First Step to Practical Machine Learning

Knowledge Sharing for CPE/SKE students

Sirakorn Lamyai

Student, Kasetsart U.

October 30, 2018

Before we start...

Make sure these are installed on your computer.

This page is a guide for installing on Windows

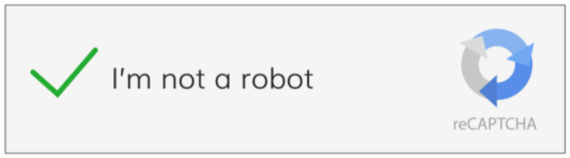
- Python 3.6: Download and install at `https://www.python.org`
- NumPy, Scipy, Matplotlib, Scikit-learn, MLxtend:
  Run `pip install numpy scipy matplotlib sklearn mlxtend`

Outline

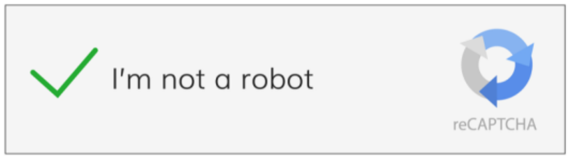
Introduction to Machine Learning

What is Machine Learning?

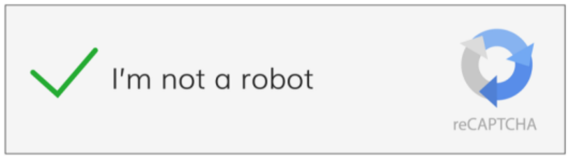
What is Machine Learning?



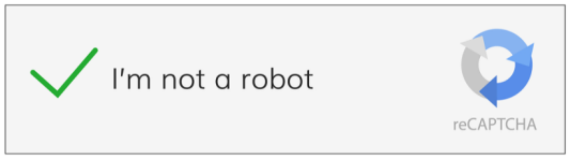
What is Machine Learning?



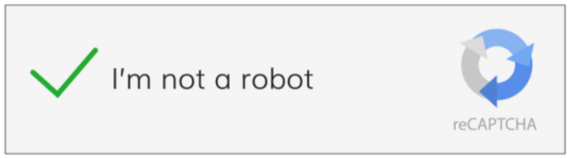
- This is Recaptcha.



- This is Recaptcha.
  - Recaptcha helps stop millions of spam a day.

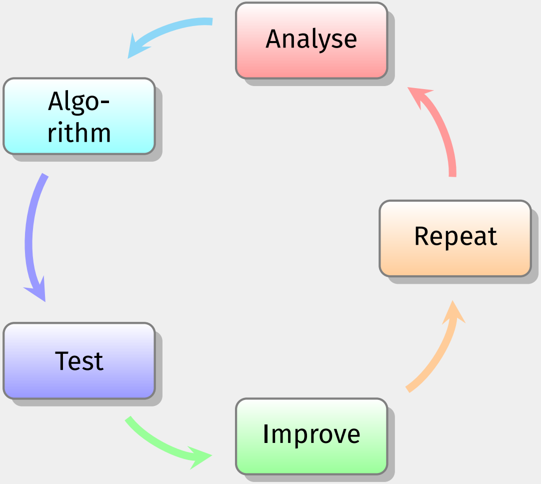


- This is Recaptcha.
  - Recaptcha helps stop millions of spam a day.
  - In some old days, we have to type Captcha texts to distinguish ourself from bots.

