DDA3020 Machine Learning: Lecture 1 Introduction

Jicong Fan
SDS, CUHK-SZ

September 02/04, 2025

Outline

Instructors and teaching assistants (Session 1)

Time and venue:

- Lecture: Tuesday/Thursday 1:30–2:50 pm, Teaching B Bldg 104
- Tutorial: Wednesday/Thursday 6:00–6:50 pm/7:00–7:50 pm, Teaching A Bldg 307

Instructor:

- **Jicong Fan** (fanjicong@cuhk.edu.cn)
- Personal homepage: `https://jicongfan.github.io/`
- Office hour (OH): 3:00–4:00 pm, Thursday, DY 502a.

TAs:

- **Meixi Zheng (leading TA):** 223040254@link.cuhk.edu.cn
  OH: Mon 11:00-12:00, TXB603
- **Dekun Chen:** dekunchen@link.cuhk.edu.cn
  OH: Wed 15:00-16:00, TXC604
- **Fanzeng Xia:** 223040232@link.cuhk.edu.cn
  OH: Fri 16:00-17:00, TXC611
- **Kuang Wang:** 224040348@link.cuhk.edu.cn
  OH: Mon 16:00-17:00, DYB101

Instructors and teaching assistants (Session 2)

Time and venue:

- Lecture: Tuesday/Thursday 3:30–4:50 pm, Teaching Complex C, Room 201
- Tutorial: Wednesday/Thursday 6:00–6:50 pm/7:00–7:50 pm, Teaching A Bldg 307

Instructor:

- **Juexiao Zhou** (juexiao.zhou@gmail.com)
- Personal homepage: `https://www.joshuachou.ink/about/`
- Office hour (OH): 12:00–1:00 pm, on every Tuesday, Zhixin Bldg 403b.
- Sending emails to request appointment is also preferred

TAs:

- **Meixi Zheng (leading TA):** 223040254@link.cuhk.edu.cn
  OH: Mon 11:00-12:00, TXB603
- **Dekun Chen:** dekunchen@link.cuhk.edu.cn
  OH: Wed 15:00-16:00, TXC604
- **Fanzeng Xia:** 223040232@link.cuhk.edu.cn
  OH: Fri 16:00-17:00, TXC611
- **Kuang Wang:** 224040348@link.cuhk.edu.cn
  OH: Mon 16:00-17:00, DYB101

Learning materials

Textbooks:

- Required: Andriy Burkov, “The Hundred-Page Machine Learning Book”, 2019. (read first, buy later: Link)
- Required: K. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press, 2012. (Link)
- Recommended: Andreas C. Muller and Sarah Guido, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, O’Reilly Media, Inc., 2017. (The PDF will be uploaded to the BB system, only used for learning)
- Recommended: Jeff Leek, “The Elements of Data Analytic Style: A guide for people who want to analyze data”, Lean Publishing, 2015. (Link)
- Recommended: C. Bishop. Pattern Recognition and Machine Learning. Springer, 2011. (Link)

Grading policy

Grading:

- 30% Written homework assignments (3 times, W1, W2, W3)
- 20% Programming homework assignments (Python) (twice, P1, P2)
- 10% Project (final code submission 5% & final report submission 5%)
- 40% Final exam

Submission: You have 2 weeks to **independently** complete each assignment (written + programming). Late submission will get discounted score: $(0, 48]$ hours $\rightarrow$ 50%; $(48, \infty)$ hours $\rightarrow$ 0%.

Requirements to finish the homework smoothly: basic knowledge of probability and linear algebra, familiarity with Python programming, and correct understanding of the teaching content.

