# Machine Learning

# Introduction

Kien C Nguyen

August 20, 2024

INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

# Lecture Plan (1)

| No | Lecture | #hrs |
|---|---|---|
| 1 | Introduction | 3 |
| 1.1 | - An overview of Machine Learning | |
| 1.2 | - Taxonomy | |
| 2 | Linear Regression & Gradient Descent | 3 |
| 3 | K-nearest neighbors | 3 |
| 4 | Logistic Regression | 3 |
| 5 | Decision Trees & Decision Tree-Based Models | 3 |

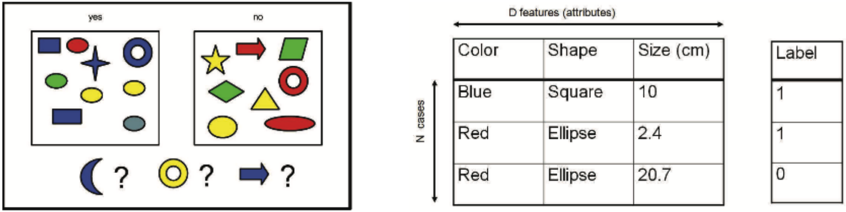
Table: Lecture plan

# Lecture Plan (2)

| No | Lecture | #hrs |
|---|---|---|
| 6 | Feature Engineering & Model Evaluation | 3 |
| 7 | K-means Clustering | 3 |
| 8 | Principal Component Analysis | 3 |
| 9 | Neural Networks | 3 |
| 10<br>10.1<br>10.2 | Reinforcement Learning<br>- Markov Decision Process<br>- Reinforcement Learning | 3 |

Table: Lecture plan

# An example of learning



| Color | Shape | Size (cm) | Label |
|---|---|---|---|
| Blue | Square | 10 | 1 |
| Red | Ellipse | 2.4 | 1 |
| Red | Ellipse | 20.7 | 0 |

(a) (b)

Figure: An example of learning

Figure from [1]

[1] Kevin P. Murphy. Machine learning: a probabilistic perspective. Cambridge, Mass. : MIT Press, 2013.

# What is Machine Learning?

## Definition

Machine Learning (ML) is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) from data, without being explicitly programmed (Wikipedia).

https://en.wikipedia.org/wiki/Machine_learning

- In supervised learning, we teach a computer program to estimate or predict a quantity, or differentiate among multiple classes. We use a training dataset with labels.
- In unsupervised learning, we do not have labels. Instead, given a dataset, we attempt to find the structure of data or relationship among different fields.

# What is Machine Learning?

https://en.wikipedia.org/wiki/Machine_learning
Tom M. Mitchell (an American computer scientist) provided a widely quoted, more formal definition of the algorithms studied in the machine learning field:

**Definition**

"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."

# A Machine Learning problem

In the "New York City Taxi Fare Prediction" Competition on Kaggle (https://www.kaggle.com/c/new-york-city-taxi-fare-predic data), competitors have to predict the cost (in USD) of a taxi ride in New York City given the following features:

- *pickup_datetime* – timestamp value indicating when the taxi ride started.
- *pickup_longitude* – float for longitude coordinate of where the taxi ride started.
- *pickup_latitude* – float for latitude coordinate of where the taxi ride started.
- *dropoff_longitude* – float for longitude coordinate of where the taxi ride ended.
- *dropoff_latitude* – float for latitude coordinate of where the taxi ride ended.
- *passenger_count* – integer indicating the number of passengers in the taxi ride.

# A Machine Learning problem

Competitors are given a training dataset (*train.csv*) with input features and target *fare_amount* values. They will then have to predict the *fare_amount* for each row of input features in a test set (*test.csv*). Using Tom M. Mitchell's definition of Machine Learning discussed in Lecture "Introduction to Machine Learning" (for Parts (a) and (b)),

**(a)** Describe the experience $E$ and the class of tasks $T$ of the algorithms used to solve this problem.

**(b)** Propose a performance measure $P$ that we can use to rank the competitors' submissions.

**(c)** Is this problem a supervised learning one or an unsupervised learning one? Justify your answer.

**(d)** Is this problem a classification or a regression problem? Justify your answer.

# Paradigms of Machine Learning (1)

Machine Learning these days can be divided into four main paradigms.

