

TAUTOLOGY  
INNOVATION  
SCHOOL



# DATA PREPARATION

BY TAUTOLOGY

MADE BY TAUTOLOGY THAILAND  
DO NOT PUBLISH WITHOUT PERMISSION

facebook/tautologyai  
www.tautology.live

# Data Preparation



# NaN

# NaN

- What is NaN?
- Problem of NaN
- Check NaN
- Listwise Deletion
- Code
- Further Reading



# What is NaN?

NaN (Not a Number) คือ การระบุถึงข้อมูลที่ขาดหายไป หรือ missing value ซึ่งอาจเกิดจากความผิดพลาดในการเก็บค่าสถิติ หรือ user กรอกข้อมูลไม่ครบ

|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 0 | 4.0      | NaN   | 1114      |
| 1 | 4.0      | 110.0 | 1088      |
| 2 | 4.0      | 117.0 | 1462      |
| 3 | 3.0      | 93.0  | 123       |
| 4 | NaN      | 92.0  | 1378      |
| 5 | 3.0      | NaN   | 726       |
| 6 | 6.0      | 96.0  | 1649      |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง, พื้นที่ของบ้าน, ราคาบ้าน

# Problem of NaN

ปัญหาของ NaN คืออะไร?



NaN ไม่ใช่ตัวเลข!  
และสิ่งที่ไม่ใช่ตัวเลขไม่สามารถนำไปสร้าง  
โมเดลได้



# Check NaN

เราสามารถตรวจสอบ NaN ผ่าน method .info()

```
1 data_nan.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   NumRooms    6 non-null     float64
1   Area        5 non-null     float64
2   SalePrice   7 non-null     int64
dtypes: float64(2), int64(1)
memory usage: 296.0 bytes
```

# Listwise Deletion

|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 0 | 4.0      | NaN   | 1114      |
| 1 | 4.0      | 110.0 | 1088      |
| 2 | 4.0      | 117.0 | 1462      |
| 3 | 3.0      | 93.0  | 123       |
| 4 | NaN      | 92.0  | 1378      |
| 5 | 3.0      | NaN   | 726       |
| 6 | 6.0      | 96.0  | 1649      |



|              | NumRooms       | Area            | SalePrice       |
|--------------|----------------|-----------------|-----------------|
| <del>0</del> | <del>4.0</del> | <del>NaN</del>  | <del>1114</del> |
| 1            | 4.0            | 110.0           | 1088            |
| 2            | 4.0            | 117.0           | 1462            |
| 3            | 3.0            | 93.0            | 123             |
| <del>4</del> | <del>NaN</del> | <del>92.0</del> | <del>1378</del> |
| <del>5</del> | <del>3.0</del> | <del>NaN</del>  | <del>726</del>  |
| 6            | 6.0            | 96.0            | 1649            |



# Code

|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 0 | 4.0      | NaN   | 1114      |
| 1 | 4.0      | 110.0 | 1088      |
| 2 | 4.0      | 117.0 | 1462      |
| 3 | 3.0      | 93.0  | 123       |
| 4 | NaN      | 92.0  | 1378      |
| 5 | 3.0      | NaN   | 726       |
| 6 | 6.0      | 96.0  | 1649      |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง, พื้นที่ของบ้าน, ราคาบ้าน

# Code

- Check NaN

```
1 data_nan.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 7 entries, 0 to 6  
Data columns (total 3 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   NumRooms    6 non-null      float64  
1   Area        5 non-null      float64  
2   SalePrice   7 non-null      int64  
dtypes: float64(2), int64(1)  
memory usage: 296.0 bytes
```

# Code

- Listwise Deletion

```
1 data = data_nan.dropna(axis=0)
```

# Code

|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 0 | 4.0      | NaN   | 1114      |
| 1 | 4.0      | 110.0 | 1088      |
| 2 | 4.0      | 117.0 | 1462      |
| 3 | 3.0      | 93.0  | 123       |
| 4 | NaN      | 92.0  | 1378      |
| 5 | 3.0      | NaN   | 726       |
| 6 | 6.0      | 96.0  | 1649      |



|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 1 | 5.0      | 100.0 | 1131.0    |
| 2 | 4.0      | 89.0  | 426.0     |
| 3 | 4.0      | 95.0  | 770.0     |
| 6 | 3.0      | 100.0 | 845.0     |

# Further Reading

- Impute Missing Values





# Data Preparation



# Outlier

# Outlier

- What is Outlier?
- Effect of Outliers
- Check Outliers
- Remove Outliers
- Code
- Further Reading

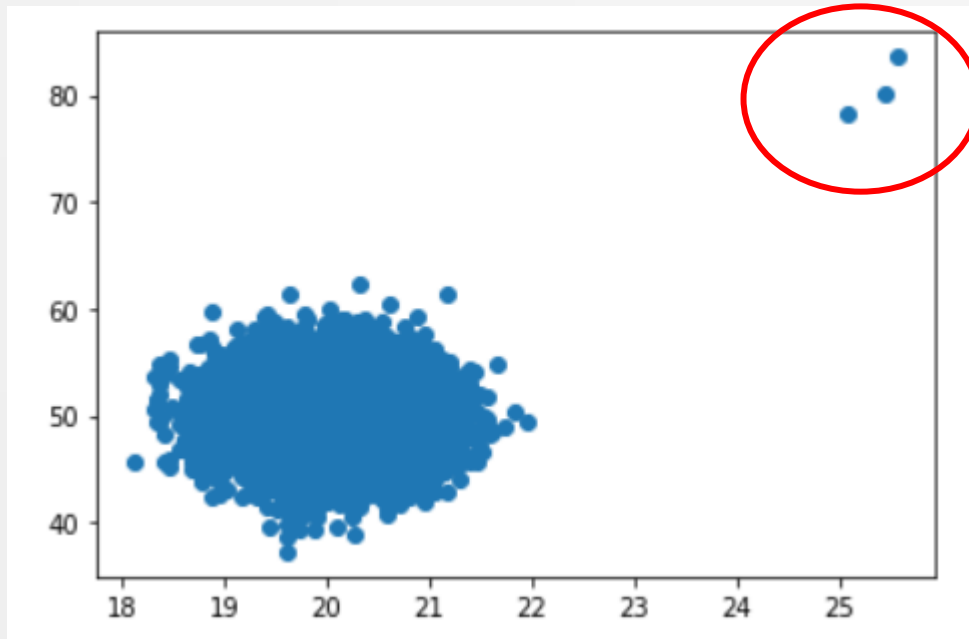
# What is Outlier?

