

Introduction and Machine Learning Basics

CSIT375 AI for Cybersecurity

Dr Wei Zong

SCIT University of Wollongong

Disclaimer: The presentation materials come from various sources. For further information, check the references section

Outline

- **Subject overview**
- **What is AI**
- **Brief introduction to cyber security**
- **How AI helps Cyber security**
- **Limitations of AI in Security**
- **Math preliminaries**
- **Linear regression**

Subject information

- **Lecturer**
 - Dr Wei Zong
 - Email: wzong@uow.edu.au
- **7 Lectures (recordings will be available via Moodle):**
 - Introduction and machine learning basics.
 - Training algorithm to learn.
 - Spam and phishing.
 - Backpropagation and anomaly detection.
 - SVM and Decision Trees.
 - Adversarial machine learning.
 - Network intrusion detection case study.
- **8 Labs:**
 - 10 tasks.
 - guided coding practice.
 - not recorded.

Subject overview

This subject:

- covers the principles, techniques and applications of AI for cybersecurity
- focuses on how to **protect** systems with **data** and **algorithms**
- provides examples of how machine learning can be applied to augment or replace rule-based or heuristic solutions to problems such as **anomaly detection**, **malware classification**, and **network traffic analysis**
- also talks about security issues of machine learning/deep learning models.
 - Adversarial examples & backdoor attacks.

Subject learning outcomes

- Analyze and solve AI for cybersecurity problems by gathering, organising and evaluating data
- Explain core AI concepts and tools in relation to cybersecurity
- Describe principles in cybersecurity and apply machine learning techniques in system protection and cyber threat detection.
- Demonstrate how AI can be applied to solve cybersecurity-related issues.

Syllabus

- Introduction to AI
- Machine learning basics
- Training algorithms to learn
- Spam and phishing detection
- Anomaly detection
- Malware analysis
- Malware threat detection
- Network traffic analysis
- Adversarial machine learning

- **Programming Language and tool:**

- Python
- Jupyter notebook



Prerequisites

Abilities and skills that are expected:

- Familiarity with Python
- Understanding of Math (basic knowledge of probability and statistics)
- Ability to understand algorithms

Assessment

- **CSIT375:**
 - Quiz 1 (5%)
 - Quiz 2 (25%)
 - Assignment (20%):
 - Final exam (50%)

Assessment

- All written assignments must be submitted electronically **via Moodle before the due date**
- **Penalties apply to all late assessments**
 - 25% mark penalty per day for late submissions
 - Submissions received **more than 3 days late** will receive a mark of **zero**
 - Submit wrong files = Late. 1s late => 1 day late
 - e.g. if the assignment is marked on a scale from 0-10 and your mark is 8.5/10
 - 1 day late the mark will be reduced to 6/10
 - 2 days late the mark will be reduced to 3.5/10
 - more than 3 days late the mark will be reduced to 0/10

Attendance requirements

- Satisfactory attendance is deemed by the University, to be attendance at 80% of the allocated contact hours
- Lecture recordings are made available via the subject Moodle site
- Labs will not be recorded

Minimum performance requirements

- To be eligible for a 'Pass' in this subject a student must achieve a mark of at least 40% in the final exam (i.e. 20/50)
 - And at least 50/100 final marks.
- Plagiarism is treated seriously. All submissions are checked for plagiarism
- **All assessment must be your own original work**
 - DO NOT: copy from others, copy from the internet, pay someone to do it
- If we suspect any work is not original, all students involved are likely to receive zero for the affected assessment

Important message

- **Make sure you check the subject's Moodle site regularly**
 - Subject materials are made available on Moodle
 - We will post announcements to Moodle from time to time
 - such as changes in assessment requirements / due date
- **Your feedback is welcome**
 - positive aspects of this subject
 - issues of this subject to be addressed

Outline

- Subject overview
- **What is AI**
- Brief introduction to cyber security
- How AI helps Cyber security
- Limitations of AI in Security
- Math preliminaries
- Linear regression

What is AI

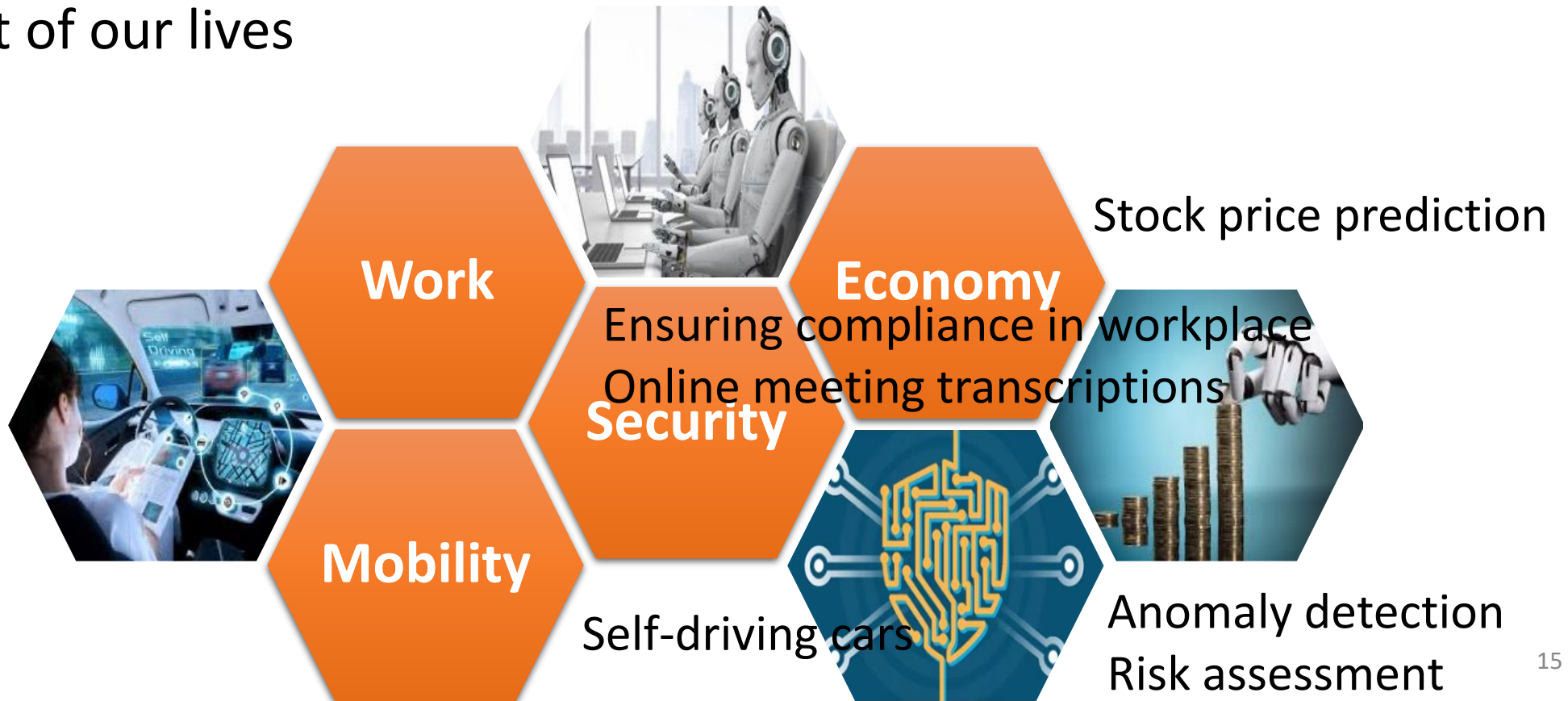
- **Artificial Intelligence (AI)**

Artificial intelligence is a popular term that indicates algorithmic solutions to complex problems typically solved by humans.

AI systems have been loosely defined to be machine-driven decision engines that can achieve near-human-level intelligence.

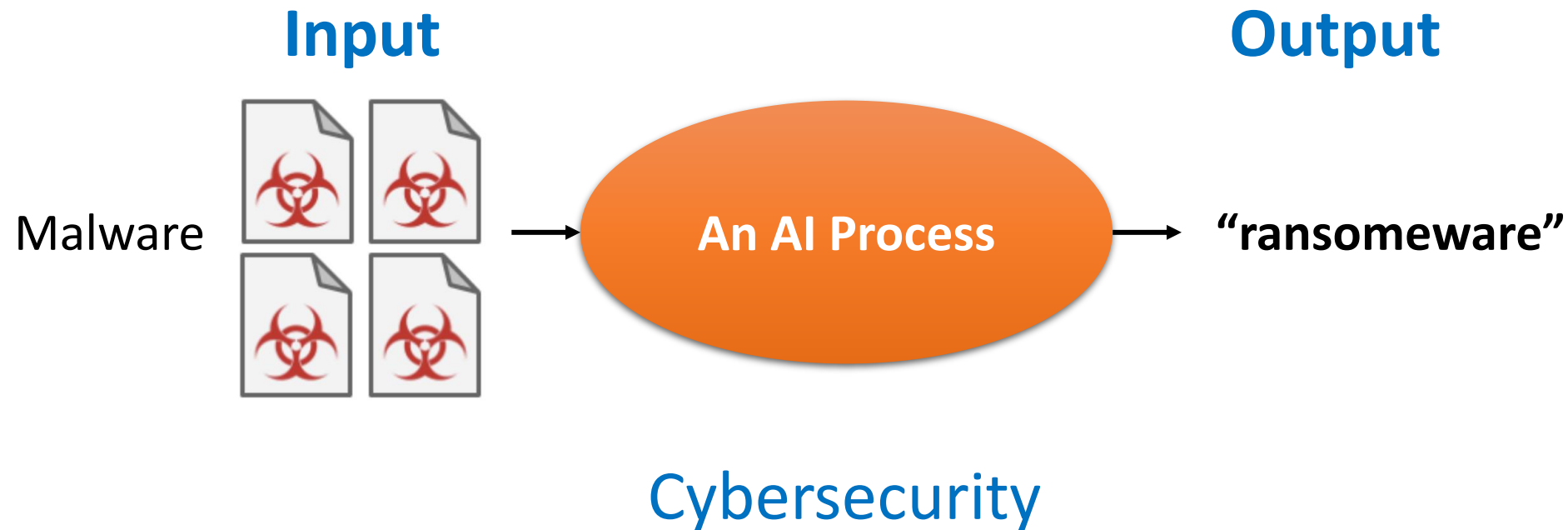
What is AI

- **Artificial Intelligence (AI)** has been moving extremely quickly in the last few years, demonstrating a potential to revolutionize every aspect of our lives



What is AI

- AI can be broadly defined as technology that can *learn* and produce intelligent behavior

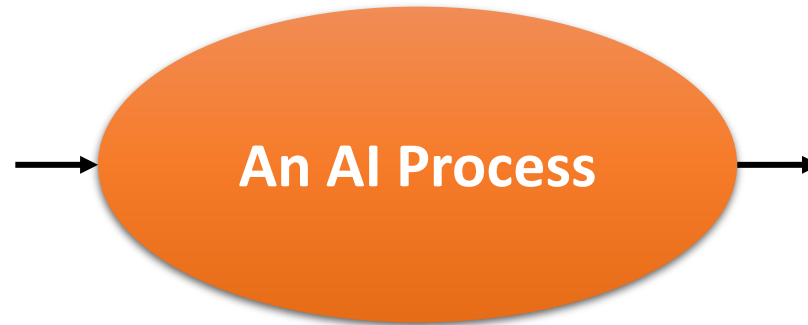


What is AI

- AI can be broadly defined as technology that can *learn* and produce intelligent behavior

Input

Pixels:



Output

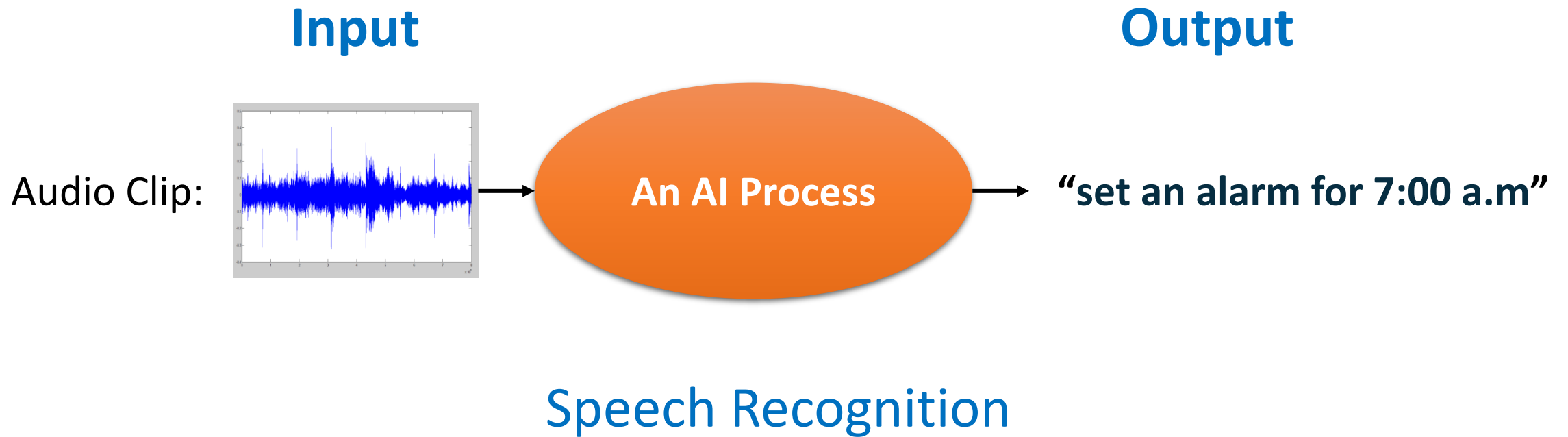
**“Four kids are playing
with a ball”**

More than just a category
about the image!

Computer Vision

What is AI

- AI can be broadly defined as technology that can *learn* and produce intelligent behavior



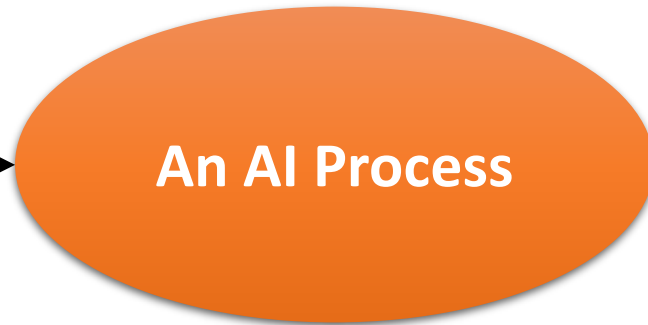
What is AI

- AI can be broadly defined as technology that can *learn* and produce intelligent behavior

Input

Output

Text: “Hello, how are you?”



An AI Process

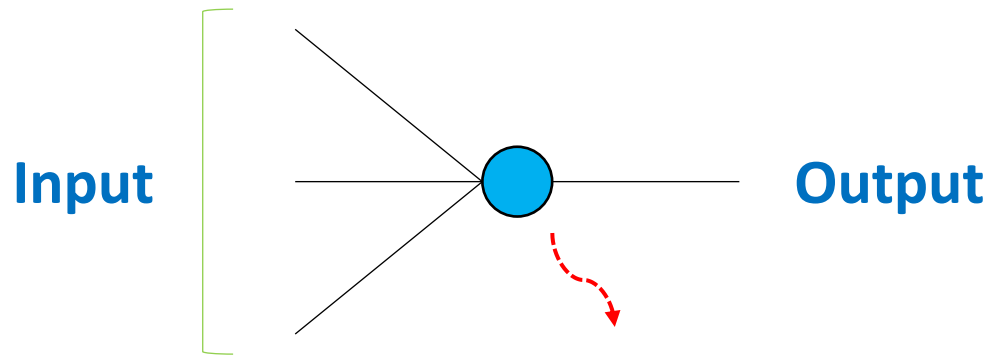


“Bonjour, comment allez-vous”

Machine Translation

What is AI

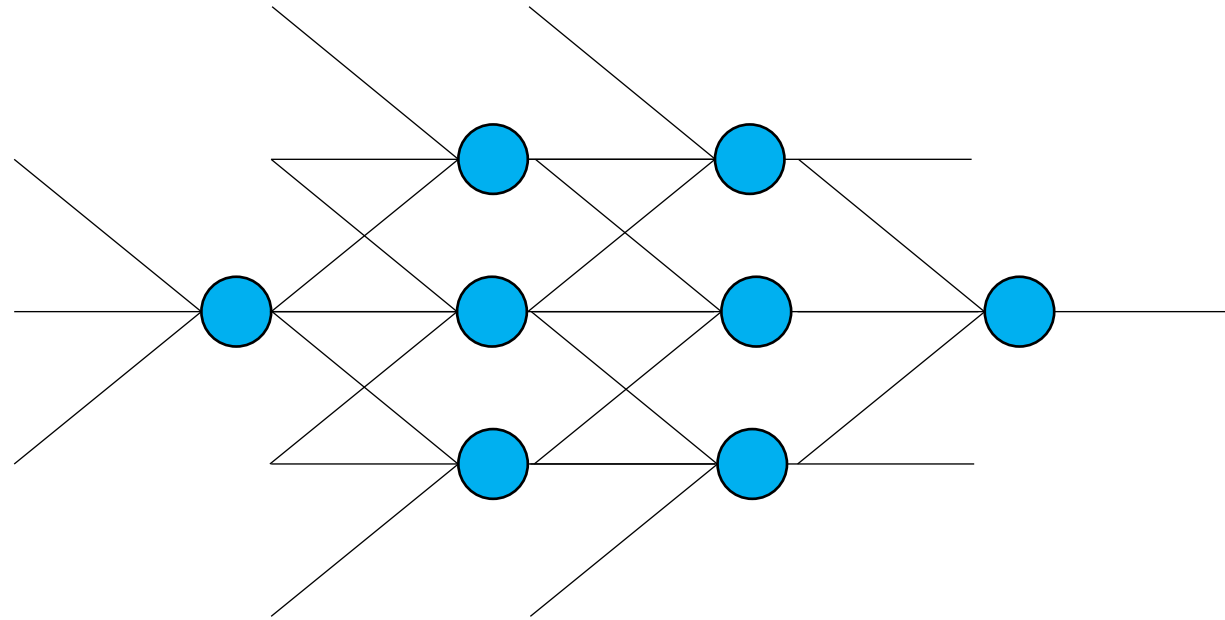
- Think of this as incoming impulses (**input**) passed from one **neuron** (AI process) to the next, if any, and finally generating an **output**



AI process, which is essentially a mathematical function-- more on this later.

