

Lab2: Experiment Report

6440126922 นิปฐน อังควิชัย

จากการสร้างแบบจำลอง Logistic Regression ทำนายประเภทของธุรกรรมทางการเงิน ตามขั้นตอนใน Google Colab notebook นี้ [lab2-dm.ipynb](#) และมีกระบวนการ Data preprocessing ดังนี้

กระบวนการ Data preprocessing

- 1) สร้างตัวแปร day_range จาก PostDate
- 2) สร้างตัวแปร is_df จาก localAmount
- 3) เพิ่มข้อความ 'payable' ไปที่ lineDescription ใน Group Acctnum (class) ที่ 30
- 4) ทำความสะอาดข้อความใน lineDescription
 - a) ลบ punctuation
 - b) ตัดคำ
 - c) ลบ stopword
 - d) ลบคำที่น้อยกว่า 3
 - e) ทำ lemmatization
 - f) ลบ non-dictionary word
 - g) ลบ blank rows จากข้อความที่ผ่านกระบวนการทำความสะอาดแล้ว

ทำให้ได้ผลลัพธ์ของแบบจำลองจาก Evaluation metrics มีดังนี้

ผลลัพธ์ของแบบจำลองจาก Evaluation metrics

	precision	recall	f1-score	support
10	0.88	0.88	0.88	525
11	0.46	0.71	0.56	42
12	0.55	0.66	0.60	62
13	0.73	0.30	0.42	81
14	1.00	0.75	0.86	4
15	0.75	0.54	0.63	536
20	0.86	0.59	0.70	32
30	1.00	1.00	1.00	2222
31	0.56	0.25	0.35	59
32	0.75	0.50	0.60	398
36	0.58	0.57	0.58	214
37	0.00	0.00	0.00	5
40	0.93	0.88	0.90	543
43	0.98	0.88	0.93	50
51	0.58	0.35	0.44	62
61	0.75	0.13	0.22	23
62	0.73	0.93	0.82	1645
63	0.80	0.62	0.70	165
accuracy			0.84	6668
macro avg	0.72	0.59	0.62	6668
weighted avg	0.84	0.84	0.83	6668

รูปที่ 1: ผลลัพธ์ของ Evaluation metrics

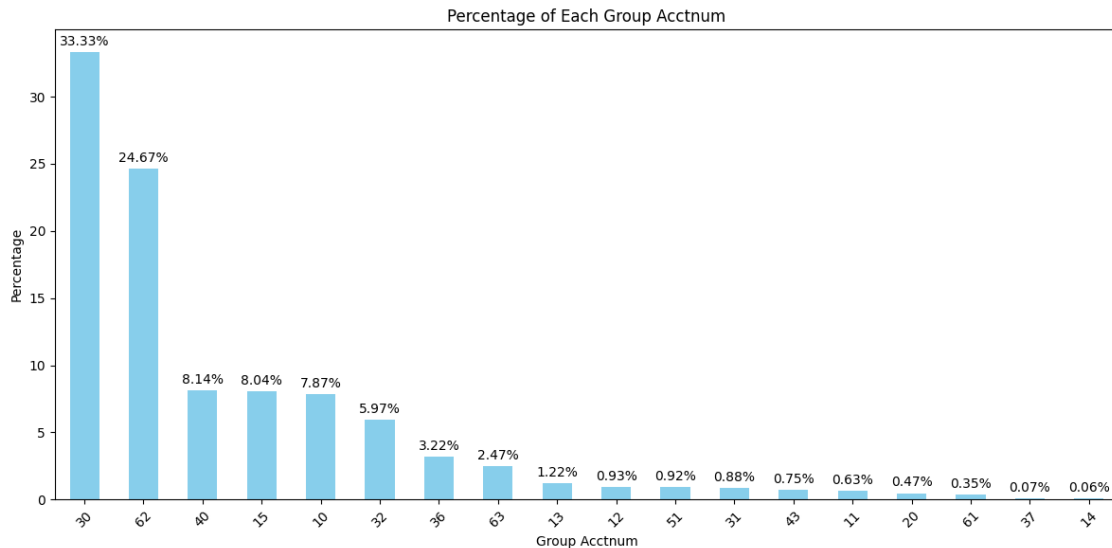
- Accuracy ได้ 84%
- Macro average F1 ได้ 62%
- Weighted average F1 ได้ 83%

Macro average เป็นการคำนวณค่าเฉลี่ย performance ของโมเดลในแต่ละ class โดยให้น้ำหนักกับทุก class เท่ากันหมด ไม่สนใจจำนวน sample ของ class นั้นๆ

Weighted average เป็นการคำนวณค่าเฉลี่ย performance ของโมเดลในแต่ละ class โดยถ่วงน้ำหนักด้วยจำนวน sample ใน class นั้นๆ ทำให้ class ที่มีจำนวน sample มากจะส่งผลต่อ overall score มาก

