



# Cardinal Operations

Summer Internship Report

2018.05~2018.08  
Hao Rong

- Training – Unilever Demand Prediction (week 1 ~ week 2)
- Daimler supply chain optimization problem
  - Data Wrangling, Exploration & API Request (week 2 ~ week 7) ← Client 1<sup>st</sup> Review, Jun 20th
  - Parts Demand Prediction (week 7 ~ week 9) ← Client 2<sup>nd</sup> Review, Jul 18th
  - CVXPY model implementation (week 9 ~ week 11) ← Client 3<sup>rd</sup> Review, Aug 15th



## Part 1

# Training – Unilever Demand Prediction

- Machine learning model – Static model
- LSTM model – Dynamic model



# DAIMLER



- Daimler supply chain optimization problem
  - Data Wrangling, Exploration & API Request (week 2 ~ week 7)
  - Parts Demand Prediction (week 7 ~ week 9)
  - CVXPY model implementation (week 9 ~ week 11)
- Set the project goal with the client (PDC optimization)
- Several version of model prototypes (single period, multi period, two period)
- Build the product (a visual-friendly decision interface)



← Client 1<sup>st</sup> Review, Jun 20th

← Client 2<sup>nd</sup> Review, Jul 18th

← Client 3<sup>rd</sup> Review, Aug 15th



1

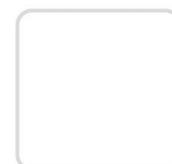


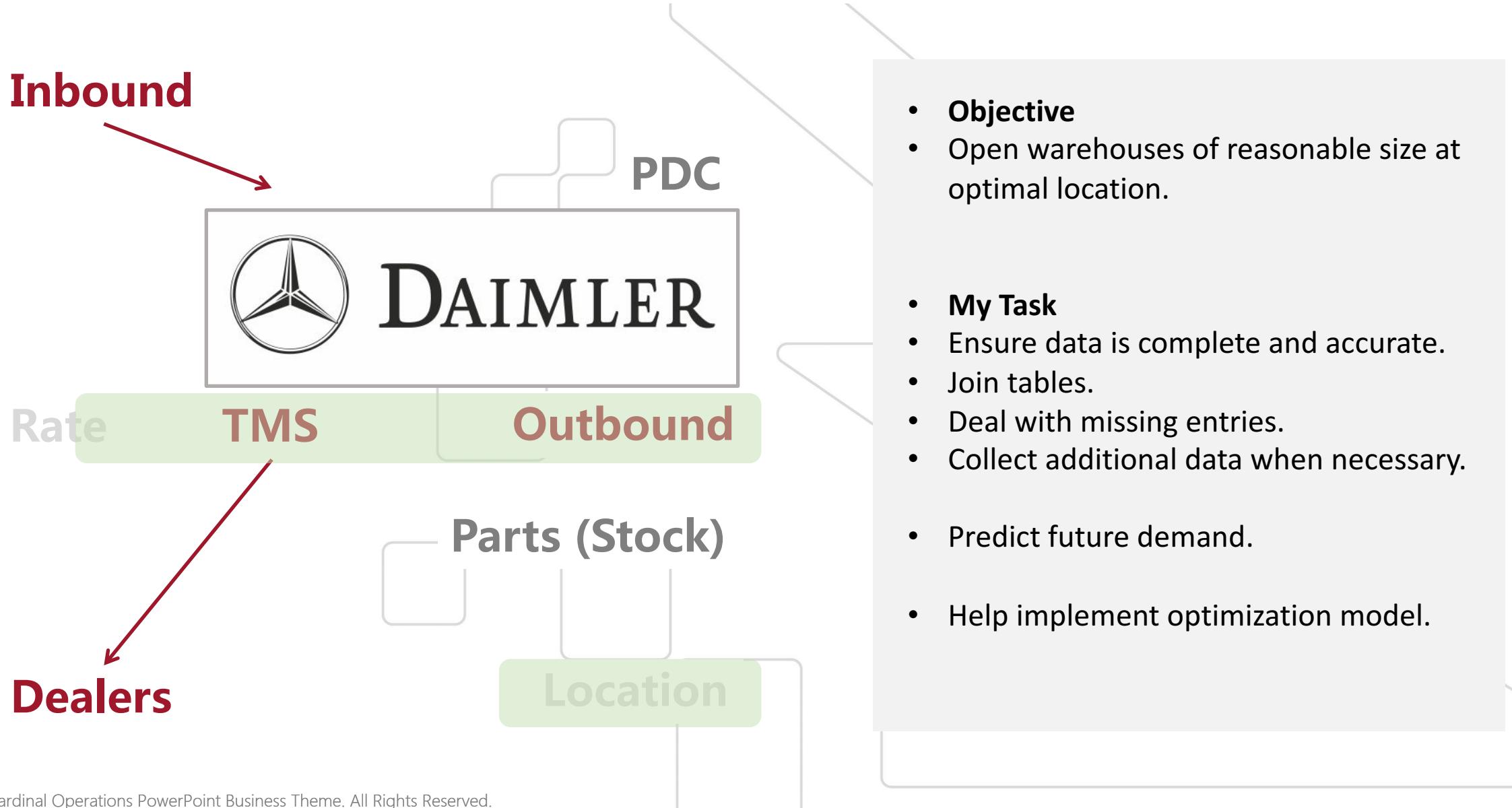
DAIMLER



## Part 2

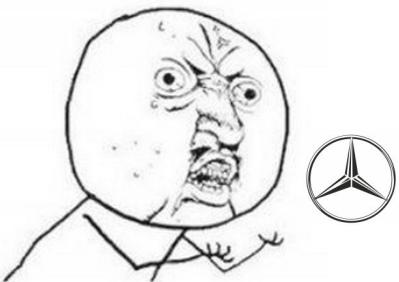
# Data Wrangling, Exploration & API Request







