Traditional ML vs MLP

สรุปไฮไลท์ที่น่าสนใจ

- ข้อมูล Structured data ที่มีการทำ feature engineering แล้ว การทดลองตัวแบบระหว่าง Traditional ML และ MLP จะให้ค่า accuracy ไม่แตกต่างกันมาก
- จากการทดลอง จะเห็นความแตกต่างของการปรับ Activation function in Hidden layer ซึ่งพบว่า การใช้ sigmoid ได้ค่าเฉลี่ยของ accuracy สูงกว่าการใช้ relu แม้จะทำการทดลองที่จำนวน layer และ batch size เท่ากัน
- การกำหนดจำนวน hidden node และ hidden layer เยอะ ทำให้ค่า accuracy ดีขึ้นจริง แต่มีโอกาสที่จะไม่ส่งผลให้โมเดลดีขึ้น อาจทำให้ model เกิดการจดจำ data train set แต่ไม่เกิดการเรียนรู้ pattern ของ data test set ส่งผลให้เกิดการ overfit ได้
- การทดลองโดยตัวแบบ MLP ใช้เวลาและทรัพยากรที่สูงกว่า ตัวแบบ Traditional ML

Introduction

การทดลองนี้จัดทำขึ้นเพื่อสร้างแบบลองสำหรับจำแนกระดับความน่าเชื่อถือของบุคคล (Credit score classification) ออกเป็น 3 กลุ่ม ซึ่งอาศัยข้อมูลที่ธนาคารสามารถรวบรวมได้มาใช้เป็น input ของแบบจำลอง เพื่อใช้ในการอนุมัติสินเชื่อทางการเงิน โดยการทดลองนี้มุ่งเน้นการเปรียบเทียบแบบจำลองที่มีความแตกต่างกันระหว่าง traditional machine learning และ deep learning

Data

Data source: https://www.kaggle.com/datasets/parisrohan/credit-score-classification?select=train.csv

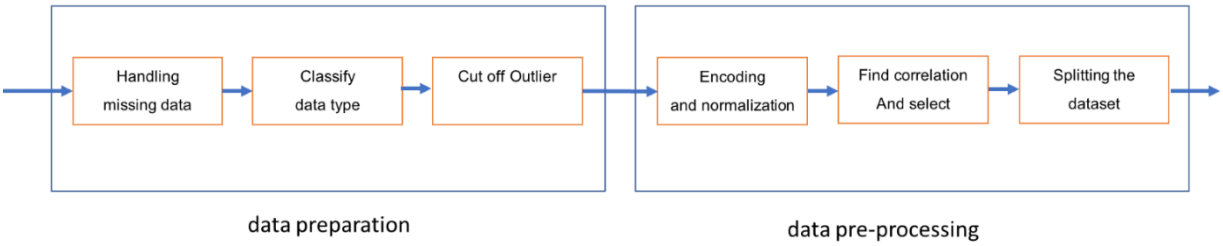
ข้อมูลมีจำนวน 27 columns มีรายละเอียดดังนี้

columns	Description
ID	Represents a unique identification of an entry
Customer_ID	Represents a unique identification of a person
Month	Represents the month of the year
Name	Represents the name of a person

Age	Represents the age of the person
SSN	Represents the social security number of a person
Occupation	Represents the occupation of the person
Annual_Income	Represents the annual income of the person
Monthly_Inhand_Salary	Represents the monthly base salary of a person
Num_Bank_Accounts	Represents the number of bank accounts a person holds
Num_Credit_Card	Represents the interest rate on credit cardRepresents the number of other credit cards held by a person
Interest_Rate	Represents the interest rate on credit card
Num_of_Loan	Represents the number of loans taken from the bank
Type_of_Loan	Represents the types of loan taken by a person
Delay_from_due_date	Represents the average number of days delayed from the payment date
Num_of_Delayed_Payment	Represents the average number of payments delayed by a person
Changed_Credit_Limit	Represents the percentage change in credit card limit
Num_Credit_Inquiries	Represents the number of credit card inquiries
Credit_Mix	Represents the classification of the mix of credits
Outstanding_Debt	Represents the remaining debt to be paid (in USD)
Credit_Utilization_Ratio	Represents the utilization ratio of credit card
Credit_History_Age	Represents the age of credit history of the person
Payment_of_Min_Amount	Represents whether only the minimum amount was paid by the person
Total_EMI_per_month	Represents the monthly EMI payments (in USD)
Amount_invested_monthly	Represents the monthly amount invested by the customer (in USD)
Payment_Behaviour	Represents the payment behavior of the customer (in USD)
Monthly_Balance	Represents the monthly balance amount of the customer (in USD)
Credit_Score	Represents the bracket of credit score (Poor, Standard, Good)

