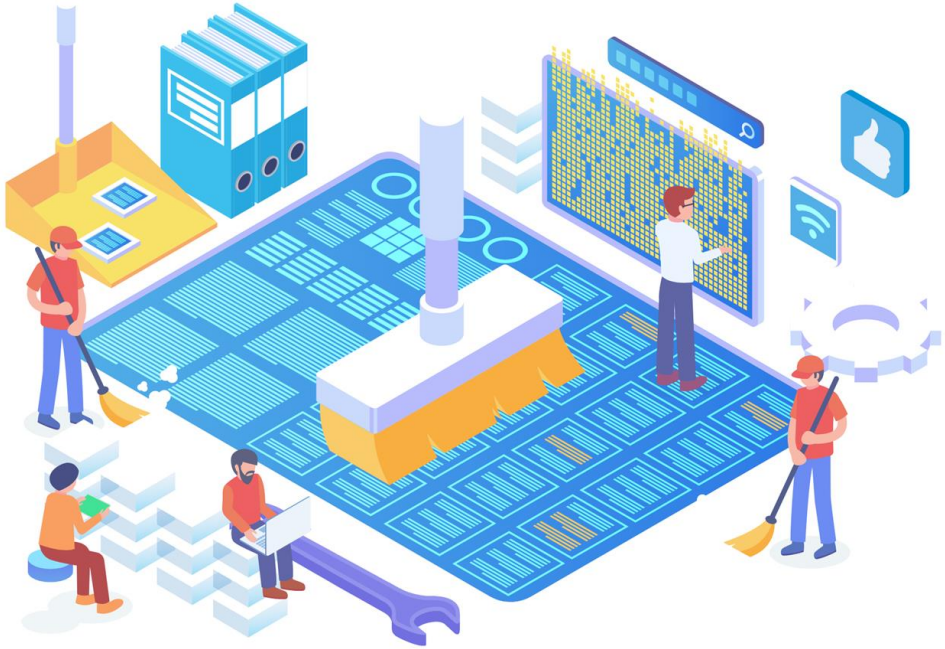


Chapter 2: Data Cleansing

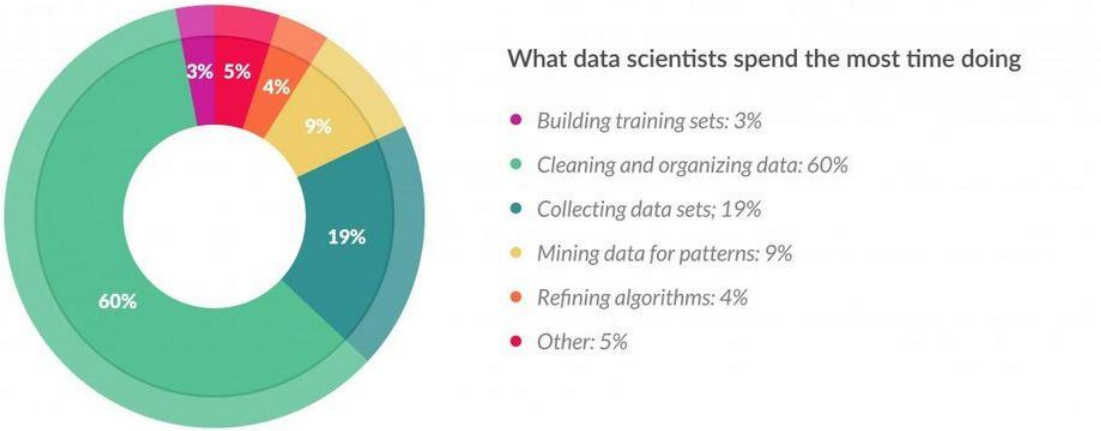
ทำความสะอาดข้อมูลให้พร้อมใช้งานด้วยเครื่องมือ
Distributed Processing: Apache Spark

DataTH

What is Data Cleansing & Data Quality

DataTH

Data Scientist ใช้เวลาส่วนใหญ่ไปกับการทำความสะอาดข้อมูล



ผลสำรวจจาก Forbes บอกว่า Data Scientist ใช้เวลากว่า 60% เพื่อเตรียมและทำความสะอาดข้อมูล

https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/

DataTH

Data Cleansing คืออะไร

Data Cleansing การทำความสะอาดข้อมูล

เป็นการพัฒนาคุณภาพของข้อมูล โดยการค้นหาและแก้ไขความผิดพลาดของข้อมูล เช่น

- ข้อมูลไม่ถูกต้องตามโครงสร้าง (Format Error) เช่น อายุติดลบ
- ข้อมูลสูญหาย (Missing Data)
- ข้อมูลสูงกว่าค่าปกติ (Outlier) เช่น อายุ 670 ปี

Tip: ควรทำ Data Cleansing คู่กับผู้เชี่ยวชาญข้อมูลด้านนั้นด้วย เช่น ผู้เชี่ยวชาญทางการแพทย์ จะบอกเราได้ว่าเลือดมีกี่ประเภท หรือค่าความดันแบบไหนที่ผิดปกติ

DataTH

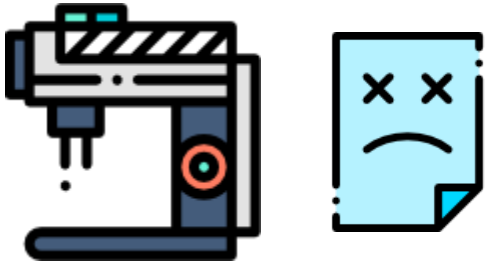
Data ทำไมถึงผิด

ถ้ายื่น Survey กระดาษ > คนพิมพ์เข้าระบบ

- คนกรอก ก็อาจจะกรอกผิดบ้าง หรือไม่อยากบอกข้อมูลบ้าง
- คนพิมพ์ใส่ระบบ ก็อาจจะพิมพ์ผิด

ถ้ามาจากเซนเซอร์เครื่องจักร > เข้าระบบอัตโนมัติ

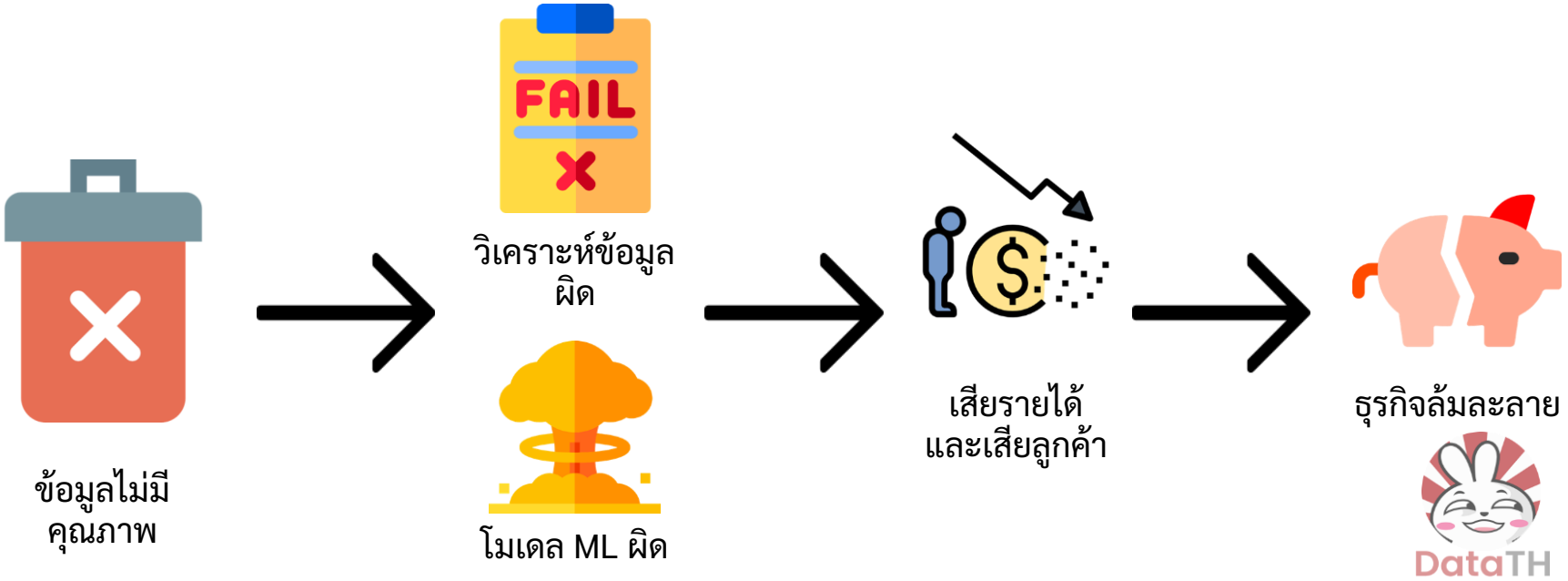
- เซ็นเซอร์อาจจะมีปัญหาบ้าง หรือวัดผลผิดบ้าง
- เกิดข้อผิดพลาดระหว่างการส่งข้อมูล



