รายงานความก้าวหน้าวิชา CE Project

ครั้งที่ 2

ระหว่างวันที่ 28 ส.ค. 65 ถึงวันที่ 09 ก.ย. 64

1. ชื่อโครงงาน (อังกฤษ) Analytics and Prediction System for CE Curriculum administrators

2. การดำเนินงานมีความก้าวหน้า 16 % (ใช้ค่า **% Complete** จาก MS Project)

   มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 9 %

   ☐ เร็วกว่าแผน ..... 3 ..... วัน ☐ ช้ากว่าแผน ..... วัน

3. รายละเอียดความก้าวหน้า

   นัดประชุมกับที่ปรึกษาจำนวน 1 ครั้ง

   ครั้งที่ 1 : หัวข้อการประชุม 2 หัวข้อ 1 Review Diagram แผนผังรวมของระบบ โดยมี Feedback : ต้องแก้ไขบางคำให้สื่อความหมายให้ชัดเจนมากขึ้น เช่น ข้อมูล file CSV ไปเป็น ข้อมูลเกรดของนักศึกษา แล้วปรับแก้ให้สอดคล้องกับ Usecase Diagram มากยิ่งขึ้น เช่นอาจต้องมีกล่องเป็น Background เพื่อครอบรวม usecase ว่าอยู่ใน Functional ไหนในระบบหลักบ้าง

   2 ชี้แนะการดำเนินงานต่อไป : รวบรวมข้อมูลทั้งจากที่ ที่ปรึกษาจัดเตรียมให้ และ จัดเตรียมเองขึ้นไปทดลองใน Google Collab เพื่อปั่นเป็น Schema ตามที่ได้ Design ไว้ หลักจากนั้นจะเริ่มทำการ Data Processing ทั้ง Clean และ Transform เพื่อให้พร้อมต่อการใช้งาน แล้ว นำไปทดลองต่อตามที่ได้วางแผนไว้ใน Gantt Chart

   หัวข้อการพัฒนาโครงงานตาม Gantt Chart

   ศึกษาทฤษฎีที่เกี่ยวข้อง Complete 75 % (late: 9 วัน)

   - ด้านเทคโนโลยี
     หลังจากสืบค้นข้อมูลได้พบ library ตัวนึงของ python ที่หน้าสนใจคือ Surprise ซึ่งเป็น lib ที่อยู่ใน subset ของ Scikit learn อีกที ซึ่งถูกออกแบบมาเพื่อทำ Recommendation System โดยเฉพาะซึ่งสามารถเลือก Algorithm ได้หลากหลาย ซึ่งเหมาะกับตัวโครงงานเป็นอย่างมากเลยเลือกที่จะใช้ lib ตัวนี้เป็น lib หลัก
   - ด้านงานวิจัยที่เกี่ยวข้อง
     ได้ทำการ Review แล้วทั้ง 2 งานวิจัยสรุปได้ว่า
     1 สุเมธ ดาราพิสุท นำเสนองานวิจัยเรื่อง การสร้างรายการเพลงโดยใช้การกรองร่วมแบบเซสชั่นที่เพิ่มขึ้นด้วยกลไกการลืมและการวิเคราะห์สถิติเชิงมุม โดยใช้ 2 วิธีร่วมกัน 1 การสร้างรายการเพลงจะพิจารณาการฟังเพลงในเซสชั่น

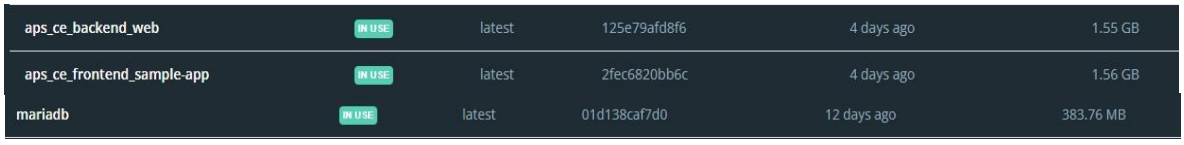
ปัจจุบันที่คล้ายกับเซสชั่นในอดีตของผู้ฟัง 2 สร้างรายการเพลงแนะนำโดยพิจารณาช่วงเวลา เฉพาะในการฟังเพลง ซึ่งแตกต่างจากช่วงเวลาอื่นอย่างมีนัยสำคัญทางสถิติในรอบวันของผู้ฟังโดยใช้ การวิเคราะห์สถิติเชิงมุม และวัดประสิทธิภาพโดย : ประสิทธิภาพ HitRatio และ Precision จากการทดลองพบว่าการใช้ 2 วิธีแยกกันนั้น ได้ผลลัพท์ที่น้อยกว่านำมาใช้ร่วมกัน 0.18-0.22 %

