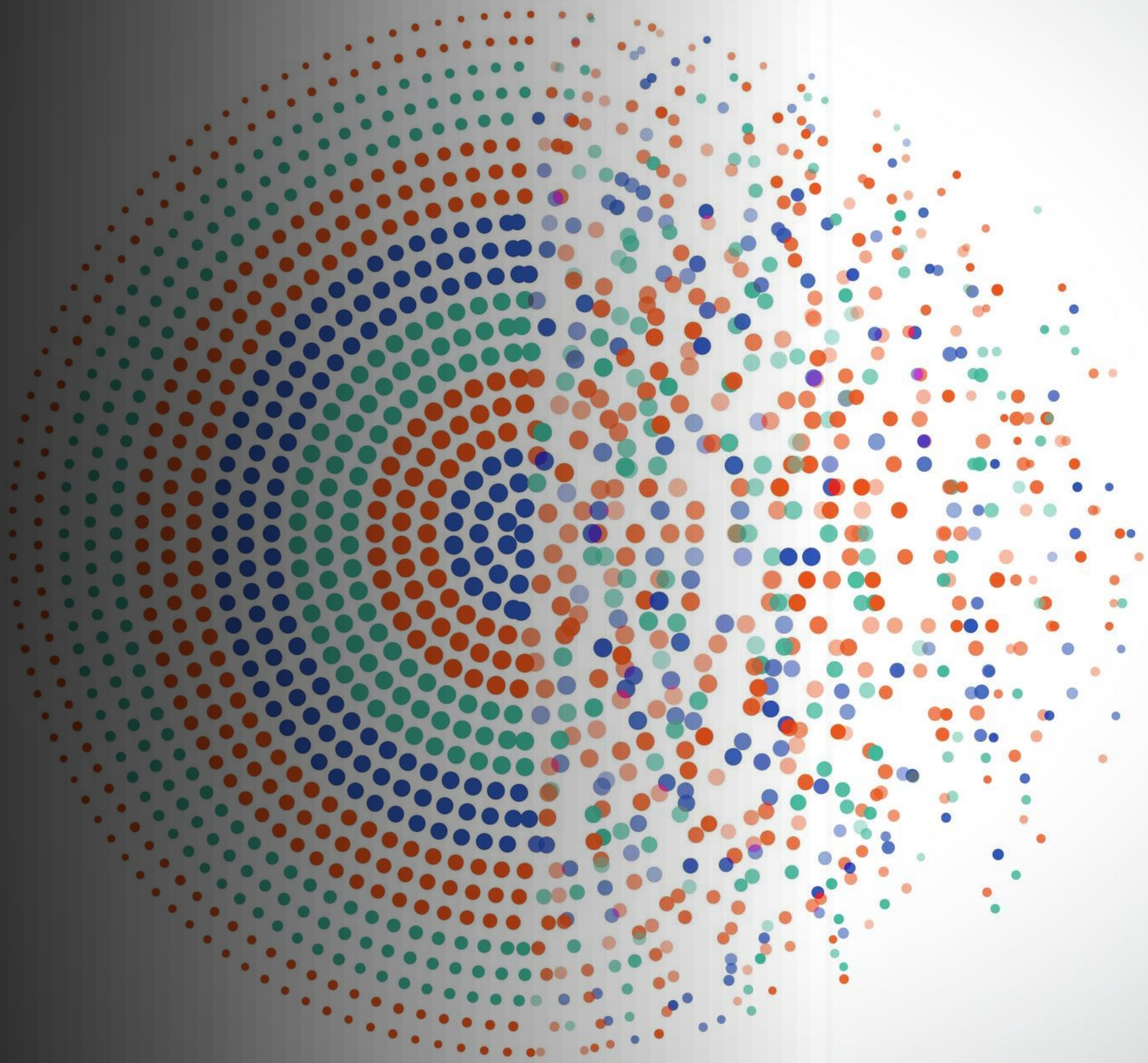
Introduction to Machine Learning

Week 1



Disclaimer and things to remember

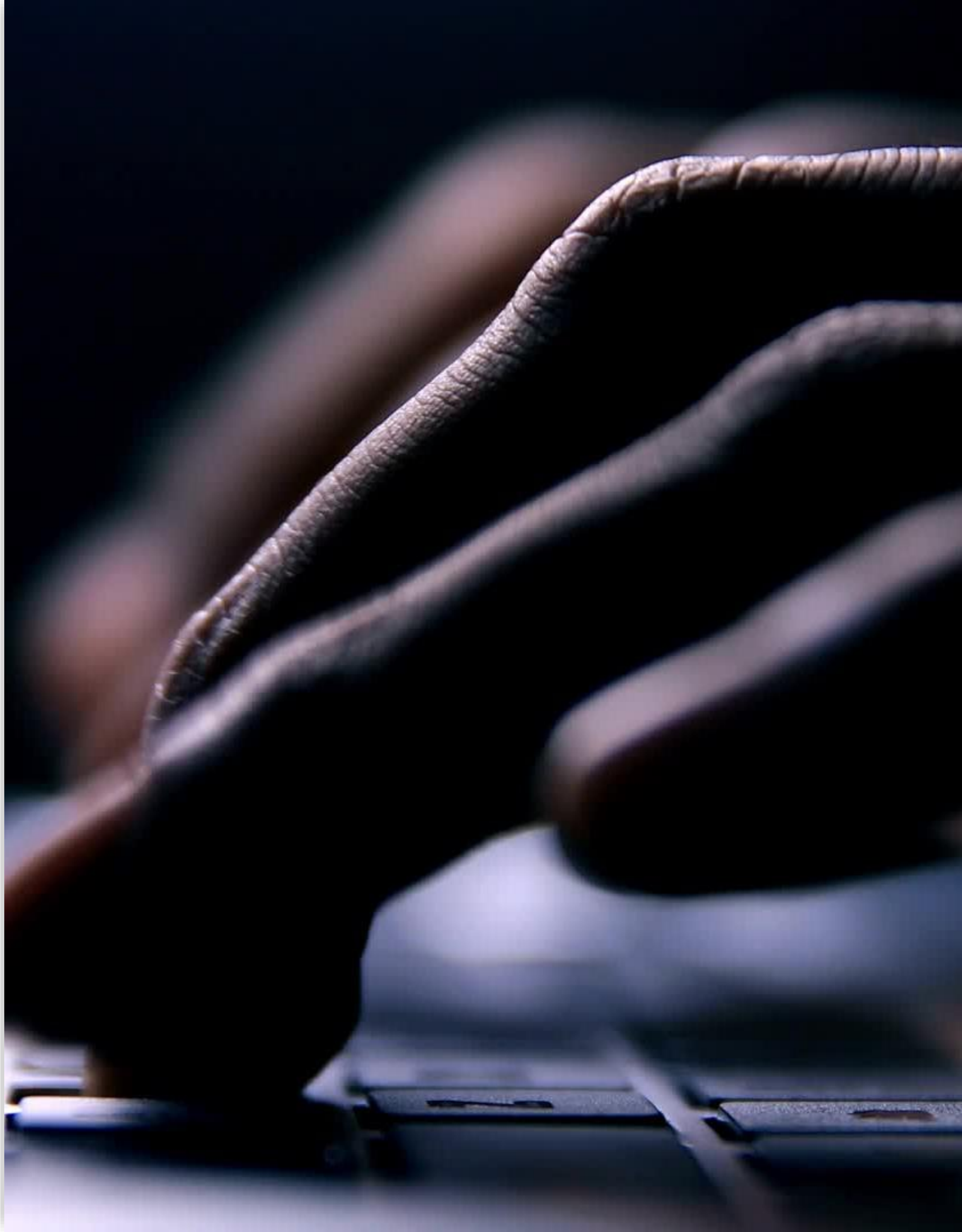
- I do not know everything and cannot explain them all entirely.
- You don't have to remember everything all at once.
- Always focus on the bigger picture.
- This lecture focuses on the overview and more on the surface-level knowledge of this course. No need to get pressured.

