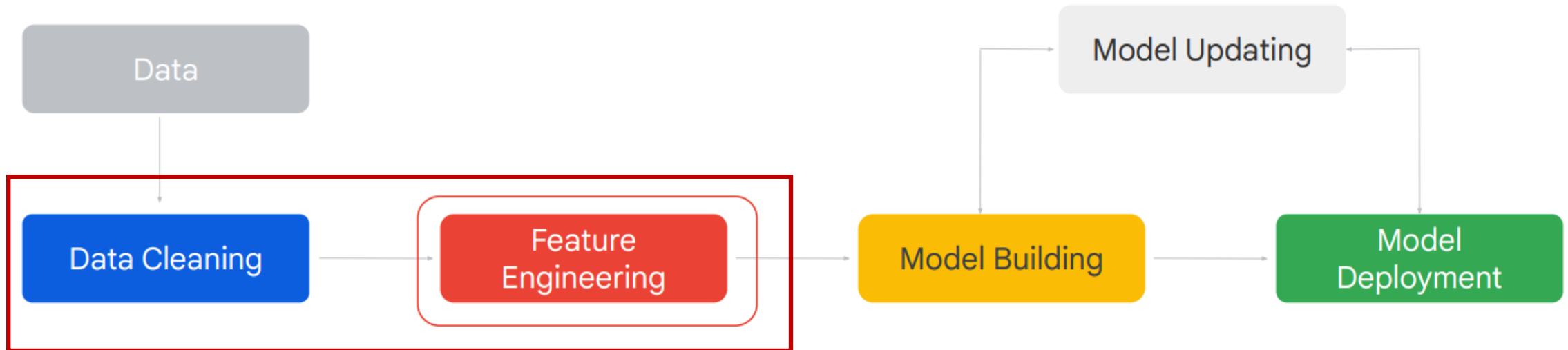


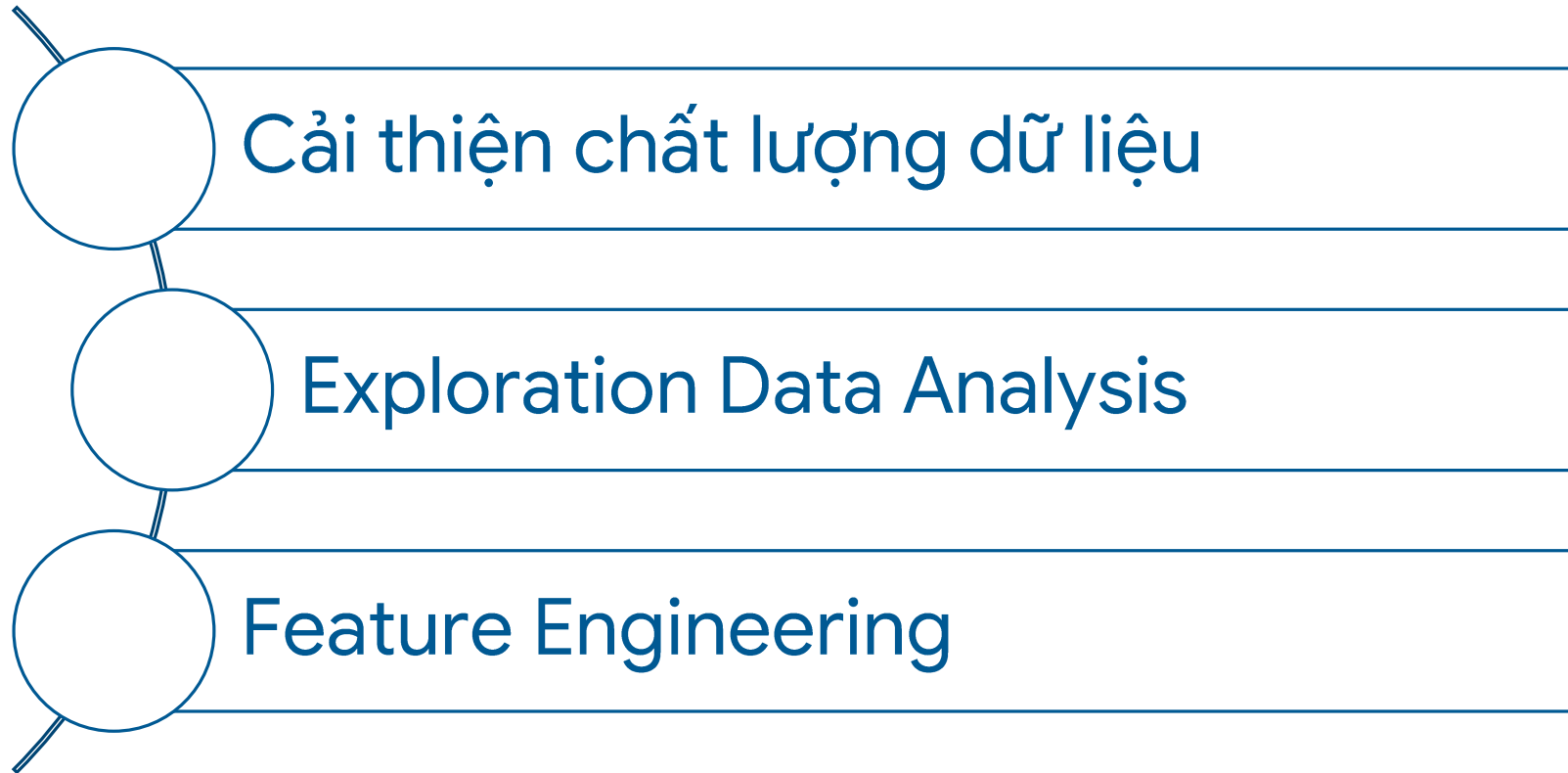
Exploration Data Analysis & Feature Engineering

2022/08, bigdata

Where are we?

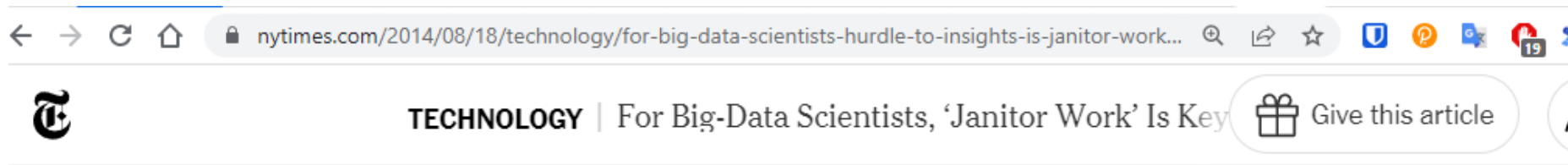


Chủ đề buổi học



80%

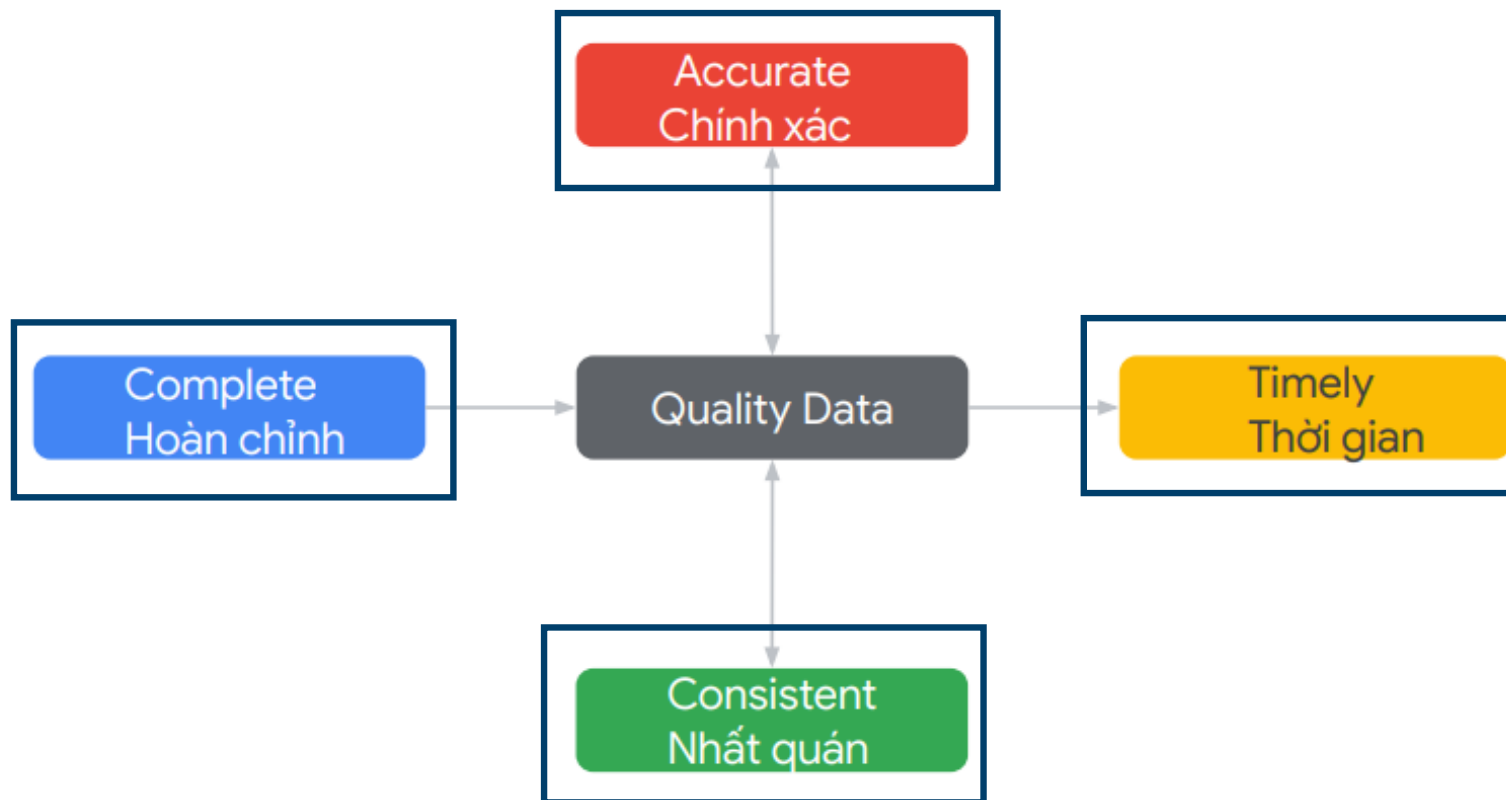
phần trăm thời gian của Data Scientist là data preparing



