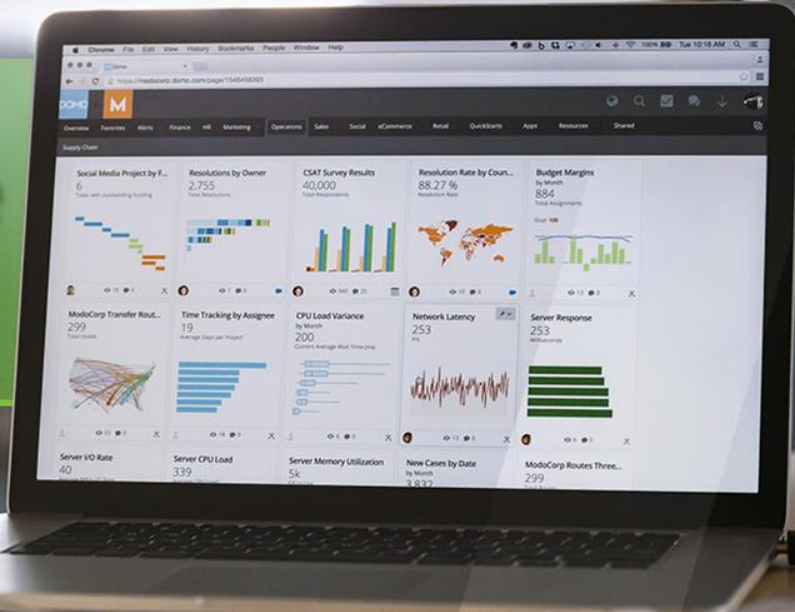


# How can data science be used in Insurtech?

*Ethan Wu- NSLI.*



# Who am I ?

-DS over 3+ years

-ETL main engineer

-GitHub Arctic Code Contributor

-LOMA ALMI



吳胤霆  
Ethan Wu



## 聯絡資訊

Email:  
ethan1126.ilink@gmail.com

## 資格認證

Github 北極計畫貢獻者  
(Github Arctic Program  
Contributor)

進修驗管理師 (LOMA ALMI):  
280, 290, 301, 307, 320, 361,  
371

## 語言能力

中文 - 母語  
英文 - TOEIC 780

## 部落格

GitHub:  
<https://github.com/cubatlin>  
Medium:  
<https://medium.com/@ethan1126.ilink>

## 個人摘要

做為 GitHub 北極計畫貢獻者之一, Ethan 是一名透過量化方法(統計&機器學習)來解決商業問題的資料科學家。我曾在北京新浪微碼透過機器學習分析客戶留言情感分析, 也主導壽險業理賠再購模型建置&基於圖論的社群網路演算法設計。最後做為一位 Kaggle, 我也致力於解決真實世界有關異常偵測、影像辨識、自然語言處理等問題, 期盼能跟著世界持續往前。

## 人工智慧競賽成績

(2019 玉山銀行人工智慧公開挑戰賽 - 信用卡盜刷偵測)

隊名: 英雄互啄(隊長)

- 亞軍(2nd), 創意暨商業應用擂台賽。
- 全國 1%(15/1366), 模型預測準確度競賽。

(2020 Shopee Code League- 亞太區 7 國 Kaggle 競賽(公開組, 超過 3000 隊))

隊名: 台灣樹皮下降第一品牌

- 15 名, 客戶評論文字情感語意分析(自然語言任務)。
- 23 名, 跨語言商品名稱翻譯準確度大賽 (自然語言任務)。
- 23 名, EDM 客戶點擊精準行銷預測(推薦系統任務)。
- 47 名, 蝦皮商家洗評價偵測(異常偵測任務)。
- 67 名, 商品影像分類(電腦視覺任務)。

