MONASH University
Information Technology

# FIT2093 INTRODUCTION TO CYBERSECURITY



## Week 11 Lecture



## Machine Learning / AI for Cybersecurity

**S1 2022**


# Outline

Machine Learning for Cybersecurity

- Machine learning: What is it?
  - types of **learning**
  - types of machine learning **problems**
  - generative vs discriminative **models**
  - *e.g. pattern recognition, feature extraction*


# Outline

Machine Learning for Cybersecurity

- Machine learning for CyberSecurity problems
  - **binary** classification
  - **anomaly** detection
  - **lie** detection
  - **biometrics**
  - **spam** classification
  - **Tasks** in classification
  - Classification **Metrics**
  - **crypto**graphy


# Question

Machine Learning for Cybersecurity

- Q: Give an example scenario where you think Machine Learning/AI might be used to **defend** a system against attacks.


# Question

Machine Learning for Cybersecurity

- Q: Give an example scenario where you think Machine Learning/AI might be used to **attack** a system.



**Activity (5 mins)**

1) Click the latest link in the Zoom chat
2) Add your question response to the Ed forum
3) Add your “hearts” to your favourite responses

MONASH University

# Machine Learning: What is it?


# Machine Learning: What is it?

Machine Learning for Cybersecurity

- Machine learning
  - learn from experience, improve over time
    - **learn** to do what?
    - how to **improve**?
    - with **time**, what **experience** is enhanced?



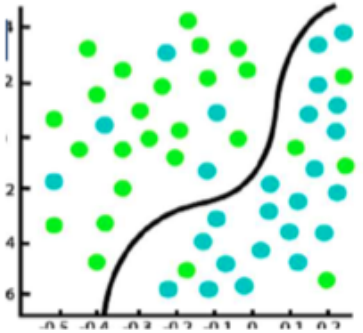
- Includes but is not just optimization
  - max or min some objective function s.t. some constraints

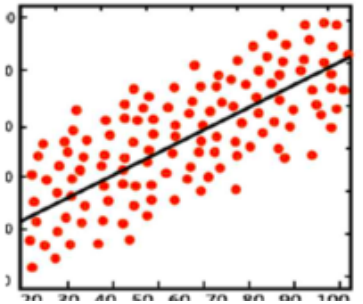
# # Machine Learning: What is it? #

Machine Learning for Cybersecurity


- Machine *learn*ing: learn from *experience*, *improve* over *time*
  - learn to do
    - **classification**: choose 1 of N labels/categories/classes
    - **regression**: predict numerical values





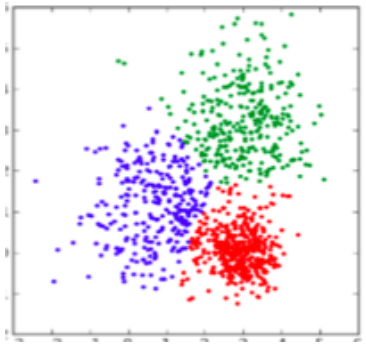


# # Machine Learning: What is it? #

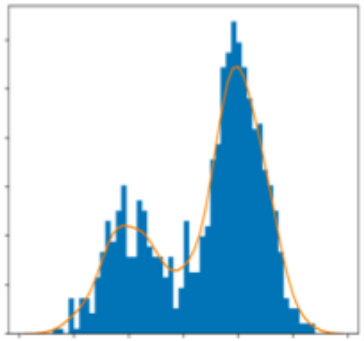
Machine Learning for Cybersecurity

- Machine *learn*ing: learn from *experience*, *improve* over *time*
  - learn to do
    - **clustering**: group similar samples





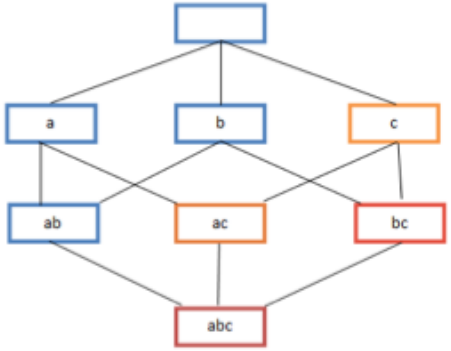
    - **density estimation**: construct the probability distribution of observed samples



# # Machine Learning: What is it? #

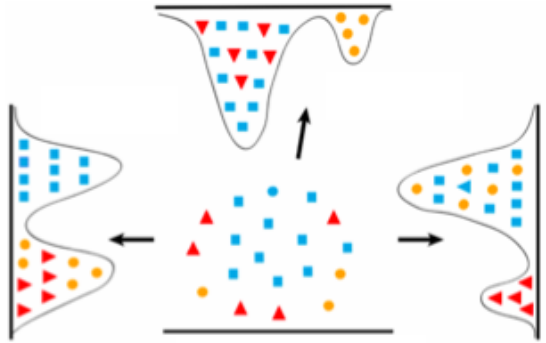
Machine Learning for Cybersecurity

- Machine *learn*ing: learn from *experience*, *improve* over *time*
  - learn to do
    - **association rule** learning: discover relations between samples






    - **dimensionality** reduction / feature learning / representation learning: reduce the dimensions of features in the dataset of samples

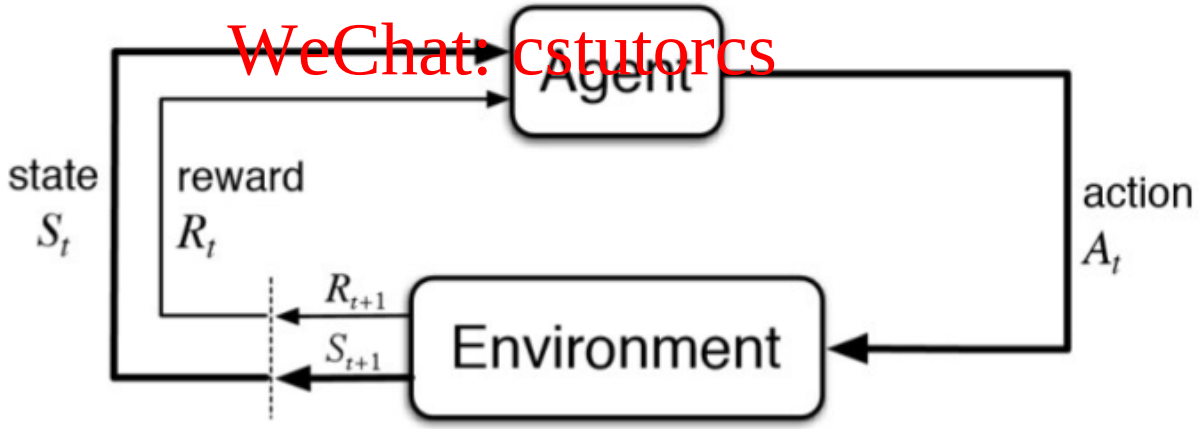


# # Machine Learning: What is it? #

Machine Learning for Cybersecurity


- Machine *learn*ing: learn from *experience*, *improve* over *time*
  - learn to do
    - **reinforcement** learning: game like training to max the reward vs penalty





# Machine Learning: What is it?

Machine Learning for Cybersecurity


- Machine *learn*ing: learn from *experience*, *improve* over *time*
  - how to **improve**?
    - ↑ accuracy of prediction/estimation
    - ↓ difference from desired
  - with **time**, what **experience** is enhanced?
    - see more (labelled) samples e.g. (un)supervised learning
    - more interactions (reinforcement learning)

