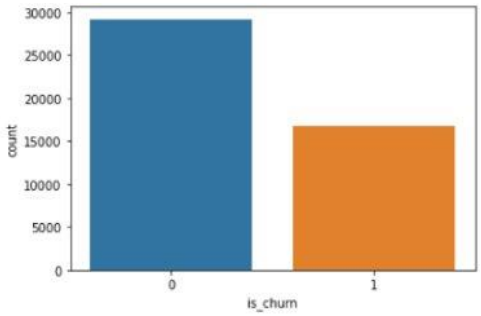
Project Objectives:

To compared Traditional ML and MLP

Dataset Description:

Dataset: Local Contractor from B2B Industry
Classification Problems: Churn Prediction
Total Customers(n): 45875
Number of Not Churn (class 0): 29156
Number of Churn (class 1): 16719



Total Features: 32
Features are:

	1 month	3 months	6 months	12 months	lifetime (3 yrs rolling)
First to last purchase day	count distinct	count distinct	count distinct	count distinct	count distinct
Total amount	sum	sum	sum	sum	sum
No order	count distinct	count distinct	count distinct	count distinct	count distinct
Active month	count distinct	count distinct	count distinct	count distinct	count distinct
revisit duration	avg / std	avg / std	avg / std	avg / std	avg / std

Number of Train Test Class:

Train/Test	Not Churn [Class 0]	Churn [Class 1]
Train	11202	11202
Test	9622	5517

Using Traditional ML:

We should Random Forest Classifier, traditional ML, to predict churn. Before we build the model, we resample the imbalanced dataset.

First, we handle an imbalanced dataset with undersampling. N of each class[0,1] = 11202 and accuracy and variance is 0.933, +/- 0.003 respectively.

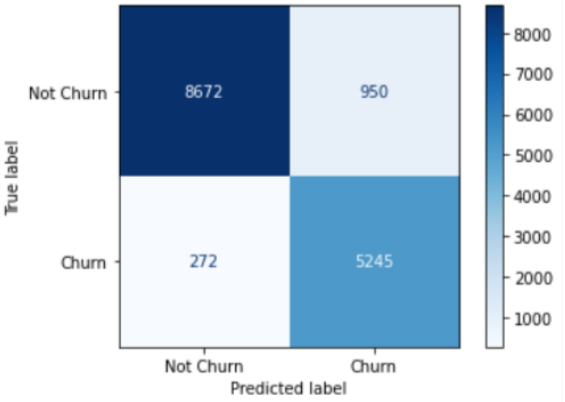
Next, we use oversampling to resample data. N of each class[0,1] = 19534 and accuracy and variance is 0.932, +/- 0.004 respectively.

So, we decide to use undersampling because of high accuracy and low variance. Then we input training data size = 0.67 into the model and test dataset with size = 0.33

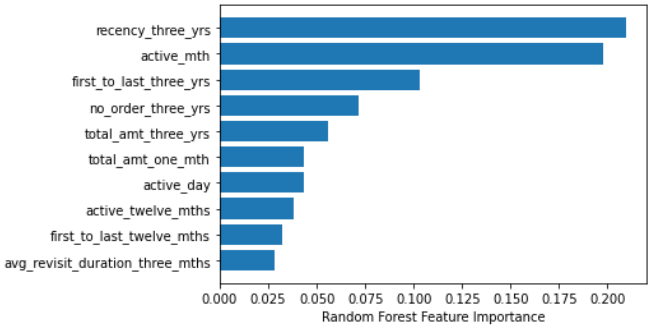
Performance (ML):

The model result is shown below:

	precision	recall	f1-score	support
0	0.97	0.90	0.93	9622
1	0.85	0.95	0.90	5517
accuracy			0.92	15139
macro avg	0.91	0.93	0.91	15139
weighted avg	0.92	0.92	0.92	15139



The top 10 features importance of this model are:



Using MLP:

การทดลองใช้ Google Colab Pro ด้วย GPU รุ่น Tesla P100-PCIE-16GB
ทำการทดลองแต่ละ Model โดยใช้ Keras tuner และตั้งค่า Hyperparameter ดังนี้

- **Number of Hidden layers:** min value = 1, max value = 3
- **Number of Units in Hidden layer:** [64, 128, 256, 512, 1024]
- **Activation function in Hidden layer:** [relu, tanh]
- **Dropout(rate=0.25):** [True, False]
- **Learning rate:** [0.01, 0.001, 0.0001]

และได้ Fix ค่าต่างๆดังนี้

- **Activation function in Output layer**: sigmoid
- **Loss function**: binary_crossentropy
- **Optimizer**: Adam
- **Batch size**: 32
- **Epoch:** 100