DataTH

ทำไมต้องทำความสะอาดข้อมูล

Garbage in, Garbage out



DataTH

Data ที่ไม่สะอาด มีราคาแพง

เศรษฐกิจอเมริกาเสียเงิน **$3.1 พันล้าน** จากข้อมูลที่ไม่สะอาด ในปี 2016

Case Study: NASA เสียเงิน $125 ล้าน จากข้อมูลไม่สะอาดในโปรเจค Mars Climate Orbiter

“The peer review preliminary findings indicate that **one team used English units (e.g., inches, feet and pounds)** while the **other used metric units (e.g. centimeter, kilometer, and kilogram)** for a key spacecraft operation.”

https://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data
https://mars.jpl.nasa.gov/msp98/news/mco990930.html

DataTH

ทำไม Data Cleaning ถึงยาก

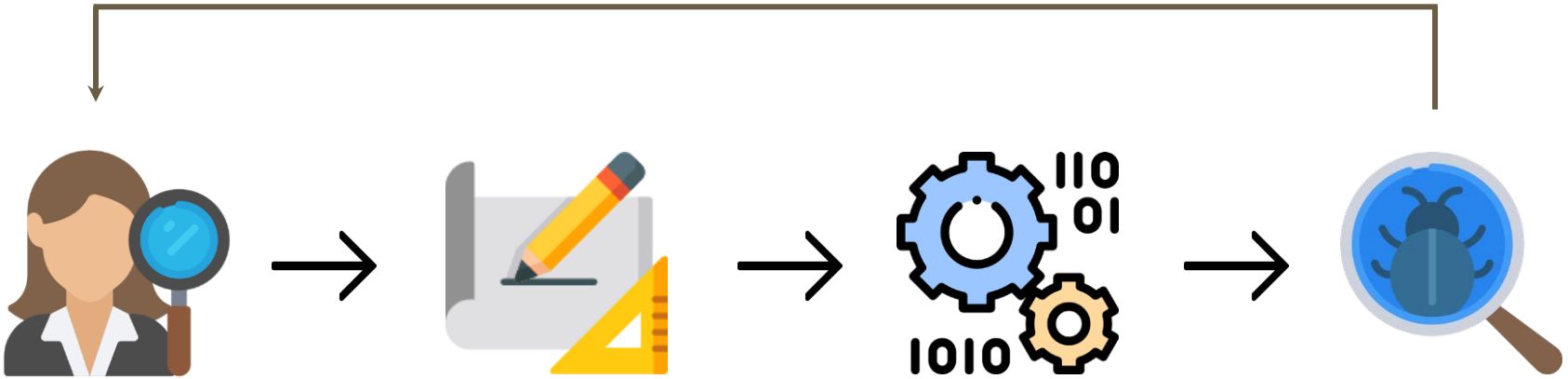
กระบวนการที่
ไม่มีวันจบสิ้น

ยากที่จะรู้ว่าเกิด
จากอะไร

แต่ละ Source มักมี
ข้อมูลโครงสร้าง
แตกต่างกัน

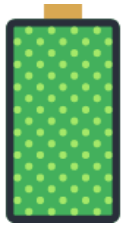
DataTH

Data cleansing เป็นกระบวนการที่ไม่มีวันจบสิ้น



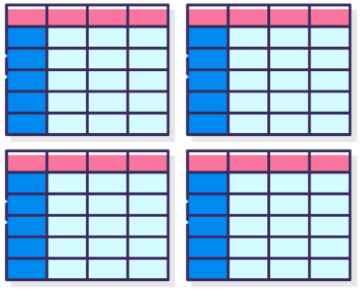
DataTH

Data Quality – คุณภาพข้อมูลที่ดี คืออะไร



Completeness
ข้อมูลครบ ไม่มี
Missing Values

Validity
ข้อมูลไม่ผิดข้อจำกัด
เช่น อายุไม่ควรติดลบ

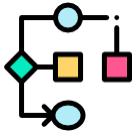


Consistency
ข้อมูลจากหลาย Data Source
ควรจะใช้โครงสร้างคล้ายกัน

DataTH

3 เครื่องมือสำหรับ Data Quality

Data Dictionary



Data Lineage

Data Catalogue

DataTH

Data Dictionary

ไฟล์ที่รวบรวมรายละเอียดของทุกคอลัมน์ในตารางข้อมูล เพื่อให้ทุกคนในองค์กรเข้าใจตรงกัน และสำคัญมากในการทำความสะอาดข้อมูล

คอลัมน์สำคัญใน Data Dictionary:

- ชื่อคอลัมน์
- ประเภทข้อมูลในคอลัมน์
- ตัวอย่างข้อมูลในคอลัมน์
- คำอธิบายของคอลัมน์

employees

Column	Data type	Nullable	Description
emp_no	integer	not null	Unique employee number
first_name	varchar(100)	not null	Employee first name
last_name	varchar(100)	not null	Employee last name
dob	date	null	Date of birth if known
card_no	char(6)	null	Employee access control card number
edu	integer	not null	Education level 9 - unknown 4 - elementary 3 - Middle school 2 - High shool 1 - University
dept_id	integer	not null	Employee department ID. Ref: departments
eval	date	null	Date of last performance review
current	bit	not null	1 - current employee, 0 - past employee

