

ระบบแนะนำแบบผสมสำหรับเว็บไซต์หางาน
**HYBRID WEB RECOMMENDATION SYSTEM FOR JOB
SEEKER**

วศิน เสริมสัมพันธ์
รหัสประจำตัว 60070157

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ คณะเทคโนโลยีสารสนเทศ
ปีการศึกษา 2563
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

HYBRID WEB RECOMMENDATION SYSTEM FOR JOB SEEKER

VASIN SERMSAMPAN

**A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR EDUCATION PROGRAM
THE DEGREE OF BACHELOR OF SCIENCE PROGRAM IN
DATA SCIENCE AND BUSINESS ANALYSIS
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2020

COPYRIGHT 2020

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

กิตติกรรมประกาศ

รายงานนี้สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ แนะนำให้คำปรึกษา และตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่งจากอาจารย์ที่ปรึกษา และอาจารย์ที่ปรึกษาร่วมทั้งสองท่าน ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย และจบลงได้ด้วยดี

1. Assoc Prof. Worapoj Kreesuradej

2. Dr. Nont Kanungsukkasem

นอกจากนี้ยังมีบุคคลท่านอื่น ๆ อีกที่ไม่ได้กล่าวไว้ ณ ที่นี้ ซึ่งให้ความกรุณาแนะนำ ในจัดทำรายงานฉบับนี้ ข้าพเจ้าจึงใคร่ขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลและให้ความเข้าใจการลงมือทำงานนี้ รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

วสิน เสริมสัมพันธ์

ผู้จัดทำรายงาน

วันที่ 10 พฤศจิกายน พ.ศ. 2561

ใบรับรองปริญญาโท ประจำปีการศึกษา 2563
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแนะนำแบบผสมสำหรับเว็บไซต์หางาน
ผู้จัดทำ วศิน เสริมสัมพันธ์
คณะ เทคโนโลยีสารสนเทศ
สาขาวิชา วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ

.....
(รศ.ดร.วรพจน์ กริสุระเดช)
อาจารย์ที่ปรึกษา

.....
(ดร.นนท์ คณิงสุขเกษม)
อาจารย์ที่ปรึกษาร่วม

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับรายงานการปฏิบัติงานสหกิจศึกษาฉบับนี้ เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ

ชื่อรายงาน	ระบบแนะนำแบบผสมสำหรับเว็บไซต์หางาน
ชื่อนักศึกษา	วสิน เสริมสัมพันธ์
รหัสนักศึกษา	60070157
สาขาวิชา	วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ
อาจารย์ที่ปรึกษา	รศ.ดร.วรพจน์ กริสุระเดช
อาจารย์ที่ปรึกษาร่วม	ดร.นนท์ คณิงสุขเกษม
ปีการศึกษา	2563

บทคัดย่อ

ในช่วงเวลาที่ผ่านมา ระบบแนะนำได้กลายเป็นที่นิยมและถูกนำไปใช้งานจากหลากหลายผู้ให้บริการ จากการประสบความสำเร็จในการลดจำนวนยูสเซอร์ที่เข้ามาค้นหาได้เป็นจำนวนมาก ด้วยการแนะนำตำแหน่งงานที่เหมาะสมกับบุคคลที่เข้ามาใช้งาน แต่ถึงอย่างนั้นเทคนิคที่ผู้ให้บริการส่วนใหญ่ใช้อยู่ ยังไม่มีศักยภาพมากพอที่จะแนะนำตำแหน่งงานที่เหมาะสมหรือสอดคล้องกับโปรไฟล์ผู้หางานได้ โดยผู้ให้บริการส่วนใหญ่จะมีการแนะนำเพียงฟิลด์หรือหมวดหมู่งานแทนแนะนำตำแหน่งงานเป็นกรณีเพื่อให้งานที่แนะนำสอดคล้องกับโปรไฟล์ยูสเซอร์มากที่สุด ด้วยประการทั้งปวงเราจึงมีวัตถุประสงค์โดย I) ออกแบบระบบรวบรวมชุดข้อมูลของตำแหน่งงาน จากเว็บไซต์หางานจ๊อบดีบี (JobsDB) และ โปรไฟล์ยูสเซอร์ที่มีความน่าเชื่อถือสูงจากเว็บไซต์ลิงก์ดอิน (LinkedIn) โดยใช้เทคนิคการสกัดข้อมูล (data scraping) II) ออกแบบโครงสร้างท่อส่งข้อมูลอัตโนมัติ (automated data pipeline) เพื่อจัดการ การไหลของข้อมูลที่มาจากการ สกัดและจัดการ กับข้อมูลเหล่านี้ให้อยู่ในรูปแบบที่เป็น โครงสร้าง (structured data) III) ออกแบบกรอบการทำงานของระบบแนะนำโดยอิงจากโปรไฟล์ทักษะวิชาชีพของยูสเซอร์ และ ข้อมูลตำแหน่งงาน IV) ออกแบบ โมเดลการจัดหมวดหมู่ตำแหน่งงาน โดยแยกประเทศตำแหน่งงานจากข้อมูลโปรไฟล์และตำแหน่งงาน เพื่อเพิ่มประสิทธิภาพในการแนะนำรวมลดระยะเวลาและทรัพยากรที่ต้องใช้ V) ดำเนินการประเมินเชิงประจักษ์ของความสามารถในการ ให้การแนะนำ โดยพิจารณาการกำหนดค่าที่แตกต่างกัน จากกรอบงานที่เสนอ VI) พัฒนาส่วนเว็บแอปพลิเคชันเพื่อให้บริการระบบแนะนำงาน

Project Title	Hybrid Web Recommendation System for Job Seeker
Student	Vasin Sermsampan
Student ID	60070157
Program	Data Science and Business Analytics
Advisor	Assoc Prof. Worapoj Kreesuradej
Sub Advisor	Dr. Nont Kanungsukkasem
Year	2020

Abstract

In the last years, the job recommendation system has become very popular. Due to its success in reducing the traffic of users who came to find a job position, By generating personalized job suggestions that are suitable for that person. However, most of them fail to recommend job vacancies that fit properly to the job seekers' profiles as they should be, Examples of problems, such as recommend only the part of the job field instead of suggesting it on a case-by-case for the job to be most consistent with the job seeker profile. We, therefore, have the following objectives: I) Collect job data sets from site JobsDB and professional profile from LinkedIn. Using data scraping techniques II) Design an automated data pipeline to manage the flow of data coming from data scraping and manipulation to structured data III) Design a recommended system framework based on jobs data and professional skills IV) Design job classification model by predicting the group of job positions based on profile and job information. To optimize the recommendation, as well as reducing the amount of time and resource that be used V) Conducting an empirical assessment of the ability to make recommendations By considering the different configurations From the proposed framework IIII) Develop a web application section to provide job guidance systems VI) Develop a web application to provide job recommendation service

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	V
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ระบบให้การแนะนำ (Recommendation System)	3
2.2 Apache Airflow	9
2.3 Docker	9
2.4 การสกัดข้อมูล (Data Scraping)	10
2.5 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)	11
2.6 การหาความสัมพันธ์ระหว่างสองสิ่ง	12
2.7 Support Vector Machine	13
2.8 เว็บแอปพลิเคชัน (Web Application)	13
2.9 เอพีไอ (API)	13
บทที่ 3 วิธีการทดลอง	16
3.1 สถาปัตยกรรมสำหรับไปป์ไลน์ข้อมูล	16
3.2 การพัฒนา API เพื่อให้บริการระบบแนะนำ	27
บทที่ 4 ผลการทดลอง	30
4.1 ข้อมูลการทดลอง	30
4.2 การทำนายกลุ่มสายงาน	33
4.3 ระบบแนะนำ	37
4.4 เว็บให้บริการระบบแนะนำ	39
บทที่ 5 สรุปผล	40
5.1 ปัญหาที่เกิดขึ้น	40
5.2 ทิศทางในอนาคต	40

สารบัญรูป

รูปที่	หน้า
2.1 [1, baptiste] อัลกอริทึมระบบผู้แนะนำประเภทต่างๆ	3
2.2 ภาพรวมการกรองแบบร่วมกัน	3
2.3 เมทริกซ์การโต้ตอบยูสเซอร์-ไอเทม	4
2.4 ภาพรวมของกระบวนการค้นหาวีธีการกรองร่วมกัน	5
2.5 [1, baptiste] วิธีกรองแบบ user-user	6
2.6 [1, baptiste] วิธีกรองแบบ item-item	7
2.7 ภาพรวมการกรองโดยอิงจากเนื้อหา	8
2.8 ภาพรวมของกระบวนการค้นหาวีธีการกรองโดยอิงจากเนื้อหา	8
2.9 comparing container and virtual machines	10
3.1 รูปภาพกรอบการทำงาน บริการที่ทำงานอยู่บน docker engine	16
3.2 รูปภาพกรอบการทำงาน การรวบรวมข้อมูลภายใต้ Airflow	17
3.3 it roadmap	19
3.4 ไฟล์รายการคำสั่งสก็ดข้อมูลบางส่วน	19
3.5 ไฟล์รายการคำสั่งสก็ดข้อมูลบางส่วน	21
3.6 รูปภาพโฟลว์การทำงานของจัดการข้อมูล	22
3.7 PCA vector ของตำแหน่งงานเมื่อทำความสะอาดข้อมูลแล้ว	23
3.8 PCA vector ของตำแหน่งงานเมื่อทำการให้น้ำหนักแก่คำแล้ว	24
3.9 PCA vector ของตำแหน่งงานผ่านเทคนิค word2vec	25
3.10 กลุ่มงานทางไอที	25
3.11 ตัวอย่างโค้ดไปป์ไลน์เทรนโมเดล	26
3.12 รูปภาพโฟลว์การทำงานระบบแนะนำ	27
3.13 context diagram	28
3.14 decomposition diagram	28
3.15 physical diagram	29
4.1 ตารางเปรียบเทียบจำนวนโปรไฟล์ในแต่ละสาขางาน	30
4.2 ตัวอย่างข้อมูลโปรไฟล์เปลี่ยนจากรูปแบบเจสันมาเป็นรูปแบบตาราง	31
4.3 ตัวอย่างข้อมูลโปรไฟล์ที่ถูกสก็ดมาในรูปแบบเจสัน	31
4.4 ตารางเปรียบเทียบจำนวนตำแหน่งงานในแต่ละสาขางาน	32

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.6 รายงานการแบ่งกลุ่มโดยใช้ข้อมูลโปรไฟล์	33
4.7 confusion matrix จากการทำนายโดยใช้ข้อมูลโปรไฟล์ผู้ใช้	34
4.8 ตาราง roc จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน	34
4.9 รายงานการแบ่งกลุ่มโดยใช้ข้อมูลโปรไฟล์	35
4.10 confusion matrix จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน	35
4.11 ตาราง roc จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน	36
4.12 ตัวอย่างคำร้องแนะนำตำแหน่งงาน	37
4.13 ตัวอย่างคำตอบรับแนะนำตำแหน่งงาน	38
4.14 แบบฟอร์มสำหรับกรอกข้อมูลโปรไฟล์ผู้ใช้	39
4.15 รายการตำแหน่งงานที่ถูกแนะนำ	39

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

การหางานที่เหมาะสมกับตัวผู้หางานนั้น เป็นสิ่งที่เป็ปัญหาอย่างช้านานจนถึงปัจจุบัน ไม่ว่าจะเป็นกลุ่มผู้เรียนจบใหม่ นักศึกษาฝึกงาน หรือแม้กระทั่งผู้ที่ต้องการเปลี่ยนงาน ปัญหานี้มีปัจจัยหลายส่วนและหลายกลุ่ม เช่น กลุ่มของผู้ฝึกงานและผู้ที่ยื่นจบใหม่ มักยังไม่ทราบความต้องการของตนเองว่าต้องการทำงานในแขนงไหน ตนเองถนัดกับสิ่งใด และปัญหาภาพรวมที่พบเจอเยอะที่สุดคือความยุ่งยากในการหางาน โดยที่ผู้หางานจำเป็นต้องค้นหางานด้วยตนเองทีละตำแหน่งและอ่านรายละเอียดตำแหน่งเหล่านั้นว่ามีความต้องการตรงกับความสามารถหรือไม่ แต่เมื่อเลือกตำแหน่งงาน ได้ก็ไม่ได้หมายความว่างานเหล่านั้นจะเหมาะกับตัวผู้หางาน นั้นทำให้ต้องกลับมาค้นหาด้วยวิธีแบบเดิมอีกรอบ จากที่กล่าวมาจะเห็นว่าปัญหาเหล่านี้เป็นปัญหาที่สำคัญและยังเจออยู่ไม่ว่ายุคไหนก็ตาม

ทั้งนี้บางเว็บไซต์หางานก็ได้มีการแก้ปัญหาเหล่านี้ด้วยการเพิ่มระบบคัดกรองและแนะนำตำแหน่งงานขึ้นมา อย่างเช่นเว็บไซต์จ๊อบบีเค (JobBKK) ที่มีระบบคัดกรองแบ่งเป็นประเภทที่ผู้หางานต้องการเช่น สถานที่ เงินเดือนขั้นต่ำ ประสบการณ์ อีกทั้งยังมีระบบจับคู่งานกับผู้หางาน แต่ถึงจะมีความละเอียดในการค้นหาและคัดกรอง แต่ต้องแลกมาด้วยความยุ่งยากและเสียเวลาเกินความจำเป็นในการค้นหาแต่ละครั้ง อีกทั้งระบบจับคู่งานกับผู้หางานเป็นการจับคู่แค่ในหมวดหมู่งานนั้นเท่านั้น ไม่ได้จับคู่งานโดยอิงจากความสามารถจริง ๆ ของผู้หางานเป็นเคสต่อเคส

ด้วยปัญหาดังกล่าวและตัวอย่างเว็บหางานส่วนใหญ่ที่พบ ทางผู้จัดทำจึงได้คิดและออกแบบระบบที่สามารถจับคู่ทักษะวิชาชีพของผู้หางานกับตำแหน่งงานให้มีความสอดคล้องและมีประสิทธิภาพมากที่สุด โดยคำนึงการใช้งานได้จริง เพื่อช่วยแก้ปัญหาการหางานในปัจจุบัน ไม่ว่าจะเป็นการไม่ทราบความต้องการของตนเอง หรือความซับซ้อนและยุ่งยากในการหางาน โดยระบบที่ผู้จัดทำขึ้นมาคือระบบให้การแนะนำ (Recommendation System) เพื่อมาช่วยสนับสนุนเว็บแอปพลิเคชัน (Web Application) จับคู่ผู้หางานกับตำแหน่งงาน ซึ่งเทคนิคที่ใช้ในการสร้างระบบแนะนำนั้น ทางผู้จัดทำได้ใช้เทคนิคการกรองแบบอิงเนื้อหา (Content Based Filtering) ซึ่งเป็นการแนะนำโดยทำการดูเนื้อหาและลักษณะของงานว่ามีคำสำคัญ (Keyword) และแนะนำงานที่มีลักษณะคล้ายกับโปรไฟล์ของผู้หางานมากที่สุด โดยคำนึงถึงทักษะวิชาชีพของผู้หางานเป็นหลัก ในการหาคำสำคัญของงานได้ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing) เข้ามาช่วยในการเข้าใจและแบ่งคำเพื่อนำไปวิเคราะห์หาคำสำคัญต่อไป

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาออกแบบ โครงสร้างทอส่งข้อมูลอัตโนมัติ เพื่อจัดการการไหลของข้อมูลที่มาจากการสัคดและจัดการกับข้อมูลเหล่านี้ให้อยู่ในรูปแบบที่เป็นโครงสร้าง

2. เพื่อออกแบบและพัฒนาระบบแนะนำตำแหน่งงาน โดยผสมผสานการอ้างอิงจากทักษะของ โปรไฟล์ยูสเซอร์ และระหว่างยูสเซอร์กับตำแหน่งงาน
3. ประยุกต์ระบบแนะนำตำแหน่งงานกับเว็บแอปพลิเคชัน ที่จะเปิดให้ใช้สำหรับหางานจาก ตำแหน่งงานที่สกัดมาจากเว็บไซต์ชั้นนำ
4. ออกแบบและพัฒนาเว็บแอปพลิเคชันหางานที่สามารถจับคู่โปรไฟล์ยูสเซอร์กับตำแหน่งงาน ได้

1.3 ขอบเขตของงานวิจัย

1. พัฒนาระบบแนะนำตำแหน่งงาน โดยใช้การกรองแบบอิงเนื้อหา (Content Based Filtering) โดยผสมระหว่างยูสเซอร์เบสและจ๊อบเบส
2. พัฒนาเว็บแอปพลิเคชันหางาน ที่เชื่อมต่อกับระบบแนะนำตำแหน่งงาน
3. ข้อมูลที่ใช้ในการสร้างระบบในส่วน of ข้อมูลดั้งเดิม โปรไฟล์ได้ใช้ข้อมูลจาก ลิงค์ดอิน (Linkedin) และส่วนของตำแหน่งงานได้ใช้ข้อมูลจาก อินดีด (Indeed)
4. ข้อมูลที่สกัดและการสร้างโมเดลรวมถึงระบบจะเป็นภาษาอังกฤษทั้งหมด
5. ขอบเขตของการแนะนำตำแหน่งงานจะอยู่ในขอบเขตของไอที

