杉数科技 Cardinal Operations

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 1st Review [redacted]
  - Parts Demand Prediction (week 7 ~ week 9) ← Client 2nd Review [redacted]
  - CVXPY model implementation (week 9 ~ week 11) ← Client 3rd Review [redacted]

Censored

Part 1

Training – Unilever Demand Prediction

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

- 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)

← Client 1st Review

← Client 2nd Review

← Client 3rd Review

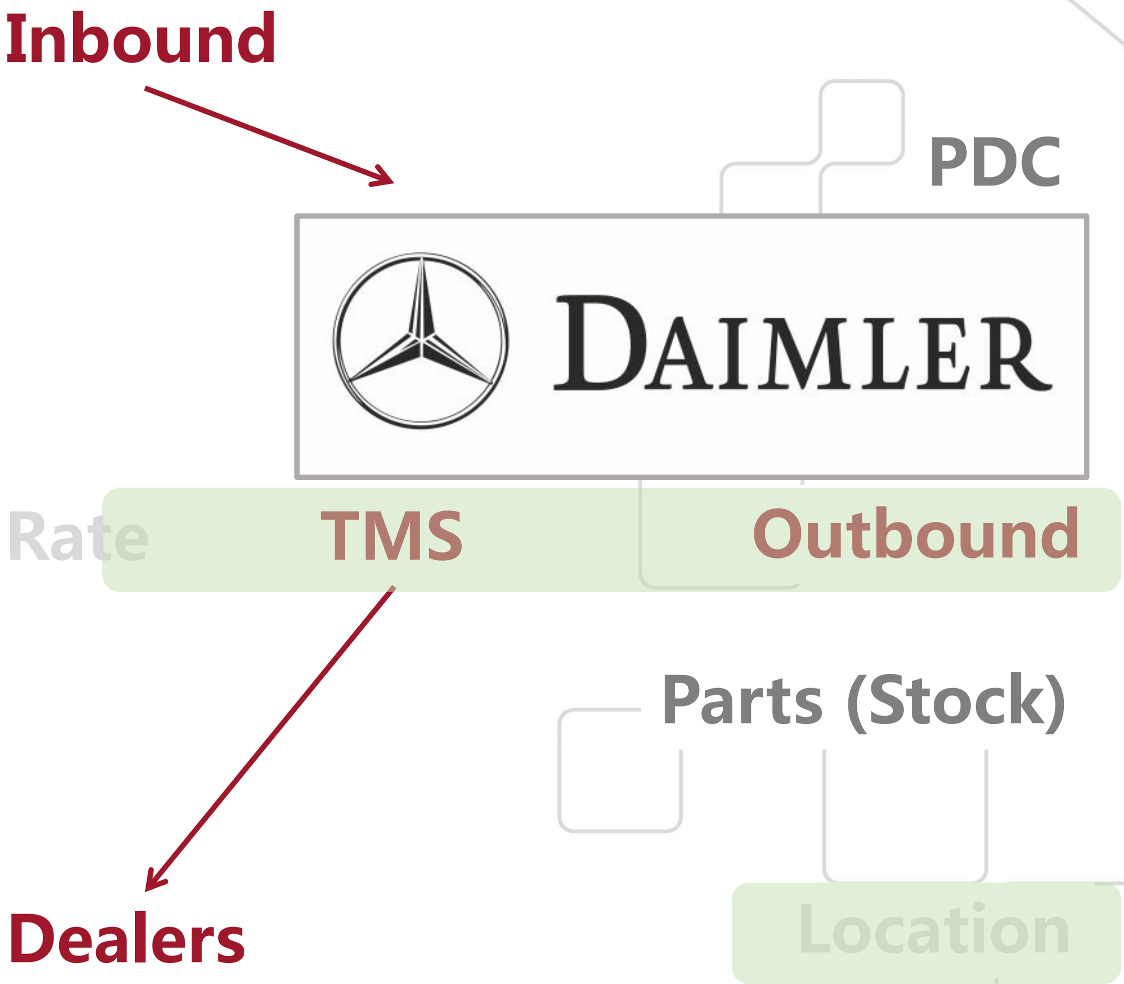
Censored

- Set the project 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)

Part 2

Data Wrangling, Exploration & API Request

Overview

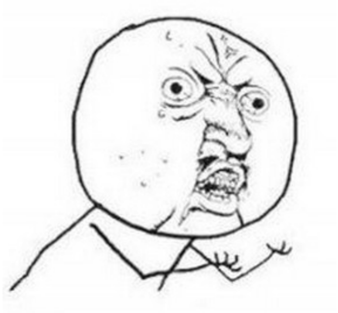


- **Objective**
- Open warehouses of reasonable size at optimal location.

