รายงานความก้าวหน้าวิชา CE Project

ครั้งที่ 6

ระหว่างวันที่ 05 พ.ย. 65 ถึงวันที่ 25 พ.ย. 65

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

2. การดำเนินงานมีความก้าวหน้า 50 % (ใช้ค่า **% Complete** จาก MS Project)

   มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 7 %

   ☐ เร็วกว่าแผน ..... วัน ☐ ช้ากว่าแผน .....40..... วัน

3. รายละเอียดความก้าวหน้า

นัดประชุมกับที่ปรึกษาจำนวน 1 ครั้ง

ครั้งที่ 1 : แจ้งรายละเอียด progression ของการพัฒนาให้ที่ปรึกษาทราบ พร้อมแจ้งผลการนำเสนอโครงงาน CEPD ให้ที่ปรึกษาทราบ หลังจากนั้นได้พูดคุยถึงรายละเอียดการเตรียมการนำเสนอในวิชา Project จริง โดยปรับให้ผู้นำเสนอ เลือกแนวทางที่สามารถแสดงให้เห็นภาพรวมของระบบมากยิ่งขึ้น และอภิปรายถึงผลลัพท์ของระบบให้น่าดึงดูดผู้ฟังมากขึ้น และที่ปรึกษาได้ชี้แจ้งแนวทางการดำเนินงานต่อไป โดยจะสรุปแผนเก็บ survey และ การดำเนินงานออกแบบ Figma เพิ่มเติม

หัวข้อการพัฒนาโครงงานตาม Gantt Chart

ศึกษาทฤษฎีที่เกี่ยวข้อง Complete 75 % (remaining 15 Hr) หมายเหตุ พักไว้สำหรับเพื่อในอนาคตมีเรื่องที่จะต้องศึกษาเพิ่มเติม

เตรียม Server Complete 50 % (remaining 7 Hr) หมายเหตุ จากที่ได้ Discuss กับที่ปรึกษาทำให้ต้องพักส่วนนี้ไปก่อนแล้วไปมุ่งเน้นกับการพัฒนาตัว Model ทั้งหมด และ แสดงผลลัพท์

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

พัฒนา Similarity Model 100 % (Edit) เนื่องด้วยผลจากการตรวจทาน architecture ของระบบทำให้ทราบว่า Data ของเกรดนักศึกษาที่ใช้นำมา train model และพัฒนา module นั้น มีการผสมของปีการศึกษาหลายปีทำให้จำเป็นต้อง edit การทำงานส่วนการนำเข้าข้อมูลมาใช้ในการ train ใหม่

พัฒนา Job Classification Model Complete 95 % หมายเหตุเนื่องด้วยจากข้อสรุปของการประชุมทำให้จำเป็นต้องเก็บ survey ใหม่อีกครั้ง ซึ่งวันที่จะเก็บ survey นั้นได้ทำการตกลงกับที่ปรึกษาไว้แล้วว่าจะเลือกเป็นวันที่นักศึกษาที่จบไป

แล้วมาซ้อมรับปริญญาครั้งใหญ่กัน ทำให้การพัฒนาในส่วนหัวข้อนี้นั้นอาจจะต้องวางไปก่อนแล้วดำเนินงานต่อหลังจากที่ได้ data ที่เก็บ survey มา

พัฒนา Django Web App Complete 12% (remaining 247 Hr)

โดยส่วนที่ได้ทำการพัฒนาไปจะมีดังนี้

พัฒนา Backend Function Complete 26 % : โดยได้ทำการพัฒนา Function Map Subject Group ให้ได้อยู่ในรูปของ REST API และนำลงไป implement ในตัว Demo ของโครงงานเป็นที่เรียบร้อยแล้ว พร้อมกับ ทำให้สามารถรองรับการเพิ่มของวิชาใหม่ๆที่เข้ามา สามารถ หาความคล้ายกับวิชาที่มีอยู่ในระบบแล้วจับกลุ่มกันได้ โดย Complete percent ของ function นี้อยู่ที่ 75% โดยจะเผื่อไว้สำหรับการเปลี่ยนแปลงหรือ optimize ให้ดีขึ้นในภายภาคหน้า ในส่วนของการพัฒนา Call Fuction similarity หลักสูตรปกติ นั้นได้ทำการทดลองใช้งานและทำการ Implement ลง Demo ของโครงงานเป็นที่เรียบร้อยแล้ว โดย Complete percent ของ function นี้อยู่ที่ 50% เช่นเดียวกับ Function Map จะเผื่อไว้สำหรับการ optimize เพิ่มในอนาคต ซึ่งจะเป็นเช่นเดียวกันกับ หัวข้อการพัฒนา Call Fuction similarity หลักสูตรต่อเนื่อง โดย Complete percent ของ function นี้อยู่ที่ 50% เช่นเดียวกัน

```
@csrf_exempt
def addSubject(df, thisdict):
    df['subject_key'] = df['subject_key'].fillna('อื่นๆ')
    for index,subject in df.iterrows():
        subject_class = addSubjectClasstoDF(thisdict, subject['subject_id'])
        dic = {
            "subject_id":subject['subject_id'],
            "subject_name_thai":subject['subject_name_thai'],
            "subject_name_eng":subject['subject_name_eng'],
            "abstract":subject['abstract'],
            "subject_key":subject['subject_key'],
            "year":subject['year'],
            "subject_class": subject_class
        }
        this_subject = Subject_Data.objects.get(subject_id = subject['subject_id'], subject_name_thai = subject['subject_name_thai'], subject_name_eng = subject['subject_name_eng'])
        if this_subject == None:
          subject_serializer = SubjectSerializer(data=dic)
          if subject_serializer.is_valid():
              subject_serializer.save()
              print(f'save {subject_serializer.data}')
          else:
              res = "Failed to add"
              print(res)
              return res
        else:
          subject_serializer = SubjectSerializer(this_subject,data=dic)
          if subject_serializer.is_valid():
              subject_serializer.save()
              print(f'Update {subject_serializer.data}')
          else:
              res = "Failed to add"
              print(res)
              return res
    res = 'Complete add'
    return res

def queryBycurriculum(df):
  curriculums = list(df["curriculum"].unique())
  dfs = {}
  for i in curriculums:
    q = "SELECT * FROM df WHERE curriculum = '{0}'".format(i)
    temp = sqldf(q)
    dic = {i:[]}
    dic[i].append(temp)
    dfs.update(dic)
  return dfs
```