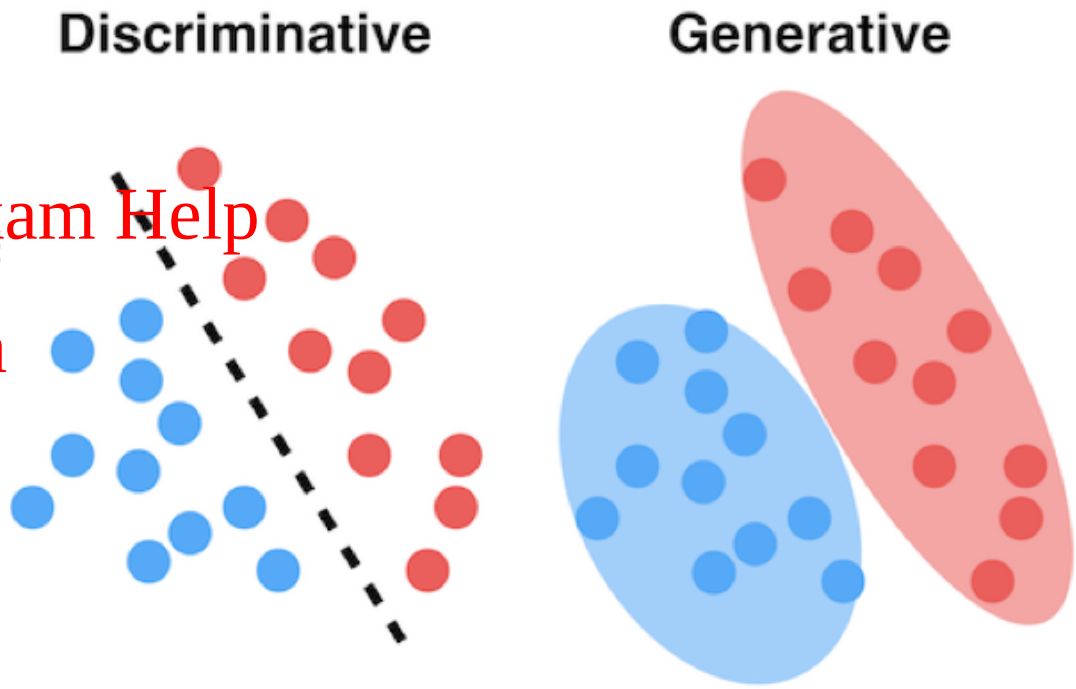
# # Machine Learning: What is  it? #

Machine Learning for Cybersecurity

- **supervised** learning: *we'll focus on this, most common*
  - ground truth samples available with class labels
- **unsupervised** learning
  - samples without labels, no ground truth
- **semi-supervised**: samples with & without labels
- **reinforcement** learning
  - no samples, learn from interactions & corresponding reward/payoff

# # Machine Learning: What is it? #

Machine Learning for Cybersecurity




- Discriminative Model
  - learn the decision **boundary** between classes among samples
  - predict the class label



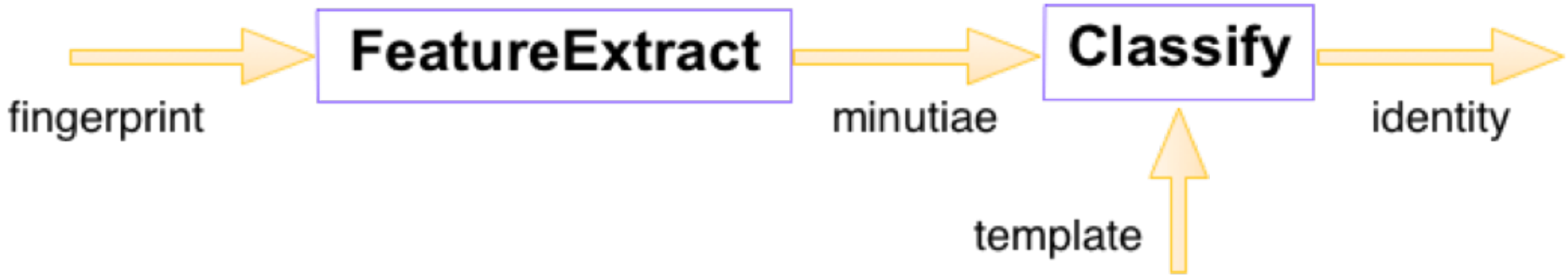
- Generative Model
  - model the underlying **distribution** of the samples
  - can generate new samples from same distribution

# Pattern Recognition ⊂ Machine Learning

Machine Learning for Cybersecurity

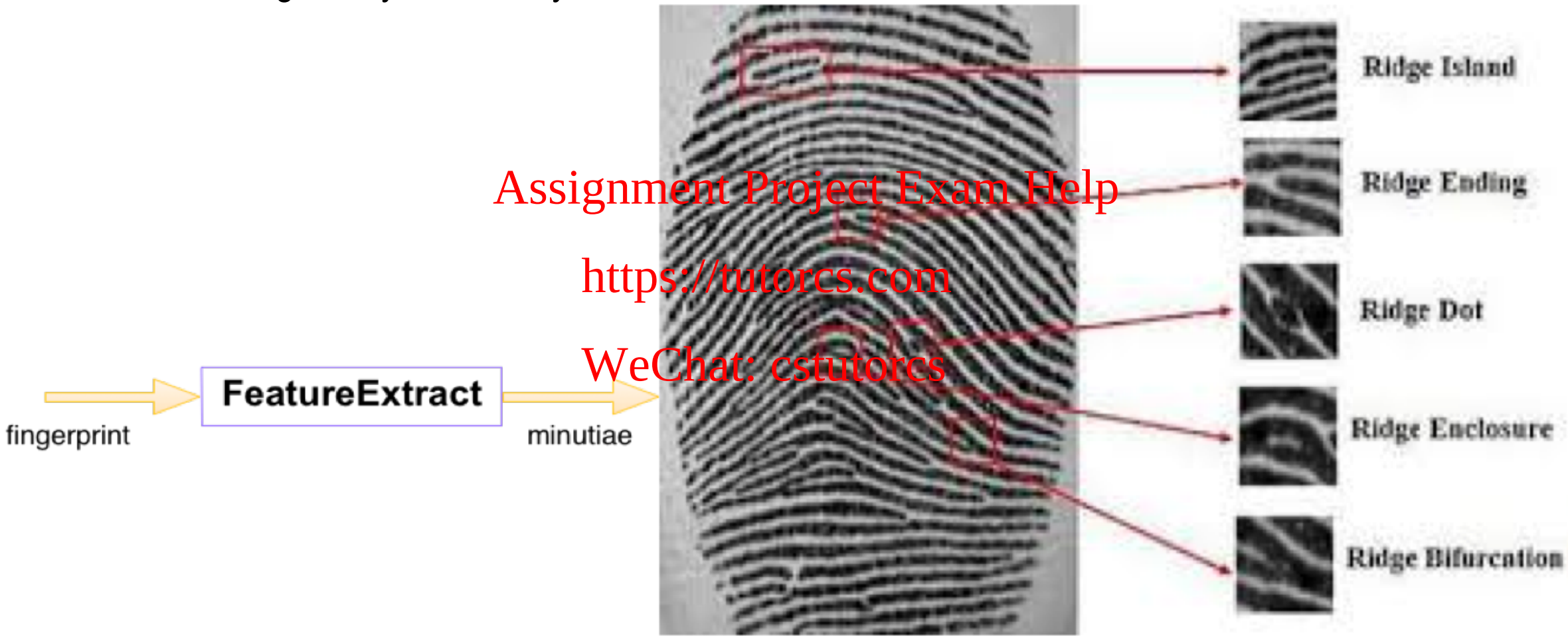

- **pattern recognition = feature learning** then **classification**
  - can treat each stage as a black box, various options
- *e.g. for fingerprint biometrics*
  - *feature = minutiae*