What is AI

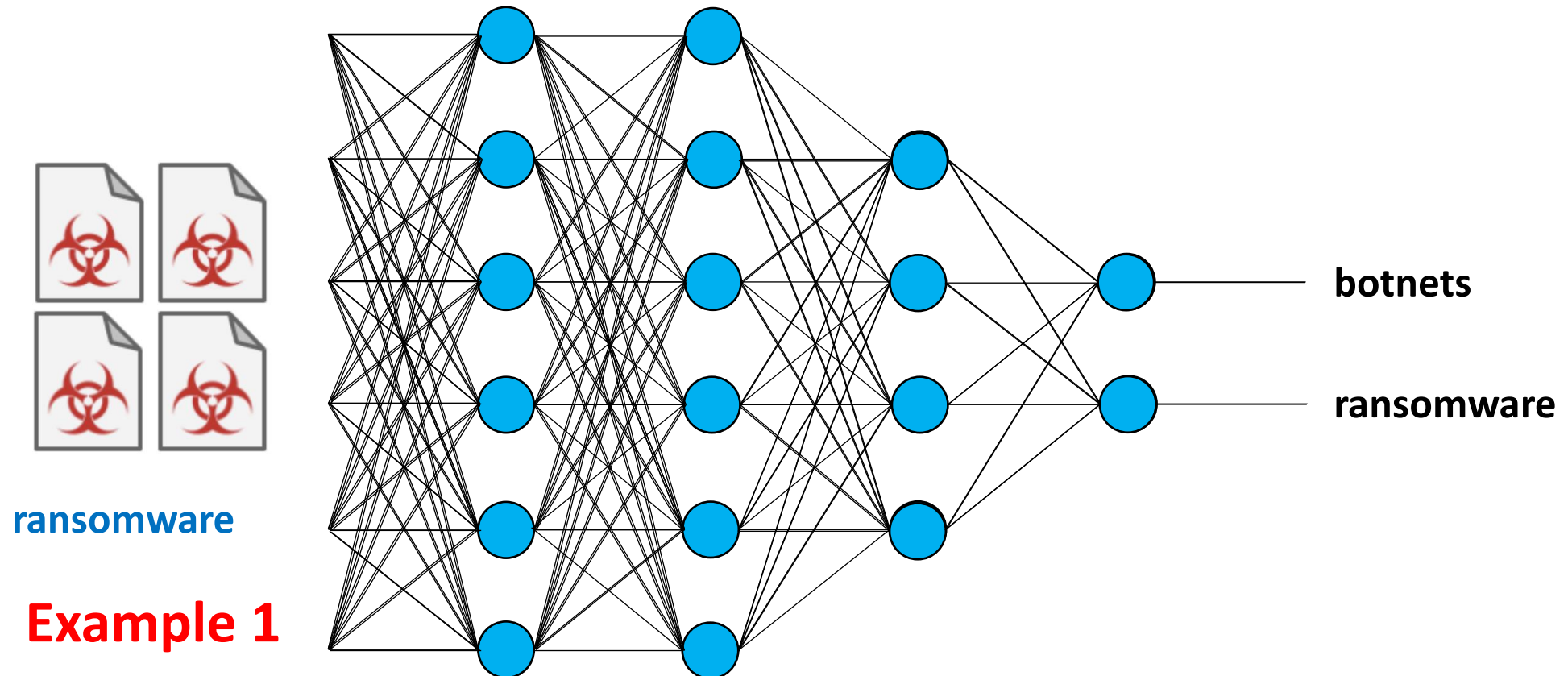
- Connect as many of these neurons as needed, resulting in what is called a **neural network** (a branch of AI)



The more layers you add, the deeper it becomes. Deep ones are referred to as **deep neural networks** or **deep learning (DL) models**

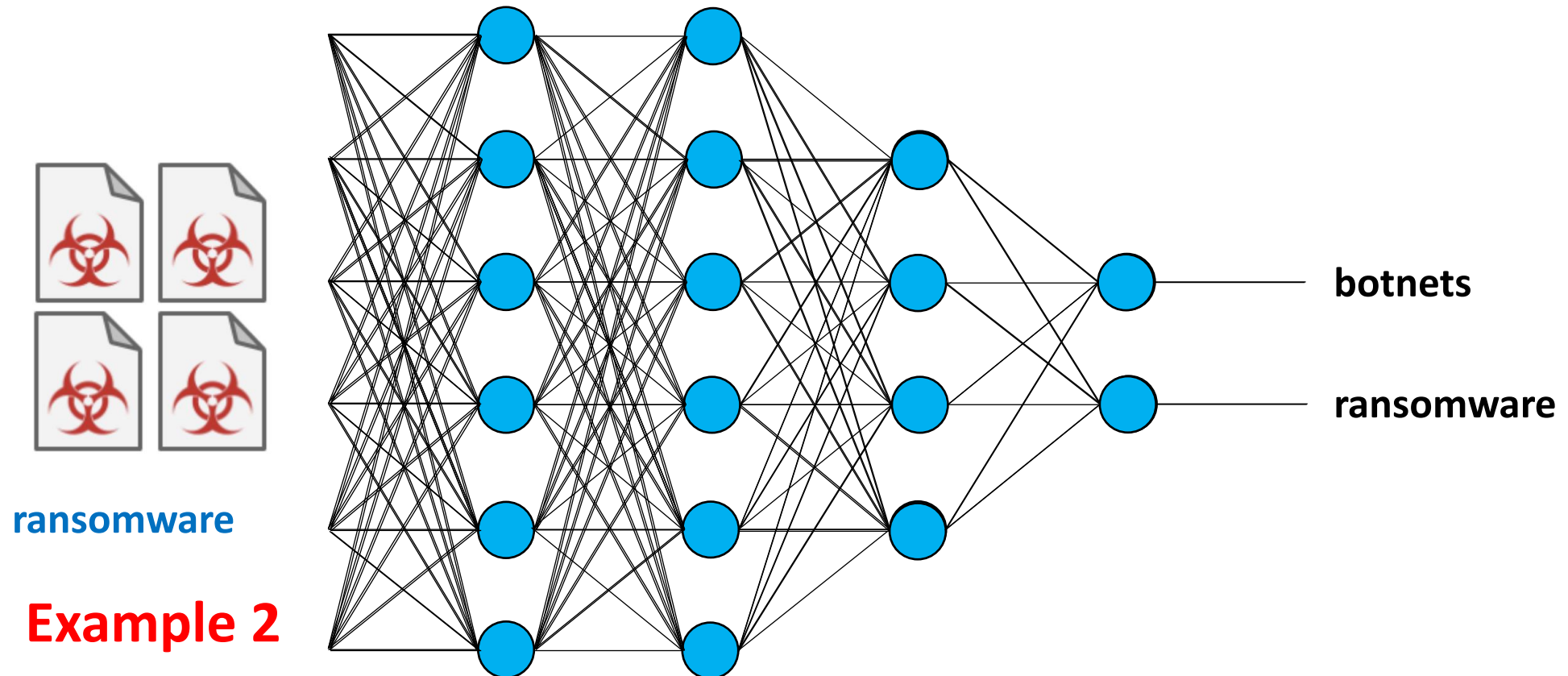
What is AI

- Subsequently, *train* the DL model



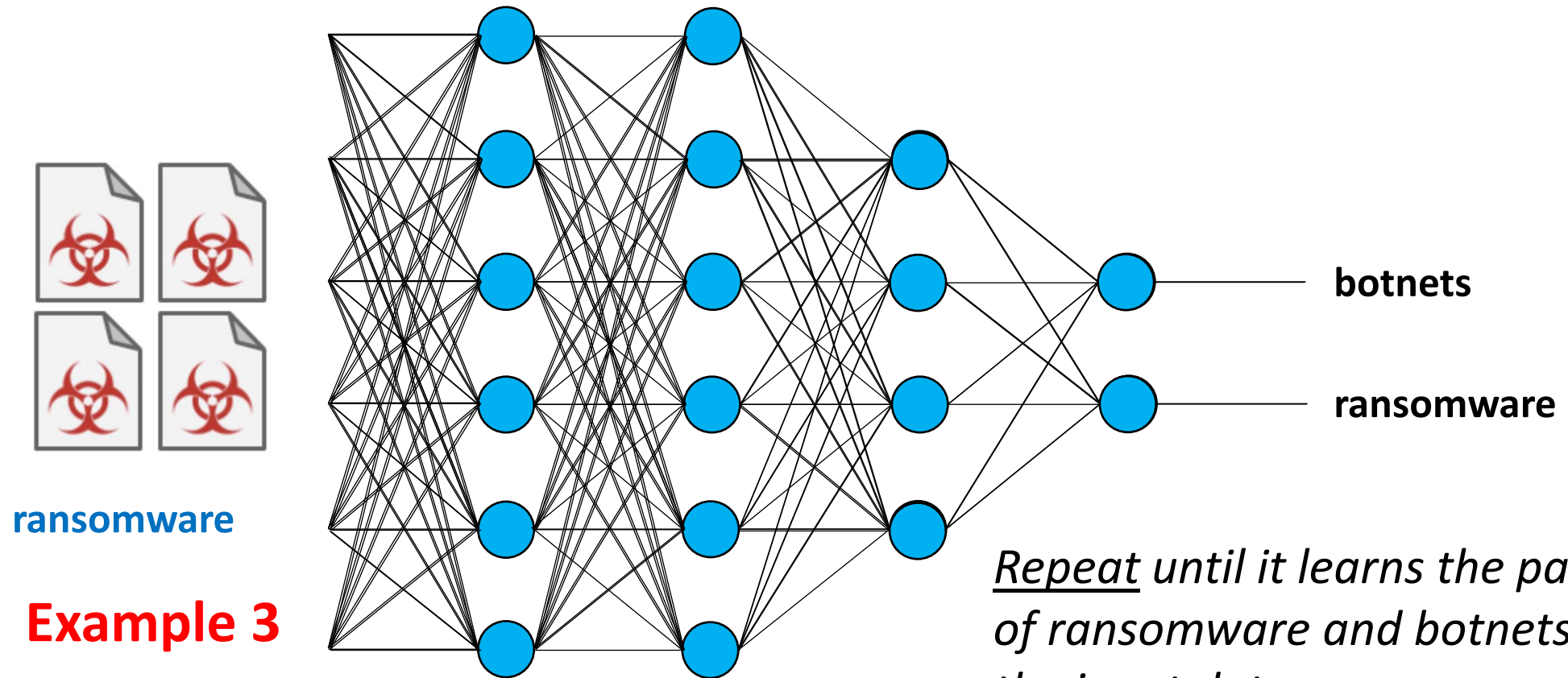
What is AI

- Again, with a different known example



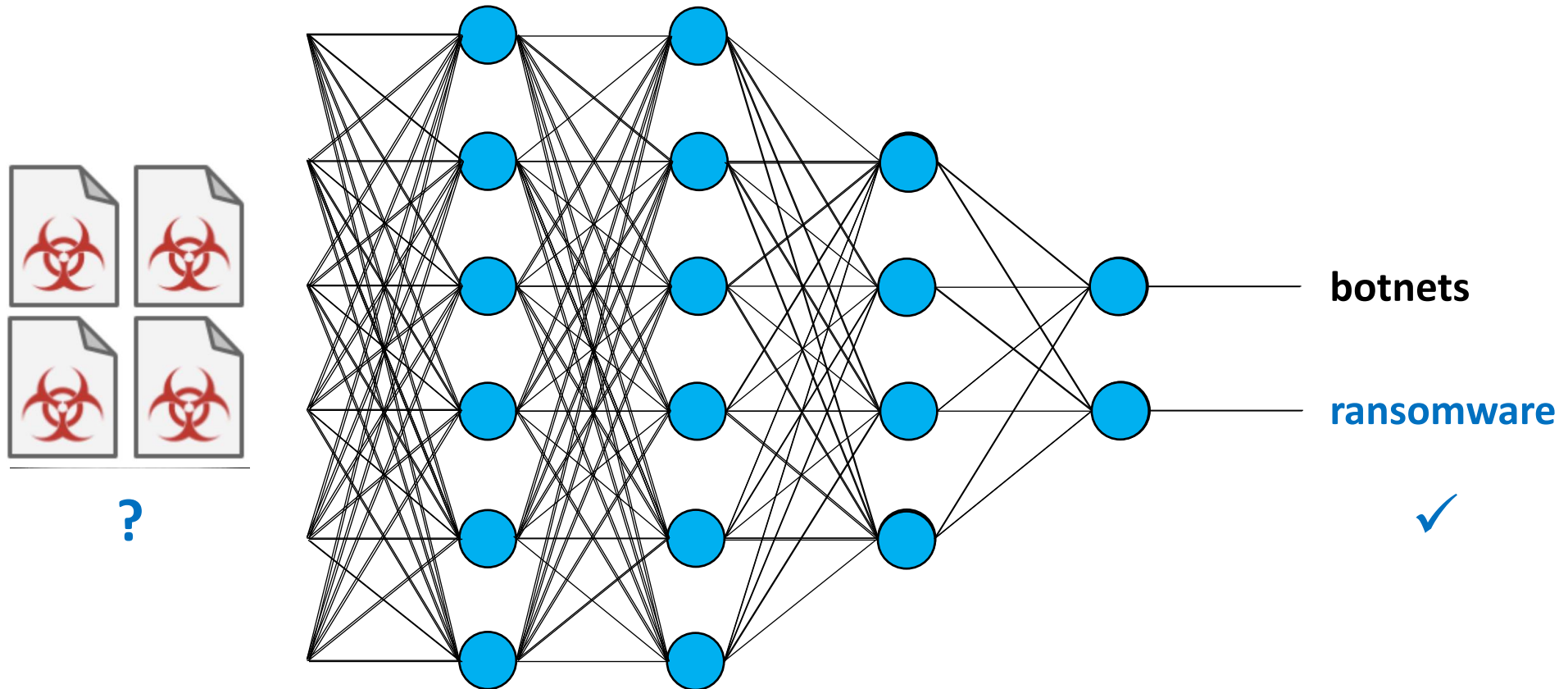
What is AI

- Yet again, with another different known example

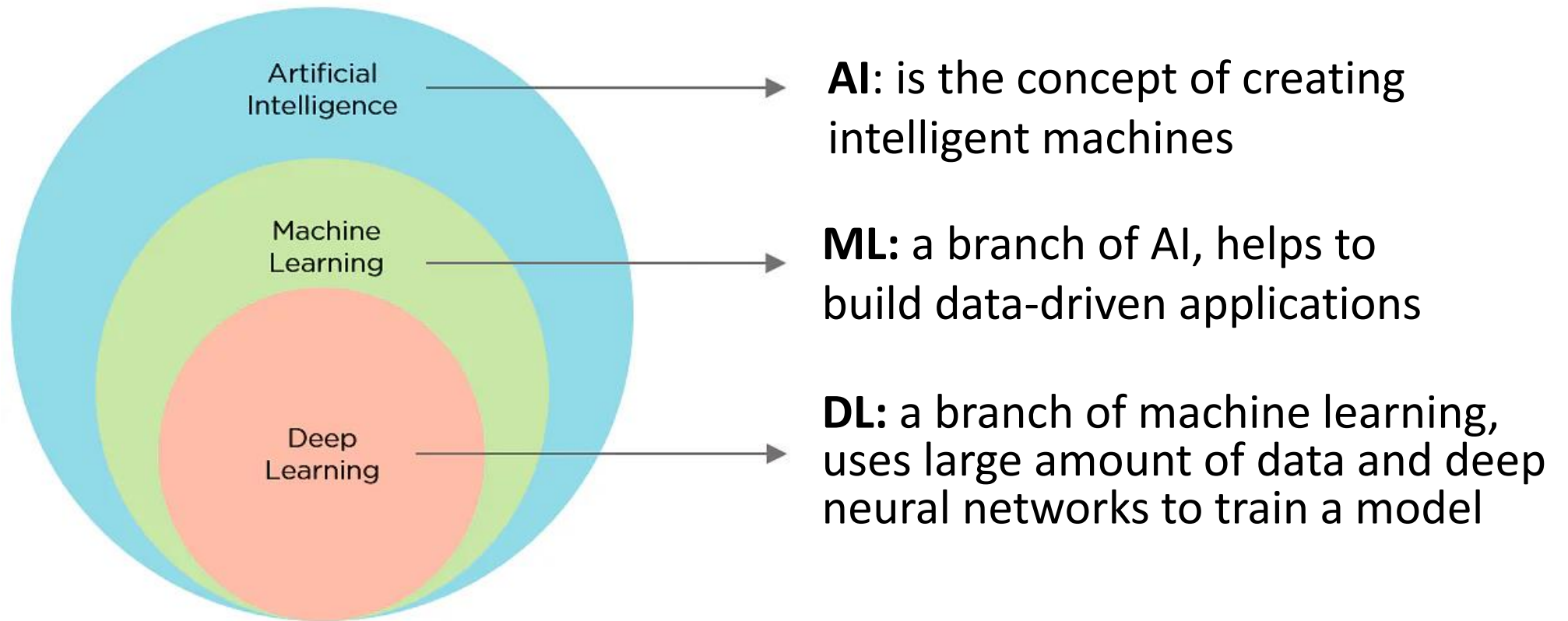


What is AI

- After training the DL model, use it to *infer* what an unknown malware is



Definitions



Outline

- Subject overview
- What is AI
- **Brief introduction to cyber security**
- How AI helps Cyber security
- Limitations of AI in Security
- Math preliminaries
- Linear regression

Definitions

- **Cyber Security**

- Cybersecurity is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access

- **Goals: C-I-A triad**

- **Confidentiality** unauthorized disclosure of information
- **Integrity** unauthorized modification of information
- **Availability** unauthorized withholding of information



What is cybersecurity all about?

- Protecting information and systems from threats and attacks
- ensuring they remain secure and functional
- In an organization, the **people, processes, and technology** need to complement one another to create an effective defense from cyber attacks
- **Users:** understand and comply with basic data security principles
- **Processes:** framework for how to deal with cyber attacks
- **Technology:** tools needed to protect from cyber attacks
 - entities need to be protected: endpoint, networks, the cloud
 - firewalls, malware protection, antivirus software, email security solutions...

