Artificial Intelligence for Medicine II

Spring 2025

Lecture 3: Supervised Learning Basic Concepts

(Many slides adapted from Bing Liu, Han, Kamber & Pei; Tan, Steinbach, Kumar and the web)

Supervised learning

- Supervised learning is a type of machine learning where an algorithm is trained on labeled data. This means that the training data includes both the input data and the corresponding correct output.
- The goal is for the algorithm to learn a mapping from inputs to outputs so that it can make accurate predictions on new, unseen data.
- The two main types of supervised learning:
  - **Classification and**
  - **Regression**

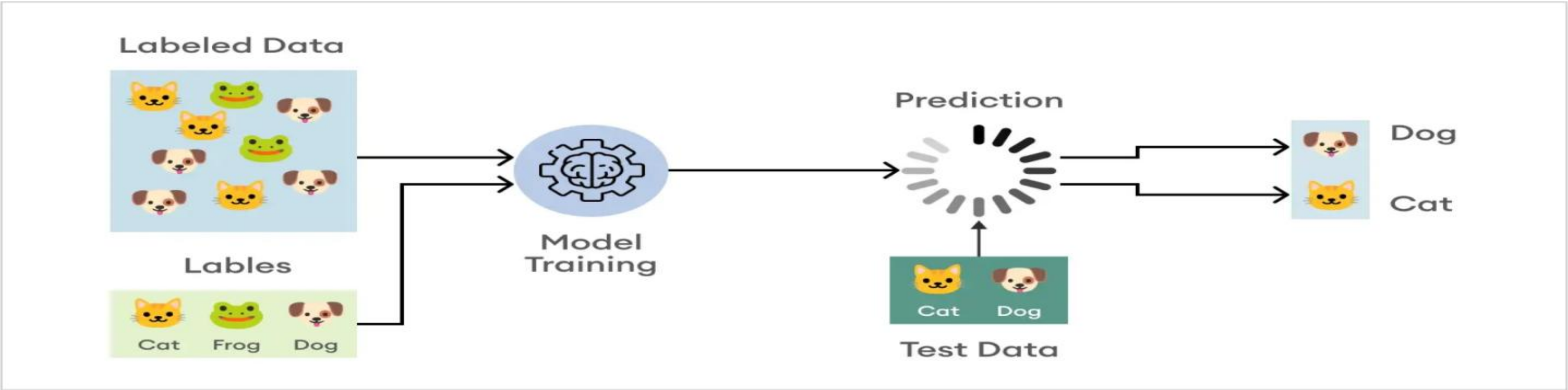
Supervised vs. unsupervised Learning

- Supervised learning: learning from examples (labeled data).
  - Supervision: The data (observations, measurements, etc.) are labeled with pre-defined classes, which is
  - like a “teacher” gives us the classes (supervision).
- Unsupervised learning (clustering)
  - Class labels of the data are not given or unknown
  - **Goal**: Given a set of data, the task is to establish the existence of classes or clusters in the data

Classification

- Classification is used when the **output variable is a categorical value**. The goal is to assign input data to one of several predefined categories.

- **Examples of classification tasks:**
  - Email spam detection (spam or not spam).
  - Image recognition (identifying objects in images, like cats or dogs).
  - Medical diagnosis (classifying diseases based on symptoms).
- **Common classification algorithms:**
  - **Logistic Regression**: Despite its name, it's used for binary classification tasks.
  - **Decision Trees**: Splits the data into branches to make decisions based on input features.
  - **Random Forest**: An ensemble method that uses multiple decision trees to improve accuracy.
  - **Support Vector Machines (SVM)**: Finds the optimal boundary between classes.
  - **Neural Networks**: Can handle complex classification tasks with multiple layers.

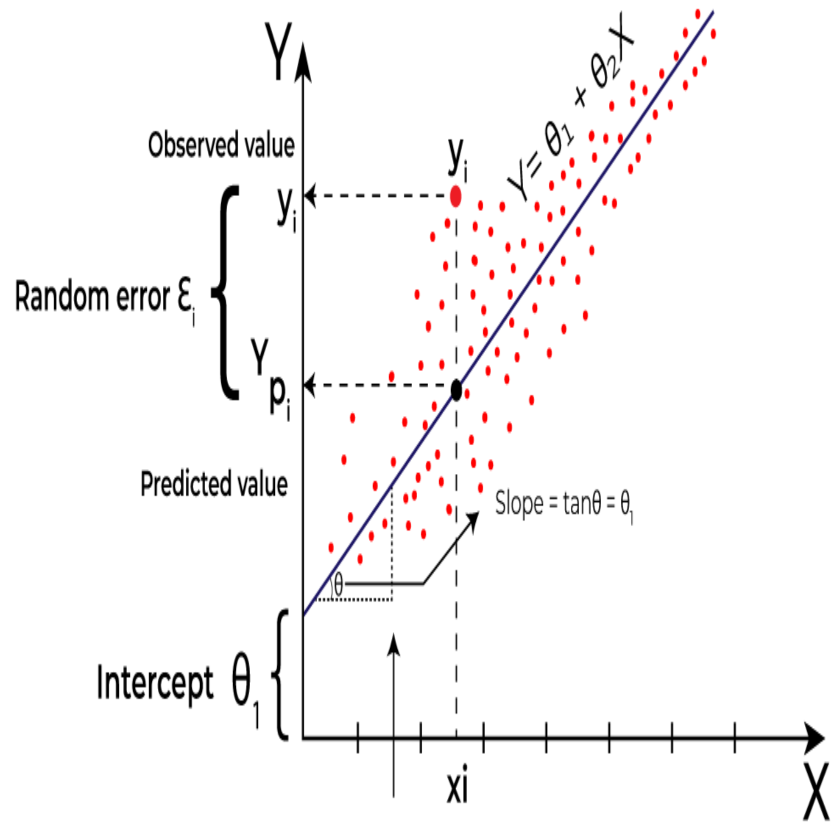


Regression

- Regression is used when the output variable is a continuous value. The goal is to predict a numerical value based on input features.