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ทางเราได้สร้างระบบแนะนำตำแหน่งงานขึ้นมาเพื่อช่วยในลดปัญหาความยุ่งยากซับซ้อนและ เสียเวลากับวิธีหางานในปัจจุบัน โดยนำเสนอการจับคู่ระหว่างโปรไฟล์และตำแหน่งงานที่เหมาะสม กัน

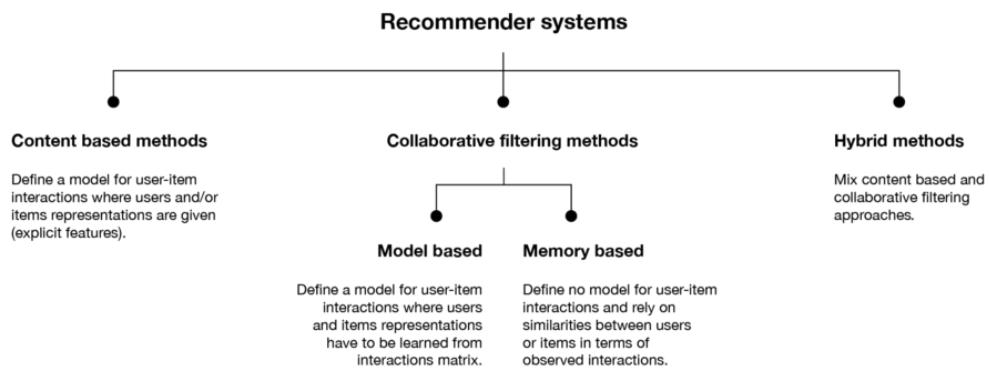
1. ช่วยให้ผู้ที่ยังไม่มีงานทำในปัจจุบันสามารถหางานได้ขึ้นผ่านขั้นตอนการจับคู่ตำแหน่งงาน ที่ทางเราสร้างขึ้น
2. เพื่อสร้างฐาน ข้อมูลที่เป็น อนาคต ในการนำข้อมูลตำแหน่งงาน ไปวิเคราะห์และ ใช้งาน ใน อนาคต
3. เพื่อวิจัยและค้นคว้าการสร้างระบบแนะนำที่มีประสิทธิภาพและต่อยอดได้ในอนาคต

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

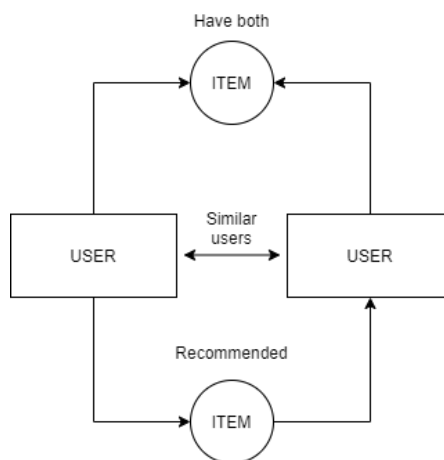
2.1 ระบบให้การแนะนำ (Recommendation System)

ระบบให้การแนะนำ [1, baptiste] เป็นระบบสนับสนุนการตัดสินใจที่จะให้การแนะนำสินค้าหรือบริการที่มีความเหมาะสมกับรูปแบบและพฤติกรรมของลูกค้าหรือยูสเซอร์แต่ละคน โดยอาศัยข้อมูลของยูสเซอร์งานร่วมกับข้อมูลประกอบภายนอกมาใช้ในการวิเคราะห์คัดกรอง ให้ได้สิ่งที่มีความหมายที่เหมาะสมกับยูสเซอร์งาน โดยมีเทคนิคแบ่งย่อยได้เป็นสองชนิดหลัก ๆ คือ การกรองแบบอิงเนื้อหา (Content Based Filtering) การกรองแบบร่วม (Collaborative Filtering) และการกรองแบบผสม (Hybrid Recognition) [3]



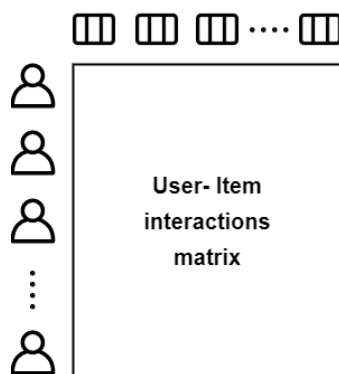
รูปที่ 2.1 [1, baptiste]อัลกอริทึมระบบผู้แนะนำประเภทต่างๆ

2.1.1 การกรองแบบร่วมกัน (Collaborative Filtering)



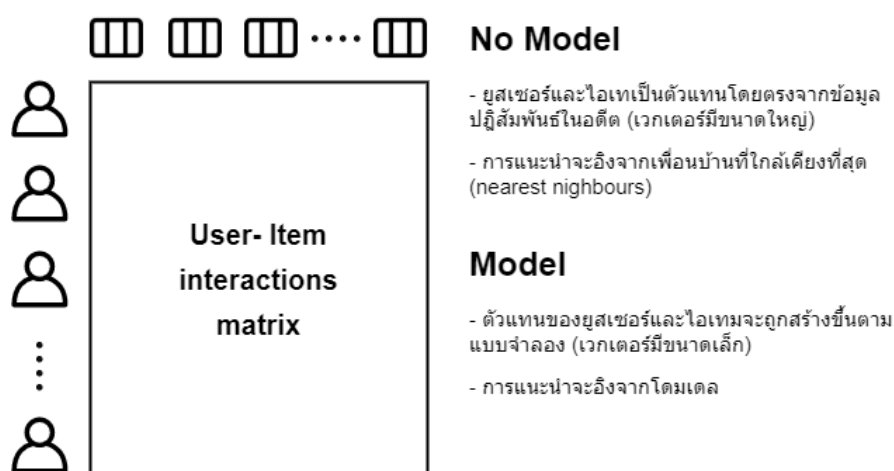
รูปที่ 2.2 ภาพรวมการกรองแบบร่วมกัน

การกรอแบบร่วม เป็นเทคนิคและนำโดยใช้ข้อมูลการโต้ตอบในอดีตที่ถูกบันทึกไว้ระหว่างยูสเซอร์และรายการเพื่อสร้างคำแนะนำใหม่ ๆ โดยข้อมูลเหล่านี้จะถูกเก็บไว้ในรูปแบบเมทริกซ์ที่เรียกว่า "user-item interactions matrix"



รูปที่ 2.3 เมทริกซ์การโต้ตอบยูสเซอร์ไอเทม

แนวคิดหลักที่เป็นกฎของการกรอแบบร่วมกันคือข้อมูลการโต้ตอบในอดีตในปริมาณที่มากเพียงพอจะสามารถตรวจหาความสัมพันธ์กันของยูสเซอร์กับไอเทมหรือยูสเซอร์กับยูสเซอร์ได้ คลาสของอัลกอริทึมการกรอแบบร่วมกันแบ่งออกเป็นสองประเภทย่อยที่โดยทั่วไปเรียกว่า อิงจากหน่วยความจำ (memory based) และ อิงจากโมเดล (model based) ในวิธีนี้การทำงานจะไม่มี การสันนิษฐานแบบจำลองแฝงแต่อัลกอริทึมจะทำงานโดยตรงกับข้อมูลการตอบโต้ยูสเซอร์และไอเทม ตัวอย่างเช่น ยูสเซอร์จะแสดงผลลัพธ์ของข้อมูลไอเทมเพื่อนบ้านที่ใกล้เคียงที่สุดเพื่อใช้ในการแนะนำ วิธีนี้มีความอคติต่ำในทางทฤษฎีแต่มีความแปรปรวนสูง และการอิงจากโมเดล วิธีนี้ถือว่าเป็นรูปแบบปฏิสัมพันธ์แฝง โมเดลนี้จะได้รับการฝึกให้สร้างค่าการตอบโต้ยูสเซอร์ไอเทมใหม่จากการแสดงยูสเซอร์และไอเทมของตัวเอง จากนั้นคำแนะนำใหม่สามารถทำได้โดยใช้โมเดลนี้ การทำนายจะมีความหมายทางคณิตศาสตร์ที่ยากต่อการตีความโดยมนุษย์ ดังนั้นวิธีการนี้ถือว่าเป็นวิธีที่มีความอคติสูงแต่มีความแปรปรวนต่ำ

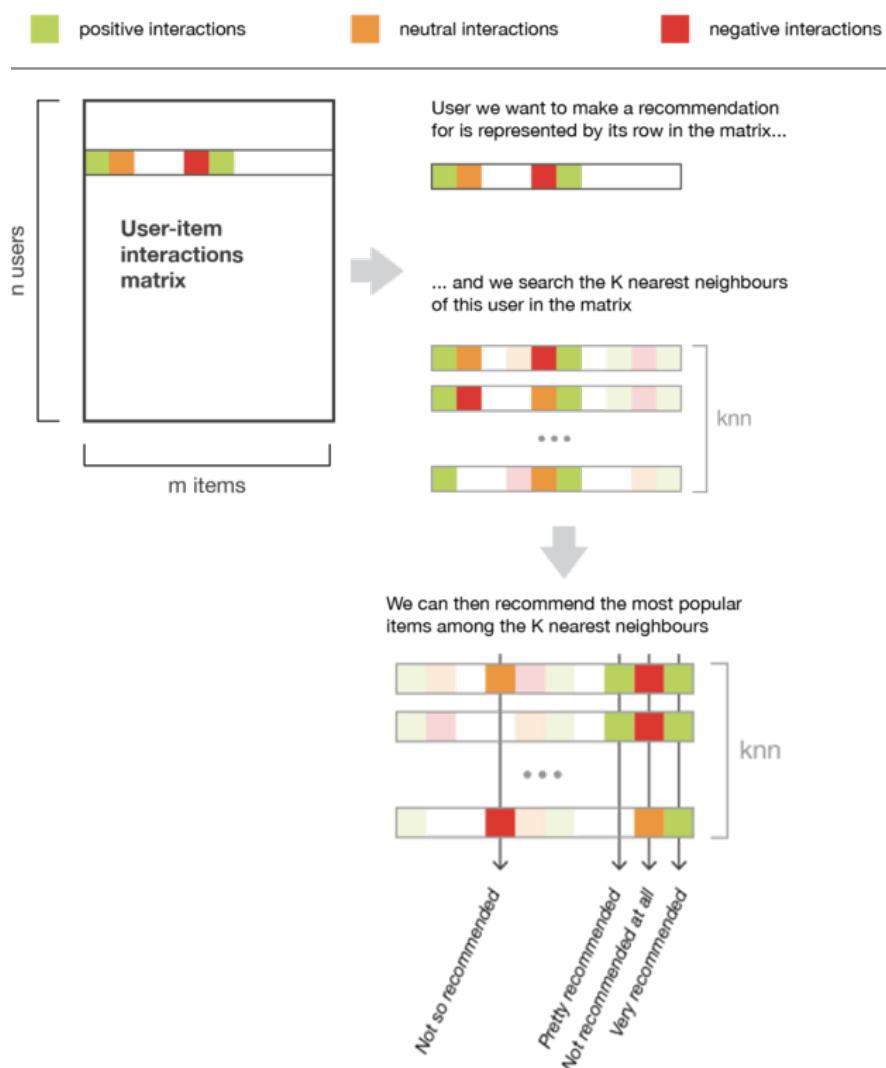


รูปที่ 2.4 ภาพรวมของกระบวนการค้นวิธีการกรองร่วมกัน

ข้อได้เปรียบหลักของการกรองแบบร่วมกันคือไม่ต้องการข้อมูลเกี่ยวกับยูสเซอร์หรือรายการเป็นตัวตั้งต้น ดังนั้นจึงสามารถใช้ได้ในหลายสถานการณ์ ยิ่งไปกว่านั้นยิ่งข้อมูลการโต้ตอบมีมากขึ้นเท่าใด คำแนะนำใหม่ก็จะยิ่งถูกต้องมากขึ้นเท่านั้น แต่ถึงอย่างนั้นเนื่องจากไม่ต้องการข้อมูลในการตั้งต้นการพิจารณาข้อมูลเพื่อแนะนำจึงเกิดปัญหา นั่นคือ "cold start problem" ในทางปฏิบัติจึงเป็นไปได้ที่จะแนะนำสิ่งใดให้กับยูสเซอร์ใหม่หรือนำรายการใหม่ให้กับยูสเซอร์ ยูสเซอร์และรายการที่มีข้อมูลการตอบโต้ที่น้อยเกินไปจะทำให้การแนะนำคลาดเคลื่อนเป็นอย่างมาก, ปัญหานี้สามารถแก้ไขได้หลายวิธีอย่างเช่น การแนะนำไอเทมแบบสุ่มให้กับยูสเซอร์ใหม่หรือนำไอเทมใหม่กับยูสเซอร์แบบสุ่ม หรือการแนะนำไอเทมยอดนิยมให้กับยูสเซอร์ใหม่หรือไอเทมใหม่ให้กับยูสเซอร์ส่วนใหญ่ หรือการแนะนำชุดรายการให้กับยูสเซอร์ใหม่หรือรายการใหม่ไปยังกลุ่มยูสเซอร์ที่หลากหลาย เป็นต้น

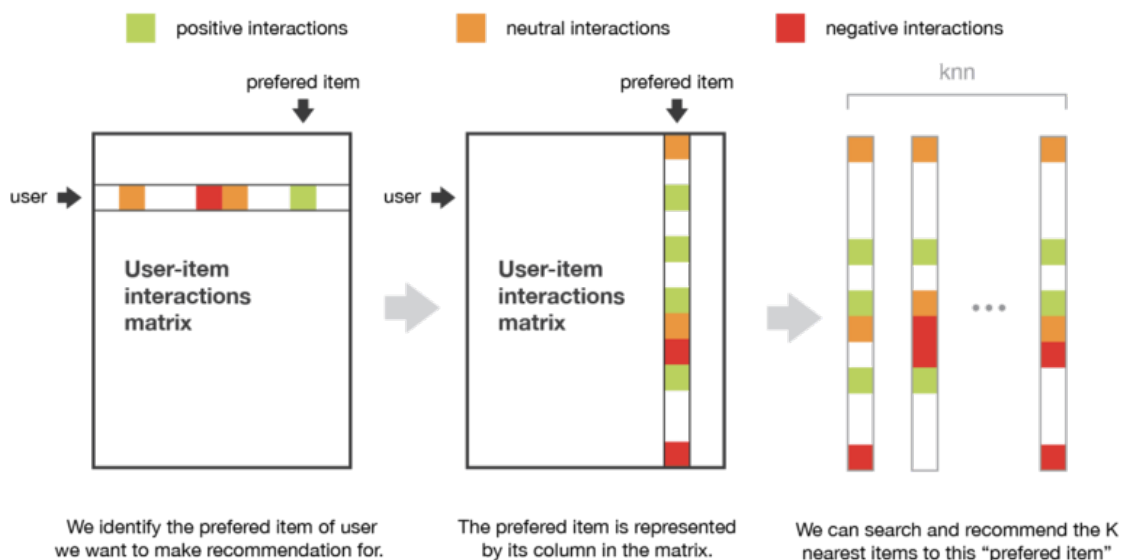
2.1.1.1 Memory Based

2.1.1.1.1 User-user ในการแนะนำให้กับยูสเซอร์ วิธี user-user จะพยามระบุผู้ใช้ที่มีข้อมูลโปรไฟล์การโต้ตอบที่คล้ายคลึงกันมากที่สุด (เพื่อนบ้านที่ใกล้ที่สุด) เพื่อแนะนำรายการที่ได้รับ ความนิยมมากที่สุดให้บรรดาเพื่อนบ้านเหล่านี้ (หมายถึงยูสเซอร์ใหม่) วิธีนี้เรียกว่า “ผู้ใช้เป็นศูนย์กลาง” (user-centered) สมมติว่าเราต้องการให้คำแนะนำสำหรับยูสเซอร์ใหม่ ขั้นแรกทุกคนจะถูกแทนที่ ด้วยเวกเตอร์ของการโต้ตอบกับรายการ หลังจากนั้นเราสามารถคำนวณความคล้ายคลึงกันระหว่าง ยูสเซอร์ที่เราสนใจกับยูสเซอร์อื่น ๆ ทุกคน การวัดความคล้ายคลึงกันคือการที่ยูสเซอร์สองคนมีปฏิ สัมพันธ์คล้ายคลึงกันในรายการเดียวกันนั้นหมายความว่าควรได้รับการพิจารณาว่าอยู่ใกล้กัน เมื่อคํ านวนความคล้ายคลึงกับยูสเซอร์ทุกคนแล้ว เราสามารถเก็บ k nearest neighbour ไว้ให้กับยูสเซอร์ของ เราจากนั้นแนะนำรายการที่ได้รับความนิยมมากที่สุดในบรรดารางกรเหล่านี้



รูปที่ 2.5 [1, baptiste] วิธีการแบบ user-user

2.1.1.1.2 Item-item การให้คำแนะนำใหม่แก่ผู้เสนอแนวคิดของวิธี item-item คือการหารายการที่สอดคล้องกับรายการที่ผู้เสนอรายการตอบได้เป็นบวก (position) สองรายการซึ่งจะถือว่าคล้ายกันหากผู้ใช้ส่วนใหญ่ที่มีปฏิสัมพันธ์กับทั้งคู่ทำในลักษณะเดียวกัน วิธีนี้เรียกว่า "item-centered"