- **My Task**
- Ensure data is complete and accurate.
- Join tables.
- Deal with missing entries.
- Collect additional data when necessary.

- Predict future demand.

- Help implement optimization model.



Ways that data can be dirty:

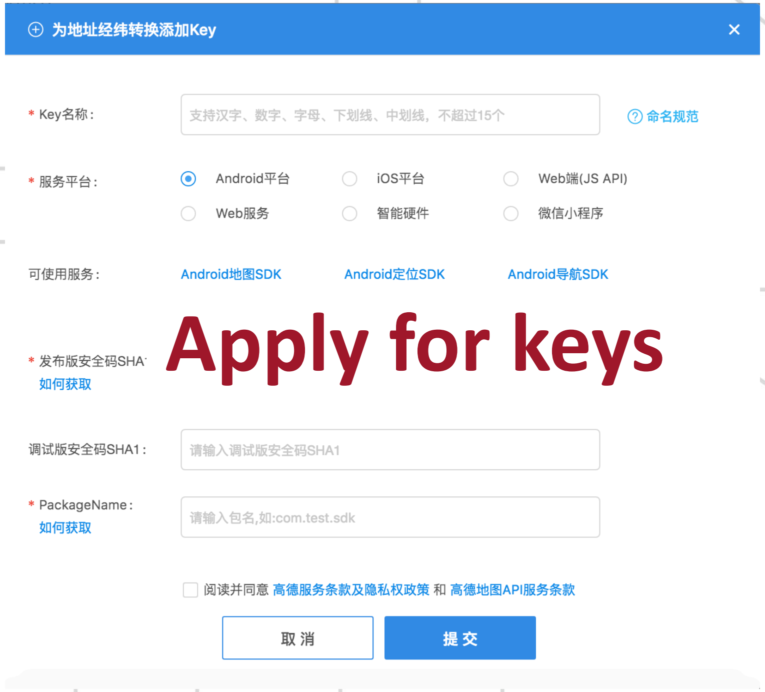
Problem	Reason / Description	Solution
Multiple keys for one records		
Multiple keys for one records		
Missing data		
Data of different magnitudes in one column		
Inconsistent primary key		
Chinese Characters VS. pinyin		
Secret Meanings behind unusual keys		
Ambiguous definition of features		

[redacted]

Censored

Amap API:

Registration → **Apply for keys** → **Use**




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

How to use it?

```python
import requests

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

Coordinates

```python
def driveDistance(origin, destination):
    parameters = {'origin': origin, 'destination': destination, 'strategy': 0, 'key': '6ec1a32b3d319596f2d05fc98e0b
    base = 'http://restapi.amap.com/v3/direction/driving'
    response = requests.get(base, parameters)
    answer = response.json()
    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)

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

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.

Censored

Parts Demand Prediction

Finding Substitutes and Complements

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

Substitutes:

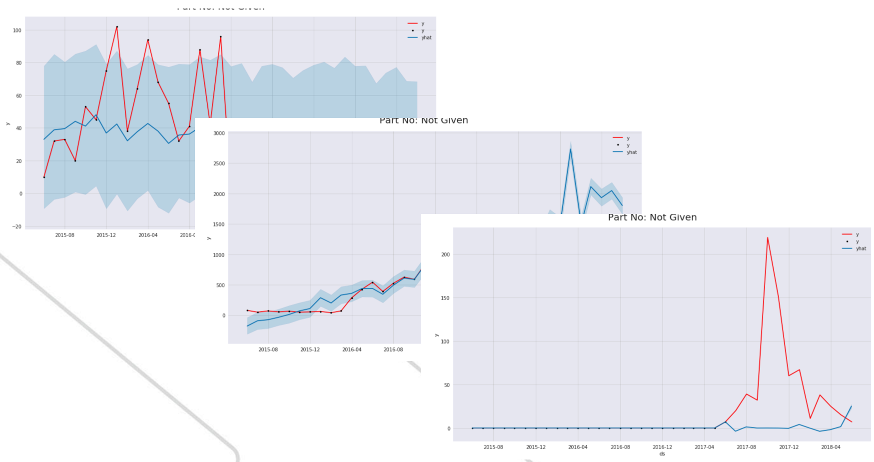
Reduce noise in demand trend.

