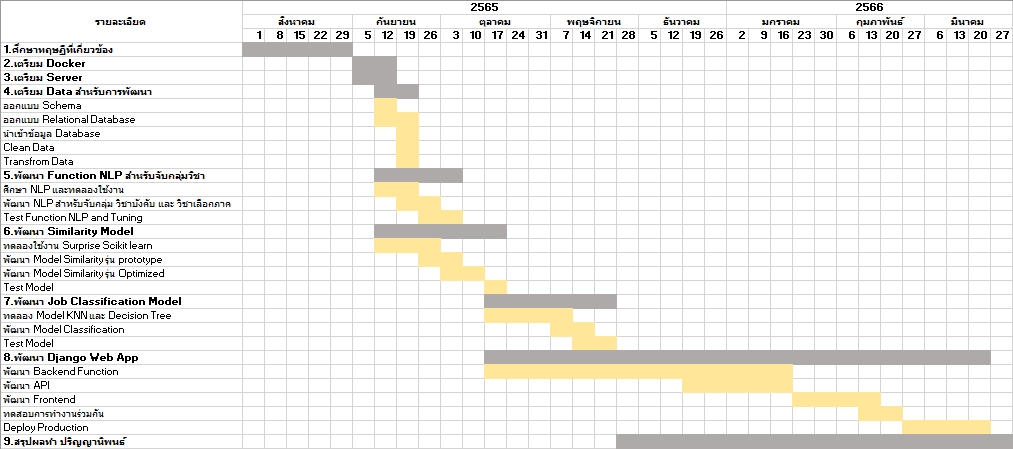
จากปัญหาที่ทางผู้จัดทำเล็งเห็นความสำคัญคือการนำข้อมูลผลการเรียนของนักศึกษาในอดีตมาใช้ประโยชน์ในการบริหารหลักสูตร และ นำมาวิเคราะห์ผลเพื่อช่วยในการวางแผนการเรียนของนักศึกษา ซึ่งการวางแผนในการเรียนของหลักสูตรจะสามารถช่วยอาจารย์และบุคลากรที่เกี่ยวข้องกับการศึกษาในด้านของการบริหารหลักสูตร เพื่อวางแผนการเพิ่มหรือลดจำนวนผู้เรียนในรายวิชาต่าง ๆ ซึ่งส่งผลต่อการผลิตบัณฑิตด้านต่าง ๆ ได้

ดังนั้นผู้จัดทำจึงได้เห็นถึงความสำคัญการประเมินสถานะขอหลักสูตร ของระบบแนะนำการวางแผนการคาดการณ์จากการใช้ความรู้ทางด้าน Data Analytics, Prediction และ Recommendation โดยใช้ข้อมูลผลการเรียนของนักศึกษาในอดีต เพื่อพัฒนาระบบช่วยเหลือ และตอบโจทย์ให้แก่นักศึกษาและบุคลากรทางการศึกษาหรือบุคคลที่เกี่ยวข้องได้

* 1. **วัตถุประสงค์**

1. เพื่อนำข้อมูลของผลการเรียนของนักศึกษาในอดีตและข้อมูลจากแบบสำรวจการมีงานทำของบัณฑิตมาใช้ ในการวางแผนการเรียนหรือประเมินอาชีพในอนาคตของนักศึกษาได้
2. ประมวลผลข้อมูลผลการเรียนของนักศึกษาในอดีต และข้อมูลจากแบบสำรวจการมีงานทำของบัณฑิต และทำแผนภาพกราฟิกเพื่อนำเสนอข้อมูล อำนวยความสะดวกให้กรรมการหลักสูตรในการวางแผนการ ทำงาน
3. เพื่อนำข้อมูลผลการเรียนของนักศึกษาในอดีต มาพัฒนาเป็นระบบแนะนำและวางแผนการเรียนตัวของ นักศึกษาได้
4. เพื่อนำข้อมูลการพยากรณ์อาชีพในอนาคตของนักศึกษาในสถาบันมาแสดงเป็นแผนภาพกราฟิกในการ อำนวยความสะดวกให้หน่วยงานภายนอกได้รับทราบว่าหลักสูตรปัจจุบันของสถาบันสามารถผลิต บุคลากรที่มีความชํานาญด้านใดได้บ้าง
   1. **ประโยชน์ของโครงงาน**
5. ได้ระบบรวบรวมข้อมูลผลการเรียนของนักศึกษาและข้อมูลแบบสำรวจการทำงานของบัณฑิต แล้วนำมาวิเคราะห์และนำเสนอข้อมูลที่เป็นประโยชน์ในการบริหารหลักสูตรของกรรมการหลักสูตร
6. มีระบบที่สามารถแนะนำ วางแผน และประเมินอาชีพในอนาคตจากผลการเรียนของนักศึกษา
   1. **ข้อจำกัดของโครงงาน**
7. ข้อมูลผลการเรียนในอดีตย้อนหลังมีเพียง 2 ปี
8. ข้อมูลผลการเรียนในอดีตจะได้จากสำนักทะเบียนและประมวลผล โดยกรรมการหลักสูตรจะเป็นผู้ร้องขอข้อมูลดังกล่าวและนำเข้าระบบ
9. การทำนายต่าง ๆ จะใช้ข้อมูลเพียง 2 แหล่งคือข้อมูลผลการเรียนของนักศึกษาจากสำนักทะเบียนและประมวลผล และแบบสอบถามการมีงานทำของบัณฑิตเท่านั้น
   1. **แผนการดำเนินงาน**

แผนการดำเนินงานในการพัฒนาโครงงานตลอดระยะเวลา 2 ภาคการศึกษา ตั้งแต่เดือน สิงหาคม พ.ศ. 2565 - มีนาคม พ.ศ. 2566 แสดงดังรูป 1.1



**รูป 1.1 แผนการดำเนินงาน**

**บทที่ 2**

**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

* 1. **ทฤษฎีที่เกี่ยวข้อง**
     1. **Classification and Predict**

Classification and Prediction คือการจำแนกประเภทของข้อมูลหรือการนำชุดข้อมูลที่มีความใกล้เคียงกันมาจัดเข้าด้วยกัน โดยจะนำมาใช้ในการวิเคราะห์ข้อมูล ซึ่งกระบวนการดั่งกล่าวสามารถแบ่งได้เป็น 2 ส่วน

* + - 1. **Training Data**

Training Data คือการนำข้อมูลมาที่ได้มาทำการเรียนรู้ให้กับคอมพิวเตอร์เพื่อนำไปสร้างเป็นโมเดลแบบจำลองและวัดประสิทธิภาพของโมเดลแบบจำลองนั้น เป็นระบบที่ใช้ในการสอนระบบ โดยจะทำการสร้างโมเดลซึ่งจะมีด้วยกันหลายวิธี เช่น Decision Tree, Naive Bayes, K Nearest Neighbors และ Neural Network เป็นต้น

* + - 1. **Predict**

Predict คือการนำข้อมูลใหม่ที่รับมานำเข้าโมเดลแบบจำลองที่เป็นผลลัพธ์จากการผ่านกระบวนการ Training Data ไปทำการคำนวณหรือพยากรณ์ประเภทของปัญหาในด้าน Classification

1. Binary classification (การจำแนกแบบไบนารี)

คือตัวแปรที่อยู่ในรูปแบบสองหมวดหมู่ที่มีแค่ 2 ผลลัพธ์ เช่น ผลลัพธ์แบบ ใช่ หรือ ไม่ใช่ ตก หรือ ผ่าน หากเปรียบในรูปแบบของตัวเลขก็คือ 0 กับ 1 อัลกอริทึมที่ใช้คู่กับการจำแนกแบบไบนารี จะมีดังนี้ k-Nearest Neighbors Decision Trees หรือ Naive Bayes

1. Multi-Class Classification (การจำแนกประเภทหลายคลาส)

ในการจำแนกรูปแบบนี้จะต่างกับการจำแนกแบบไบนารี โดยจะมีผลลัพธ์หรือหมวดหมู่มากกว่า 2 ผลลัพธ์ ตัวอย่างของการจำแนกประเภทนี้ เช่น รูปภาพที่มีองค์ประกอบคล้ายคลึงกับรูปภาพที่อยู่ในฐานข้อมูลเพื่อค้นหาคำศัพท์ที่คาดว่าจะพิมพ์ใน predictive keyboard โดยผลลัพธ์ที่อาจเกิดนั้นจะมีได้มากกว่า 2 หมวดหมู่ อัลกอริทึมที่ใช้คู่กับการจำแนกประเภทนี้สามารถใช้อัลกอริทึมคล้ายกับการจำแนกแบบไบนารีได้

1. Multi-Label Classification (การจำแนกประเภทหลายเลเบล)

เปรียบให้เข้าใจง่ายโดยการยกตัวอย่างเช่น รูปภาพรูปหนึ่งสามารถมีรูปดอกไม้ ท้องฟ้าก้อนเมฆได้ แต่รูปภาพรูปนั้นจะจัดว่าเป็นหมวดหมู่รูปวาด รูปถ่าย หรือรูปเสีย Multi-Label Classification ก็คือการทำเลเบลให้กับชุดข้อมูล หรือการติดฉลากให้รูปนั้น ๆ ว่ามีดอกไม้หรือเปล่ามีก้อนเมฆหรือไม่ส่วน Multi-Class Classification จะจำแนกว่ารูปนั้นเป็นรูปที่เกิดจากการวาดหรือรูปที่เกิดจากการถ่ายหรือรูปเสีย

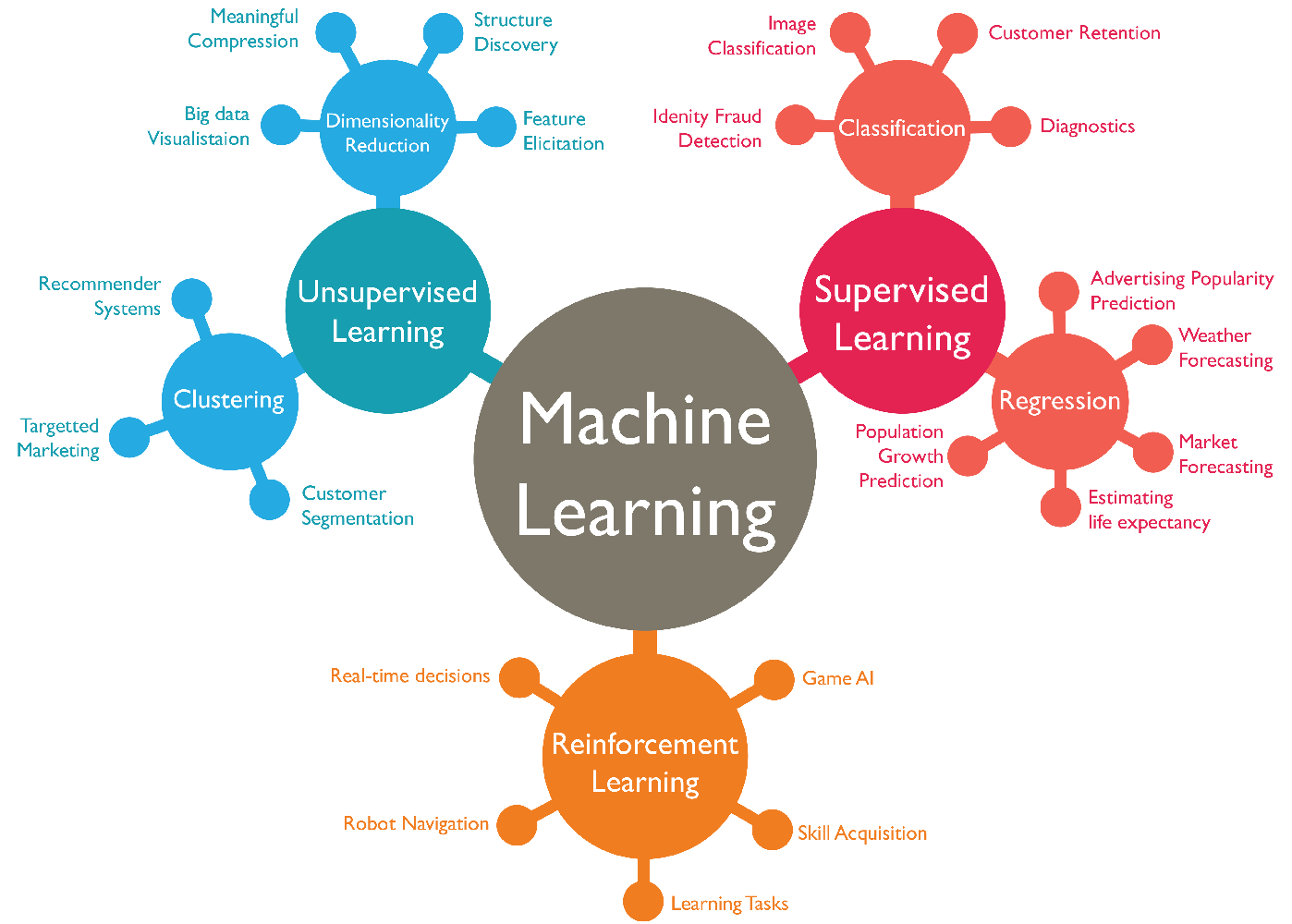
1. Imbalanced Classification (การจำแนกแบบข้อมูลไม่เท่าเทียม)

คือปัญหาที่เกิดจากข้อมูลที่มีไม่เท่าเทียมกัน (Imbalanced dataset) ตัวอย่างเช่นข้อมูลของการทุจริตโดยข้อมูลส่วนใหญ่ย่อมเป็นข้อมูลที่จัดว่า “ไม่ทุจริต” และจะมีเปอร์เซ็นต์น้อยที่จัดว่าเป็น “ทุจริต” เป็นต้น โดยจะเปรียบโดยง่ายคือกรณีที่ชุดข้อมูลมีการแยกประเภทกันแต่จำนวนของประเภทนั้นมีอัตราส่วนของข้อมูลที่ห่างกันค่อนข้างมาก

* + 1. **Machine Learning**

Machine Learning คือ การทำให้ระบบคอมพิวเตอร์มีความสามารถเรียนรู้ได้ด้วยตนเอง โดยจะใช้ข้อมูล ด้วยวิธีการใส่ข้อมูลและผลลัพธ์เข้าไป เหมือนกันเป็นการสอนระบบคอมพิวเตอร์ เพื่อทำให้โปรแกรมนำผลลัพธ์นั้นไปประมวลผลและพยากรณ์ Output และ Input ของข้อมูลใหม่ โดยแบ่ง Machine Learning ออกได้เป็น 3 ประเภท คือ

1. Supervised Learning คือการเรียนรู้ที่เครื่องจักรหรือคอมพิวเตอร์นั้นจำเป็นต้องอาศัยข้อมูลในการฝึกฝน ซึ่งจำเป็นที่จะต้องอาศัยชุดของข้อมูล ที่จะประกอบไปด้วยชุดของข้อมูลและชุดของผลลัพธ์ของข้อมูลที่ต้องการจะนำมาให้เครื่องจักรหรือคอมพิวเตอร์ใช้ในการเรียนรู้
2. Unsupervised Learning เป็นการเรียนรู้ที่ให้เครื่องจักรหรือคอมพิวเตอร์นั้นสามารถเรียนรู้ได้ด้วยตนเองแบบไม่ต้องมีผู้สอน โดยไม่จำเป็นต้องมีค่าเป้าหมายหรือผลลัพธ์ของแต่ละชุดข้อมูล ซึ่งวิธีการนี้คือการที่มนุษย์นั้นจะเป็นผู้ใส่ชุดข้อมูล และกำหนดสิ่งที่ต้องการจากชุดข้อมูลเหล่านั้น โดยให้เครื่องจักรหรือคอมพิวเตอร์วิเคราะห์จากการจำแนกและทำการสร้างแบบแผนจากข้อมูลที่ได้รับมา
3. Reinforcement Learning คือการเรียนรู้สิ่งต่างๆ โดยลองผิดลองถูกต่างๆ ภายใต้แนวคิดที่ว่าจะเลือกกระทำสิ่งใดที่ทำให้ได้ผลลัพธ์มากที่สุด โดยจะทำการเรียนรู้จากในสถานการณ์ในอดีตหรือระบบจำลองและพยายามที่จะพัฒนาระบบการตัดสินใจของตัวเองให้ดีขึ้นอย่างต่อเนื่อง โดยที่อาจจะสามารถพัฒนาด้วยการพยายามสร้างแบบจำลองสถานการณ์ต่าง ๆ ขึ้นมา



