

# 大數據在2021年 企業應用與發展

Ethan Wu, 2021/4





# Machine Learning

Dataset

Competition

Discussion

自我介紹



# 自我介紹

- 學歷&經歷:

時間	身份	學校&組織
2018-Now	Data Mining Analyst	南山人壽
2021	Top 30 teams in Open Category	Shopee Code League 2021
2019-2020	人工智慧競賽職業組隊長	T. Brain – 菜雞互啄 Kaggle – 台灣梯度下降第一品牌
2016-2018	碩士(MBA)	國立中山大學- 企業管理學系
2017	台灣代表實習生	新浪微博總部 – 天氣通產品部門 @北京
2012-2016	經濟學學士	東海大學經濟系

- 專業證照與資格:

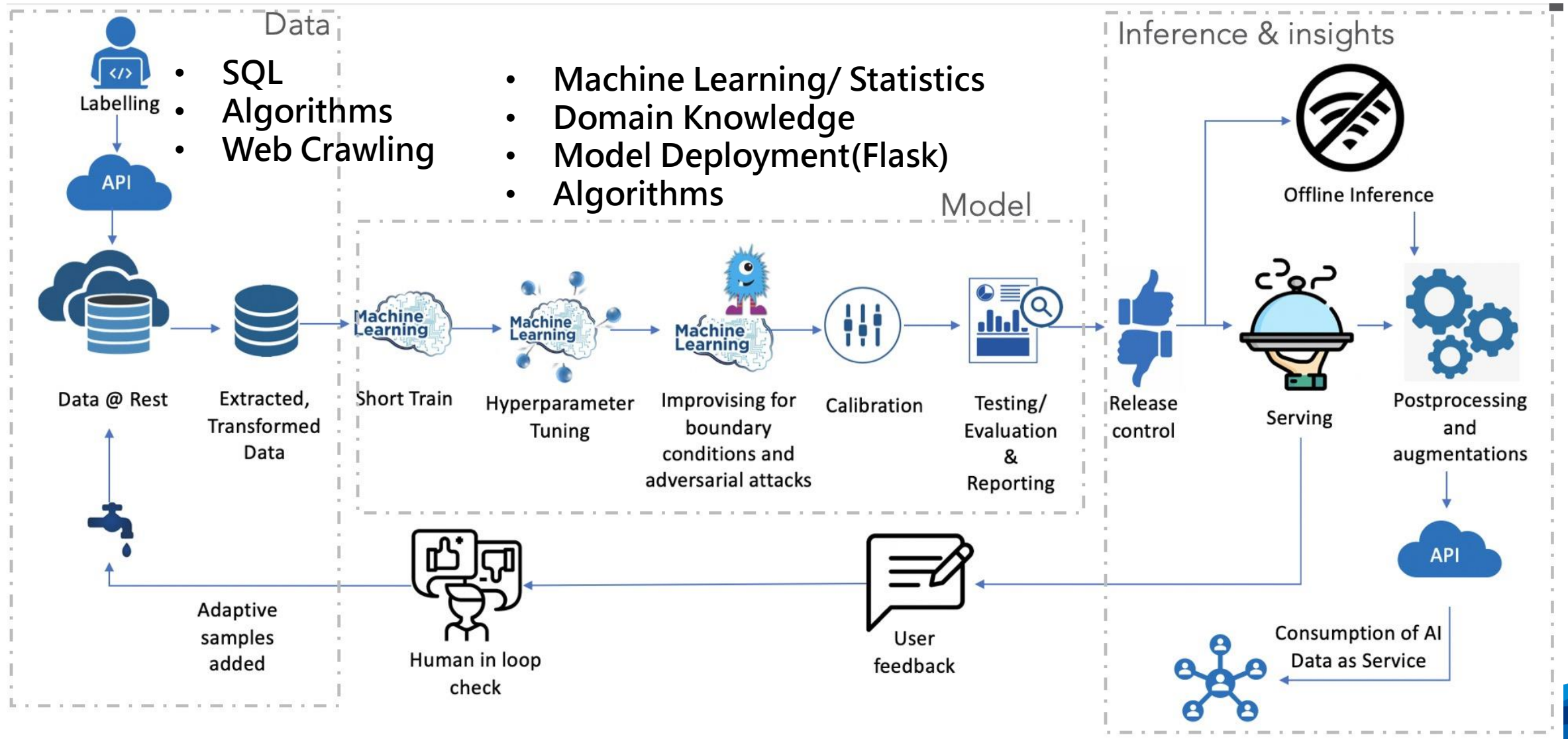
資格	認證組織
壽險管理師(FLMI Level 3)/ACS	美國壽險管理學會 (Life Office Management Association)
北極計畫開源代碼認證貢獻者	GitHub軟體原始碼代管服務平台 (Arctic Code Vault Contributor)
開源代碼開發者專案成員	GitHub軟體原始碼代管服務平台 (Developer Program Member)

# 數據分析經驗

領域	專案項目	性質	成效
人壽保險領域 (Nan Shan)	演算法設計 (圖論演算法)	專案項目	尋找鄰近高資產客戶之普客，挖掘潛在購買商機&推薦合適保障缺口
人壽保險領域 (Nan Shan)	理賠再購模型建置	產學合作 / 內部開發	分析客戶個體偏好&行銷獎勵機制，行銷名單內外再購率8倍
投資金融領域 (Portfolio)	台股動能指標訊號模型 (網路爬蟲、GBDT)	專案項目	回測期間2018-2020 年化報酬率 > 100%之模型訊號
銀行個金領域 (E. Sun)	信用卡盜刷偵測模型 (台灣區競賽: 異常偵測)	職業組商業競賽: 2 <sup>th</sup> 模型準確度: Top 1%	計算各卡別消費金額是否超出正常消費範圍，預防銀行與客戶盜刷損失
電子商務領域 (Shopee)	自然語言處理- 顧客評論語意情緒分析	亞太區競賽: 15 <sup>th</sup>	處理不平衡資料、找尋特定情緒字眼作為優化商城服務依據
電子商務領域 (Shopee)	精準行銷、推薦系統 挖掘高資產價值客戶	亞太區競賽: 24 <sup>th</sup> 台灣區競賽: 5 <sup>th</sup>	製作精準行銷名單，提升電子商務廣告投放轉換率/ 預測高資產價值客戶
電子商務領域 (Shopee)	商城劣質商家異常偵測 (Anomaly Detection)	亞太區競賽: 23 <sup>th</sup>	偵測出極短時間內同一群帳戶互相洗商城評價，提高優良商城認證公信力
電子商務領域 (Shopee)	商品影像分類 (電腦視覺任務)	亞太區競賽: 67 <sup>th</sup>	應用類神經網路&梯度提升樹算法，辨別並刪除商城賣家品質不良商品圖示