- **Examples of regression tasks:**
  - Predicting house prices based on features like size, location, and number of bedrooms.
  - Forecasting stock prices based on historical data.
  - Estimating the amount of rainfall based on weather conditions.

- **Common regression algorithms:**
  - **Linear Regression**: Models the relationship between the input variables and the output by fitting a linear equation to the observed data.
  - **Polynomial Regression**: Extends linear regression by fitting a polynomial equation to the data.
  - **Support Vector Regression (SVR)**: Uses support vector machines to perform regression tasks.
  - **Neural Networks**: Can model complex relationships between inputs and outputs.



An application of supervised learning

- **Endless applications of supervised learning**.
- An emergency room in a hospital measures 17 variables (e.g., blood pressure, heart rate, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit (ICU).
  - Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

Another application

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: Decide whether an application should approved, i.e., **classify** applications into two categories, approved and not approved.

Supervised machine learning

- We humans learn from past experiences.
- A computer does not "experience."
  - A computer system learns from data, which represents "past experiences" in an application domain.
- Our focus: learn a target function that can be used to predict the values (**labels**) of a discrete class attribute, e.g.,
  - high-risk or low risk and approved or not-approved.
- The task is commonly called: supervised learning, classification, or inductive learning.

The data and the goal

- Data: A set of data records (also called examples, instances, or cases) described by
  - *k* data attributes: A_1, A_2, … A_k.
  - One class attribute: a set of pre-defined class labels
  - In other words, each record/example is labelled with a class label.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future or test) instances/cases.

An example: data (loan application)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

An example: the learning task

Sub-tasks:

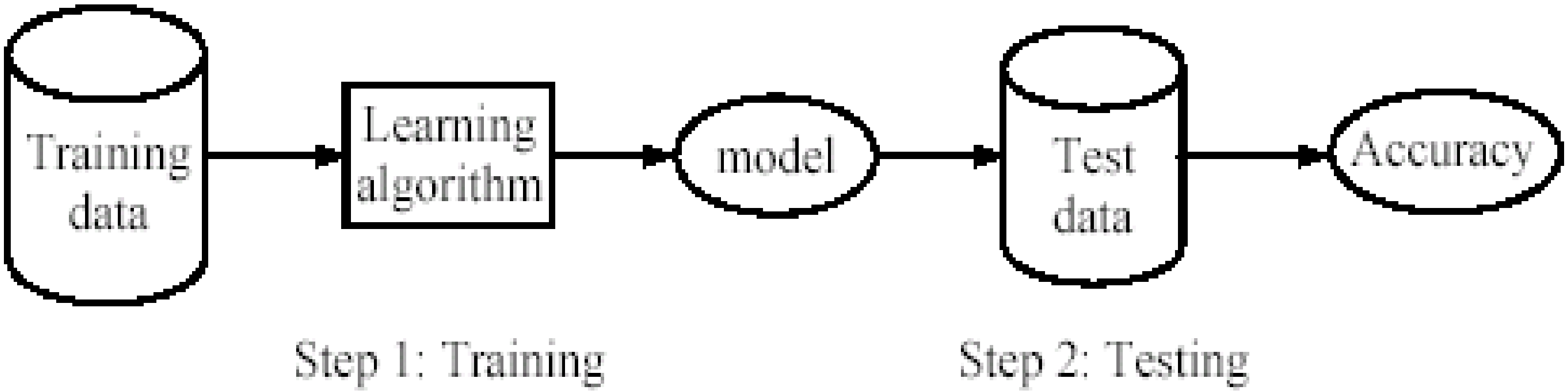
- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following applicant/case?

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

Supervised learning process: two steps

- Learning or training: Learn a model using the training data (with labels)
- Testing: Test the model using unseen test data (without labels) to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



What do we mean by learning?

- Given
  - a data set *D*,
  - a task *T*, and
  - a performance measure *M*,
- A computer system is said to **learn** from *D* to perform the task *T*,
  - if after learning, the system's performance on *T* improves as measured by *M*.
  - In other words, the learned model helps the system to perform *T* better as compared to without learning.

An example

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: accuracy.
- No learning: classify all future applications (test data) to the majority class (i.e., Yes):

  Pr(Yes) = 9/15 = 60%.

  – Expected accuracy = 60%.
- Can we do better (> 60%) with learning?

Fundamental assumption of learning

- **Assumption**: The data is *independent and identically distributed* (*i.i.d*).
- Given the data D = {**X**, y} with N examples ($\mathbf{X}_i$, y_i), and a joint distribution $\mathbb{P}(\mathbf{X}, y)$, mathematically i.i.d means

$$(\mathbf{X}_i, y_i) \sim \mathbb{P}(\mathbf{X}, y), \forall i = 1, \ldots, N$$
$$(\mathbf{X}_i, y_i) \text{ independent of } (\mathbf{X}_j, y_j), \forall i \neq j \in \{1, \ldots, N\}$$

Fundamental assumption of learning

- **The data** is split into training and test data.
  - The distribution of training examples is identical to the distribution of test examples (including future unseen examples).
  - To achieve good accuracy on the test data,
    - training examples must be sufficiently representative of the test data.
- In practice, this assumption is often violated to certain degree.
  - Strong violations will clearly result in poor classification accuracy.