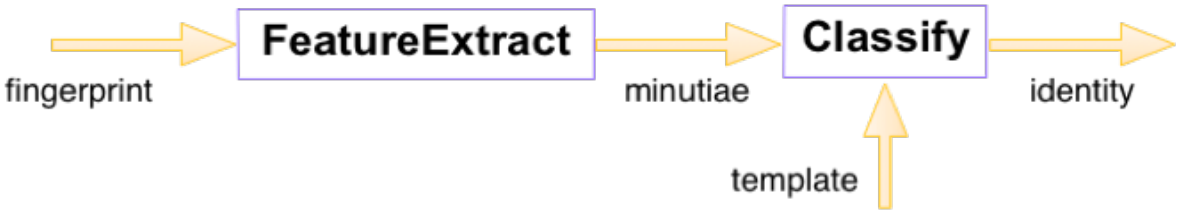
# # Fingerprint Minutiae #

Machine Learning for Cybersecurity



# Pattern Recognition

Machine Learning for Cybersecurity




- biometrics = pattern recognition applied to a security problem

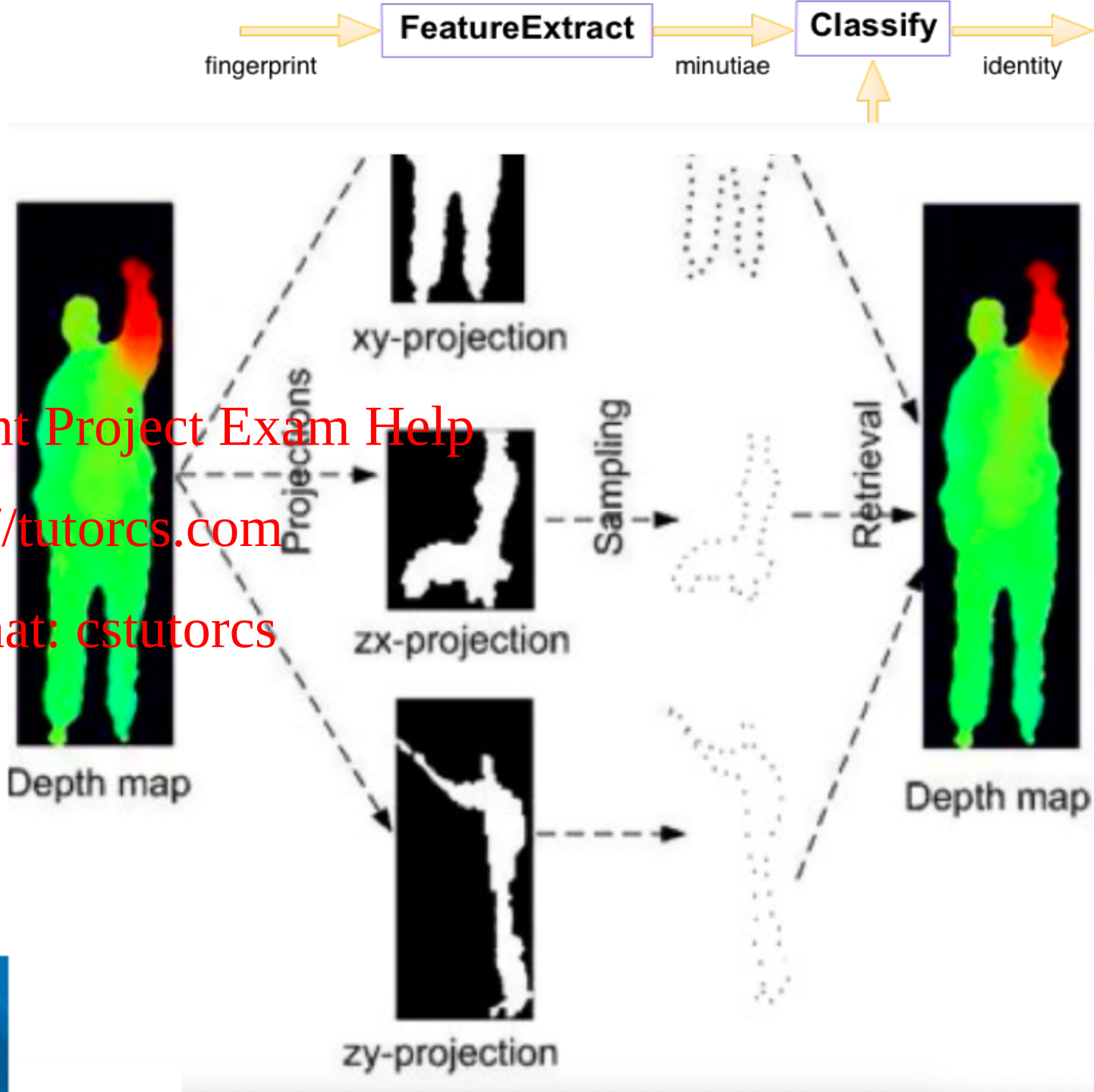


- Q: *Why need to do feature extraction? Why not directly classify?*

# Feature Extraction

Machine Learning for Cybersecurity

- Q: *Why need to do feature extraction? Why not directly classify?*
- *A: this allows to obtain various projection views, features from various viewpoints are suited for different classification problems*
  *e.g. top-down view of a human does not help to predict his/her height, side view is better*