# Summary of Data Skillset on 2021

- Tableau/Power BI





# Machine Learning

Dataset

Competition

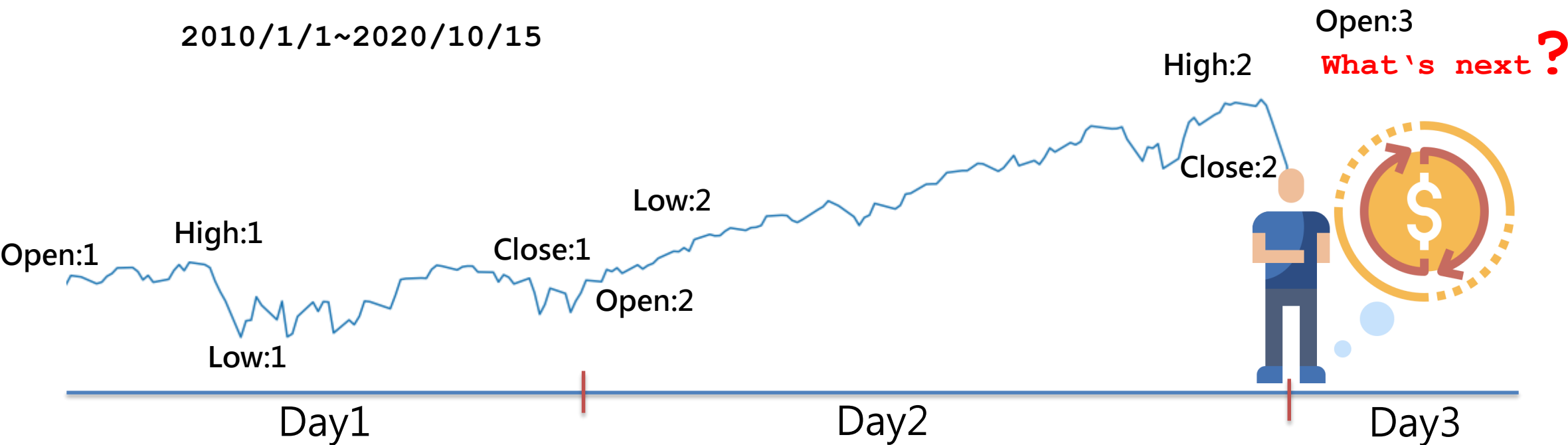
Discussion

## 專案1. 動能投資模型

# Transform Time-Series data to cross-section data

Duration:

2010/1/1~2020/10/15



	TxDate	StockID	Open	High	Low	Close	Adj Close	Volume	type	有價證券名稱	市場別	產業別	公開發行/上市(櫃)/發行日
0	2010-01-04	1101	27.1635	27.483101	27.083599	27.4032	13.858723	10321349.0	stock	台泥	上市	水泥工業	1962-02-09
1	2010-01-05	1101	27.6429	28.601601	27.563000	28.3619	14.343570	60016780.0	stock	台泥	上市	水泥工業	1962-02-09
2	2010-01-06	1101	28.3619	29.080900	28.282000	28.9611	14.646606	44831404.0	stock	台泥	上市	水泥工業	1962-02-09
3	2010-01-07	1101	29.0410	29.080900	28.361900	28.4018	14.363750	18095530.0	stock	台泥	上市	水泥工業	1962-02-09
4	2010-01-08	1101	28.4018	28.601601	28.082300	28.2820	14.303162	13307856.0	stock	台泥	上市	水泥工業	1962-02-09



