

รายงานความก้าวหน้าวิชา CE Project

ครั้งที่ 3

ระหว่างวันที่ 10 ก.ย. 65 ถึงวันที่ 23 ก.ย. 65

1. ชื่อโครงการ (อังกฤษ) Analytics and Prediction System for CE Curriculum administrators
2. การดำเนินงานมีความก้าวหน้า 28 % (ใช้ค่า **% Complete** จาก MS Project)

มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 12 %

☒ เร็วกว่าแผน11..... วัน ☐ ช้ากว่าแผน วัน

3. รายละเอียดความก้าวหน้า

นัดประชุมกับที่ปรึกษาจำนวน 1 ครั้ง

ครั้งที่ 1 : Discuss เรื่องการจัดการกับวิชาเลือก (GenED) โดยความคิดเห็นของผู้จัดทำคือ วิชา GenED นั้นไม่ได้จำเป็นในการนำมาเป็น parameter ในการ train แบบจำลอง Similarity เนื่องจากผลลัพธ์ที่โครงการนี้ต้องการที่จะนำเสนอ นั้นไม่ได้มีผลลัพธ์ที่ขึ้นตามกับวิชาหมวด GenED และในแง่ของด้านการพัฒนาแบบจำลองนั้นการนำวิชาหมวด GenED เข้ามานั้นอาจทำให้ค่า RMSE มีค่าที่สูงขึ้น FeedBack: จากที่ปรึกษา ที่ปรึกษาต้องการข้อมูลที่เป็นตัวเลขในเชิงคณิตศาสตร์เพื่อช่วยในการตัดสินใจ ต่อมาคือสิ่งที่ ที่ปรึกษาต้องการให้เพิ่มคือการใช้ NLP มาช่วยในเรื่องการจัดหมวดหมู่ของวิชาเพื่อลด workload ในการจัดกลุ่มแบบมือ และ เพื่อให้รองรับกับวิชาที่จะเข้ามาใหม่ในอนาคต ต่อมาคือการ Review การทำงานของ Surprise Model ที่ได้พัฒนาขึ้น FeedBack: ถือว่าเป็นที่น่าพอใจแล้วสามารถใช้งานเป็น Algorithm หลักได้เลย แต่การ Tuning Parameter ในการ train แบบจำลองนั้นยังต้องทดลองกับ parameter ที่อยู่ในรูปแบบ การจัดกลุ่มวิชาแบบ NLP

หัวข้อการพัฒนาโครงการตาม Gantt Chart

ศึกษาทฤษฎีที่เกี่ยวข้อง Complete 75 % (late: 23 วัน) หมายเหตุ พักไว้สำหรับเพื่อในอนาคตมีเรื่องที่จะต้องศึกษาเพิ่มเติม

- ด้านเทคโนโลยี

หลังจากที่ได้มีการ Discuss กับที่ปรึกษาเรื่องการจัดกลุ่มวิชาผ่าน NLP จึงได้ไปทำการศึกษาเพิ่มเติมจนได้ข้อสรุปเป็นการใช้ Lib ของ NLTK ร่วมกับ Stop word เพื่อเพิ่มความแม่นยำในการหา Similarity

เตรียม Server Complete 50 % (late: 16 วัน) หมายเหตุ จากที่ได้ Discuss กับที่ปรึกษาทำให้ต้องพักส่วนนี้ไปก่อนแล้วไปมุ่งเน้นกับการพัฒนาตัว Model Similarity