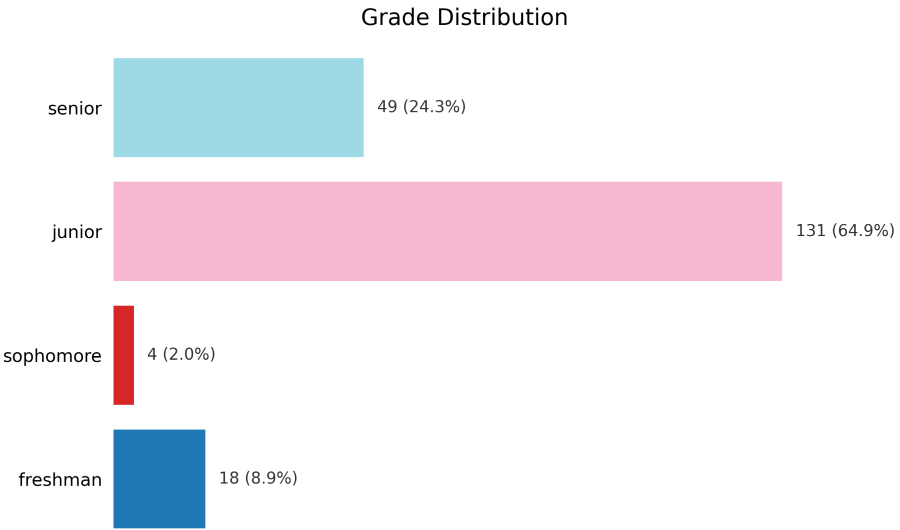
Plagiarism: Zero mark is given for the whole assignment (including written and programming) in the first plagiarism case. Students will **FAIL** the whole course for repeated plagiarism. **Note**: if there are heavy overlaps between two answers, then both will be identified as plagiarism (we don't have time to distinguish). Thus, discussions are encouraged, but you must finish the assignment by yourself, and don't share your answer with others.

Agenda with tutorials

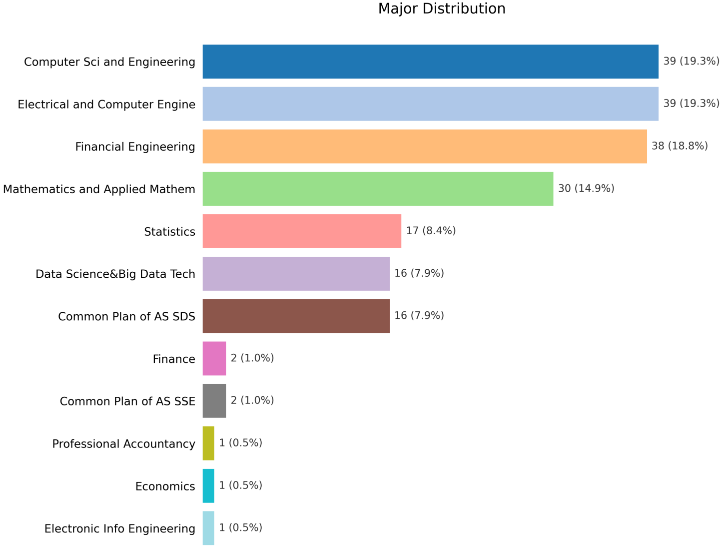
Week	Content	Homework	Tutorial
W1	Introduction		None
W2	Review of Probability & Linear Algebra etc.		python and sklearn
W3	Linear Regression I		Linear regression
W4	Linear Regression II & Logistic Regression	W1+P1 release	Logistic regression
W5	Support Vector Machines		SVM coding
W6	Decision Tree and Random Forest		W1 + P1
W7	Neural Networks I (MLP & CNN)		Pytorch coding
W8	Neural Networks II (RNN & Transformer)	W2+P2+Project release	Pytorch coding
W9	Over-fitting, Bias-Variance Trade-off		Demo
W10	Performance Evaluation		W2 + P2
W11	Intro. to Unsupervised Learning, K-Means		K-means coding
W12	Mixture Models, EM algorithm	W3 release	GMM coding
W13	PCA		PCA coding
W14	Review		W3

Students' background

- The students come from **diverse years and majors**, thus, may have different expectations and experiences.