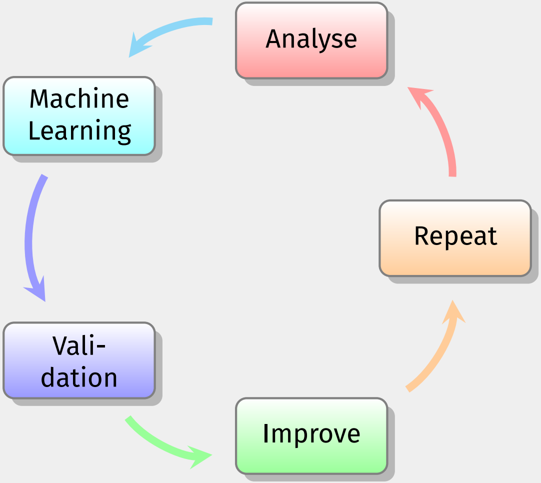


- This is Recaptcha.
  - Recaptcha helps stop millions of spam a day.
  - In some old days, we have to type Captcha texts to distinguish ourself from bots.
  - How is it possible that with a single click, an automated system can distinguish bots from humans?

Traditional programming approach



Machine learning approach



In other words...

Machine Learning

In other words...

Machine Learning

= Data + Data analysis algorithm

Machine Learning

= Data + Data analysis algorithm

= Adapt to change

Machine Learning Problems

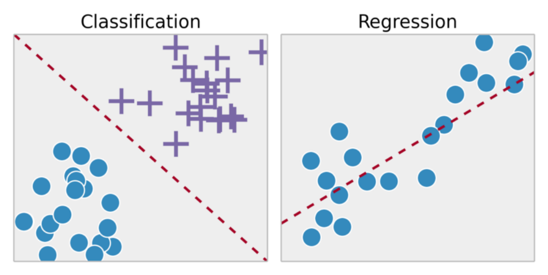
Types of Machine Learning problems

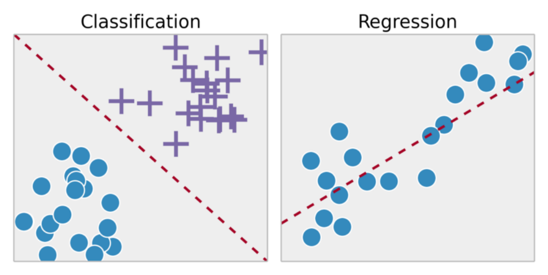
1. Supervised learning

Types of Machine Learning problems

1. Supervised learning
2. Unsupervised learning

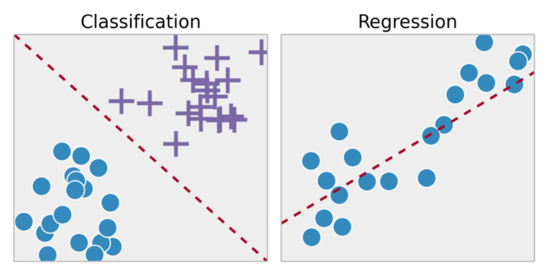
1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning


Classification
Regression

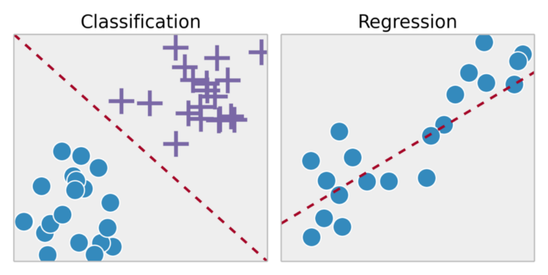


- Given a **training set** for the data, find a **model** to **generalise** well to **unseen** data.

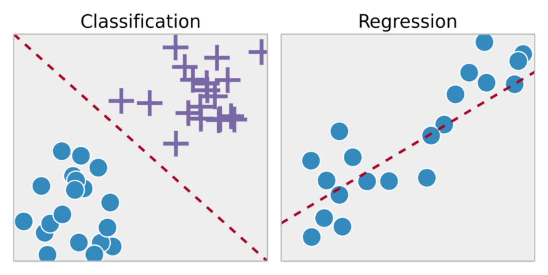
Supervised learning



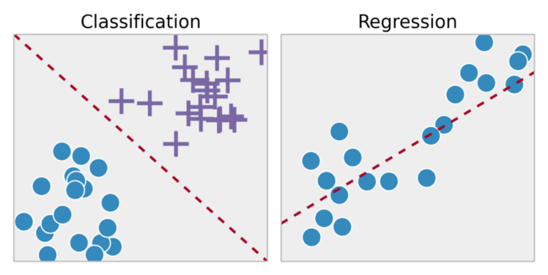
- Given a **training set** for the data, find a **model** to **generalise** well to **unseen** data.
- Two main supervised learning problems