## Ways that data can be dirty:



Problem	Reason / Description	Solution
Multiple keys for one records	Lower case & Upper case.	Unify characters to upper/lower case.
Multiple keys for one records	Special characters inside the key.	Unify data with regular expression.
Missing data	Clients didn't output the whole dataset.	Confirm with clients. Ask for complete data.
Data of different magnitudes in one column	Inconsistency in clients' database.	Data visualization. Outliers. Check other fields.
Inconsistent primary key	Primary key is unique in every subgroup dataset. It is not unique in main dataset.	Concatenate with other fields to make it unique.
Chinese Characters VS. pinyin	Suzhou? Suzhou?	Well..... Exclude the unlikely possibilities.
Secret Meanings behind unusual keys	While case ID of goods shipping to normal dealer is ordinary length 7 string, case ID of goods shipping to other could be casual 'C001', 'C002'..... And they are repeatedly used	Ask clients about it. (They might not know....) Guess it by yourself.
Ambiguous definition of features	ICT? Facing PDC?	Ask clients.



# Data Wrangling, Exploration & API Request

数据驱动 运筹帷幄

## Amap API:

Registration

① 为地址经纬转换添加Key

\* Key名称: 支持汉字、数字、字母、下划线、中划线, 不超过15个 ② 命名规范

\* 服务平台:  Android平台  iOS平台  Web端(JS API)  
 Web服务  智能硬件  微信小程序

可使用服务: [Android地图SDK](#) [Android定位SDK](#) [Android导航SDK](#)

\* 发布版安全码SHA1: 如何获取  
请输入调试版安全码SHA1

\* PackageName: 如何获取  
请输入包名,如:com.test.sdk

阅读并同意 [高德服务条款及隐私权政策](#) 和 [高德地图API服务条款](#)

[取消](#) [提交](#)

Apply for keys

地址经纬转换-杉数科技-戴姆勒两点驾车路程9-Web服务-5c938b9a56b75953... 基础API

服务	今日调用量	调用量上限(次/日)	并发量上限(次/秒)	状态	操作
地理编码	免费	0 已用 0%	3000000	1000	正常 提升配额
逆地理编码	免费	0 已用 0%	3000000	1000	正常 提升配额
输入提示	免费	0 已用 0%	300000	200	正常 提升配额
搜索	免费	0 已用 0%	300000	300	正常 提升配额
公交路径规划	免费	0 已用 0%	300000	200	正常 提升配额
驾车路径规划	免费	0 已用 0%	300000	200	正常 提升配额
步行路径规划	免费	0 已用 0%	300000	200	正常 提升配额
骑行路径规划	免费	0 已用 0%	300000	200	正常 提升配额
行驶距离测量	免费	0 已用 0%	300000	200	正常 提升配额
抓路服务	免费	0 已用 0%	300000	200	正常 提升配额
行政区查询	免费	0 已用 0%	300000	200	正常 提升配额
交通态势	免费	0 已用 0%	300000	200	正常 提升配额
IP定位	免费	0 已用 0%	3000000	1000	正常 提升配额
坐标转换	免费	0 已用 0%	3000000	1000	正常 提升配额
静态地图	免费	0 已用 0%	6000000	1000	正常 提升配额
天气	免费	0 已用 0%	3000000	1000	正常 提升配额

Use



## How to use it?

```
1 import requests  
2  
3 def geocode(address):  
4     parameters = {'address': address, 'key': '6ec1a32b3d319596f2d05fc98e0bf07f'}  
5     base = 'http://restapi.amap.com/v3/geocode/geo'  
6     response = requests.get(base, parameters)  
7     answer = response.json()  
8     return answer['geocodes'][0]['location']
```

## Coordinates

```
1 def driveDistance(origin, destination):  
2     parameters = {'origin': origin, 'destination': destination, 'strategy': 0, 'key': '6ec1a32b3d319596f2d05fc98e0bf07f'}  
3     base = 'http://restapi.amap.com/v3/direction/driving'  
4     response = requests.get(base, parameters)  
5     answer = response.json()  
6     return answer
```

## Driving Route



## Some tips for using Amap API:

- Limited request amount per day per key! (Although we might not use that much)
  - Apply for multiple accounts
  - Apply for multiple keys
  - Apply for the commercial account (10 times more amount! a little trick)
- Save your requested data constantly while in progress
  - Every request could takes up to 2 seconds (ex. route planning)
  - Work could span more than 1 day. / Server could break.
  - Implement saving actions every N iterations (CSV, pickles)
  - Implement saving actions whenever the request amount in a key is exhausted
- Parallel requests (a lot faster even with requests upper bound)



# Data Wrangling, Exploration & API Request

数据驱动 运筹帷幄

我的应用(1)

+ 创建新应用

## 服务调用量配额说明

Key平台类型	服务	个人开发者		认证个人开发者		企业开发者 [当前]	
		日配额 (次)	QPS	日配额 (次)	QPS	日配额 (次)	QPS
地理	地理编码	6000	100	300000	200	3000000	1000
	逆地理编码	6000	100	300000	200	3000000	1000
	输入提示	2000	50	30000	50	300000	200
	搜索	2000	50	30000	50	300000	300
	公交路径规划	2000	50	30000	50	300000	200