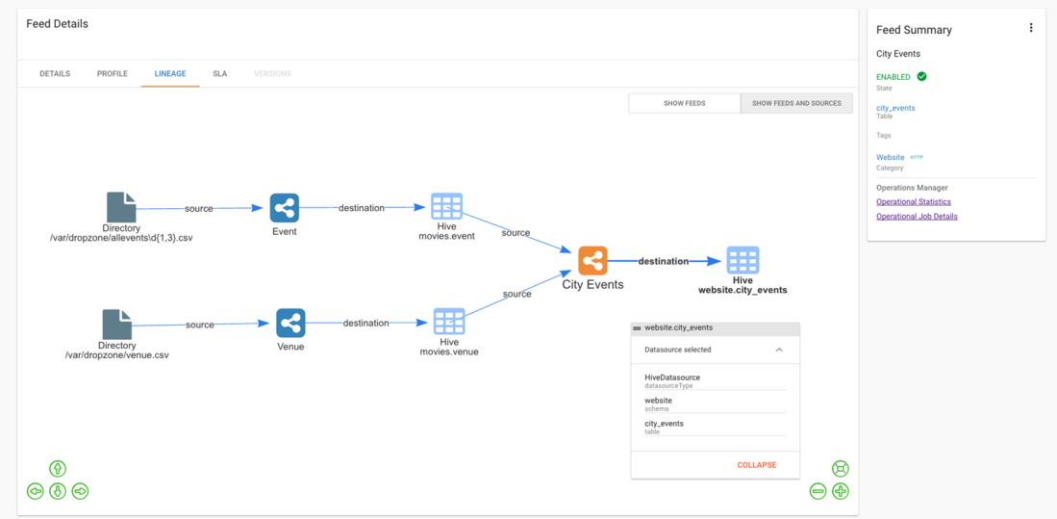
DataTH

https://dataedo.com/blog/why-is-metadata-important-for-databases

Data Lineage

การเดินทางของข้อมูลตั้งแต่ต้นจนจบ

มีประโยชน์มากในการตรวจสอบ Error ของข้อมูล ว่ามาจากจุดไหนในระบบ



Apache Atlas

kylo

https://kylo.readthedocs.io/en/v0.9.0/how-to-guides/FeedLineage.html

DataTH

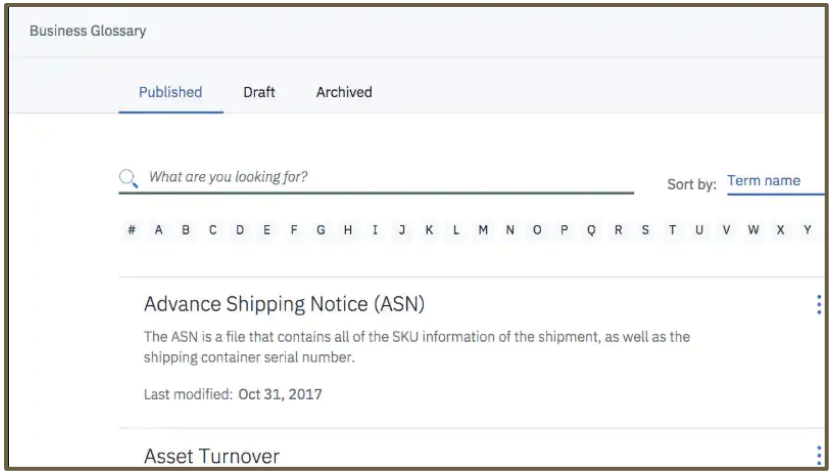
Data Catalog

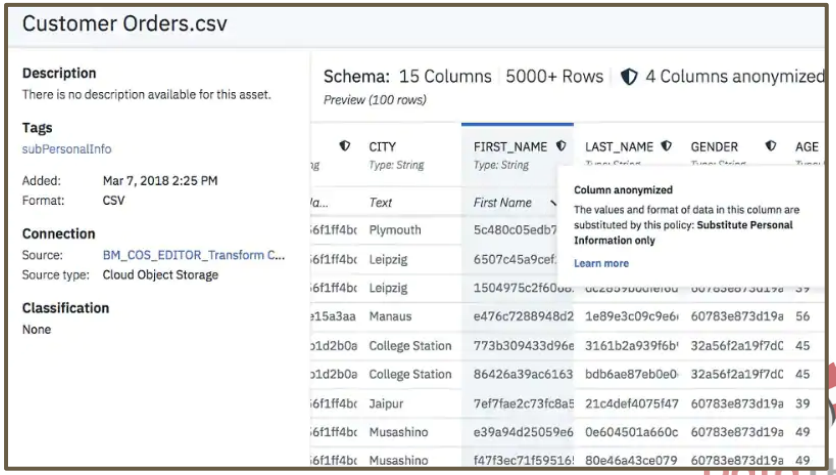
แหล่งรวมข้อมูล และรายละเอียดเกี่ยวกับข้อมูล (metadata) ทั้งหมดให้ทั้งฝ่าย Business และที่ใช้งาน Data นำข้อมูลไปใช้ต่อได้ง่าย

ตัวอย่างข้อมูลใน Data Catalog

- ระบบค้นหาข้อมูล
- Data Dictionary
- ตัวอย่างข้อมูล ที่ซ่อนข้อมูลสำคัญให้เห็นได้เฉพาะพนักงานบางกลุ่ม (Data Masking)

ตัวอย่าง: IBM Watson Knowledge Catalog, Azure Data Catalog, Talend Data Catalog





Exploratory Data Analysis (EDA)

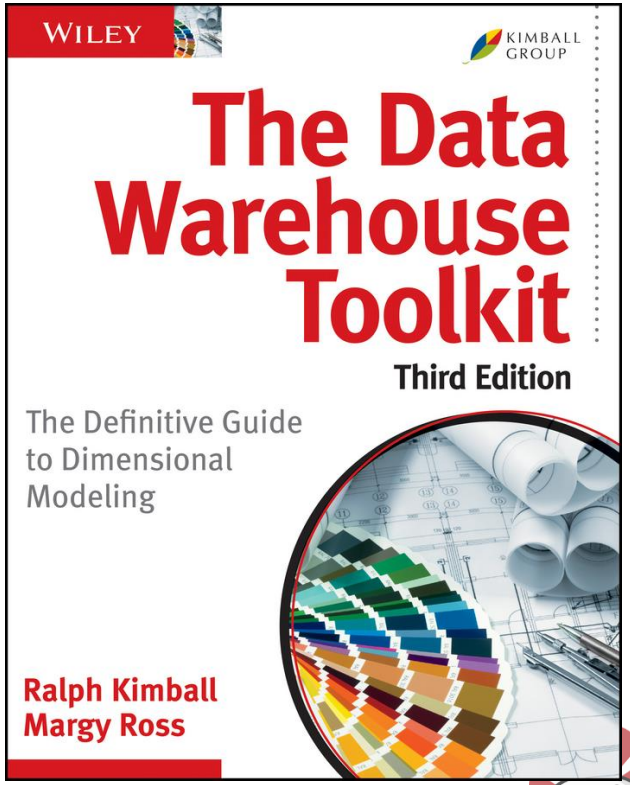
DataTH

Data Profiling

“Data Profiling ใช้การดึงข้อมูลจากแหล่งข้อมูล เพื่อดูเนื้อหาและความสัมพันธ์ แทนที่จะดูจากข้อมูลที่อาจจะไม่อัพเดทจากเอกสารการใช้งาน”

- Ralph Kimball, Margy Ross

วิธีการดูภาพรวมของข้อมูล เพื่อหาว่ามีปัญหาอะไรกับข้อมูลมั้ย



DataTH

ตัวอย่างของการทำ Data Profiling

ใช้ข้อมูล Hotel Booking Demand บน Kaggle

https://www.kaggle.com/jessemostipak/hotel-booking-demand

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000
mean	0.370416	104.011416	2016.156554	27.165173	15.798241
std	0.482918	106.863097	0.707476	13.605138	8.780829
min	0.000000	0.000000	2015.000000	1.000000	1.000000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000
75%	1.000000	160.000000	2017.000000	38.000000	23.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000

DataTH

ประเภทของ EDA

(Exploratory Data Analysis)

วิธีการเจาะลึกเพื่อดูรายละเอียดของข้อมูล และความสัมพันธ์ของแต่ละคอลัมน์ / แถว

ใช้ตัวเลขหรือกราฟฟิก

1. **แบบใช้ตัวเลข** - ใช้ตัวเลขทางสถิติ เช่น หาค่า Min, Max, Mean
2. **แบบใช้กราฟฟิก** - ใช้การพลอตกราฟ (Data Visualisation) เช่น Boxplot, Histogram

ดูรายละเอียดตัวแปรกี่ตัว

1. **ดูทีละตัว (Univariate)** - เช่น ดูค่า Mean ของคอลัมน์เดียว
2. **ดูทีละหลายตัว (Multivariate)** - เช่น การคำนวณ Covariance หรือ การวาด Scatterplot เปรียบเทียบระหว่าง 2 ตัวแปร

DataTH

EDA แบบใช้ตัวเลข

(Exploratory Data Analysis)

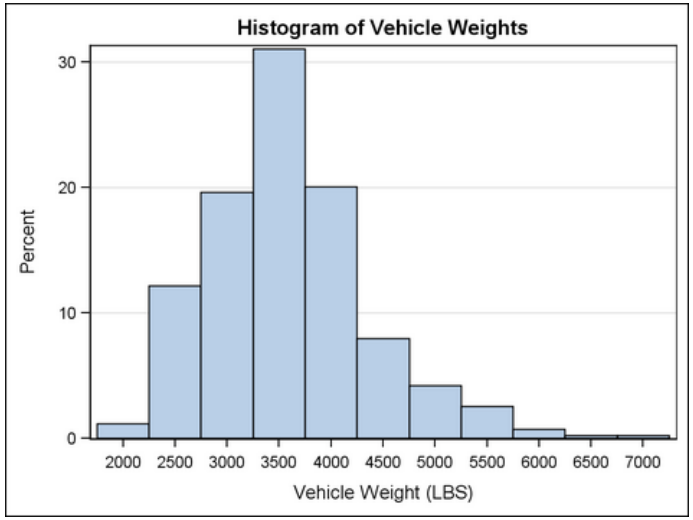
สถิติเชิงพรรณา (Descriptive Statistics)

- **การกระจายตัวของข้อมูล (Distribution):**
  - จำนวนข้อมูล (count)
  - ค่าเฉลี่ย (mean)
  - ข้อมูล 25%, 50%, 75%
  - ค่าสูงสุด - ต่ำสุด
- **ความแปรปรวนของข้อมูล (Variance):**
  - ค่าความเบี่ยงเบนมาตรฐาน (Standard Deviation)

	is_canceled	lead_time
count	119390.000000	119390.000000
mean	0.370416	104.011416
std	0.482918	106.863097
min	0.000000	0.000000
25%	0.000000	18.000000
50%	0.000000	69.000000
75%	1.000000	160.000000
max	1.000000	737.000000

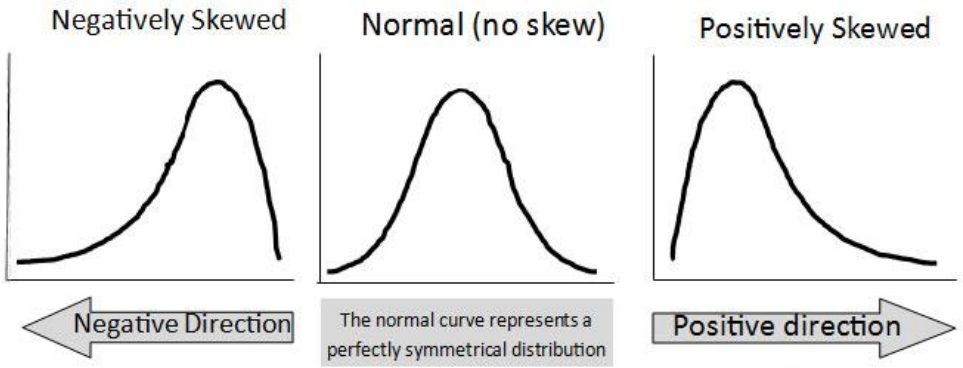
https://support.sas.com/documentation/cdl/en/grstatgraph/65377/HTML/default/p1sxw5gidyzrygn1ibkzfmc5c93m.htm
https://www.managedfuturesinvesting.com/what-is-skewness/