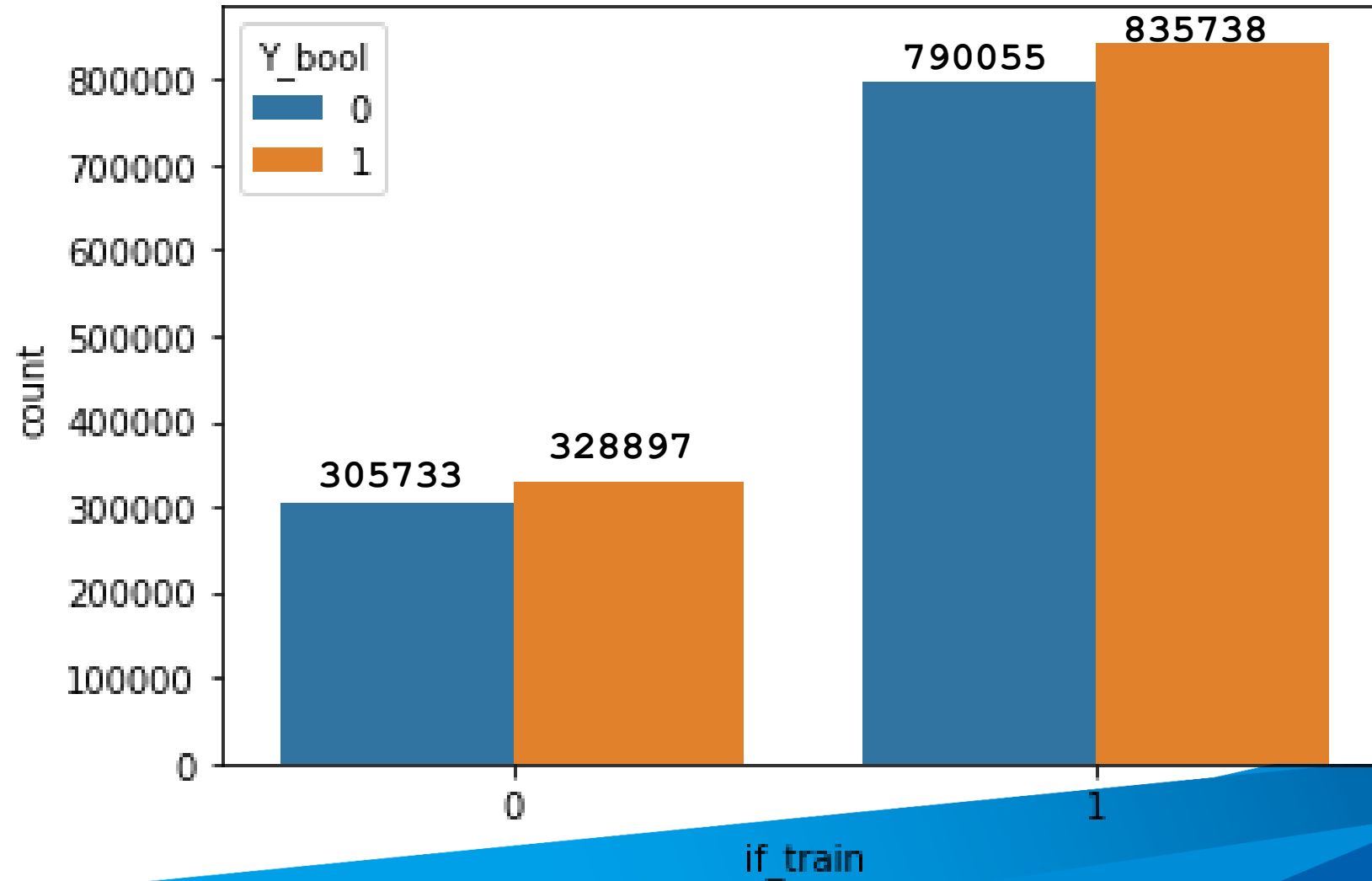
# Split Train/Test data

Test Duration:

2018/1/2~2020/10/15

Train Duration:

2010/1/1~2017/12/29





# Data Description

Dataset Shape: (2288783, 23)

	Name	dtypes	Missing	Uniques	First Value	Second Value	Third Value	Entropy
0	TxDate	datetime64[ns]	0	2650	2010-01-04 00:00:00	2010-01-05 00:00:00	2010-01-06 00:00:00	11.36644
1	StockID	int64	0	950	1101	1101	1101	9.8469
2	Open	float64	7368	254932	27.1635	27.6429	28.3619	14.27262
3	High	float64	7368	259958	27.4831	28.6016	29.0809	14.33147
4	Low	float64	7368	255687	27.0836	27.563	28.282	14.31288
5	Close	float64	7368	258628	27.4032	28.3619	28.9611	14.33336
6	Adj Close	float64	7368	868154	13.8587	14.3436	14.6466	18.82828
7	Volume	float64	7368	959554	1.03213e+07	6.00168e+07	4.48314e+07	17.26461
8	type	object	0	2	stock	stock	stock	0.03970
9	有價證券名稱	object	9757	943	台泥	台泥	台泥	9.83935
10	市場別	object	9757	1	上市	上市	上市	0.00000
11	產業別	object	9757	28	水泥工業	水泥工業	水泥工業	4.48537
12	公開發行/上市(櫃)/發行日	datetime64[ns]	9757	708	1962-02-09 00:00:00	1962-02-09 00:00:00	1962-02-09 00:00:00	8.77633

# Step1: Stock information Clustering

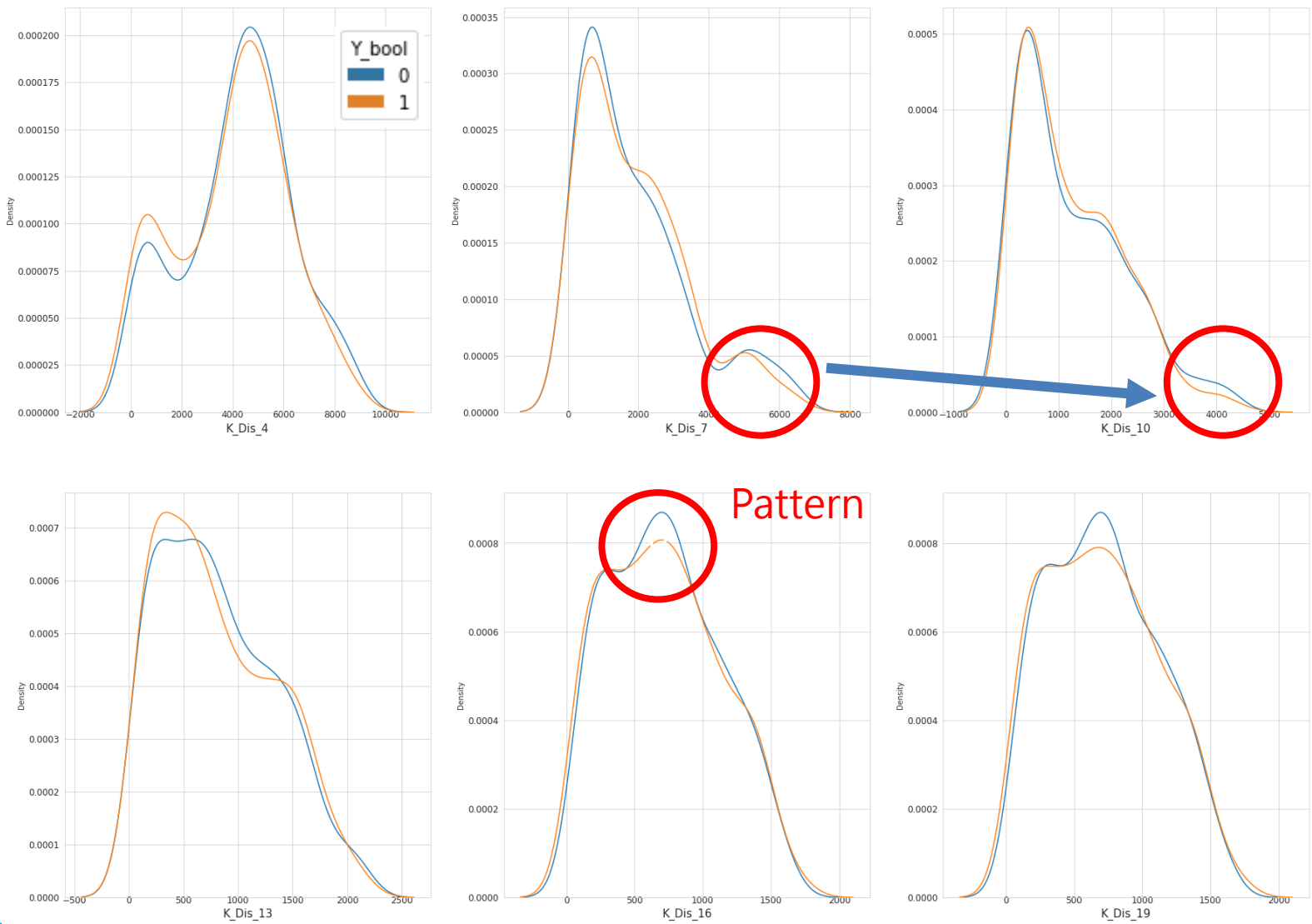
大市值 Group :