杉数科技-戴姆勒两点驾车路程

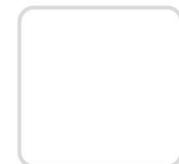
c63241db75daf91de90169daf62767ea

Web服务

设置 查看配额 删除



DAIMLER



### Part 3

## Parts Demand Prediction



# Parts Demand Prediction

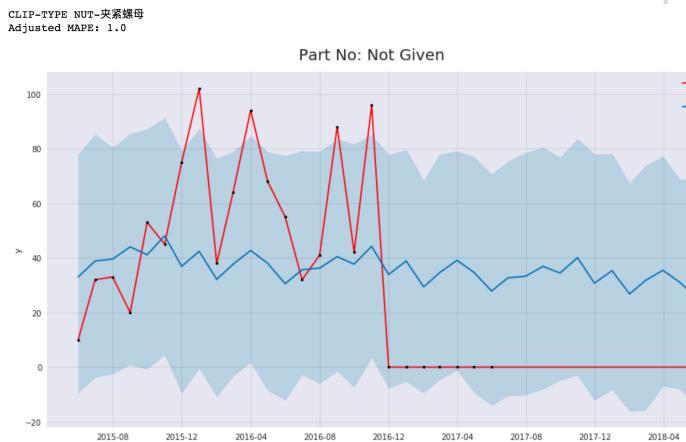
数据驱动 运筹帷幄

## Goal:

- Make reliable prediction of demand at parts level to infer total space needed for PDC.

## Problem:

- Parts of small demands usually have very unstable demand pattern.
- Even with large demands, some parts could still have burst or drop in temporal pattern.





# Parts Demand Prediction

数据驱动 运筹帷幄

## Finding Substitutes and Complements

- Check correlation of parts within certain group.....?
- Do check correlation among groups.

### Substitutes:

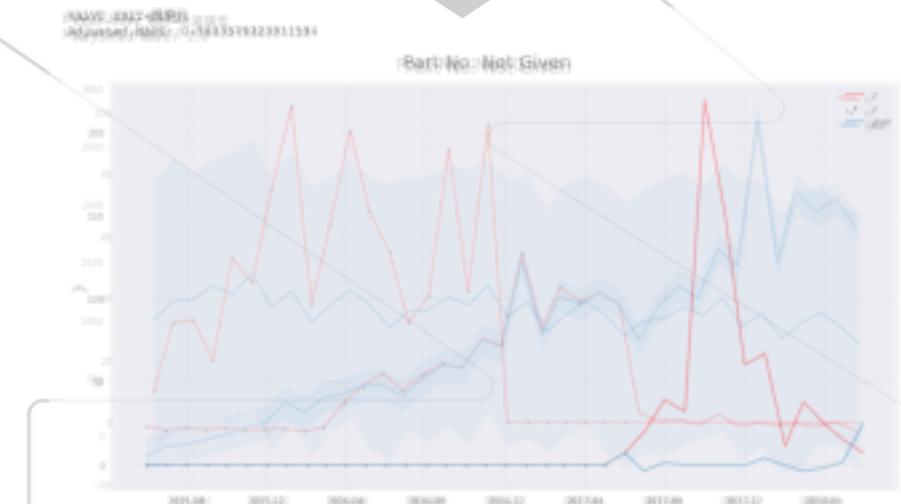
Reduce noise in demand trend.

- Clustering by parts properties.
- Clustering by description.
- Clustering for better model performance.

### Complements:

Infer trend of noisy data from trends of its complement group.

- Too little data. how can you infer complements relationship?
- Negative correlation?
- Require profound domain knowledge in car parts.
- Clustering by Jaccard Coefficient.