รูปที่ 1 Function Map Subject ใน Demo ของโครงงาน

```python
@csrf_exempt
def reqPredictPerUser(df_user, model_id):
  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']['model']
      model_type = i['args']['type']
  print(model)
  response = []
  q_subject_data = list(Subject_Data.objects.all().values())
  df_subject = pd.DataFrame(q_subject_data)
  subId_name = {}
  for index_sub, row_sub in df_subject.iterrows():
    dic_sub = {row_sub['subject_id']:row_sub['subject_name_eng']}
    subId_name.update(dic_sub)
  print(subId_name)
  thisdict = genSubjectDict(df_subject)
  df_user = transfromGrade(df_user)
  df_user = df_user[df_user.grade != 'Zero']      You, 4 weeks ago • ...
  qdata = list(Student.objects.filter(curriculum='วิศวกรรมคอมพิวเตอร์').values())
  df = pd.DataFrame(qdata)
  df = transfromGrade(df)
  dfs = []
  copy_origin_df = df
  non_Zero_df = copy_origin_df[copy_origin_df.grade != 'Zero']
  dfs.append(df)
  dfs.append(non_Zero_df)
  if model_type == 'Class':
    tempAVG = dfs[1]
    tempAVG_user = df_user
    df_user['subject_class'] = df.apply (lambda row: findSubjectClass(row['subject_id'], thisdict), axis=1)
    dfs[1]['subject_class'] = dfs[1].apply (lambda row: findSubjectClass(row['subject_id'], thisdict), axis=1)
    q_find_AVG = "SELECT student_id, AVG(grade) as grade, semester, year, curriculum, subject_class FROM tempAVG GROUP BY subject_class, student_id ORDER BY student_id"
    q_find_AVG_user = "SELECT student_id, AVG(grade) as grade, semester, year, curriculum, subject_class FROM tempAVG_user GROUP BY subject_class, student_id ORDER BY student_id"
    tempDFS_AVG = sqldf(q_find_AVG)
    tempDF_AVG_USER = sqldf(q_find_AVG_user)
    dfs[1] = tempDFS_AVG
    df_user = tempDF_AVG_USER
    # model_file = joblib.load('recommend/ML_model/Class_test_New_01_วิศวกรรมคอมพิวเตอร์.pkl')
    subject_id_in_dataset = dfs[1]['subject_class'].unique()
    subject_id = df_user['subject_class']
  else:
    # model_file = joblib.load('recommend/ML_model/Grade_test_New_01_วิศวกรรมคอมพิวเตอร์.pkl')
    subject_id_in_dataset = dfs[1]['subject_id'].unique()
    subject_id = df_user['subject_id']
  subject_ids_to_pred = np.setdiff1d(subject_id_in_dataset, subject_id)
  test_set = [['Optional', sub, 4] for sub in subject_ids_to_pred]
  # model_file = all_models[-1]['args']['model']
  # model_file = cPickle.loads(model_file)
  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()[:NumOfSub]
  for j in index_max:
    sub = subject_ids_to_pred[j]
    if model_type == 'Grade':
      lisForFindSubId = list(subId_name.keys())
      if sub in lisForFindSubId:
        dic = {"subject_id" : sub, "sub_name" : subId_name[sub], "grade" : round(pred_ratings[j], 2)}
      else:
        dic = {"subject_id" : sub, "sub_name" : "ไม่พบเกรดในฐานข้อมูล", "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)
  return response
```