Outlier คือ ข้อมูลที่สูงกว่า หรือ ต่ำกว่าข้อมูลทั่วไปใน feature เดียวกัน อย่างผิดปกติ

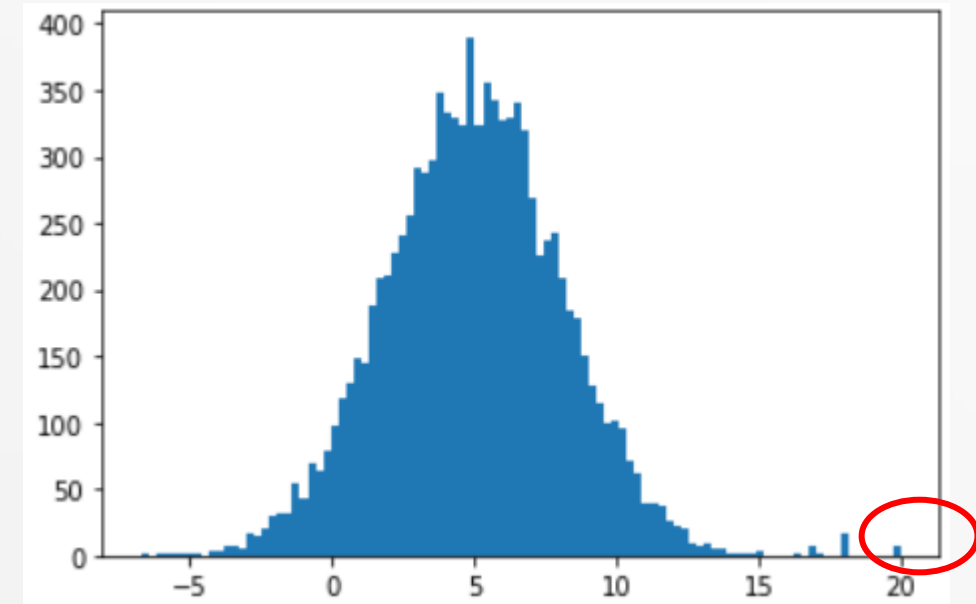
|   | NumRooms | Area   | SalePrice |
|---|----------|--------|-----------|
| 0 | -300     | -100   | 560       |
| 1 | 4        | 107    | 1388      |
| 2 | 3        | 105    | 1013      |
| 3 | 5        | 114    | 1811      |
| 4 | 100000   | 100    | 1344      |
| 5 | 3        | 900000 | 1055      |
| 6 | 3        | 105    | 820       |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง, พื้นที่ของบ้านและ ราคาบ้าน

# What is Outlier?



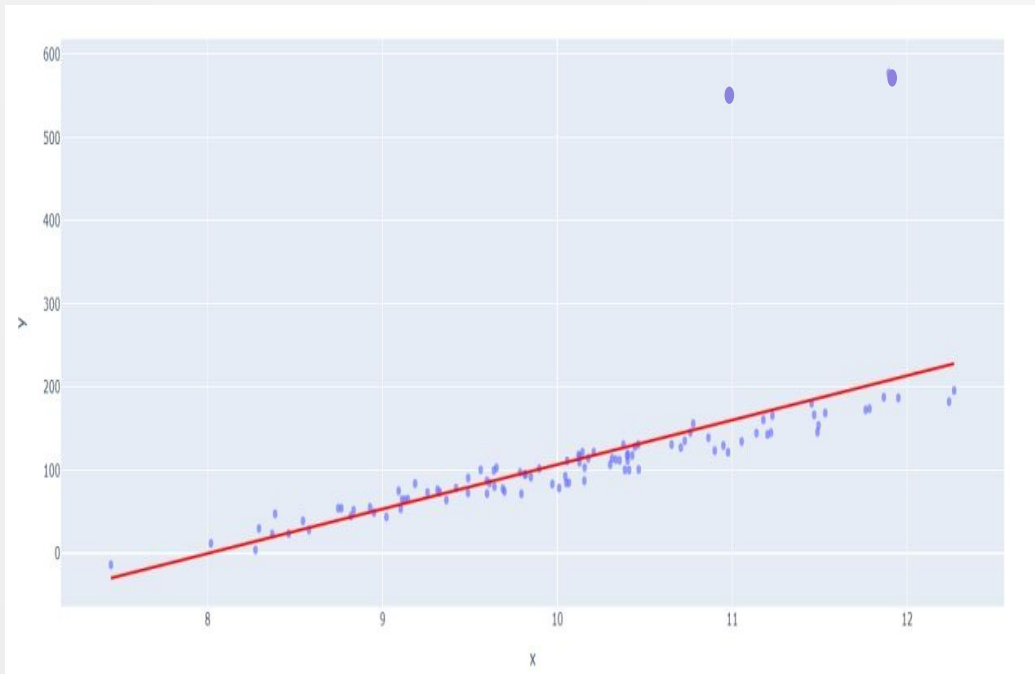
กราฟของข้อมูลระหว่างเงินเดือน(พันบาท) และ  
อายุของแต่ละคน(ปี)



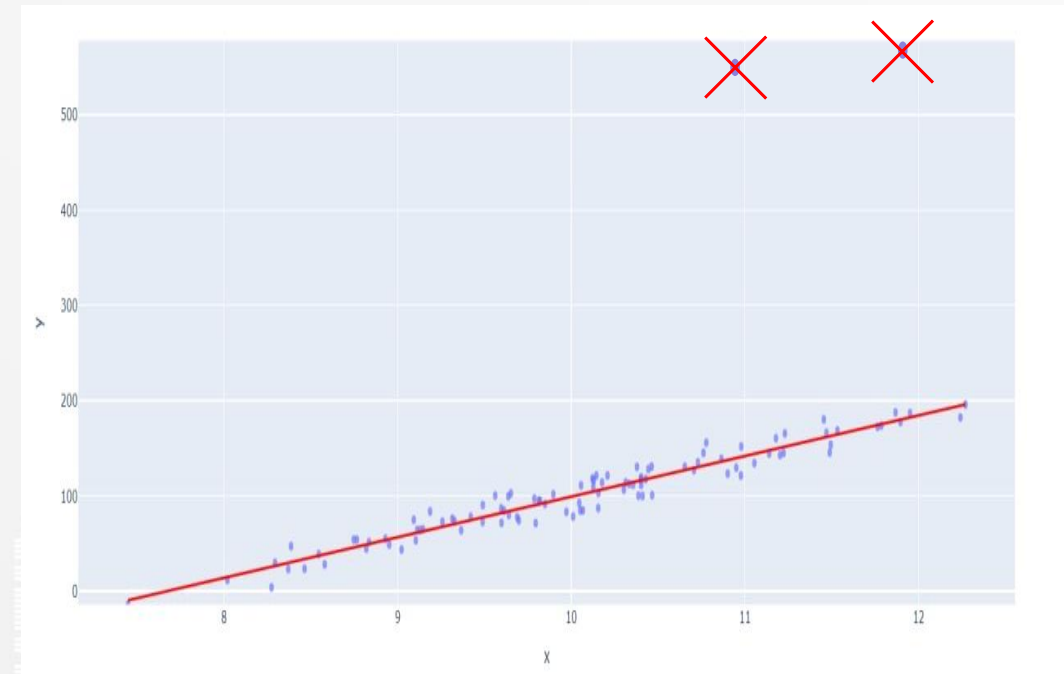
กราฟ Histogram ของอุณหภูมิสำหรับเก็บยา



# Effect of Outliers



ข้อมูลที่มี Outliers



ข้อมูลที่ไม่มี Outliers

# Check Outliers

เราสามารถตรวจสอบ outliers ผ่าน method .describe()