Students' background



Course difficulty

- Machine learning is a **complex** and **evolving** discipline. Our course will cover the **fundamental concepts and algorithms** in machine learning.
- **Fundamental $\neq$ SIMPLE**. Machine learning is built on several other foundation subjects, including **probability, linear algebra, optimization, computer science**, *etc.*
- If you find it **difficult** to understand the teaching content, please don't hesitate to tell me or TAs, and we will help you overcome the challenges. If you feel the content is **too easy**, no worry, we can provide further reading materials on advanced and emerging topics (*e.g.*, deep learning, adversarial machine learning). Wish each of you an enjoyable learning experience!

Communications

Smooth communications between you and us are very important to build a successful course. If any questions/difficulties/suggestions, you are welcome to

- Talk to me directly, after class or during office hours
- Email or talk to your TAs, questions will be collected and sent to me weekly for responses
- Questions about programming are handled by TAs
- You may also send me emails if you need my help

Definition of machine learning

- Arthur Samuel: "the field of study that gives computers the ability to learn without being explicitly programmed."
- Tom Mitchell: "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."

Definition of machine learning

Tom Mitchell: "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."

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

- Classifying emails as spam or not spam.
- Watching you label emails as spam or not spam.
- The number (or fraction) of emails correctly classified as spam/not spam.
- None of the above – this is not a machine learning problem.

ML is a branch of artificial intelligence

- Artificial intelligence (AI) is intelligence demonstrated by **machines** (*e.g.*, computer, robots), unlike the natural intelligence displayed by humans and animals, which involves consciousness and emotionality.
- AI covers many topics, such as machine learning (ML), computer vision (CV), natural language processing (NLP), and speech processing.

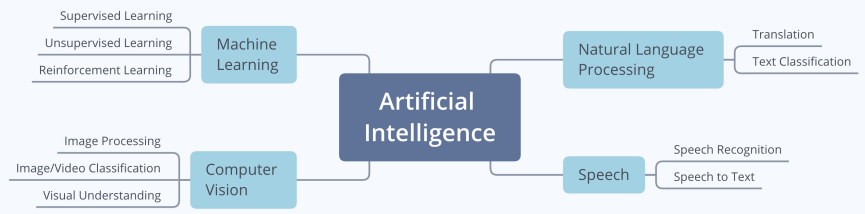


Figure: Machine learning is one of the most important branches of artificial intelligence.

Connection with other disciplines

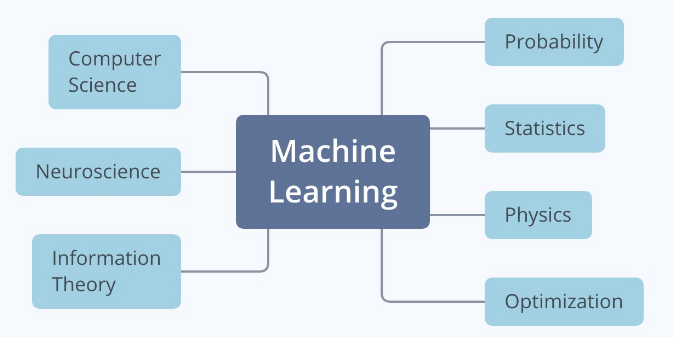


Figure: Machine learning is interdisciplinary.

Applications of machine learning

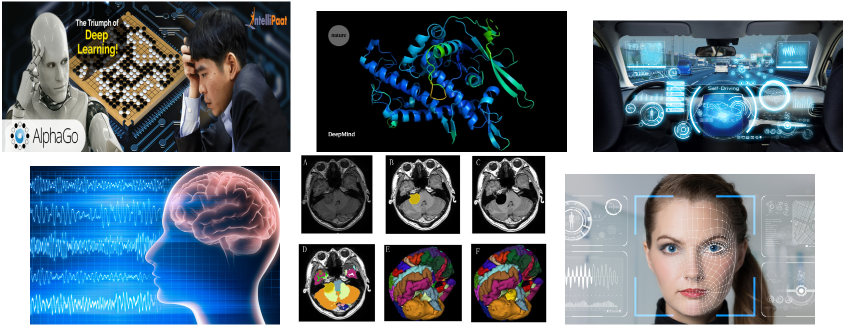


Figure: Machine learning has been widely used in many mission-critical tasks, such as Game (AlphaGo), protein structure prediction (AlphaFold2), EEG signal processing, medical image diagnosis, face recognition, etc.

