คุณานนต์ รัตนโกเศศ 6332003721

Individual Study in Computer Engineering Report

ประกอบรายวิชา 2110391

Depression classification by using NLP

Introduction

เนื่องจากโจทย์เรื่องการแยกว่าเป็นผู้ป่วยโรคซึมเศร้า เป็นโจทย์ที่น่าสนใจ ทำให้เกิดคำถามว่าจะมีข้อมูลอะไรบ้าง ที่สามารถใช้เป็น feature ในการวิเคราะห์ สุดท้ายในเทอมนี้ถึงตัดสินใจที่จะลอง traditional approch ML ในการศึกษา ประกอบไปด้วย N-gram + Naive Bayes, TF-IDF(n-gram) + Logistic regression

Data preparation

สำหรับ Dataset ที่ใช้ในการทดสอบนี้มาจากโปรเจ็คที่ อาจารย์เอกพลทำอยู่ โดยเป็นชุดข้อความ text และ label(isDepressed) ว่าเป็นผู้ป่วยหรือไม่ โดยมีข้อมูลทั้งหมด 32 รายการ แบ่งเป็น isDepressed=1 จำนวน 12 ข้อมูล และ isDepressed=0 จำนวน 22 คน

Participant_ID	text	isDepressed
302	im fine how about yourself im from los angeles...	0
307	laughter um moscow um my family moved to the u...	0
331	yes okay connecticut um to be an actor laughte...	0
335	yes im okay uh im from here originally los ang...	1
346	yes im okay here in los angeles theres a lot o...	1

Tokenization

การแบ่งคำ เนื่องจากเป็นภาษาอังกฤษเลยทำให้สามารถแบ่งได้ง่าย โดยการใช้ library word_tokenize ของ nltk หลังจากนั้นเมื่อแบ่งเสร็จแล้ว ก็จะทำการแปลงให้อยู่ในรูปแบบ lower case ทั้งหมดก่อน

Train Model

Model 1 N-gram + Naive Bayes

ทำการแปลงกลุ่มคำที่เราจะใช้ในการ train ทั้งหมดให้เป็น feature vector โดยในเคสนี้เนื่องจาก model ที่จะใช้คือ Naive Bayes ทำให้ต้องทำ feature ของแต่ละ document เป็น vector ของจำนวนคำที่เจอต่อ vocab ต่างๆ เลยใช้ countvectorizer() ก่อนนำเข้าโมเดล

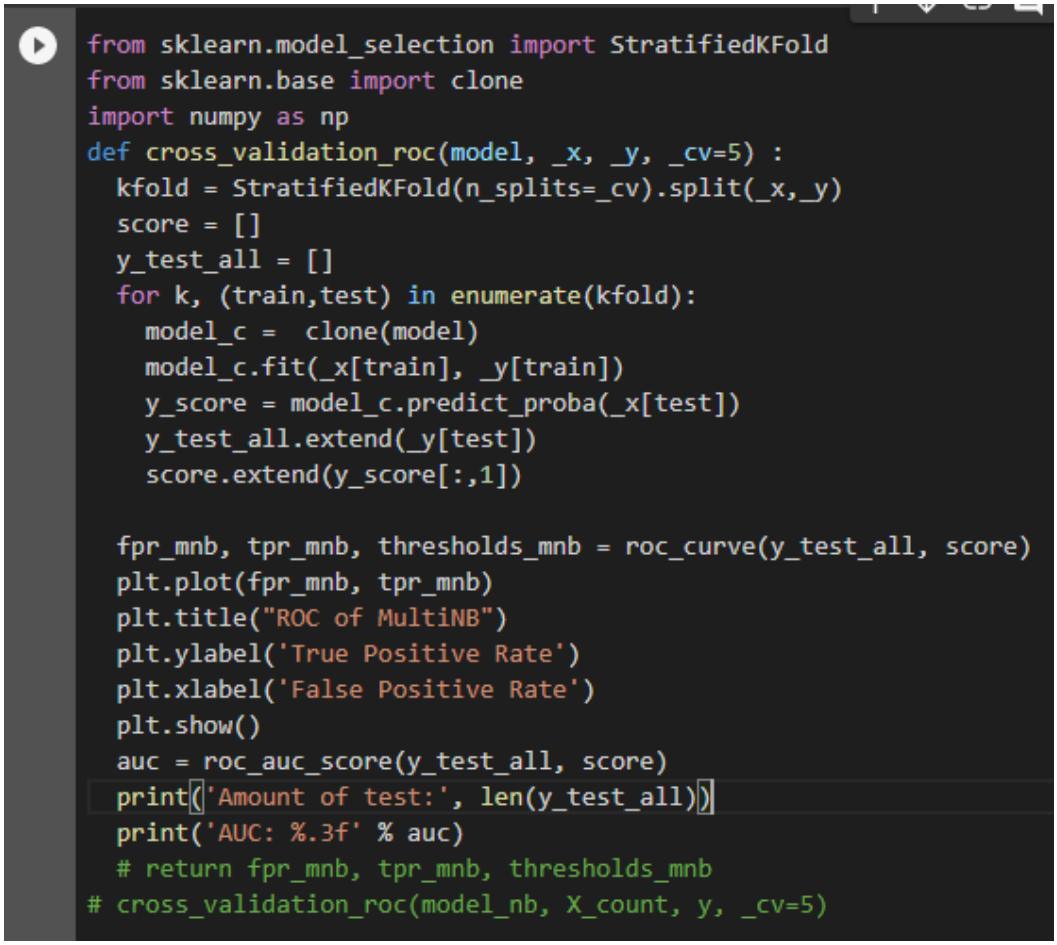
```
count_vec = CountVectorizer(
    ngram_range = (1,2),
    tokenizer=tokenizer,
    min_df = 3,
    max_df=0.9
)
model_nb = MultinomialNB()
X_count = count_vec.fit_transform(X)
cross_validation_roc(model_nb, X_count, y, _cv=5)
```