ทำการเลือก Hyperparameter แต่ละ Model โดยใช้ RandomSearch Tuner ได้ผลลัพธ์ดังนี้

Model	#Layer	#Unit1	#Unit2	#Unit3	Activation function	Dropout (rate=0.25)	Learning rate
1	1	512	-	-	tanh	True	0.001
2	1	64	-	-	tanh	False	0.001
3	1	64	-	-	relu	True	0.01
4	1	64	-	-	relu	True	0.01
5	1	1024	-	-	relu	True	0.01
6	3	1024	256	256	tanh	False	0.001
7	1	512	-	-	tanh	True	0.0001
8	3	512	64	64	tanh	True	0.01
9	2	512	128	-	tanh	True	0.0001
10	1	64	-	-	tanh	True	0.001

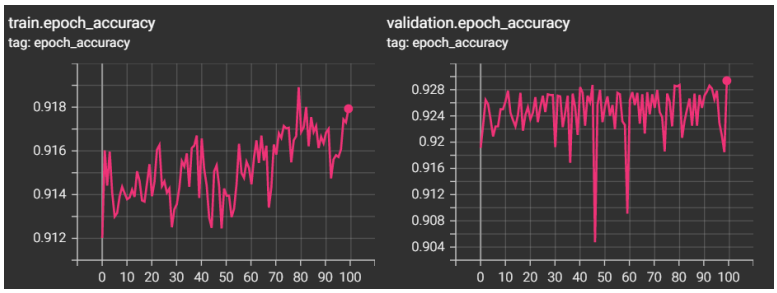
Performance (MLP):

หลังจากทำการ train model ครบทั้ง 10 model จะได้ผลลัพธ์ดังนี้

Model	Train Accuracy	Validation Accuracy
1	0.9235	0.9273
2	**0.9304**	0.9248
3	0.9222	0.9258
4	0.9211	0.9271
5	0.9216	0.9273
6	0.9281	0.9262
7	0.9210	0.9261
8	0.9179	**0.9294**
9	0.9250	0.9272
10	0.9238	0.9263

โดย Model 8 ให้ค่า Validation Accuracy สูงสุด และมี Hyperparameter ดังนี้

- **Number of Hidden layers:** 3
- **Number of Units in Hidden layer:**
    - Layer 1: 512
    - Layer 2: 64
    - Layer 3: 64
- **Activation function in Hidden layer:** tanh
- **Dropout(rate=0.25):** True
- **Learning rate:** 0.01
- **Activation function in Output layer**: sigmoid
- **Loss function**: binary_crossentropy
- **Optimizer**: Adam
- **Batch size**: 32
- **Epoch:** 100



รูปที่ 1 ภาพแสดงค่า Accuracy ขณะเรียนรู้ของ Model 8

รูปที่ 2 ภาพแสดงค่า loss ขณะเรียนรู้ของ Model 8

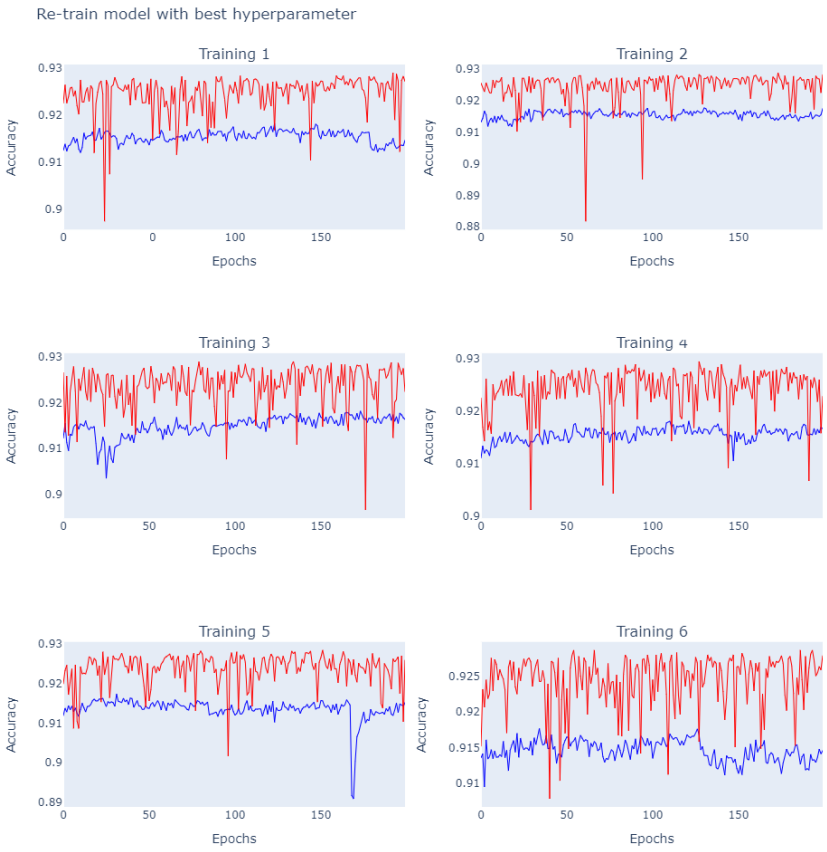
ทำการ Re-train model ด้วย Hyperparameter ที่ให้ค่า Validation Accuracy สูงสุด โดยใช้ epoch เท่ากับ 200 ทั้งหมด 6 รอบ โดยให้ผลลัพธ์ดังนี้