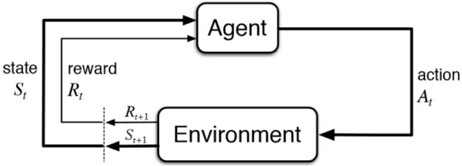
Two basic paradigms of ML

Given some data $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$, you can do

- **Supervised learning**: you are also provided some human-labeled outputs $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots$, and your task is to learn a mapping function from one input data $\boldsymbol{x}_i$ to one output $\boldsymbol{y}_i$. Learning from teacher
- **Unsupervised learning**: your task is to build/learn a good model of $\boldsymbol{x}$, such that some characteristics of the data could be revealed, such as clustering, dimensionality reduction, *etc.* Learning by oneself

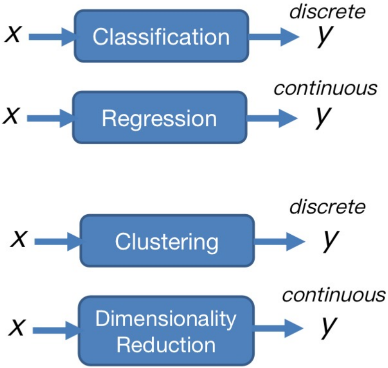
Other learning paradigms

- **Reinforcement learning**: you can make some actions a_i to change the data $\boldsymbol{x}_i$ (the state of the environment), and you will receive some rewards/punishments r_i. Gradually, you will automatically learn to make suitable actions for different data to get more rewards. It is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. Learning from rewards/punishments



In this course, we will focus on supervised learning and unsupervised learning.

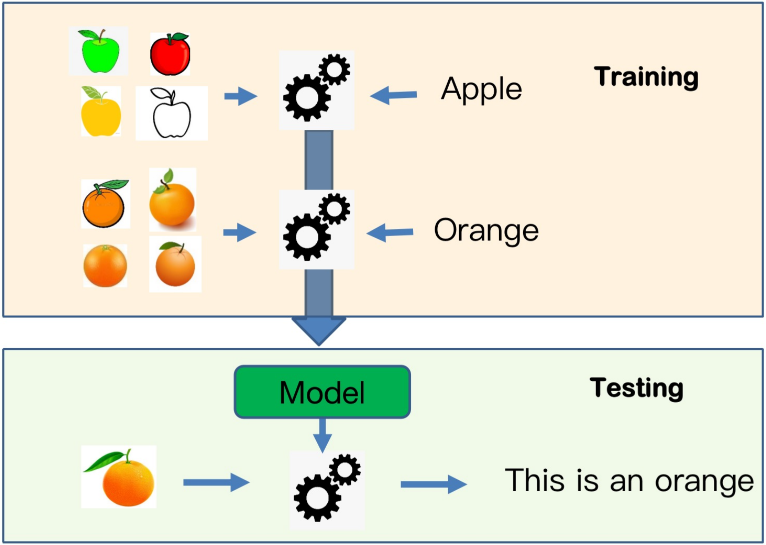
Supervised vs. unsupervised learning



	Supervised Learning (with labels)	Unsupervised Learning (without labels)
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality Reduction