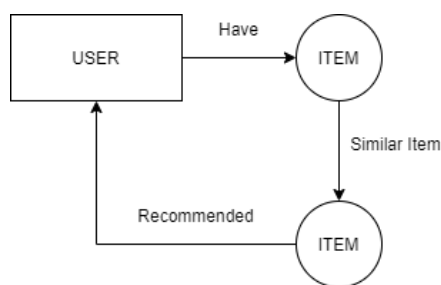


รูปที่ 2.6 [1, baptiste] วิธีการแบบ item-item

2.1.1.2 Model Based

การกรองแบบร่วมกัน โดยใช้โมเดล อาศัยข้อมูลการโต้ตอบของผู้เสนอไอเทม และใช้โมเดลในการอธิบายข้อมูลการโต้ตอบเหล่านี้ ตัวอย่างเช่น อัลกอริทึมการแยกตัวประกอบเมทริกซ์ (matrix factorization) โดยสลายเมทริกซ์การโต้ตอบของผู้เสนอไอเทมที่มีขนาดใหญ่และกระจายให้ให้เป็นตารางเมทริกซ์ที่มีขนาดเล็กและหนาแน่นจำนวนสองเมทริกซ์

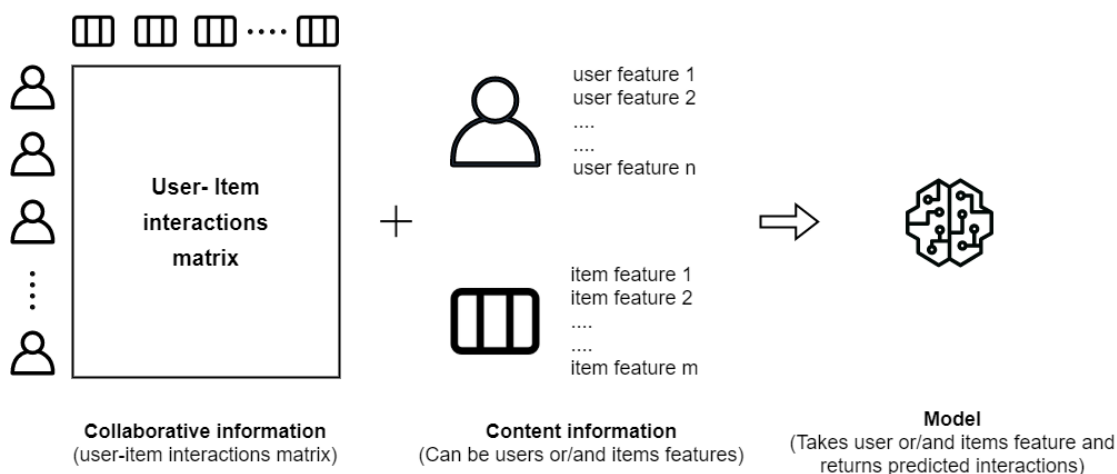
2.1.2 การกรองแบบอิงเนื้อหา (Content Based Filtering)



รูปที่ 2.7 ภาพรวมการกรองโดยอิงจากเนื้อหา

การกรองโดยอิงจากเนื้อหาต่างจากการกรองแบบร่วมกันที่อาศัยข้อมูลการตอบโต้ยูสเซอร์และไอเทม การกรองโดยอิงจากเนื้อหาใช้ข้อมูลเพิ่มเติมเกี่ยวกับยูสเซอร์ และ/หรือ ไอเทม ยกตัวอย่างเช่นระบบแนะนำภาพยนตร์ ข้อมูลเพิ่มเติมนี้อาจจะเป็น เพศ, อายุ, งาน หรือข้อมูลส่วนตัวอื่น ๆ ของยูสเซอร์ เพราะฉะนั้นแนวคิดนี้คือการพยายามสร้างแบบจำลองตามคุณลักษณะเพื่อพยายามอธิบายยูสเซอร์และไอเทม ตัวอย่างเช่น เมื่อพิจารณาภาพยนตร์ เราจะพยายามจำลองความจริงที่ว่าผู้หญิงมักจะให้คะแนนภาพยนตร์บางเรื่องตามที่เพศหญิงชอบ และผู้ใช้จะ ให้คะแนนภาพยนตร์บางเรื่องตามที่เพศตัวเองชอบ เป็นต้น หากทำการพิจารณาจากตัวอย่างข้างต้นเราเพียงแค่อัปโหลดเพศเราก็สามารถแนะนำภาพยนตร์ที่เพศนั้นๆ ชอบได้

วิธีการอิงจากเนื้อหานี้จะไม่ประสบปัญหา "cold start problem" น้อยกว่าวิธีการอิงแบบร่วมกัน



รูปที่ 2.8 ภาพรวมของกระบวนการค้นวิธีการกรองโดยอิงจากเนื้อหา

ยูสเซอร์ใหม่หรือไอเทมใหม่สามารถอธิบายได้ตามลักษณะ (เนื้อหา) ของตัวมันเอง และคำแนะนำที่เกี่ยวข้องสามารถทำได้สำหรับเอนทิตีใหม่เหล่านี้ เฉพาะยูสเซอร์ใหม่หรือผู้ใช้ใหม่ที่มีคุณสมบัติที่ไม่เคยเจอมาก่อนเท่านั้นที่ได้รับผลกระทบจากข้อเสียนี้ แต่เมื่อมีข้อมูลมากเพียงพอปัญหานี้จะหมดไป

2.1.3 ระบบให้คำแนะนำแบบผสม (Hybrid Recommendation)

ระบบการแนะนำแบบผสม เป็นการประยุกต์ระบบแนะนำหลายหรือหลายข้อมูลเข้าด้วยกันเพื่อเพิ่มประสิทธิภาพในการทำนาย และแก้ไขปัญหาคือของของแต่ละเทคนิค

2.2 Apache Airflow

Apache airflow [24] เป็นแพลตฟอร์มการจัดการเวิร์กโฟลว์แบบโอเพนซอร์สสามารถเขียนโปรแกรมเพื่อกำหนดเวลาขั้นตอนการทำงานและตรวจสอบผ่านอินเทอร์เฟซผู้ใช้ได้ Airflow เขียนด้วยภาษา python และเวิร์กโฟลว์ถูกสร้างผ่านสคริปต์ python โดยได้รับการออกแบบภายใต้หลักการ "configuration as code" แม้ว่าแพลตฟอร์มอื่น ๆ ที่ใช้หลักการนี้จะอยู่ภายใต้มาร์กอัพ เช่น XML แต่การใช้ python ช่วยให้นักพัฒนานำเข้าไลบรารีและคลาสเพื่อช่วยในการสร้างเวิร์กโฟลว์ได้ง่ายและมีประสิทธิภาพมากยิ่งขึ้นกว่าการตั้งค่าโคด ๆ แบบ XML

Airflow ใช้กราฟ acyclic กำกับ (DAG) เพื่อจัดการระเบียบเวิร์กโฟลว์งาน และการอ้างอิงถูกกำหนดไว้ใน python จากนั้น airflow จะจัดการตั้งเวลาและดำเนินการ DAG ตามเวลาที่กำหนด (เช่น รายชั่วโมงรายวัน) หรือตามทริกเกอร์เหตุการณ์ภายนอก

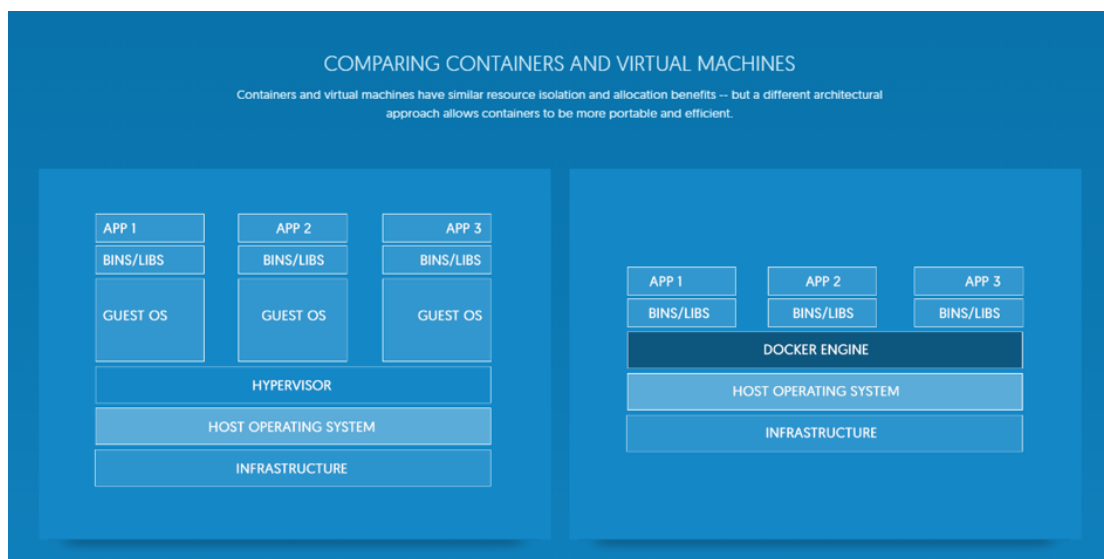
2.3 Docker

Docker [25] เป็นเอ็นจินที่มียุทธศาสตร์การทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่องเซิร์ฟเวอร์เพื่อใช้ในการรันเซอร์วิสที่ต้องการ มีการทำงานคล้ายคลึงกับเครื่องเสมือน (virtual machine) เช่น VMWare, VirtualBox, XEN, KVM แต่ข้อแตกต่างที่ชัดเจนคือ เครื่องเสมือนที่กล่าวมาจำเป็นต้องจำลองทั้งระบบปฏิบัติการ (OS) เพื่อใช้งานและหากต้องการใช้บริการใด ๆ จำเป็นต้องติดตั้งเพิ่มบนระบบปฏิบัติการนั้น แต่สำหรับ docker แล้วจะใช้สิ่งที่เรียกว่าคอนเทนเนอร์ ในการจำลองสภาพแวดล้อมขึ้นมาเพื่อใช้งานสำหรับ 1 บริการ ที่ต้องการใช้งานเท่านั้น โดยไม่ต้องมีส่วนระบบปฏิบัติการเข้าไปเกี่ยวข้องด้วยเหมือนเครื่องเสมือนอื่น ๆ

โดย docker นั้นเป็นที่รู้จักกันอย่างแพร่หลายในช่วง 1-2 ปีที่ผ่านมา เนื่องจากสามารถใช้งานได้สะดวกและตอบสนองความต้องการของผู้พัฒนาโปรแกรมหรือผู้ดูแลระบบ

Docker image เป็นตัวต้นแบบของคอนเทนเนอร์ซึ่งภายในจะประกอบด้วยแอปพลิเคชันต่าง ๆ ที่มีการติดตั้งไว้เพื่อนำมาใช้งานสำหรับบริการนั้น ๆ รวมทั้งมีการตั้งค่าต่าง ๆ ไว้อย่างเรียบร้อย จากนั้นจึงนำมาสร้างเป็นอิมเมจบนรีจิสทรีเพื่อนำมาใช้งานทั้งนี้ผู้ใช้งานสามารถสร้าง docker image ของตัวเองได้อีกด้วย

Docker container เป็นกล่องเหมือนซึ่งนำ docker image มาติดตั้งเพื่อให้สามารถใช้งานบริการที่ต้องการได้จากอิมเมจนั้น ๆ โดยในคอนเทนเนอร์แต่ละตัวจะมีการใช้งาน RAM, CPU ไฟล์ตั้งค่าต่าง ๆ เป็นของตัวเอง



รูปที่ 2.9 comparing container and virtual machines

2.4 การสกัดข้อมูล (Data Scraping)

การสกัดข้อมูล [12] เป็นเทคนิคในการเข้าถึงข้อมูลจากเว็บไซต์เพื่อที่หาและสกัดข้อมูลที่ต้องการในการสกัดข้อมูลจากข้อความที่ดึงมาจากเว็บไซต์สามารถใช้ไลบรารี beautifulsoup ของภาษา python เพื่อช่วยในการสกัดข้อมูลให้มีความง่ายขึ้นได้ กรณีที่เว็บไซต์ที่ทำงาน โดยการเรนเดอร์หน้าเพจทั้งหน้าแล้วส่งมาให้ยูสเซอร์ (client) เราสามารถดึงข้อมูลของทั้งหน้ามาใช้ได้โดยตรงและสกัดข้อมูลจากที่กล่าวมาข้างต้น แต่บางเว็บไซต์ที่มีการทำงานแบบฝั่งไคลเอนต์ มีการแสดงผลข้อมูลเป็นแบบ Asynchronous ซึ่งทำให้ข้อมูลปรากฏขึ้นไม่พร้อมกัน โดยจะขึ้นอยู่กับการทำงานของยูสเซอร์เช่น คลิ๊กเปิด เลื่อนลงเพื่อโหลดฟีด จะไม่สามารถดึงข้อมูลทั้งหน้าได้จำเป็นต้องจำเป็นต้องแก้ปัญหาโดยทำการจำลองเบราว์เซอร์เพื่อจำลองการทำงานของยูสเซอร์ขึ้นมา

2.4.1 Puppeteer

Puppeteer เป็นไลบรารี Node ซึ่งมีอีพีไอระดับสูงเพื่อควบคุม Chrome หรือ Chromium ผ่านหน้าพัฒนา โดย puppeteer จะทำการรันเป็นเบราว์เซอร์ล่องหน (headless browser) หรือคือไม่มีอินเทอร์เฟซผู้ใช้งานแบบกราฟิก โดยเบราว์เซอร์ล่องหนสามารถควบคุมหน้าเว็บได้โดยอัตโนมัติในสภาพแวดล้อมที่คล้ายกับเว็บเบราว์เซอร์ แต่ดำเนินการผ่านอินเทอร์เฟซบรรทัดคำสั่งหรือใช้การสื่อสารผ่านเครือข่าย มีประโยชน์เป็นอย่างมากสำหรับการทดสอบหน้าเว็บเนื่องจากสามารถแสดงผลและทำความเข้าใจ HTML ได้ อีกทั้งยังสามารถประยุกต์ใช้เบราว์เซอร์ล่องหนในการสกัดข้อมูลจากเว็บไซต์ที่ต้องการผ่านการจำลองเสมือนเพื่อเข้าถึงข้อมูลที่ต้องการ

2.5 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)

การประมวลผลภาษาธรรมชาติ (NLP) [7] เป็นแขนงหนึ่งของสาขาปัญญาประดิษฐ์ (Artificial Intelligence) ที่ทำให้เครื่องจักรมีความสามารถในการอ่านทำความเข้าใจและเข้าใจความหมายของภาษามนุษย์ได้ กล่าวคือ NLP แสดงถึงการจัดการภาษามนุษย์โดยอัตโนมัติเช่นการพูด ข้อความ หรือแม้กระทั่งแนวคิดที่สนใจ โดยได้มีการนำไปประยุกต์ใช้ในแขนงมากมายเช่น ช่วยในการทำความเข้าใจและคาดการณ์กลุ่มของยูสเซอร์จากโปรไฟล์ของยูสเซอร์เหล่านั้น เป็นต้น

2.5.1 Word Embedding

Word Embedding [8] คือการจับบริบทของคำในเอกสารที่มีความคล้ายคลึงกับคำอื่น ๆ และแปลงคำให้เป็นตัวเลขในรูปแบบเวกเตอร์ โดยถือเป็นหนึ่งในวิธีการสร้างฟีเจอร์จากคำวิธีหนึ่ง โดยทำการลดขนาดเวกเตอร์ลงด้วย เช่น ทำการ word embedding กับคำว่า "I, liked, the, hotel" เราจะได้เวกเตอร์ออกมาคือ $I[0.3, 0.2, 0.8, 0.1]$, $liked[0.4, 1.2, 0.1, 0.9]$, $the[1.3, -2.1, 0, 1.2]$, $hotel[0.5, 1.4, 0.3, -0.4]$ เป็นต้น

2.5.1.1 Word2Vec

Word2Vec [8] Pre-trained weight model หรือแบบจำลองน้ำหนักที่ผ่านการเทรนมาแล้วหน้าแล้ว word2vec มีสองแบบที่สามารถใช้เพื่อทำ word embeddings คือ CBOW และ Skip-gram

1. **Bag-of-Words Models (CBOW)** โมเดลนี้จะทำนายคำถัดไปโดยอ้างอิงจาก n คำก่อนหน้า และ n คำต่อท้ายคำถัดไป ตัวอย่างเช่นประโยคต่อไปนี้

Lorem ipsum dolor sit amet

CBOW จะทำนายคำ *dolar* โดยให้อินพุต $n = 2$ ก่อนและหลังคำซึ่งจะได้ว่า *Lorem, ipsum, sit* และ *amet* คำเหล่านี้เรียกว่าบริบทของคำเป้าหมายและปริมาณจะเป็นพารามิเตอร์ของแบบจำลอง

2. **Skip-gram** จากที่จะคาดเดาตามบริบทของคำ skip-gram จะทำนายบริบทแค่คำเดียว จากตัวอย่างก่อนหน้านี้เมื่อการทำนายด้วย skip-gram ตัว skip-gram จะพยายามทำนายคำว่า *Lorem, ipsum, sit* และ *amet* โดยมีคำว่า *dolar* เป็นอินพุต