Introduction

- In the era of big data, individuals and companies generate vast amounts of data through various activities.
- Both individuals and companies desire personalized products and services based on their needs and interests.
- Predicting customer behavior, like identifying which products a customer is likely to buy, is a complex task due to changing patterns over time and location.
- Traditional algorithmic approaches may not exist for certain tasks, such as predicting spam emails or customer preferences.

Before Machine Learning

- Early intelligent applications used hand-coded rules for decision-making, like creating blacklists for spam filters.
- Hand-coded rules have limitations: they are domain-specific, requiring rewriting for slight task changes, and demand deep human expertise.
- Hand-coding faces in images is challenging due to the difference in computer and human perception.



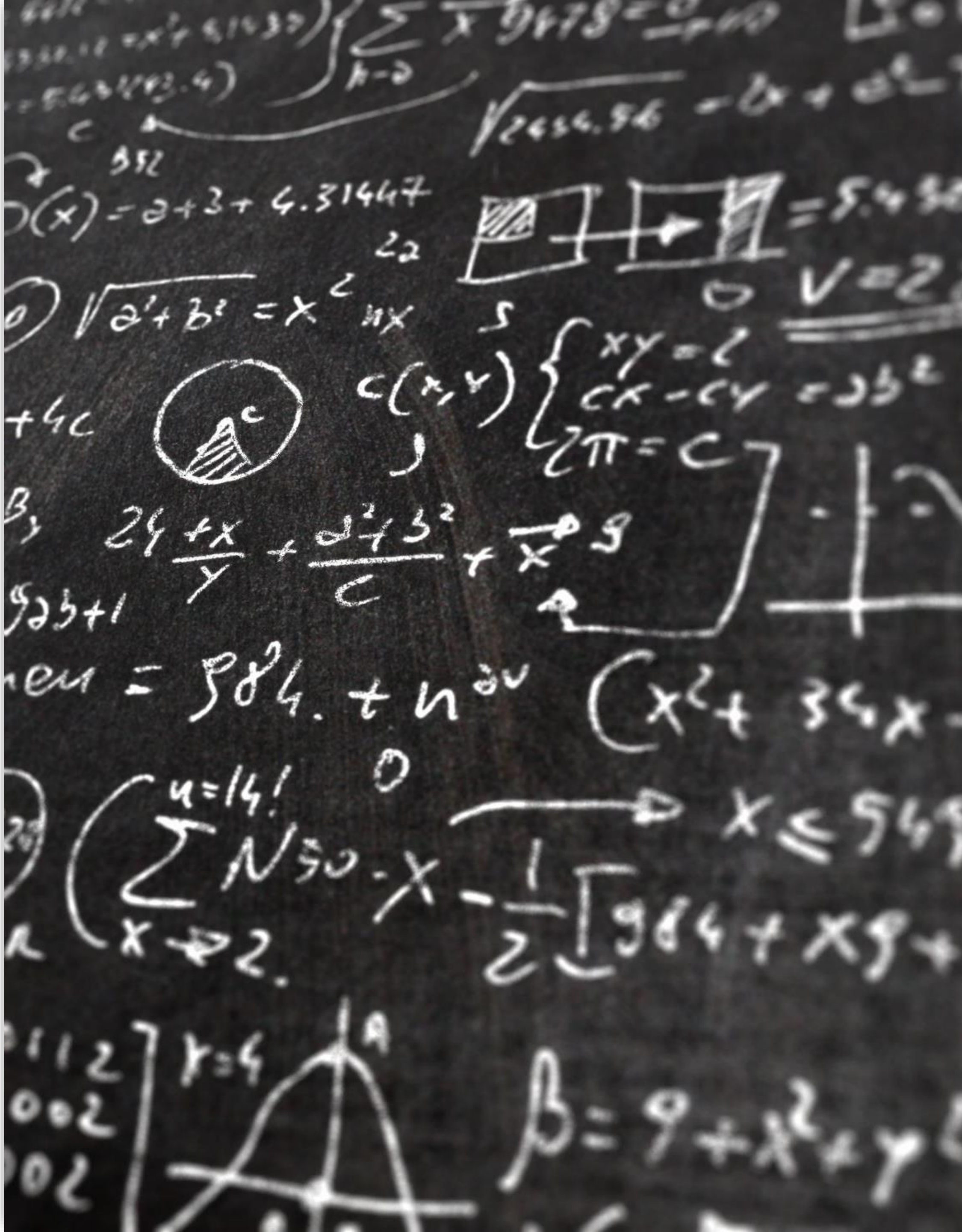
Early application of ML

- ML's earlier applications can be traced from **Samuel's Checkers** playing program.
- The app tends to play on its own, which was considered as **one of the first**, first-learning program (Arthur Samuel, 1959)
- This application somehow **played better** than Samuel.
- The application **learned from thousands of games**, **recognizing patterns** of **how one lose and how one win**.

What is Learning?

- *"The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something."* **- Merriam-Webster dictionary**

- *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."* **- Tom Mitchell**



What is Machine Learning?

Leverages data to automatically extract patterns and create algorithms for tasks where traditional algorithms are unavailable.

Involves constructing approximations to understand or predict patterns in data.

Application to large databases is termed "data mining," used in various fields like retail, finance, manufacturing, medicine, telecommunications, and science.

Enable systems to learn and adapt to changing environments (with limitations).

The efficiency of the learning or inference algorithm is crucial in certain applications, along with predictive accuracy.

Contributes to solutions in vision, speech recognition, and robotics, exemplified by face recognition based on capturing specific patterns.

Involves programming computers to optimize performance using example data or past experiences.

Uses statistical theory to build mathematical models, relying on efficient algorithms for training, data processing, and inference.

Why Machine Learning?

Machine learning overcomes the old approaches by learning from a large collection of examples, enabling algorithms to determine characteristics without explicit rules.

It excels in supervised learning, where users provide input-output pairs for the algorithm to generalize from known examples.

Supervised learning solves various problems, such as spam classification, tumor benignity detection, and credit card fraud detection.