เตรียม Data สำหรับการพัฒนา Complete 72 %

ยังไม่ได้มีอะไรที่ทับหน้าไปจากเดิมเนื่องจากได้มีการปรับแผนการพัฒนาให้อยู่ในรูปการทำงานบน Google Colab ทำให้ยังไม่จำเป็นต้องใช้ Data ที่มาจาก Database แต่ใช้ Data จาก File โดยตรงแทน โดยจะใช้ File หลักอยู่ 2 file ด้วยกันคือ file สำหรับเกรด นศ และ file รายวิชา ซึ่ง file เกรดของ นศ (HashidComData.csv) นั้นจะเป็น file ที่จะนำเข้าไป train model similarity ส่วน file รายวิชา (2560-2564.csv) นั้นจะนำไปใช้ในการหา รูปแบบการจัดกลุ่มผ่าน NLP

```
student_id,subject_id,grade,semester,year,curriculum
f31df81d081367bb50462e95405acd57,01006028,C+,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01006030,D,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076001,C,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076002,B+,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90201001,C+,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90306003,B+,1,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01006031,F,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076003,D,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076004,D+,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076012,D,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90201002,C,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90402008,C+,2,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01006031,D+,3,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,03456227,A,3,2560,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076006,D+,1,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90108007,B,1,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90201026,B,1,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076005,D+,2,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90106002,B,2,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90306008,F,2,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90401013,B+,2,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,03456216,A,3,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,90201016,B,3,2561,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076008,C+,1,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076032,F,1,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076263,C+,1,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076013,C,2,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076014,B+,2,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076023,A,2,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076423,B+,2,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076582,C+,2,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01006004,S,3,2562,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076007,D,1,2563,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076011,C,1,2563,วิศวกรรมคอมพิวเตอร์
f31df81d081367bb50462e95405acd57,01076015,B,1,2563,วิศวกรรมคอมพิวเตอร์
```

รูปที่ 1 File Hashid_ComData.csv

```
subject_id,subject_name_thai,subject_name_eng,abstract,subject_key,subject_type,year,subject_class
01006030,แคลคูลัส 1,CALCULUS 1,"Function, Limit, Continuity and their applications, Mathematical induction, Introduction to deriva
01006031,แคลคูลัส 2,CALCULUS 2,"Functions of several variables and their applications, Vector algebra in three dimensions, Polar
01006032,สมการอนุพันธ์เชิงเส้นอันดับหนึ่งและสอง,ELEMENTARY DIFFERENTIAL EQUATIONS AND LINEAR ALGEBRA,"Systems of linear equations and solu
01006004,การรู้จำและรู้จำภาษา,INDUSTRIAL TRAINING,"During their four-year selected studies, students are required to complete a sho
01006028,เตรียมความพร้อมก่อนเข้าสู่วิศวกรรม,PRE-ENGINEER ACTIVITIES,Participates in activities organized by the Faculty of Engineering of adv
01076003,วงจรไฟฟ้าและอิเล็กทรอนิกส์,CIRCUITS AND ELECTRONICS,"Fundamentals electric circuit. Ohm's law, Kirchhoff's law, Thevenin's and
01076253,ความน่าจะเป็นและสถิติ,PROBABILITY AND STATISTICS,Introduction to probability and statistics; probability; random variables; d
01076002,พื้นฐานการเขียนโปรแกรมคอมพิวเตอร์,PROGRAMMING FUNDAMENTAL,"Introduction to fundamental concepts of procedural programming and
emphasis in problem solving. Topics include data types, control structures, functions,
arrays, files, pointer and the mechanics of running, testing, and debugging. ",ซอฟต์แวร์,2560,
01076001,วิศวกรรมคอมพิวเตอร์เบื้องต้น,INTRODUCTION TO COMPUTER ENGINEERING,"Introduction to computer Engineering roles --hardware and so
provide students with an understanding of the role computation can play in solving problems. It also
aims to help students to feel justifiably confident of their ability to hands-on experience, assemble and
test hardware projects that allow them to accomplish setting goals. This class will use the Arduino
microcontroller board to build hardware project.",ฮาร์ดแวร์,2560,
01076004,การพัฒนาโปรแกรมเชิงวัตถุ,OBJECT ORIENTED PROGRAMMING,"The design and implementation of software using object-oriented program
techniques that include objects, class, polymorphism, exception handling, inheritance and interfaces.
Students will analyze program specifications and identify appropriate objects and classes. Additional
programming topics include simple data structure, simple searching and sorting, graphical user interfaces
and UML modeling of small systems.",ซอฟต์แวร์,2560,
01076005,โครงสร้างข้อมูลและอัลกอริทึม,DIGITAL SYSTEM FUNDAMENTALS,"Fundamental of solving problems using data structures including link
stacks, queues, hash tables, and graphs. Algorithms for sorting, searching, and other fundamental
operations. Introduction to foundations for analysis of iterative and recursive algorithms. Implementation
of selected algorithms using object-oriented paradigm.",ซอฟต์แวร์,2560,
01076006,พื้นฐานระบบดิจิทัล,DIGITAL SYSTEM FUNDAMENTALS,"Introduction to Digital Systems; Binary Systems; Boolean Algebra and Simplif
Combinational Circuit; Sequential Components i.e. latches, flip-flops, registers and counters; Sequential
Circuits; Basic ALU (Arithmetic and Logic Unit) and Control Unit; Hardware Description Language;",ฮาร์ดแวร์,2560,
01076007,การสื่อสารข้อมูล,DIGITAL COMMUNICATIONS,"Basic concepts of data communication; OSI model; signal characteristics; encoding and
modulating; transmission of analog and digital data; communication interface device; transmission
media; data multiplexing; high speed digital access; error detection and correction; data link flow control;
data link protocols and Ethernet networks.",เน็ตเวิร์ก,2560,
01076008,กระบวนการพัฒนาซอฟต์แวร์,SOFTWARE DEVELOPMENT PROCESSES,"Introduction to modern software development principles and practice"
```