- Given a **training set** for the data, find a **model** to **generalise** well to **unseen** data.
- Two main supervised learning problems
  - Classification: On the discrete data

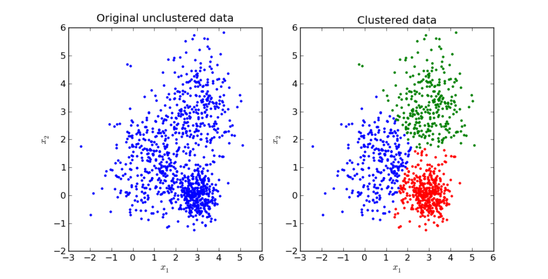


- Given a **training set** for the data, find a **model** to **generalise** well to **unseen** data.
- Two main supervised learning problems
  - Classification: On the discrete data
  - Regression: On the continuous data

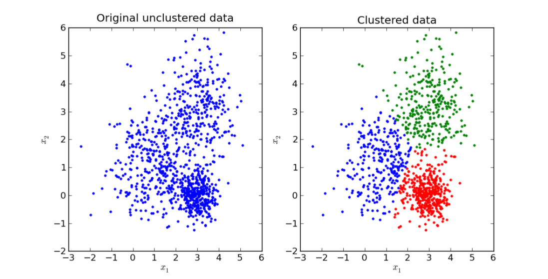


- Given a **training set** for the data, find a **model** to **generalise** well to **unseen** data.
- Two main supervised learning problems
  - Classification: On the discrete data
  - Regression: On the continuous data
- Example problems: Spam E-mail detection, Facial recognition

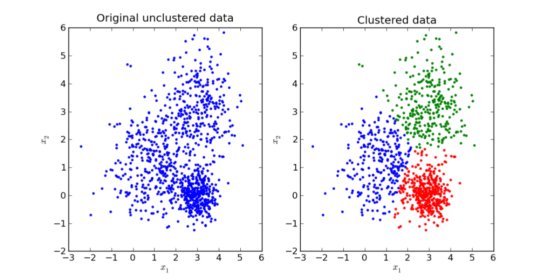
Unsupervised learning



Unsupervised learning

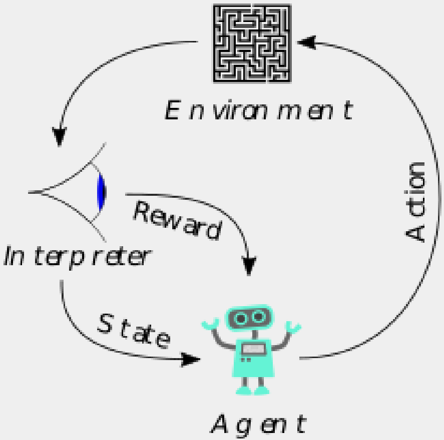


- Discover **hidden** structure in **non-labelled** data.



- Discover **hidden** structure in **non-labelled** data.
- Example: Clustering, Generative models

Reinforcement learning



Model

Model

Model

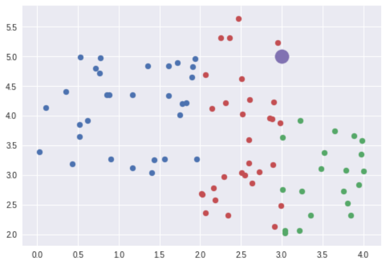
- A result of the combination between...