- In the predictive or **supervised learning** approach, given an input vector $\mathbf{x}$, we have to predict an output $y$, where $y$ can be categorical or numerical.
  - The goal is to learn a mapping from inputs $\mathbf{x}$ to outputs y, given a labeled set of input-output pairs $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$.
  - Here $D$ is called the training set, and $N$ is the number of training examples.
- In the descriptive or **unsupervised learning** approach, we are given only inputs $D = \{\mathbf{x}_i\}_{i=1}^{N}$, and the goal is to find structures / patterns or relationships in the data. This is sometimes called knowledge discovery.

# Paradigms of Machine Learning (2)

- **Self-supervised learning** is a type of machine learning where the model learns to predict part of the input from other parts of the input.
  - This approach allows the model to generate its own labels from the input data,
  - It eliminates the need for manually labeled data, which is a significant advantage when labeled data is scarce or expensive to obtain.
- **Reinforcement learning** (RL) is a type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize the cumulative reward.

# Supervised learning [1]

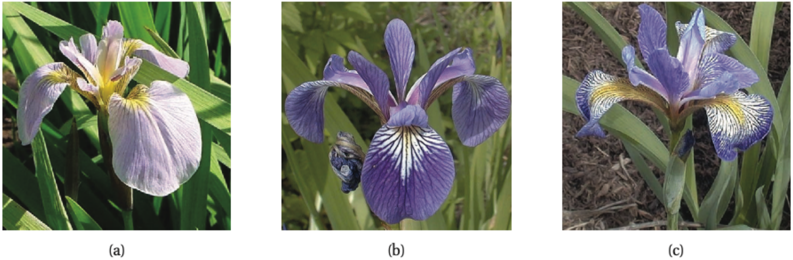
For supervised learning,

- when $y_i$ is categorical, the problem is known as classification or pattern recognition. For example, the problem of classifying emails into 'spam' and 'not spam'.
- When $y_i$ is real-valued, the problem is called regression. For example, the problem of predicting the income level.
- Another variant, known as ordinal regression, occurs where label space Y has some natural ordering, such as grades A–F.

# Classification [1]

- Here the goal is to learn a mapping from inputs $\mathbf{x}$ to outputs $y$, where $y \in \{0, \ldots, C-1\}$, with $C$ being the number of classes.
- If $C = 2$, this is called binary classification ($y \in \{0, 1\}$
- if $C > 2$, this is called multiclass classification.
- If the class labels are not mutually exclusive (e.g., somebody may be classified as tall and strong), we call it multi-label classification, but this is best viewed as predicting multiple related binary class labels (a so-called multiple output model).

# Classification - Example [1]



(a)

(b)

(c)

Figure: Three types of iris flowers: setosa, versicolor and virginica [1].

# Classification - Example [1]

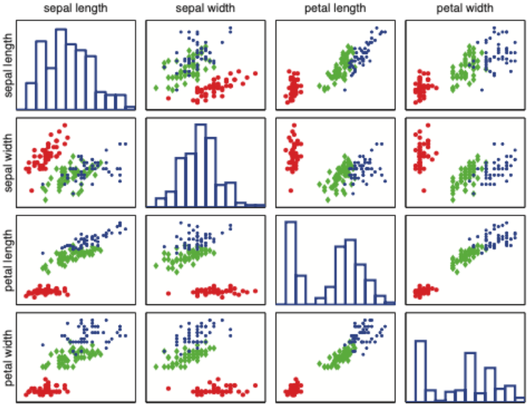
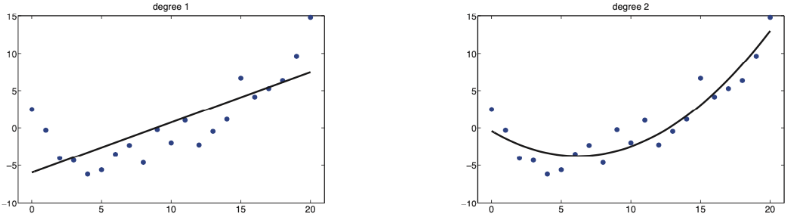


Figure: Visualization of the Iris data as a pairwise scatter plot. The diagonal plots the marginal histograms of the 4 features. The off diagonals contain scatterplots of all possible pairs of features. Red circle

# Regression [1]

- Regression is just like classification except the response variable is continuous.
- The next slide shows a simple example: we have a single real-valued input $x_i \in \mathbb{R}$, and a single real-valued response $y_i \in \mathbb{R}$.
- We consider fitting two models to the data: a straight line and a quadratic function.
- Various extensions of this basic problem can arise, such as having high-dimensional inputs, outliers, non-smooth responses

# Regression - Example [1]



(a)

(b)

Figure: (a) Linear regression on some 1D data. (b) Same data with polynomial regression (degree 2) [1].