組數越多, Pattern越穩定

	StockID	Open	High	Low	Close	Adj Close
count	453.0	453.000000	453.000000	453.000000	453.000000	453.000000
mean	3008.0	3482.229581	3504.216336	3407.737307	3425.584989	3174.811207
std	0.0	1151.787326	1155.651750	1140.939156	1143.179969	1109.788865
min	3008.0	2080.000000	2080.000000	2005.000000	2010.000000	1815.583370
25%	3008.0	2525.000000	2525.000000	2455.000000	2455.000000	2246.780030
50%	3008.0	3075.000000	3090.000000	2980.000000	3000.000000	2748.621830
75%	3008.0	4445.000000	4495.000000	4385.000000	4430.000000	4163.034180
max	3008.0	5995.000000	6015.000000	5955.000000	5970.000000	5655.832520

$f(\text{不同Group} | K \text{ Dis } 4)$



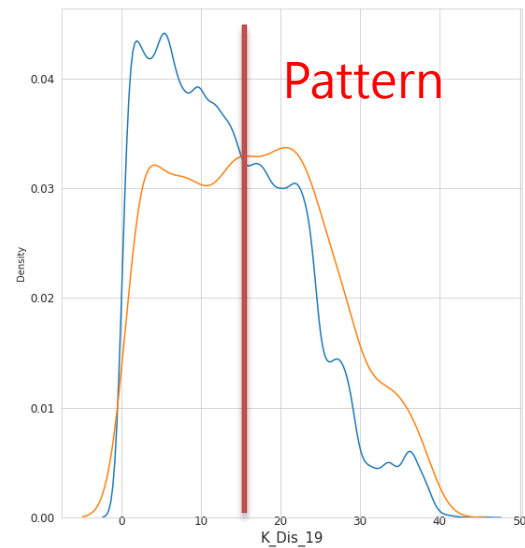
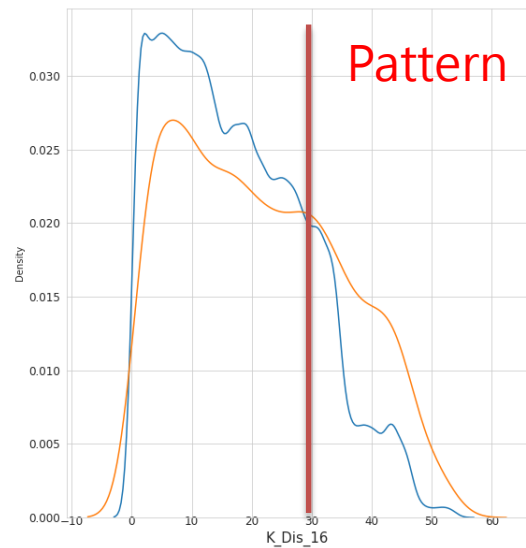
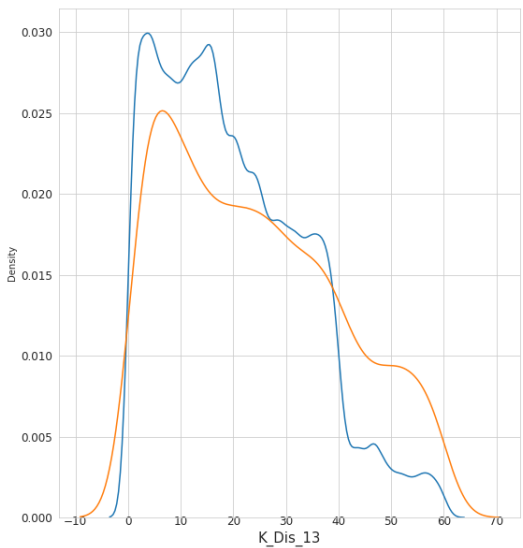
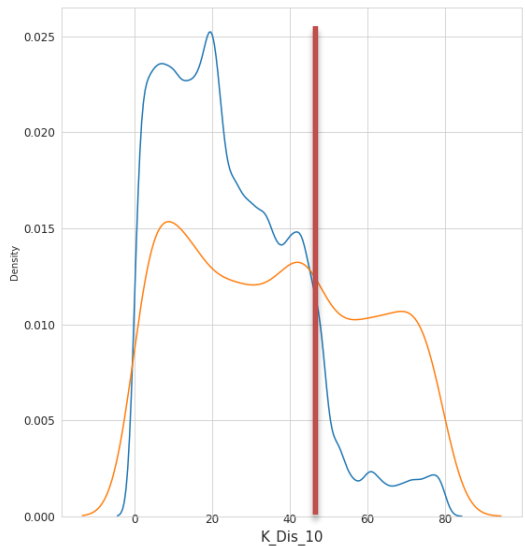
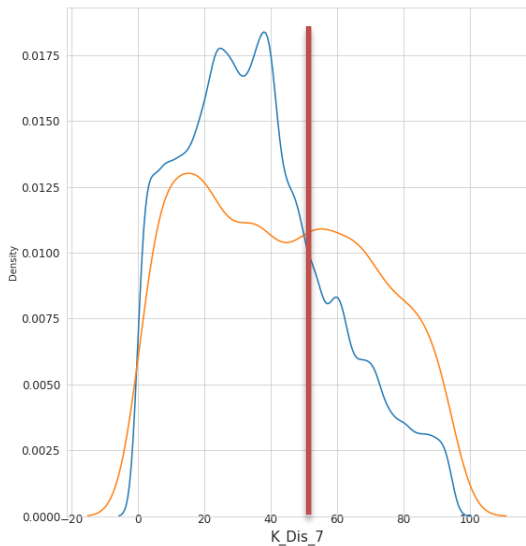
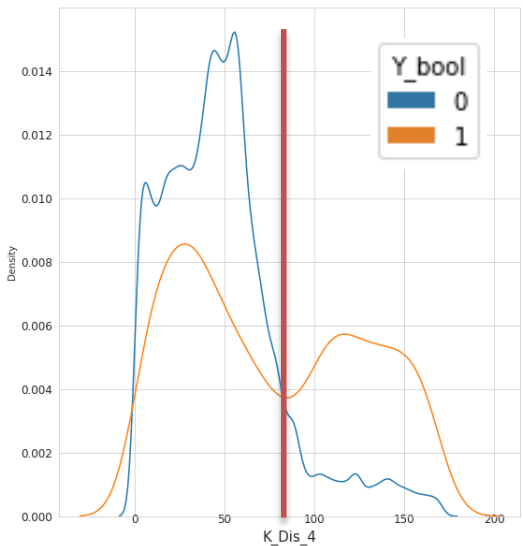
# 小型市值 Group :



