

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  
INFORMATION TECHNOLOGY AND BUSINESS ANALYSIS  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2020**

**COPYRIGHT 2020**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

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

## บทคัดย่อ

ในช่วงเวลาที่ผ่านมา ระบบแนะนำงานได้กลายเป็นที่นิยมเป็นอย่างมาก เนื่องจากประสบความสำเร็จในการลดจำนวนการเข้าถึงของยูสเซอร์ที่เข้ามาใช้ค้นหาหางานได้เป็นจำนวน ด้วยการแนะนำงานที่เหมาะสมกับบุคคลนั้น ๆ แต่ถึงอย่างนั้นเทคนิคที่มีใช้อยู่ในระบบทุกวันนี้ ส่วนใหญ่ยังไม่สามารถแนะนำตำแหน่งงานที่เหมาะสมกับโปรไฟล์ผู้หางานได้ดีเท่าที่ควรเช่น แนะนำเฉพาะส่วนของฟิลด์งานแทนที่จะแนะนำงานเป็นกรณีเพื่อให้งานนั้นสอดคล้องกับโปรไฟล์ผู้หางานที่สุด ทั้งนี้เราจึงมีวัตถุประสงค์โดย

I) รวบรวมชุดข้อมูลของตำแหน่งงาน จากเว็บไซต์หางานต่าง ๆ และ โปรไฟล์ผู้ใช้จากเว็บไซต์ลิงก์ดอิน (linkedin) ด้วยเครื่องมือค้นหาของเว็บไซต์ (search engine) เหล่านั้น II) ออกแบบกรอบการทำงานระบบแนะนำโดยขึ้นอิงจากโปรไฟล์ทักษะวิชาชีพของผู้ใช้ III) ดำเนินการประเมินเชิงประจักษ์ของความสามารถในการให้การแนะนำ โดยพิจารณาการกำหนดค่าที่แตกต่างกัน จากกรอบงานที่เสนอ IIII) 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 various job websites and user profiles from LinkedIn a website. With the search engines of those websites II) Design a recommended system framework based on professional skills III) 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

# สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญตาราง	IV
สารบัญรูป	V
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ระบบให้การแนะนำ (Recommendation System)	3
2.2 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)	5
2.3 การหาความสัมพันธ์ระหว่างสองสิ่ง	6
2.4 Support Vector Machine	7
2.5 การสกัดข้อมูล (Data Scraping)	8
2.6 เว็บแอปพลิเคชัน (Web Application)	8
บทที่ 3 วิธีการทดลอง	9
3.1 ส่วนของระบบแนะนำ	9
3.2 Recommendation	11
บทที่ 4 ผลการทดลอง	13
4.1 การทำนายกลุ่มของโปรไฟล์	13
4.2 การจับคู่งานกับโปรไฟล์	15
บทที่ 5 สรุปผล	17
5.1 ผลการทดลองและการแก้ปัญหาในอนาคต	17
5.2 ปัญหาที่เกิดขึ้น	17
บรรณานุกรม	18

# สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางอธิบาย word embeddings	11

# สารบัญรูป

รูปที่	หน้า
2.1	ขั้นตอนทำงาน การกรองแบบอิงเนื้อหา
2.2	ขั้นตอนทำงาน การกรองแบบรวม
2.3	ขั้นตอนทำงาน การแนะนำแบบผสม
3.1	ไปป์ไลน์ของเฟรมเวิร์กการแนะนำ
3.2	คำสำคัญอาชีพไอที พร้อมความสามารถ และคำอธิบายบางส่วน
3.3	ข้อมูลงานที่รวบรวมจากเว็บไซต์ indeed บางส่วน
3.4	ข้อมูลโปรไฟล์จากเว็บไซต์ linkedin บางส่วน
3.5	การทำความสะอาดข้อมูลก่อนและหลัง
3.6	ลดรูปคำในและตัดคำรูปแบบ ngram
3.7	การสร้างขั้นตอนโมเดลด้วยไปป์ไลน์
3.8	การเลือกตัวแปรที่ดีที่สุดให้แก่โมเดล
4.1	การแยกกลุ่มของโปรไฟล์
4.2	รายงานการจำแนกประเภท
4.3	ตาราง confusion matrix
4.4	ตัวอย่างข้อมูลผู้ใช้ในการแนะนำตำแหน่งงาน
4.5	ผลการแนะนำตำแหน่งงานที่สอดคล้องกับผู้ใช้มากที่สุด 10 อันดับแรก



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

การหางานที่เหมาะสมกับตัวผู้หางานนั้น เป็นสิ่งที่เป็ปัญหาอย่างช้านานจนถึงปัจจุบัน ไม่ว่าจะเป็นกลุ่มผู้เรียนจบใหม่ นักศึกษาฝึกงาน หรือแม้กระทั่งผู้ที่ต้องการเปลี่ยนงาน ปัญหานี้มีปัจจัยหลายส่วนและหลายกลุ่ม เช่น กลุ่มของผู้ฝึกงานและผู้ที่ยื่นจบใหม่ มักยังไม่ทราบความต้องการของตนเองว่าต้องการทำงานในแขนงไหน ตนเองถนัดกับสิ่งใด และปัญหาภาพรวมที่พบเจอเยอะที่สุดคือความยุ่งยากในการหางาน โดยที่ผู้หางานจำเป็นต้องค้นหางานด้วยตนเองทีละตำแหน่งและอ่านรายละเอียดตำแหน่งเหล่านั้นว่ามีความต้องการตรงกับความสามารถเราหรือไม่ 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 หลังจากนั้นจึงนำมารวมเข้ากับเทคนิคการกรองแบบร่วม (Collaborative Filtering) ในขอบเขตของกรองร่วมโดยอิงจากยูสเซอร์ (User Based Filtering) โดยเป็นการแนะนำที่อ้างอิงทักษะวิชาชีพของยูสเซอร์อื่นที่ทำงานแล้วว่ามีความคล้ายคลึงกับเรามากเพียงใด โดยสามารถตีความได้ว่าถ้ายูสเซอร์คนนั้นมีทักษะคล้ายกับเรา เราก็อาจจะได้คำแนะนำงานของยูสเซอร์คนนั้นเช่นกัน ซึ่งการนำเทคนิคทั้งสองวิธีมาใช้รวมกันเพื่อให้คำแนะนำ เราเรียกเทคนิคนี้ว่าการแนะนำแบบผสม (Hybrid Recommendation)