Common types of cybersecurity threats

- **Phishing**

- sending fraudulent emails that resemble emails from reputable sources
- steal sensitive data, e.g. credit card numbers, login information

- **Ransomware**

- extort money by blocking access to files or the computer systems until the ransom is paid

- **Malware**

- gain unauthorized access or to cause damage to a computer

- **Social engineering**

- trick people into revealing sensitive information

Outline

- Subject overview
- What is AI
- Brief introduction to cyber security
- **How AI helps Cyber security**
- Limitations of AI in Security
- Math preliminaries
- Linear regression

How AI helps security

- **AI + ML + Cyber Security**

Machine learning has been quickly adopted in cybersecurity for its potential to automate the **detection** and **prevention** of attacks, particularly for next-generation antivirus AI systems

AI has a lot of potential for cybersecurity applications, such as learning from existing cyber incidents, predicting attacker behavior, and taking proactive defensive measures to protect critical infrastructures

How AI helps security

- AI lets computers learn without being explicitly programmed
- Challenges in managing threat information
 - track and correlate massive data
 - not feasible to manage with only people => automate the analysis
- In security, AI continuously learns by analyzing data to find patterns and predict threats in massive data sets
 - detect malware
 - find insider threats
 - keep people safe when browsing
 - uncovering suspicious user behavior
 - ...

How AI helps security

- **Find threats on a network**

- detect threats by constantly monitoring the behavior of the network for anomalies
- AI engines process massive amount of data in near real time to discover critical incidents
- detection of insider threats, unknown malware, and policy violations

- **Keep people safe when browsing**

- predict “bad neighborhoods” online to help prevent people from connecting to malicious websites
- analyze Internet activity to automatically identify attack infrastructures staged for current and emergent threats

How AI helps security

- **Provide endpoint malware protection**

- detect unknown malware that is trying to run on endpoints
- identify new malicious files and activity based on the attributes and behaviors of known malware

- **Protect data in the cloud**

- analyzing suspicious cloud app login activity
- detecting location-based anomalies
- conducting IP reputation analysis to identify threats and risks in cloud apps and platforms

Applications of AI in Security

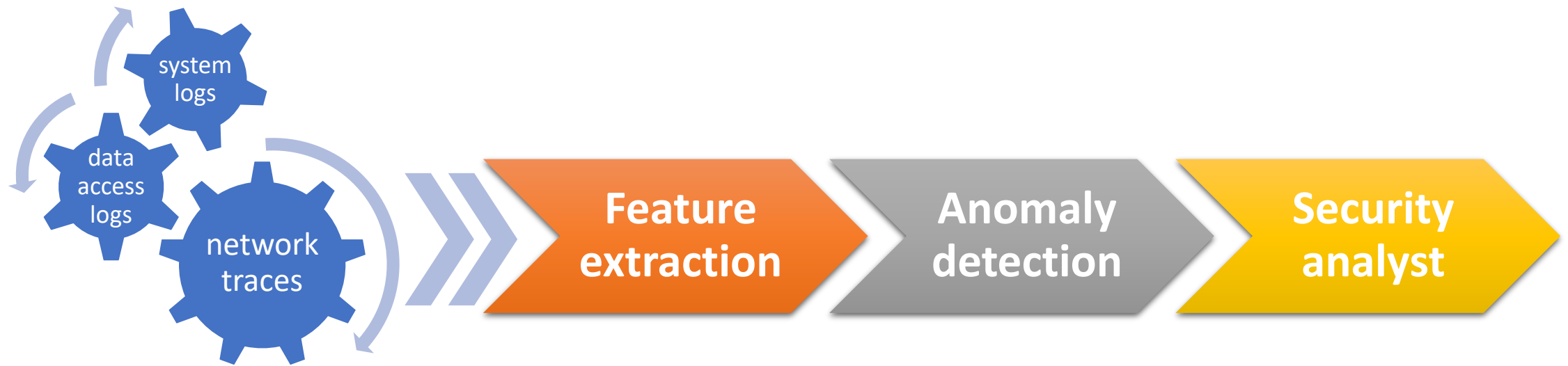
- **Monitoring**

- Automated, continuous, feed data to analysts
- Things to look for: intrusion, privacy violation, exfiltration, ...

- **Analysis**

- Not necessarily continuous
- Often initiated by humans
- Applications: threat intelligence, incident investigation, vulnerability assessment, ...

High-level view of a monitoring pipeline



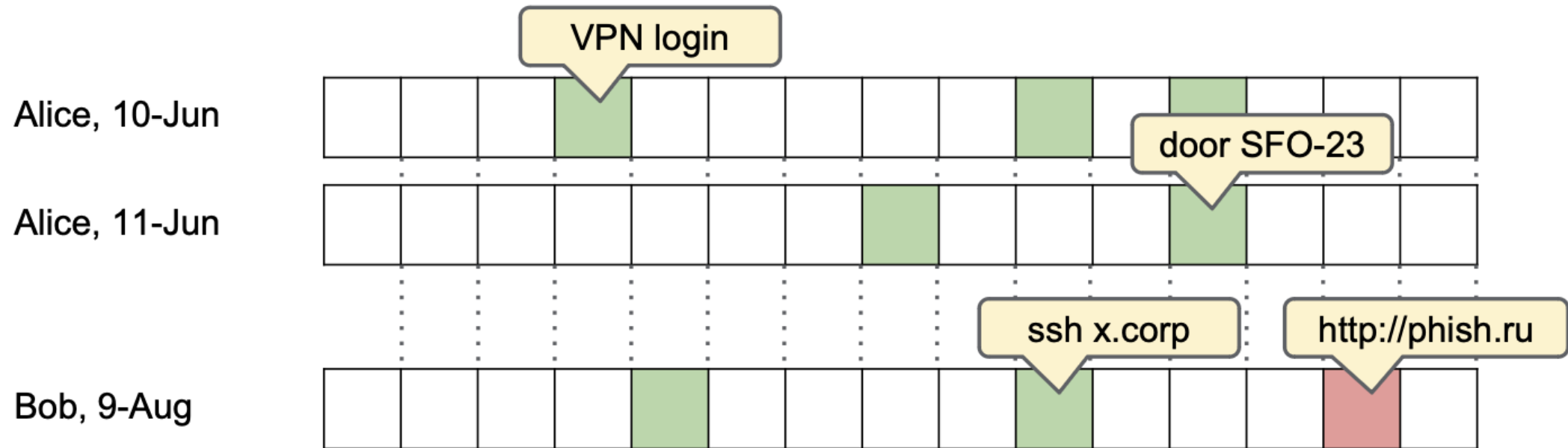
Example: detecting anomalous actor behavior

- Some reasons to care
 - Employee account compromised by malware?
 - Intentional malicious activity?
- Goal: model actor behavior, find anomalies
- What we need to do
 - Identify useful features
 - Model normalcy
 - Find outliers



Feature extraction: modeling actors

- Partition logs by actor and time
- Represent (actor, time) pairs as vectors of binary variables

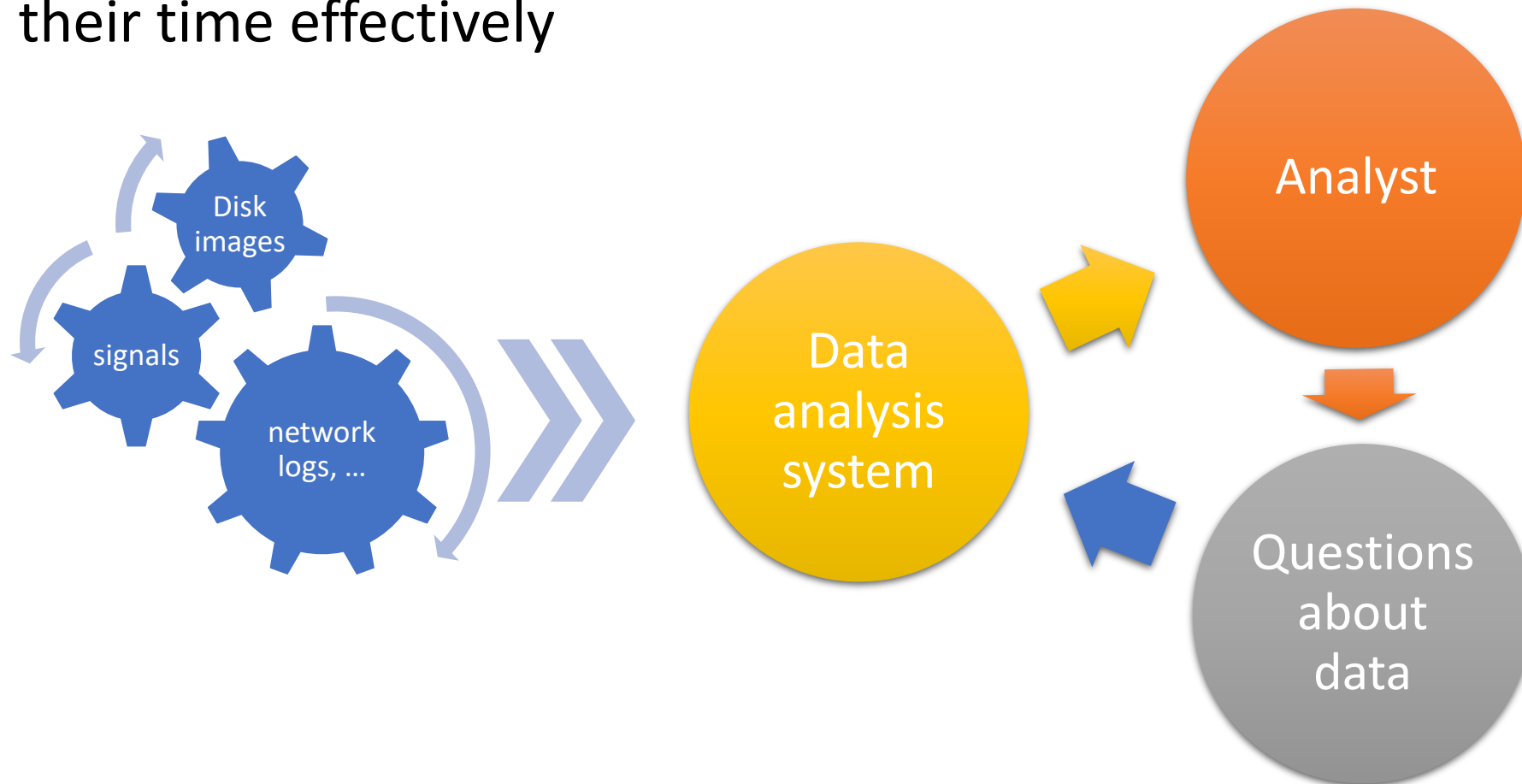


Modeling normalcy and finding outliers

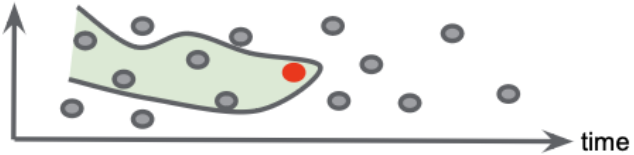
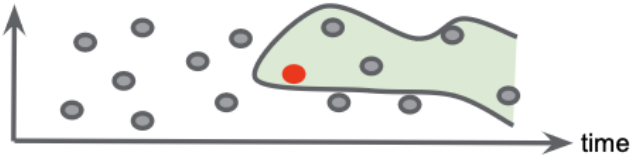
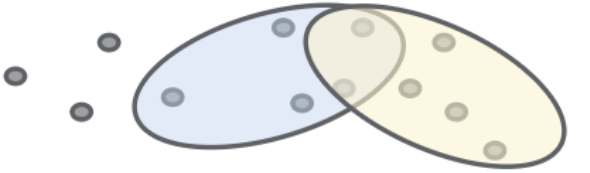
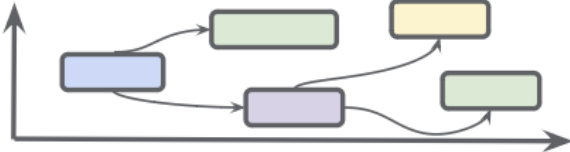
- Need to find low-probability features or combinations of features
- Many possible approaches
- Nearest neighbors \Rightarrow similarity metric between actors
 - Intuition: find users that are not very similar to any other users
 - Variant: compare a user to her past
 - “Neighbors” are feature vectors in user’s past
 - Identify changes in behavior
- “Strange pairs” \Rightarrow features that rarely appear together
 - Intuition: identify users with pairs of features that occur frequently individually but rarely together
 - E.g., “accessed source code” and “works in HR”

AI for security analysis

- Skilled analysts are a valuable resource: give them the tools to use their time effectively



Questions an analyst might ask

Causation	How did the attacker get root?	
Consequence	What was the effect of running the script?	
Correlation	Which signals fired simultaneously?	
Summarization	What was the user doing last night?	

Broad spectrum of tools

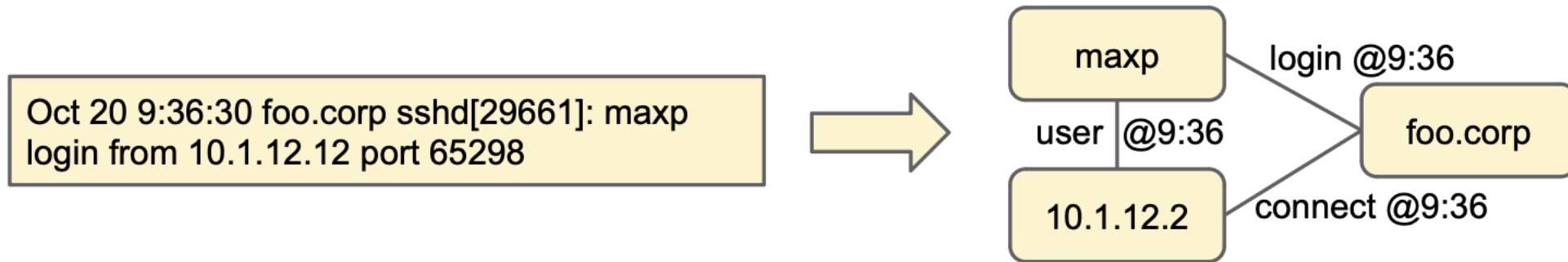
- Looking for causes and effects
 - graph traversal
- Triaging malware
 - classification
- Statistical (as well as graph-based) approaches are effective in this problem domain

Example 1: graph traversal for incident investigation

- Some questions to answer:
 - Were any machines affected by watering hole attack?
 - A watering hole attack is a form of cyberattack that targets groups of users by infecting websites that they commonly visit.
 - User A downloaded malware. What should be cleaned up?
- Given a graph representation of all relevant logs, can be framed as a large-scale graph search problem

Graph creation

- Log lines induce graph components



- Edges annotated with times and semantics
- Many different log sources in one huge graph

Sample graph query

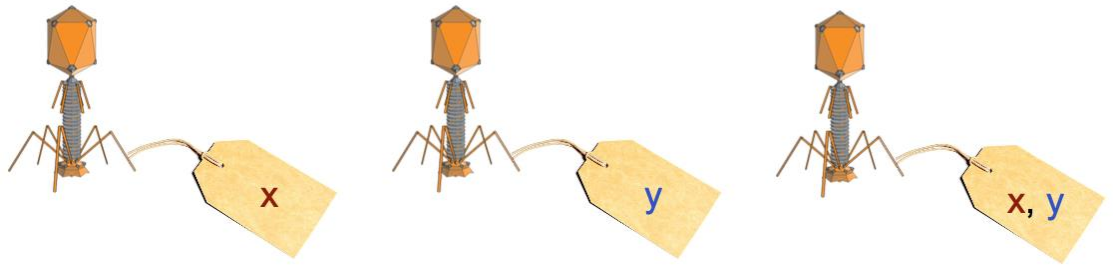
Given watering hole hostname X ...

- IPs that it resolved to
 - internal IPs that talked to them
 - machines (assets) those internal IPs belonged to
 - users who used those machines
 - other machines those users have logged into

Hours of manual research replaced by a ~10-second query

Example 2: malware classification

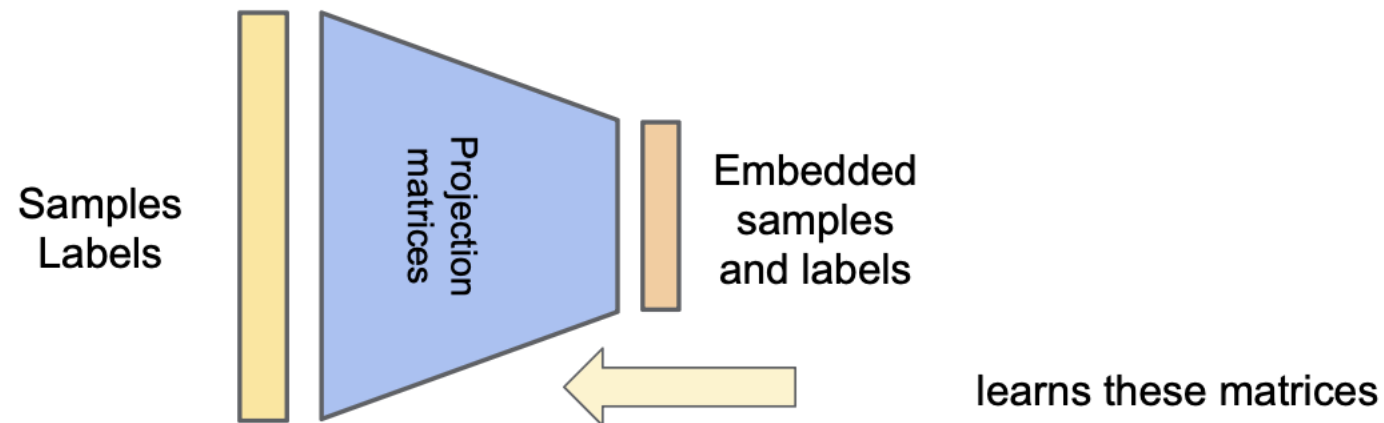
- Given a binary, is it malware? If so, what kind?
- Each sample is an executable. It has indicators (features) from static and dynamic analysis (e.g., basic block structure, registry changes, ...)
- Malware in training corpus also has one or more labels (from manual labeling, signatures, etc.) denoting its families



- Why is this useful?
 - Incident triage – is this malware we should care about?