จากข้อมูลจะตัดส่วนที่ไม่ส่งผลต่อการสร้างตัวแบบจำลองออกไป และเลือกเฉพาะ columns ที่เป็นปัจจัยที่อาจส่งผลต่อ Credit_Score จำนวน 17 columns

มีกระบวนการจัดเตรียมข้อมูลดังนี้



- Data preparation

- Handing missing data ในกระบวนการนี้จะมีการเติมตัวเลขที่ไม่ทราบค่าเป็น 0 แทน

- Classify data type เป็นการจัดประเภทให้ถูกต้องจากเดิมเป็น object ซึ่งเป็นข้อมูลที่มีประเภท int และ string ปะปนกัน

- Cut off outlier เป็นตรวจสอบและตัดข้อมูลที่ผิดปกติออก เช่น การตัด age ที่มากกว่า 100 ปี หรือต่ำกว่า 18 เป็นต้น

- Data pre-processing

- Encoding and normalization เป็นการนำข้อมูลประเภท Category Encoding เป็น Number เพื่อให้สามารถนำเข้าแบบจำลองได้ และทำการ normalization เพื่อตัดเรื่องของความแตกต่างกันหน่วยของข้อมูล

- Find correlation นำข้อมูลมาหา correlation และเลือกเฉพาะ columns ที่มี correlation มากกว่า 0.1

- Splitting the dataset นำข้อมูลมาแยก test set และ train set ในอัตราส่วน 30:70

Traditional ML

จากการทดลองนี้ เราเลือกใช้ Machine Learning 2 Models ได้แก่

- **Support Vector Machine (SVM)**: มีการตั้งค่า hyperparameters ดังนี้
  - C: float, default=1.0
  - kernel: rbf
  - gamma: scale

  เมื่อทำการทดลอง Support Vector Machine Model ให้ประสิทธิภาพ ดังตาราง

	precision	recall	f1-score	support
0	0.691	0.573	0.627	4796
1	0.784	0.688	0.733	10688
2	0.504	0.782	0.613	3873
accuracy			0.678	19357
macro avg	0.660	0.681	0.658	19357
weighted avg	0.705	0.678	0.683	19357

- **Decision Tree**: มีการตั้งค่า hyperparameters ดังนี้
  - criterion: gini
  - splitter: {“best”, “random”}, default=”best”
  - min_samples_spli: tint or float, default=2

  เมื่อทำการทดลอง Decision Tree Model ให้ประสิทธิภาพ ดังตาราง

	precision	recall	f1-score	support
0	0.634	0.653	0.643	4796
1	0.733	0.727	0.730	10688
2	0.576	0.569	0.572	3873
accuracy			0.677	19357
macro avg	0.648	0.649	0.648	19357
weighted avg	0.677	0.677	0.677	19357

จากการทดลอง Traditional ML ทั้ง 2 Models พบว่า Support Vector Machine Model มีค่า accuracy สูงกว่า โดยมีค่า เป็น 0.678