2.5.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TFIDF [17] ใช้เพื่อชั่งน้ำหนักของคำสำคัญ (Keyword) ในเอกสารใด ๆ เพื่อกำหนดความสำคัญให้กับคำสำคัญเหล่านั้นตามจำนวนครั้งที่ปรากฏในเอกสาร หรือก็คือยิ่งคะแนน $TF * IDF$ (น้ำหนัก) สูงเท่าไรหาคำนั่นก็จะสำคัญเท่านั้น ในทุกคำหรือคำศัพท์แต่ละคำจะมีคะแนน TF และ IDF อยู่เสมอ ผลคูณของคะแนน TF และ IDF ของคำหนึ่งจะเรียกว่าน้ำหนัก $TF*IDF$ ของคำนั้น ๆ

ความถี่ (TF: Term Frequency) ของคำคือจำนวนครั้งที่ปรากฏในเอกสาร เมื่อทราบถึง TF แล้ว เราจะสามารถบอกได้ว่ามีคำนั้นปรากฏในเอกสารบ่อยเท่าใด

$$TF(t) = \text{จำนวนครั้งที่ } t \text{ ปรากฏบนเอกสาร} / \text{จำนวนคำทั้งหมดในเอกสาร} \quad (2.1)$$

ความถี่เอกสารผกผัน (IDF: Inverse Document Frequency) ของคำคือการวัดความสำคัญของคำ เหล่านั้นในคลังข้อมูลคำ (Corpus) ทั้งหมด

$$IDF(t) = \log_e(\text{จำนวนเอกสารทั้งหมด} / \text{จำนวนเอกสารที่มีคำศัพท์อยู่ในนั้น}) \quad (2.2)$$

$$W_{x,y} = TF_{x,y} \cdot \log \left(\frac{N}{DF_x} \right) \quad (2.3)$$

$TF_{x,y}$ = frequency of x in y

DF_x = number of documents containing x

N = total number of document

เมื่อเราทำ TF-IDF แล้วเราสามารถเห็นความสำคัญของข้อความสำคัญได้

2.6 การหาความสอดคล้องระหว่างสองสิ่ง

ในการหาความสอดคล้องระหว่างสองสิ่ง [10] เราสามารถทำได้โดยใช้เทคนิคความคล้ายคลึงของโคไซน์ (Cosine Similarity)

$$sim_{A,B} = \frac{A \cdot B}{||A|| ||B||} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.4)$$

ตัวอย่างข้อความ "backend developer", "senior software developer" เมื่อนำมาเปลี่ยนเป็นเมตริกซ์ เทคนิคการนับคำ (count vectorizer) จะได้เมตริกซ์ [1, 1, 0, 0] และ [0, 1, 1, 1] หลังจากมาหาความสอดคล้องจากการแทนค่าจากสมการดังกล่าวจะได้

$$sim_{A,B} = \frac{(1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1)}{\sqrt{(1^2 + 1^2 + 0^2 + 0^2)} \sqrt{(0^2 + 1^2 + 1^2 + 1^2)}} \quad (2.5)$$

$$sim_{A,B} = \frac{1}{\sqrt{2} \sqrt{3}} \quad (2.6)$$

ดังนั้นแล้วความสอดคล้องระหว่าง "backend developer" และ "senior software developer" คือ 0.408

2.7 Support Vector Machine

Support Vector Machine (SVM) [11] เป็นเทคนิค Pattern Recognition แบบ Supervised Learning ถูกใช้ในเคส Classification และ Regression โดยภายในงานนี้ได้ถูกใช้เพื่อ Classification ตำแหน่งงาน ด้วยการสร้าง Hyper-plane ที่เหมาะสมที่สุด (Optimal) เพื่อแยกข้อมูลสองกลุ่มด้วย Optimal Hyper-plane นั้น $w \times x - b = 0$ จะทำหน้าที่แบ่งข้อมูลสองกลุ่มออกจากกันด้วยมี Support Vector ทำหน้าที่เป็นกั้นชนระหว่างข้อมูลที่ใกล้กัน SVM จะสร้างพื้นที่การตัดสินใจขึ้นมา หรือก็คือพื้นที่ระหว่าง $w \times x - b = 1$ และ $w \times x - b = -1$ โดยจะปรับให้ระยะห่างหรือความกว้างระหว่างทั้งสองนั้นมีค่าสูงที่สุด แต่บางกรณีข้อมูลไม่สามารถแบ่งแยกได้ด้วยเส้นตรง จำเป็นต้องแบ่งข้อมูลแบบ Non-linear ซึ่ง SVM สามารถใช้ Kernel เข้ามาช่วยในการเปลี่ยนมิติของข้อมูลเพื่อให้สามารถแบ่งแยกข้อมูลทั้งสองกลุ่มได้ด้วย Linear Hyper-plan

2.8 เว็บแอปพลิเคชัน (Web Application)

Web Application [13] ทำหน้าที่ในการเป็นช่องทางในการเชื่อมต่อระหว่างเว็บไซต์กับผู้ใช้ให้บริการไอพีโอจากที่อื่นเป็นตัวกลางที่ทำให้โปรแกรมสามารถประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น ๆ ได้ เช่น google map ที่ทาง google ให้บริการให้ยูสเซอร์สามารถนำเว็บไซต์ของตนเองเชื่อมต่อกับแผนที่ของ google ได้

2.8.1 จาวาสคริปต์ (Javascript)

เป็นภาษาคอมพิวเตอร์ที่นิยมใช้ในการพัฒนาเว็บแอปพลิเคชัน เนื่องจากจาวาสคริปต์มีความสามารถในการจัดการได้ทั้งฝั่งไคลเอนต์ (client) และฝั่งเซิร์ฟเวอร์ (server) ภาษาจาวาสคริปต์เป็นภาษาที่มีคุณสมบัติอะซิงโครนัส (asynchronous) ซึ่งแก้ไขปัญหาการขัดกันระหว่างคำสั่งที่ต้องรอในการรันคำสั่งถัดไปของภาษาที่เป็นซิงโครนัส (synchronous)

2.8.2 วีว (Vue)

Vue.js เป็นไลบรารีจาวาสคริปต์ที่มุ่งเน้นไปที่เลเยอร์ของมุมมอง (view) สำหรับพัฒนามุมมองผู้ใช้ (user interface) โดยในตัวไลบรารีสามารถรองรับแอปพลิเคชันที่ซับซ้อนเช่นระบบจัดการเส้นทาง (routing) ระบบจัดการสถานะ (state) และการสร้าง (build)

2.9 เอพีไอ (API)

Application Programming Interface (API) [29] คือส่วนต่อประสานโปรแกรมประยุกต์ เป็นวิธีการที่ระบบปฏิบัติการ, ไลบรารี และบริการอื่น ๆ เปิดให้โปรแกรมคอมพิวเตอร์สามารถติดต่อเรียกใช้งานได้ โดยเอพีไอสร้างขึ้นจากส่วนสำคัญสองส่วนคือ

1. ข้อกำหนดที่จะอธิบายการแลกเปลี่ยนข้อมูลระหว่างโปรแกรม ที่ทำออกมาในลักษณะของเอกสาร เพื่อกำหนดการร้องขอและการตอบสนองต้องเป็นอย่างไร

2. ซอฟต์แวร์ที่เขียนขึ้นมาตามข้อกำหนดดังกล่าว และทำการเผยแพร่ออกไปให้ใช้งานได้

โดยทั่วไปแล้วแอปพลิเคชันที่มีเอพีไอจะต้องถูกเขียนเป็นภาษาโปรแกรมมิ่ง และเพื่อการพัฒนาในอนาคต จึงจำเป็นต้องมีการตรวจสอบโครงสร้างของเอพีไอดังนั้น ผู้ออกแบบจึงต้องให้ความสำคัญกับการทดสอบ เพื่อตรวจสอบเงื่อนไขที่สามารถเกิดขึ้นได้จากการใช้งาน

การใช้งานเอพีไอ

ปัจจุบันเอพีไอถูกใช้งานงานในแอปพลิเคชันเพื่อสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์ บริษัทใหญ่หลายบริษัทมีการเปิดให้บริการเอพีไอ เพื่อใช้งานภายนอก เช่น twitter, google, facebook โดยใครก็ตามที่สนใจนำบริการเหล่านี้ไปประยุกต์ใช้ สามารถส่งคำร้องเพื่อรับข้อมูลที่ต้องการ หรือถึงส่งคำร้องเพื่อขอบริการได้

ไลบรารีและเฟรมเวิร์ค โดยปกติแล้วเอพีไอ จะเกี่ยวข้องกับไลบรารีซอฟต์แวร์ เอพีไอนี้จำเป็นต้องอธิบายข้อกำหนด เอพีไอเดียวสามารถมีการใช้งานได้หลากหลาย (หรือไม่มีเลย) ในรูปแบบของไลบรารีต่าง ๆ ที่ใช้อินเทอร์เฟซการเขียน โปรแกรมร่วมกัน การแยกเอพีไอออกจากการทำงาน สามารถทำให้โปรแกรมที่เขียนภาษาหนึ่งใช้ไลบรารีที่เขียนด้วยอีกภาษาหนึ่งได้ ตัวอย่างเช่น เนื่องจากภาษา scala และภาษา java คอมไพล์เป็น bytecode ที่เข้ากันได้ นักพัฒนา scala จึงสามารถใช้ประโยชน์จากภาษาโปรแกรมที่เกี่ยวข้องกับเอพีไอภาษา Java ได้เป็นต้น

ระบบปฏิบัติการ เอพีไอสามารถระบุอินเทอร์เฟซระหว่างแอปพลิเคชันและระบบปฏิบัติการได้ ตัวอย่างเช่น microsoft ได้สร้างความมุ่งมั่นอย่างยิ่งต่อเอพีไอที่เข้ากันได้กับไลบรารี windows api (win32) ดังนั้นแอปพลิเคชันรุ่นเก่าอาจทำงานบน windows เวอร์ชันใหม่โดยใช้งานดังกล่าวเฉพาะปฏิบัติการที่เรียกว่า "โหมดความเข้ากันได้" เป็นต้น

ริโมทเอพีไอ ริโมทเอพีไอถูกใช้ให้นักพัฒนาสามารถเข้าควบคุมทรัพยากรผ่านทาง โปรโตคอล เพื่อให้มีมาตรฐานการสื่อสารเดียวกัน ถึงแม้ว่าจะเป็นคนละเทคโนโลยี เช่น ฐานข้อมูลเอพีไอสามารถอนุญาตให้นักพัฒนาเข้าถึงข้อมูลในฐานข้อมูลได้หลากหลายชนิดได้ผ่านฟังก์ชันเดียวกัน เพราะฉะนั้นริโมทเอพีไอจึงถูกใช้บ่อยในงานรักษาด้วยทำงานที่ฝั่งไคลเอนต์ให้ไปดึงข้อมูลจากเซิร์ฟเวอร์กลับมาทำงาน

เว็บเอพีไอ เว็บเอพีไอถูกใช้กันอย่างแพร่หลายในปัจจุบัน เนื่องจากเป็นเอพีไอที่อยู่ในกลุ่มของ HTTP และขยายออกไปสู่รูปแบบต่าง ๆ เช่น XML และ JSON ซึ่งโดยรวมแล้วจะอยู่บนเว็บเซอร์วิส เช่น SOUP หรือ REST เป็นต้น

ตัวอย่างเอพีไอที่นิยมในปัจจุบัน

1. **Google Maps API** เปิดให้ใช้งานเพื่อนำเอาแผนที่ของ Google มาลงใน webpage โดยอาศัย JavaScript หรือ Flash

2. **Youtube API** Google ยอมให้ developer สามารถนำเอา Clip video บน YouTube ไปลงใน website หรือ application ได้
3. **Flickr API** เพื่อให้ developer สามารถเข้าถึง คลังรูปภาพใน community
4. **Twitter API** มี REST API ให้ค้นหา แล้วตรวจสอบข้อมูล trends ได้
5. **Amazon product advertising API** เปิด API ให้ใช้ค้นหาสินค้า และ การโฆษณาผ่านทาง website

2.9.1 Flask

เฟลค คือเว็บเฟรมเวิร์กเป็นเฟรมเวิร์กที่เขียนขึ้นมาสำหรับใช้งานในภาษาPython ไพทอน (ไพธอน) เพื่อใช้ในการสร้างเว็บไซต์ ทำให้ภาษาไพธอนนั้น มีความสามารถในการจัดการกับเว็บไซต์ซึ่งทำให้มีความสามารถคล้ายๆภาษา PHP (พีเอชพี) ซึ่งแทบจะใช้แทนกันได้เลย ในปัจจุบันมีผู้ใช้ Flask Framework ค่อนข้างจะเยอะมากซึ่งเป็นผลมาจากการใช้งานที่ง่ายและผนวกกับมีผู้ใช้ภาษาไพธอนเพิ่มขึ้นนั่นเอง

Python ภาษาโปรแกรม Python [30] คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น

บทที่ 3

วิธีการทดลอง

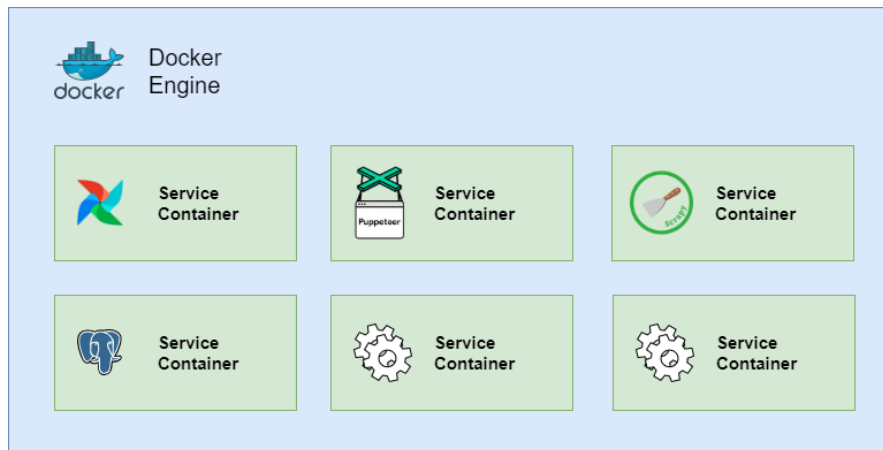
ในบทนี้จะกล่าวถึงขั้นตอนและกรอบการทำงานในการพัฒนาระบบแนะนำตำแหน่งงาน การจัดการข้อมูล และการพัฒนาเว็บแอปพลิเคชัน โดยมีจุดประสงค์เพื่อให้ระบบแนะนำตำแหน่งงานสามารถทำงานได้อย่างมีประสิทธิภาพสามารถใช้งานได้จริง รวมถึงอภิปรายภาพรวมระบบทั้งหมด

3.1 สถาปัตยกรรมสำหรับไปป์ไลน์ข้อมูล

สถาปัตยกรรมที่ผู้เขียนเลือกใช้นั้นเป็นเทคโนโลยี โอเพน ซอร์สทั้งหมดเพื่อให้ทุกขั้นตอนของท่อดึงข้อมูลสามารถทำงานจริงได้ในระยะยาวโดยคำนึงถึงต้นทุนและประสิทธิภาพที่ตามมา

3.1.1 โครงสร้างพื้นฐานของระบบ

บนเครื่องเซิร์ฟเวอร์นั้นทางผู้เขียนได้เลือกเทคโนโลยี docker เข้ามาใช้ในการจำลองเครื่องเสมือน โดยแบ่งบริการเป็นคอนเทนเนอร์ต่าง ๆ เพื่อความง่ายในการควบคุมและจัดการตัวบริการนั้น ๆ อีกทั้งสามารถสเกลได้เมื่อบริการนั้นมีการใช้งานในปริมาณที่มากในอนาคตและง่ายต่อการติดตั้งเมื่อมีการย้ายเซิร์ฟเวอร์ โดยบริการที่ทำงานอยู่บน docker เช่น ฐานข้อมูล ระบบจัดการตารางงาน ระบบสกัดข้อมูล โมดูลทำความสะอาดและแปลงข้อมูล โมดูลจำแนกประเภทกลุ่มของข้อมูล ระบบแนะนำ บริการเว็บเซอร์วิส เป็นต้น



รูปที่ 3.1 รูปภาพกรอบการทำงาน บริการที่ทำงานอยู่บน docker engine

3.1.2 แพลตฟอร์มจัดการเวิร์กโฟลว์

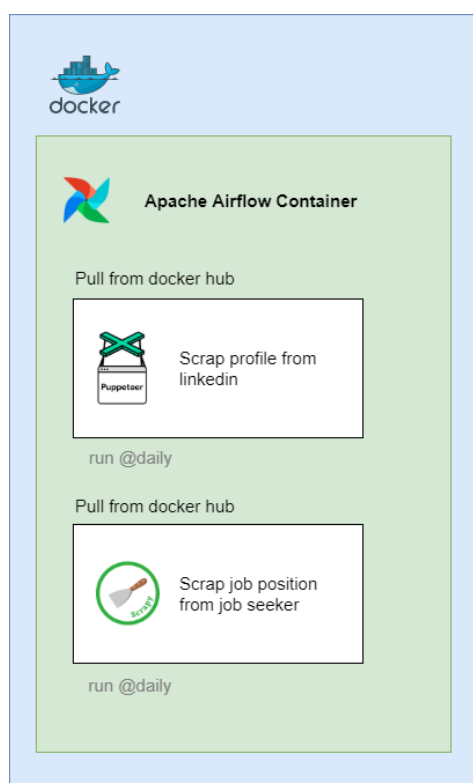
Apache airflow เป็นแพลตฟอร์มการจัดการ เวิร์กโฟลว์สามารถกำหนดเวลาหรือขั้นตอน การทำงานได้ด้วยการเขียนโปรแกรมผ่าน python โดยแพลตฟอร์มนี้ถูกติดตั้งเป็นบริการอยู่บน docker

engine เพื่อทำการควบคุมตารางการทำงานและไฟล์ของคอนเทนเนอร์อื่น ๆ โดยมีลำดับการทำงานดังนี้