**รูป 2.1 ประเภทของ Machine Learning**

**(ที่มา :** [**medium.com**](https://medium.com/investic/machine-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-fa8bf6663c07)**, 2018)**

* + 1. **Extract-Transform-Load (ETL)**

Extract-Transform-Load คือ กระบวนการหนึ่งที่อยู่ในระบบของ Data Warehouse ซึ่งระบบนี้จะทำการออกแบบมาเพื่อให้สามารถดึงข้อมูลมาจากหลายๆแหล่งมาใช้งานได้ โดยจะนำกระบวนการตรวจสอบคุณภาพของชุดข้อมูลมาประยุกต์ร่วมใช้ ซึ่งมีการเชื่อมโยงและปรับชุดของข้อมูลให้เป็นไปในรูปแบบเดียวกันทั้งหมดเพื่อให้ ชุดของข้อมูลจากหลากหลายแหล่งสามารถใช้งานร่วมกันได้ และทำการส่งมอบ

* + - 1. **Extract**

เป็นกระบวนการเริ่มต้นของระบบที่ดึงข้อมูลจากแหล่งของข้อมูล จะประกอบด้วยข้อมูลจากหลากหลายแหล่งที่มา ข้อมูลที่อยู่ต่างที่กันนั้นอาจจะอยู่ในรูปแบบที่แตกต่างกัน ยกตัวอย่างเช่น อาจจะอยู่ในรูปแบบของฐานข้อมูลคนละชนิด หรือ ไม่ใช่ฐานข้อมูลแท้จริงซึ่งอาจจะเป็นระบบไฟล์ข้อมูลธรรมดา

* + - 1. **Transforming**

Transforming ขั้นตอนการแปลงรูปแบบของข้อมูลนี้จะมีการใช้กฎหรือฟังก์ชัน (Function) มากมายเพื่อที่จะแปลงข้อมูลให้ได้อยู่ในรูปแบบตามที่ต้องการก่อนที่จะนำข้อมูลเหล่านั้นเข้าไปยังปลายทาง ข้อมูลจากต้นทางนั้นบางแหล่งข้อมูลมีความจำเป็นน้อยมากหรือแทบจะไม่ต้องการ การแปลงข้อมูลเลย แต่ในบางแหล่งอาจจะต้องการกระบวนการที่ซับซ้อนในการแปลงข้อมูล ซึ่งจะกินทรัพยากรของระบบที่ใช้และเวลาในการประมวลผลของระบบ ซึ่งความซับซ้อนของข้อมูลจะขึ้นอยู่กับความต้องการของเชิงธุรกิจ หรือ เป้าหมายของการนำข้อมูลไปใช้งาน โดยจะมีกระบวนการตัวอย่างต่อไปนี้

1. Selection คือ การเลือก Column ที่ต้องการที่จะนำไปใช้งานหรือเก็บลงฐานข้อมูล โดยยกตัวอย่าง เช่น ถ้าต้นทางของข้อมูลมีอยู่ด้วยกัน 3 Column หรือ 3 attributes เช่น enroll\_num, age และ salary จะมีการแปลงข้อมูล เกิดขึ้นและ เลือกที่จะไม่มีการแปลงข้อมูลหากพบว่า record นั้นมีค่าของข้อมูล column salary เป็นค่าว่าง
2. Translation คือ การแปลข้อมูล ตัวอย่างเช่น หากข้อมูลต้นทางนั้นมีการเก็บข้อมูลของเพศโดยให้ 1 เป็นเพศชาย และ 2 เป็นเพศหญิง จะต้องมีการแปลจากชุดตัวเลขที่กำหนดก่อนหน้านี้ให้ 1 = Male และ 2 = Female กระบวนการนี้เรียกว่า data cleaning หรือ กระบวนการทำความสะอาดข้อมูล
3. Encoding free form การ mapping ค่าจากค่าหนึ่งสู่อีกค่าหนึ่งเพื่อให้ง่ายต่อการใช้งาน
4. Filtering คือ กระบวนการกรองเฉพาะข้อมูลที่กำหนด
5. Sorting คือ กระบวนการเรียงข้อมูลที่ต้องการ
6. Joining คือ กระบวนการเชื่อมโยงข้อมูลระหว่างตารางข้อมูล
7. Aggregation คือ การหาผลรวมของชุดข้อมูลหรือหาผลรวมของผลลัพธ์
8. Transposing or pivoting คือการสลับทิศทางตำแหน่งของการแสดงข้อมูล เช่นการย้ายระเบียบไปเป็น Column หรือ ย้าย Column มาเป็นระเบียบ เพื่อให้ง่ายต่อการนำข้อมูลไปใช้
   * + 1. **Loading**

Loading กระบวนการโหลดข้อมูลเข้าไปยังฐานข้อมูลเช่น นำข้อมูลเข้าไปในระบบ Data Warehouse ทั้งนี้ขึ้นอยู่กับความต้องการของ บางงานจะมีการสะสมของข้อมูล ความถี่ของการนำข้อมูลเข้าสู่ระบบ อาจจะมีการล้างข้อมูลแล้วทับข้อมูลใหม่

* + 1. **Natural Language Processing (NLP)**

Natural Language Processing (NLP) เป็นเครื่องมือที่ให้คอมพิวเตอร์เข้าใจภาษาของมนุษย์ที่มีความซับซ้อน เป็นศาสตร์หนึ่งที่สำคัญทางด้าน Machine Learning โดยเป็นสาขาวิชาหนึ่งที่ประกอบด้วยองค์ความรู้จากหลากหลายแขนง อาทิ ภาษาศาสตร์ (Linguistics) วิทยาการคอมพิวเตอร์ (Computer Science) ปัญญาประดิษฐ์ (Artificial Intelligence: AI) รวมไปถึงสถิติ (Statistics) โดย NLP มีมาตั้งแต่ช่วงกลางศตวรรษที่ 19 และมีการพัฒนามาเรื่อย ๆ จนถึงปัจจุบัน โดยแบ่งออกเป็น 3 ยุค ดังนี้

1. ยุค Rule-based Method (ช่วง ค.ศ.1950-1990)

ในยุคแรกของ NLP มีการใช้งานตามกฎ (Rule-based Method) โดยนักภาษาศาสตร์ผู้มีความเชี่ยวชาญโครงสร้างของภาษาที่สนใจ จะเป็นผู้เขียนกฎต่าง ๆ ขึ้นมาเพื่อให้คอมพิวเตอร์สามารถคำนวณข้อความของโจทย์ต่างๆได้

1. ยุค Machine Learning (ช่วง ค.ศ.1990-2010)

ในยุคนี้ พบว่ามีการเขียนกฎด้วยมือไม่สามารถตอบโจทย์ที่มีความซับซ้อนได้ จึงมีสิ่งที่ได้มาทดแทนในยุคนี้คือ ความสามารถของเครื่องคอมพิวเตอร์ รวมถึงความรู้ทางด้านสถิติ และ Machine Learning ซึ่งได้ถูกนำมาพัฒนาเพื่อใช้ในการทำงานด้าน NLP โดยมีการนำเข้าข้อมูลเพื่อให้คอมพิวเตอร์สามารถเรียนรู้ด้วยตนเองแทนการใช้ผู้เชี่ยวชาญทางด้านภาษา

1. ยุค Deep Learning (ช่วง ค.ศ.2010-ปัจจุบัน)

ในยุคปัจจุบัน ด้วยพลังการคำนวณของคอมพิวเตอร์ที่มีการพัฒนาสูงขึ้นอย่างต่อเนื่อง ทำให้เทคโนโลยีที่มีความซับซ้อนสูงอย่าง การเรียนรู้เชิงลึก (Deep Learning) ถูกนำมาใช้งานแทนที่ Machine Learning ซึ่งใช้ความรู้ทางด้านสถิติแบบดั้งเดิมอย่างแพร่หลายมากขึ้น รวมถึงในงานด้าน NLP ด้วยเช่นกัน อาทิ การสร้างแบบจำลองทางภาษา (Language Model) และการวิเคราะห์โครงสร้างของข้อความ (Parsing)

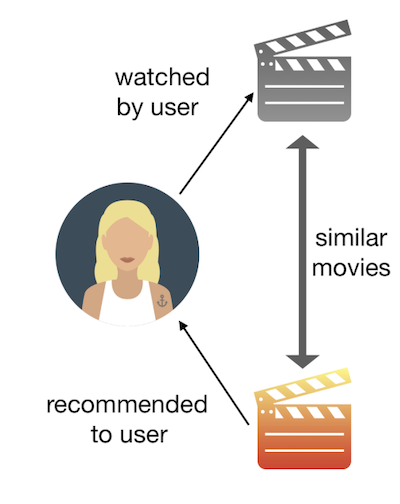
ตัวอย่างการประยุกต์ใช้ NLP ในด้านต่าง ๆ

1. ด้านการทำงานวิจัย การวิจัยมีแหล่งของข้อมูลทางภาษาขนาดใหญ่ ซึ่งทำให้ NLP สามารถเข้ามามีบทบาทได้อย่างหลากหลาย ตัวอย่างเช่น การใช้ Topic Model ในการจัดหมวดหมู่บทความ
2. ด้านพาณิชย์อิเล็กทรอนิกส์ การซื้อของผ่านช่องทางออนไลน์ เข้ามามีบทบาทสำคัญเป็นอย่างมากในระบบเศรษฐกิจ ซึ่งทำให้เกิดปริมาณธุรกรรมขนาดใหญ่ ไม่ว่าจะเป็น คำอธิบายสินค้าและบริการ การแสดงความคิดเห็นของผู้บริโภค รวมถึงการสนทนากันระหว่างผู้ซื้อและผู้ขายผ่านทางช่องแช็ต
3. ด้านการแพทย์ ข้อมูลทางการแพทย์มีการบันทึกข้อมูลด้วยข้อความตัวอย่างเช่น บทสนทนาระหว่างแพทย์และผู้ป่วย การวินิจฉัยโรคโดยแพทย์ และประวัติการรักษาของผู้ป่วย
4. ด้านกฎหมาย สำหรับงานด้าน มีข้อมูลทางด้านภาษาที่แตกต่างและหลากหลาย เช่นเดียวกัน เช่น ประมวลกฎหมายต่าง ๆ คำร้องต่อศาล คำให้การของคู่ความ และคำพิพากษาของศาล ซึ่งสามารถประยุกต์ใช้เครื่องมือ NLP ได้ในหลายมิติไม่ว่าจะเป็นการใช้ PoS Tagging และ NER เพื่อช่วยในการตีความประมวลกฎหมาย
   * 1. **Recommendation System**

Recommendation System เป็นระบบที่จะทำการแนะนำสิ่งที่ผู้ใช้งานมีความสนใจไม่ว่าจะเป็น เนื้อหา เพลง course เรียน ไปจนถึงสินค้าที่ขายในร้าน online โดยสามารถแนะนำได้ผ่านโมเดลที่ส่วนใหญ่มักจะถูกใช้กันมีอยู่ด้วยกันสามประเภท ได้แก่

* + - 1. **Content-based Filtering**

เป็นรูปแบบของโมเดลที่จะแนะนำลักษณะของตัวบริการหรือสินค้าเป็นตัวตั้งและทำการแนะนำสิ่งค้าที่มีลักษณะที่คล้ายกัน



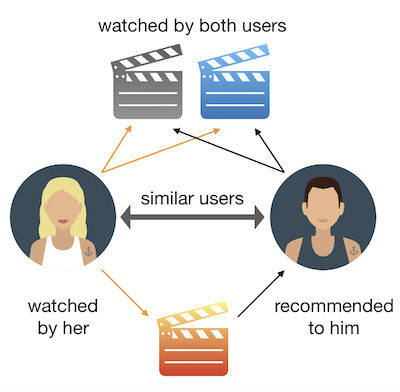
**รูป 2.2 รูปแบบของ Content-based Filtering**

**(ที่มา :** [towardsdatascience.com](https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243)**, 2018)**

* + - 1. **Collaborative Filtering**

เป็นรูปแบบโมเดลที่เรียนรู้จากพฤติกรรมของผู้ใช้กับผู้ใช้คนอื่น ๆ ที่คล้ายคลึงกัน

1. Memory-based เป็นการดูข้อมูลแล้วหาความสัมพันธ์ ระหว่างผู้ใช้หรือ สินค้าจากข้อมูลโดยตรง



**รูป 2.3 รูปแบบของ Memory-based**

**(ที่มา :** [towardsdatascience.com](https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243)**, 2018)**

1. Model-based ใช้เทคนิคของ machine learning เพื่อหา user embedding และ item embedding มาทำการทำนาย rating ที่ผู้ใช้จะให้กับสินค้า หรือ relevance score
2. Hybrid ใช้หลาย ๆ วิธีการมารวมกัน Hybrid system เป็นการมัดรวมทั้งสองอัลกอริทึมของ Model-based และ Memory-based เอาไว้เพื่อทำให้ระบบการแนะนำสมบูรณ์ขึ้น ซึ่งระบบนี้ถูกนำไปใช้ในปัจจุบันมากที่สุดแทบจะทุกแพลตฟอร์มใหญ่ที่มีการแนะนำสินค้าและบริการ
   * + 1. **Hybrid system**

เป็นการมัดรวมทั้งสองระหว่าง Content-based Filtering และ Collaborative Filtering เพื่อทำให้ระบบการแนะนำสมบูรณ์ขึ้น

* 1. **เครื่องมือที่เกี่ยวข้อง**
     1. **Docker**

Docker เป็นเครื่องมือแบบ open-source ที่ช่วยจำลองสภาพแวดล้อม ในการรัน service หรือ server โดยการสร้าง container เพื่อจัดการกับ library ต่างๆ และยังช่วยจัดการในเรื่องของ version control เพื่อให้ง่ายต่อการจัดการกับปัญหาต่างๆ ที่เกิดขึ้นองค์ประกอบต่างๆของ Docker

1. Docker image

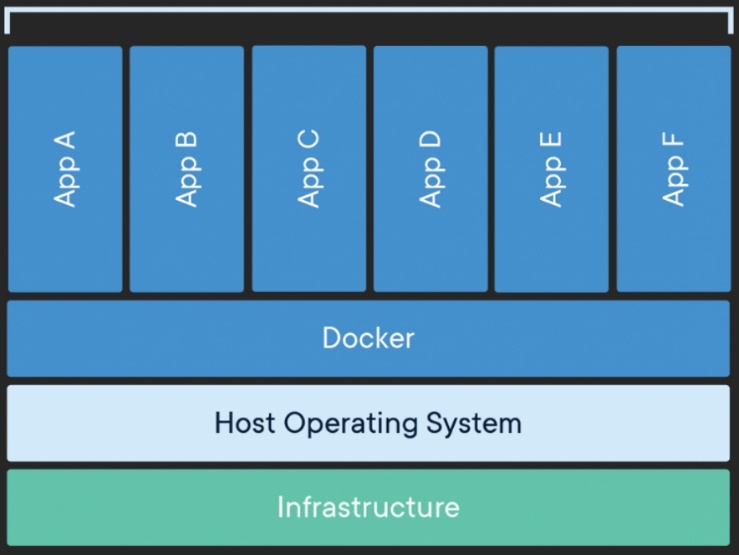
คือแม่พิมพ์หรือต้นแบบของ Container โดยจะเป็นระบบปฏิบัติการ Linux ที่มีการติดตั้ง Application และจะมีการ Configuration tool ต่างๆเอาไว้ ซึ่งเกิดจากการ build ไฟล์ Docker file ขึ้นมาเป็น image

1. Docker container

เป็นสิ่งที่ถูกสร้างขึ้นมาจาก Docker Image ที่เป็นต้นแบบหรือแม่พิมพ์ เกิดเป็น container และจะได้ Service หรือ Application ที่สามารถเรียกใช้งานได้ทันที

1. Docker registry

การสร้าง Docker Image แล้วนำไปเก็บรวบรวมไว้บน server ลักษณะเดียวกับการเก็บ Source Code ไว้บน (Github) โดยมี Docker Hub เป็น Docker registry หลักในการเรียกใช้(pull) Docker Image และนอกจากนี้ยังมีผู้ให้บริการ docker registry อื่นๆด้วย