# Unsupervised Learning

- We now consider unsupervised learning, where we are just given input data, without labels.
- The goal is to discover structures and patterns in the data
  - this is sometimes called knowledge discovery
- Unlike supervised learning, we are not told what the desired output is for each input.

# Unsupervised Learning - K-means clustering [1]

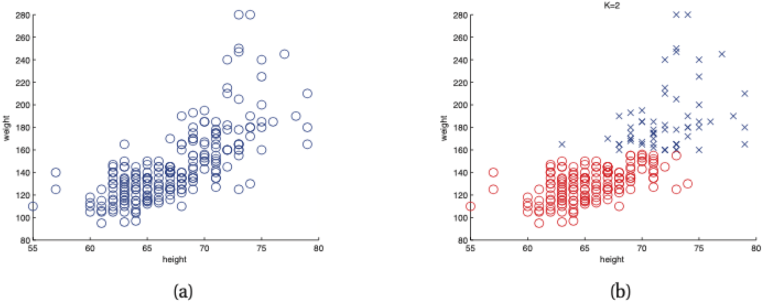
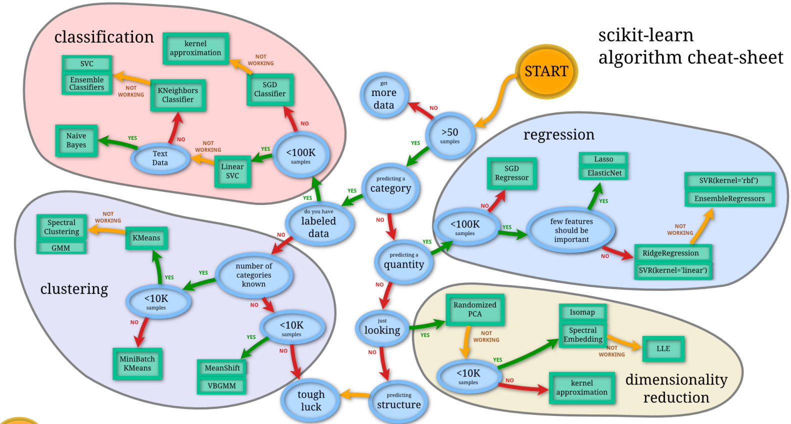


Figure: (a) The height and weight of some people. (b) A possible clustering using K = 2 clusters [1].

# scikit-learn algorithms



Back

Figure: scikit-learn algorithms

# Linear model for regression

Linear model for regression is a linear combination of the input variables. It assumes the dependency of the response variable $y$ on the explanatory variables $\mathbf{x}$ is linear.

## Formula

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D = w_0 + \sum_{j=1}^{D} w_j x_j$$

where

- $y \in \mathbf{R}$: response variable, dependent variable, outcome.
- $D$: number of dimensions of the input vector $\mathbf{x}$.
- $\mathbf{x} = (x_1, ..., x_D)^T$: input vector (explanatory variable, independent variable, features).
- $\mathbf{w} = (w_0, ..., w_D)$: parameters.
- $D + 1$: total number of parameters.

# Linear regression

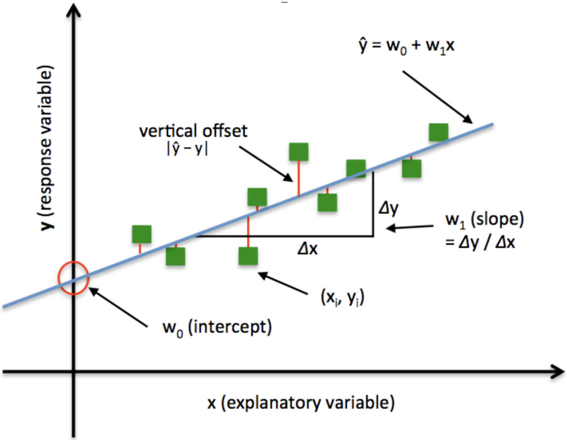


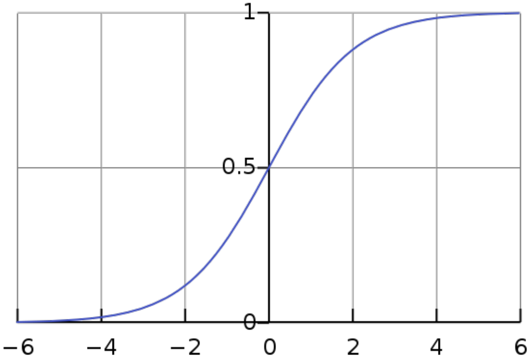
Figure: http://rasbt.github.io/

# Logistic Regression

- Recall that in linear regression, $\hat{y} = \mathbf{w}^T\mathbf{x}$.
- This model can only be used if $y$ is not upper-bounded and not lower-bounded.
- In logistic regression, we predict the probability of a Positive Class (vs a Negative Class).
- E.g. Probability that an email is Spam, probability that a customer is a Bad customer.