|              | NumRooms  | Area      | SalePrice |
|--------------|-----------|-----------|-----------|
| <b>count</b> | 7.00      | 7.00      | 7.00      |
| <b>mean</b>  | 14245.43  | 128633.00 | 1141.57   |
| <b>std</b>   | 37814.38  | 340140.88 | 411.40    |
| <b>min</b>   | -300.00   | -100.00   | 560.00    |
| <b>25%</b>   | 3.00      | 102.50    | 916.50    |
| <b>50%</b>   | 3.00      | 105.00    | 1055.00   |
| <b>75%</b>   | 4.50      | 110.50    | 1366.00   |
| <b>max</b>   | 100000.00 | 900000.00 | 1811.00   |

# Remove Outliers

|   | NumRooms | Area   | SalePrice |
|---|----------|--------|-----------|
| 0 | -300     | -100   | 560       |
| 1 | 4        | 107    | 1388      |
| 2 | 3        | 105    | 1013      |
| 3 | 5        | 114    | 1811      |
| 4 | 100000   | 100    | 1344      |
| 5 | 3        | 900000 | 1055      |
| 6 | 3        | 105    | 820       |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง, พื้นที่ของบ้าน, ราคาบ้าน

# Code

|   | NumRooms | Area   | SalePrice |
|---|----------|--------|-----------|
| 0 | -300     | -100   | 560       |
| 1 | 4        | 107    | 1388      |
| 2 | 3        | 105    | 1013      |
| 3 | 5        | 114    | 1811      |
| 4 | 100000   | 100    | 1344      |
| 5 | 3        | 900000 | 1055      |
| 6 | 3        | 105    | 820       |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง, พื้นที่ของบ้าน, ราคาบ้าน

# Code

- Check Outliers

```
1 data_outlier.describe()
```

|       | NumRooms      | Area          | SalePrice   |
|-------|---------------|---------------|-------------|
| count | 7.000000      | 7.000000      | 7.000000    |
| mean  | 14245.428571  | 128633.000000 | 1141.571429 |
| std   | 37814.380910  | 340140.883966 | 411.399586  |
| min   | -300.000000   | -100.000000   | 560.000000  |
| 25%   | 3.000000      | 102.500000    | 916.500000  |
| 50%   | 3.000000      | 105.000000    | 1055.000000 |
| 75%   | 4.500000      | 110.500000    | 1366.000000 |
| max   | 100000.000000 | 900000.000000 | 1811.000000 |



# Code

- Remove Outliers

```
1 _filter = (0 < data_outlier['NumRooms']) & (data_outlier['NumRooms'] < 10)  
2           & (0 < data_outlier['Area']) & (data_outlier['Area'] < 1000)  
3 data = data_outlier[_filter]
```

# Code

|   | NumRooms | Area   | SalePrice |
|---|----------|--------|-----------|
| 0 | -300     | -100   | 560       |
| 1 | 4        | 107    | 1388      |
| 2 | 3        | 105    | 1013      |
| 3 | 5        | 114    | 1811      |
| 4 | 100000   | 100    | 1344      |
| 5 | 3        | 900000 | 1055      |
| 6 | 3        | 105    | 820       |



|   | NumRooms | Area  | SalePrice |
|---|----------|-------|-----------|
| 1 | 4.0      | 110.0 | 1088      |
| 2 | 4.0      | 117.0 | 1462      |
| 3 | 3.0      | 93.0  | 123       |
| 6 | 6.0      | 96.0  | 1649      |

# Further Reading

- Standard Deviation Method
- Interquartile Range Method
- Isolation Forest
- Minimum Covariance Determinant
- Local Outliers Factor
- One-Class SVM