3.1.3 การรวบรวมข้อมูล

การรวบรวมข้อมูลมีการรวบรวมจากสองแหล่งคือเว็บไซต์ linkedin และเว็บไซต์ indeed โดยทั้งสองเว็บไซต์นี้มีขั้นตอนการสกัดและรวบรวมไม่เหมือนกัน โดยเว็บไซต์ linkedin มีความซับซ้อนและความยากในการสกัดข้อมูลมากเนื่องจากเป็นเว็บไซต์ระดับโลกที่มีการป้องกันบอทและการเข้าถึงต่าง ๆ ที่ไม่ใช่มนุษย์อีกทั้งหน้าเว็บยังมีการทำงานเป็นแบบ client-side rendering ซึ่งจำเป็นต้องใช้การจำลองเสมือนมนุษย์มาทำหน้าที่เป็นบอทผ่านไลบรารี puppeteer ส่วนเว็บไซต์ indeed เนื่องจากมีการทำงานเป็น server-side rendering จึงสามารถดึงข้อมูลได้ตรงจากการสร้างคำร้องไปที่เซิร์ฟเวอร์ และเซิร์ฟเวอร์จะตอบกลับมาเป็นหน้าเว็บเพจที่เป็น static

ทั้งนี้การสกัดข้อมูลจำเป็นต้องมีคำสั่งหรือเป้าหมายที่เจาะจงในการสกัดข้อมูล ทางผู้จัดทำได้



รูปที่ 3.2 รูปภาพรอบการทำงาน การรวบรวมข้อมูลภายใต้ Airflow

รวบรวมตำแหน่งงานทางเทคโนโลยีสารสนเทศโดยอิงจาก CompTIA certification roadmap โดยแบ่งสายงานทางเทคโนโลยีสารสนเทศเป็นทั้งหมด 8 สายงานและ 62 ตำแหน่งดังนี้

1. service and infrastructure

helpdesk, system admin, virtualization engineer, system engineer, system architect

2. network technology

network technician, network analyst, telecommunication, network security, network admin, network engineer

3. it business and strategy

it operation, business architect, business analyst, policy advisor, policy consultant, enterprise architect

4. it management

it manager, it deputy director, it director, project manager, program manager, cto, cio

5. information security

security trainee, security technician, security analyst, security manager, security engineer, security architect, it auditor, risk compliance, incident, forensics, malware developer

6. devops and cloud technology

sysops engineer, devops admin, devops engineer, reliability engineer, devops consultant, cloud engineer, cloud architect

7. storage and data

data center, data analyst, database admin, business intelligence, data warehouse, data scientist, database architect, data engineer, database engineer

8. software development

software developer, software tester, software support, applications developer, qa, web developer, applications security, web manager, software engineer, software architect, software system architect



รูปที่ 3.3 it roadmap

3.1.3.1 Linkedin

การสกัดข้อมูลโปรไฟล์ผู้ใช้งานถึงคือนจะใช้ไลบรารี puppeteer มาใช้ในการจำลองการกระทำของมนุษย์โดยมีกระบวนการหลักทั้งหมดสามขั้นตอนคือ

1. การกำหนดรายการคำสั่งสำหรับการสกัดข้อมูล

```

{
  "helpdesk": {
    "exp": "early",
    "group": "service and infrastructure",
    "limit": 100,
    "prev": 100
  },
  "system admin": {
    "exp": "early",
    "group": "service and infrastructure",
    "limit": 30,
    "prev": 30
  },
  "virtualization admin": {
    "exp": "mid",
    "group": "service and infrastructure",
    "limit": 50,
    "prev": 50
  }
}

```

รูปที่ 3.4 ไฟล์รายการคำสั่งสกัดข้อมูลบางส่วน

2. สกัดข้อมูลโปรไฟล์ผู้ใช้งานจากการค้นหาผ่านคำสำคัญที่กำหนดในรายการคำสั่ง
3. สกัดข้อมูลโปรไฟล์ผู้ใช้งานจากการเข้าสู่หน้าหลักโปรไฟล์ผ่านยูอาร์แอลที่สกัดมาจากขั้นตอนก่อนหน้า

Algorithm 1: Scrap profile algorithms

```

1  Function Main(order):
2      Login(env.username, env.password)
3      GetUrl()
4      GetData()
5  Function Login(username, password):
6      if exist cookies then
7          login with cookie
8      else
9          login with username, password form
10     save cookie
11 Function GetUrl():
12     read order
13     read backlist
14     if not exist backlist then
15         generate backlist file from order
16     while order do
17         profiles = []
18         while order.keyword do
19             search keyword
20             scroll all page
21             urls = querySelectorAll(all profile).href
22             urls = backlist filter(urls)
23             urls = realurl filter(urls)
24             profiles.push(url)
25         save profiles to file
26         update order file
27         update backlist file
28 Function getData():
29     read url files while files do
30         data = []
31         while files.url do
32             goto url
33             validate page is exist
34             scroll all page
35             scrap name
36             scrap about
37             scrap experience
38             scrap skill
39             scrap interest
40             data.push(name, about, experience, skill, interest)
41         save data to file

```

3.1.3.2 Indeed

การสกัดข้อมูลตำแหน่งงานจากเว็บไซต์อินดีดจะใช้เฟรมเวิร์ค scrapy มาใช้ในการรวบรวมข้อมูลจากกรีเครสที่ส่งไปเพื่อค้นหาตำแหน่งงานจากคำสำคัญที่กำหนดไว้โดยจะมีขั้นตอนการทำงานหลักสามขั้นตอนคือ

1. การกำหนดรายการคำสั่งสำหรับการสกัดข้อมูล

```
{
  "helpdesk": {
    "exp": "early",
    "group": "service and infrastructure",
    "limit": 100,
    "prev": 100
  },
  "system admin": {
    "exp": "early",
    "group": "service and infrastructure",
    "limit": 30,
    "prev": 30
  },
  "virtualization admin": {
    "exp": "mid",
    "group": "service and infrastructure",
    "limit": 50,
    "prev": 50
  }
}
```

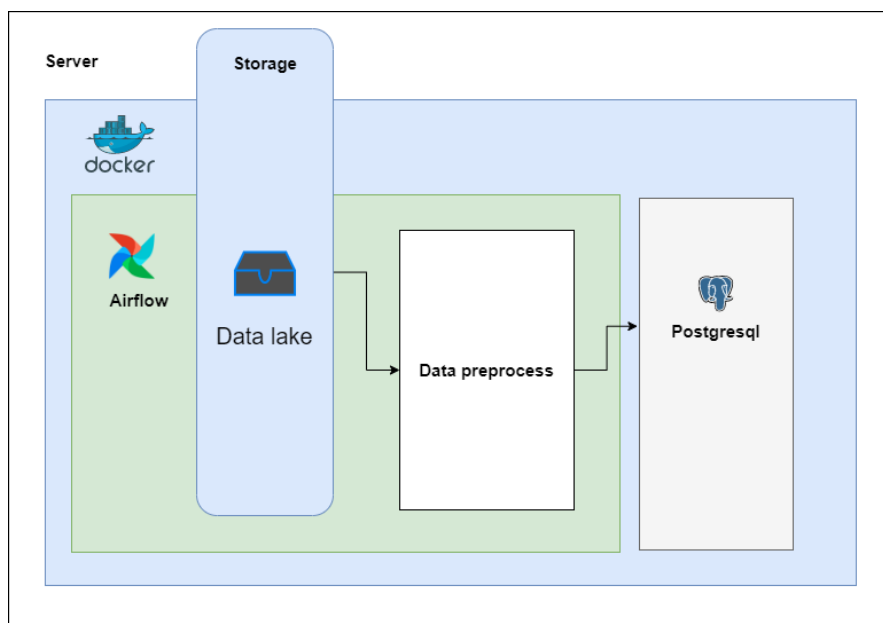
รูปที่ 3.5 ไฟล์รายการคำสั่งสกัดข้อมูลบางส่วน

2. สกัดยูอาร์แอลตำแหน่งงานจากการค้นหาผ่านคำสำคัญที่กำหนดในรายการคำสั่ง
3. สกัดข้อมูลหน้าตำแหน่งงานจากการเข้าสู่หน้าตำแหน่งงานจากยูอาร์แอลที่สกัดมาจากขั้นตอนก่อนหน้า

3.1.4 การจัดการข้อมูล

หลังจากทำการรวบรวมข้อมูลจากทั้งสองแหล่ง (linkedin, indeed) และบันทึกเป็นรูปแบบไฟล์อยู่ในทะเลสาบข้อมูลแล้ว airflow จะทำการรันและเริ่มกระบวนการทำความสะอาดข้อมูล(Data cleaning) โดยเป็นกระบวนการตรวจสอบแก้ไขหรือลบรายการข้อมูลที่ไม่ถูกต้องหรือไม่สอดคล้องออกจากชุดข้อมูลโดยมีขั้นตอนดังนี้

1. ตรวจสอบฟอร์แมตและความถูกต้องของไฟล์ข้อมูล
2. ลดรูปข้อมูลที่ซ้ำกัน ถ้าข้อมูลระบุกลุ่มที่แตกต่างกันให้รวมกลุ่มนั้นเป็นหลายรายการในข้อมูลเดียว
3. ลบข้อมูลที่ไม่มีฟิลด์สำคัญคือฟิลด์ about และ skill
4. ข้อมูลที่ผ่านขั้นตอนทั้งหมดจะถูกบันทึกลงฐานข้อมูลโดยมีรูปแบบที่ชัดเจนและพร้อมใช้งาน



รูปที่ 3.6 รูปภาพโฟลว์การทำงานของจัดการข้อมูล

3.1.5 การเทรนโมเดลแบ่งกลุ่มสายงาน

ในการเทรน โมเดลมีขั้นตอนหลายอย่างที่ต้องคำนึงถึง เพื่อให้โมเดลมีความแม่นยำในการแบ่งสายงาน จึงจำเป็นต้องนำเทคนิคต่าง ๆ มาประยุกต์ใช้ทั้งกับในด้านข้อมูลและด้านโมเดล โดยเทคนิคหลักที่นำมาใช้มีดังนี้

การทำความสะอาดข้อมูล

ก่อนขั้นตอนการเทรน โมเดล ข้อมูลจำเป็นต้องอยู่ในรูปแบบที่เหมาะสมต่อการใช้ในการเทรน โมเดลมากที่สุดโดยขั้นตอนการ "clean data" จะประกอบไปด้วยขั้นตอนง่ายๆ ทั่วไปอย่างเช่น

1. การเปลี่ยนข้อมูลให้เป็นตัวเลข อักษรพิเศษทั้งหมด ลบตัวอักษร โคด ลบแท็ก ลบช่องว่างที่เกินกำหนด
2. การใช้ "stopword" คัดกรองคำที่ไม่จำเป็นออกจากข้อมูล
3. ใช้เทคนิค "Lemmatizer" หรือการลดรูปคำให้เป็นคำรากศัพท์เช่น "am", "are", "is" จะถูกเปลี่ยนเป็น "be"
4. ใช้เทคนิค "Tokenize" หรือเทคนิคการแบ่งคำมาแบ่งข้อมูลให้อยู่ในรูปแบบโทเค็นหรือในรูปแบบคำต่อคำ
5. พิจารณารูปแบบของคำที่สะกดผิดเช่น "cool", "kewl", "coool"

การสร้างโครงสร้างของคำ

เมื่อทำความสะอาดข้อมูลแล้วจะสังเกตว่าข้อมูลยังมีการกระจายที่ยังไม่แน่นอนและไม่สามารถแยกแยะด้วยตาเปล่าได้ เทคนิคต่อไปนี้เป็นทำให้ข้อมูลมีความชัดเจนมากยิ่งขึ้น โดยมีขั้นตอนดังนี้



รูปที่ 3.7 PCA vector ของตำแหน่งงานเมื่อทำความสะอาดข้อมูลแล้ว

TF-IDF เพื่อช่วยให้โมเดลสามารถโฟกัสความหมายของคำได้จึงได้นำเทคนิค "TF-IDF (Term Frequency, Inverse Document Frequency)" เข้ามาใช้ในการให้น้ำหนักคำตามความหายากในชุดข้อมูล และลดคำที่เกิดขึ้นบ่อยเกินไปแล้วไปเพิ่ม noise ให้กับข้อมูลโดยรวม

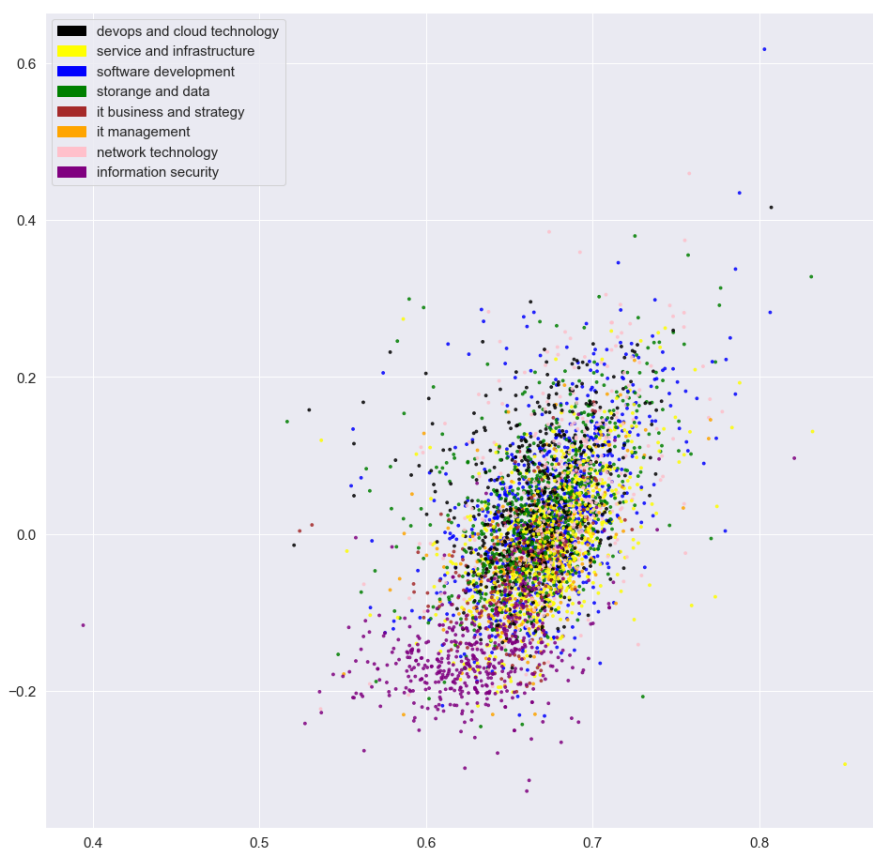
Word2Vec ถึงจะสามารถจัดการกับความถี่ของคำได้แล้วแต่อย่างไรก็ตามมีความเป็นไปได้มากกว่าหากเราปรับใช้โมเดลนี้เราจะพบคำศัพท์ที่เราไม่เคยเห็นในชุดข้อมูลของเรามาก่อน รุ่นก่อนหน้านี้อาจไม่สามารถจำแนกคำเหล่านี้ได้อย่างถูกต้องแม้ว่าคำจะเป็นคำที่คล้ายกันมากในการเทรนก็ตาม



รูปที่ 3.8 PCA vector ของตำแหน่งงานเมื่อทำการให้น้ำหนักแก่คำแล้ว

เพื่อแก้ปัญหา การจับความหมายของคำ "semantic meaning of word" โดยเครื่องมือที่ใช้เพื่อจับความหมายเรียกว่า "Word2Vec"

pre-trained words word2vec เป็นเทคนิคในการการฝังคำอย่างต่อเนื่อง(continuous embeddings) โดยเรียนรู้จากการอ่านข้อความจำนวนมากและจดจำคำที่มีแนวโน้มที่ปรากฏในบริบทที่คล้ายคลึงกัน หลังจากที่เทรนมาพอแล้วจะสร้างเวกเตอร์ 300 มิติ สำหรับแต่ละคำในคำศัพท์ โดยคำที่มีความหมายใกล้เคียงกันจะมีระยะใกล้เคียงกัน โดยผู้จัดทำได้นำ pre-trained ของ "GoogleNews" ที่ประกอบไปด้วยเวกเตอร์ของคำมากกว่า 3 ล้าน หลังจากการทำ word embeddings ด้วย word2vec แล้วจะสังเกตว่าการกระจายตัวมีความแน่นหนาขึ้นแต่การแบ่งแยกสายงานไม่แตกต่างจากการทำ TFIDF มากนัก