- Use a function $\Phi(\mathbf{w}^T\mathbf{x})$
- As this is a probability, we want $0 \leq \Phi(\mathbf{w}^T\mathbf{x}) \leq 1$
- Sigmoid function (Logistic function)

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



Source:
https://en.wikipedia.org/wiki/Logistic_regression

# Perceptron

1. SVMs are extensions of **perceptron** classifier
2. Given that the training data is **linearly separable**, the perceptron algorithms find a $d-1$ dimensional hyperplane that perfectly separates the $+1's$ from the $-1's$
3. Mathematically, the goal is to learn a weight $w \in \mathbb{R}^D$ and a bias term $b \in \mathbb{R}$, that satisfy the linear separability constrains:

$$\forall i, = \begin{cases} w^T x_i - b \geq 0 & \text{if } y_i = 1 \\ w^T x_i - b \leq 0 & \text{if } y_i = -1 \end{cases} \tag{1}$$

   Equivalently, $\forall i, y_i(w^T x_i - b) \geq 0$

4. The resulting decision boundary is a hyperplane $H = \{x : w^T x - b = 0\}$

# Motivation for SVMs

- Perceptrons have two major shortcomings
  - If data is not linearly separable, the perceptrons fails to find a stable solution
  - If the data is linearly separable, the perceptrons could find infinitely many decision boundaries ⇒ generalization issues

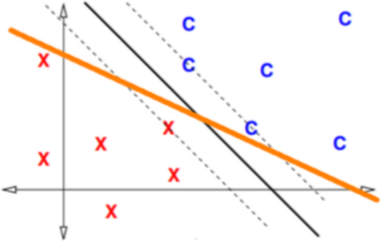


Figure: Two possible decision boundaries under the perceptron. The X's and C's represent the +1's and -1's respectively. Source: UC Berkeley CS189

# Feature Engineering Overview

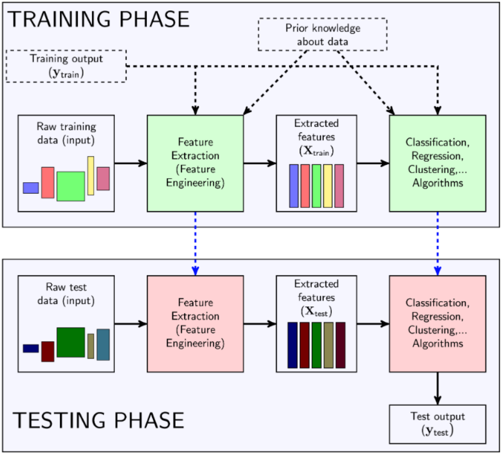

TRAINING PHASE
Prior knowledge about data
Training output ($\mathbf{y}_{\text{train}}$)
Raw training data (input)
Feature Extraction (Feature Engineering)
Extracted features ($\mathbf{X}_{\text{train}}$)
Classification, Regression, Clustering,... Algorithms
Raw test data (input)
Feature Extraction (Feature Engineering)
Extracted features ($\mathbf{X}_{\text{test}}$)
Classification, Regression, Clustering,... Algorithms
Test output ($\mathbf{y}_{\text{test}}$)
TESTING PHASE

Figure: A standard machine learning pipeline (source: Machine Learning Co Ban, Vu Huu Tiep)

# Different types of variable in statistics

**Numerical (quantitative)**

- **Discrete:** integer values, typically counts. E.g. age, sick days per year.
- **Continuous:** takes any value in a range of values. E.g. weight, height.

**Categorical (qualititive)**

- **Nominal:** mutually exclusive and unordered categories. E.g. sex (male/female), blood types (A/B/AB/O).
- **Ordinal:** mutually exclusive and ordered categories . E.g. disease stage (mild/moderate/severe).

## Dimensionality Reduction

- When we work with a dataset with a lot of features, it is difficult to understand or explore the relationships between the features.
- Without a thorough understanding of the data, we might overfit our model or overlook violations of the assumptions of the algorithm, like the independence of features in linear regression.
- This is where dimensionality reduction comes in.
- In machine learning, dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.
- By reducing the dimension of your feature space, we have fewer relationships between features to consider which can be explored and visualized easily and also you are less likely to overfit your model.