Modeling the data

- Each sample X is a sparse N -dimensional vector ($N \approx \text{millions}$)
- Each label is an integer in $[1, k]$ ($k \approx \text{thousands}$)
- Learns a projection into a low-dimensional embedding space
 - Makes the problem computationally feasible
 - Provides meaningful metric inside embedding space



Using the model

- Once the projection matrices are learned, we can do useful things
- Compare two samples? Project into embedding space, measure distance
- Closest family to a sample? Project sample and all families, find smallest distance
- Approximate nearest sample? Filter samples by closest family

Outline

- Subject overview
- What is AI
- Brief introduction to cyber security
- How AI helps Cyber security
- **Limitations of AI in Security**
- Math preliminaries
- Linear regression

Limitations of AI in Security

- “Security is a process”
- Technology (AI, ML, etc.) is only a tool, not a complete solution
- User education (social engineering is surprisingly successful)
- System hardening (auth, secure engineering, timely patches, ...)
- Operational procedures
 - Adapting to growth (new hires / platforms)
 - Maintaining alertness (in the absence of major incidents)
 - Etc.

Limitations of AI in Security

- **False negatives can be very expensive**
 - Could cause arbitrary damage to our users
 - Network intrusion.
- **False positives can also be expensive**
 - Analyst time is valuable
- **Alerts should make sense to a human**
 - The analyst (security expert) is key
 - False positives + inexplicable results → signal fatigue
 - Inexplicable predictions are useless for the judicial purpose.

AI Models Have Limitations

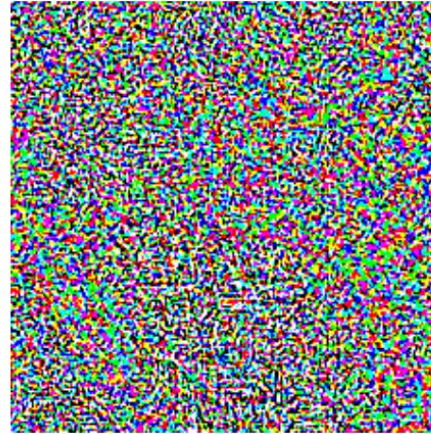


x

“panda”

57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

$=$



$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$

“gibbon”

99.3 % confidence

Real-world Attacks against AI Models



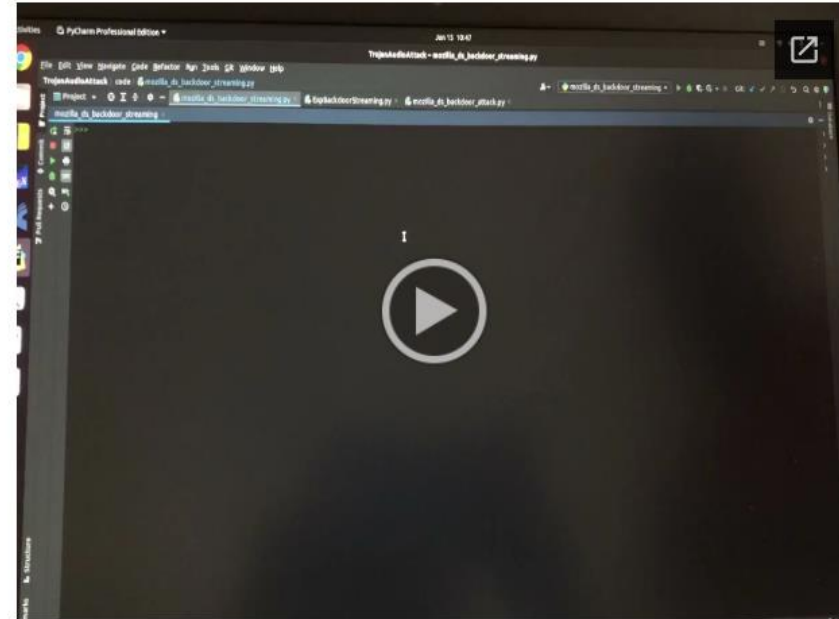
Real-world Attacks against AI Models

Real-world Attack Demo 1

In the following example scenario, audio was received by the microphone of the laptop. The laptop was used for recording and transcribing input audio. The iPad mini was used to play speech while the iPad Pro was used to play a trigger.

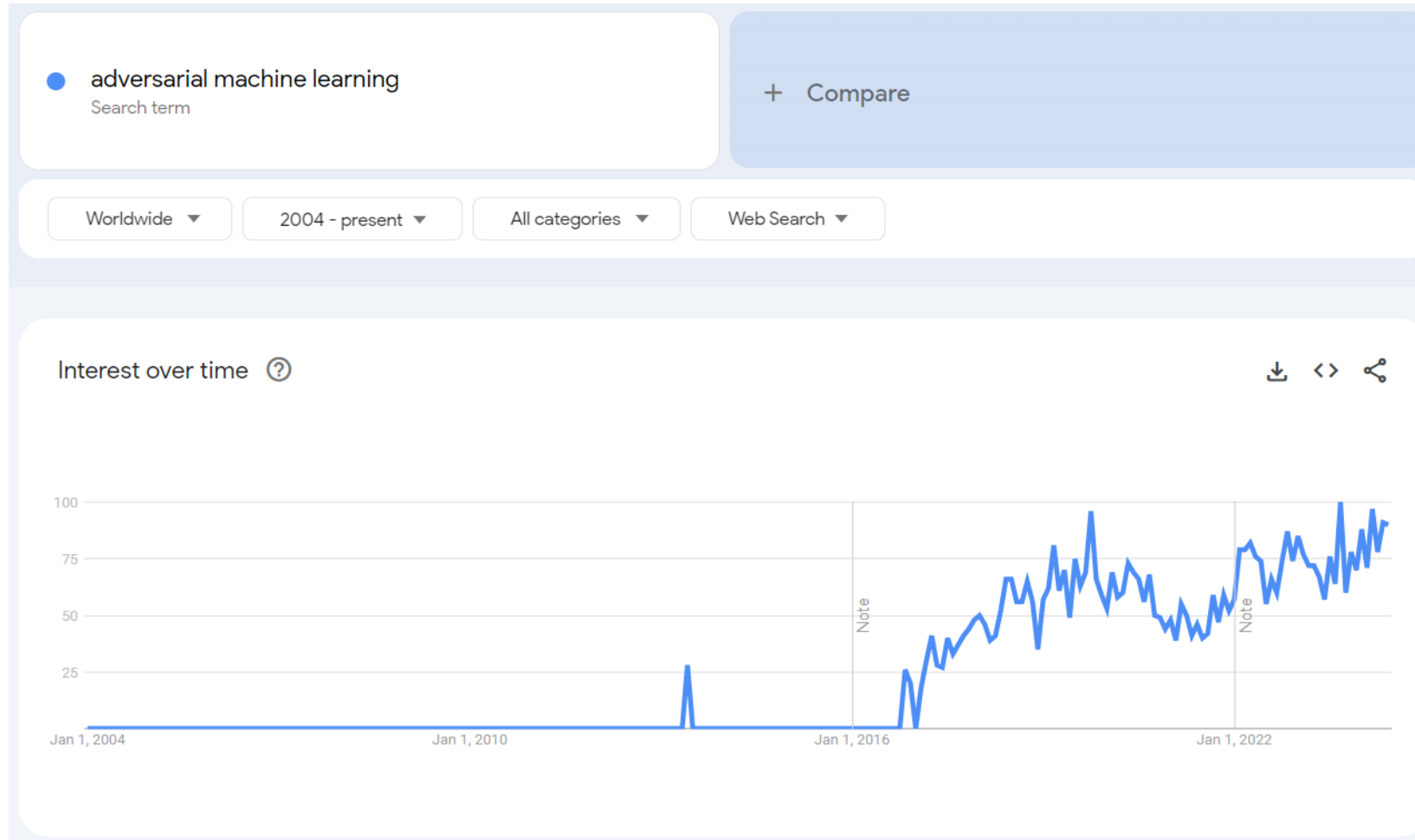


The trigger was played at a distance of 1 meter away from



The benign speech was correctly transcribed.

Adversarial Machine Learning



<https://trends.google.com/>

Outline

- Subject overview
- What is AI
- Brief introduction to cyber security
- How AI helps Cyber security
- Limitations of AI in Security
- **Math preliminaries**
- Linear regression

Plotting Data in 2D Space

- How to plot values of something (e.g., *network traffic*) over something else (e.g., *time*)?

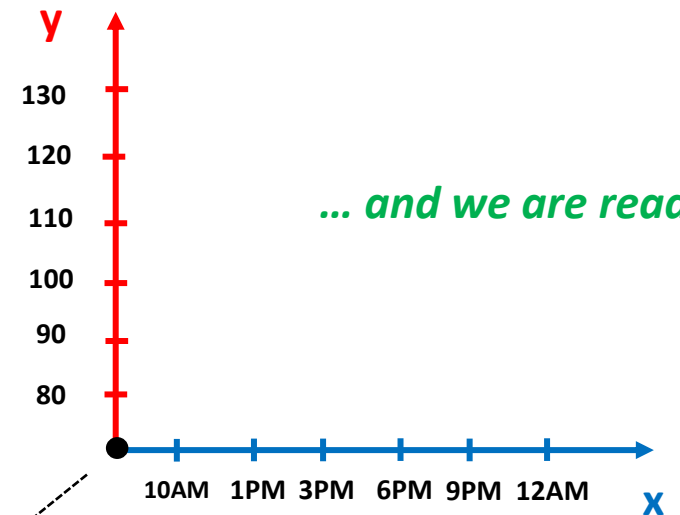


The horizontal direction is commonly called **x-axis**



The vertical direction is commonly called **y-axis**

Where they cross over is the **origin (0,0)**



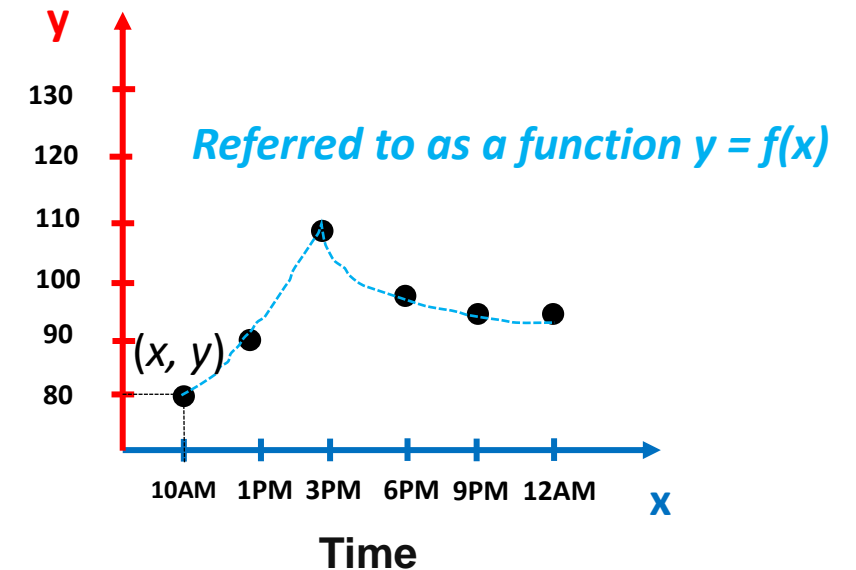
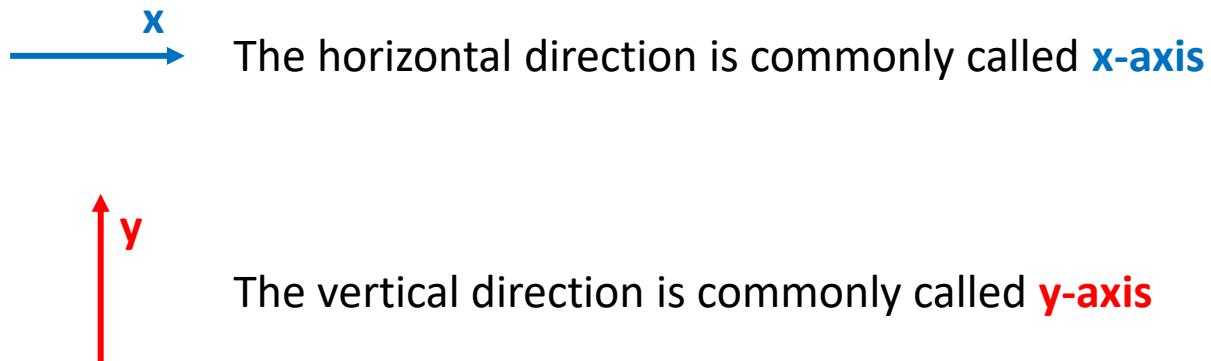
... and we are ready to go

Time

1. Put them together on a chart (or *graph*)
2. Define the *units* on the axes and label them

Plotting Data in 2D Space

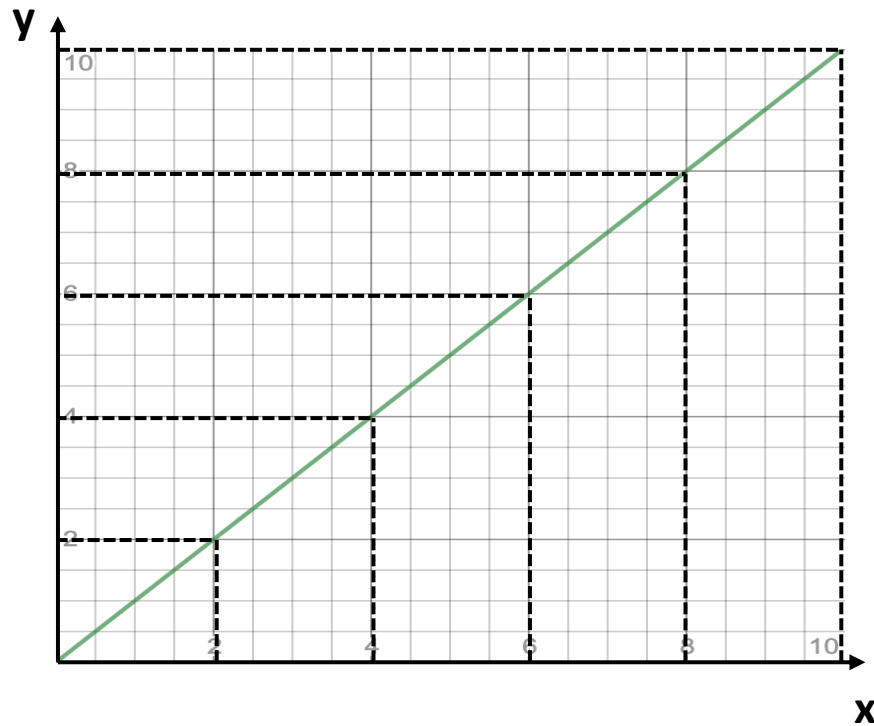
- How to plot values of something (e.g., *network traffic*) over something else (e.g., *time*)?



1. Put them together on a chart (or *graph*)
2. Define the *units* on the axes and label them

Mathematical Functions

- Here is a very simple function



x	y = f(x)
2	2
4	4
6	6
8	8
10	10

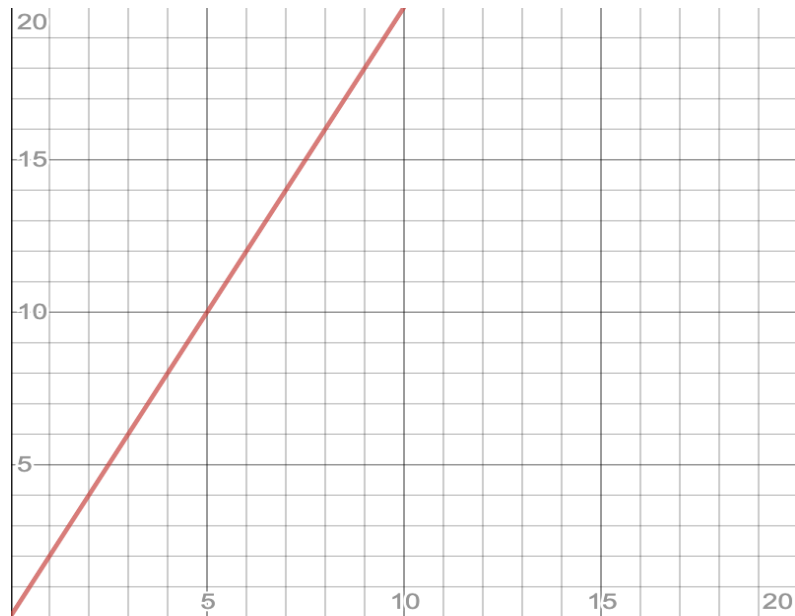
What is this function?

A line: $y = \underline{1x}$

For any given input (x), the output (y) is exactly the same

Mathematical Functions

- Here is another very simple function



x	y = f(x)
2	4
4	8
6	12
8	16
10	20

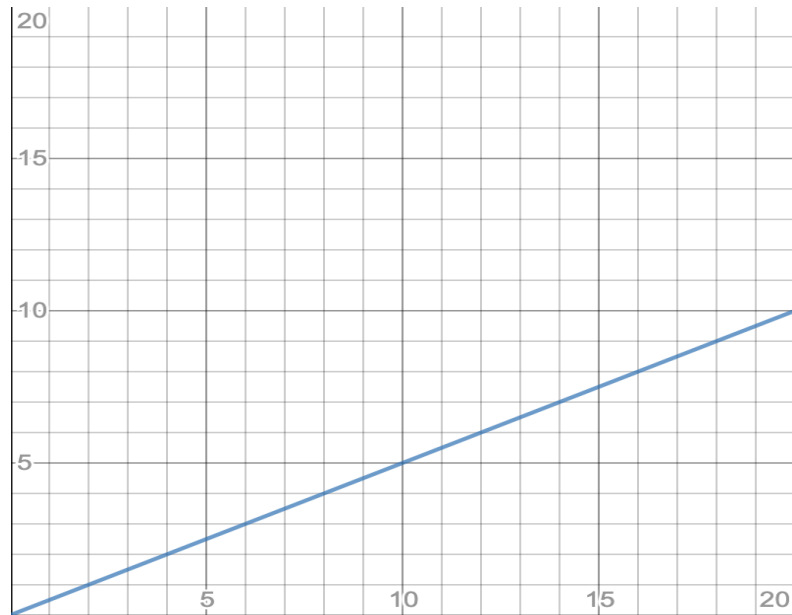
What is this function?