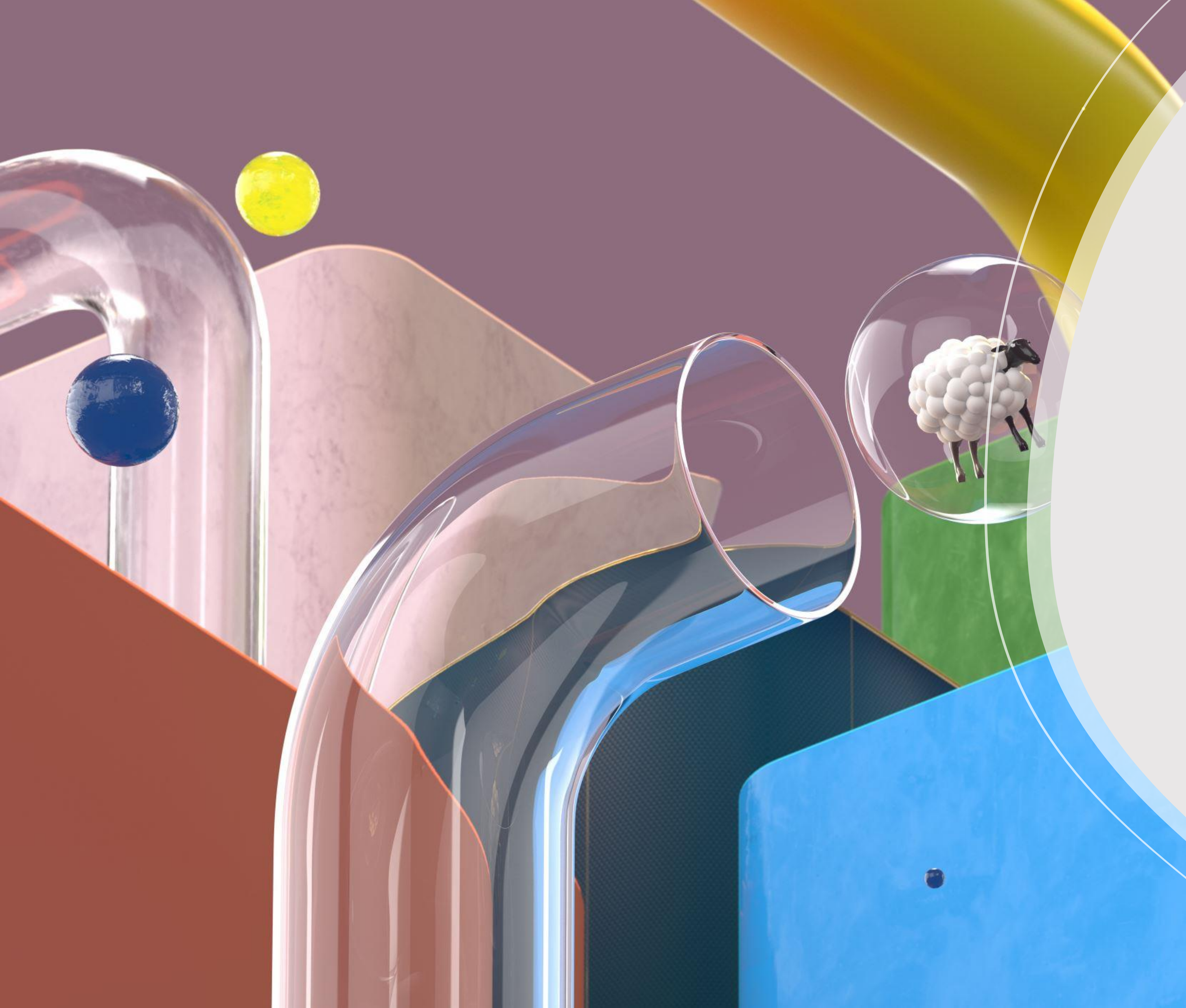
Data collection processes for supervised learning tasks vary widely, from easy and cheap (reading envelopes) to expensive and complex (medical imaging).

Unsupervised learning lacks known output data; examples include identifying topics in blog posts, segmenting customers based on preferences, and detecting abnormal website access patterns.

Both supervised and unsupervised learning require a representation of input data, often conceptualized as a table with rows as data points and columns as features.

Features are the properties describing data points, and the process of creating a good data representation is called feature extraction or feature engineering.

The quality of predictions depends on the information contained in the data, emphasizing the importance of relevant features for accurate machine learning outcomes.



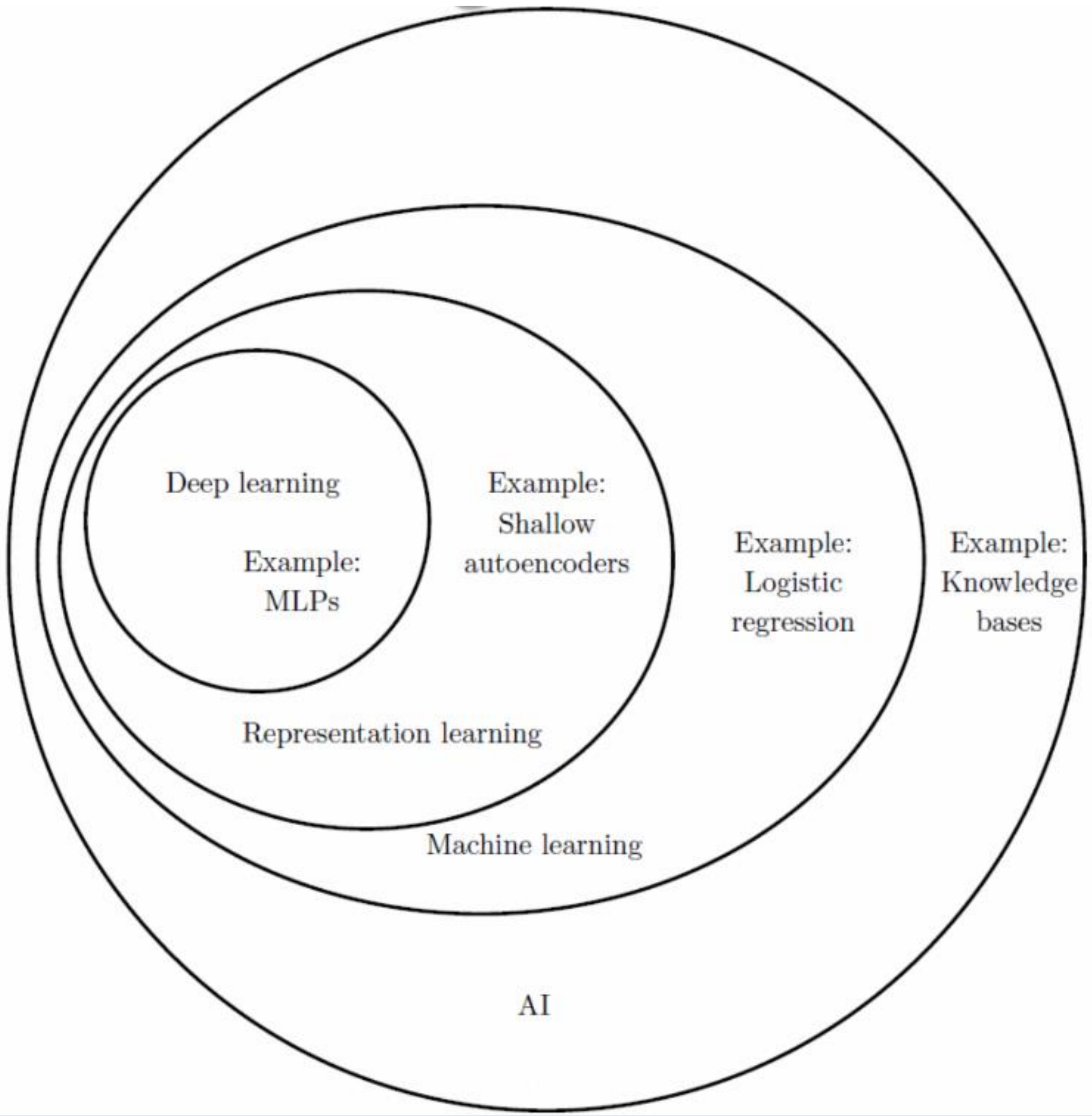
How Machine Learning works?

- For example, suppose an environmental conservation organization wants volunteers to identify and catalog different species of wildflower using a phone app.
- The following animation shows how machine learning can be used to enable this scenario.

ML Application Example

- A team of botanists and scientists collect data on wildflower samples.
- The team labels the samples with the correct species.
- The labeled data is processed using an algorithm that finds relationships between the features of the samples and the labeled species.
- The results of the algorithm are encapsulated in a model.
- When new samples are found by volunteers, the model can identify the correct species label.