Pattern一直都在,  
但應可Ensemble抓去更多  
細微特徵

	StockID	Open	High	Low	Close
count	687240.000000	687240.000000	687240.000000	687240.000000	687240.000000
mean	3440.647349	25.151667	25.275494	24.682782	24.811403
std	2258.812614	15.277821	15.356455	14.994905	15.072366
min	50.000000	0.067640	0.067640	0.050730	0.050730
25%	2009.000000	13.068725	13.137800	12.820600	12.900000
50%	2537.000000	21.049800	21.156450	20.647100	20.750000
75%	3711.000000	34.300000	34.482800	33.669300	33.839100
max	9958.000000	71.000000	71.600000	67.115400	67.840300

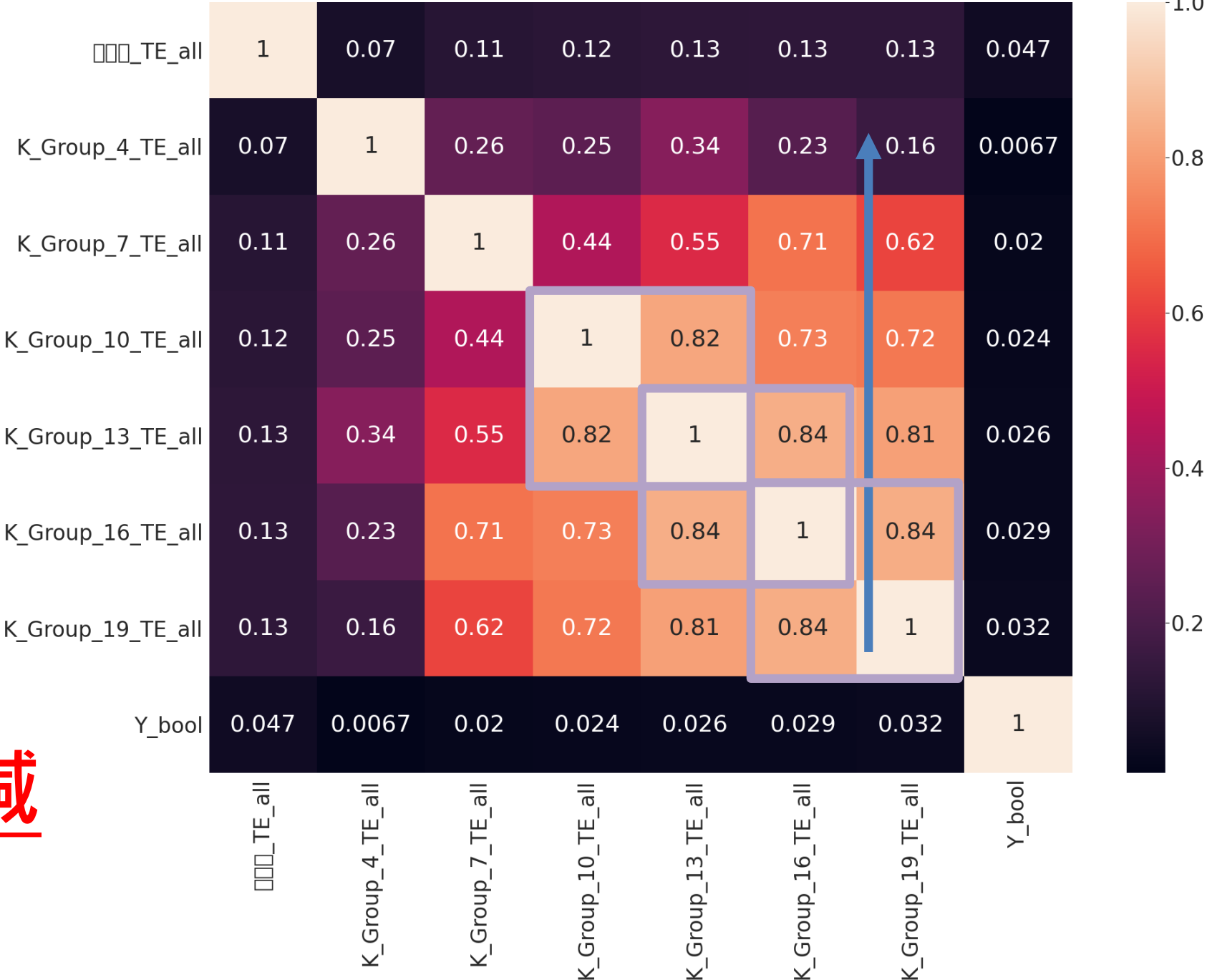
**f(不同Group|K Dis 4)**



# Grouping Correlation

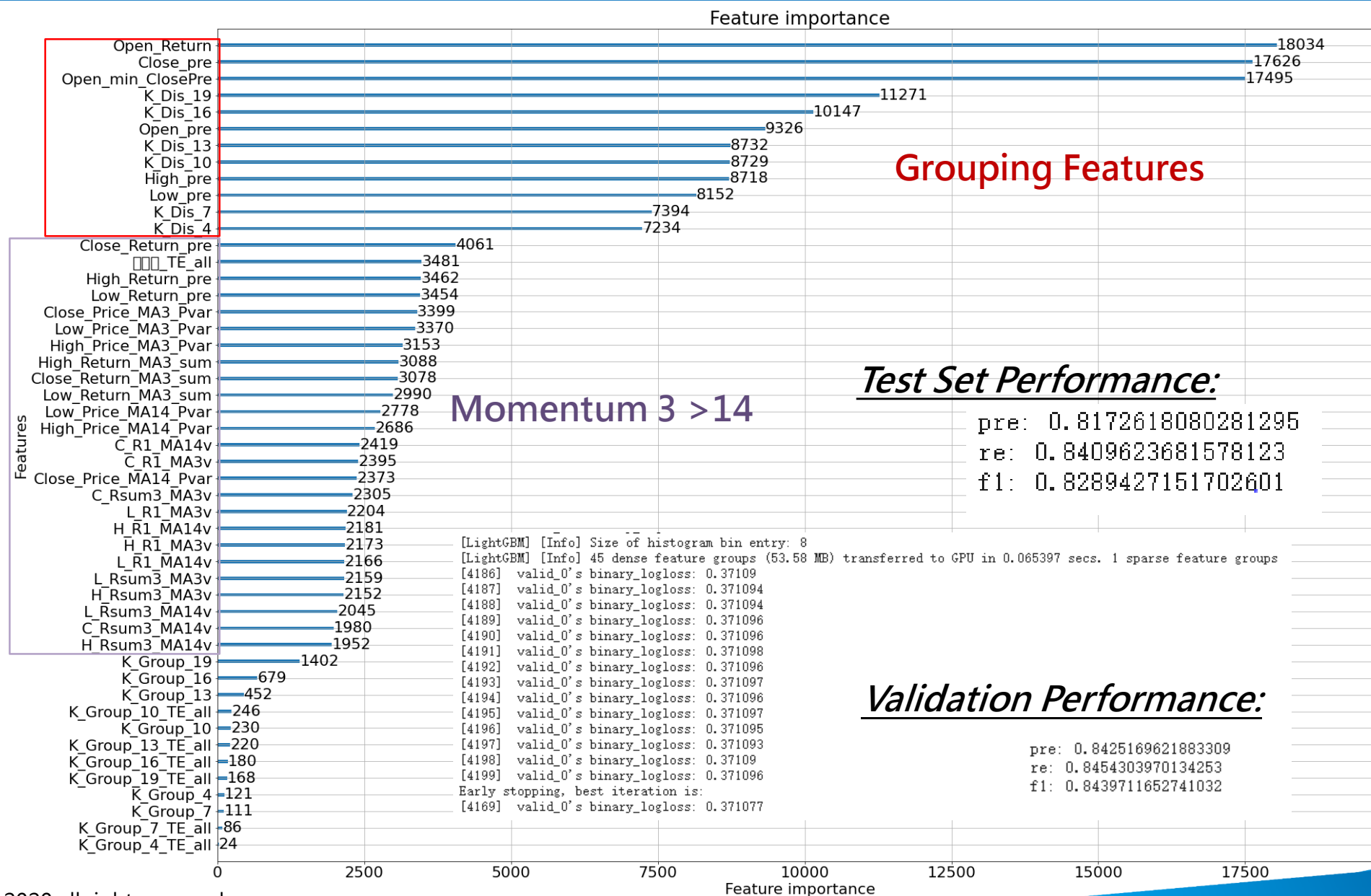
10,13,16,19 Encode Value  
線性強度高(基本上是遞減)