**รูป 2.4 การทำงานของแอปพลิเคชันต่าง ๆ บน Docker Engine**

**(ที่มา :** [docker.com](https://docker.com/resources/what-container)**)**

* + 1. **Django**

Django Framework เป็นชุด Framework สำหรับ การพัฒนาเว็บไซต์ โดยใช้ภาษา Python ในการพัฒนา โดยทุกวันนี้ Framework สำหรับการเขียนเว็บไซต์ด้วยภาษา Python มีค่อนข้างที่จะเยอะ ซึ่ง Django Framework ก็เป็นหนึ่งใน Framework สำหรับการพัฒนาเว็บไซต์ และทำเว็บไซต์ด้วยภาษา Python ด้วยเช่นกัน

คุณสมบัติของ Django Framework

1. Object-relational mapper ใช้ในการทำงานด้านข้อมูล และช่วยสนับสนุน dynamic database-access API
2. Automatic admin interface ช่วยในการสร้าง Interface อัตโนมัติสำหรับการ add, edit, delete และ search
3. Elegant URL design ทำให้ URL มีความสั้น กระชับ สามารถสื่อความหมายของหน้านั้น ๆ ได้อย่างชัดเจน
4. Template system มีการออกแบบ Template Language เพื่อการเขียนแยกส่วนระหว่าง Design และ Business Logic
5. Cache system สามารถบันทึก หรือจัดการข้อมูลที่มีการดาวน์โหลดไปแล้ว เพื่อเพิ่มประสิทธิภาพการทำงานของเว็บไซต์ด้านความเร็ว และด้านอื่น ๆ
6. Internationalization คือ Django สนับสนุน Application ที่มีความหลากหลายด้านภาษาในการแสดงผล
   * 1. **Scikit-learn**

Scikit-learn เป็นแพ็กเกจที่รวบรวม Library ด้าน การเรียนรู้ของเครื่อง (Machine Learning) เอาไว้ โดยใช้ภาษา Python ในการพัฒนา และถูกออกแบบมาให้ทำงานร่วมกับ Library อย่าง NumPy และ SciPy

Scikit-learn ยังเป็น Open Source ที่เปิดให้สามารถเข้าไปพัฒนาต่อยอดได้และเป็นแหล่งรวม Library และอัลกอริทึมที่เน้นไปในด้านของ การเรียนรู้ของเครื่อง (Machine Learning) ซึ่งมีส่วนในการทำ แบบจำลองข้อมูล (Data Modeling) อีกหนึ่งปัจจัยที่ทำให้มีผู้ใช้เยอะ เพราะเป็น Interface ระดับสูง ทำให้มือใหม่สามารถเข้าใจภาพรวมและ ขั้นตอนการทำงาน ของการเรียนรู้ของเครื่อง (Machine Learning) ได้เครื่องมือที่ผู้ใช้งานสามารถนำไปใช้ได้

* + 1. **MariaDB**

MariaDB คือ เป็น Open Source สำหรับจัดการกับฐานข้อมูล MariaDB ถูกพัฒนาขึ้นโดยนักพัฒนาเดิมของ MySQL เนื่องจากความกังวลที่เกิดขึ้นเมื่อ MySQL ถูกซื้อโดย Oracle Corporation ในปี 2009

MariaDB ได้รับการพัฒนาเป็นซอฟต์แวร์โอเพนซอร์ส และเป็นฐานข้อมูลเชิงสัมพันธ์แบบ SQL สำหรับการเข้าถึงข้อมูล เวอร์ชันล่าสุดของ MariaDB มีคุณลักษณะ GIS และ JSON

MariaDB เปลี่ยนข้อมูลเป็นฐานข้อมูลที่มีโครงสร้างในหลากหลายแอปพลิเคชัน เป็นการปรับปรุงและแทนที่ด้วยการแทนที่ของ MySQL เพราะมีความรวดเร็วและสามารถปรับขนาดได้ มีระบบแวดล้อมที่อุดมไปด้วยปลั๊กอินส์เอนจินและเครื่องมืออื่น ๆ ทำให้สามารถใช้งานได้หลากหลาย

* + 1. **React**

React เป็น Library ของภาษา JavaScript โดยใช้สำหรับการสร้าง user interface ทำให้สามารถเขียนโค้ดในการสร้าง UI ที่มีความซับซ้อนแบ่งเป็นส่วนเล็กๆออกจากกันได้ ซึ่งแต่ละส่วนสามารถแยกการทำงานออกจากกันได้อย่างอิสระ และทำให้สามารถนำชิ้นส่วน UI เหล่านั้นไปใช้ซ้ำได้

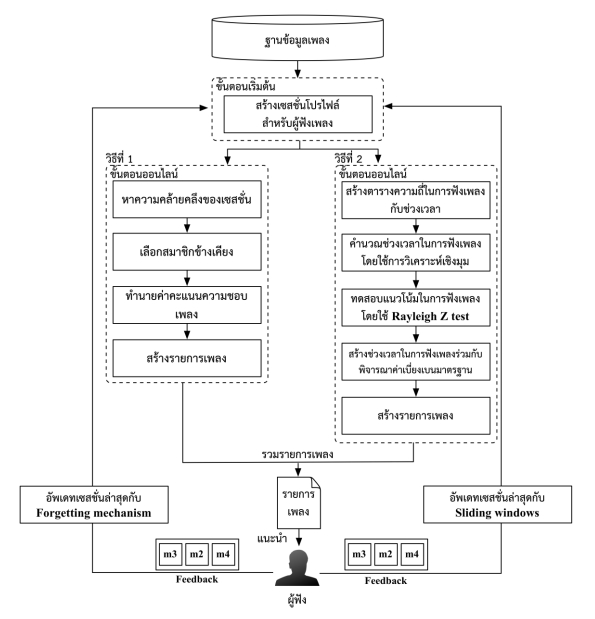
* + 1. **Node.JS**

Node.js เป็น open-source และ cross-platform ของภาษา JavaScript โดยทำงานแบบ asynchronous สามารถใช้ Node.js ในการพัฒนาแอปพลิเคชันแบบ Command line แอปพลิเคชัน Desktop หรือแม้แต่เว็บเซิร์ฟเวอร์ได้ โดยที่ Node.js จะมี APIs ที่จะสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก และยังเป็นเป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows, Linux และ Mac OS X โดยสามารถเขียนโปรแกรมในภาษา JavaScript และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดย Node.js

* 1. **งานวิจัย**ที่เกี่ยวข้อง
     1. **การสร้างรายการเพลงโดยใช้การกรองร่วมแบบเซสชั่นที่เพิ่มขึ้นด้วยกลไกการลืมและการวิเคราะห์สถิติเชิงมุม**

สุเมธ ดาราพิสุท นำเสนองานวิจัยเรื่อง การสร้างรายการเพลงโดยใช้การกรองร่วมแบบเซสชั่นที่เพิ่มขึ้นด้วยกลไกการลืมและการวิเคราะห์สถิติเชิงมุม โดยใช้ 2 วิธีร่วมกัน 1 การสร้างรายการเพลงจะพิจารณาการฟังเพลงในเซสชั่นปัจจุบันที่คล้ายกับเซสชั่นในอดีตของผู้ฟัง 2 สร้างรายการเพลงแนะนำโดยพิจารณาช่วงเวลา เฉพาะในการฟังเพลงซึ่งแตกต่างจากช่วงเวลาอื่นอย่างมีนัยสำคัญทางสถิติในรอบวันของผู้ฟังโดยใช้ การวิเคราะห์สถิติเชิงมุม และวัดประสิทธิภาพโดย ประสิทธิภาพ Hit Ratio และ Precision จากการทดลองพบว่าการใช้ 2 วิธีแยกกันนั้นได้ผลลัพธ์ที่น้อยกว่านำมาใช้ร่วมกัน 0.18-0.22 % โดยวัตถุประสงค์ในการทำเพื่อวิเคราะห์ปัญหาที่เกิดขึ้นในการสร้างรายการเพลงแนะนำแบบออฟไลน์ พัฒนาขั้นตอนวิธีการสร้างรายเพลงแนะนำให้มีประสิทธิภาพเพิ่มขึ้นทั้งทางด้านความเร็วและความถูกต้องในการสร้างรายการเพลง

โดยในโครงงานของผู้จัดทำนั้นได้นำสวนของการออกแบบ Diagram ในงานวิจัยนี้มาใช้งาน โดยใช้วิธีที่ 1ซึ่งของผู้จัดทำจะเป็น 1 หาความคล้ายคลึงของหมวดหมู่วิชา 2 เลือกสมาชิกข้างเคียง 3 ทำนายค่าผลลัพธ์การเรียนหรือเกรด 4 นำไปสร้างรายการสำหรับขั้นตอนต่อไป

**รูป 2.5 ประเภทของ Machine Learning**

**(ที่มา :** ดาราพิสุทธิ์, 2016**)**

* + 1. **ระบบแนะนำสินค้าอาหารโดยใช้ระบบแนะนำแบบผสมผสาน**

นิภาภรณ์ พันธ์นาม นำเสนองานวิจัย ระบบแนะนำสินค้าอาหารโดยใช้ระบบแนะนำแบบผสมผสาน ใช้เทคนิค Content based filtering แบบหลักการ Cosine และสร้างแบบจำลองโดยใช้ lib Surprise ซึ่งมีอัลกอริทึม SVD, NMF, Baseline และ KNN และวัดประสิทธิภาพโดย RMSE, MAE จากการทดลองพบว่า 1 เทคนิคการกรองแบบอิงเนื้อหานำวิธีการ TF-IDF เข้ามาช่วยในการทำ Vectorization ส่วนใหญ่ค่าความเหมือนออกมาค่อนข้างที่จะตำเนื่องมาจากข้อมูลที่น้อยเกินไป 2 เทคนิคการกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วม ผ่าน library Surprise ของ Scikitlearn ซึ่งโมเดลที่มีผลคะแนนโดยรวมดีที่สุดคืออัลกอริทึมของ SVD ซึ่ง ได้ค่ำ RMSE 1.2528 และ MAE 0.9376 และ 3 ระบบแนะนำแบบผสมผสาน โดยผลลัพธ์นั้นจะไม่ชัดเจนเนื่องจากวิธีนี้ได้มีการทำนายค่า Rating ซึ่งวิธีการของระบบแนะนำแบบผสมผสานนั้น ได้มีนำเทคนิคการกรองแบบอิงเนื้อหา ที่ไม่ได้มีการทำนายค่าอะไรมารวมในการทำงานของแบบจำลองด้วย ซึ่งถ้าต้องการวัดผลลัพธ์สามารถอ้างอิงจากค่า RMSE, MAE ได้

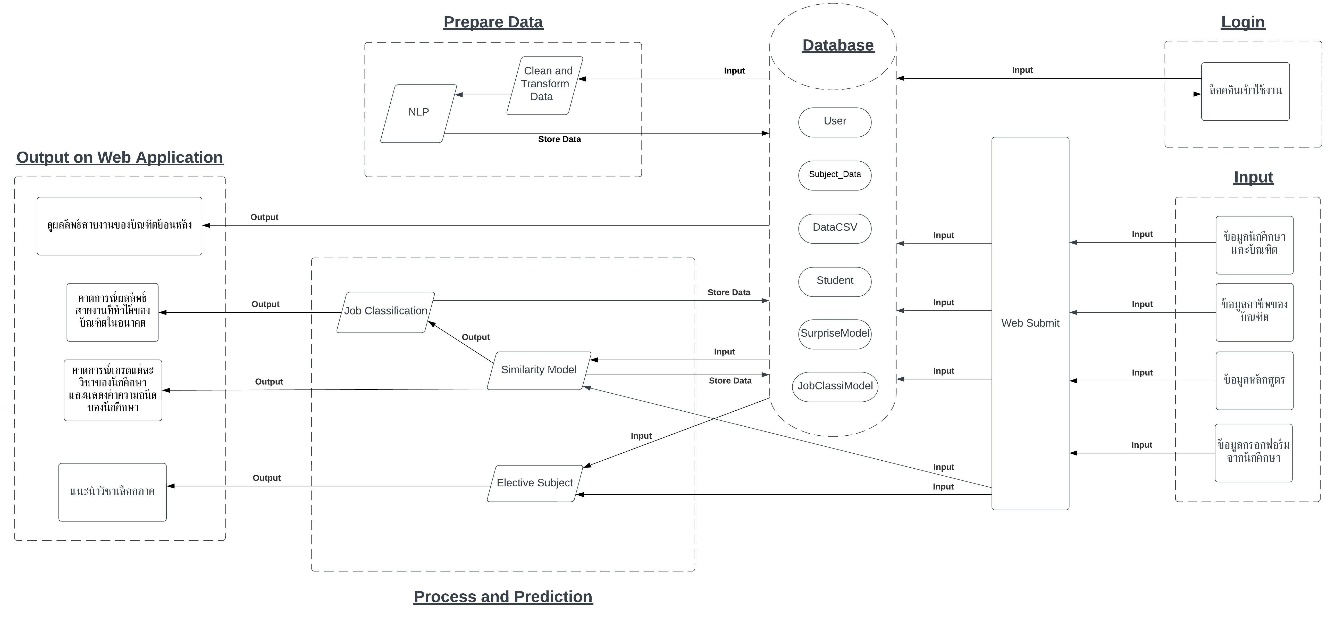
โดยในงานโครงงานของผู้จัดทำนั้นได้นำผลลัพธ์การทดลองของงานวิจัยนี้ที่สรุปได้ว่าผลคะแนนโดยรวมดีที่สุดคืออัลกอริทึม SVD เป็นตัวตัดสินในการเลือกใช้อัลกอริทึมนี้และได้นำวิธีการการวัดประสิทธิภาพของแบบจำลองนี้จากงานวิจัยมาปรับใช้ในรูปแบบเดียวกันกับตัวโครงงาน

**บทที่ 3**

**การออกแบบ**

* 1. โครงสร้างการทำงานของระบบ

โครงสร้างการทำงานของระบบได้อธิบายถึงการเชื่อมต่อระหว่างส่วนต่างๆของระบบ เริ่มตั้งแต่ส่วนของ Input ที่ทำหน้าที่รับข้อมูลเข้ามา แล้วเก็บไว้ในส่วนของ Database จากนั้นส่วนของ Prepare Data จะนำข้อมูลจากส่วนของ Database เมื่อทำเสร็จแล้ว จะทำการส่งกลับไปอัปเดตยัง Database ส่วนของ Process and Prediction จะนำข้อมูลที่ได้ไปเข้าอัลกอริทึมเพื่อ Process ผลลัพธ์ออกมาแสดงผลบน Web Application

****

**รูป 3.1 โครงสร้างการทำงานของระบบ**

* 1. การทำงานของระบบ

จากรูป 1 โครงสร้างการทำงานของระบบนั้นประกอบไปด้วยองค์ประกอบทั้งหมด 6 ส่วน ซึ่งมีรายละเอียด ดังนี้

* + 1. **Login**

เป็นส่วนสำหรับไว้ให้กรรมการหลักสูตรได้ทำการล็อกอินเข้าสู่ระบบเพื่อให้กรรมการหลักสูตรป้อนข้อมูลเกรดของนักศึกษาปัจจุบัน เกรดและอาชีพของบัณฑิต และข้อมูลของหลักสูตร

* + 1. **Input**

เป็นส่วนที่ทำการรับข้อมูลของนักศึกษาและบัณฑิต ข้อมูลของหลักสูตร แล้วเก็บเข้ายังส่วนของ Database และข้อมูลการกรอกฟอร์มของนักศึกษาจะส่งข้อมูลไปยังส่วนของ Process and Prediction โดยตรง

* + 1. **Database**

ทำหน้าที่ในการจัดเก็บและบันทึกข้อมูล โดยจะประกอบไปด้วย Table User, Subject\_Data, DataCSV, Graduate, Student

* + 1. **Prepare Data**

ทำหน้าที่เตรียมพร้อมข้อมูลเพื่อให้พร้อมต่อการนำไปใช้ในส่วนของ Prediction and Prediction โดยแบ่งออกเป็น 2 ส่วนย่อย ดังนี้