The AI bubble



Interpreting Machine Learning

- A computer is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task in T, as measured P, improves with experience E. (Tom Mitchell, 1998)

Learning Algorithms

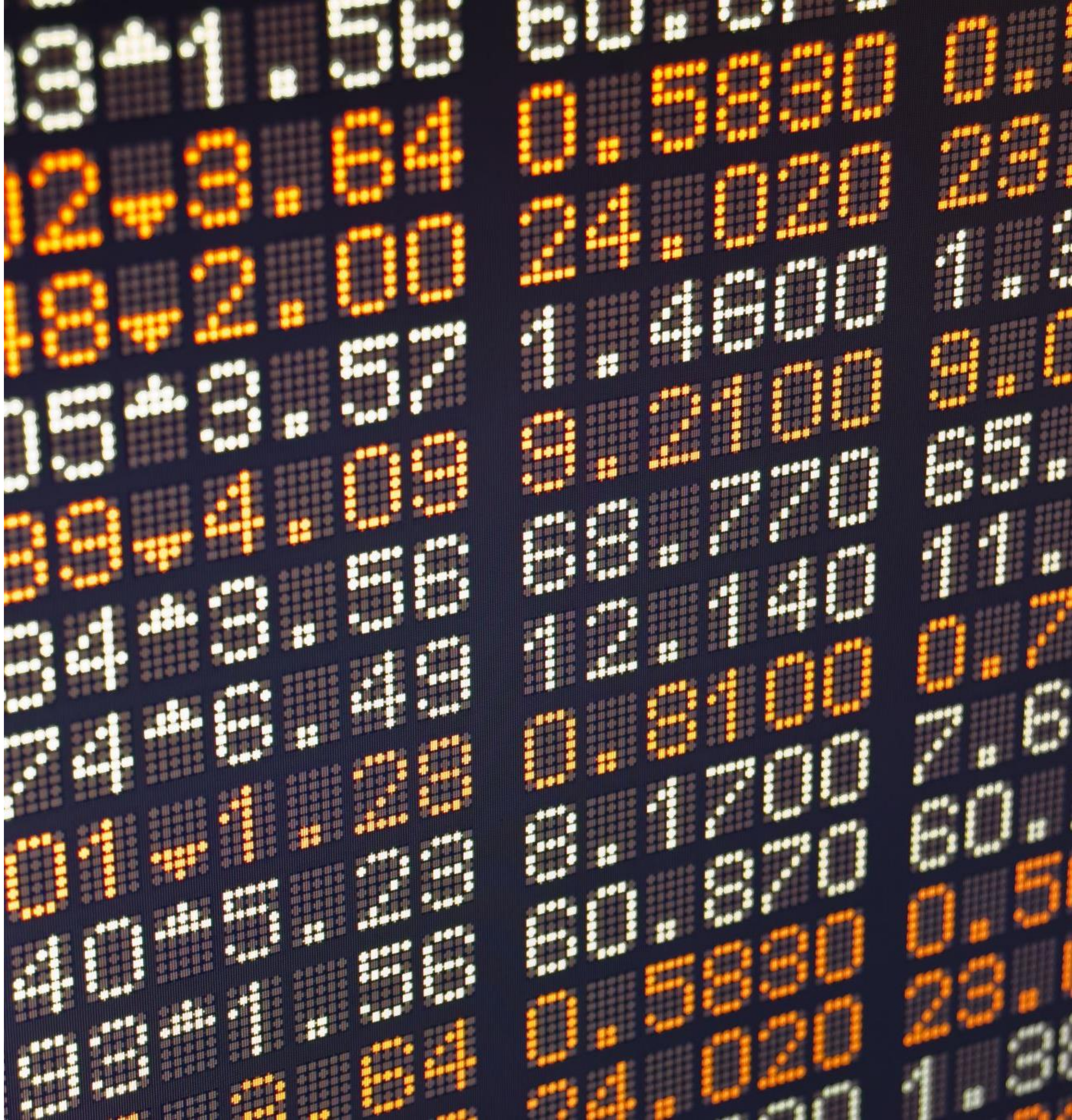
- A field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959)
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Michell, 1997).

- E: the experience of playing thousands of games
- T: playing checker's game
- P: the fraction of games it wins against human opponents

- By its definition, Samuel's program has learned to play checkers

Task T

- Classification
- Regression
- Clustering



Performance P

- Loss Function

Experience E

- Data with discrete labels
- Data with continuous labels
- Data without labels

The concept of learning in an ML system

- Learning = Improving with E at some T with respect to P
- $Learning = f(T, E, P)$

ML Systems

The Checker Learning Problem

- A computer program that learns to play checkers might improve its performance as measured by its ability to win at the class of tasks involving playing checkers games, through experience obtained by playing games against itself.
- T: Playing checkers
- P: % of game won against opponents
- E: Playing practice game against itself

The Handwritten Recognition Problem

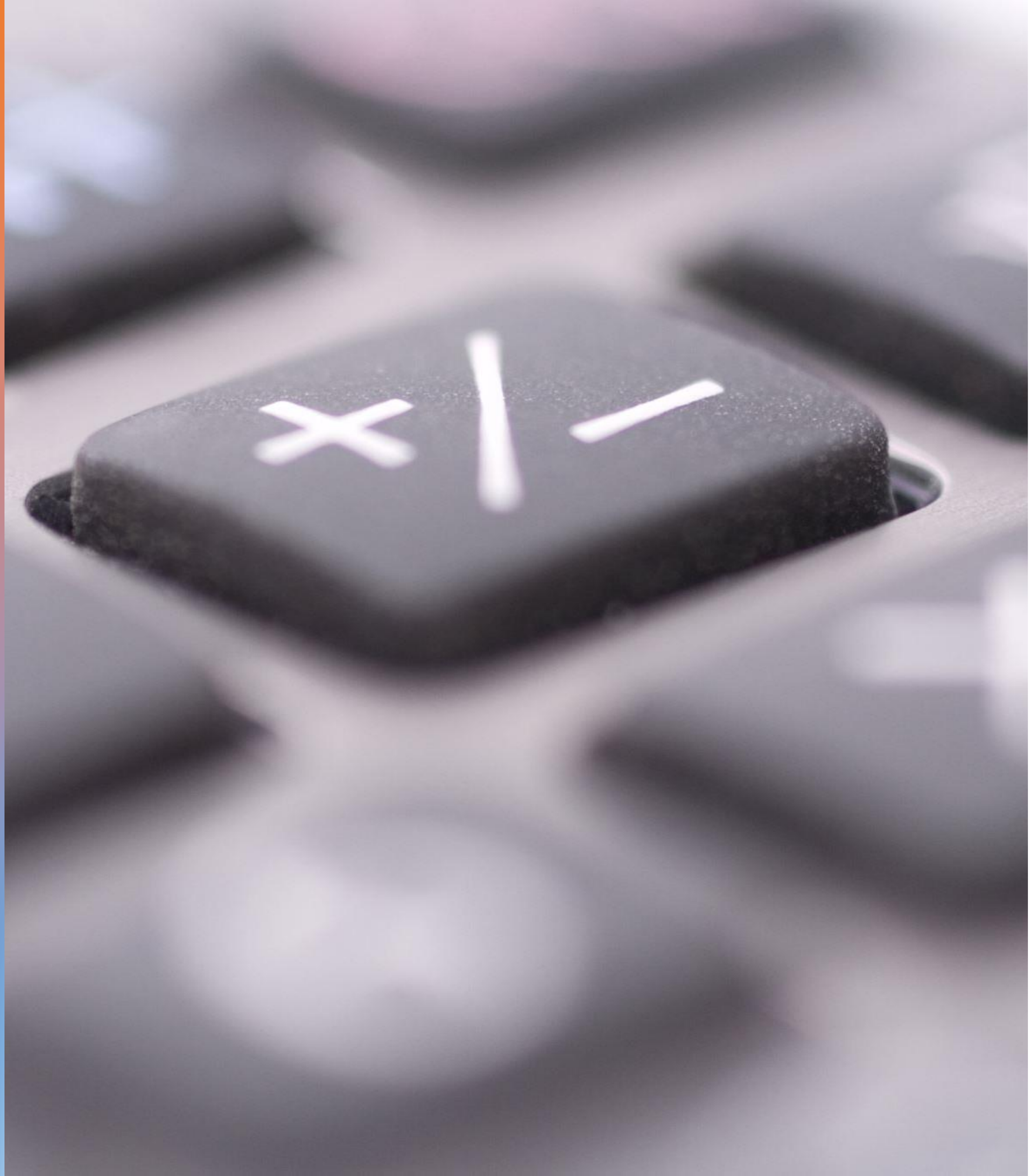
- T: Recognizing and classifying handwritten word within images
- P: % of words correctly classified
- E: Database of handwritten words with given classifications

Robot Driving Learning Problem

- T: Driving on public highways using vision sensors.
- P: Distance traveled before an error occurs based on a person's perspective.
- E: Sequence of images and steering commands recorded while observing a human driver.