Substitute effect



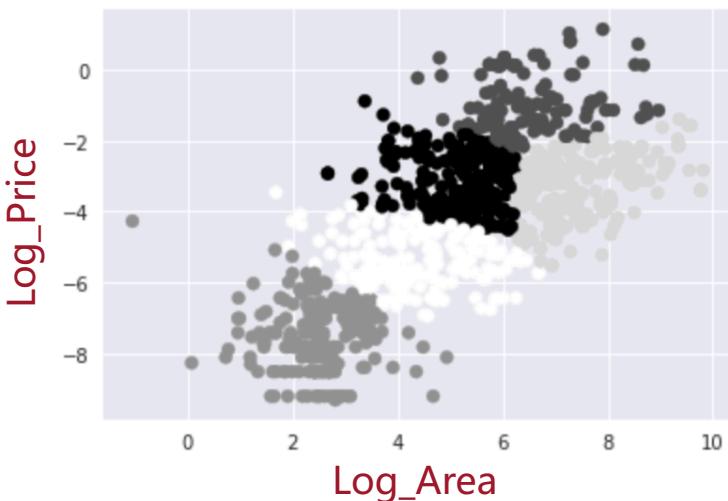
# Parts Demand Prediction

数据驱动 运筹帷幄

## Clustering by Parts Properties

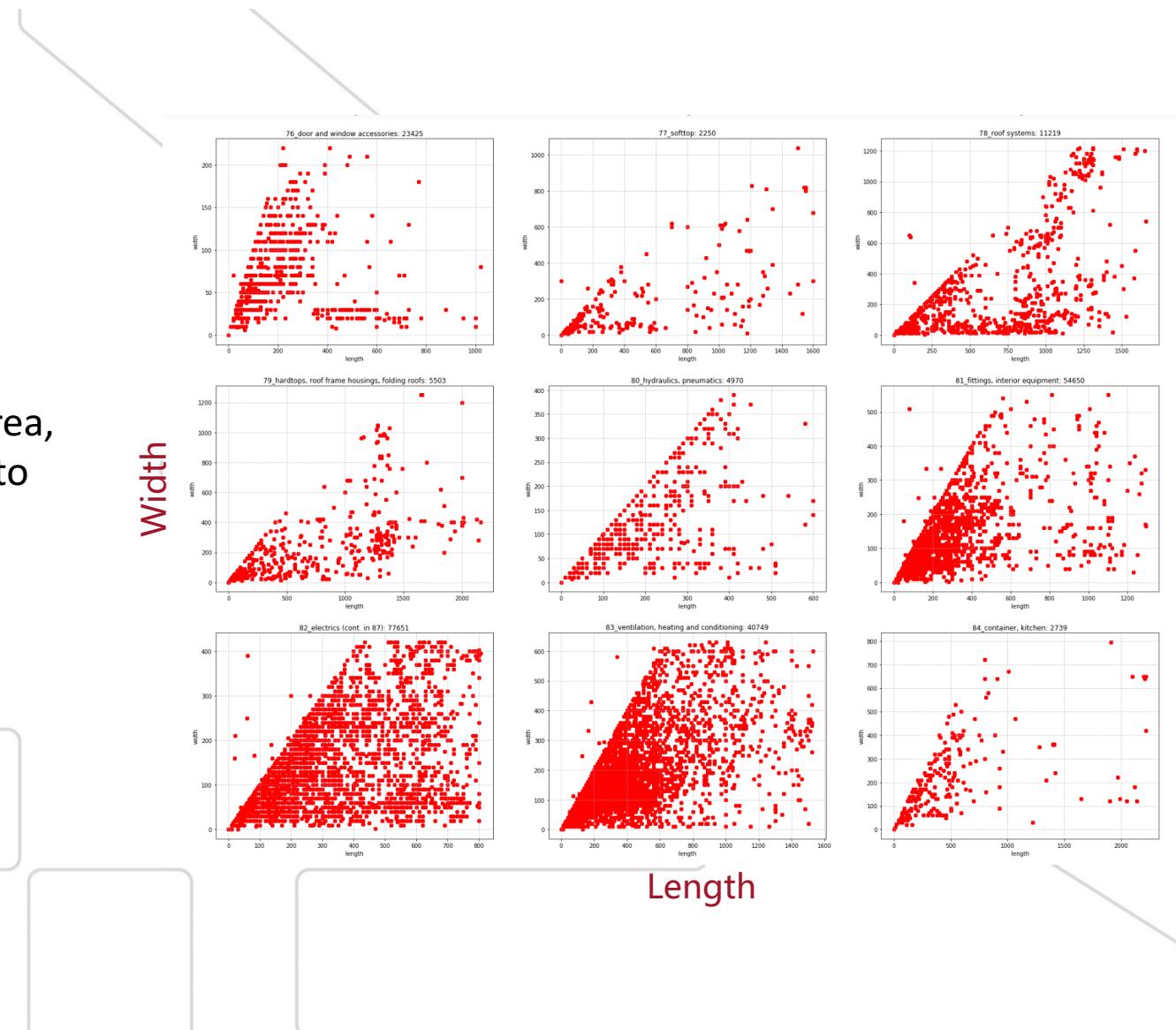
Hierarchical clustering over length, width, height.

- Does not make too much sense to me. How is demand related to parts' size?



K-means clustering regarding to price and area.

- The final properties we care about are price, area, and chargeable weight. And area is correlated to chargeable weight.





## Clustering by description

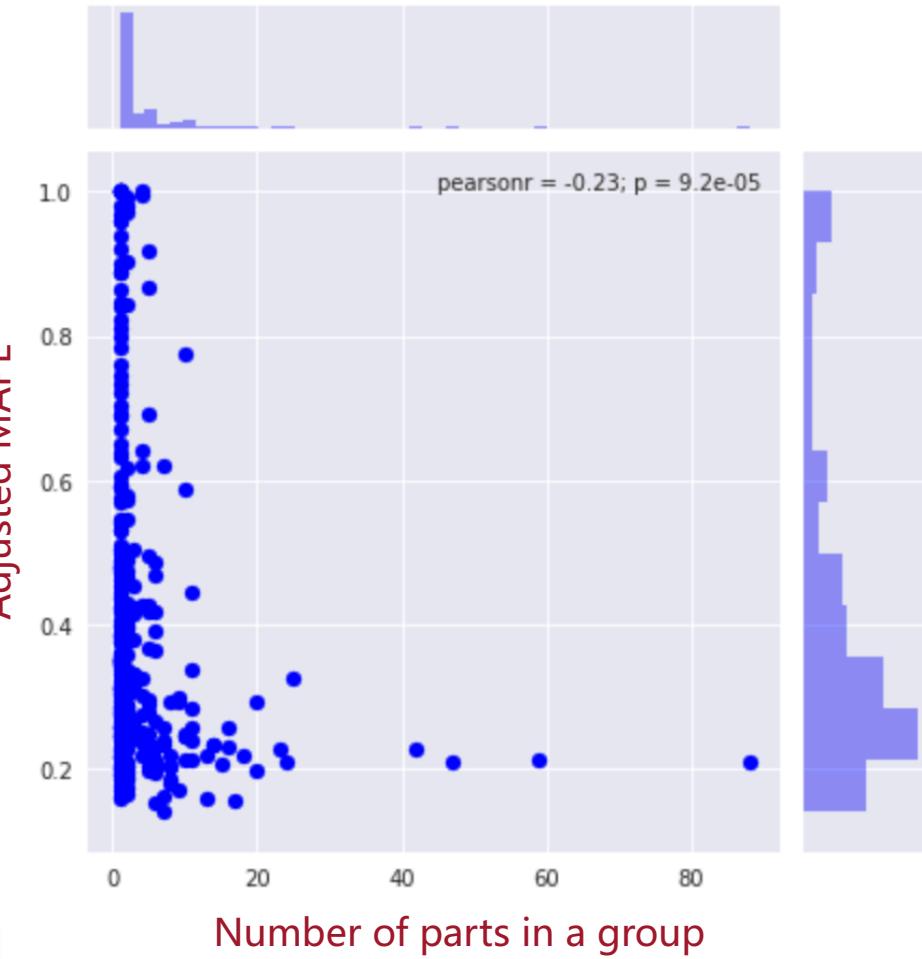
- A good indication of substitutes.
- Clusters that aggregate more parts tend to have better model performance.

	group_name
108	SEALING ADHESIVE-密封胶粘剂
177	BATTERY-蓄电池
263	GROMMET-固定索环
176	PAN HEAD SCREW A WASHER-平头螺栓
84	PROFILE MOUNTING-成型固定框架
23	CLIP-TYPE NUT-夹紧螺母
153	SCREW W HEXAGON SOCKET-埋头螺栓
188	OPERATOR'S MANUAL-操作手册
28	CORE BUNDLE SEALING-电缆线圈密封
223	V-RIBBED BELT-多楔带

### Furthermore:

- Clustering by similar descriptions.
- Descriptions that share same words.