## 1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบแนะนำตำแหน่งงาน โดยผสมผสานการอ้างอิงจากทักษะของ โปรไฟล์ผู้ใช้ และระหว่างผู้ใช้ด้วยกัน
2. ประยุกต์ระบบแนะนำตำแหน่งงานกับเว็บแอปพลิเคชัน ที่จะเปิดให้ใช้สำหรับหางานและลงตำแหน่งงาน
3. ออกแบบและพัฒนาเว็บแอปพลิเคชันหางานที่สามารถจับคู่โปรไฟล์ผู้ใช้กับตำแหน่งงานได้

## 1.3 ขอบเขตของงานวิจัย

1. พัฒนาระบบแนะนำตำแหน่งงาน โดยใช้เทคนิคผสมระหว่าง การกรองแบบอิงเนื้อหา (Content Based Filtering) และ การกรองแบบร่วม (Collaborative Filtering)
2. พัฒนาเว็บแอปพลิเคชันหางาน ที่เชื่อมต่อกับระบบแนะนำตำแหน่งงาน
3. ข้อมูลที่ใช้ในการสร้างระบบ ในส่วนของข้อมูลตั้งต้น โปรไฟล์ได้ใช้ข้อมูลจาก ลิงกด์อิน (Linkedin) และส่วนของตำแหน่งงานได้ใช้ข้อมูลจาก อินดีด (Indeed)
4. ภาษาที่ใช้จะเป็นภาษาอังกฤษทั้งหมด
5. ขอบเขตของการแนะนำจะอยู่ในขอบเขตของไอที

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

ทางเราได้สร้างระบบแนะนำตำแหน่งงานขึ้นมาเพื่อช่วยในลดปัญหาความยุ่งยากซับซ้อนและเสียเวลากับวิธีหางานในปัจจุบัน โดยนำเสนอการจับคู่ระหว่างโปรไฟล์และตำแหน่งงานที่เหมาะสมกัน

1. ช่วยให้ผู้ที่ยังไม่มีงานทำในปัจจุบันสามารถหางานได้ผ่านขั้นตอนการจับคู่ตำแหน่งงานที่ทางเราสร้างขึ้น
2. เพื่อสร้างฐานข้อมูลที่เป็นอนาคต ในการนำข้อมูลตำแหน่งงาน ไปวิเคราะห์และใช้งานในอนาคต
3. เพื่อวิจัยและค้นคว้าการสร้างระบบแนะนำที่มีประสิทธิภาพและต่อยอดได้ในอนาคต

## บทที่ 2

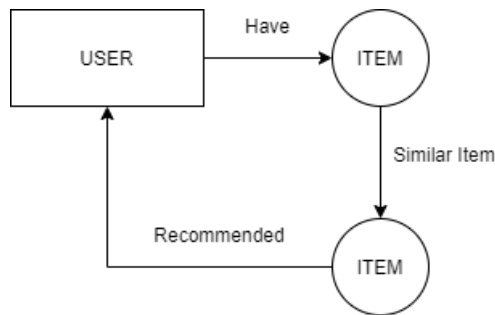
# แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 ระบบให้การแนะนำ (Recommendation System)

ระบบให้การแนะนำ [1] เป็นระบบสนับสนุนการตัดสินใจที่จะให้การแนะนำสินค้าหรือบริการที่มีความเหมาะสมกับรูปแบบและพฤติกรรมของลูกค้าหรือผู้ใช้แต่ละคน โดยอาศัยข้อมูลของผู้ใช้งานร่วมกับข้อมูลประกอบภายนอกมาใช้ในการวิเคราะห์คัดกรอง ให้ได้สิ่งที่มีความหมายที่เหมาะสมกับผู้ใช้งาน โดยมีเทคนิคแบ่งย่อยได้เป็นสองชนิดหลัก ๆ คือ การกรองแบบอิงเนื้อหา (Content Based Filtering) การกรองแบบร่วม (Collaborative Filtering) และการกรองแบบผสม (Hybrid Recognition) [2]

#### 2.1.1 การกรองแบบอิงเนื้อหา (Content Based Filtering)

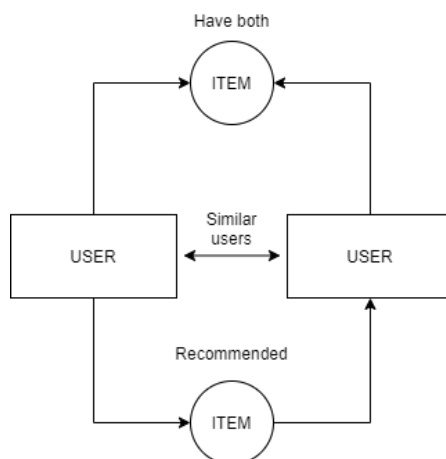
การกรองแบบอิงเนื้อหา [3] เป็นการกรองข้อมูลโดยอิงจากเนื้อหาและคุณสมบัติที่สอดคล้องกับโปรไฟล์ของผู้ใช้ การใช้เทคนิคนี้ เนื่องจากการดูโปรไฟล์ของแต่ละคนแยกจากกันทำให้การแนะนำไอเทมนั้นค่อนข้างเหมาะสมกับผู้ที่มีโปรไฟล์ที่แตกต่างกัน แต่มีข้อเสียคือจำเป็นต้องเตรียมแคตตาล็อกหมวดหมู่ให้กับข้อมูล และสร้างฟีเจอร์ที่เหมาะสมที่จะอธิบายตัวข้อมูลเหล่านั้น และการแนะนำจะไม่สามารถแนะนำไอเทมที่แตกต่างไปจากโปรไฟล์ของผู้ใช้เหล่านั้นได้มากนัก