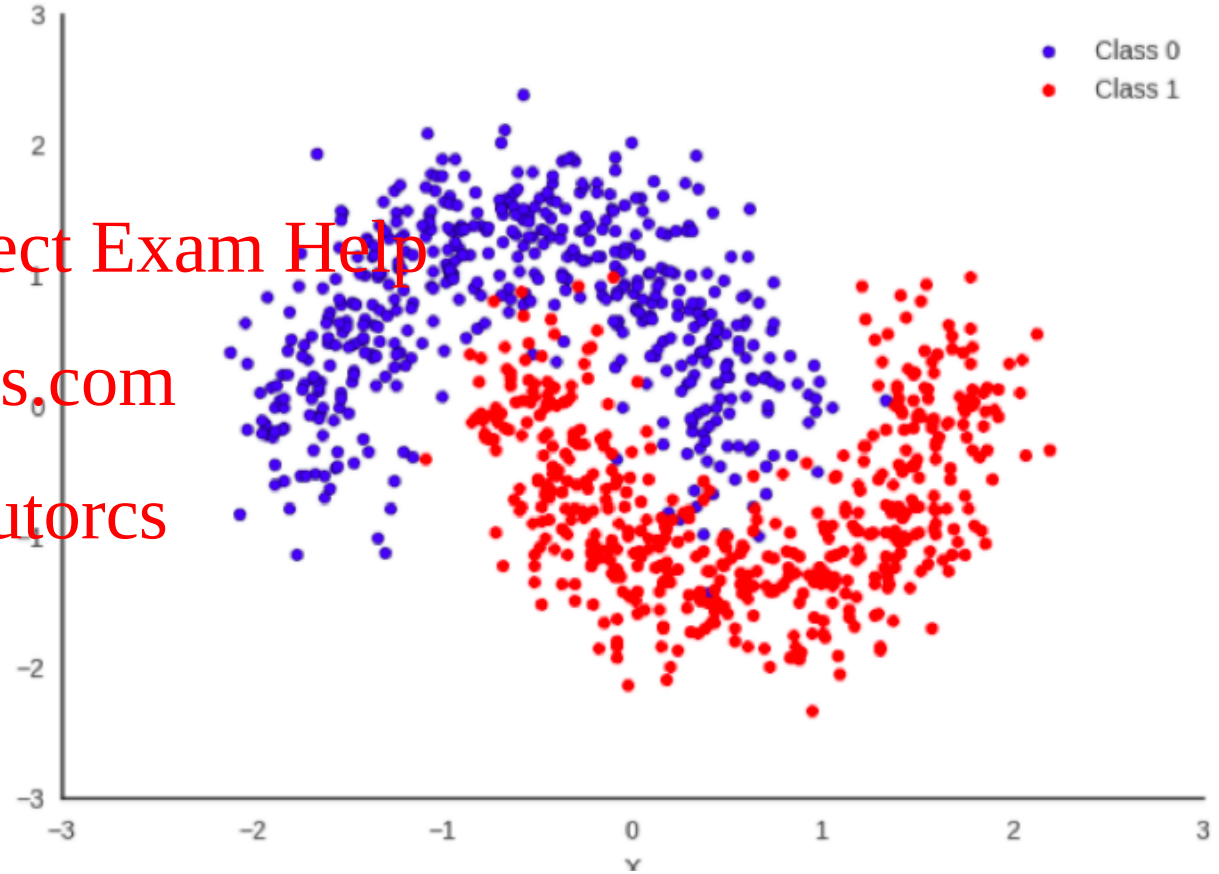
MONASH University

# Machine Learning for CyberSecurity Problems

# Security Problems as Classification Problems

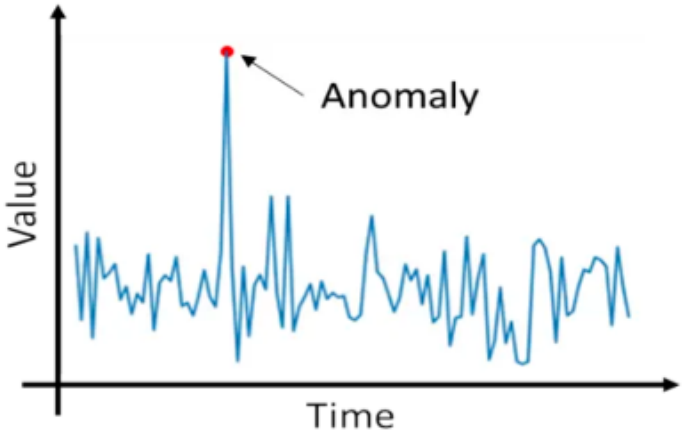
Machine Learning for Cybersecurity

- binary classification
  - 2 classes/categories
    - yes / no
  - involves feature extraction/learning before classification
    - *not everything is useful, some redundant / noise-like*
    - *we only need to focus on some key aspects*



# Anomaly Detection as Binary Classification

Machine Learning for Cybersecurity






- binary classification: 2 classes
  - normal / abnormal
  - benign / malicious traffic/behaviour *e.g. intrusion/malware detection*
  - (I) detect **presence** of known **malicious** signatures
  - (II) detect **absence** of known **normal** behaviour

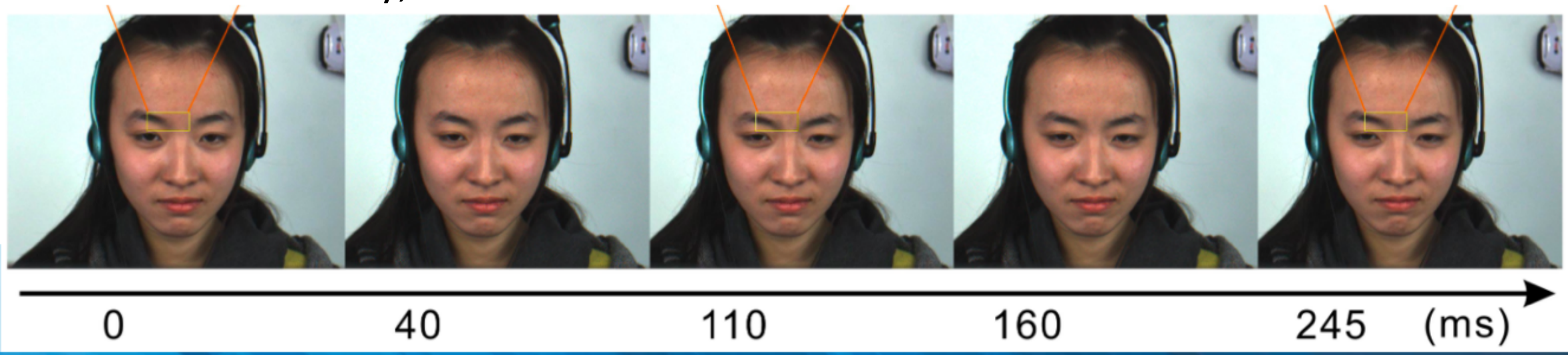
- Q: when is approach (II) better than (I)?
- Q: when will both approaches not work?
- Q: how could an attacker cause both approaches to fail?

# Lie Detection as Binary Classification

Machine Learning for Cybersecurity

- lie detection:
  - innocent vs suspicious
    - e.g. presence of micro-expressions
      - subtle facial movements, fraction of a second
      - involuntary, sub-conscious reaction to emotion