2 นิภาภรณ์ พันธ์นาม นำเสนองานวิจัย ระบบแนะนำสินค้าอาหารโดยใช้ระบบแนะนำแบบผสมผสาน ใช้เทคนิค Content based filtering แบบหลักการ Cosine และสร้างแบบจำลองโดยใช้ lib Surprise ซึ่งมีอัลกอริทึม SVD, NMF, Baseline และ KNN และวัดประสิทธิภาพโดย RMSE, MAE จากการทดลองพบว่า 1 เทคนิคการกรองแบบอิงเนื้อหานำวิธีการ TF-IDF เข้ามาช่วยในการทำ Vectorization ส่วนใหญ่ค่าความเหมือนออกมาค่อนข้างที่จะต่ำเนื่องมาจากข้อมูลที่น้อยเกินไป 2 เทคนิคการกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วม ผ่าน library Surprise ของ Scikitlearn ซึ่งโมเดลที่มีผลคะแนนโดยรวมดีที่สุดคืออัลกอริทึมของ SVD ซึ่ง ได้ค่า RMSE 1.2528 และ MAE 0.9376 และ 3 ระบบแนะนำแบบผสมผสาน โดยผลลัพท์นั้นจะไม่ชัดเจนเนื่องจากวิธีนี้ไม่ได้ เนื่องจากกรณีนี้ได้มีการทำนายค่า Rating ซึ่งวิธีการของระบบแนะนำแบบผสมผสานนั้น ได้มีนำเทคนิคการกรองแบบอิงเนื้อหาที่ไม่ได้มีการทำนายค่าอะไรมารวมในการทำงานของแบบจำลองด้วย ซึ่งถ้าต้องการวัดผลลัพท์สามารถอ้างอิงจากค่า RMSE, MAE ได้

เตรียม Docker Complete 100% (early: 3 วัน)

โดยจากครั้งก่อนที่มีการรวมทุกอย่างไว้ใน Image เดียวนั้นได้เปลี่ยนมาเป็นการแยกส่วนของ Image ออกมาเป็นส่วนของ Backend ที่ใช้ Framework ของ Django และ มี Image ของ Database Server เป็น SQL โดยใช้ MariaDB และ Image ของ Frontend ที่ใช้เป็น React

ซึ่งหลังจากวางโครงสร้างและสรุปการทำงานกันจะได้ว่าจะนำ Image ของ Backend และ Image ของ Database Compose ขึ้นไปเป็น Container พร้อมกัน โดยจะเปิด Port สำหรับ Development อยู่ที่ 8000 สำหรับ Backend และ 3306 สำหรับ Database Server และในส่วนของ Image Frontemd นั้นได้ทำเป็น Compose เดี่ยวขึ้นไปและทำการเปิด Port สำหรับ Development อยู่ที่ 3000



เตรียม Server Complete 50 % (late: 1 วัน)

จากการเปลี่ยนแผนใหม่ทำให้ยังไม่มีอะไรคืบหน้านอกจากทดลองนำ Docker Compose ไปสร้าง Container บน Server

เตรียม Data สำหรับการพัฒนา Complete 72 %

หลังจากได้ Data มาจากที่ปรึกษาและที่หามาเองได้เอา Data มา Clean เช่นใน Data เกรดและวิชาของนักศึกษา column year นั้นไม่ได้เป็นประโยชน์จึง Drop ทิ้งไปและได้ Transform โดยใน Column Grade ได้เปลี่ยนจากระบบ Char (A, B+, B, …, F) ไปเป็น Int(4, 3.5, 3, …, 0) และได้ให้เปลี่ยนเกรด S ให้มีค่าเท่ากับ 4 และ U เท่ากับ 0 หลังจากนั้นได้ใช้ pandasql ในการ query จากตารางหลักออกมาเป็นตารางย่อยที่จะนำไปใช้ประโยชน์ต่อไป