## 工作經歷

資料探勘分析師(DA), 南山人壽保險股份有限公司, 台北

08/2018 - 至今

- SAS Server 管理者。資料存取、轉換、儲存(ETL)主要工程師, 在 SAP HANA 與 SAS 系統間設計語系轉碼解決方案。治理並設計保單資料庫供超過 10 個部門使用。
- 基於圖論的家戶社群網路演算法工程師。
- 協助台灣大學學生於產學合作建置理賠再購模型: 帶領台灣大學學生分析 2020 COVID-19 下壽險市場趨勢。
- 透過 Tableau 繪製 CEO 儀錶板。

實習生, 新浪微博總部產品部門, 北京

07/2017 - 08/2017

- 文字情感探勘- 分析客戶於新浪微博 App 上留言情緒。
- 新浪微博廣告投放損失函數設計演討會參與者。

## 教育背景

企業管理碩士- 國立中山大學, 高雄, 2016-2018

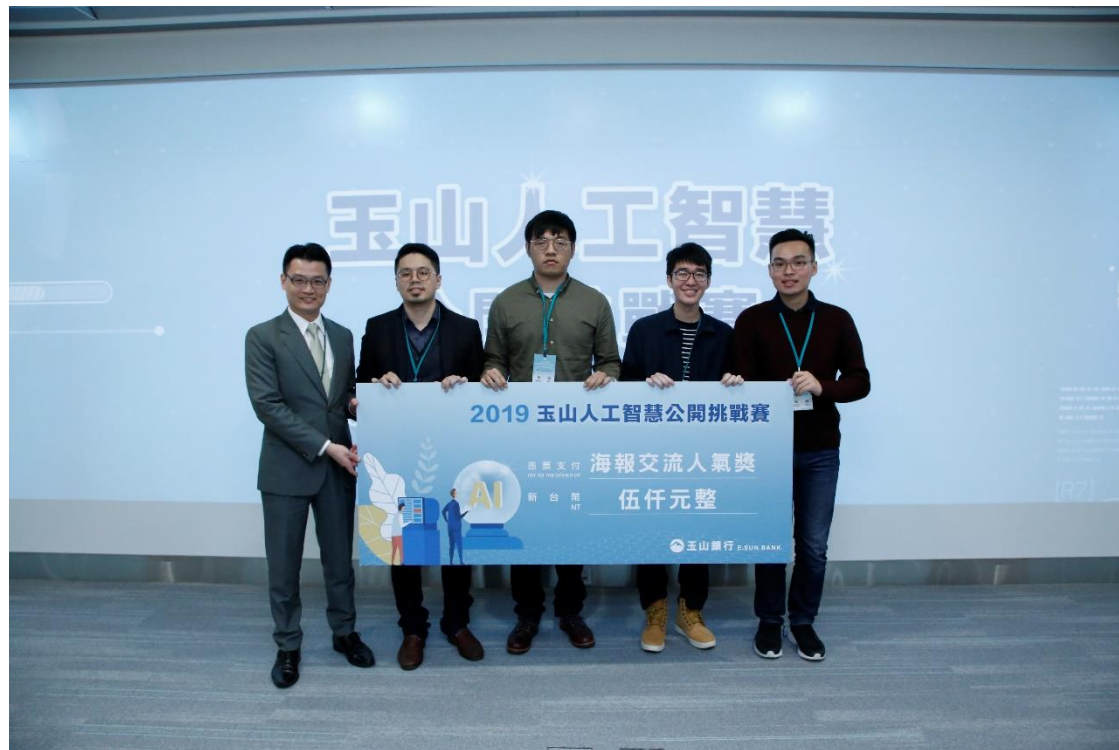
經濟學國際交換生- 武漢大學, 武漢, 2017

經濟學學士- 東海大學, 台中, 2012 - 2016

## 技術能力

程式語言: Python, R, SQL, SAS  
資料視覺化: Tableau, Matplotlib, Seaborn, Plotly, ggplot  
機器學習方法: XGBoost, LightGBM, VGG16, XceptionNet, EfficientNet, Autoencoder, BERT, Kmeans, KNN, KDE, DFS, Seq2seq, LSTM  
Server 管理能力: SAS Management Console