รูปที่ 2 File 2560-2564.csv

พัฒนา Function NLP สำหรับจับกลุ่มวิชา 66%

หลังจากทำการศึกษารียบร้อยแล้วจึงได้ข้อสรุปว่าจะใช้ NLTK ซึ่งจากคำแนะนำของที่ปรึกษาซึ่งแนะนำให้
นำเอา algorithm ในการตัด generic word หรือ stop word ออกไปเพื่อเพิ่มความแม่นยำ

โดยการนำหลักการ NLP มาใช้งานนั้นเพื่อให้รองรับต่อการใช้งานของระบบในอนาคตในกรณีที่มิมีวิชาใหม่
เข้ามาโดยจะทำการจัดกลุ่มของวิชาผ่านการหาค่าของ similarity ของบทคัดย่อของวิชาแล้วจัดกลุ่มวิชาที่มีค่า similarity
ใกล้เคียงกันจะอยู่ในกลุ่มเดียวกัน

โดยปัจจุบันนั้นได้ทำการจัดกลุ่มวิชาบังคับและวิชาเลือกของคณะวิศวกรรมคอมพิวเตอร์ผ่าน algorithm นี้แล้ว
เป็นไปด้วยดี โดยการทำงานทั้งหมดของ Function นี้จะมีดังนี้

ทำการ import ไฟล์ CSV ทำการจัดรูป rows โดยการ drop rows ที่มีรหัสวิชาที่ซ้ำออก โดยเก็บแบบแรกไว้ นำ
abstract มาใช้ในการหาค่า NLP อันดับแรก ทำการจัดรูปแบบประโยคโดยการตัดคำเชื่อมออก ให้เหลือเป็นคำเฉพาะ จัด
กลุ่มค่า Similarity ของ NLP ที่มีค่าใกล้เคียงกันมากกว่า 0.90 ให้อยู่ในกลุ่มเดียวกัน เพื่อทำการแยก class วิชา ประกาศ
Dict 1 ตัวเพื่อมาเก็บค่ากลุ่มของวิชาที่ใกล้เคียงกัน ทำการเทียบค่าความใกล้เคียงแบบวนลูป โดยหาค่าใกล้เคียงที่มาก
ที่สุด แล้วเก็บเข้าไปใน Value ของ Key ตัวนั้นๆ

ผลลัพธ์

- 1.สามารถนำไปใช้ในการหา recommendation ของเกรดได้แม่นยำขึ้น
- 2.จับ class วิชาตัวเดียวกันถึงแม้ว่าจะมีการเปลี่ยนเลขรหัสวิชา ลดปัญหาการนำข้อมูลระหว่างหลักสูตรมาใช้

```

- NLP

[ ] pip install --user -U nltk

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.7)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from nltk) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.7/dist-packages (from nltk) (2022.6.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from nltk) (4.64.1)

[ ] import os
import gensim
import spacy
import en_core_web_sm
import nltk
import math
nltk.download('stopwords')
from nltk.corpus import stopwords
STOPWORDS = set(stopwords.words('english'))

nlp = en_core_web_sm.load()

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

df_subject_pre_nlp = pd.read_csv('2560+2564.csv',dtype={0:'string'},encoding='TIS-620')
df_subject_pre_nlp = df_subject_pre_nlp.drop_duplicates('subject_id', keep='first')
df_subject_pre_nlp = df_subject_pre_nlp.reset_index()
df_subject_pre_nlp