# Data Preparation





# Feature Encoding



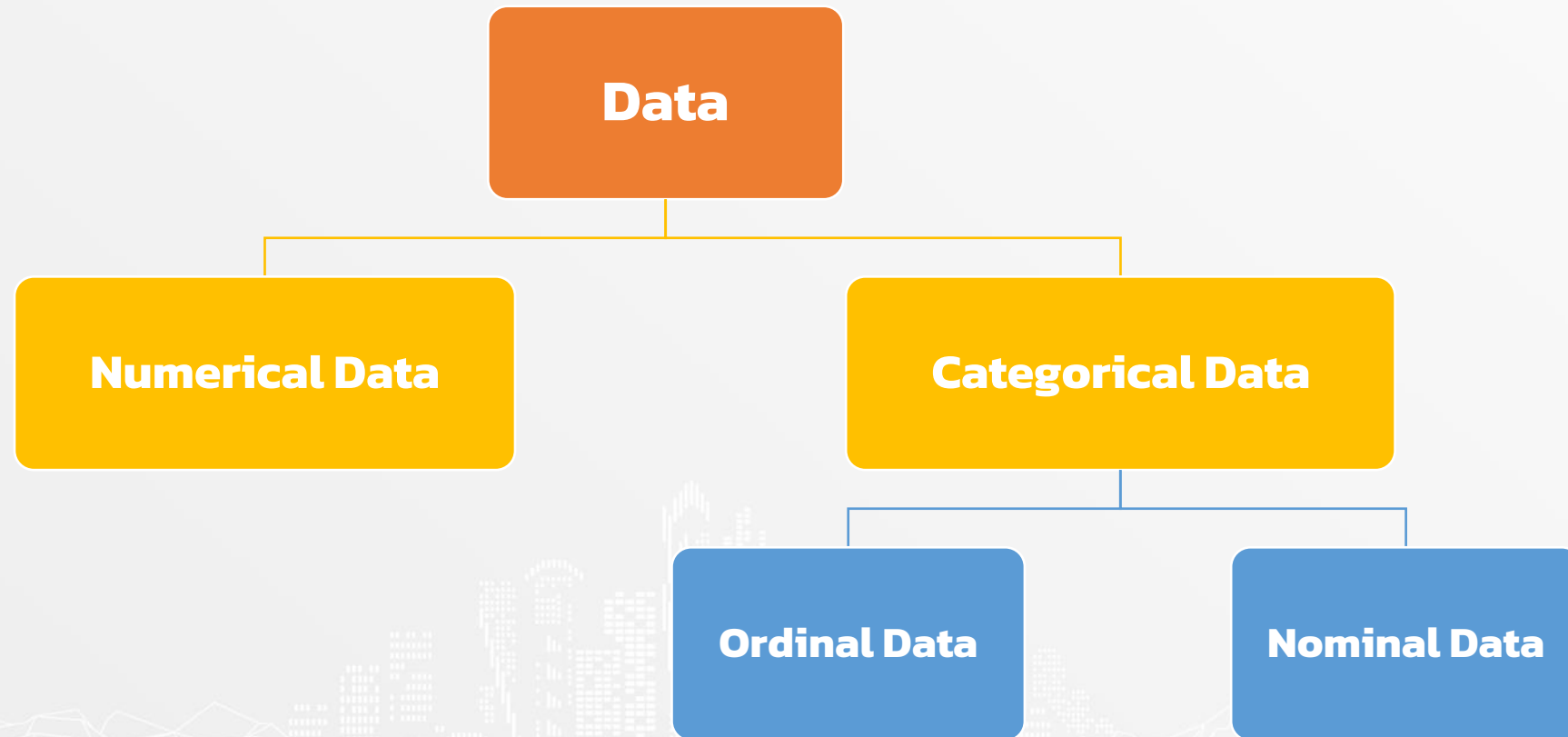
# Feature Encoding

Type of Data

Ordinal  
Encoding

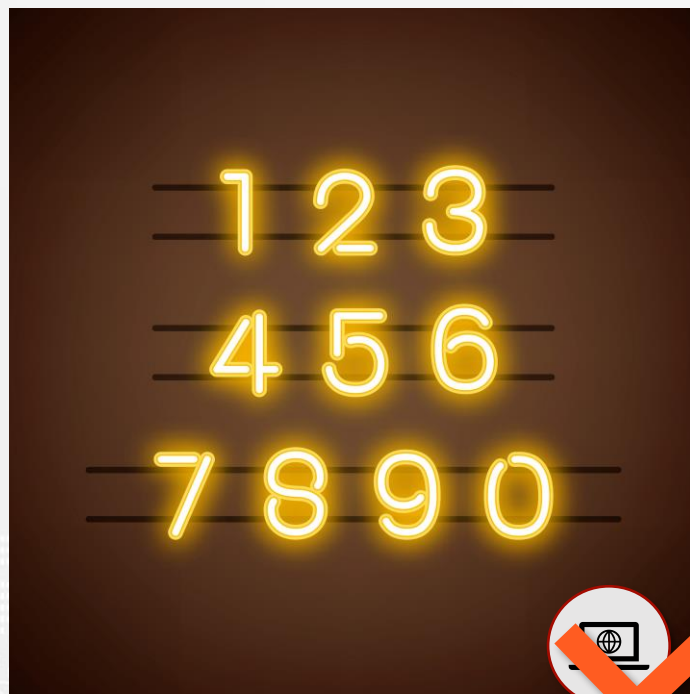
One Hot  
Encoding

# Type of Data

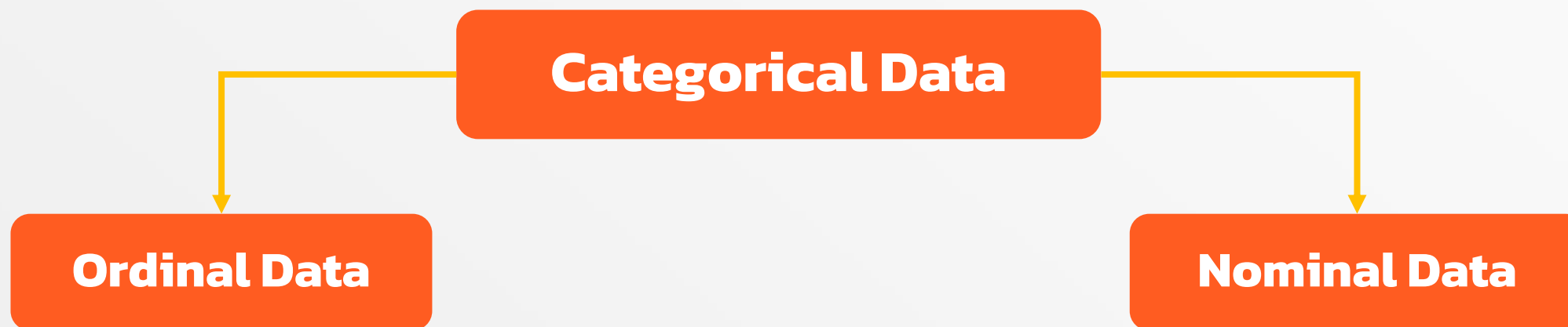


# Numerical Data

Numerical Data คือ ข้อมูลที่ใช้แทนจำนวน อาจอยู่ในรูปของจำนวนเต็ม หรือ ทศนิยม



# Categorical Data



# Ordinal Data

**Ordinal Data** คือ categorical data ที่มีการเรียงลำดับอย่างชัดเจน และไม่สามารถสลับลำดับได้ (ลำดับมีความหมาย)

**เช่น** เหรียญทอง เงิน ทองแดง, เกรด A B C D F, คะแนนแบบประเมิน



A B C D F



# Nominal Data

**Nominal Data** คือ categorical data ที่ไม่มีลำดับของข้อมูล  
**เช่น** ชาย/หญิง, วันหยุด/วันธรรมดา, ประเภทของการขนส่ง, สัญชาติ





# Categorical Data



# Feature Encoding

**Type of Data**



**Ordinal  
Encoding**



**One Hot  
Encoding**



# Ordinal Encoding

- What is Ordinal Encoding?
- How to define number
- Example
- Code

# What is Ordinal Encoding?

**Ordinal Encoding** คือ การแปลง ordinal data ให้อยู่ในรูปแบบของ numerical data ที่มีระยะห่างเท่ากัน



↓

2 1 0



↓

0 1 2 3 4

A B C D F → 4 3 2 1 0



# How to define number



|    |    |   |   |  |
|----|----|---|---|--|
| 2  | 1  | 0 | ✓ | ★★ เหมาะกับการใช้งาน<br>ทางคอมพิวเตอร์ |
| 1  | 2  | 3 | ✓ |  |
| -1 | 0  | 1 | ✓ |  |
| 1  | -1 | 2 | ✗ |  |
| 1  | 1  | 2 | ✗ |  |

# Example

|   | Grade | Medal  |
|---|-------|--------|
| 0 | B     | Gold   |
| 1 | A     | Gold   |
| 2 | B     | Silver |
| 3 | D     | Bronze |
| 4 | F     | Bronze |
| 5 | C     | Silver |

ตารางแสดงผลการเรียนรู้และ เหรียญรางวัลที่ได้



# Example

|   | Grade | Medal  |
|---|-------|--------|
| 0 | B     | Gold   |
| 1 | A     | Gold   |
| 2 | B     | Silver |
| 3 | D     | Bronze |
| 4 | F     | Bronze |
| 5 | C     | Silver |

**Grade**

**A = 4, B = 3, C = 2, D = 1, F = 0**

**Medal**

**Gold = 2, Silver = 1, Bronze = 0**



|   | Grade | Medal |
|---|-------|-------|
| 0 | 3     | 2     |
| 1 | 4     | 2     |
| 2 | 3     | 1     |
| 3 | 1     | 0     |
| 4 | 0     | 0     |
| 5 | 2     | 1     |