# Topic 1: 2019 E.Sun AI Fall Competition



*Solutions QR Code:*



# Topic 2: 2020 Shopee Code League



## 【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 · 10 min read



*Article QR Code:*



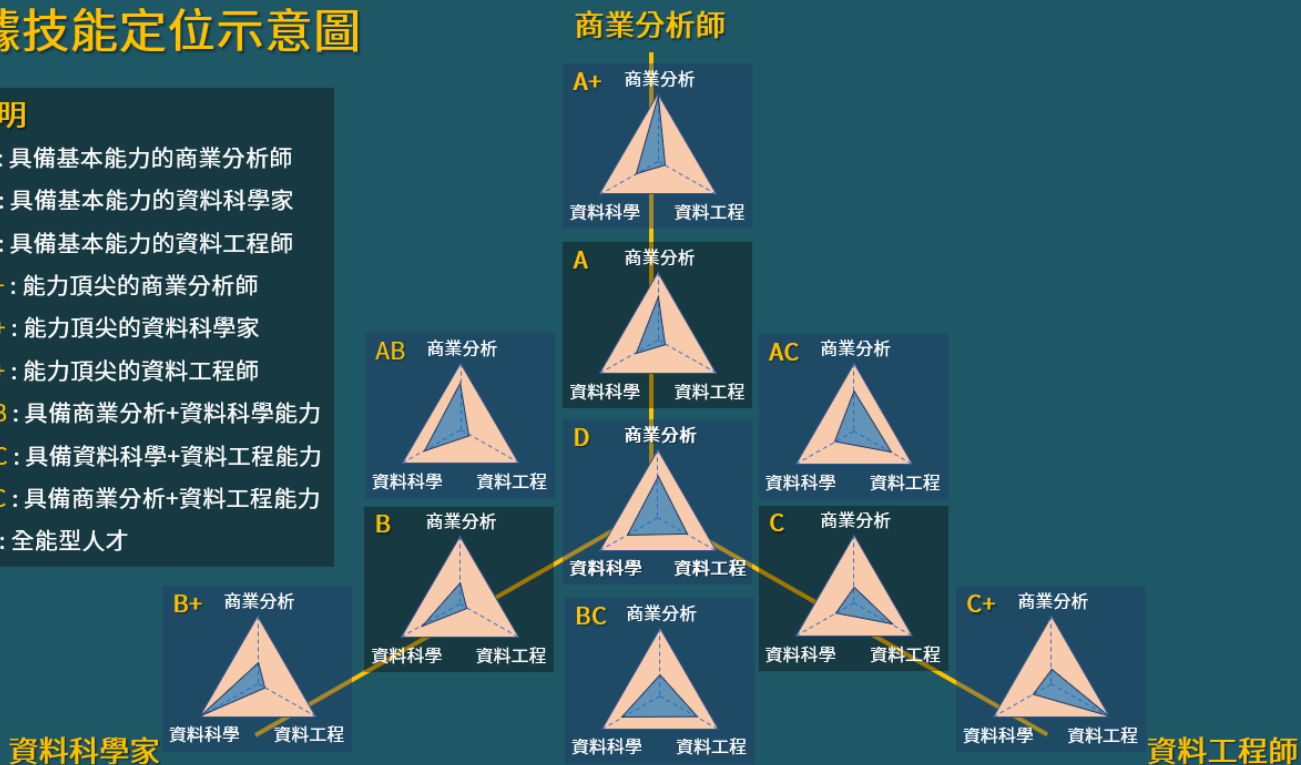
# 3 types of roles in data region



## 數據技能定位示意圖

### 說明

- A: 具備基本能力的商業分析師
- B: 具備基本能力的資料科學家
- C: 具備基本能力的資料工程師
- A+: 能力頂尖的商業分析師
- B+: 能力頂尖的資料科學家
- C+: 能力頂尖的資料工程師
- AB: 具備商業分析+資料科學能力
- BC: 具備資料科學+資料工程能力
- AC: 具備商業分析+資料工程能力
- D: 全能型人才



