

Machine Learning 3(2-2-5), 2/2565

Unit 1

What is “Data Science” and “Machine Learning”

อ.อานันท์ ไม่ประดิษฐ์

**What is
DATA SCIENCE?**

Data + Science

What is Data?

Data is new electricity.

ข้อมูล เป็น พลังงานไฟฟ้าใหม่

Science as we know it.

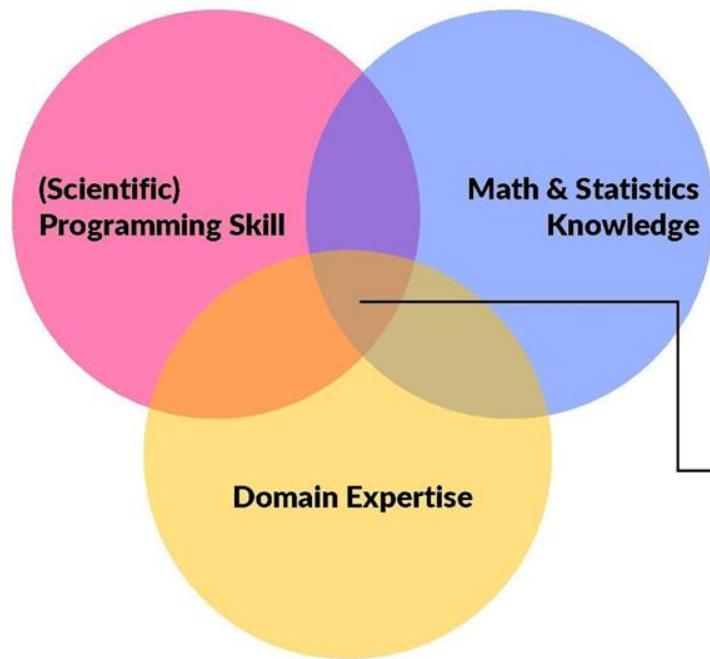
Scientists use data to test hypothesis, theory formation.

วิทยาศาสตร์ตามที่เรารู้ คือ นักวิทยาศาสตร์จะใช้ข้อมูลมาทดสอบสมมิฐานและสร้างทฤษฎี



Andrew Ng

DS Venn Diagram



Data Science
is the intersection of three skills
Data Science គឺ គួរតំណែងនៃ 3 skills

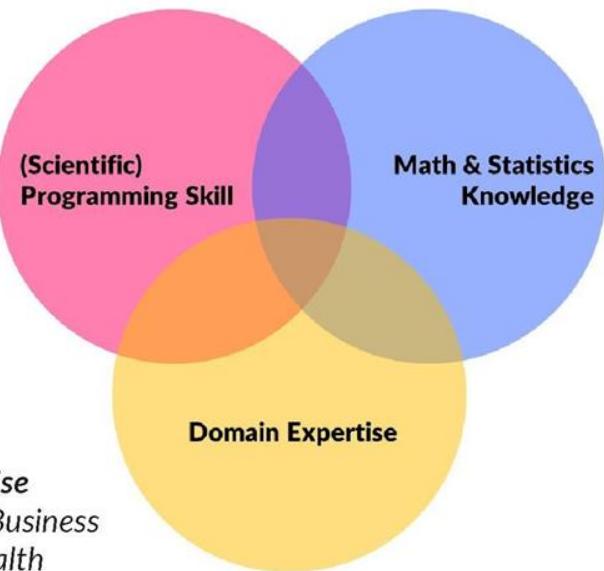
<http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>

Essential Knowledge

ความรู้ที่จำเป็น

Programming skill

- R
- Python
- SQL
- Spark
- Hadoop



Domain Expertise

- Marketing | Business
- Medical | Health
- Politic
- Engineer
- Could be anything

การตลาด | ธุรกิจ
ทางแพทย์ | สุขภาพ
การเมือง
วิศวกรรม
ฯลฯ

Math & Statistics

- Linear algebra
- Calculus
- Descriptive statistics
- Inferential statistics
- Bayesian statistics
- Probability

พื้นฐานทางคณิตศาสตร์

แคลคูลัส

สถิติเชิงบรรยาย

สถิติอ้างอิง

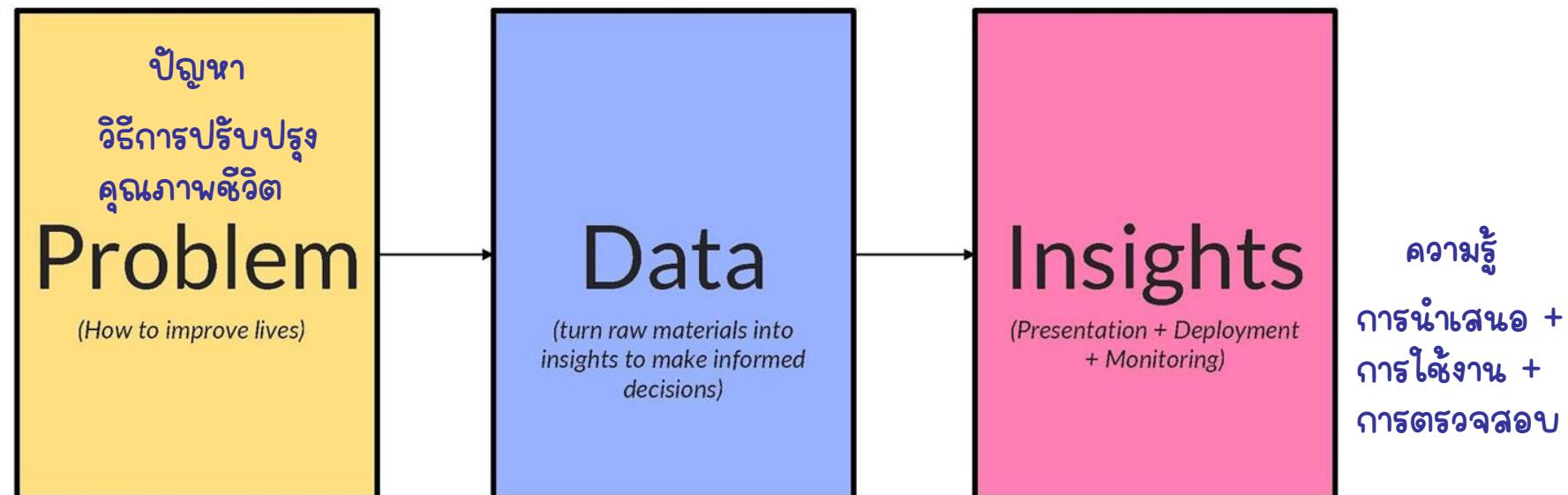
สถิติเบบย์เซียโน

ความน่าจะเป็น

Always

การทำวิทยาการข้อมูลที่ดีนั้น เริ่มต้นจากคำถามที่ดี

Good data science starts with good question



ข้อมูล

แปลงวัสดุเดิมเป็นข้อมูลที่เชื่อถือได้
เพื่อตัดสินใจที่มีข้อมูลอย่างมีลึก

ความรู้
การนำเสนอ +
การใช้งาน +
การตรวจสอบ

What is MACHINE LEARNING?

Can you guess?

x (input)	1	2	3	4	5	6	7	8	9	10
y (output)	1	4	9	16	25	36	49	64	81	<input type="text"/>

You guess the function is x^2



Ask her for a date?

Right! Can you guess this one?

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub Bar	Many	Saturday	OK	Yes
Occasion 3	Pub Bar	Few	Monday	OK	No
Occasion 4	Restaurant	Few	Friday	Bad	Yes
Occasion 5	Restaurant	Many	Friday	Good	No
Occasion 6	Pub Bar	Few	Saturday	OK	Yes
Occasion 7	Pub Bar	Many	Monday	Good	No
Occasion 8	Restaurant	Few	Saturday	OK	No
Occasion 9	Pub Bar	Many	Friday	Bad	No
Occasion 10	Restaurant	Few	Friday	Good	<input type="text"/>

Yes or No?

Machine Learning
is *function approximation*

การประมาณค่าฟังก์ชัน

Machine Learning is *inductive reasoning*

เรียนรู้จากประสบการณ์

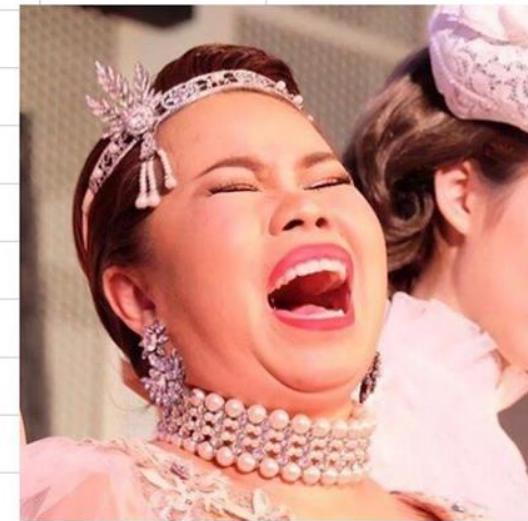
Learning from
experience

การให้เหตุผลแบบอุปนัย

กระบวนการคิดหรือตีความที่มีการใช้อ้อมูลที่มีอยู่เพื่อทำนายหรือสรุปความรู้ใหม่ที่มีความน่าจะเป็นสูง โดยไม่ใช้หลักการทางตรรกศาสตร์.

ML algorithms trying to **MAP** inputs (x's) to output (y)

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub Bar	Many	Saturday	OK	Yes
Occasion 3	Pub Bar	Few	Monday	OK	
Occasion 4	Restaurant	Few	Friday	Bad	
Occasion 5	Restaurant	Many	Friday	Good	
Occasion 6	Pub Bar	Few	Saturday	OK	
Occasion 7	Pub Bar	Many	Monday	Good	
Occasion 8	Restaurant	Few	Saturday	OK	
Occasion 9	Pub Bar	Many	Friday	Bad	
Occasion 10	Restaurant	Few	Friday	Good	



It's good to have lots data but ...

REPRESENTATION

is the quality we desire

การมีจั่วหัวมากมีประโยชน์นี้ แต่การแสดงจั่วหัวเป็นสิ่งที่เราต้องการคุณภาพ

Representation = วิธีที่จั่วหัวถูกแทนที่ในรูปแบบที่ระบบคอมพิวเตอร์สามารถทำงานการคำนวณได้. การเลือกหรือสร้าง representation ที่เหมาะสมสามารถมีผลมากต่อประสิทธิภาพของระบบ ML ใน การเรียนรู้และดำเนินการ ในการนี้, ความหมายของ "representation" ใน ML นั้นจะเกี่ยวข้องกับการแปลงจั่วหัวที่เหมาะสมสำหรับการนำเข้าไปในตัวแบบ (model) หรืออัลกอริทึมที่กำลังถูกพัฒนา

ML Example

ตัวอย่าง Machine Learning

- Predict breast cancer การคำนวณว่าผู้ป่วยมีโอกาสติดเชื้อมะเร็งเต้านม
- Weather forecast การคำนวณสภาพอากาศ
- Watson analytics platform พัฒนาโดย IBM
- Google flu trend การระบุความนิยมในชุมชน
- Obama winning the US election



Predicting crime before it happens

คำจำกัดความที่แนะนำสมบูรณ์ Machine Learning

A proper definition of ML

ความสามารถในการเรียนรู้

“ Machine Learning: Field of study that gives computers **the ability to learn** without being explicitly programmed.

การเรียนรู้ของเครื่อง: สาขาวิชาที่ให้
ความสามารถให้คอมพิวเตอร์เรียนรู้โดย
ไม่ต้องถูกโปรแกรมไปอย่างชัดเจน



Arthur Samuel (1959)

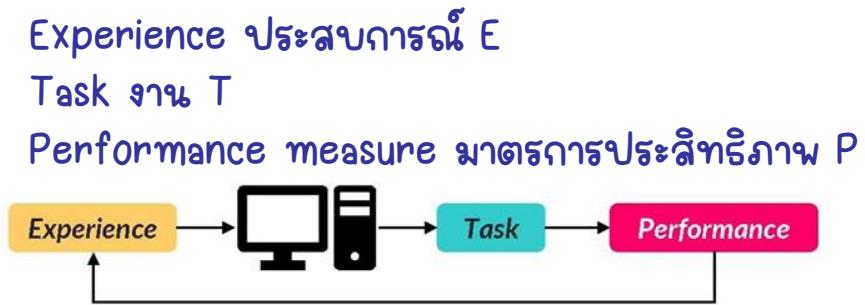
A more modern def in 1998

Well posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.



Tom Mitchell (1998)

ปัญหาการเรียนที่ดูดต้อง: โปรแกรมคอมพิวเตอร์ดูดต้องเรียนรู้จากประสบการณ์ต่อเนื่องกับงานและวัดประสิทธิภาพด้วยมาตรฐาน P



ถ้าประสิทธิภาพของมันในการทำงาน P on T
ตามที่ดูดได้โดยมาตรฐานประสิทธิภาพ measure by P
พัฒนาด้วยประสบการณ์ improves with Experience E

Types of ML algorithms

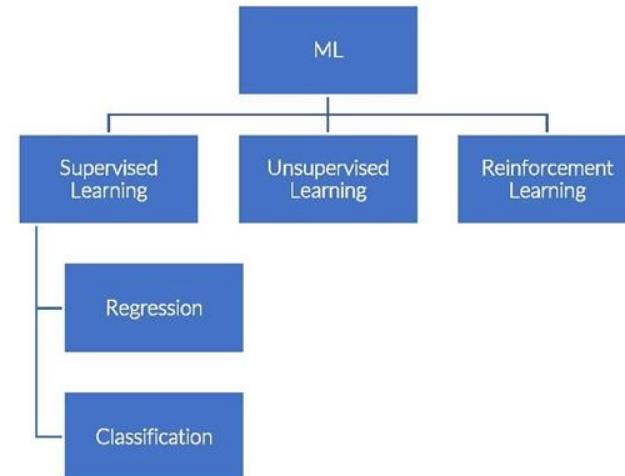
1. Supervised Learning

ກາຕກອຍ (ຄາເຕເຈາ) ຕຳລາວ

- i. Regression
- ii. Classification

2. Unsupervised Learning

3. Reinforcement Learning



What's Artificial Intelligence



Function approximation การประมาณค่าฟังก์ชัน

$$Y = f(X)$$

Model = Algorithm (Data)

Predicting new unseen data

เราสร้างโมเดลเพื่อแกนตัวโลก (ปัญหาที่เราพยายามแก้)

We build models to represent the world (problem we're trying to solve)

In the past
ในอดีต

Model

การทำนายผลลัพธ์ของข้อมูลที่ไม่
เคยมีมาก่อนหรือไม่เคยเห็นมาก่อน



And the goal of Machine Learning is **GENERALIZATION**

Beyond (past) data you used to build the model

ແລະ ຖຸດມູ່ຂໍ້ມາຍບອງ machine learning ຄື່ອ generalization ນອກເໜີ້ອຈາດຈົ້ນມູລໃນອົດຕະກູ່ ໃຊ້ສ້າງໄມ້ເດລ

Generalization หมายถึงการให้แน่ใจว่าโมเดลของเรารู้จักฝึกหัดกับข้อมูลมากเกินไป (overfit)

**Generalization means we make sure
our models don't overfit the data**

Also don't underfit

ไม่ให้เกิดการฝึกหัดไม่เพียงพอ

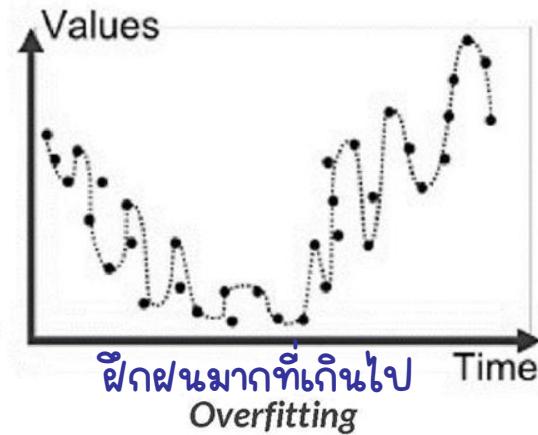
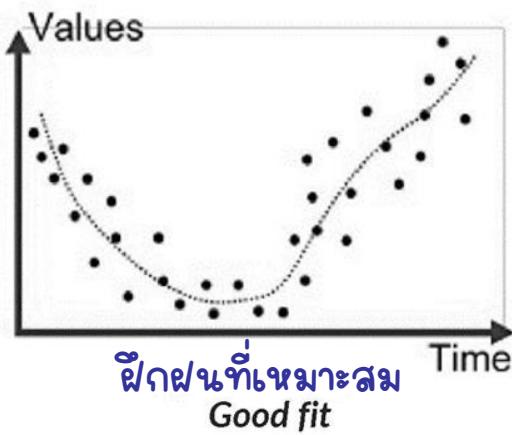
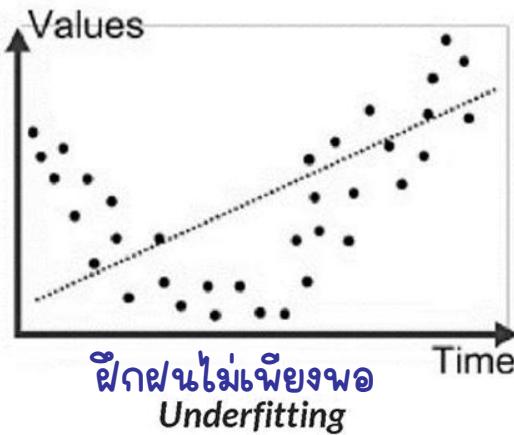
We mean
data in the
past

หมายถึง ข้อมูลในอดีต

ไม่สามารถทำนายข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้

If your model **can't generalize to new unseen data**, then you're in trouble.

ถ้าไม่เดลอนองคุณไม่สามารถใช้แบบจำลองในการทำนายข้อมูลใหม่ที่ไม่เคยเป็นที่รู้จักมาก่อน, แล้วคุณจะพบว่ามีปัญหา



https://cdn-images-1.medium.com/max/1600/1*6vPGzBNppqMHllg1o_se8Q.png

This model looks okay as it fit underlying patterns in the data well. Though, not 100% correct ?

ไม่เดลอนี้ดูโอเค เพราะมันเข้ากับรูปแบบซ่อนเร้นในข้อมูลได้ดี
แม้กระทั่งไม่สูงต้องทั้งหมด 100%?

SUMMARY

สรุป

- Data Science in Thailand is still very young วิชาการจัดการข้อมูลในประเทศไทยยังเป็นเยาวชนอยู่
- ML is data scientist's arsenal tool เป็นเครื่องมือที่สำคัญในชุดเครื่องมือของนักวิชาการจัดการข้อมูล
- ML is **Function Approximation**, learning from experience เป็นการประมาณฟังก์ชัน, การเรียนรู้จากประสบการณ์
- **Representation & Generalization** (avoid overfitting | underfitting) Representation & Generalization (การหลักเลี้ยงการจำลอง/มากเกินไป | การจำลองไม่เพียงพอ)
- 3 types of ML algorithms including supervised, unsupervised and reinforcement learning 3 ประเภท
- Regression predicts numbers | Classification predicts categories การ Regression ทำนายตัวเลข | การ Classification ทำนายหมวดหมู่

Classification

- 1) Features selection
- 2) Category selection
- (3)

Group of 5-6

Red ball	Brown shoe
Blue cube	Yellow flower pot
Green toy car	Black smartphone
Yellow rubber duck	Blue backpack
Purple dinosaur toy	Green umbrella
Orange building block	Purple flower
Pink teddy bear	Red toy train
Small yellow plastic boat	Green marker
Blue toy robot	Blue puzzle piece
Brown wooden toy horse	Yellow rubber ball
Silver key	Pink headphones
White seashell	Wooden spoon
Black sunglasses	White clock
Goldfish plush toy	Orange toy airplane
Wooden block	Black backpack
Purple crayon	Brown coffee mug
Green plastic dinosaur	Blue sunglasses
Laptop	Silver keychain
Red apple	Gray laptop bag
Yellow tennis ball	Green plant
Blue bicycle helmet	Yellow toy dump truck
Pink plastic cup	Red stapler
Orange kite	Blue water bottle
Gray book	Pink notebook
White candle	Wooden picture frame

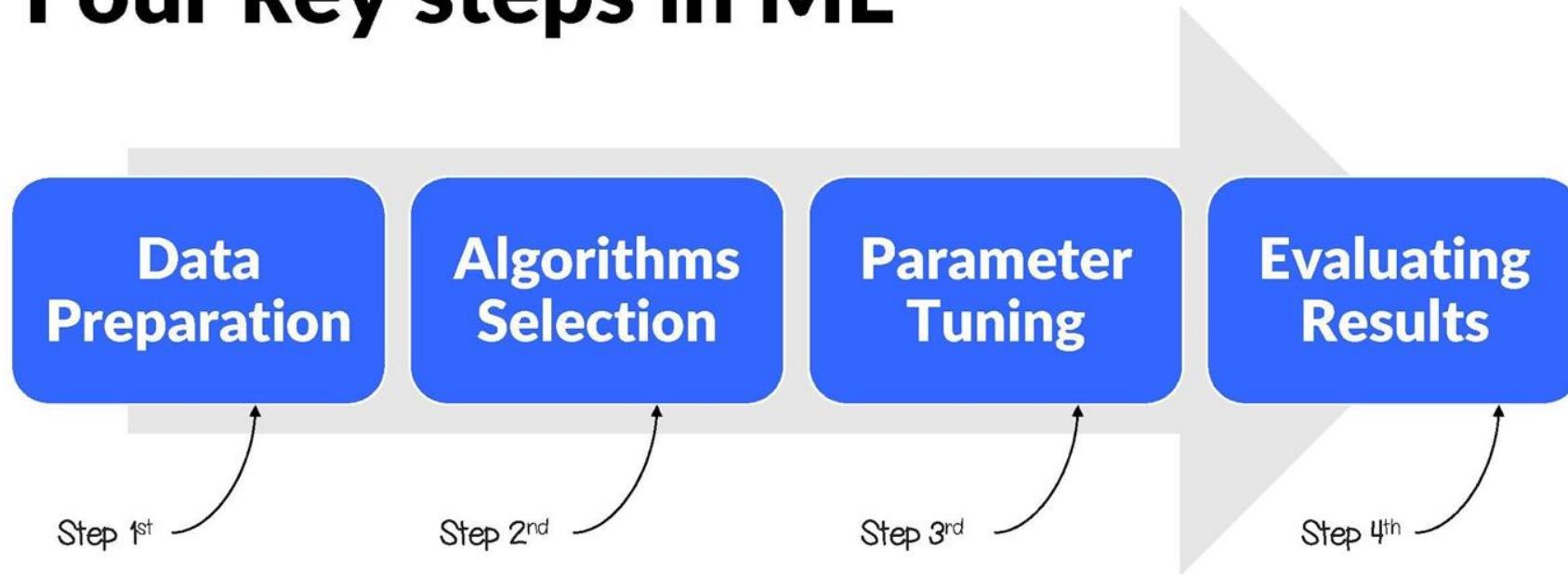
Machine Learning 3(2-2-5), 2/2565

Unit 2

Unit 2: Four key steps in Machine Learning

อ.อานันท์ ไม่ประดิษฐ์

Four key steps in ML



DATA PREPARATION

กระบวนการเตรียมข้อมูลก่อนที่จะนำไปใช้ในการวิเคราะห์หรือการประมวลผลในงานทางด้านวิทยาศาสตร์ข้อมูล (Data Science) หรือการเรียนรู้ของเครื่อง (Machine Learning)

Data Format

The diagram illustrates the structure of a dataset. At the top, a pink double-headed arrow spans across the first four columns (X1 to X4) and is labeled *Variables | Features*. To the right of the fifth column (Y), another pink double-headed arrow is labeled *Target Variable*. A vertical pink double-headed arrow on the far right is labeled *Data Points | Observations*, pointing from the bottom row to the top row of the table.

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub Bar	Many	Saturday	OK	Yes
Occasion 3	Pub Bar	Few	Monday	OK	No
Occasion 4	Restaurant	Few	Friday	Bad	Yes
Occasion 5	Restaurant	Many	Friday	Good	No
Occasion 6	Pub Bar	Few	Saturday	OK	Yes
Occasion 7	Pub Bar	Many	Monday	Good	No
Occasion 8	Restaurant	Few	Saturday	OK	No
Occasion 9	Pub Bar	Many	Friday	Bad	No
Occasion 10	Restaurant	Few	Friday	Good	No

Variable Types

1. Binary (1/0)
2. Categorical
3. Integer
4. Continuous ຕ່ອນເນື້ອງ

การเลือกคุณลักษณะ

Feature Selection

- Choose relevant set of features that represent the problem
- Curse of dimensionality: 2^n feature

เลือกเซตของคุณลักษณะที่เกี่ยวข้อง
และแทนปัญหา

กระบวนการการปรับแต่ง

Feature Engineering

- Recode การแปลงรหัสหรือค่า
- Combine multiple variables (PCA) การรวมกันของตัวแปรหลาย ๆ ตัว Principal Component Analysis
- Extract new information from existing feature

การสกัดข้อมูลใหม่จากคุณลักษณะที่มีอยู่ (เดิมข้อมูลที่มีประโยชน์มาก)

การสร้างข้อมูลใหม่จากคุณลักษณะที่มีอยู่

Gentle intro to Feature Engineering

ID	Name	DOB	Math	Stats	Art	Music	Design
1	Mr. David Beckham	13 / 05 / 1975	85	90	50	42	49
2	Ms. Angelina Jolie	20 / 09 / 1988	50	60	90	95	88
3	Mr. Khalid Khan	05 / 01 / 1950	32	65	70	72	65
4	Mr. Hibino Takamoto	31 / 12 / 1995	75	76	80	50	30

Question:

What new variable can we create from this dataset?

Think at least 5 new variables

We can use
`dplyr::mutate()` to
create new variables

ข้อมูลที่สูญหาย

Missing Data

Mean | Median
imputation

การเติมค่าเฉลี่ย, ค่ามัธยฐาน

การประมาณค่า

1. Approximated

การคำนายค่า

2. Predicted (Supervised Learning)

การลบข้อมูลที่หายไป

3. Removed

If we have large data points and NA is not much, we may consider removing them

ถ้าเรามีจำนวนข้อมูลมากและข้อมูลที่หายไป (NA) ไม่มีมากนัก เราอาจพิจารณาที่จะลบข้อมูลที่หายไปนั้นได้

Approximated or Removed?

```
Console ~/Desktop/R PROGRAMMING/
> head(complete.cases(biopsy), 200)
 [1] TRUE  TRUE
[14] TRUE  FALSE TRUE  TRUE
[27] TRUE  TRUE
[40] TRUE  FALSE TRUE  TRUE
[53] TRUE  TRUE
[66] TRUE  TRUE
[79] TRUE  TRUE
[92] TRUE  TRUE
[105] TRUE  TRUE
[118] TRUE  TRUE
[131] TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  FALSE TRUE  TRUE  TRUE  TRUE
[144] TRUE  TRUE  FALSE TRUE  TRUE
[157] TRUE  TRUE  FALSE TRUE  TRUE  TRUE  TRUE  TRUE  FALSE TRUE  TRUE  TRUE  TRUE  TRUE
[170] TRUE  TRUE
[183] TRUE  TRUE
[196] TRUE  TRUE  TRUE  TRUE  TRUE
```

R functions:

`complete.cases(df)` สำหรับตรวจสอบว่าข้อมูลมี NA หรือเปล่า?

`tidyverse::drop_na(df)` สำหรับ drop case ที่มีข้อมูลไม่ครบถ้วน

EmployeeID	Salary (\$)	Experience (years)	Department
1	65000	4	HR
2	?	6	IT
3	75000	8	Sales
4	60000	?	IT
5	70000	5	Marketing

EmployeeID	Salary (\$)	Experience (years)	Department	Salary / Exp
1	65000	4	HR	16250
2	67500	6	IT	11250
3	75000	8	Sales	9375
4	60000	5.75	IT	10434.78
5	70000	5	Marketing	14000

ALGORITHM SELECTION

There are 3 types of algorithms

	Algorithms	Main Tasks
Unsupervised learning	K-Means Principal Component Analysis (PCA) Association Rules Social Network Analysis (SNA)	<i>Tell me what patterns exist in my data - descriptive</i>
Supervised learning	Linear Regression Logistic Regression Decision Tree Random Forests KNN Support Vector Machine Neural Networks	<i>Use the pattern in my data to make predictions - predictive</i>
Reinforcement learning	Multi-Armed Bandits (UBC, Thompson Sampling)	<i>Use the pattern in my data to make predictions and - improve - these predictions as more results come in</i>

Source: Data Science For The Layman (2017)

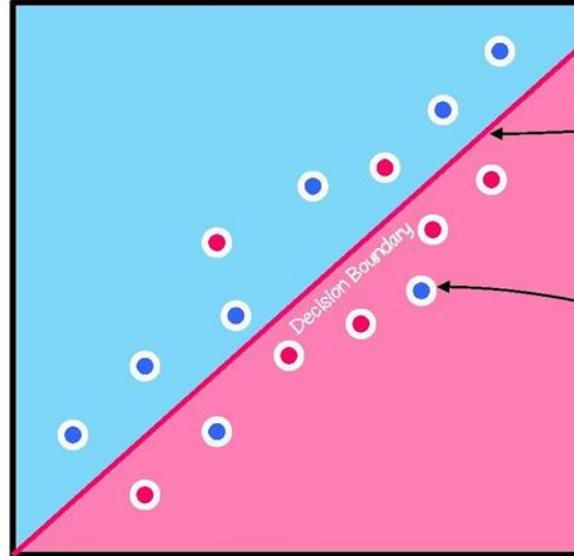
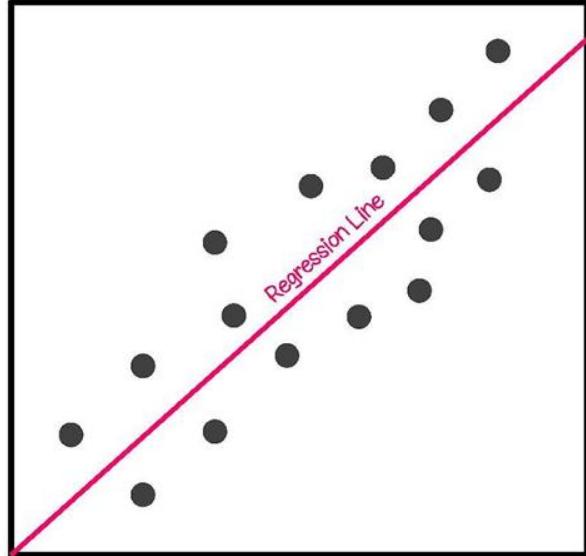
การเรียนรู้แบบไม่มีการสอน: บอกฉันว่ามีลักษณะอะไรในข้อมูลของฉัน - บรรยาย

การเรียนรู้แบบมีการสอน: ใช้ลักษณะในข้อมูลของฉันเพื่อทำการพยากรณ์ - ทำนาย

ใช้ลำดับข้อมูลเพื่อทำการพยากรณ์และปรับปรุงการพยากรณ์เหล่านี้เมื่อมีผลลัพธ์มากขึ้น ลองผิดลองถูก เซ่น

การทดลองทางการตลาด: เลือกวิธีการโฆษณาหลายรูปแบบเพื่อดูว่าความต้องของโฆษณาไหนที่ได้รับการคลิกมากที่สุด.