Refer to:
`https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d`

Supervised learning



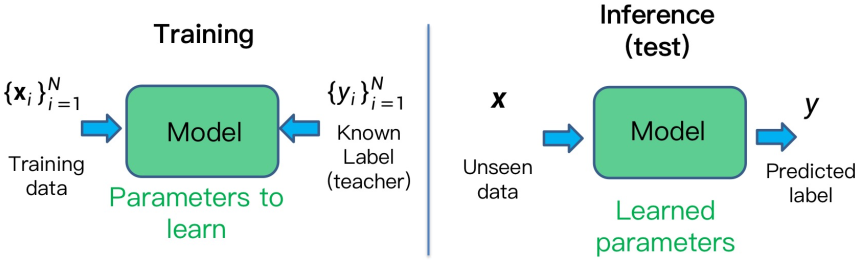
Supervised learning

In **supervised learning**, the **dataset** is the collection of labeled examples, denoted as $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$,

$$\boldsymbol{x}_i = [x_i^{(1)}, \ldots, x_i^{(j)}, \ldots, x_i^{(D)}]^\top,\ i = 1, \ldots, N$$

- Each element $\boldsymbol{x}_i$ is called a feature vector: it is a vector in which each dimension $j = 1, \ldots, D$ contains a value that describes the example somehow.
- The label y_i can be either an element belonging to a finite set of classes $\{1, 2, \ldots, C\}$, or a real number.

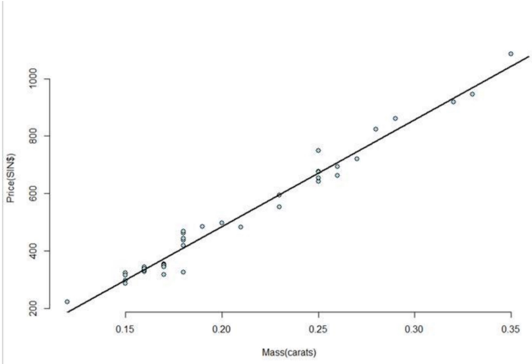
How supervised learning works



General procedure of supervised learning:

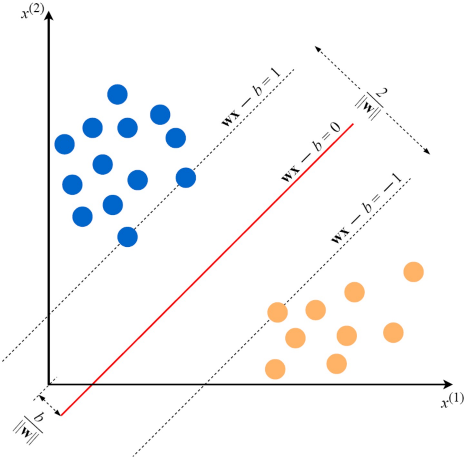
- **Data collection**: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$
- **Training**: conducting model training on the training data to learn the model parameters
- **Inference (test)**: Using the trained model to predict the output of unseen data $\boldsymbol{x}$

Regression example



- **Task**: To predict the price of diamond as a function of mass
- **Performance**: Accuracy of predicted values
- **Experience**: Historical data

Classification example



- **Task**: To classify the input data into two categories
- **Performance**: Classification accuracy
- **Experience**: Historical data

Regression vs. classification

Suppose that you are running a company, and you want to develop learning algorithms to address the following two problems.

Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.

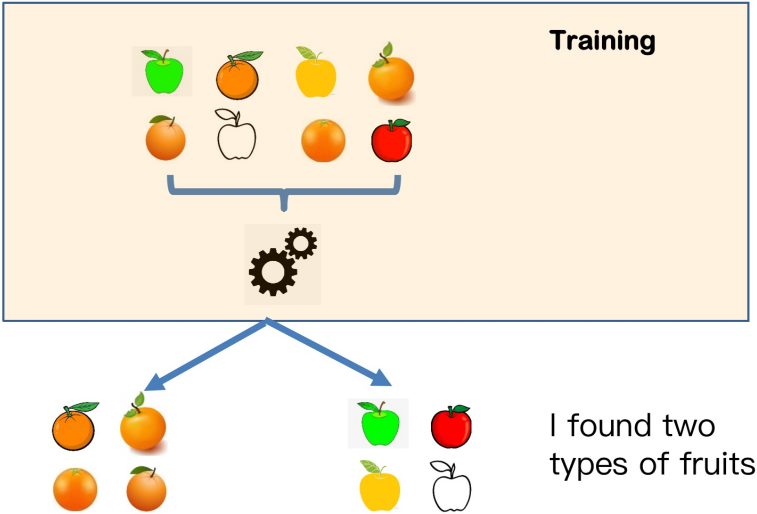
Problem 2: You'd like a program to examine individual customer accounts, and for each account, decide if it has been hacked or compromised.