- A result of the combination between...
  - a **method** to recognise the data, and

Model

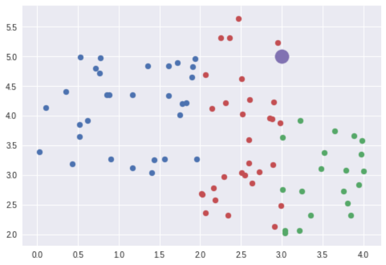
- A result of the combination between...
  - a **method** to recognise the data, and
  - **sample datas** for such the method

- A result of the combination between...
  - a **method** to recognise the data, and
  - **sample datas** for such the method



Data

- A result of the combination between...
  - a **method** to recognise the data, and
  - **sample datas** for such the method



Determine which group should the purple dot be in (red/green/blue) by **checking the colour of its nearest dot.**

Data

Method

- We're going to write our **first own** machine learning algorithm called ***k*-Nearest Neighbour** (*k*-NN)

- We're going to write our **first own** machine learning algorithm called ***k*-Nearest Neighbour** (*k*-NN)
  - *k*-NN is known to be very simple, with its concept as

- We're going to write our **first own** machine learning algorithm called k**-Nearest Neighbour** (k-NN)
  - k-NN is known to be very simple, with its concept as

k-NN algorithm

To classify label of a data point, get k nearest data points to the data point, and select the major label among those data points.

Coding time!

Machine Learning Process

Machine Learning Process

- Train

- Train
- Test

Machine Learning Process

- Train
- Test

(There'll be more of this, trust me.)

Choosing the parameter for k-NN algorithm

What is the bad way to choose k?

Choosing the parameter for k-NN algorithm

What is the bad way to choose k?

- What if we choose k = # of all points?

Choosing the parameter for k-NN algorithm

What is the bad way to choose k?

- What if we choose k = # of all points?
    - What will happen if our dataset's got 3 labels of A, B, C with 10, 20, and 30 data points of each?

Choosing the parameter for k-NN algorithm

What is the bad way to choose k?

- What if we choose k = # of all points?
  - What will happen if our dataset's got 3 labels of A, B, C with 10, 20, and 30 data points of each?
  - Answer: Our model will always answer the labels with the highest data point count.

What is the bad way to choose k?

- What if we choose k = # of all points?
  - What will happen if our dataset's got 3 labels of A, B, C with 10, 20, and 30 data points of each?
  - Answer: Our model will always answer the labels with the highest data point count.
- What if we choose $k = 1$?

Choosing the parameter for k-NN algorithm

What is the bad way to choose k?

- What if we choose k = # of all points?
  - What will happen if our dataset's got 3 labels of A, B, C with 10, 20, and 30 data points of each?
  - Answer: Our model will always answer the labels with the highest data point count.
- What if we choose $k = 1$?
  - Let's try!

Coding time!

Training and Testing set

- We separate our dataset into 2 parts: the **training set** and **testing set**

Training and Testing set

- We separate our dataset into 2 parts: the **training set** and **testing set**
  - Most of the time, the testing set will be around 10-25% of the entire dataset

Training and Testing set

- We separate our dataset into 2 parts: the **training set** and **testing set**
  - Most of the time, the testing set will be around 10-25% of the entire dataset
  - What will happen if we train on the testing set?

Training and Testing set

- We separate our dataset into 2 parts: the **training set** and **testing set**
  - Most of the time, the testing set will be around 10-25% of the entire dataset
  - What will happen if we train on the testing set?
  - What will happen if we test on the training set?

- We separate our dataset into 2 parts: the **training set** and **testing set**
    - Most of the time, the testing set will be around 10-25% of the entire dataset
    - What will happen if we train on the testing set?
    - What will happen if we test on the training set?
        - **Cheating!** Like letting the model *remembers* the answer instead of **generalising** the data pattern.