Regression vs. Classification



ເລື່ອນແບ່ງຄລາສ: ເລື່ອນທີ່ແບ່ງຈົ້ນມູນລົງແດນແລະຈົ້ນມູນລັບນຳເຈີນໄດ້ດີທີ່ສຸດ

Decision Boundary:
The line that best split
red vs. blue data points

100% in reality may not
be feasible, some blue
points are mistakenly
classified as red

ໃນຄວາມເປົ້າຂອງ 100% ອາຈະໄມ່ເປົ້າໄປໄດ້, ບາງຈຸດລັບນຳເຈີນ
ອາຈຸດຈາກຈຳແນດຜິດເປົ້າສີແດນບໍ່ຈາກ

NO FREE LUNCH

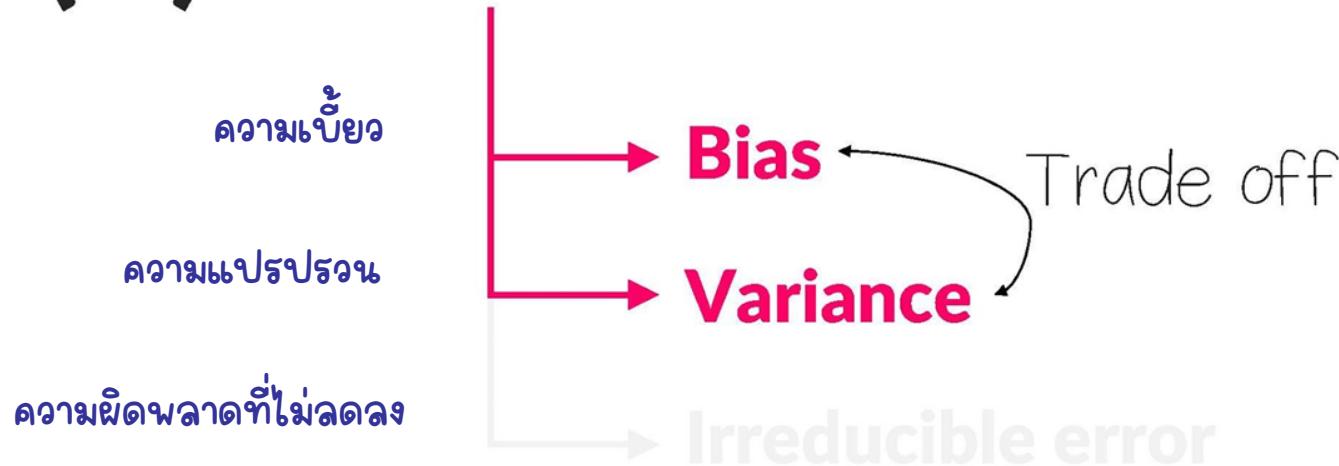
- Speed
- Accuracy
- Bias & Variance Tradeoff



โมเดลของเราระบด้วยจั๊มพิดพลาด 3 ประการ

Our model consists of 3 types of error

$$Y = f(X) + \text{Error}$$



กระบวนการหรือการตัดสินใจที่ต้อง^{ต่อ}
ทำอะไรอ้างสองเรื่องหรือมาตรฐานที่
จัดແຍ້ງດັນ

Low variance algorithm(high bias):

→ Less complex, consistent but inaccurate on average → underfit

Low bias algorithm(high variance):

→ More complex, accurate on average but inconsistent → overfit

อัลกอริทึมที่มีความแปรปรวนต่ำ (เบี้ยงหลง):

→ น้อยซับซ้อน, มีน้อยแต่ไม่แม่นยำในเฉลี่ย → การปรับสัมผู้ลงน้อย (underfit)

อัลกอริทึมที่มีเบี้ยงหลง (ความแปรปรวนสูง):

→ ซับซ้อนมาก, แม่นยำในเฉลี่ยแต่ไม่มั่นคง → การปรับสัมผู้ลงมาก (overfit)

นี่เป็นอุดมคติ แต่ยากในการปฏิบัติ

This is ideal,
but difficult
in practice

Low
variance

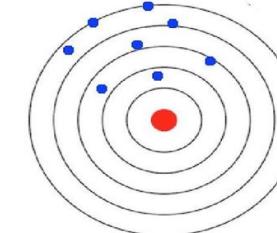
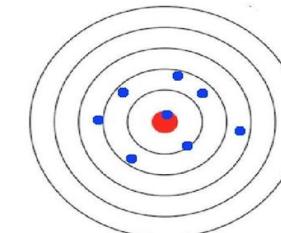
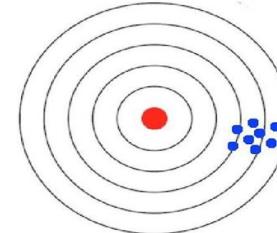
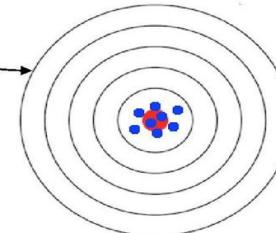
Low bias

High bias

High
variance

ถึงนี้จะต้องหลีกเลี่ยง

This must
be avoided



Example Models

High Bias

Having lots of assumptions to do function approximation

การมีเบี้ยວสูงนั้นหมายถึงการมีการสมมติมากหมายเพื่อทำฟังก์ชันประมาณค่า

Linear Regression
Logistic Regression
Linear Discriminant Analysis

Low Bias

Having little or few assumptions about the function approximation

Decision Tree
KNN
SVM

การมีเบี้ยວต่ำนั้นหมายถึงการมีการสมมติน้อยหรือไม่มีเลยเดียวกับการประมาณค่าของฟังก์ชัน

ความแปรปรวนต่ำ: ข้อมูลใหม่ไม่ทำให้โมเดลเปลี่ยนแปลงมาก

Low Variance

New data does not cause model to change much

High Variance

New data causes model to change much i.e. prediction results vary significantly compared to low variance algorithms

ความแปรปรวนสูง: ข้อมูลใหม่ทำให้โมเดลเปลี่ยนแปลงมาก เช่น ผลการคำนวณมีความแปรผันมากเมื่อเทียบกับอัลกอริทึมที่มีความแปรปรวนต่ำ

การปรับพารามิเตอร์

PARAMETER TUNING

กระบวนการในการปรับค่าของพารามิเตอร์ในโมเดลเพื่อทำให้โมเดลมีประสิทธิภาพมากที่สุด



**Music Composer is
a very good example**

นักแต่งเพลงเป็นตัวอย่างที่ดีมาก

វិជ្ជាគុណភាព

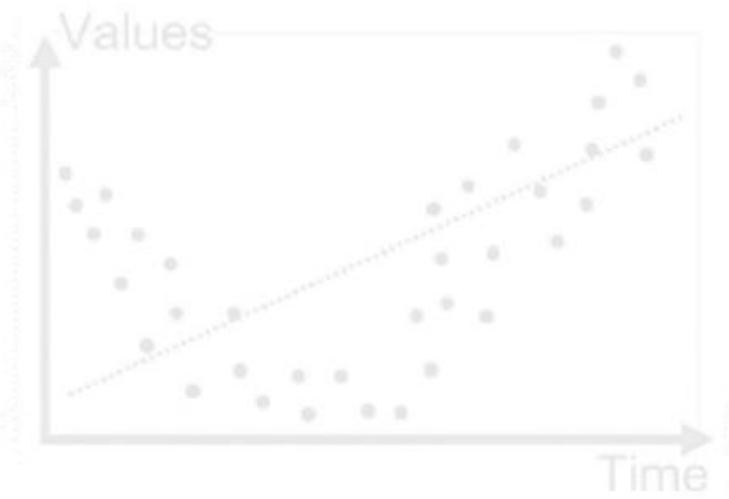
There are so many learners.

**Even the same learner can have
different performances.**

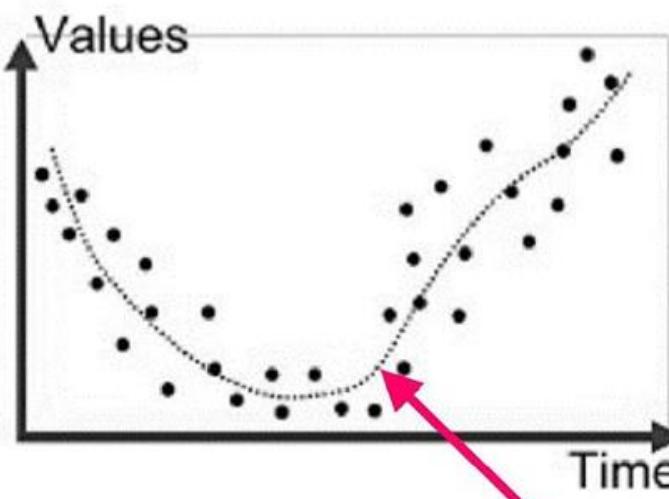
នៅពេលវិជ្ជាគុណភាពរបស់គ្រួសារមិនស្ម័គ្រប់គ្នា

เราปรับพารามิเตอร์เพื่อให้ผู้เรียนของเราได้ผลลัพธ์ที่ดีที่สุด (Good Fit)

We tune parameter to get the best results (**GOOD FIT**) from our learner.



Underfitting



Good fit



Overfitting

The parameters that can be tuned called "**HYPERPARAMETERS**"

https://cdn-images-1.medium.com/max/1600/1*6vPGzBNppqMHllg1o_se8Q.png

พารามิเตอร์ที่สามารถปรับแต่งได้เรียกว่า 'ไฮเปอร์พารามิเตอร์'

Some models have hyperparameters, some don't.

Stochastic Gradient Boosting

```
method = 'gbm'
```

Type: Regression, Classification

Tuning parameters:

- n.trees (# Boosting Iterations)
- interaction.depth (Max Tree Depth)
- shrinkage (Shrinkage)
- n.minobsinnode (Min. Terminal Node Size)



Hyperparameters of GBM algorithms

ไฮเปอร์พารามิเตอร์ของอัลกอริทึม GBM

Required packages: gbm , plyr

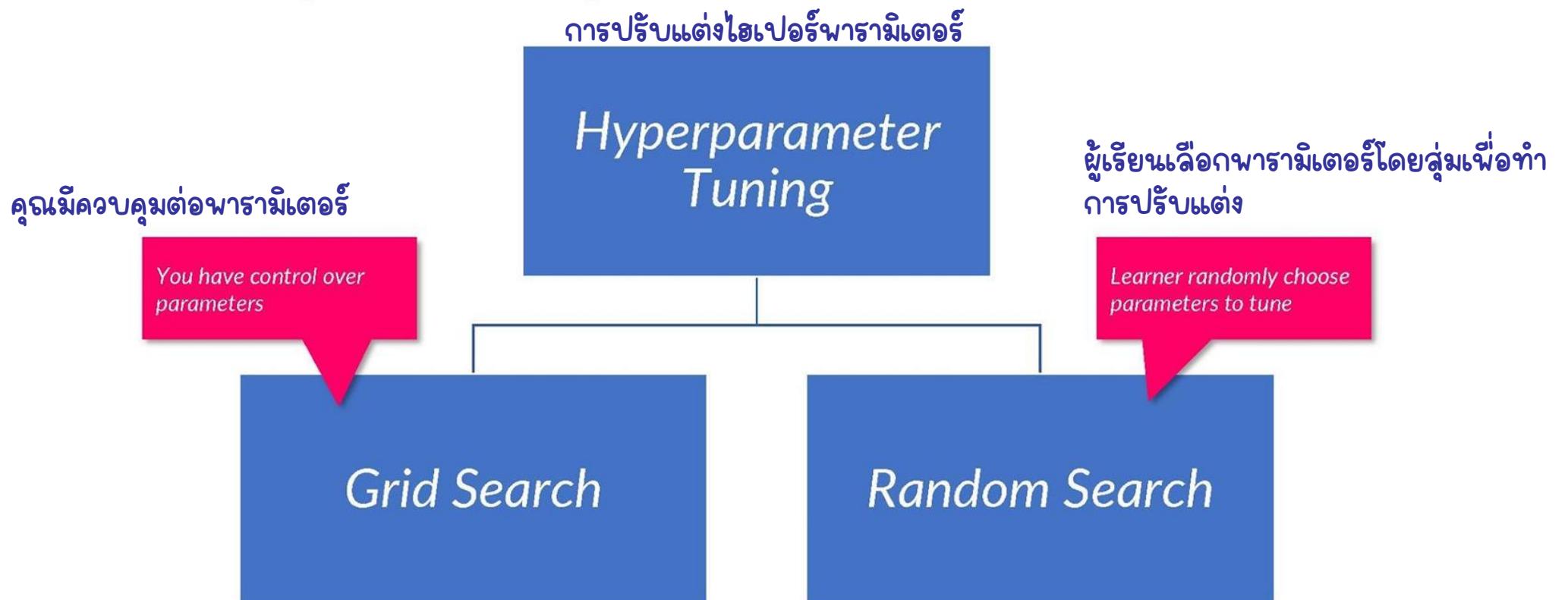
A model-specific variable importance metric is available.

<http://topepo.github.io/caret/train-models-by-tag.html>

การค้นหาการปรับแต่งที่ดีที่สุดใน machine learning เป็นปัญหาการปรับแต่ง

SEARCH for the best tune

ML is optimization problem



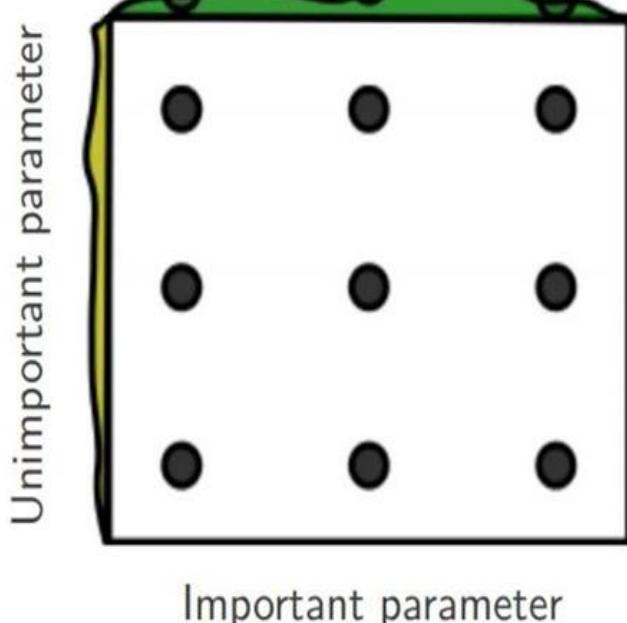
การปรับค่าของ hyperparameters ของโมเดลโดยการทดลองทุกกฎแบบค่าที่เป็นไปได้จากการริดหรือตาราง

การปรับค่าของ hyperparameters ของโมเดลโดยการทดลองค่า hyperparameters ที่สุ่มมาจากการซ่อนที่กำหนดไว้

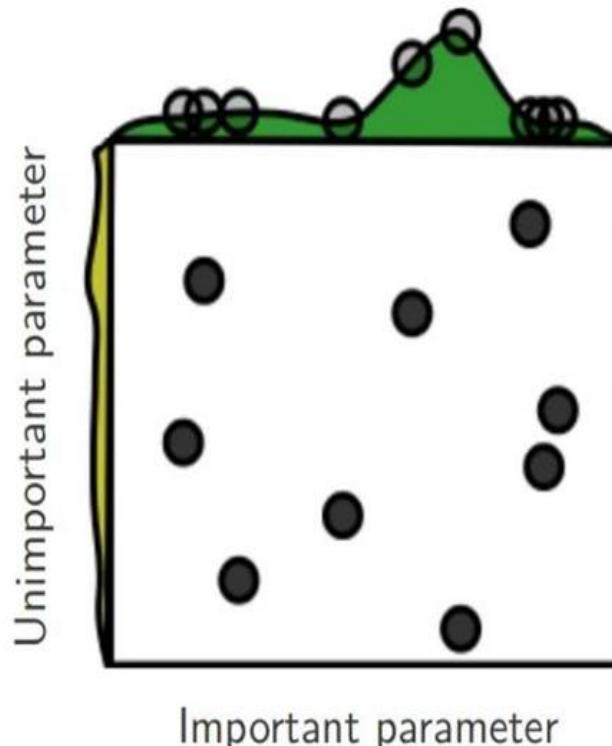
និងនឹងលើកបង្កើរឱ្យនា បានគ្រែការនៃ random search ជាមីត្តភាពភាពមាត្រីន បានគ្រែក្រោមផ្សេង

It really depends on the learners. Sometimes it's more efficient to use random search, sometimes it's not.

Grid Layout



Random Layout



กระบวนการในการวิเคราะห์ผลลัพธ์หรือการประเมินผลของภารกิจดังนี้
ไม่เดล

EVALUATING RESULTS

Performance Metrics

ตัววัดประสิทธิภาพ

ปั๊มจากการจำแนกประเภท

Classification problems

- **Confusion Matrix**
- **% Accuracy**
- **Area Under Curve (AUC)**
- **Logarithmic Loss (Log Loss)**
- **Cohen's KAPPA #for imbalanced problems**

เมทริกซ์ความสับสน ตาราง cross tabulation
ความแม่นยำเป็น %
พื้นที่ใต้เส้นโค้ง²
การสูญเสียทางลօดการพิม
ค่าปีกของโคเอนลำหรับปั๊มาก็ไม่สมดุล

Regression problems

ปั๊มจากการคำนวณค่าต่อเนื่อง

- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**

ค่าเฉลี่ยของความผิดพลาดสหูรณ์
รากที่สองของค่าเฉลี่ยของความผิดพลาดกำลังสอง

Cancer Detection



Confusion Matrix

Total n = 100		Reference	
Predicted	Cancer	No Cancer	
Cancer	60	3	
No Cancer	2	35	

คือตาราง cross tabulation (เรียกสั้นๆว่า cross-tab) แบบพิเศษที่นำ
ข้อมูลสองคอลัมน์คือ actual, prediction มาทำเป็นตารางลรุปผล cross

ความแม่นยำคือผลรวมของแนว角 диагonal ด้วยขนาดตัวอย่างทั้งหมด
Accuracy is the sum of diagonal divided by total sample size

$$60+35 / 100 = 95\%$$

Confusion Matrix

Total n = 100		Reference	
Predicted	Cancer	No Cancer	
Cancer	60 (A)	3 (B)	
No Cancer	2 (C)	35 (D)	

ถ้าผู้ป่วยเป็นมะเร็ง เราหายถูกเท่าไร? (%)

$$\text{Sensitivity} = \mathbf{A} / \mathbf{A+C}$$

ถ้าผู้ป่วยไม่ได้เป็นมะเร็ง เราหายถูกเท่าไร? (%)

$$\text{Specificity} = \mathbf{D} / \mathbf{B+D}$$

ผล prediction "cancer" ของเรากลับกี่ %

$$\text{Precision} = \mathbf{A} / \mathbf{A+B}$$

ในผู้ป่วยโรคมะเร็งทั้งหมด เรายำนาถูกเท่าไร? (%)

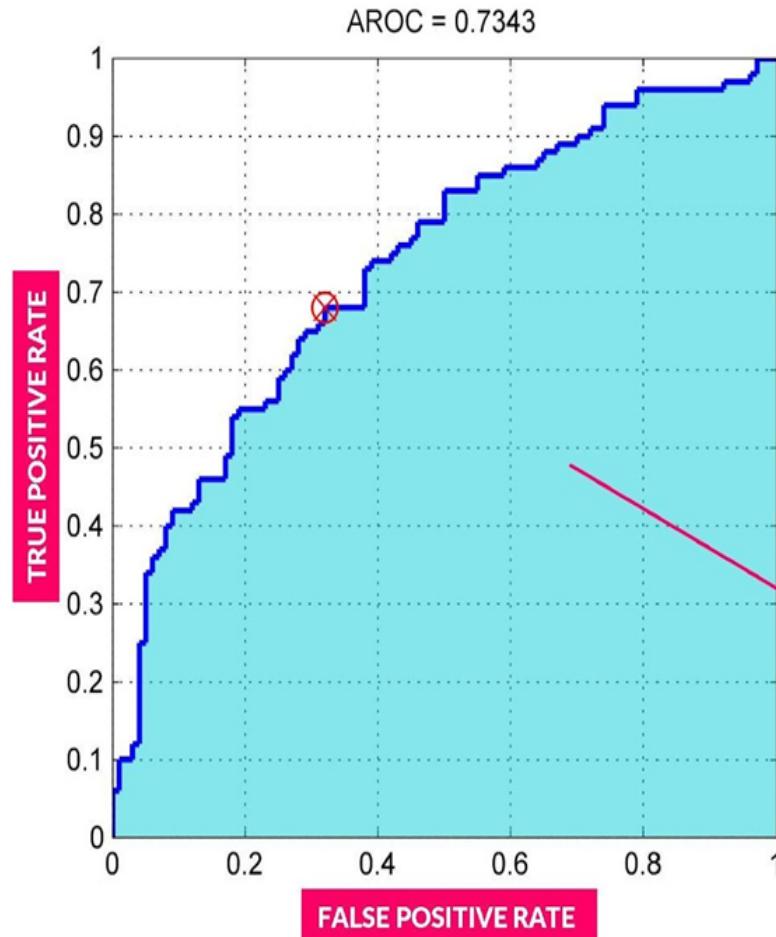
$$\text{Recall} = \mathbf{A} / \mathbf{A+C}$$

Measure of overall test's accuracy

$$\text{F1} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Area Under Curve (AUC)

พื้นที่ใต้เส้นโค้ง

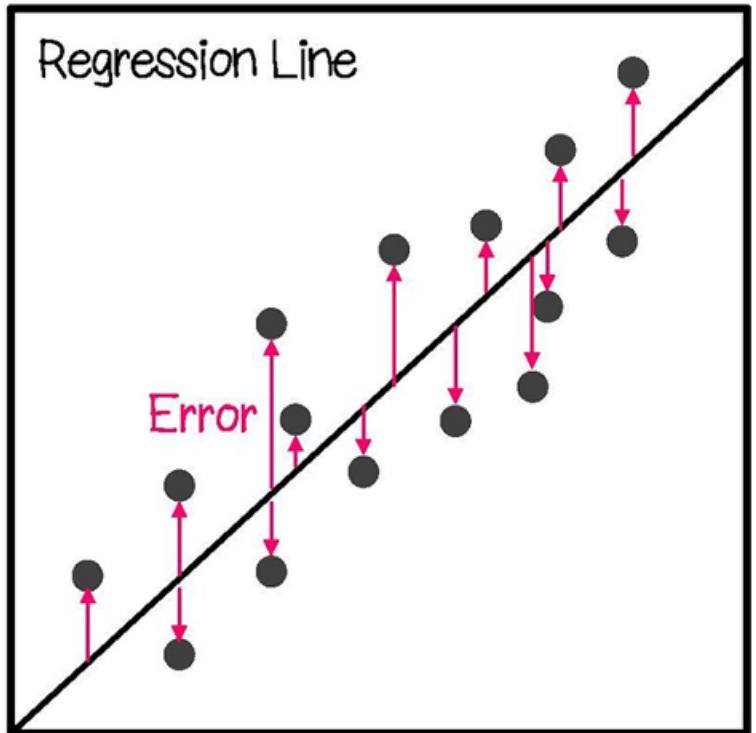


For binary classification, sometimes
we want to maximize AUC

สำหรับการจำแนกแบบใบหน้าบางครั้งเราต้องการเพิ่ม AUC

ค่าเฉลี่ยของความคลาดเคลื่อนที่สูงมากกำลังสอง

Root Mean Squared Error (RMSE)



นิพจน์ทางคณิตศาสตร์มักดูน่ากลัวเสมอ

Mathematical Expression always look scary

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Actual - Predicted y_i)^2}{N}}$$

$$RMSE = \sqrt{\frac{\text{sum(error}^2)}{N}}$$

Remember this one

$$RMSE = \sqrt{\text{mean } (error^2)}$$

ค่าเฉลี่ยของความคลาดเคลื่อนของค่าทำนาย (หรือค่าโมเดล) จากค่าจริงที่หลากหลาย ค่า เป็นการนำความคลาดเคลื่อนของแต่ละค่ามายกกำลังสอง, หาค่าเฉลี่ย, และทำการดึงรากที่สองของผลลัพธ์นั้น เพื่อให้ได้มาตรฐานที่บ่งบอกถึงความคลาดเคลื่อนรวมของโมเดลในการทำนาย สำหรับ ประสิทธิภาพที่ดีของโมเดล, RMSE ควรมีค่าน้อย ๆ มักใช้ในงานที่เกี่ยวข้องกับการทำนายค่าต่อเนื่อง เช่น การทำนายราคา, อุณหภูมิ, หรือค่าทางการอิเล็กทรอนิกส์ทางอิเล็กทรอนิกส์.

Recall the goal of ML is “GENERALIZATION”

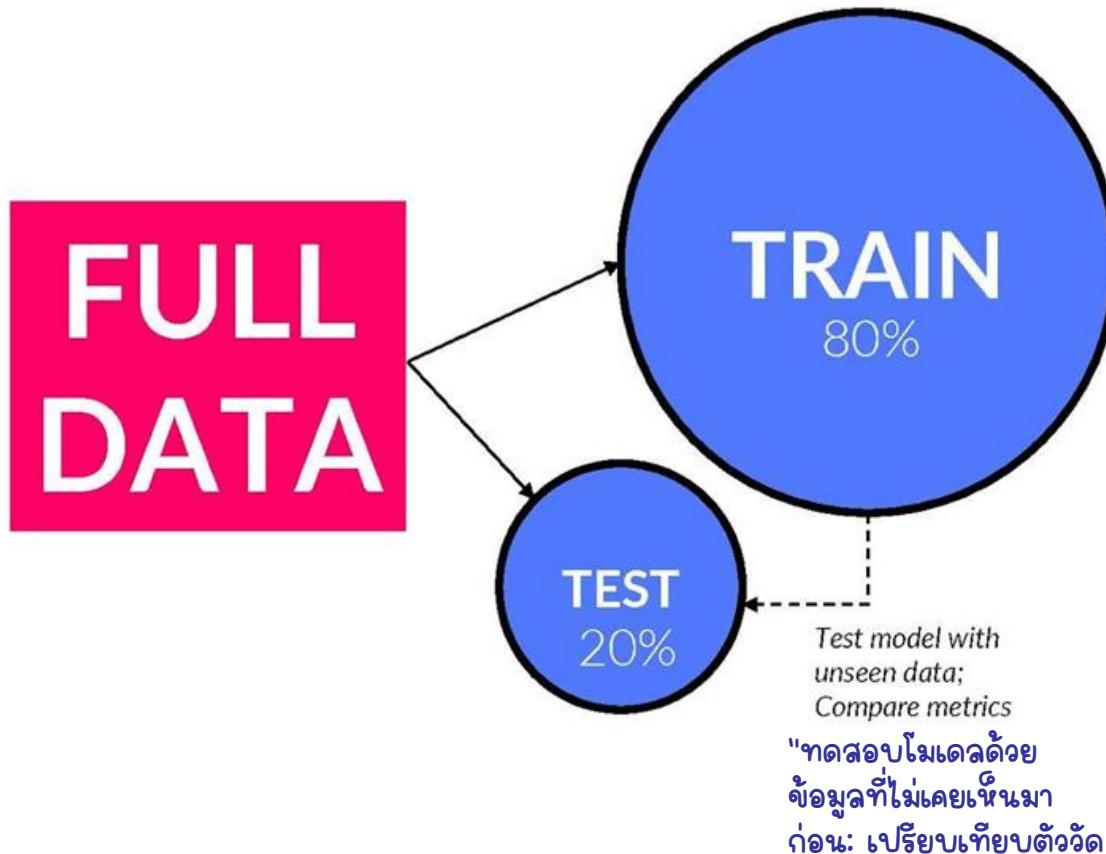
จำไว้ว่าเป้าหมายของ Machine Learning คือการทำนายที่ว่าไป

We must test our model on new unseen dataset

เราต้องทดสอบโมเดลของเราด้วยชุดข้อมูลที่เรารายก่อนมาก่อน

ໂນເມລີຂອງເຮົາກໍອມກຳຈານໄດ້ແລ້ວຮູ້ອ່ານ?

Is our model good to go?



Training set (build model)
ຊູ້ຜູ້ກ (ສໍາຮັບເບຈຳລອງໂນເມລີ)

Testing set (evaluate model)
ຊູ້ກາດສອບ (ປະເມີນໂນເມລີ)

Decision Criterion: ແກ່ນ໌ກາຣຕັດສິນ
The model **passes the test** if
tested performance is similar
trained performance [metric]
ໂນເມລີຜ່ານກາຣກາດສອບໜາກປະສິກິພາພທີ່ກຳລົບມີຄວາມ
ຄລ້າຍຄລືງດັບປະສິກິພາພທີ່ສູງຜູ້ກ [ຕິວວັດ]

ເທົ່ານີ້ມີການສູ່ມື້ອໍາຍ່າງໜີ້ອໍາຈົ້ວລະບົ້າ

Resampling techniques

1. Train-test split

ກາຮະບົ້າຈົ້ວລະບົ້າປິ່ນຫຼຸດເພື່ອກະລຸດກາລົກສອບ

*This method is preferred
but it's very expensive for
large dataset*

2. k-Fold Cross-Validation

ກາຮຳ Cross-validation ແບບ k-fold

3. Repeated k-Fold Cross-Validation

ກາຮຳ Cross-validation ແບບ k-fold ຊຳ

4. Bootstrap

ເທົ່ານີ້ Bootstrap

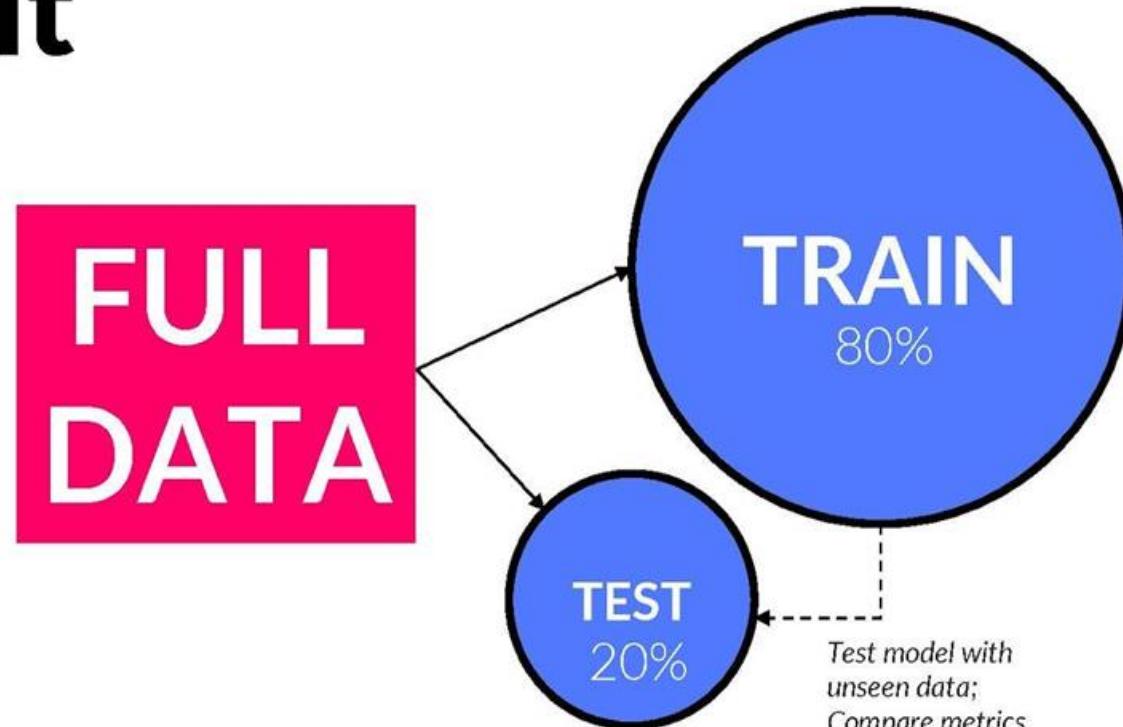
5. Leave One Out Cross-Validation

ກາຮຳ Cross-validation ແບບ Leave-one-out

វិធីទីនៃប្រតិបត្តិការណ៍ដែលបង្កើតឡើងដើម្បីបង្កើតនូវមុននឹងការសរស់សុំ

The most efficient method is train-test split

"Train-test split" គឺ "ការបែងចែកមូលដំឡូងជាពាណិជ្ជកម្ម" ដោយការស្វែងរកបែងចែកមូលដំឡូងសំខាន់ៗ: ជូនដែលបានបង្កើតឡើងដើម្បីសរស់សុំ និងជូនដែលបានបង្កើតឡើងដើម្បីពន្លាសម្រាប់ប្រតិបត្តិការណ៍។



“ការសរស់សុំនូវមុននឹងការសរស់សុំ
ដែលបានបង្កើតឡើងដើម្បីសរស់សុំ
តាមរយៈប្រព័ន្ធថឹបបញ្ជី

K-Fold Cross Validation

FULL DATASET



[1]	Train	Train	Train	Train	Test
[2]	Train	Train	Train	Test	Train
[3]	Train	Train	Test	Train	Train
[4]	Train	Test	Train	Train	Train
[5]	Test	Train	Train	Train	Train

People normally
do 5, 10 K-Fold

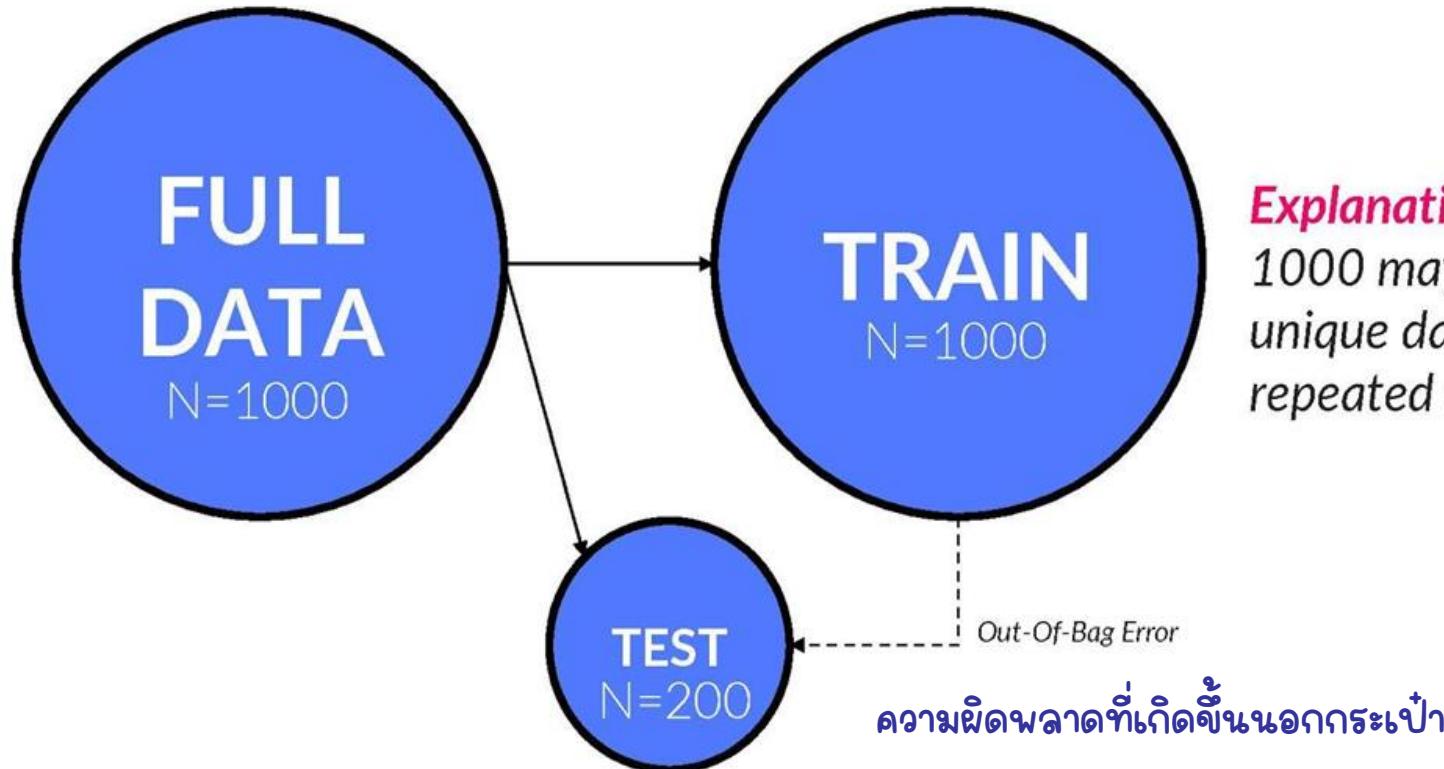
"Repeated k-fold cross-validation" คือ "การทำ Cross-validation แบบ k-fold ซ้ำ" โดยทำ k-fold cross-validation ซ้ำ n ครั้ง เพื่อเพิ่มความเสถียรในการประมาณประสีกติกา.