# Code

|   | grade | medal  |
|---|-------|--------|
| 0 | B     | gold   |
| 1 | A     | gold   |
| 2 | B     | silver |
| 3 | D     | bronze |
| 4 | F     | bronze |
| 5 | C     | silver |

ตารางแสดงผลการเรียนรู้และ เหรียญรางวัลที่ได้

# Code

```
1 from sklearn.preprocessing import OrdinalEncoder
2
3 categories = [
4     np.array(['F', 'D', 'C', 'B', 'A']),
5     np.array(['bronze', 'silver', 'gold'])
6 ]
7
8 ordinal_encoder = OrdinalEncoder(categories=categories)
9 data_transformed = ordinal_encoder.fit_transform(data)
10
11 data_transformed = pd.DataFrame(data_transformed, columns=feature_name)
```

# Code

|   | grade | medal  |
|---|-------|--------|
| 0 | B     | gold   |
| 1 | A     | gold   |
| 2 | B     | silver |
| 3 | D     | bronze |
| 4 | F     | bronze |
| 5 | C     | silver |



|   | grade | medal |
|---|-------|-------|
| 0 | 3     | 2     |
| 1 | 4     | 2     |
| 2 | 3     | 1     |
| 3 | 1     | 0     |
| 4 | 0     | 0     |
| 5 | 2     | 1     |

# Feature Encoding

**Type of Data**



**Ordinal  
Encoding**



**One Hot  
Encoding**



# One Hot Encoding

- What is One Hot Encoding?
- Example
- Code



# What is One Hot Encoding?

One Hot Encoding คือ การแปลง nominal data ให้อยู่ในรูปแบบของ numerical data โดยแบ่งข้อมูลเป็นหลาย ๆ column ตามชนิดของข้อมูล และกำหนดค่าแต่ละ column ในรูปแบบของ binary (0 หรือ 1)



# Example

|   | Sex    | Transport |
|---|--------|-----------|
| 0 | Male   | Bus       |
| 1 | Female | Train     |
| 2 | Female | Car       |
| 3 | Male   | Train     |
| 4 | Female | Bus       |
| 5 | Male   | Bus       |

ตารางแสดงข้อมูลเพศ และวิธีการเดินทาง

# Example

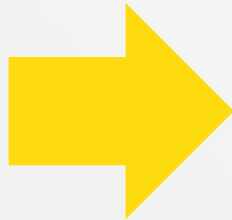
|   | Sex    | Transport |
|---|--------|-----------|
| 0 | Male   | Bus       |
| 1 | Female | Train     |
| 2 | Female | Car       |
| 3 | Male   | Train     |
| 4 | Female | Bus       |
| 5 | Male   | Bus       |



|   | Female | Male | Transport |
|---|--------|------|-----------|
| 0 | 0      | 1    | Bus       |
| 1 | 1      | 0    | Train     |
| 2 | 1      | 0    | Car       |
| 3 | 0      | 1    | Train     |
| 4 | 1      | 0    | Bus       |
| 5 | 0      | 1    | Bus       |

# Example

|   | Sex    | Transport |
|---|--------|-----------|
| 0 | Male   | Bus       |
| 1 | Female | Train     |
| 2 | Female | Car       |
| 3 | Male   | Train     |
| 4 | Female | Bus       |
| 5 | Male   | Bus       |



|   | Female | Male | Bus | Car | Train |
|---|--------|------|-----|-----|-------|
| 0 | 0      | 1    | 1   | 0   | 0     |
| 1 | 1      | 0    | 0   | 0   | 1     |
| 2 | 1      | 0    | 0   | 1   | 0     |
| 3 | 0      | 1    | 0   | 0   | 1     |
| 4 | 1      | 0    | 1   | 0   | 0     |
| 5 | 0      | 1    | 1   | 0   | 0     |

# Code

|   | sex    | transport |
|---|--------|-----------|
| 0 | male   | bus       |
| 1 | female | train     |
| 2 | female | car       |
| 3 | male   | train     |
| 4 | female | bus       |
| 5 | male   | bus       |

ตารางแสดงข้อมูลเพศ และวิธีการเดินทาง

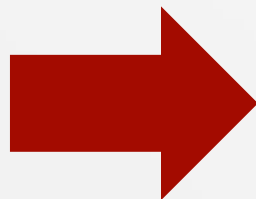
# Code

```
1 from sklearn.preprocessing import OneHotEncoder
2
3 one_hot_encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
4 data_transformed = one_hot_encoder.fit_transform(data)
5
6 data_transformed = pd.DataFrame(data_transformed,
7                                 columns=['female', 'male', 'bus', 'car', 'train'])
```



# Code

|   | sex    | transport |
|---|--------|-----------|
| 0 | male   | bus       |
| 1 | female | train     |
| 2 | female | car       |
| 3 | male   | train     |
| 4 | female | bus       |
| 5 | male   | bus       |



|   | female | male | bus | car | train |
|---|--------|------|-----|-----|-------|
| 0 | 0.0    | 1.0  | 1.0 | 0.0 | 0.0   |
| 1 | 1.0    | 0.0  | 0.0 | 0.0 | 1.0   |
| 2 | 1.0    | 0.0  | 0.0 | 1.0 | 0.0   |
| 3 | 0.0    | 1.0  | 0.0 | 0.0 | 1.0   |
| 4 | 1.0    | 0.0  | 1.0 | 0.0 | 0.0   |
| 5 | 0.0    | 1.0  | 1.0 | 0.0 | 0.0   |

# Feature Encoding

**Type of Data**



**Ordinal  
Encoding**



**One Hot  
Encoding**



# Data Preparation



# Feature Scaling

# Feature Encoding

What is Feature Scaling?

Why need Feature Scaling?