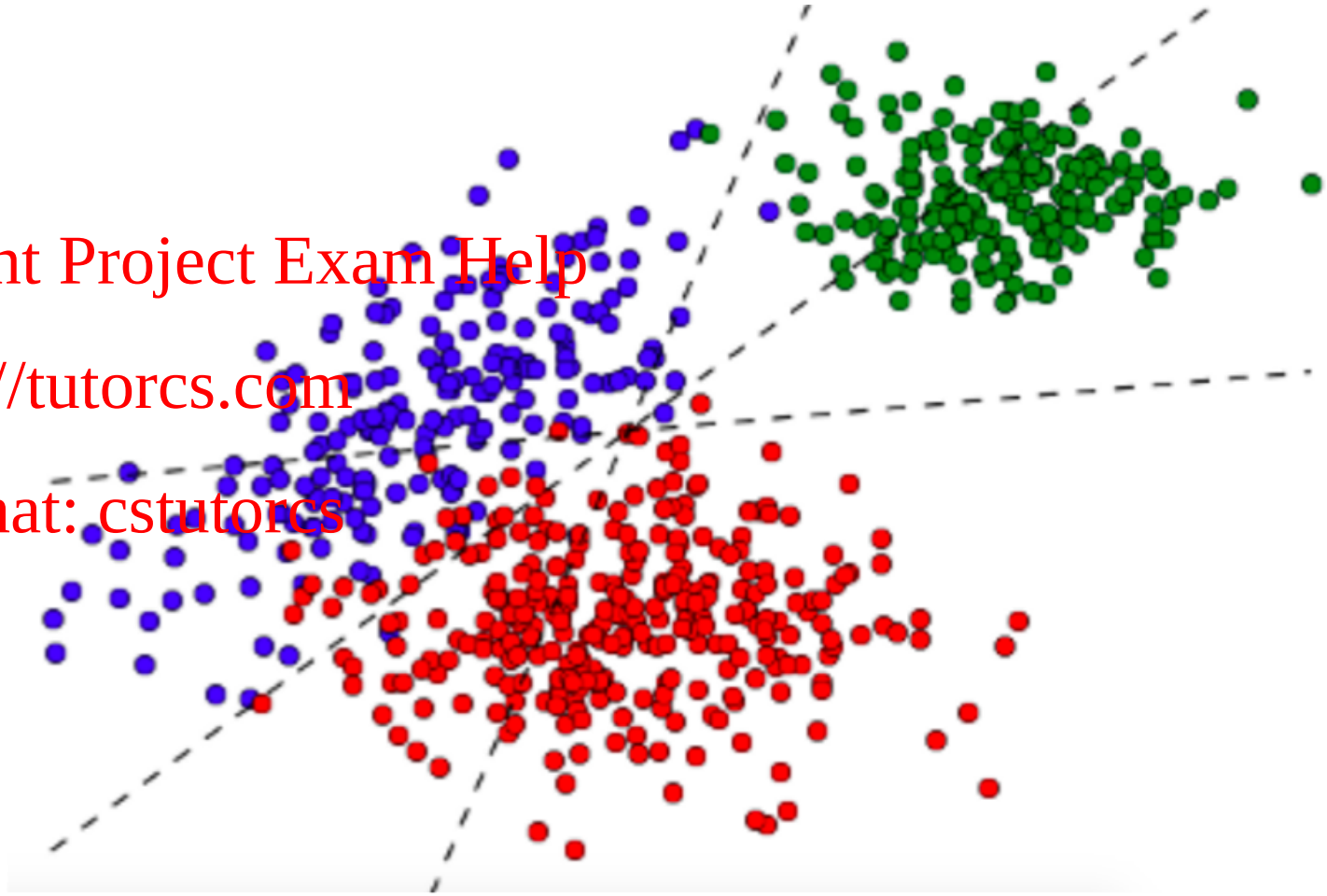




# **Biometrics** as **Multiclass Classification**

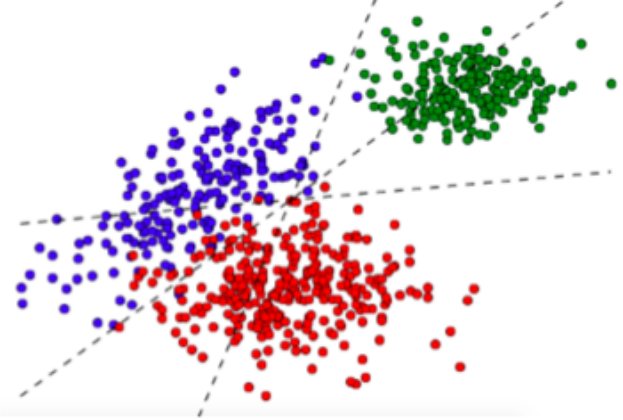
Machine Learning for Cybersecurity

- Biometrics identification:
  - each ID is a class label
  - multi-class classification



# **Biometrics** as **Multiclass Classification**

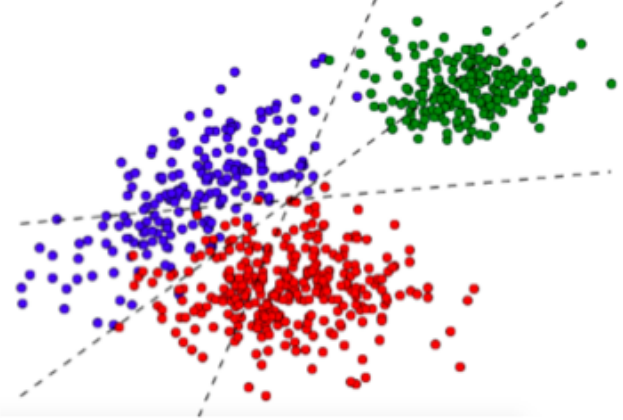
Machine Learning for Cybersecurity



- *Recap*:
  - biometrics forgery problem
    - biometrics considered by most to be highly unforgeable
      - thus much reliance & false sense of security due to this assumption
      - thus consequences devastating if forged
  - anti-forgery solution: liveness detection
    - real vs forgery/fake
    - assumption: real has liveness, fake does not
      - *e.g. sweat, pulsation, capacitance, ...*

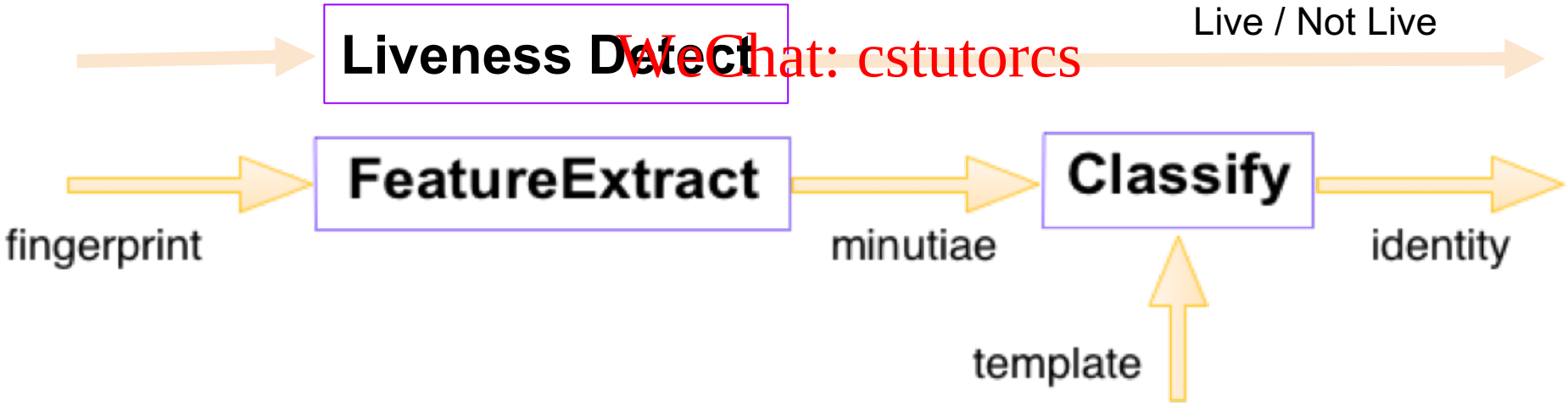
# Biometrics as Multiclass Classification

Machine Learning for Cybersecurity



- counter anti-forgery attacks e.g. [Bowden-Peters et al. 2012]
  - Q: why liveness detection not work?
    - liveness separately detected from feature extraction





# Spam Detection as Binary Classification

Machine Learning for Cybersecurity

- *Spam detection:*
  - *Spam vs Non-Spam*
  - *datasets comprise the entries (label,text)*
    - *where label* ∈ {non-spam,spam}
  - *could use the Natural Language Processing (NLP)*
    - *breaks language into shorter pieces*

