

## รายงานความก้าวหน้าวิชา CE Project 2

ครั้งที่ 2

ระหว่างวันที่ 21 ม.ค. 66 ถึงวันที่ 03 ก.พ. 66

1. ชื่อโครงการ (อังกฤษ) Curriculum Output Prediction From Student Academic Data
2. การดำเนินงานมีความก้าวหน้า 60 % (ใช้ค่า **% Complete** จาก MS Project)

มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 3 %

☐ เร็วกว่าแผน ..... วัน ☒ ช้ากว่าแผน ..... 57 ..... วัน

## 3. รายละเอียดความก้าวหน้า

นัดประชุมกับที่ปรึกษาจำนวน 1 ครั้ง

ครั้งที่ 1 : การประชุมในครั้งที่ผ่านมามีหัวข้อการประชุมดังนี้ 1) สรุปการพัฒนา requirement ครั้งก่อนที่ได้จากที่ปรึกษา  
2) แนวทางในการพัฒนาต่อ

- 1) สรุปการพัฒนา requirement ครั้งก่อนที่ได้จากที่ปรึกษา โดยส่วนของ requirement เพิ่มเติมที่ได้ทำการคุยไปกับที่ปรึกษาในการประชุมครั้งก่อนนั้นได้พัฒนาเสร็จไปแล้ว 2 ส่วนจาก 3 โดย 1) พัฒนา requirement ในส่วนการร้องขอผลการทำนายของแบบจำลองโดยทำให้ผู้ใช้งานสามารถเลือกเฉพาะวิชาที่ต้องการทำนายได้ โดยจะปรับเปลี่ยนการทำงานในการหารายวิชาที่ต้องการทำนายผ่านการ filter ข้อมูลที่ user นำเข้ามาด้วยการเลือกเฉพาะวิชาที่มี column Want\_To\_Predict เป็นเครื่องหมาย '?' 2) พัฒนา requirement ในส่วนของการทำ recommend วิชาเลือกอิสระของภาควิชาให้มีลูกเล่นเพิ่มมากขึ้นโดยทางผู้จัดทำได้ทำการนำการหาค่า cosine similarity มาใช้งานกับข้อมูลเกรดที่ได้จากการ prediction ซึ่งผลออกมาไม่เป็นที่น่าพอใจสักเท่าไรเนื่องจากความสัมพันธ์ของรายวิชาที่ออกมานั้นค่อนข้างไม่สมเหตุสมผลถ้ามองในมุมมองของความสัมพันธ์ของรายวิชา โดยทางที่ปรึกษาได้ความคิดเห็นว่าสิ่งที่ส่งผลออกมานั้นเป็นผลลัพธ์ทางสถิติอยู่แล้วเราอาจจะไม่สามารถนำเสนอข้อมูลในส่วนนี้ในเชิงการแนะนำเชิงความสัมพันธ์ทางรายวิชาได้แต่อาจจะนำเสนอในรูปแบบเชิงผลลัพธ์ทางสถิติแทนและที่ปรึกษาได้เสนอแนวทางในการปรับปรุงการทำงานของส่วนนี้ว่าให้นำข้อมูลดิบมาใช้งานแทนข้อมูลที่ผ่านการ prediction เนื่องจากอาจมีปัญญการสะสมของ error ที่มากขึ้น
- 2) แนวทางการพัฒนาต่อจากที่ได้พูดคุยกับทางที่ปรึกษานั้นทางที่ปรึกษาได้แนะนำให้ทางผู้จัดทำนั้นแจกแจง function หลักของ Web Application ออกมาเป็น working process excel เพื่อที่จะได้ดำเนินการพัฒนาได้ง่ายยิ่งขึ้น

## หัวข้อการพัฒนาโครงการตาม Gantt Chart

ศึกษาทฤษฎีที่เกี่ยวข้อง Complete 75 % (remaining 15 Hr) หมายถึง พักไว้สำหรับเพื่อในอนาคตมีเรื่องที่จะต้องศึกษาเพิ่มเติม

เตรียม Data สำหรับการพัฒนา Complete 89 % (remaining 0 Hr) หมายถึง เสร็จด้วย Data test สำหรับ Model ทั้งหมดคือ Data ของนักศึกษาปี 1-4 ปัจจุบัน ซึ่งที่ปรึกษาจะเตรียม Data ชุดนี้ให้ในเทอมการศึกษาหน้า

พัฒนา Job Classification Model Complete 95 % หมายถึง เสร็จด้วยจากข้อสรุปของการประชุมทำให้จำเป็นต้องเก็บ survey ใหม่อีกครั้ง ซึ่งวันที่จะเก็บ survey นั้นได้ทำการตกลงกับที่ปรึกษาไว้แล้วว่าจะเลือกเป็นวันที่นักศึกษาที่จบไปแล้วมาขอรับปริญญาครั้งใหญ่กัน ทำให้การพัฒนาในส่วนหัวข้อนี้นั้นอาจจะต้องวางไปก่อนแล้วดำเนินงานต่อหลังจากที่ได้ data ที่เก็บ survey มา