EDA แบบใช้กราฟฟิก



1 LBS = 0.45 kg

Skewness - ความเบี่ยงเบนของข้อมูล



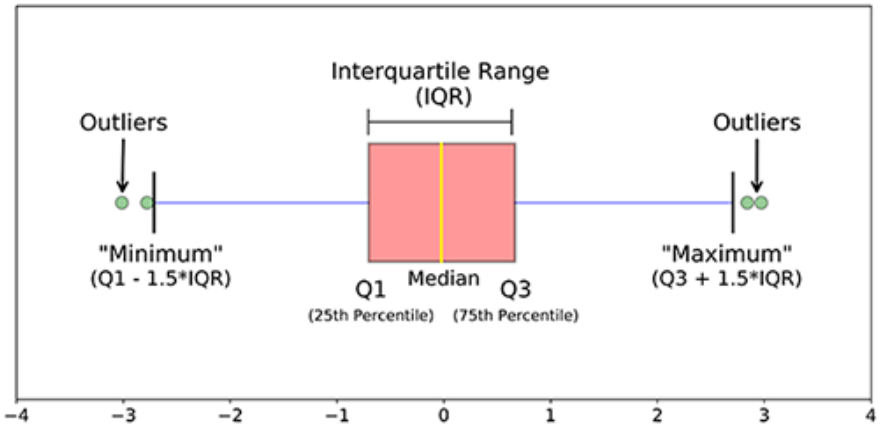
Histogram

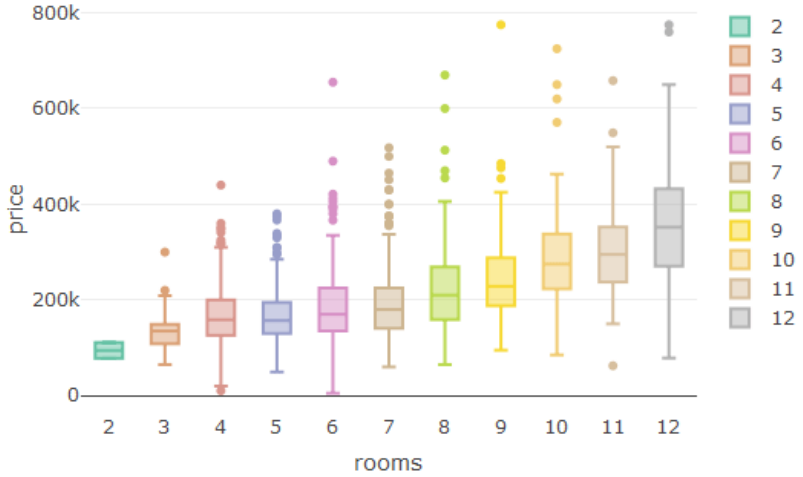
แสดงการกระจายตัวของข้อมูล (เป็นลักษณะยอดเขาเดียว หรือหลายยอด หรือไม่มียอดเลย) และความเบี่ยงเบนของข้อมูล (Skewness)

DataTH

https://medium.com/@agarwal.vishal819/outlier-detection-with-boxplots-1b6757fafa21
https://www.edureka.co/community/8659/how-to-create-a-box-plot-using-plotly-in-r

EDA แบบใช้กราฟฟิก (2)





Boxplot

แสดงการกระจายตัวของข้อมูล และค่าที่เกินจากปกติมาก (Outliers)

DataTH

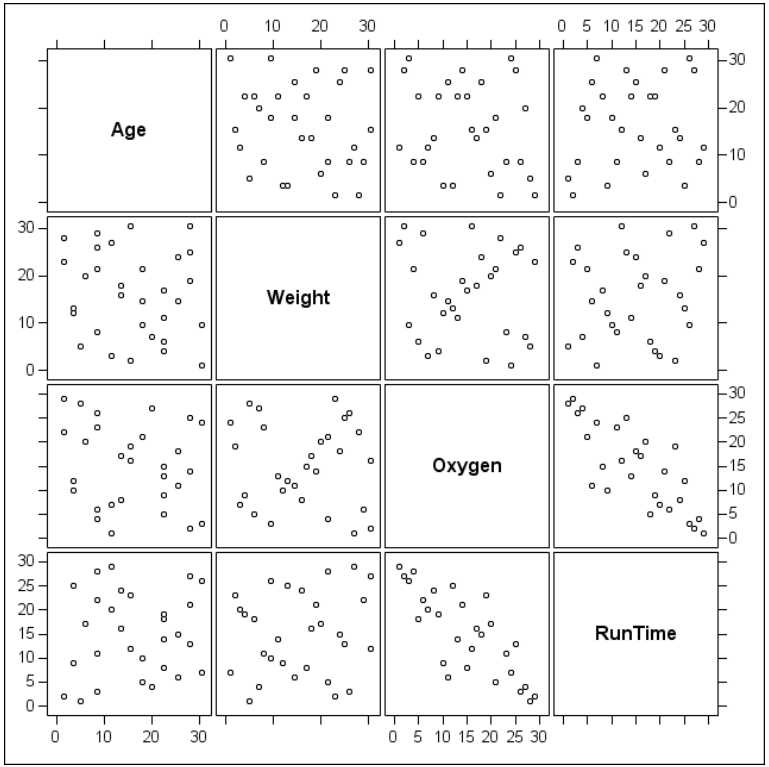
EDA แบบใช้กราฟฟิก (3)

Scatterplot

ใช้หาความสัมพันธ์ระหว่าง 2 ตัวแปร

Scatterplot Matrix

ใช้หาความสัมพันธ์ระหว่างทุกคู่ตัวแปร ->



https://support.sas.com/kb/37/000.html

DataTH

Data Anomaly

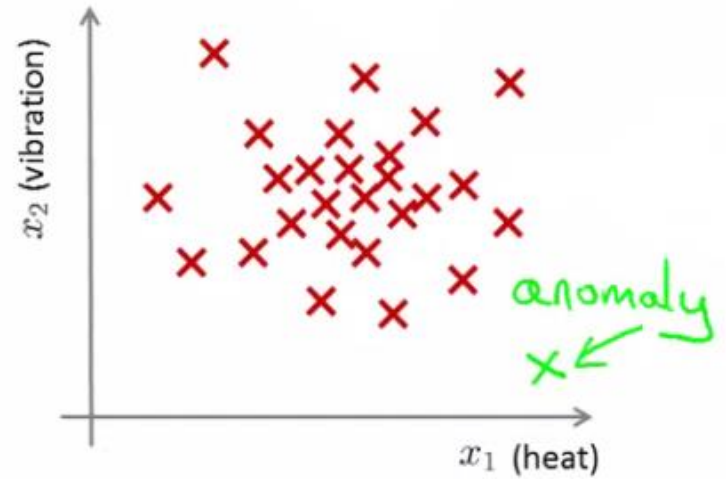
DataTH

What is Data Anomaly

ความผิดปกติของข้อมูลที่เกิดขึ้นได้จากตอนเก็บข้อมูล ซึ่งทำให้ข้อมูลไม่สมบูรณ์

ตัวอย่าง Data Anomaly:

- พิมพ์ผิด (Lexical Error)
- ข้อมูลซ้ำ (Duplication)
- ข้อมูลไม่สม่ำเสมอ (Inconsistency)
- ข้อมูลหายบางส่วน (Missing Values)
- ข้อมูลเกินค่าปกติ (Outliers)
- ฯลฯ



DataTH

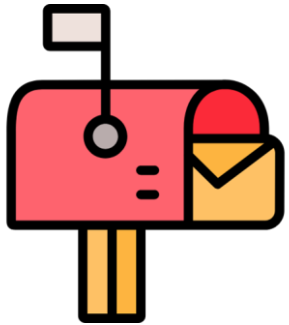
http://upscfever.com/upsc-fever/en/data/en-data-chp59.html

