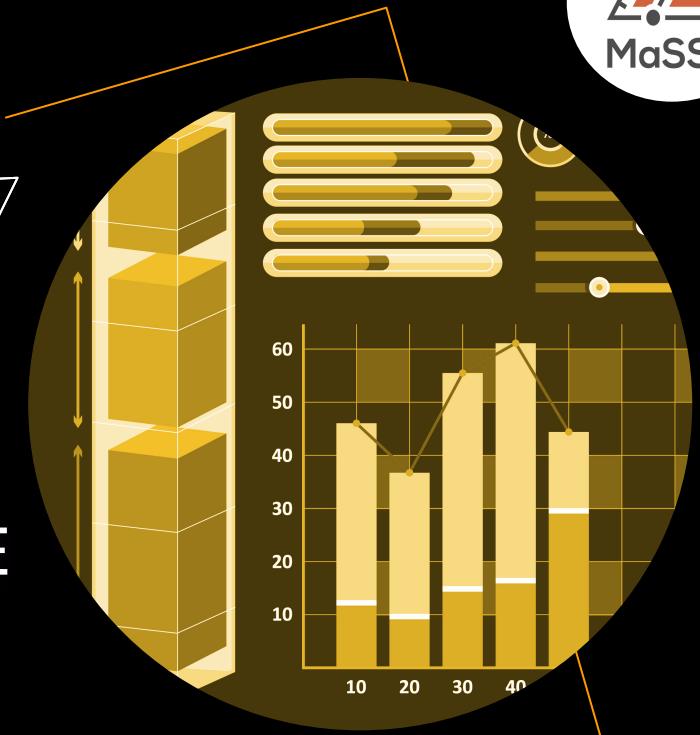


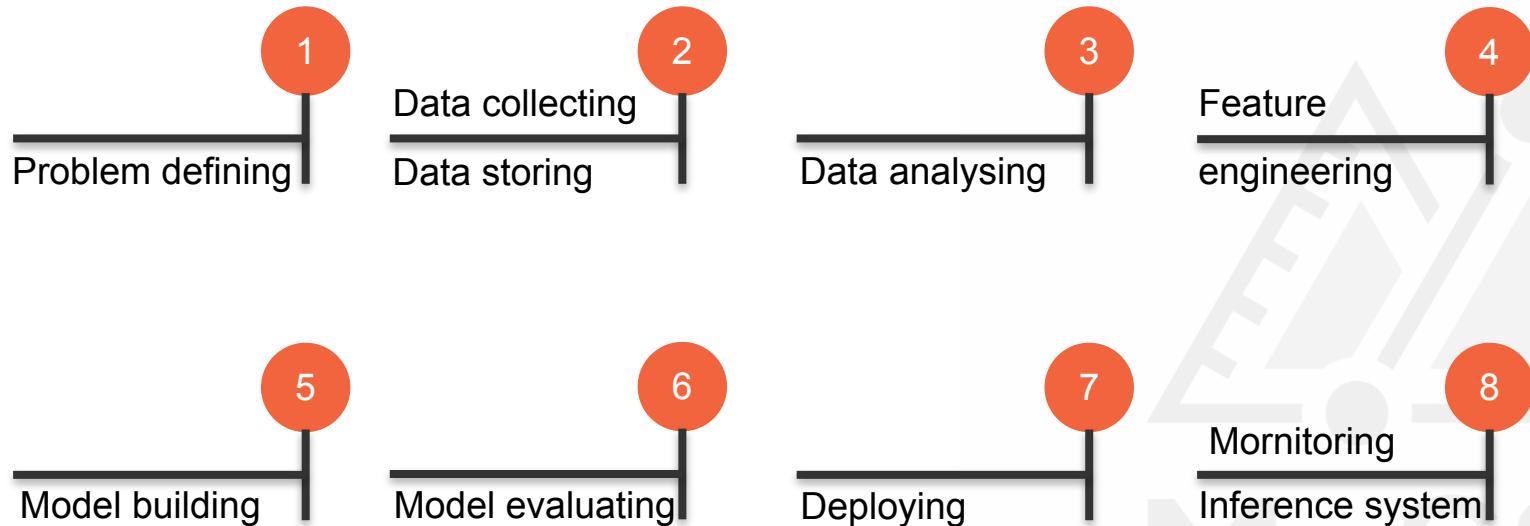


# A SIMPLE CLASSIFICATION PROBLEM CODING TO UNDERSTAND STRUCTURE OF A MODELING PROBLEM

Date / Người giảng dạy



# Complete machine learning pipeline



Chúng ta sẽ lần lượt giới thiệu các bước cùng với ví dụ thực tế

Bài toán đang được ứng dụng tại **Sendo**®

# Problem defining

- Các ứng dụng công nghệ phần mềm kĩ thuật cao hiện tại đều sinh ra để giải quyết 1 vấn đề nào đó trong hệ thống hoặc giúp cải thiện năng suất lao động, thay thế con người làm những công việc lặp lại.
- AI cũng vậy, chúng ta luôn cần phải define 1 bài toán, 1 vấn đề đang gặp phải và liệu có thể sử dụng AI hay ko?

## Bài toán thực tế: personalized-recommender system

Phân tích và trích thông tin tương tác của 1 user (**click, view, sản phẩm đã mua...**), từ đó suy ra các thông tin chi tiết về cá nhân như: **độ tuổi, sở thích, thu nhập, thói quen**, từ đó khi user vào trang Sendo thì sẽ nhận được kết quả hiển thị sản phẩm phù hợp nhất với các thông tin của user đó.

**Mục tiêu cần đạt được:** các sản phẩm user quan tâm cần được đẩy lên trang hiển thị cao nhất có thể  
Ví dụ các sản phẩm trong top 5 gợi ý cần có tỉ lệ click cao  
**Các metric sử dụng:** CTR@k (click through rate), recall@k (độ bao phủ sản phẩm tại vị trí k)

# Problem defining - example

Sendo

Tài ứng dụng Chăm sóc khách hàng Kiểm tra đơn hàng

Tim kiếm trên Sendo...

Vuong Thanh

Cho bạn Áo sơ mi nam Bỉm, Tã, Vệ sinh Đồ bảo hộ mô tô, xe máy Thiết bị & dụng cụ chăm s... Máy xay, máy ép