รูปที่ 2.1 ขั้นตอนทำงาน การกรองแบบอิงเนื้อหา

### 2.1.2 การกรองแบบร่วม (Collaborative Filtering)

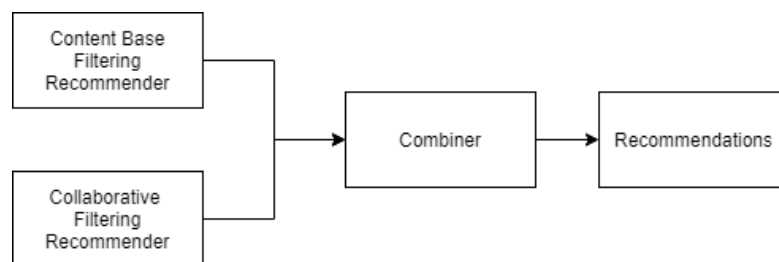
การกรองแบบร่วม [4] เป็นการกรองข้อมูลโดยอิงจากโปรไฟล์ผู้ใช้อื่น ๆ เข้ามาช่วยในการตัดสินใจ โดยการดูโปรไฟล์ของเราเทียบกับความสอดคล้องกับโปรไฟล์ผู้ใช้อื่นๆ ถ้าโปรไฟล์นั้น ๆ มีความสอดคล้องกับโปรไฟล์ของเรามากนั้นหมายความว่าเราอาจมีความสนใจเหมือนกับโปรไฟล์นั้น ๆ และระบบจะแนะนำไอเทมของโปรไฟล์นั้นออกมา แต่เทคนิคนี้มีข้อเสียข้อใหญ่นั่นคือปัญหา Cold Start [5] คือระบบไม่สามารถแนะนำไอเทมให้เหมาะสมได้ เนื่องจากข้อมูลไม่มากพอในการแนะนำ



รูปที่ 2.2 ขั้นตอนทำงาน การกรองแบบร่วม

### 2.1.3 ระบบให้การแนะนำแบบผสม (Hybrid Recommendation)

ระบบการแนะนำแบบผสม [2] เป็นการรวมการทำงานของระบบแนะนำที่ใช้เทคนิคการกรองแบบร่วม (Collaborative Filtering) และระบบแนะนำจากการกรองโดยอิงจากเนื้อหา (Content Based Filtering) เข้าด้วยกันเพื่อประสิทธิภาพที่ดียิ่งขึ้น และแก้ไขข้อด้อยของแต่ละเทคนิคเช่น การกรองแบบร่วมที่มีปัญหาใหญ่คือ Cold Start [5] ที่ต้องการข้อมูลตั้งต้นเป็นจำนวนหนึ่งเพื่อจะได้แนะนำได้อย่างถูกต้อง ตรงจุดการกรองโดยอิงจากเนื้อหาสามารถแก้ไขข้อด้อยตรงนี้ได้เนื่องจากการกรองโดยอิงจากเนื้อหานั้น แ่จากโปรไฟล์ของเรากับไอเทมเป็นแคสต่อเคสเท่านั้น



รูปที่ 2.3 ขั้นตอนทำงาน การแนะนำแบบผสม

## 2.2 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)

การประมวลผลภาษาธรรมชาติ (NLP) [6] เป็นแขนงหนึ่งของสาขาปัญญาประดิษฐ์ (Artificial Intelligence) ที่ทำให้เครื่องจักรมีความสามารถในการอ่านทำความเข้าใจและเข้าใจความหมายของภาษามนุษย์ได้ กล่าวคือ NLP แสดงถึงการจัดการภาษามนุษย์โดยอัตโนมัติเช่นการพูด ข้อความ หรือแม้กระทั่งแนวคิดที่สนใจ โดยได้มีการนำไปประยุกต์ใช้ในแขนงมากมายเช่น ช่วยในการทำความเข้าใจและคาดการณ์กลุ่มของผู้ใช้จากโปรไฟล์ของผู้ใช้เหล่านั้น เป็นต้น

### 2.2.1 Word Embedding

Word Embedding [7] 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

#### 2.2.1.1 Word2Vec

Word2Vec [7] Pre-trained weight model หรือแบบจำลองน้ำหนักที่ผ่านการเทรนมาแล้วล่วงหน้า แล้ว 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 *dolor* 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 *dolor* เป็นอินพุต

### 2.2.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TFIDF [16] ใช้เพื่อชั่งน้ำหนักของคำสำคัญ (Keyword) ในเอกสารใด ๆ เพื่อกำหนดความสำคัญให้กับคำสำคัญเหล่านั้นตามจำนวนครั้งที่ปรากฏในเอกสาร หรือก็คือยิ่งคะแนน  $TF * IDF$  (น้ำหนัก) สูงเท่าไรคำนั้นก็ยิ่งสำคัญเท่านั้น ในทุกคำหรือคำศัพท์แต่ละคำจะมีคะแนน TF และ IDF อยู่เสมอ ผลคูณของคะแนน TF และ IDF ของคำหนึ่งจะเรียกว่าน้ำหนัก  $TF*IDF$  ของคำนั้น ๆ

ความถี่ (TF: Term Frequency) ของคำคือจำนวนครั้งที่ปรากฏในเอกสาร เมื่อทราบถึง TF แล้วเราจะสามารถบอกได้ว่ามีคำนั้นปรากฏในเอกสารบ่อยเท่าใด

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

ความถี่เอกสารผกผัน (IDF: Inverse Document Frequency) ของคำคือการวัดความสำคัญของคำเหล่านั้นในคลังข้อมูลคำ (Copus) 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

### 2.3 การหาความสอดคล้องระหว่างสองสิ่ง

ในการหาความสอดคล้องระหว่างสองสิ่ง [9] 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" เมื่อนำมาเปลี่ยนเป็นเมทริกซ์เทคนิคการนับคำ (count vectorizer) จะได้เมทริกซ์ [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

## 2.4 Support Vector Machine

Support Vector Machine (SVM) [10] เป็นเทคนิค Pattern Recognition แบบ Supervised Learning ถูกใช้ในเคส Classification และ Regression โดยภายในงานนี้ได้ถูกใช้เพื่อ Classification ตำแหน่งงาน ด้วยการ สร้าง 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