"K-fold cross-validation" คือ "การทำ Cross-validation แบบ k-fold" โดยแบ่งจุดอยู่ใน k ชุดเท่า ๆ กันและทำการทดสอบ k ครั้ง โดยในแต่ละครั้งจะใช้หนึ่งใน k ชุดเป็นชุดทดสอบและ k-1 ชุดคงเหลือเป็นชุดฝึก

Bootstrap Sampling

(aka. Random sampling with replacement)

ນີ້ອໍາທີ່ເຮັດວ່າ ການສຸ່ມຕົວຢ່າງແບບສຸ່ມທຶນ



"Bootstrap" ຕື່ອ "ເຫລົ້າໂບດ" ໂດຍກາຮຸ່ມຈຳຂ່ອງລຈາກຜູ້ດັ່ງ
ຈຳຂ່ອງລເດີມເພື່ອລັບໃຈຕົວຢ່າງທດສອບແລະໄສ່ໃນການປະມານຄ່າຕ່າງໆ
ຈາກປະສິກົນການ.

Explanation:

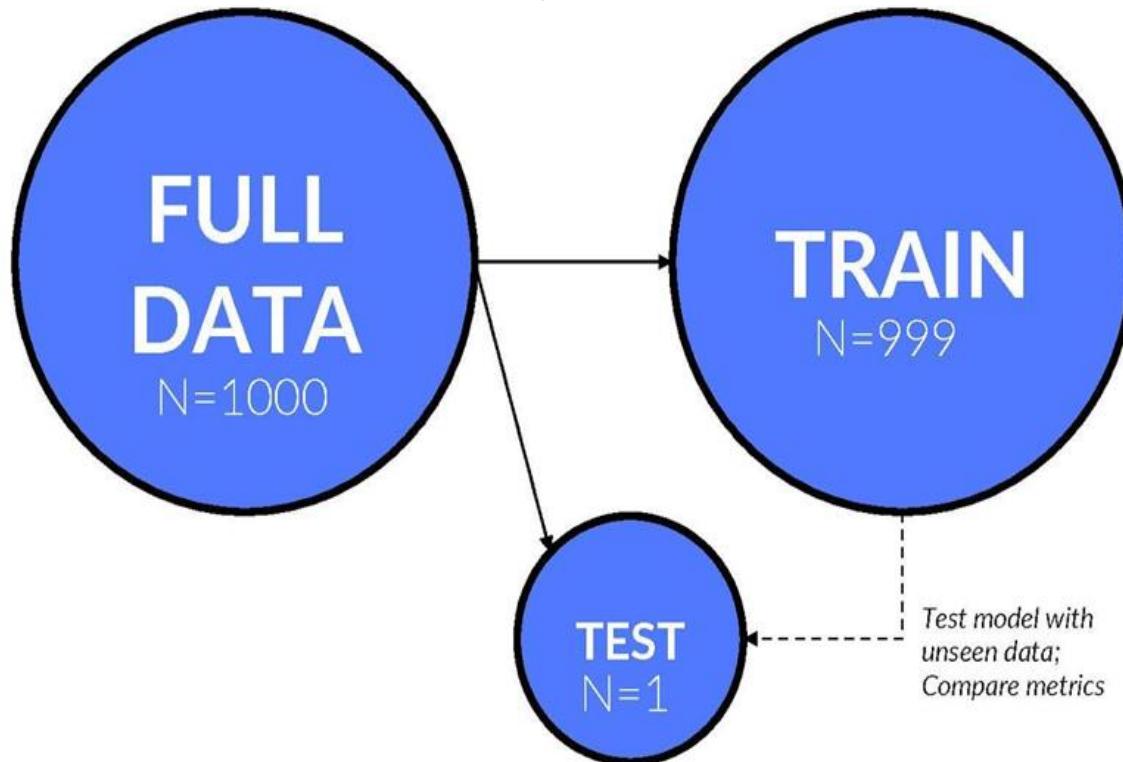
1000 may come from 800 unique data points + 200 repeated data points

ອົກບາຍ: 1000 ອາຈານາຈາກ 800
ຈຸດຈຳຂ່ອງລຫຼຸ້າໄໝ່ຫັກນີ້ + 200 ຈຸດ
ຈຳຂ່ອງລກໜັກນີ້

Leave One Out CV

This is the most expensive method – time consuming

นี่คือวิธีที่แพงที่สุด - ใช้เวลามาก



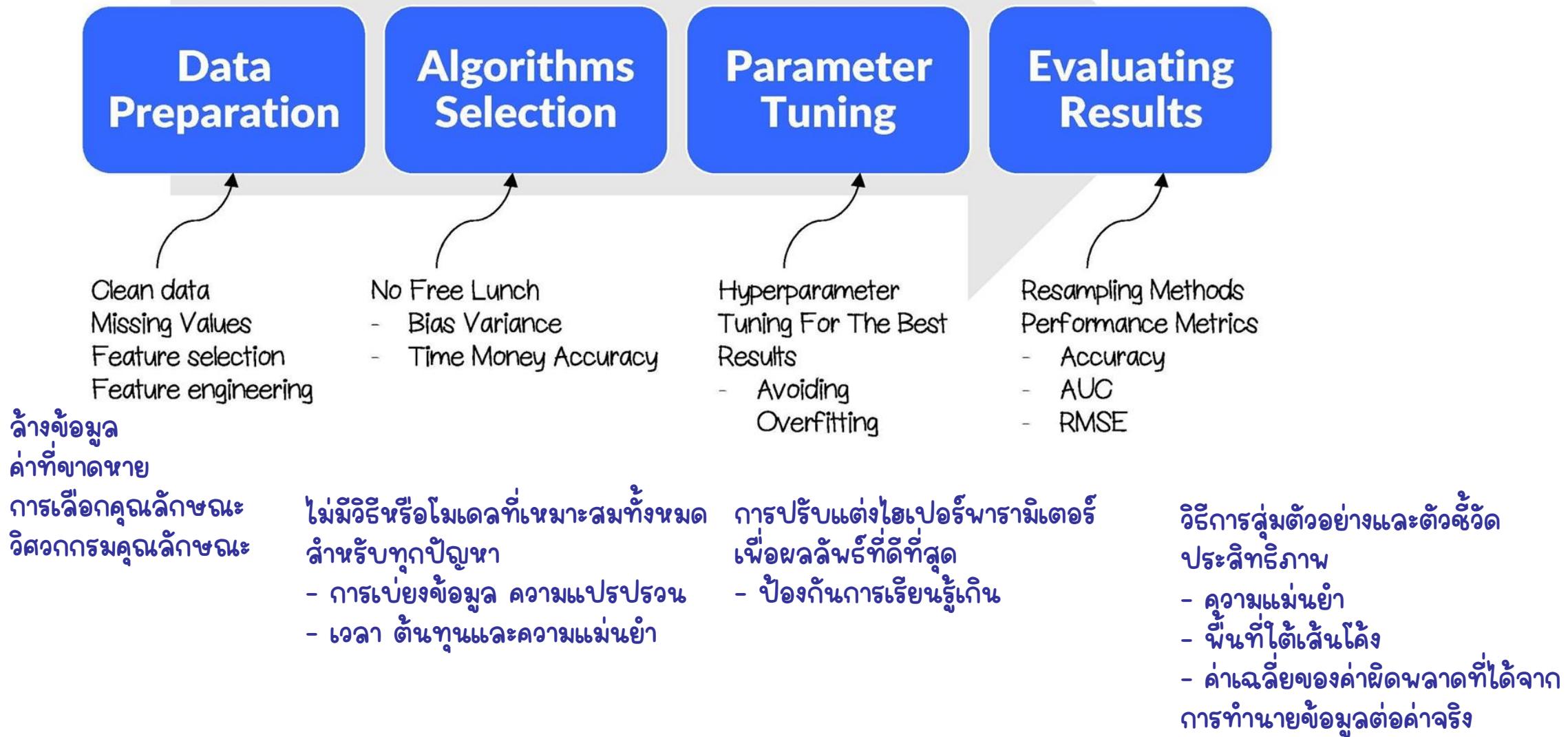
Explanation:

Leave 1 data point to test the model. For $n=1000$, this means we need to run model 1000 times

อธิบาย: ปล่อย 1 จุดข้อมูลไว้เพื่อทดสอบโดยเดล สำหรับ $n=1000$ นี่หมายความว่าเราต้องเรียกใช้โดยเดล 1000 ครั้งเพื่อทดสอบทุกรอบ ๆ โดยที่แต่ละรอบจะใช้ข้อมูลที่ไม่เหลือของจากฐานข้อมูลในการทดสอบครั้งถัดไป

"Leave-one-out cross-validation" คือ "การทำ Cross-validation แบบ Leave-one-out" โดยการทดสอบโดยใช้ข้อมูลที่เหลืออยู่ทั้งหมดนอกเหนือจุดเดียวที่นั่งในแต่ละรอบของการทดสอบ

#SIMPLE :)



SUMMARY

- Data preparation takes 70-80% of our time. การเตรียมข้อมูลใช้เวลา 70-80% ของเวลาทั้งหมดของเรา
 - "Garbage in, Garbage out" "ขยะเข้า, ขยะออก" หรือ "ข้อมูลที่ไม่ดีเข้าไปจะทำให้ผลลัพธ์ที่ไม่ดีออกมา"
 - Feature selection | Feature engineering | Missing Value การเลือกคุณลักษณะ | การพัฒนาคุณลักษณะ | ค่าที่หายไป
- No free lunch theorem ไม่มีอิฐหรือไม่เคลื่อนที่เหมาะสมสมทั้งหมดสำหรับทุกปัญหา
 - Bias vs. Variance Tradeoff การคำนว่าที่ระยะห่างความคลาดเคลื่อนและความแปรปรวน
 - Speed vs. Accuracy Tradeoff การคำนว่าที่ระยะห่างความเร็วและความแม่นยำ
- [Hyper]Parameter tuning to get the best out of our learning algorithms การปรับแต่งพารามิเตอร์เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดจากอัลกอริทึมการเรียนรู้ของเรา
- Resampling techniques used in ML research incl. train-test, k-fold CV, repeated k-fold CV, bootstrap, and LOOCV

เทคนิคการสุ่มตัวอย่างที่ใช้ในการวิจัยเครื่องเรียนรู้รวมถึงการแบ่ง train-test, k-fold cross-validation, repeated k-fold, bootstrap, และ leave-one-out cross-validation (LOOCV)

OKAY !! Let's recap a bit

มาพบกันสรุปเล็กน้อย

ML is **function approximation**

Machine Learning (ML) เป็นการประมาณฟังก์ชัน

Learn from past data to **predict future (unseen) data point**

เรียนรู้จากข้อมูลที่ผ่านมาเพื่อทำนายข้อมูลในอนาคต (ที่ยังไม่เคยเห็น)

The goal is **representation & generalization** (avoid overfitting)

เป้าหมายคือการแทนเล็กน้อยและการทำหัวไป (ป้องกันการเรียนรู้เกินไป)

Regression or Classification problems depend **target variable (y)**

ปัญหาการ Regression หรือ Classification จึงอยู่กับตัวแปรเป้าหมาย (y)

No Free Lunch: bias vs. variance tradeoff

ไม่มีวิธีใดหรือโมเดลที่เหมาะสมสมบูรณ์ทุกปัญหา และต้องมีการตัดสินใจเพื่อทำค้าง่าที่ระหว่างความคลาดเคลื่อนและความแม่นยำ

Algorithms can be tuned via **hyperparameter**

อัลกอริทึมสามารถปรับแต่งผ่านไฮเปอร์พารามิเตอร์ได้

K-Fold cross validation is popular resampling technique

การทำ Cross Validation แบบ k-fold เป็นเทคนิคการสุ่มตัวอย่างที่นิยมในการทำทางด้านความหลากหลายของข้อมูล (resampling technique)

Our model passes the test if **training vs. testing performances are similar** (not overfit)

โมเดลของเราผ่านการทดสอบถ้าประสิทธิภาพในกระบวนการฝึก (training) และการทดสอบ (testing) คล้ายกัน (ไม่มีการเรียนรู้เกินไป)

อัลกอริทึมที่ควรทราบ

ALGORITHMS YOU SHOULD KNOW

We cover 8 algorithms

Linear Regression	Regression
Logistic Regression	Classification
Decision Tree	Regression & Classification
Random Forest	(mainly) Classification
KNN	(mainly) Classification
Support Vector Machine	(mainly) Classification
Neural Networks	Regression & Classification
K-Means	Unsupervised Learning

Supervised
Learning

การเรียนรู้ของเครื่อง (Neural Network)
เป็นอัลกอริทึมพิเศษ ที่ทำงานคล้ายกับ
การเรียนรู้ด้วยการรีวิวน์ฟอร์สเมนต์
(Reinforcement Learning)

NN is special algorithm, it works
like reinforcement learning

Machine Learning 3(2-2-5), 2/2565

Unit 3

Supervised Learning

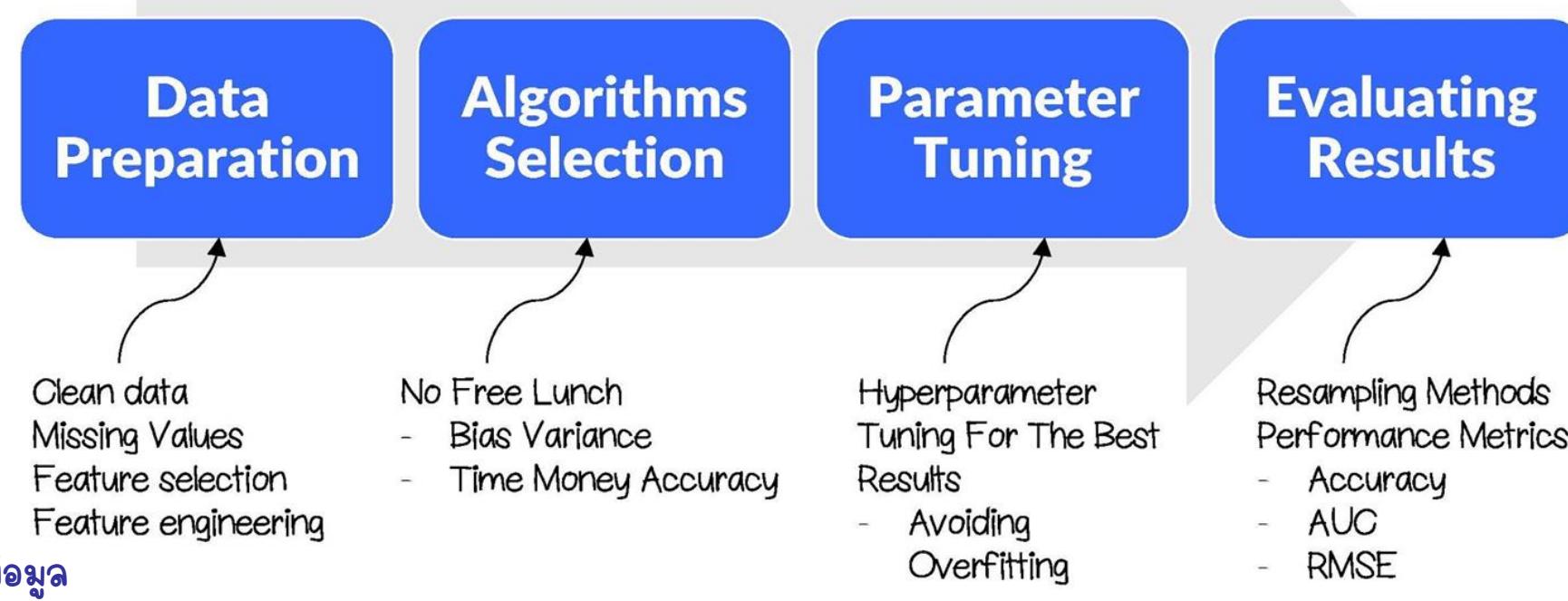
อ.อานันท์ ไม่ประดิษฐ์

LINEAR REGRESSION

Hello World in
Machine Learning

คือหนึ่งในเทคนิคทางสถิติและเครื่องมือเรียนรู้เครื่องของคณิตศาสตร์ที่ใช้ในการวิเคราะห์และพยากรณ์ความสัมพันธ์ระหว่างตัวแปรตัวนึง (independent variables) และตัวแปรตาม (dependent variable) ที่มีความลึกซึ้งหรือเปลี่ยนแปลงเส้น直線ใน การวิเคราะห์ผ่านกระบวนการของตัวแปรตัวนึงต่อตัวแปรตาม

Recall these steps



ລຳຈົ້ວມູລ

ຄ່າທີ່ຈະຫາຍ

ການເລືອດຄຸນເລັກຊະນະ
ວິສະວະກຮມຄຸນເລັກຊະນະ

ໄມ່ມີວິທີ່ໂມແດລທີ່ເໝາະສົມທັງໝົດ
ລຳຫັບຖຸກປຶກ
- ການເບ່ຍຈົ້ວມູລ ຄວາມແປປປວນ
- ເວລາ ຕິ່ນຖຸນແລະຄວາມແມ່ນຍໍາ

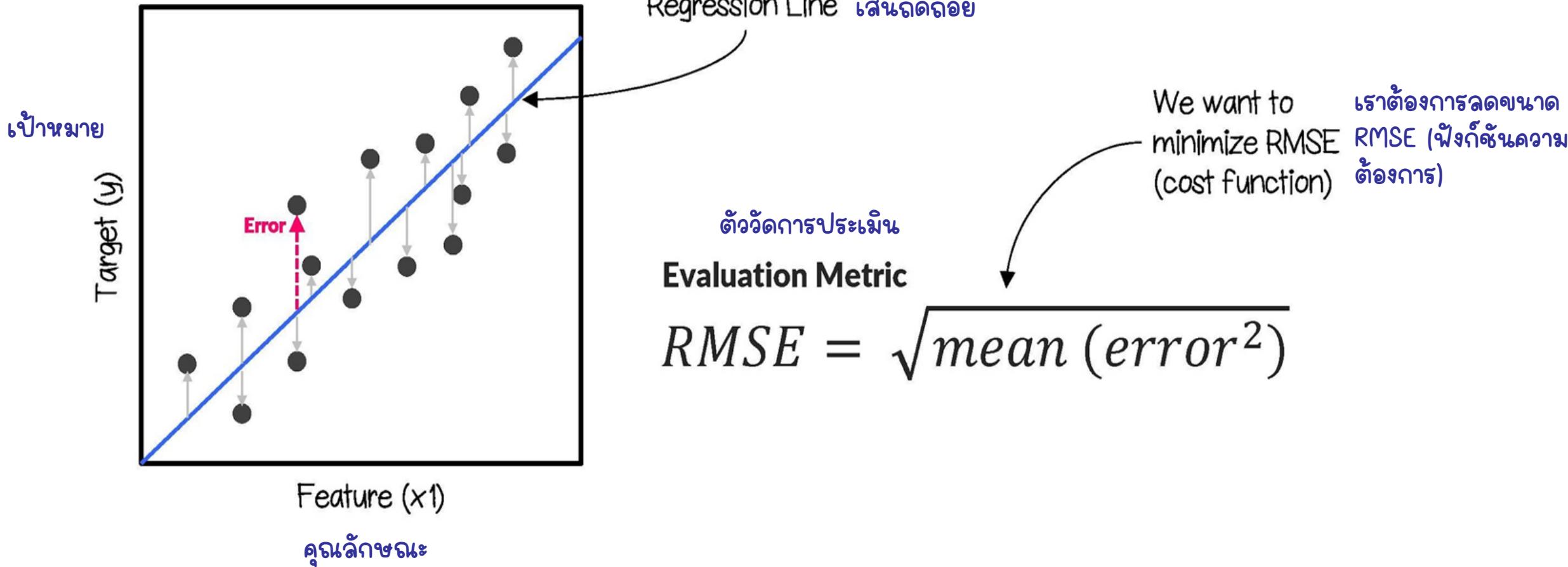
ການປັບແຕ່ງໄອເປົອ໌ພາຣາມີເຕອ້ຣ
ເພື່ອຜລລົງທີ່ດີທີ່ສູດ
- ປົ່ອກັນການເຮືອນຮູ້ເກີນ

ວິທີການສຸ່ມຕົວຢ່າງແລະຕັ້ງຈັ້ງ
ປະລິກິດການ
- ຄວາມແມ່ນຍໍາ
- ຈົ່າທີ່ໄຕເສັ້ນໂຄງ
- ຄ່າເລື່ອຍາອັນຄ່າຜິດພາດທີ່ໄດ້ຈາກ
ການທຳນາຍຈົ້ວມູລຕ່ອງຄ່າຈະກິງ

អំពីកម្រិតការសម្រាប់លេខរឹងទូទៅ: វិធីការវាជលេខទូទៅដើម្បីបង្កើតក្នុងក្រុមចំណែកដែលត្រូវបាន

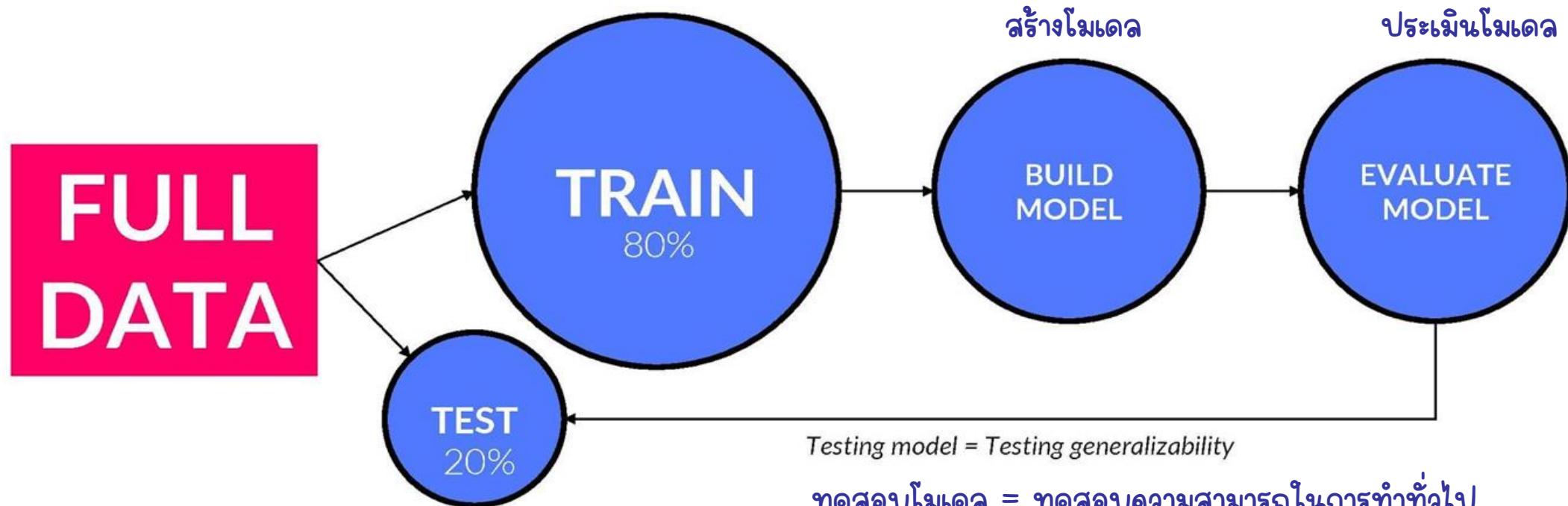
Algorithm explained:

How to draw a single straight line that fit best with given data points



ការបង្រៀនឯងតុលិភាគ - ការសរស់រាយប្រព័ន្ធប្រព័ន្ធផែនក្នុងការសរស់រាយ

Let's do a **train-test split** for linear regression problem



Advantages:

- *Easy to train* ง่ายต่อการฝึกอบรม
- *Always find the optimum solutions (least squared)*
- *Foundation of advanced algorithms like neural nets*

มีความสามารถในการค้นหาค่าตอบที่ดีที่สุด (น้อยที่สุด)

เป็นพื้นฐานของอัลกอริทึมที่ใช้ใน深度学习 Neural Networks

Disadvantages:

- *Only work for regression problems (continuous y)*
- *Full of assumptions (high bias algorithms)*

ใช้ได้เฉพาะลำดับปัญหาการถดถอย (ตัวแปรตามเป็นตัวเลข)

เต็มไปด้วยการสมมติฐาน (อัลกอริทึมที่มีความเอียงสูง)

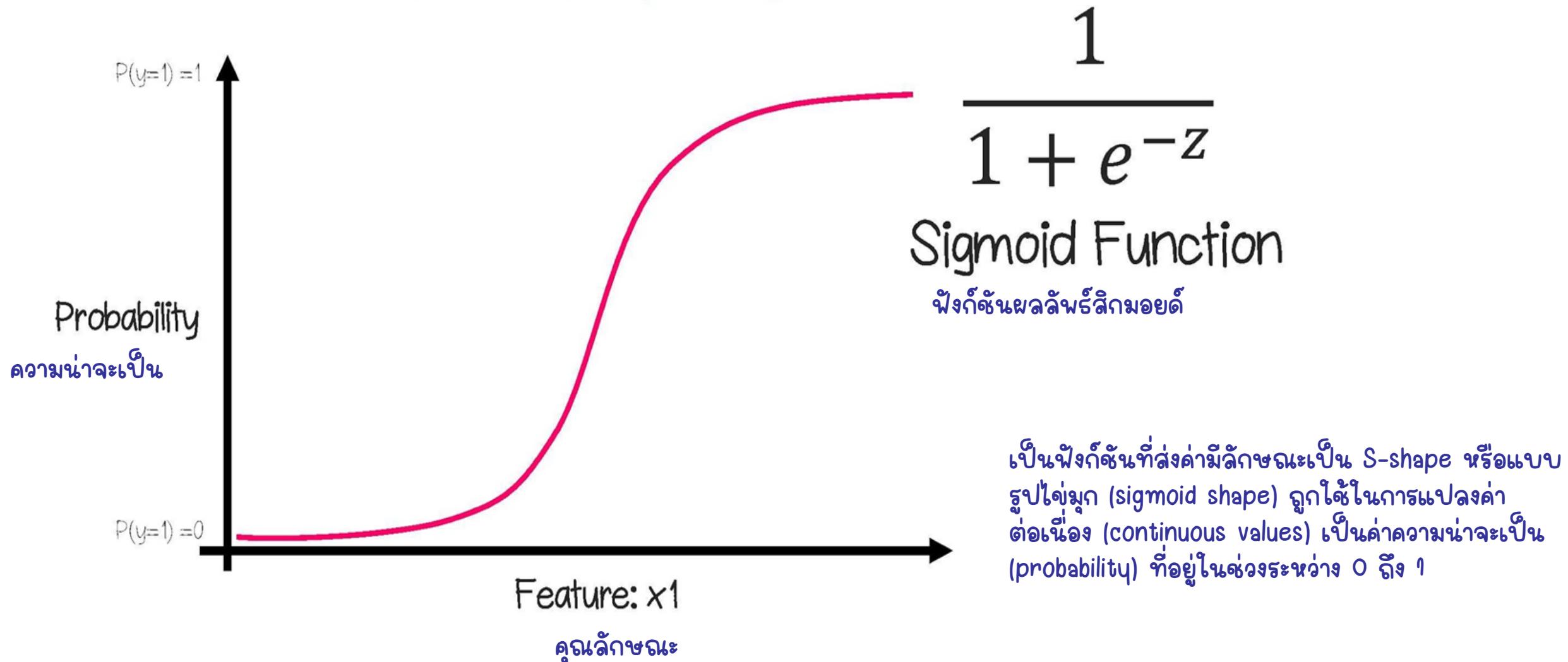
LOGISTIC REGRESSION

การจำแนกชั้นอยู่ในรูปแบบของคลาส (Class) ที่เป็นไปได้สองกลุ่ม โดยมีค่าเป็นตัวแปรตัวที่เป็นตัวเลขและตัวแปรตามที่มีค่าเป็นคลาส 0 หรือ 1 (ทำนายว่าอยู่ในกลุ่มไหน)

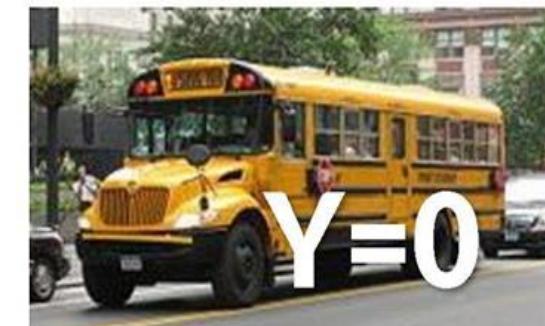
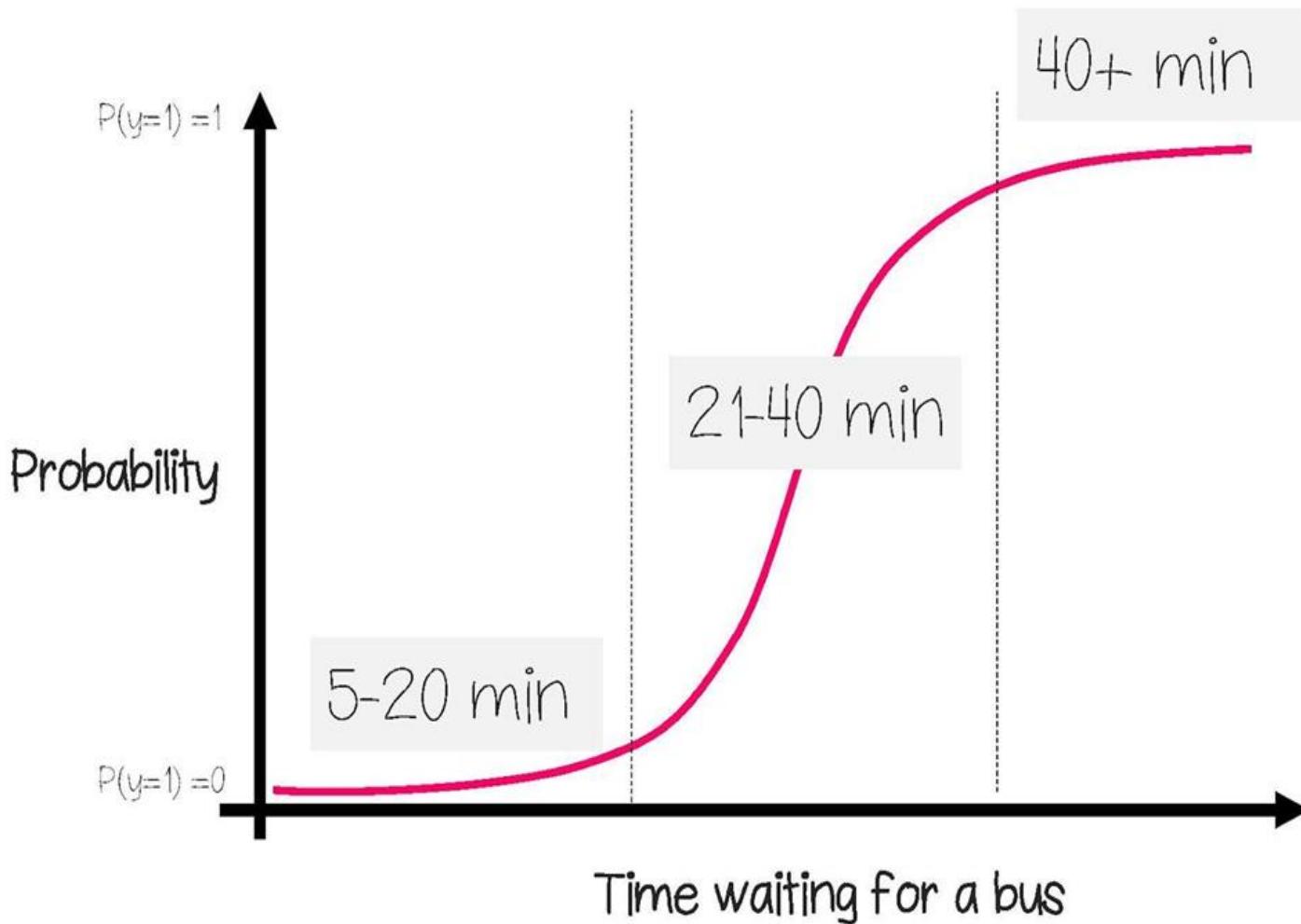
อัลกอริธึมที่อธิบาย: การจำแนกข้อมูลด้วยการคำนวณความน่าจะเป็นที่ $Y = 1$

Algorithm explained:

Classify data points by calculating the **probability** that $Y=1$



Should I get taxi or a bus?