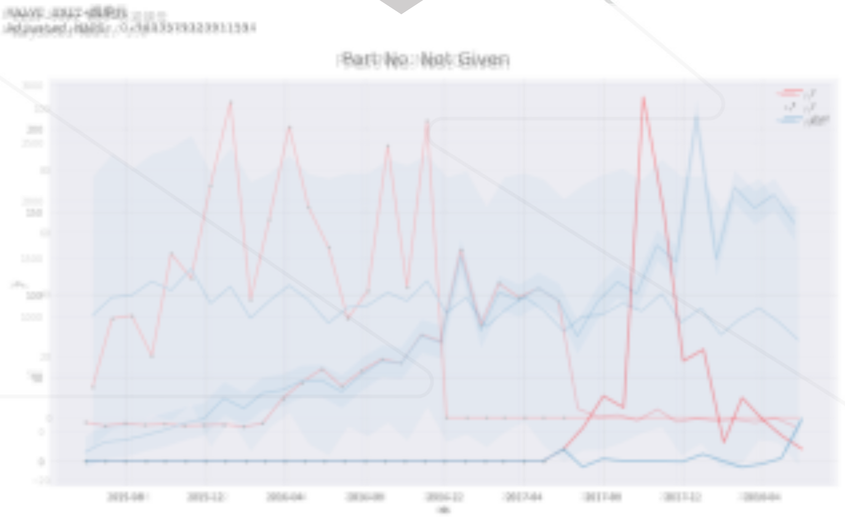
- Clustering by parts properties.
- Clustering by description.
- Clustering for better model performance.

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

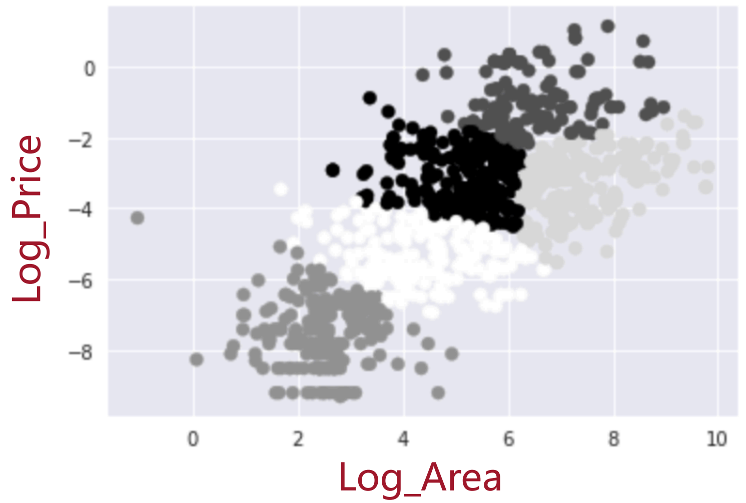
Clustering by Parts Properties

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.