## 2.5 การสกัดข้อมูล (Data Scraping)

การสกัดข้อมูล [11] เป็นเทคนิคในการเข้าถึงข้อมูลจากเว็บไซต์เพื่อทำและสกัดข้อมูลที่ต้องการในการสกัดข้อมูลจากข้อความที่ดึงมาจากเว็บไซต์สามารถใช้ไลบรารี beautifulsoup ของภาษา python เพื่อช่วยในการสกัดข้อมูลให้มีความง่ายขึ้นได้ กรณีที่เว็บไซต์ที่ทำงานโดยการเรนเดอร์หน้าเพจทั้งหน้าแล้วส่งมาให้ผู้ใช้ (client) เราสามารถดึงข้อมูลของทั้งหน้ามาใช้ได้โดยตรงและสกัดข้อมูลจากที่กล่าวมาข้างต้น แต่บางเว็บไซต์ที่มีการทำงานแบบฝั่งไคลเอนต์ มีการแสดงผลข้อมูลเป็นแบบ Asynchronous ซึ่งทำให้ข้อมูลปรากฏขึ้นไม่พร้อมกัน โดยจะขึ้นอยู่กับการทำงานของผู้ใช้เช่น คลิกเปิดเลื่อนลงเพื่อโหลดฟีด จะไม่สามารถดึงข้อมูลทั้งหน้าได้จำเป็นต้องจำเป็นต้องแก้ปัญหาโดยทำการจำลองบราวเซอร์เพื่อจำลองการทำงานของใช้ขึ้นมาโดยใช้ไลบรารีชื่อว่า ซีลีเนียม (selenium)

## 2.6 เว็บแอปพลิเคชัน (Web Application)

Web Application [12] ทำหน้าที่ในการเป็นช่องทางในการเชื่อมต่อระหว่างเว็บไซต์กับผู้ใช้บริการไอพีโอจากที่อื่นเป็นตัวกลางที่ทำให้โปรแกรมสามารถประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น ๆ ได้ เช่น google map ที่ทาง google ให้บริการให้ผู้ใช้สามารถนำเว็บไซต์ของตนเองเชื่อมต่อกับแผนที่ของ google ได้

### 2.6.1 จาวาสคริปต์ (Javascript)

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



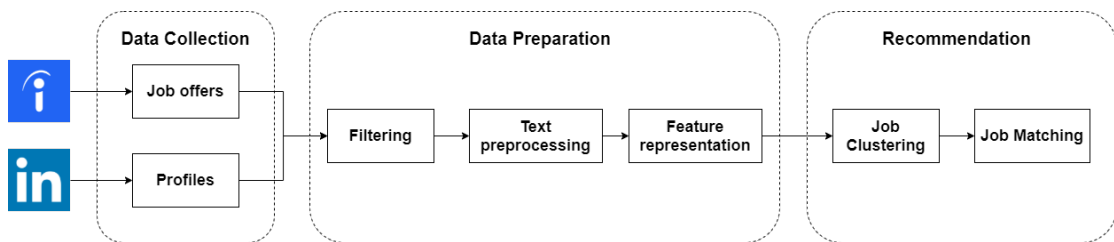
## บทที่ 3

### วิธีการทดลอง

ในบทนี้เรากล่าวถึงขั้นตอนและกระบวนการทำงานของระบบแนะนำงานตามทักษะอาชีพและโปรไฟล์ของผู้ใช้ในส่วนแรก และเว็บแอปพลิเคชันในส่วนที่สอง โดยมีจุดประสงค์เพื่อให้ระบบสามารถแนะนำตำแหน่งงานได้อย่างมีประสิทธิภาพ และดำเนินการทดลองเพื่อวัดประสิทธิภาพของโมเดลที่เรานำเสนอนี้ว่าให้ผลอย่างไร สามารถแนะนำงานได้เหมาะสมเพียงพอที่จะนำไปใช้จริงหรือไม่

#### 3.1 ส่วนของระบบแนะนำ

ในส่วนของระบบแนะนำ จะเน้นไปที่การแสดงกรอบและวิธีการสร้างโมเดลแนะนำโดยใช้เทคนิคการกรองโดยอิงจากเนื้อหาและการกรองแบบรวม



รูปที่ 3.1 ไปป์ไลน์ของเฟรมเวิร์กการแนะนำ

##### 3.1.1 การรวบรวมข้อมูล

เรารวบรวมชุดข้อมูลตำแหน่งงานจากเว็บไซต์จัดหางานอินดีด (Indeed) และชุดโปรไฟล์ผู้ใช้จากลิงก์ดอิน (Linkedin) ที่เป็นเว็บไซต์เครือข่ายสังคมด้านเครือข่ายธุรกิจที่ทุกคนรู้จักกันดีด้วยการสกิดข้อมูล (Data Scapper) จากเว็บไซต์ทั้งสอง โดยส่วนของตำแหน่งงานนั้น เราได้สร้างรายการคำหลักจากอุตสาหกรรมไอทีและใช้เป็นคำค้นหาโดยใช้เครื่องมือค้นหา (search engine) ของทางเว็บไซต์ indeed แล้วบันทึกผลลัพธ์ลงบนไฟล์ csv เช่นเดียวกับโปรไฟล์เราสกิดข้อมูลโปรไฟล์มืออาชีพจากเว็บไซต์ linkedin ซึ่งใช้เครื่องมือค้นหาของทางเว็บไซต์ในการค้นหาโดยเราตั้งบริษัทที่เป็นมาตรฐานอย่างเช่นอโกด้า (agoda) และกสิกร (kasikorn) เพื่อให้ได้ผู้มีอาชีพที่ข้อมูลโปรไฟล์ครบถ้วน หลังจากนั้นจึงบันทึกข้อมูลลงบน csv เพื่อนำไปใช้งานต่อไป

0	Mobile Application	Java Application Developer Mobile Software Eng...	Mobile developers are fluent in object-orient...
1	Database Administrator	Database Administrator ETL SQL Server Database...	Implement, support and manage the corporate da...
2	Web Developer	Web Designer .Net Web Developer Web Developer ...	Are fluent in the core web development scripti...
3	Help Desk Technician	Help Desk Technician IT Support Specialist Des...	Diagnose computer errors and provide technical...
4	Network Administrator	Network Administrator Network Manager Cisco Ne...	Configure and maintain the organization's inte...
5	Graphic Designer	Graphic Designer Packaging Designer Copy Write...	Design websites, email and newsletter template...
6	IT Security	Information Security Consultant Information Se...	Develop plans to safeguard computer files agai...
7	Software Engineer	Software QA Engineer Software Systems Engineer...	Software engineers analyze end-users' needs an...
8	IT Manager	Chief Information Officer (CIO) Chief Technolo...	Are responsible for strategic IT planning, inc...