Model 2: TF-IDF(n-gram) + Logistic Regression

ทำการแปลงกลุ่มคำที่เราจะใช้ในการ train ทั้งหมดให้เป็น feature vector โดยในเคสนี้เนื่องจาก model ที่จะใช้คือ Logistic Regression ทำให้ต้องทำ feature ของแต่ละ document สามารถใช้ TF-IDF ได้ เพราะการทำ word encoder แบบนี้จะทำการหาคำที่มีโอกาสเจอเฉพาะเอกสารด้วย และไม่ให้ความสำคัญกับคำที่มีทุกๆ ประโยค

Result

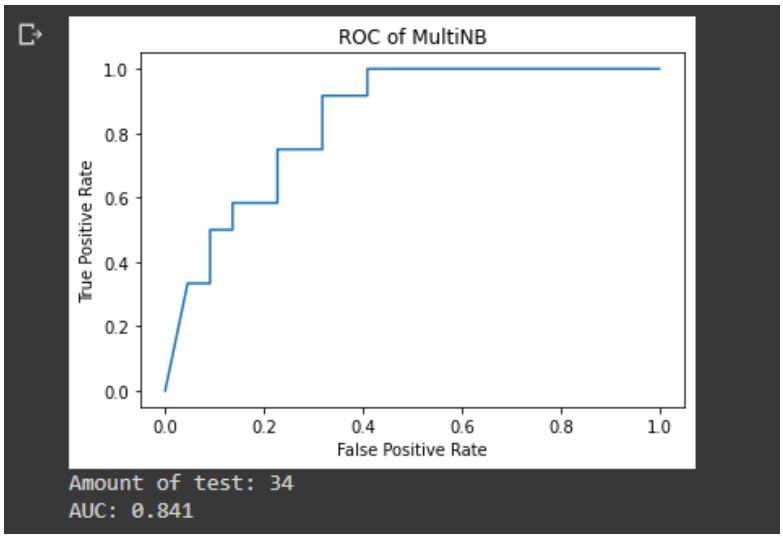
ขั้นตอนการแบ่งข้อมูล เนื่องจากข้อมูลที่มีน้อย เลยทำการวัดผลโดยใช้ K-fold validation โดยเก็บผลลัพท์ที่ได้ในแต่ละครั้ง แล้วนำมา stack กันเป็นผลลัพท์สุดท้าย โดยใช้การดู ROC และ AUC เพราะโมเดลที่เรียนรู้ได้อาจจะต้องปรับตาม threshold เพื่อหาจุดเหมาะสมระหว่าง True positive rate กับ false positive rate