Training	Train Accuracy	Validation Accuracy
1	0.9212	0.9272
2	0.9211	0.9283
3	0.9157	0.9224
4	0.9159	0.9228
5	0.9202	0.9259
6	0.9210	0.9278

และมีค่าเฉลี่ยเท่ากับ

Avg. Train Accuracy = 0.9192

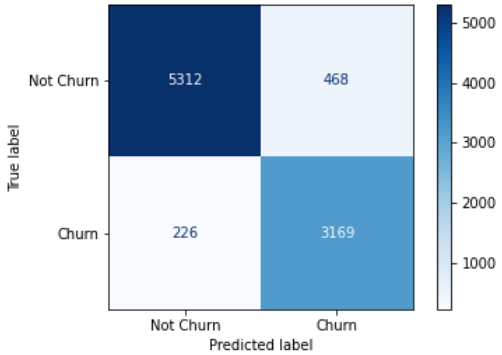
Avg. Validation Accuracy = 0.9257



รูปที่ 3 ภาพแสดงค่า Accuracy ของ train set (สีฟ้า) และ validation set (สีแดง) ขณะเรียนรู้

คำนวณ Confusion matrix จาก model ให้ผลลัพธ์ดังนี้

	Precision	Recall	F1 score
Not Churn	0.9592	0.9190	0.9387
Churn	0.8713	0.9334	0.9013



รูปที่ 4 ภาพแสดง Confusion Matrix

Comparing Result Traditional ML and MLP:

Metric	Traditional ML	MLP
Accuracy	0.92	0.93
Precision	0.91	0.92
Recall	0.93	0.93
F1 score	0.91	0.92

Discussion

จากการทดสอบเพื่อเปรียบเทียบประสิทธิภาพระหว่าง Traditional ML และ MLP กับชุดข้อมูลเพื่อสร้าง Churn Prediction model พบว่าทั้งสอง model ให้ประสิทธิภาพใกล้เคียงกันโดยที่มีค่า F1 score สำหรับ Traditional ML เท่ากับ 0.91 และ MLP เท่ากับ 0.92 ดังนั้นการสร้าง model กับข้อมูลที่เป็น Structure data โดยใช้ MLP นั้นไม่ได้ส่งผลให้ model มีประสิทธิภาพเพิ่มขึ้นได้อย่างมีนัยสำคัญและ MLP model มี Hyperparameter ที่ต้องทดสอบจำนวนเยอะกว่าทำให้ใช้เวลาในการทดสอบนาน

Pros and Cons

	Traditional ML	MLP
Pro	It takes less resource than an MLP Can explain how model work and what are the important features that impact to model that business can use in the further	Can solve many problems e.g. Classification prediction problems, Regression prediction problems Can input many types of data Such as Image data, Text Data, Time series data and Other types of data No need for feature engineering
Con	Need structured data Need to do feature engineering to get feature importance	Consume many resources Many hyperparameter to tune Hard to interpret the features that impact to the business

Recommendation:

- ในกรณีของ structured data ข้อมูลประเภทนี้เป็นข้อมูลที่เราสามารถทำการทดลองเพื่อเลือก feature สำหรับการ train model ได้ด้วยวิธีต่างๆอยู่แล้วเช่นการทำ EDA หรือ การทำ feature engineering ดังนั้นการใช้ Traditional ML จึงมีความเหมาะสมมากกว่า
- ในการ Train Model โดยปกติแล้ว MLP ต้องการจำนวนของข้อมูลในการ Train มากกว่า Traditional ML ดังนั้นหากข้อมูลไม่เยอะเพียงพอ อาจจะทำให้ประสิทธิภาพของ MLP แย่กว่า Traditional ML
- MLP ใช้เวลา และ resource อื่นๆ มากกว่า ทั้งในส่วนของการ Train และ Test แต่ท้ายที่สุดผลลัพธ์ที่ได้มีความใกล้เคียงกับ Traditional ML
- หากเป็นการทำงานร่วมกับฝั่ง business user การใช้ Traditaional ML ทำให้เราสามารถอธิบายได้ว่า feature ไหนบ้างที่มีผลต่อ model performance ในขณะที่ MLP เราไม่สามารถอธิบายได้
- MPL ใช้เวลาในการTrainมากกว่า Traditional ML
- MLP ยากต่อการทำความเข้าใจมากกว่า Traditional ML และต้องการ hardware ที่มีศักยาภาพสูงกว่า Traditional ML
- การเลือกใช้ Traditional ML ในการจัดการกับ structured data น่าจะมีประสิทธิภาพดีกว่าในขณะที่ประสิทธิผลมีความใกล้เคียงกัน