# รายงานความก้าวหน้าวิชา CE Project

**ครั้งที่ 4**

**ระหว่างวันที่ 24 ก.ย. 65 ถึงวันที่ 21 ต.ค. 65**

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

## การดำเนินงานมีความก้าวหน้า 37 % (ใช้ค่า **% Complete** จาก MS Project)

มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 9 %

□ เร็วกว่าแผน วัน □ ช้ากว่าแผน 0 วัน

## **รายละเอียดความก้าวหน้า**

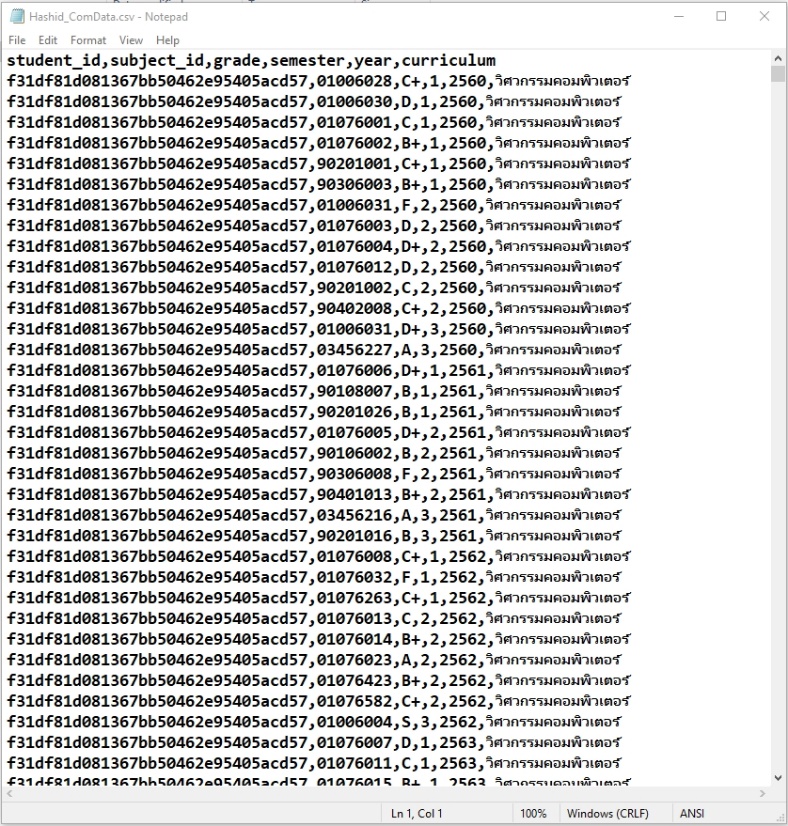
หัวข้อการพัฒนาโครงงานตาม Gantt Chart

**ศึกษาทฤษฎีที่เกี่ยวข้อง Complete 75 % (late: 51 วัน)** หมายเหตุ พักไว้สำหรับเพื่อในอณนาคตมีเรื่องที่จะต้องศึกษาเพิ่มเติม

**เตรียม Server Complete 50 % (late: 45 วัน)** หมายเหตุ จากที่ได้ Discuss กับที่ปรึกษาทำให้ต้องพักส่วนนี้ไปก่อนแล้วไปมุ่งเน้นกับการพัฒนาตัว Model Similarity

**เตรียม Data สำหรับการพัฒนา Complete 78 % (late: 30 วัน)** หมายเหตุเนื่องด้วยจำเป็นต้องมี data ที่เป็นแบบสอบถามการทำงานของนักศึกษาหลังจากเรียนจบ ซึ่งต้องใช้เวลาในการรอการตอบกลับของแบบสอบถาม

**พัฒนา Function NLP สำหรับจับกลุ่มวิชา Complete 100 %** หลังจากที่ผ่านการ Tuning model มาแล้ว และจากการแบ่งข้อมูลแบบ K-Fold Cross Validation โดยแบ่งเป็น 10 Fold โดยค่า Mean ของ Fold ทั้งหมดอยู่ที่ RMSE = 0.5646 และ MAE = 0.4309 ในกรณีของ Data ที่เป็นของนักศึกษาภาคปกติ และค่า Mean ของ Fold ทั้งหมดในกรณีของนักศึกษาภาคต่อเนื่อง RMSE = 0.7816 และค่า MAE = 0.6154 โดย Algorithm ที่จะใช้ในการสร้าง Model นี้คือ SVD (Singular Value Decomposition) หลังจากนั้นได้นำ code ที่ทดลองใน Google Collab ไปลงบน Backend Function ของ Demo Website ที่เตรียมไว้และทดสอบอีกครั้ง โดยค่า Mean ของ Fold ทั้งหมดอยู่ที่ RMSE = 0.5683 และ MAE = 0.4303 ในกรณีของ Data ที่เป็นของนักศึกษาภาคปกติ และค่า Mean ของ Fold ทั้งหมดในกรณีของนักศึกษาภาคต่อเนื่อง RMSE = 0.8070 และค่า MAE = 0.6388 แล้วได้ทำการ Implement ให้อยู่ในรูปแบบของการขอ Request prediction เกรดของ User โดย