รูปที่ 3.2 คำสำคัญอาชีพไอที พร้อมความสามารถ และคำอธิบายบางส่วน

job_type	link	job_title	company	desc	skill
0	Mobile Application Developer	Native Mobile Application Developer (iOS/Android)	Haneke Design-Tampa, FL	love crafting beautiful apps squeezing device ...	risk,git,solid,swift,kotlin,design,mobile,comm...
1	Mobile Application Developer	Mobile Application Developer (iOS)	Love's Travel Stops & Country Stores2,438 rev...	req id basic purpose member application develo...	testing,git,swift,hospitality,travel,scrum,des...
2	Mobile Application Developer	Mobile Application developer	Ace-stack LLC-Tampa, FL	job detail position mobile application develop...	testing,security,swift,kotlin,design,analysis,...
3	Mobile Application Developer	Senior Mobile Application Developer	DISCOVERTEC LLC4 reviews-Jacksonville, FL 32256	description mobile application developer role ...	debugging,swift,analytics,xcode,mobile,product...
4	Mobile Application Developer	Mobile Application Developer	ProBar10 reviews-United States	mobile application developer position requires...	wireless,integration,mobile,design

รูปที่ 3.3 ข้อมูลงานที่รวบรวมจากเว็บไซต์ indeed บางส่วน

	name	position	company	location	bio	education	exp_position	exp_company	skill
0	aditya singh	senior software engineer	agoda	bangkok metropolitan area, thailand	a software engineer with 2.5+ years of industr...	shri l.r. tiwari college of engineering,thakur...	senior software engineer,on a career break	agoda,career break	java,nodejs,python,web development,algorithms,...
1	kan-anek atichativat	software engineer	agoda	bangkok metropolitan area, thailand	experienced software engineer adept in all pro...	university of essex,king mongkut's institute o...	software engineer (data team),software enginee...	agoda,mohara,burda international asia,kurume l...	node.js,react.js,mongodb,software engineering,...
2	ambas chobsanti	ios developer	agoda	bangkok metropolitan area, thailand	i'm ambas chobsanti, an experienced programmer ...	king mongkut's institute of technology ladkrab...	ios engineer,senior ios developer,ios develop...	agoda,lazada group,taskworid,humanica ltd.,kin...	swift,objective-c,ios,ios development,web serv...
3	patipol sittiyanon	software engineer	agoda	bangkok metropolitan area, thailand	an energetic and enthusiastic person who enjoy...	king mongkut's university of technology thonbu...	software engineer,backend developer,programmer...	agoda,scala360,dst thailand,soffico gmbh,tesa	scala,akka,play framework,scrum,docker,javascr...
4	chutimon tangtanaporn	ux designer	agoda	yannawa, bangkok, thailand	ux & visual designer ..	chiang mai university,regina coeli college	photographer & graphic designer,visual designe...	freelance,cpf it center co.,ltd,amarin printin...	graphic design,graphics,photography,web design...

รูปที่ 3.4 ข้อมูลโปรไฟล์จากเว็บไซต์ linkedin บางส่วน

### 3.1.2 การเตรียมข้อมูล

ถึงแม้ว่าเราจะดึงข้อมูลงานโดยใช้คำสำคัญไอทีแล้ว แต่ก็ยังมีบางงานที่ไม่สอดคล้องกับฟิลด้นั้นจริง ๆ เราได้เราแก้ปัญหาด้วย I.) การใช้พจนานุกรมศัพท์ไอทีเพื่อจับคู่ว่างานนั้น ๆ อยู่ในฟิลดที่ควรจะอยู่จริง ๆ II.) เมื่อเราก็คัดกรองตำแหน่งงานได้แล้วต่อไปคือการประมวลผลข้อความล่วงหน้าในส่วนนี้เราทำการลบคำหยุดคำพิเศษช่องว่างต่าง ๆ และลดรูปของคำอยู่ในรูปแบบรากศัพท์ III.) แสดงคุณสมบัติข้อมูล (feature representation) โดยมีจุดมุ่งหมายเพื่อแสดงข้อความเหล่านั้นในรูปแบบแบบเวกเตอร์โมเดล เพื่อจุดประสงค์นี้เราได้นำวิธีการสองแบบคือ word embeddings และ TF-IDF โดยเทคนิค word embeddings เราเลือก Word2Vec ซึ่งมีหลากหลายรูปแบบเช่น CBOW และ Skip-Gram และ N-Gram(bigrams และ trigrams) มาใช้ในการทดลองเพื่อดูว่ารูปแบบใดจะดีที่สุด เรา

สามารถดูคำอธิบาย corpus ที่ใช้สำหรับ word embeddings

Dataset	Documents	Tokens
Profile	417	1581
Jobs	4748	24333

ตารางที่ 3.1 ตารางอธิบาย word embeddings

Before: Native Mobile Application Developer in Thailand (iOS/Android)  
 After: native mobile application developer in thailand ios android

รูปที่ 3.5 การทำความสะอาดข้อมูลก่อนและหลัง

['native', 'mobile', 'application', 'developer', 'thailand', 'io', 'android']

รูปที่ 3.6 ลดรูปคำในและตัดคำรูปแบบ ngram

## 3.2 Recommendation

ในส่วนสุดท้ายนี้ เทคนิคการกรอง โดยอิงจากเนื้อหาด้วยเราเลือกกลุ่มงาน โดยอ้างอิงจากระยะทางของผู้ใช้ (profile) ด้วยระยะทางโคไซน์ (cosine distance) ซึ่งเราได้นำโมเดล SVM เข้ามาใช้ในการทำนายกลุ่มและใช้ Gradient Descent เข้ามาช่วยเช่นเดียวกับเทคนิคการกรองแบบอิงร่วม (Collaborative Filter) ที่มีวิธีการเหมือนกันแต่เปลี่ยนข้อมูลสำหรับโมเดลเป็นข้อมูลทักษะของผู้ใช้ (profile) ทั้งหมดแทนโดยหาระยะทางผู้ใช้ที่ใกล้เคียงกับเรามากที่สุด และทำการแนะนำตำแหน่งงานที่ผู้ใช้นั้นทำงาน

สำหรับการสร้างโมเดลนั้นเราได้แบบข้อมูลสำหรับเทรนโมเดลไว้ที่ 77% และส่วนของทดสอบ 33% โดยข้อมูลที่ใช้เป็นข้อมูลรายละเอียดตำแหน่งงานและมีเฉลยเป็นตำแหน่งงานที่ค้นหาจากบนเว็บไซต์ เมื่อทำการเทรนโมเดลเราได้ใช้ Pipeline ในการสร้างลำดับการทำโดยแบ่งเป็น 3 ขั้นตอนคือ Count Vectorizer, TFIDF และ SGDClassifier(SVM) และนำ GridSearchCV เข้ามาใช้ในการ Optimized โมเดลตามตัวเลือกที่ตั้งไว้นั้นคือ ngram, penalty และ alpha