Width

Censored

Length

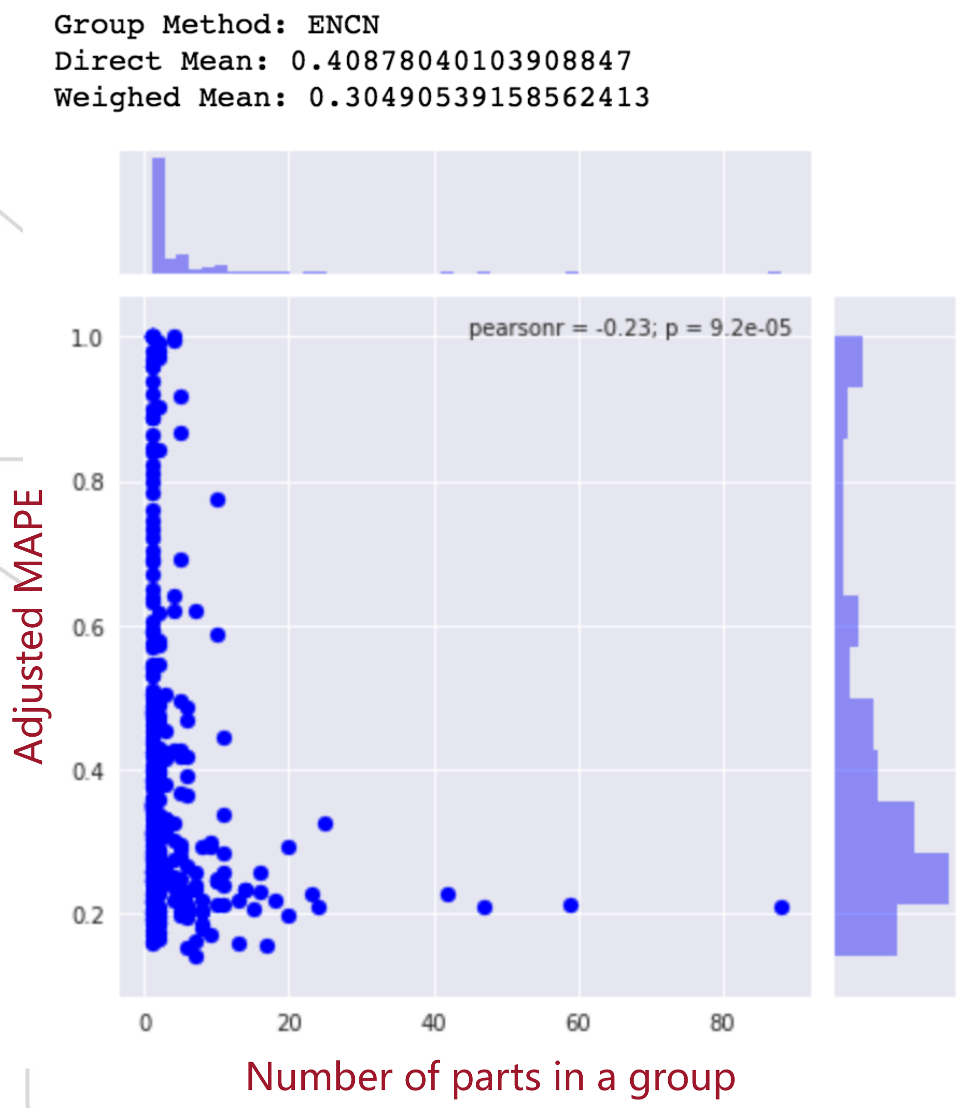
Clustering by description

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

Censored

Furthermore:

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



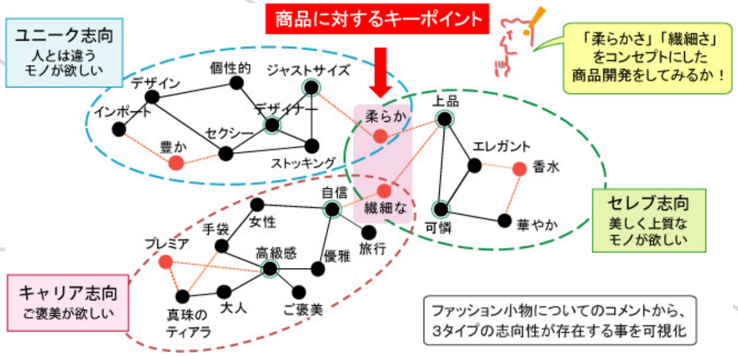
Clustering by Jaccard Coefficient

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:

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

- 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

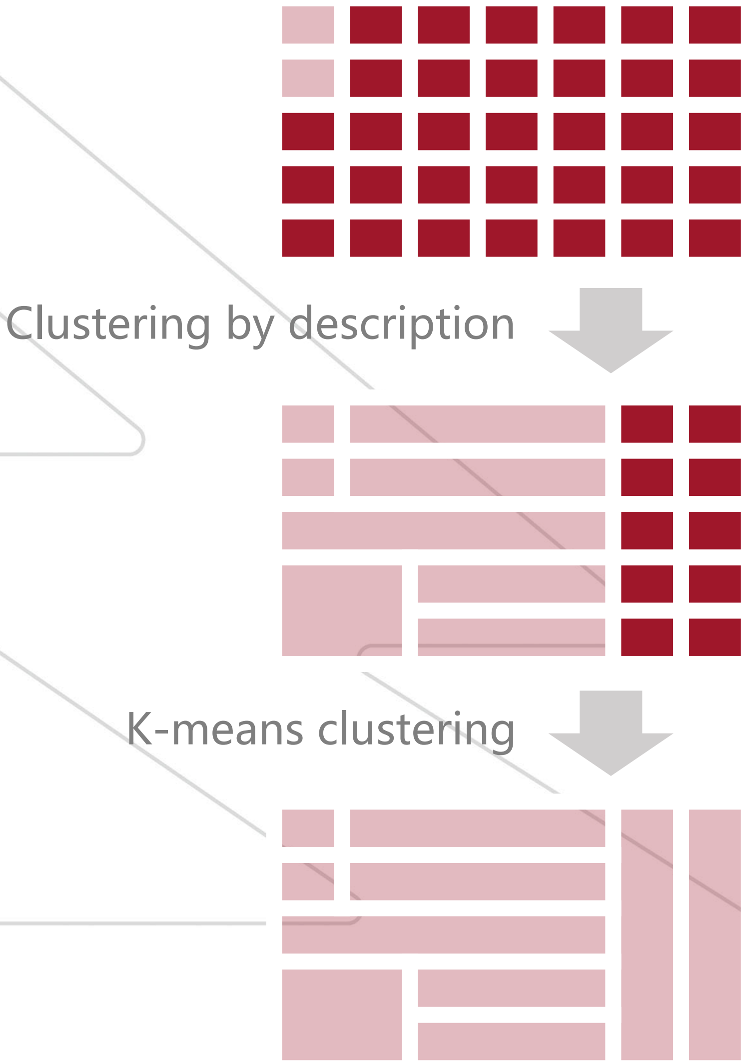
After few trials of clustering heuristics, we have a handful of methods can be applied to clustering.

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?