จากผลลัพธ์จะเห็นได้ว่า Weighted average F1 มีค่าสูงกว่า Macro average F1 กล่าวคือเมื่อคำนวณ performance ของโมเดลด้วยการให้น้ำหนักแต่ละ class ไม่เท่ากัน (Weighted average) โมเดลของเราได้ performance ดีกว่า อาจตีความได้ว่าการกระจายตัวของ class ใน dataset ของเราไม่สมดุล จะเห็นได้รูปที่ 2 ด้านล่างว่า Group Acctnum ที่ 30 และ 62 มีจำนวนเกิน 50% ของ dataset ชุดนี้ ถือว่าเป็น majority class และเมื่อดู F1 score ของ 2 class นี้จะพบว่าได้ score เป็น 100% และ 82% ตามลำดับซึ่งสูงมาก ดังนั้นเมื่อมีการคำนวณโดยให้น้ำหนักความสำคัญกับ majority class จึงทำให้ได้ performance สูงกว่าการคำนวณแบบให้น้ำหนักกับทุก class เท่ากันอย่างการคำนวณแบบ Macro average

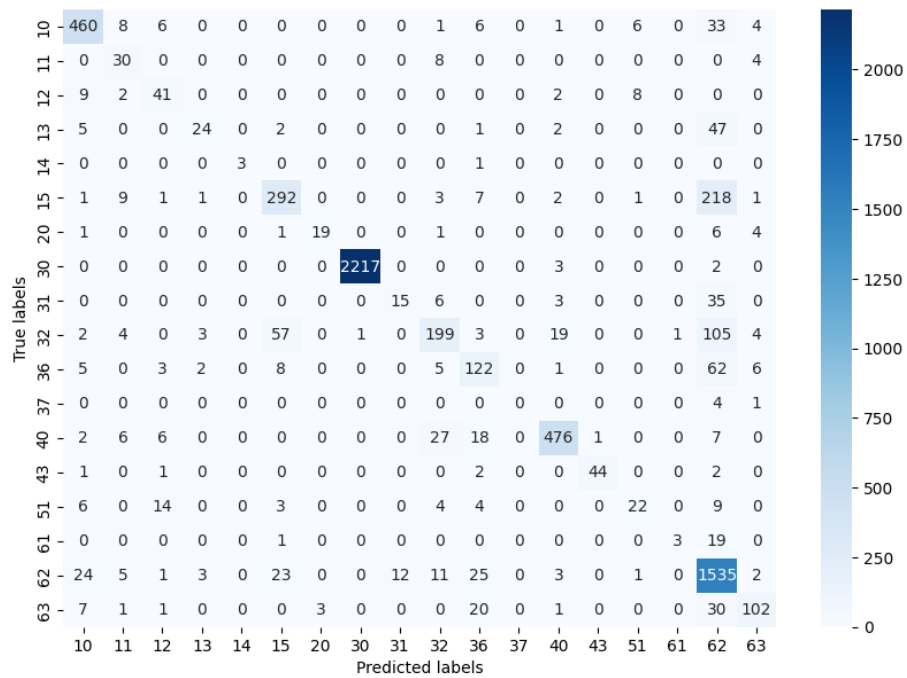
หากเรางานของเราให้ความสำคัญกับ majority class จากผลลัพธ์ของโมเดลก็ถือว่าโมเดลนี้มีประสิทธิภาพที่ดี มี Weighted average F1 มากกว่าร้อยละ 80 แต่หากงานของเราให้ความสำคัญกับทุก class เท่าๆ กัน โมเดลของเรายังต้องการพัฒนาเพิ่มเติมอีก เนื่องจากได้ Macro average เพียงร้อยละ 62



รูปที่ 2: การกระจายตัวของ class ใน dataset

Weighted average recall และ precision มีค่าเท่ากัน ในขณะที่ Macro average recall และ precision มีค่าแตกต่างกันถึง 13% โดยมีค่า recall & precision เป็น 59% และ 72% ตามลำดับ แสดงให้เห็นว่าโดยภาพรวมโมเดลยัง detect positive class ไม่ดีเท่าที่ควร จากการที่ได้ recall 59% แต่ว่าโมเดลมีความแม่นยำในการทำนาย class ที่เราสามารถเชื่อถือได้ จากการที่ได้ precision สูงถึง 72%

จากการวิเคราะห์ Confusion matrix ในรูปที่ 3 ด้านล่าง จะเห็นความน่าสนใจที่โมเดลของเราทำนาย Group Acctnum ที่ 15 ผิดเป็น Group Acctnum ที่ 62 ถึง 218 ครั้ง อีกทั้งยังทาย Group Acctnum ที่ 32 ผิดเป็น Group Acctnum ที่ 62 ถึง 105 ครั้ง แสดงให้เห็นว่าความน่าเชื่อถือในการทำนาย Group Acctnum เป็น 62 ค่อนข้างน้อย เมื่อดู recall ของ Group Acctnum ที่ 62 ในรูปที่ 1 จะพบว่ามีค่า 73% ซึ่งมิต่ำน้อยเมื่อเทียบกับ Group Acctnum ที่มีความถี่ใกล้เคียงกันอย่าง Group Acctnum ที่ 30 และ 40 ซึ่งมี precision สูงถึง 100% และ 93% ตามลำดับ จากตรงนี้อาจหา feature ที่โดดเด่นของ Group Acctnum ที่ 62 เพื่อให้โมเดลสามารถทำนาย class นี้ได้แม่นยำยิ่งขึ้น



รูปที่ 3: ผลลัพธ์ของ Confusion metrics