Yet far too much handcrafted work — what data scientists call “data wrangling,” “data munging” and “data janitor work” — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

Clean data

Chất lượng dữ liệu cho model



Clean data

Cải thiện chất lượng dữ liệu



1

Resolve Missing
Values



2

Convert the
Date feature
column to
Datetime
Format



3

Parse date/time
features



4

Remove
unwanted values



5

Convert
categorical
columns to
"one-hot
encodings"

Data về balance khách hàng

	CIF	cob_dt	current_bal	prv_month	prv_month_bal	prv_year	prv_year_bal
0	1401895852	2019-12-31	1.600000e+09	2019-11-30	3.350000e+09	2018-12-31	2.100000e+09
1	2400002097	2020-02-29	2.800000e+09	2020-01-31	2.800000e+09	2019-02-28	1.880000e+09
2	2400002747	2019-07-31	4.283642e+08	2019-06-30	4.283642e+08	NaN	NaN

Thống kê null data

0s

```
1 data_balance.isnull().sum()
```

CIF	0
cob_dt	0
current_bal	0
prv_month	0
prv_month_bal	0
prv_year	123947
prv_year_bal	123947
next_1month	0
next_1month_bal	0
next_2months	10601
next_2months_bal	10601
next_3months	21184
next_3months_bal	21184
dtype: int64	

Missing value

Clear missing/ null value

```
# Bỏ những feature null value
data_balance = data_balance[~data_balance['prv_year_bal'].isnull()]
```

```
# Fill feature null với avg / mean / max / most frequency value
mean_value = data_balance['prv_year_bal'].mean()

data_balance['prv_year_bal'].fillna(mean_value, inplace = True)
```

🔗 <https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html>

Data với dữ liệu datetime

```
1 data_balance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172134 entries, 0 to 172133
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CIF                   172134 non-null int64
1   cob_dt                172134 non-null object
2   current_bal           172134 non-null float64
3   prv_month             172134 non-null object
4   prv_month_bal         172134 non-null float64
5   prv_year              48187 non-null object
6   prv_year_bal          48187 non-null float64
7   next_1month           172134 non-null object
8   next_1month_bal       172134 non-null float64
9   next_2months          161533 non-null object
10  next_2months_bal      161533 non-null float64
11  next_3months          150950 non-null object
12  next_3months_bal      150950 non-null float64
dtypes: float64(6), int64(1), object(6)
memory usage: 17.1+ MB
```

```
1 data_bal['cob_dt'] = pd.to_datetime(data_bal['cob_dt'])
2 data_balance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172134 entries, 0 to 172133
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CIF                   172134 non-null int64
1   cob dt                172134 non-null datetime64[ns]
2   current_bal           172134 non-null float64
3   prv_month             172134 non-null object
4   prv_month_bal         172134 non-null float64
5   prv_year              48187 non-null object
6   prv_year_bal          48187 non-null float64
7   next_1month           172134 non-null object
8   next_1month_bal       172134 non-null float64
9   next_2months          161533 non-null object
10  next_2months_bal      161533 non-null float64
11  next_3months          150950 non-null object
12  next_3months_bal      150950 non-null float64
dtypes: datetime64[ns](1), float64(6), int64(1), object(5)
memory usage: 17.1+ MB
```

```
1 data_balance['year'] = data_bal['cob_dt'].dt.year
2 data_balance['month'] = data_bal['cob_dt'].dt.month
3 data_balance['day'] = data_bal['cob_dt'].dt.day
4 data_balance.info()
```