Should you treat these as classification or regression problems?

- Treat both as classification problems.
- Treat problem 1 as classification, and problem 2 as regression.
- Treat problem 1 as regression, and problem 2 as classification.
- Treat both as regression problems.

Regression with continuous output vs. Classification with finite and discrete or categorical outputs

Unsupervised learning

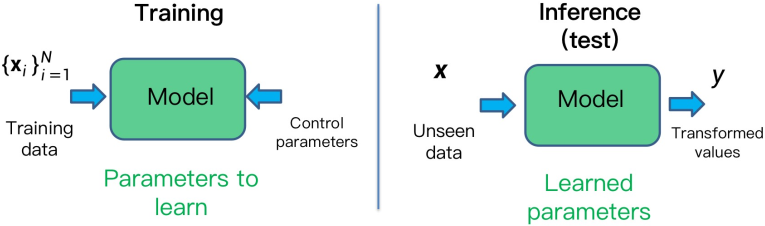


I found two types of fruits!

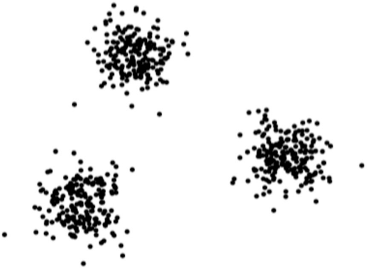
Unsupervised learning

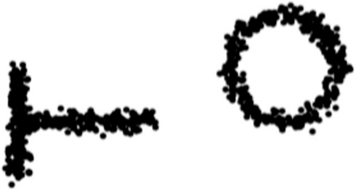
- In **unsupervised learning**, the **dataset** is a collection of unlabeled examples, *i.e.*, $\{\boldsymbol{x}_i\}_{i=1}^N$.
- Again, $\boldsymbol{x}$ is a feature vector, and the goal of an unsupervised learning algorithm is to create a **model** that takes a feature vector $\boldsymbol{x}$ as input and either transforms it into another vector or into a value that can be used to solve a practical problem.
- Its **main task** is to **analyze the structure of data** for future inference.

How unsupervised learning works



Clustering example

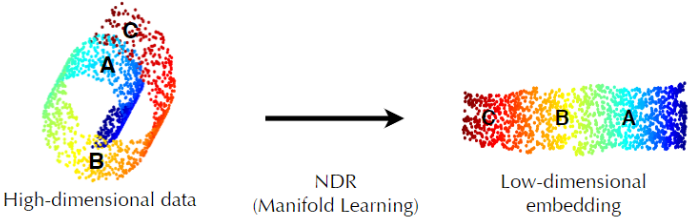




- Task: to partition a set of unlabeled points to clusters.
- Performance (for test data):
    - points within the same cluster are close to each other
    - points from different clusters are far from each other
    - the clusters have an appropriate coverage of all data
- Experience: available data

The key question is how to define and measure close or far, depending on what features are adopted (*e.g.*, global Euclidean distance, local distance, shape ...)