ประเภทของ Data Anomaly

1. **Syntactical Anomalies** - เกิดจากข้อผิดพลาดในการกรอกข้อมูล เช่น พิมพ์ผิด (spelling mistake), ประเภทข้อมูลผิด (domain format error), มีตัวอักษรผิดปกติ (syntactical error), มีค่าแปลก (irregularity)
2. **Semantic Anomalies**: เกิดจากข้อผิดพลาดในการเก็บข้อมูล เช่น เก็บข้อมูลซ้ำ (duplication), เก็บข้อมูลไม่ตรงตามเงื่อนไข (integrity constraint violation)
3. **Coverage Anomalies**: เกิดจากข้อผิดพลาดในความสมบูรณ์ของข้อมูล เช่น ค่าหาย (missing value)
4. **Outliers:** ข้อมูลที่สูงหรือต่ำกว่าปกติ

DataTH

1. Syntactical Anomalies คืออะไร และวิธีแก้ไข



Syntactical Anomalies เช่น พิมพ์ผิด (Spelling Mistake) หรือค่าแปลก (Irregularities) เช่น รหัสไปรษณีย์ 6 หลัก, ชื่อจังหวัดที่ไม่มีจริง

วิธีแก้ไข:

1. <u>แก้ไขที่การเก็บข้อมูล:</u> เช่น ถ้าเป็นฟอร์มออนไลน์ สามารถเขียนโค้ดตรวจสอบข้อมูล (Data Validation) หรือทำเป็นช้อยส์ให้เลือก
2. <u>แก้หลังจากเก็บข้อมูลมาแล้ว:</u> เปรียบเทียบกับแหล่งอ้างอิงข้อมูลที่น่าเชื่อถือ เช่น ตรวจสอบที่อยู่จาก Google Map API

DataTH

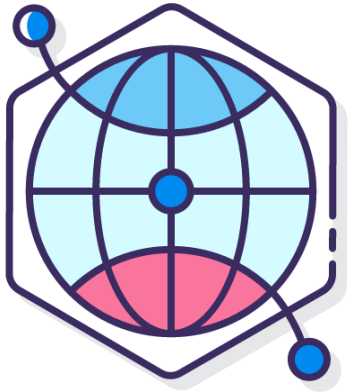
2. Semantic Anomalies คืออะไร และวิธีแก้ไข

ประเภทของ Semantic Anomalies

1. **ค่าไม่ตรงตามเงื่อนไข (Integrity Constraints)**
   เช่น อายุติดลบ หาค่าที่ผิด และแก้ไขให้ถูก

2. **ค่าซ้ำ (Duplication)**
   เช็คก่อนว่าคอลัมน์นั้นซ้ำได้มั้ย แล้วลบหรือแก้ไขตามความเหมาะสม

3. **ค่าขัดกัน (Contradictions)**
   เช่น วันจบมาก่อนวันเริ่ม แก้ไขเหมือนค่าซ้ำ



DataTH

วิธีค้นหา Syntactical และ Semantics Anomalies

Regular Expression (regex) = pattern สำหรับค้นหาตัวหนังสือ

ทำไม regex ถึงมีประโยชน์

- ฟีเจอร์เยอะ และยืดหยุ่น
- เขียนได้สั้น ๆ
- ง่ายสำหรับการค้นหาและแทนที่
- เรียนครั้งเดียว นำไปใช้ได้ในหลายภาษาโปรแกรมมิ่ง

ตัวอย่าง

[A-Z] = ค้นหา 1 ตัวอักษร จาก A ถึง Z
[A-Z]+ = ค้นหา 1+ ตัวอักษร จาก A ถึง Z
[a-zA-Z0-9]* = ค้นหา 0+ ตัวอักษร จาก a ถึง z, A ถึง Z, และ 0 ถึง 9

Tip: ใช้ https://regex101.com/ สำหรับเขียน Regular Expression

DataTH

3. Coverage Anomalies หรือ Missing Values

ข้อมูลที่หายไปจากระบบ สามารถเกิดโดยบังเอิญ (MAR - Missing At Random)
หรือไม่บังเอิญ (MNAR - Missing Not At Random)

สำหรับ Data Analyst, Data Scientist

- มองหาข้อมูลหาย
- ตรวจเช็คว่าข้อมูลหายเพราะอะไร
- หาวิธีแก้ไขปัญหาข้อมูลหายที่เหมาะสมที่สุด เช่น ตัดทิ้งจากการวิเคราะห์ หรือแทนค่าที่หาย (impute) ด้วยค่าเฉลี่ย, Regression Model

สำหรับ Data Engineer

- มองหาข้อมูลหาย
- ตรวจสอบข้อมูลดิบ (Raw Data) ที่มาจากแหล่งข้อมูล
- แจ้งปัญหากับทีมที่ดูแลแหล่งข้อมูล

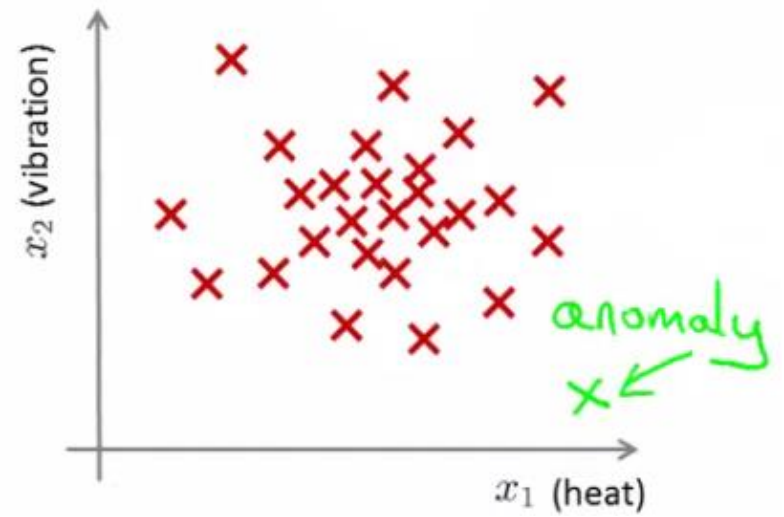
Tip: ปัญหาถ้าแก้ได้ที่ต้นเหตุ (แหล่งข้อมูล) จะดีกับงานมากกว่าในระยะยาว

DataTH

Outliers คืออะไร

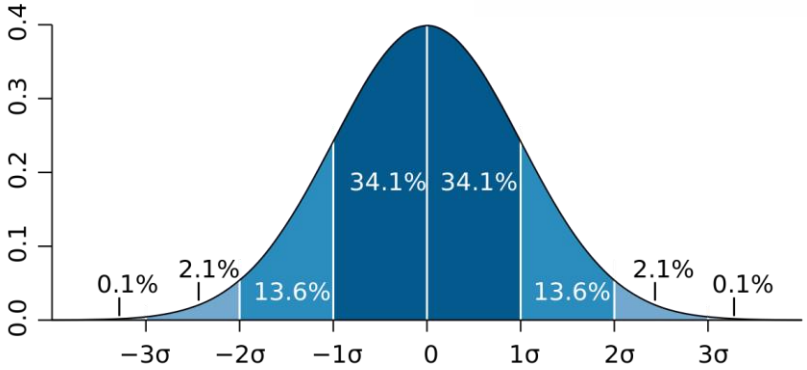
Outliers คือ ค่าที่ห่างจากค่าส่วนใหญ่มาก ซึ่งแปลว่าค่าอาจจะผิดปกติ

มีประโยชน์มากสำหรับการตรวจสอบการโกง ในบัตรเครดิต (Fraud Detection)



วิธีการเช็ค Outliers

1. Boxplot
2. กฎ 3 sigma



https://en.wikipedia.org/wiki/Standard_deviation

DataTH

Distributed Data Processing
(การประมวลผลข้อมูล แบบกระจายศูนย์)

DataTH

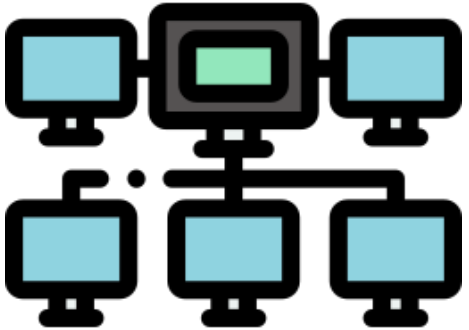
Distributed Data Processing คืออะไร

Standalone (1 Node)

ประมวลผลด้วยคอมพิวเตอร์ 1 เครื่อง

ทำงานได้แค่ทีละอย่าง

Cluster (2+ Nodes)



ประมวลผลด้วยคอมพิวเตอร์หลายเครื่อง

ทำได้พร้อมกันหลายอย่าง หรือทำอย่างเดียวก็เร็วขึ้น เพราะกระจายงานกันทำได้

DataTH

ตัวอย่าง Distributed Data Processing :
Word Count การนับจำนวนของแต่ละคำ จากหนังสือ 1 เล่ม

Standalone (1 คน)

Apple: 10 คำ
Bee: 5 คำ
Cat: 9 คำ
Dog: 13 คำ
. . .