ยังไม่ได้มีอะไรที่คืบหน้าไปจากเดิมเนื่องจากได้มีการปรับแผนการพัฒนาให้อยู่ในรูปการทำงานบน Google Colab ทำให้ยังไม่จำเป็นต้องใช้ Data ที่มาจาก Database แต่ใช่ 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

A page of a book

Description automatically generated with low confidence  *รูปที่ 1 File Hashid\_ComData.csv*

*รูปที่ 2 File 2560-2564.csv*

**พัฒนา Function NLP สำหรับจับกลุ่มวิชา 66%**

หลังจากทำการศึกษารียบร้อยแล้วจึงได้ข้อสรุปว่าจะใช้ NLTK ซึ่งจากคำแนะนำของที่ปรึกษาซึ่งแนะนำให้นำเอา algorithm ในการตัด generic word หรือ stop word ออกไปเพื่อเพิ่มความแม่นยำ

โดยการนำหลักการ NLP มาใช้งานนั้นเพื่อให้รองรับต่อการใช้งานของระบบในอณนาคตในกรณีที่มีวิชาใหม่เข้ามาโดยจะทำการจัดกลุ่มของวิชาผ่านการหาค่าของ similarity ของบทคัดย่อของวิชาแล้วจัดกลุ่มวิชาที่มีค่า similarity ใกล้เคียงกันในจะอยู่ในกลุ่มเดียวกัน

โดยปัจจุบันนั้นได้ทำการจัดกลุ่มวิชาบังคับและวิชาเลือกของคณะวิศวกรรมคอมพิวเตอร์ผ่าน algorithm นี้แล้วเป็นไปได้ด้วยดี โดยการทำงานทั้งหมดของ Function นี้จะมีดังนี้

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

ผลลัพธ์

1.สามารถนำไปใช้ในการหา recommendation ของเกรดได้แม่นยำขึ้น

Text

Description automatically generated2.จับ class วิชาตัวเดียวกันถึงแม้ว่าจะมีการเปลี่ยนเลขรหัสวิชา ลดปัญหาการนำข้อมูลระหว่างหลัดสูตรมาใช้

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

**Text

Description automatically generated***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 ที่ยังไม่เต็มของเกรด (นักศึกษาบางคนไม่ได้ลงเรียนบางวิชาทำให้ซึ่งอาจทำให้การ Prediction หรือการ Train Model สำหรับการหาอาชีพนั้นอาจเกิดข้อผิดพลาดได้)

โดยปัจจุบันได้ทำการพัฒนาตัว Prototype ของ Function นี้เสร็จเป็นที่เรียบร้อยแล้ว โดยผลลัพท์ออกมาเป็นที่น่าพอใจทั้งที่ปรึกษาและผู้จัดทำโดย RMSE อยู่ที่ 0.534 โดยที่ n\_factor = 20 และ n\_epochs = 20 ซึ่งการทำงานมีดังนี้

เริ่มด้วยใช้ pandas อ่าน CSV ที่เข้ามาหลังจากนั้นจะใช้ pandasql ในการ query มาเฉพาะ record ที่เป็นวิศวกรรมคอมพิวเตอร์ปกติและต่อเนื่องโดยจะเลือกมาแค่เฉพาะวิชาที่ไม่ใช้ 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

Text

Description automatically generatedGraphical user interface, application

Description automatically generated*รูปที่ 5 Import Lib Surprise และ query ค่าเฉลี่ยของเกรดของนักศึกษาในวิชาแต่ละ Type มาเป็น Dataframe*

Text

Description automatically generated *รูปที่ 6 ตรวจสอบความถูกต้องของข้อมูลโดยการ query มา 1 User*

Text

Description automatically generated*รูปที่ 7 เลือก Algorithm สำหรับสร้าง Model และตรวจ ค่า RMSE และ MAE*

*รูปที่ 8 เลือกค่า n\_factors และ n\_epochs ที่ดีที่สุดสำหรับการสร้าง Model*

Text

Description automatically generated*รูปที่ 9 นำ Model ที่สร้างมาใช้งานเป็น Function*

Text

Description automatically generated

*รูปที่ 10 ทดสอบการใช้งานของ Function*

## 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

เนื่องด้วยปัญหานี้เกิดจากที่ว่าในข้อมูลของนักศึกษานั้นมีวิชาที่ในข้อมูลของรายละเอียดวิชาไม่มีอยู่ทำให้เมื่อนำ column ของ subject types มาเพิ่มใน Dataframe ของที่จะ trian model นั้นเกิดเป็น Not Found Subject Type โดยปัจจุบันแก้ปัญหาโดยการใส้เป็นหมวดอื่นๆไปก่อน

## สิ่งที่จะดำเนินการต่อไป

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