# Classification: Empirical Tasks

Machine Learning for Cybersecurity

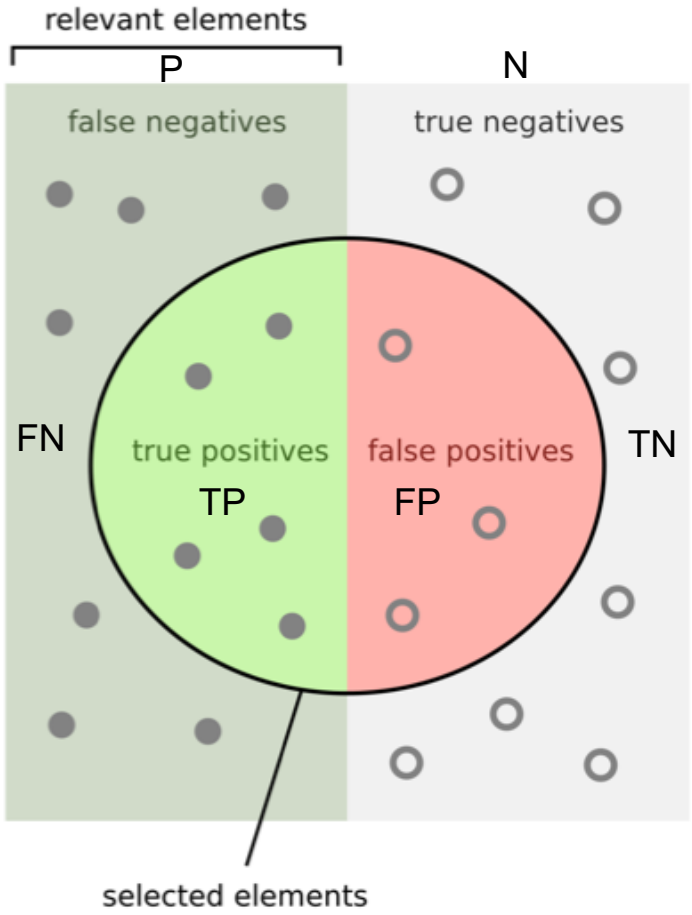
- separate dataset into **training set** & **test set**
- **train**/build the classifier
  - let it learn from seeing labelled samples from the training set, learn the decision boundary
- **test** the classifier
  - ...

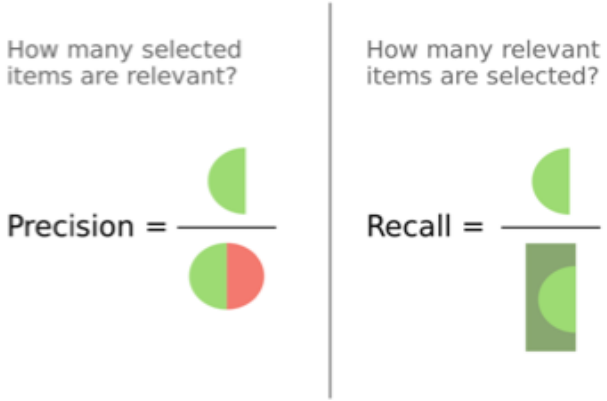
# Classification: Metrics

Machine Learning for Cybersecurity

- **test** the classifier
  - give it the samples from the test set, without the labels, it should **predict** what the label is for each sample
  - compute the **metrics**:
    - **accuracy** = #correct / #totalSamples
      = (TP+TN) / (P+N)
    - **precision** = TP / all P predictions = TP / (TP+FP)
    - **recall** = TP / all samples in that class = TP / (TP+FN)
      (a.k.a. TP rate or sensitivity)





# Classification: Metrics

Machine Learning for Cybersecurity

- **test** the classifier: compute the **metrics**
  - accuracy
  - precision
  - recall
  - **F1 score** = harmonic mean of precision & recall

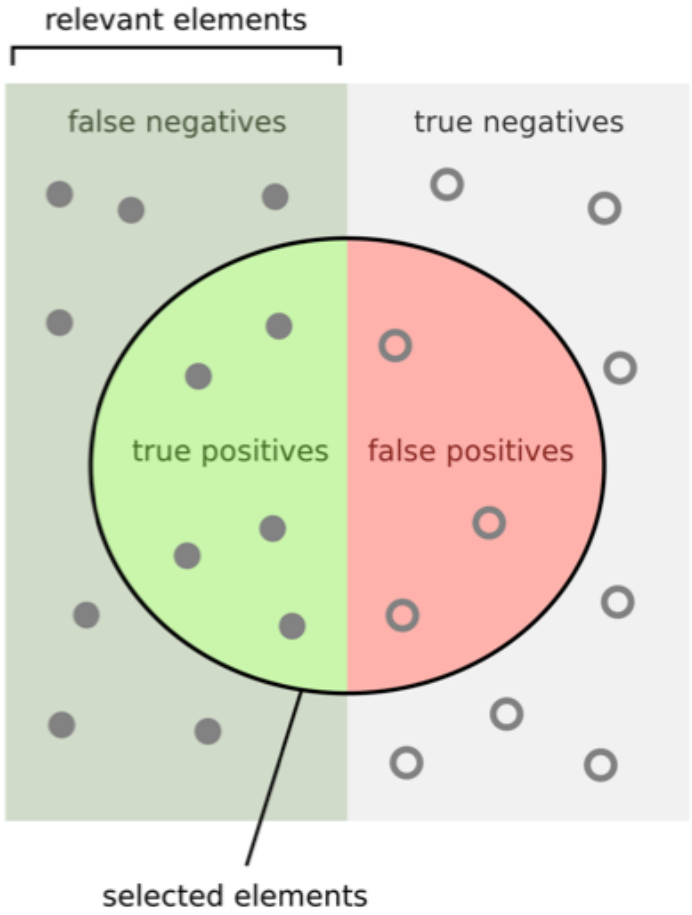


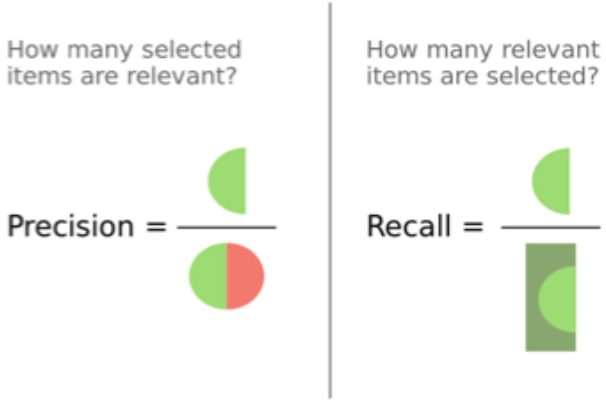
$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \cdots + \frac{1}{x_n}}$$





$$\frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$



# Classification: Metrics

Machine Learning for Cybersecurity

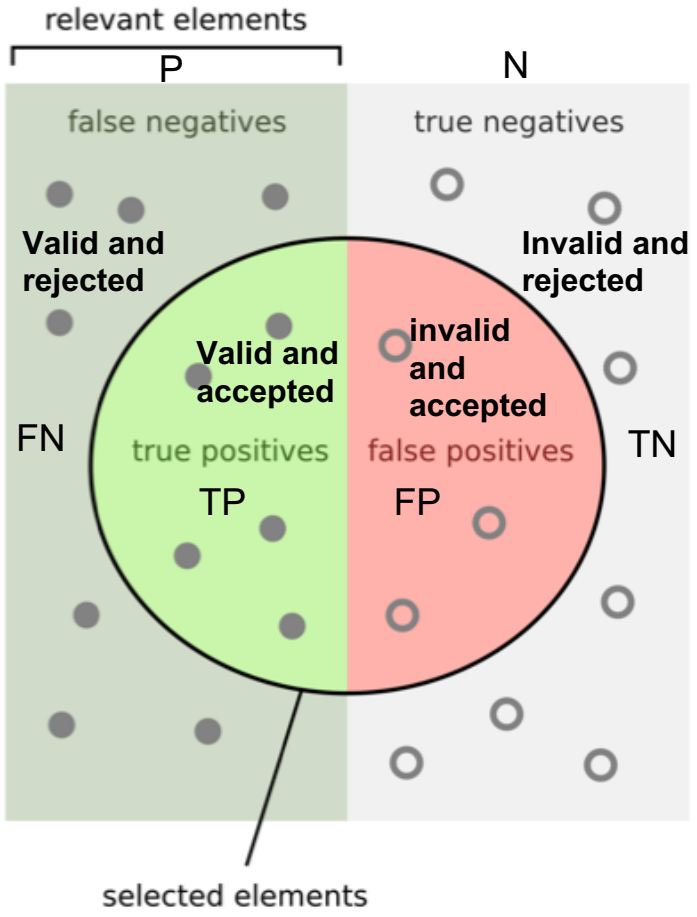
- **Recall last lecture's alternative classification metrics**
- **(commonly used in biometrics)**
  - **False Accept Rate (FAR) =** FP / all N samples = FP/(TN+FP)
    - compare with (1-precision) = FP / (TP+FP)
  - **False Reject Rate (FRR) =** FN / all P samples = FN/(FN+TP)
    - compare with (1-recall) = FN / (FN+TP)