พัฒนา Django Web App Complete 21% (remaining 302 Hr) หลังจากที่ได้ทำการคุยแนวทางการพัฒนาต่อกับที่ปรึกษาในครั้งก่อนนั้นได้ทำการ optimize algorithms ในการทำงานของ function req prediction ให้มีการ maintain ได้ การ scalability ที่ง่ายขึ้น ตัวอย่างการแก้ไขเช่นในบางส่วนของ algorithms นั้นได้ใช้ sqldf ในการ query dataframe เพื่อทำ aggregation operation ซึ่งจะมีการใช้ query string ที่มีความยาวซับซ้อนและไม่จำเป็น โดยการแก้ไขนั้นได้เปลี่ยนไปเป็นการใช้ method aggregation ใน pandas แทน หรือ จะมีบางส่วนของ algorithms ที่ดึงข้อมูลจากฐานข้อมูลโดยไม่จำเป็นก็ได้ตัดออกไป โดยการแก้ไข optimize ข้างต้นนี้เป็นการแก้ไข เฉพาะเส้น api ที่จะ req predict ในรูปแบบเกรดของทุกวิชายังไม่ได้ทำการแก้ไขในส่วนของการ predict ที่เป็นส่วนของกลุ่มรายวิชา ในส่วนของการพัฒนา Frontend นั้นการ Design ใหม่ในส่วนมากเป็นไปได้ด้วยดี

```

@curl_example
def reqPredictPartner(df_user, model_id):
    """ 3 months ago + 10h complete for DEMO """
    # Get Curriculum from User session
    this_user_curri = df_user.loc[0, 'curriculum']
    # Create a list Model by 1 to 10 years
    all_models = list(SurpriseModel.objects.all().values())
    model = "NOTFOUND"
    model_type = "NOTFOUND"
    for i in all_models:
        if str(i['id']) == model_id:
            model = i['args'][0]
            model_type = i['args'][1]
            print(model)
            # =====
            response = []
            q_subject_data = list(subject_data.objects.all().values())
            df_subject = pd.DataFrame(q_subject_data)
            # Create a Subject dict (ID-NUM) session
            subId_name = {}
            for index, sub in df_subject.iterrows():
                dic_sub = {row.sub['subject_id']:row.sub['subject_name_eng']}
                subId_name.update(dic_sub)
            # print(subId_name)
            # =====
            thidict = genSubjectDict(df_subject)
            df_user = transformGrade(df_user)
            df_user = df_user[df_user.grade != 'Zero']
            qdata = list(Student.objects.filter(curriculum=this_user_curri).values())
            df = pd.DataFrame(qdata)
            df = transformGrade(df)
            df = []
            copy_origi_df = df
            row_Zero_df = copy_origi_df[copy_origi_df.grade != 'Zero']
            df.append(row_Zero_df)
            if model_type == 'Class':
                tempDF0 = df[1:]
                df_user['subject_class'] = df.apply(lambda row: findSubjectClass(row['subject_id'], thidict), axis=1)
                df[1]['subject_class'] = df[1].apply(lambda row: findSubjectClass(row['subject_id'], thidict), axis=1)
                q_find_Avg = "SELECT student_id, avg(grade) as grade, semester, year, curriculum, subject_class FROM tempDF0"
                q_find_Avg_user = "SELECT student_id, AVG(grade) as grade, semester, year, curriculum, subject_class FROM tempDF0"
                tempDF5_Avg = sqlf(q_find_Avg)
                tempDF_Avg_USER = sqlf(q_find_Avg_user)
                df[1] = tempDF5_Avg
                df_user = tempDF_Avg_USER
            # model file = pathlib.Path('recommend/Model/class_test_new_01_testrecommanidtest.pkl')
            subject_id_in_dataset = df[1]['subject_class'].unique()
            subject_id = df_user['subject_class']
            class:
            # model file = pathlib.Path('recommend/Model/grade_test_new_01_testrecommanidtest.pkl')
            subject_id_in_dataset = df[1]['subject_id'].unique()
            subject_id = df_user['subject_id']
            subject_id_to_pred = np.setdiff1d(subject_id_in_dataset, subject_id)
            test_set = ([Optional, sub, 4] for sub in subject_id_to_pred)
            # model file = all_model[-1]['args'][0]
            # model file = os.path.join(model_dir, model_file)
            predictions = model.test(test_set)
            pred_ratings = np.array(pred_set for pred in predictions)
            handData = list(subject_id_in_dataset)
            index_max = (-pred_ratings).argmax()[1:handData]
            for j in index_max:
                sub = subject_id_to_pred[j]
                if model_type == 'Grade':
                    listforindexdata = list(subId_name.keys())
                    if sub in listforindexdata:
                        dic = {subject_id: sub, "sub name": subId_name[sub], "grade": round(pred_ratings[j], 2)}