Quay trúng 1 triệu Thời trang nữ -70% Sân deal mỗi ngày Chợ Tết Sendo Voucher & Mã Freeship Quần áo nam sale sốc Nạp tiền điện thoại Thanh toán hóa đơn Vé Xe Rẻ -50% ZaloPay Freeship 0Đ

CÒN 26 NGÀY ĐẾN TẾT

TRANG HOÀNG KHÔNG GIAN DEAL BÍ MẬT GIẢM 50K

DUY NHẤT 17.01

QUAY TRÚNG VOUCHER ĐẾN 200K

FLASH SALE 03 : 14 : 01

Xem tất cả

Product Image	Name	Original Price	Sale Price	Discount (%)
	GIAO HÀNG VIỆT	55.000đ	121.000đ	-55%
	FREE SHIP	62.000đ	139.000đ	-56%
	GIAO HÀNG VIỆT	59.000đ	81.000đ	-28%
	KOKOMI	18.000đ	49.000đ	-64%
	ROMANO	62.000đ	250.000đ	-76%
	1800W	39.000đ	155.000đ	-75%
	Cháo Tứ Bánh	60.000đ	219.000đ	-73%
	Cháo Tứ Bánh	69.000đ	126.591đ	-46%

# Problem defining - example



# Data collecting and storing

- Câu hỏi đầu tiên cần đặt ra: Chúng ta đã có dữ liệu chưa ?



- Nếu chưa có hoặc chưa đủ dữ liệu, dữ liệu có thể được thu thập theo thời gian bằng một số cách:

- Online content analysis
- Online focus groups
- Online interviews
- Online qualitative research
- Online questionnaires
- Social network analysis

- Lưu dữ liệu:

Dữ liệu sẽ được lưu vào các database như MongoDB, SQL,..

với 1 số lượng record lớn để dễ dàng biến đổi và sử dụng cho bài toán.

Thông thường dữ liệu được lưu vào MongoDB với dạng các json chứa các record

Ví dụ:

```
{user:'mr john cena', phone:01234567, address : "abc abc", image:'path'}
```



# Data collecting and storing - Example

Trước khi làm bắt đầu làm bài toán, các câu hỏi về dữ liệu được đặt ra, để có thể tính toán mức độ khả thi, tránh mất thời gian và tiền thuê database lưu trữ.