Standardization

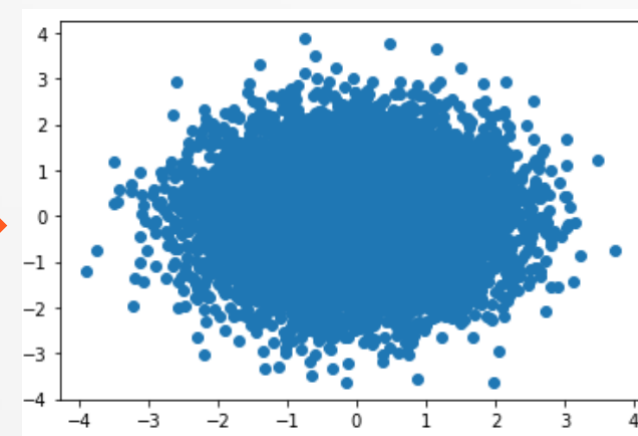
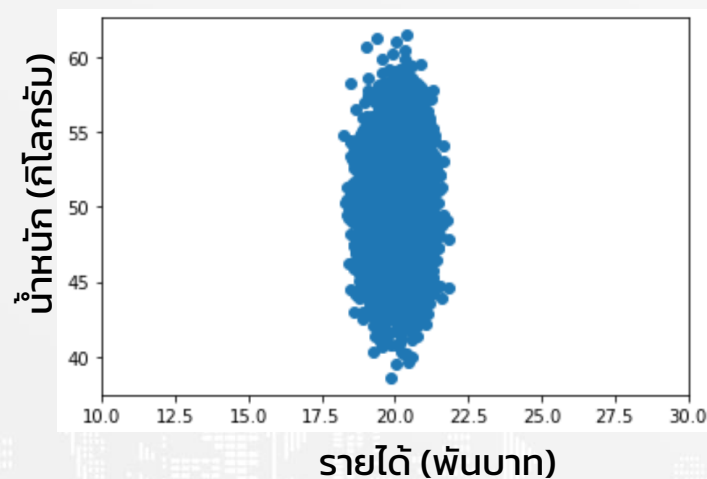
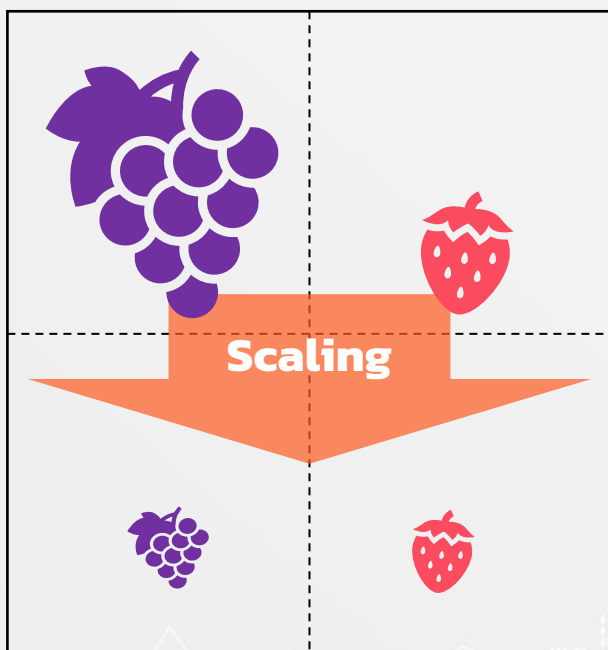
Min-Max Scaling

Conclusion



# What is Feature Scaling?

Feature Scaling คือ การทำให้ทุก feature อยู่ใน scale เดียวกัน





# Feature Encoding

**What is Feature Scaling?**



**Why need Feature Scaling?**



**Standardization**



**Min-Max Scaling**

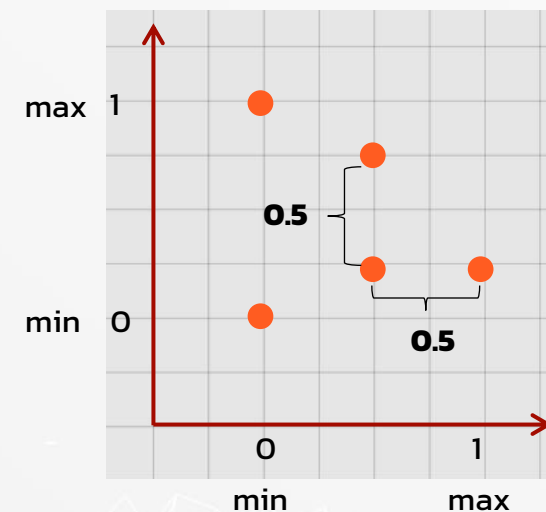
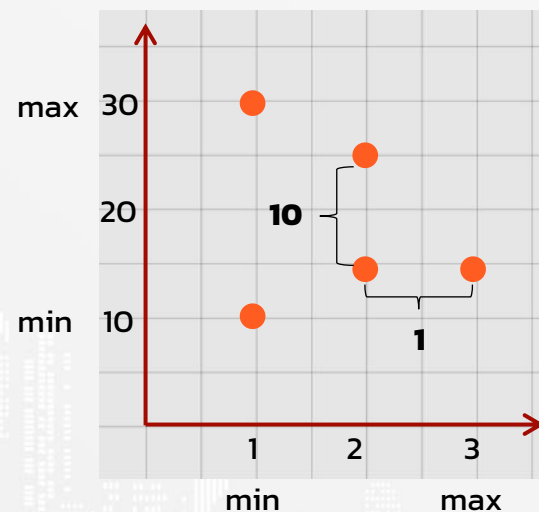
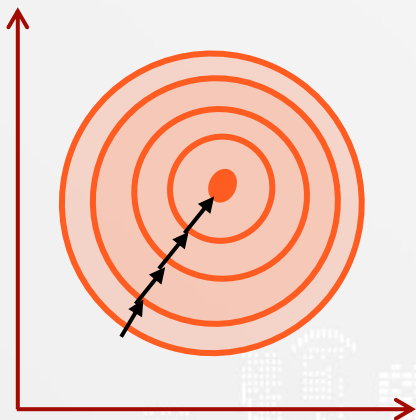
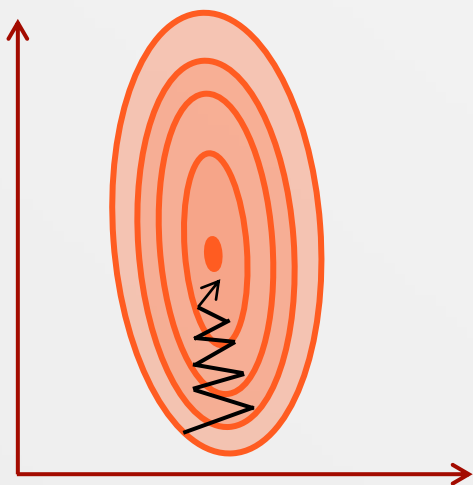


**Conclusion**



# Why need Feature Scaling?

เพื่อแก้ปัญหา bias จาก scale ที่ไม่เท่ากันของแต่ละ feature



# Why need Feature Scaling?

- ทำให้ model ประเภท distance-based model มีประสิทธิภาพดีขึ้น เช่น k nearest neighbor, support vector machine
- ทำให้ model ประเภท gradient-based model เรียนรู้ได้เร็วขึ้น เช่น multiple regression, logistic regression, deep learning
- ทำให้ใช้ cost ในการคำนวณน้อยลง สำหรับบาง dataset
- ทำให้ใช้ memory ในการคำนวณน้อยลง สำหรับบาง dataset