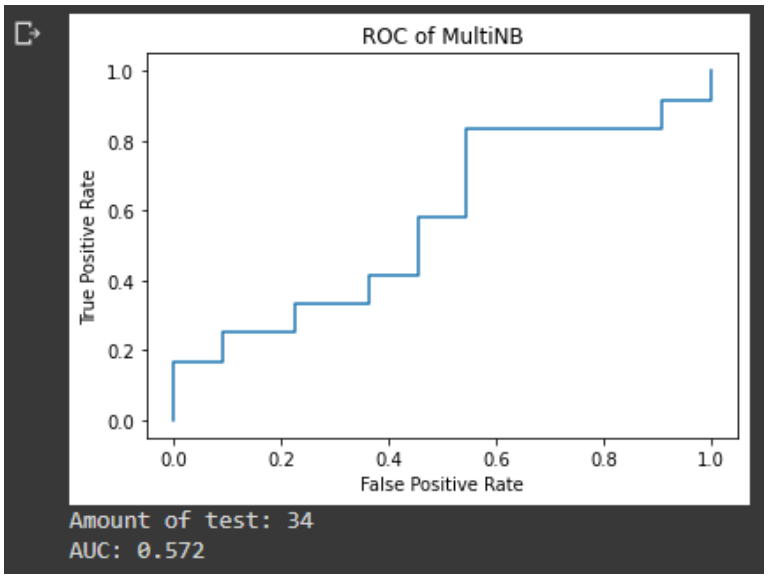
```python
from sklearn.model_selection import StratifiedKFold
from sklearn.base import clone
import numpy as np
def cross_validation_roc(model, _x, _y, _cv=5) :
  kfold = StratifiedKFold(n_splits=_cv).split(_x,_y)
  score = []
  y_test_all = []
  for k, (train,test) in enumerate(kfold):
    model_c =  clone(model)
    model_c.fit(_x[train], _y[train])
    y_score = model_c.predict_proba(_x[test])
    y_test_all.extend(_y[test])
    score.extend(y_score[:,1])

  fpr_mnb, tpr_mnb, thresholds_mnb = roc_curve(y_test_all, score)
  plt.plot(fpr_mnb, tpr_mnb)
  plt.title("ROC of MultiNB")
  plt.ylabel('True Positive Rate')
  plt.xlabel('False Positive Rate')
  plt.show()
  auc = roc_auc_score(y_test_all, score)
  print('Amount of test:', len(y_test_all))
  print('AUC: %.3f' % auc)
  # return fpr_mnb, tpr_mnb, thresholds_mnb
# cross_validation_roc(model_nb, X_count, y, _cv=5)
```

ผลลัพท์ที่ได้ N-gram + Naive Bayes



ผลลัพท์ที่ได้ TF-IDF(n-gram) + Logistic Regression



อุปสรรคที่เกิดขึ้น วิธีแก้ปัญหา

1. เนื่องจากข้อมูลน้อยในตอนแรก เลยยังไม่เข้าใจวิธีการวัดผล แต่อาจารย์ได้แนะนำว่าให้ทำการ stack result หลังจากแบ่งกลุ่มการทดลอง
2. เนื่องจากปี 3 เนื้อหาที่เรียนอัดกันเยอะมาก ตอนแรกอยากลอง model bert ด้วย แต่มีเวลาไม่พอ ทำให้ได้ลองแค่ โมเดลพื้นฐาน แต่ก็ได้ผลลัพท์ที่ค่อนข้างดี

คุณานนต์ รัตนโกเศศ 6332003721

สิ่งที่ได้รับ

- ได้ลองโมเดล classical ML ว่ามีประสิทธิภาพเป็นยังไงได้บาง
- ได้ลองนำโมเดลเดิม ไปทดสอบกับ dataset :https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned?datasetId=2400767&sortBy=voteCount ซึ่งเห็นว่าผลลัพท์ค่อนข้างดี ไม่แพ้พวก deep learning เลย
- ได้เข้าใจ feature กลุ่มที่เป็น text มากขึ้น
- เห็นผลลัพท์ว่าการตัด stop word ออกมีผลทำให้ผลลัพท์ดีขึ้น
- การวัดผลไม่สามารถดูแค่ accuracy กับ confusion matrix อย่างเดียวได้ ต้องดูตาม ROC หรือ AUC ด้วย