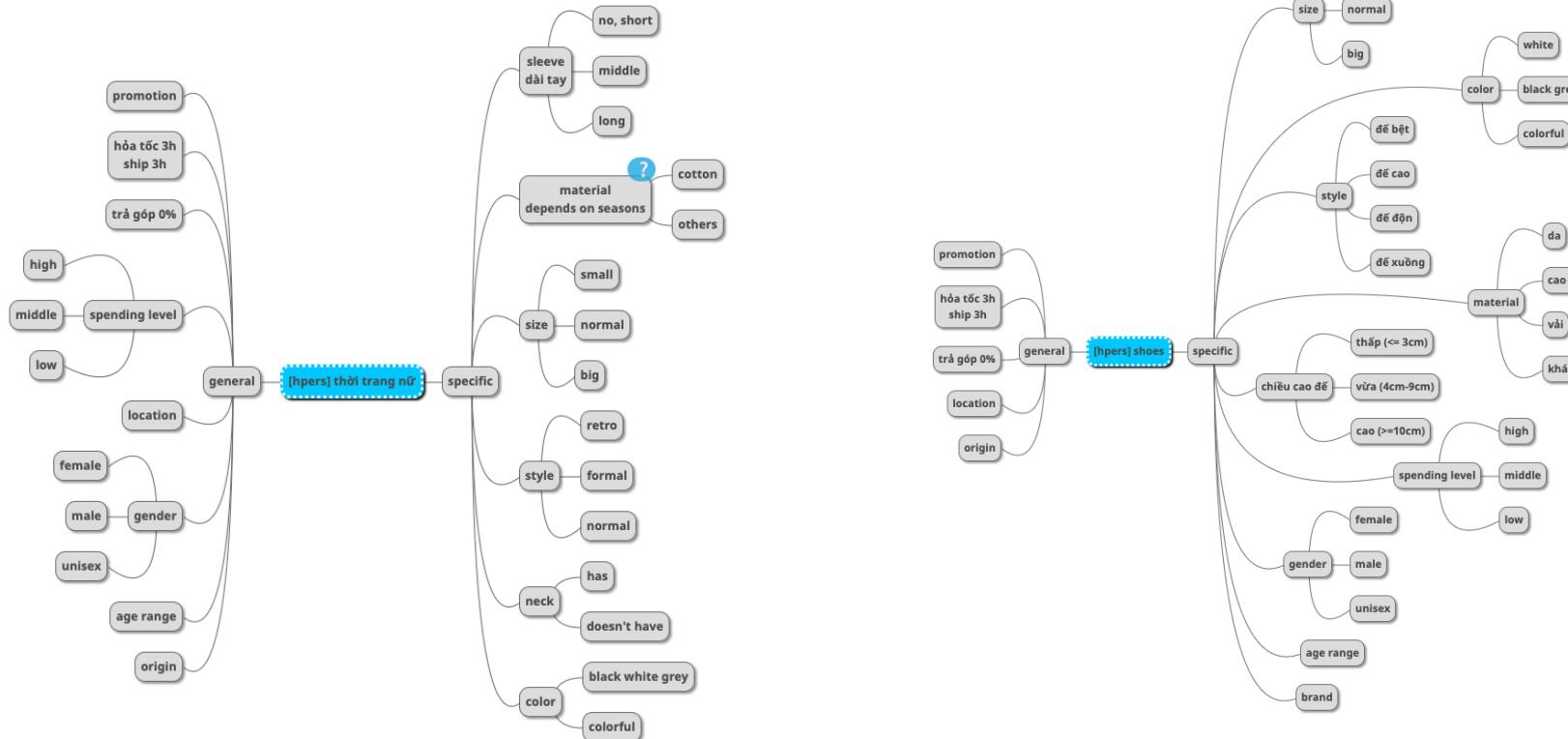
Feature	How	Question
Favorite category	<ul style="list-style-type: none"> <li>Từ hành vi access vào category, click sản phẩm?</li> </ul>	<ul style="list-style-type: none"> <li>Cần trọng số, nên như thế nào?</li> <li>Có nguồn hay cách nào khác để đánh giá mức độ yêu thích này?</li> </ul>
Favorite shops	<ul style="list-style-type: none"> <li>Từ dữ liệu shop liked?</li> <li>Từ dữ liệu shop click?</li> </ul>	<ul style="list-style-type: none"> <li>Câu hỏi tương tự như với Favorite category.</li> </ul>
Location	<ul style="list-style-type: none"> <li>Từ đơn hàn/Từ GPS/Từ IP?</li> </ul>	<ul style="list-style-type: none"> <li>Có tác dụng như thế nào với các category trong ngành fashion?</li> </ul>