做維度縮減



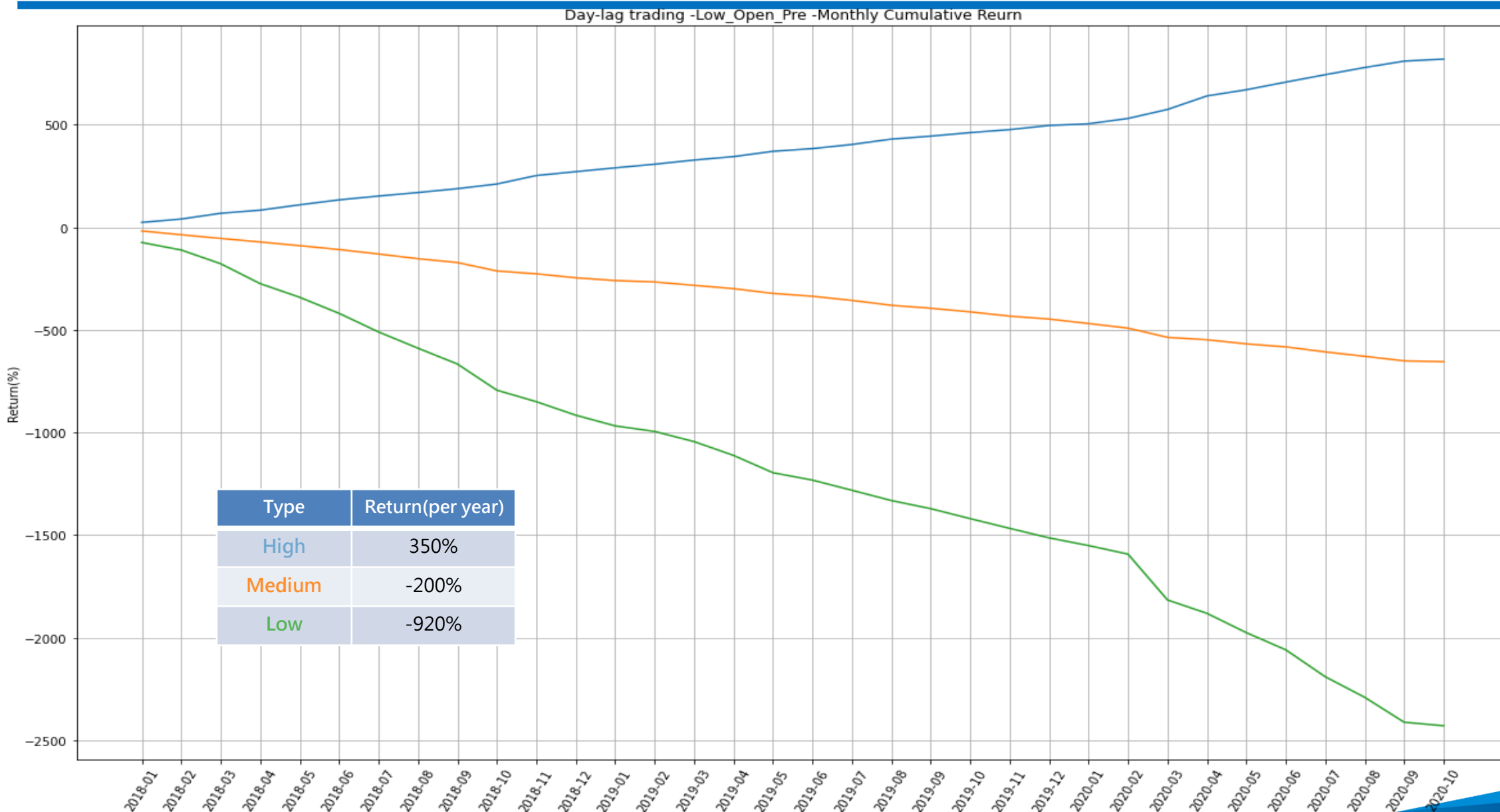


# Evaluation of Model



# 專案1-2. 隔日沖銷實測績效

Low-Cost(Open)





# Machine Learning

Dataset

Competition

Discussion

## 專案2. Fraud Detection



# 專案2. 信用卡盜刷偵測海報(異常偵測任務)

Assume

## 對比賽做了哪些假設?

-消費面切入

Features

## 基於信用卡特性做了哪些特徵?

-抓出異常差異



視卡號為最小觀察個體

-即使一人多卡，也會因活動模式不同而產生不同的消費模式



經常性消費居多

-消費金額集中在NTD3000內



分析玉山信用卡商品類別

-利用歷史消費偏好(mcc,國外消費,網購消費)作為卡片類別客戶貼標依據



特定通路任務有加碼，幹嘛去刷其他通路?

-統計每張卡號在此筆消費以前，過去消費紀錄之特店代號眾數類別



功能卡的消費金額容易有級距上的差異?

-計算本筆消費金額與過去平均消費金額之差異



消費分數差異!

-統計常用消費組合並依Baseline Model之特徵重要度給予權重

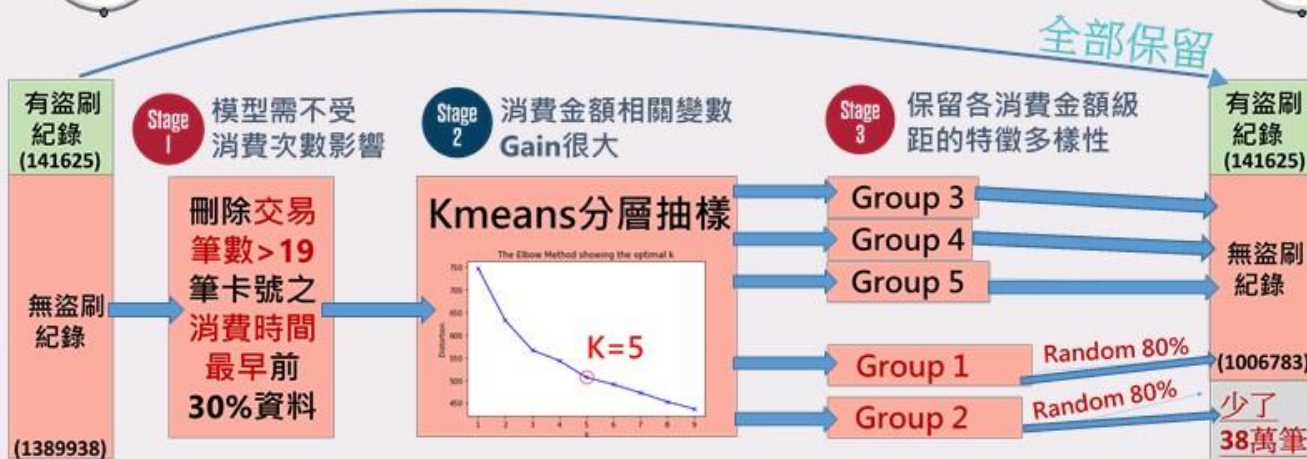
How

## 如何處理不平衡資料?

-對無盜刷紀錄資料做  
Undersampling

Model

## 編碼,stacking & RIPPER 演算法觀察分裂規則

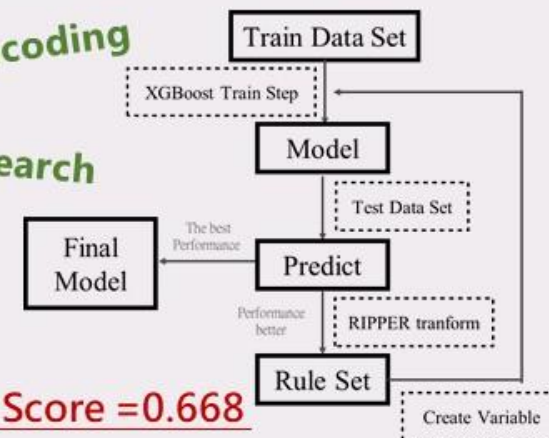


Target Encoding

GridSearch

Ripper

F1 Private Score =0.668



QR Code:







# Machine Learning

Dataset

Competition

Discussion