เมื่อทำการสร้างโมเดลเพื่อทำนายกลุ่มของผู้ใช้งานแล้วนั้น เรานำกลุ่มเหล่านั้นมาความสอดคล้องกับโปรไฟล์ที่ต้องการแนะนำโดยอ้างอิงจากทักษะวิชาชีพของโปรไฟล์และตำแหน่งงานทั้งหมดในหมวดนั้น ด้วยการหาระยะทางทักษะกับรายละเอียดของตำแหน่งงานแต่ละกัน โดยใช้เทคนิคระยะทางโคไซน์ (cosine distance) ซึ่งเราจะได้ผลอันดับความสอดคล้องออกเพื่อแนะนำงานให้ผู้ใช้งานต่อไป

```
SVM_MODEL = Pipeline([('vect', CountVectorizer()),
                        ('tfidf', TfidfTransformer()),
                        ('clf-svm', SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, random_state=42)),])
```

### รูปที่ 3.7 การสร้างขั้นตอนโมเดลด้วยไปป์ไลน์

```
parameters_svm = {'vect__ngram_range': [(1, 1), (1, 2)],
                  'tfidf__use_idf': (True, False),
                  'clf-svm__penalty': ['l2'],
                  'clf-svm__alpha': (1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3),}
gs_clf_svm = GridSearchCV(SVM_MODEL, parameters_svm, n_jobs=-1)
gs_clf_svm = gs_clf_svm.fit(X_train, y_train)
gs_clf_svm.best_score_
```

### รูปที่ 3.8 การเลือกตัวแปรที่ดีที่สุดให้แก่โมเดล

## บทที่ 4

### ผลการทดลอง

#### 4.1 การทำนายกลุ่มของโปรไฟล์

ในส่วนนี้เรานำเสนอการทดลองเชิงประจักษ์ที่เน้นการประเมินคุณภาพของการแนะนำงาน โดยการทดลองนี้เราได้มีข้อมูลโปรไฟล์ผู้เชี่ยวชาญจาก LinkedIn จำนวน 417 ตัวอย่าง และตำแหน่งงานจาก Indeed จำนวน 4,748 ตัวอย่าง, ทั้งโปรไฟล์และตำแหน่งงานเนื่องจากเฟิร์ดไอทีที่กว้างขวางโปรไฟล์จึงมีความแตกต่างกันบ้างเล็กน้อยในแต่ละกลุ่มโปรไฟล์ จากภาพด้านล่างแสดงการกระจายของฟิลด์ย่อยภายในของโปรไฟล์ 417 รายการซึ่งแสดงให้เห็นถึงจำนวนนักพัฒนาที่มีเป็นจำนวนมาก

chief information officer cio	30
cloud engineer	5
data scientist	33
database administrator	22
devops engineer	10
graphic designer	23
help desk technician	1
manager	12
mobile application developer	40
network administrator	1
network architect	8
programmer	27
project manager	45
security	3
software engineer	59
system analyst	23
web developer	75

รูปที่ 4.1 การแยกกลุ่มของโปรไฟล์

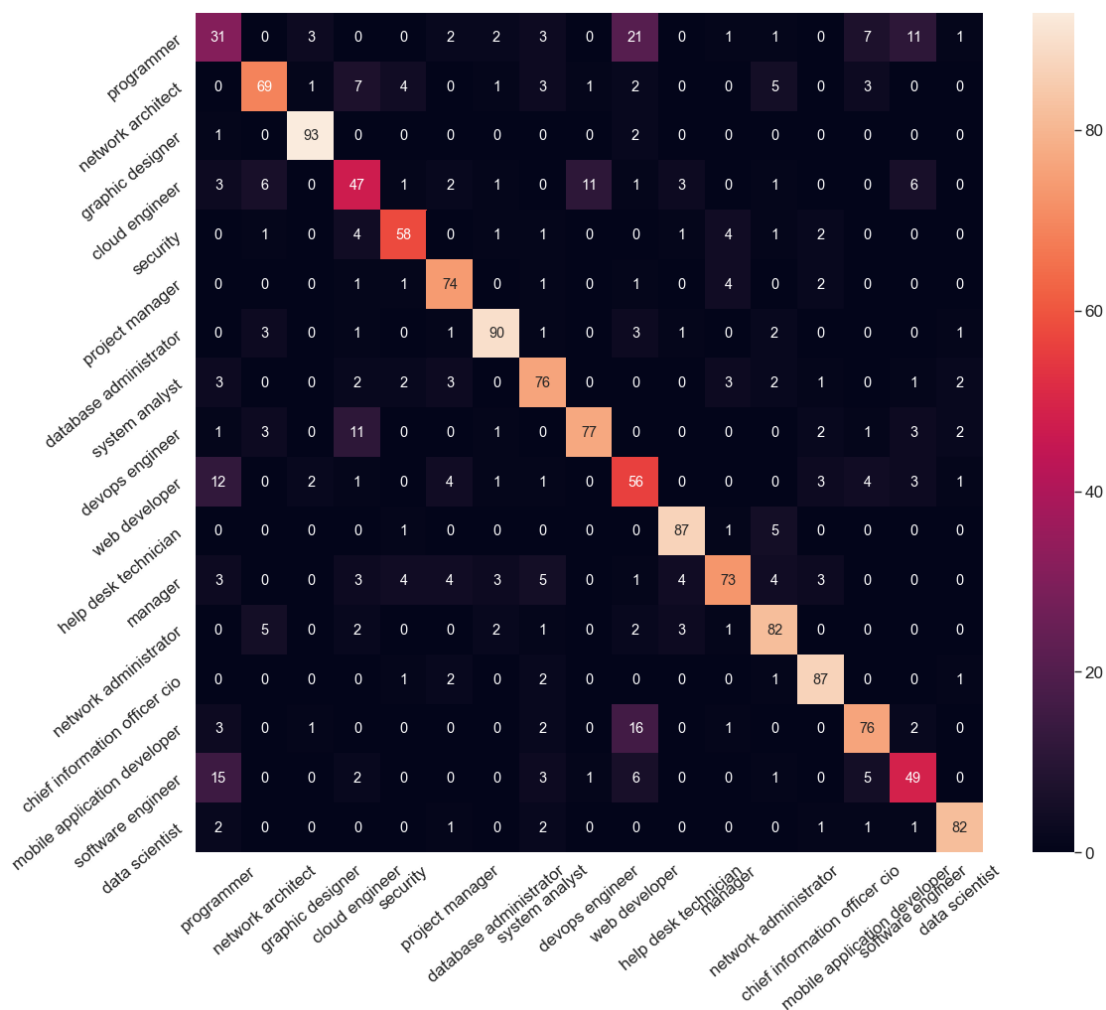
ในการประเมินพวกเราได้ใช้ระบบแนะนำสร้างทำนายข้อมูลตำแหน่งงานสำหรับทดสอบ โดยใช้โมเดลที่ผ่านการเทรนมาแล้วได้โดยข้อมูลทดสอบนั้นเป็นข้อมูลรายละเอียดงานที่แบ่งมาจากส่วนเทรนจำนวน 33% หรือ 1,567 ตัวอย่าง โดยจากรายงานการจำแนกประเภทเราจะเห็นว่าในกลุ่มตำแหน่งงานที่มีความคลุมเครือและใกล้เคียงกันเช่น 'web developer' และ 'software engineer' มีความแม่นยำที่น้อยเมื่อเทียบกับกลุ่มอื่น ๆ