E-Mail Spam Filtering

- T: Identify if the e-mail is a spam e-mail.
- P: % of spam e-mails that were filtered; % of ham e-mails that were incorrectly filtered-out
- E: A data source of ham and spam e-mails that were labeled by knowledgeable people.

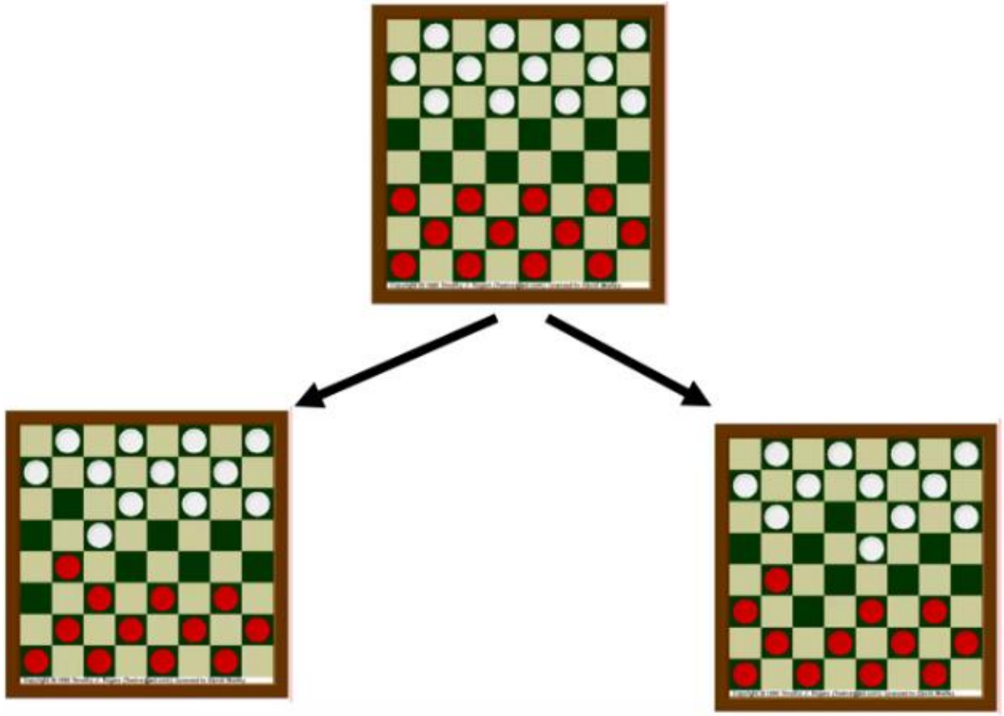


Choosing the Training Experience

- The type of training experience E available to a system can have a significant impact on success or failure of the learning system.
- A key attribute is whether E provides direct or indirect feedback about the choices made by the performance system.

Direct Learning

- Approach of a T learning from E, having an input-output pairs to learn the meaning of the input toward a specific output.

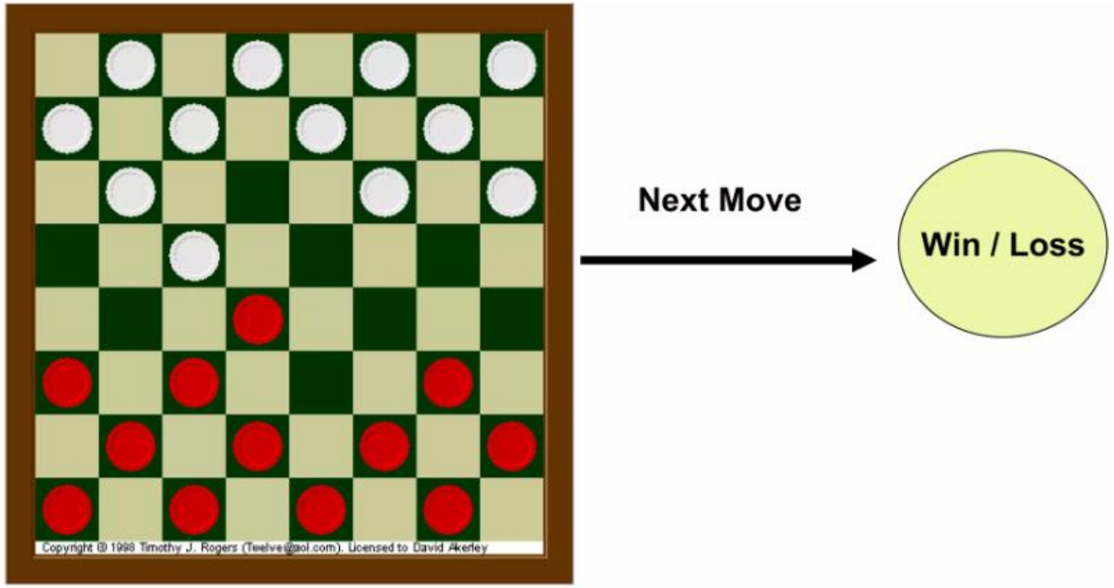


Supervised Learning

- Learns from a dataset $\mathcal{D} = \{(x^1, y^1), (x^2, y^2), \cdots, (x^n, y^n)\}$

Indirect Learning

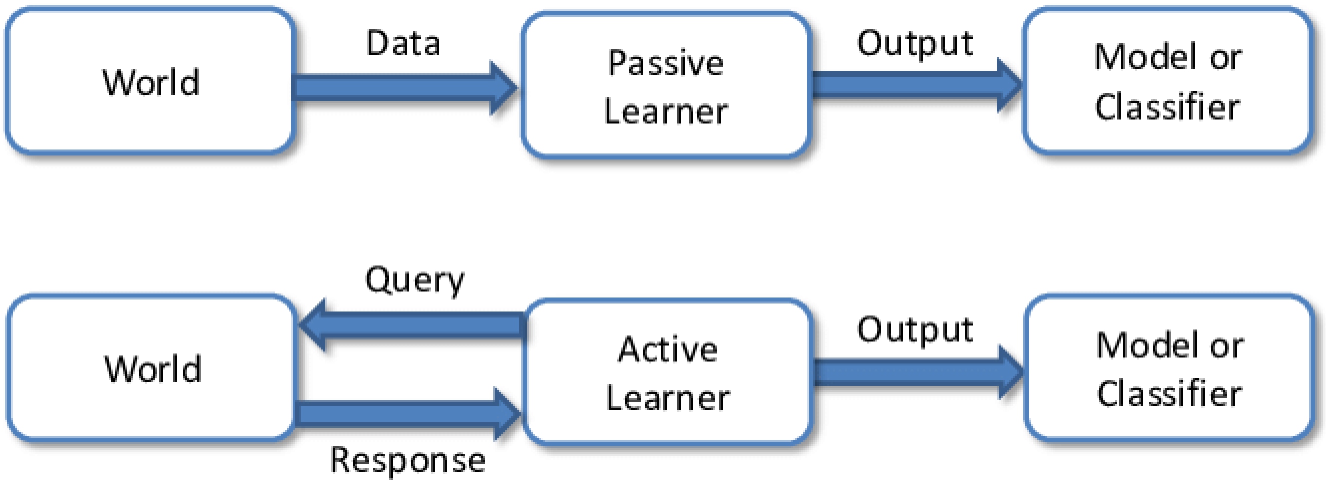
- Approach of a T learning from E without any given information or meaning while achieving a certain goal.