Drop Curriculum that not Computer Engineer and Computer Engineer Continue

```
[ ] df = df.drop(df[(df['curriculum'] != 'Computer Engineer') & (df['curriculum'] != 'Computer Engineer Continue')].index)
    df
```

	student_id	subject_id	grade	semester	year	curriculum
0	f31df81d081367bb50462e95405acd57	1006028	C+	1	2560	Computer Engineer
1	f31df81d081367bb50462e95405acd57	1006030	D	1	2560	Computer Engineer
2	f31df81d081367bb50462e95405acd57	1076001	C	1	2560	Computer Engineer
3	f31df81d081367bb50462e95405acd57	1076002	B+	1	2560	Computer Engineer
4	f31df81d081367bb50462e95405acd57	90201001	C+	1	2560	Computer Engineer
...	...	...	...	...	...	...
12505	8b6c9ad43a4fd390abdebfb034b4b330	1076112	NaN	1	2564	Computer Engineer Continue
12506	8b6c9ad43a4fd390abdebfb034b4b330	1076118	NaN	1	2564	Computer Engineer Continue
12507	8b6c9ad43a4fd390abdebfb034b4b330	90641001	NaN	1	2564	Computer Engineer Continue
12508	8b6c9ad43a4fd390abdebfb034b4b330	90641003	NaN	1	2564	Computer Engineer Continue
12509	8b6c9ad43a4fd390abdebfb034b4b330	90644008	NaN	1	2564	Computer Engineer Continue

12510 rows × 6 columns

Transfrom Grade Char() to Float()

```
df.loc[df["grade"] == "A", "grade"] = 4
df.loc[df["grade"] == "S", "grade"] = 4
df.loc[df["grade"] == "T(A)", "grade"] = 4

df.loc[df["grade"] == "B+", "grade"] = 3.5
df.loc[df["grade"] == "T(B+)", "grade"] = 3.5

df.loc[df["grade"] == "B+", "grade"] = 3.5
df.loc[df["grade"] == "T(B+)", "grade"] = 3.5

df.loc[df["grade"] == "B", "grade"] = 3
df.loc[df["grade"] == "T(B)", "grade"] = 3

df.loc[df["grade"] == "C+", "grade"] = 2.5
df.loc[df["grade"] == "T(C+)", "grade"] = 2.5

df.loc[df["grade"] == "C", "grade"] = 2
df.loc[df["grade"] == "T(C)", "grade"] = 2

df.loc[df["grade"] == "D+", "grade"] = 1.5
df.loc[df["grade"] == "T(D+)", "grade"] = 1.5

df.loc[df["grade"] == "D", "grade"] = 1
df.loc[df["grade"] == "T(D)", "grade"] = 1

df.loc[df["grade"] == "F", "grade"] = 0
df.loc[df["grade"] == "T(F)", "grade"] = 0
df.loc[df["grade"] == "U", "grade"] = 0


df['grade'] = df['grade'].fillna(0)
df
```

Create query of student data

```
q_student = "SELECT DISTINCT(student_id), curriculum, delflag FROM df;"
df_student = sqldf(q_student)
df_student
```

	student_id	curriculum	delflag
0	f31df81d081367bb50462e95405acd57	Computer Engineer	0
1	48c159db28a39eb366ac8ec31b975fb8	Computer Engineer	0
2	63ff5b2e627f0162eb706e16098a6001	Computer Engineer	0
3	025f749d7a5d9b5c3f3d57b68e1de9e9	Computer Engineer	0
4	2a408033cc779781e686fb63ed6b8ce2	Computer Engineer	0
...	...	...	...
598	10a9ed59eced4212d4cf8ecb61e5a83d	Computer Engineer Continue	0
599	58bef7a3a232e045c2fdfd7b8060fcec	Computer Engineer Continue	0
600	ca026e043902fbe88a85696e9ae4152a	Computer Engineer Continue	0
601	15ccc2e8ddb2d6c32a7d065d00643fa9	Computer Engineer Continue	0
602	8b6c9ad43a4fd390abdebfb034b4b330	Computer Engineer Continue	0