```

รูปที่ 3 Import lib NLP (NLTK) และดึง Data (2560-2564.csv) มาใช้งาน

Add Subject Group Into Rating Data

```
def findSubjectType(subject_id):
    lisValue = list(thisdict.values())
    lisKeys = list(thisdict.keys())
    subTypes = "อื่นๆ"
    for i in lisKeys:
        if int(subject_id) in thisdict[i]:
            subTypes = str(i)
    return subTypes

# def addSubjectTypeDFS(dfs, traindfs):
#     for i in dfs:
#         for indexj, rowj in

for index, row in df_for_Com_Train.iterrows():
    subId = row['subject_id']
    df_for_Com_Train.at[index, 'subjectTypes'] = findSubjectType(subId)

for index, row in df_for_Com.iterrows():
    subId = row['subject_id']
    df_for_Com.at[index, 'subjectTypes'] = findSubjectType(subId)

df_for_Com.head(50)
```

รูปที่ 4 นำเอา Subject Type ที่หามาได้มาเพิ่มเป็น Column ให้กับ Dataframe นักศึกษา

พัฒนา Similarity Model 59 %

หลังการทดลองใช้งาน Model ของ Lib Surprise ได้นำมาปรับใช้กับ Data จริงคือ Data เกรดของนักศึกษา (HashidComData.csv) และ Data รายละเอียดของรายวิชา (2560-2564.csv) โดยสิ่งที่ Lib นี้ต้องการคือ Data การให้ Rating ของ User กับสินค้าต่างๆในระบบซึ่งเราจะมอง เกรดของนักศึกษาคือค่า Rating และตัวสินค้าในระบบคือ ตัวรายวิชา โดยหลักการนี้คือการทำ Recommendation User Base โดยเราจะใช้ประโยชน์จากการที่ Model Prediction เกรดของนักศึกษากับวิชาที่ไม่ได้ลงมาเป็นการ Fill Column ที่ยังไม่ได้เติมของเกรด (นักศึกษามบางคนไม่ได้ลงเรียนบางวิชาทำให้ซึ่งอาจทำให้การ Prediction หรือการ Train Model สำหรับการหาอาชีพนั้นอาจเกิดข้อผิดพลาดได้)

โดยปัจจุบันได้ทำการพัฒนาตัว Prototype ของ Function นี้เสร็จเป็นที่เรียบร้อยแล้ว โดยผลลัพธ์ออกมาเป็นที่น่าสนใจทั้งที่ปริกษาและผู้จัดทำโดย RMSE อยู่ที่ 0.534 โดยที่ n_factor = 20 และ n_epochs = 20 ซึ่งการทำงานมีดังนี้

เริ่มด้วยใช้ pandas อ่าน CSV ที่เข้ามาหลังจากนั้นจะใช้ pandasql ในการ query มาเฉพาะ record ที่เป็นวิศวกรรมคอมพิวเตอร์ปกติและต่อเนื่องโดยจะเลือกมาแค่เฉพาะวิชาที่ไม่ใช่ GenED ต่อมาจะทำการแยก Dataframe ออกเป็น 2 ส่วนคือ Com ปกติ และ ต่อเนื่อง หลังจากนั้นจะนำมา Transform ให้เกรดอยู่ในรูปแบบของตัวเลข และทำการอ่าน csv ของรายละเอียดวิชา (2560-2564.csv) แล้วเข้ากระบวนการ NLP แล้วจะนำหมวดหมู่วิชามาเพิ่มเป็น Column ใน Dataframe ที่จะนำมา train model หลักจากนั้นจะนำ Dataframe เข้าไป train ด้วยการเลือก algorithm เป็น SVD และทำการ Tune ค่า n_factors และ n_epochs ให้ได้ค่า RMSE ที่ดีที่สุดแล้วจะนำ Model ไป Prediction Missing Value ใน Dataframe