ໂນໂລນີ້ຄໍາລັກກຳກາຮອດຍເສີ່ງເລື່ນມາກ (ສແນຕິກວາມລົ້ມພັນຮູ້ເສີ່ງເລື່ນຮະຫວ່າງ X ແລະ Y)

The model is very similar to linear regression

(assuming linear relationship between x and y)

$$z = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \rightarrow \text{then } \frac{1}{1+e^{-z}} \text{ to get probability scores}$$

ດາວໆນໍາ...ມາໃຊ້ເພື່ອໄຫ້ໄດ້ຄະແນນຄວາມໜ້າເປັນ

ID	X1 (minute)	X2 (crowd)	X3 (time)	X4 (weather)	P(Y=1) (get Taxi)	Prediction
1	50	Many	Friday	Good	.95	Yes
2	20	Many	Saturday	OK	.32	No
3	20	Few	Monday	OK	.31	No
4	25	Few	Friday	Bad	.85	Yes
5	30	Many	Friday	Good	.68	Yes
6	35	Few	Saturday	OK	.25	No
7	12	Many	Monday	Good	.52	Yes
8	5	Few	Saturday	OK	.12	No
9	10	Many	Friday	Bad	.20	No
10	65	Few	Friday	Good	.98	Yes



We set threshold = 0.5
If $p(y=1) \geq 0.5$, then we predicted YES get taxi!

ເຮັດວຽກຄ່າຄ່າມາຕຽບຮູ້ທີ່ 0.5 ຊໍາ $p(y=1) \geq 0.5$
ແລ້ວຮາກໍານາຍ່ອງຈຳ 'ໃໝ່' ຂະໜາດີຮັບຮອດເທິກສີ'

Advantages:

- One of the most powerful algorithms to date หนึ่งในอัลกอริทึมที่มีประสิทธิภาพมากที่สุดในปัจจุบัน
- Easy to train ง่ายต่อการฝึกอบรม
- Can be developed to create complex decision boundaries (e.g. x^2 , y^2) สามารถพัฒนาเพื่อสร้างขอบเขตการตัดสินใจที่ซับซ้อน

Disadvantages:

- Also a lot of assumption e.g. no outlier, no noise
- Logistic regression is also a linear algorithm
- Prone to multicollinearity (check correlation first)

(sigmoid function transform an output into non-linear)

นอกจากนี้ยังมีการสมมติฐานหมาย เช่น ไม่มีข้อมูลผิดปกติ, ไม่มีเสียงรบกวน
โลจิสติกเรสเซ็นต์เป็นอัลกอริทึมเชิงเส้น

มีความเสี่ยงต่อ multicollinearity (ตรวจสอบความสัมพันธ์กัน)

multicollinearity เกิดขึ้นเมื่อตัวแปรตัวเดียวกันมีความลักษณะที่คล้ายคลึงกันอย่างมาก
ซึ่งอาจทำให้การวิเคราะห์ผลลัพธ์ของแต่ละตัวแปรที่ไม่แน่นอนและทำให้การคำนวณเป็นไปได้ยาก

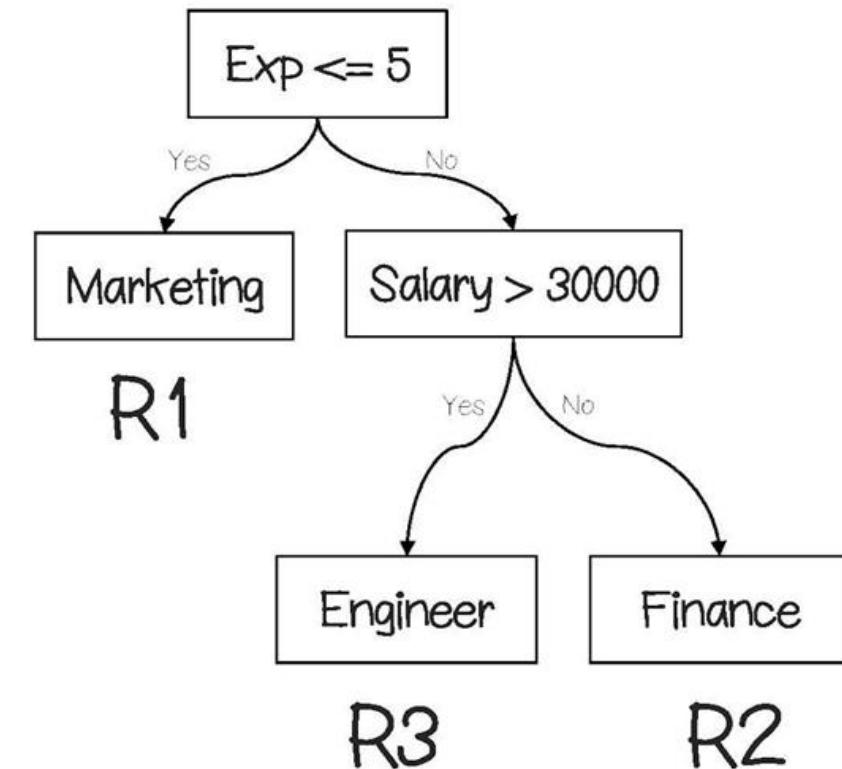
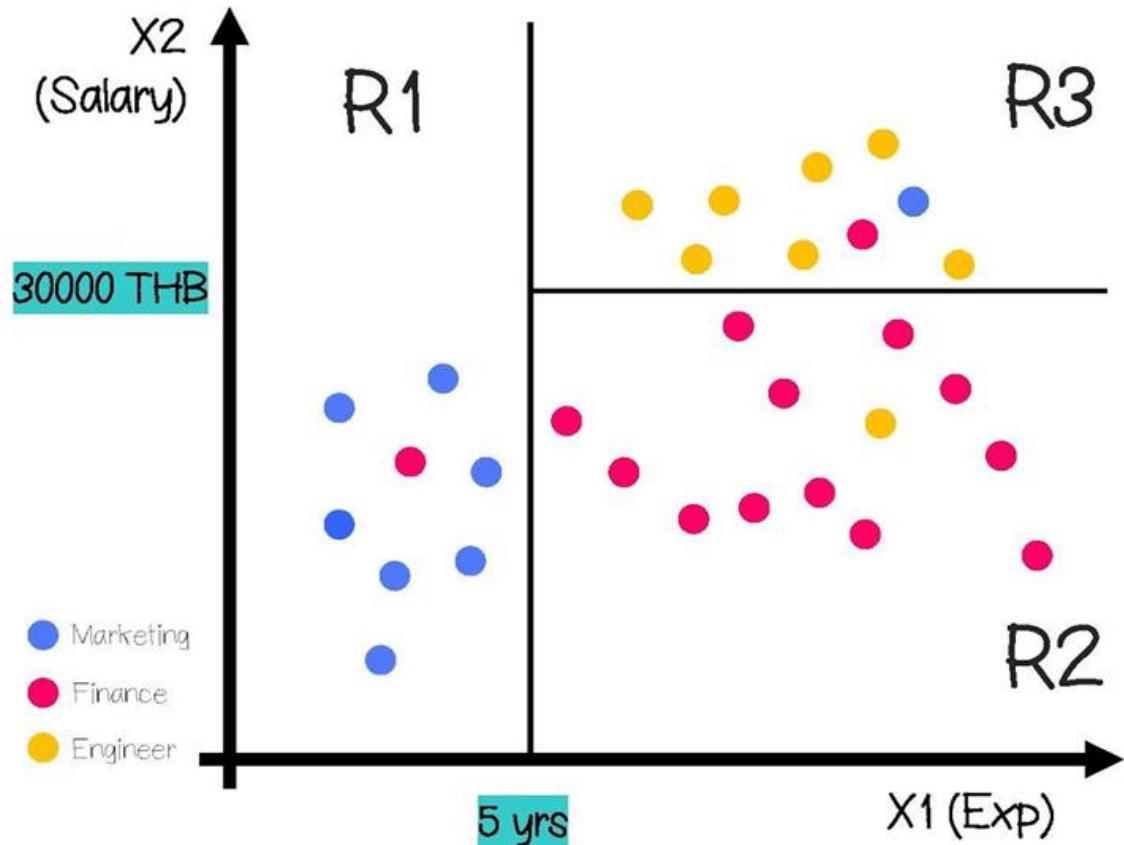
"ຕົ້ນໄມ້ຕັດສິນໄຈ" ຄືອອັລກອຮູມໃຫຍ່ໃນກາຣເຮັຍນີ້ແຄຣ່ອງຈັກທີ່ໃຈໂຄຮງສ້າງວອນຕົ້ນໄມ້ເພື່ອທຳນາຍພລລິພຣ໌ທີ່ອຕັດສິນໄຈ

DECISION TREE

ອັດກອຣິຖິມທີ່ອົບາຍໍ: ແບ່ງພື້ນທີ່ເປັນລູ້ມີຄາຕີ່ກໍານາຍຈົ່ວລຸດ (ຄລາສ)

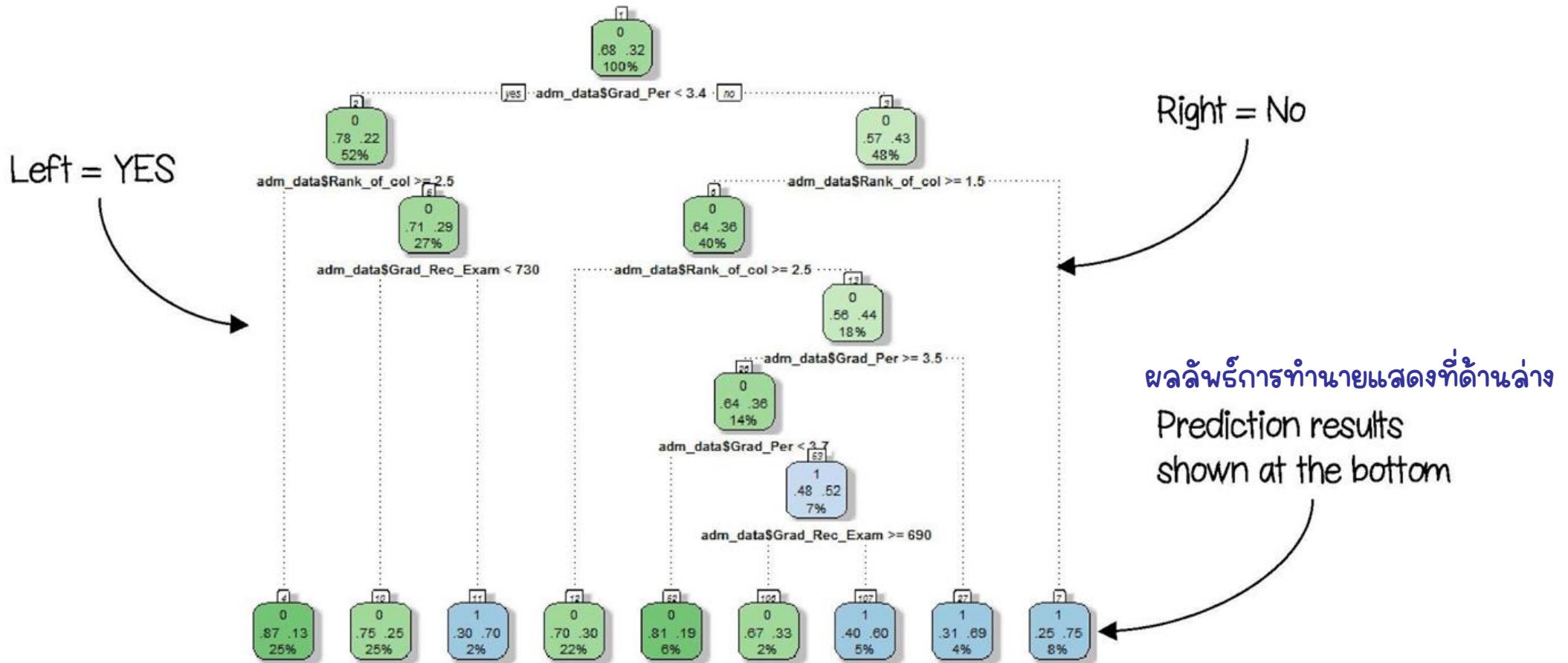
Algorithm explained:

Divide space into regions that best predict data points (classes)



ຕົ້ນໄໝ້ຕັດສິນໃຈສາມາຮຽນໄດ້ຈ່າຍໆ ດ້ວຍການໃໝ່ແຜນກາງ

Decision Tree is easily presented using diagram



Advantages:

- Easy to train ง่ายต่อการฝึกอบรม
- Easy to interpret results
- Widely used in many disciplines
ผลลัพธ์ที่อ่านและทำความเข้าใจได้ง่าย
ใช้กันอย่างแพร่หลายในหลายวิชา

The main reason why เหตุผลหลักที่คนนิยมใช้ตัดสินใจ
people use decision tree

Disadvantages:

- Prone to overfitting: Greedy algorithms

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

ข้อความเสียด้วยต่อการเรียนรู้เกินไป: อัลกอริทึมโล่ห์ (Greedy algorithm)

Locally optimal decision
are made at each node,
but can't guarantee globally
optimal decision trees

กฎการตัดสินใจที่เหมาะสมสมทำ
ลิ่นที่แต่ละโหนดตัดสินใจ
แบบห้องถูน แต่ไม่สามารถร
รับประคัญญาจะได้ตัวเอง
ตัดสินใจที่เหมาะสมสมทำหะนุด

Required Packages:

ແພື່ນເກມທີ່ຕ້ອງການ

library(**caret**) – train & test models ຜິກຜະນະລະກາດສອບໂນມເລ

library(**caTools**) – split data into training & testing sets ແບ່ງຄໍ້ມູນເປົ້າໃຈແລະຫຼຸດການສອບ

library(**rpart**) – train decision tree (CART): recursive partitioning and

regression tree

CART

ຝັກຕິ່ນໄມ້ຕັດສິນໃຈ: ດາວໂຫຼດທີ່ຈະເປັນຮັບອະນຸຍາດ

```
method = 'rpart'
```

Type: Regression, Classification

Tuning parameters:

- cp (Complexity Parameter)

Required packages: rpart

A model-specific variable importance metric is available.

Cp: complexity parameter
to penalize larger trees

ຈໍາຍຕ່ອດກາຮັກ

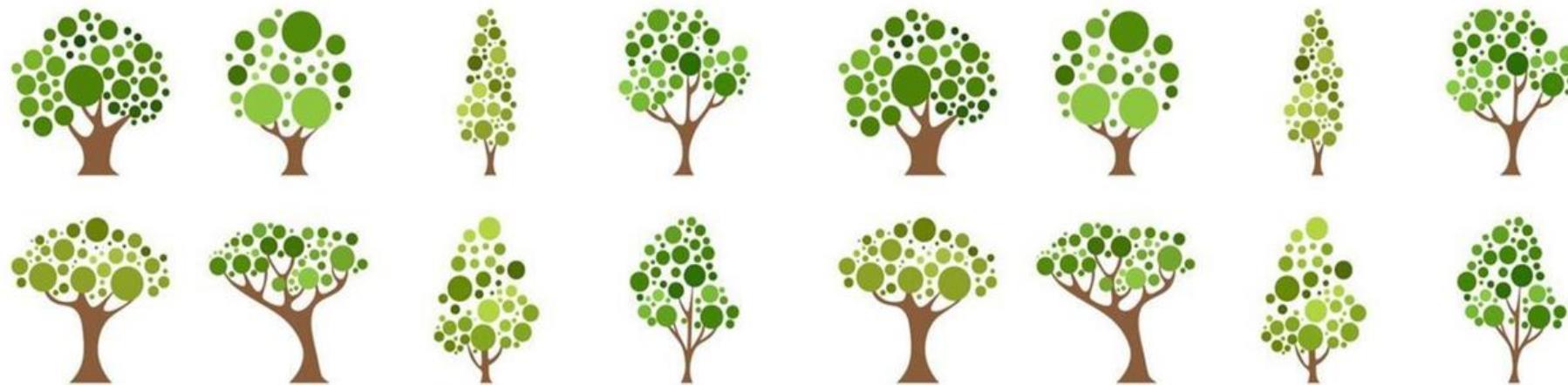


RANDOM FOREST

อัลกอริทึมที่อธิบาย: การเพาะปลูกพันต้นไม้และหาค่าเฉลี่ยของผลการคำนวณของทั้งหมด

Algorithm explained:

Grow thousand trees and average their prediction results



Grow **uncorrelated** 1000 trees

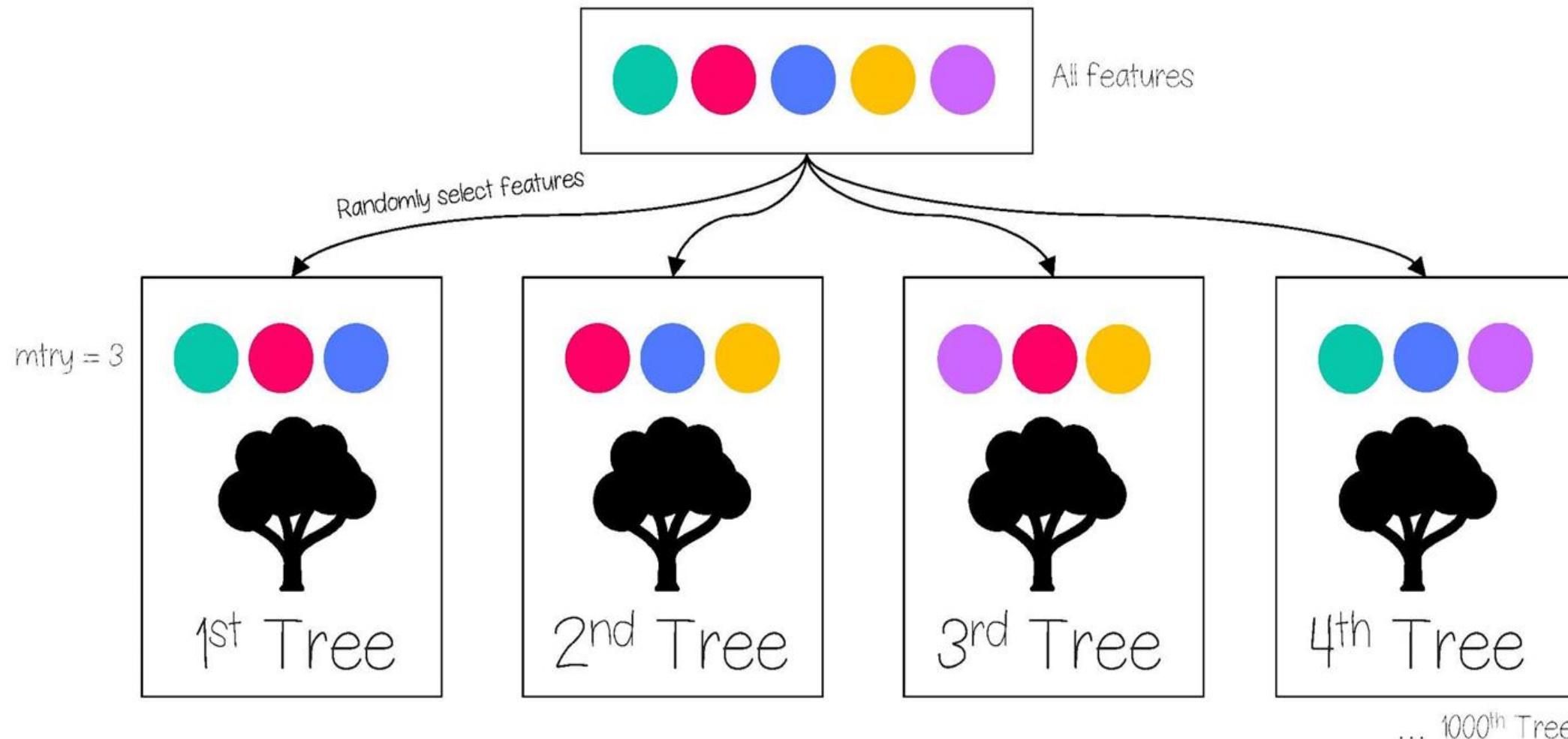
เพาะปลูก 1000 ต้นไม้ที่ไม่มีความลับพันธุ์กัน

ເຮັດວາປະລູກ 100 ຕິ່ນໄສ້ອ່າງໄຮ?

ຄວາມສຸ່ມຄົວເພື່ອນວອນຄຸນ, ພາຣາມີເຕອຣ໌ໃໝ່ເພວົ້າເຕອຣ໌ຄົວ mtry

How do we grow 1000 trees?

Randomness is your friend, hyperparameter is **mtry**



ປ័ត៌មានទូទៅនៃអ្នកជនដែលធានាយេត្តិយោចនៅពេលបង្កើតម៉ោងទីផ្សារ

Wisdom of the crowds

Find the average prediction results of all grown trees



... 1000th Tree

Advantages:

ຈົດ

- *Amongst the best learners in the world*
- *Easy to train (Improvement from Decision Tree)*

ອູ້ໃນຮະດັບອົງຜູ້ເຮືອນທີ່ລື້ອໍ້ສຸດໃນໂລກ

ຈ່າຍຕ່ອກາຮືກອບຮມ (ພິກາຮັບແນວຈາກຕົ້ນໄໝເຕັດລື້ອນໄຈ)



Disadvantages:

ຈົດເລີຍ

- *Can be slow to train for large dataset*
- *Black Box Algorithm (not easy to explain)*

ກາຮືກອບຮມລາມາຮຽນສໍາໜັບຜູ້ຈົດຈ້ອນມູລທີ່ໃໝ່

ອຳລັກອຣີທີ່ມີກລ່ອດຳ (ໄໝຈ່າຍຕ່ອກາຮືກບາຍ)

ເພື່ອນບ້ານທີ່ໄດ້ລຶກທີ່ສຸດ K ຄນ

KNN (K-NEAREST NEIGHBOR)

ຄໍອອລາດອຣິກິມທີ່ໃຊ້ໃນການຈຳແນກຂໍ້ອມູນແລະກຳນາຍໄ ໂດຍກາຮຽດູ້ຄົງຂໍ້ອມູນທີ່ໄດ້ທີ່ສຸດ K ຕ້ອອຍ່າງຈາກຫຼຸດຂໍ້ອມູນເພື່ອຕັດສິນໃຈປະເກາຫ້ອື່ອຄ່າກຳນາຍຈອງຕ້ອອຍ່າງໃໝ່ ຖ້າ ໂດຍໃຊ້ວິທີກາຣຄໍານວກຮະຢະໜ່າງ (distance) ຮະໜວ່າງຕ້ອອຍ່າງທີ່ຕ້ອງການຈຳແນກດັບຕ້ອອຍ່າງໃນຫຼຸດຂໍ້ອມູນ

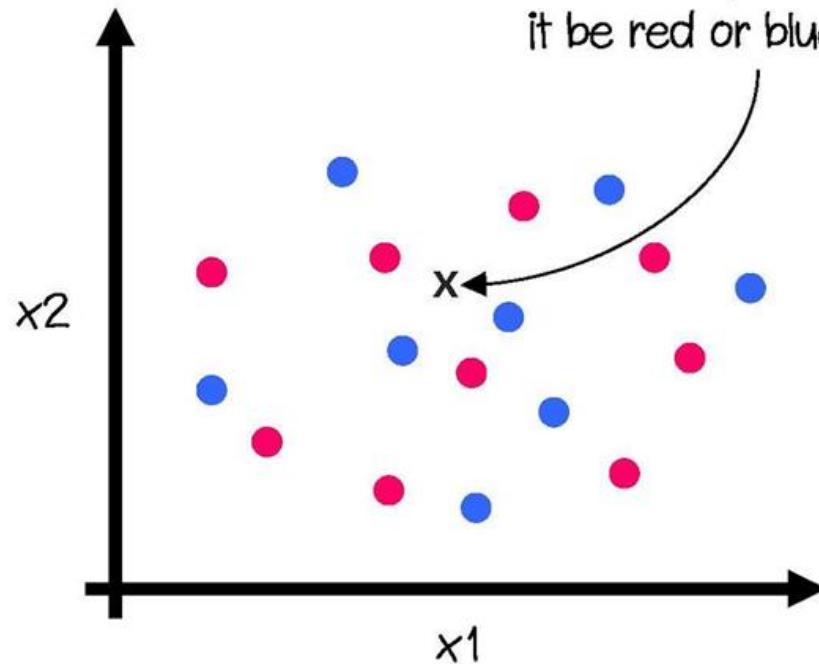
อัลกอริทึมที่เรียกว่า: กำหนดอยู่จุดใดใน簇 ตามระยะทางจากเพื่อนบ้านของมัน

Algorithm explained:

Predict new data according to the distance from its neighbors

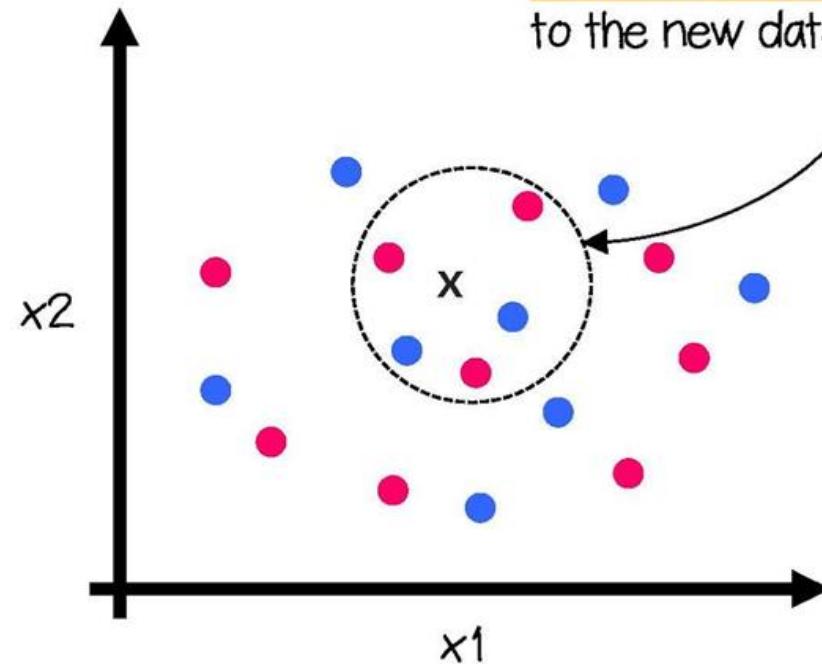
จุดข้อมูลใหม่, ควรที่จะเป็นสีแดงหรือสีน้ำเงิน

New data point, should
it be red or blue?



" k " คือจำนวนของเพื่อนบ้าน
ที่ใกล้ที่สุดกับจุดข้อมูลใหม่
($k=5$)

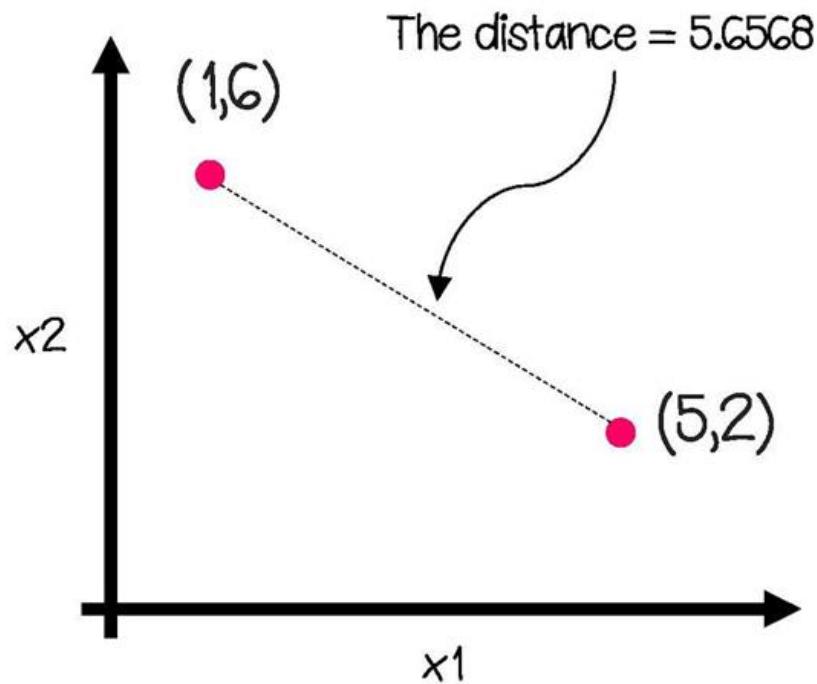
**k is number of closest neighbors
to the new data point ($k=5$)**



ระยะห่างยูคลิดเดียน: วิธีการคำนวณระยะห่างที่ได้รับความนิยมที่สุด

Euclidean Distance:

The most popular distance calculation method



$$d = \sqrt{\sum (x_i - y_i)^2}$$
$$d = \sqrt{(1 - 5)^2 + (6 - 2)^2}$$
$$d = \sqrt{16 + 16}$$

$d = 5.656854$

Very simple
to compute

คำนวณได้ง่ายมาก

Another Example

Consider the nutrient dataset provided with the `flexclust` package. The dataset contains measurements on the nutrients of 27 types of meat, fish, and fowl. The first few observations are given by

```
> data(nutrient, package="flexclust")
> head(nutrient, 4)
```

		energy	protein	fat	calcium	iron
BEEF	BRAISED	340	20	28	9	2.6
HAMBURGER		245	21	17	9	2.7
BEEF	ROAST	420	15	39	7	2.0
BEEF	STEAK	375	19	32	9	2.6

We can calculate
many dimensions
(more than two)

"ความสามารถคำนวณ
หลากหลายมิติ (มากกว่า
สอง)"

and the Euclidean distance between the first two (beef braised and hamburger) is

$$d = \sqrt{(340 - 245)^2 + (20 - 21)^2 + (28 - 17)^2 + (9 - 9)^2 + (2.6 - 2.7)^2} = 95.64$$

Advantages:

- *Very simple* ง่ายมาก
- *Easy to train (remember all data)* ง่ายต่อการฝึกอบรม (จำจ้อมูลทั้งหมด)

Disadvantages:

- *Worst for large dataset* ไม่เหมาะสมสำหรับชุดข้อมูลที่ใหญ่
- *Not good with high dimensional data (many features)*
- *Don't work well with categorical data*

ไม่ดีกับข้อมูลมิติสูง (มีคุณลักษณะมากหลาย)

ทำงานไม่ดีกับข้อมูลแบบหมวดหมู่

Example Code:

```
5 ## Load library  
6 library(caret)  
7  
8 ## train model  
9 ## define trControl  
10 my_trcontrol = trainControl(method = "cv",  
11                               number = 3,  
12                               verboseIter = TRUE,  
13                               search = "random")  
14  
15 knn_model <- train(Species ~ . , data = m,  
16                      method = "knn",  
17                      metric = "Accuracy",  
18                      tuneLength = 10,  
19                      trControl = my_trcontrol)
```

กำหนดเทคนิคของเรา (ในการฝึกโมเดล)

Define our technique
(to train the model)