Deep Learning

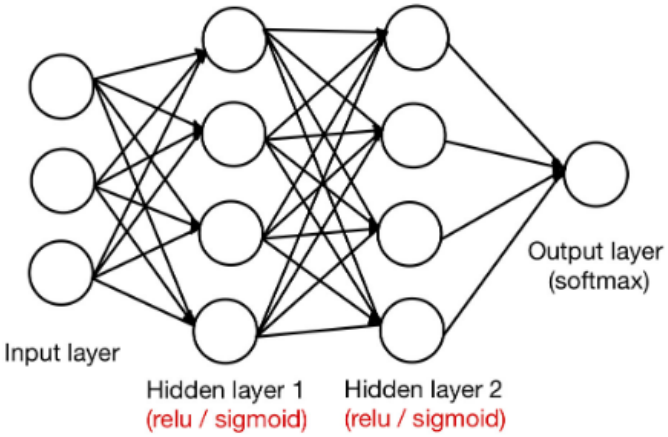
Network architecture

รายละเอียด Model หลัก แบ่งออกเป็น 2 กลุ่ม แบบ 2 hidden layer และ 3 hidden layer

โดยปรับ Activation function in Hidden layer เป็น relu และ sigmoid

และกำหนด Activation function in Output layer เป็น softmax

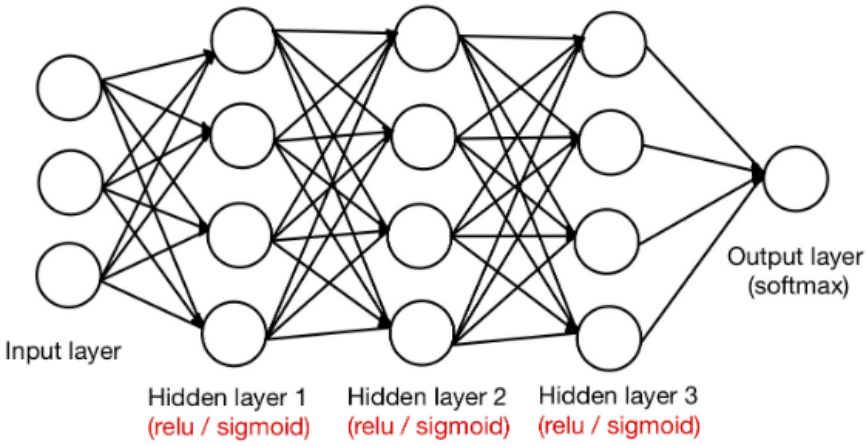
- กลุ่ม 2 hidden layer มี architecture ดังนี้



โดยมีการกำหนดจำนวน hidden node ตามตารางด้านล่างนี้

Model	Hidden layer 1	Hidden layer 2
1	27	9
2	81	9
3	243	9
4	27	27
5	81	27
6	243	27
7	27	81
8	81	81
9	243	81
10	27	243
11	81	243
12	243	243

- กลุ่ม 3 hidden layer มี architecture ดังนี้



โดยมีการกำหนดจำนวน hidden node ตามตารางด้านล่างนี้

Model	Hidden layer 1	Hidden layer 2	Hidden layer 3
1	27	9	9
2	81	9	9
3	243	9	9
4	27	27	9
5	81	27	9
6	243	27	9
7	27	81	27
8	81	81	27
9	243	81	27
10	27	243	81
11	81	243	81
12	243	243	81
13	27	243	243
14	81	243	243
15	243	243	243

Training ทดลองโดยใช้ Google Colab ด้วย GPU รุ่น Tesla T4 ทำการทดลองแต่ละ Model โดยใช้ Keras และตั้งค่า hyperparameter ดังนี้

- Number of Hidden layers ได้แก่ 2 และ 3 hidden layers
- Number of Units in Hidden layer ได้แก่ 9, 27, 81 และ 243
- Activation function in Hidden layer ได้แก่ relu และ sigmoid

และได้ Fix ค่าต่างๆ ดังนี้

- Batch size เป็น 512
- Dropout rate เป็น 0.3
- Learning rate เป็น 0.001
- Activation function in Output layer เป็น softmax
- Loss function เป็น Cross-entropy
- Optimizer เป็น Adam
- Epoch เป็น 500