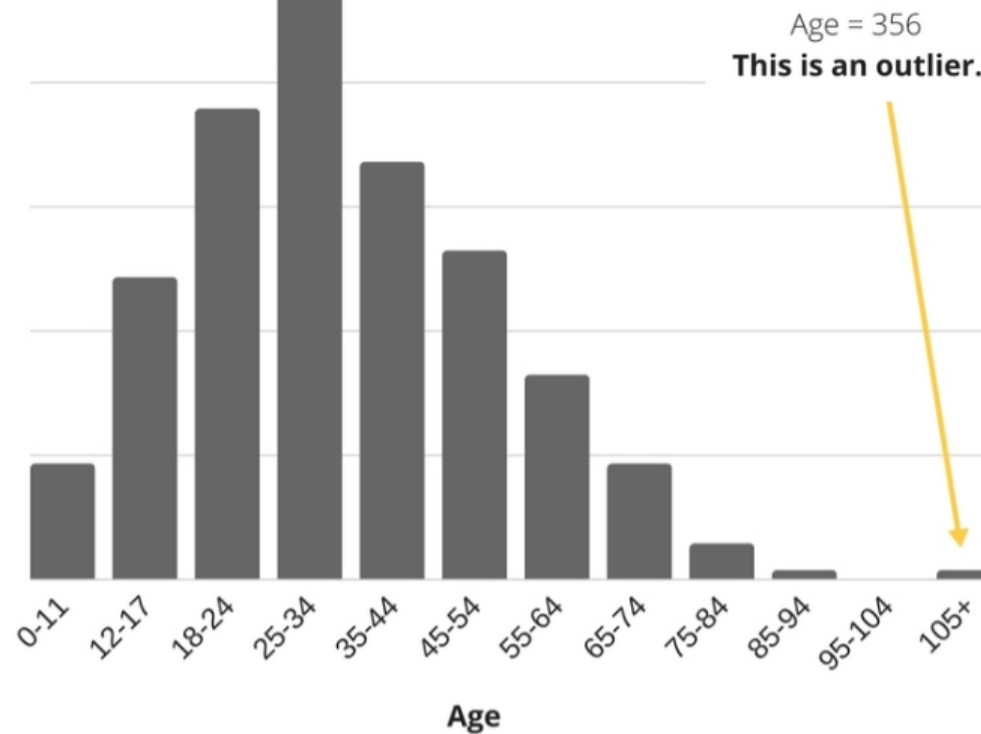
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172134 entries, 0 to 172133
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CIF                   172134 non-null  int64
1   cob_dt                172134 non-null  datetime64[ns]
2   current_bal           172134 non-null  float64
3   prv_month             172134 non-null  object
4   prv_month_bal         172134 non-null  float64
5   prv_year              48187 non-null   object
6   prv_year_bal          48187 non-null   float64
7   next_1month           172134 non-null  object
8   next_1month_bal       172134 non-null  float64
9   next_2months          161533 non-null  object
10  next_2months_bal      161533 non-null  float64
11  next_3months          150950 non-null  object
12  next_3months_bal      150950 non-null  float64
13  year                  172134 non-null  int64
14  month                 172134 non-null  int64
15  day                   172134 non-null  int64
dtypes: datetime64[ns](1), float64(6), int64(4), object(5)
memory usage: 21.0+ MB
```

Tách date với Ngày / Tháng / Năm

Clean data

Un-wanted (outlier) data

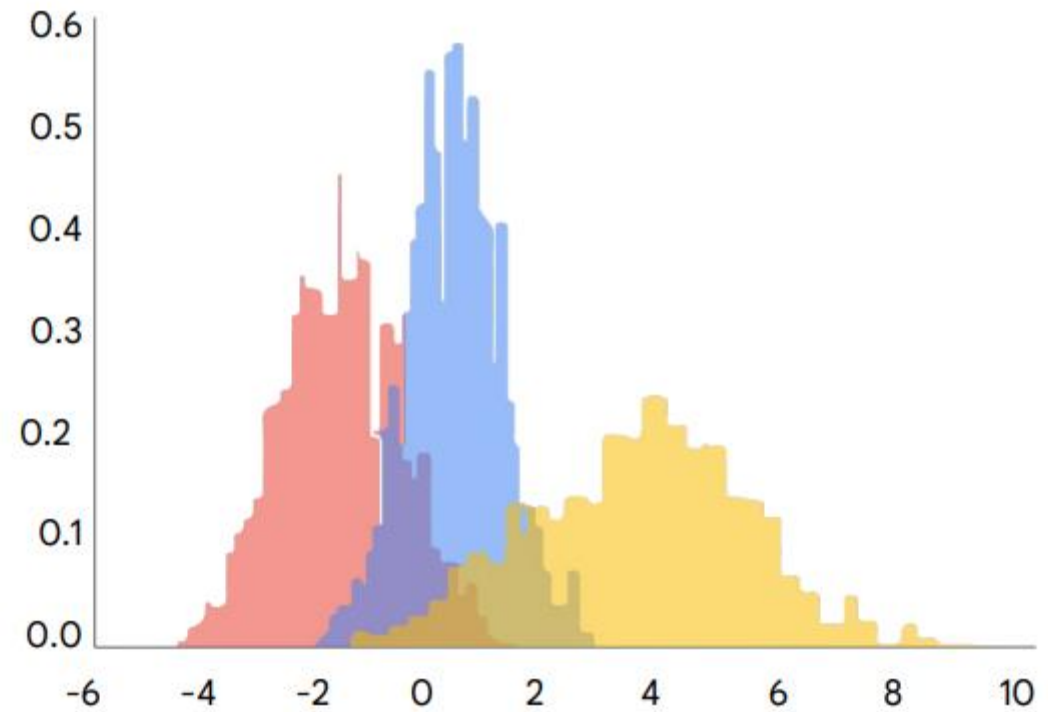
Xóa
Outlier data



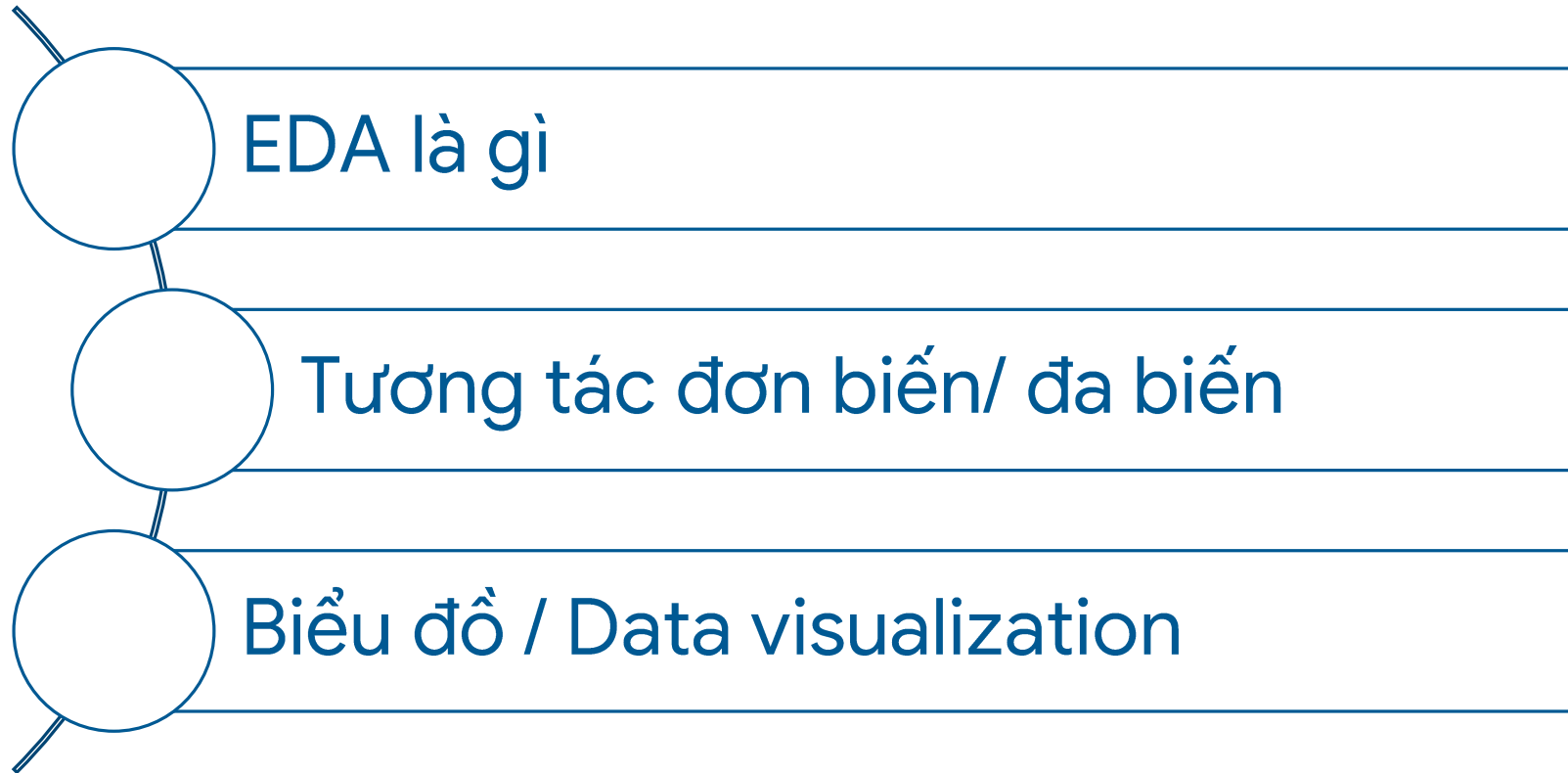
*lưu ý: outlier vs anomaly

EDA

Exploration Data Analysis



EDA





Data



Algorithm



Predictive insight



Decision

EDA là đặc biệt quan trọng trong dữ liệu dạng bảng

Ý nghĩa của các trường dữ liệu

Ý nghĩa của bộ dữ liệu, từng trường dữ liệu. Việc hiểu trường dữ liệu quyết định trực tiếp đến quyết định xử lý/ biến đổi về sau