1. Clean and Transform Data

เป็นการเรียกข้อมูลจากใน Database มาทำให้ข้อมูลมีความถูกต้อง ครบถ้วน สมบูรณ์ ไม่มีค่าที่ผิดปกติ เพื่อเตรียมพร้อมสำหรับการประมวลผลข้อมูล โดยเลือกเฉพาะข้อมูลที่เกี่ยวข้องกับวัตถุประสงค์ของการวิเคราะห์ และ ประมวลผล พร้อมทั้งตัดข้อมูลส่วนที่ไม่ได้นำมาใช้ออก

1. NLP หรือ Natural Language Processing

นำข้อมูลรหัสวิชามาเข้ากระบวนการ NLP เพื่อทำการหาค่า Similarity ของกลุ่มวิชาที่สามารถอยู่ในกลุ่มเดียวกันได้ โดยใช้บทคัดย่อของแต่ละวิชา เพื่อลดปัญหาการเปลี่ยนรหัสวิชาระหว่างหลักสูตร เมื่อทำเสร็จกระบวนการแล้วจะนำข้อมูลที่ได้กลับไปอัปเดตที่ Database

* + 1. **Process and Prediction**

เป็นส่วนการประมวลผลหลักของระบบ ประกอบไปด้วย Process 3 ส่วนดังนี้

1. Similarity Model

มีหน้าที่ในการทำนายผลลัพธ์ของเกรดในรายวิชาที่ไม่ได้มีการลงทะเบียนของผู้ใช้งานตามที่ผู้ใช้งานต้องผ่านการเทียบความคล้ายคลึงของ User Behavior จาก File ผลลัพธ์เกรดที่ผู้ใช้งานได้นำเข้ามาในระบบ โดยใช้กระบวนการหลักคือการทำ Recommender Systems และใช้ Library Surprise ของ Scikit ในภาษา Python เป็นตัวช่วยในการ เทียบความคล้ายคลึงของ User Behavior

1. Job Classification

มีหน้าที่ทำนายผลลัพธ์กลุ่มของสายงานอาชีพที่เป็นไปได้ของผู้ใช้งานหรือกลุ่มของนักศึกษาที่มีข้อมูลในระบบผ่านการวิเคราะห์เชิงสธิติ โดยใช้กระบวนการหลักคือการทำ Classification และใช้ Library Scikit-learn ในภาษา Python เป็นตัวช่วยในการทำ Classification

1. Elective Subject

มีหน้าที่ประมวลผลจัดกลุ่มของวิชาเลือกภาค เพื่อนำไปแนะนำให้แก่นักศึกษาที่มีความสนใจเฉพาะจุดได้

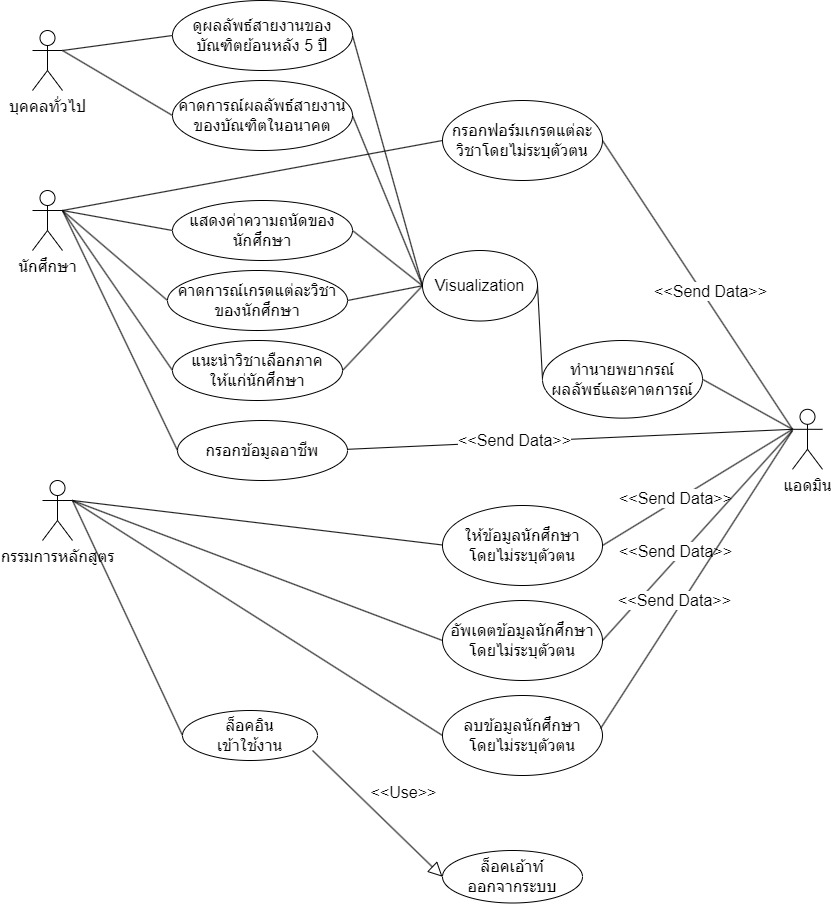
* + 1. **Web Application**

เป็นส่วนที่ทำหน้าที่ในการแสดงผลข้อมูลและเป็น interface สำหรับผู้ใช้งาน

* 1. Use Case Diagram

การใช้งานระบบจะแบ่งผู้ใช้ออกเป็น 4 ประเภท ได้แก่ บุคคลทั่วไปหรือนักศึกษา กรรมการหลักสูตร และ แอดมิน

1. บุคคลทั่วไปหรือนักศึกษา มีสิทธิ์เข้าถึงการดูข้อมูลการทำนายผล
2. กรรมการหลักสูตร มีสิทธ์เข้าถึงในการส่งไฟล์ข้อมูลของนักศึกษา
3. แอดมิน มีสิทธ์เข้าถึงการทำงานทั้งหมดของระบบวิเคราะห์ และ พยากรณ์

****

**รูป 3.2 Use Case Diagram**

**ตาราง 3.1 Use Case ดูผลลัพธ์สายงานของบัณฑิตย้อนหลัง**

|  |
| --- |
| **Use Case:** ดูผลลัพธ์สายงานของบัณฑิตย้อนหลัง |
| **Use Case ID:** UC-01 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถดูผลลัพธ์สายงานของบัณฑิตย้อนหลังได้ |
| **Precondition:** เข้า web application หน้าผลลัพธ์สายงานของบัณฑิตย้อนหลัง |
| **Flow of Events:**   1. เลือกหน้าดูผลลัพธ์สายงานของบัณฑิตย้อนหลัง 2. เลือกปีที่ต้องการดู 3. เลือกหลักสูตรที่ต้องการดู |
| **Postcondition:** หน้าเว็บแสดงผลผลลัพธ์สายงานของบัณฑิตย้อนหลัง |

**ตาราง 3.2 Use Case คาดการณ์ผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต**

|  |
| --- |
| **Use Case:** คาดการณ์ผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต |
| **Use Case ID:** UC-02 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถดูผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคตได้ |
| **Precondition:** เข้า web application หน้าผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต |
| **Flow of Events:**   1. เลือกหน้าดูผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต 2. เลือกปีที่ต้องการดู 3. เลือกหลักสูตรที่ต้องการดู |
| **Postcondition:** หน้าเว็บแสดงผลผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต |

**ตาราง 3.3 Use Case กรอกแบบฟอร์มเกรดสำหรับการวิเคราะห์ข้อมูล**

|  |
| --- |
| **Use Case:** กรอกแบบฟอร์มเกรดสำหรับการวิเคราะห์ข้อมูล |
| **Use Case ID:** UC-03 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถกรอกแบบฟอร์มการวิเคราะห์ข้อมูลได้ |
| **Precondition:** เข้า web application หน้าสำหรับการวิเคราะห์ข้อมูลของนักศึกษา |
| **Flow of Events:**   1. เลือกหน้าสำหรับการวิเคราะห์ข้อมูลของนักศึกษา 2. เลือกปีการศึกษา 3. เลือกหลักสูตร 4. โหลดแบบฟอร์มสำหรับการวิเคราะห์ข้อมูล 5. อัปโหลดแบบฟอร์มสำหรับการวิเคราะห์ข้อมูล |
| **Postcondition:** Redirect ไปหน้าแสดงผลวิเคราะห์ข้อมูลนักศึกษา |

**ตาราง 3.4 Use Case แสดงค่าความถนัดของนักศึกษา**

|  |
| --- |
| **Use Case:** แสดงค่าความถนัดของนักศึกษา |
| **Use Case ID:** UC-04 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถดูผลแสดงค่าความถนัดของนักศึกษาได้ |
| **Precondition:** เข้า web application หน้าแสดงผลวิเคราะห์ข้อมูลนักศึกษา |
| **Flow of Events:**   1. เลือกหน้าแสดงผลสำหรับการวิเคราะห์ข้อมูลของนักศึกษา |
| **Postcondition:** หน้าเว็บแสดงผลผลลัพธ์การวิเคราะห์ข้อมูลค่าความถนัดของนักศึกษา |

**ตาราง 3.5 Use Case คาดการณ์เกรดแต่ละวิชาของนักศึกษา**

|  |
| --- |
| **Use Case:** คาดการณ์เกรดแต่ละวิชาของนักศึกษา |
| **Use Case ID:** UC-05 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถดูผลแสดงค่าคาดการณ์เกรดแต่ละวิชาของนักศึกษาได้ |
| **Precondition:** เข้า web application หน้าแสดงผลวิเคราะห์ข้อมูลนักศึกษา |
| **Flow of Events:**   1. เลือกหน้าแสดงผลสำหรับการวิเคราะห์ข้อมูลของนักศึกษา |
| **Postcondition:** หน้าเว็บแสดงผลผลลัพธ์การวิเคราะห์ข้อมูลค่าคาดการณ์เกรดแต่ละวิชาของนักศึกษา |

**ตาราง 3.6 Use Case แนะนำวิชาเลือกภาคให้แก่นักศึกษา**

|  |
| --- |
| **Use Case:** แนะนำวิชาเลือกภาคให้แก่นักศึกษา |
| **Use Case ID:** UC-06 |
| **Actor:** บุคคลทั่วไปหรือนักศึกษา |
| **Description:** สามารถดูผลแนะนำวิชาเลือกให้ภาคให้แก่นักศึกษาได้ |
| **Precondition:** เข้า web application หน้าแนะนำวิชาเลือกภาค |
| **Flow of Events:**   1. เลือกหน้าแนะนำวิชาเลือกภาค 2. เลือกสิ่งที่สนใจภายในตัวเลือกที่มีให้ |
| **Postcondition:** หน้าเว็บแนะนำวิชาเลือกภาค |

**ตาราง 3.7 Use Case ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร**

|  |
| --- |
| **Use Case:** ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Use Case ID:** UC-07 |
| **Actor:** กรรมการหลักสูตร |
| **Description:** ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตรได้ |
| **Precondition:** เข้า web application หน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Flow of Events:**   1. เลือกหน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร 2. อัปโหลดไฟล์ CSV 3. กดปุ่มอัปโหลด |
| **Postcondition:** อัปโหลดข้อมูลสำเร็จ |

**ตาราง 3.8 Use Case อัปเดตข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร**

|  |
| --- |
| **Use Case:** อัปเดตข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Use Case ID:** UC-08 |
| **Actor:** กรรมการหลักสูตร |
| **Description:** อัปเดตนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตรได้ |
| **Precondition:** เข้า web application หน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Flow of Events:**   1. เลือกหน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร 2. อัปเดตไฟล์ CSV 3. กดปุ่มอัปโหลด |
| **Postcondition:** อัปเดตข้อมูลสำเร็จ |

**ตาราง 3.9 Use Case ลบข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร**

|  |
| --- |
| **Use Case:** อัปเดตข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Use Case ID:** UC-09 |
| **Actor:** กรรมการหลักสูตร |
| **Description:** ลบนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตรได้ |
| **Precondition:** เข้า web application หน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Flow of Events:**   1. เลือกหน้าให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร 2. ลบไฟล์ CSV 3. กดปุ่มยืนยัน |
| **Postcondition:** ลบข้อมูลสำเร็จ |

**ตาราง 3.10 Use Case ล็อกอินเข้าใช้งาน**

|  |
| --- |
| **Use Case:** ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร |
| **Use Case ID:** UC-10 |
| **Actor:** กรรมการหลักสูตร |
| **Description:** กรรมการหลักสูตรสามารถล็อกอินเข้าใช้งานได้ |
| **Precondition:** เข้า web application หน้า login |
| **Flow of Events:**   1. กรอกข้อมูล username และ password 2. กดปุ่ม Login |
| **Postcondition:** Redirect ไปหน้าหลัก |

**ตาราง 3.11 Use Case ทำนายพยากรณ์ผลลัพธ์และคาดการณ์**

|  |
| --- |
| **Use Case:** ทำนายพยากรณ์ผลลัพธ์และคาดการณ์ |
| **Use Case ID:** UC-11 |
| **Actor:** แอดมิน |
| **Description:** แอดมินสามารถทำนายพยากรณ์ผลลัพธ์และคาดการณ์ได้ |
| **Precondition:** เป็นแอดมิน |
| **Flow of Events:**   1. ทำการทำนายพยากรณ์และคาดการณ์ 2. ส่งออกผลลัพธ์ไปยังหน้าเว็บ |
| **Postcondition:** ทำนายพยากรณ์ผลลัพธ์และคาดการณ์ได้สำเร็จ |

**ตาราง 3.12 Use Case จัดเตรียมข้อมูล**

|  |
| --- |
| **Use Case:** จัดเตรียมข้อมูล |
| **Use Case ID:** UC-12 |
| **Actor:** แอดมิน |
| **Description:** จัดเตรียมข้อมูลสำหรับการทำนายและคาดการณ์ได้ |
| **Precondition:** เป็นแอดมิน |
| **Flow of Events:**   1. นำเข้าข้อมูล 2. จัดรูปแบบข้อมูล 3. เก็บเข้าฐานข้อมูล |
| **Postcondition:** จัดเตรียมข้อมูลสำหรับการทำนายและคาดการณ์ได้สำเร็จ |

**ตาราง 3.13 Use Case กรอกข้อมูลอาชีพ**

|  |
| --- |
| **Use Case:** กรอกข้อมูลอาชีพ |
| **Use Case ID:** UC-13 |
| **Actor:** นักศึกษา |
| **Description:** กรอกข้อมูลอาชีพแรกของการทำงานได้ |
| **Precondition:** เข้า web application หน้ากรอกข้อมูลอาชีพ |
| **Flow of Events:**   1. เลือกหน้ากรอกข้อมูลอาชีพ 2. ใส่รหัสนักศึกษา 3. ใส่ตำแหน่งอาชีพ 4. กดยืนยัน |
| **Postcondition:** กรอกข้อมูลอาชีพสำเร็จ |

* 1. Sequence **Diagram**

****

**รูป 3.3 Sequence Diagram สำหรับ UC-01**

**Diagram

Description automatically generated**

**รูป 3.4 Sequence Diagram สำหรับ UC-02**

**Diagram

Description automatically generated**

**รูป 3.5 Sequence Diagram สำหรับ UC-03**

**Diagram

Description automatically generated**

**รูป 3.6 Sequence Diagram สำหรับ UC-04**

**Diagram

Description automatically generated**

**รูป 3.7 Sequence Diagram สำหรับ UC-05**

**Diagram

Description automatically generated**

**รูป 3.8 Sequence Diagram สำหรับ UC-06**

**Diagram

Description automatically generated**

**รูป 3.9 Sequence Diagram สำหรับ UC-07**

**Diagram

Description automatically generated**

**รูป 3.10 Sequence Diagram สำหรับ UC-08**

**Diagram

Description automatically generated**

**รูป 3.11 Sequence Diagram สำหรับ UC-09**

**Diagram

Description automatically generated**

**รูป 3.12 Sequence Diagram สำหรับ UC-10**

**Diagram

Description automatically generated**

**รูป 3.13 Sequence Diagram สำหรับ UC-11**

**Diagram

Description automatically generated**

**รูป 3.14 Sequence Diagram สำหรับ UC-12**

**Diagram

Description automatically generated**

**รูป 3.15 Sequence Diagram สำหรับ UC-13**

* 1. **การออกแบบฐานข้อมูล**

ระบบฐานข้อมูลของโครงงานนี้ผู้จัดทำได้เลือกใช้ MariaDB ซึ่งเป็นฐานข้อมูลแบบ SQL ที่ถูกพัฒนามาเพื่อสำหรับเก็บข้อมูล โดยฐานข้อมูลของระบบประกอบไปด้วยทั้งหมด 5 ตาราง ดังรูป