## 專案3. Sentiment Analysis

# 【Shopee Code League】2020 蝦皮數據競賽系列賽參賽心得&亞 太區15th做法分享



Ethan Wu Aug 14, 2020 · 10 min read



QR Code:



# 專案4. – Source Code

-by 台灣梯度下降第一品牌

原始資料

翻譯成英文

資料清洗

Emoji

😊 → smile

移除重複字元

Thhhheeee beeesst goooood → The best good

刪除噪音

~~Mjnmnmnmmsdffr~~

Text Cleansing

評論詞情緒抽取

Step 1:

刪除詞長度大於2的單詞，  
除了no這個字

Step 2:

計算各ranking評論中前2000個  
最常出現的詞，把每個評論僅用  
這2000個詞代表

Step 3:

用詞在該ranking出現的頻率當  
作詞的機率密度函數，對各個  
review進行加總/平均/變異數分  
析，抽樣並取出各ranking下，  
具代表性的review

Text Extraction

Text Augumentation

樣本增強

同意詞/反意詞增強  
在樣本少的類別，解  
決不平衡資料問題



BERT

TTA

XGB

Predict

Modeling

QR Code:







# Machine Learning

Dataset

Competition

Discussion

## What is LeetCode?



# LeetCode Example 1

LeetCode Day 16 Explore new Problems Mock Contest Discuss Store

Description Solution Discuss (999+) Submissions

Success Details >

Runtime: 44 ms, faster than 94.77% of Python3 online submissions for Two Sum.