Unsupervised Learning

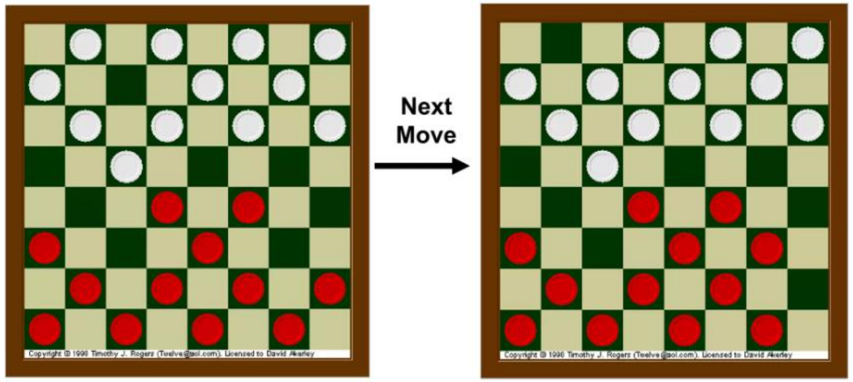
- Learns from a dataset $\mathcal{D} = \{x^1, x^2, \cdots x^n\}$

Active and Passive Learner



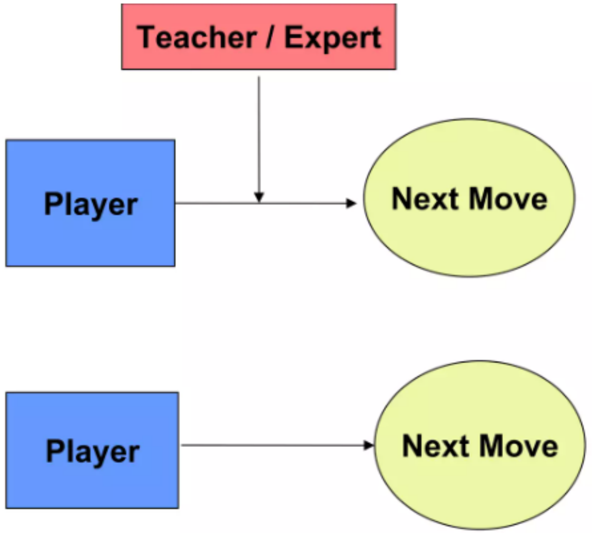
Passive Learning

- A scenario where the learning algorithm does not actively select or query for specific training examples but instead learns from a fixed set of labeled data



Active Learning

- Degree to which the learner controls the sequence of training examples.
- The learner can interact with an oracle or teacher to query for information, choose specific instances for learning, or even generate its own examples.



- How well the training experience represents the distribution of examples over which the final system performance P must be measured.
- Learning will be most reliable when the training example follow a distribution similar to that of future test examples.

Target Function

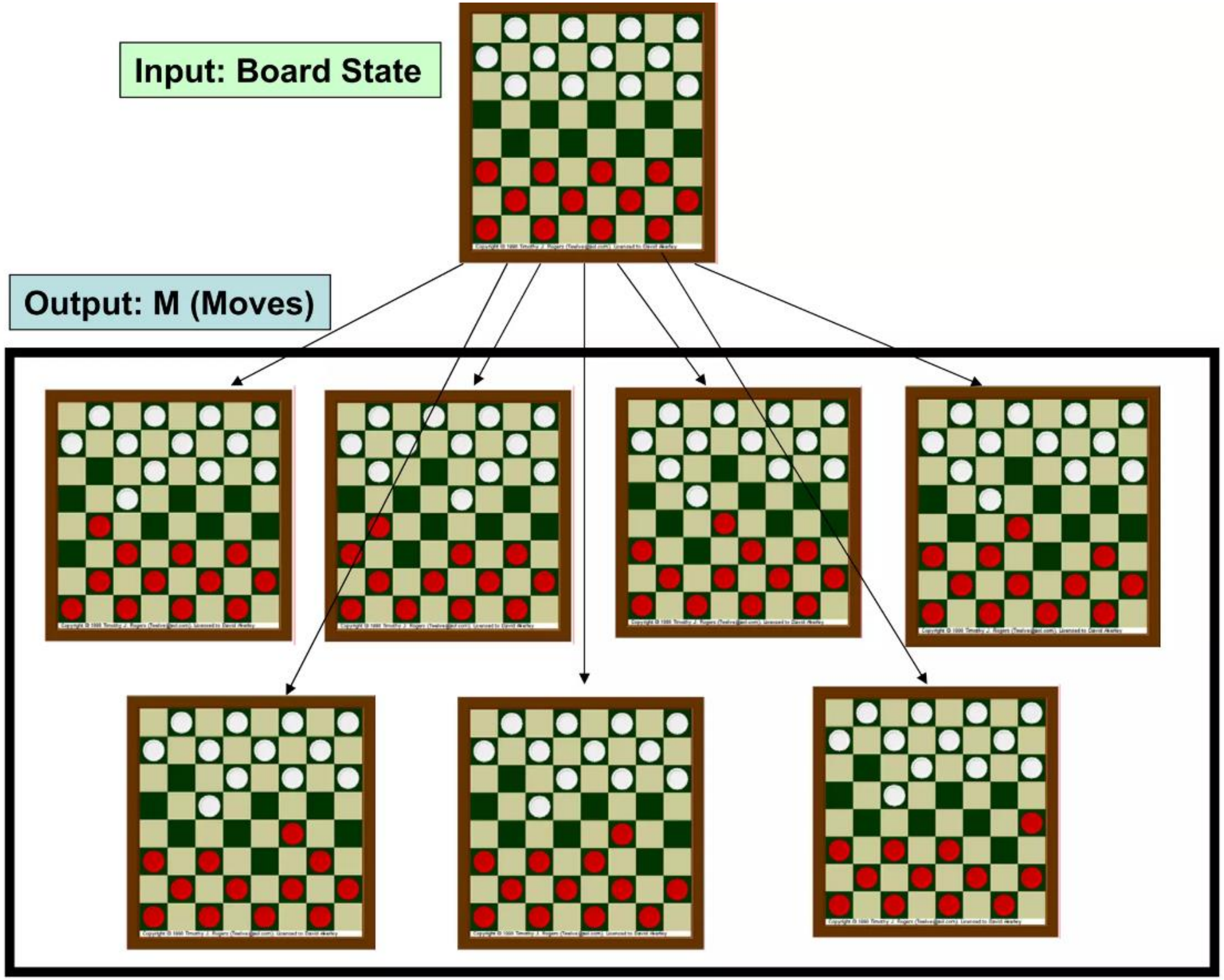
- Choose move: $B \to M$
- Choose move is a function $f(\cdot)$, where input B is the set of legal board states and produces M which is the set of legal moves.
- This also represents the relationship between the mapped input and label on a dataset.

- B = Board states that produce possible or legal moves.
- M = The legal moves that the player can do.

Human Perspective

$$M = f(B)$$

Human cognition can define exactly the board states, aiding in the identification of the possible moves.