	precision	recall	f1-score	support
chief information officer cio	0.86	0.93	0.89	94
cloud engineer	0.58	0.57	0.58	82
data scientist	0.91	0.91	0.91	90
database administrator	0.88	0.87	0.88	103
devops engineer	0.86	0.76	0.81	101
graphic designer	0.93	0.97	0.95	96
help desk technician	0.88	0.93	0.90	94
manager	0.83	0.68	0.75	107
mobile application developer	0.78	0.75	0.77	101
network administrator	0.78	0.84	0.81	98
network architect	0.79	0.72	0.75	96
programmer	0.42	0.37	0.39	83
project manager	0.80	0.88	0.84	84
security	0.81	0.79	0.80	73
software engineer	0.64	0.60	0.62	82
system analyst	0.75	0.80	0.78	95
web developer	0.50	0.64	0.56	88
accuracy			0.77	1567
macro avg	0.77	0.77	0.76	1567
weighted avg	0.77	0.77	0.77	1567

**รูปที่ 4.2** รายงานการจำแนกประเภท

ซึ่งจากการวิเคราะห์เราคาดว่าเกิดจากการใช้คำสำคัญ (keyword) ของ โปรไฟล์ที่ดึงข้อมูลมาจาก linkedin โดยโปรไฟล์เหล่านี้มักใส่ทักษะวิชาชีพครอบคลุมในส่วนของการพัฒนาหรืออีกนัยคือนักพัฒนาส่วนใหญ่เป็น full-stack ที่สามารถทำได้หลายตำแหน่งงาน แต่เนื่องจากตำแหน่งงานที่ทำในปัจจุบันของโปรไฟล์เหล่านั้นสามารถตั้งได้แค่ตำแหน่งเดียว ทำให้เกิดความคลาดเคลื่อนในการทำนาย ทั้งนี้เมื่อเราทำการพล็อตการฟ confution matrix เราจะเห็นถึงความผิดพลาดในการทำนายของตำแหน่ง 'software engineer' 'devops engineer' และ web developer เป็นจำนวนหนึ่งส่งผลให้ค่าความแม่นยำรวมเหลือแค่ 77% ซึ่งทางเราถือว่าเกือบพอใช้ได้





รูปที่ 4.3 ตาราง confusion matrix

## 4.2 การจับคู่งานกับโปรไฟล์

เราทำการแบ่งกลุ่มของโปรไฟล์ออกเป็นกลุ่มย่อยๆ เนื่องจากในขั้นตอนการจับคู่งานกับโปรไฟล์ โดยเทคนิคการหาระยะทางโคไซน์ (cosine distance) นั้นจำเป็นต้องใช้ทรัพยากรเครื่องที่ประมวลผลเป็นจำนวนมาก การแบ่งกลุ่มออกเป็นกลุ่มย่อยๆ จะช่วยแบ่งเบาภาระของการประมวลผลเพื่อหาคำแนะนำที่เหมาะสมกับโปรไฟล์ผู้ใช้งานมากที่สุด

ผลลัพธ์การแนะนำตำแหน่งงาน โดยการหาความสอดคล้องระหว่าง โปรไฟล์ผู้ใช้และงานในกลุ่ม เรยกตัวอย่างผู้ใช้งานหนึ่งซึ่งเป็น senior software engineer ทำงานอยู่ agoda เพื่อเป็นตัวอย่างผลลัพธ์การแนะนำตำแหน่งงาน

```

name          aditya singh
position       senior software engineer
company        agoda
location       bangkok metropolitan area, thailand
bio            a software engineer with 2.5+ years of industry experience having worked on 10+ projects in 6+ languages and ac
ross
education      shri l.r. tiwari college of engineering,thakur college of science and commerce,ryan international school
exp_position    senior software engineer,on a career break
exp_company     agoda,career break
skill          java,nodejs,python,web development,algorithms,data structures,unit testing,design patterns,web applications,obj
ect-oriented programming (oop),software development,php,mongodb,c,mysql,javascript,sql,junit,node.js,problem solving,redis,elas
ticsearch,logstash
predicted      software engineer