Phân phối xác suất của từng trường

Phân phối xác suất của từng trường dữ liệu

- Số giá trị trong trường?
- Số giá trị khuyết / null?
- Giá trị không hợp lệ (outlier)?
- Giá trị ngoại lệ (anomaly)?

Kiểu dữ liệu của mỗi trường

*Dạng chuỗi/ số/ ngày tháng.
Tránh bugs khi vận hành model*

Mối tương quan giữa các trường dữ liệu

Mối quan hệ tương quan giữa:

- Trường dữ liệu vs label
- Trường dữ liệu với nhau

Kiểu dữ liệu

Ý nghĩa từng trường dữ liệu

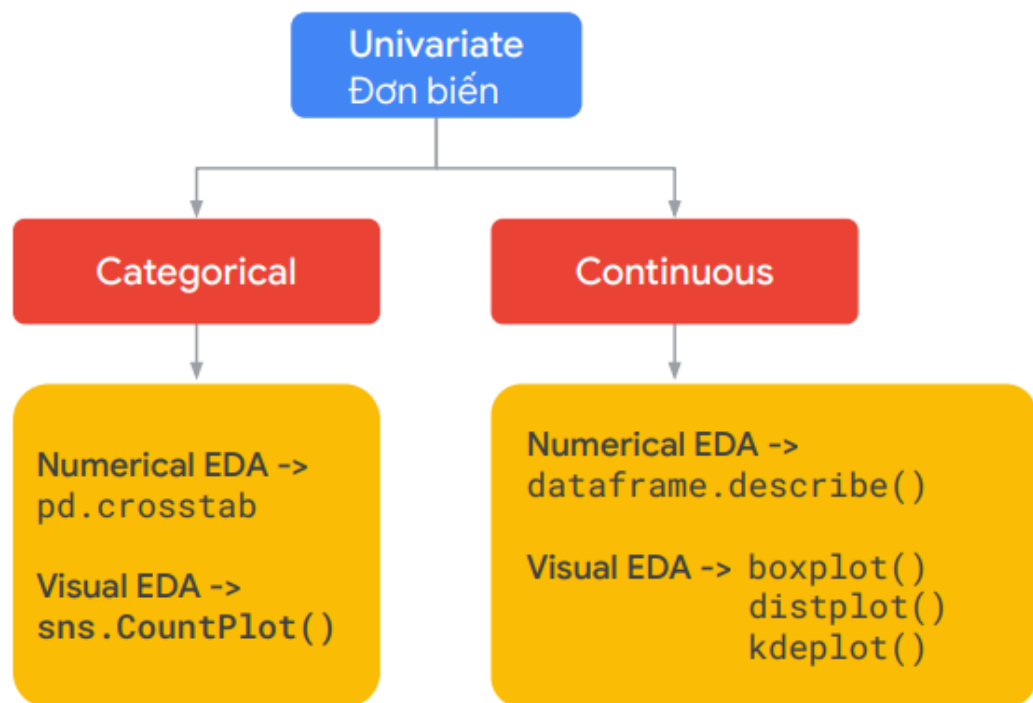
```
1 data_balance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172134 entries, 0 to 172133
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CIF                    172134 non-null int64
1   cob_dt                 172134 non-null object
2   current_bal            172134 non-null float64
3   prv_month              172134 non-null object
4   prv_month_bal          172134 non-null float64
5   prv_year               48187 non-null  object
6   prv_year_bal           48187 non-null  float64
7   next_1month            172134 non-null object
8   next_1month_bal        172134 non-null float64
9   next_2months           161533 non-null object
10  next_2months_bal        161533 non-null float64
11  next_3months            150950 non-null object
12  next_3months_bal        150950 non-null float64
dtypes: float64(6), int64(1), object(6)
memory usage: 17.1+ MB
```

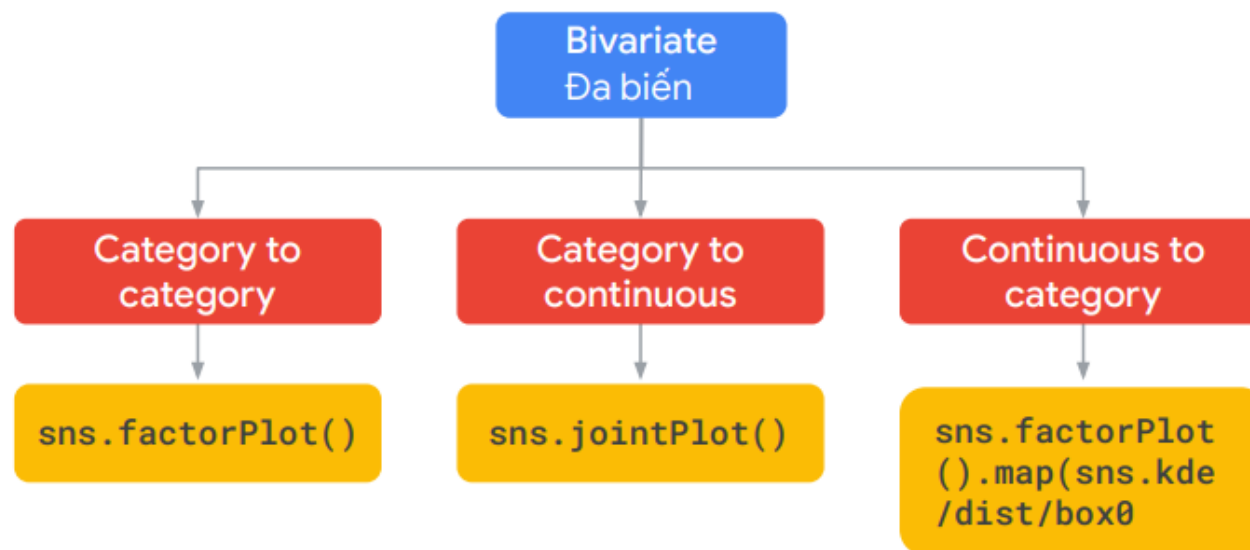

Phân phối xác suất của từng trường dữ liệu (tổng quan)