รูปที่ 3.9 PCA vector ของตำแหน่งงานผ่านเทคนิค word2vec

หลังจากที่ข้อมูลอยู่ในรูปแบบที่พร้อมใช้งานแล้วจึงนำข้อมูลมาทำการสร้าง โมเดลแบ่งกลุ่มตำแหน่งงาน โดยกลุ่มจะถูกแบ่งออกเป็นทั้งหมด 8 กลุ่มใหญ่และ 62 ตำแหน่งงาน [26, CompTIA] โดยมีลักษณะดังนี้

	name	group_id	subgroup	exp	group_name
0	helpdesk intern	1	3	early	service and infrastructure
1	desktop support intern	1	5	early	service and infrastructure
2	polymer lab tech	1	1	early	service and infrastructure
3	hardware technician	1	2	early	service and infrastructure
4	helpdesk technician	1	3	early	service and infrastructure
...
133	principal software engineer	8	2	late	software development
134	software architect	8	3	late	software development
135	senior software architect	8	3	late	software development
136	qa director	8	4	late	software development
137	software system architect	8	2	extended	software development

รูปที่ 3.10 กลุ่มงานทางไอที

การเทรนโมเดลนั้นจะถึงขั้นการ โดย airflow ให้ทำการเทรนทุก ๆ หนึ่งวันเพื่อเป็นการอัปเดตตัวโมเดลให้มีความแม่นยำและถูกต้องมากที่สุดและโมเดลที่ทำการเทรนจะถูกเก็บไว้ที่ storage ของ

เซิร์ฟเวอร์หรือถูกเรียกใช้โดยระบบแนะนำต่อไปโดย ทั้งนี้ตัวโมเดลจะใช้เทคนิค "support vector machine" เข้ามาใช้ในการเทรน โมเดล โดยตัวอย่างไปป์ไลน์ที่ใช้ในการเทรนโมเดลจะมีลักษณะดังนี้

```
svm_job = Pipeline([('vect', CountVectorizer(max_df=0.75, ngram_range=(1, 2))),
                    ('tfidf', TfidfTransformer(use_idf=True)),
                    ('clf-svm', CalibratedClassifierCV(SGDClassifier(random_state=42, loss='hinge')))])
```

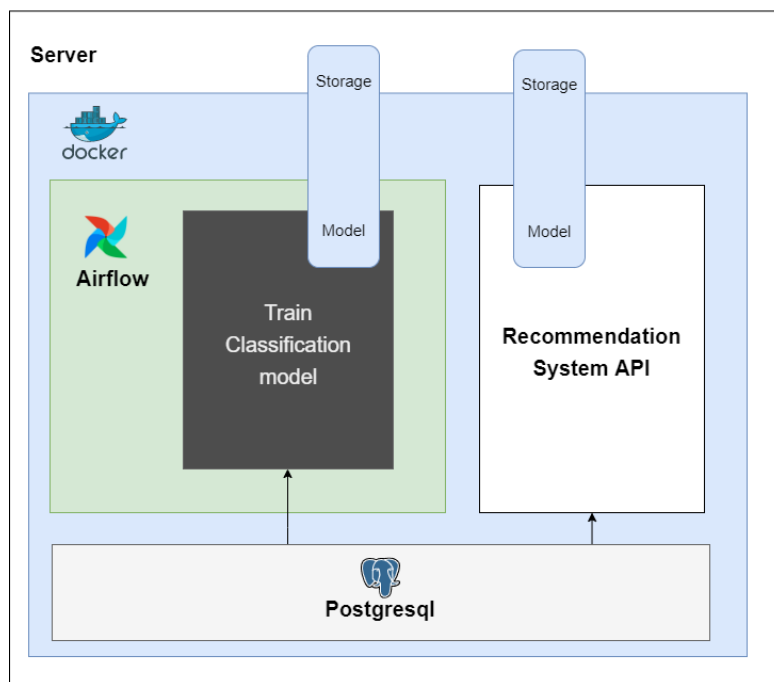
รูปที่ 3.11 ตัวอย่างโค้ดไปป์ไลน์เทรนโมเดล

3.1.6 ระบบแนะนำ

ระบบแนะนำ จะใช้เทคนิคการกรองโดยอิงจากเนื้อหา โดยเบสข้อมูลที่แตกต่างกันคือ ข้อมูลยูสเซอร์เบส และข้อมูลจ๊อบเบส โดยการเลือกกลุ่มงานจะอ้างอิงจากระยะทางของยูสเซอร์และจ๊อบด้วยระยะทางโคไซน์ (cosine distance) โดยการนำโมเดล SVM ที่เทรนจากข้อมูลที่แตกต่างกันมาเข้ามามีใช้ในการทำนาย

Algorithm 2: Job recommendation algorithm

```
1 Function Main(payload): Response(Record<string, any>[]):
2     read jobs from db
3     read profiles from db
4     load job-based pre-train model
5     load profile-based pre-train model
6     load accuracy weight by model
7     return recommendation([job model, profile model], payload, jobs, acc, 20)
8 Function recommendation(models, payload, jobs, acc, n): Record<string, any>[:]:
9     payload = clean payload text
10    groups = predict 'payload' in each 'models'
11    jobs = filter all jobs by groups(labels)
12    acc = calculate accuracy in each group by weight 100%
13    jobs_sim = calculate similarity between all job and payload then drop duplicate
14    result = sort similarity job jobs_sim
15    return result[:n]
```



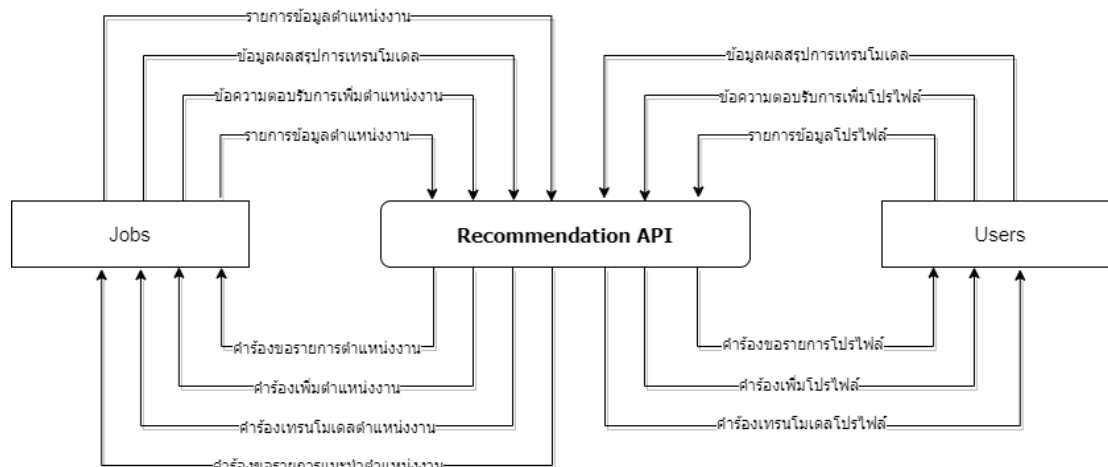
รูปที่ 3.12 รูปภาพโฟลว์การทำงานของระบบแนะนำ

3.2 การพัฒนา API เพื่อให้บริการระบบแนะนำ

ในการพัฒนา API เพื่อให้บริการระบบแนะนำนั้นผู้จัดทำได้เลือกเฟรมเวิร์คที่มีความคุ้นชินและเหมาะสมในการพัฒนามากที่สุดโดยเฟรมเวิร์คที่เลือกนั้นคือเฟรมเวิร์คของภาษา "python" ที่ชื่อว่า "flask" มาพัฒนาเป็น API โดยมีขั้นตอนดังนี้

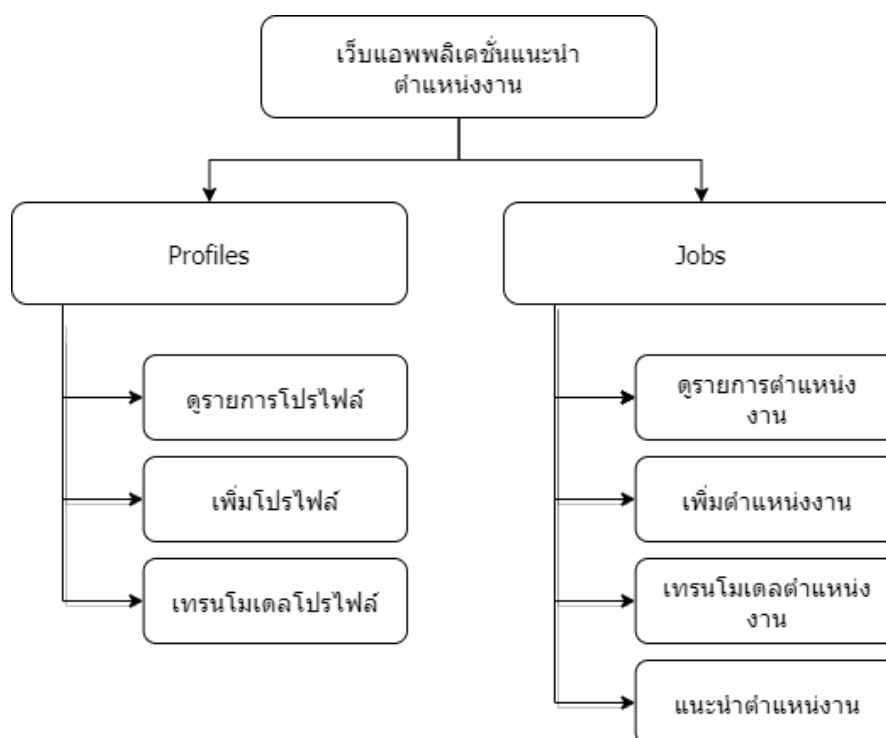
1. ออกแบบผังระบบ (Context Diagram)
2. ผังการแยกฟังก์ชันงานย่อย (Decomposition Diagram)
3. ออกแบบฐานข้อมูลเชิงคุณภาพ (Physical Design)

3.2.1 ออกแบบผังระบบ (Context Diagram)



รูปที่ 3.13 context diagram

3.2.2 ผังการแยกฟังก์ชันงานย่อย (Decomposition Diagram)



รูปที่ 3.14 decomposiiton diagram

3.2.3 ออกแบบฐานข้อมูลเชิงคุณภาพ (Physical Design)

Users		Jobs	
PK	<u>id</u> int NOT NULL	PK	<u>id</u> int NOT NULL
	name TEXT() NOT NULL interest TEXT() NULL url TEXT() NOT NULL group TEXT() NOT NULL job TEXT() NOT NULL about TEXT() NULL exp TEXT() NULL skill TEXT() NULL		job_type TEXT() NOT NULL job_title TEXT() NOT NULL company TEXT() NOT NULL desc TEXT() NOT NULL

รูปที่ 3.15 physical diagram

บทที่ 4

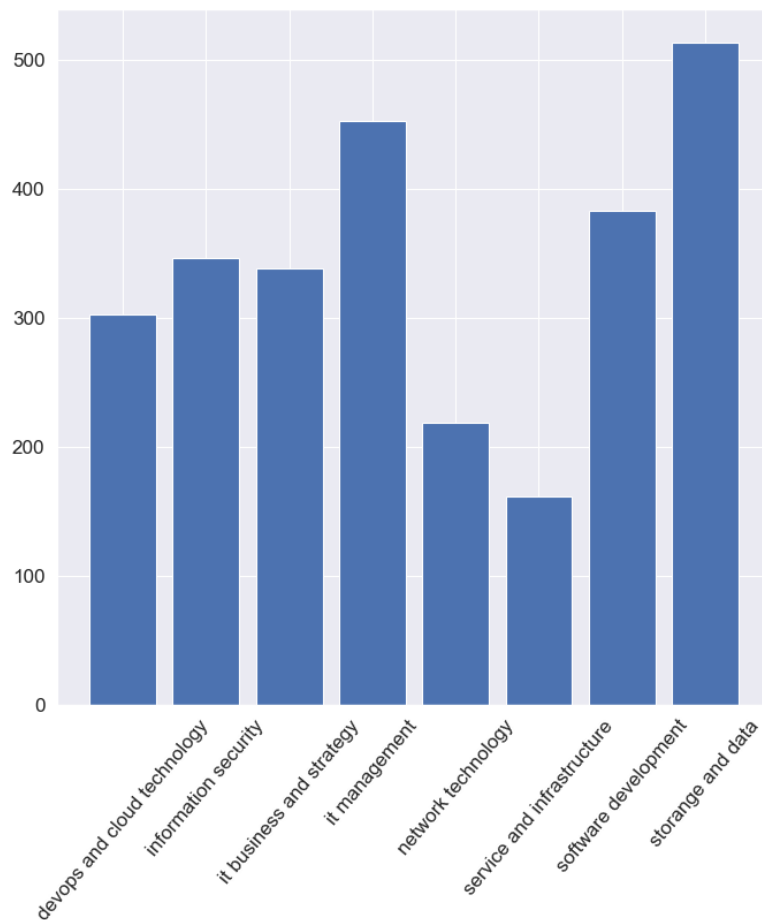
ผลการทดลอง

4.1 ข้อมูลการทดลอง

ในส่วนของการทดลองเชิงประจักษ์จะใช้ข้อมูลสองชนิดคือ ยูสเซอร์เบสเป็นข้อมูลโปรไฟล์ผู้ใช้จากเว็บไซต์ลิงก์อิน(linkedin) จำนวน 2,720 คน และตำแหน่งงานจากเว็บไซต์อินดีด(indeed) จำนวน 4,748 ตำแหน่ง โดยใช้คำสำคัญในการค้นหาข้อมูลจำนวนทั้งสิ้น 62 คำซึ่งเป็นตำแหน่งงานทางไอทีที่อิงจากสายงานทั้งหมด 8 สายงาน

4.1.1 ข้อมูลโปรไฟล์ผู้ใช้

ข้อมูลโปรไฟล์ผู้ใช้จะถูกสกัดโดยตรงจากเว็บไซต์ลิงก์อินโดยสกัดออกมาในรูปแบบเจสัน(json) และถูกนำมาแปลงเป็นรูปแบบตารางในขั้นตอนของการเตรียมการข้อมูลเพื่อบันทึกลงฐานข้อมูล จำนวนข้อมูลทั้งหมดที่สกัดมาจากเว็บไซต์ลิงก์อินคือ 2,720 คน



รูปที่ 4.1 ตารางเปรียบเทียบจำนวนโปรไฟล์ในแต่ละสายงาน

	name	interest	url	group	job	about	exp	skill
0	Visarut Tirataworawan	[Bruce Kasanoff, Shell, Diego Rodriguez, Netwo...	https://www.linkedin.com/in/visarut	software development	applications developer	Software Engineer with five years of experience...	[[('position': ['Senior Backend Developer'], 'c...	[Software Development, Computer Network Operat...
1	Amit Ughade	[Amdocs, Amadeus, Big Data and Analytics, Acce...	https://www.linkedin.com/in/amit-ughade-8304713a	software development	applications developer	Highly accomplished and experienced Java devel...	[[('position': ['Company Name Allianz Tec...	[Java, Spring Boot, Pivotal Cloud Foundry (PCF...
2	Ekbundit Wangthammang	[Spring Users, Kasetsart University, Java Deve...	https://www.linkedin.com/in/ekbunditw	software development	applications developer	Experienced Java Developer with a demonstrated...	[[('position': ['Senior Software Engineer'], 'c...	[Java, Objective-C, Android Development, Softw...
3	Chumpol J.	[Microsoft, Bank of Thailand, Hitachi, KASIKOR...	https://www.linkedin.com/in/chumpol-j-91a11b44	software development	applications developer	๓ คติประจำตัวในการทำงาน คือ OPEN & CHALL...	[[('position': ['Application Developer'], 'comp...	[IT Management, Purchase Orders, Information S...
4	Borriwat H.	NaN	https://www.linkedin.com/in/bthppong	software development	applications developer	- ASP.NET C# (since 2013) ๓ - React Nati...	[[('position': ['Mobile Application Developer']...	[C#, jQuery, ASP.NET, ASP.NET MVC, JavaScript...