Group Method: ENCN  
Direct Mean: 0.40878040103908847  
Weighed Mean: 0.30490539158562413





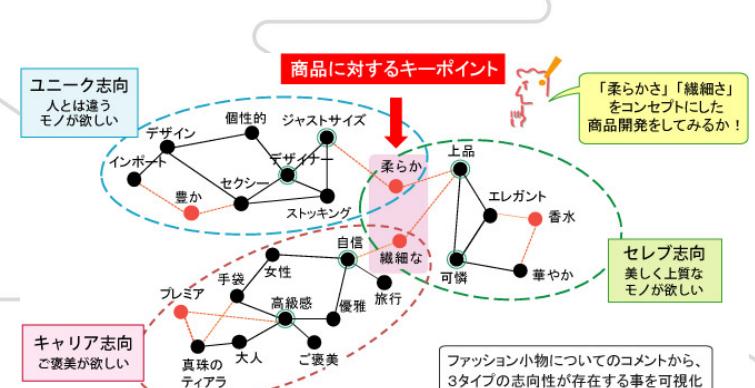
## Clustering by Jaccard Coefficient

Construct the clusters of high frequency items and the pairs

- ▶ Those items with high occurrence frequency are represented as black nodes in graph G.
- ▶ The item-pairs that co-occur in the same sentences are then identified, and the item-pairs are sorted by their occurrence frequency. item-pairs are represented as black solid lines in graph G.
- ▶ Jaccard Coefficient:

$$J(I_i, I_j) = \frac{Freq(I_i \cap I_j)}{Freq(I_i \cup I_j)}$$

- Failed at first. Cannot find co-occurrence because I used co-occurrence in same case.
- Should use co-occurrence in same order.





## Optimize clustering for better model performance

After few trials of clustering heuristics, we have a handful of methods can be applied to clustering.

A way to optimize our current process is to learning a combination of clustering methods regarding a specific objective, which in our case is model performance.

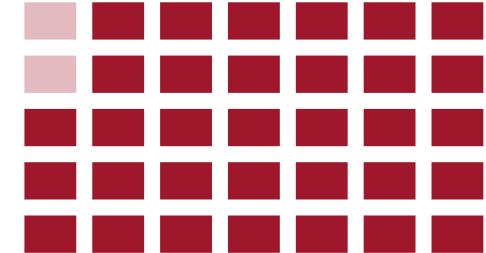
```
if performance(cluster(A, B)) > mean(performance(A), performance(B)):  
    keep cluster(A, B)
```

.....

### Drawback:

- Evaluating model performance is a very slow process.
- Maybe there are better evaluation methods?