```

#### รูปที่ 4.4 ตัวอย่างข้อมูลผู้ใช้ในการแนะนำตำแหน่งงาน

โดยเราทำการหา similarity distance ของตำแหน่งงานที่สอดคล้องกับโปรไฟล์นี้มากที่สุด 10 อันแรกจะได้ผลลัพธ์ดังนี้

```

[0.24659848095803594]
[ 1 ] Associate, Software Development Engineer
job skill: sql,excel,integration,ci,automation,management,design,strategy,programming,nosql,business,mongodb,fix,organization,s
crum,backend,rest,web,java,agile,soap,oop,engineering,science

[0.18238430103502123]
[ 2 ] Web Developer (Junior)
job skill: html,leadership,mongodb,management,python,organization,angular,design,marketing,backend,web,integration,mysql,postgr
esql,agile,javascript,react

[0.18189719562018503]
[ 3 ] Software Engineer
job skill: ui,hiring,youtube,search,mobile,scalability,javascript,english,security,leadership,c,storage,design,programming,busi
ness,go,linux,organization,web,unix,java,python,processing,android,networking,engineering,science,communication

[0.18144497650147817]
[ 4 ] Software Engineer II
job skill: screening,excel,production,architecture,integration,javascript,powershell,security,c,design,programming,business,aw
s,innovation,organization,java,agile,testing,python,hardware,engineering,science,communication,recruiting

[0.17202219458852253]
[ 5 ] Software Engineer, Front End
job skill: ui,optimization,youtube,search,mobile,scalability,javascript,security,leadership,accessibility,c,storage,angularjs,d
esign,programming,business,go,web,java,python,processing,android,networking,engineering,science

[0.16943857979778443]
[ 6 ] Junior Software Engineer
job skill: access,security,research,leadership,linux,training,python,organization,design,programming,science,web,integration,in
frastructure,unix

[0.13890151155800623]
[ 7 ] Software Engineer, University Grad
job skill: php,facebook,travel,python,running,engineering,web,business,xhtml,science,java,javascript,perl

[0.12926993860203706]
[ 8 ] Junior Software Engineer
job skill: mvc,security,html,insurance,spring,training,redux,python,amazon,design,scalability,engineering,web,science,javascript,react

[0.12197569533584585]
[ 9 ] Associate Software Engineer
job skill: testing,jms,linux,mathematics,containerization,engineering,science,java,agile,troubleshooting

[0.0432628913961467]
[ 10 ] Software Engineer
job skill: communication,solid,linux,storage,design,science,web,multimedia,administration

```

#### รูปที่ 4.5 ผลการแนะนำตำแหน่งงานที่สอดคล้องกับผู้ใช้งานมากที่สุด 10 อันแรก



## บทที่ 5

### สรุปผล

#### 5.1 ผลการทดลองและการแก้ปัญหาในอนาคต

ในการทดลองนี้ เราได้เสนอกรอบการทำงานของระบบแนะนำตำแหน่งงานโดยใช้เทคนิคการกรองแบบเนื้อหา โดยการทำงานถูกแบ่งออกเป็นสองส่วนคือ ส่วนของการแบ่งกลุ่มผู้ใช้ในหมวดฟิลด์ไอทีต่าง ๆ ที่กำหนดไว้โดยใช้โมเดล SVM ซึ่งจากผลการทดลอง โมเดลมีความแม่นยำในการแบ่งกลุ่มโปรไฟล์เพียง 77% ซึ่งถือว่าเกือบพอใช้ได้ และในส่วนที่สองจะเป็นการจับคู่ระหว่างโปรไฟล์ที่ทราบกลุ่มฟิลด์แล้วกับตำแหน่งงานในฟิลด์นั้น ๆ โดยประสิทธิภาพจะขึ้นอยู่กับการแบ่งกลุ่มของโมเดลการแบ่งกลุ่มโปรไฟล์ว่ามีความแม่นยำเพียงใด

ทิศทางในอนาคตของงานเราจะมุ่งเน้นไปที่การประเมินผลที่มีความละเอียดยิ่งขึ้นโดยพิจารณาการรวบรวมข้อมูลให้มีความชัดเจนมากที่สุด และจำนวนข้อมูลที่มากกว่าเดิม รวมถึงประเมินที่กลุ่มที่ครอบคลุมมากขึ้นซึ่งหมายถึงตำแหน่งงานที่ผู้หางานจะได้รับหลากหลายมากขึ้น และในเฟสต่อไปหลังจากที่ปรับปรุงโมเดลให้มีประสิทธิภาพตามที่คาดหวังไว้แล้ว จะดำเนินการสร้างเว็บแอปพลิเคชันที่รองรับระบบนี้อย่างเต็มรูปแบบ และเปิดใช้งานในอนาคต

#### 5.2 ปัญหาที่เกิดขึ้น

เนื่องจากความกว้างขวางของฟิลด์ไอที ทำให้ในการดึงข้อมูลจำเป็นต้องมีความรอบคอบใน keyword ที่ใช้ในการค้นหาเหมือนนั้นจะทำให้การความคลาดเคลื่อนเป็นจำนวนมากกับโมเดล และในส่วนของโปรไฟล์นั้นอคติที่เกิดขึ้นกับโปรไฟล์แต่ละนั้นทำให้การคิดว่าดึงข้อมูลโดยอิงจากบริษัทที่ดึงและเป็นมาตรฐาน เป็นความคิดที่ผิดเนื่องจากผู้คนที่ได้ใส่ทักษะความสามารถครอบคลุมเกินฟิลด์ที่ตัวเองทำงาน ดังนั้นในอนาคตจึงคิดเปลี่ยนการดึงข้อมูลโปรไฟล์จากบริษัท เปลี่ยนเป็นดึงผ่านฟิลด์ไอทีแทนซึ่งคาดว่าจะสามารถทำให้การวัดประสิทธิภาพครอบคลุมในส่วนของโปรไฟล์ด้วย

## บรรณานุกรม

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

- [12] Margaret Rouse: Web application  
[https:// searchsoftwarequality.techtarget.com/ definition/ Web-application-Web-app](https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app)
- [13] Huizhi Liang: Real-time Collaborative Filtering Recommender Systems *Department of Computing and Information Systems* The University of Melbourne. 2005.
- [14] Philip Lenhart: Combining Content-based and Collaborative Filtering for Personalized Sports News Recommendations *Department of Informatics*. 2016
- [15] PyOhio Lenhart: “Large-Scale Recommendation System with Python and Spark  
<https://www.youtube.com/watch?v=oAByzl71Ak4>
- [16] 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)
- [17] Sung-Hwan Min: Recommender Systems Using Support Vector Machines *Graduate School of Management, Korea Advanced Institute of Science and Technology* 207-43 Cheongrangi-dong, Dongdaemun-gu, Seoul 130-722, Korea shmin@kgsm.kaist.ac.kr
- [18] 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)
- [19] Erion Çano Min: Hybrid Recommender Systems: A Systematic Literature Review *Charles University in Prague Prague, CZ, Czechia*
- [20] 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)
- [21] 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
- [22] Jorge Valverde-Rebaza: Job Recommendation based on Job Seeker Skills: An Empirical Study *Ricardo Puma's*