603 rows × 3 columns

Create query of subject data

```
q_subject = "SELECT subject_id, student_id, grade, semester, case when subject_id LIKE '90%' then 1 else 0 end as delflag FROM df order by subject_id;"
df_subject = sqldf(q_subject)
df_subject
```

	subject_id	student_id	grade	semester	delflag
0	1006001	0197dc3d32f1d32bbff2a3bff89e69f9	4.0	3	0
1	1006004	f31df81d081367bb50462e95405acd57	4.0	3	0
2	1006004	48c159db28a39eb366ac8ec31b975fb8	4.0	3	0
3	1006004	63ff5b2e627f0162eb706e16098a6001	4.0	3	0
4	1006004	025f749d7a5d9b5c3f3d57b68e1de9e9	4.0	3	0
...	...	...	...	...	...
12505	90644014	28bc2819925b694bbe8d1bb739eb9ba4	0.0	1	1
12506	90644014	891a12be0266ea9a88733c30a42a3d2c	0.0	1	1
12507	90644014	7d40d9021507c0ca14eb4beddccfa397	0.0	1	1
12508	90644014	0db42e9bb839d237a396f3991bf5fd56	0.0	1	1
12509	90644014	15ccc2e8ddb2d6c32a7d065d00643fa9	0.0	1	1

12510 rows × 5 columns

Load df to csv

```
df_student.to_csv('student_data.csv',index=False)
df_subject.to_csv('subject_data.csv', index=False)
```

4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาจากครั้งก่อน

1 Task และ Workload

ปัจจุบันทั้งงานและ workload นั้นเป็นไปตามที่เหมาะสมเรียบร้อยแล้ว

2 Data Processing

ปัจจุบันได้มีการวางแผนและแตก task ที่ต้องทำออกมาอย่างมีระบบและขั้นตอนแล้ว

ปัญหา ณ ปัจจุบัน

1 Docker Image Design

เนื่องจากตอนแรกได้ Design ให้ Docker Image นั้นรวบร่วมนำสิ่งที่ต้องใช้งานไว้ใน Image เดียวกันทำให้เกิดความสับสนในการวาง port การติดต่อและ development code ได้ยาก จึงได้แก้ปัญหาโดยการแยกนำ Backend และ Database และ Frontend ออกมาเป็นอย่างละ Image ซึ่งได้ทำการ Compose รวม Backend และ Database สร้าง Compose Container ขึ้นมาเพื่อให้ Django setup SQL Database Server ได้ง่ายยิ่งขึ้น และแยก Frontend เป็น Compose Container เนื่องจากการ Run เป็น Compose up นั้นจะทำให้ผู้จัดทำ ทำงานได้ถนัดกว่าเป็น Image ซึ่งจะยังไม่ได้ทดลองติดต่อกับ Database Server ผ่านตัว Backend ทำให้ยังไม่ทราบว่าจะประสบกับปัญหาใดบ้าง

2 Data Transform

เนื่องจากตัว Algorithm ส่วนมากใน Recommendation ของ Surprise นั้นจำเป็นต้องใช้ข้อมูล Rating ที่เป็นตัวเลขจึงต้องแก้ไข Grade ของนักศึกษาที่เป็นรูปแบบ Char (A, B+, B, …, F) ไปเป็น Int(4, 3.5, 3, …, 0) ซึ่งปัญหาคือไม่สามารถทราบได้ว่าจะนำเกรด S, U ไปเทียบกับเลขใด โดยปัจจุบันแก้ปัญหาโดยการ เปลี่ยนเกรด S ให้มีค่าเท่ากับ 4 และ U เท่ากับ 0

5. สิ่งที่จะดำเนินการต่อไป

- ทำการศึกษาทฤษฎีที่เกี่ยวข้องให้ครบเพื่อนำไปประกอบกับรายงานหลัก
- ทำการ Map, Transform, Clean ข้อมูลให้ครบตาม Design ที่ได้ทำไว้ ผ่าน Google Collab
- ทดลองนำข้อมูลที่ได้เตรียมมาใช้กับ library Surprise ของ Scikit learn เพื่อเลือก algorithm ที่ดีที่สุด