ใช้ผู้เชี่ยวชาญ 1 คน นับทั้งเล่ม

อ่านได้แค่ทีละหน้า

Cluster (2+ คน)

← node
(1 คน)

ใช้เด็กหลายคน นับคนละบท

อ่านได้พร้อมกันหลายหน้า

DataTH

Hadoop MapReduce

ส่วนหนึ่งของ Hadoop: Hadoop MapReduce เป็นวิธีหนึ่งในการทำ Distributed Data Processing สำหรับข้อมูลขนาดใหญ่ ที่ได้รับความนิยมมากในช่วงก่อนหน้านี้

สร้างขึ้นโดย Yahoo! ในปี 2006 จากงานวิจัยของ Google ที่ตีพิมพ์ในปี 2003

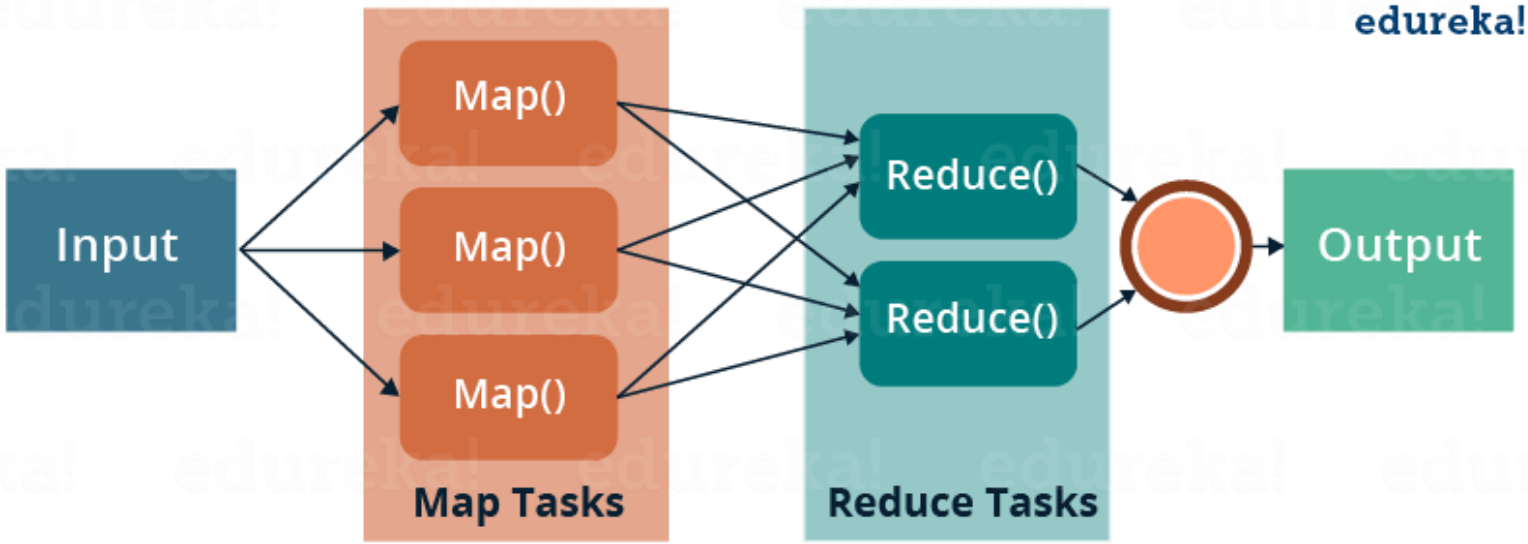
ใช้คอมพิวเตอร์ทั่วไป (Commodity Hardware) มาทำงานร่วมกัน

เขียนยาก ทำให้ความนิยมตกลงไปในปัจจุบัน

ทำงานช้า เพราะต้องเขียนไฟล์เก็บข้อมูลทุกขั้นตอน

DataTH

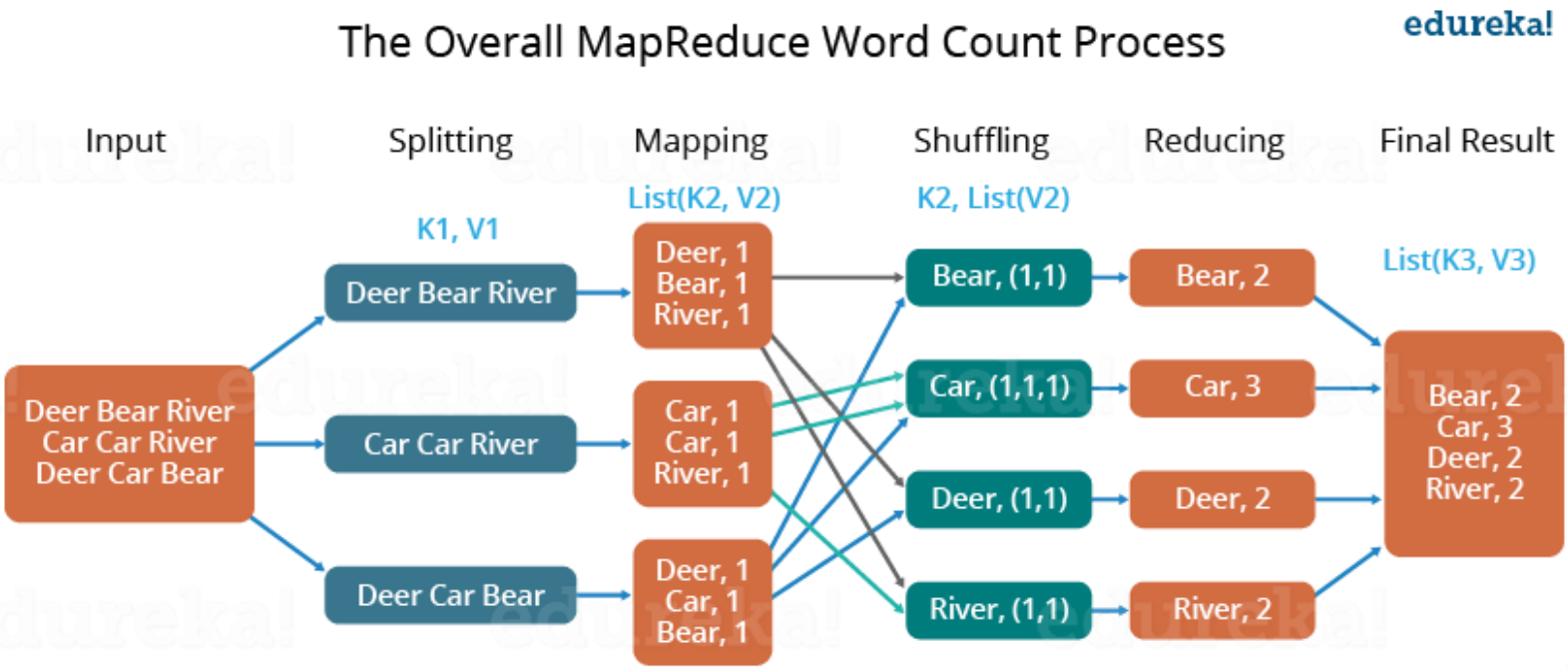
การทำงานของ Hadoop MapReduce



https://www.edureka.co/blog/mapreduce-tutorial/

DataTH

ตัวอย่าง Hadoop MapReduce



https://www.edureka.co/blog/mapreduce-tutorial/

DataTH

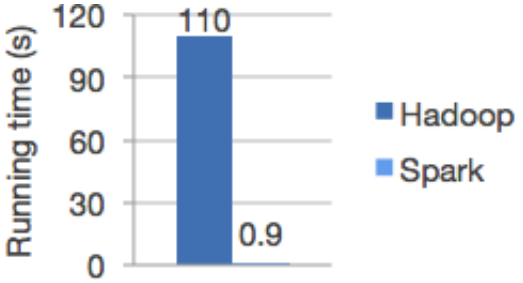
มารู้จักกับ Apache Spark

DataTH

Apache Spark

เทคโนโลยีในการทำ Distributed Data Processing สำหรับข้อมูลขนาดใหญ่ ที่คิดค้นโดย University of California ในปี 2014

- ✓ เก็บข้อมูลบนหน่วยความจำ (Memory) แทนการเขียนไฟล์เก็บข้อมูลทุกขั้นตอน
- ✓ เก็บการเปลี่ยนแปลงของข้อมูล แทนการเก็บข้อมูลที่เปลี่ยนแล้ว
- ✓ ทนต่อการล่ม (Fault-tolerant) ด้วย RDD (Resilient Distributed Dataset)
- ✓ สามารถใช้งานได้จากหลายภาษา เช่น Python, R, Java, Scala
- ✓ มี Component ให้ใช้งานเยอะ เช่น MLlib, Spark SQL, GraphX