```
1 data_bal.describe()
```

	cif_no	zbal_current	zbal_prv_month	zbal_prv_year	next_1month_bal
count	1.721340e+05	1.721340e+05	1.721340e+05	4.818700e+04	1.721340e+05
mean	1.131977e+09	2.398324e+09	2.381323e+09	2.265397e+09	2.373845e+09
std	1.233375e+09	3.375059e+09	3.337812e+09	3.082321e+09	3.399013e+09
min	2.000354e+08	2.500000e+01	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.001932e+08	5.000000e+08	5.000000e+08	5.000000e+08	4.799984e+08
50%	3.002620e+08	1.360000e+09	1.350000e+09	1.300000e+09	1.310000e+09
75%	3.014211e+09	3.000000e+09	2.996160e+09	2.930000e+09	2.995641e+09
max	3.017716e+09	1.826119e+11	1.585939e+11	7.452788e+10	1.826119e+11



Phân phối dữ liệu đơn biến

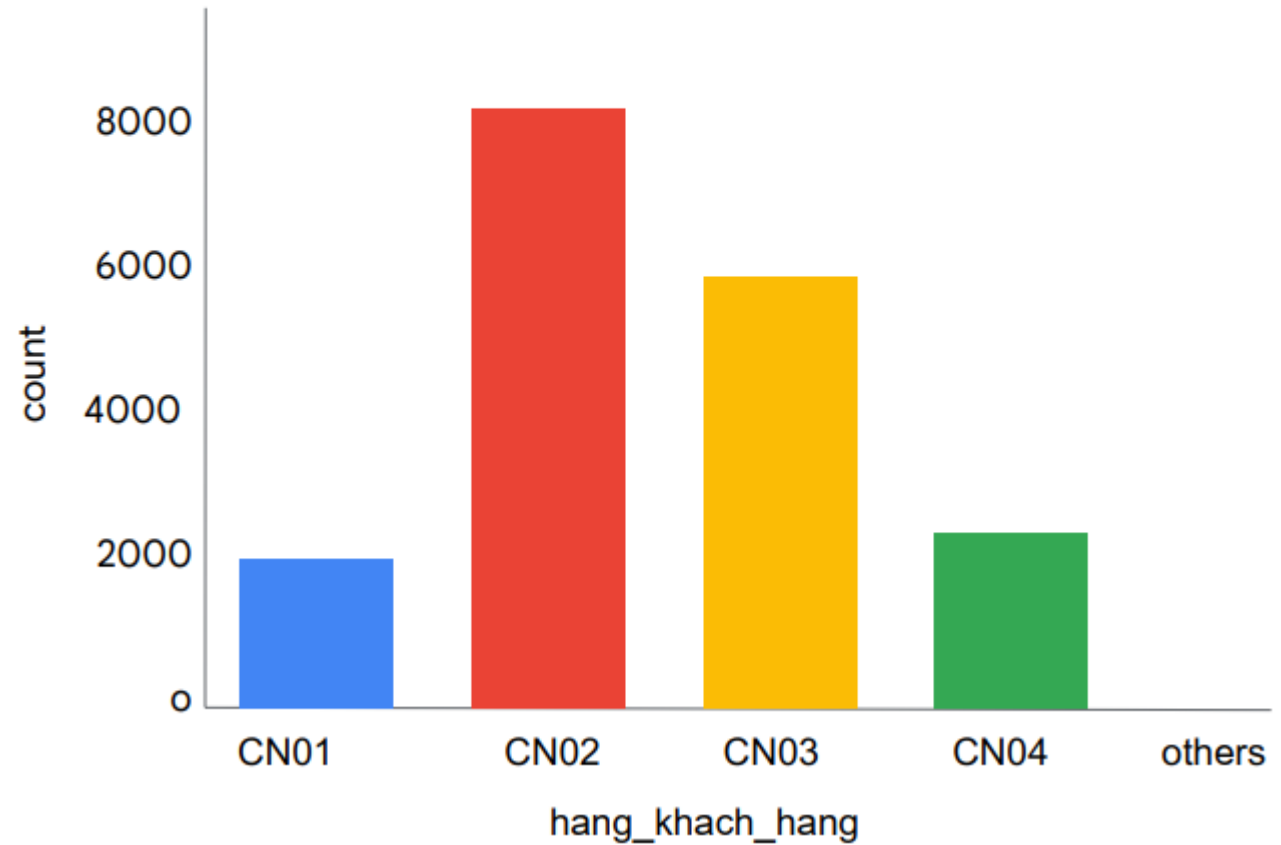


Tương tác đa biến với nhau

Univariate

Đơn biến

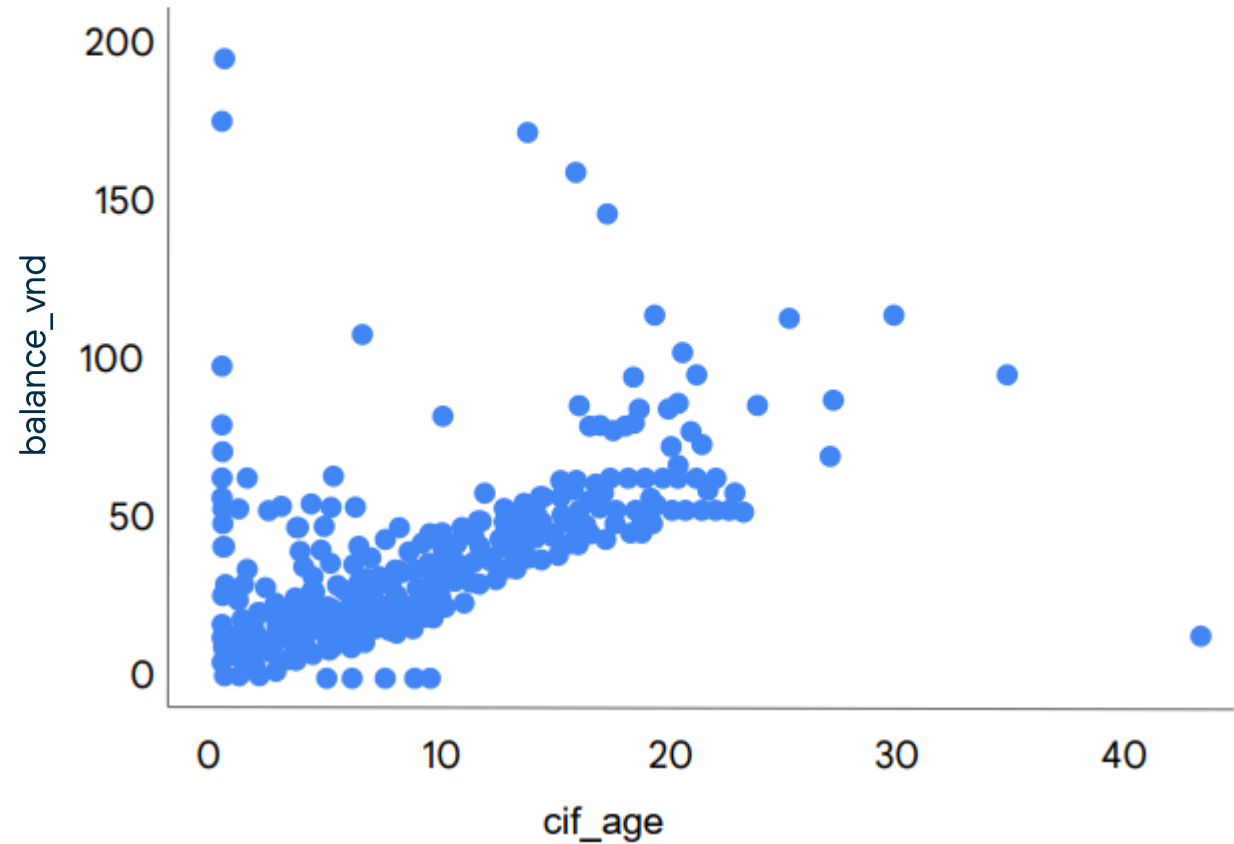
```
[17]: sns.countplot(x = `hang_khach_hang`, data=df)  
[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba4ef400>
```



Bivariate

Đa biến

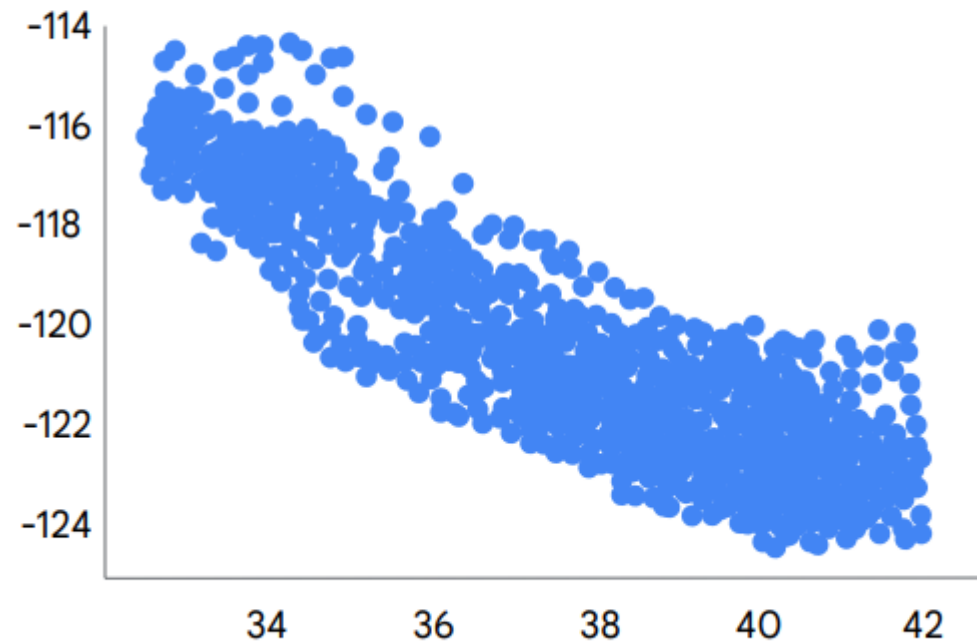
```
ax = sns.regplot(x="t_cif_age", y="f_balance_vnd",  
                 fit_reg=False, ci=None, truncate=True, data=trips)  
ax.figure.set_size_inches(10, 8)
```



Scatter plot

Biểu diễn mối quan hệ của các trường dữ liệu

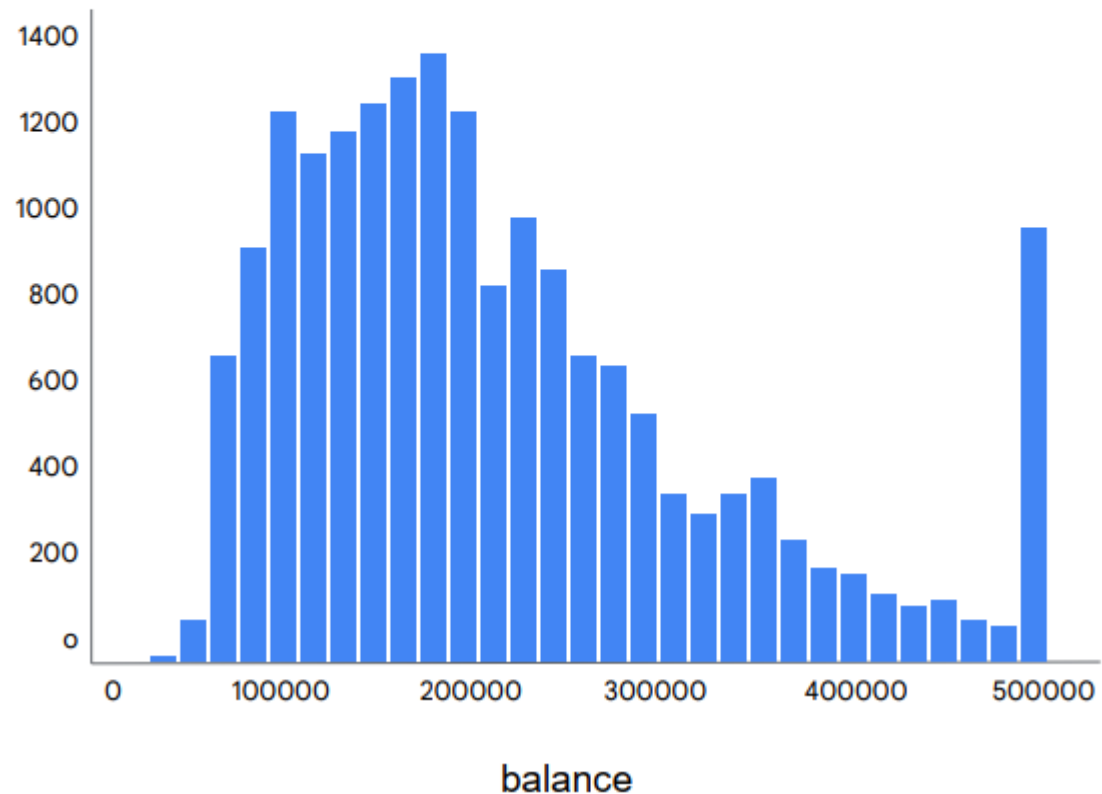
```
X = df_USAhousing[ `latitude` ]  
Y = df_USAhousing[ `longitude` ]  
  