Dimensionality reduction example



The purposes of dimensionality reduction:

- **Data simplification**: non-linear $\rightarrow$ linear
- **Data visualization**: high dimensional $\rightarrow$ low-dimensional
- **Reduce noise**: some dimensions of the input data may be noises
- **Variable selection for prediction**: learn a sparse model, if there are redundancies among different dimensions

Basic concepts in supervised learning

- **Data for supervised learning**:
  - **Training set**: $D_{train} = \{(\boldsymbol{x}_i, y_i)\}_i^n$, with $\boldsymbol{x}_i \in \mathcal{X}$ being the feature presentation (could be scalar, image/video, text sequence, graph, cloud point, *etc.*), $y_i \in \mathcal{Y}$ being the supervision/ground-truth value (could be a discrete label, continuous value, vector, sequence, tensor, *etc.*).
  - **Testing set**: $D_{test} = \{(\boldsymbol{x}_i, y_i)\}_i^m$ is used to evaluate the performance of the trained model.
- **Data for unsupervised learning**
  - **Training set**: $D_{train} = \{\boldsymbol{x}_i\}_i^n$, with $\boldsymbol{x}_i \in \mathcal{X}$ being the feature presentation (could be scalar, image/video, text sequence, graph, cloud point, *etc.*).
  - **Testing set**: $D_{test} = \{\boldsymbol{x}_i\}_i^m$ is used to evaluate the performance of the trained model.
- **Independent and identically distributed (i.i.d.) assumption**: In standard machine learning, it assumes that all samples are observations (or realizations) of independent and identical random variables, and training and testing sets follow the same distribution.

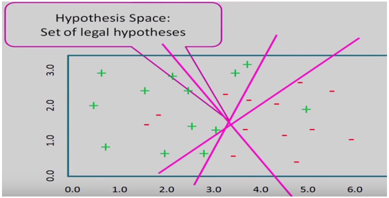
Basic concepts in supervised learning

- **Target function** $t : \mathcal{X} \to \mathcal{Y}$: the ground-truth mapping function from the input to the output behind the training/testing data. It is unknown, and our goal is to find it.
- **Hypothesis** h: A hypothesis is a candidate function that describes the unknown target function, for example

$$h(\boldsymbol{x}) = 1 \times x_1 + 2 \times x_2 = [1, 2] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- **Hypothesis space** $\mathcal{H}$: Hypothesis space is the set of all the possible legal hypotheses, for example

$$\mathcal{H}_{\boldsymbol{w}}(\boldsymbol{x}) = w_1 \times x_1 + w_2 \times x_2 = \boldsymbol{w}^\top \boldsymbol{x}$$



Basic concepts in supervised learning

- **Cost function** is a measure of how good/bad the hypothesis is in terms of its ability to estimate the relationship between $\boldsymbol{x}$ and y, such as the square loss $(h(\boldsymbol{x}) - y)^2$.
- **Objective function** is the function that we want to optimize (minimize, maximize, or minimax). When we are minimizing it, we may also call it the cost function, loss function, or error function. In this course, we use these terms interchangeably.
- **Training/learning** is the process of searching a good hypothesis h in the hypothesis space $\mathcal{H}$, through optimizing the objective function on a training set D_{train} using the optimization method, such as

$$h^* = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{(\boldsymbol{x}_i, y_i) \in D_{train}} (h(\boldsymbol{x}_i) - y_i)^2$$

- **Testing/evaluation**: evaluating the performance of the learned model h^* on a test set D_{test}, such as

$$\frac{1}{m} \sum_{(\boldsymbol{x}_i, y_i) \in D_{test}} (h^*(\boldsymbol{x}_i) - y_i)^2$$

A general machine learning workflow

1. **Collecting data**: Be it the raw data from Excel, access, text files, etc., this step (gathering past data) forms the foundation of future learning. The better the variety, density, and volume of relevant data, the better the learning prospects for the machine become.
2. **Preprocessing data**: One needs to spend time determining the quality of data and then taking steps for fixing issues such as missing data and treatment of outliers.
3. **Determining the hypothesis space, objective function, optimization method**.
4. **Training**: learning the parameters of the hypothesis function through optimizing the objective function.
5. **Testing**: evaluating the learned model on testing data.
6. **Improving the performance**: This step might involve choosing a different model altogether or introducing more variables to augment the efficiency. That's why a significant amount of time needs to be spent on data collection and preparation.