# Dimensionality Reduction

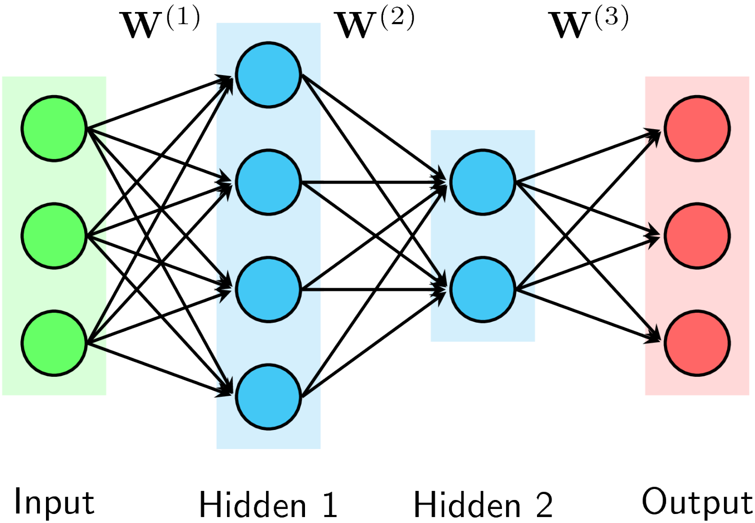
Dimensionality reduction can be achieved in the following ways:

- Feature Elimination: You reduce the feature space by eliminating features. This has a disadvantage though, as you gain no information from those features that you have dropped.
- Feature Selection: You apply some statistical tests in order to rank them according to their importance and then select a subset of features for your work. This again suffers from information loss and is less stable as different test gives different importance score to features. You can check more on this here.
- Feature Extraction: You create new independent features, where each new independent feature is a combination of each of the old independent features. These techniques can further be divided into linear and non-linear dimensionality reduction techniques.

# Principal Component Analysis (PCA)

- Principal Component Analysis or PCA is a linear feature extraction technique.
- It performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.
- It does so by calculating the eigenvectors from the covariance matrix.
- The eigenvectors that correspond to the largest eigenvalues (the principal components) are used to reconstruct a significant fraction of the variance of the original data.

# Multi-layer Perceptrons



# K-means Algorithm

**Input:**

- K (number of clusters)
- $\{x^{(i)}\}_{i=1}^{N}$

```
Initialization:
Randomly initialize $K$ cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^D$
while Assignment changes from the last iteration do
    Assignment:
    for $i = 1$ to $N$ do
        Assign $x^{(i)}$ to the cluster with the minimum distance
        $d(x^{(i)}, \mu_k)$
    end
    Update:
    for $j=1$ to $K$ do
        $\mu_k$ = mean of all the points assigned to cluster $k$
    end
end
```

# Self-supervised Learning

The model learns to predict part of the input from other parts of the input, generating its own labels from the input data.

- Pretext Task: Surrogate tasks created from the input data to provide learning objectives.
- Representation Learning: Learning useful data representations for downstream tasks.

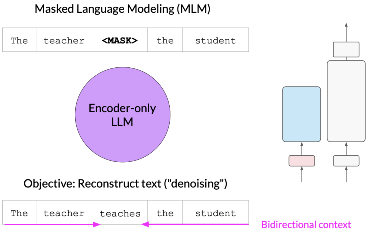


Figure: Masked Language Modeling (Coursera - Generative AI and LLMs)

# Reinforcement Learning

- Definition: Reinforcement Learning $(\mathrm{RL})$ is a type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize cumulative rewards.
- Key Concepts:
  - Agent: The learner or decision-maker.
  - Environment: The external system the agent interacts with.
  - State: A representation of the current situation of the environment.
  - Action: Choices the agent can make.
  - Reward: Feedback from the environment, can be positive or negative.
  - Policy: Strategy used by the agent to determine actions.
  - Value Function: Estimates the expected reward for states or state-action pairs.
  - Exploration vs. Exploitation: Balancing between exploring new actions and exploiting known ones.

# References

[1] K. P. Murpy – Machine Learning – A Probabilistic Perspective, MIT Press, 2012.

[2] Vu Huu Tiep – Machine Learning co ban, https://machinelearningcoban.com/2017/02/24/mlp/

[3] UIUC CS 446 Machine Learning

[4] Andrew Ng – Coursera Machine Learning

[5] Thân Quang Khoát – Machine Learning Basics Lectures, 2021