# Feature Encoding

**What is Feature  
Scaling?**



**Why need  
Feature Scaling?**



**Standardization**



**Min-Max Scaling**



**Conclusion**

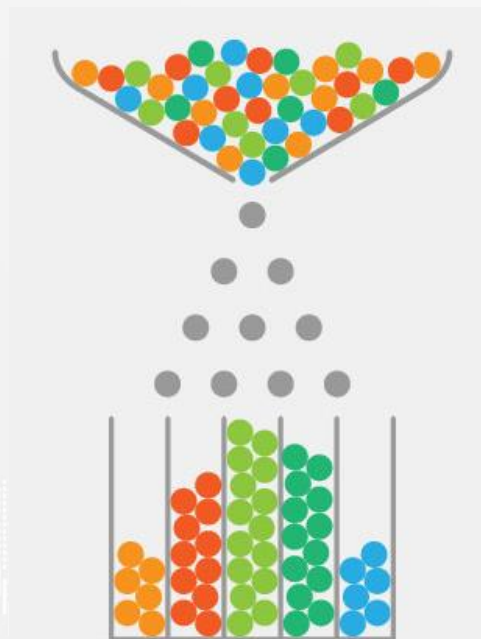


# Standardization

- What is Standardization?
- Formula
- Step to Calculate Standardization
- Example
- Code

# What is Standardization?

Standardization คือ เทคนิคการปรับ scale ของข้อมูลให้มีค่า mean เป็น 0 และ standard deviation เป็น 1



Ref: <https://www.appsflyer.com/blog/data-standardization-effective-analysis/>



# Formula

$$x' = \frac{x - mean}{s.d.}$$

- $x$  คือ ข้อมูลแต่ละตัวใน feature ที่กำลังพิจารณา
- $mean$  คือ ค่าเฉลี่ยของ feature ที่กำลังพิจารณา
- $s.d$  คือ ค่าส่วนเบี่ยงเบนมาตรฐานของ feature ที่กำลังพิจารณา

# Step to Calculate Standardization

1. หาค่า *mean* และ *s.d* ของแต่ละ feature
2. ปรับค่าข้อมูลแต่ละตัวใน feature ตามสูตรของ standardization

# Example

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |

เลือก feature ที่ต้องการจะ  
ทำ feature scaling



| Area |
|------|
| 99   |
| 110  |
| 117  |
| 93   |
| 92   |
| 99   |
| 96   |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง  
และพื้นที่ของบ้าน

# Example

**Area = [99.0, 110.0, 117.0, 93.0, 92.0, 99.0, 96.0]**

วิธีการปรับค่าด้วย standardization มีดังต่อไปนี้

1. หาค่า *mean* และ *s.d* ของข้อมูล Area

- *mean* = 100.86

- *s.d* = 8.58

2. ปรับค่าข้อมูลแต่ละตัวใน Area ตามสูตรของ standardization

$$x' = \frac{x - \text{mean}}{s.d} = \frac{x - 100.86}{8.58}$$

# Example

| Area  | Area_scaled   |
|-------|---|
| 99.0  | $= \frac{99 - 100.86}{8.58} = \frac{-1.86}{8.58} = -0.22$ |
| 110.0 | $= \frac{110 - 100.86}{8.58} = \frac{9.14}{8.58} = 1.07$  |
| 117.0 | $= \frac{117 - 100.86}{8.58} = \frac{16.14}{8.58} = 1.88$ |
| 93.0  | $= \frac{93 - 100.86}{8.58} = \frac{-7.86}{8.58} = -0.92$ |
| 92.0  | $= \frac{92 - 100.86}{8.58} = \frac{-8.86}{8.58} = -1.03$ |
| 99.0  | $= \frac{99 - 100.86}{8.58} = \frac{-1.86}{8.58} = -0.22$ |
| 96.0  | $= \frac{96 - 100.86}{8.58} = \frac{-4.86}{8.58} = -0.57$ |

# Example

| Area  | Area_scaled |
|-------|-------------|
| 99.0  | -0.22       |
| 110.0 | 1.07        |
| 117.0 | 1.88        |
| 93.0  | -0.92       |
| 92.0  | -1.03       |
| 99.0  | -0.22       |
| 96.0  | -0.57       |



# Code

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |

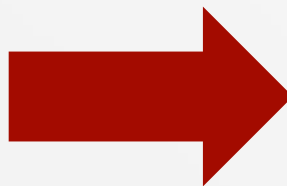
ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง และพื้นที่ของบ้าน

# Code

```
1 from sklearn.preprocessing import StandardScaler
2
3 scaler = StandardScaler()
4 data_scaled = scaler.fit_transform(data)
5
6 data_scaled = pd.DataFrame(data_scaled, columns=feature_name)
```

# Code

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |



|   | NumRooms  | Area      |
|---|-----------|-----------|
| 0 | -0.144338 | -0.216546 |
| 1 | -0.144338 | 1.066075  |
| 2 | -0.144338 | 1.882288  |
| 3 | -1.154701 | -0.916158 |
| 4 | 0.866025  | -1.032760 |
| 5 | -1.154701 | -0.216546 |
| 6 | 1.876388  | -0.566352 |

# Feature Encoding

**What is Feature  
Scaling?**



**Why need  
Feature Scaling?**



**Standardization**



**Min-Max Scaling**



**Conclusion**

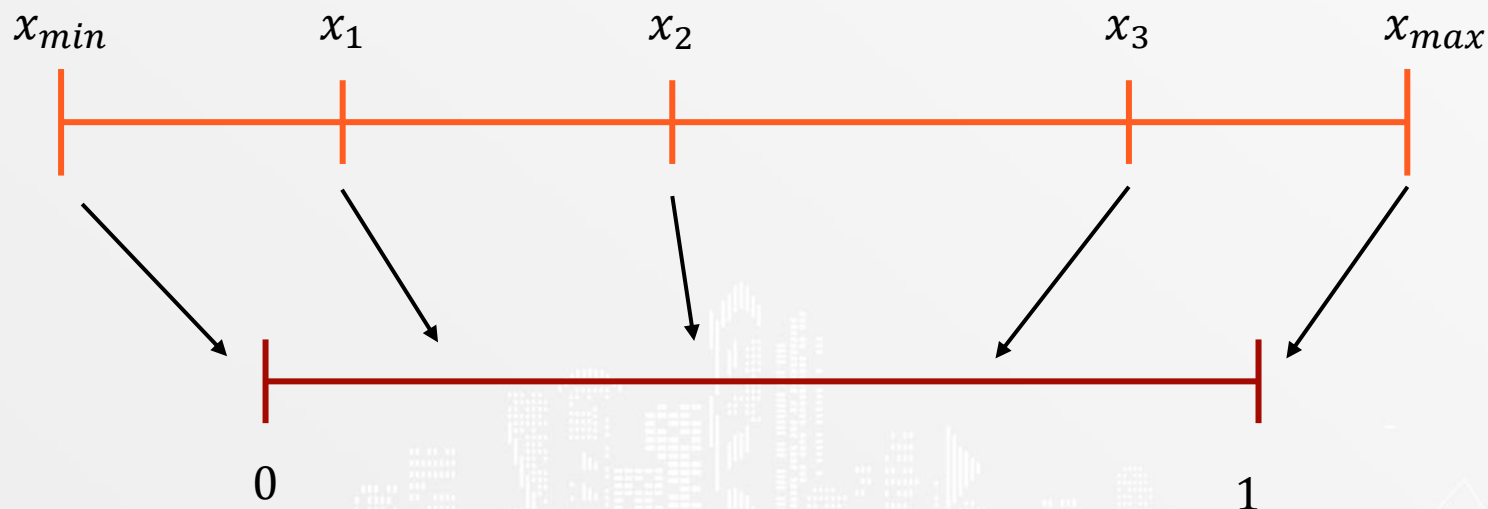


# Min-Max Scaling

- What is Min-Max Scaling?
- Formula
- Step to Calculate Min-Max Scaling
- Example
- Code