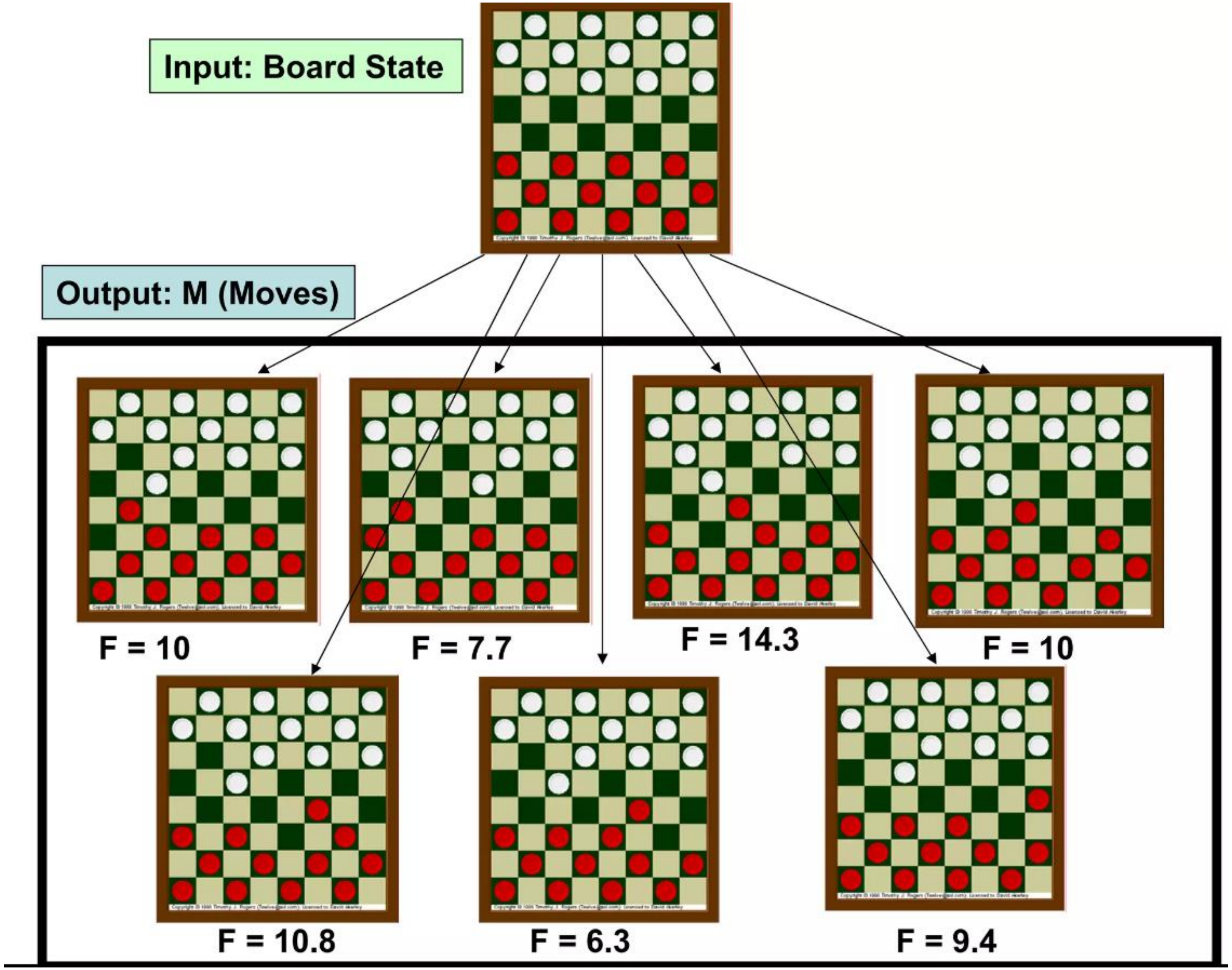
Machine Perspective

$$R = f(B)$$

R set of real numbers that represent the ideal target function. What we want to achieve!

The real numbers will give a certain representation of M based on the value representation of B.

The given set of real numbers will receive a random set of weights w^j that will adjust depending on the error rate.



Goal of achieving the target function

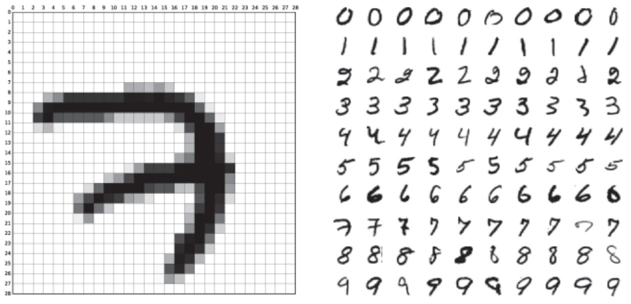
- The representation $\hat{R}$ aims to attain R based on the given weights assigned and other parameters with it.
- if b_i is a final board state that is won, then $R(b_i) = 100$
- if b_i is a final board state that is lost, then $R(b_i) = -100$
- if b_i is a final board state that is drawn, then $R(b_i) = 0$

- If b_i is a not a final state in the game, then $R(B) = R(b_i)$, where b_i is the best final board state that can be achieved starting from b_i and playing optimally until the end of the game (assuming the opponent plays optimally, as well).

Representation

- Refers to the way data is encoded or structured to be fed into a machine learning algorithm.
- It's about how you present the data to the model.

- Example: In a handwriting recognition task, representation of a something are defined by pixel values assigned in a specific manner and specific values.



(a) MNIST sample belonging to the digit '7'. (b) 100 samples from the MNIST training set.

Target function representation

Representation	Description
x^1	Number of white pieces on the board
x^2	Number of red pieces on the board
x^3	Number of white kings on the board
x^4	Number of red kings on the board
x^5	Number of white pieces threatened by red (capturable by red on the next turn)
x^6	Number of red pieces threatened by white (capturable by white on the next turn)

$$\hat{R}(b_i) = w^0 + w^1x^1 + w^2x^2 + w^3x^3 + w^4x^4 + w^5x^5 + w^6x^6$$

Target Function and Objective Function

- Target function is the ideal function you want your model to learn (True relationship between size and price).
- The objective function (or loss function) is the mathematical measure used to quantify how well your model is currently performing with respect to the target function.

- Target Function: Price of House = 50 × Size of House + 10,000.
- Objective Function: The difference from the real price and predicted price.

Choosing a Representation for the Target Function

- T: Playing checkers
- P: % of games won in the world tournament
- E: games played against itself
- F: $R = f(B)$
- $\hat{R}(b_i) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$

- The problem of learning a checkers strategy reduces to the problem of learning values for the coefficients w_0 through w_6 in the target function representation.

Determining the Approximation Algorithm

- Attaining the target function $\hat{R}$ requires training data.
- Training samples define the board state and training values.
- We can define them as an ordered pair $\left(b, R_{train}(b)\right)$.

- Example: the board state b represents a scenario where black has won (notably, x2 = 0 signals that red has no remaining pieces). In this context, the target function value $R_{train}(b)$ is designated as +100.

- $\left\langle\langle w_1 = 3, w_2 = 0, w_3 = 1, w_4 = 0, w_5 = 0, w_6 = 0\rangle, +100\right\rangle$

- We can say that +100 is an arbitrary value that indicates that the black player has won the game based on *b.*

Weight Adjustment

- To make accurate predictions in machine learning, we fine-tune our weight values.
- Think of weights as factors that control the importance of different features.
- Our goal is to find the best weights that match our training data.
- We measure how well our predictions match reality using the squared error.
- It's like calculating the difference between our prediction and the actual outcome for each example, squaring it, and adding them up.
- The squaring helps emphasize larger errors, ensuring we pay more attention to significant mistakes.
- Considering the best weights should adhere to the best fit to the training data.
- Weight adjustment bases from the closest values of $\hat{R}$ that has the least errors.
- The approximation algorithm of choice would be basing on these goals.
- $E \equiv \sum_{\langle b, R_{train}(b)\rangle \, \epsilon \, training \, samples} \left(R_{train}(b) - \hat{R}(b)\right)^2$

Least Mean Square (LMS) Rule

- Our go-to method for adjusting weights is the Least Mean Squares (LMS) rule.
- It's like a smart trial-and-error process. For each training example, LMS adjusts the weights a bit to reduce the error.
- It repeats this process, gradually finding the best weights that minimize errors across all examples.

Why do we square?

- Penalizing Mistakes: Squaring magnifies bigger errors, making sure we don't ignore important deviations from the correct values.

- Consistent Direction: Squaring ensures all errors contribute positively to our total error. This consistency prevents the cancellation of positive and negative errors.
- $E \equiv \sum_{\langle b, R_{train}(b)\rangle \, \epsilon \, training \; samples} \left(R_{train}(b) - \hat{R}(b) \right)^2$

- This equation quantifies the overall difference between our predictions and the actual outcomes across all training examples.