Surprise

```
[ ] from surprise import Dataset
    from surprise import Reader
    from surprise import SVD
    from surprise.model_selection import cross_validate
    from surprise.model_selection import train_test_split
    from surprise.model_selection import GridSearchCV
    import numpy as np
```

Find Grade AVG For SubTypes.

```
q_find_AVG = "SELECT student_id, AVG(grade) as grade, semester, year, curriculum, subjectTypes FROM df_for_Com_Train GROUP BY subjectTypes, student_id ORDER BY student_id"
df_for_Com_Train_AVG = sqldf(q_find_AVG)
```

รูปที่ 5 Import Lib Surprise และ query ค่าเฉลี่ยของเกรดของนักศึกษาในวิชาแต่ละ Type มาเป็น Dataframe

Find User And Their Grade.

```
q_u_grade = "SELECT * FROM df_for_Com_Train_AVG WHERE student_id = 'f31df81d081367bb50462e95405acd57'"
df_user_grade = sqldf(q_u_grade)
```

รูปที่ 6 ตรวจสอบความถูกต้องของข้อมูลโดยการ query มา 1 User

```
[ ] min_rating = 0
    max_rating = 4

    reader = Reader(rating_scale=(min_rating, max_rating))
    data = Dataset.load_from_df(df_for_Com_Train_AVG[['student_id', 'subjectTypes', 'grade']], reader)

    svd = SVD(n_epochs=10)
    results = cross_validate(svd, data, measures=['RMSE', 'MAE'], cv=10, verbose=True)
```

Evaluating RMSE, MAE of algorithm SVD on 10 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
RMSE (testset)	0.5315	0.5705	0.5485	0.5827	0.5475	0.5598	0.5660	0.5389	0.5692	0.5865	0.5601	0.0173
MAE (testset)	0.3910	0.4367	0.4103	0.4378	0.4171	0.4107	0.4192	0.4031	0.4303	0.4323	0.4188	0.0147
Fit time	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.11	0.11	0.10	0.00
Test time	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00

รูปที่ 7 เลือก Algorithm สำหรับสร้าง Model และตรวจ ค่า RMSE และ MAE

```
[ ] param_grid = {
    'n_factors': [20, 50, 100],
    'n_epochs': [5, 10, 20]
}

gs = GridSearchCV(SVD, param_grid, measures=['rmse', 'mae'], cv=10)
gs.fit(data)

print(gs.best_score['rmse'])
print(gs.best_params['rmse'])

0.5340764094678272
{'n_factors': 20, 'n_epochs': 20}

[ ] best_factor = gs.best_params['rmse']['n_factors']
    best_epoch = gs.best_params['rmse']['n_epochs']
    trainset, testset = train_test_split(data, test_size=.20)

[ ] svd = SVD(n_factors=best_factor, n_epochs=best_epoch)
    svd.fit(trainset)

<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7f2f11edf590>
```

รูปที่ 8 เลือกค่า n_factors และ n_epochs ที่ดีที่สุดสำหรับการสร้าง Model

```
[ ] def nan_grade_sub(student):
    df_onlyNull = df_for_Com.loc[df_for_Com['student_id'] == student]
    df_onlyNull = df_onlyNull[df_onlyNull['grade'].isna()]
    lis = df_onlyNull['subject_id'].tolist()
    return lis

def nan_grade_sub_Type(student):
    df_onlyNull = df_for_Com.loc[df_for_Com['student_id'] == student]
    df_onlyNull = df_onlyNull[df_onlyNull['grade'].isna()]
    lis = df_onlyNull['subjectTypes'].unique().tolist()
    return lis

def generate_recommendation(model, student_id, df, subject, n_items):
    # Get a list of all movie IDs from dataset
    subject_ids = df["subject_id"].unique()
    targetSub = nan_grade_sub(student_id)
    # Get a list of all movie IDs that have been watched by user
    subject_ids_student = df.loc[df["student_id"] == student_id, "subject_id"]
    # Get a list off all movie IDS that that have not been watched by user
    subject_ids_to_pred = np.setdiff1d(subject_ids, subject_ids_student)

    # Apply a rating of 4 to all interactions (only to match the Surprise dataset format)
    test_set = [[student_id, subject_id, 4] for subject_id in subject_ids_to_pred]

    # Predict the ratings and generate recommendations
    predictions = model.test(test_set)
    pred_ratings = np.array([pred.est for pred in predictions])
    print("Top {0} item recommendations for student {1}:".format(n_items, student_id))
    # Rank top-n movies based on the predicted ratings
    index_max = (-pred_ratings).argsort()[:n_items]
    for i in index_max:
        subject_id = subject_ids_to_pred[i]
        if subject_id in targetSub:
            print(subject_id, pred_ratings[i])
```