Train the model using
random search +
tuneLength = 10

ฝึกโมเดลโดยใช้การค้นหาแบบสุ่ม + ปรับค่า
Tune = 10

วิธีการในการเรียนรู้ของเครื่องที่ใช้ในการจำแนกรูปภาพและการจำแนกรูปแบบในอ้อมูล

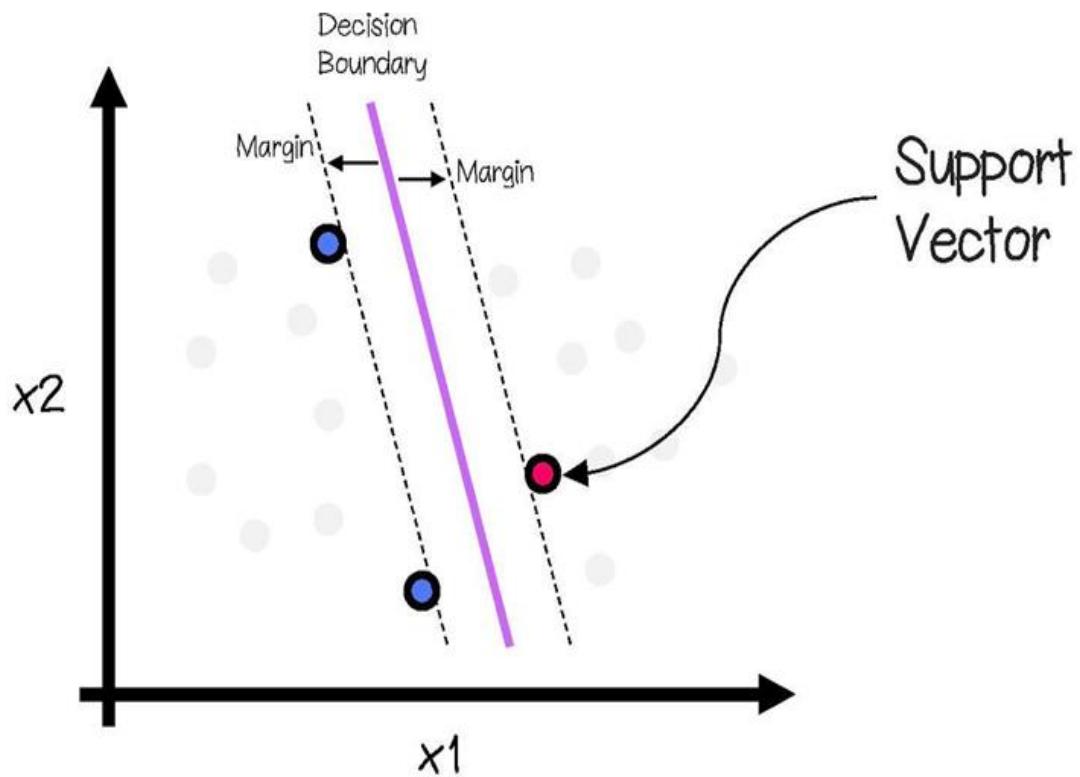
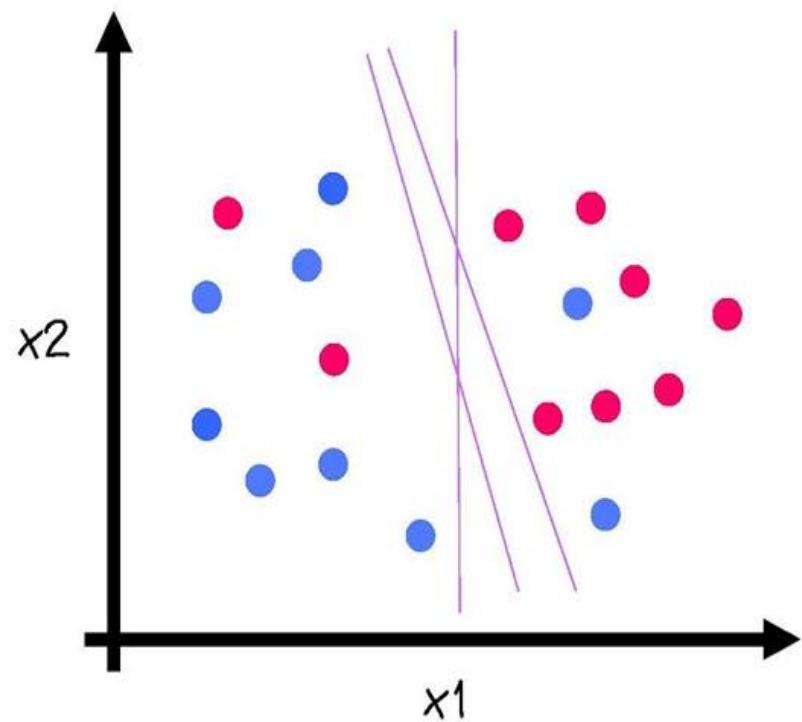
SUPPORT VECTOR MACHINE

มีการใช้แนวคิดของการคืนนาเลสันแบบ (hyperplane) ที่มีความซับซ้อนระหว่างกลุ่มข้อมูลที่แตกต่างกัน. ในกรณีที่ข้อมูลไม่สามารถแบ่งแยกได้ด้วยเส้นตรงเดียว, SVM สามารถใช้เครื่องหมาย (markers) เพิ่มเติ่มหรือทางคณิตศาสตร์ในการทำให้มีความยืดหยุ่น (flexibility) ในการจำแนกรูปแบบ

អំពីការស្នើសុំរបៀបនេះទៅកាន់ការត្រួតពិនិត្យ

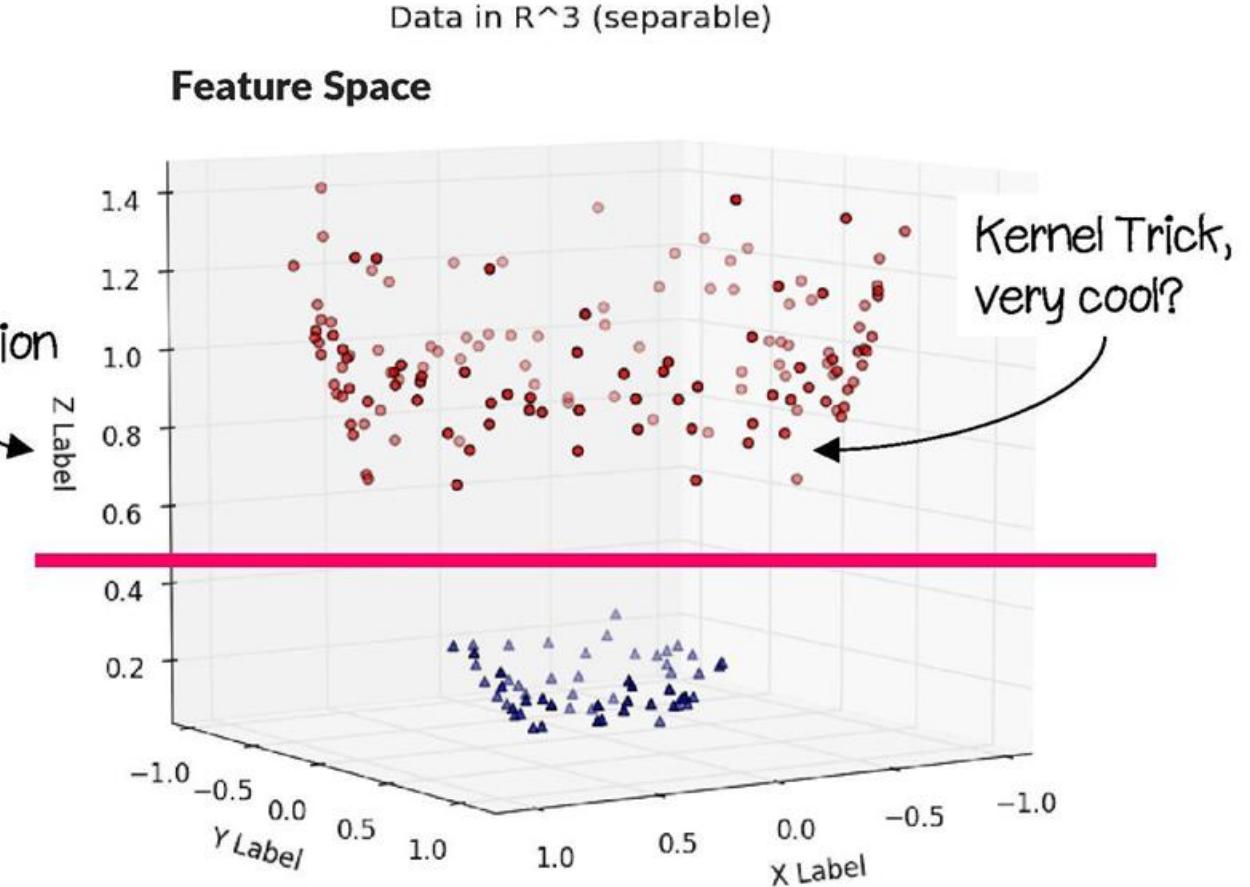
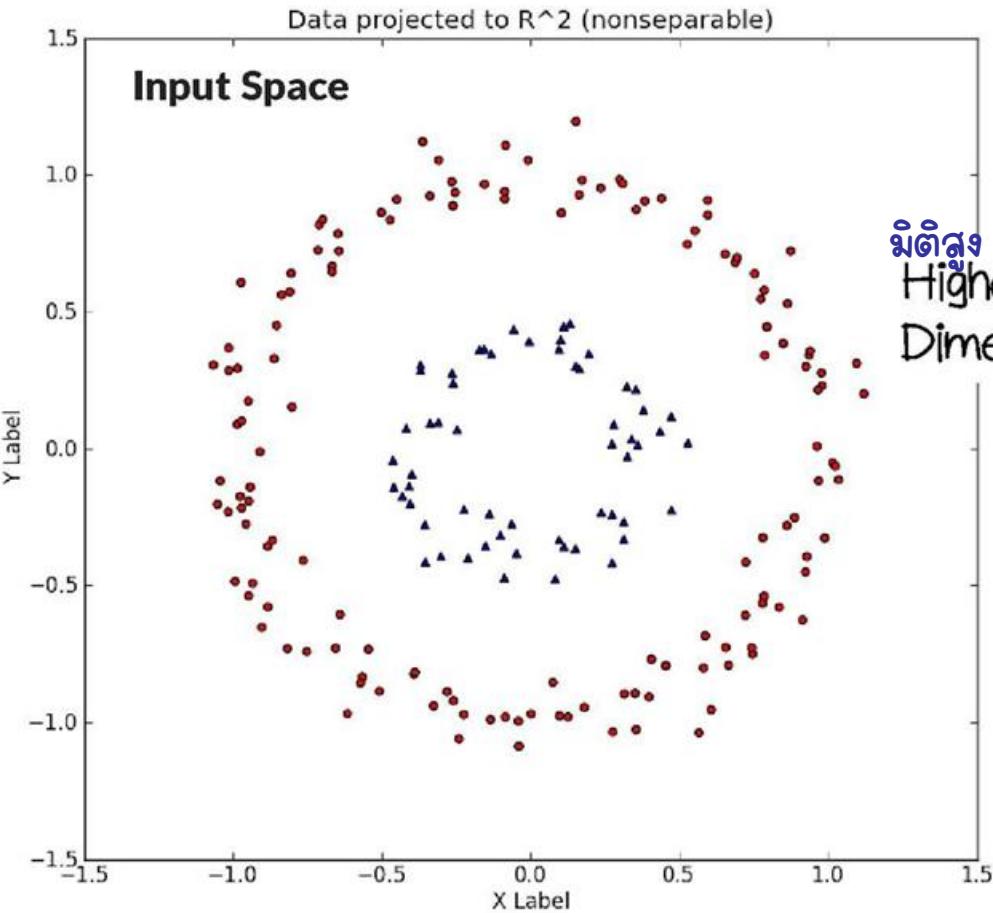
Algorithm explained:

We want to maximize the **margin** from decision boundary



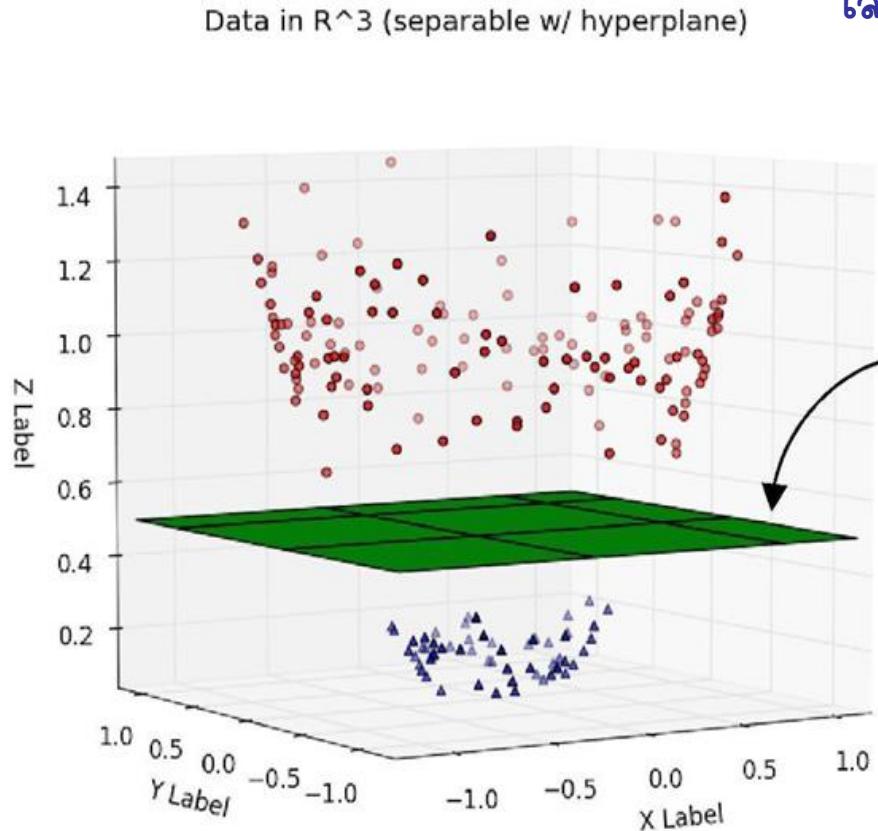
ເຮັດວຽກສູງສຸດຮະຍະໜ່າງຈາກເລີ່ມຕົ້ນຂອບກາຣຕິດລິນໄຈ

But can you divide this with a single straight line?



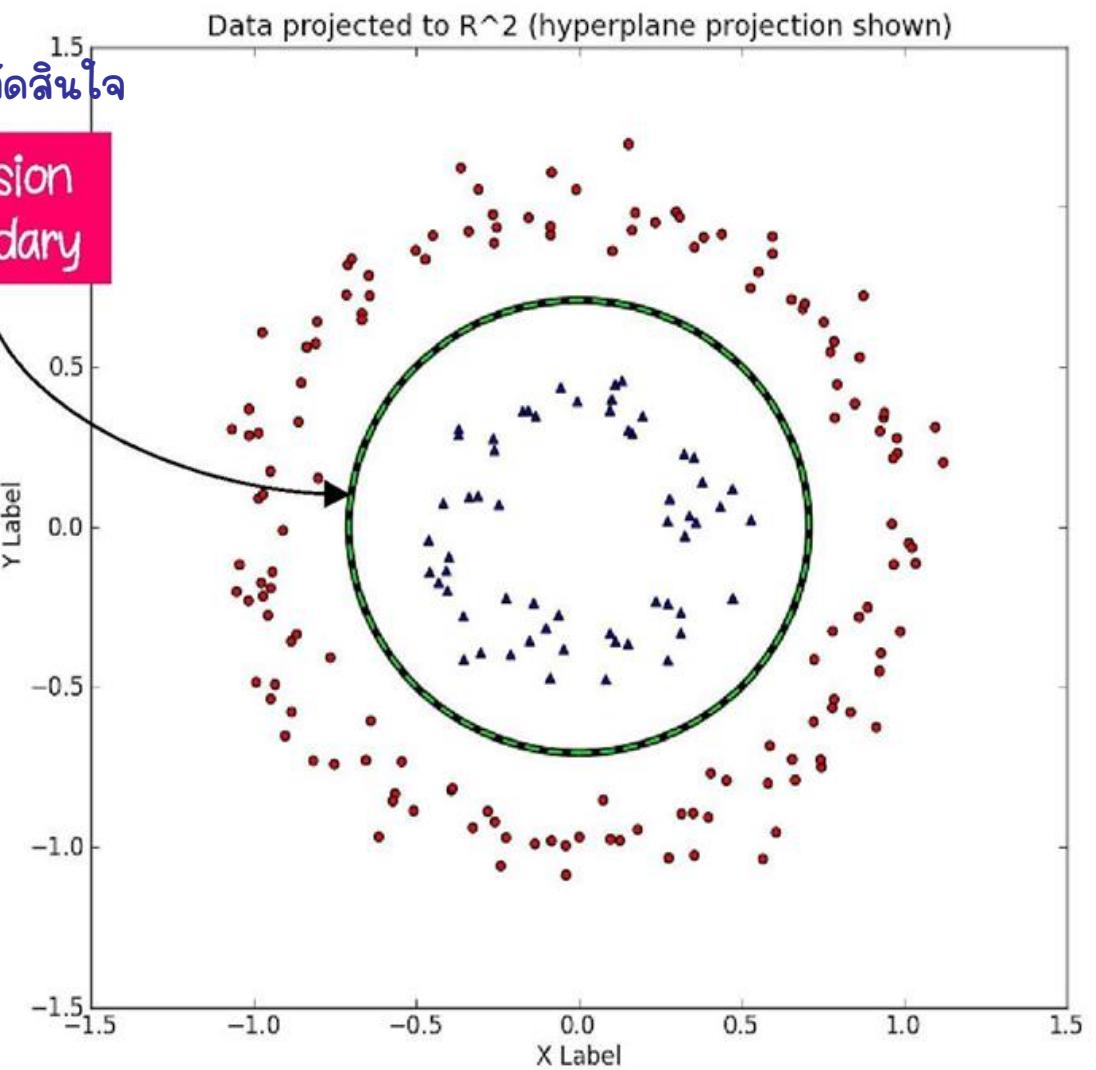
Impressive Result!!

ผลลัพธ์ที่น่าประทับใจ



เลียนแบบการตัดสินใจ

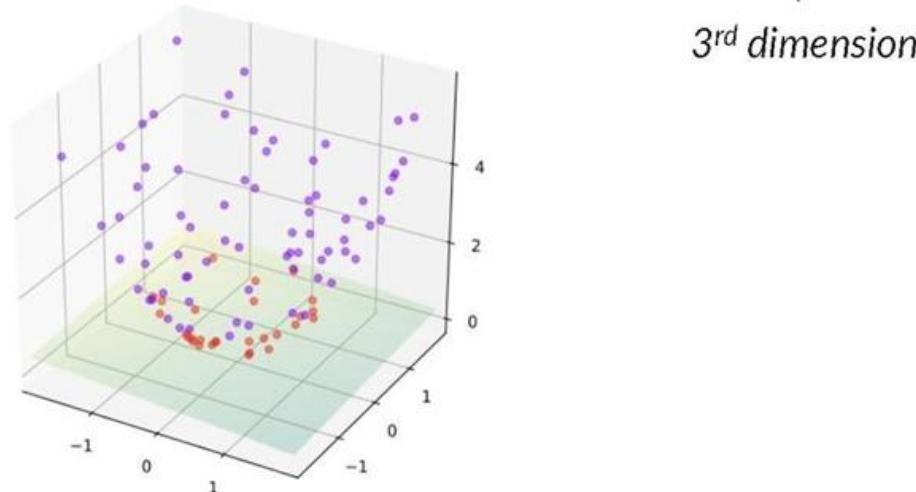
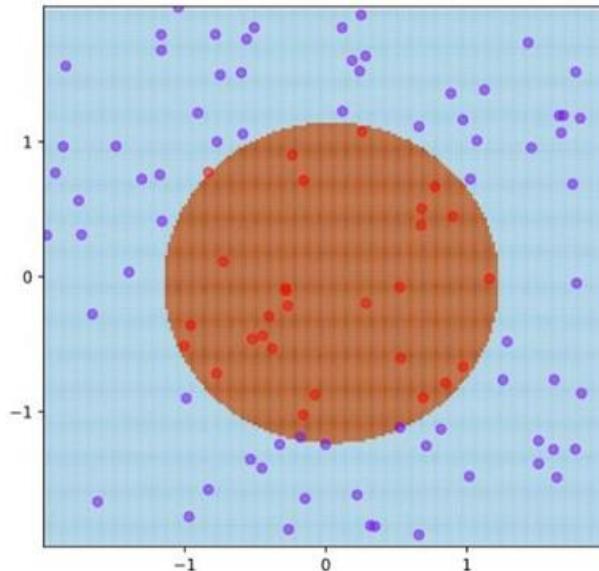
Decision
Boundary



ມາລຳຮວຈໃນນຸ່ມມອງລັກງາເກີຍອັກັບເທັກນິດ Kernel

Let's look closely at kernel trick

- we have (x,y) coordinates ເຮັມສືພັດ (x, y)
- we project a new higher dimension: $(x, y, x^2 + y^2)$
ເຮັມການໂປຣແກຣມເພື່ອເຈົ້າສູ່ມືຕິສູງໃໝ່



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

Advantages: จ้อดี

- Work with Regression | Classification problems
- Superior Performance ประสิทธิภาพที่ดีเยี่ยม
- Having lots of available kernel tricks

<https://topepo.github.io/caret/train-models-by-tag.html#support-vector-machines>

มีจำนวนหลาย kernel tricks ที่สามารถใช้งานได้มาก

ใช้งานกับปัญหาการทำนาย
(regression) หรือการจัดคลุ่ม
(classification)

Disadvantages: จ้อเสีย

- Expensive แพง
- It's lazy algorithm (remember data like KNN)

เป็นอัลกอริทึมที่ 'เล่นเป็นมาก' (lazy algorithm) (เปรียบเสมือนกับ KNN) โดยที่มีความจำต้องให้คำจำกัดของจ้อมูล

ແພັດເກຈໍ້ຕ້ອງການ

Required Packages:

library(**caret**) – train & test models ໜຶກຜົນແລະທຸດສອບໂນມເຈລ

library(**caTools**) – split data into training & testing sets ແບ່ງຈຳຂູ້ລເປີນຫຼຸດຜົນແລະທຸດສອບ

library(**e1071**) – load some algorithms e.g. SVM ໂໜລດວັດກອຣິກິ່ມບາງວັນ

Support Vector Machines with Linear Kernel

```
method = 'svmLinear2'
```

Type: Regression, Classification

Tuning parameters:

- cost (Cost)

Required packages: e1071

การสร้างโมเดลที่สามารถปรับค่าให้ดีขึ้นตามการฝึกอบรมเพื่อให้ระบบสามารถทำนายผลลัพธ์หรือจำแนกจังหวะในลักษณะที่ไม่เคยเจอนมาก่อน

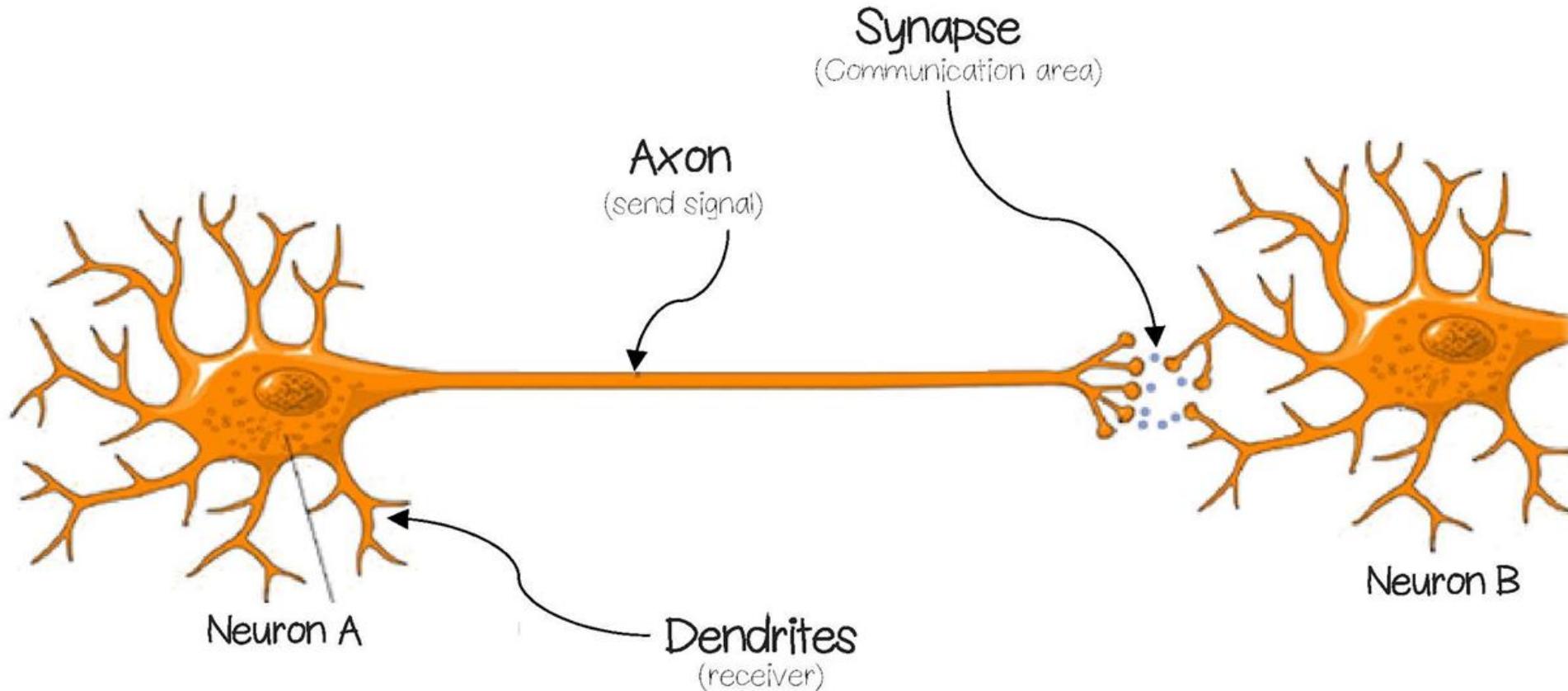
NEURAL NETWORK



Neuron

វិធាន់ថែល់សមរុបនគ្គល់សេវាសារ

How your brain cells communicate



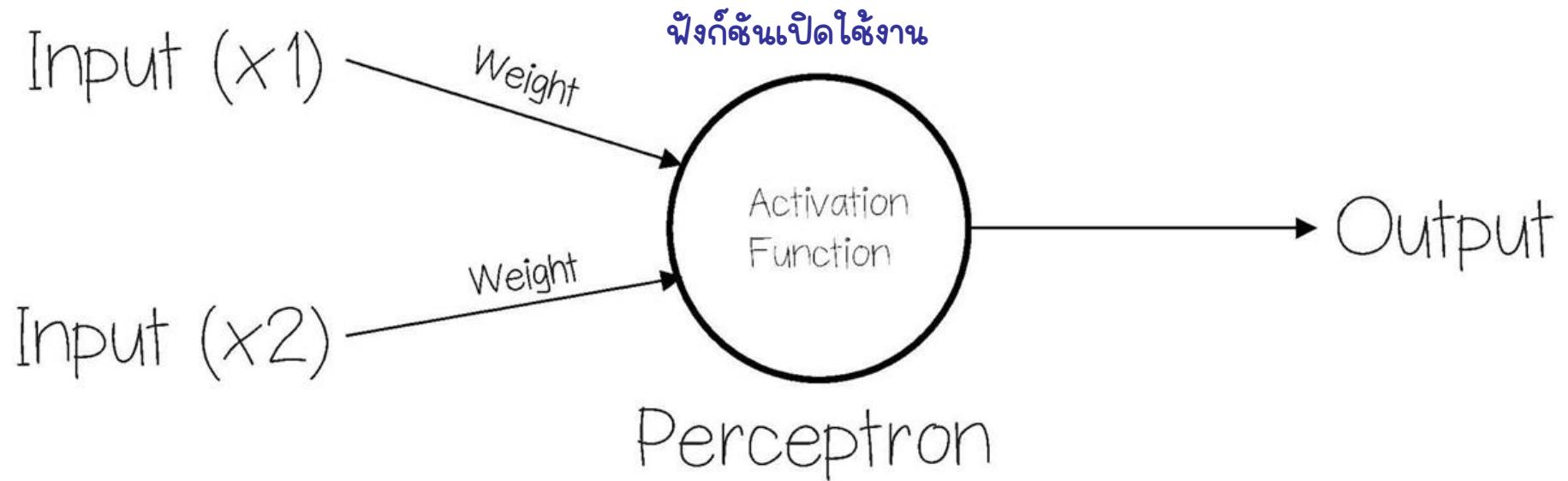
หากทำการสื่อสารได้สำเร็จ, นิวรอนจะสูบกระตุ้น



អំពីការត្រួតពិនិត្យ: នឹងរាយបន្ថែមលើស្ថាបនាទៀតមែនដឹងទៀតដោយចាប់ផ្តើម

Algorithm Explained:

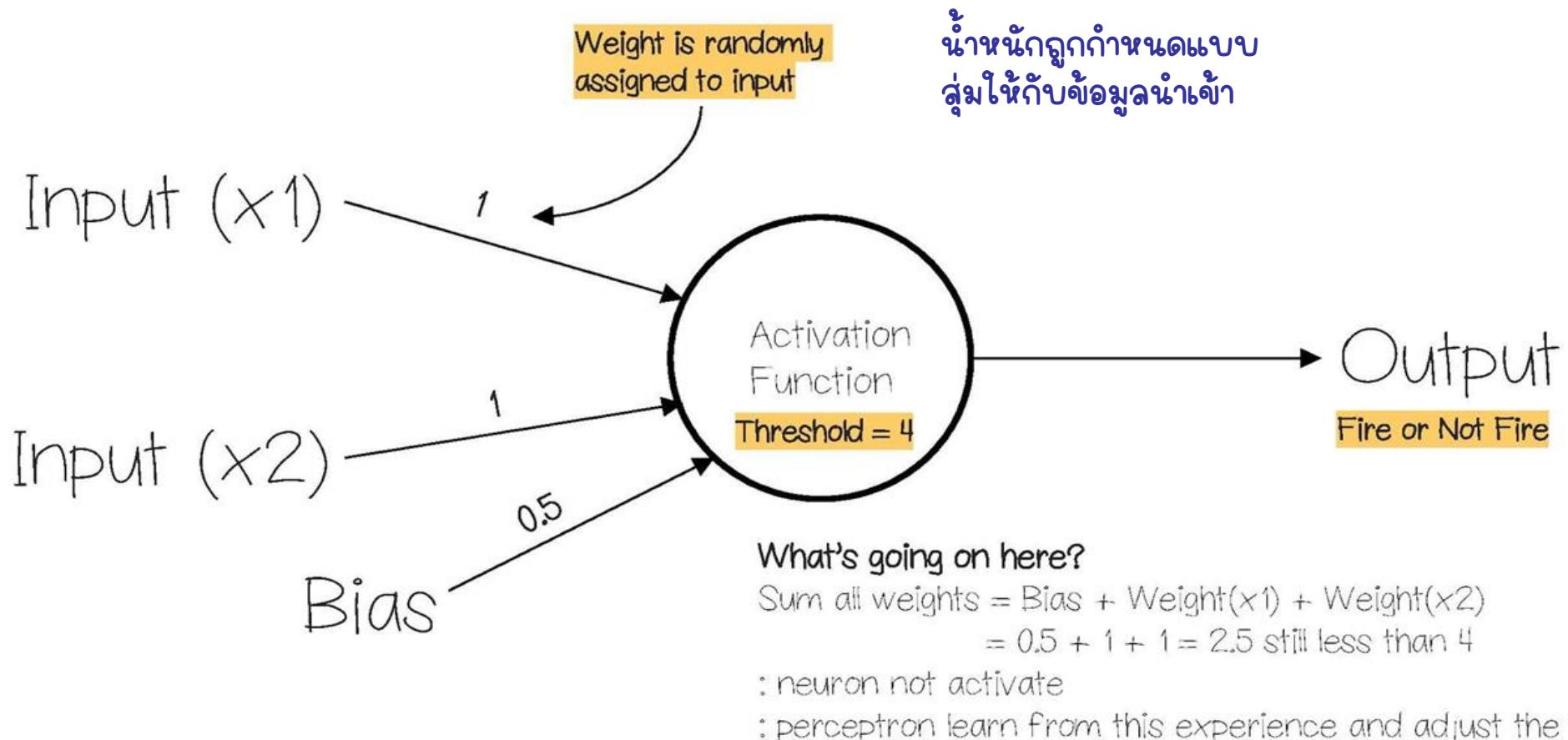
Neurons fire only when it's the signal strong enough to activate them



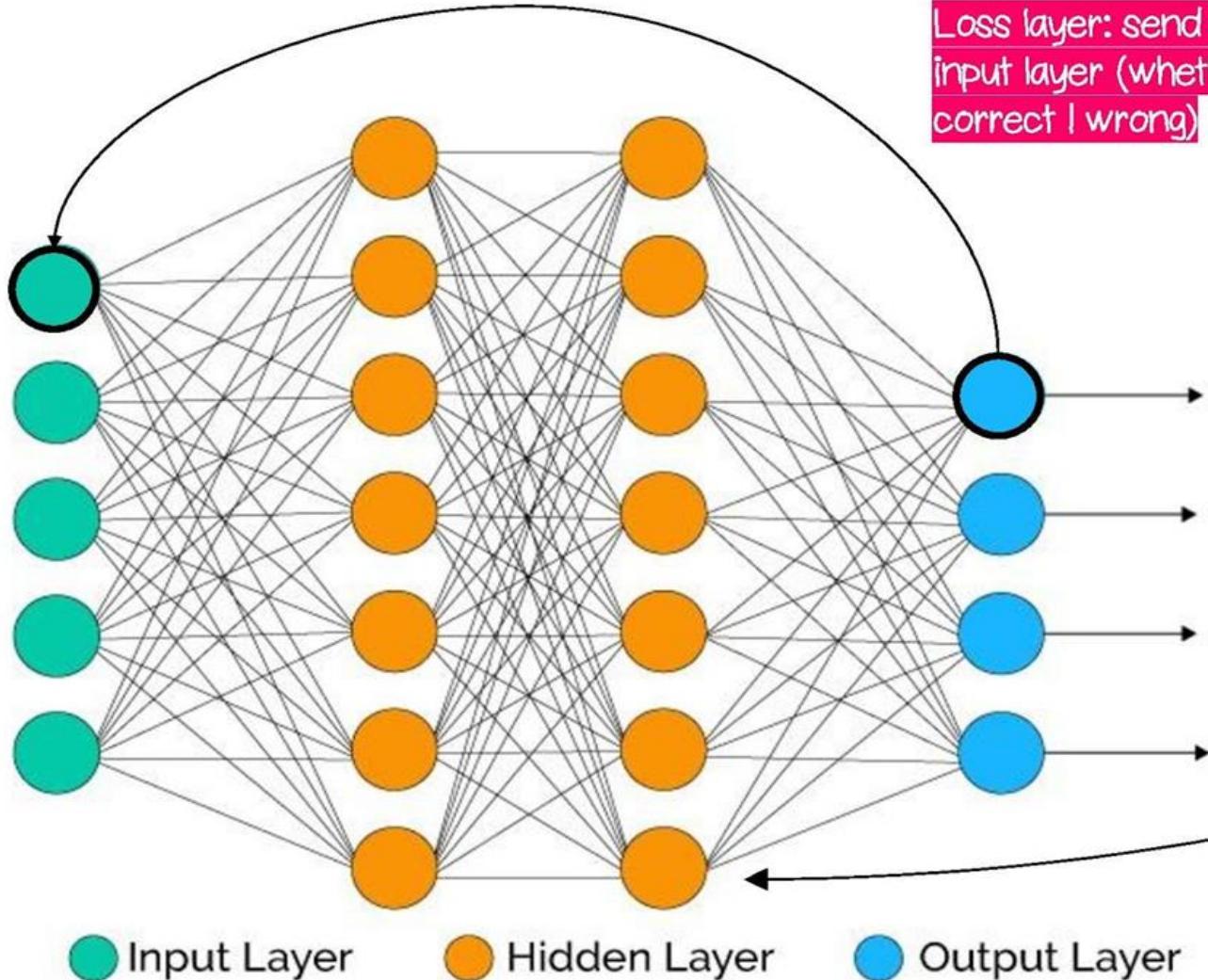
NOT USE

Algorithm Explained:

Neurons fire only when it's the signal strong enough to activate them



NOT USE



Loss layer: send feedback to input layer (whether prediction is correct | wrong)

Loss layer: សំគាល់តិចមកលប់ប្រព័ន្ធដែលមូលដ្ឋាន (វាតារមុខគោរពណ៍ត្រួតពិនិត្យ)

សំគាល់គោរពតាមរយៈប្រភពភាពហើយមីនុយទីនេះជាន់អ្នកបង្កើត ឬបោរិចអាមិនធូរ គឺតុលាការនៃលេខរឹងនៅក្នុងបញ្ហាទីនេះ។ សំគាល់គោរពតាមរយៈប្រភពភាពហើយមីនុយទីនេះជាន់អ្នកបង្កើត ឬបោរិចអាមិនធូរ គឺតុលាការនៃលេខរឹងនៅក្នុងបញ្ហាទីនេះ។

For basic neural networks,
hyperparameter is number of hidden
layer and learning rate (alpha)

Remarks:

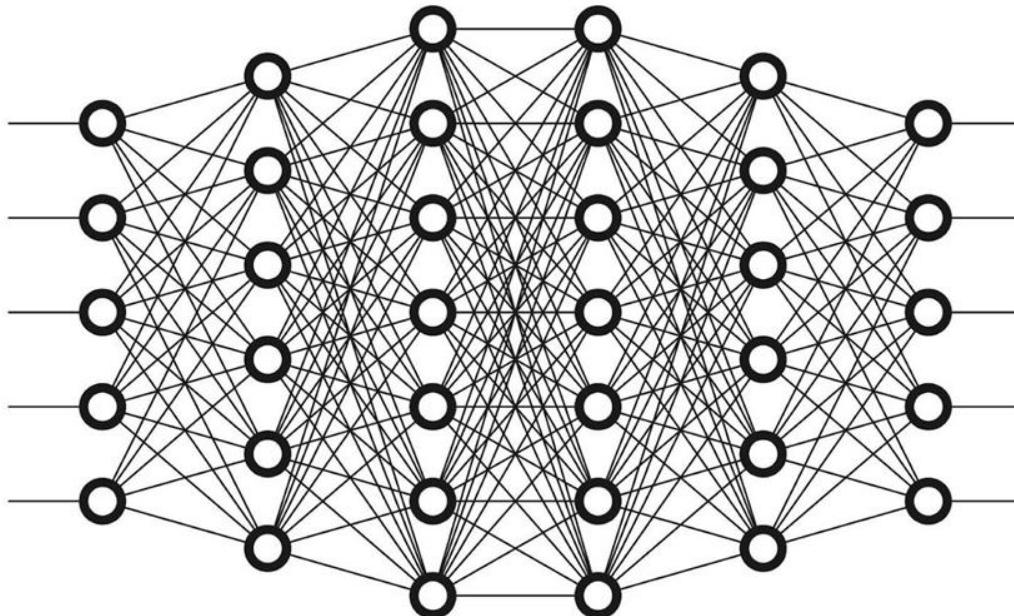
NN is a good example of supervised learning that also a reinforcement learning (learn from experience)

<https://medium.com/@xenonstack/overview-of-artificial-neural-networks-and-its-applications-2525c1addff7>

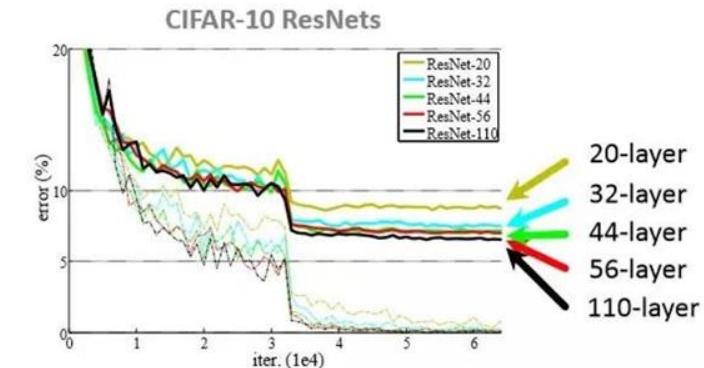
NN បើនពីរមួយចំណាំដែលទទួលបានការបង្កើតឡើងពេញលេញនៅក្នុងការសារព័ត៌មាន (supervised learning) និងការបង្កើតឡើងពេញលេញនៅក្នុងការសារព័ត៌មាន (reinforcement learning)

Deep Learning

Expands the number of hidden layers



การเรียนรู้เชิงลึกหลายชั้นที่ซ่อน



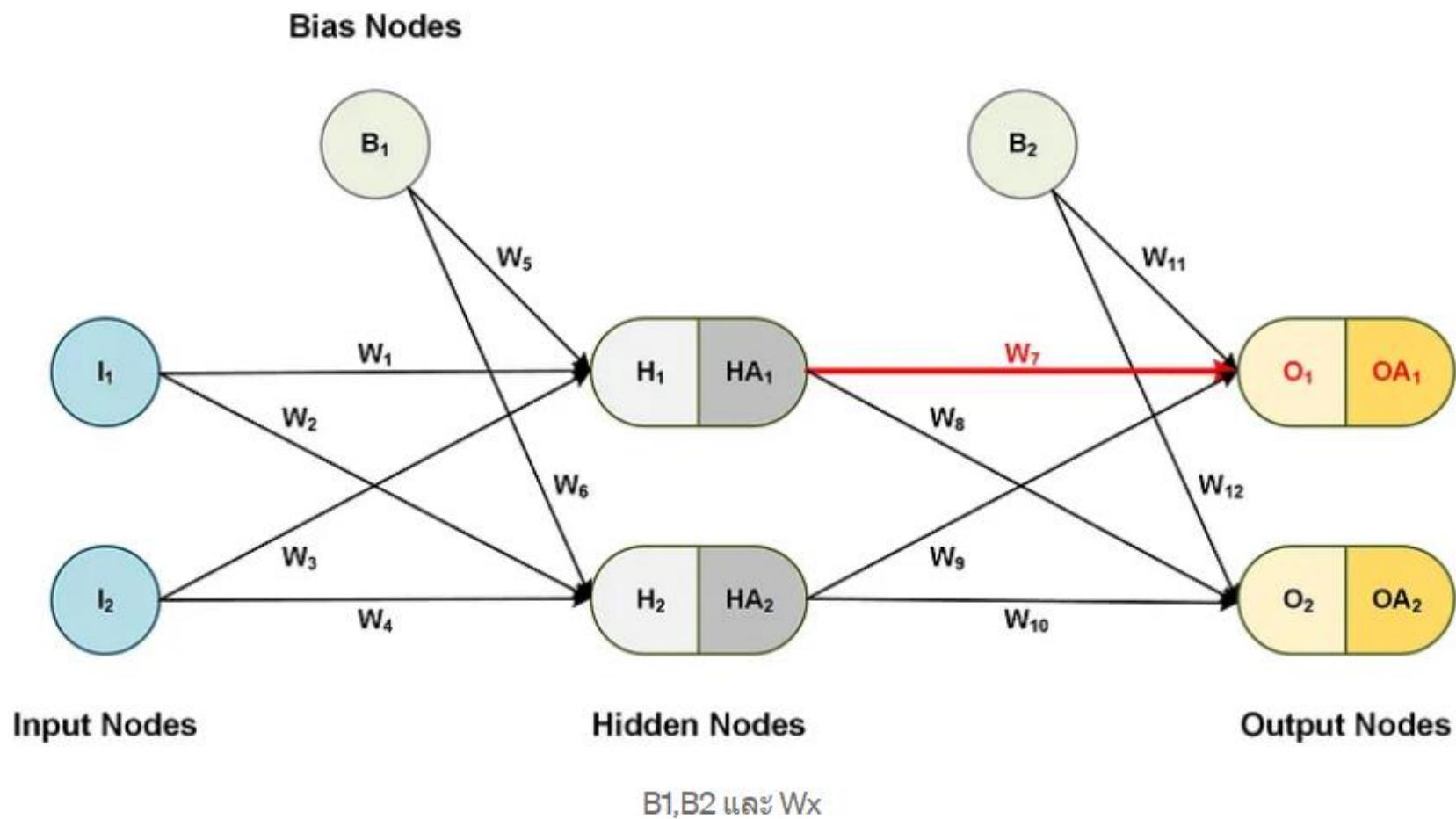
Hieu Pham, Has done some machine learning

Updated Apr 29, 2017

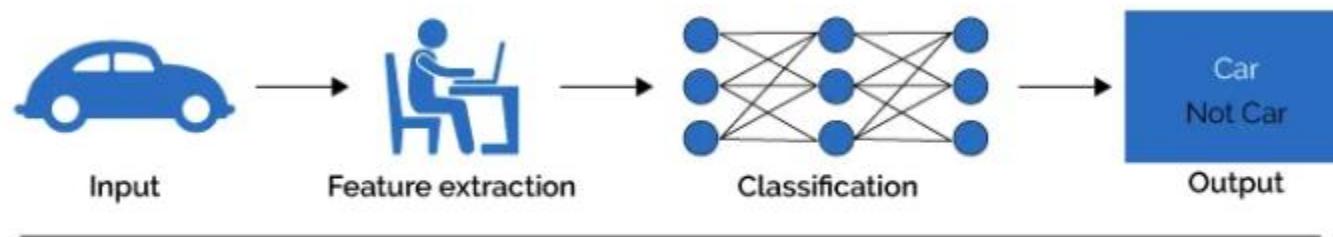


In this paper (Huang et al, 2016), the authors report that they "can increase the depth of residual networks **even beyond 1200 layers** and still yield meaningful improvements in test error (4.91% on CIFAR-10)".

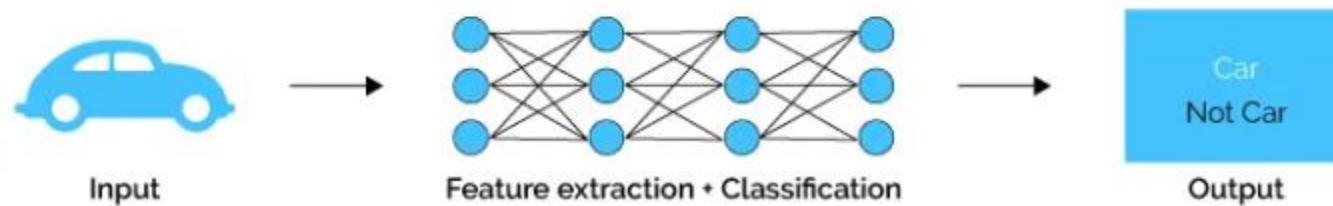
I have never heard of anything deeper than that.



Machine Learning

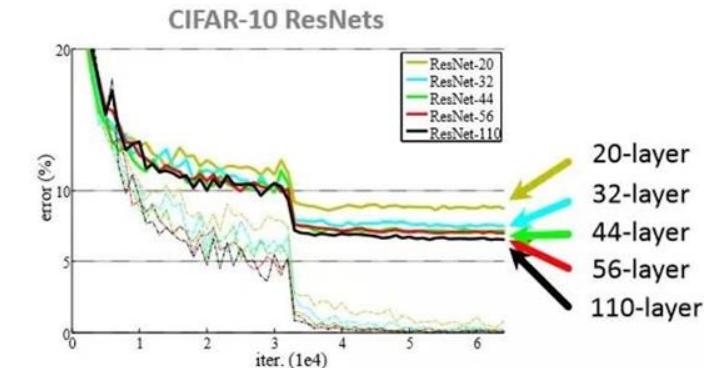
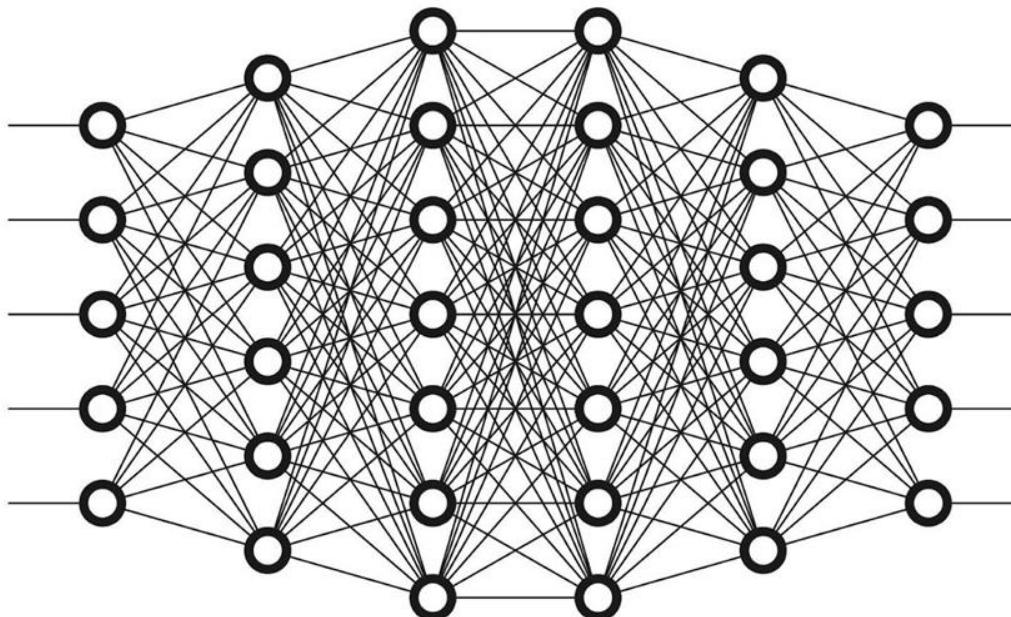


Deep Learning



Deep Learning

Expands the number of hidden layers



Hieu Pham, Has done some machine learning

Updated Apr 29, 2017

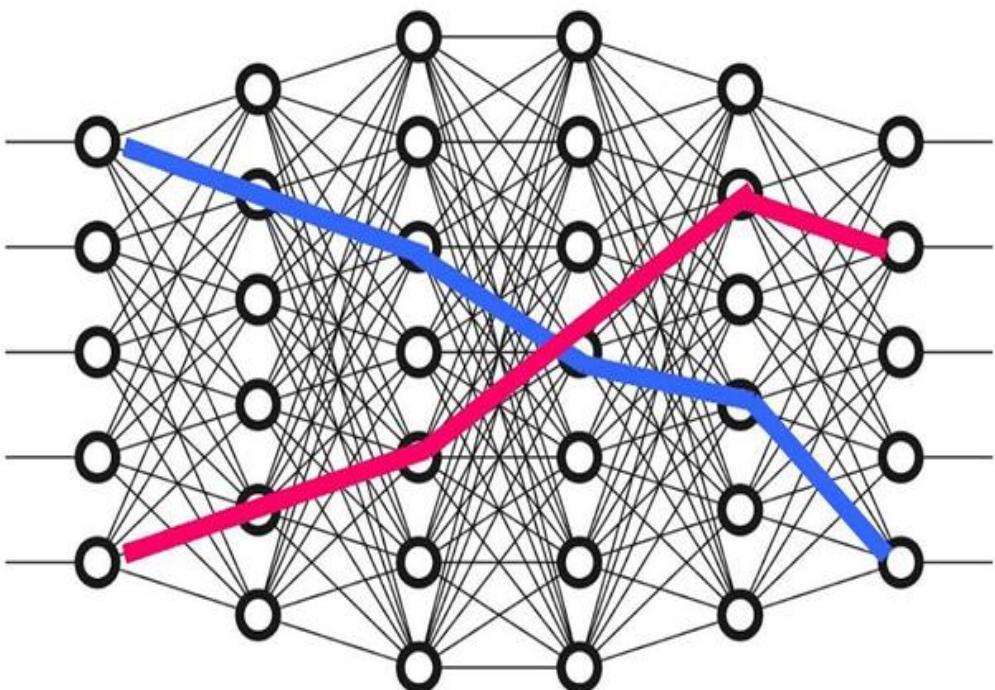


In this paper (Huang et al, 2016), the authors report that they “can increase the depth of residual networks **even beyond 1200 layers** and still yield meaningful improvements in test error (4.91% on CIFAR-10)”.

I have never heard of anything deeper than that.

Backpropagation

The backbone of artificial neural networks



Backpropagation เป็นร้อยรื่อในโครงสร้าง
ประสาทเทียม

Wrong

Right

Send feedback to
input layers and
adjust the weights

ส่งคำติชมกลับไปยังชั้นป้อนข้อมูลนำเข้าและ
ปรับน้ำหนัก

NO FREE LUNCH

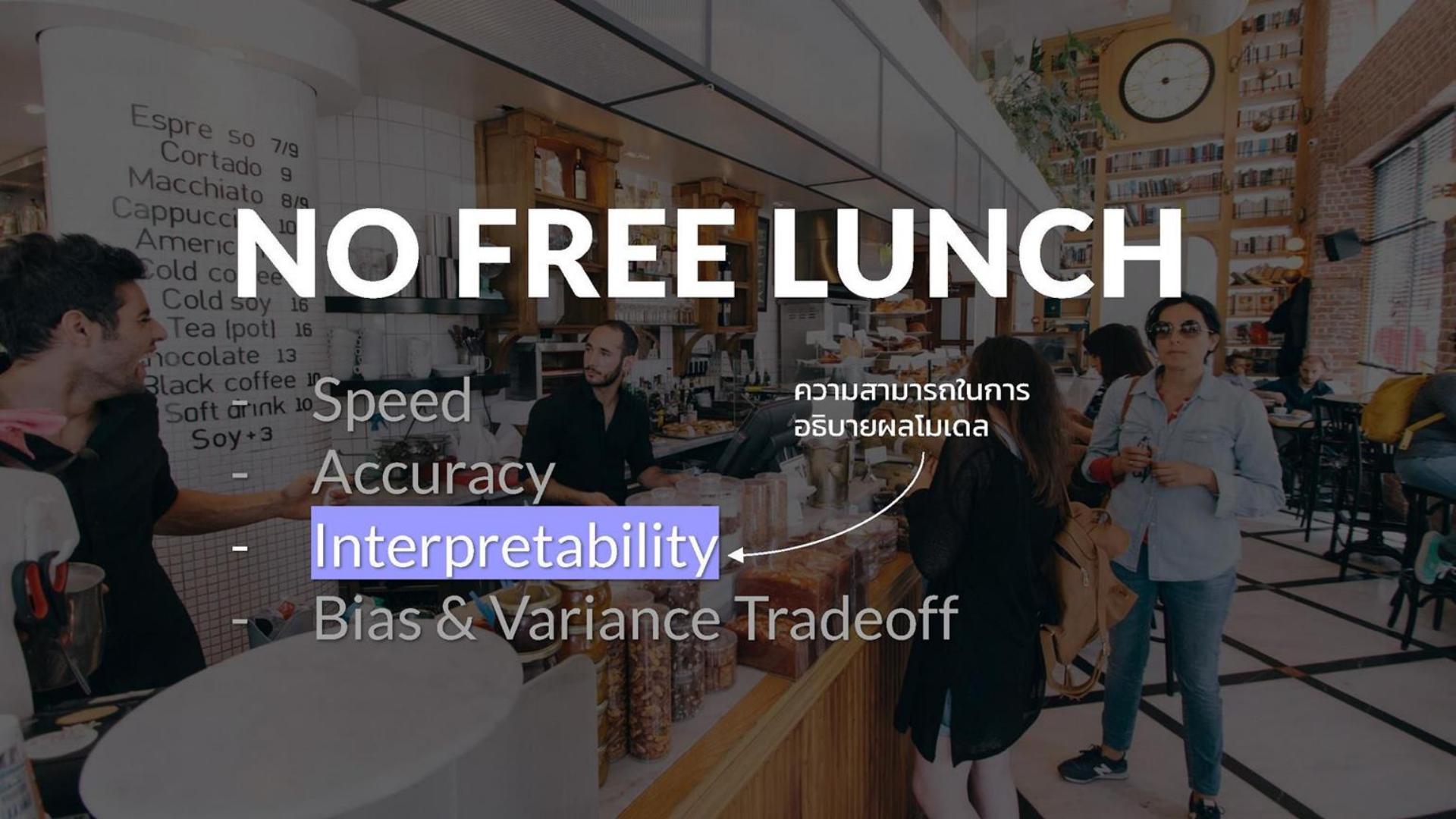
Speed

- Accuracy

- Interpretability

- Bias & Variance Tradeoff

ความสามารถในการ
อธิบายผลไมเดล



Advantages:

- Amongst the very best performers | learners in the world
- Can be developed into deep learning algorithms
- Work with both regression and classification problems

Disadvantages:

- Requires large data set
- Black Box
- Prone to overfitting problem
- Need more power if more hidden layers

Time to buy a
graphic card :D



- อยู่ในระดับผู้แสวงความสามารถที่ดีเยี่ยมในโลก
- สามารถพัฒนาเป็นอัลกอริทึมการเรียนรู้เชิงลึกได้
- ทำงานถูกปั๊บหากการติดต่ออย่างและการจำแนกประเภททั้งคู่

- ต้องการซุ่มจ้มูลน้ำใจไว้
- ห่อหุ้มด้วย
- มีโอกาสต่อการเกิดปัญหาการหลุดละเมิดไป
- ต้องการพลังงานมากจึงเมื่อมีชั้นที่ซ่อนมากก็จะ

Required Packages:

`library(caret)` – train & test models

`library(caTools)` – split data into training & testing sets

`library(nnet)` – used to train neural network in caret

Neural Network

```
method = 'nnet'
```

Type: Classification, Regression

Tuning parameters:

- `size` (#Hidden Units)
- `decay` (Weight Decay)

Required packages: `nnet`

A model-specific variable importance metric is available.

ដំណោល៖ ការសរុបមែន
បង្កើតឡើងជាប្រព័ន្ធដែល
ធ្វើនៅក្នុងការដំណោល
នៃការគិតតាមភាពនៅក្នុង caret

មិនអាចរាយការណ៍តាមរយៈ
ការសរុបមែនដែលត្រូវបានបង្កើតឡើង

Machine Learning 3(2-2-5), 2/2565

Unit 4

Unsupervised Learning

อ.อานันท์ ไม่ประดิษฐ์

K-Means เป็นอัลกอริทึมที่ใช้ในการจัดกลุ่มข้อมูลที่มีอยู่ในกลุ่มหลาย ๆ กลุ่มโดยพยากรณ์ทำให้ข้อมูลในกลุ่มเดียวกันมีความคล้ายคลึงกันมากที่สุด โดยการนำมาวิเคราะห์และแบ่งกลุ่มข้อมูลในลักษณะที่ไม่ได้ระบุล่วงหน้า การจัดกลุ่มนี้มักถูกนำมาใช้ในการพบรคความลับเพื่อจัดกลุ่มหรือสร้างโมเดลทำงานโดยที่ไม่มีการระบุชัดเจน

UNSUPERVISED: K-MEANS

การจัดจำแนกที่หลักของอัลกอริทึมแบบไม่มีการสอน

Recall the main task of unsupervised algorithm:

To find patterns in our data set

(Descriptive – no right | wrong answer)

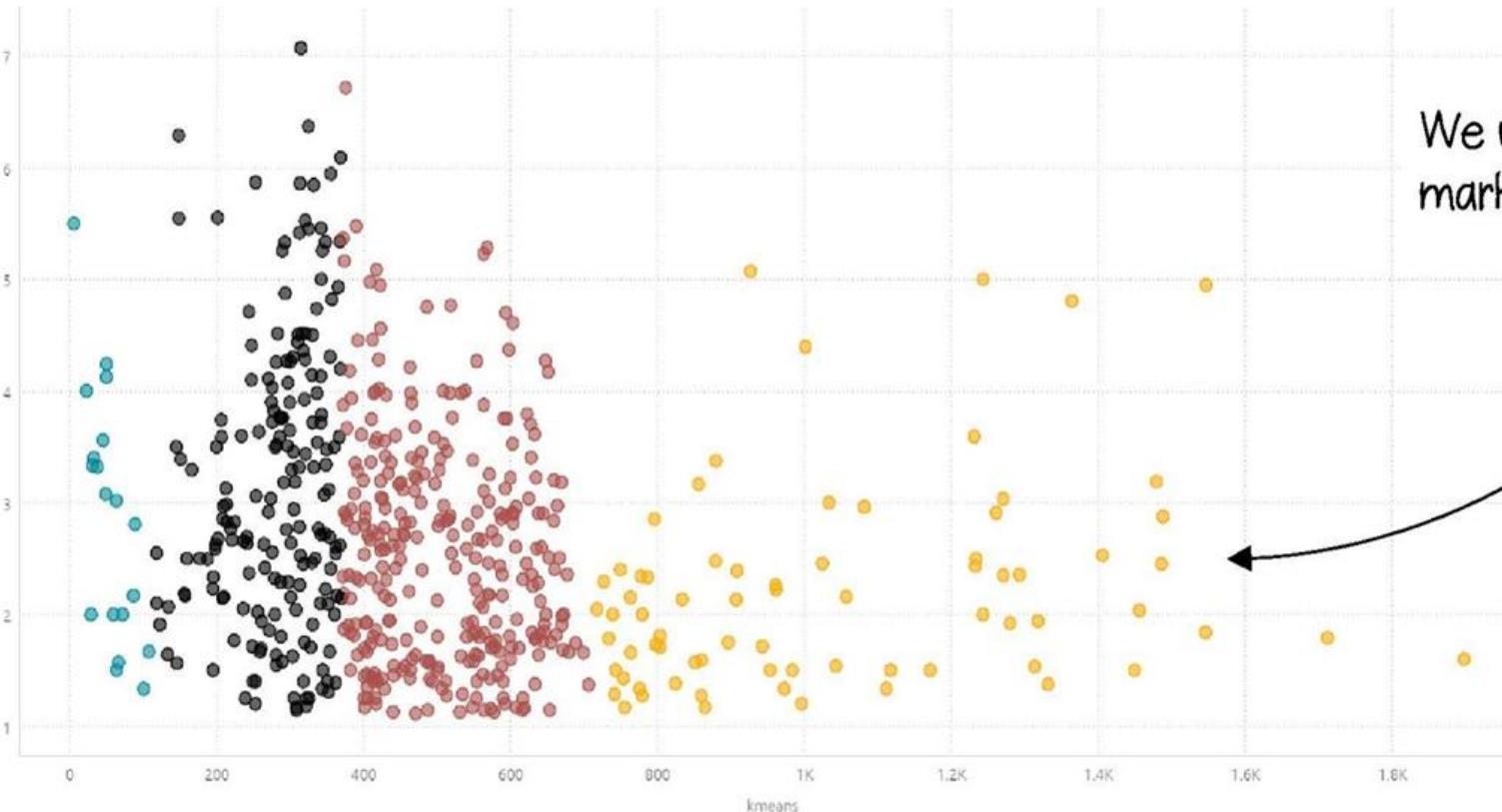
ค้นหากฎแบบในชุดข้อมูลของเรา (คำอธิบาย – ไม่มีคำตอบที่ถูกหรือผิด)

มันเป็นอย่างไรที่มีการวิเคราะห์ตามบุคคลมาก แต่เมื่อ
เทคโนโลยีดิจิทัลสูงบางอย่างที่ช่วยนำทางการ
ตัดสินใจของเรา (ไม่ได้รับการอภิปรายที่นี่)

It's very subjective. But there
are some advanced techniques
to guide our decisions (not
discussed here).

K-Means is very popular unsupervised learning algorithms

K-Means เป็นอัลกอริทึมการเรียนรู้แบบไม่มีการสอนที่มีความนิยมมาก

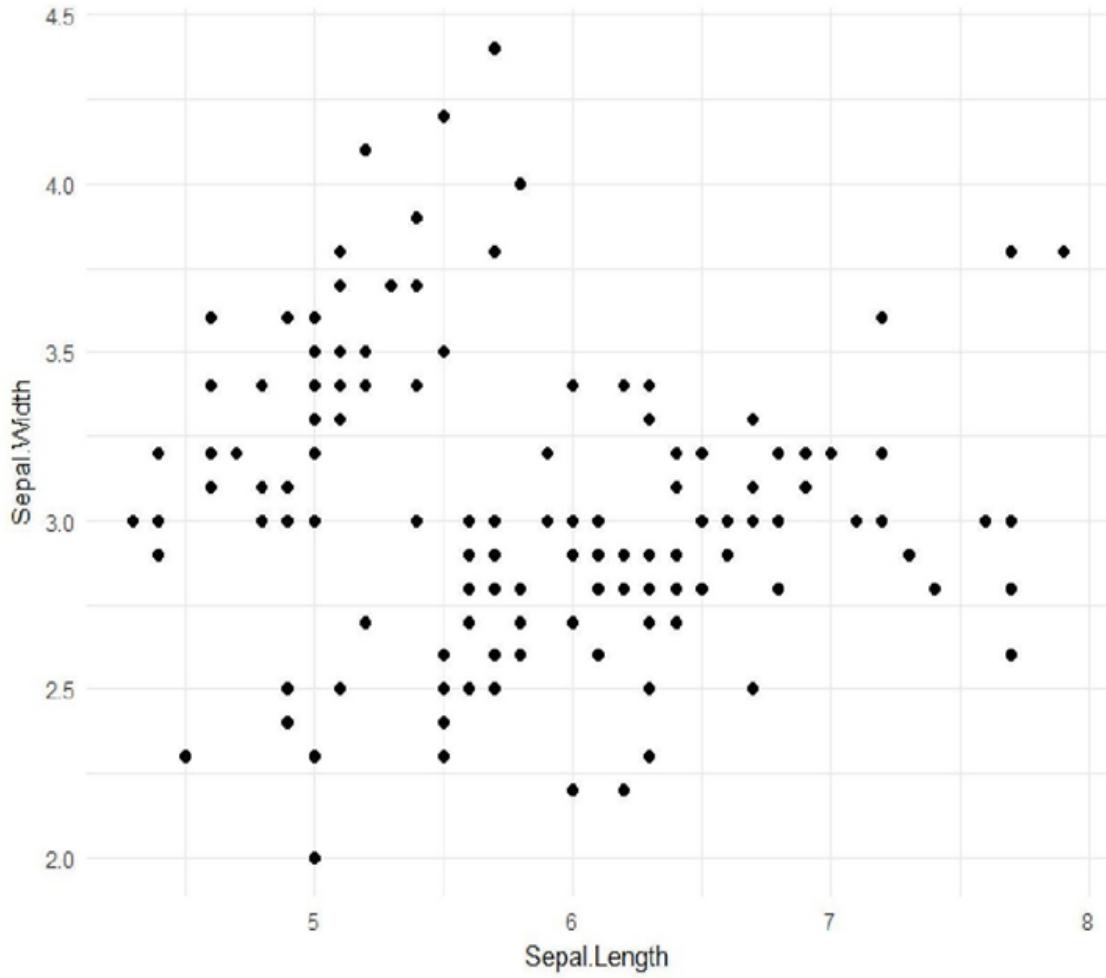


We use it extensively in market segmentation study

เราใช้มันอย่างแพร่หลายในการศึกษาsegmentation เต็มที่ในตลาด

K-Means is **very efficient**, easily done in one line

K-Means มีประสิทธิภาพมาก และง่ายต่อการใช้เพียงบรรทัดเดียว



```
> library(ggplot2)  
> ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +  
  geom_point(size = 2) + theme_minimal()
```

```
> set.seed(2)  
> km_fit <- kmeans(iris[1:4], centers = 3)
```

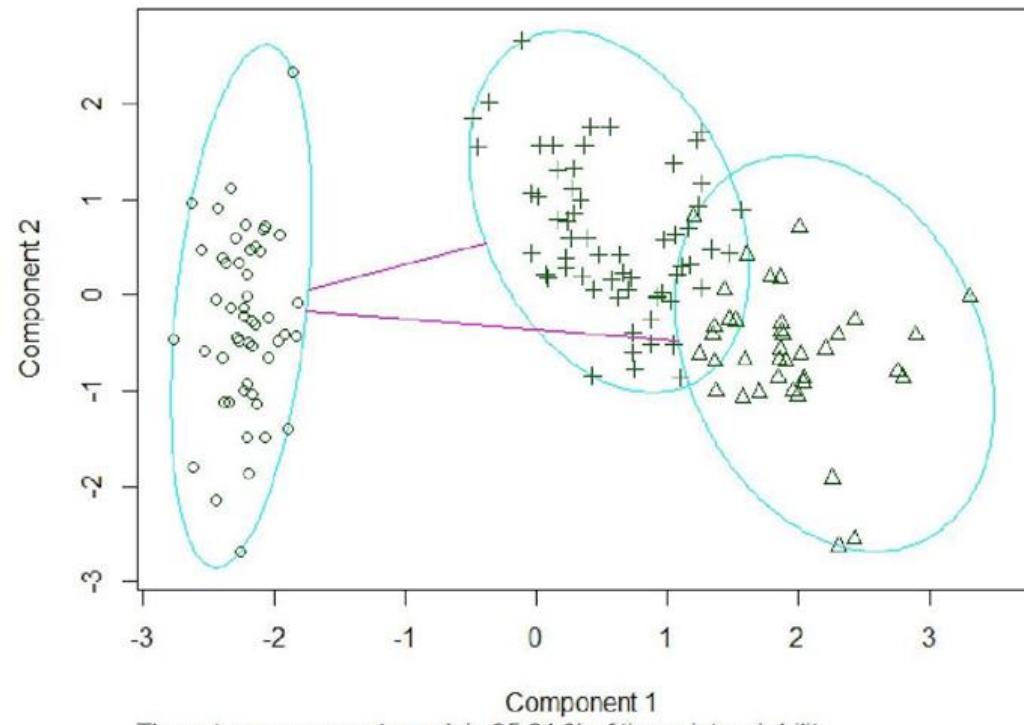
Specify (guess)
number of clusters

โดยการระบุ (เดา) จำนวนของกลุ่ม

In reality, we don't know how accurate our model is.