plt.scatter(x,y)  
plt.show()
```



Histogram

Biểu diễn phân phối dữ liệu

```
[14]: sns.set_style('whitegrid')  
data_balance['balance'].hist(bins=30)  
plt.xlabel('balance')  
  
[14]: Text(0.5, 0, 'balance')
```

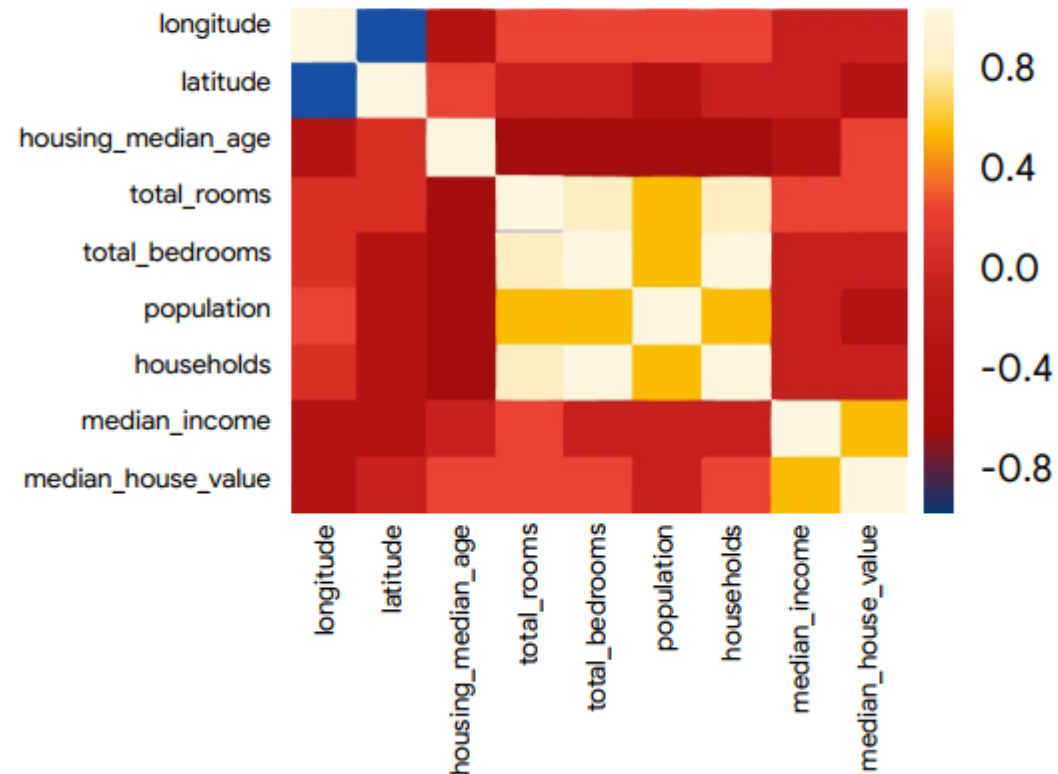


Correlation plot

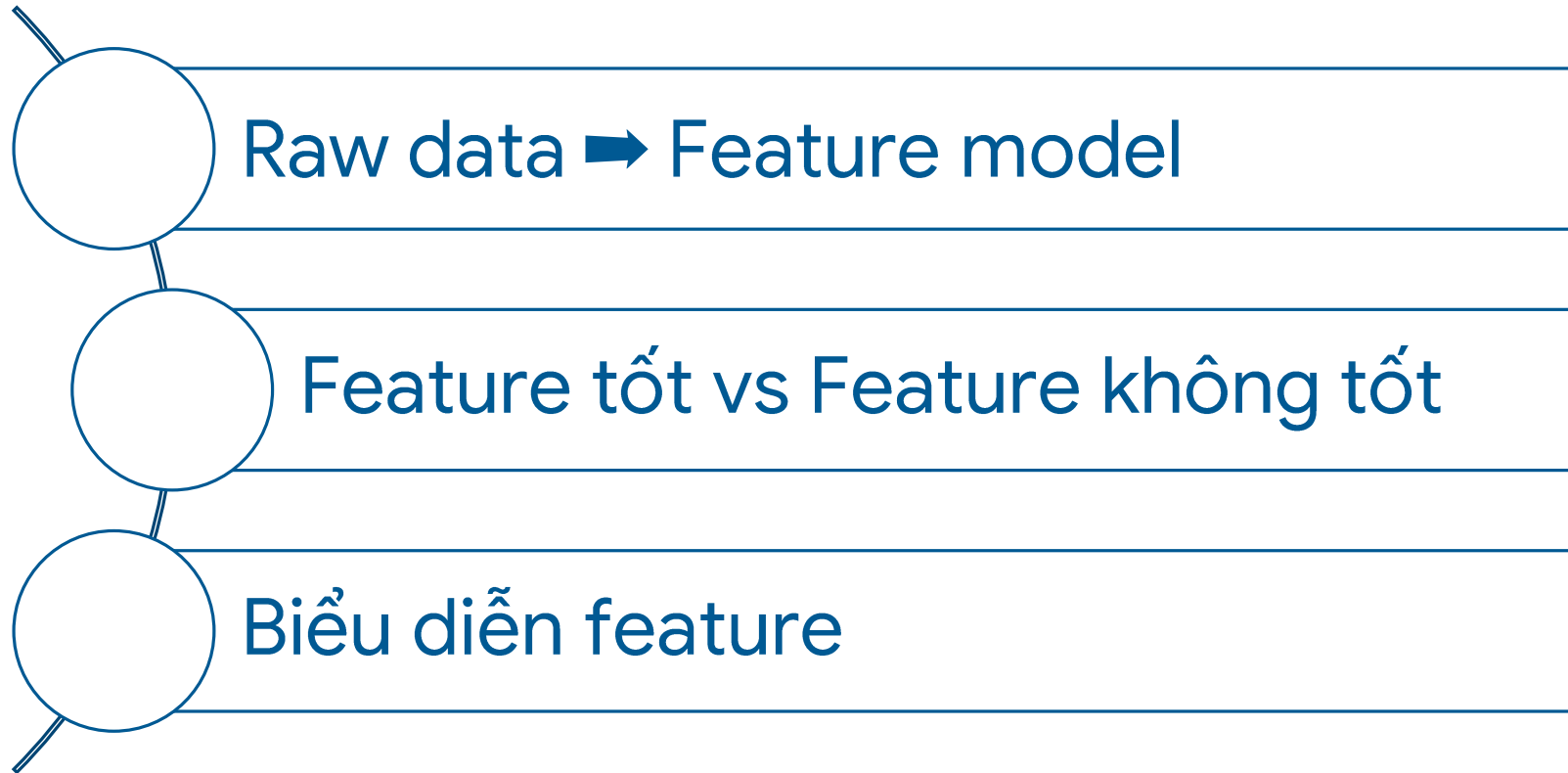
Biểu diễn mối tương quan của các trường dữ liệu

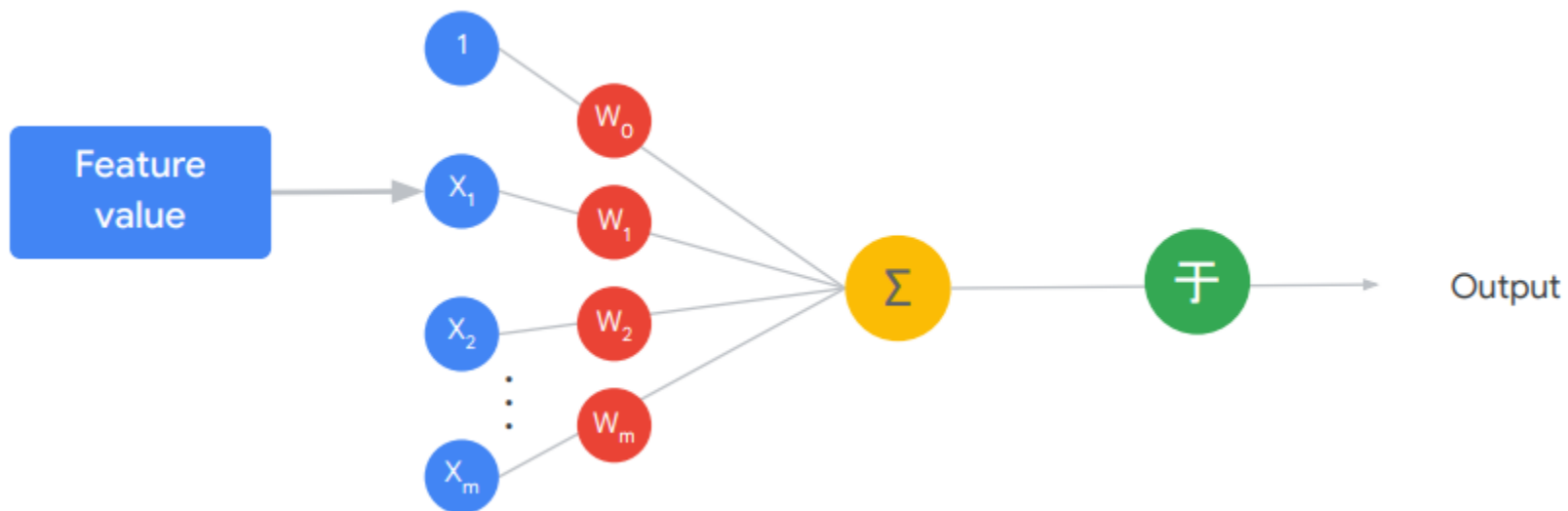
```
[12]: sns.heatmap(df_USAhousing.corr())
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba7c1dd8>
```



Feature Engineering





Feature là đặc điểm có thể đo lường của một mẫu
Được sử dụng làm input của ML model

Là quá trình biến đổi/ tạo mới feature từ raw data thành feature mang khả năng dự đoán trong thuật toán ML

“

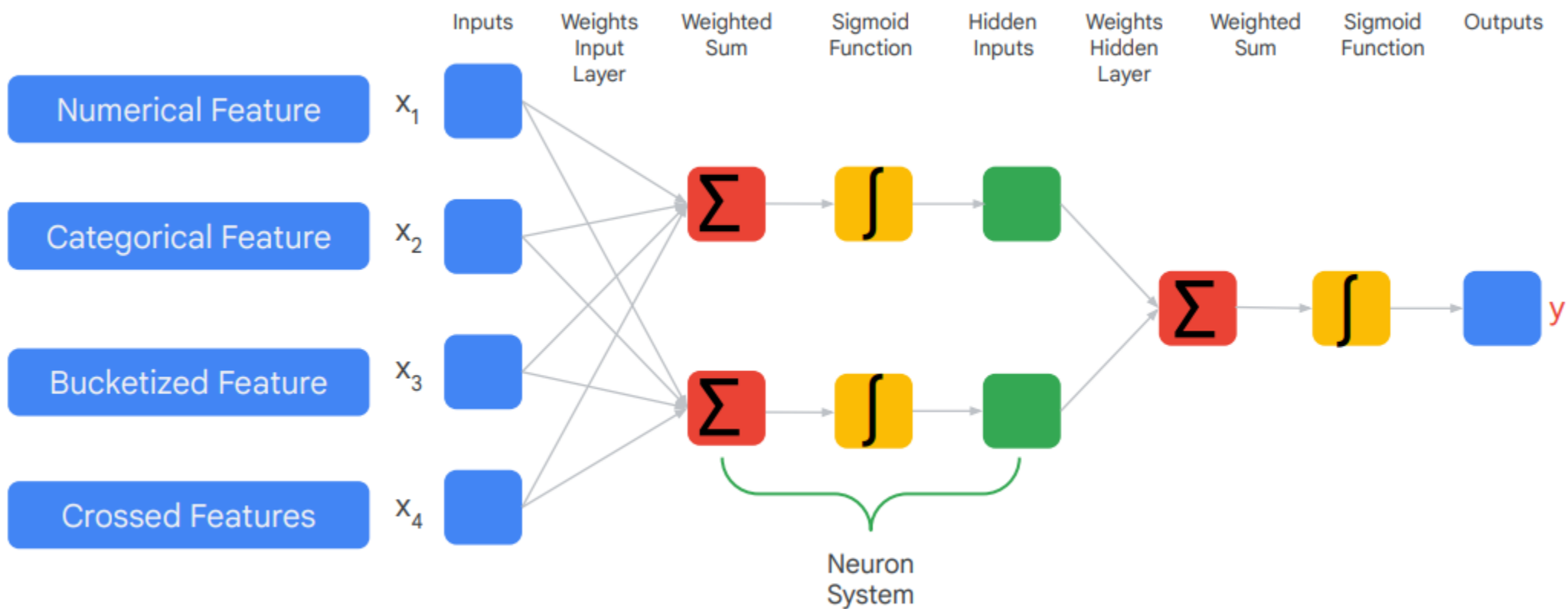
Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in **improved model accuracy** on unseen data.

”

Prof. Andrew Ng

Why FE?

Nhằm cải thiện độ chính xác của thuật toán ML thông qua việc sử dụng dữ liệu đầu vào tốt hơn



Phân tích

- Phân tích yêu cầu
- Tìm kiếm dữ liệu
- EDA

Biểu diễn

- Feature extraction
- Biến đổi feature
- Tạo mới feature

Feature

Numeric

Categorical

Bucketized

Crossed

...

Thế nào là feature tốt?



1

Liên quan đến
bài toán



2

Sẵn sàng tại
thời điểm dự
đoán



3

Dạng số với
định dạng phù hợp



4

Đa dạng mẫu



5

Mang human
insight giải
quyết vấn đề

Phân tích

Data nào cần thiết để giải quyết bài toán dự đoán giá nhà?



Diện tích đất
Số phòng ngủ



Lịch sử
giá nhà



Địa điểm

Input data

dien_tich/
so_phong_ngu/
gara/
loai_dat/
nha_dat/
chung_cu

ML Model

Giá: \$4 tỷ



"Features"

Output

Biểu diễn feature

Feature phải được biểu diễn dưới dạng số

```
{
  "transactionId": 42,
  "name": "Ice Cream", text/categorical
  "price": 2.50, numeric
  "tags": ["cold", "dessert"], text/categorical
  "servedBy": {
    "employeeId": 72365,
    "waitTime": 1.4,
    "customerRating": 4},
  "storeLocation": {
    "latitude": 35.3,
    "longitude": -98.7}
},
```



```
[ ..., 1, 2.50, ..., ]
[ ..., 0, 8.99, ..., ]
[ ..., 0, 3.45, ..., ]
...
```

Biểu diễn feature - numeric

Dạng numeric để nguyên / scale

Biểu diễn feature - id

Dạng đặc biệt/ id nên bỏ