Result

ผลการทดลองปรับ hyperparameters กลุ่ม 2 hidden layer จะได้ค่า accuracy ของ model เป็นตามตาราง

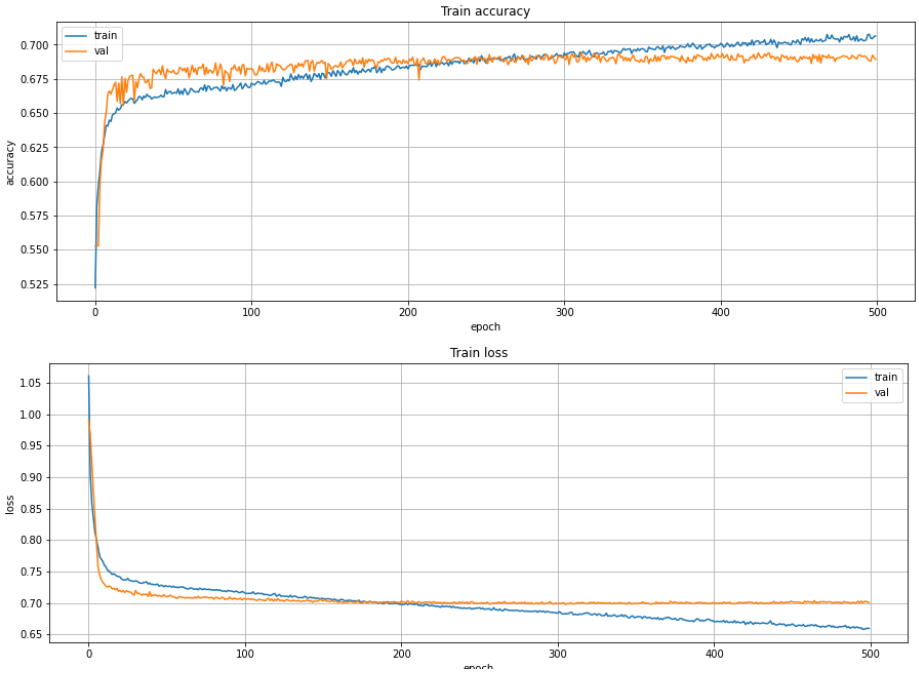
Model	Hidden layer 1	Hidden layer 2	Accuracy test set	
			relu	sigmoid
1	27	9	0.6767	0.6777
2	81	9	0.6788	0.6742
3	243	9	0.6764	0.6713
4	27	27	0.6813	0.6791
5	81	27	0.6778	0.6809
6	243	27	0.6747	0.6800
7	27	81	0.6802	0.6802
8	81	81	0.6817	0.6825
9	243	81	0.6777	0.6845
10	27	243	0.6802	0.6853
11	81	243	0.6821	0.6899
12	243	243	0.6755	0.6864

ผลการทดลองปรับ hyperparameters กลุ่ม 3 hidden layer จะได้ค่า accuracy ของ model เป็นตามตาราง