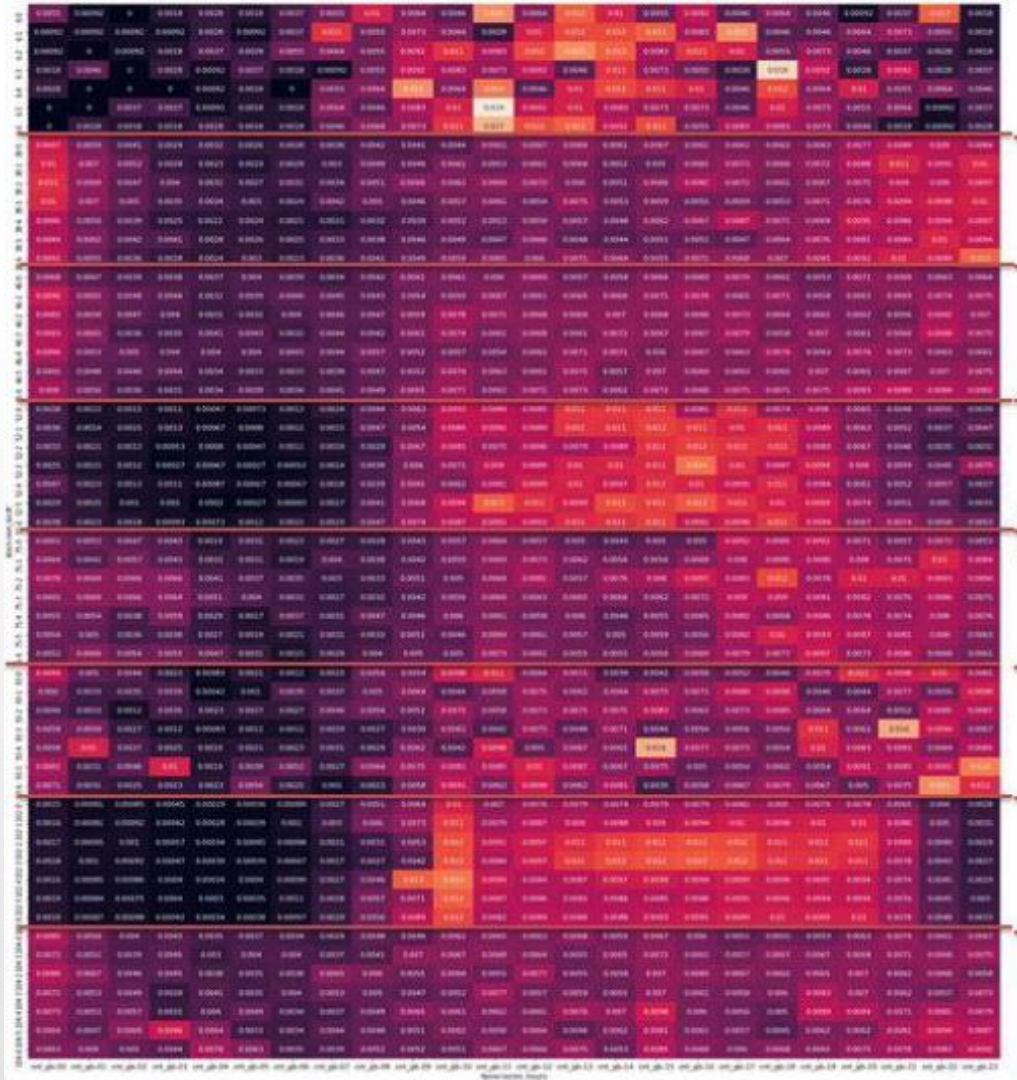


# Data Analytics-

## *Exploratory Data Analysis*

*Heatmap*

*7\*24?*



# Data Scientist –CV , NLP

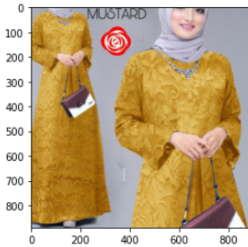
## VGG Prediction

```
In [ ]: img_path = train_path+df_train['fname_full'][1]
        print(img_path)

/content/drive/My Drive/SCL_2/shopee-product-detection-dataset/train/train_share/train/01/a0c202321099b202db6198c23f5899de.jpg
```

```
In [ ]: sample_img = load_img(img_path)
        plt.imshow(sample_img)
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x7fecde513128>
```



```
In [ ]: img = image.load_img(img_path, target_size=(224, 224))
        x = image.img_to_array(img)
        print(x.shape)

(224, 224, 3)
```

```
In [ ]: #Sample 1
        model = VGG16(weights='imagenet', include_top=True)