ในความเป็นจริงเราไม่ทราบว่าโมเดลของเรามีความแม่นยำอย่างไร

cluster plot



```
> library(clusplot)  
> clustplot(iris[1:4], km_fit$cluster, main = "cluster plot")
```

```
> # cross-check with actual class in iris dataframe  
> table(km_fit$cluster, iris$Species)
```

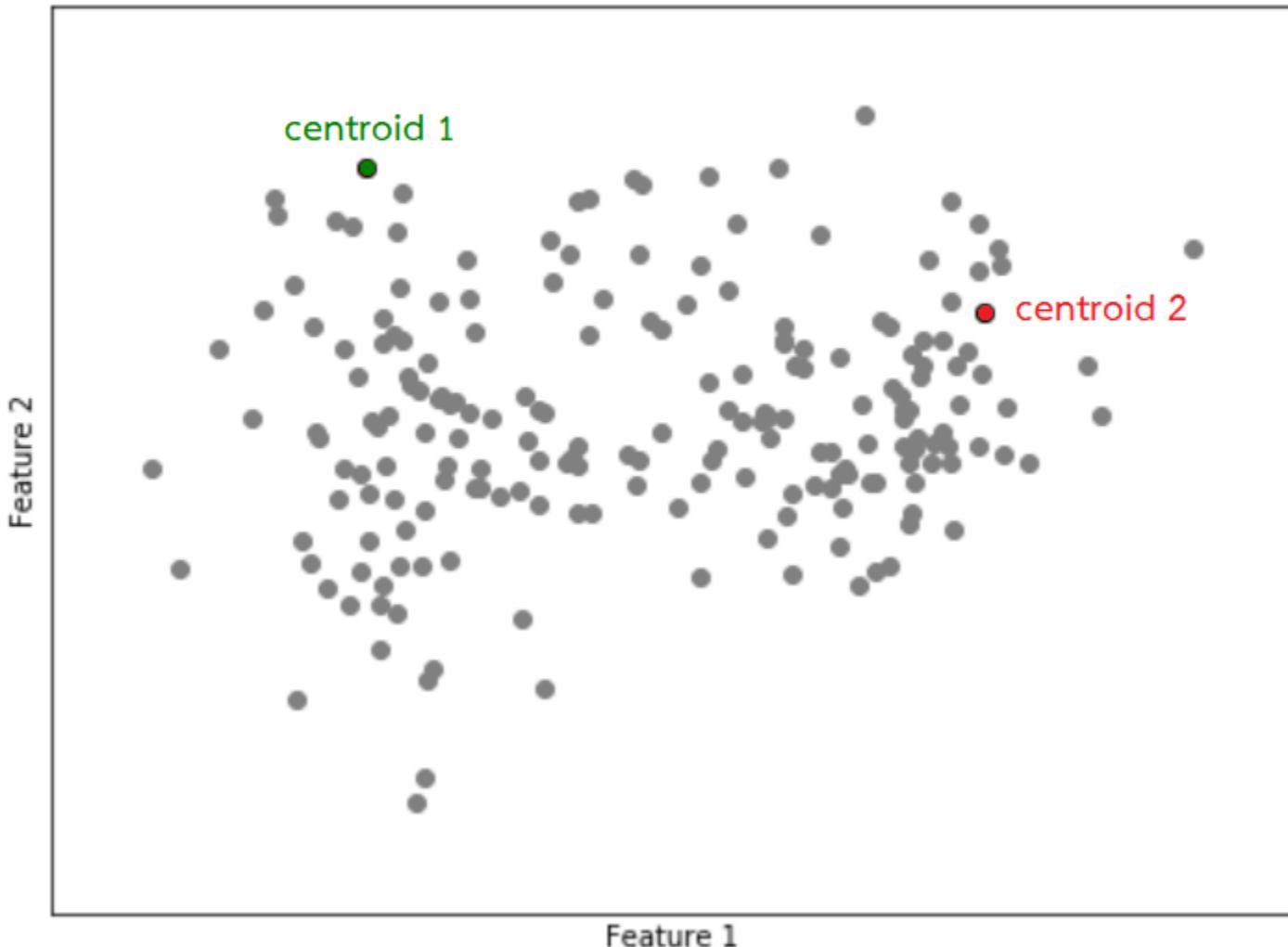
K-means Cluster

คลุ่มของ K-means

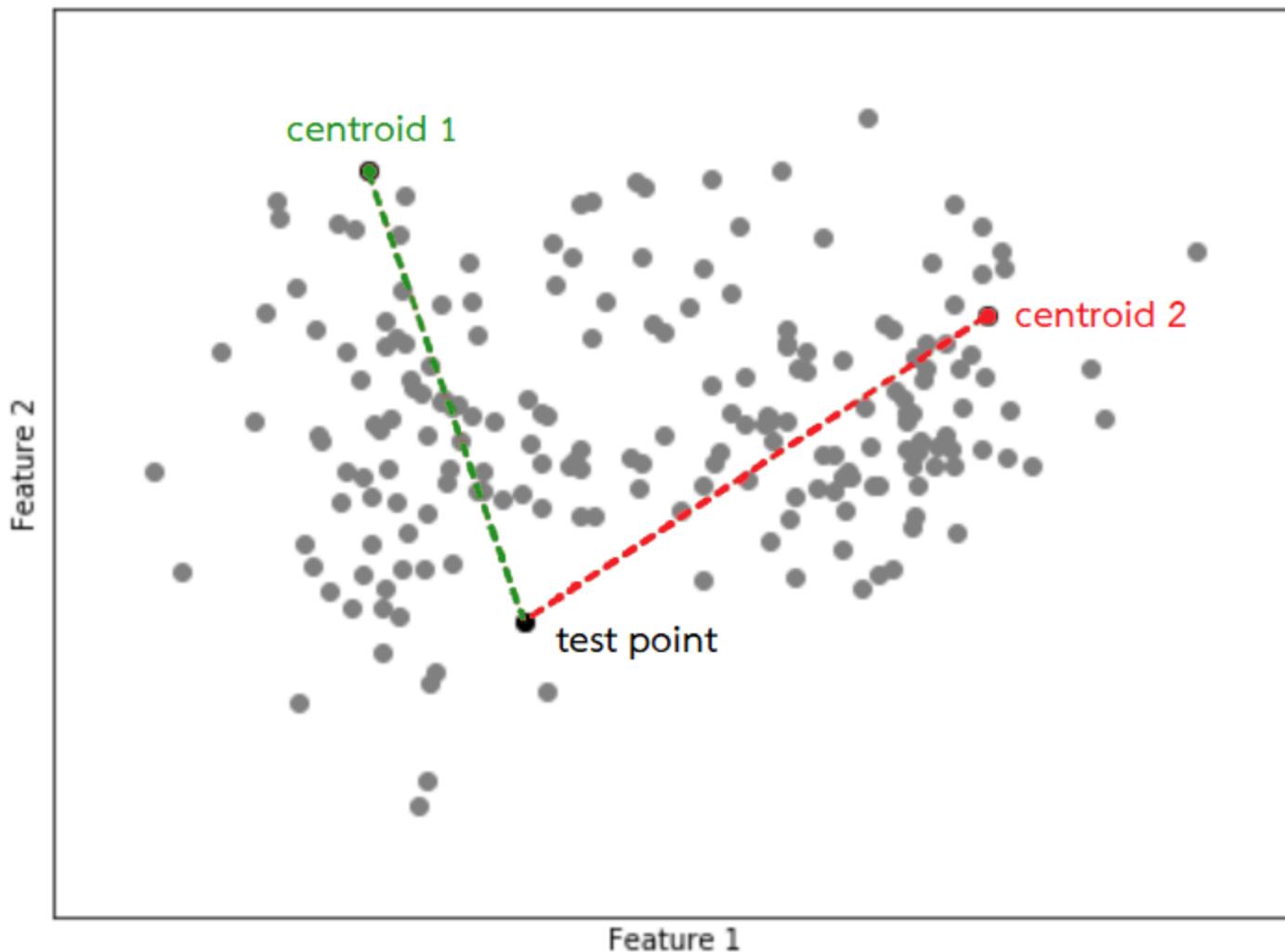
Actual Class

คลาสที่แท้จริง

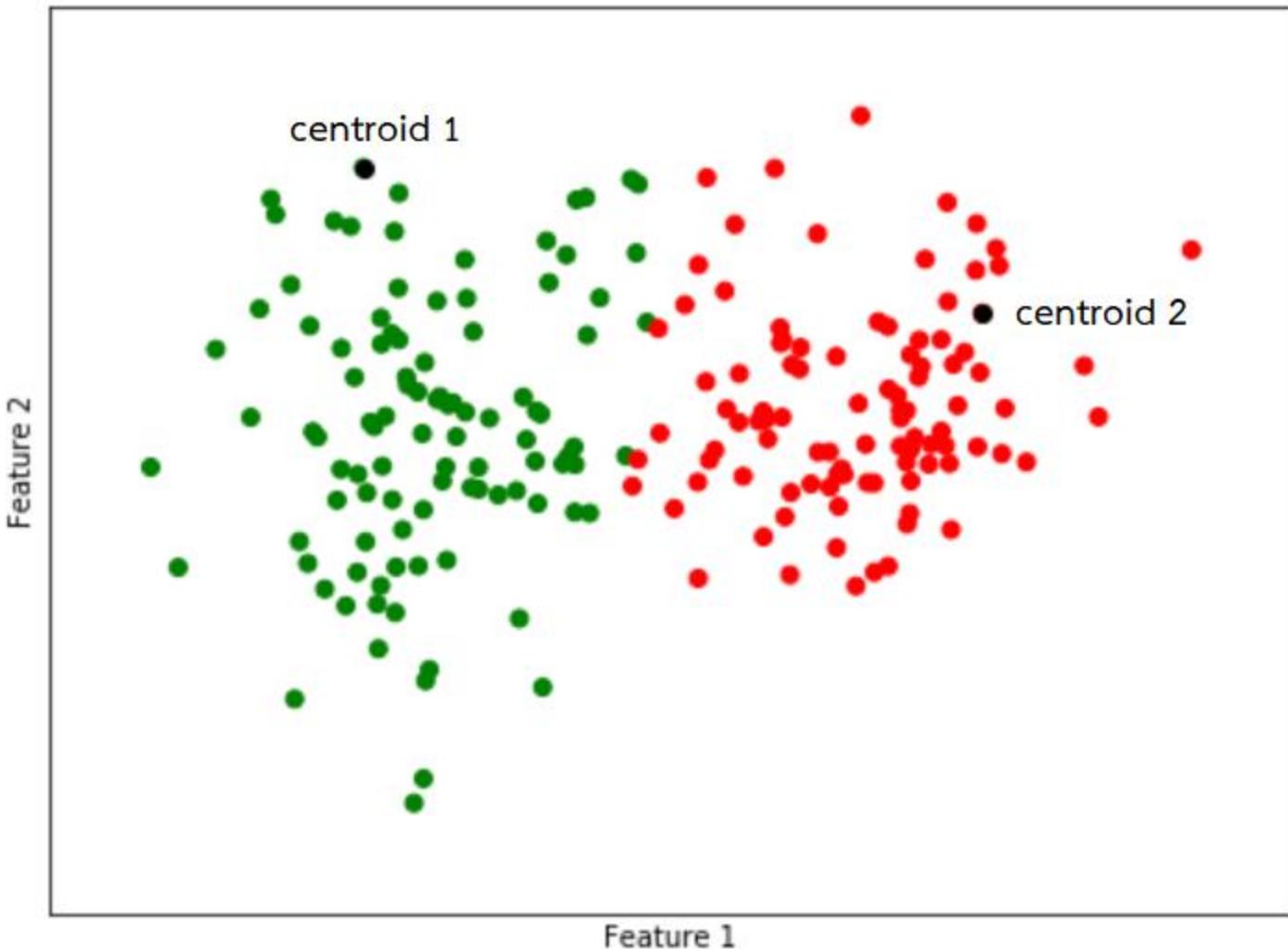
1. ลุ่มข้อมูลมา k records เพื่อใช้เป็น centroid (ในกรณีนี้ $k = 2$)



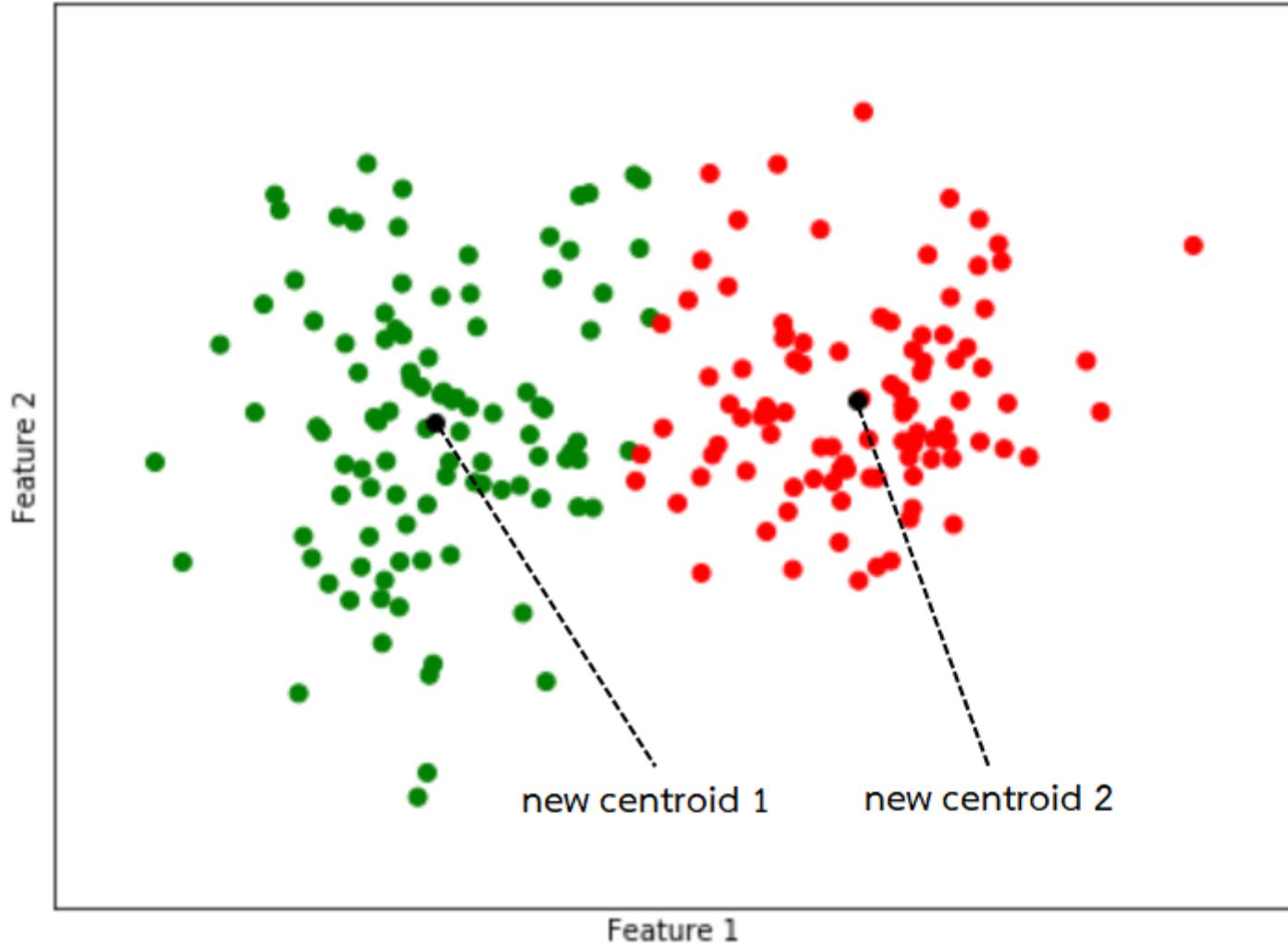
2. คำนวณระยะห่างระหว่างข้อมูลแต่ละดวงกับ centroid ที่มีอยู่ 4 จุด



3. จัดกลุ่มจุดข้อมูลให้ไปอยู่กับลุ่มเดียวกับ centroid กี่อยู่ใกล้กี่สุด



4. ปรับตำแหน่งของ centroid ในแต่ละกลุ่มไปเป็นค่าเฉลี่ยของข้อมูลทั้งหมดที่อยู่ที่ในกลุ่มนั้น



5. ทำขั้นตอนที่ 2 ถึง 4 จนกว่าจะถึงเกณฑ์ที่ใช้ในการหยุด

Advantages:

- Very easy to train
- Often produce reasonable classification results

ง่ายมากในการฝึก
มักผลลัพธ์การจำแนกหมวดหมู่ที่มีมาตรฐาน

Disadvantages:

- Number of clusters sometimes difficult to determine
- Not all features are numeric (can't find mean)

จำนวนของกลุ่มบางครั้งยากที่จะกำหนด
ไม่ทุกค่าเป็นตัวเลข (ไม่สามารถหาค่าเฉลี่ยได้)

គៀវិតការណ៍នេះនៅក្នុងបច្ចេកទេសទីមួយទៀតែនៅក្នុងការរើនភ្លើការតំលើនូវការដែលត្រូវការពិនិត្យ។ ការរើនភ្លើការតំលើនូវការនេះត្រូវបានធ្វើឡើងនៅក្នុងសាធារណៈសាស្ត្រភាព (Q-Learning) ដែលបានគិតឡើងដោយភាពិស់ន័យ។ នៅក្នុងការរើនភ្លើការតំលើនូវការនេះ គឺជាប្រព័ន្ធទឹកដែលបានគិតឡើងដោយភាពិស់ន័យ។ ការរើនភ្លើការតំលើនូវការនេះ គឺជាប្រព័ន្ធទឹកដែលបានគិតឡើងដោយភាពិស់ន័យ។

Reinforcement Learning: Q-Learning

So far... Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map $x \rightarrow y$

Examples: Classification,
regression, object detection,
semantic segmentation, image
captioning, etc.

- การจำแนกประเภท
- การจดจำ
- การตรวจสอบวัตถุ
- การแปลงล้วนด้วยความหมาย
- การใส่คำอธิบายในภาพ



Classification

So far... Unsupervised Learning

Data: x

Just data, no labels!

ข้อมูลเท่านั้น ไม่มีป้ายเลื่อน

Goal: Learn some underlying hidden *structure* of the data

เรียนรู้โครงสร้างที่ซ่อนอยู่ของข้อมูลบางส่วน

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

การจัดกลุ่ม การลดขนาด การเรียนรู้คุณลักษณะ การประมาณค่าความหนาแน่น

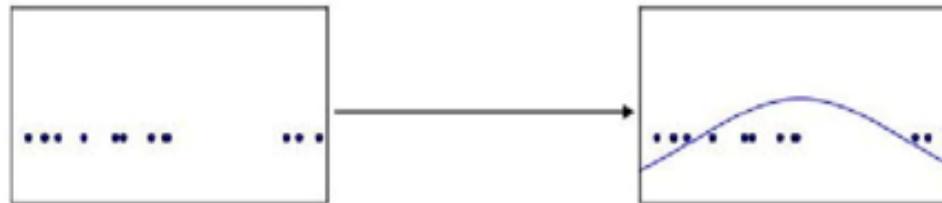
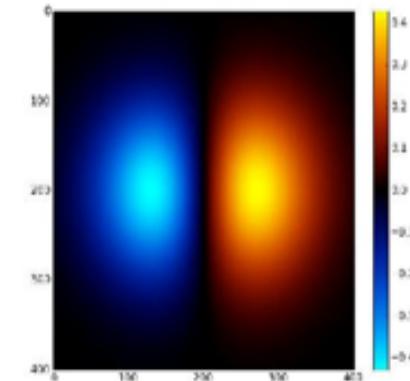
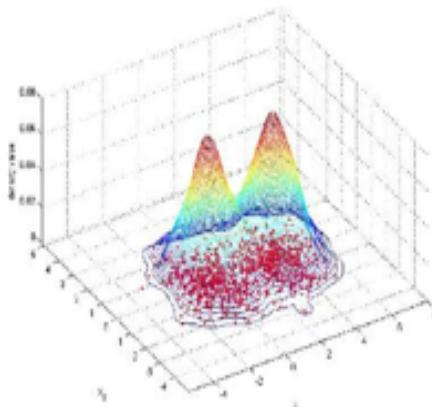


Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-d density estimation

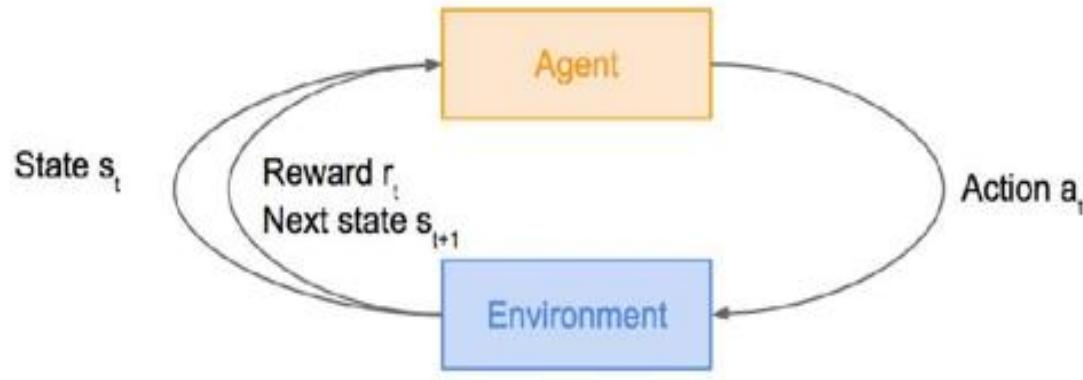


2-d density estimation

Today: Reinforcement Learning

ปัญหาที่เกี่ยวข้องกับตัวแทนที่มีปฏิสัมพันธ์กับสิ่งแวดล้อม ซึ่งให้สัญญาณ
รางวัลที่เป็นตัวเลข

Problems involving an **agent**
interacting with an **environment**,
which provides numeric **reward**
signals



Goal: Learn how to take actions
in order to maximize reward

เรียนรู้การดำเนินการเพื่อสร้างรางวัลให้มากที่สุด



Overview

- What is Reinforcement Learning?
- Markov Decision Processes
- Q-Learning
- Policy Gradients