Table

Description automatically generated

**รูป 3.16 แสดงความสัมพันธ์ระหว่างตารางของฐานข้อมูลในระบบ**

* + 1. **ตาราง Rec\_User**

เป็นตารางที่ใช้เก็บข้อมูลผู้ใช้งานในระบบ โดยจะมี Attribute ดังนี้

1. user\_id : UUID สำหรับเก็บ id ของผู้ใช้

2. username : VARCHAR สำหรับเก็บ username ของผู้ใช้

3. password : VARCHAR สำหรับเก็บ password ของผู้ใช้

* + 1. **ตาราง Subject\_Data**

เป็นตารางที่ใช้เก็บข้อมูลหลักสูตรวิชา โดยจะมี Attribute ดังนี้

1. subject\_id : UUID สำหรับเก็บ id ของรหัสวิชา

2. subject\_name\_thai : VARCHAR สำหรับเก็บชื่อรายวิชาภาษาไทย

3. subject\_name\_eng : VARCHAR สำหรับเก็บชื่อรายวิชาภาษาอังกฤษ

4. abstract : VARCHAR สำหรับเก็บบทคัดย่อรายวิชา

5. subject\_key : VARCHAR สำหรับเก็บ Keyword ของรายวิชาเลือกภาค

6. subject\_class : VARCHAR สำหรับเก็บหมวดหมู่ของแต่ละวิชา

7. year : VARCHAR สำหรับเก็บปีของเล่มหลักสูตรวิชา

* + 1. **ตาราง Student\_Grade**

เป็นตารางที่ใช้เก็บข้อมูลเกรดนักศึกษาโดยจะมี Attribute ดังนี้

1. student\_id : UUID สำหรับเก็บ id encrypt ของนักศึกษา

2. subject\_id : VARCHAR สำหรับเก็บ id ของรหัสวิชา

3. grade : VARCHAR สำหรับเก็บเกรดแต่ละรายวิชา

4. semester : VARCHAR สำหรับเก็บเทอมที่ลงเรียนรายวิชาของนักศึกษา

5. year : VARCHAR สำหรับเก็บปีที่ลงเรียนรายวิชาของนักศึกษา

6 curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

7. career : VARCHAR สำหรับเก็บรายชื่อตำแหน่งอาชีพแรกของนักศึกษา

8. status : VARCHAR สำหรับเก็บสถานะการเป็นนักศึกษา

* + 1. **ตาราง Student\_Data**

เป็นตารางที่ใช้เก็บข้อมูลนักศึกษาโดยจะมี Attribute ดังนี้

1. student\_id : UUID สำหรับเก็บ id encrypt ของนักศึกษา

2 curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

3. status : VARCHAR สำหรับเก็บสถานะการเป็นนักศึกษา

4. career : VARCHAR สำหรับเก็บรายชื่อตำแหน่งอาชีพแรกของนักศึกษา

5. start\_year : VARCHAR สำหรับเก็บปีการศึกษาของนักศึกษา

6. curriculum\_year : VARCHAR สำหรับเก็บปีของหลักสูตรของนักศึกษา

* + 1. **ตาราง CSV\_File**

เป็นตารางที่ใช้เก็บข้อมูลไฟล์ CSV โดยจะมี Attribute ดังนี้

1. data\_id : UUID สำหรับเก็บ id ของไฟล์ CSV

2. name : VARCHAR สำหรับเก็บชื่อของไฟล์ CSV

3. upload\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปโหลดไฟล์ CSV

4. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดตไฟล์ CSV

5. del\_flag : VARCHAR สำหรับเก็บสถานะไฟล์ CSV (0 = ไฟล์ยังมีอยู่, 1 =ไฟล์ถูกลบ)

6. type\_data : VARCHAR สำหรับเก็บประเภทข้อมูลของ Data

7. file : BINARY สำหรับเก็บไฟล์ CSV

* + 1. **ตาราง SurpriseModel**

เป็นตารางที่ใช้เก็บ Surprise Model โดยจะมี Attribute ดังนี้

1. model\_id : UUID สำหรับเก็บ id ของไฟล์ Model

2. name : VARCHAR สำหรับเก็บชื่อของ Model

3. curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

4. year : VARCHAR สำหรับเก็บปีของหลักสูตรของชุดข้อมูลที่ใช้สร้าง

5. rmse : VARCHAR สำหรับเก็บค่าความแม่นยำของ Model

6. create\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่สร้าง Model

7. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดต Model

8. del\_flag : VARCHAR สำหรับเก็บสถานะ Model (0 = Modelยังมีอยู่, 1 =Modelถูกลบ)

9. type\_pred : VARCHAR สำหรับเก็บประเภทของ Model

10. rec\_model : BINARY สำหรับเก็บตัว Model

* + 1. **ตาราง CareerModel**

เป็นตารางที่ใช้เก็บ Job Classification Model โดยจะมี Attribute ดังนี้

1. model\_id : UUID สำหรับเก็บ id ของไฟล์ Model

2. name : VARCHAR สำหรับเก็บชื่อของ Model

3. curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

4. year : VARCHAR สำหรับเก็บปีของหลักสูตรของชุดข้อมูลที่ใช้สร้าง

5. accuracy : VARCHAR สำหรับเก็บค่าความแม่นยำของ Model

6. create\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่สร้าง Model

7. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดต Model

8. del\_flag : VARCHAR สำหรับเก็บสถานะ Model (0 = Modelยังมีอยู่, 1 =Modelถูกลบ)

9. train\_set\_cols : BINARY สำหรับเก็บ Columns ของชุดข้อมูลที่ใช้สร้าง Model

10. career\_model : BINARY สำหรับเก็บตัว Model

**บทที่ 4**

**ผลการดำเนินงาน**

ในบทนี้จะอธิบายถึงการทดลองที่ได้ทำในภาคการศึกษานี้ เพื่อแสดงให้เห็นถึงความก้าวหน้าในโครงงาน

* 1. การสร้าง Django Framework และเชื่อมต่อฐานข้อมูล MariaDB บน Docker Compose

โดยการทดลองการสร้างจะตัว Django Project ให้เชื่อมต่อกับ Maria DB ได้นั้น ทางผู้จัดทำได้ทำการเตรียม requirements.txt ซึ่งเป็น File เพื่อให้ Docker นั้นจัดเตรียม Environment ของตัวระบบดังรูปที่ 4.1



**รูป 4.1 File Requirements**

หลังจากได้มีการเตรียม File Requirements แล้วทางผู้จัดทำได้ทำการสร้าง Docker File เพื่อนำ Requirements ที่ได้มาจัดสรร Library ที่จำเป็นต่อการพัฒนาโครงงาน โดยได้กำหนด Version ของ Python เป็น 3.7.14 ดังรูปที่ 4.2

Text

Description automatically generated

**รูป 4.2 Docker file**

ต่อมาผู้จัดทำได้สร้าง File Docker Compose ขึ้นมาเพื่อนำ Docker Container ในส่วนของ Django Application และ MariaDB นำมารวมเป็น Docker Compose โดยกำหนด Internal Port ของ MariaDB เป็น 3306 และ External Port เป็น 3308 เพื่อจัดระเบียบในการใช้งานของฐานข้อมูล โดยทางผู้จัดทำออกแบบให้ Django Application ติดต่อกับฐานข้อมูลผ่าน Internal Port โดยกำหนด Port ของ Django Application เป็น 8000 ทั้ง Internal และ External Port ดังรูปที่ 4.3

Text

Description automatically generated

**รูป 4.3 File Docker - compose.yml**

ซึ่งเมื่อหลังจากการสร้าง Docker Compose ขึ้นมาทางผู้จัดทำจะตรวจสอบการทำงานของระบบทั้งหมดผ่าน Application Docker Desktop ร่วมกับ Log Terminal ของ Visual Studio Code ดังรูปที่ 4.4

Graphical user interface, text

Description automatically generated

**รูป 4.4 ภาพรวมของการตรวจสอบการทำงานของระบบ**

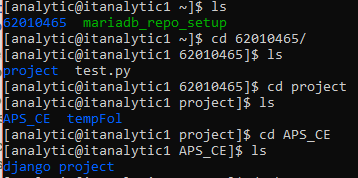
จากผลการทดลองที่ 4.1 พบว่าการการสร้าง Django Framework และเชื่อมต่อฐานข้อมูล MariaDB บน Docker Compose พบว่าการทำงานของ Container ทั้ง Django Application และ MariaDB Database นั้นทำงานร่วมกันเป็นไปอย่างราบรื่น และการตรวจสอบการทำงานของ Docker Compose ผ่าน Docker Desktop นั้นทำให้การทำงานและการตรวจสอบได้สะดวกยิ่งขึ้น

* 1. การเตรียม Server ที่ใช้สำหรับ Deploy Application

โดยทางผู้จัดทำได้ทำการทดลองการใช้งาน Server สำหรับ Deploy โดยได้ทำการสร้าง Directory ของผู้ทดลองขึ้นมาและได้ทำการทดลองติดตั้ง Program ที่จำเป็นในการรองรับการทำงานของ Application มี Python, MariaDB

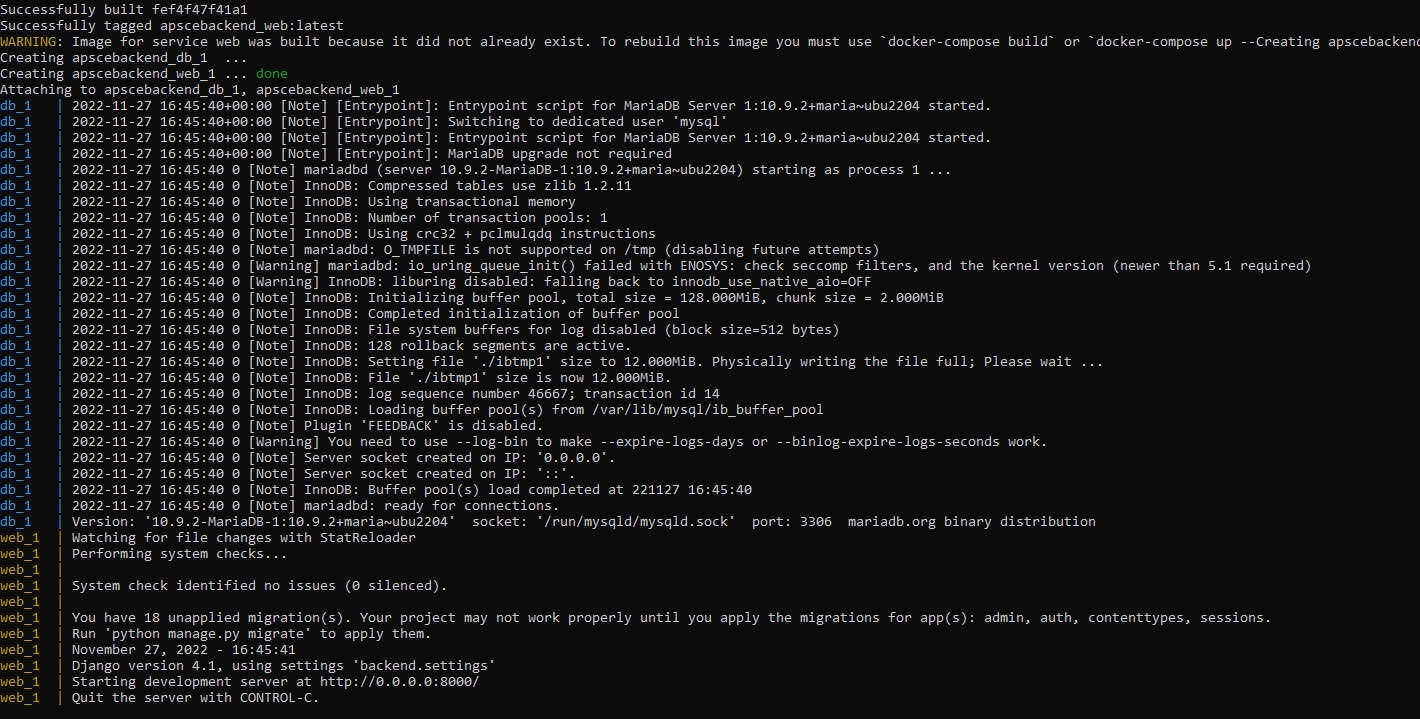
ผลการทดลองพบว่าไม่สามารถติดตั้ง Python Version 3.10.2 ที่ผู้จัดทำต้องการได้ซึ่งปัญหาเกิดจากการแตก File Python ที่ติดตั้งไม่สำเร็จ

โดยทางผู้จัดทำจึงแก้ปัญหาโดยการนำ Docker Container ไป Deploy บน Server แทนโดยทางผู้จัดทำได้ทำการนำ Directory ที่เก็บ Django Application รวมทั้ง Docker File, requirements.txt และ docker-compose.yml โคลนผ่าน Github ลงไปยัง server ที่เตรียมไว้ ดังรูปที่ 4.5



**รูป 4.5 Directory ที่เก็บ Django Application**

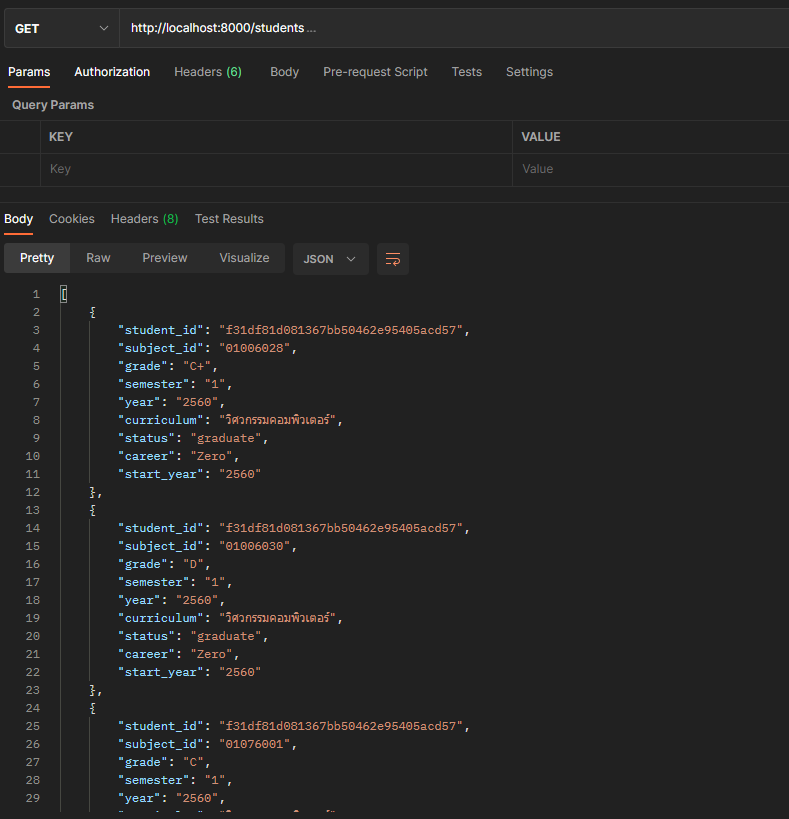
โดยต่อมาทางผู้จัดทำได้ทำการทดลองใช้คำสั่งในการสร้าง Docker Compose ขึ้นมาใน Server โดยจะสร้าง Docker Compose ผ่านการอ่าน File Docker File, requirements.txt และ docker-compose.yml ที่มากับ Directory ของ Django Application ดังรูปที่ 4.6

**รูป 4.6 ผลลัพธ์การสร้าง Docker Compose บน Server**

จากผลการทดลองที่ 4.2 พบว่าสามารถนำ Django Application มาบน Server ที่เตรียมไว้ได้ และสามารถสร้าง Docker Compose ได้