```
{  
  "transactionId": 42,  
  "name": "Ice Cream",  
  "price": 2.50,  
  "tags": ["cold", "dessert"],  
  "servedBy": {  
    "employeeId": 72365,  
    "waitTime": 1.4,  
    "customerRating": 4},  
  "storeLocation": {  
    "latitude": 35.3,  
    "longitude": -98.7}  
},
```



Biểu diễn feature - categorical

Dạng categorical rời rạc nên được encode: one-hot

one-hot
encode

Khu vực
KV1
KV2
KV3
KV4
KV5
KV6



KV1	KV2	KV3	KV4	KV5	KV6
0	0	0	1	0	0
1	0	0	0	0	0

Biểu diễn feature - categorical

Dạng categorical nên được encode: ordinal encode

Khi feature mang tính chất **thứ tự**

ordinal
encode

CIF	Hạng khách hàng
xxx	AA+
yyy	A
zzz	BB-
abc	B



CIF	Hạng khách hàng encoded
xxx	4
yyy	3
zzz	2
abc	1

Biểu diễn feature - categorical

Dạng categorical nên được encode: count encode

Khi feature mang tính **tần suất**

count
encode

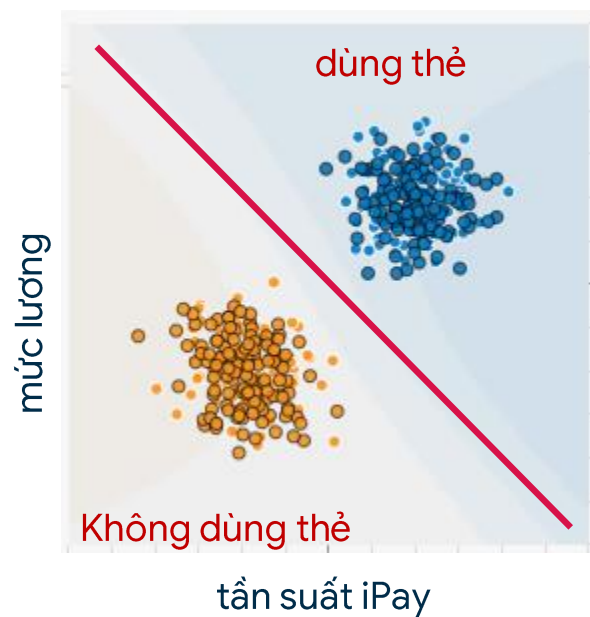
CIF	Chi nhánh	Rời bỏ
xxx	CN HN	0
yyy	CN 1	1
zzz	CN 1	1
abc	CN HN	1



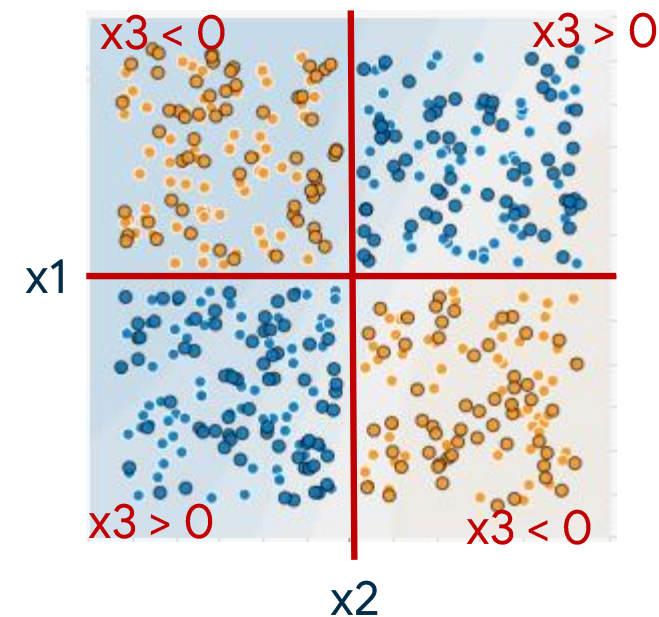
CIF	Chi nhánh
xxx	1
yyy	2
zzz	2
abc	1

Biểu diễn feature – crossed

Question 1?
Vẽ đường phân lớp

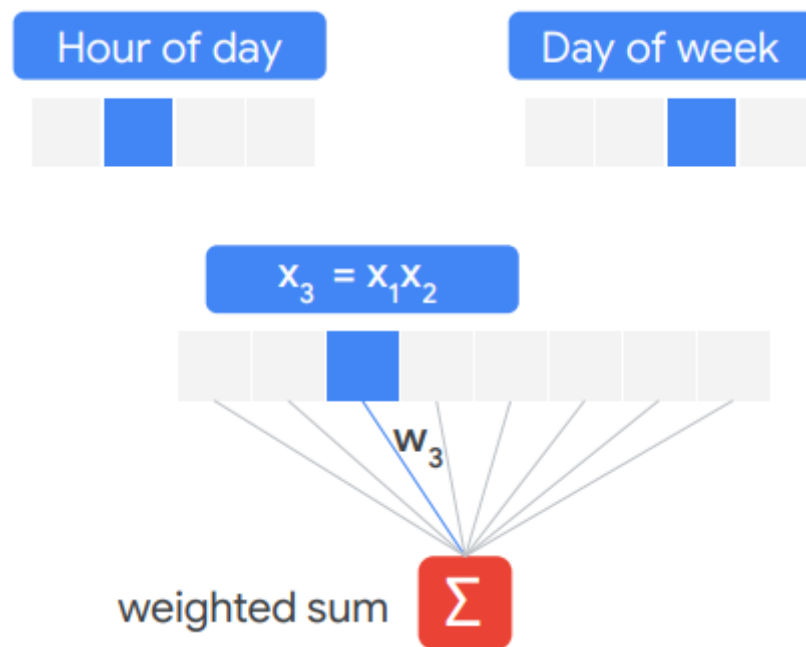


Question 2?
Vẽ đường phân lớp



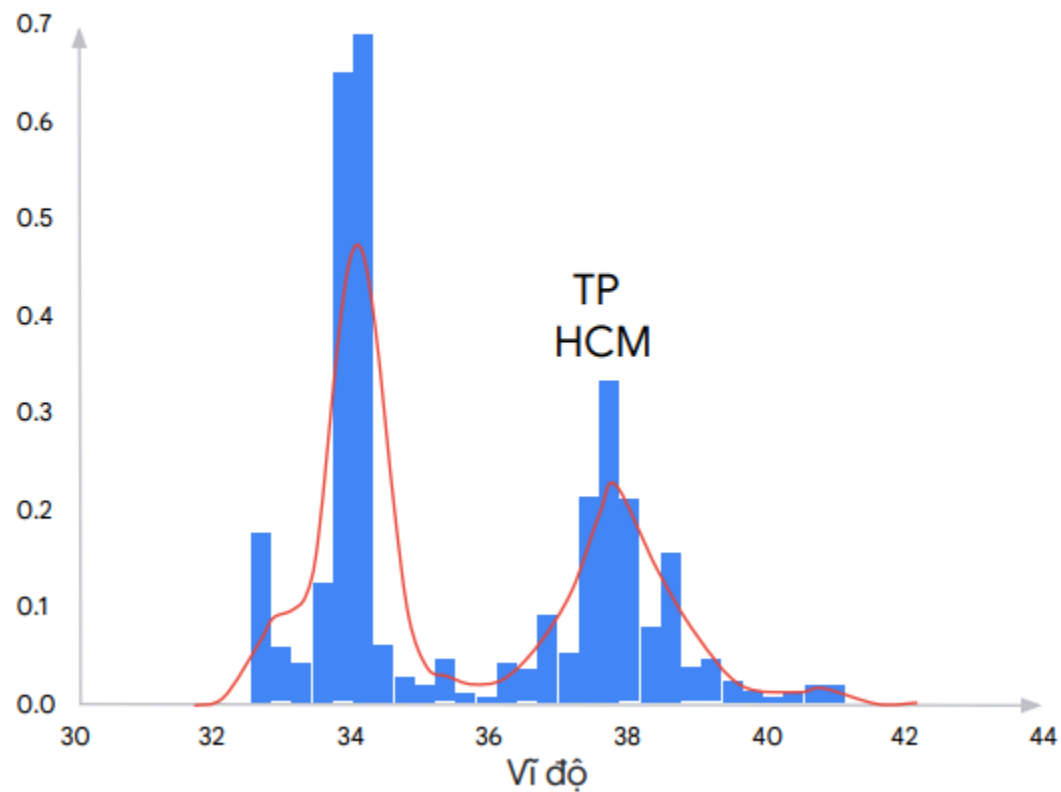
Ý tưởng: $x_3 = x_1 * x_2$

Biểu diễn feature – crossed