Clustering by description



K-means clustering

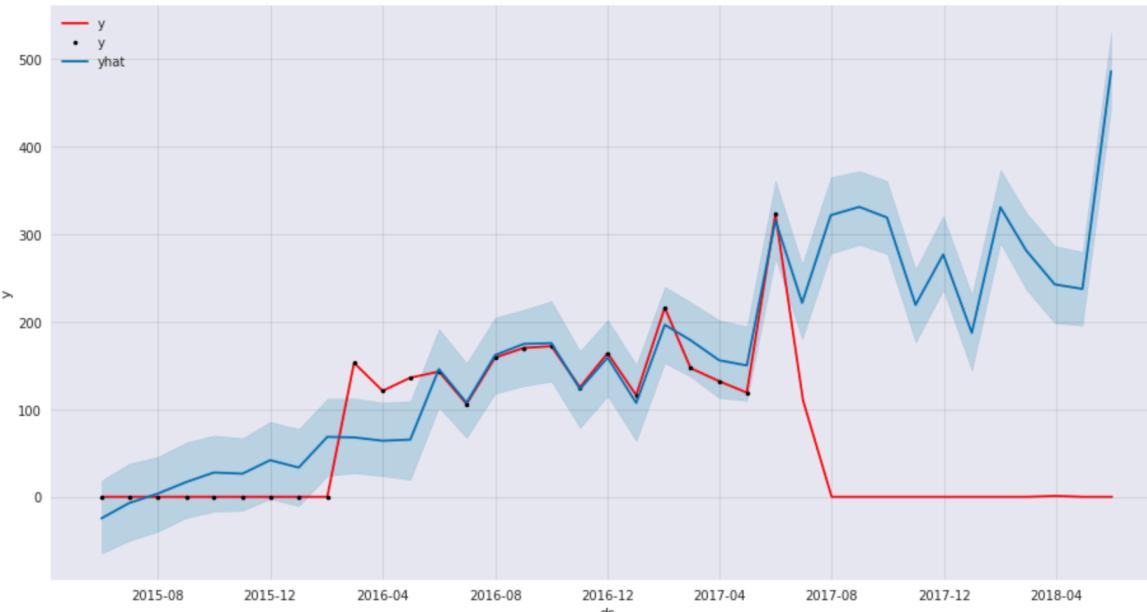




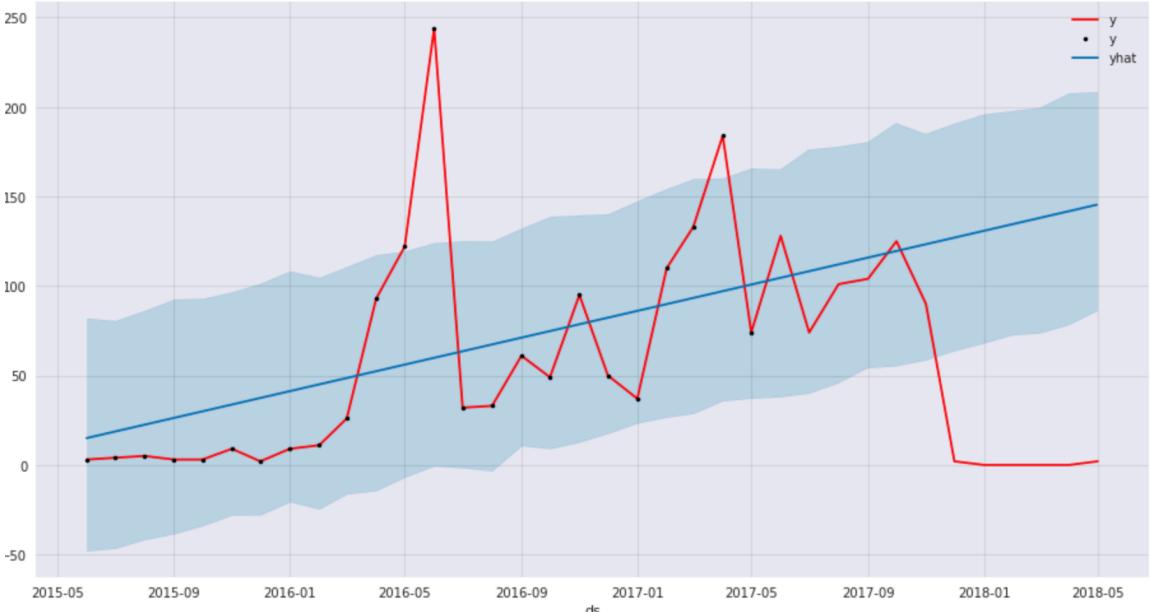
## Prediction model: (predict 12 month)

- FB prophet: 0.16 adjusted MAPE on Area demand
- ARIMA (Echo): similar adjusted MAPE
- holts winter forecasting: Data does not have obvious seasonality

### Weird thing in using FB prophet



Training set: 24 month / Test set: 12 month



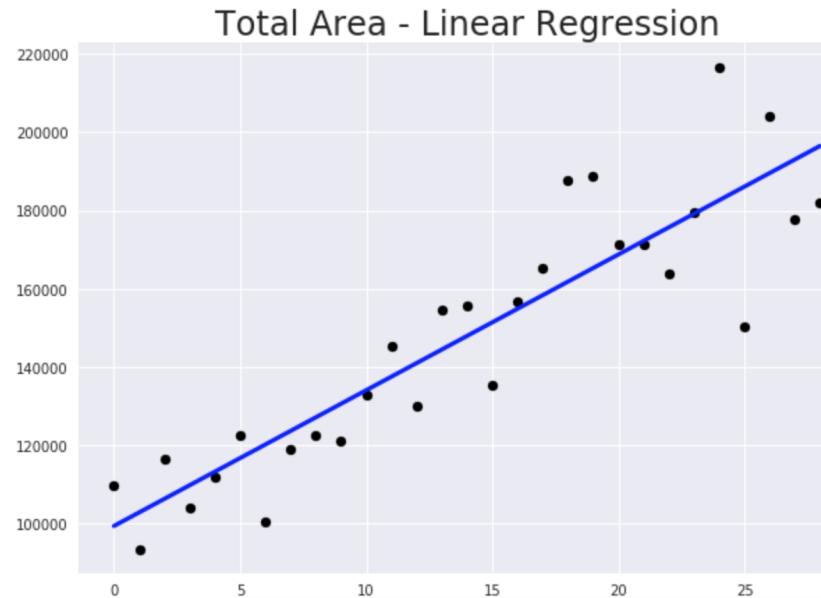
Training set: 23 month / Test set: 12 month



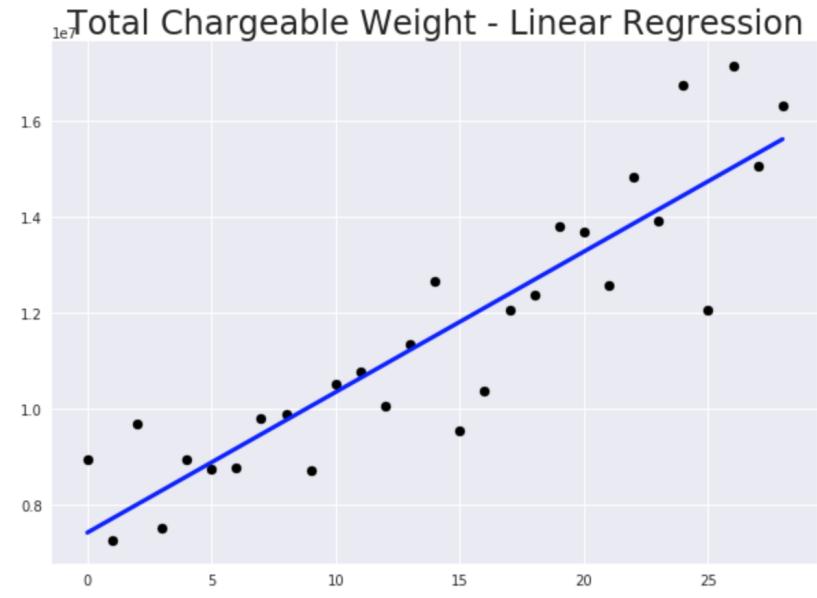
## What we use at last:

- Linear regression on country level demand.
- For robustness.
- For simpler optimization model formulation.