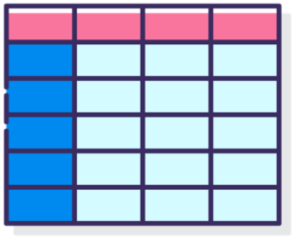
เปรียบเทียบความเร็วในการรัน logistic regression บน Hadoop MapReduce และ Spark

พบว่า Spark เร็วกว่าถึง 100 เท่า

https://spark.apache.org/

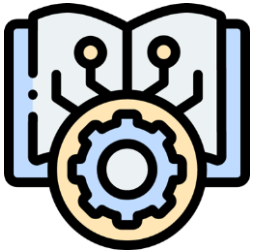
DataTH

ส่วนเสริม (Module) ที่มากับ Spark



Spark SQL
สามารถเขียน SQL เพื่อทำงานกับข้อมูลบน Spark ได้

Spark Streaming
เขียนโปรแกรมรองรับข้อมูลที่เข้ามาอย่างรวดเร็ว



MLlib
สร้างโมเดล Machine Learning สำหรับข้อมูลขนาดใหญ่

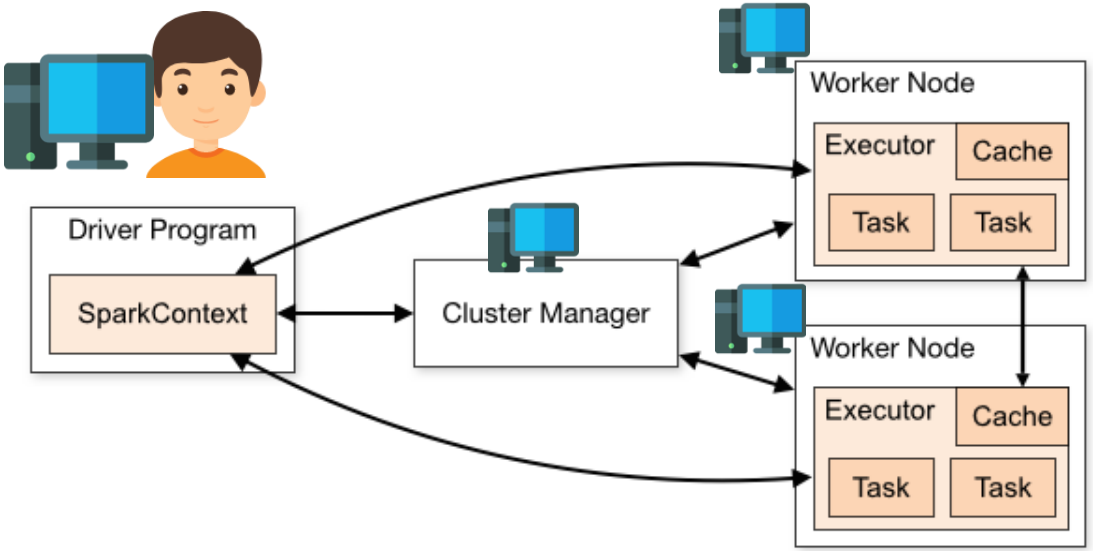
GraphX
การประมวลผลข้อมูลในรูปแบบของกราฟ

รันบน APACHE Spark™

3 วิธี ในการรันคำสั่ง Spark

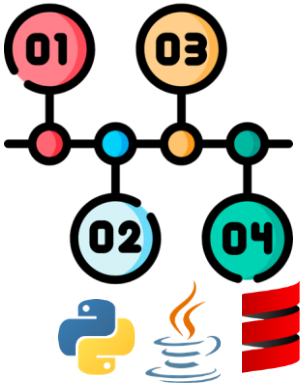
1. **สั่งงาน Spark ผ่านการเขียนโปรแกรม** เช่น PySpark, SparkR
2. **สั่งงาน Spark ทีละคำสั่ง** ด้วย Spark Shell
3. **สั่งงาน Spark ด้วยการส่งงาน (Job) ไปรันบน cluster** ด้วย Spark Submit

ตัวอย่างการใช้ Spark Submit



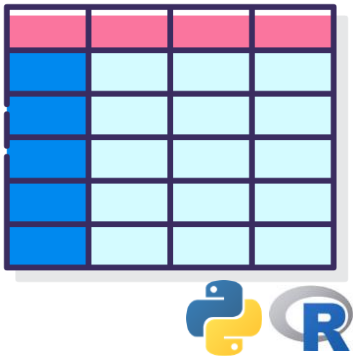
https://spark.apache.org/docs/latest/cluster-overview.html

ประเภทข้อมูลใน Spark



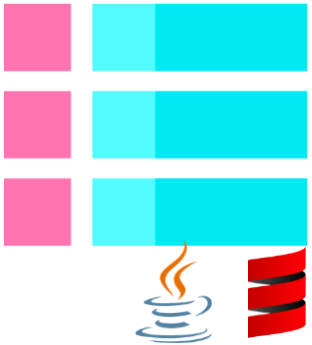
Resilient Distributed Datasets (RDD) (Java / Scala / Python)

วิธีทำงานกับข้อมูลแบบพื้นฐาน มีคำสั่งซับซ้อน



Spark DataFrame & Spark SQL (Python / R)

วิธีทำงานกับข้อมูลแบบตาราง Relational Database



DataSet (Java / Scala)

ข้อมูลแบบ DataFrame ที่มีการเช็คโค้ดตอน Compile และบังคับกำหนด Type ข้อมูล

RDD (Resilient Distributed Datasets)

วิธีการเก็บข้อมูลแบบพื้นฐานของ Spark

สามารถรันคำสั่งบน RDD ได้ โดยแบ่งเป็นคำสั่ง 2 แบบ

Transformation

- ไม่ทำงานทันที (Lazy) เก็บเป็นประวัติไว้ก่อน
- ตอนที่ทำงาน จะสร้าง RDD ใหม่ทุกครั้ง (ข้อมูล RDD ไม่สามารถเขียนทับได้ - Immutable)
- ตัวอย่างคำสั่ง: map(), filter()

Action

- ทำงานทันที และบังคับ Transformation ก่อนหน้านี้ให้ทำงานด้วย
- ผลลัพธ์ไม่ได้เป็น RDD เช่น เป็นตัวเลข หรือ อาจจะไม่มีผลลัพธ์
- ตัวอย่างคำสั่ง: count(), collect(), take(5)

RDD: Transformation vs Action

Transformation:
5.33 ms

ยังไม่ทำงานทันที

```
%%timeit

from pyspark.sql.functions import when

dt.withColumn("CountryNoEIRE", when(dt['Country'] == 'EIRE', 'Ireland').otherwise(dt['Country']))
```

```
100 loops, best of 3: 5.33 ms per loop
```

Action:
63.1 ms

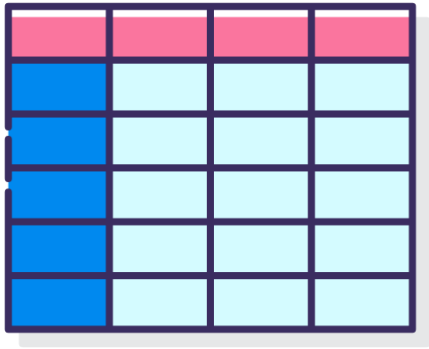
ทำงานทันที และสั่งให้ Transformation ก่อนหน้านี้ทำงานด้วย

```
%%timeit

from pyspark.sql.functions import when

dt.withColumn("CountryNoEIRE", when(dt['Country'] == 'EIRE', 'Ireland').otherwise(dt['Country'])).take(1)
```

```
10 loops, best of 3: 63.1 ms per loop
```

DataTH

Spark DataFrame



แปลงข้อมูลเป็นตาราง ทำให้ทำงานกับข้อมูลได้ง่ายขึ้น
คล้าย DataFrame ใน Pandas และ R

- คำสั่ง **select** สำหรับเลือกข้อมูลด้วยเงื่อนไขต่าง ๆ
- คำสั่ง **withColumn** สำหรับเพิ่มคอลัมน์ใหม่
- คำสั่ง **na.fill** สำหรับเติมข้อมูล Missing Values
- แปลงเป็น Pandas DataFrame เพื่อทำ Data Visualisation ได้
- พร้อมใช้งานกับ SQL ด้วย Spark SQL

na ย่อมาจาก Not Applicable ซึ่งหมายถึง null value หรือข้อมูลที่หายไป

Tip: บางอย่างทำใน DataFrame ง่ายกว่า แต่เราอยากได้ RDD เราก็สามารถแปลง RDD → DataFrame → RDD ได้