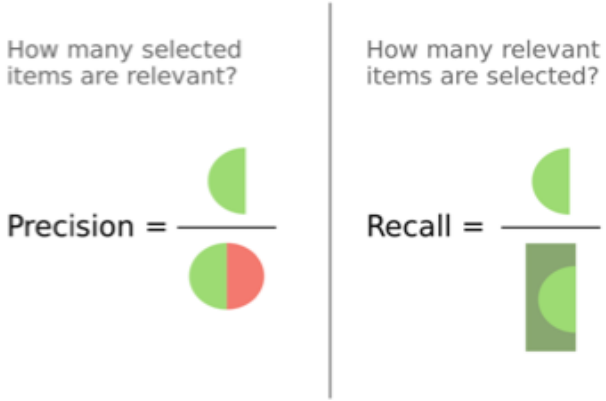
**Note:** above definitions assume

P = "positive" = "valid", N= "negative" = "invalid" (see next slide)



**QUESTION**: Which metric do you think is more relevant for security classification (assume P ="valid", N = "invalid"):
FAR or (1-precision)? Why?



# A note on definitions

- Definition of “positive” (P) / “negative” (N) can depend on the scenario / context
  - can be defined explicitly in each scenario to avoid confusion.



- However, **regardless of the definition of “Positive” or “Negative”:**
  - In security scenarios, the following natural convention for the meaning of Accept and Reject is usually followed:



    - Accept = gets through system = (if no errors) honest/valid/good
    - Reject = gets blocked = (if no errors) malicious/invalid/bad



    - Therefore FAR and FRR error rates can be defined generally as:
      - **False Accept Rate, FAR** = (# of malicious samples falsely classified as accept)/(total # of malicious samples)
      - **False Reject Rate, FRR** = (# of honest samples falsely classified as reject)/(total # of honest samples)

# **Spam Detection** as **Binary Classification**

Machine Learning for Cybersecurity

- feature learning
  - bag of words (BoW)
    - word & frequency

e.g.
(1) John likes to watch movies. Mary likes movies too.



BoW =
{"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};



(source: Wikipedia)

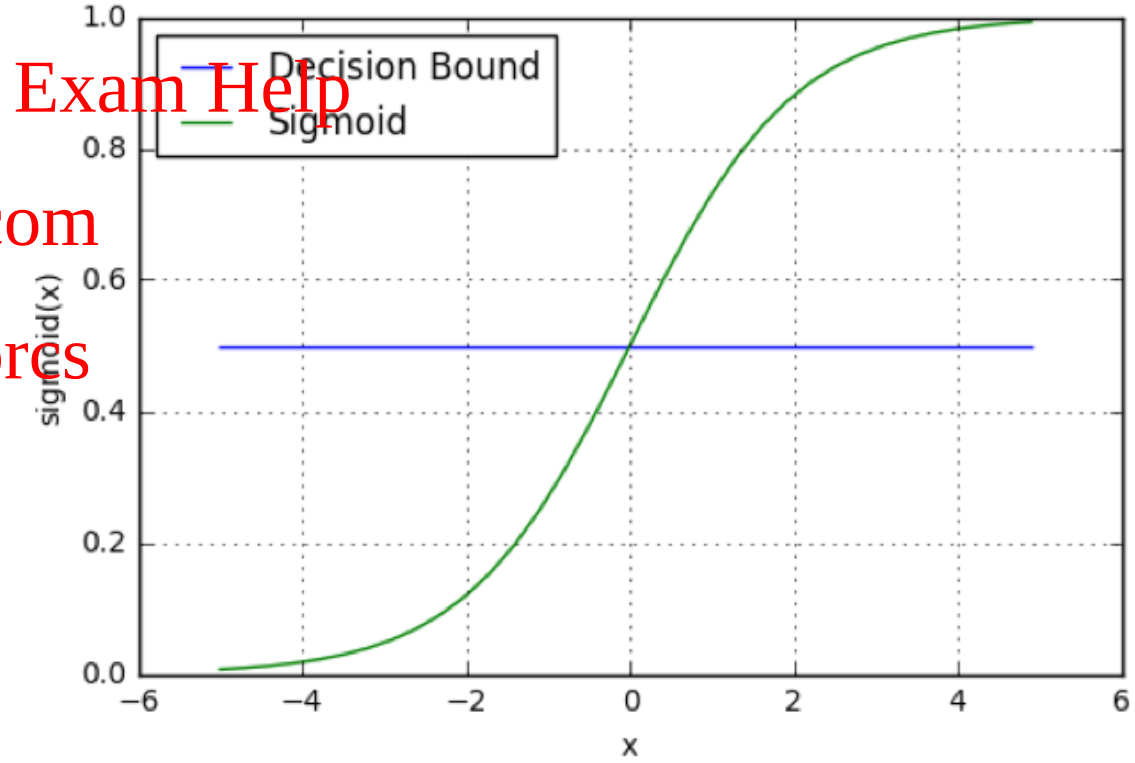
# Spam Detection as Binary Classification

Machine Learning for Cybersecurity

- classification
  - logistic regression classifier
    - features (in this case, BoW incl. frequency count) are input to sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

      outputting a value between 0 and 1 i.e. probability p
    - p > 0.5 → class = 1, else class = 0



# Cryptanalysis as Binary Classification

Machine Learning for Cybersecurity

- *Recap*:
  - CONFidentiality problem aims to safeguard secrecy of message m

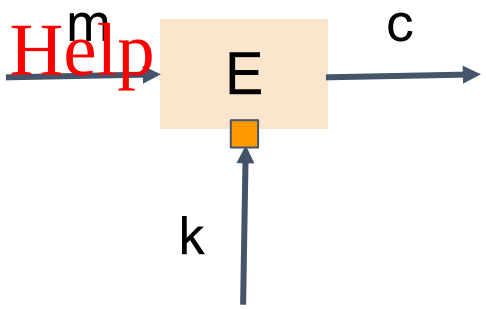


- How? ensure only specific parties can access
  - lock it up: unlock only if have key
  - transform it: reverse transform needs a key
    - encryption

# Encryption for CONF?

Machine Learning for Cybersecurity

- Encrypt? c = E(m,k)
  - m = message a.k.a. plaintext p
  - k = secret key
  - c = output ciphertext





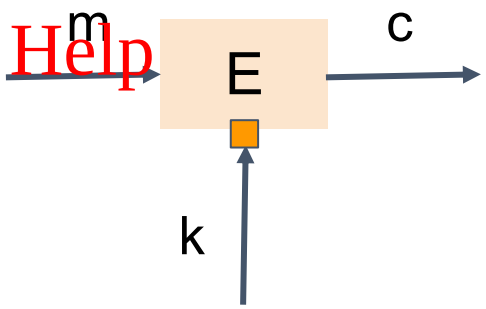

- Q: *since m needs to be CONF, how to ensure m remains CONF although c can be seen by anyone*?