Review of this lecture

- Machine learning 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."
- Three main paradigms of machine learning:
  - **Supervised learning**: learning from teacher
  - **Unsupervised learning**: learning by oneself
  - Reinforcement learning: learning from rewards/punishments
  - Many other learning paradigms ...
- Supervised learning: regression and classification.
- Unsupervised learning: clustering and dimensionality reduction.
- Some basic concepts

Further reading

Other learning paradigms:

- **Reinforcement learning**: an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. (Link)
- **Semi-supervised learning**: labeled data + unlabeled data (Link)
- **Ensemble learning**: learning with multiple ML models, and the final prediction is obtained by combining these models. Improving the performance of individual models 三个臭皮匠，顶个诸葛亮 (Link)
- **Transfer learning**: source domain data + target domain data, useful especially when the target data is insufficient (Link)
- **Federated learning**: learning the model at local servers using local data, then uploading locally updated parameters to the central server to obtain the unified parameters. Protecting users' privacy (Link)
- **Machine unlearning**: erasing the effect of some particular training samples from a trained model. Protecting users' privacy (Link)

Practice: Scikit-Learn

https://scikit-learn.org/stable/

scikit learn | Install | User Guide | API | Examples | Community | More | Go

scikit-learn

Machine Learning in Python

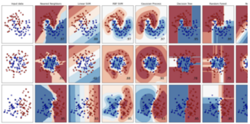
Getting Started | Release Highlights for 1.4 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

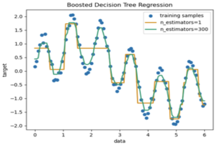


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

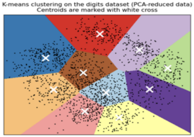


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

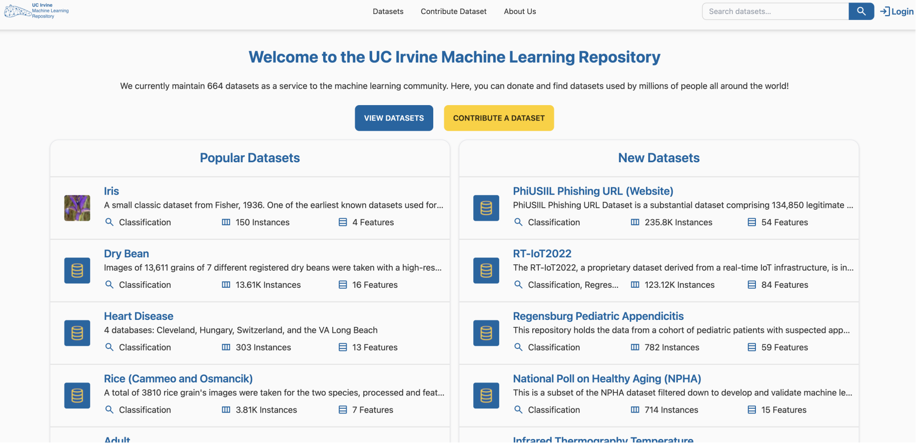
Dimensionality reduction

Model selection

Preprocessing

Practice: UCI Data Repository

https://archive.ics.uci.edu/



Practice: Kaggle

https://www.kaggle.com/

kaggle Competitions Datasets Models Code Discussions Courses ••• Search Sign In Register

Level up with the largest AI & ML community

Join over 17M+ machine learners to share, stress test, and stay up-to-date on all the latest ML techniques and technologies. Discover a huge repository of community-published models, data & code for your next project.

Register with Google Register with Email

Who's on Kaggle?

Learners
Dive into Kaggle courses, competitions & forums.

Developers
Leverage Kaggle's models, notebooks & datasets.

Researchers
Advance ML with our pre-trained model hub & competitions.