Idea of LMS at work

- $New\ Weight = Old\ Weight + Learning\ Rate\ \times\ (Actual\ Outcome\ - Predicted\ Outcome)\ \times Corresponding\ Feature$

- Here, the learning rate controls how much we change the weights in each step. It’s like taking small, careful steps to reach the best weights. This prevents overestimation and provides stability.

- This method might sound simple, but in certain situations, it’s proven to converge to the best weights that give us the closest approximation to our training data.

- So, by adjusting weights smartly, we're training our machine learning model to make better predictions!

Finding the learning Rate

- There isn't a one-size-fits-all solution, and the optimal learning rate often depends on the specific characteristics of your data and the model architecture.

Problem Framing

Understanding the Problem

- State the goal for the product you are developing or refactoring.
- Determine whether the goal is best solved using, predictive ML, or a non-ML solution.
- Verify you have the data required to train a model if you're using a predictive ML approach.

Stating your goals

- Begin by stating your goal in non-ML terms. The goal is the answer to the question, "What am I trying to accomplish?"
- The following table clearly states goals for hypothetical apps:

Application	Goal
Mail app	Detect spam emails
Banking app	Detect fraud transactions
Ecommerce app	Determine bad comments

Clear use of ML

- Most people will think ML is a general tool.
- ML is a specialized or tailored tool for specific problems.
- ML may come simple for certain problems but consider the effort in bulk.
- Focus on quality and cost-efficiency when determining if ML is needed.

Data is a driving force for ML

- Abundant
- Consistent and reliable
- Trusted
- Available
- Correct
- Represents the real-world

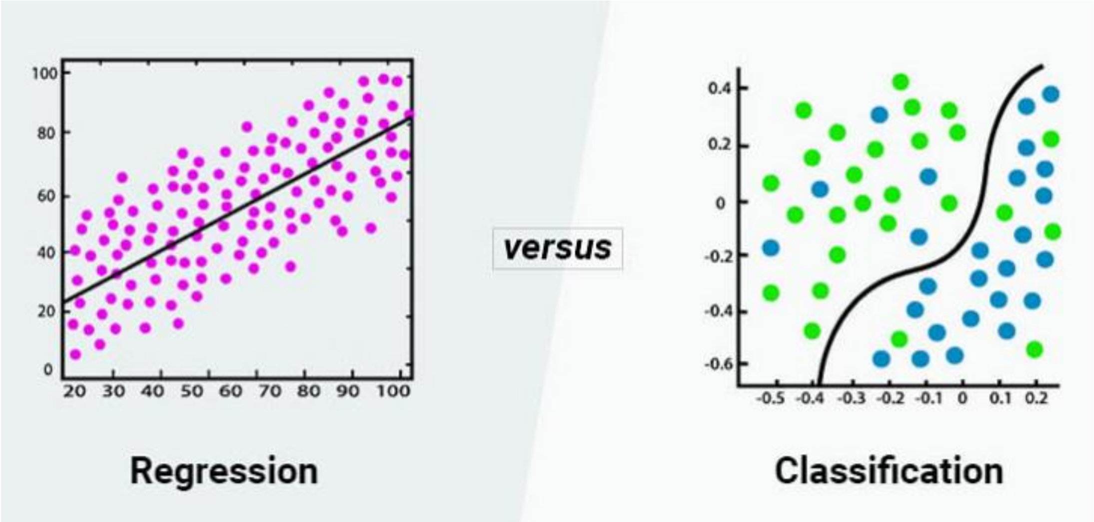
Machine Learning Problems

Supervised

Unsupervised

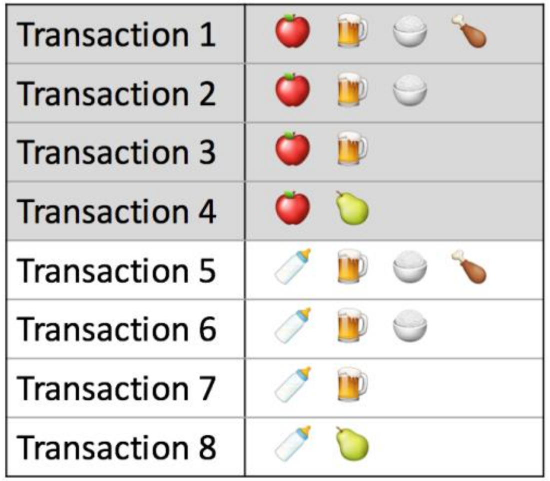
Common Supervised ML problems

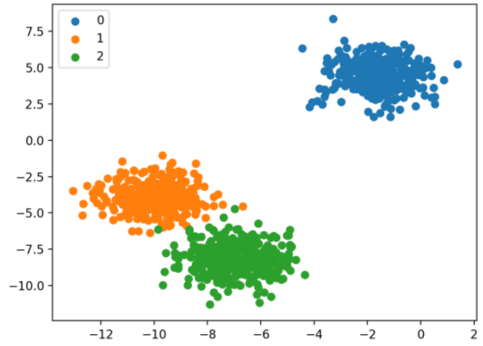
- Classification
- Regression

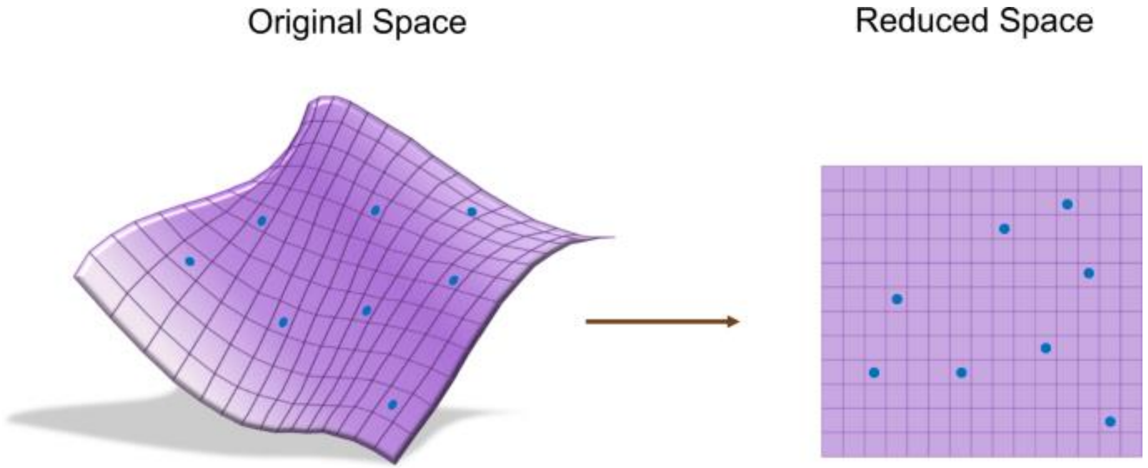


Common Unsupervised ML problems

- Clustering
- Dimensionality reduction
- Association analysis







Some applications of supervised ML solutions

- Spam email detection (classification)
- Animal breed classification (classification)
- House price prediction (regression)
- Crop yield prediction (regression)

Some applications of unsupervised ML solutions

- Customer segmentation (clustering)
- Fraud detection (clustering)
- File Compression (dimensionality reduction)
- Face Recognition (dimensionality reduction)
- Market isle arrangement (association analysis)
- Supply chain optimization (association analysis)

Remember

Supervised ML learns from **data with features and labels** and **predict unseen samples** to determine their labels.

Unsupervised ML learn **common data patterns** or **trends** and determine how each point **comes together** or **separate**.

ML works on a **specific task**, **gains experience**, and is **defined based on its performance** in reducing errors, where it requires a certain amount of learning phase.

Aside from data, it requires computing resource.

Activity

- Look for a pair of a problem and a solution (No application of software or algorithms).
- From the identified problem and solution, think of an idea on how Machine Learning can provide a better solution.
- Identify the type of problem, whether supervised or unsupervised would be the best choice.
- If supervised, determine the possible features of the data and its paired label. If unsupervised, determine the goal adequately.
- Determine its T, P, and E
- Write a one-page paper with references.