        img = image.load_img(img_path, target_size=(224, 224))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        x = preprocess_input(x)

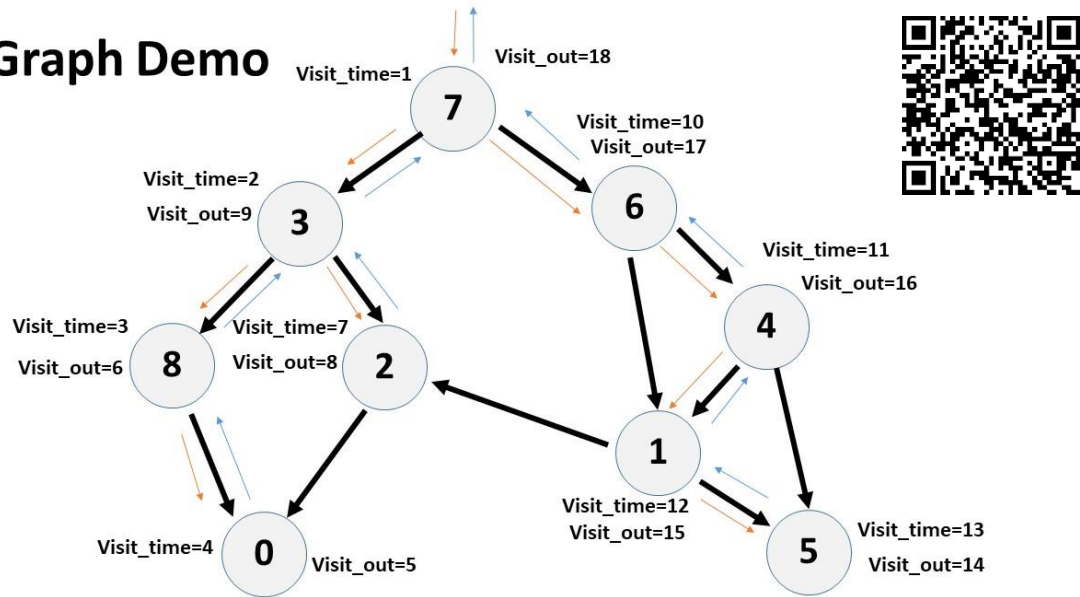
        features = model.predict(x)
        print('Predicted:', decode_predictions(features, top=5)[0])
```

<https://udn.com/news/story/7239/4253246>



# Algorithm Engineer

## Graph Demo



<https://github.com/cubatlin>


<https://udn.com/news/story/7239/4425271>





# Algorithm Engineer - LeetCode

## Demonstrate on 560. Subarray Sum Equals K

 Explore Day 28 Problems Mock new 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](#)

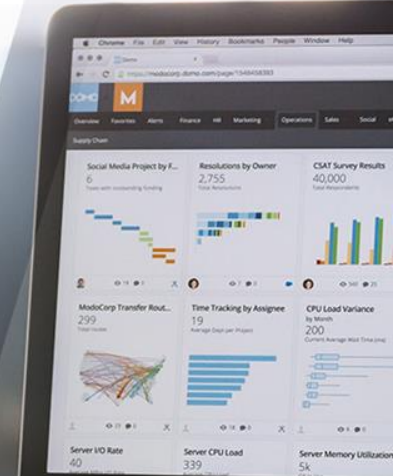
[Find Pivot Index](#) [Subarray Sums Divisible by K](#)

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
```





# Want to know more about me?