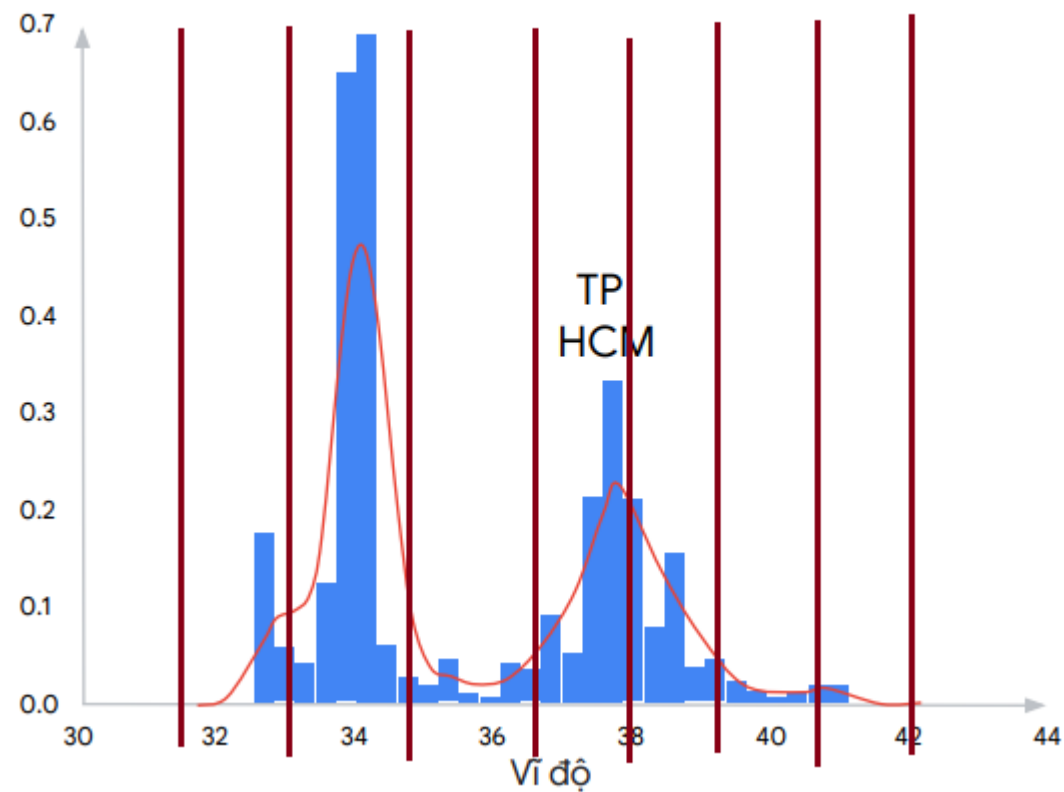


Biểu diễn feature – bins (chia khoảng)

Hà Nội

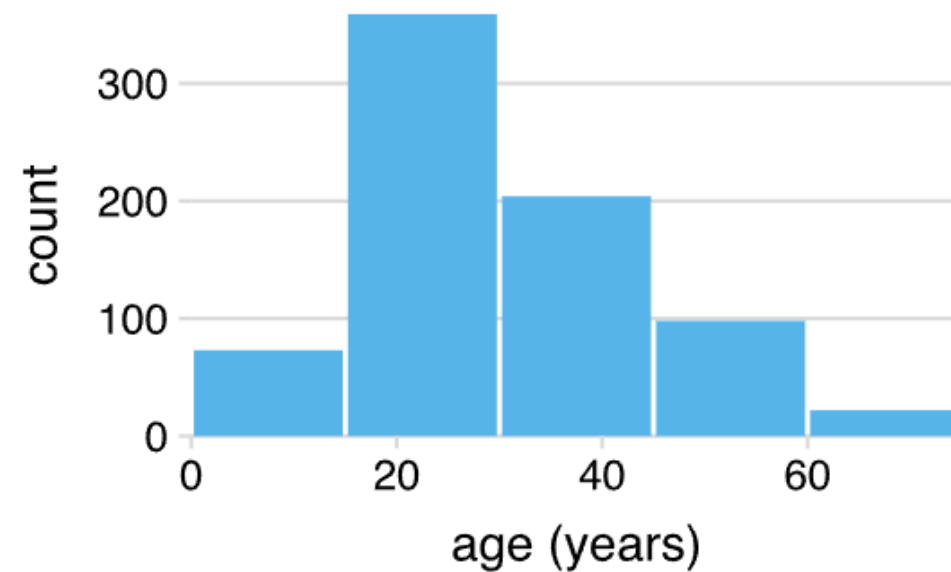
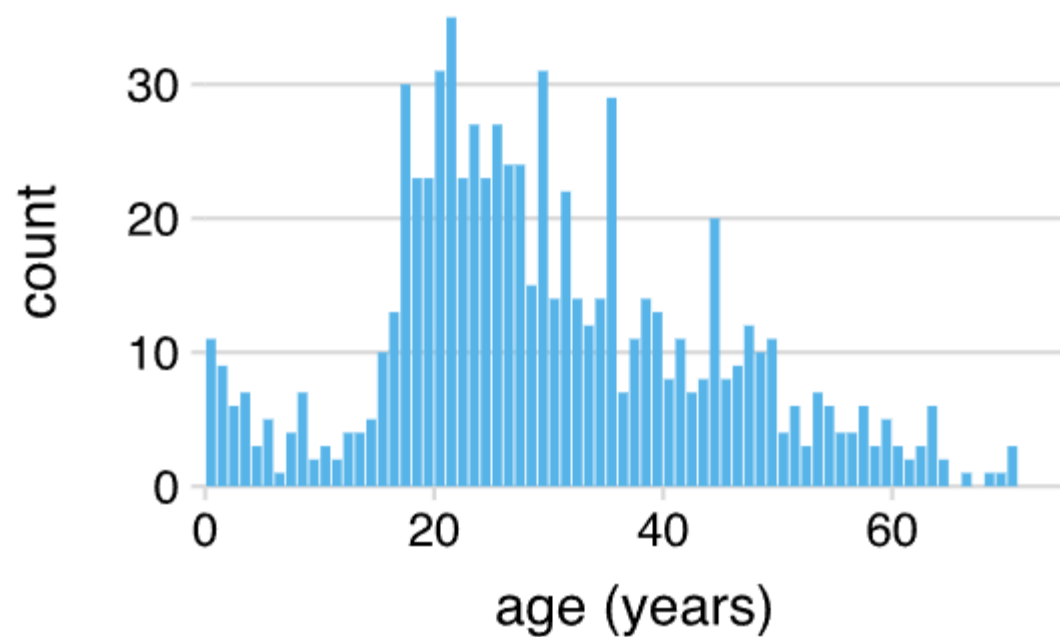


Hà Nội

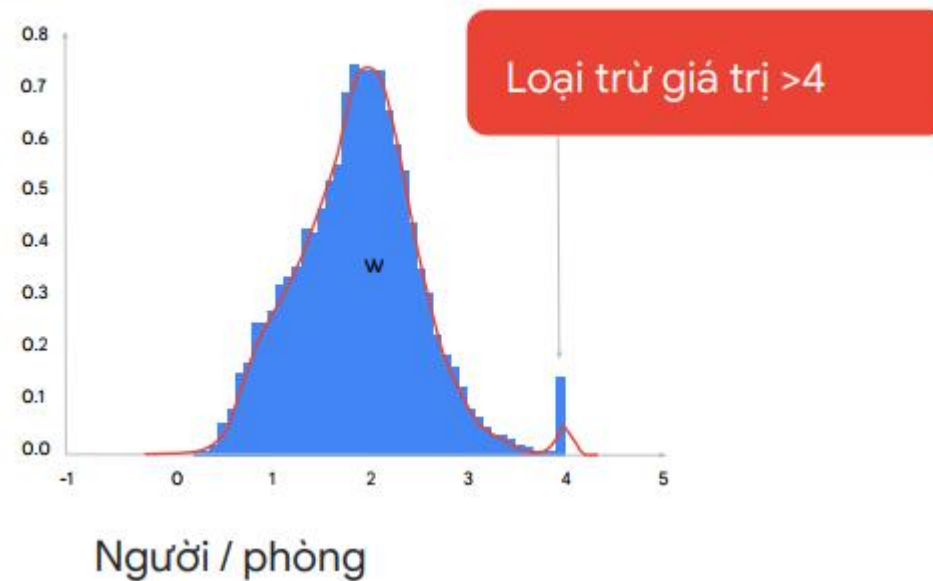
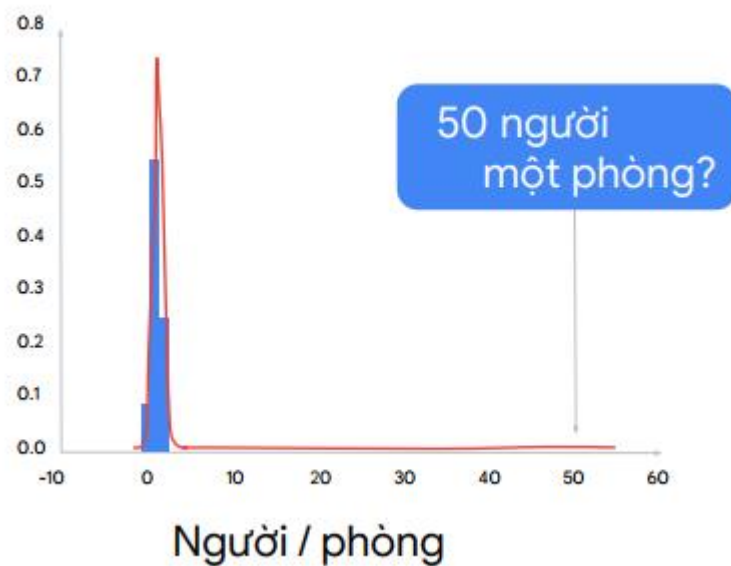


- Số liên tục (continuous number)
- Có tính chất chia khoảng

Biểu diễn feature – bins (chia khoảng)



Biểu diễn feature – clipping (anomaly)



Tạo mới feature

```
ua = 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/56.0.2924.76 Chrome/56.0.2924.76 Safari/537.36'
```

```
# Parser thông tin user agent
```

```
ua = user_agents.parse(ua)
```

```
# Khai thác các thuộc tính của user
```

```
print('Is a bot? ', ua.is_bot)
```

```
print('Is mobile? ', ua.is_mobile)
```

```
print('Is PC? ',ua.is_pc)
```

```
print('OS Family: ',ua.os.family)
```

```
print('OS Version: ',ua.os.version)
```

```
print('Browser Family: ',ua.browser.family)
```

```
print('Browser Version: ',ua.browser.version)
```

Biểu diễn feature – scale



Playground

<https://playground.tensorflow.org>

Lab 3

github.com/VietinBank/training-resources

Reading

- Feature scaling & Normalization

<https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>

- AI For Everyone – Andrew Ng

<https://www.coursera.org/learn/ai-for-everyone>

kahoot.it