Training and Testing set

- We separate our dataset into 2 parts: the **training set** and **testing set**
  - Most of the time, the testing set will be around 10-25% of the entire dataset
  - What will happen if we train on the testing set?
  - What will happen if we test on the training set?
    - **Cheating!** Like letting the model *remembers* the answer instead of **generalising** the data pattern.
  - In other words, **don't test and train model on the same set of data.**

Choosing the best k

Choosing the best k

- **Train** with the training set, to let our model know how will the data looks like.

Choosing the best k

- **Train** with the training set, to let our model know how will the data looks like.
- **Test** with the testing set, to see on how our model performs.

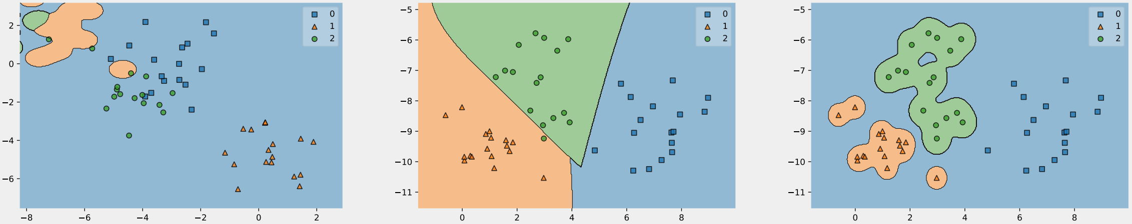
Choosing the best k

- **Train** with the training set, to let our model know how will the data looks like.
- **Test** with the testing set, to see on how our model performs.

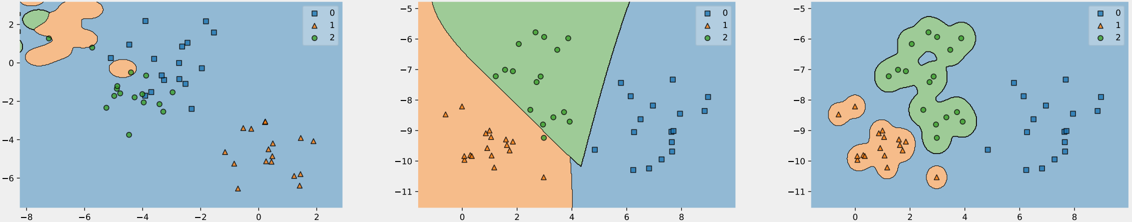
Warning! This is a simplified Machine Learning model training process, there are more to concerns!

Overfitting and underfitting

Which decision region is good?

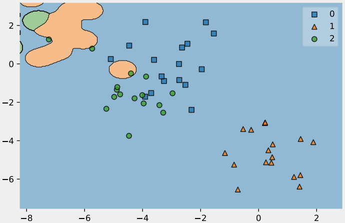


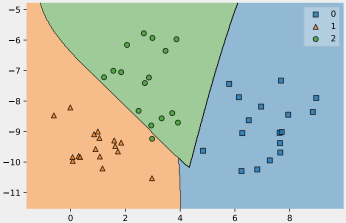
Which decision region is good?

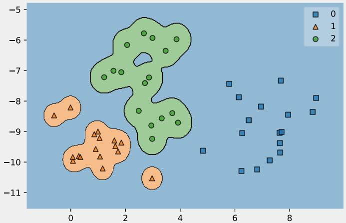


Underfit: The model fails to recognise data pattern

Which decision region is good?





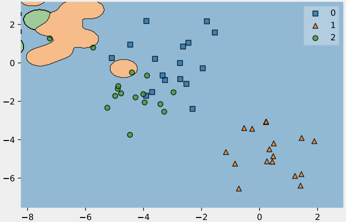


Underfit: The model fails to recognise data pattern

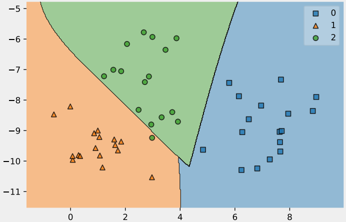
Overfit: The model **remembers** data pattern instead of generalising.

