Introduction to Artificial Intelligence

Machine Learning

greenwich.edu.vn

UNIVERSITY OF GREENWICH

Alliance with FPT Education

Machine Learning
Introduction

- An **agent** is **learning** if it improves its performance after making observations about the world.
- When the agent is a **computer**, we call it **machine learning.**
- Machine learning is:
  - a computer **software**
  - has been built with a **model** based on the **data**,
  - and uses the model as a **hypothesis** about the world to solve problems.

Machine Learning
Basic concepts

- **Model:** is a simplified representation of a system. Models may be atomic, factored and relational and can be based on logic or probability.
- **Training Data:** contains the knowledge that the learning algorithm extracts and learns.
- **Testing Data:** used to test the generalization capability of the learning algorithm on unknown data.
- **Loss function:** is a function evaluate the error between hypothesis and the real output value.
- **Weight space:** The space defined by all possible settings of the weights or model's parameters.
- **Training**: involves **modifying a model’s parameters (weights)** to **minimize the loss function** on the training set.

Forms of Learning

- Any component of an agent program can be improved by machine learning. The improvements, and the techniques used to make them, depend on these factors:
  - Which **component** is to be improved.
  - What **prior knowledge** the agent has, which influences the model it builds.
  - What **data** and **feedback** on that data is available.

Forms of Learning

- There are 3 types of feedback that determine three main types of learning:
  - **Supervised learning:** the agent observes input-output pairs and learns a function that maps from input to output. An output in this case is called a **label**.
  - **Unsupervised learning:** the agent learns patterns in the input without any explicit feedback.
  - **Reinforcement learning:** the agent learns from a series of reinforcements feedback: rewards and punishments.

Forms of Learning

- Some main types of model based on its output:
  - **Classification:** when the output is one of a finite set of values (such as sunny/cloudy/rainy or true/false).
  - **Regression:** when the model's output is a number.
  - **Clustering:** the output that groups data points into clusters based on similarity.
  - **Dimensionality Reduction:** when the output creates a lower-dimensional representation of the data while preserving important information.
  - **Association Rule Learning:** discovers relationships between variables in large datasets.
  - **Others: generation, reinforcement.**