# Encryption for CONF?

Machine Learning for Cybersecurity

- Encrypt? c = E(m,k)
  - m = message a.k.a. plaintext p
  - k = secret key
  - c = output ciphertext



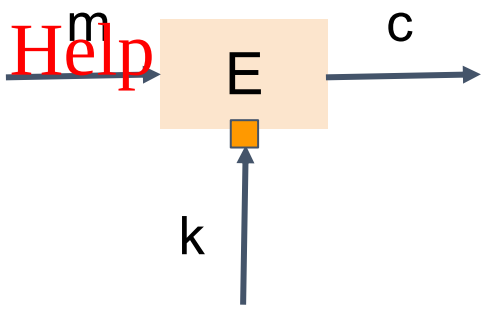



- Q: *since m needs to be CONF, how to ensure m remains CONF although c can be seen by anyone*?
- A: reversing E needs the k, only knowing k can E be reversed
- the end goal = CONF of m, the means = CONF of k

# Breaking Encryption CONF

Machine Learning for Cybersecurity

- the end goal = CONF of m,
  the means = CONF of k
- guessing k leads to breaking CONF of m





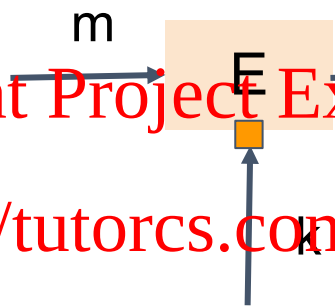

- cryptanalysis = security analysis of crypto techniques e.g. encryption

- basic cryptanalysis of encryption = binary classification problem
  - we'll discuss a few variations of cryptanalysis problems, any of these would indicate some weakness in the encryption design

# Cryptanalysis of Encryption: Case I

Machine Learning for Cybersecurity

- User
  - chooses secret k





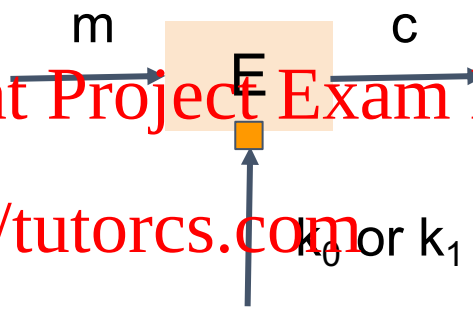

- Attacker
  - sees c
  - wants to guess k

- Case I: key-recovery attack

# Cryptanalysis of Encryption: Case II

Machine Learning for Cybersecurity

- User
  - flips a coin {0,1}
    - if 0: use $k_0$
    - if 1: use $k_1$






- Attacker
  - sees c
  - wants to guess if $k_0$ or $k_1$

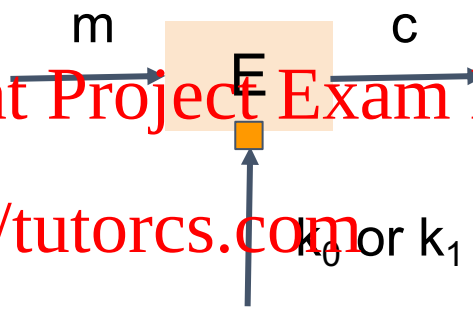
- Case II: key-distinguishing attack

- Q: *which is easier for attacker to do? Case I or II?*

# Cryptanalysis of Encryption: Case II

Machine Learning for Cybersecurity

- User
  - flips a coin {0,1}
    - if 0: use $k_0$
    - if 1: use $k_1$





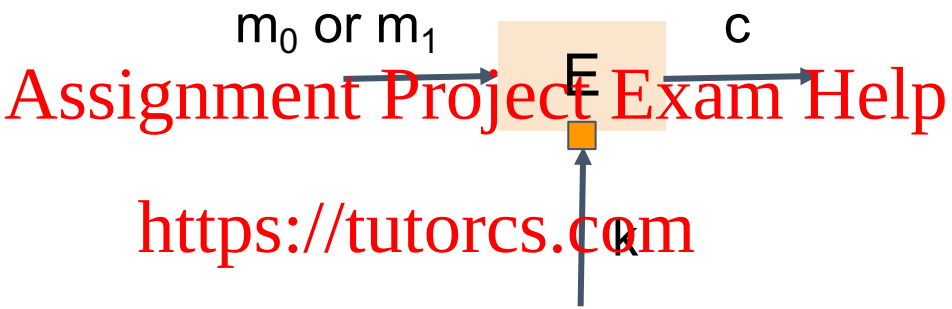

- Attacker
  - sees c
  - wants to guess if $k_0$ or $k_1$

- Q: *which is easier for attacker to do? Case I or II?*
- A: Case II is easier, just guess 1 of 2 options, vs Case I i.e. guess all bits of k

# Cryptanalysis of Encryption: Case III

Machine Learning for Cybersecurity

- User
  - flips a coin {0,1}
    - if 0: use $m_0$
    - if 1: use $m_1$






- Attacker
  - sees c
  - wants to guess if $m_0$ or $m_1$

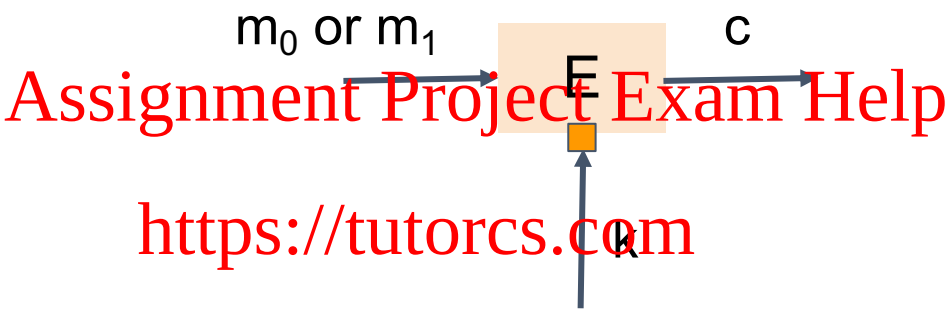
- Case III: plaintext-distinguishing attack

- Case II and Case III would be similarly difficult for the attacker since it is guessing 1 of 2 options

# Cryptanalysis of Encryption: Case III

Machine Learning for Cybersecurity

- User
  - flips a coin {0,1}
    - if 0: use $m_0$
    - if 1: use $m_1$






- Attacker
  - sees c
  - wants to guess if $m_0$ or $m_1$

- Case III: plaintext-distinguishing attack

- Q: *what weakness or problem could cause an attacker to be able guess if it is* $m_0$ or $m_1$?

# Further Reading

- A. Polyakov, ”Machine Learning for Cybersecurity”, article available at: https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b
  - This article discusses most of the topics we discussed, and provides links to further references.



**Note:** Our focus in this week’s lecture is on a high-level overview on the types of ML techniques and their potential applications in cyber security. We do not expect students to learn the details of how ML algorithms work – this is outside the scope of this unit.