Feature	How	Question
Spending Level	<ul style="list-style-type: none"> <li>Phân loại ngành hàng thuần chủng (ngành hàng mà sản phẩm khá đồng nhất)</li> <li>Phân loại price range trong từng ngành hàng</li> </ul>	<ul style="list-style-type: none"> <li>Phân loại price range như thế nào cho hợp lý?</li> </ul>
Gender	<ul style="list-style-type: none"> <li>Từ hành vi người dùng?</li> <li>Từ thông tin account?</li> </ul>	<ul style="list-style-type: none"> <li>Global gender info vs contextual gender info?</li> </ul>

Ngoài thông tin về user, những thông tin khác cần lưu trữ?

# Data collecting and storing - Example

Ví dụ về các thông tin của sản phẩm cần lưu



# Data collecting and storing - Example

Data tại Sendo chủ yếu được lưu trong mongo DB, theo các bảng được định dạng schema rõ ràng

conf["data\_pquet\_buyer\_info\_INC\_filepattern"] (e.g. "/user/xa-m/shared/tiepmk/fdms/buyer\_info\_incremental/upto\_date={}/baid\_group={}")

## a. Schema

1	account_id	bigint	186152
2	account_email	string	vivi24112009@yahoo.com
3	account_phone	string	0397642983
4	facebook_id	string	# mostly empty. 99.9%
5	first_name	string	linh
6	last_name	string	trần mỹ
7	full_name	string	linh trần mỹ
8	gender	float	1.0
9	address_full	string	dội 4, thôn cổ mỹ
10	address_city	string	quảng trị
11	address_city_id	bigint	46
12	address_district	string	huyện vĩnh linh
13	address_district_id	bigint	563
14	address_ward	string	xã vĩnh giang
15	address_ward_id	bigint	19423
16	total_order	float	17.0
17	total_order_cancel	float	0.0
18	total_order_close	float	0.0
19	total_order_complete	float	17.0
20	total_rating	float	0.0
21	last_payment_method	string	# all empty value
22	latest_order_date	string	2019-04-17 11:27:39 +0700
23	email_verified	float	0.0
24	is_otp_phone	float	1.0
25	is_bad_buyer	float	None
26	created_ip	string	# all empty value
27	utc_created_at	string	2014-03-21 17:31:04 +0700
28	utc_updated_at	string	2020-05-25 00:16:01 +0700
29	utc_rad2_insert	string	2019-07-20 01:04:37 +0700
30	utc_rad2_update	string	2020-05-17 09:55:34 +0700
31	baid_group	int	2

# Data analysing

Data analysing sẽ bao gồm 2 steps chính: Data cleansing và Data exploratory

## Data cleansing:

- Có những dữ liệu được thu thập có thể không hữu ích hoặc không liên quan đến mục tiêu phân tích của bạn, do đó dữ liệu cần được làm sạch.
- Dữ liệu được thu thập có thể chứa các bản ghi trùng lặp, khoảng trắng hoặc lỗi.
- Giai đoạn này phải được thực hiện trước phân tích vì dựa trên việc làm sạch dữ liệu, đầu ra phân tích của bạn sẽ gần hơn với kết quả mong đợi.

## Data exploratory:

- Sau khi dữ liệu được thu thập, làm sạch và xử lý, dữ liệu có thể sử dụng để phân tích.
- Trong quá trình phân tích, có thể thấy thông tin chính xác bạn cần hoặc bạn cần thu thập thêm dữ liệu.
- Trong giai đoạn này, bạn có thể sử dụng các công cụ và phần mềm phân tích dữ liệu để giúp bạn hiểu, giải thích và rút ra kết luận dựa trên các yêu cầu.

# Data analysing - example

Phân tích tương tác của user với hệ thống để tìm ra các insight quan trọng có thể tạo feature:

- K - top category người dùng tương tác nhiều nhất => favourite category => tạo feature suggest các sản phẩm trong category này nhiều hơn ở trang Home
- K - top shop có lượng tương tác cao nhất => favourite shop => tạo feature suggest các sản phẩm thuộc các shop yêu thích nhiều hơn Trong mỗi category, lấy hết tất cả các tên sản phẩm được tương tác và tạo thành 1 bộ corpus đưa qua CountVectorizer và lấy ra kết quả các từ có trọng số cao nhất → Đưa ra sản phẩm yêu thích và các thuộc tính của sản phẩm nếu user tìm thuộc tính đó nhiều
- Ví dụ 1 user có id: 1234, các sản phẩm user này tương tác nhiều trong thời gian 1 tuần qua:

['máy\_may gia\_dình juki', 'máy\_may gia\_dình', 'máy\_may gia\_dình juki hzl 8800', 'máy\_may  
gia\_dình bernina đa\_năng nhật\_bản', 'máy\_may gia\_dình janome', 'máy\_may gia\_dình juki flora  
5000', 'máy\_may gia\_dình juki flora 5000', 'máy\_may gia\_dình cổ\_diễn truyền\_thống', 'máy\_may  
gia\_dình juki', 'máy\_may gia\_dình đa\_năng hàng nội\_địa nhật may được hơn 10 kiều chỉ',  
'máy\_may gia\_dình juki', 'máy\_may gia\_dình tiện\_lợi toyota nội\_địa nhật', 'máy\_may gia\_dình  
nhật\_bản đa\_năng chính\_hãng', 'máy\_may gia\_dình máy\_may nhật\_bản singer đời\_mới', 'máy\_may  
gia\_dình juki flora', 'máy\_may gia\_dình nhật\_bản', 'máy\_may  
gia\_dình juki missin 7700', 'máy\_may gia\_dình janome nhật chính\_hãng', 'máy\_may gia\_dình  
juki', 'máy\_may mini gia\_dình máy\_khâu với 12 kiều máy khác\_nhau phù\_hợp\_với mọi loại vải -  
bảo\_hành 12 tháng', 'máy\_may gia\_dình tiện\_lợi toyota nội\_địa nhật', 'máy\_may gia\_dình jagua  
bàn ga nhật', 'máy\_may gia\_dình bán\_công nghiệp', 'máy\_may gia\_dình bán\_công nghiệp',  
'máy\_may gia\_dình tiện\_lợi toyota nội\_địa nhật', 'máy\_may gia\_dình vừa may vừa thêu brother  
tendy 7000', 'máy\_may gia\_dình nhật', 'máy\_may gia\_dình nhật\_bản', 'máy\_may mini gia\_dình  
singer merrit hàng nhật\_bản - được kiềm\_hàng', 'máy\_may gia\_dình bán\_công nghiệp', 'máy\_may  
gia\_dình singer', 'máy\_may gia\_dình nhật\_bản', 'máy\_may gia\_dình nhật', 'máy\_may gia\_dình  
bán\_công nghiệp', 'máy\_may gia\_dình nhật brother', 'máy\_may gia\_dình đa\_năng juki nhật\_bản',  
'máy\_may gia\_dình nhật brother', 'máy\_may gia\_dình đa\_năng juki nhật\_bản', 'máy\_may gia\_dình  
nội\_địa nhật brother tendy 7000', 'máy\_may bán\_công nghiệp gia\_dình nội\_địa nhật', 'máy\_may  
gia\_dình sinco tăng 5 khẩu\_trang y\_tê', 'máy\_may gia\_dình tiện\_lợi toyota nội\_địa nhật',  
'máy\_may bán\_công nghiệp gia\_dình nội\_địa nhật', 'máy\_may gia\_dình bán\_công nghiệp',  
'máy\_may janome công\_nghiệp để bàn siêu\_khỏe của nhật', 'máy\_may gia\_dình bán\_công nghiệp',  
'máy\_may gia\_dình acme 588 mới', 'máy\_may gia\_dình bernina đa\_năng nhật\_bản', 'máy\_may  
gia\_dình jagua bàn ga nhật', 'máy\_may gia\_dình bán\_công nghiệp', 'máy\_may gia\_dình bán\_công  
nghiệp', 'máy\_may gia\_dình có kết\_hợp thêu', 'máy\_may gia\_dình acme 588 mới', 'máy\_may  
gia\_dình jagua bàn ga nhật', 'máy\_may gia\_dình juki', 'máy\_may gia\_dình brother nội\_địa  
nhật', 'máy\_may gia\_dình singer fitline 6580', 'máy\_may gia\_dình đa\_năng juki nhật\_bản',  
'máy\_may gia\_dình juki', 'máy\_may gia\_dình đa\_năng juki nhật\_bản', 'máy\_may gia\_dình juki',  
'máy\_may gia\_dình tiện\_lợi toyota nội\_địa nhật', 'máy\_may gia\_dình nhật máy\_may singer  
đa\_năng', 'máy\_khâu gia\_dình với 12 chế\_độ may máy\_may mini cao\_cấp sản\_xuất theo công\_nghệ  
nhật\_bản bảo\_hành 1 năm - được kiềm\_hàng', 'máy\_may gia\_dình 12 kiều may có vắt\_số may  
lại\_mũi siêu tiện\_lợi giá ưu\_đãi - được kiềm\_hàng', 'máy\_khâu gia\_dình với 12 chế\_độ may  
máy\_may mini cao\_cấp sản\_xuất theo công\_nghệ nhật\_bản bảo\_hành 1 năm', 'máy\_may gia\_dình  
juki', 'máy\_may gia\_dình đa\_năng juki nhật\_bản', 'máy\_may gia\_dình jagua bàn ga nhật',  
'máy\_may gia\_dình vừa may vừa thêu brother tendy 7000', 'máy\_may gia\_dình brother profeel',  
'bóng đèn năng\_lượng mặt\_trời cảm\_ứng ngày\_đêm đèn solar light 40w', 'đèn năng\_lượng  
mặt\_trời', 'đèn năng\_lượng mặt\_trời 40w solar light đèn\_phá led năng\_lượng mặt\_trời vnled  
vietnamled dt 0936395395', 'bóng đèn led sạc tích\_diện đèn led sạc pin ánh\_sáng trắng có  
móc\_treo kèm\_theo bóng đèn gia\_dụng ánh\_sáng trắng chống\_thâm\_nước tiết\_kiệm năng\_lượng  
hl146', 'đèn năng\_lượng mặt\_trời', 'đèn năng\_lượng mặt\_trời', 'cây bơm\_thút thông tắc  
bồn\_cầu', 'dụng\_cụ thông tắc cổng bồn\_cầu', 'máy\_hút\_bụi gia\_dình', 'bộ 2 dụng\_cụ nút  
đa\_năng mở\_đậy nắp bồn\_cầu nhà\_vệ\_sinh thông\_minh bảo\_vệ súc\_khoẻ và kháng\_khuẩn tiên\_dung  
cho trẻ\_em và người\_già']