* 1. การใช้งานฐานข้อมูลร่วมกับ Application

โดยการทดลองการใช้งานร่วมกันของฐานข้อมูลกับ Application นั้นทางผู้จัดทำได้ทำการทดลองการใช้งานผ่านทางเครื่องคอมพิวเตอร์ส่วนบุคคลของผู้จัดทำเองในรูปแบบ Local Host ซึ่งการทำงานร่วมกันของฐานข้อมูลกับ Application นั้นทางผู้จัดทำได้ทดลองเป็นการทำงานในรูปแบบการใช้ API ในการติดต่อระหว่างฐานข้อมูลกับ Application ทั้งกระบวนการค้นหาข้อมูลดังรูปที่ 4.7, เพิ่มข้อมูลดังรูปที่ 4.8, แก้ไขข้อมูลดังรูปที่ 4.9 และ การลบข้อมูลดังรูปที่ 4.10



**รูป 4.7 ผลลัพธ์การค้นหาข้อมูลผ่าน API**

A screenshot of a computer

Description automatically generated with medium confidence

**รูป 4.8 ผลลัพธ์การแก้ไขข้อมูลผ่าน API**

A screenshot of a computer

Description automatically generated with medium confidence

**รูป 4.9 ผลลัพธ์การเพิ่มข้อมูลผ่าน API**

**A screenshot of a computer

Description automatically generated with medium confidence**

**รูป 4.10 ผลลัพธ์การลบข้อมูลผ่าน API**

จากการทดลองที่ 4.3 พบว่าการทำงานร่วมกันของ Application ในส่วนของ Backend และฐานข้อมูลเป็นไปได้ด้วยดี

* 1. การพัฒนา Similarity Model สำหรับการทำนายเกรดของผู้ใช้งานตามที่ต้องการ

โดยการทดลองการพัฒนา Similarity Model นี้ทางผู้จัดทำได้ทำการทดลองใน Google Colab โดยทางผู้จัดทำได้เลือกใช้อัลกอริทึม SVD ของ Surprise จาก Library ของ Scikit-Learn โดยการทดลองนี้ผู้จัดทำได้นำชุดข้อมูลผลลัพธ์การเรียนของบัณฑิตปีการศึกษา 2560 และ 2561 มาทดลองโดยผลลัพธ์ที่พึงประสงค์ของ Similarity Model ที่ผู้จัดทำคาดหวังคือ ทำนายข้อมูลผลลัพธ์การเรียนของบัณฑิตในวิชาที่บัณฑิตผู้นั้นไม่ได้ลงทะเบียนไว้จากหลักการของอัลกอริทึม SVD โดยผู้จัดทำได้กำหนด Parameter ว่า Ratings คือ Grade ของบัณฑิต UserID คือรหัสนักศึกษาของบัณฑิต และ ProductID คือรหัสวิชา โดยชุดข้อมูลที่ได้มานั้นประกอบไปด้วยข้อมูลของบัณฑิตหลายหลักสูตรด้วยกันโดยทางผู้จัดทำได้เลือกใช้แค่เพียงหลักสูตร วิศวกรรมคอมพิวเตอร์ และ วิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) ดังรูปที่ 4.11 โดยจะแบ่งส่วนของการทดลองออกเป็น 2 ส่วนคือ การเตรียมข้อมูลการทดลอง และ การ Train Model

Table

Description automatically generated

**รูป 4.11 ชุดข้อมูลในการทดลอง**

* + 1. การเตรียมข้อมูลการทดลอง

โดยในทุกรูปแบบการทดลองทางผู้จัดทำจะ ทำการจัดการเตรียมการชุดข้อมูลในรูปแบบเดียวกัน โดยทางผู้จัดทำจะ ทำการนำข้อมูลชุดที่ไม่ใช้งานออกในที่นี้คือชุดข้อมูลของบัณฑิตที่ไม่ได้อยู่ในหลักสูตรวิศวกรรมคอมพิวเตอร์ และ วิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) และได้ทำการเปลี่ยนรูปแบบของข้อมูล Column Grade จากรูปแบบของตัวอักษรให้อยู่ในรูปแบบของตัวเลข และแยกส่วนของชุดข้อมูลออกตามหลักสูตร วิศวกรรมคอมพิวเตอร์ และ วิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) เพื่อทำการสร้าง Model ให้ทั้งสองหลักสูตรโดยจะทำการจัดรูปแบบของ Column ของข้อมูลตามนี้ student\_id, subject\_id, grade, semesters, year, curriculum และ job ดังรูปที่ 4.12

Text

Description automatically generated

**รูป 4.12 ชุดข้อมูลหลังจากผ่านการเตรียมข้อมูล**

* + 1. **การ** Train **Model**

หลังจากที่ผู้จัดทำได้ทำการเตรียมชุดข้อมูลเรียบร้อยแล้วจะนำชุดข้อมูลที่ได้มาทำการ Tuning Hyperparameter ด้วย Gird Search โดยช่วงของ Parameter ที่ได้กำหนดสำหรับอัลกอริทึม SVD มีดังนี้ n\_factors = [20, 50, 100] และ n\_epochs = [5 ,10, 20] และได้ทำการแบ่ง

ชุดข้อมูลในการทดสอบออกเป็น 80 (สำหรับ Train) ต่อ 20 (สำหรับทดสอบ) ส่วน และทำการวัดประสิทธิภาพโดยใช้ค่า RMSE

จากผลลัพธ์การทดลองที่ 4.4 พบว่าชุดข้อมูลในส่วนของหลักสูตร วิศวกรรมคอมพิวเตอร์ มีค่า RMSE อยู่ที่ 0.6319 และ ในส่วนของหลักสูตร วิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) คอมพิวเตอร์ มีค่า RMSE อยู่ที่ 0.7871 ซึ่งเมื่อวัดจากช่วงของ Rating หรือ Grade ของชุดข้อมูลที่อยู่ตั้งแต่ 0 ถึง 4

ซึ่งยังไม่เป็นที่น่าพอใจทางผู้จัดทำเลยทำการสันนิษฐานว่าการที่จำนวนของวิชาที่มีจำนวนที่มากนั้นอาจจะส่งผลต่อการคำนวณของอัลกอริทึม จึงแก้ปัญหาโดยการจัดกลุ่มวิชา

* 1. การพัฒนา NLP สำหรับการจับกลุ่มวิชา

**เนื่องจากการทำนายผลเกรดของนักศึกษานั้น ต้องประกอบไปด้วยข้อมูล** subject\_id **หรือข้อมูลรหัสของรายวิชานั้นๆ ซึ่งหลักสูตรรายวิชามีการถูกปรับทุกๆ 4 ปี จึงทำให้เกิดปัญหาในการ** Train Model **เพราะมีข้อมูลรหัสรายวิชาที่ไม่เหมือนกัน แต่ว่าตัววิชานั้นมีความคล้ายคลึงกับตัววิชาเดิม ทางผู้จัดทำจึงนำบทคัดย่อของรายวิชามาจัดกลุ่มวิชา โดยการหาค่า** Similarity **ผ่านกระบวนการ** NLP **ความคล้ายคลึงของบทคัดย่อแต่ละวิชา โดยมีตัวอย่างของข้อมูลดังรูปที่ 4.13**

**Graphical user interface, text, application, email

Description automatically generated**

**รูป 4.13 ตัวอย่างข้อมูล Data Subject**

ขั้นตอนแรกทำการนำเข้าข้อมูลจาก Table subject\_data สำหรับการหาค่า Similarity ใน Table subject\_data เป็นการรวมข้อมูลรายวิชาของแต่หลายหลักสูตร จึงทำให้มีรหัสวิชาบางรายวิชาซ้ำ ใช้วิธีการแก้ไขโดยการ Drop Record ที่ซ้ำออก แล้วเก็บ Record แรกเอาไว้ ผลลัพธ์ดังรูปที่ 4.14

**Graphical user interface, text

Description automatically generated**

**รูป 4.14 ข้อมูล Data Subject หลังจากตัดรหัสวิชาที่ซ้ำออก**

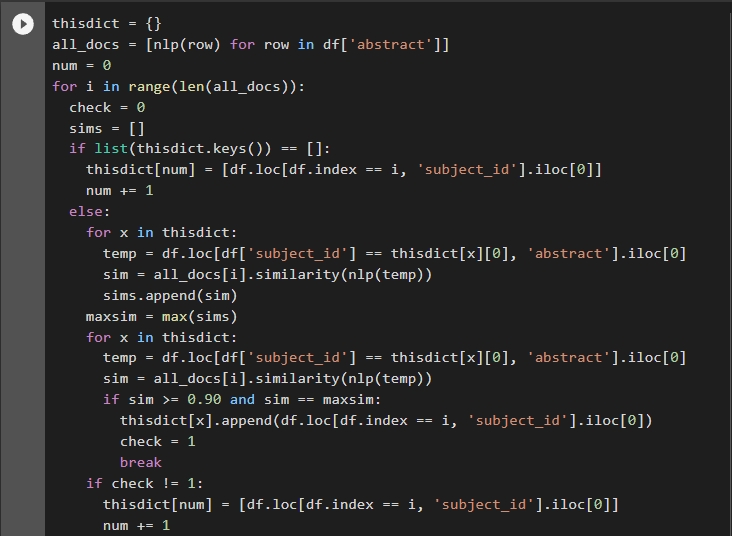
หลังจากนั้นตัวบทคัดย่อที่จะนำไปหาค่า Similarity นั้น มีคำเชื่อมต่างๆไว้ ซึ่งส่งผลต่อการนำไปหาค่า NLP จึงนำคำเชื่อมเหล่านั้นออกจากตัวบทคัดย่อของแต่ละหลายวิชา เพื่อให้เหลือแต่คำที่เฉพาะเจาะจงมากขึ้น ผลลัพธ์ดังรูปที่ 4.15

**A computer screen capture

Description automatically generated with medium confidence**

**รูป 4.15 ข้อมูล Data Subjectหลังจากจัดแต่งบทคัดย่อ**

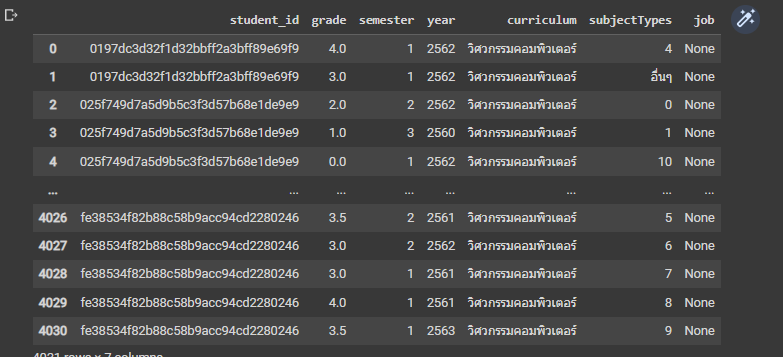
เมื่อบทคัดย่อมีความพร้อมต่อการนำเข้าหาค่า NLP แล้ว ต่อมาทำการสร้าง Dict เอาไว้ โดยให้ค่า Key เป็นเลขกลุ่ม โดยเริ่มจาก 0 ค่า Value ทำการเก็บเป็น List ของรหัสวิชาต่างๆที่อยู่ในกลุ่มเดียวกัน ต่อมาได้ทำการวนลูปตามจำนวนของวิชาทั้งหมด และทำการเทียบค่า Similarity ของแต่ละวิชา โดยจะจัดกลุ่มจากค่า Similarity ที่มีค่ามากกว่าเท่ากับ 0.90 ให้อยู่ในกลุ่มเดียวกัน และสร้าง Column เพิ่ม เพื่อเก็บค่าเลขกลุ่มเอาไว้ ตัวอย่างของ code ดังรูปที่ 4.16

****

**รูป 4.16 Coding การเทียบค่า Similarity**

การนำชุดข้อมูลวิชาที่จัดกลุ่มไปใช้กับ Similarity Modelโดยการเตรียมข้อมูลในการทดลองนั้นจำเป็นต้องนำชุดข้อมูลกลุ่มของวิชามารวมกับชุดข้อมูลของผลลัพธ์การเรียนของบัณฑิต แล้วรวมกลุ่มข้อมูลของผลลัพธ์การเรียนของบัณฑิต

ในวิชาที่อยู่ในกลุ่มเดียวกันโดยรวมข้อมูลผ่านการทำค่าเฉลี่ย แล้วจึงนำข้อมูลไปทดลองกับการ Train แบบจำลอง Similarity Model โดยจะทำการจัดรูปแบบของ Column ของข้อมูลตามนี้ student\_id, grade, semesters, year, curriculum, subjectTypes และ job ผลลัพธ์ดังรูปที่ 4.17

**Text

Description automatically generated**

**รูป 4.17 ชุดข้อมูลหลังจากผ่านการเตรียมข้อมูลผลลัพธ์จาก NLP**

จากผลลัพธ์การทดลองที่ 4.5 พบว่าชุดข้อมูลในส่วนของหลักสูตร วิศวกรรมคอมพิวเตอร์ มีค่า RMSE อยู่ที่ 0.5751 และ ในส่วนของหลักสูตร วิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) คอมพิวเตอร์ มีค่า RMSE อยู่ที่ 0.7351 ซึ่งเมื่อวัดจากช่วงของ Rating หรือ Grade ของชุดข้อมูลที่อยู่ตั้งแต่ 0 ถึง 4 ดังรูปที่ 4.18

Calendar

Description automatically generated

**รูป 4.18 ผลลัพธ์การทดลองร่วมกับ NLP**

จากผลลัพธ์ที่ได้เลยสามารถสรุปได้ว่าการสันนิษฐานของผู้จัดทำในเรื่องของจำนวนวิชาที่มีมากเกินไปทำให้ค่า RMSE เพิ่มขึ้นนั้นถูกต้อง และผลลัพธ์ที่ได้ออกมาเป็นที่น่าพอใจเพียงพอแล้ว สรุปผลการนำไปปรับใช้งานจริง เนื่องด้วยการ prediction ที่ใช้การจับกลุ่มของรายวิชานั้นไม่มีความสามารถที่จะตอบรับความต้องการของการใช้งานจริงได้เนื่องด้วยการใช้งานจริงนั้นได้ทำการกำหนดผลลัพธ์การทำนายไว้ว่าจำเป็นต้องแสดงผลการทำนายออกมาเป็นรายวิชาแต่การทำนายในรูปแบบของการทดลองนี้นั้นจะได้เป็นผลลัพธ์ของกลุ่มรายวิชาแทนซึ่งไม่สอดคล้องกับที่ต้องการ ทำให้การปรับใช้งานจริงของแบบจำลองนั้นจะใช้ตามการทดลองที่ 4.4

* 1. การพัฒนา Job Classification Model

โดยการทดลองนี้จะเป็นการทดลองการสร้างแบบจำลองการทำนายผลลัพธ์สายงานอาชีพ **โดยมีข้อมูลที่ใช้จะเป็นข้อมูลที่ได้จากการเก็บแบบสอบถามการมีงานทำจากบัณฑิตหลักสูตรวิศวกรรมคอมพิวเตอร์หลักสูตรปกติและต่อเนื่อง** 2560 **ปีการศึกษา** 2560 **และ** 2561**ดังรูปที่ 4.19**