รูปที่ 2 Similarity ใน Demo ของโครงงาน

พัฒนา API Complete 14% : โดยได้ทำการพัฒนา API Simple CSV Generator ให้พร้อมใช้งานในรูปแบบ REST API และได้ทำการ implement ลงไปใน Demo ของตัวของงานเป็นที่เรียบร้อย และทำการทำสอบการทำงานเป็นไปได้ด้วยดี โดย Complete percent ของ function นี้อยู่ที่ 75% โดยจะเผื่อไว้สำหรับการเปลี่ยนแปลงหรือ optimize ให้ดีขึ้นในภายภาคหน้า

```python
@csrf_exempt
def csv2560Download(request, curri):
    subjects = list(Subject_Data.objects.all().values())
    if curri == 'computer':
        response = HttpResponse(content_type='text/csv')
        response.write(codecs.BOM_UTF8)
        writer = csv.writer(response)
        writer.writerow(['student_id','subject_id','grade', 'semester', 'year', 'curriculum'])
        for i in subjects:
            writer.writerow(['Optional',i['subject_id'],'Your Grade', 'Optional', 'Optional', 'วิศวกรรมคอมพิวเตอร์'])
        response['Content-Disposition'] = 'attachment; filename="2560fileformat.csv"'
    else:
        response = HttpResponse(content_type='text/csv')
        writer = csv.writer(response)
        writer.writerow(['student_id','subject_id','grade', 'semester', 'year', 'curriculum'])
        for i in subjects:
            writer.writerow(['Optional',i['subject_id'],'Your Grade', 'Optional', 'Optional', 'วิศวกรรมคอมพิวเตอร์ (ต่อเนื่อง)'])
        response['Content-Disposition'] = 'attachment; filename="2560fileformat.csv"'
    return response
```

รูปที่ 3 Function Create CSV ใน Demo ของโครงงาน

4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาจากครั้งก่อน

1. การสร้าง file csv ผ่านภาษา Python : เนื่องด้วยผู้จัดทำได้ทำการทดลองใช้งานแล้วพบปัญหาว่าเมื่อใช้ Program Microsoft Excel ในการ edit หรือ view ตัว file นั้นจะทำให้ภาษาไทยใน column curriculum นั้นกลายเป็นภาษาประหลาด โดย ปัญหานี้ได้แก้ไขเป็นที่เรียบร้อยแล้ว โดยได้ทำการเพิ่มส่วนของการ Encoding เข้าไปใน Function ในขนาดที่ทำการสร้าง File CSV ขึ้นมา โดยได้เลือก Encoding Rule ให้เป็น UTF-8 โดยผลลัพท์ที่ออกมาเป็นไปอย่างน่าพอใจโดยเมื่อเปิด File ผ่าน Editor MS Excel ตัว Column Curriculum นั้นเป็นภาษาไทยปกติ โดยหลังจากนั้นได้นำเอา Function นี้ไป Implement ลงใน Demo ของตัวโครงงานเป็นที่เรียบร้อย

ปัญหา ณ ปัจจุบัน

1. Data ของเกรดนักศึกษาที่ผสมกัน : หลังจากที่ได้ทำการแก้ไขตัว algorithm ให้เป็นไปตาม sequence diagram ทำให้พบว่าตลอดมาของการพัฒนานั้นทางผู้จัดทำได้นำข้อมูลที่ผสมกันของนักศึกษา ไปใช้ในการทดลองและพัฒนาซึ่งการ train model นั้นจำเป็นที่จะต้องใช้ ชุดข้อมูลของนักศึกษาที่ได้จบการศึกษาไปแล้วเท่านั้นถึงจะได้ผลลัพท์ของการ prediction ได้สอดคล้องที่สุด

5. สิ่งที่จะดำเนินการต่อไป

1 จัดการกับปัญหา Data ของเกรดนักศึกษาที่ผสมกัน โดย Solution ที่ได้วางแผนไว้จะเป็นการใช้ปีการศึกษาที่เข้าเรียนของนักศึกษาคนนั้นๆมา แล้ว เทียบกับ ค่าปีของ Datetime ณ ปัจจุบันลบด้วย 4 เพื่อจะได้ทราบว่านักศึกษาคนนั้นๆเรียนจบแล้วรึยัง แล้วหลังจากนั้นจะทำการ filter ข้อมูลที่ query ขึ้นมาใน function similarity เฉพาะนักศึกษาที่เรียนจบไปแล้วในการ train model

2 จัดระเบียบ Function ใน Demo App ให้เป็นไปตาม Usecase Diagram และ Sequence Diagram มากขึ้นเนื่องด้วยปัจจุบันมี Function ที่เกิดขึ้นมานอกเหนือจาก Usecase และ Sequence Diagram มากมาย จะจัดระเบียบโดยการไล่ดูที่ละ Usecase และ Sequence ตัด Function ที่ไม่จำเป็นออกรวมทั้งแก้ไขชื่อให้สอดคล้อง

3 เพิ่ม log การทำงานของตัวระบบ เนื่องด้วยบาง request ใช้เวลาในการ process ค่อนข้างนานผู้จัดทำเล็งเห็นถึงความไม่สะดวกในการ Monitoring Process จึงจะเพิ่ม print log ลงไปในแต่ละ Function ว่าทำงานถึง process ไหนแล้ว