Với hướng xử lí dữ liệu trên, ta có kết quả như sau:

-----important word: [('máy\_may', 74), ('gia\_đình', 72), ('nhật', 21), ('juki', 19), ('nhật\_bản', 15), ('đa\_năng', 11)]

-----fav shop: [793624, 828521, 742643, 347717, 781093]

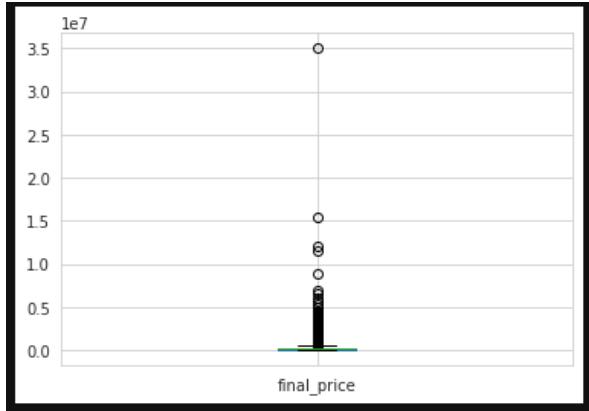
-----fav cat: ['1/2/1108/4000', '1/2/2849/3343', '1/2/1019/1074', '1/2/1108/3185', '1/2/1019/1067']

Check khoảng giá trung bình của user khi tương tác vào 1 category:

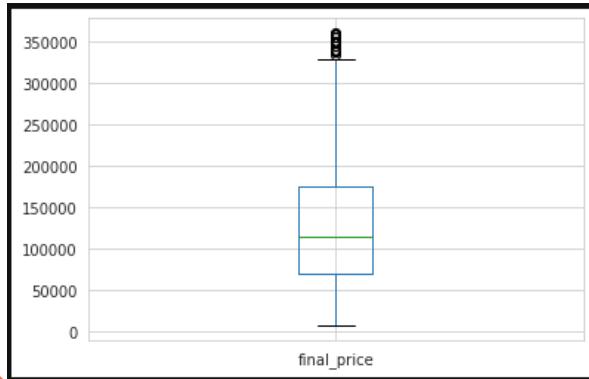
- Chia khoảng giá của các sản phẩm trong category ra làm 4 phần, ví dụ category "nhẫn, trang sức":

final_price	
count	8005.00
mean	693644.56
std	36060384.00
min	9000.00
25%	99000.00
50%	165000.00
75%	250000.00
max	3220000000.00