Overfitting and underfitting

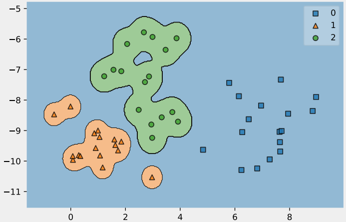
Which decision region is good?



Underfit: The model fails to recognise data pattern



Good fit: The model recognises data pattern **generally**



Overfit: The model **remembers** data pattern instead of generalising.

Good model must **generalise**

- Actually, the key point in k-NN algorithm is choosing k points with the least **distant**.
- What is **distant**?

Norm for k-NN algorithm

Norm

In linear algebra, a **norm** is a function that assigns a strictly positive length or size to each vector in a vector space - except for the zero vector, which is assigned a length of zero.

Given $\vec{x}$ as an N-dimension vector of $\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$

- l_1 Norm: $|x|_1 = \sum_{i=0}^{N} |x_i|$ (Manhattan)
- l_2 Norm: $|x|_2 = \sqrt{\sum_{i=0}^{N} x_i^2}$ (Euclidian)
- l_p Norm: $|x|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ (Minkowski)

Algorithms for Machine Learning Classification Problem

k-NN is a very simple intuition for machine learning algorithms. However, there exists more algorithm that performs well to other problems.
Example algorithms:

- Naïve Bayes
- SVM
- Decision Tree
- Logistic Regression

	Gender	Hair
1	M	Long
2	M	Short
3	F	Long
4	F	Long
5	F	Short

Can we *guess* the gender from hair's length?

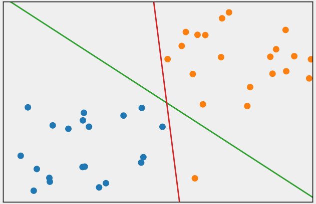
- $P(\text{Male}|\text{Long hair}) = \frac{1}{3}$
- $P(\text{Female}|\text{Long hair}) = \frac{2}{3}$

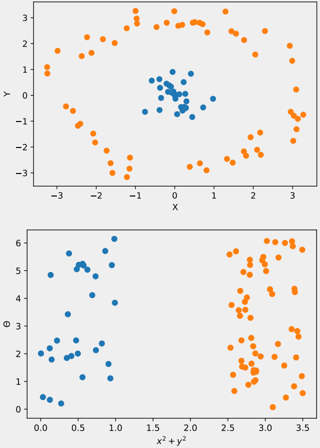
Therefore, we guess that the long-haired person is more likely to be a female.

Bayes Theorem

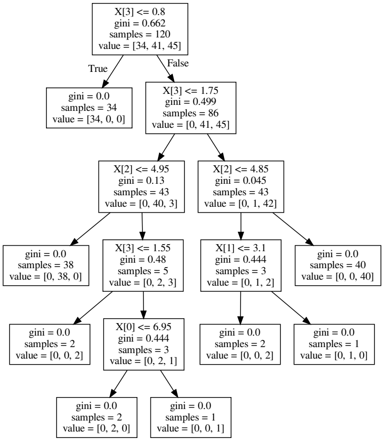
$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

Support Vector Machines (SVM)





- Goal: to draw a line to separate groups of data
- Ideal good line: maximising the distant between the line and classes of data points
- What if the data is not linearly separable? **Kernel tricks**



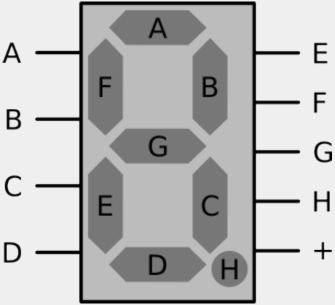
- Creating an if-else conditions automatically
- Nested conditions with a parameter to determine how does the separating of the "tree" performs.