DataTH

Spark SQL

ใช้ **SQL** ในการดึงข้อมูลจาก Spark DataFrame

มีประโยชน์มากสำหรับการเขียน ETL ด้วย Spark

แปลง Spark DataFrame เป็น **TempView** หรือ **GlobalTempView**

↓

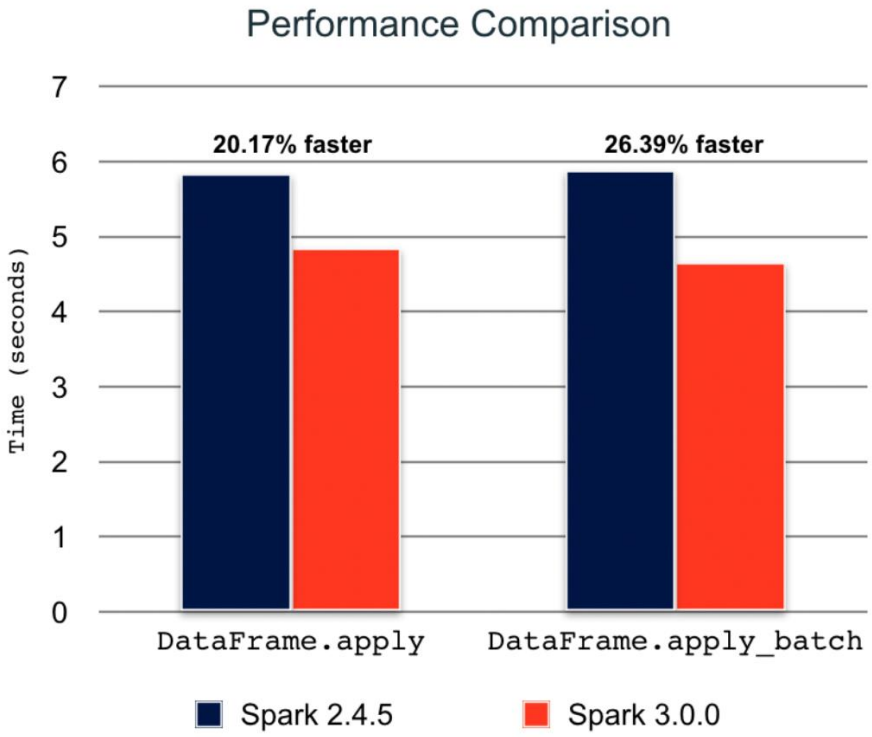
ใช้คำสั่ง SQL ในการดึงหรือแปลงข้อมูลบน TempView

DataTH

เกร็ดเสริม: Spark 3

Spark 3 ออกมาช่วง June 2020

1. เร็วกว่าเดิม
   - **Adaptive query execution (AQE)** ปรับการอ่านข้อมูลในระหว่างที่รันงานให้ใช้เวลาน้อยที่สุด
   - **Dynamic partition pruning** ปรับปรุงการ join ให้เร็วขึ้นโดยการใช้ partition
   - **GPU Aware** ใช้งานร่วมกับการ์ดจอ โดยอัตโนมัติ
2. รองรับ SQL ตามมาตรฐาน ANSI
   (American National Standards Institute)
3. PySpark <3 Pandas



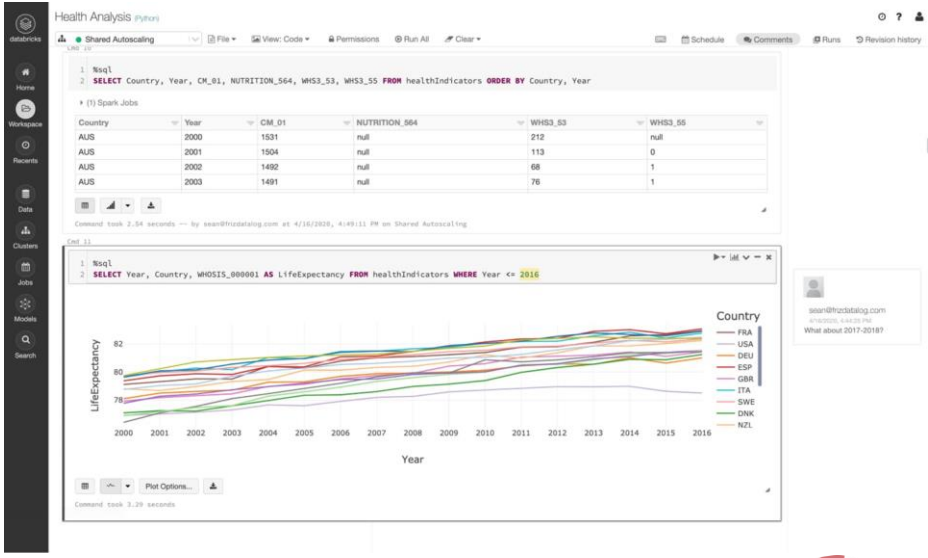
https://databricks.com/blog/2020/06/24/introducing-koalas-1-0.html

เกร็ดเสริม: Databricks

databricks

Data Platform ออนไลน์ โดยทีมพัฒนา Spark

- มี UI แบบ Notebook ให้ใช้งานง่าย รองรับ Python, R, SQL
- รองรับการรัน Spark job ด้วย Spark-submit หรือ Pyspark
- สามารถใช้ได้บน AWS, GCP, และ Microsoft Azure



https://databricks.com

DataTH

เกร็ดเสริม: Spark Cheatsheets

By DataCamp

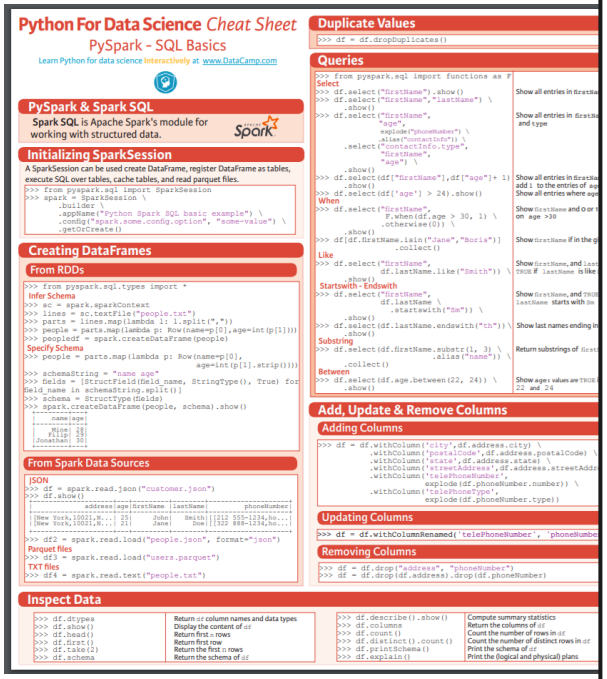
(ภาษาอังกฤษ)

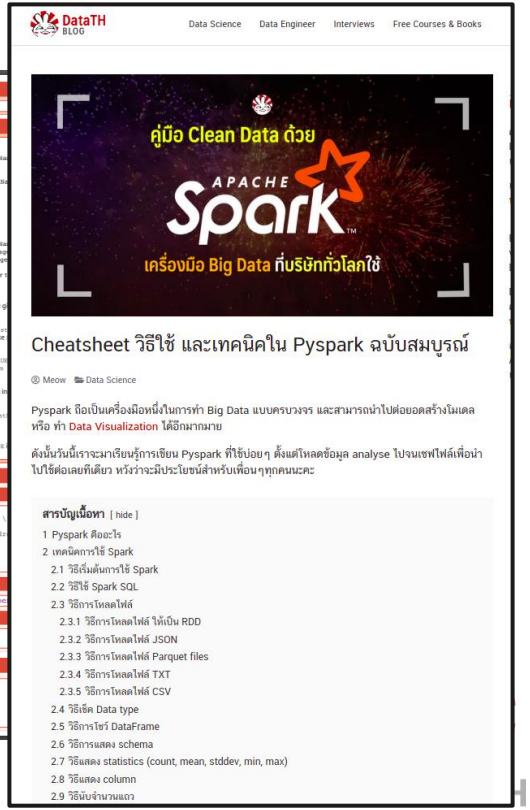
RDD:
https://www.datacamp.com/community/blog/pyspark-cheat-sheet-python

DataFrame:
https://www.datacamp.com/community/blog/pyspark-sql-cheat-sheet

By DataTH

(ภาษาไทย)
https://blog.datath.com/cheatsheet-pyspark/





Workshop 2:
Data Cleansing with Spark

Workshop 2 – Data Cleansing with Spark

มาลองใช้ Apache Spark เพื่อให้ข้อมูลของเรามีคุณภาพกันเถอะ

Input:

- ข้อมูลดิบ (CSV)

Output:

- ข้อมูลที่ทำความสะอาดเรียบร้อย (CSV)

DataTH