Coefficients:  
[3471.58520311]  
Intercept:  
99272.16426454959



Coefficients:  
[293095.47783742]  
Intercept:  
7421163.756482931

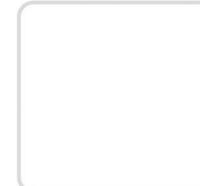




DAIMLER



## Part 4 CVXPY model implementation





## A brief introduction:

```
import cvxpy as cvx

# Create two scalar optimization variables.
x = cvx.Variable()
y = cvx.Variable()

# Create two constraints.
constraints = [x + y == 1,
                x - y >= 1]

# Form objective.
obj = cvx.Minimize((x - y)**2)

# Form and solve problem.
prob = cvx.Problem(obj, constraints)
prob.solve() # Returns the optimal value.
print("status:", prob.status)
print("optimal value", prob.value)
print("optimal var", x.value, y.value)
```

```
status: optimal
optimal value 0.999999999761
optimal var 1.0000000001 -1.19961841702e-11
```

- **Variables**
- **Parameters**
- **Constraints**
- **Objective**
- **Problem**

**Result:** **Problem.attributes**  
**Variables.value**



# CVXPY model implementation

数据驱动 运筹帷幄

决策变量	$x_{ijt}, i \in \mathcal{I}, j \in \mathcal{J}$	如果 $i$ Dealer 在 $t$ 周期选择 $j$ 仓库运输则为 1, 否则为 0
$z_{jm}^k, j \in \mathcal{J}^n, m \in \mathcal{M}, k \in \mathcal{T}^k$		如果仓库 $j$ 在 $k$ 周期的开始开放且面积为 $m$ 则为 1, 否则为 0
$z_j^k, j \in \mathcal{J}^e, k \in \mathcal{T}^k$		如果仓库 $j$ 在 $k$ 周期的末尾关闭则为 1, 否则为 0
已知参数		
$m'_j$		$j$ 仓库的已知面积, $j \in \mathcal{J}^e$
$s_{ij}$		$i$ Dealer 到 $j$ 仓库的距离矩阵
$dm_{it}$		$t$ 周期 $i$ Dealer 所需求的面积总量
$dh_{it}$		$t$ 周期 $i$ Dealer 所需求的价格总量
$w_{it}$		$t$ 周期 $i$ Dealer 需求的计费重量总量

$$\min_{\{x_{ijt}, z_{jm}^k, z_j^k\}} \sum_{j \in \mathcal{J}^n} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{T}^k} (F_m + (|T| - k + 1)m\alpha) z_{jm}^k + \sum_{j \in \mathcal{J}^e} \sum_{k \in \mathcal{T}^k} (f + km'_j\alpha) z_j^k + \sum_{j \in \mathcal{J}^e} |T|m'_j\alpha(1 - \sum_{k \in \mathcal{T}^k} z_j^k) \quad (1)$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} w_{ijt}^l r_{ij}^l s_{ij} x_{ijt} + \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} dh_{it} x_{ijt} \omega / trm$$

$$\text{subject to } \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{T}^k} z_{jm}^k \leq 1, \forall j \in \mathcal{J}^n \quad (2)$$

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}^n} z_{jm}^k \leq 1, \forall k \in \mathcal{T}^k \quad (3)$$

$$\sum_{k \in \mathcal{T}^k} z_j^k \leq 1, \forall j \in \mathcal{J}^e \quad (4)$$

$$\sum_{j \in \mathcal{J}} x_{ijt} = 1, \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (5)$$

$$\beta \sum_{i \in \mathcal{I}} dm_{it} x_{ijt} / trm \leq \sum_{k \in \mathcal{T}^k: k \leq t} \sum_{m \in \mathcal{M}} mz_{jm}^k, \forall j \in \mathcal{J}^n, t \in \mathcal{T} \quad (6)$$

$$\beta \sum_{i \in \mathcal{I}} dm_{it} x_{ijt} / trm \leq m'_j(1 - \sum_{k \in \mathcal{T}^k: k < t} z_j^k), \forall j \in \mathcal{J}^e, t \in \mathcal{T} \quad (7)$$

$$x_{ijt} \in \{0, 1\}, \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \quad (8)$$

$$z_{jm}^k \in \{0, 1\}, \forall j \in \mathcal{J}^n, m \in \mathcal{M}, k \in \mathcal{T}^k \quad (9)$$

$$z_j^k \in \{0, 1\}, \forall j \in \mathcal{J}^e, k \in \mathcal{T}^k \quad (10)$$

• Variables

• Parameters

• Objective

• Constraints



## Somewhat advance tricks:

- Objectives can be broken into pieces

```
# warehouse bursting fee  
bursting_fee = cvx.sum(m_tilde * phi)  
  
# change penalty fee  
change_penalty = sum(cvx.sum(m) for m in c[:-1])  
  
obj_func = nonexistent_open_fee + nonexistent_rental_fee + exist_close_fee + exist_closing_rental_fee + exist_not_closing_rental_fee +\\  
Transport_fee + exist_expand_fee + expand_rental_fee + bursting_fee + change_penalty
```

- Constraints can be broken into pieces

```
constr = []  
  
constr += [sum(cvx.sum(m, axis=1) for m in Z1) <= 1]  
constr += [cvx.sum(Z2, axis=1) + sum(cvx.sum(m, axis=1) for m in Z3) <= 1]  
  
for t in range(len(X)):  
    constr += [cvx.sum(X[t], axis=1) == 1]
```

- Use of Parameters
  - similar to `tf.placeholder`



## Difficulties in model formulations:

- CVXPY variables cannot have more than 2 dimensions
  - no numpy style broadcast:  $\mathbf{X}[:, :, :, \text{None}]$
  - How to initiate such kind of variables?

```
x = list(cvx.Variable((I,J), boolean=True) for t in T)
```