Memory Usage: 15.4 MB, less than 22.28% of Python3 online submissions for Two Sum.

Next challenges:

3Sum 4Sum Two Sum III - Data structure design  
Subarray Sum Equals K Two Sum IV - Input is a BST  
Two Sum Less Than K

Show off your acceptance:



dict (iter equal to n, but less memory usage)

```
1 class Solution:
2     def twoSum(self, nums, target):
3         s = set(nums)
4         for i, n in enumerate(nums):
5             r = (target - n)
6             if r in s and nums.index(r) != i:
7                 return([i, nums.index(r)])
8
```

Example 1:

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Output: Because nums[0] + nums[1] == 9, we return [0, 1].

# LeetCode Example 2

## LeetCode 560. Subarray Sum Equals K

Hint:  $\text{Sum}[i:j] = \text{Sum}[0:j] - \text{Sum}[0:i]$ , for  $j > i$

1	1	1	2	1	2	1
---	---	---	---	---	---	---

思路: 依次遍歷數列，並在每次遍歷到新數字j的時候，計算是否有與 $\text{Sum}[0:j]$ 相差為K的 $\text{Sum}[0:i]$

1	1	1	2	1	2	1
---	---	---	---	---	---	---


Sum=5

← Sum(1+1+1+2)-Sum(1)=4

← Get it Sum(1+1+1+2)-Sum(1+1)=3

所以這兩個數列中間相差的元素  
就是我們要找的Subarray

# DP Solution

 Explore <sup>Day 28</sup> Problems Mock <sup>new</sup> Contest Discuss Store

Description

Solution

Discuss (737)

Submissions

Success [Details >](#)




Runtime: 124 ms, faster than 99.20% of Python3 online submissions for Subarray Sum Equals K.

Memory Usage: 16.4 MB, less than 23.53% of Python3 online submissions for Subarray Sum Equals K.

Next challenges:

[Continuous Subarray Sum](#) [Subarray Product Less Than K](#)

[Find Pivot Index](#) [Subarray Sums Divisible by K](#)

Show off your acceptance:   


Python3

Autocomplete

```
1 class Solution(object):
2     def subarraySum(self, nums, k):
3         d = collections.defaultdict(int)
4         d[0] = 1 #for first subarray k=0=k
5         tmp_sum = 0; res = 0
6         for i in range(len(nums)):
7             tmp_sum += nums[i]
8             if tmp_sum - k in d:
9                 res += d[tmp_sum - k]
10            d[tmp_sum] += 1
11        return res
```

# It's Your Turn!


<https://www.kaggle.com/c/scl-2021-da/overview>

 InClass Prediction Competition

## Shopee Code League - Multi-Channel Contacts

Data Analytics Challenge

964 teams · 15 days ago



[Overview](#) [Data](#) [Code](#) [Discussion](#) [Leaderboard](#) [Rules](#)

Try to apply dfs /Union Find Algorithms on this dataset!

Overview

[Description](#)  
Examples  
Evaluation  
Partners

### Background

Customer service is an important element of the Shopee business, as providing a good service for our customers end-to-end is critical for business growth and brand image. Our goal is to resolve the customer's issue within the least amount of time while requiring the least amount of customer effort.

One measure for customer effort is the number of times a customer has to approach customer service over a particular issue, this is also known as the metric "Repeat Contact Rate" or RCR. Shopee is interested in studying the RCR in order to improve the effectiveness of our customer service.

Customers can contact customer service via various channels such as the livechat function, filling up certain forms or calling in for help. Each time a customer contacts us with a new contact method, a new ticket is automatically generated. A complication arises when the same customer contacts us using different phone numbers or email addresses resulting in multiple tickets for the same issue. Hence, our challenge here is to identify how to merge relevant tickets together to create a complete picture of the customer issue and ultimately determine the RCR.



附件

# Learning Source

---

- Algorithm : Leetcode/ Hackerearth
- Data Science/ Data Analysis :Udemy/Coursera/ Kaggle

# Appendix – Shopee Code League 2021 & Shopee Taiwan



Hi 台灣鬼滅之刃線上看,

Congratulations! Your team has made it to the top 30 teams in the Open Category for Shopee Code League 2021.

Please check and verify your scores by **26 Mar 2021, 3PM (GMT+8)**. Your scores will be considered confirmed after this date & time.

Competition	Raw Score	Normalised Score
Data Analytics	0.95326	100
Data Science	0.5988	85.35872618
Programming	4.5	4.5
Total Score		189.8587262

Thank you!

Best regards,

Shopee Code League Team



InClass Prediction Competition

[Open] I'm the Best Coder Challenge! 2020

Open Category

60 teams · a month ago

Overview

Data

Notebooks

Discussion

Leaderboard

Rules

Team

My Submissions

Late Submission

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
output (3).csv	19 hours ago	0 seconds	0 seconds	0.70622

Complete

[Jump to your position on the leaderboard](#)

Public Leaderboard

Private Leaderboard

The private leaderboard is calculated with approximately 90% of the test data.  
This competition has completed. This leaderboard reflects the final standings.

Refresh

#	Δpub	Team Name	Notebook	Team Members	Score	Entries	Last
📍		answer.csv			1.00000		
1	—	開車了還有誰沒上車/台股崩盤/...			0.69794	13	1mo
2	▲ 1	—Lue相挺			0.69278	4	1mo
3	▲ 1	JAP			0.69179	4	1mo
4	▼ 2	Robust			0.68989	9	1mo
5	—	企業規模下降第一品牌			0.68861	5	1mo