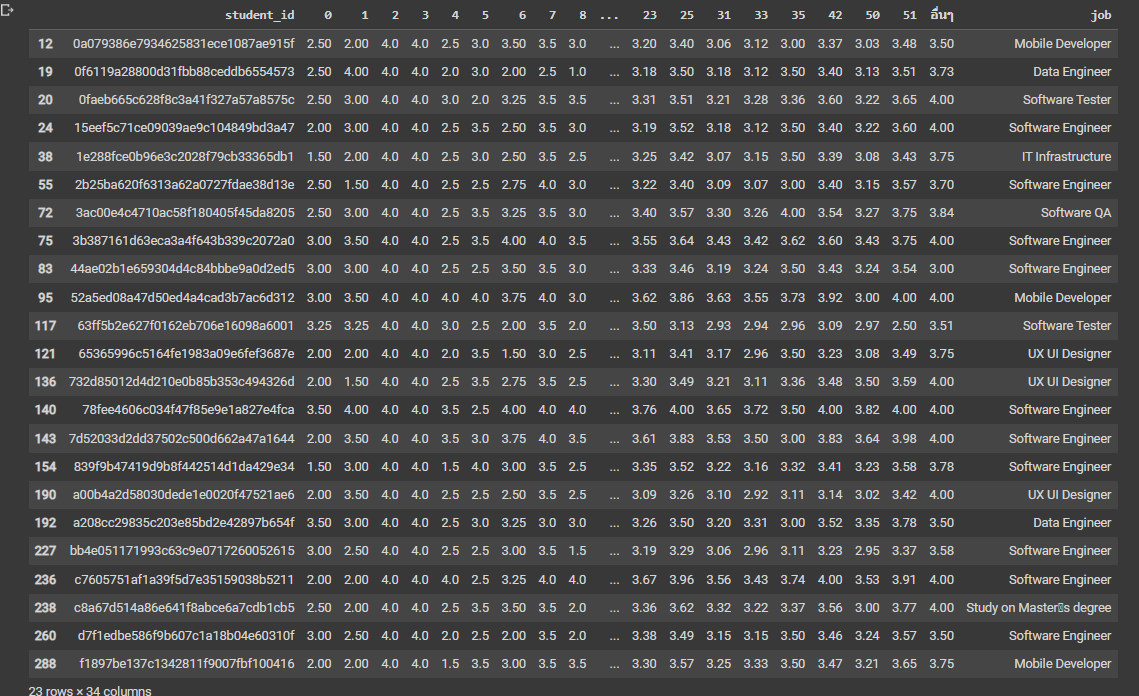
โดยการทดลองนี้ได้ใช้อัลกอริทึมในการสร้าง Model ด้วยกันสองอัลกอริทึมคือ KNN Classifier และ Decision Tree Classifier จาก Library Scikit-learn โดยชุดข้อมูลที่ใช้จะมี ชุดข้อมูลที่เป็นผลลัพธ์เกรดของบัณฑิตในระบบ และชุดข้อมูลแบบสอบถามการมีงานทำของบัณฑิต โดยการวัดประสิทธิภาพของ Model จะใช้ค่า Accuracy ที่ได้จากการคำนวณของ Function Metrics Accuracy จาก Library Scikit-Learn

**Text

Description automatically generated**

**รูป 4.19 ชุดข้อมูลแบบสอบถามการมีงานทำของบัณฑิต**

โดยขั้นตอนต่อมาคือการเตรียมข้อมูล ทางผู้จัดทำได้ทำการนำผลลัพธ์เกรดของบัณฑิตในระบบมาทำการรวมข้อมูลของสองชุดข้อมูลเข้าด้วยกันโดยใช้ student\_id ในการรวมกันของสองชุดข้อมูล โดยถ้าในชุดข้อมูลผลลัพธ์เกรดของบัณฑิตในระบบไม่มีข้อมูลรหัสนักศึกษาที่ชุดข้อมูลแบบสอบถามการมีงานทำของบัณฑิตจะไม่นำมาเป็นข้อมูลในการสร้าง Model หลังจากนั้นจะทำการ Transpose ข้อมูลให้อยู่ในรูปแบบดังรูปที่ 4.20 เพื่อนำไป Train Model

**รูป 4.20 ชุดข้อมูลที่ผ่านการ Transpose**

จากผลลัพธ์การทดลองที่ 4.6 พบว่าค่า Accuracy ของอัลกอริทึม KNN Classifier อยู่ที่ 0.4 และ Accuracy ของอัลกอริทึม Decision Tree Classifier อยู่ที่ 0.2 จากคะแนนเต็ม 1 ซึ่งการทางผู้จัดทำได้สันนิษฐานว่าข้อผิดพลาดที่เกิดขึ้นมาจากจำของชุดข้อมูลแบบสอบถามการมีงานทำของบัณฑิตที่มีน้อยเกินไปรวมทั้งจำนวนของทั้งกลุ่มของผลลัพธ์สายอาชีพและจำนวนวิชาที่มากเกินไป

* 1. การพัฒนาเพิ่มประสิทธิภาพของ Job Classification Model

**เนื่องด้วยข้อสันนิษฐานจากการทดลองที่ 4.6 ทางผู้จัดทำจึงได้ทำการเปลี่ยนรูปแบบการทดลองใหม่โดยได้ทำการเปลี่ยนอัลกอริทึมในการสร้าง** Model **โดยใช้เป็น** Random Forest Classification **และทำการปรับกลุ่มของรายอาชีพใหม่เพื่อทำการลดจำนวนกลุ่มของอาชีพ เช่น จาก** Study on master Degree **เป็น** Software Engineer, Mobile Developer **เป็น** Software Developer, Data Engineer **และ** Data Analytic **เป็น** Data Career **ดังรูปภาพที่** 4.21



**รูป 4.21 ชุดข้อมูลแบบสอบถามการมีงานทำของ บัณฑิต ที่ทำการลดจำนวนกลุ่ม**

**หลังจากนั้นทางผู้จัดทำได้ทำการทดลองแล้ววัดผลลัพธ์ด้วยการใช้ค่า** Accuracy **เช่นเดิมโดยได้ทำการ** Hyperparameter Tuning **ด้วย** parameter **ดังนี้** 'n\_estimators': [**50, 100, 200, 500**, **1000]**, 'max\_features': ['auto', 'sqrt', 'log**2**', None], 'max\_depth': [**5, 10, 20,** None], 'min\_samples\_split': [**2, 5, 10]**, 'min\_samples\_leaf': [**1, 2, 4]** **และได้ทำการแบ่งชุดข้อมูลด้วยในการสร้างและทดสอบแบบจำลองออกเป็น** 9:1 **โดยสร้าง** 9 **ส่วนและทดสอบ** 1 **ส่วน**

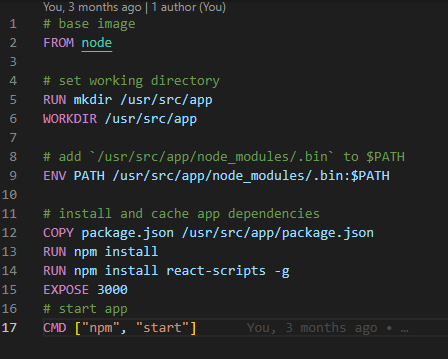
**จากผลลัพธ์การทดลองที่ 4.**7 **พบว่าค่า** Accuracy **ของอัลกอริทึม** Random Forest Classification **ร่วมกับการปรับลดจำนวนของกลุ่มผลลัพธ์นั้นออกมาอยู่ที่** 0.75 **หรือคิดเป็น** 75% **ความถูกต้องซึ่งเป็นผลลัพธ์ที่น่าพึงพอใจแล้ว แล้วจะนำกระบวนการนี้ไปปรับใช้กับระบบจริง**

* 1. การพัฒนาหน้าต่างผู้ใช้งานของ Application

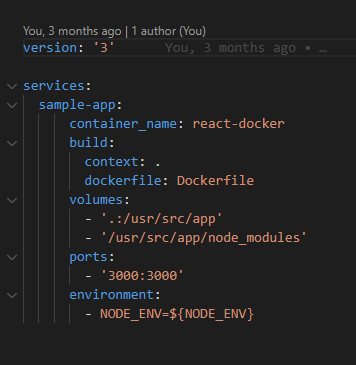
**โดยจะแบ่งส่วนของการทดลองออกเป็น 3 ส่วนคือ การทดลองสร้าง** React Project **โดย**Docker Compose, **การทดลองการทำงานร่วมกันของ** React Project **และ** Django Application **และ** **การนำ** Docker Compose **ของ** React Project **ขึ้นไปอยู่บน** Server

* + 1. **การทดลองสร้าง React Project โดย Docker Compose**

โดยการทดลองนี้ทางผู้จัดทำได้ทำการเตรียม File สำหรับการสร้างโดยมี Dockerfile ที่ระบุการสร้าง Directory ของ React Project และ Version ของ NodeJS ดังรูปที่ 4.22 ต่อมาคือ File docker-compose.yml ดังรูปที่ 4.23

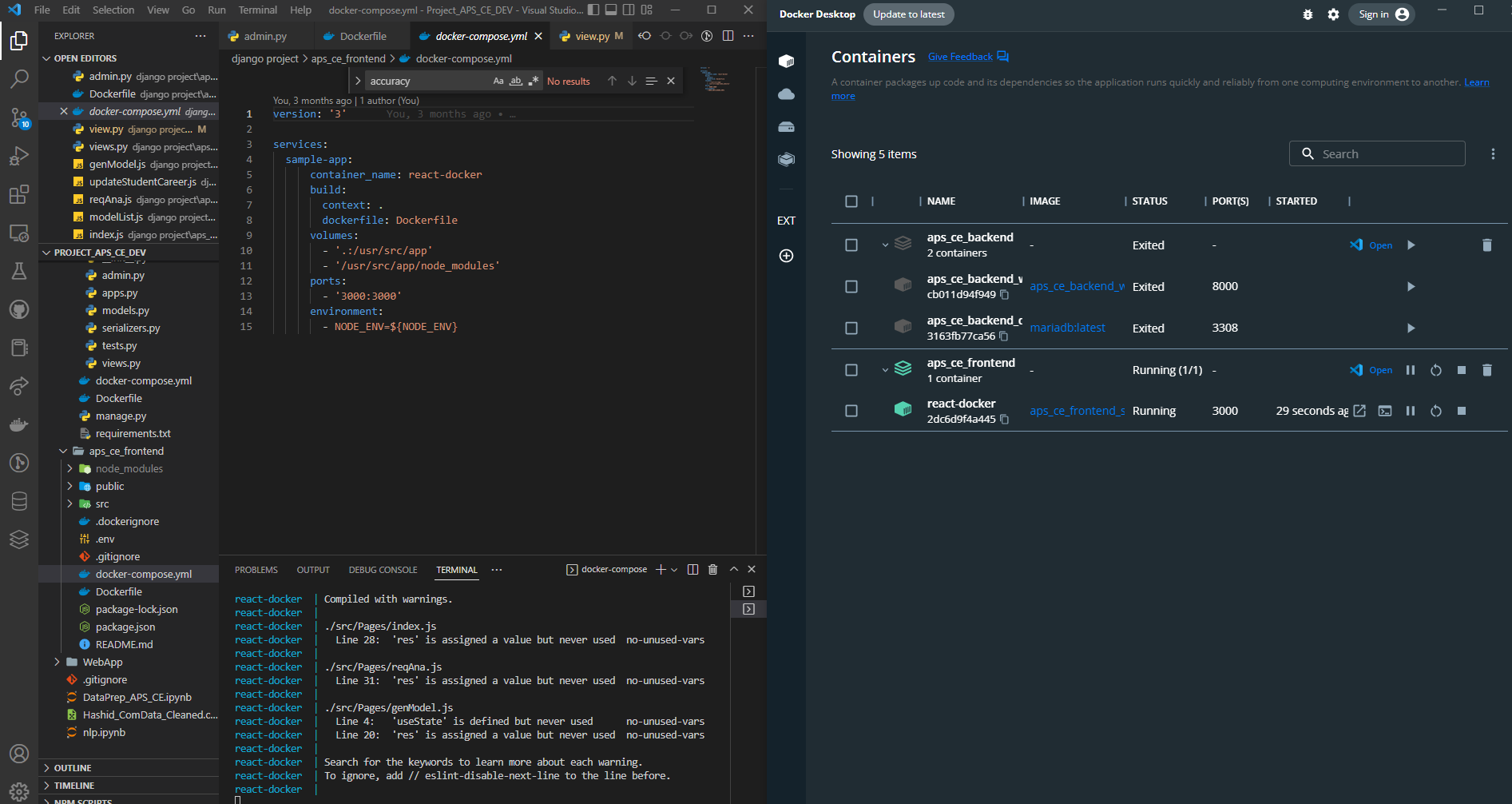


**รูป 4.22 Dockerfile React Project**

****

**รูป 4.23 docker-compose.yml React Project**

ซึ่งเมื่อหลังจากการสร้าง docker compose ขึ้นมาทางผู้จัดทำจะตรวจสอบการทำงานของระบบทั้งหมดผ่าน Application Docker Desktop ร่วมกับ log terminal ของ Visual Studio Code ดังรูปที่ 4.24 โดยได้กำหนด Port ในการเข้าถึงทั้งภายนอกและภายในเป็น 3000

**รูป 4.24 ภาพรวมของการตรวจสอบการทำงานของ React Project**

จากผลการทดลองที่ 4.8.1 พบว่าสามารถสร้าง Docker Compose ของ React Project ได้ และสามารถตรวจสอบการทำงานต่างๆ ของ React Project ผ่าน Visual Studio Code Terminal และ Application Docker Desktop ได้

* + 1. **การทดลองการทำงานร่วมกันของ React Project และ Django Application**

โดยในการทดลองนี้ทางผู้จัดทำได้กำหนดสภาพแวดล้อมของการทดลองโดยให้ทั้งสองส่วน (React Project และ Django Application) ทำงานอยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลในรูปแบบ Local Host ดังรูปที่ 4.25 โดยด้านซ้ายคือหน้าหลักของ React Project และด้านขวาคือหน้าของ Django Application

Graphical user interface, application, Word

Description automatically generated

**รูป 4.25 ภาพรวมของ React Project และ Django Application ที่เรียกดูผ่าน Web Browser**

จากผลลัพธ์การทดลองที่ 4.8.2 พบว่าทั้งสองสามารถติดต่อส่งข้อมูลให้กันและกันได้ราบรื่น โดย React Project จะติดต่อกับ Django Application ผ่าน Port 8000 และ Django Application จะติดต่อกับ React Project ผ่าน Port 3000 ผลลัพธ์ดังรูปที่ 4.26

Graphical user interface, application

Description automatically generated

**รูป 4.26 ทดสอบการติดต่อระหว่าง React Project และ Django Application**

* + 1. **การนำ Docker Compose ของ React Project ขึ้นไปอยู่บน Server**

โดยในขั้นตอนนี้ทางผู้จัดทำได้นำ Directory ของ React Project ไปยัง Server ที่ได้จัดเตรียมไว้ โดยจะเป็น Server เดียวกันกับในหัวข้อการทดลอง 4.2 ผลลัพธ์ดังรูปที่ 4.27

Text

Description automatically generated

**รูป 4.27 React Project ที่นำขึ้นไปยัง Server ที่เตรียมไว้**

โดยต่อมาได้ทำการทดลองโดยการสร้าง Docker Compose ของ React Project ขึ้นมา โดยจากผลลัพธ์การทดลองที่ 4.8.3 พบว่าสามารถสร้าง Docker Compose ของ React Project ได้สำเร็จและตัว React Project สามารถดำเนินการได้อย่างราบรื่น

จากผลลัพธ์การทดลองที่ 4.8 พบว่าสามารถสร้าง React Project จาก Docker Compose ได้และติดต่อกับ Django Application ได้อย่างราบรื่น และสามารถนำ React Project Directory ขึ้นไปสร้าง Docker Compose บน Server ได้และตัว React Project เองสามารถทำงานได้เป็นปกติ

* 1. การนำกระบวนการสร้าง Model และการเรียกใช้ลงบน Django Application

โดยในการทดลองนี้ทางผู้จัดทำจะนำกระบวนการสร้างและเรียกใช้ที่ได้ทำการทดลองไปในการทดลองที่ 4.4 , 4.5 และ 4.6 ซึ่งจะนำกระบวนการสร้างและเรียกใช้ Similarity Model และ NLP ให้อยู่ในรูปแบบของ API และ Call Function โดยกระบวนการทำงาน และ การเรียกใช้งานของ Similarity Model และ NLP มีดังนี้