# What is Min-Max Scaling?

Min-Max Scaling คือ เทคนิคการปรับ scale ของข้อมูลให้อยู่ในช่วง 0 ถึง 1 โดยข้อมูลที่มีค่ามากที่สุดจะมีค่าใหม่เป็น 1 และข้อมูลที่มีค่าน้อยที่สุดจะมีค่าใหม่เป็น 0





# Formula

$$x' = \frac{x - \min X}{\max X - \min X}$$

- $X$  คือ feature ที่กำลังพิจารณา
- $x$  คือ ข้อมูลแต่ละตัวใน feature ที่กำลังพิจารณา
- $\min X$  คือ ค่าที่น้อยที่สุดของ feature ที่กำลังพิจารณา
- $\max X$  คือ ค่าที่มากที่สุดของ feature ที่กำลังพิจารณา

# Step to Calculate Min-Max Scaling

1. หาค่า min และ max ของแต่ละ feature
2. ปรับค่าข้อมูลแต่ละตัวใน feature ตามสูตรของ min-max scaling

# Example

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |

เลือก Feature ที่ต้องการจะ  
ทำ Feature Scaling



| Area |
|------|
| 99   |
| 110  |
| 117  |
| 93   |
| 92   |
| 99   |
| 96   |

ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง  
และพื้นที่ของบ้าน

# Example

**Area = [99.0, 110.0, 117.0, 93.0, 92.0, 99.0, 96.0]**

วิธีการปรับค่าด้วย Min-Max Scaling มีดังต่อไปนี้

1. หาค่า min และ max ของข้อมูล Area

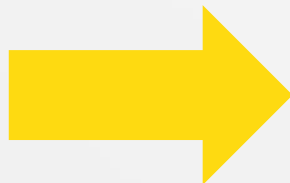
- $\max X = 117.0$
- $\min X = 92.0$

2. ปรับค่าข้อมูลแต่ละตัวใน Area ตามสูตรของ min-max scaling

$$x' = \frac{x - \min X}{\max X - \min X} = \frac{x - 92}{117 - 92} = \frac{x - 92}{25}$$

# Example

| Area  |
|-------|
| 99.0  |
| 110.0 |
| 117.0 |
| 93.0  |
| 92.0  |
| 99.0  |
| 96.0  |



| Area_scaled                                    |
|--|
| $= \frac{99 - 92}{25} = \frac{7}{25} = 0.28$   |
| $= \frac{110 - 92}{25} = \frac{18}{25} = 0.72$ |
| $= \frac{117 - 92}{25} = \frac{25}{25} = 1$    |
| $= \frac{93 - 92}{25} = \frac{1}{25} = 0.04$   |
| $= \frac{92 - 92}{25} = \frac{0}{25} = 0$      |
| $= \frac{99 - 92}{25} = \frac{7}{25} = 0.28$   |
| $= \frac{96 - 92}{25} = \frac{4}{25} = 0.16$   |

# Example

| Area  | Area_scaled |
|-------|-------------|
| 99.0  | 0.28        |
| 110.0 | 0.72        |
| 117.0 | 1           |
| 93.0  | 0.04        |
| 92.0  | 0           |
| 99.0  | 0.28        |
| 96.0  | 0.16        |



# Code

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |

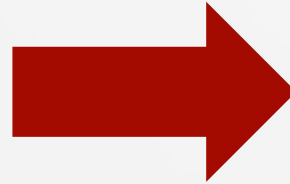
ตารางแสดงข้อมูลของบ้าน โดยมีจำนวนห้อง และพื้นที่ของบ้าน

# Code

```
1 from sklearn.preprocessing import MinMaxScaler
2
3 scaler = MinMaxScaler()
4 data_scaled = scaler.fit_transform(data)
5
6 data_scaled = pd.DataFrame(data_scaled, columns=feature_name)
```

# Code

|   | NumRooms | Area |
|---|----------|------|
| 0 | 4        | 99   |
| 1 | 4        | 110  |
| 2 | 4        | 117  |
| 3 | 3        | 93   |
| 4 | 5        | 92   |
| 5 | 3        | 99   |
| 6 | 6        | 96   |



|   | NumRooms | Area |
|---|----------|------|
| 0 | 0.333333 | 0.28 |
| 1 | 0.333333 | 0.72 |
| 2 | 0.333333 | 1.00 |
| 3 | 0.000000 | 0.04 |
| 4 | 0.666667 | 0.00 |
| 5 | 0.000000 | 0.28 |
| 6 | 1.000000 | 0.16 |

# Feature Encoding

**What is Feature  
Scaling?**



**Why need  
Feature Scaling?**



**Standardization**



**Min-Max Scaling**



**Conclusion**



# Conclusion

## Standardization

- mean =  $-5.31 \times 10^{-16}$
- s.d = 1.08
- ไม่มีขอบเขตของข้อมูล

## Area\_scaled

-0.22

1.07

1.88

-0.92

-1.03

-0.22

-0.57

## Area\_scaled

0.28

0.72

1

0.04

0

0.28

0.16

## Min-Max Scaling

- mean = 0.333
- s.d = 0.329
- ค่าอยู่ในช่วง [0,1]

# Conclusion

## Standardization

- ใช้ได้กับทุก distribution
- ไม่เปลี่ยน distribution ของข้อมูล
- เหมาะกับข้อมูลที่ไม่มีขอบเขต เช่น ข้อมูลส่วนสูง น้ำหนัก
- algorithm บางตัวควรต้องทำให้ dataset มี  $\text{mean}=0$ ,  $\text{s.d}=1$  เช่น SVM

## Min-Max Scaling

- ใช้ได้กับทุก distribution
- ไม่เปลี่ยน distribution ของข้อมูล
- เหมาะกับข้อมูลที่มีขอบเขต เช่น Indicator RSI
- algorithm บางตัวควรต้องปรับค่าให้อยู่ในช่วง 0-1 เช่น image processing



# Feature Encoding

**What is Feature Scaling?**



**Why need Feature Scaling?**



**Standardization**



**Min-Max Scaling**



**Conclusion**



# Data Preparation