รูปที่ 9 นำ Model ที่สร้างมาใช้งานเป็น Function

```
# define which user ID that we want to give recommendation
userID = 'f31df81d081367bb50462e95405acd57'
# define how many top-n movies that we want to recommend
n_items = NumOfSub
# generate recommendation using the model that we have trained
generate_recommendation_for_Types(svd,userID,df_for_Com_Train_AVG,df_SubjectData,n_items)

subject_ids_student ['0', '1', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '2', '21', '22', '3', '4', '46', '5', '6', '7', '8', '9', 'Not Found']
subject_ids_to_pred ['20', '24', '29', '32', '38', '48']
Top 49 item recommendations for student f31df81d081367bb50462e95405acd57:
38 2.7599125682988053
24 2.657175760366085
32 2.492031826535238
48 2.4866372026648085
29 2.3883020586129566
20 2.017986129318273
```

รูปที่ 10 ทดสอบการใช้งานของ Function

4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาจากครั้งก่อน

1 Docker Image Design

ปัจจุบันได้มีการข้ามเรื่องนี้ไปก่อนแล้ว Focus กับงานส่วนของการพัฒนา Model แทน

2 Data Transform

ปัจจุบันในส่วนของ Data ที่ใช้งานได้แก้ไขให้เกรด S มีค่าเทียบเท่ากับ 4 และเกรด U เท่ากับ 0 ปัญหาตรงนี้จึงหมดไป

ปัญหา ณ ปัจจุบัน

1 NLP ในเรื่อง Similarity

จากการทดลองใช้งานจริงพบว่าการจับกลุ่มแบบนี้ทำให้บางวิชาที่ควรอยู่หมวดเดียวกันดันอยู่คนละหมวดเช่น Cal 1 และ Cal 2 นั้นจะอยู่กันคนละหมวดกันถ้าปรับการจัดหมวดหมู่อยู่ที่ มากกว่า 90% แต่ถ้าปรับอยู่ที่ 80% จะจัดหมวดหมู่กันได้

2 การเพิ่ม column subject type ให้ Dataframe นักศึกษา

เนื่องด้วยปัญหานี้เกิดจากที่ไว้ในข้อมูลของนักศึกษานั้นมีวิชาที่ในข้อมูลของรายละเอียดวิชาไม่มีอยู่ทำให้เมื่อนำ column ของ subject types มาเพิ่มใน Dataframe ของที่จะ train model นั้นเกิดเป็น Not Found Subject Type โดยปัจจุบันแก้ปัญหาดังกล่าวโดยการใส่เป็นหมวดอื่นๆไปก่อน

5. สิ่งที่จะดำเนินการต่อไป

- พัฒนา Function Model Similarity ให้อยู่ในรูปแบบ Optimized โดยจะทำให้เมื่อข้อมูลเข้ามาจะทำการแยก Dataframe ตามหลักสูตร และเก็บ Dataframe ให้อยู่ในรูปแบบของ Dict และ ทำการ Train Model แยกกันไปตามหลักสูตร
- Tuning NLP Function ปรับให้ค่าการจัดกลุ่มดูเหมาะสมมากขึ้น และทำให้ตัว Function Optimized มากขึ้นเพื่อการใช้งานจริง