* + 1. **กระบวนการ การทำงานของ NLP**

1. ผู้ใช้งานนำ File CSV ของรายวิชาที่ต้องการเพิ่มในระบบ Upload เข้าไป
2. ได้ผลลัพธ์ออกมาเป็น List ของ Dictionary ที่เก็บ Class และวิชาใน Class
3. นำ List ที่ได้มารวมกับตารางจาก File CSV ที่ผู้ใช้งาน Upload
4. รูปให้ข้อมูลอยู่ในรูปแบบที่สามารถเพิ่มลงในฐานข้อมูลได้
5. ข้อมูลเพิ่มเข้าฐานข้อมูล
6. แสดงผลลัพธ์การเรียกใช้งาน API
   * 1. **กระบวนการ การสร้าง Similarity Model**
7. ผู้ใช้งานเลือกรูปแบบของ Model ที่จะสร้าง (หลักสูตร, รูปแบบการพยากรณ์ รายวิชา หรือ กลุ่มของวิชา)
8. ค้นหาข้อมูลของบัณฑิตในฐานข้อมูล
9. นำข้อมูลมาสร้าง Model
10. แปลงรูปแบบข้อมูลของ Model ให้อยู่ในรูป Pickeled File
11. คำนวณหา RMSE ของ Model นั้น
12. จัดรูปให้ข้อมูลอยู่ในรูปแบบที่สามารถเพิ่มลงในฐานข้อมูลได้
13. นำข้อมูลเพิ่มเข้าฐานข้อมูล
14. แสดงผลลัพธ์การเรียกใช้งาน API
    * 1. **กระบวนการ การเรียกใช้งาน Similarity Model**
15. ผู้ใช้งาน Upload File CSV ผลลัพธ์การเรียนของตน
16. ผู้ใช้งานเลือกหลักสูตรและปีของหลักสูตร
17. ระบบดึงข้อมูลรายวิชาจากฐานข้อมูล
18. ระบบดึงข้อมูลเกรดนักศึกษ
19. ระบบดึงข้อมูลนักศึกษา
20. ระบบนำข้อมูลของผู้ใช้งานและข้อมูลตามข้อ 4 และ 5 มารวมกัน
21. ระบบสร้างแบบจำลอง Similarity ขึ้นมาจากข้อมูลตามข้อที่ 6
22. ระบบทำนายผลรายวิชาที่ต้องการของผู้ใช้งาน
23. แสดงผลลัพธ์การเรียกใช้งาน API
    * 1. **กระบวนการ การสร้าง Job Classification Model**
24. ผู้ใช้งานใส่ชื่อ Model เลือกหลักสูตรและปีของหลักสูตรสำหรับการใช้งาน
25. ระบบดึงข้อมูลของบัณฑิตจากในระบบ
26. นำข้อมูลมาสร้าง Model
27. แปลงรูปแบบข้อมูลของ Model ให้อยู่ในรูป Pickeled File
28. คำนวณหา Accuracy ของ Model นั้น
29. จัดรูปให้ข้อมูลอยู่ในรูปแบบที่สามารถเพิ่มลงในฐานข้อมูลได้
30. นำข้อมูลเพิ่มเข้าฐานข้อมูล
31. แสดงผลลัพธ์การเรียกใช้งาน API
    * 1. **กระบวนการเรียกใช้งาน Job Classification Model**
32. ผู้ใช้งาน Upload File CSV ผลลัพธ์การเรียนของตน
33. ผู้ใช้งานเลือกหลักสูตรและปีของหลักสูตร
34. ระบบดึง Model จากฐานข้อมูล
35. ระบบแปลงข้อมูลของผู้ใช่งานให้สอดคล้องกับ Model
36. ระบบทำการทำนายผลลัพธ์ของสายงานอาชีพของผู้ใช้งาน
37. แสดงผลลัพธ์การเรียกใช้งาน API

จากผลลัพธ์การทดลองที่ 4.8 พบว่าทั้งสามกระบวนการการทำงานนั้นสามารถทำงานได้บน Django Application ได้อย่างราบรื่น

## การใช้งานหน้า **Web Application**

ในส่วนของหน้าหลัก

ในส่วนของหน้าสายงานบัณฑิต การเป็นการโชว์ข้อมูลสถิติทางด้านสายงานของนักศึกษา โดยจะแบ่งเป็นสองส่วนคือ ส่วนสถิติในอดีตของนักศึกษา จะเป็นการดึงข้อมูลด้านสายงานและจำนวนในแต่ละปีจากฐานข้อมูลมาแสดงผ่านกราฟ ส่วนที่สองส่วนคาดการณ์สถิติบัณฑิต จะเป็นการส่งข้อมูลปีที่ต้องการคาดการณ์ไปเข้าโมเดลเพื่อทำนายสายงานในอนาคตของนักศึกษาปีที่เลือกออกมา โดยแสดงผ่านกราฟ

ในส่วนของหน้าพยากรณ์นักศึกษา จะมีให้เลือกหลักสูตรและปีที่เรียน เพื่อดาวน์โหลดแบบฟอร์มรูปแบบ CSV ในการทำการกรอกเกรด เพื่อนำไปทำนายผล เมื่อทำการกรอกเสร็จ จะมีปุ่มให้อัปโหลดไฟล์เพื่อกดพยากรณ์

ในส่วนของหน้าผลลัพธ์การพยากรณ์นักศึกษา จะโชว์เกรดวิชาที่นักศึกษาต้องการจะทำนายเกรด และอาชีพที่เป็นไปได้ที่นักศึกษาสามารถทำได้

ในส่วนของหน้าแนะนำวิชาเลือกภาค จะทำการเลือกคณะและหลักสูตรที่เรียน ระบบจะทำการโชว์ key ของแต่ละรายวิชาเลือกภาคขึ้นมา เพื่อให้นักศึกษาทำการเลือกสิ่งที่สนใจ และกดปุ่มแนะนำเพื่อประมวลผลวิชาเลือกภาคที่นักศึกษามีความสนใจ โดยจะมีบอกรายละเอียดรหัสวิชาและบทคัดของวิชานั้นๆ

ในส่วนของหน้าเข้าสู่ระบบจะมีการเข้าสู่ระบบด้วยสองสถานะ คือกรรมการหลักสูตร และ นักศึกษา

สถานะของกรรมการหลักสูตรมีไว้เพื่อเข้าถึงหน้าอัพโหลดไฟล์ สถานะของนักศึกษามีไว้เพื่อกรอกข้อมูลอาชีพของนักศึกษาที่จบไปแล้ว

ในส่วนของหน้าอัปโหลดไฟล์ จะมีให้อัปโหลดไฟล์ CSV 2 ประเภท คือ ข้อมูลของนักศึกษาและข้อมูลหลักสูตรวิชา หน้าเว็บจะทำการโชว์ไฟล์ในระบบที่มีเพื่อโชว์ให้ดูและสามารถ ลบและแก้ไขไฟล์นั้นได้

โดยส่วนของหน้าแก้ไขไฟล์ จะสามารถลบ แก้ไข เพิ่ม record ได้

ในส่วนของหน้านักศึกษาจะมีให้กรอกข้อมูลรหัสนักศึกษาและอาชีพของนักศึกษาที่เรียนจบไปแล้ว เพื่อนำไปเทรนตัวโมเดล

**บทที่ 5**

**สรุป**

* 1. บทสรุป

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาระบบประมวลผลข้อมูลผลการเรียนของนักศึกษาในอดีต ข้อมูลของรายวิชาต่างๆ และข้อมูลจากแบบสำรวจการมีงานทำของบัณฑิต เพื่อนำเสนอข้อมูลสถิติต่างๆ วิเคราะห์ข้อมูลผลการผลิตบัณฑิตเพื่อให้ได้ผลลัพธ์ว่าที่ผ่านมาหลักสูตรสามารถผลิตบัณฑิตกลุ่มใดได้บ้าง มีจำนวนมากน้อยเพียงใด สามารถพยากรณ์ว่าในอนาคตหลักสูตรสามารถผลิตบัณฑิตกลุ่มใดได้เป็นจำนวนเท่าใด เพื่อเป็นประโยชน์และอำนวยความสะดวกให้กรรมการหลักสูตรในการวางแผนการบริหารหลักสูตรในอนาคต และแสดงเป็นแผนภาพกราฟิกในการอำนวยความสะดวกให้หน่วยงานภายนอกได้รับทราบว่าหลักสูตรปัจจุบันของสถาบันสามารถผลิตบุคลากรที่มีความชํานาญด้านใดได้บ้างโดยการทำงานของ Application นี้มี Web Application, Server, Database

* + 1. **การทำงานในส่วนของ Web Application**

1. Web Application มีระบบสร้าง File CSV Template สำหรับให้ผู้ใช้งานใส่ผลลัพธ์การเรียนของตน
2. Web Application สามารถสร้าง Job Classification Model แล้วนำลงฐานข้อมูลได้
3. Web Application สามารถนำเข้ารายวิชาแล้วเรียกใช้การจับกลุ่มรายวิชาด้วย NPL ได้
4. Web Application สามารถนำเข้าแบบสำรวจการมีงานทำของบัณฑิตแล้วแก้ไขข้อมูลของบัณฑิตคนนั้นในฐานข้อมูลได้
5. Web Application สามารถนำเข้า File CSV ผลลัพธ์การเรียนของผู้ใช้งานได้
6. Web Application มีระบบรองรับการร้องขอผลลัพธ์การพยากรณ์ได้
7. Web Application สามารถติดต่อกับ Database ได้
8. Web Application สามารถจัดเก็บ File ในระบบได้
9. Web Application สามารถแก้ไข File ในระบบได้
10. Web Application สามารถลบ File ในระบบได้
11. Web Application สามารถกู้คืน File ในระบบได้
12. Web Application สามารถลบ File ในระบบได้อย่างถาวร
13. Web Application สามารถ login ใช้งานด้วยสิทของกรรมการหลักสูตรได้
    * 1. **การทำงานในส่วนของ Server**
14. Server สามารถรัน Web Application ผ่าน Dockerได้
15. Server สามารถสร้าง Database ผ่าน Docker ได้
    * 1. **การทำงานในส่วนของ Database**
16. Database สามารถเก็บข้อมูลผลลัพธ์การเรียนของบัณฑิตได้
17. Database สามารถเก็บข้อมูลรายวิชาได้
18. Database สามารถเก็บข้อมูลของ Model ได้
19. Database สามารถเก็บข้อมูลของ File CSV ในระบบได้
    1. **ปัญหา**และ**อุปสรรคที่พบ**
20. ข้อมูลจากแบบสอบถามการมีงานทำของบัณฑิตที่มีน้อยเกินไปทำให้ผลลัพธ์ของสายงานอาชีพในการทำนายนั้นออกมาไม่ได้หลากหลายตามที่ต้องการ โดยได้วางแนวทางในการแก้ไขปัญหาไว้โดยได้ทำการเพิ่ม QR Code ในการกรอกแบบฟอร์มการมีงานทำของบัณฑิตเอาไว้ และมี function ในการให้กรรมการหลักสูตรอัปเดตข้อมูลดังกล่าวลงยังฐานข้อมูล
    1. ข้อเสนอแนะ
21. ค่าผลลัพธ์ Accuracy ของ Model Job Classification นั้นควรอยู่ที่ 0.9

**บรรณานุกรม**

สุเมธ ดาราพิสุทธิ์. 2559. “การสร้างรายการเพลงโดยใช้การกรองร่วมแบบเซสชั่นที่เพิ่มขึ้นด้วย

กลไกการลืมและการวิเคราะห์สถิติเชิงมุม.” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์, มหาวิทยาลัยบูรพา.

นิภาภรณ์ พันธ์นาม. 2563. “ระบบแนะนำสินค้าอาหารโดยใช้ระบบแนะนำแบบผสมผสาน.”

สารนิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล,

มหาวิทยาลัยศรีนครินทรวิโรฒ.

Punna Rirkvaleekul. 2020. Machine Learning – คำศัพท์ที่ควรรู้. [Online].

Available : <https://twinsynergy.co.th/machine-learning>.

พิพัฒน์ สมโลก. 2020. Machine Learning สิ่งใกล้ตัวแห่งโลกยุคใหม่. [Online].

Available : https://www.depa.or.th/th/article-view/article11-2563.

อาทิตย์ สกุลเมือง. 2022. Natural Language Processing (NLP): เครื่องมือที่ช่วยให้คอมพิวเตอร์ เข้าใจภาษามนุษย์. [Online]. Available : <https://bigdata.go.th/big-data-101/what-is-natural-language-processing>.

DUSADEEVIROJ. 2020. REVIEW ETL PROCESS. [Online].

Available : <https://www.fusionsol.com/blog/review-etl-process>.

LDA. 2019. Recommendation System ระบบผู้ช่วยแนะนำที่รู้ใจเรามากกว่าตัวเราเอง. [Online].

Available : <https://www.ldaworld.com/recommendation-system-lda>.

สรรพโชค สิงหสุวรรณ. 2022. การใช้งาน Docker เบื้องต้น. [Online].

Available : <https://race.nstru.ac.th/home_ex/blog/topic/show/4200>

Pattanapong Cherthong. 2016. ทำความรู้จัก Docker และ Software Container. [Online].

Available : [https://medium.com/@teamteam](https://medium.com/@teamteam/%E0%B8%97%E0%B8%B3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-docker-%E0%B9%81%E0%B8%A5%E0%B8%B0-software-container-c6338629da11)

2013. เริ่มต้นการเขียนเว็บไซต์ ทำเว็บไซต์ด้วยภาษา Python กับ Django Framework. [Online].

Available : <https://www.amplysoft.com/knowledge/what-is-django-framework-python.html>

Pavarudh. 2022. ระบบจัดการฐานข้อมูล MariaDB คืออะไร. [Online].

Available : <https://km.cc.swu.ac.th/archives/4177>.

Pavarudh. 2021. ทำความรู้จักกับ Node.js. [Online].

Available : http://marcuscode.com/tutorials/nodejs/introducing-nodejs.

**ภาคผนวก ก**

**การพิจารณาปริมาณงานขั้นต่ำสำหรับโครงงานที่ 1**

การพิจารณาปริมาณงานขั้นต่ำสำหรับโครงงานที่ 1 นั้น เป็นการพิจารณาปริมาณความคืบหน้าของโครงงาน 1 ของนักศึกษาชั้นปีการศึกษาที่ 4 คณะวิศวกรรมศาสตร์ สาขาภาควิชาวิศวกรรมคอมพิวเตอร์ ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อให้ทราบว่าการพัฒนาโครงงาน 1 นั้นมีความคืบหน้าหรือปริมาณงานขั้นต่ำที่มากเพียงสำหรับการพิจารณาการให้ผ่านหรือไม่ ซึ่งนักศึกษาชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาภาควิชาวิศวกรรมคอมพิวเตอร์ จำเป็นต้องอภิปรายปริมาณงานของตนให้คณะกรรมการทราบโดยมีรายละเอียดดังนี้

1. ชื่อโครงงาน (ภาษาอังกฤษ) Curriculum output prediction from student academic data
2. ประเภทของโครงงาน

|  |  |  |
| --- | --- | --- |
| ระบุ | ประเภทโครงงาน | ปริมาณงานขั้นต่ำ |
|  | 1. HW + SW | ระบบที่เชื่อมโยงทุกส่วนเข้าด้วยกัน |
|  | 1. SW Dev | Complete Design Front End and Back End และ Implement 30% ของฟังก์ชันทั้งหมด |
|  | 1. Research | มีผลการทดลองจาก algorithm หลักด้วยข้อมูลที่ใช้งานจริง |

1. แจกแจงปริมาณงานที่ได้ทำในวิชาโครงงาน 1 อิงตามประเภทโครงงาน

* โครงสร้างของระบบ การออกแบบดังรูปที่ ก1 นั้นเป็นการออกภาพรวมของระบบแล้วได้นำไปเป็นรากฐานในการพัฒนาของโครงงาน

**Diagram, engineering drawing

Description automatically generated**

**รูป ก1 แผนภาพการทำงานโดยรวมของระบบ**

* ส่วนการ Implement ฟังก์ชัน

1. ฟังก์ชัน Map Subject Group สำเร็จ
2. ฟังก์ชัน similarity หลักสูตรปกติ สำเร็จ
3. ฟังก์ชัน similarity หลักสูตรต่อเนื่อง สำเร็จ
4. ฟังก์ชัน Analytic Data ไม่สำเร็จ
5. ฟังก์ชัน Sum Occupations ไม่สำเร็จ
6. ฟังก์ชัน Recommend Elective Subject ไม่สำเร็จ
7. API Simple CSV Generator สำเร็จ
8. API Request All Prediction Handler ไม่สำเร็จ
9. API Request All Outcome Predict of All Student ไม่สำเร็จ
10. API Calculate overall (old) Outcome ไม่สำเร็จ
11. API Request Elective Subject ไม่สำเร็จ
12. API Add Update Edit สำเร็จ
13. API Login Handler ไม่สำเร็จ

Implementation สำเร็จทั้งหมด 5 ฟังก์ชันจากทั้งหมด 13 ฟังก์ชันคิดเป็น 38.46%