- How to do calculation with such kind of variables?

```
# Storage fee
obj_func += omega/trm * cvx.sum(sum(i[0] * i[1] for i in zip(np.moveaxis(DH,-1,0), X)))
```

## Be cautious:

- computation override
- cvx computation is different from np computation

$f(t)$  for  $t$  in  $T$ , will return a list

zip function will return iterable list of pairs

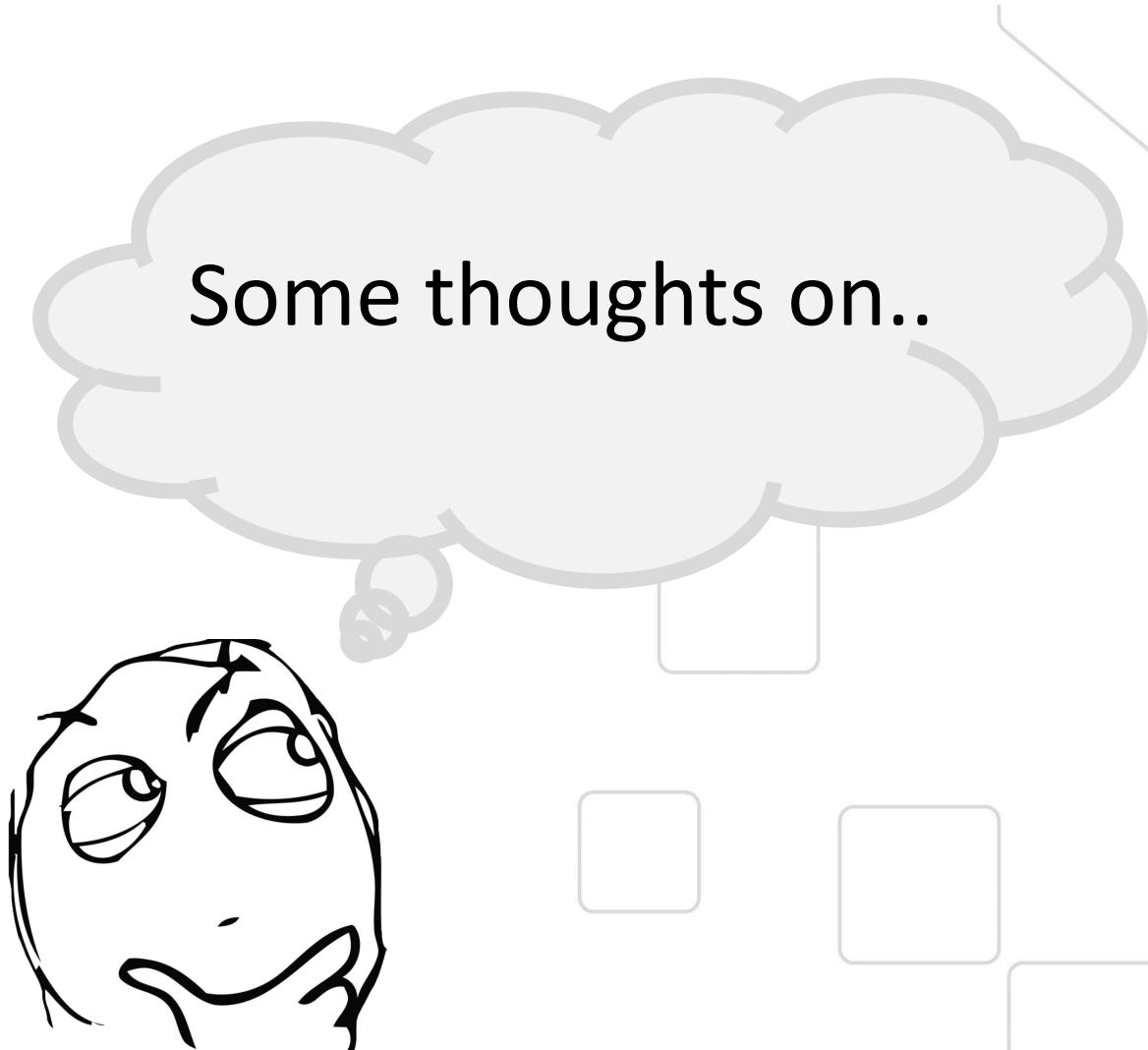


## New Skills:

- LSTM (Long Short Term Memory)
- ARIMA (Auto Regressive Integrated Moving Average)
- ETS (Error Trend Seasonality)
- MILP (Mixed Integer Programming)
- Data wrangling packages
- Tableau
- CVXPY
- Object Oriented Programming with Python
- Data instinct

## Skills learned from others:

- Multi-level modeling
- Discrete feature encoding
- Feature engineering on time series data
- Ensemble
- Parameter Tuning
- Industry classification
- Unit Test
- Web data visualization
- PySpark
- Parallel computing
- Read source code
- Code standard



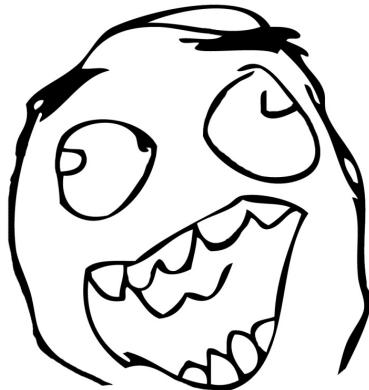
- A lot of hidden story in data.
- Be very cautious in data processing.
- Balance between dirty and clean.
- Balance between usability and accuracy.
- Balance between project and self-development.
- What clients need and what we need.
- Working style. Work with colleagues.
- Mentorship.



# Summary

数据驱动 运筹帷幄

Data Scientist  
LV. 1



Thank you!