A line: $y = 2x$

For any given input (x), the output (y) is twice the input

Mathematical Functions

- Here is another very simple function



x	y = f(x)
2	1
4	2
6	3
8	4
10	5

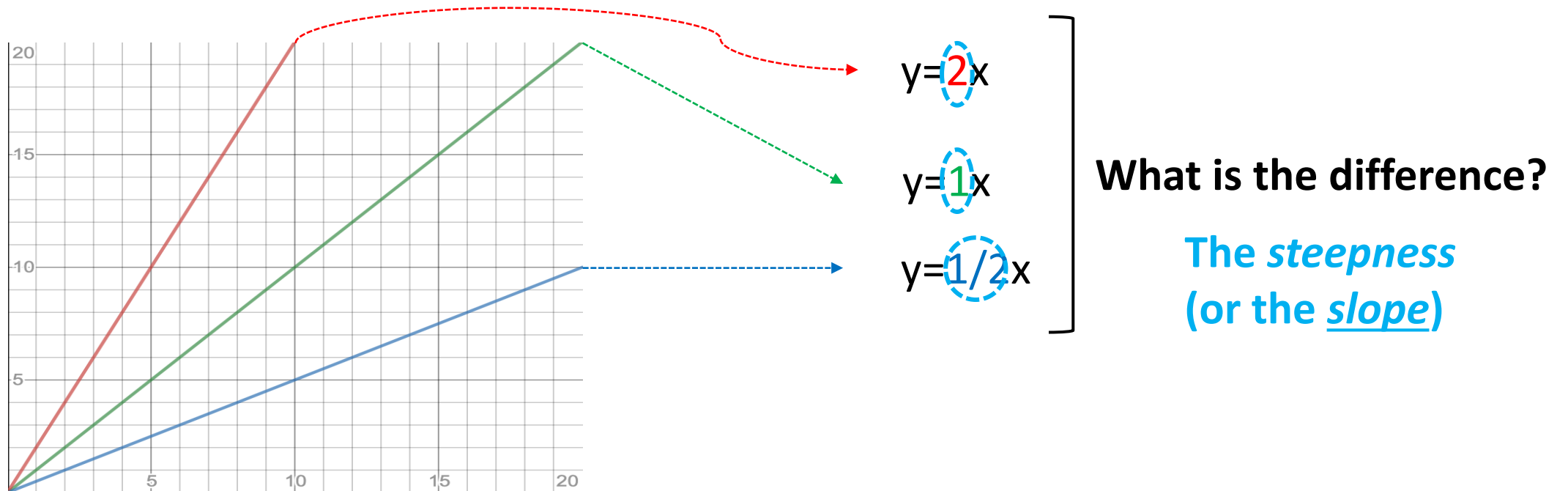
What is this function?

A line: $y = \underline{1/2}x$

For any given input (x), the output (y) is half the input

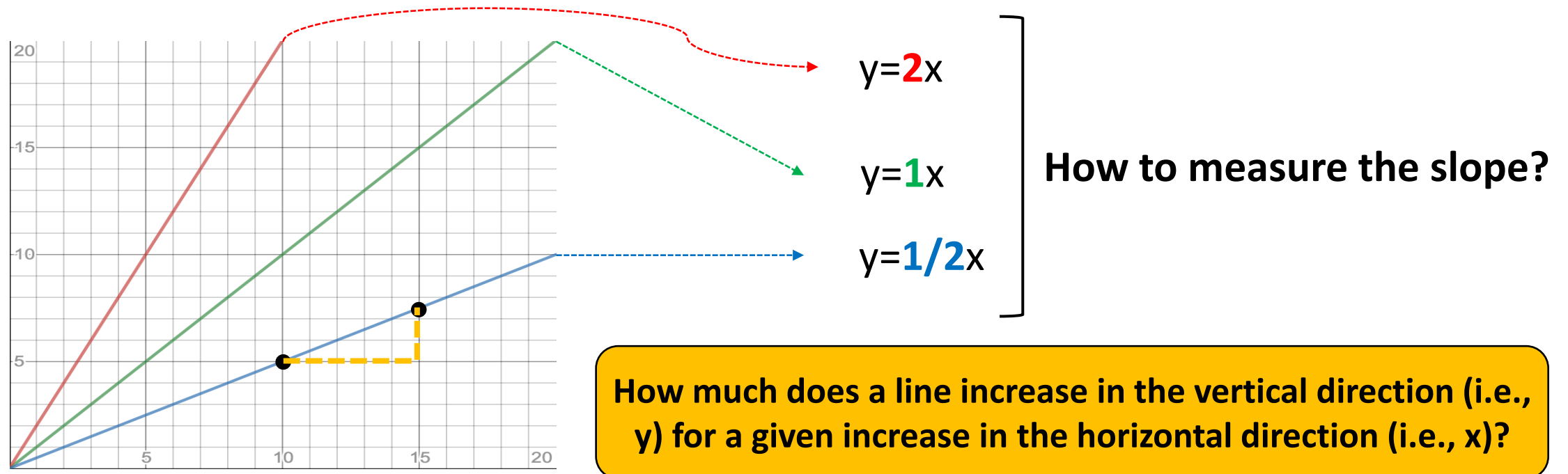
Slope

- Here are the 3 functions together



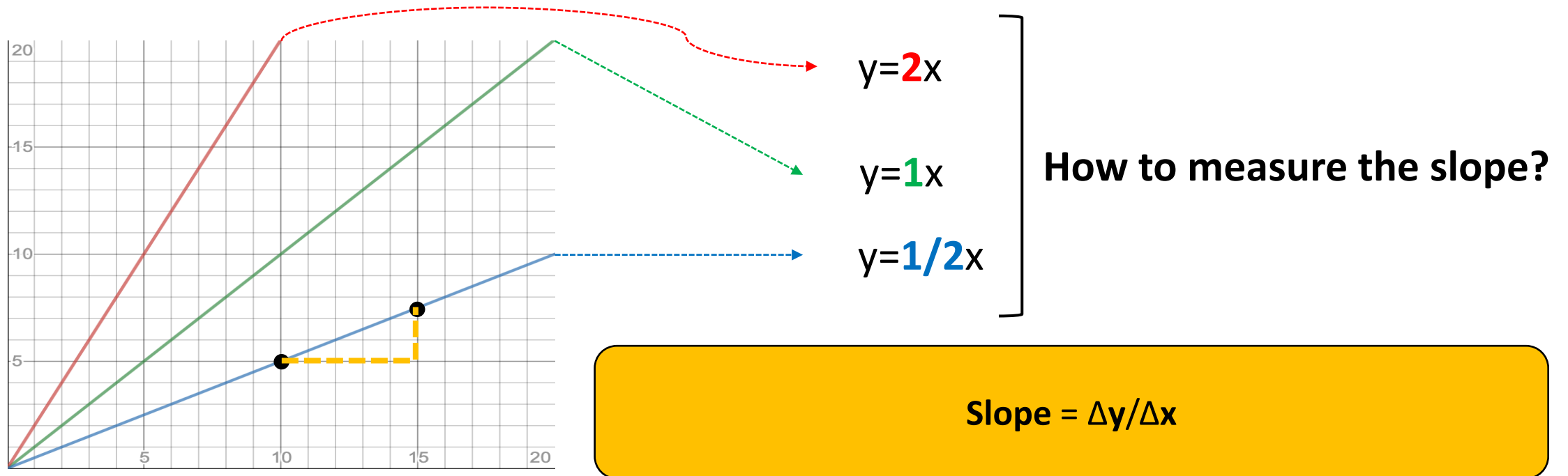
Slope

- Here are the 3 functions together



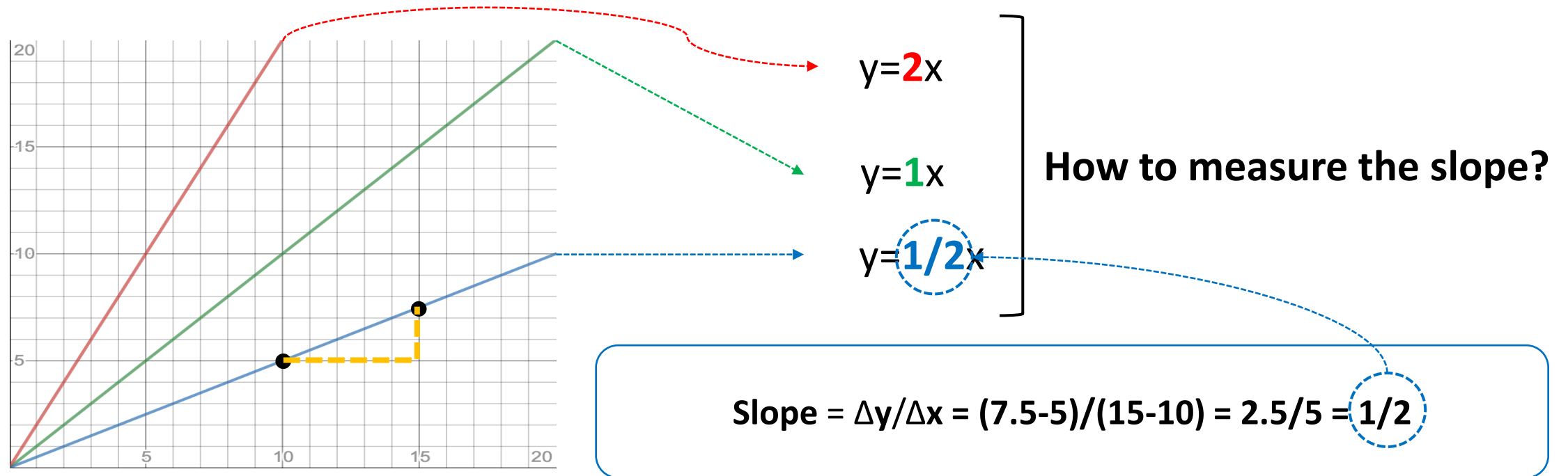
Slope

- Here are the 3 functions together



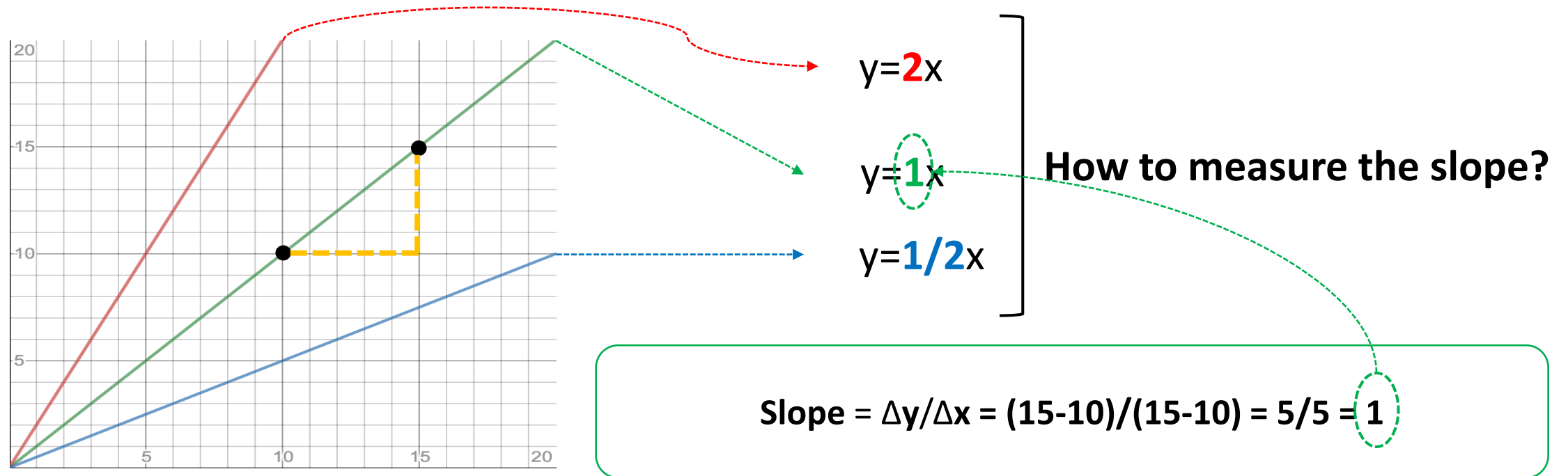
Slope

- Here are the 3 functions together



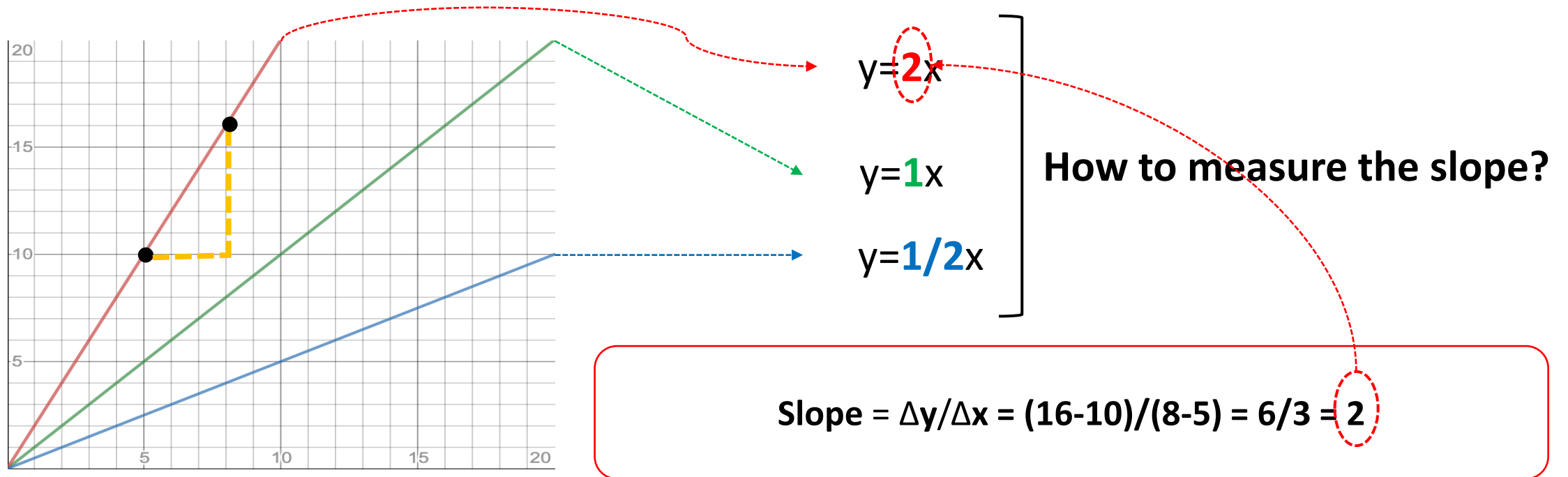
Slope

- Here are the 3 functions together



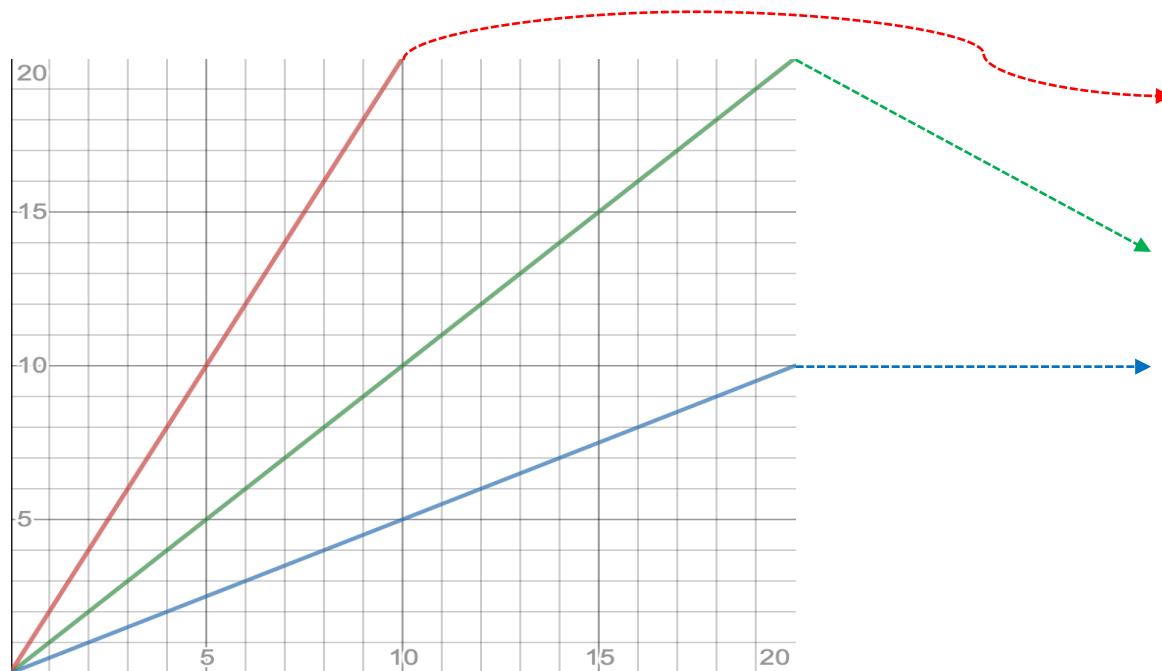
Slope

- Here are the 3 functions together



Slope

- Here are the 3 functions together



$$y = 2x$$

$$y = x$$

$$y = \frac{1}{2}x$$

What is the general formula of the line?

$$y = mx + b$$

slope


y-intercept, which describes where the line crosses the y-axis

Outline

- Subject overview
- What is AI
- Brief introduction to cyber security
- How AI helps Cyber security
- Limitations of AI in Security
- Math preliminaries
- **Linear regression**

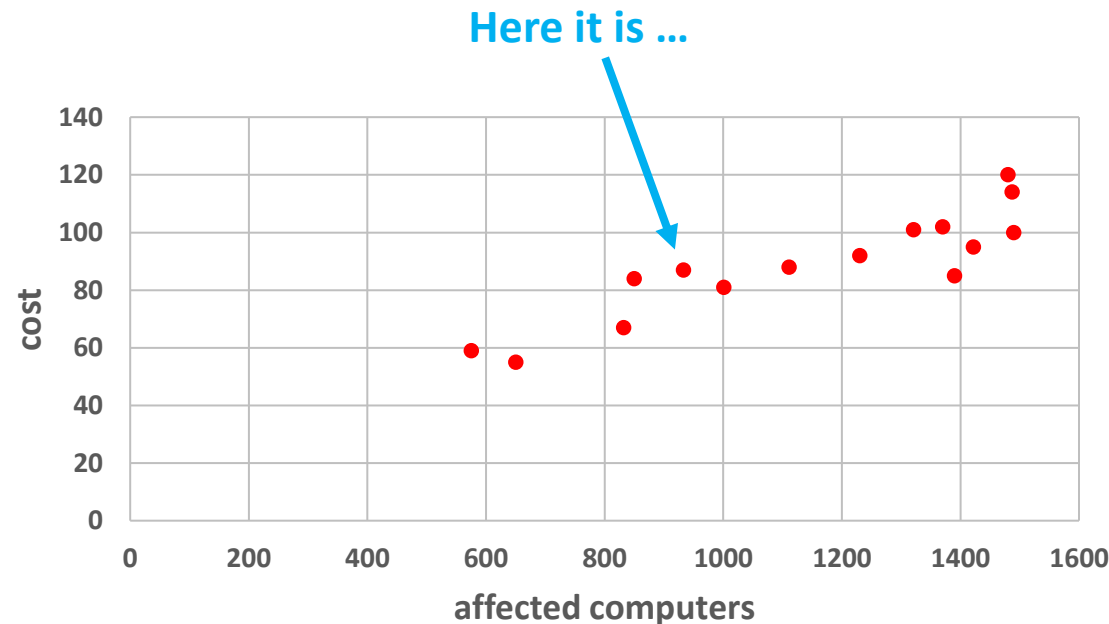
Example

- You are given the “right answer” (i.e., cost of security incident) for each example (i.e., #affected computers) in the data



# computer	cost
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



What is the cost of the security incident where there are **933** computers that have been affected?

Answer: 87

Scatter Plot

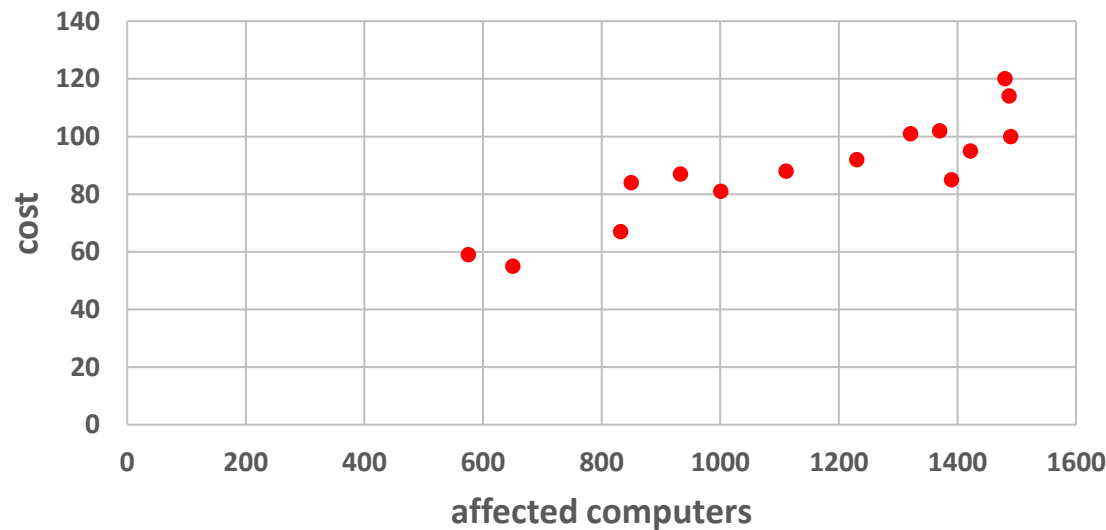
Example

- You are given the “right answer” (i.e., cost of security incident) for each example (i.e., #affected computers) in the data

?

# computer	cost
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



Scatter Plot

What is the cost of the security incident where there are **900** computers that have been affected?

Answer: NOT in the dataset!

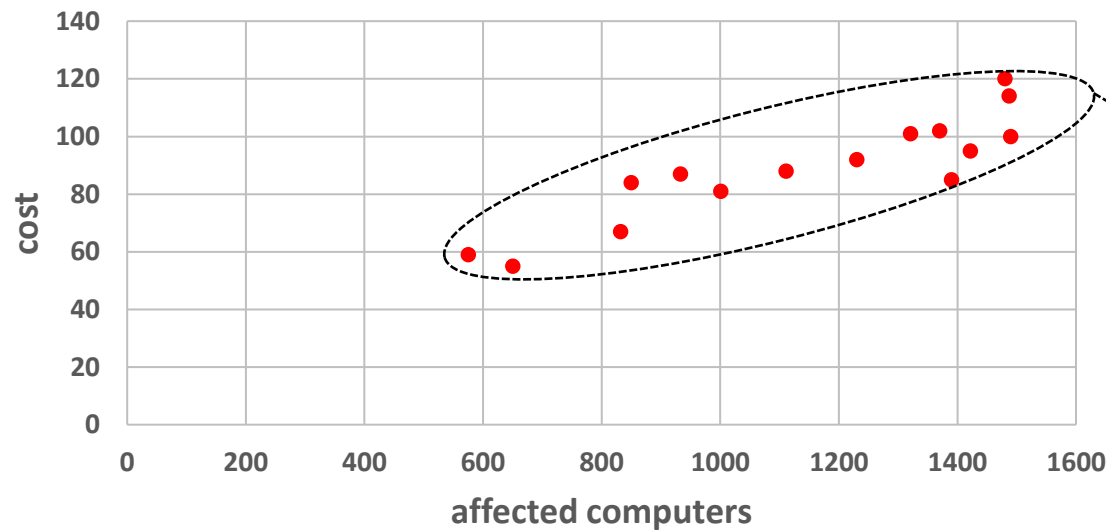
But we can *predict* it using machine learning

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset

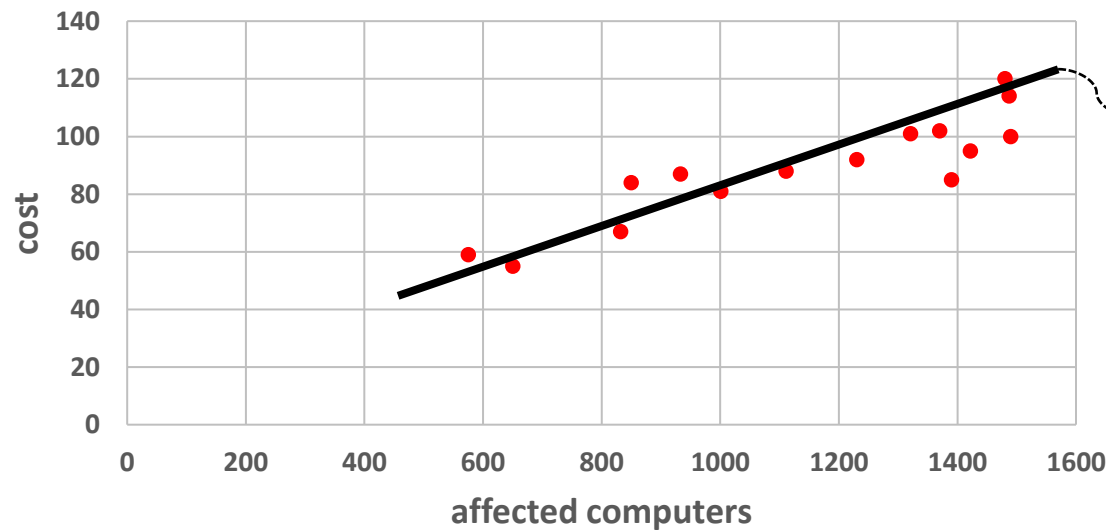


Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



Scatter Plot

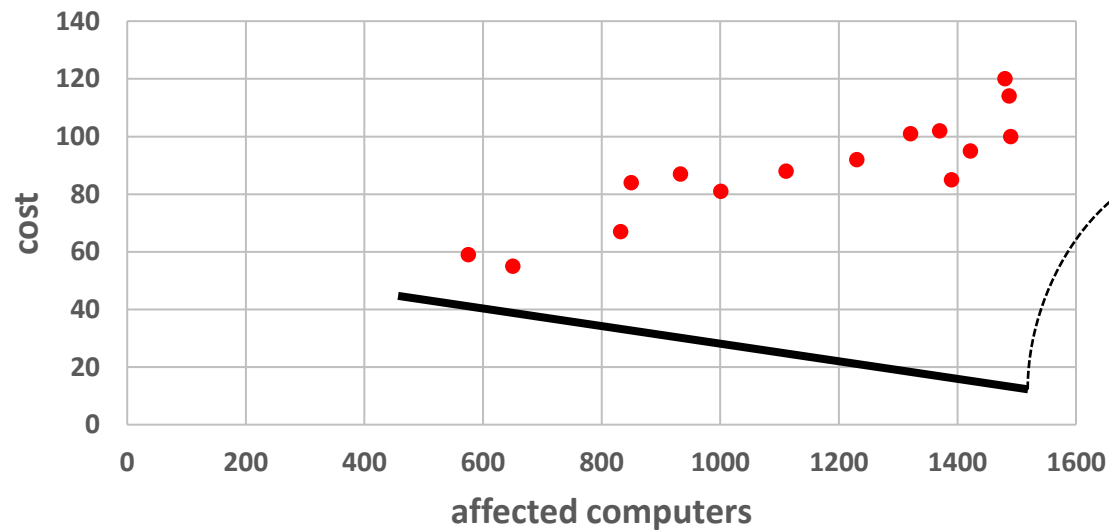
Which could be potentially captured (or fit) through a *line*

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



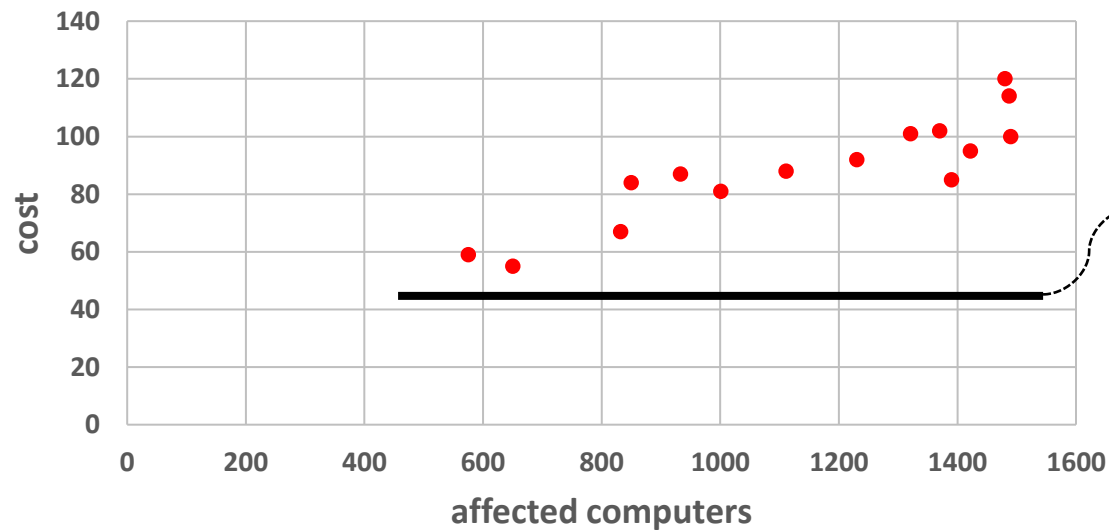
Scatter Plot

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



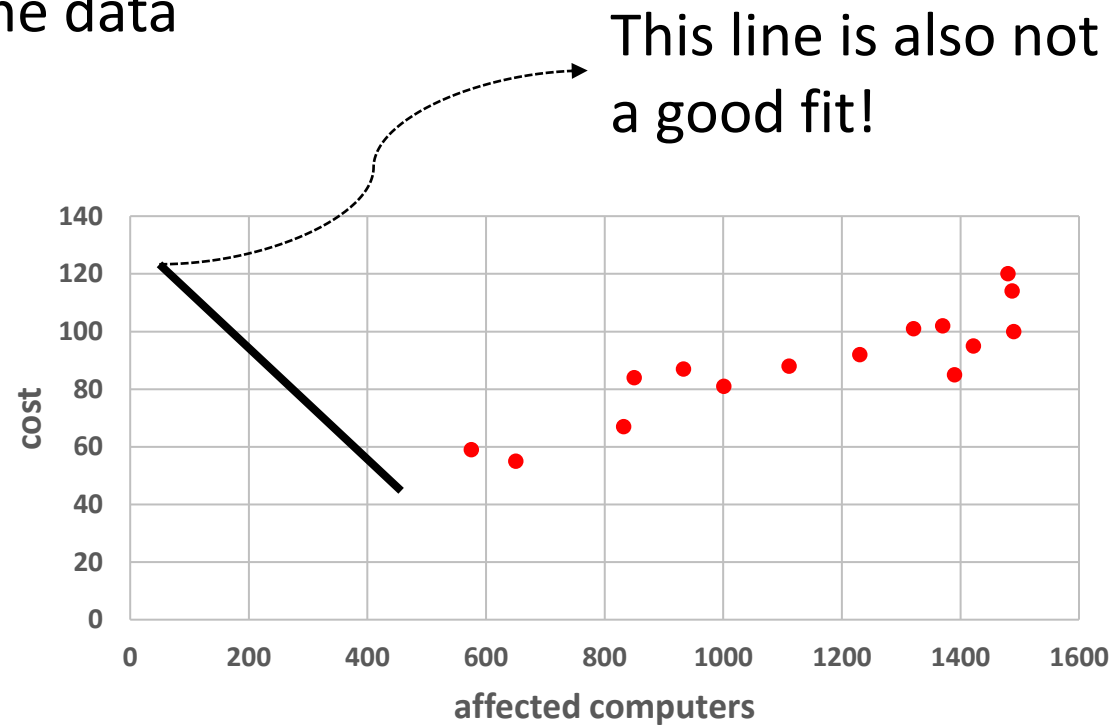
Scatter Plot

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



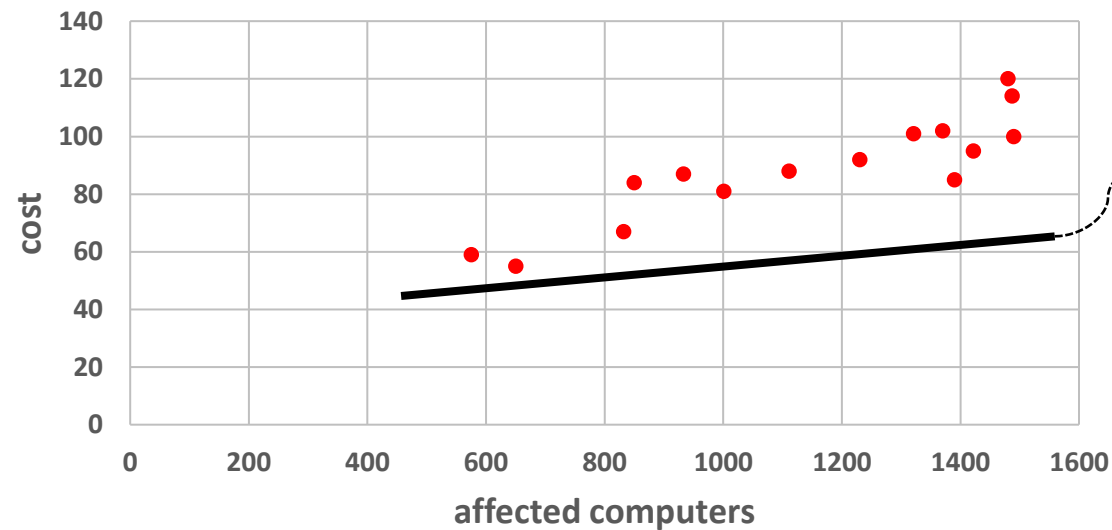
Scatter Plot

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



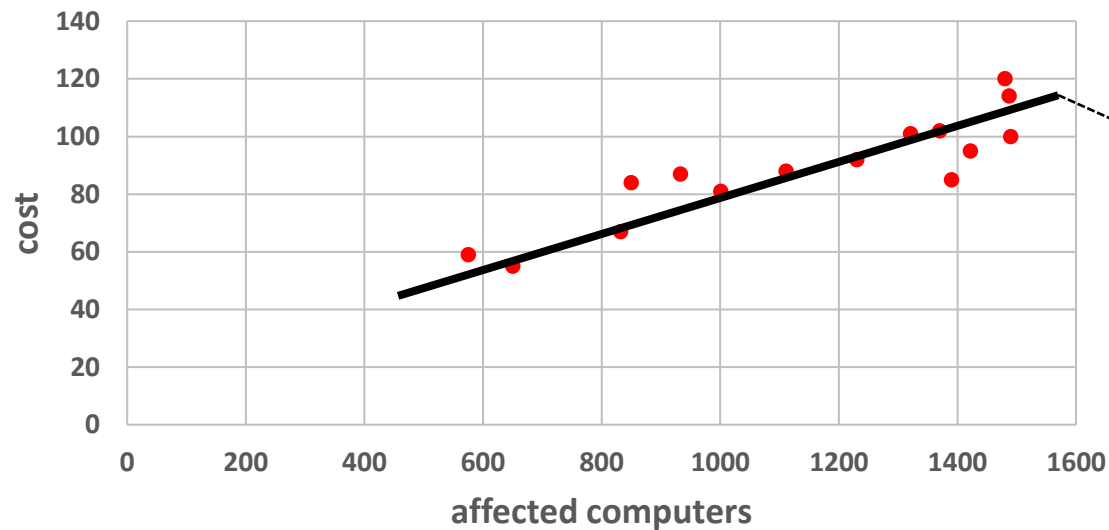
Scatter Plot

Predicting the Unknown

- How to start?
 - Let us visualize the data

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



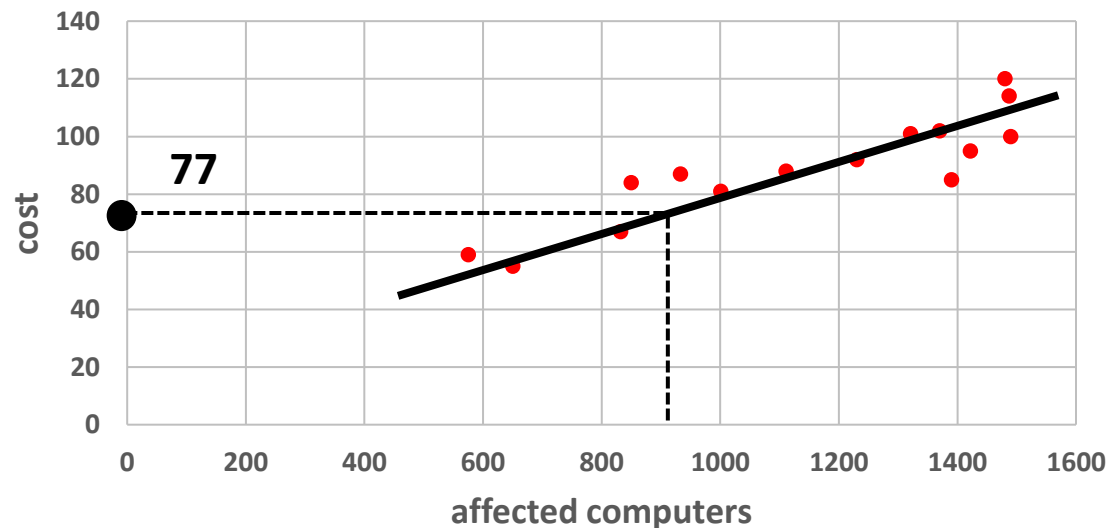
Scatter Plot

Predicting the Unknown

- Assume there is a way for us to find and fit this line around the given dataset, how can we use it to predict an unseen data point?

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



Scatter Plot

What will be the cost of the security incident where there are **900** computers that have been affected?

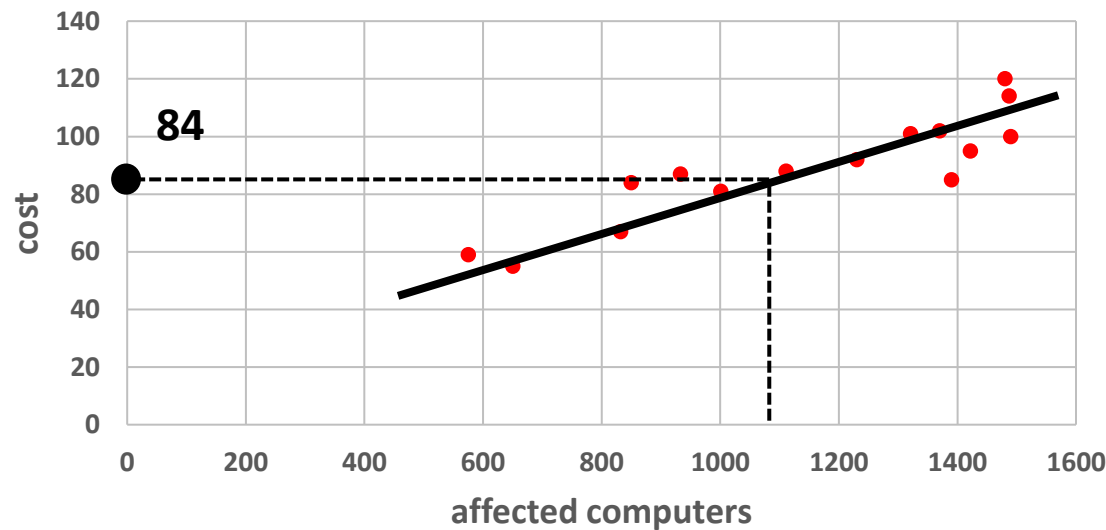
Answer: 77

Predicting the Unknown

- Assume there is a way for us to find and fit this line around the given dataset, how can we use it to predict an unseen data point?

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



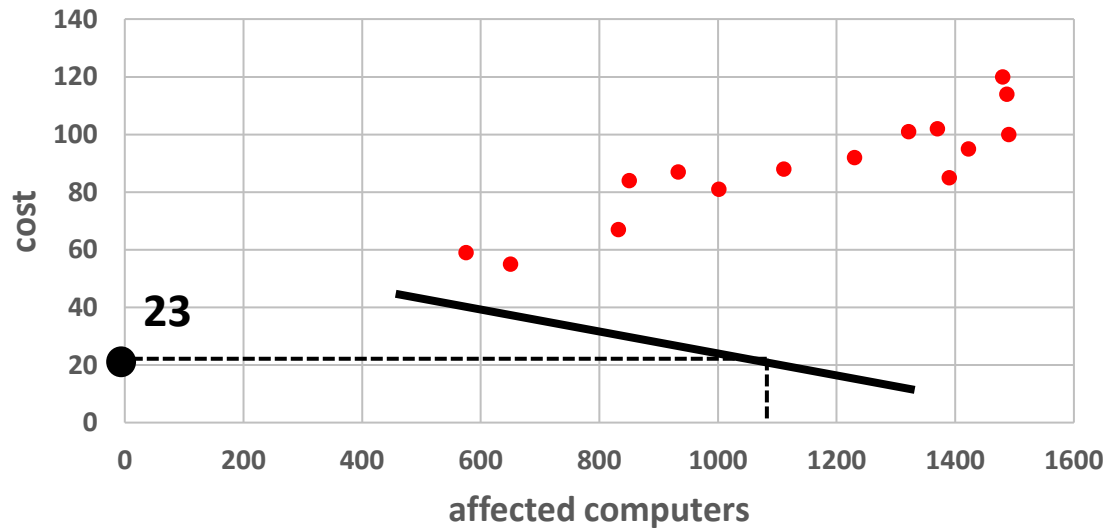
Scatter Plot

What will be the cost of the security incident where there are **1090** computers that have been affected?

Answer: 84

Predicting the Unknown

- The better the fit, the higher the accuracy of prediction

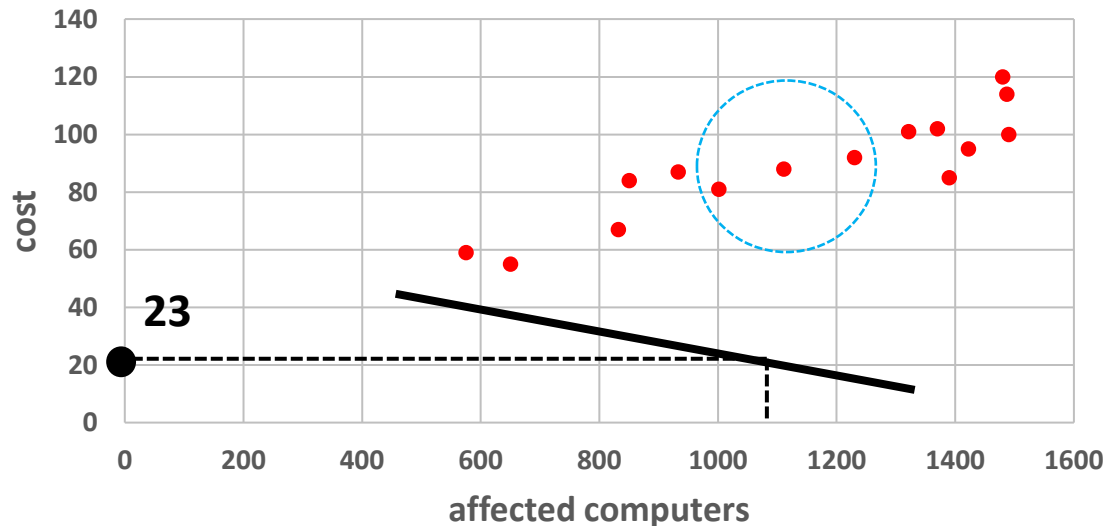


What will be the cost of the security incident where there are **1090** computers that have been affected?

Answer: 23

Predicting the Unknown

- The better the fit, the higher the accuracy of your predictions

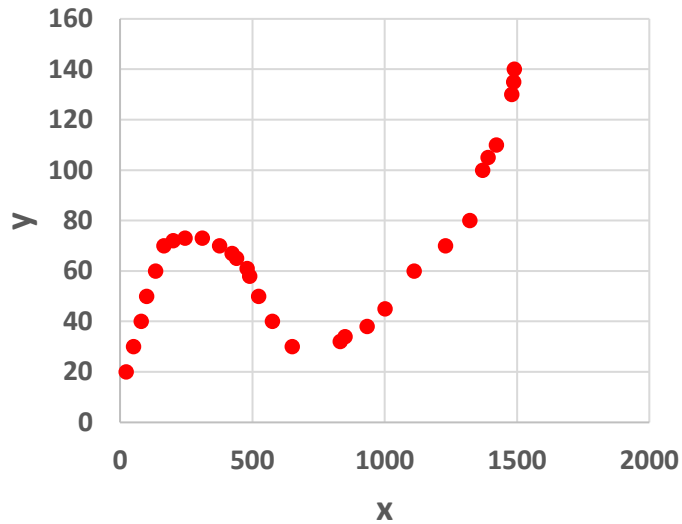


The prediction given by this line is unreasonable

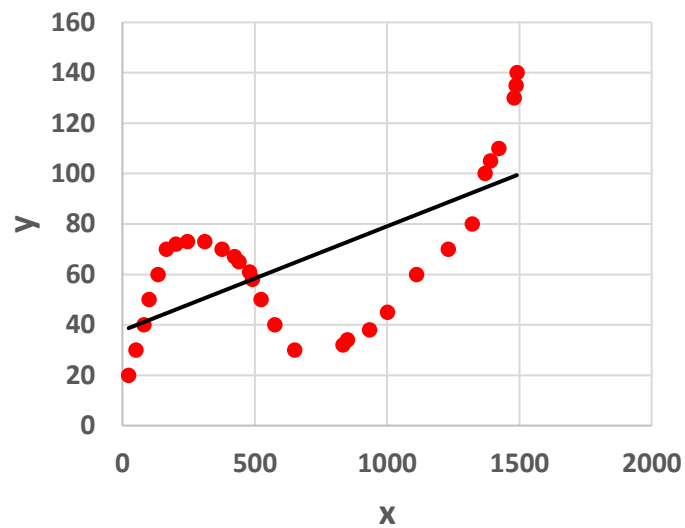
The whole idea becomes, how to fit a good function (here a *line*) which is more generally referred to as a **model**

Mathematical Models

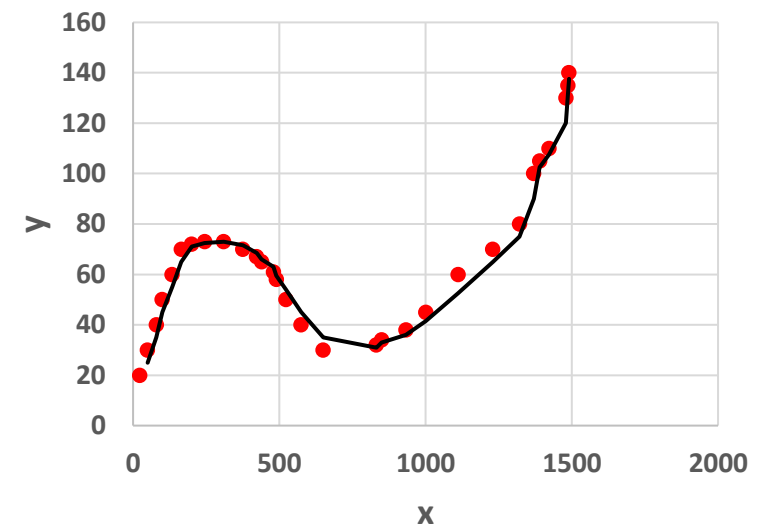
- Of course, a model needs not be a line!



Consider this data



Line is not a good fit



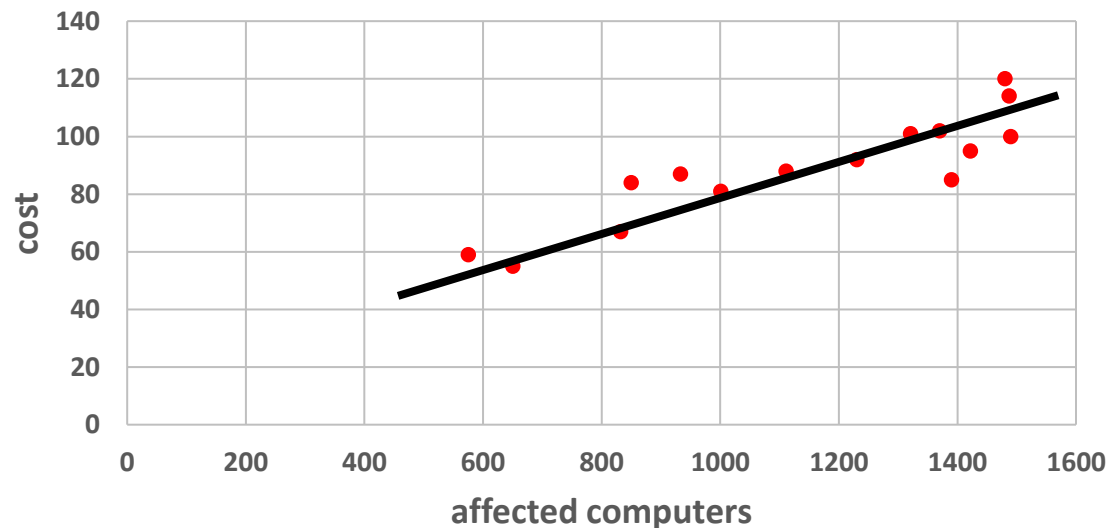
This model is a better fit

Learning a Model

- How to fit a mathematical model, after which you can predict any target value (e.g., cost) given a feature value (e.g., # computers)?

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



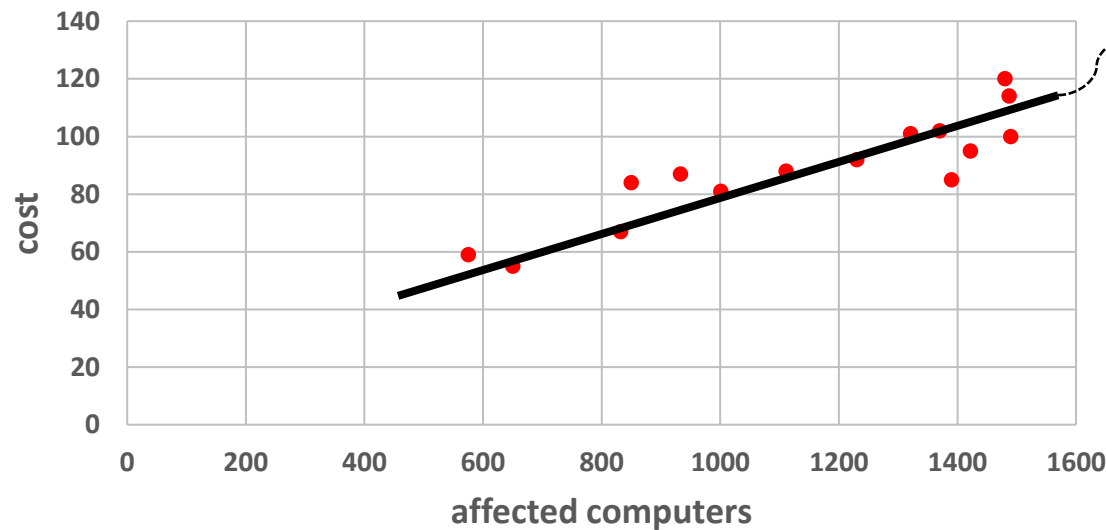
In other words, how to *learn* a model (e.g., a line) given a dataset?

Learning a Model

- How do we represent a line in mathematics?
 - $y = mx + b$

# computer (x)	cost (y)
575	59
650	55
832	67
850	84
933	87
1001	81
1111	88
1230	92
1321	101
1370	102
1390	85
1422	95
1480	120
1487	114
1490	100

Dataset



For this best fitting line:

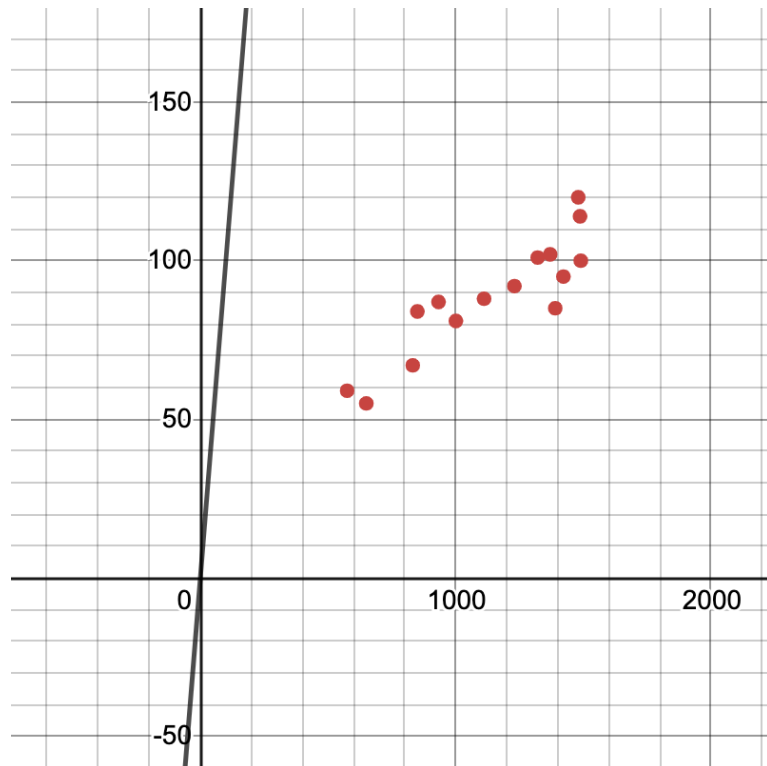
$$m = 0.052$$

$$b = 29.21$$

But, how can we find
(or *learn*) m and b ?

Learning a Model

- Let us try different random values of m and b :
 - $m = 1$ and $b = 0$

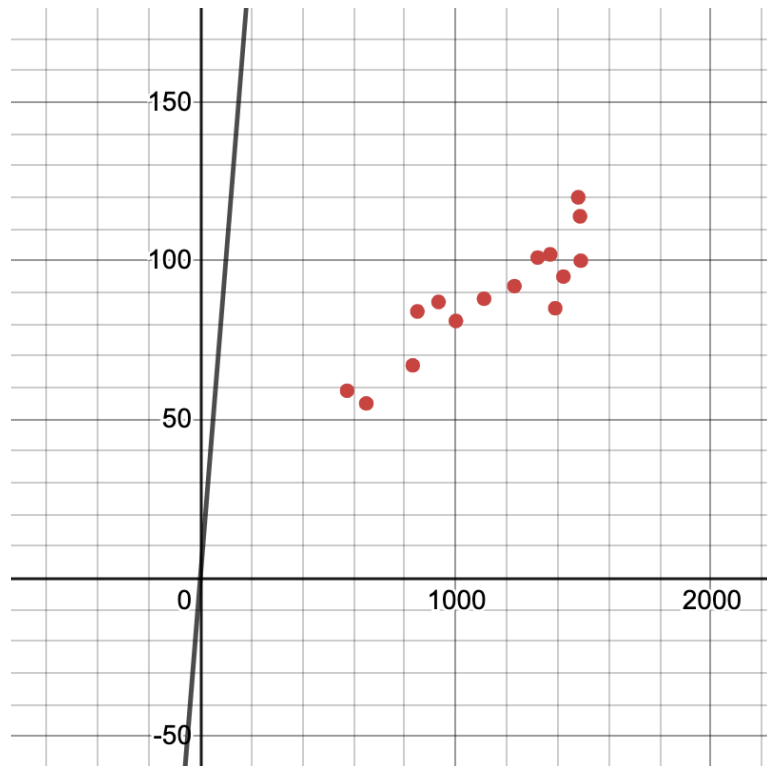


# computer (x)	Actual cost (y)	Predicted cost (y')
575	59	575
650	55	650
832	67	832
850	84	850
933	87	933
1001	81	1001
1111	88	1111
1230	92	1230
1321	101	1321
1370	102	1370
1390	85	1390
1422	95	1422
1480	120	1480
1487	114	1487
1490	100	1490

$$\begin{aligned}y' &= mx + b \\ &= 1 * 575 + 0 \\ &= 575\end{aligned}$$

Learning a Model

- Let us try different random values of m and b :
 - $m = 1$ and $b = 0$



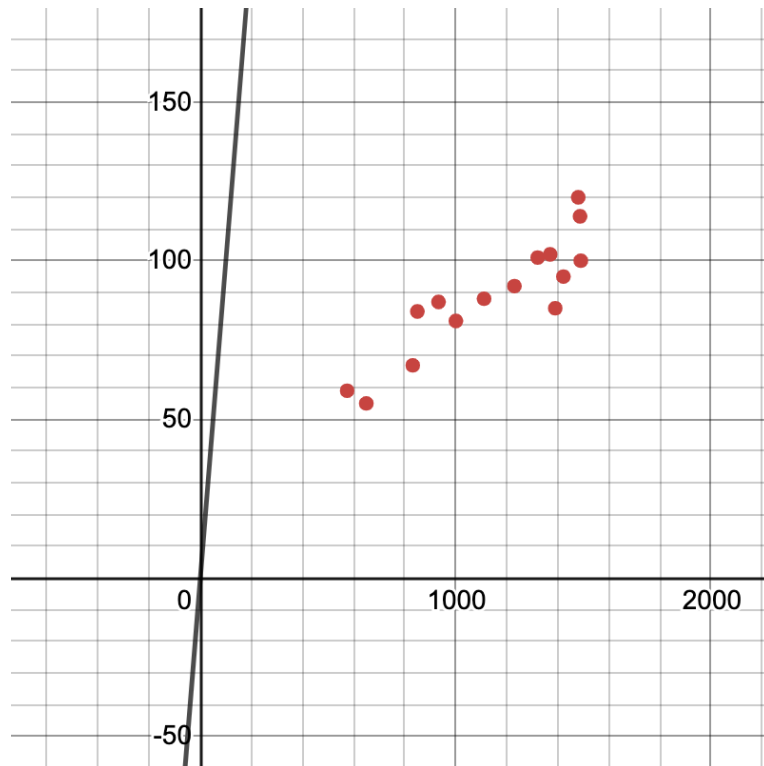
# computer (x)	Actual cost (y)	Predicted cost (y')
575	59	575
650	55	650
832	67	832
850	84	850
933	87	933
1001	81	1001
1111	88	1111
1230	92	1230
1321	101	1321
1370	102	1370
1390	85	1390
1422	95	1422
1480	120	1480
1487	114	1487
1490	100	1490

$$\begin{aligned}y' &= mx + b \\ &= 1 * 1001 + 0 \\ &= 1001\end{aligned}$$

How close are these predicted cost to the actual ones?

Learning a Model

- Let us try different random values of m and b :
 - $m = 1$ and $b = 0$



# computer (x)	Actual cost (y)	Predicted cost (y')	Predicted – Actual ($y' - y$)
575	59	575	516
650	55	650	595
832	67	832	765
850	84	850	766
933	87	933	846
1001	81	1001	920
1111	88	1111	1023
1230	92	1230	1138
1321	101	1321	1220
1370	102	1370	1268
1390	85	1390	1305
1422	95	1422	1327
1480	120	1480	1360
1487	114	1487	1373
1490	100	1490	1390

The Error

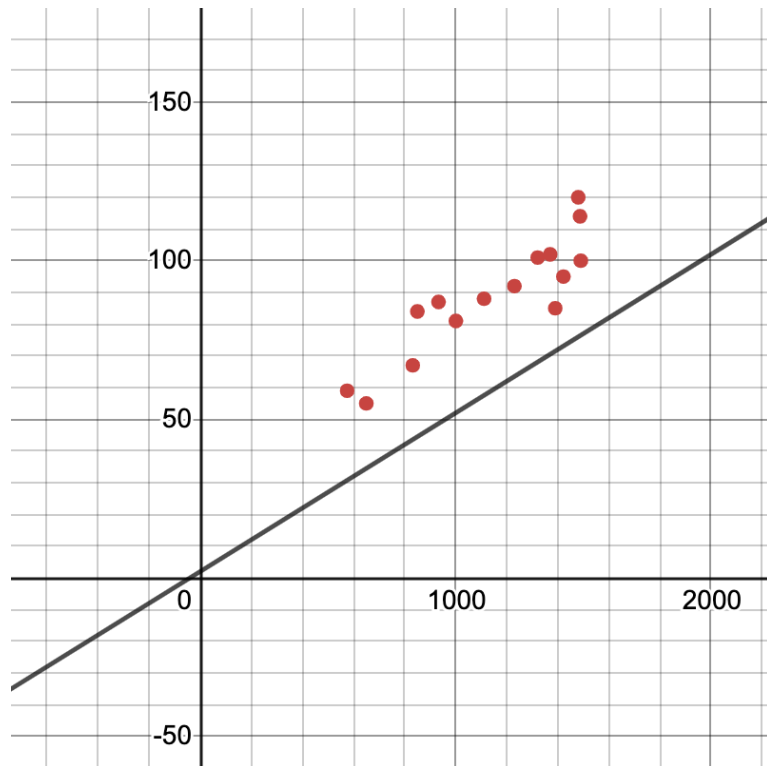
Sum = 15812

≡

$\Sigma = 15812$

Learning a Model

- Let us try different random values of m and b :
 - $m = 0.05$ and $b = 2$



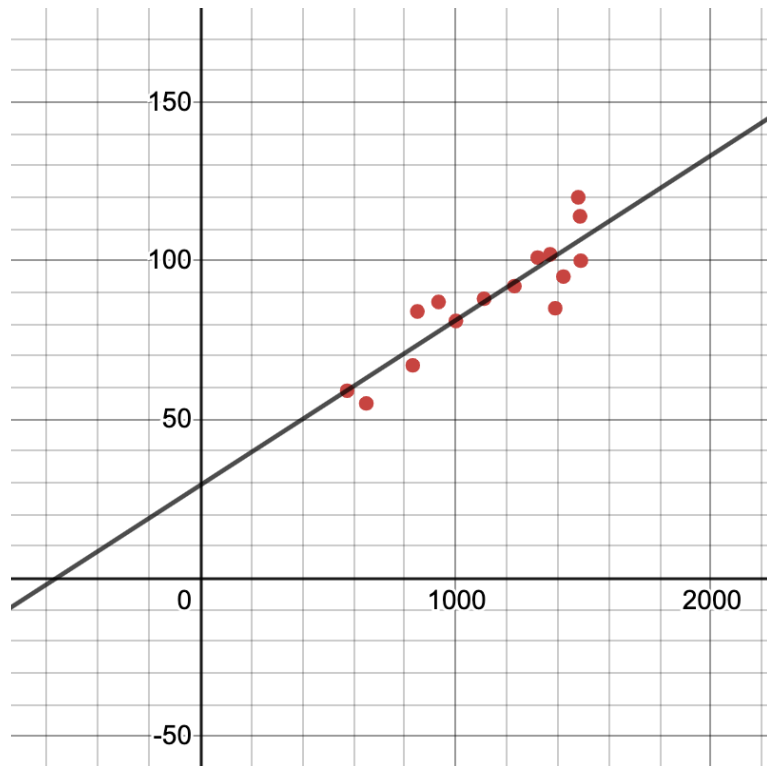
# computer (x)	Actual cost (y)	Predicted cost (y')	Predicted – Actual ($y' - y$)
575	59	30.75	-28.25
650	55	34.5	-20.5
832	67	43.6	-23.4
850	84	44.5	-39.5
933	87	48.65	-38.35
1001	81	52.05	-28.95
1111	88	57.55	-30.45
1230	92	63.5	-28.5
1321	101	68.05	-32.95
1370	102	70.5	-31.5
1390	85	71.5	-13.5
1422	95	73.1	-21.9
1480	120	76	-44
1487	114	76.35	-37.65
1490	100	76.5	-23.5

Error:

$\Sigma = -442.9$

Learning a Model

- Let us try different random values of m and b :
 - $m = 0.052$ and $b = 29.21$



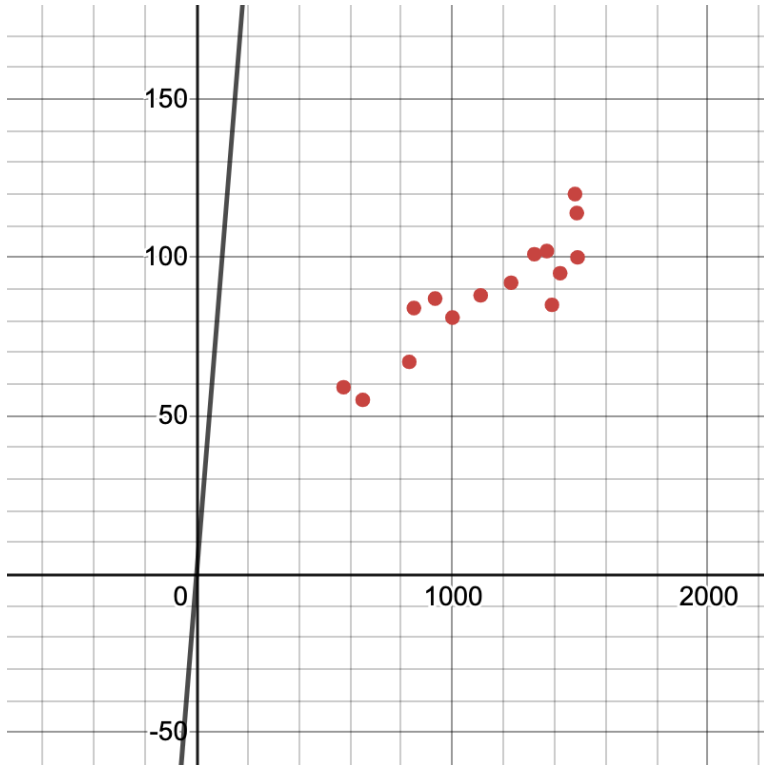
# computer (x)	Actual cost (y)	Predicted cost (y')	Predicted – Actual ($y' - y$)
575	59	59.11	0.11
650	55	63.01	8.01
832	67	72.474	5.474
850	84	73.41	-10.59
933	87	77.726	-9.274
1001	81	81.262	0.262
1111	88	86.982	-1.018
1230	92	93.17	1.17
1321	101	97.902	-3.098
1370	102	100.45	-1.55
1390	85	101.49	16.49
1422	95	103.154	8.154
1480	120	106.17	-13.83
1487	114	106.534	-7.466
1490	100	106.69	6.69

Error:

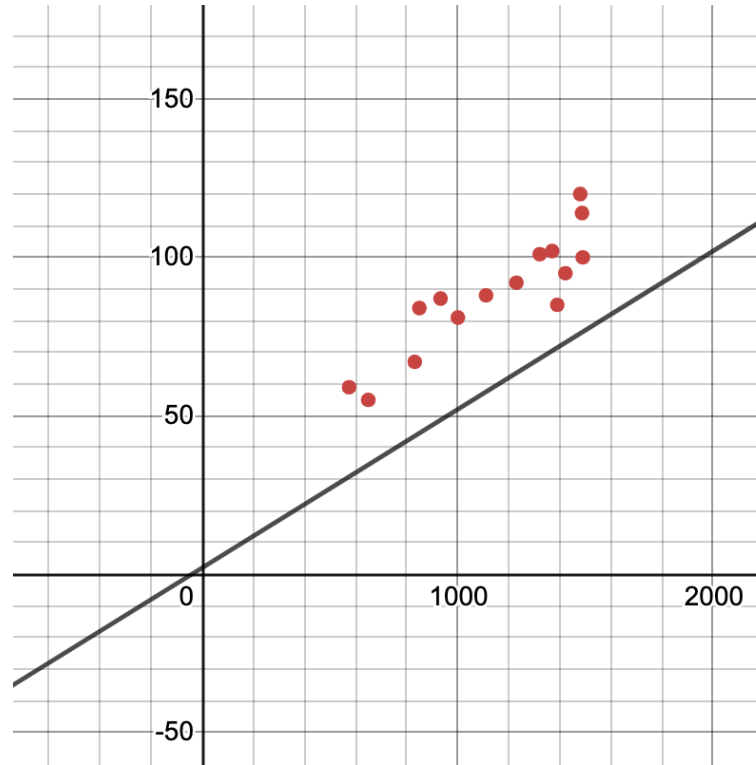
$\Sigma = -0.466$

Learning a Model

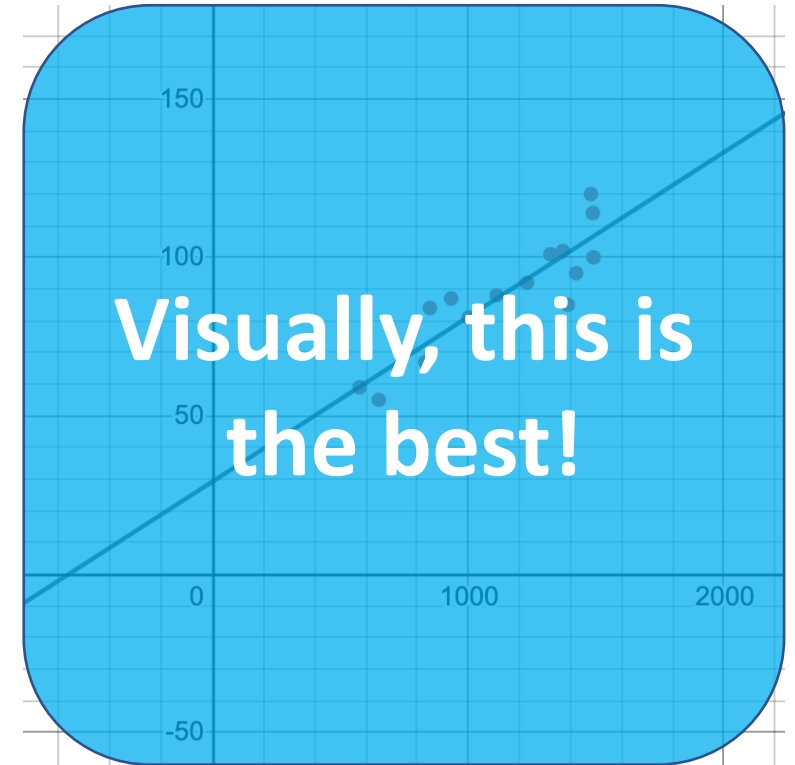
- Let us observe the three options besides each other



$m = 1$ and $b = 0$



$m = 0.05$ and $b = 2$



$m = 0.052$ and $b = 29.21$

Learning a Model

- Let us compare their errors

Learning a Model

- Let us compare their errors

$m = 1$ and $b = 0$

$m = 0.05$ and $b = 2$

$m = 0.052$ and $b = 29.21$

# computer (x)	Actual cost (y)	Predicted cost (y')	(Predicted – Actual) ² ($y' - y$) ²	Predicted cost (y')	(Predicted – Actual) ² ($y' - y$) ²	Predicted cost (y')	(Predicted – Actual) ² ($y' - y$) ²
575	59	59.11	266256	30.75	798.0625	59.11	0.0121
650	55	63.01	354025	34.5	420.25	63.01	64.1601
832	67	72.474	585225	43.6	547.56	72.474	29.964676
850	84	73.41	586756	44.5	1560.25	73.41	112.1481
933	87	77.726	715716	48.65	1470.7225	77.726	86.007076
1001	81	81.262	846400	52.05	838.1025	81.262	0.068644
1111	88	86.982	1046529	57.55	927.2025	86.982	1.036324
1230	92	93.17	1295044	63.5	812.25	93.17	1.3689
1321	101	97.902	1488400	68.05	1085.7025	97.902	9.597604
1370	102	100.45	1607824	70.5	992.25	100.45	2.4025
1390	85	101.49	1703025	71.5	182.25	101.49	271.9201
1422	95	103.154	1760929	73.1	479.61	103.154	66.487716
1480	120	106.17	1849600	76	1936	106.17	191.2689
1487	114	106.534	1885129	76.35	1417.5225	106.534	55.741156
1490	100	106.69	1932100	76.5	552.25	106.69	44.7561

SQUARED ERRORS: $\Sigma = 17922958$

$\Sigma = 14019.9$

$\Sigma = 936.9$



Recap

- What is the idea?
 - start with a generic *model* or a *hypothesis* (e.g., a line),
 - then keep varying its parameters (i.e., m and b for the line) until you get the *minimum squared error*
 - This process is called *learning*
 - And an algorithm that can be used to vary the parameters to locate the minimum squared error is called *gradient descent*
- Once you learn a model, you can use it to predict unseen values!

References

- *Machine Learning , Analytics & Cyber Security the Next Level Threat Analytics, Manjunath N V*
- *Data mining for security at Google, Max Poletto Google security team*
- *WSABIE: scaling up to large vocabulary image annotation. Jason Weston, Samy Bengio, and Nicolas Usunier*
- *Machine learning foundational course, google developers*
- *AI for Medicine, Mohammad Hammoud, CMU Qatar*
- *Practical Machine Learning in Infosecurity, Clarence Chio and Anto Joseph*
- *Machine Learning Core Concept, Mohammad Hammoud, CMU in Qatar*
- *Linear regression, Mohammad Hammoud, CMU in Qatar*