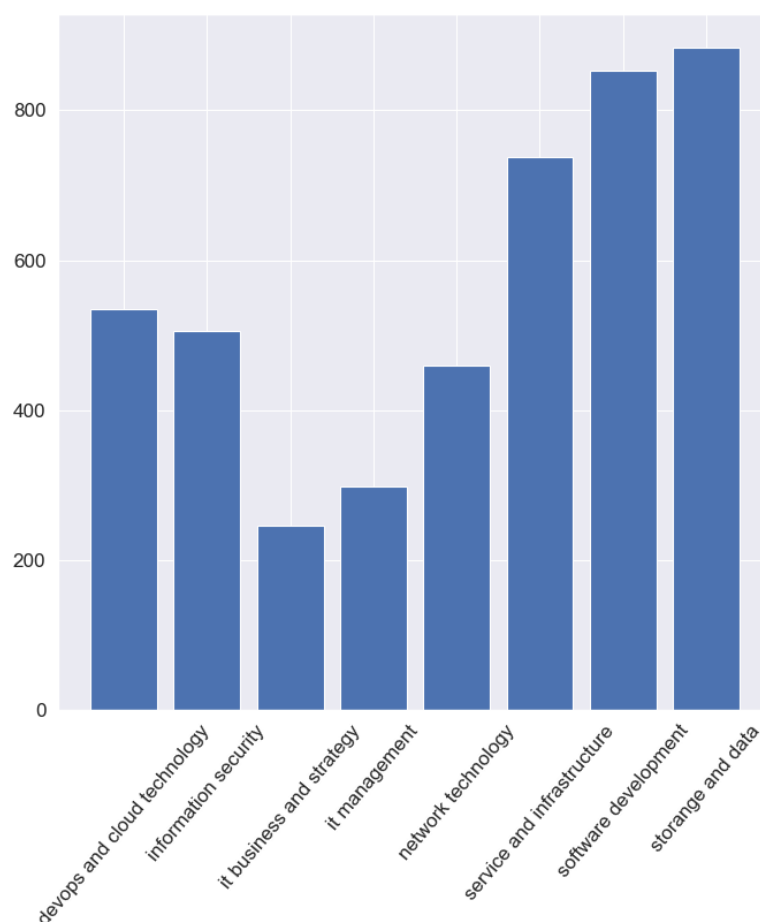
รูปที่ 4.2 ตัวอย่างข้อมูลโปรไฟล์เปลี่ยนจากรูปแบบเจ้านมาเป็นรูปแบบตาราง

```
{
  "data": {
    "name": "Wathiwut Kongjan",
    "about": "I working closely with the business intelligence team in agile methodol
    dashboard based on customer requiremets and research new tool and technology f
    "exp": [
      {
        "position": ["Business Intelligence Developer"],
        "company": ["G-ABLE Group Full-time"],
        "exp": ["1 yr 5 mos"],
        "detail": [
          "Working closely with Business Intelligence team to design and develop a re
          requirement in various industries such as government hotel and banking .
          Business Intelligence developer team. - Coordinate with vendor tr
          Developments Data Visualization and Dashboard for user find insight in da
          problem solving."
        ]
      }
    ],
    "skill": [
      "Business Intelligence (BI)",
      "Information Technology",
      "Data Visualization",
      "Data Modeling",
      "Software Project Management",
      "Government",
      "SQL",
      "Tableau",
      "Microsoft Power BI",
      "Database",
      "SQL Server Management Studio",
      "Data Transformation",
      "SQL Server Analysis Services (SSAS)",
      "Self Learning",
      "Cross-team Collaboration",
      "Hospitality"
    ],
    "interest": [
      "Stat CBS Chula Professional Network",
      "Hewlett Packard Enterprise",
    ]
  }
}
```

รูปที่ 4.3 ตัวอย่างข้อมูลโปรไฟล์ที่ถูกสกัดมาในรูปแบบเจ้าน

4.1.2 ข้อมูลตำแหน่งงาน

ข้อมูลตำแหน่งงานจะถูกสกัดมาจากเว็บไซต์อินดีด(indeed) ผ่านการส่งคำร้องไปที่เซิร์ฟเวอร์ โดยตรงทำให้ง่ายต่อการได้มาของข้อมูล โดยข้อมูลที่สกัดมานั้นจะอยู่ในรูปแบบตารางจำนวนทั้งสิ้น 4,748 ตำแหน่ง



รูปที่ 4.4 ตารางเปรียบเทียบจำนวนตำแหน่งงานในแต่ละสายงาน

	field	title	company	desc
0	service and infrastructure	HelpDesk Technician	Northeast Credit Union36 reviews-Portsmouth, N...	We are seeking a Helpdesk Technician for our g...
1	software development	Web Application Developer	LeafFilter Gutter Protection245 reviews-Hudson...	Why Work at LeafFilter?\nLeafFilter Gutter Pro...
2	service and infrastructure	IT service and infrastructure	Wellness Pointe15 reviews-Longview, TX 75601	Company Description\nAt Wellness Pointe, every...
3	software development	Mobile Developer	Bhuvi IT Solutions-United States	If you're passionate about mobile platforms an...
4	software development	PHP software development	HostMaker-Dallas, TX	JOB SUMMARY: Design and construct web pages/sl...

รูปที่ 4.5 ตัวอย่างข้อมูลตำแหน่งงานที่ถูกสกัดมาในรูปแบบตาราง

4.2 การทำนายกลุ่มสายงาน

ในการทำนายสายงานจะใช้โมเดลการเรียนรู้ของเครื่องจักรมาใช้ในการแบ่งแยกประเภทของสายงานทางไอทีซึ่งมาทั้งหมด 8 สายงานและใช้ข้อมูล โปรไฟล์ผู้ใช้เป็นยูสเซอร์เบส และข้อมูลตำแหน่งงานเป็นคอนเท้นเบส โดยใช้ทั้งสองข้อมูลนี้มาเทรนมาเทรนโมเดลแยกกัน เพื่อความหลายหลายในการแนะนำและแก้ปัญหาความคลุมเครือระหว่างสายงาน เช่น สายงาน devops and cloud technology ซึ่งทำงานใกล้ชิดกับฮาร์ดแวร์และการวางระบบต่าง ๆ ซึ่งใกล้เคียงอย่างมากกับสายงาน service and infrastructure ที่มีหน้าที่วางระบบและซัพพอร์ทเซอร์วิสต่าง ๆ

เทคนิคการสร้างโครงสร้างของคำผู้จัดทำได้เลือกเทคนิค TFIDF ซึ่งจากการเปรียบเทียบกับเทคนิคอื่นๆ เช่น word2vec หรือ word vectorizer แล้วเทคนิค TFIDF ให้ค่าความแม่นยำที่สูงที่สุด และเมื่อดูจากการกระจายตัวของคำในกราฟแล้วจะเห็นว่า TFIDF มีการกระจายตัวของกลุ่มที่เหตุซัดมากที่สุด ต่อมาคือเทคนิคที่ใช้ในการเทรนโมเดล NLP จะใช้เทคนิค "support vector machine" มาใช้ในการเทรนโมเดล เนื่องจากมีความแม่นยำสูงสุดเมื่อเทียบกับอื่นๆ เช่น "logistic regress" และมีผลลัพธ์ดังนี้

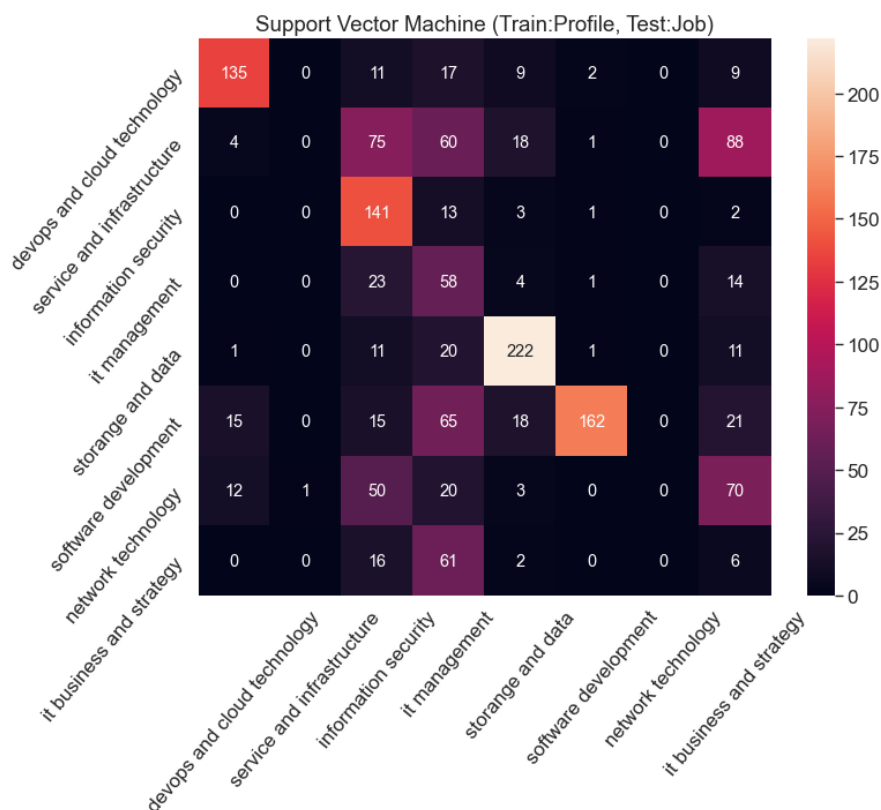
4.2.1 ยูสเซอร์เบส

ยูสเซอร์เบสเป็นการใช้ข้อมูล โปรไฟล์ผู้เข้ามาใช้เป็นฐานในการเทรนโมเดลและทำนายตำแหน่งงานจะสรุปได้ดังนี้

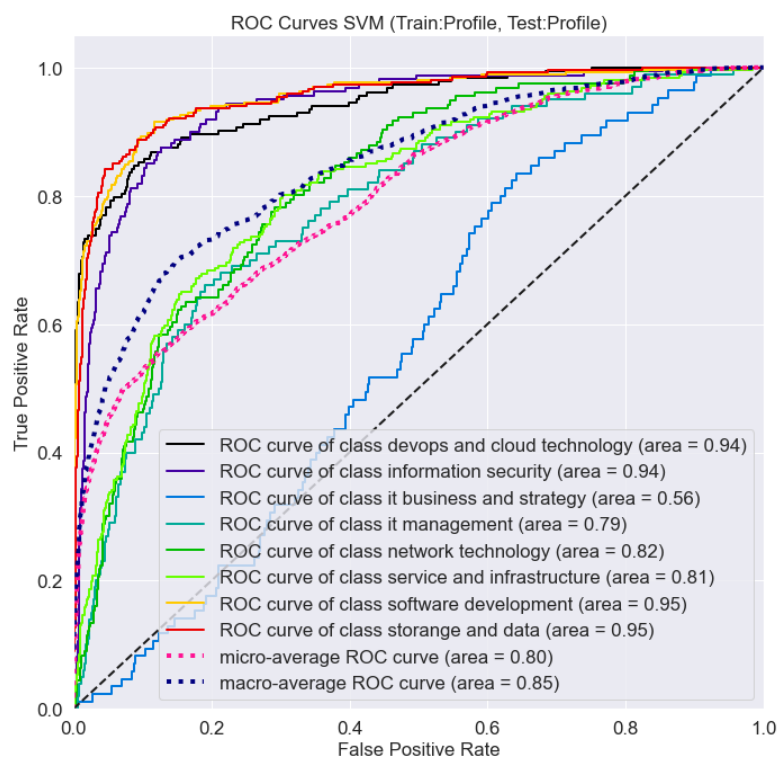
	precision	recall	f1-score	support
devops and cloud technology	0.74	0.81	0.77	167
information security	0.88	0.41	0.56	342
it business and strategy	0.07	0.03	0.04	221
it management	0.58	0.18	0.28	314
network technology	0.00	0.00	0.00	0
service and infrastructure	0.00	0.00	0.00	1
software development	0.55	0.96	0.70	168
storage and data	0.83	0.80	0.81	279
accuracy			0.49	1492
macro avg	0.46	0.40	0.40	1492
weighted avg	0.63	0.49	0.51	1492

รูปที่ 4.6 รายงานการแบ่งกลุ่มโดยใช้ข้อมูล โปรไฟล์

จากการวิเคราะห์จะคาดการณ์ได้ว่า โปรไฟล์ผู้ใช้งานอื่นนั้นมีความแม่นยำที่ค่อนข้างต่ำ โดยมีความแม่นยำอยู่ที่ 49% ซึ่งเหตุผลอาจเป็นเพราะผู้ใช้งานใส่ทักษะวิชาชีพครอบคลุมทุกสิ่งทุกอย่างที่รู้จักโดยไม่คำนึงว่าผู้ใช้นั้นมีความเชี่ยวชาญหรือไม่ และในส่วนของคำอธิบายส่วนตัว ผู้ใช้บางส่วนไม่ได้เขียนถึงความเชี่ยวชาญหรือการทำงานของตนเองแต่อาจเขียนถึงสิ่งที่ไม่เกี่ยวกับสิ่งที่ทำเลย เช่น การแนะนำตัวบอกถึงสิ่งที่ชอบสิ่งที่รักหรือกลอน เป็นต้น



รูปที่ 4.7 confusion matrix จากการทำนายโดยใช้ข้อมูลโปรไฟล์ผู้ใช้



รูปที่ 4.8 ตาราง roc จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน

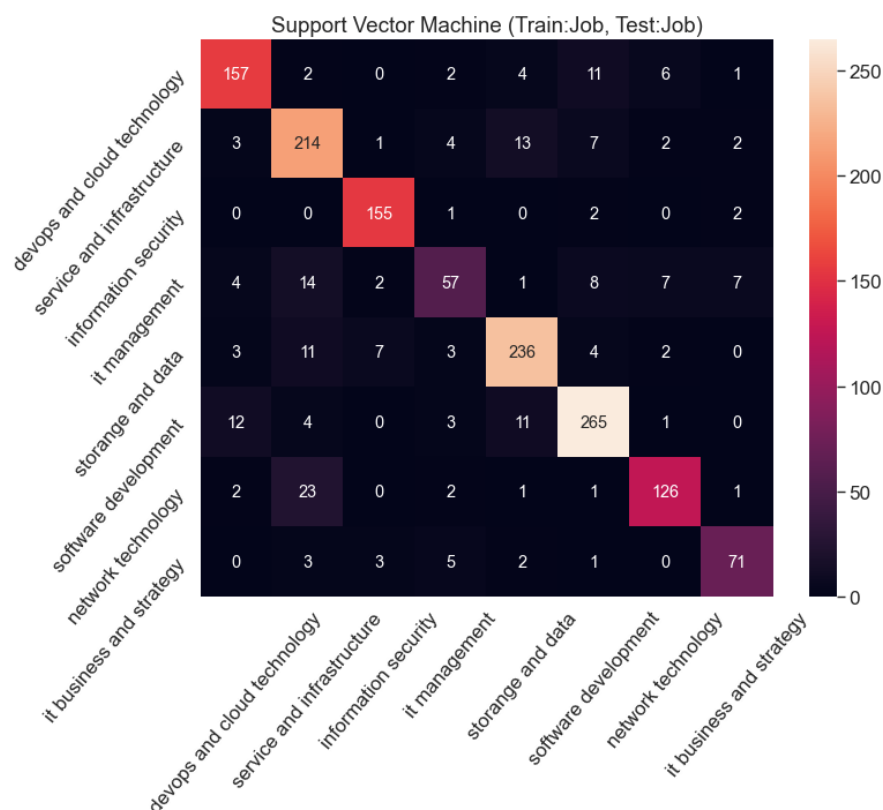
4.2.2 จีอบเบส

จีอบเบสการใช้ข้อมูลตำแหน่งงานมาใช้เป็นฐานในการเทรน โมเดลและทำนายตำแหน่งงานจะสรุปได้ดังนี้

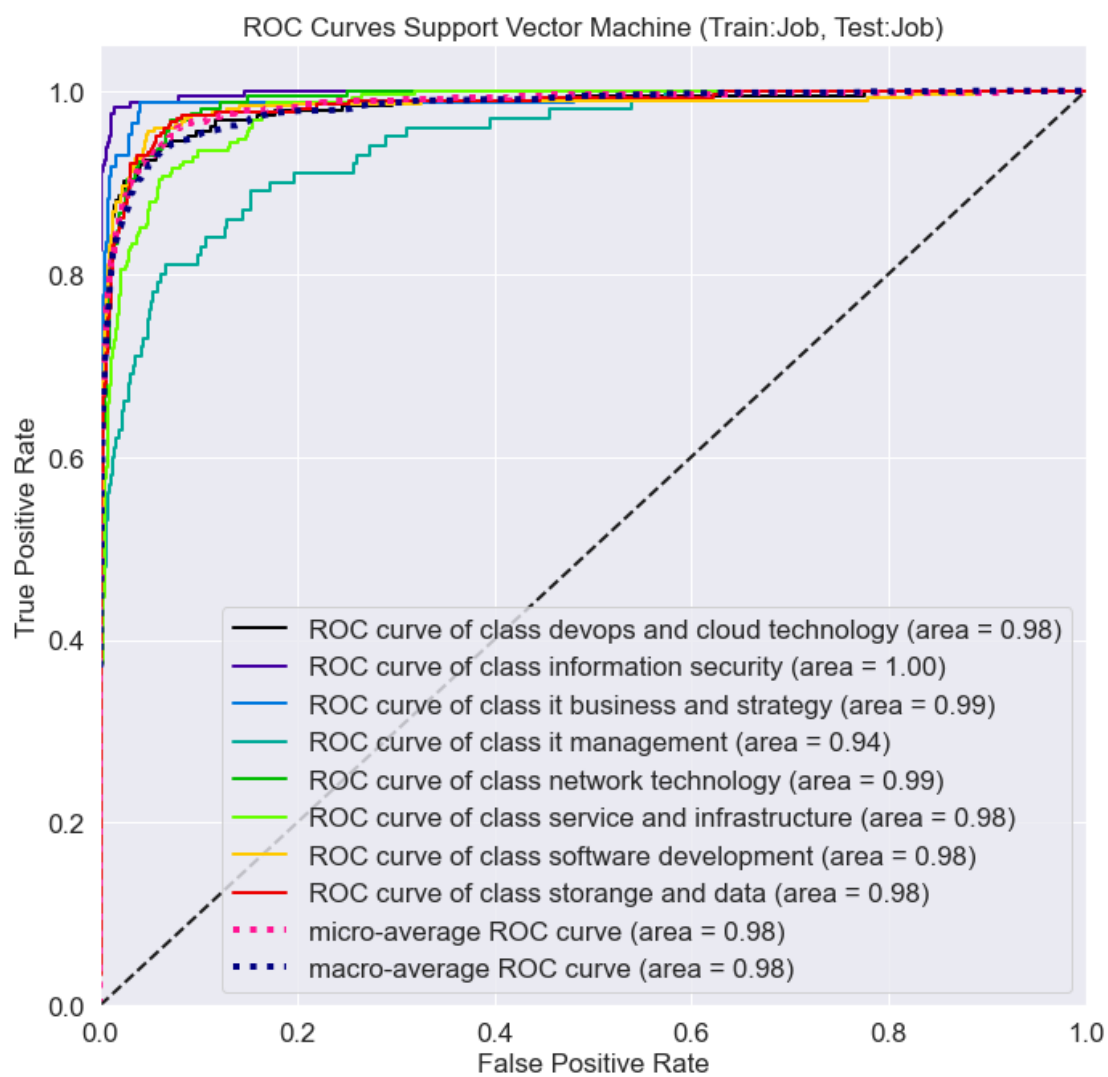
	precision	recall	f1-score	support
devops and cloud technology	0.87	0.86	0.86	183
information security	0.92	0.97	0.95	160
it business and strategy	0.85	0.84	0.84	85
it management	0.74	0.57	0.64	100
network technology	0.88	0.81	0.84	156
service and infrastructure	0.79	0.87	0.83	246
software development	0.89	0.90	0.89	296
storage and data	0.88	0.89	0.88	266
accuracy			0.86	1492
macro avg	0.85	0.84	0.84	1492
weighted avg	0.86	0.86	0.86	1492