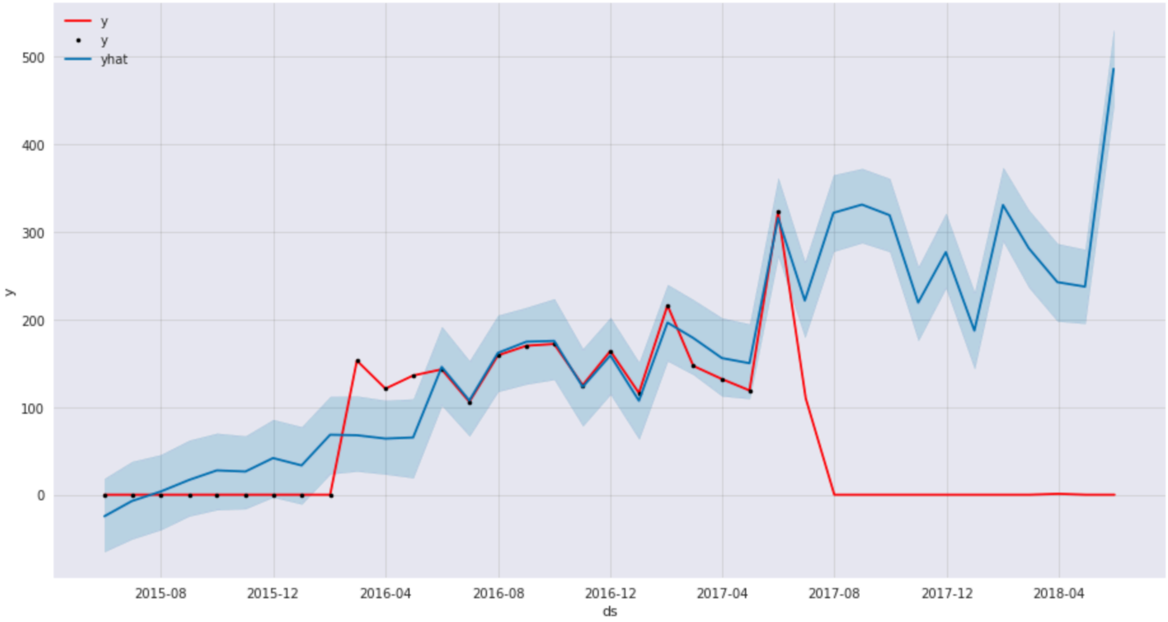


Parts Demand Prediction

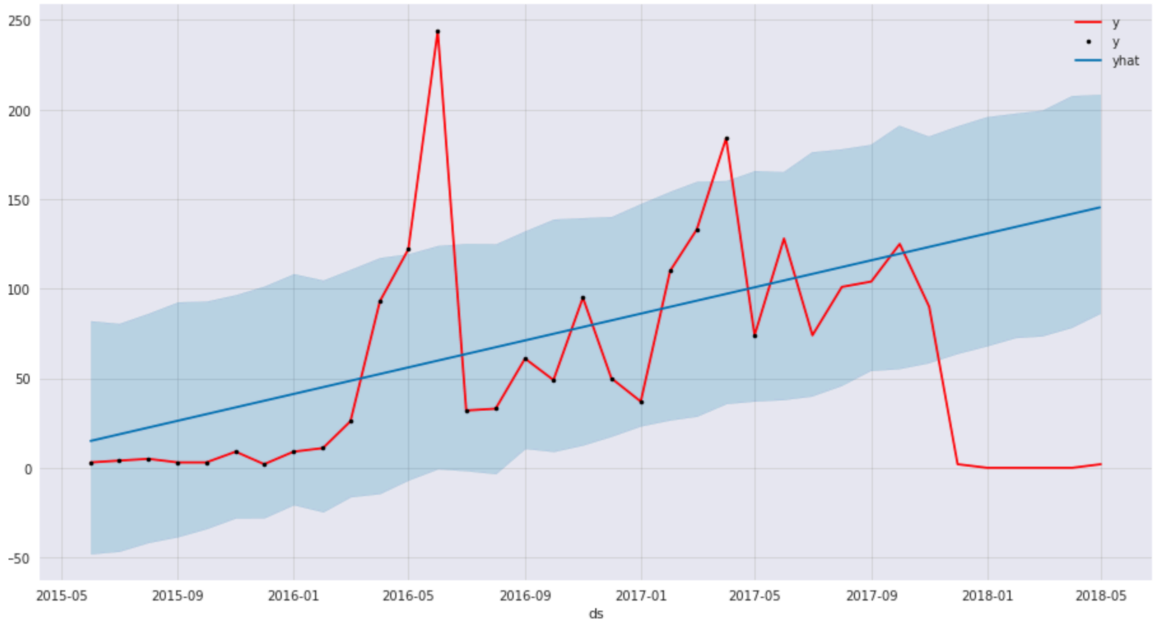
Prediction model: (predict 12 month)

- 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

Parts Demand Prediction

What we use at last:

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

Censored

Part 4

CVXPY model implementation

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 1.00000000001 -1.19961841702e-11
```

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

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

CVXPY model implementation

[redacted]

Censored

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 = nonexist_open_fee + nonexist_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: ~~X[:, :, :, None]~~
  - How to initiate such kind of variables?

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

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

- 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)))
```

zip function will return iterable list of pairs

Be cautious:

- computation override
- cvx computation is different from np computation

Summary

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:

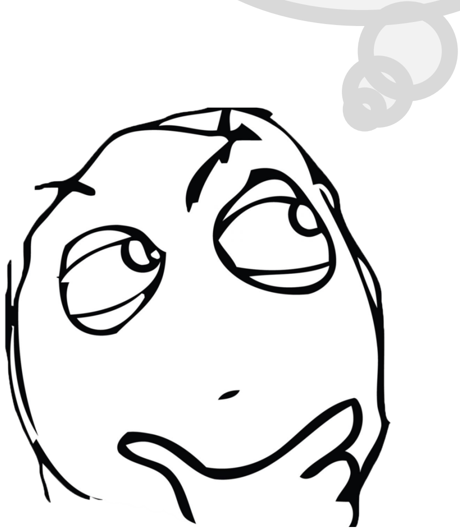
- Multi-level modeling
- Discrete feature encoding
- Feature engineering on time series data
- Ensemble
- Parameter Tuning
- Industry classification

- Unit Test
- Web data visualization
- PySpark
- Parallel computing

- Read source code
- Code standard

Summary

Some thoughts on..



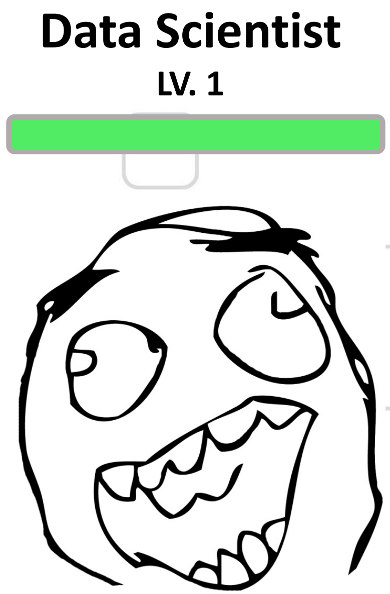
- A lot of hidden story in data.
- Be very cautious in data processing.

- 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



Thank you!