Min và max của category lêch nhau rất nhiều, khi vẽ boxplot có thể thấy rất nhiều outlier:



filter các outlier này bằng cách lấy các sản phẩm trong khoảng  $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$  trong đó  $Q1 = \text{quantile } 25\%$ ,  $Q3 = \text{quantile } 75\%$ ,  $IQR = Q3 - Q1$



Ta có thể lấy 4 khoảng quantile sau khi đã loại outlier để chia thành 4 khoảng giá của 1 category

# Feature engineering

Sau quá trình phân tích, ta có thể thấy được correlation giữa các thông tin với nhau, những thông tin nào có ích và ý nghĩa với bài toán hiện tại. Những thông tin nào nhiều outlier và thiếu dữ liệu (missing data) có thể bị loại bỏ.

Model ML chủ yếu tính toán dựa trên số thực, nên các feature quan trọng ở dạng str, list cần được vector hoá.

Ví dụ chúng ta có 1 cột thông tin là “Đảng phái” gồm các thông tin “Trump voter”, “Biden voter”, “kanye west” chúng ta cần đưa thông tin này về dạng số hoá, như `trump_voter = 1, biden_voter =2`

MaSSP

# Feature engineering - example

3 Steps included:

## 1/ Sparse Feature Remover

```
# step 1: sparse feature remover
cname_feat = np.asarray(cname_feat)
cname_feat_mask = []
pdf02_nrows = pdf02.loc[ind_train].shape[0]
cname_feat_mask = \
    (pdf02.loc[ind_train, cname_feat].isna().sum()/pdf02_nrows) \
        <= threshold_nan
cname_sparse_rm = cname_feat[cname_feat_mask]
pdf02.drop(cname_feat[~cname_feat_mask], axis=1, inplace=True)
```

## 2/ Outlier Handling

```
# step 2: outlier handling
t_qlower, t_qupper = quantile_range
qoutlier_lower, qoutlier_upper = [], []
for col in cname_sparse_rm:
    t_lower = pdf02.loc[ind_train, col].quantile(t_qlower)
    t_upper = pdf02.loc[ind_train, col].quantile(t_qupper)
    qoutlier_lower.append(t_lower)
    qoutlier_upper.append(t_upper)

pdf02[col].clip(t_lower, t_upper, inplace=True)
```

## 3/ Simple Imputation

```
# step 3: imputation
fill_value = []
for col in cname_sparse_rm:
    if fill_method == 'median':
        t_fval = pdf02.loc[ind_train, col].quantile(0.5)
        assert ~np.isnan(t_fval), f'col: {col}, {pdf02[col].isnull().sum()}, {pdf02.shape[0]}'
    elif fill_method == 'min':
        t_fval = pdf02.loc[ind_train, col].min()
    elif fill_method == 'max':
        t_fval = pdf02.loc[ind_train, col].max()
    elif fill_method == 'zero':
        t_fval = 0
    else:
        raise ValueError(
            'fill_method for imputation must be one of ["median", "min", "max", "zero"]')
    fill_value.append(t_fval)
pdf02[col].fillna(t_fval, inplace=True)
```

# Model training and evaluating

Khi train model chúng ta cần chia dữ liệu thành 3 tập: Training data, Validation data và Testing data

- Training data: sử dụng để huấn luyện model
- Validation data: sử dụng để đánh giá quá trình học của model
- Testing data: sử dụng 1 bộ dữ liệu không nằm trong tập train và validate để đánh giá hiệu quả của model khi gặp data mới.

2 khái niệm cơ bản để đánh giá 1 model là recall (độ bao phủ) và accuracy(độ chính xác)

Quá trình evaluate model sẽ dựa vào 2 chỉ số recall và accuracy để đánh giá, chọn ra model và bộ hyper-parameter phù hợp để đưa vào ứng dụng.