รูปที่ 4.9 รายงานการแบ่งกลุ่มโดยใช้ข้อมูลโปรไฟล์

การใช้ข้อมูลตำแหน่งงานหรือจีอบเบสมาใช้ในการเทรน โมเดลจะเห็นว่าตัวโมเดลมีความแม่นยำเป็นอย่างมากโดย ความแม่นยำอยู่ที่ 89% จารการวิเคราะห์จะพบว่าเนื่องจากเนื้อหาของตำแหน่งงานเป็นสิ่งที่ค่อนข้างตายตัวในตัวของเนื้อหาอยู่แล้ว เช่นในสายงานของ software development ตำแหน่งงานส่วนใหญ่จะเขียนความต้องการเป็นภาษาที่สามารถเขียนได้เป็นต้น



รูปที่ 4.10 confusion matrix จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน



รูปที่ 4.11 ตาราง roc จากการทำนายโดยใช้ข้อมูลตำแหน่งงาน

4.3 ระบบแนะนำ

ในขั้นตอนแนะนำตำแหน่งงานข้อมูลนำเข้าจากผู้ใช้งานที่ต้องการตำแหน่งงานที่เหมาะสมกับโปรไฟล์ จะถูกนำมาใช้ในการหาระยะทางโคไซน์ โดยขั้นตอนนี้จำเป็นต้องใช้ทรัพยากรเครื่องจำนวนมาก ด้วยเหตุนี้ระบบแนะนำผลของการแบ่งกลุ่มสายงาน มาใช้ในการแบ่งข้อมูลตำแหน่งงานที่มีอยู่เพื่อลดปริมาณข้อมูลที่ต้องคำนวณ และเวลาที่ใช้ในการรอผลลัพธ์ โดยผลลัพธ์ของการแนะนำตำแหน่งงานจะแสดงเรียงตามความคล้ายคลึงกันระหว่างโปรไฟล์ผู้ใช้ของเรา กับตำแหน่งงานทั้งหมดที่ถูกแบ่งโดยขั้นตอนแบ่งกลุ่มสายงาน ผลลัพธ์ที่แสดงออกมาจากการตอบกลับของเอพีไอจะเป็นดังนี้

คำร้อง payload ของข้อมูลที่ส่งไปกับคำร้องคือไฟล์ผู้ใช้ที่ต้องการหางานที่เหมาะสมโดยอยู่ในรูปแบบ json ดังนี้

```
▼ Request Headers
  User-Agent: "PostmanRuntime/7.26.8"
  Accept: "*/*"
  Postman-Token: "875cea3a-ccc6-4d9a-8aa6-6449e40ed144"
  Host: "localhost:5000"
  Accept-Encoding: "gzip, deflate, br"
  Connection: "keep-alive"
  Content-Type: "multipart/form-data; boundary=-----100035934168235"
  Cookie: "Cookie_1=value"
  Content-Length: 455
▼ Request Body
  profile: "I am a fullstack development for 3 years, Specialize in frontend stack (Vue)
  now looking for an opportunities for fullstack, data science field as co-op or Interns"
```

รูปที่ 4.12 ตัวอย่างคำร้องแนะนำตำแหน่งงาน

คำตอบรับ ข้อมูลที่ตอบกลับมาจากเซิร์ฟเวอร์จะเป็นข้อมูลรายการตำแหน่งงานที่อยู่ในรูปแบบ json ดังนี้

```
[
  {
    "company": "Earth Resources Technology, Inc19 reviews-Pasadena, CA",
    "desc": "Will participate in the design, development, and delivery of solutions ,
      performant code, developing unit tests to support code coverage requirements
      deployment, and automation requests for a subset of products.\nRequired Skill
      languages). Must possess solid knowledge of AWS services.\nExtensive experie
      troubleshooting IAM Policies, Resource permissions issues during migrations ,
      CloudFront, SNS, SQS, DynamoDB, Cloudwatch, Elasticache, Docker and Applicat
      pipelines - GitHub, Maven, Jenkins\nExperience deploying and working with va
      process\nFull stack development knowledge utilizing frontend frameworks such
      Authorization.\nDesired\n5+ years development experience in python or simila
      Data Architecture.\nExperience or familiarity with newer AWS data and analyt
      and verbal communication skills\n\nEducation\nBS in Computer Engineering, In
    "job_title": "Cloud software development",
    "job_type": "devops and cloud technology",
    "sim": 0.304020285
  },
  {
    "company": "Fuse Engineering LLC-United States",
    "desc": "Description:\n\nAN ACTIVE SECURITY CLEARANCE AND POLYGRAPH ARE REQUIRED
      containers.\nFamiliarity with at least one DevOps automation tool such as Pu
      including Windows workstations and Linux.\n\nRequirements:\nQualifications\n
      Engineering or related field, can be applied for 4 years credit; Master's de
      Linux distributions and creating new installation media.\nExperience with ce
      Grafana, or Tableau.\n\n",
    "job_title": "Inspired devops and cloud technology",
    "job_type": "devops and cloud technology",
    "sim": 0.2671056971
  },
]
```

รูปที่ 4.13 ตัวอย่างคำตอบรับแนะนำตำแหน่งงาน

4.4 เว็บไซต์บริการระบบแนะนำ

แบบฟอร์ม แบบฟอร์มสำหรับกรอกข้อมูลโปรไฟล์ผู้ใช้เพื่อนำข้อมูลนี้มาใช้ในการจับคู่ตำแหน่งงานที่เหมาะสม


JOB RECOMMENDATION

I am a fullstack development for 3 years, Specialize in frontend stack (Vue), backend stack (express, go(new)) and also have experience in DevOps (CI/CD with github actions, gitlab CI, Docker, GCP, AWS) now looking for an opportunities for fullstack, data science field as co-op or Internship.

MATCHING

รูปที่ 4.14 แบบฟอร์มสำหรับกรอกข้อมูลโปรไฟล์ผู้ใช้

ตำแหน่งงาน รายการตำแหน่งงานที่เหมาะสมกับโปรไฟล์ผู้ใช้




Cloud software development

Earth Resources Technology, Inc19

Closing December 14, 2020 12:00 AM


♥
×



Inspired devops and cloud technology

Fuse Engineering LLC-United


♥
×



Infrastructure devops and cloud technology

PIÑATA SCIENCES-Boston, MA


♥
×



devops and cloud technology

Abiomed48 reviews-Danvers, MA

♥
×



Cloud software development
Earth Resources Technology, Inc19 reviews-Pasadena, CA

Will participate in the design, development, and delivery of solutions on AWS technology stack in support of the MGSS AMMOS suite of products. Will be responsible for developing efficient and highly performant code, developing unit tests to support code coverage requirements, and deliver into a continuous build, integration, and deployment environment. Will support existing application development, deployment, and automation requests for a subset of products.

Required Skills6 years of experience in a software development role. Must have strong Python, JavaScript, or Java skills (or other related languages). Must possess solid knowledge of AWS services.

Extensive experience in Application migrations to Cloud with Cloud Native Patterns and provide support for Applications running in Cloud.

Experience in troubleshooting IAM Policies, Resource permissions issues during migrations of Applications.

Experience working with AWS Services Technologies EC2, ALB/ELB, Elastic BeanStalk, ACM, RDS, S3, LAMBDA, API Gateway, CloudFront, SNS, SQS, DynamoDB, Cloudwatch, ElastiCache, Docker and Application Runtimes.

Experience in building with Automation tools such as (Jenkins, Nexus, Maven and JUnit) as well as knowledge with CI/CD pipelines - GitHub, Maven, Jenkins.

Experience deploying and working with various relational or NoSQL databases.

Knowledge & demonstrated experience in Agile methodologies and practice.

Knowledge of ETL process/full stack development knowledge utilizing front-end frameworks such as Angular or React and web frameworks such as Django, Flask or Express.

Must be eligible to obtain any required Export Authorization.

รูปที่ 4.15 รายการตำแหน่งงานที่ถูกแนะนำ

บทที่ 5

สรุปผล

ในการทดลองนี้ ได้เสนอกรอบการทำงานของระบบแนะนำตำแหน่งงาน โดยใช้เทคนิคการกรองแบบเนื้อหา โดยใช้ข้อมูลที่แตกต่างกันสองแหล่งคือ จ๊อบเบสหรือข้อมูลตำแหน่งงานและยูสเซอร์เบสหรือข้อมูลโปรไฟล์ผู้ใช้งานใช้ในการการเทรน โมเดลโดยใช้เทคนิค "Support vector machine" ซึ่งให้ผลลัพธ์ที่แม่นยำที่สุดเมื่อเทียบกับเทคนิคอื่น ๆ ผลลัพธ์ของโมเดลการแบ่งกลุ่มสายจะได้ความแม่นยำอยู่ที่ 86% เมื่อใช้ข้อมูลจากตำแหน่งงาน และ 49% เมื่อใช้ข้อมูลยูสเซอร์ โดยความแม่นยำทั้งสองนี้จะถูกถ่วงน้ำหนักและสเกลด้วย 100 เพื่อใช้ในการแบ่งข้อมูลในขั้นตอนการแนะนำงาน โดยผลลัพธ์ของตำแหน่งงานที่แนะนำมามีความแม่นยำอยู่ในระดับที่น่าพึงพอใจเป็นอย่างมาก ถึงแม้ว่าชื่อและประเภทงานอาจมีความคลาดเคลื่อน แต่เนื้อหาของงานค่อนข้างตรงกับการจับคู่กับโปรไฟล์ผู้ใช้งาน

5.1 ปัญหาที่เกิดขึ้น

ปัญหาที่เกิดขึ้นจากการทดลอง ผู้จัดทำได้พบว่าข้อมูลที่สกัดมาจากเว็บไซต์ลิงค์อื่นนั้น โปรไฟล์ผู้ใช้งานมักเขียนคำอธิบายตนเองค่อนข้างไม่เกี่ยวข้องกับลักษณะงานที่ทำ อีกทั้งเว็บไซต์ลิงค์อื่นมีความยากในการสกัดข้อมูลเป็นอย่างมาก ทำให้ข้อมูลที่ได้นั้นมีจำนวนยังไม่เพียงพอต่อการใช้งานในความเห็นของผู้จัดทำ โดยปริมาณโปรไฟล์ที่สกัดมานั้นมีจำนวน 2,720 คน จากที่คาดหวังไว้ 6,000+

ปัญหาต่อมาที่พบคือเนื่องจากตำแหน่งงานในแต่ละรายการ นั้นมีความไม่เหมือนใคร ในด้านเนื้อหาของงานถึงแม้ว่าหัวข้อจะเหมือนกันก็ตาม รวมถึงตำแหน่งงานมีเวลาหมดอายุหรือปิดรับสมัคร ทำให้การแนะนำรายการเดิมนั้นเป็นไปไม่ได้ ทำให้การแนะนำด้วยเทคนิคการกรองแบบร่วมกัน ไม่สามารถใช้ได้

5.2 ทิศทางในอนาคต

ทิศทางในอนาคตผู้จัดทำมุ่งเน้นไปที่เรื่องของข้อมูลที่ได้นั้นมากกว่าตัวโมเดล โดยระบบสกัดข้อมูลที่ใช้อยู่ตอนนี้ยังไม่มีประสิทธิภาพและต้องมีการปรับปรุงอีกมากในการสกัดข้อมูลจากทั้งสองแหล่ง ผู้จัดทำจึงมีแผนในการพัฒนาโครงสร้างท่อข้อมูลให้เป็นระบบที่เป็นระบบอัตโนมัติ เพื่อให้ได้ข้อมูลที่ใหม่อยู่เสมอและความแม่นยำที่แม่นยำขึ้นในการแบ่งกลุ่มสายงาน

บรรณานุกรม

- [1] Baptiste Rocca: Introduction to recommender systems
<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>
- [2] Collaborative filtering ฟังก์ชันการแนะนำเพลงของ Spotify
<https://tupleblog.github.io/spotify/>
- [3] Robin Burke : Hybrid Web Recommender Systems *University of Colorado Boulder*
- [4] Nikita Sharma: Recommender Systems with Python — Part I: Content-Based Filtering
<https://heartbeat.fritz.ai/recommender-systems-with-python-part-i-content-based-filtering-5df4940bd831>
- [5] Prince Grover: Various Implementations of Collaborative Filtering
<https://towardsdatascience.com/various-implementations-of-collaborative-filtering100385c6dfe0>
- [6] Farshad Bakhshandegan Moghaddam : Cold Start Solutions For Recommendation Systems *Institute for Automation and Applied Informatics*
- [7] Diego Lopez Yse: Your Guide to Natural Language Processing (NLP)
<https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>
- [8] Lukkidd: Word Embedding and Word2Vec
<https://lukkidd.com>
- [9] Cory Maklin: TF IDF | TFIDF
<https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76>
- [10] Selva Prabhakaran: Cosine Similarity – Understanding the math and how it works
<https://www.machinelearningplus.com/nlp/cosine-similarity/>
- [11] Cory Maklin: Support Vector Machine Python Example
<https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8>
- [12] Choochart Haruechaiyasak: A Data Mining Framework for Building A Web-Page Recommendation System *Information Research and Development Division. 2015*

- [13] Margaret Rouse: Web application
[https:// searchsoftwarequality.techtarget.com/ definition/ Web-application-Web-app](https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app)
- [14] Huizhi Liang: Real-time Collaborative Filtering Recommender Systems *Department of Computing and Information Systems* The University of Melbourne. 2005.
- [15] Philip Lenhart: Combining Content-based and Collaborative Filtering for Personalized Sports News Recommendations *Department of Informatics*. 2016
- [16] PyOhio Lenhart: “Large-Scale Recommendation System with Python and Spark
<https://www.youtube.com/watch?v=oAByzl71Ak4>
- [17] Cory Maklin: Support Vector Machine Python Example
[https:// towardsdatascience.com/ support- vector- machine- python-example-d67d9b63f1c8](https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8)
- [18] Sung-Hwan Min: Recommender Systems Using Support Vector Machines *Graduate School of Management, Korea Advanced Institute of Science and Technology* 207-43 Cheongrangri-dong, Dongdaemun-gu, Seoul 130-722, Korea shmin@kgsml.kaist.ac.kr
- [19] Chhavi Saluja: Collaborative Filtering based Recommendation Systems exemplified
[https:// towardsdatascience.com/ collaborative- filtering- based-recommendation-systems-exemplified-ecbffe1c20b1](https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1)
- [20] Erion Çano Min: Hybrid Recommender Systems: A Systematic Literature Review *Charles University in Prague Prague, CZ, Czechia*
- [21] Adam Lineberry: Hybrid Content-Collaborative Movie Recommender Using Deep Learning
[https:// towardsdatascience.com/ creating- a- hybrid- content- collaborative- movie- recommender- using- deep- learning- cc8b431618af](https://towardsdatascience.com/creating-a-hybrid-content-collaborative-movie-recommender-using-deep-learning-cc8b431618af)
- [22] Qing Li : An Approach for Combining Content-based and Collaborative Filters *Dept. of Computer Sciences Kumoh National Institute of Technology* Kumi, kyungpook, 730-701, South Korea liqing@se.Kumoh.ac.kr
- [23] Jorge Valverde-Rebaza: Job Recommendation based on Job Seeker Skills: An Empirical Study *Ricardo Puma's*
- [24] Apache Airflow: Apache Airflow. Archived from the original on August 12, 2019. Retrieved September 30, 2019.
<https://airflow.apache.org/docs/stable/project.html>

- [25] ทำความรู้จัก Docker และการใช้งานบน CentOS 7
<https://www.hostpacific.com/using-docker-on-centos7/>
- [26] CompTIA Certification Roadmap
<http://certification.comptia.org/why-certify/roadmap>
- [27] Paul Goodman, Practical Implementation of Software Metrics, 1993.
- [28] PSM Group, Practical Software and Systems Measurement, Version 4.0b, October 2000.
- [29] application programming interface (API), wikipedia
<https://en.wikipedia.org/wiki/API>
- [30] Michal Jaworski and Tarek Ziade, Expert python programming, 2nd edition, PACKT, 2016