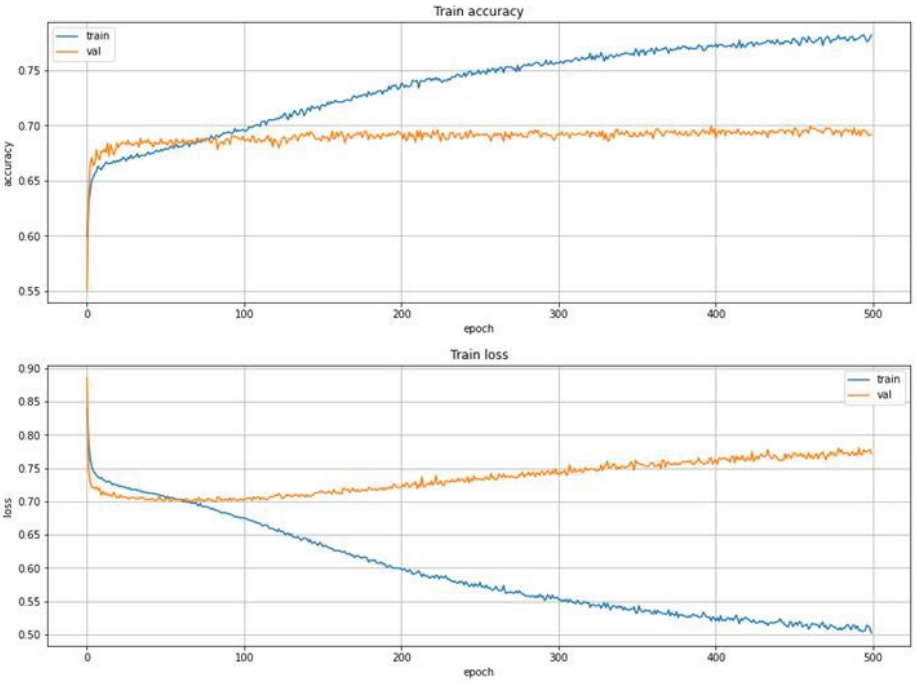
Model	Hidden layer 1	Hidden layer 2	Hidden layer 3	Accuracy test set	
				relu	sigmoid
1	27	9	9	0.6760	0.6753
2	81	9	9	0.6772	0.6795
3	243	9	9	0.6789	0.6805
4	27	27	9	0.6766	0.6752
5	81	27	9	0.6729	0.6818
6	243	27	9	0.6721	0.6785
7	27	81	27	0.6781	0.6842
8	81	81	27	0.6743	0.6793
9	243	81	27	0.6676	0.6773
10	27	243	81	0.6754	0.6871
11	81	243	81	0.6643	0.6806
12	243	243	81	0.6733	0.6751
13	27	243	243	0.6725	0.6922
14	81	243	243	0.6733	0.6875
15	243	243	243	0.6757	0.6792

จากการทดลองในกลุ่ม 2 hidden layer พบว่า Model ที่ 11 ได้ ค่า accuracy สูงสุดในกลุ่ม เป็น 0.6899 และจากการทดลองในกลุ่ม 3 hidden layer พบว่า Model ที่ 13 ได้ ค่า accuracy สูงสุดในกลุ่ม เป็น 0.6922

เพื่อหาตัวแบบที่เหมาะสม จึงมาพิจารณาจากกราฟ Train accuracy และ Train loss ของ data train set และ validation ในแต่ละ epoch



กราฟแสดง accuracy และ loss ของ กลุ่ม 2 hidden layer Model ที่ 11



กราฟแสดง accuracy และ loss ของ กลุ่ม 3 hidden layer Model ที่ 13

พบว่ากราฟของกลุ่ม 3 hidden layer Model ที่ 13 เกิดการ overfit โดยพิจารณาจากกราฟ train accuracy เมื่อเริ่มต้นทำการ train model เส้นกราฟ train และ validation มีค่าสูงขึ้นเรื่อยๆและมาตัดกันที่ช่วง epoch ≈ 70 หลังจากนั้น เส้นกราฟ train ยังคงสูงขึ้นเรื่อยๆ ส่วน validation จะเริ่มคงที่

และจากกราฟ train loss จะเห็นว่าสอดคล้องกับกราฟ train accuracy คือ เมื่อเริ่มต้น เส้นกราฟ train และ validation มีค่าลดลงเรื่อยๆ และมีจุดตัดกันที่ช่วง epoch ≈ 70 หลังจากนั้น เส้นกราฟ train ยังคงลดลงเรื่อยๆ แต่ validation กลับสูงขึ้น จึงเลือกตัวแบบ MLP จาก กลุ่ม 2 hidden layer Model ที่ 11

จากตัวแบบ MLP ที่เลือกใช้ ทำการนำค่า initial random weights ออกเพื่อหาประสิทธิภาพเฉลี่ยได้ผลตามตารางด้านล่าง

No.	Accuracy ของกลุ่ม 2 hidden layer Model ที่ 11
1.	0.6904
2.	0.6871
3.	0.6868
Mean	0.6881
SD	0.0017

จากตางรางประสิทธิภาพเฉลี่ย สรุปได้ว่า การ Train MLP โดยใช้ตัวแบบข้างต้น ได้ค่าเฉลี่ย accuracy เท่ากับ 0.6881 และ SD เท่ากับ 0.0017
เมื่อนำมาหาค่า F1 – Score ได้ค่าดังนี้

	precision	recall	f1-score	support
0	0.701	0.608	0.651	4796
1	0.763	0.730	0.746	10688
2	0.531	0.682	0.597	3873
accuracy			0.690	19357
macro avg	0.665	0.673	0.665	19357
weighted avg	0.701	0.690	0.693	19357

Discussion

- กรณี Overfit ใน กลุ่ม 3 hidden layer ของ Model ที่ 13 คาดว่าอาจเกิดจากการกำหนดจำนวน hidden node ในแต่ละ layer ที่มากเกินไป จึงทำให้ model เกิดการจดจำ data train set แต่ไม่เกิดการเรียนรู้ pattern ของ data test set
- กรณี Imbalanced dataset แก้โดยใช้วิธี UnderSampling โดยสุ่มตัวอย่างจาก class ที่มีจำนวนมากให้มีขนาดใกล้เคียงหรือเท่ากับ class ที่มีจำนวนน้อย

Conclusion

จากการทดสอบตัวแบบ Traditional ML และ MLP สามารถจำแนกระดับความน่าเชื่อถือของบุคคล (Credit score classification)ได้ โดยการใช้ตัวแบบ Traditional ML จาก Support Vector Machine(SVM) มีค่า F1-score เท่ากับ 0.678 และ การใช้ตัวแบบ MLP กลุ่ม 2 hidden layer model ที่ 11 มีค่า F1-score เท่ากับ 0.690

เมื่อเปรียบเทียบประสิทธิภาพระหว่าง Traditional ML และ MLP พบว่าการใช้ตัวแบบ MLP มีค่า F1-score สูงกว่า การใช้ตัวแบบ Traditional ML แต่ให้ประสิทธิภาพไม่แตกต่างกันมาก เนื่องจากข้อมูลที่นำมาเป็นข้อมูล Structured data ซึ่งเป็นการคัดเลือก feature มาแล้วจากกระบวนการทำ data pre-processing จึงส่งผลให้การใช้ MLP มีค่า accuracy ไม่ต่างกับการใช้ Traditional MLทั่วไป

MLP มีคุณลักษณะสำคัญที่ไม่จำเป็นต้องทำ feature engineering ก่อนนำมา train model การใช้ MLP จึงเหมาะกับการ train ข้อมูล ที่เป็น Unstructured data เช่น รูปภาพ และเสียง เป็นต้น เพราะข้อมูล Unstructured data นำมาทำ feature engineering ได้ค่อนข้างยาก

<u>รายชื่อสมาชิกกลุ่ม</u>

1. กานต์ เกริกชัยวัน รหัส 6410422006 (25%)

- Prepare and clean dataset
- Train and tune ML & MLP model
- Write result, discussion, conclusion report

2. ชวิศ เตชจินดาวงศ์ รหัส 6410422010 (25%)

- Prepare and clean dataset
- Train and tune ML & MLP model
- Write ML, discussion, conclusion report

3. ธาริกา อาจิตรนุภาพ รหัส 6410422011 (25%)

- Prepare and clean dataset
- Train and tune MLP model
- Write network architecture, training, discussion, conclusion report

4. นทีธร ชุลีกราน รหัส 6410422028 (25%)

- EDA and feature selection
- Design, train, and tune MLP model
- Write introduction, data, discussion, conclusion report

งานชิ้นนี้เป็นส่วนหนึ่งของรายวิชา Deep Learning (DADS7202)

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาการวิเคราะห์ข้อมูลและวิทยาการข้อมูล

คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์