Figure: xkcd - Compiling

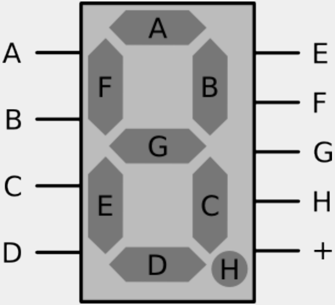
Figure: xkcd - Compiling

Problems for Machine Learning

7-Segment display

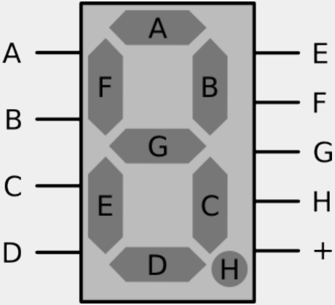


7-Segment display

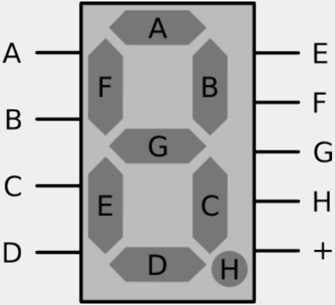


- This is a 7-segment display.

7-Segment display



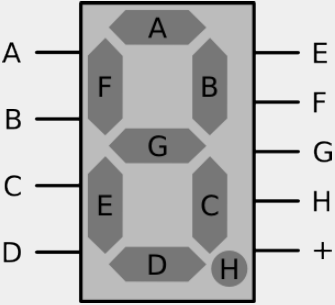
- This is a 7-segment display.
- It consists of a bulb labelled from A-G that could form a number.



- This is a 7-segment display.
- It consists of a bulb labelled from A-G that could form a number.

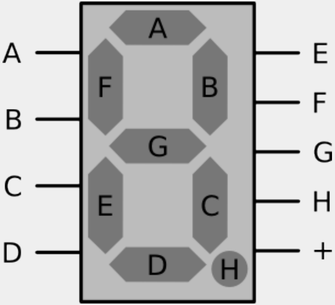
Problem

When the list of the bulb that went on were given, can we determine the number?



Problem

When the list of the bulb that went on were given, can we determine the number?

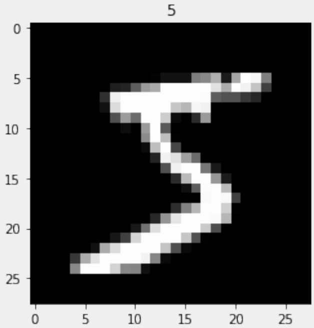


Problem

When the list of the bulb that went on were given, can we determine the number?

Not only yes, but *easily* yes!

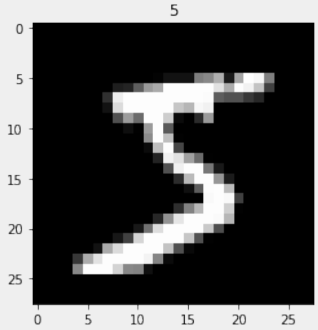
```
if led_on == (b, c):
    return 1
elif led_on == (a, b, g, e, d):
    return 2
...
```



Problem

When the image of the handwriting were given, can we determine the number?

Handwriting



Problem

When the image of the handwriting were given, can we determine the number?

With an **explicit algorithm**? Obviously no! There are too many ways of drawing the number!

MNIST Dataset

MNIST Dataset

- 28*28 pixel images of handwritten numbers (0-9)

- 28*28 pixel images of handwritten numbers (0-9)
- 60,000 training images

MNIST Dataset

- 28*28 pixel images of handwritten numbers (0-9)
- 60,000 training images
- 10,000 testing images

k-NN WITH MNIST

- Training: Pretty fast, no calculations on training phase
- Testing: **thinking**
  - 60,000 data points to calculate the distant + 10,000 data points to test
  - = 600,000,000 calculations to be made
    (this excludes sorting, of which is a $\mathcal{O}(n)$ process)
  - = **(relatively) slow**