MaSSP

# Model training and evaluating – example

Tập train bao gồm dữ liệu tương tác của user trong 10 ngày

Tập validate gồm dữ liệu tương tác của user trong 2 ngày tiếp theo

Tập test lấy dữ liệu 8 ngày tiếp sau đó

Kết quả sau khi huấn luyện với bộ parameters default của xgboost:

Recall@30 tăng từ 80% lên 92% so với tập dữ liệu ko sử dụng ML model để rerank kết quả

	set_name	cname_rank	map_10	map_20	map_30	map_5	recall_10	recall_20	recall_30	recall_5	ndcg_10	ndcg_20	ndcg_30	ndcg_5	precision_10	precision_20	precision_30	precision_5
0	train	base_score	0.4069	0.4121	0.4127	0.3886	0.7725	0.8451	0.8606	0.6368	0.4945	0.5133	0.5167	0.4502	0.3074	0.3117	0.3122	0.2945
1	train	pscore_model	0.5312	0.5332	0.5334	0.5204	0.8352	0.8635	0.8685	0.7563	0.6054	0.6128	0.6139	0.5794	0.3136	0.3126	0.3125	0.3184
0	valid	base_score	0.3822	0.3862	0.3867	0.3676	0.7026	0.7588	0.7705	0.5950	0.4593	0.4738	0.4763	0.4242	0.2913	0.2945	0.2949	0.2818
1	valid	pscore_model	0.4851	0.4866	0.4868	0.4762	0.7504	0.7725	0.7763	0.6854	0.5500	0.5558	0.5566	0.5286	0.2961	0.2952	0.2951	0.2999
0	ind_test_0	base_score	0.1819	0.1954	0.1995	0.1574	0.5055	0.6987	0.8063	0.3206	0.2569	0.3060	0.3290	0.1972	0.0855	0.0927	0.0956	0.0780
1	ind_test_0	pscore_model	0.3005	0.3110	0.3137	0.2750	0.6979	0.8491	0.9157	0.5075	0.3938	0.4324	0.4468	0.3321	0.1048	0.1003	0.0993	0.1155
0	ind_test_1	base_score	0.1899	0.2032	0.2074	0.1640	0.5173	0.7086	0.8112	0.3211	0.2657	0.3143	0.3362	0.2025	0.0879	0.0950	0.0977	0.0792
1	ind_test_1	pscore_model	0.3026	0.3135	0.3160	0.2768	0.7037	0.8588	0.9204	0.5125	0.3971	0.4365	0.4498	0.3349	0.1065	0.1025	0.1013	0.1173
0	ind_test_2	base_score	0.1804	0.1951	0.1991	0.1572	0.4880	0.7041	0.8073	0.3121	0.2517	0.3065	0.3287	0.1947	0.0830	0.0915	0.0941	0.0758
1	ind_test_2	pscore_model	0.2888	0.2996	0.3023	0.2623	0.6923	0.8507	0.9209	0.4955	0.3834	0.4238	0.4389	0.3194	0.1035	0.0988	0.0978	0.1126
0	ind_test_3	base_score	0.1761	0.1903	0.1946	0.1514	0.4904	0.6951	0.8015	0.3030	0.2488	0.3007	0.3235	0.1886	0.0821	0.0897	0.0925	0.0739
1	ind_test_3	pscore_model	0.2856	0.2972	0.2997	0.2595	0.6837	0.8488	0.9129	0.4862	0.3790	0.4211	0.4350	0.3152	0.1016	0.0975	0.0962	0.1107

# Model training and evaluating – example

Tuning: việc tuning model để tìm ra bộ tham số phù hợp nhất với data, có thể tăng kết quả của model

```
{'is_cname_subset': 0,
 'model_options': {'eval_k_list': [5, 10, 20, 30],
                   'eval_metric_list': ['ndcg', 'map', 'recall', 'precision'],
                   'fit_params': {'early_stopping_rounds': 30,
                                 'num_boost_round': 500},
                   'max_features': 400,
                   'max_iter': 9,
                   'params_dict': {'fix_params': {'eval_metric': 'ndcg',
                                                 'important_type': 'gain'},
                                   'tune_params': {'booster': ['gbtree'],
                                                 'colsample_bytree': [0.9],
                                                 'gamma': [0],
                                                 'learning_rate': [0.01,
                                                                  0.1,
                                                                  0.3],
                                                 'max_depth': [3, 4, 5],
                                                 'n_estimators': [1000],
                                                 'objective': ['rank:pairwise'],
                                                 'reg_alpha': [0],
                                                 'reg_lambda': [1],
                                                 'subsample': [0.9]}}}}
```

Các bộ tham số được define sẵn sau đó sử dụng gridsearch hoặc randomSearch để tìm ra bộ tham số tốt nhất

## Deploying and monitoring inference system



Sau khi đã hoàn thiện model và pipeline xử lý dữ liệu, ta sẽ đưa model này lên production và quan sát, đánh giá model khi chạy thực tế.

Machine learning inference system:

