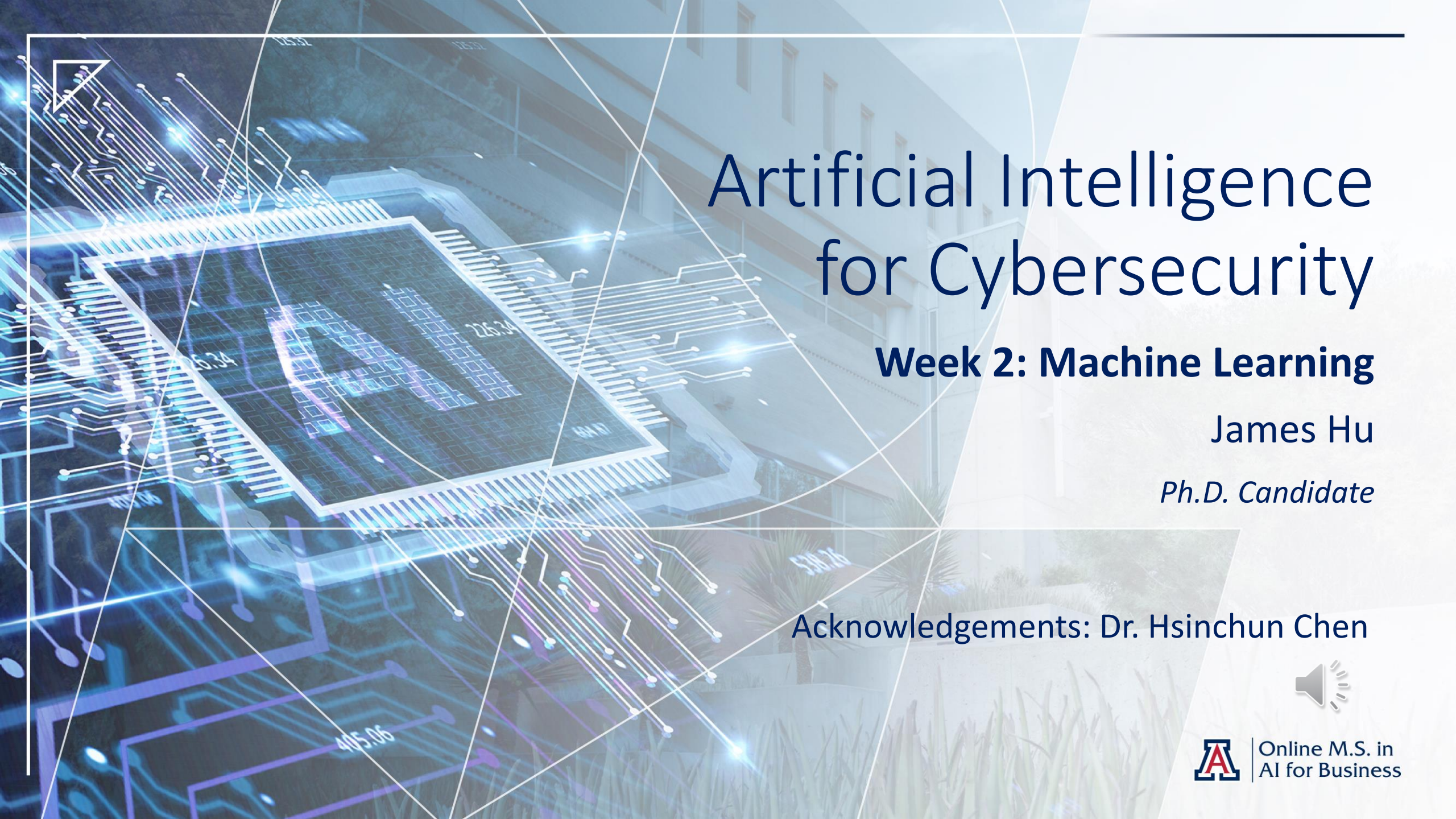




Artificial Intelligence for Cybersecurity

Week 2: Machine Learning

James Hu

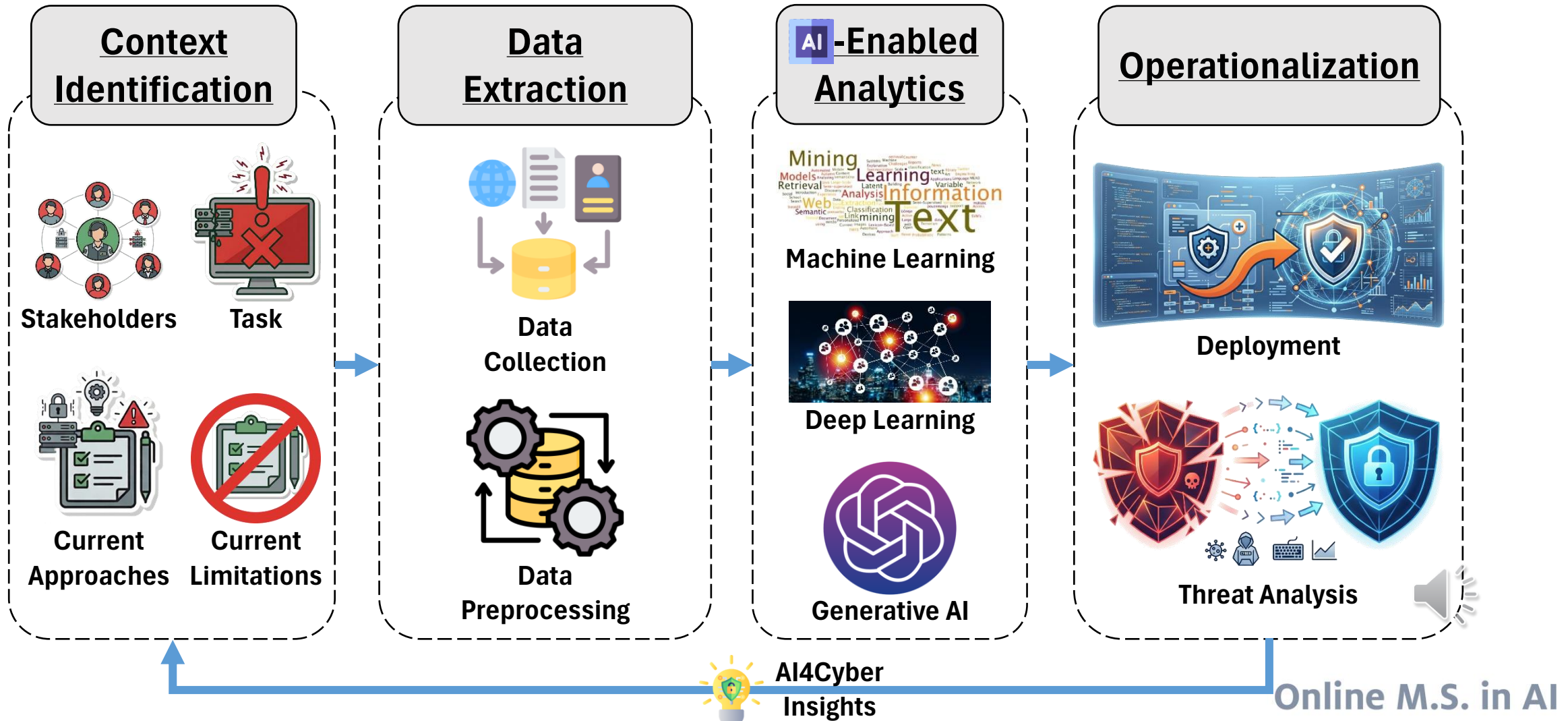
Ph.D. Candidate

Acknowledgements: Dr. Hsinchun Chen



Online M.S. in
AI for Business

From Week 1: AI4Cyber Lifecycle





Today's Learning Objectives

- ▶ Understand how machines can learn patterns from big data.
- ▶ Understand the core differences between unsupervised and supervised learning.
- ▶ Explain how **K-means** and **Latent Dirichlet Allocation (LDA)** perform unsupervised learning.
- ▶ Explain how **Decision Trees** and **Support Vector Machines (SVM)** perform supervised learning.



How do Machines Learn?

- ▶ How do humans learn?
- ▶ Imagine studying for a test. You may undertake the following steps:
 1. Review your study notes to answer a practice problem
 2. Grade the answer, focusing on your mistakes
 3. Revise your understanding based on your mistakes
 4. Repeat until you are ready for the test.



How do Machines Learn?

- ▶ How do humans learn?
- ▶ Imagine studying for a test. **You** may undertake the following steps:
 1. Review your **study notes** to **answer** a practice problem
 2. Grade the **answer**, focusing on your **mistakes**
 3. Revise your **understanding** based on your **mistakes**
 4. Repeat until you are ready for the test.
- ▶ Machine Learning (ML) is similar.
- ▶ To learn a task, **ML models** undergo training:
 1. **Model Input & Output:** Inputs a **datapoint** to produces an **output**.
 2. **Loss Calculation:** Use a loss function to calculate the **output's loss value**
 3. **Optimization:** Adjust the model's **parameters** based on the **loss value**.
 4. Repeat until the model converges. 

Overview of Classes of ML Models

- ▶ For the purposes of this class, we can describe ML models (h) based on its:
 - ▶ **Input (x):** What the model perceives.
 - ▶ **Parameters (θ):** The model's "numerical dials" controlling how it answers.
 - ▶ **Output ($\hat{y} := h_{\theta}(x)$):** Given θ , what happens inside the model to produce an answer.
 - ▶ **Loss Function ($\mathcal{L}(\hat{y})$):** The criteria determining how "wrong" the model's answer is.
 - ▶ **Optimization:** How the model changes its parameters to produce "less wrong" answers.
- ▶ **Core Intuition:** When learning, a model aims to find the optimal parameter (θ^*) that produces the overall least wrong answers.
 - ▶ Often happens iteratively, with the model slightly adjusting its parameters many times.



Tokenization

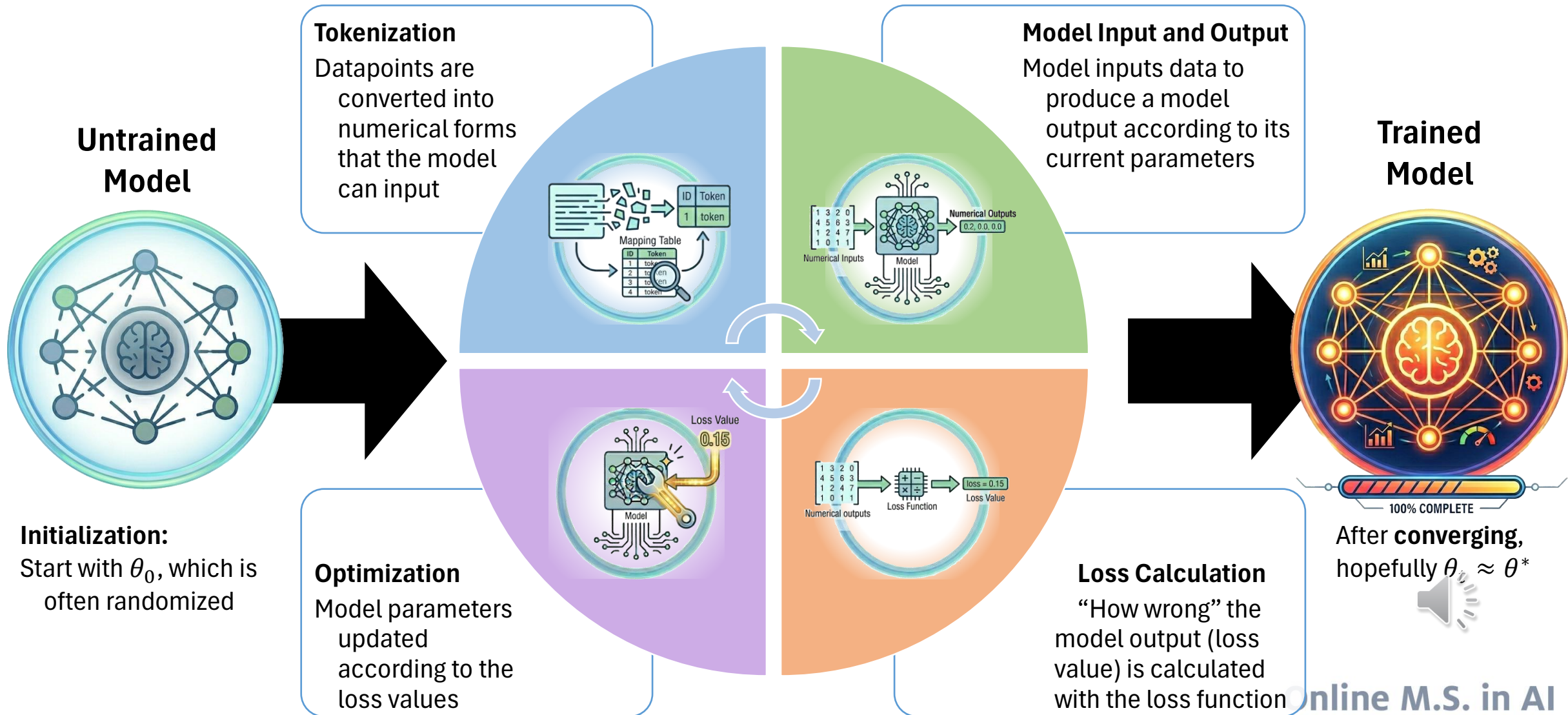
- ▶ However, unlike humans, machines can only “perceive” numbers.
 - ▶ **Tokenization:** Converting non-numerical features into numbers for ML models

Data Type	Tokenization Procedure	Example Untokenized datapoint	Example Tokenized datapoint
Numerical	None, already in number form	67	[67]
Textual	Assign a number to each unique token (i.e., character, words, etc.)	i like cats	[9, 0, 12, 9, 11, 5, 0, 3, 1, 20, 19]
Categorical	Assign a number to each group or category	Employed, unemployed	[1, 0]
Datetime	Treated as mix of categorical (day, month, year) and numerical (time)	1:09am on March 7 th	[69, 3, 7]
Images	Leverage each pixel's red, green, and blue values		[112, 173, 71], [237, 125, 49], [112, 48, 160]

- ▶ Also, machines can only output numbers.
 - ▶ Converting model outputs back into human-readable data often involves the tokenization process in reverse.



The ML Training Loop



Key to ML Effectiveness

- ▶ Fundamentally, all AI models are math equations that extract a dataset's patterns into its parameters.
- ▶ **Example Dataset Patterns:**
 - ▶ **Simple:** Datapoints with a certain feature inside a certain range (external IP addresses) correspond with a certain category (untrusted devices).
 - ▶ **Complex:** Certain sequences of text (Exam Question) correspond to other certain sequences of text (Exam Answer).

informatics

Article

Quality Assessment of Generative AI in Cybersecurity Certification

Vanessa G. Félix ^{1,2}, Rodolfo Ostos ^{1,2}, Luis J. Mena ¹, Homero Toral-Cruz ³, Alberto Ochoa-Brust ⁴, Pablo Velarde-Alvarado ⁵, Apolinar González-Potes ⁴, Ramón A. Félix-Cuadras ⁴, José A. León-Borges ⁶ and Rafael Martínez-Peláez ^{1,7,*}

4.4. Phase 2 Results: Full Practice Examination Performance

Table 2 summarizes the results of the full 100-question practice examination. All models exceeded typical Security+ passing thresholds, with Gemini-2.5 Pro achieving 97%, Copilot-2.5 Pro 96%, and ChatGPT-5 93%.

Table 2. Full practice exam results (100 questions).

Model	Correct	Accuracy
ChatGPT-5	93	93%
Copilot-2.5 Pro	96	96%
Gemini-2.5 Pro	97	97%

The findings indicate that LLMs can perform consistently well under summative, certification-like conditions, in which questions are presented in a continuous, mixed-domain format. This raises important considerations for assessment design, particularly regarding the ability of traditional multiple-choice formats to differentiate between human learners and AI-assisted responses in environments where such tools are accessible.

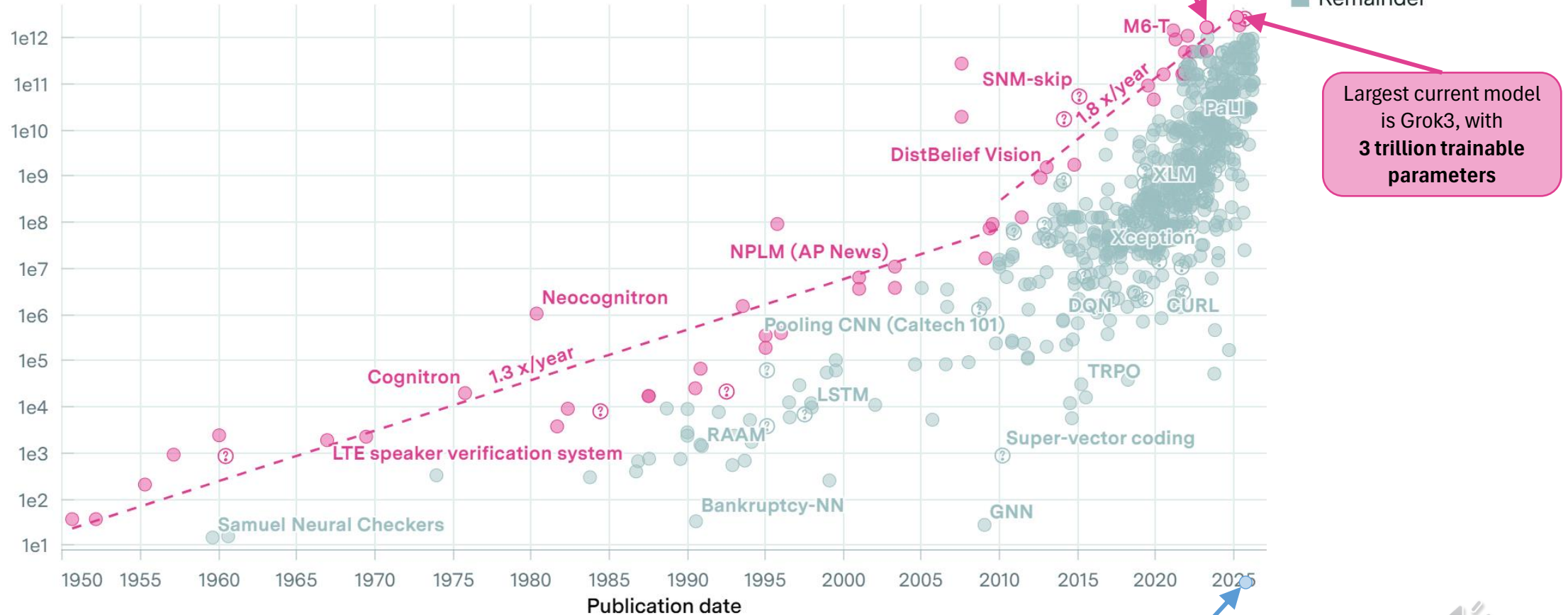
Key to ML Effectiveness

- ▶ Digitalization has increased the volume and complexity of data.
 - ▶ Larger and more complex datasets contain more comprehensive and complicated patterns.
- ▶ **Cybersecurity Examples:**
 - ▶ Is there a pattern hidden among cve.org's 229K published reports that tells us about overall trends in cyberattacks?
 - ▶ Is there a pattern connecting an email's text (~6k common English words) and if it is phishing?
 - ▶ What is the pattern between your study materials and the correct answer on the CompTIA Security+ exam?
- ▶ To be able to learn these patterns, AI models must also become more complicated.



Notable AI models

Number of trainable parameters



From Week 1: AI-enabled Analytic



Descriptive analytic:

- **Purpose:** Summarize and interpret historical data to understand past events.
- **Question to answer:** What happened?
- Often supported by Unsupervised Learning



Predictive analytic:

- **Purpose:** Forecast future outcomes or probabilities based on historical data
- **Question to answer:** What will happen?
- Often supported by Supervised Learning

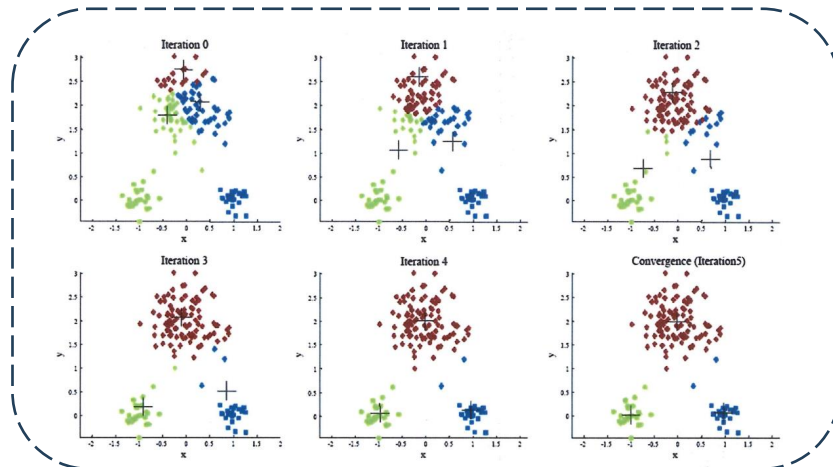


Prescriptive analytic:

- **Purpose:** Recommend specific actions to take to achieve the objective.
- **Question to answer:** What should I do?
- Often supported by Agentic AI



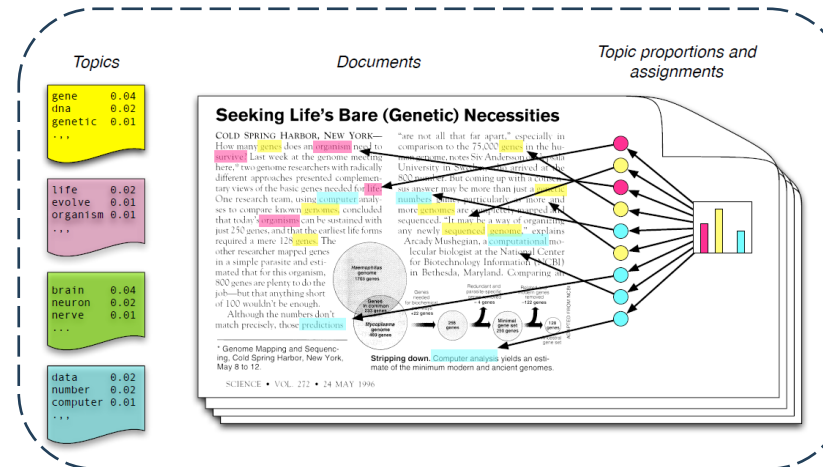
From Week 1: Descriptive Methods



Clustering

Examples

- **Network Device Segmentation:** Grouping devices in a network based on similar network behaviors.



Topic Modeling

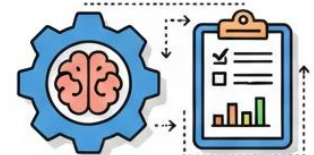
Examples

- **Dark Web Discovery:** Parsing posts from hacker forums or Reddit to identify emerging adversary trends.

1. Data



2. Training

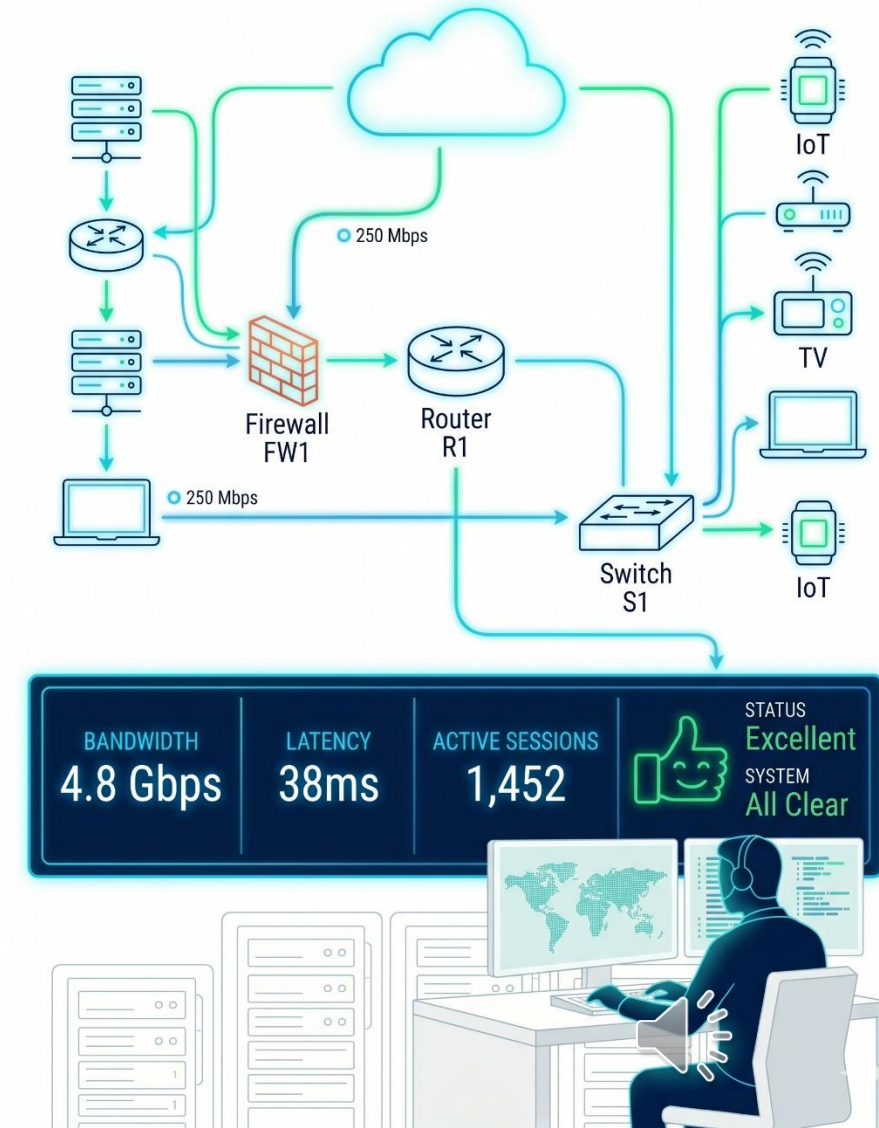


3. Insights



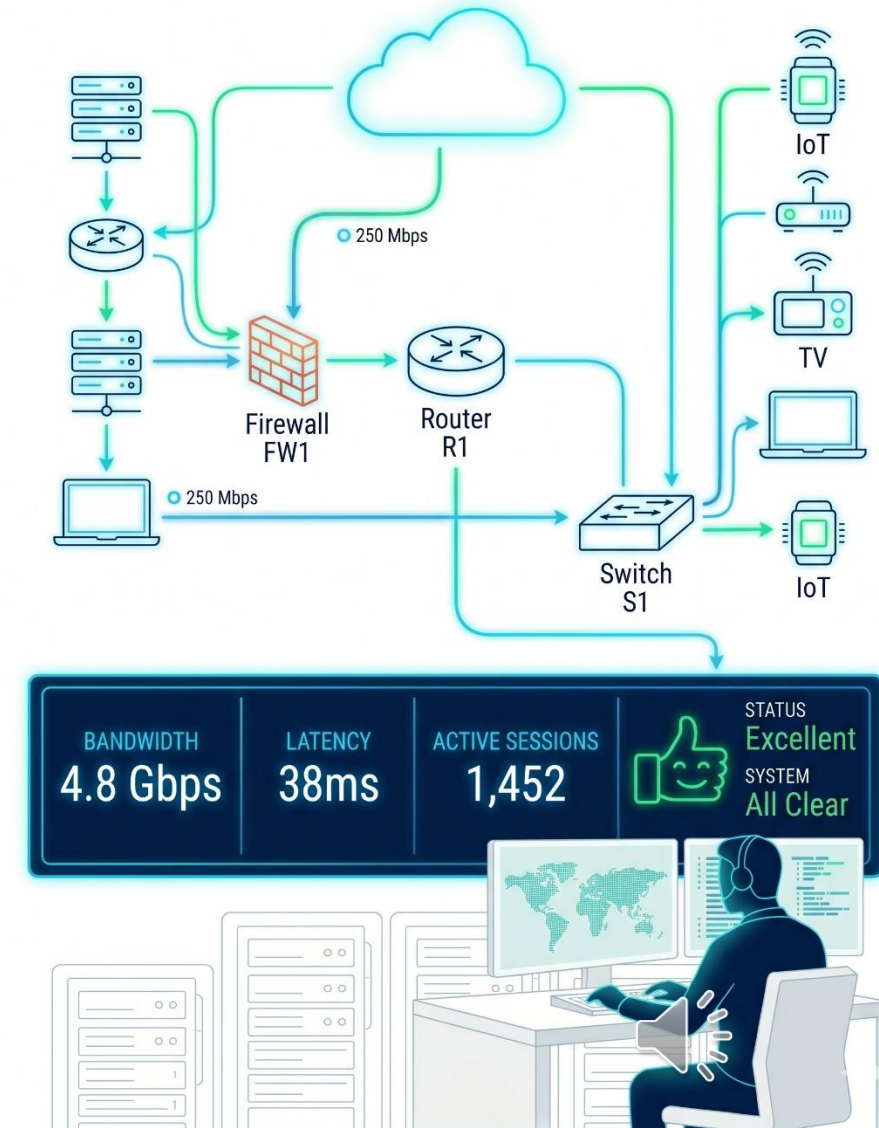
Clustering – Motivation

- Suppose you are a company's network administrator monitoring network traffic of connected devices. You can observe:
 - Source/destination IPs
 - Port number
 - Payload size
 - Connection duration and time



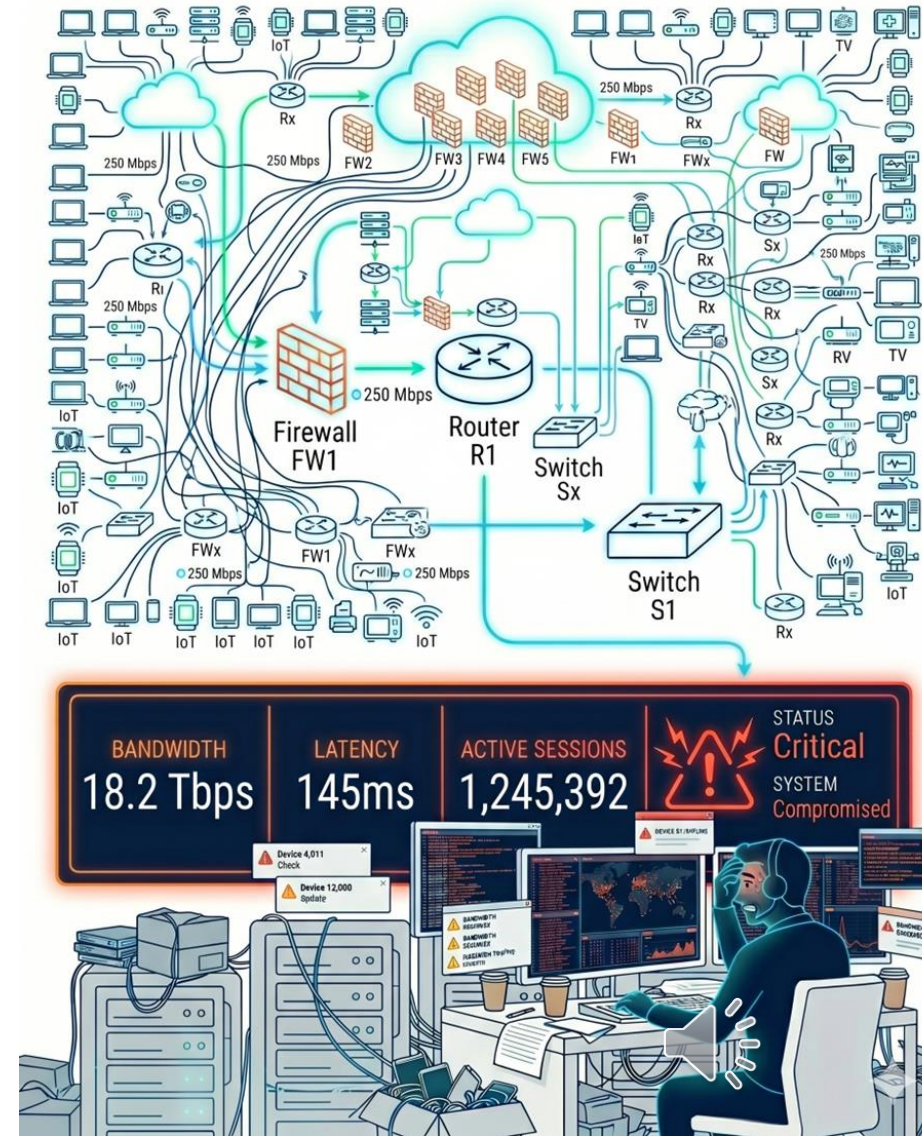
Clustering – Motivation

- ▶ Would you treat every device on the network the same?
 - ▶ If the CEO's laptop and the smart-fridge send a 1GB packet to the company's database, would you treat these instances as the same?
- ▶ Different devices can be grouped based on their network behavior.
- ▶ If a device suddenly behaved differently, it would be a cause for concern.



Clustering – Motivation

- ▶ Suppose you work for the University of Arizona, where **thousands** of devices are connecting and disconnecting each day.
 - ▶ Do you want to evaluate and monitor every device?
- ▶ You may greatly appreciate an AI-enabled analytic that can automatically identify clusters of devices based on their network behavior.



Online M.S. in AI

Clustering

- ▶ Clustering algorithm groups datapoints into clusters such that datapoints within the same cluster are similar to each other.
- ▶ For example, we might find several clusters of devices within the UofA that exhibit similar network behavior:
 - ▶ **Student Personal Devices:** High volume of encrypted web traffic, frequent roaming between different Wi-Fi access points.
 - ▶ **Private Servers/Databases:** Consistent high-bandwidth transfers, communication restricted to admin IPs, regular midnight activities related to data backup.
 - ▶ **Scientific Equipment:** Static IP with low-frequency communication, high-volume bursts of communication to a single destination, use of specialized protocols.



Clustering – K-means (*Overview*)

- ▶ K-means algorithm is a simple ML model that partitions a given dataset into a user-specified number of clusters K :
 - ▶ **Input:** Dataset of datapoints, each with values for the same v features (a v -dimensional input).
 - ▶ **Parameters:** The current v -dimensional value (location) of each centroid.
 - ▶ **Initialization:** We randomly select K datapoints as the centroid for each cluster.
 - ▶ **Output (Data Assignment):** Each datapoint is assigned to its closest centroid, forming the centroid's cluster.



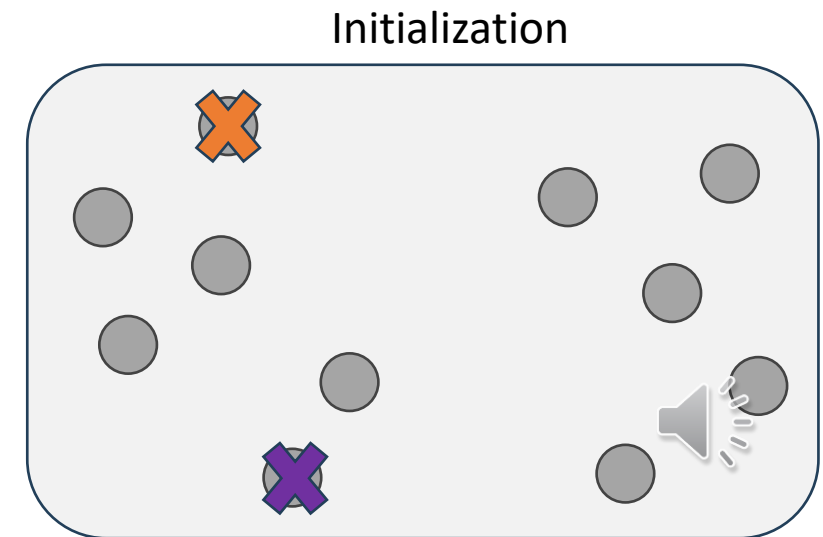
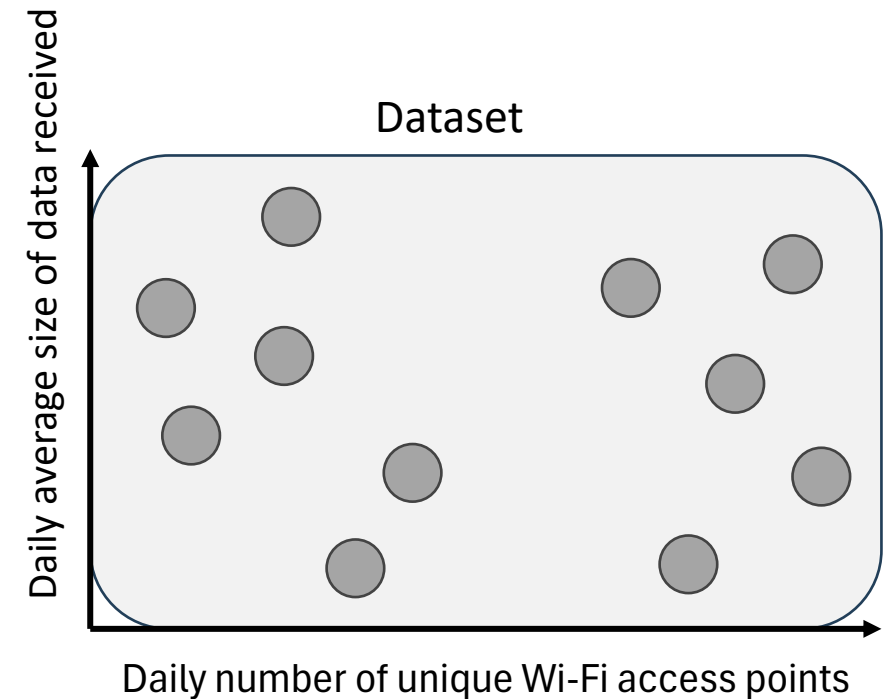
Clustering – K-means (*Overview*)

- ▶ K-means algorithm is a simple ML model that partitions a given dataset into a user-specified number of clusters K :
 - ▶ **Loss Function:** For each cluster, calculate the vector between the centroid and the center (mean) of all its datapoints.
 - ▶ **Optimization (Relocation of Means):** Move each centroid according to the loss vector.
 - ▶ **Converge:** Stop when each cluster's centroid stays at its center.
- ▶ **Core Intuition:** When datapoints with v features are plotted on a v -dimensional space, related or similar datapoints are located closer to each other.



Clustering – K-means

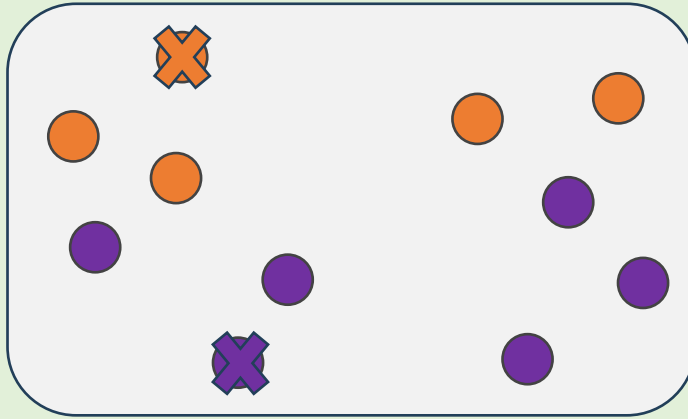
- ▶ Consider this example with 11 devices as datapoints, each with two features:
 - ▶ X Axis: Daily number of unique Wi-Fi access points
 - ▶ Y Axis: Daily average size of data received
 - ▶ Our datapoints can be plotted in this 2-dimensional space
- ▶ Let's use K-means to cluster this dataset.
 - ▶ Initialization: Set $K = 2$ and select 2 datapoints as initial centroids



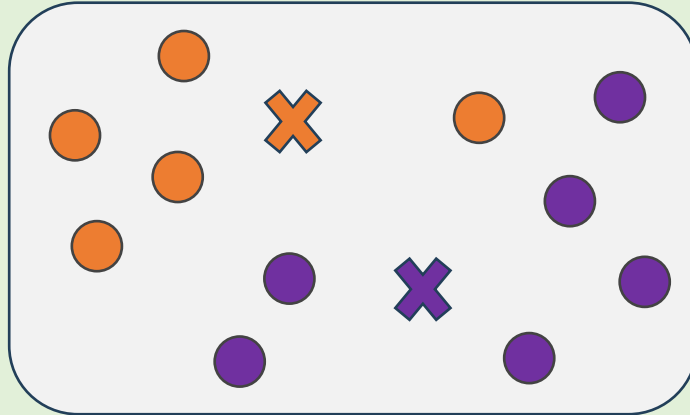
Data Assignment

Relocation of means

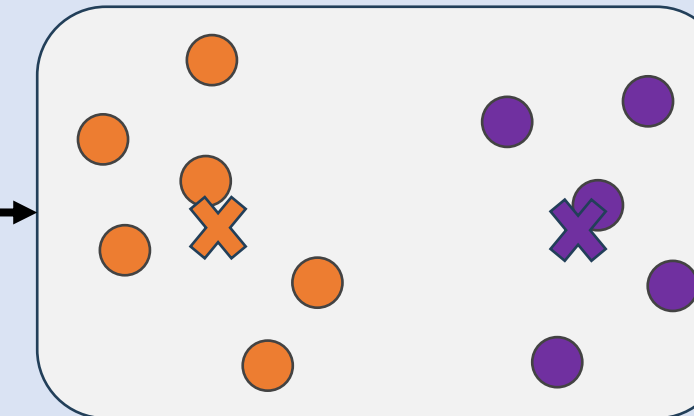
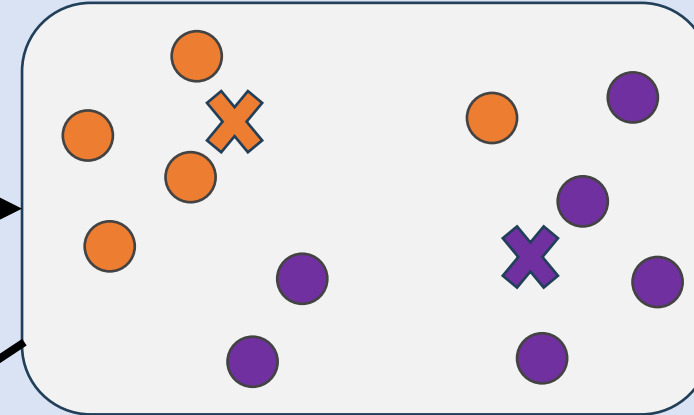
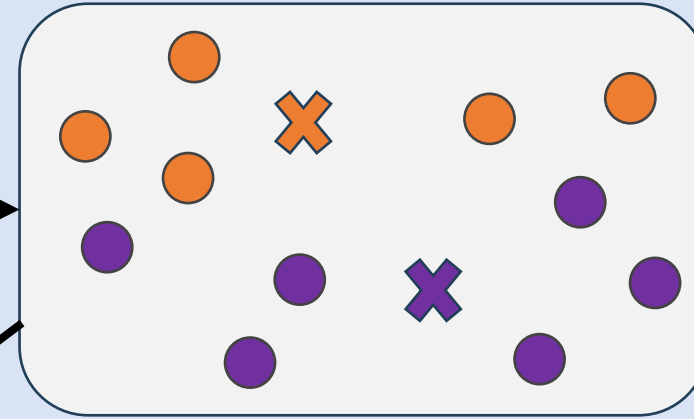
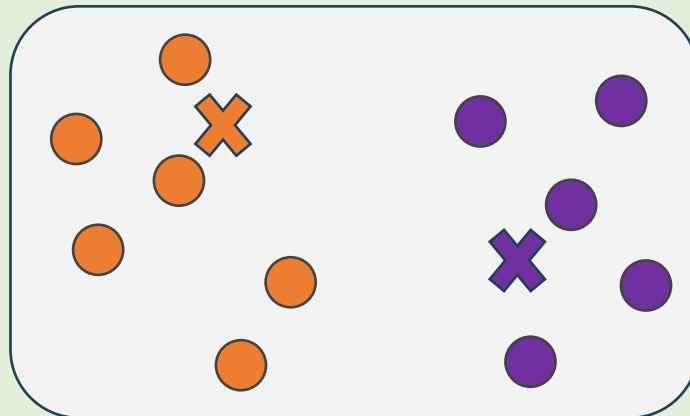
Step 1



Step 2



Step 3

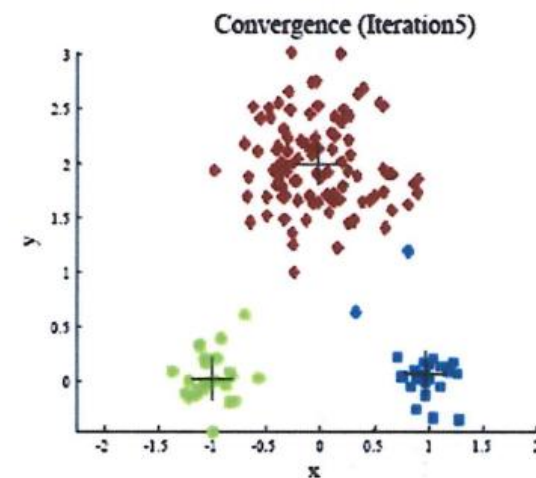
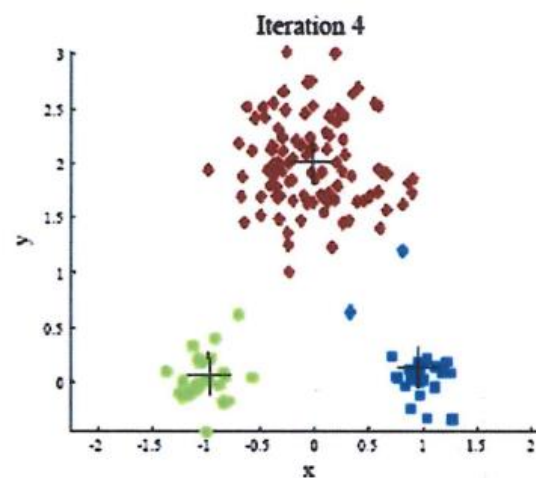
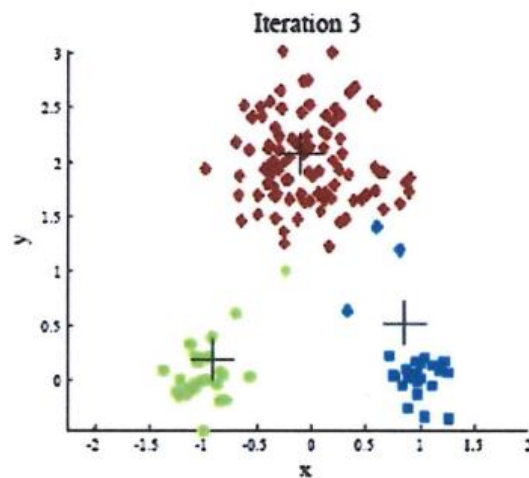
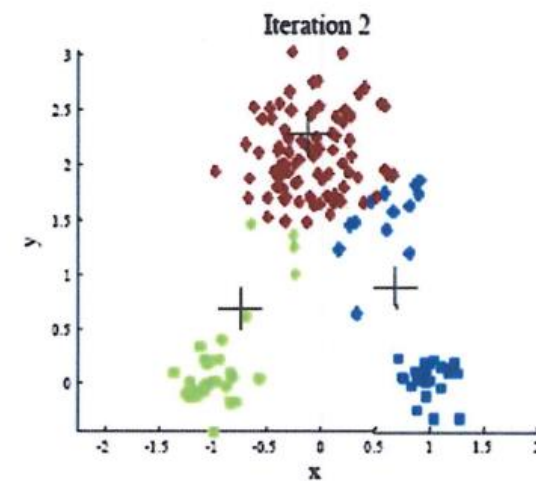
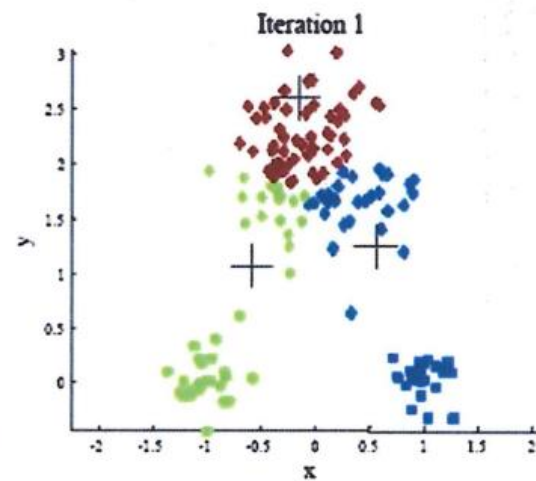
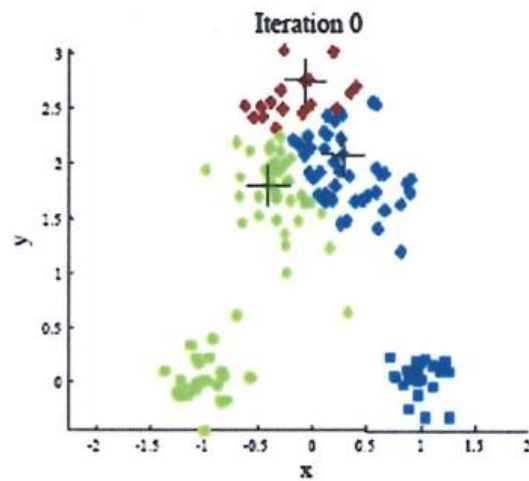


Possible Interpretation:

- Purple denotes Student Personal Devices
- Orange denotes Scientific Equipment

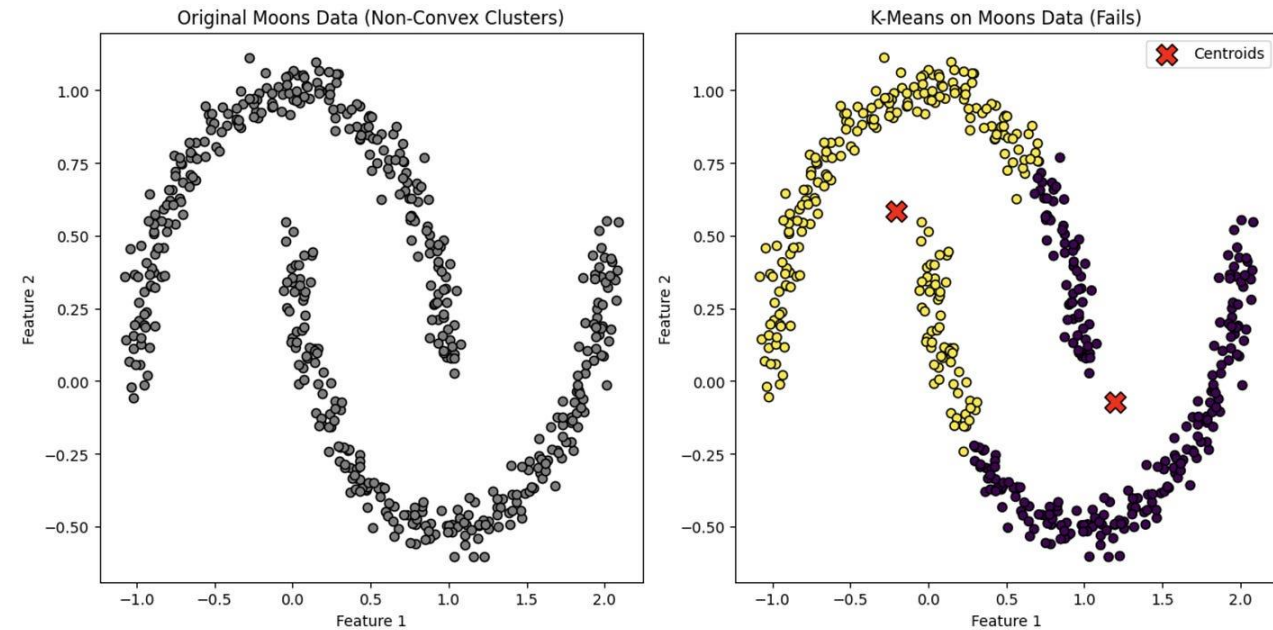


Clustering – K-means

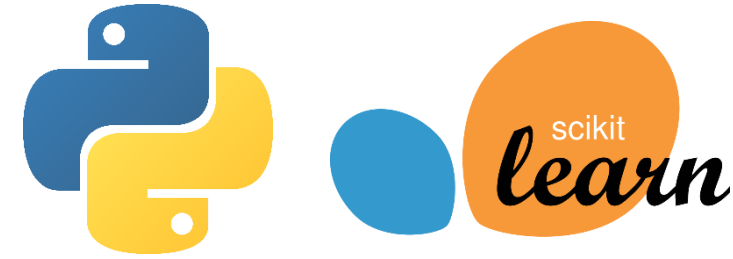


Clustering – K-means

- ▶ K-means has some limitations:
 - ▶ The number of cluster k has to be decided in advance
 - ▶ Clustering result depend on the initial partition
 - ▶ No guarantee on the best solution (global optimum)
 - ▶ It doesn't work well when clusters are not compact
- ▶ More advanced clustering models were developed to address these limitations (e.g., DBScan).
 - ▶ Outside the scope of this class, but feel free to explore and use these models for your project.



Clustering – Python Example



- ▶ **Python example:** network device segmentation
 - ▶ Refer to supplemental materials (*clustering.ipynb*)
- ▶ **Data:** network device information (e.g., ip address, connection time, etc.)
- ▶ **Key Packages:** sklearn.cluster



Topic Modeling – Motivation

- ▶ Textual data is one of the more common data types in cybersecurity.
 - ▶ Can we find clusters in textual data?
- ▶ Suppose you are working as a cyber analyst monitoring hacker forums. There are thousands of forum posts per day.
 - ▶ How do you know what trends they are discussing?
 - ▶ Is there a pattern that may hint towards a planned future cyberattack?

Looking to Zero-Day Vulnerability and Exploitation Methods

Zero_Day_Scout

We have analyzed several recent zero-day vulnerability reports and discussed exploitation methods for an RCE. Who has proof-of-concepts? Share your code and methods below.

[Read More](#) · November 30, 2025 · [Read More](#)



Thinking about a new Ransomware-as-a-Service (RaaS) Trends Campaign



Hey, looking to move from a declining RaaS operation. Which of the new ransomware-as-a-service providers do you recommend for higher affiliate payouts and reliable decryption for a new campaign?

[Read More](#) · November 30, 2025 · [Read More](#)

For Sale: Recent Data Breach PII Dataset

DB_Trader_X

WTS: Fresh Data Breach of customer PII. Large dataset from recent e-commerce hack. Contains emails, passwords, and addresses. Asking 10 BTC for full DB or 1 BTC per 10k rows. Sample available. Serious offers only.

[Read More](#) · November 30, 2025 · [Read More](#)



Advanced Persistent Threat (APT) Group TTP Shifts Discussion

Have you observed recent shifts in Advanced Persistent Threat group tactics, techniques, and procedures? We are tracking a group that appears to be changing its initial access and lateral movement patterns. Share observations on code snippets and tool discussion.

[Read More](#) · November 30, 2025 · [Read More](#)



Topic Modeling

- ▶ Topic models are a suite of algorithms for discovering the main themes (i.e., textual clusters) from a collection of documents.
- ▶ Topic models can be applied to massive collections of documents to automatically organize, understand, search, and summarize large electronic archives.
- ▶ Latent Dirichlet Allocation (LDA), a technique based on Bayesian Modeling, is more commonly used nowadays.
 - ▶ Useful shorthand: “Bayesian” typically means something pertaining to conditional probabilities:

$P(A|B, C)$ denotes how likely event A will happen if events B and C already happened

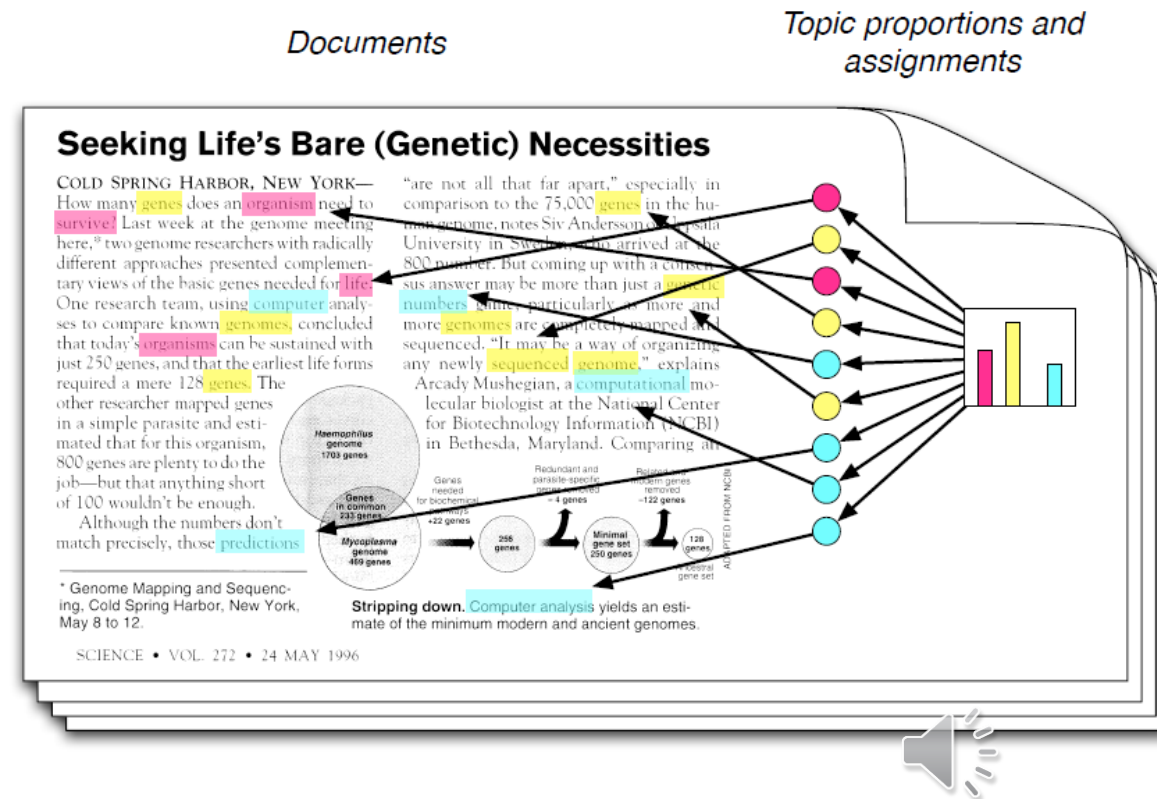


Topic Modeling – LDA

- ▶ LDA uses several definitions:

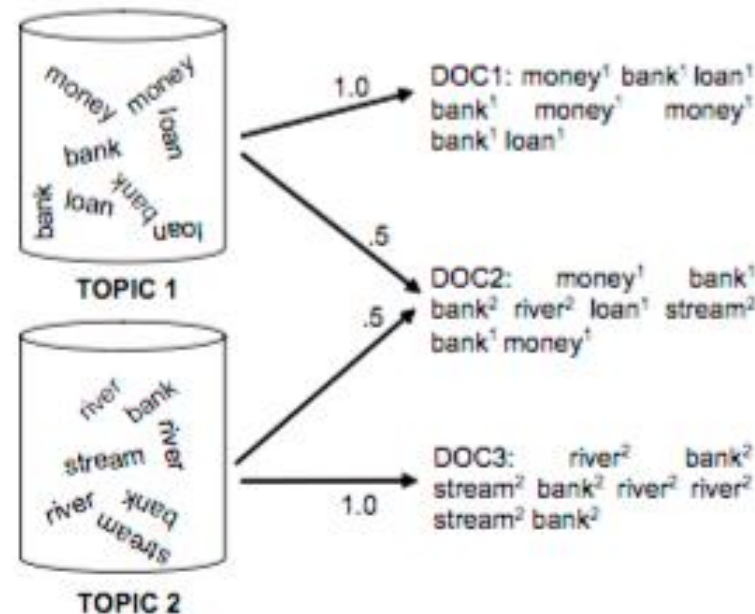
- ▶ Each **word** is a basic unit of discrete data.
- ▶ Each **topic** is a probability distribution of many words.
- ▶ Each **document** is assigned a probability distribution of topics.

Topics	
gene	0.04
dna	0.02
genetic	0.01
...	
life	0.02
evolve	0.01
organism	0.01
...	
brain	0.04
neuron	0.02
nerve	0.01
...	
data	0.02
number	0.02
computer	0.01
...	



Topic Modeling – LDA

- ▶ LDA simplifies textual data by viewing each document as a bag-of-words.
- ▶ From an LDA's point-of-view, documents are written by:
 1. Probabilistically sampling a topic its topic distribution.
 2. Probabilistically sampling a word from the chosen topic.
 3. Repeated for each word in the document



How LDA Perceives the Documents

Tokenized as:

[money, loan, bank, river, stream]

[3, 2, 3, 0, 0]

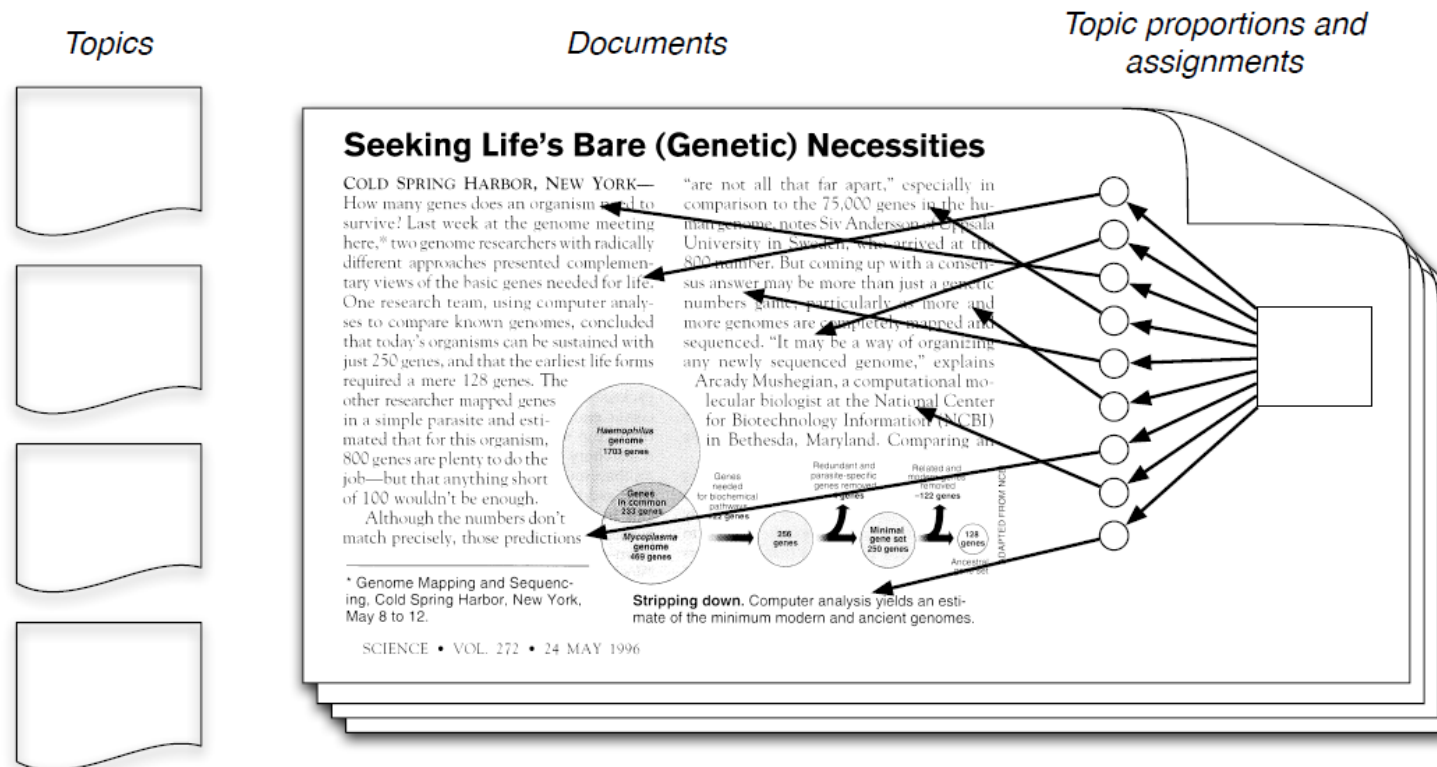
[2, 1, 3, 1, 2]

[0, 0, 3, 3, 2]



Topic Modeling – LDA

- ▶ However, an untrained LDA model does not yet know the topics.
 - ▶ By observing documents and their bag-of-words, an LDA model must **infer** the hidden topics.



Topic Modeling – LDA (*Overview*)

- ▶ The details of LDA involves complex Bayesian math that are outside the scope of this class. We will keep the overview of LDA at a high-level:
 - ▶ **Input:** Documents as a bag-of-words (the frequency of each word in a document)
 - ▶ Input is typically an $M \times N$ matrix, where M is the total number of unique words across all N documents in your training dataset.
 - ▶ **Parameters:** With K total topics, an LDA model contains two sets of parameters:
 - ▶ **Document Context:** Probability that sampling from a topic results in that word for each of the M unique words and K topics ($P(word|topic)$).
 - ▶ **Topic Context:** Probability that a topic is assigned to a document for each of the K topics and N documents ($P(topic|document)$).
 - ▶ **Output:** An LDA model uses its two contexts to probabilistically assigns:
 1. Each word to a topic based on the word's $P(topic|word, document)$
 2. Each document to a topic based on the document's $P(topic|document)$

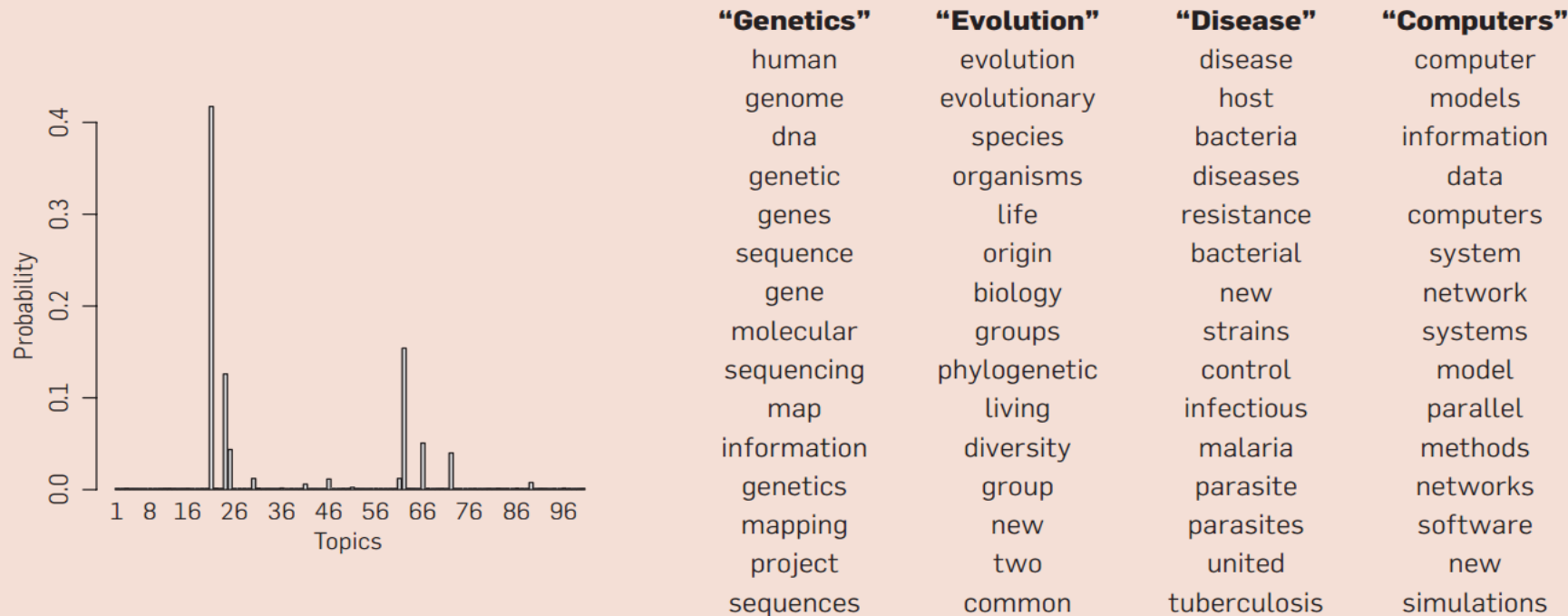


Topic Modeling – LDA (Overview)

- ▶ The details of LDA involves complex Bayesian math that are outside the scope of this class. We will keep the overview of LDA at a high-level:
 - ▶ **Loss Function:** Calculate how unlikely it is for the LDA current probabilities to output the new topic assignments (i.e., perplexity).
 - ▶ **Optimization:** Adjust the LDA's probabilities to lower perplexity by making the new topic assignments more likely.
 - ▶ **Converge:** When topic assignments stop changing.
- ▶ **Core Intuition:** Related words often appear together. For example:
 - ▶ “Phishing” and “Email” may appear more often in the same forum post (document).
 - ▶ That forum post is assigned the topic “SE” for social engineering
 - ▶ In the next training step, “Phishing” and “Email” are more likely to be assigned “SE” due to the forum post’s increased $P(SE|post)$ affecting each word’s $P(SE|word, post)$.

Topic Modeling – LDA

The resulting output from an LDA model would be sets of topics, each containing words, which would then be manually labeled. On the left are the inferred topic proportions for the example article from the previous figure.



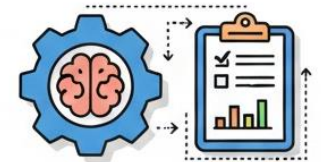
Unsupervised Learning

- ▶ K-means and LDA are both descriptive analytics that conduct unsupervised learning.
 - ▶ **Unsupervised Learning:** Extracts any available patterns from a dataset.
- ▶ Unsupervised learning models are **good at discovering the existence of unknown patterns.**
 - ▶ However, they do not interpret what the pattern is.
- ▶ What if you already know the pattern you want your AI model to learn?

1. Data



2. Training



3. Insights

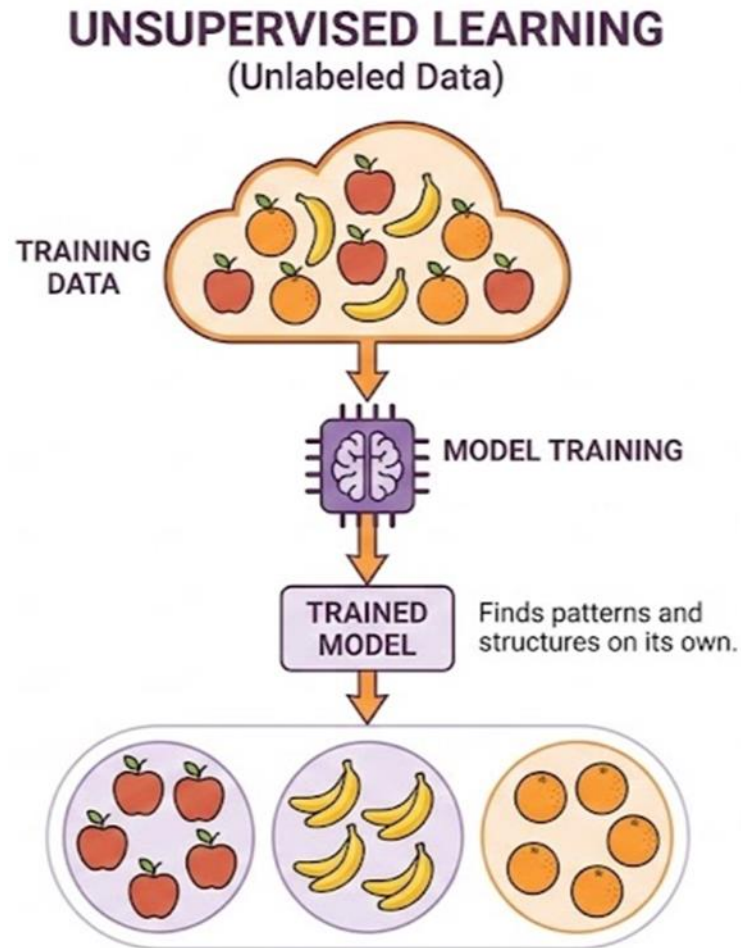


Unsupervised vs Supervised Learning

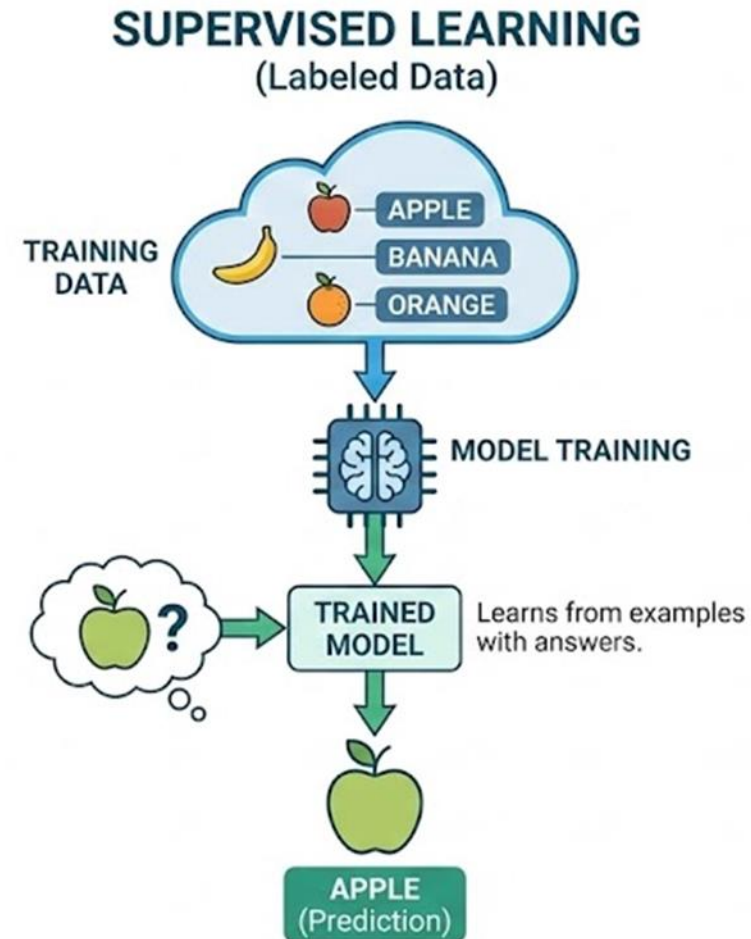
- ▶ Consider developing an AI-enabled phishing detector.
 - ▶ **Target Pattern:** Relationship between an email's (datapoint's) features and if it is phishing or benign (categorical label).
- ▶ We can attempt to use K-means with $K = 2$, but it is difficult to ensure that K-means' two clusters will correspond to our two labels.
 - ▶ The two clusters could correspond to something as trivial as “big and small emails.” We just don't know until the K-means model finishes learning!
- ▶ For tasks with labeled data, the label can “supervise” the AI model's learning so that it learns the correct pattern.



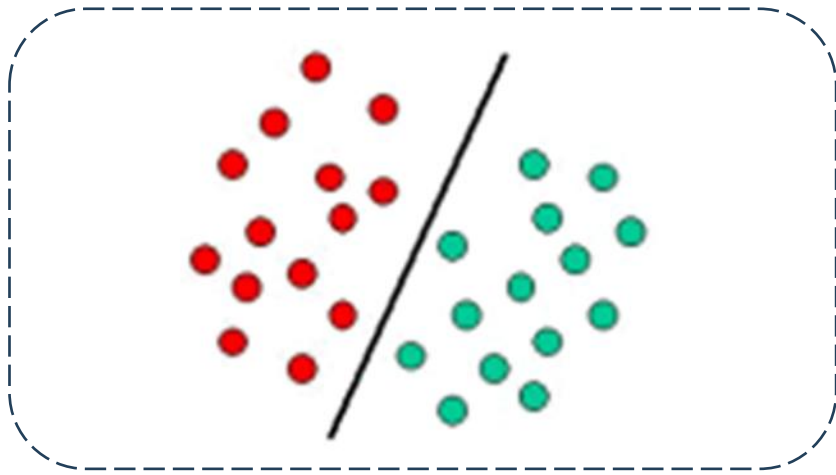
Unsupervised vs Supervised Learning



VS



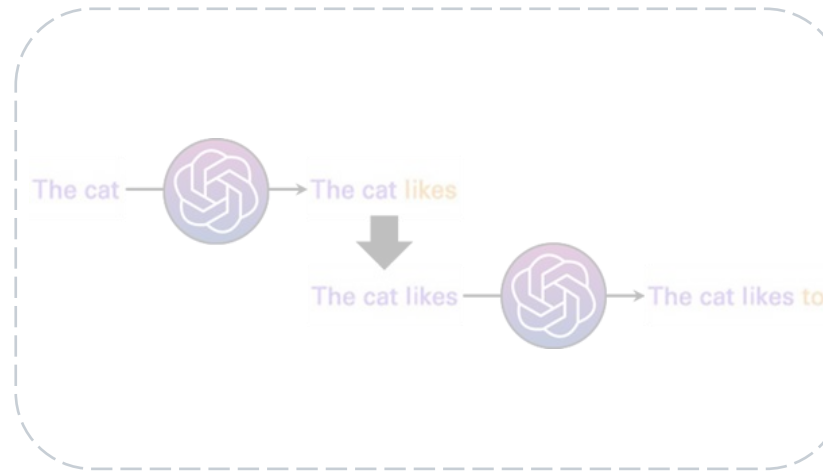
From Week 1: Predictive Methods



Classification

Examples

- **Phishing Detection:** Flagging emails that have a high probability of being phishing
- **Malware Detection:** Determine which files are suspicious or malicious



Generation

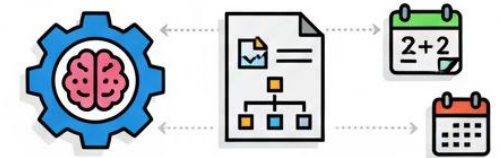
Examples

- **After Action Summarization:** Generate reports that summarize key points of a cybersecurity incident.

1. Data



2. Training & Testing



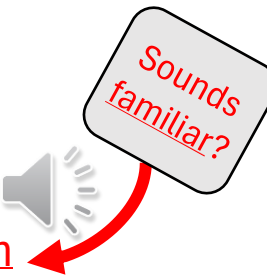
3. Prediction & Forecasting



Online M.S. in AI

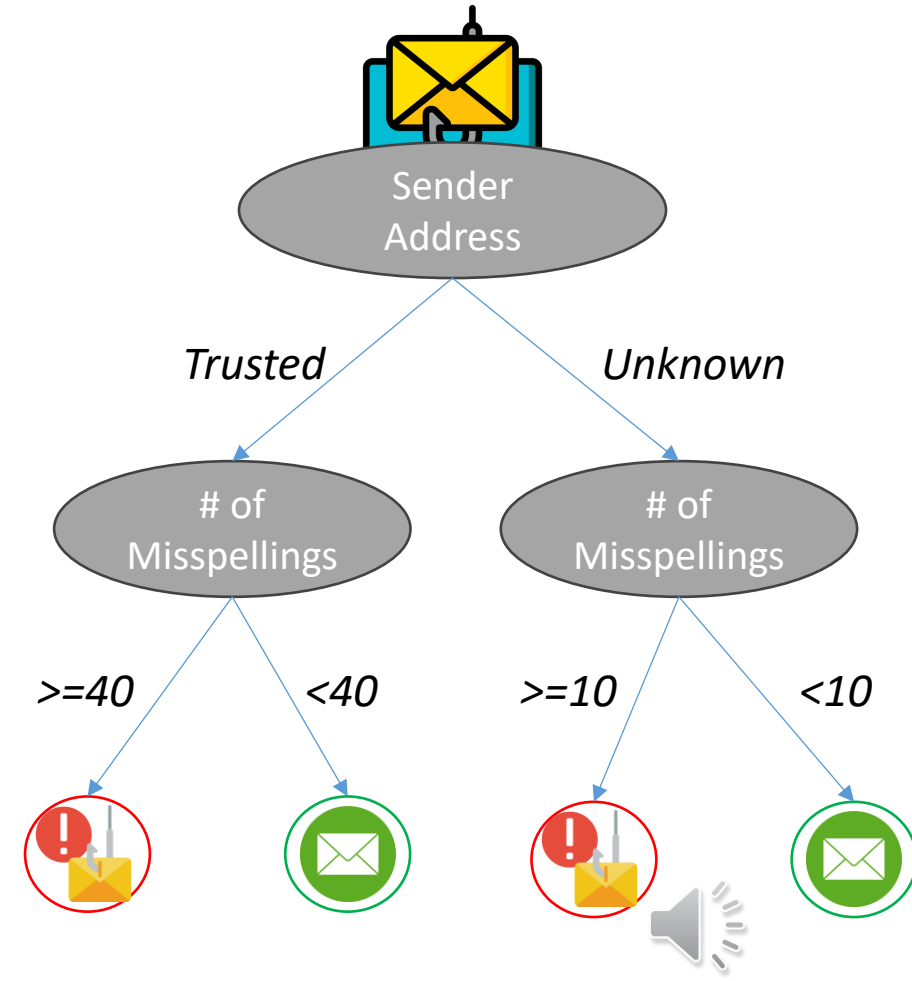
Classification – Motivation

- ▶ Suppose you are a firm's email security administrator, and you are asked to evaluate a new email and classify it as phishing or benign. You can observe:
 - ▶ Sender/recipient email addresses
 - ▶ Email text
 - ▶ Attachments
 - ▶ Datetime sent
- ▶ How would your approach this task?
 - ▶ Would you carefully examine each individual word from beginning to end? Or ...
 - ▶ Would you scan for “red flags” that indicate phishing (e.g., suspicious attachments, external sender address, misspellings, etc.)?
 - ▶ Perhaps you learned which “red flags” to look for by observing previous emails and improving from emails you incorrectly classified.



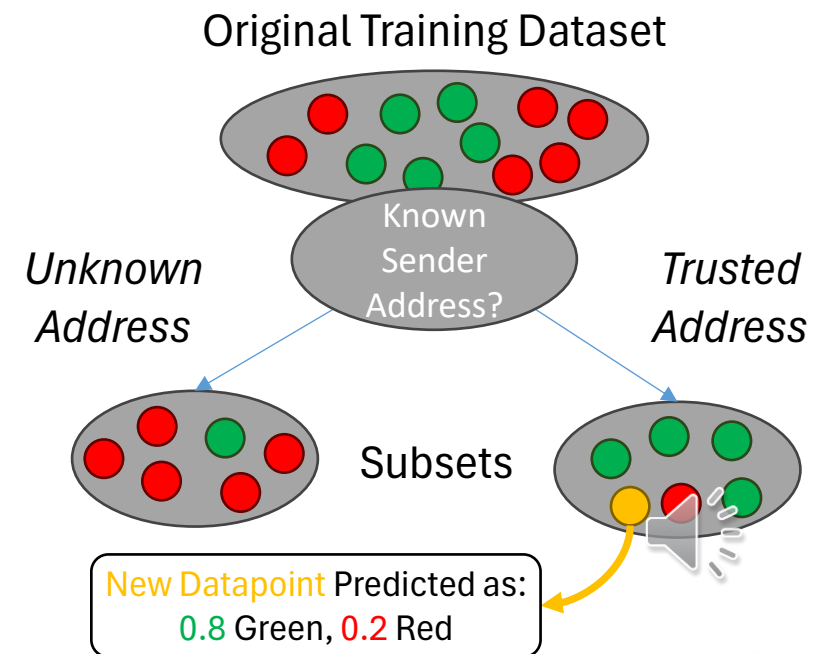
Classification – Analytic Description

- ▶ Classification models learn from a labeled training dataset to predict the label of a datapoint based on its features.
- ▶ A common procedure is the **Decision Tree**, which breaks down a classification problem into a **sequence of tests**
 - ▶ Each internal node = test on an attribute
 - ▶ Each branch = outcome of the test
 - ▶ Each leaf = classification
- ▶ In the previous example, your “red-flags” were the internal nodes of a human decision tree (you).



Classification – Decision Tree (Overview)

- ▶ ID3 (Iterative Dichotomiser 3) is an algorithm that trains a decision tree model from a labeled dataset:
 - ▶ **Input:** Dataset of datapoints, each with categorical values for the same v features (a v -dimensional input) and a categorical label.
 - ▶ **Parameters:** Current nodes, branches, and leaves in the tree
 - ▶ **Initialization:** The tree starts empty
 - ▶ **Output:** Predicted label is the probability makeup of the subset based on tree traversal.



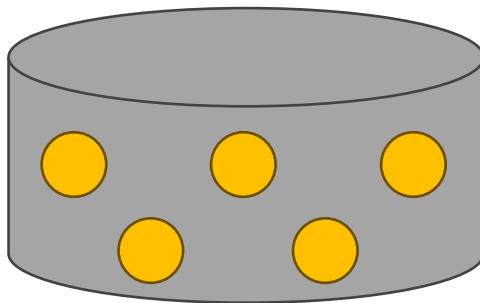
Classification – Decision Tree (*Overview*)

- ▶ ID3 (Iterative Dichotomiser 3) is an algorithm that trains a decision tree model from a labeled dataset:
 - ▶ **Loss Function:** Calculate the entropy for all nodes with features that haven't yet been chosen for nodes (averaged across all of the node's branches).
 - ▶ **Optimization:** For any mixed subset, continue by adding the node and branches with the least entropy.
 - ▶ **Converge:** All subsets are homogenous (all datapoints have the same label) or all features are chosen for nodes.
- ▶ **Core Intuition:** We want subsets where we are 100% certain these datapoints are a certain label (homogenous).
 - ▶ At each step, we choose the test that results in subsets with proportionally the most same-label datapoints.
 - ▶ Entropy is a mathematical representation of this selection criteria.

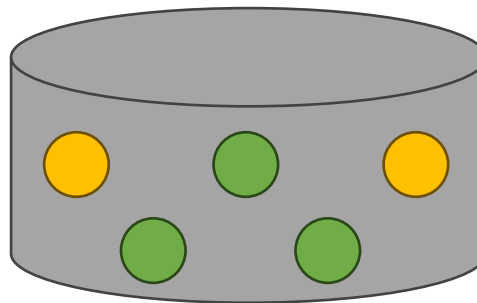


Classification – Decision Tree

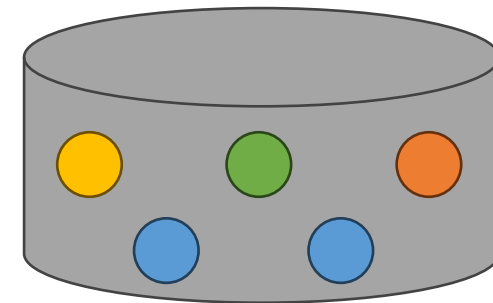
- ▶ **Entropy** is a measure of the amount of uncertainty in the dataset.
- ▶ Consider taking a ball from some jars. How uncertain is your outcome?



100% Yellow



60% Green
40% Yellow

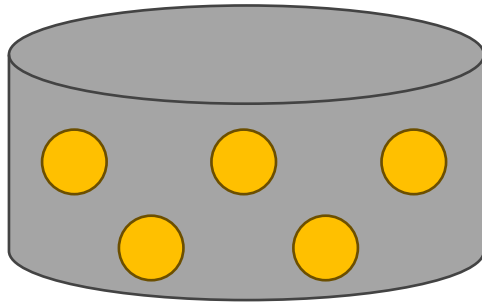


40% Blue
20% Yellow
20% Green
20% Orange

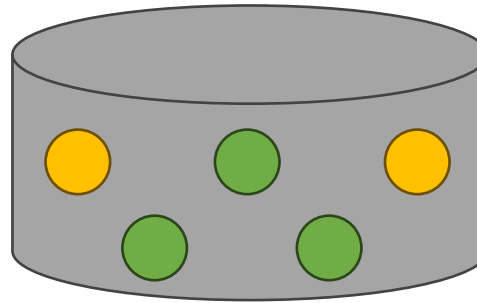


Classification – Decision Tree

- Entropy $H = -\sum_{i=1}^n p_i \log_2(p_i)$



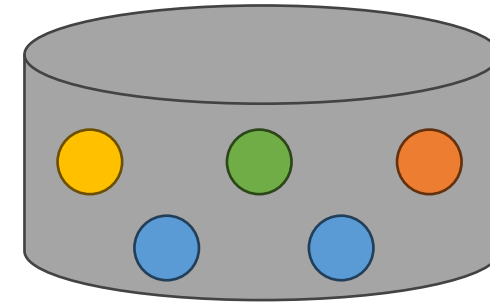
$$P_{yellow} = 1$$
$$H = -(1.0 \cdot \log_2(1.0))$$
$$= 0$$



$$P_{green} = 0.6$$
$$P_{yellow} = 0.4$$

$$H = -(0.6 \log_2(0.6) + 0.4 \log_2(0.4))$$
$$= 0.971$$

$$H$$
$$= -(0.4 \log_2(0.4) + 0.2 \log_2(0.2)$$
$$+ 0.2 \log_2(0.2) + 0.2 \log_2(0.2))$$
$$= 1.922$$

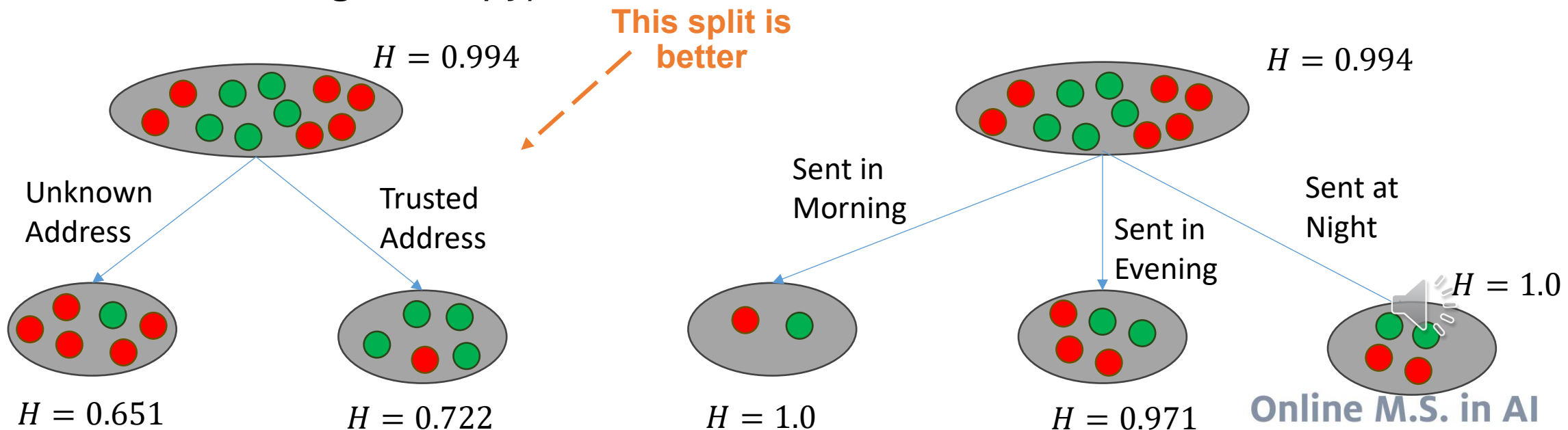


$$P_{blue} = 0.4$$
$$P_{yellow} = 0.2$$
$$P_{green} = 0.2$$
$$P_{orange} = 0.2$$



Classification – Decision Tree

- ▶ When creating branches, ID3 calculates the entropy before dividing.
 - ▶ For each attribute, compute the sum of entropy after dividing.
- ▶ ID3 chooses the criteria that lead to the largest uncertainty reduction (largest decrease in average entropy).



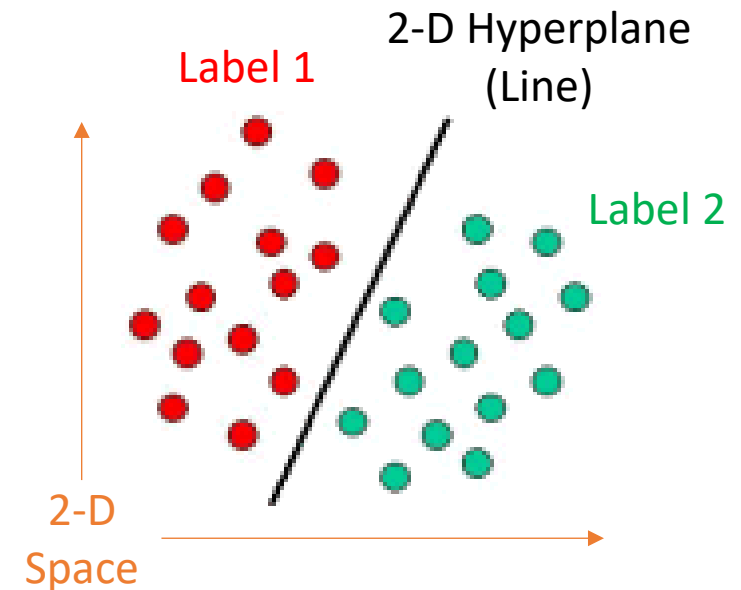
Classification – Decision Tree

- ▶ Other algorithms to train a decision tree model:
 - ▶ **C4.5**: Handles continuous attributes and missing values
 - ▶ **CART**: Uses *Gini index* (classification) or *MSE* (regression)
 - ▶ **CHAID**: Uses *chi-square* significance testing to choose splits
- ▶ Decision trees are useful because they:
 - ▶ Are **interpretable** (rule-like decisions)
 - ▶ Map naturally to human reasoning (if-then rules)
 - ▶ Are computationally efficient
 - ▶ Can represent complex classification boundaries by successive splits



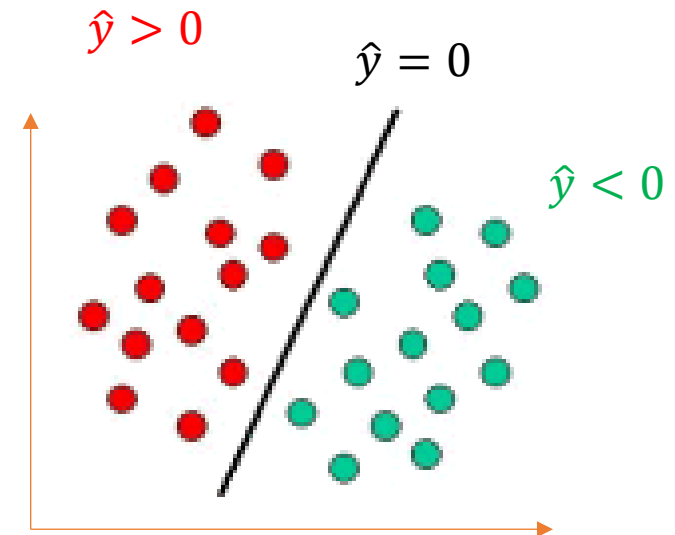
Classification – SVM

- ▶ Support Vector Machines (SVMs) are another class of supervised learning models.
 - ▶ For this class, we will focus on SVMs for classification.
- ▶ SVM is based on the same spatial concept of K-means:
 - ▶ All datapoints with v features reside in a v -dimensional space.
 - ▶ K-means cluster these points together based on proximity.
 - ▶ SVM assumes there are a few v -dimensional hyperplanes that can separate datapoints based on their labels.
 - ▶ These hyperplanes are referred to as potential **Decision Boundaries**.



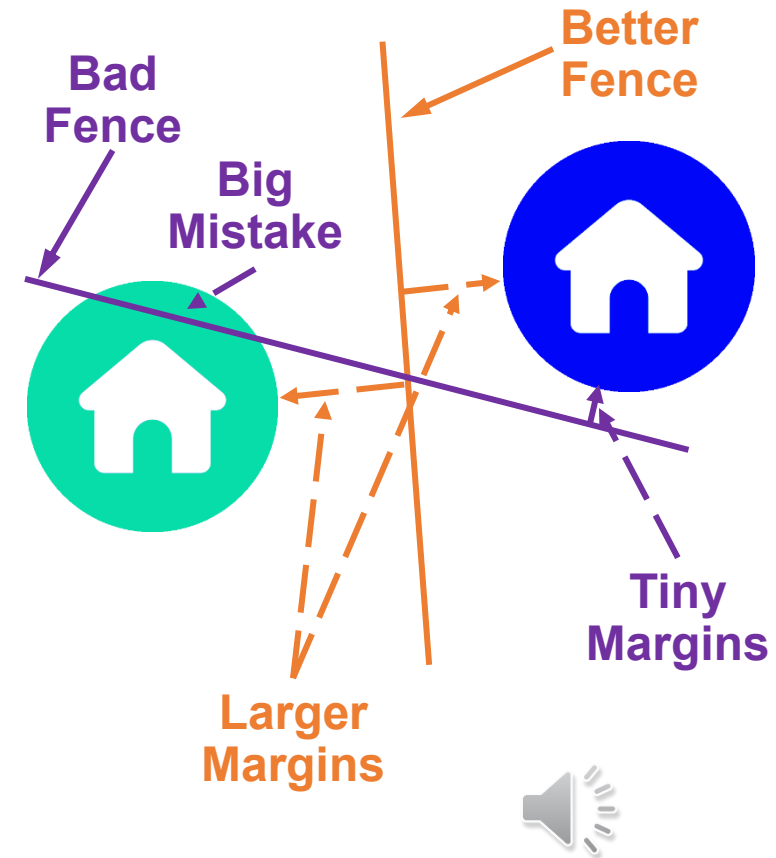
Classification – SVM (Overview)

- ▶ SVM aims to learn the “best” decision boundary:
 - ▶ **Input:** Dataset of datapoints, each with values for the same v features and a categorical label.
 - ▶ **Parameters:** The current parameters of the decision boundary function
 - ▶ For example: The function for a 2-D decision boundary is $\hat{y} = mx_1 + nx_2 + b$, with $[m, n, b]$ as the SVM’s parameters.
 - ▶ **Initialization:** Parameters are randomized
 - ▶ **Output:** Predicted labels is the sign of $\hat{y}$, which represents which “side” of the decision boundary a datapoint is located on.



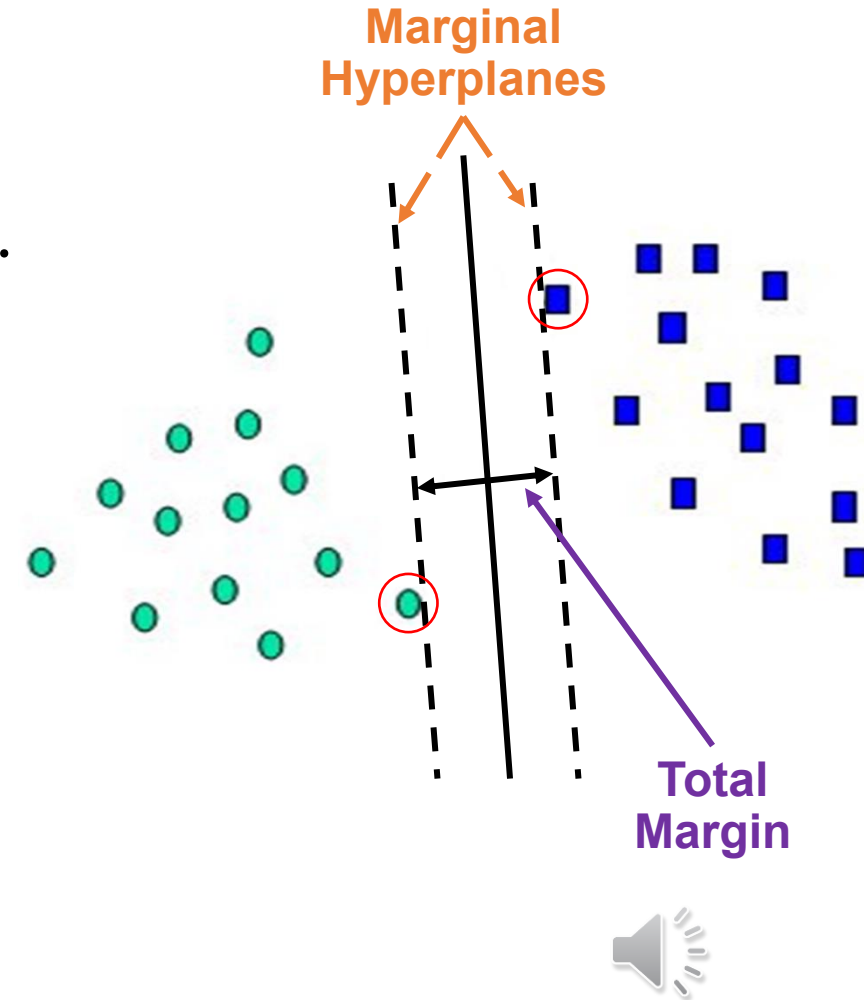
Classification – SVM (Overview)

- ▶ SVM aims to learn the “best” decision boundary:
 - ▶ **Loss Function (The Hinge Loss):**
 - ▶ Big mistake for incorrect predicted labels (datapoints on wrong side)
 - ▶ Small mistake based on current margins
 - ▶ **Optimization:** Adjust the parameters to minimize the loss.
 - ▶ **Converge:** Hinge Loss can no longer be further minimized.
- ▶ **Core Intuition:** Imagine a fence that separates your house from your neighbor’s house.
 - ▶ The fence should at least not cross into a house. More preferable fences should be further from the houses.
 - ▶ You would evaluate based on fence’s nearest distances to each house (margins).



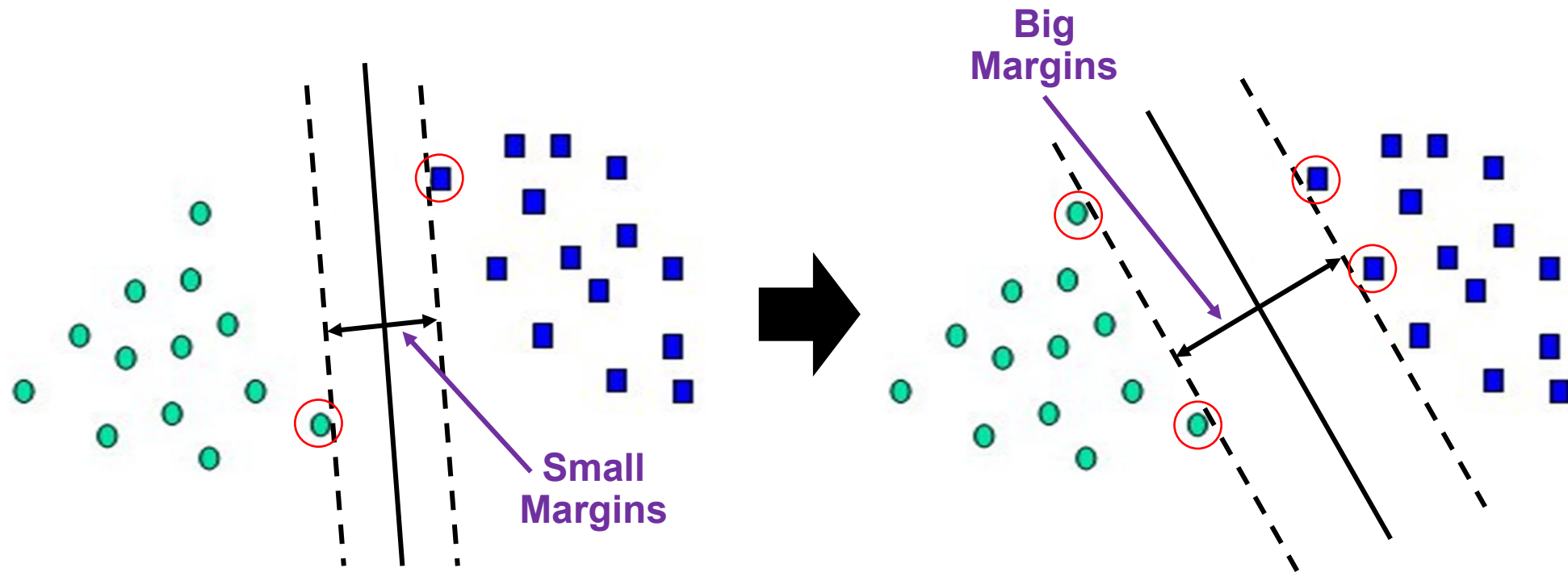
Classification – SVM

- ▶ SVMs calculate the margins of their decision boundary by constructing two marginal hyperplanes.
 - ▶ **Marginal Hyperplanes:** Furthest parallel hyperplanes from the decision boundary before touching a datapoint.
 - ▶ The margins are the distances between the boundary and the marginal hyperplanes.
- ▶ The vectors (datapoints) just touching the marginal hyperplanes are the support vectors.



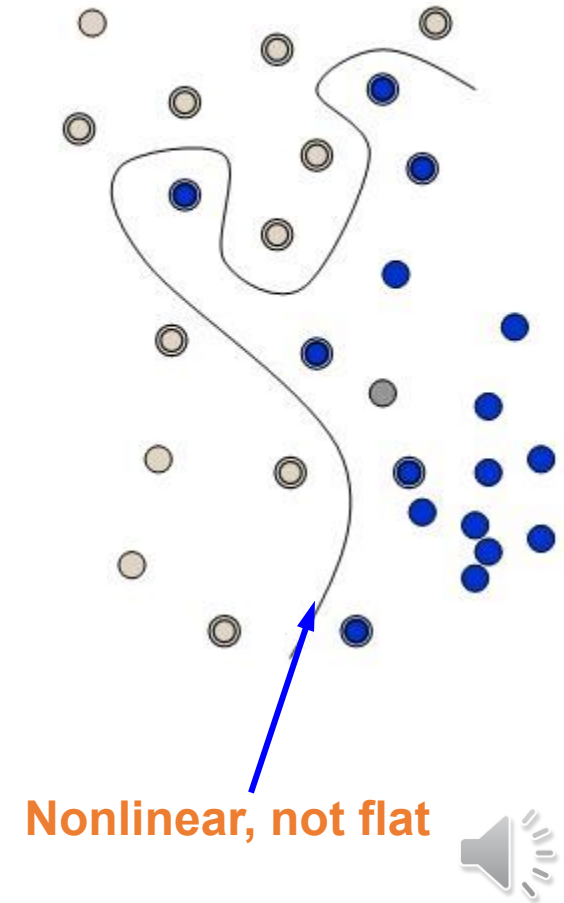
Classification – SVM

- ▶ When an SVM optimizes, it adjusts the orientation of its decision boundary to increase its margins.



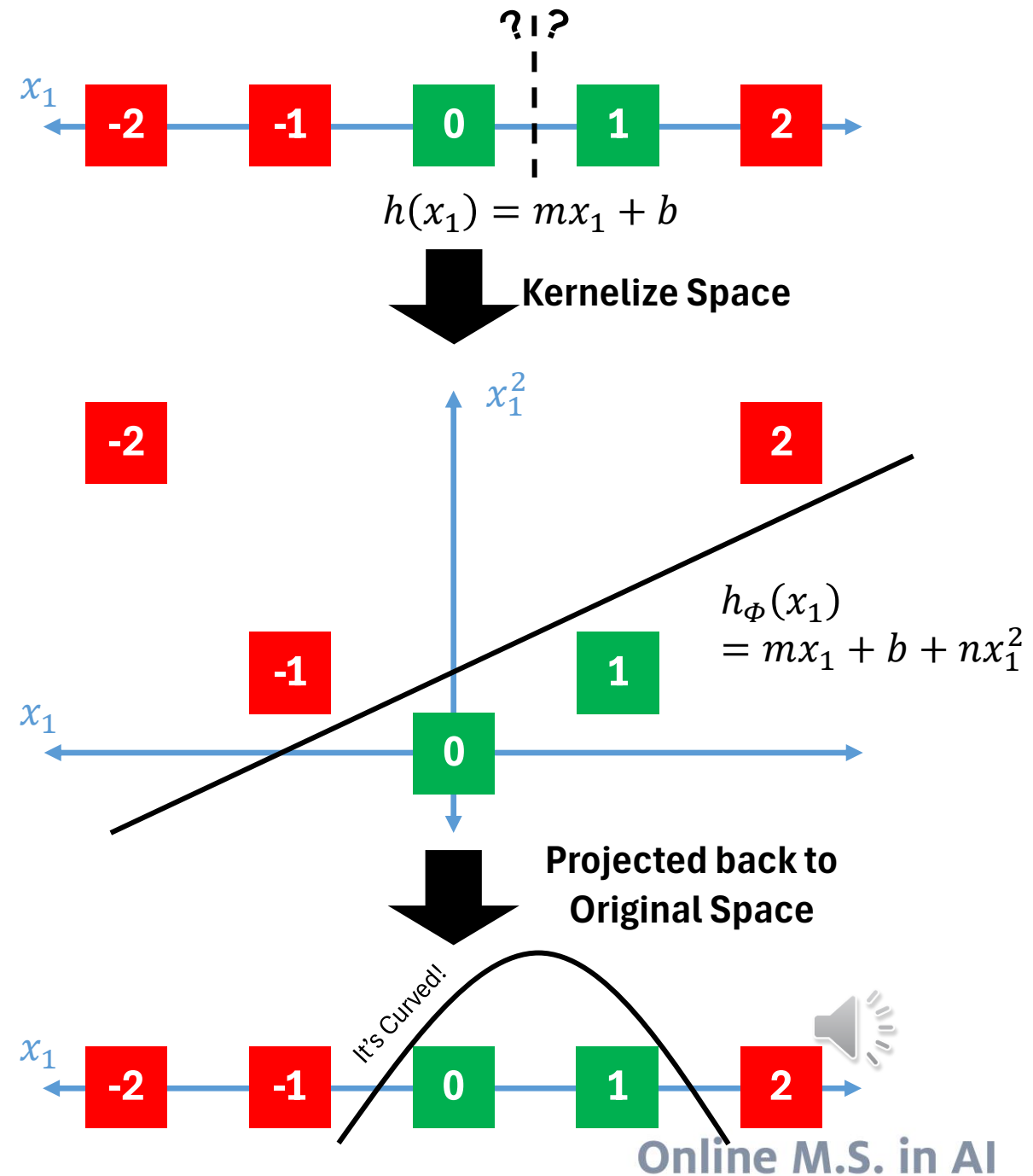
Classification – SVM

- ▶ However, datapoints are not always linearly separable.
- ▶ Can the decision boundary curve?
 - ▶ SVM's marginal hyperplanes must be parallel to the decision boundary.
 - ▶ Staying parallel with a curved high-dimensional hyperplane is mathematically difficult.
 - ▶ Thus, SVMs tend to keep decision boundaries linear.
- ▶ Instead of a curved decision boundary, SVMs change the v -dimensional space into an even higher dimensional space where the datapoints are linearly separable.



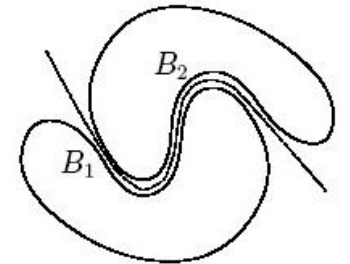
Classification – SVM

- ▶ Consider a 1-D space with the labeled datapoints:
 - ▶ It is impossible to cleanly separate the red/green labels with a single 1-D decision boundary
- ▶ Let's change this into a 2-D space with a new axes of x_1^2 .
 - ▶ We just kernelized the space with a simple **Kernel Function**: $\Phi(x_1) = x_1^2$.
 - ▶ Datapoints are now linearly separable in the kernelized space.
 - ▶ In the original 1-D space, we can see the 2-D boundary.



Classification – SVM

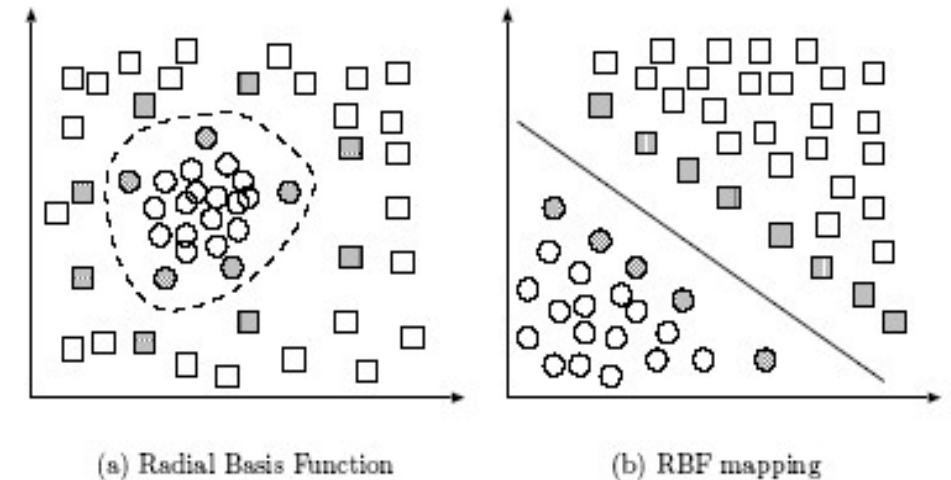
- ▶ Kernel functions increase an SVM's model complexity:
 - ▶ Each variable in a kernel function adds another dimension and trainable parameter.
 - ▶ For example, learning a 3-D boundary in a 2-D space (x_1, x_2) may involve:
 - ▶ **Kernel Function:** $\Phi(x_1, x_2) = x_1^3 + x_2^3 + x_1^2x_2 + x_1x_2^2 + x_1x_2$
 - ▶ **Decision Boundary:** $h(x_1, x_2) = mx_1 + nx_2 + b \rightarrow h_\Phi(x_1, x_2) = mx_1 + nx_2 + b + cx_1^3 + dx_2^3 + ex_1^2x_2 + fx_1x_2^2 + gx_1x_2$
 - ▶ Kernelized SVM now learns a linear 7-D boundary with 8 trainable parameters (266% increase in parameter count).
- ▶ Increased model complexity allows SVMs to learn more complicated patterns!
 - ▶ **Tradeoff:** More parameters means more computation to train.



Classification – SVM

- Common SVM Kernel Functions include:
 - Linear kernel (equivalent to no mapping)
 - Polynomial kernel (Prior examples)
 - Radial basis kernel (In Figure)
 - Convolution kernel
 - Your own kernel function

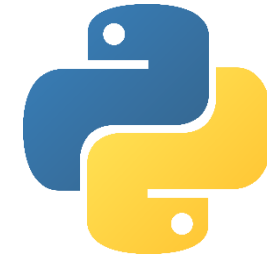
Radial basis function: $\exp(-\gamma|u-v|^2)$



Separable classification with Radial Basis kernel functions in different space. Left: original space. Right: feature space.



Classification – Python Example



- ▶ **Python example:** Phishing detection using Decision Tree and SVM
 - ▶ Refer to supplemental materials (*classification.ipynb*, *phishing.csv*)
- ▶ **Data:** Prior phishing websites with 50+ different features per website
- ▶ **Key Packages:** `sklearn.tree`, `sklearn.svm`





Review Questions

- ▶ Explain the 5 general attributes we are using in this class to describe an AI/ML model.
- ▶ What does K-means or LDA tell us about the clusters/topics they discover?
- ▶ What is entropy and how does ID3 use it to determine which attribute should be the next decision node?
- ▶ What is the intuition behind kernel functions and their role in allowing SVMs to learn more complicated patterns?



Next Week