Reinforcement Learning

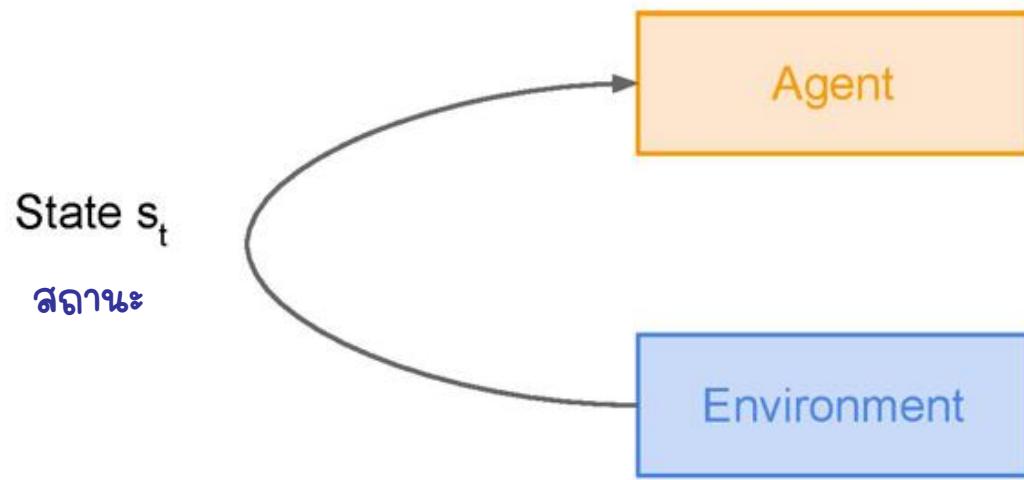
ตัวแทน

Agent

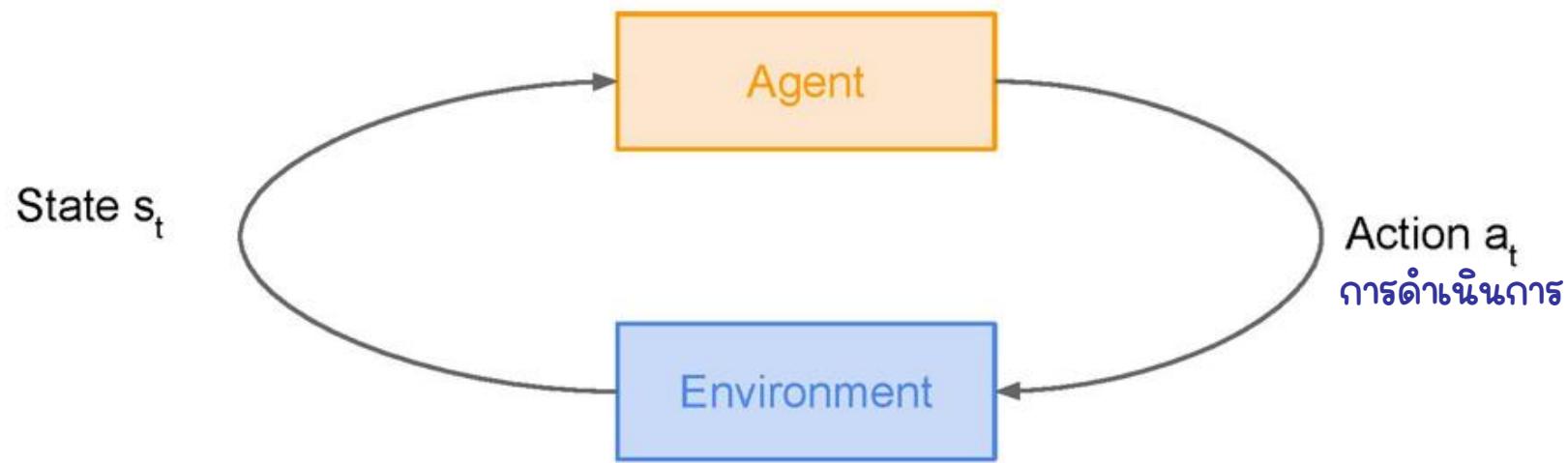
Environment

สิ่งแวดล้อม

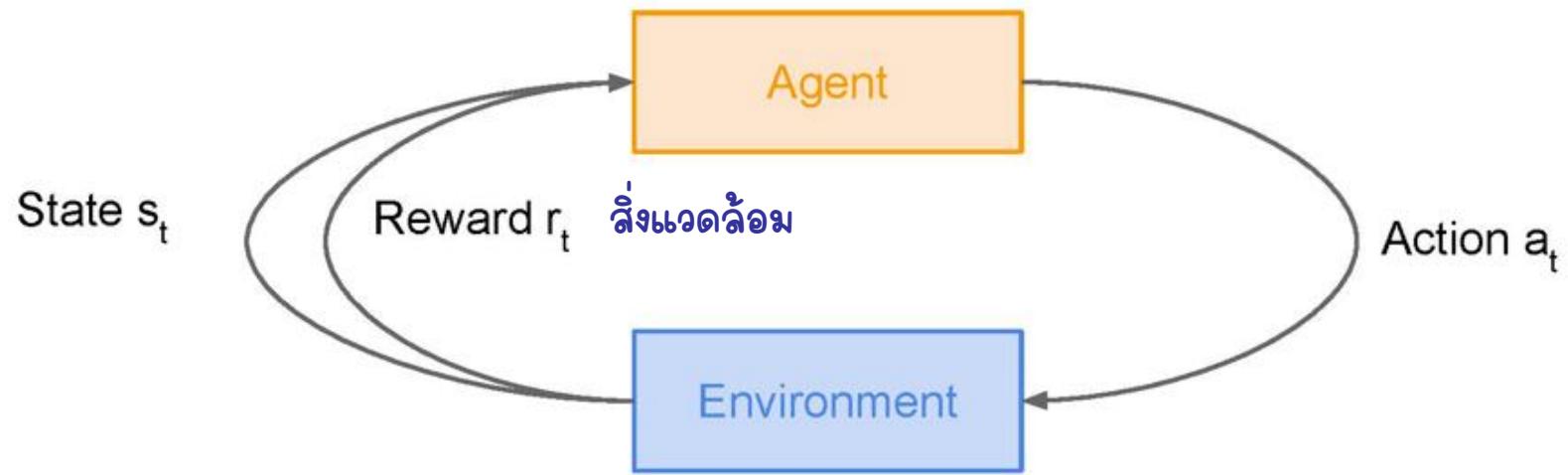
Reinforcement Learning



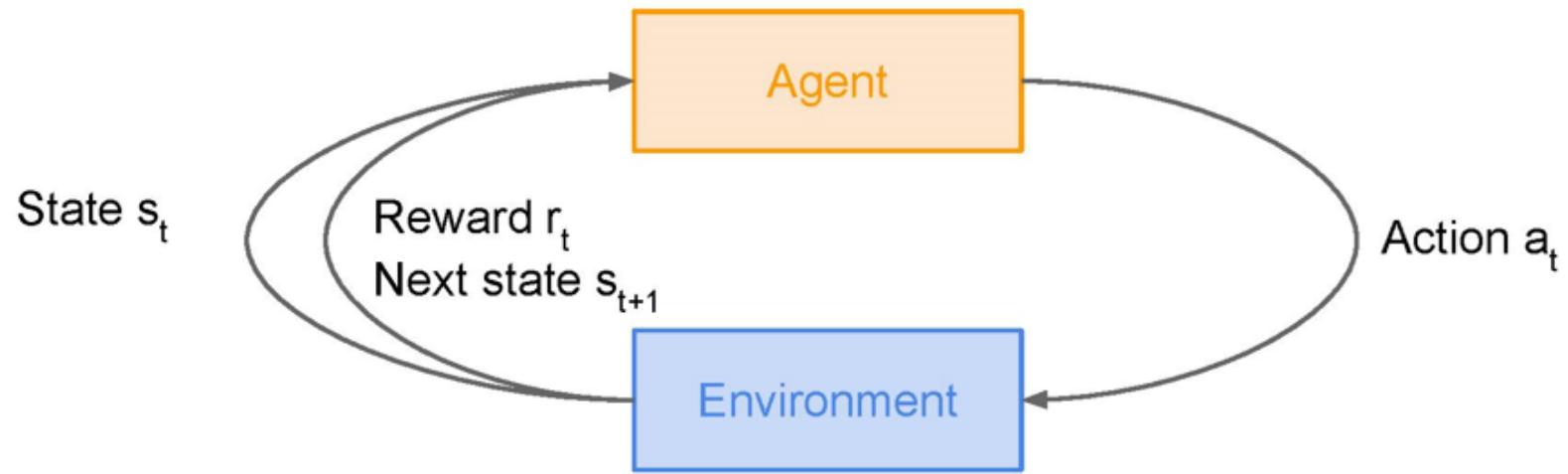
Reinforcement Learning



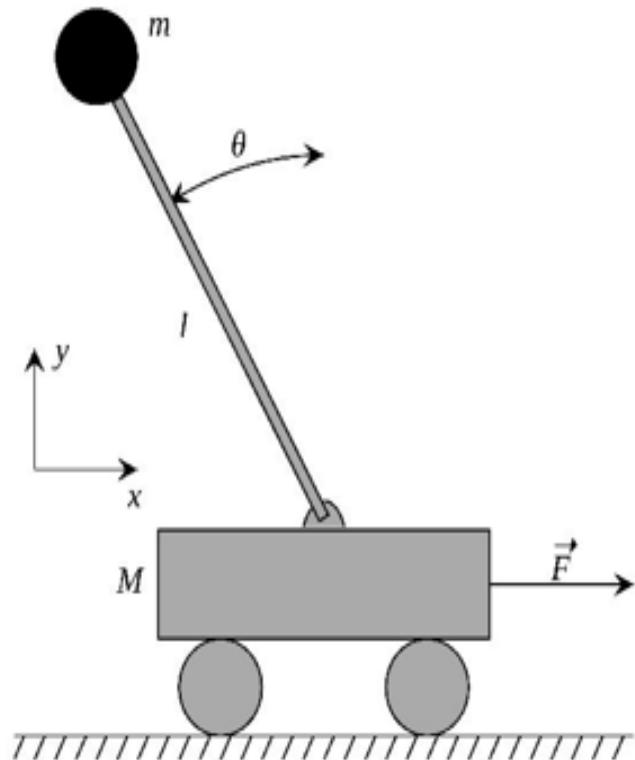
Reinforcement Learning



Reinforcement Learning



Cart-Pole Problem



Objective: Balance a pole on top of a movable cart

State: angle, angular speed, position, horizontal velocity

Action: horizontal force applied on the cart

Reward: 1 at each time step if the pole is upright

ปัญหา Cart-Pole

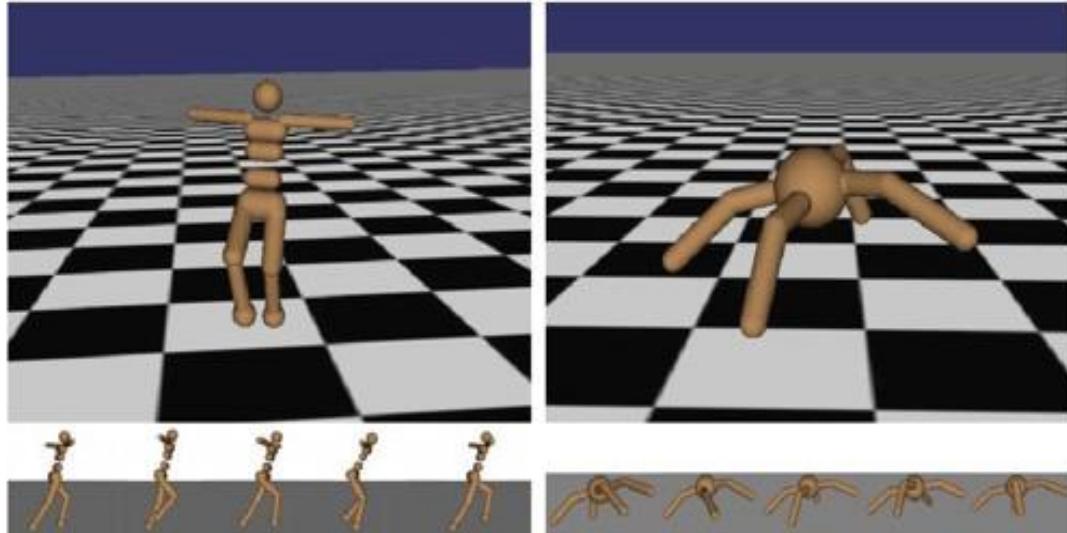
Obj : การปรับสมดุลควบคุมให้平衡ได้

State : มุม, ความเร็วในมุม, ตำแหน่ง, ความเร็วแนวอนุ

Action : แรงโน้มถ่วงที่ใช้บันรัฐบาล

Reward : 1 ที่ทุกจังหวะเวลาหากเสาตั้งตรง

Robot Locomotion



Objective: Make the robot move forward

State: Angle and position of the joints

Action: Torques applied on joints

Reward: 1 at each time step upright + forward movement

Obj : ทำให้หุ่นยนต์เคลื่อนที่ไปทางหน้า

State : มุมและตำแหน่งของข้อต่อ

Action : แรงบิดที่ใช้กับข้อต่อ

Reward : 1 ที่หากหุ่นยนต์เคลื่อนไหวเมื่อตั้งตรงขึ้นและเคลื่อนที่ไปทางหน้า

Atari Games



Objective: Complete the game with the highest score

State: Raw pixel inputs of the game state

Action: Game controls e.g. Left, Right, Up, Down

Reward: Score increase/decrease at each time step

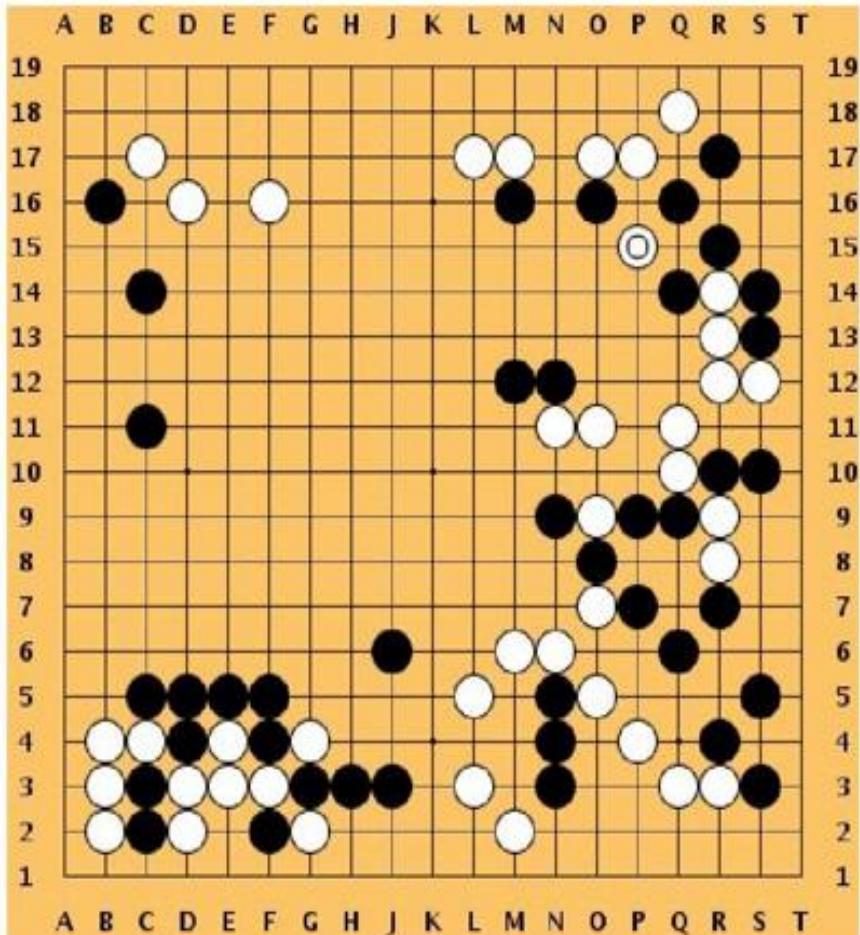
Obj : លោក្រួចណែនាំរាយការណ៍នេនស្តុង្នុត

State : ឯកសារលម្អិតទៅលក្ខណៈបែបរាយការណ៍នេន

Action : ការគូបគុមណែនាំ ថែង ច័យ ឬវាទី ឬឯកសារ

Reward : គេនេនដំឡើ/លទ្ធផលនៃពូនិនិត្យ

Go



Objective: Win the game!

State: Position of all pieces

Action: Where to put the next piece down

Reward: 1 if win at the end of the game, 0 otherwise

Obj : ຜະນະເກມ

State : ຕຳແໜ່ງນໍາລູອງຈົ່ງເກມທີ່ໜີມດ

Action : ທີ່ຈະວາງຈົ່ງເຄີດໄປລົງ

Reward : 1 ອີ້າຜະນະທີ່ສູງຂອງເກມ 0 ມີລະຫັ້ນ

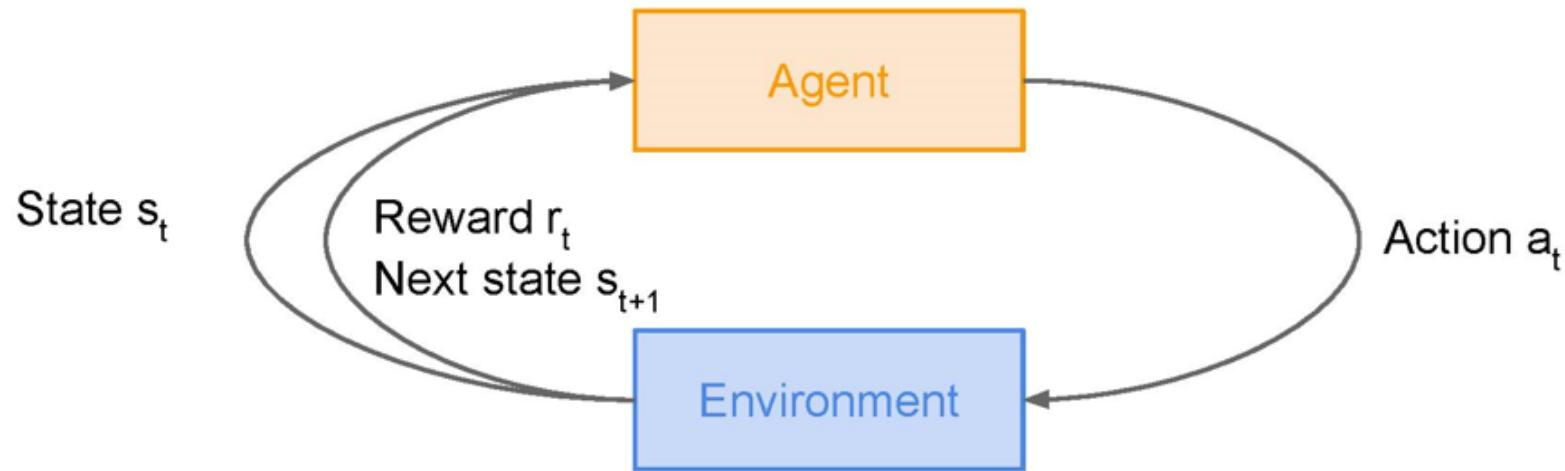


[OpenAI Five: Dota Gameplay](#)



I tried to make a Valorant AI using computer vision

How can we mathematically formalize the RL problem?



Markov Decision Process

ករាបវណ្ណាពាណិជ្ជកម្មនៃលទ្ធផលនៃ Markov

- Mathematical formulation of the RL problem
- **Markov property:** Current state completely characterises the state of the world

ការស្ថិតិសាលាលទ្ធផលនៃប័ណ្ណានៅក្នុងសេវា

គុណភាពបំពើនៃ Markov: ឥឡូវនេះបានបង្ហាញថាអ្នកអាបាសមួយនៅក្នុងសេវាដែលមានការផ្តល់សារដល់ខ្លួន

Defined by: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{P}, \gamma)$

\mathcal{S} : set of possible states

\mathcal{A} : set of possible actions

\mathcal{R} : distribution of reward given (state, action) pair

\mathbb{P} : transition probability i.e. distribution over next state given (state, action) pair

γ : discount factor

ចែងចាយនៃសារីប័ណ្ណានៅក្នុងសេវា

ចែងចាយនៃការការណ៍ការងារក្នុងសេវា

ការការណ៍ការងារក្នុងសេវាដែលត្រូវបានបង្ហាញ

គោលនៃការបង្ហាញនៃការការណ៍ការងារក្នុងសេវាដែលត្រូវបានបង្ហាញ

តាមប័ណ្ណានៅក្នុងសេវាដែលត្រូវបានបង្ហាញ

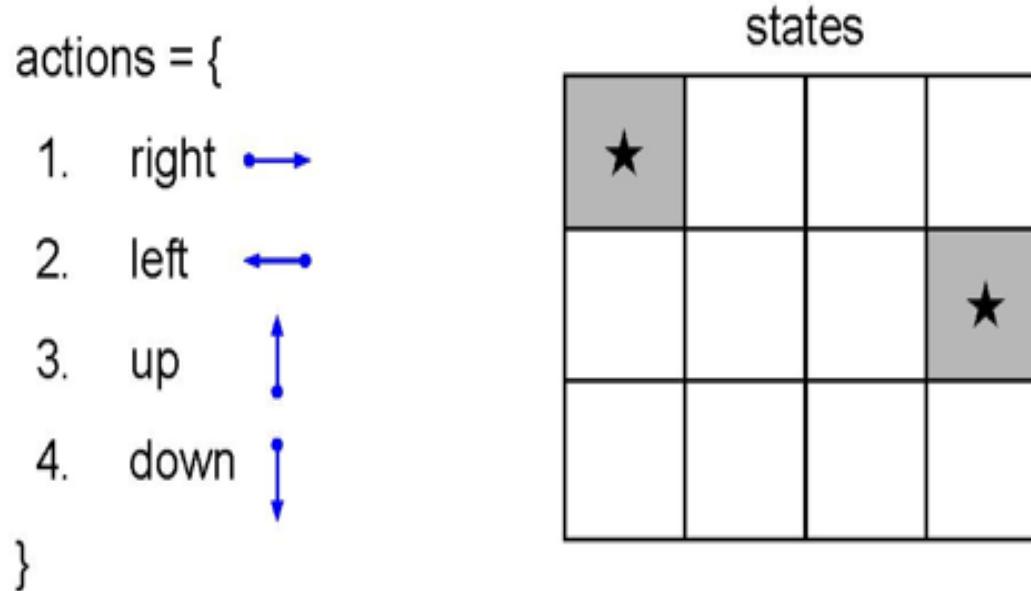
Markov Decision Process

- At time step $t=0$, environment samples initial state $s_0 \sim p(s_0)$
- Then, for $t=0$ until done:
 - Agent selects action a_t
 - Environment samples reward $r_t \sim R(\cdot | s_t, a_t)$
 - Environment samples next state $s_{t+1} \sim P(\cdot | s_t, a_t)$
 - Agent receives reward r_t and next state s_{t+1}
- In each time step, the agent observes the current state s_t and takes an action a_t . The environment returns a reward r_t and the next state s_{t+1} .
- A policy π is a function from S to A that specifies what action to take in each state.
- **Objective:** find policy π^* that maximizes cumulative discounted reward:
$$\sum_{t \geq 0} \gamma^t r_t$$

นโยบายเป็นฟังก์ชันจาก S ไปยัง A ซึ่งระบุว่าจะดำเนินการอย่างไรในแต่ละสถานะ ด้านนโยบายที่สูงสุดในรางวัลรวมที่ได้รับล่วงหน้า

A simple MDP: Grid World

กำหนด "ราชวัล" เป็นค่าตอบสำหรับการเปลี่ยนสถานะแต่ละครั้ง

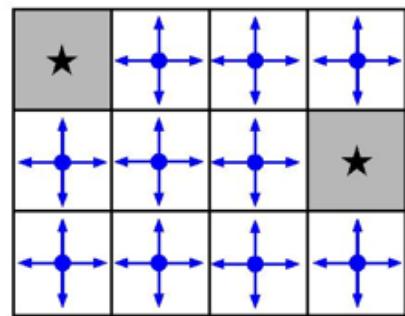


Set a negative “reward”
for each transition
(e.g. $r = -1$)

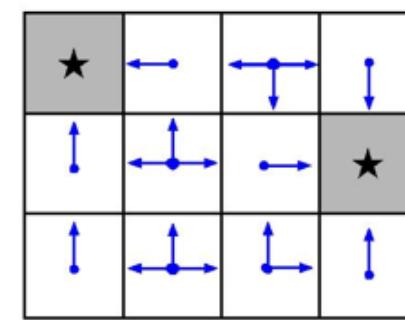
Objective: reach one of terminal states (greyed out) in
least number of actions

Obj : ให้เรียนรู้ใช้ชั้นนี้ในสถานะเป้าหมาย (ເທອງມີນັລສເຕັ້ງທີ່ເປັນລືເກາ) ດ້ວຍຈຳນວນດາວກະກະທີ່ນ້ອຍທີ່ສູດ

A simple MDP: Grid World



Random Policy



Optimal Policy

The optimal policy π^*

We want to find optimal policy π^* that maximizes the sum of rewards.

How do we handle the randomness (initial state, transition probability...)?

นโยบายที่เหมาะสม

เราต้องการหานโยบายที่เหมาะสมที่สุดที่มีประสิทธิภาพในการสะสมรางวัลรวม

เราจะจัดการกับความลุ่ม (สถานะเริ่มต้น, ความน่าจะเป็นในการเปลี่ยนสถานะ) อย่างไร

The optimal policy π^*

We want to find optimal policy π^* that maximizes the sum of rewards.

How do we handle the randomness (initial state, transition probability...)?
Maximize the **expected sum of rewards!**

Formally: $\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | \pi \right]$ with $s_0 \sim p(s_0), a_t \sim \pi(\cdot | s_t), s_{t+1} \sim p(\cdot | s_t, a_t)$

นโยบายที่เน้นกำไร

เราต้องการหานโยบายที่เน้นกำไรที่สุดที่มีประสิทธิภาพในการลงทะเบียนรางวัลรวม

เราจะจัดการกับความลับ (สถานะเริ่มต้น, ความน่าจะเป็นในการเปลี่ยนสถานะ) อย่างไร?

สูงสุดค่าเดียวของผลรวมของรางวัล

Definitions: Value function and Q-value function

Following a policy produces sample trajectories (or paths) $s_0, a_0, r_0, s_1, a_1, r_1, \dots$

How good is a state?

The **value function** at state s , is the expected cumulative reward from following the policy from state s :

$$V^\pi(s) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, \pi \right]$$

คำนิยาม: ฟังก์ชันค่าและฟังก์ชันค่า Q

การตามนโยบายจะสร้างเส้นทางต่ออย่าง (หรือเส้นทาง)

ความดีของสถานะคืออย่างไร?

ฟังก์ชันค่าที่สถานะ s , คือค่าเดียวของรางวัลรวมจากการตามนโยบายจากสถานะ s :

How good is a state-action pair?

The **Q-value function** at state s and action a , is the expected cumulative reward from taking action a in state s and then following the policy:

$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi \right]$$

ความดีของคู่สถานะ-การกระทำเป็นอย่างไร?

ฟังก์ชันค่า Q ที่สถานะและการกระทำคือค่าดูของการหักห้ามของรางวัลรวมจากการดำเนินการในสถานะเดิมตามนโยบาย

Bellman equation

The optimal Q-value function Q^* is the maximum expected cumulative reward achievable from a given (state, action) pair:

$$Q^*(s, a) = \max_{\pi} \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi \right]$$

ສະດາຣເບລ່ລ໌ແມ່ນ

ຝຶກໍ່ຮັນຄ່າ Q ທີ່ເໜີມະສົມ Q^* ເປີນຄ່າຄາດໜວ້າສູງສຸດຂອງຮາງວັດຮອມທີ່ເປີນໄປໄດ້ຈາກຄູ່ (ສຖານະ, ດາວກຮະໜຳ) ທີ່ກຳນົດໃໝ່

Stop !!!

Q^* satisfies the following **Bellman equation**:

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]$$

Intuition: if the optimal state-action values for the next time-step $Q^*(s', a')$ are known, then the optimal strategy is to take the action that maximizes the expected value of $r + \gamma Q^*(s', a')$

Q^* ประสบความสำเร็จในสมการเบลล์เม่นต่อไปนี้

ความคิดอย่างง่าย: หากค่าความคาดหวังสูงสุดสำหรับค่าสถานะ-การกระทำในขั้นตอนต่อไป Q^* รู้อยู่แล้ว ได้ยกตัวอย่างว่าการกระทำการที่ดีที่สุดคือการเลือกการกระทำการที่ทำให้ค่าความคาดหวังมากที่สุดของค่า

The optimal policy π^* corresponds to taking the best action in any state as specified by Q^*

นโยบายที่เหมาะสมสอดคล้องกับการดำเนินการที่ดีที่สุดในสถานะใด ๆ ตามที่ระบุโดย Q^*

Solving for the optimal policy

การแก้ปัญหาจ่องใจอย่างที่เหมาะสม

Value iteration algorithm: Use Bellman equation as an iterative update

อัลกอริทึมการແບ່ງຄ່າຄູ່ລັກຈະນະ: ໃຊ້ສ່ວນເບລື້ມ
ແມນແປ່ນການປັບປຸງແບບຮະບບເອັດລາຍ

$$Q_{i+1}(s, a) = \mathbb{E} \left[r + \gamma \max_{a'} Q_i(s', a') | s, a \right]$$

Q_i will converge to Q^* as $i \rightarrow \infty$

Q_i จะคงเป็น Q^* เมื่อ $i \rightarrow \infty$

What's the problem with this? ไม่สามารถวิเคราะห์ได้ เนื่องจากต้องคำนวณ $Q(r, s)$ สำหรับทุกคู่สถานะ-การณ์ หากสถานะเป็นตัวอ่อนๆ เช่น รูปภาพ
ปุ่มจุบันของเกม เราจะไม่สามารถคำนวณให้ทั้งหมดที่คู่ของสถานะทั้งหมดได้โดยที่มีประสิทธิภาพทางคณิตศาสตร์!
Not scalable. Must compute $Q(s, a)$ for every state-action pair. If state is e.g. current game state pixels, computationally infeasible to compute for entire state space!

Solution: use a function approximator to estimate $Q(s, a)$. E.g. a neural network!

ວິທີแก้ปัญหา: ໃຊ້ຕົວປະມານເຟັກສົນເພື່ອປະມານຄ່າ Q ເຊັ່ນ ໂຄງຈ່າຍປະສາກເຫຼີມ

การแก้ปัญหาของนโยบายที่เหมาะสม

Solving for the optimal policy: Q-learning

ใช้เครื่องมือประมาณค่าทางฟังก์ชันเพื่อประมาณค่าฟังก์ชันค่าการกระทำ

Q-learning: Use a function approximator to estimate the action-value function

$$Q(s, a; \theta) \approx Q^*(s, a)$$

function parameters (weights)

If the function approximator is a deep neural network => **deep q-learning!**

หากเครื่องมือประมาณค่าเป็นเครื่องจ่ายประสาทเชิงลึก => Deep Q-learning

Solving for the optimal policy: Q-learning

Remember: want to find a Q-function that satisfies the Bellman Equation: จำไว้: ต้องการหาฟังก์ชัน Q ที่สอดคล้องกับสมการเบล์ล์เมน

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]$$

Forward Pass ผ่านไปข้างหน้า

$$\text{Loss function: } L_i(\theta_i) = \mathbb{E}_{s, a \sim \rho(\cdot)} \left[(y_i - Q(s, a; \theta_i))^2 \right]$$

$$\text{where } y_i = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a \right]$$

Backward Pass ผ่านย้อนกลับ

Gradient update (with respect to Q-function parameters θ):

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s, a \sim \rho(\cdot); s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i) \right] \nabla_{\theta_i} Q(s, a; \theta_i)$$

การอัปเดตเกรเดียนต์ (เดียวกับพารามิเตอร์ของฟังก์ชัน Q)

Iteratively try to make the Q-value close to the target value (y_i) it should have, if Q-function corresponds to optimal Q^* (and optimal policy π^*)

พยายามอย่างต่อเนื่องให้ค่า Q ใกล้เคียงกับค่าเป้าหมายที่ควรจะ หากฟังก์ชัน Q สอดคล้อง กับ Q^* ที่เหมาะสม (และ นโยบายที่เหมาะสม)

วันนี้เราจะมาลองเขียนโปรแกรมอย่างง่ายให้ AI ของเรารีเล่นเกมส์กัน และอัลกอริทึมที่เราจะใช้กันก็คือ Q-Learning ซึ่งนำเอาสมการทางคณิตศาสตร์ที่เรียกว่า Bellman Equation มาช่วย

$$Q(s,a) = r + \gamma \max(Q(s',a'))$$

พูดถึง Q-Learning อย่างง่ายๆ ก็คือการที่เรามีตารางที่ค่อยบวกกว่า ถ้าเราอยู่ state (s) นี้ เราทำ action (a) นี้ แล้วเล่นอย่างดีที่สุดจนจบเกมส์ จะได้ reward (r) เท่าไหร่ โดยมีการ discount reward (γ)

$$Q^\pi(s_t, a_t) = \underline{E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t, a_t]}$$

Q value for that state given that action

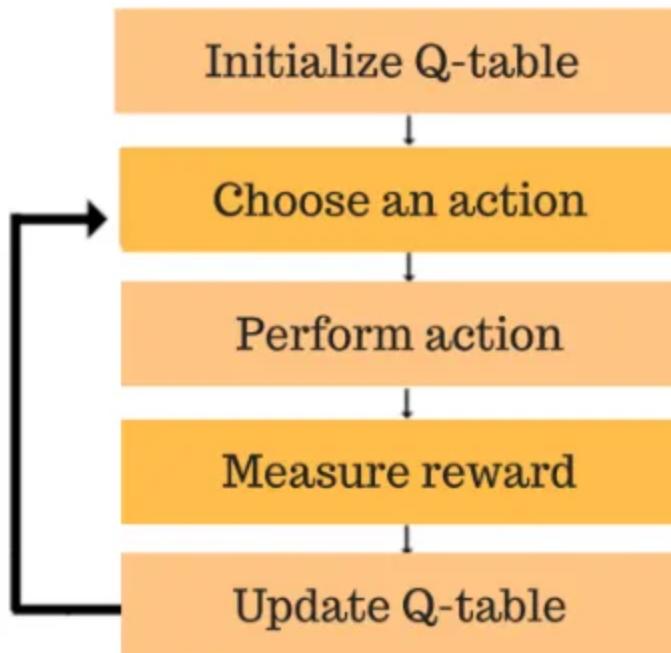
Expected discounted cumulative reward ...

given that state and that action

Thanks to [Thomas Simonini](#)

Simple Q-learning algorithm Process

เนื้องจากเป็น simple q-learning ขอ implement ในรูปแบบที่ เรายังจำนวน state และ action ทั้งหมด เพื่อให้ง่ายต่อการเรียนรู้



Step 1: Initialize Q-table

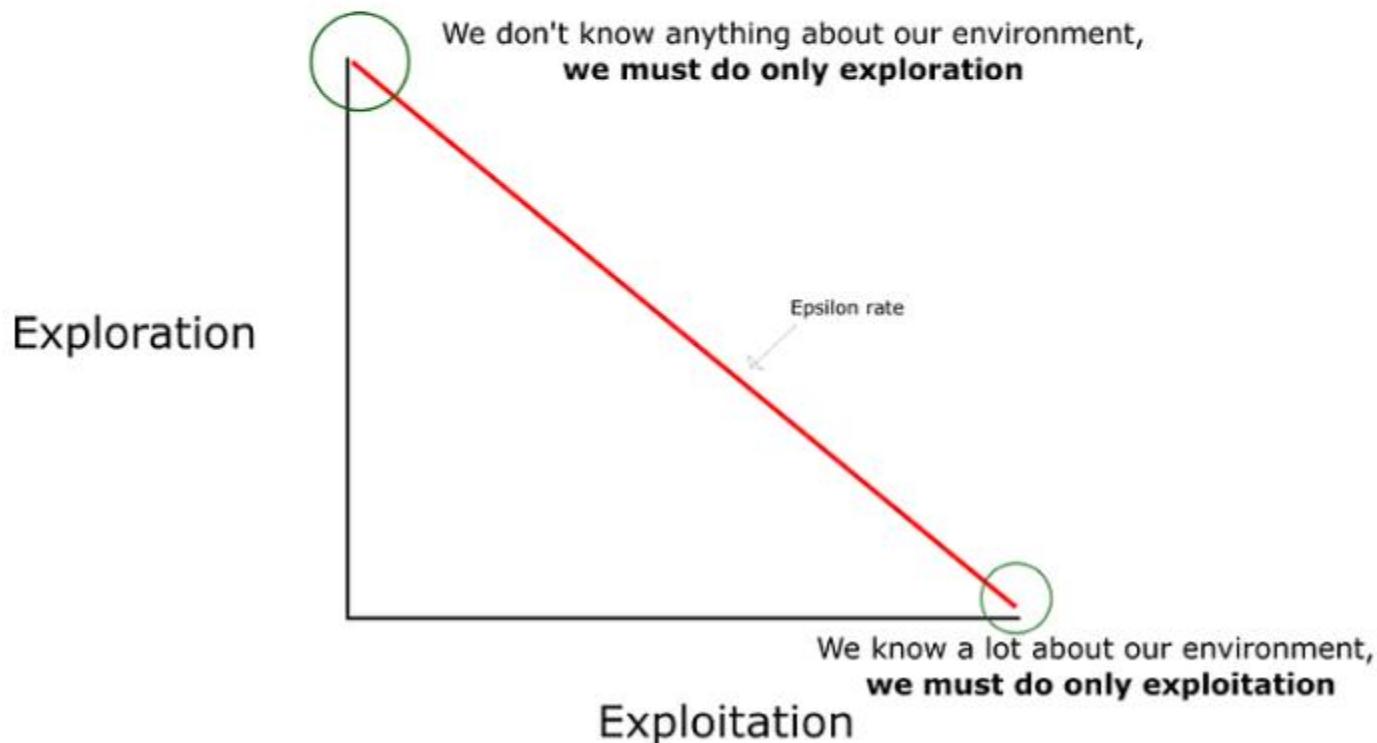
สำหรับขั้นแรก ก็คือการสร้าง ตารางที่มีขนาดเท่ากับ จำนวน n และ ($n =$ จำนวน state ที่เป็นไปได้) และ m หลัก ($m =$ จำนวน action ที่เป็นไปได้) ซึ่งเราจะให้ค่าทั้งหมด เป็น 0

		Actions			
		0	0	0	0
States	0	0	0	0	0
	0	0	0	0	0
...					
		0	0	0	0

Step 2: Choose an action

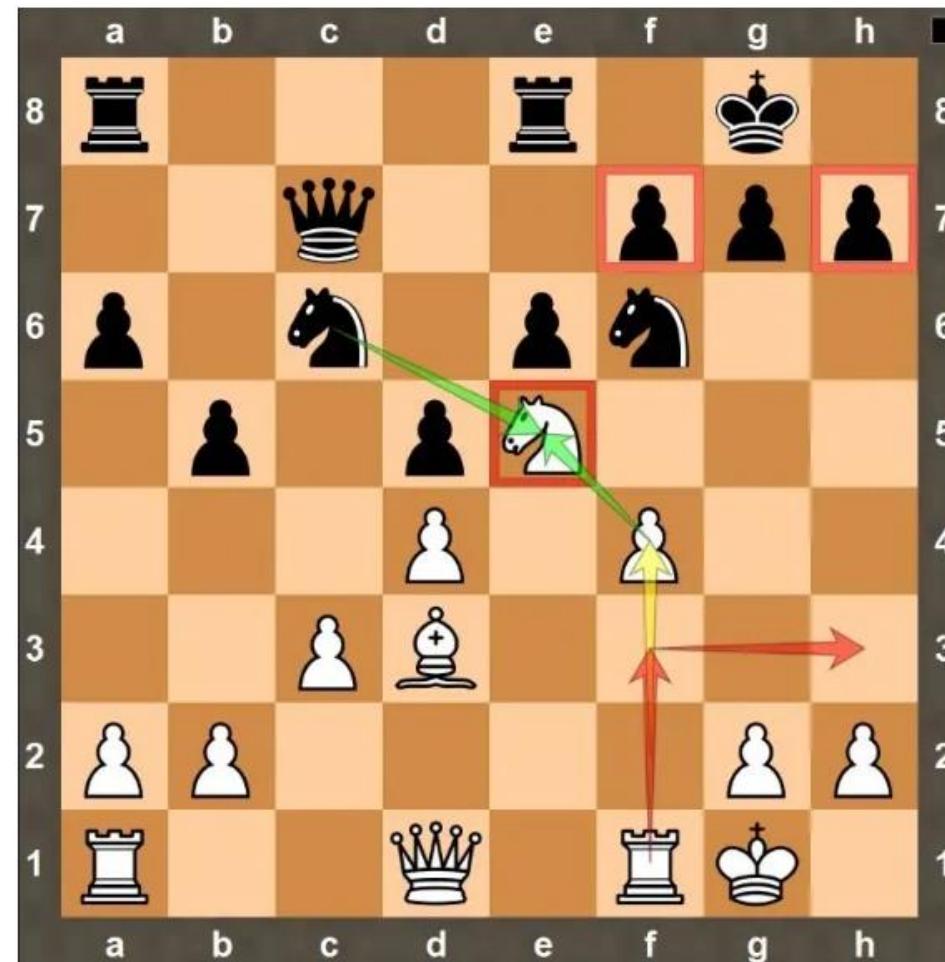
ก่อนที่เราจะเลือกว่าจะทำ action อะไร ส่าหรับ state ต่างๆ นั้นก็จะทำให้เกิดคำถามว่า

- เริ่มต้นเราเริ่มจากไม่รู้อะไรเลย เราจะรู้ได้ยังไงว่า เลือก action ไหนแล้วจะดี ?
- แล้วถ้าเรารู้ว่า action ไหนได้ผลลัพธ์ดีในระดับนึงแล้ว เราคิดว่ามันดีที่สุดรึยังจาก state ตอนนี้ ?



ทำให้เกิดสิ่งที่จะมาช่วยเราก็คือ Epsilon greedy โดยหน้าที่ของมันก็คือการบอกร่วมกับเราครรตัดสินใจเลือกรห่วง การค้นหารูปแบบใหม่ (explore) หรือจะเลือกทำแบบที่คิดว่าดีอยู่แล้ว (exploit)

ส่วนมากเขามักจะให้ค่าของ epsilon อยู่ในช่วง 0-1 โดย 1 หมายถึง ให้ explore 100% และยิ่งมีค่าน้อยลงก็หมายถึง explore ให้น้อยลงนั่นเอง



Step 3-4: Perform action & Measure reward

หลังเราเลือกแล้วว่า state นี้ เราจะ action อะไร เรา ก็นำสิ่งที่เราเลือกไปใช้กับ environment ซึ่ง environment ก็จะบอกเราว่า หลังจากทำ action นี้ เราจะไปอยู่ที่ state 'ใหม่ต่อไป' (s') และเราได้ reward (r) มาเท่าไหร่หลังจากทำ action นั้น