```

รูปที่ 1 function req prediction ตัวเก่า

```
def reqPredictPerUser_Production(df_user):
    all_models = list(SurpriseModel.objects.all().values())
    model = None
    for i in all_models:
        if str(i['id']) == '14':
            model = i['args']['model']
            model_type = i['args']['type']
    if model is None:
        print("Model not found")
        return
    df_user = transformGrade(df_user)
    selected_values = df_user.query("Want To Predict == '?'")['subject_id'].tolist()
    df_user = df_user[df_user.grade != 'Zero']
    subId_name = {row['subject_id']:row['subject_name_eng'] for row in Subject_Data.objects.values()}
    subject_ids_to_pred = selected_values
    test_set = [['Optional', sub, 4] for sub in subject_ids_to_pred]
    predictions = model.test(test_set)
    pred_ratings = [pred.est for pred in predictions]
    pred_ratings = [(subject_ids_to_pred[i], pred_ratings[i]) for i in range(len(subject_ids_to_pred))]
    pred_ratings.sort(key=lambda x: x[1], reverse=True)
    response = []
    for sub_id, grade in pred_ratings:
        if sub_id in subId_name:
            dic = {"subject_id": sub_id, "sub_name": subId_name[sub_id], "grade": round(grade, 2)}
            response.append(dic)
    return response
```

รูปที่ 2 function req prediction ตัวใหม่

#### 4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาจากครั้งก่อน

1. ส่วนของการ Design Frontend ที่ยังไม่ลงตัวเนื่องด้วยผู้จัดทำที่มีหน้าที่รับผิดชอบ Backend นั้นยังต้องปรับตัว call function ของระบบหลังบ้านที่ยังไม่ได้ optimize ให้เป็นไปตามที่ต้องแก้ไข โดยปัญหานี้ขึ้นอยู่กับช่วงดำเนินการแก้ไขเนื่องด้วยจำเป็นต้องทำการเปลี่ยนโครงสร้างการทำงานหลายจุดเนื่องด้วยการพัฒนาในส่วนของการทอมนั้นเน้นไปในการทดลองผลการใช้งานทำให้ function ส่วนมากถูกออกแบบมาสำหรับแสดงผลการทดลองจึงได้วางแนวทางการปรับแก้ไว้ว่าจะนำส่วนที่ไม่จำเป็นออกไปแล้วเรียกใช้แค่ส่วนที่จำเป็น

ปัญหา ณ ปัจจุบัน

1. ปัญหาการ prediction รายวิชาที่ไม่ครบ เนื่องด้วยการ optimize ที่กล่าวไปข้างต้นนั้นทำให้ทางผู้จัดทำได้ทำการสำรวจการทำงานของ algorithms ของ function ใหม่อีกครั้งแล้วพบว่า function นั้นไม่ได้คืนค่าการ prediction ให้ได้ครบทุกรายวิชา โดยทางผู้จัดทำได้ตั้งสมมติฐานไว้ว่าอาจเกิดจากการทำงานในส่วนสุดท้ายของ function ที่ทำการ loop + sorted ค่าของ prediction ที่ได้มา โดยทางผู้จัดทำได้วางแนวทางการแก้ไขไว้ว่าทำการเขียนส่วนการทำงานของ algorithms ในส่วนนี้ใหม่อีกครั้ง

```
predictions = model.test(test_set)
pred_ratings = np.array([pred.est for pred in predictions])
NumOfSub = len(subject_id_in_dataset)
index_max = (-pred_ratings).argsort()[0:NumOfSub]
for j in index_max:
    sub = subject_ids_to_pred[j]
    if model_type == 'Grade':
        listForFindSubId = list(subId_name.keys())
        if sub in listForFindSubId:
            dic = {"subject_id": sub, "sub_name": subId_name[sub], "grade": round(pred_ratings[j], 2)}
        else:
            dic = {"subject_id": sub, "sub_name": "ไม่พบวิชาในฐานข้อมูล", "grade": round(pred_ratings[j], 2)}
    else:
        dic = {"subject_class": sub, "subject_in_class": thisdict[sub], "grade": round(pred_ratings[j], 2)}
    response.append(dic)
```

รูปที่ 3 ส่วนการทำงานของ algorithms ที่ผู้จัดทำคิดว่ามีปัญหา

## 5. สิ่งที่จะดำเนินการต่อไป

1. ทำการพัฒนาโครงงานต่อตามแนวทางที่ได้ปรึกษากับที่ปรึกษา
2. พัฒนา requirement ในส่วนของการทำ recommend วิชาเลือกอิสระของภาควิชาให้มีลูกเล่นเพิ่มมากขึ้น โดย  
แนวทางการพัฒนาต่อจะทำการเปลี่ยนชุดข้อมูลในการทำ cosine similarity โดยใช้ข้อมูลดิบแทน
3. พัฒนา requirement ในส่วนของการทำให้ผู้ใช้งานที่เป็น admin สามารถ edit file ที่ตัวเอง upload อยู่ได้ โดย  
แนวทางการพัฒนานั้นจะทำการนำ file csv ที่อยู่ใน Database มาแสดงเป็น Column และ row ใน user interface  
แล้วทำ interact แต่ละ record เพื่อ edit แล้วนำข้อมูลที่ edit ไป update