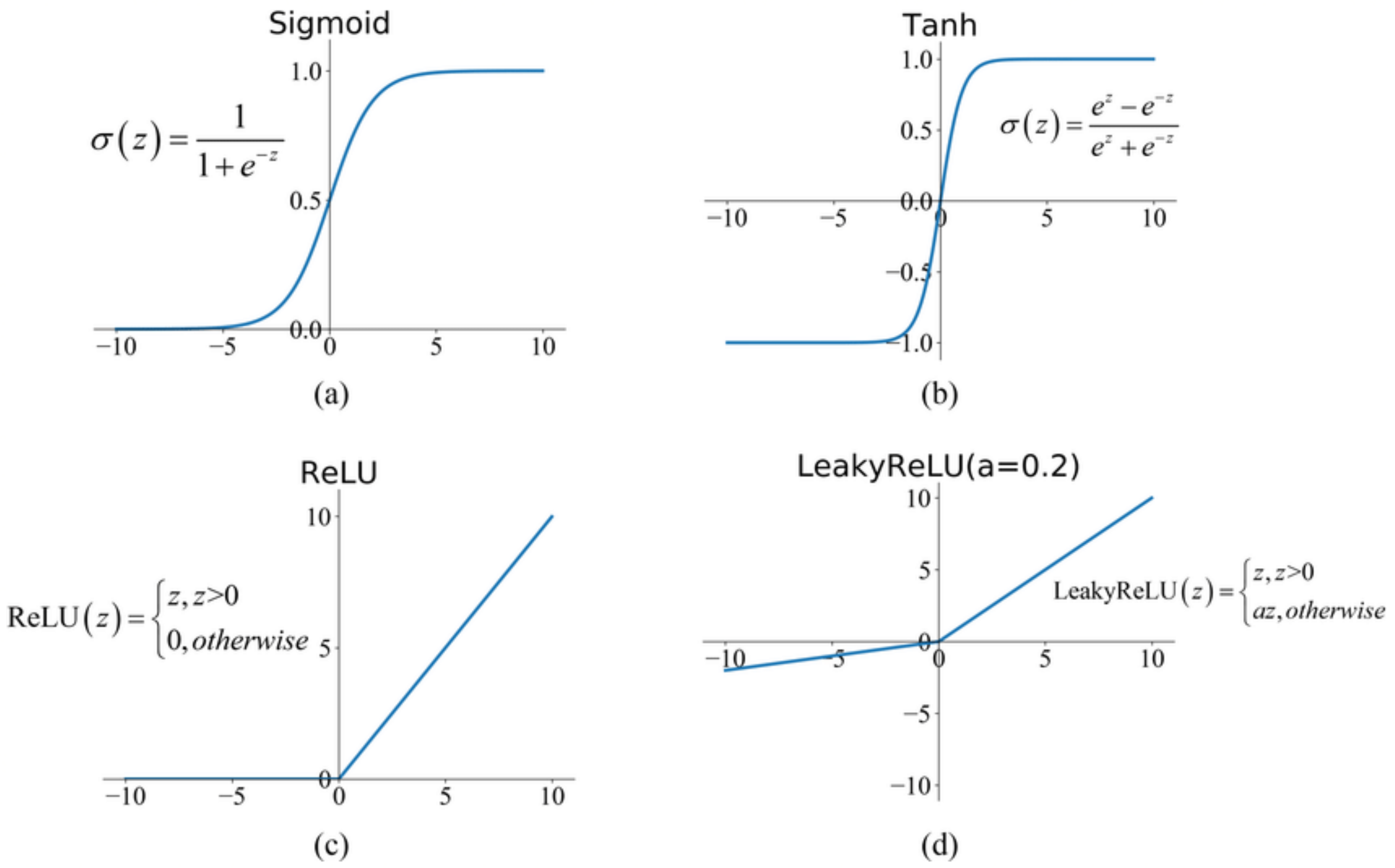
Key concepts of Machine Learning

LOSS function

- •Loss function: cost function or error function $L(f, \hat{f}) = \|f - \hat{f}\|_2^2$
- •Mean Square Errror $\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$
- •Mean Bias Error $MBE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}$
- •Mean Squared Logarithmic Error $MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(Y_i) - \log(\hat{Y}_i))^2$
- •And many others loss function

Activation function

Activation functions: This function determines the output.

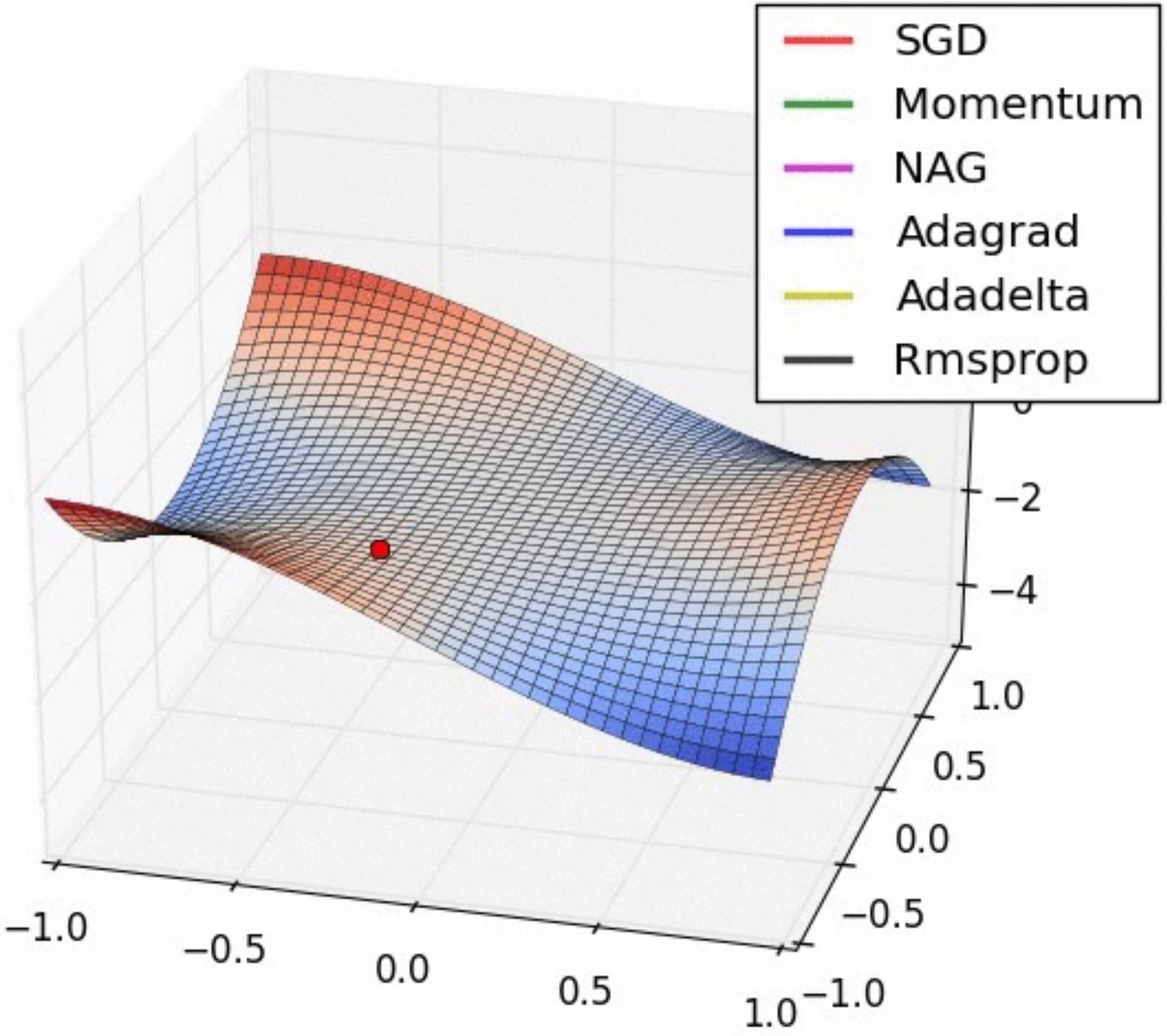


ML core – Optimization (Training)

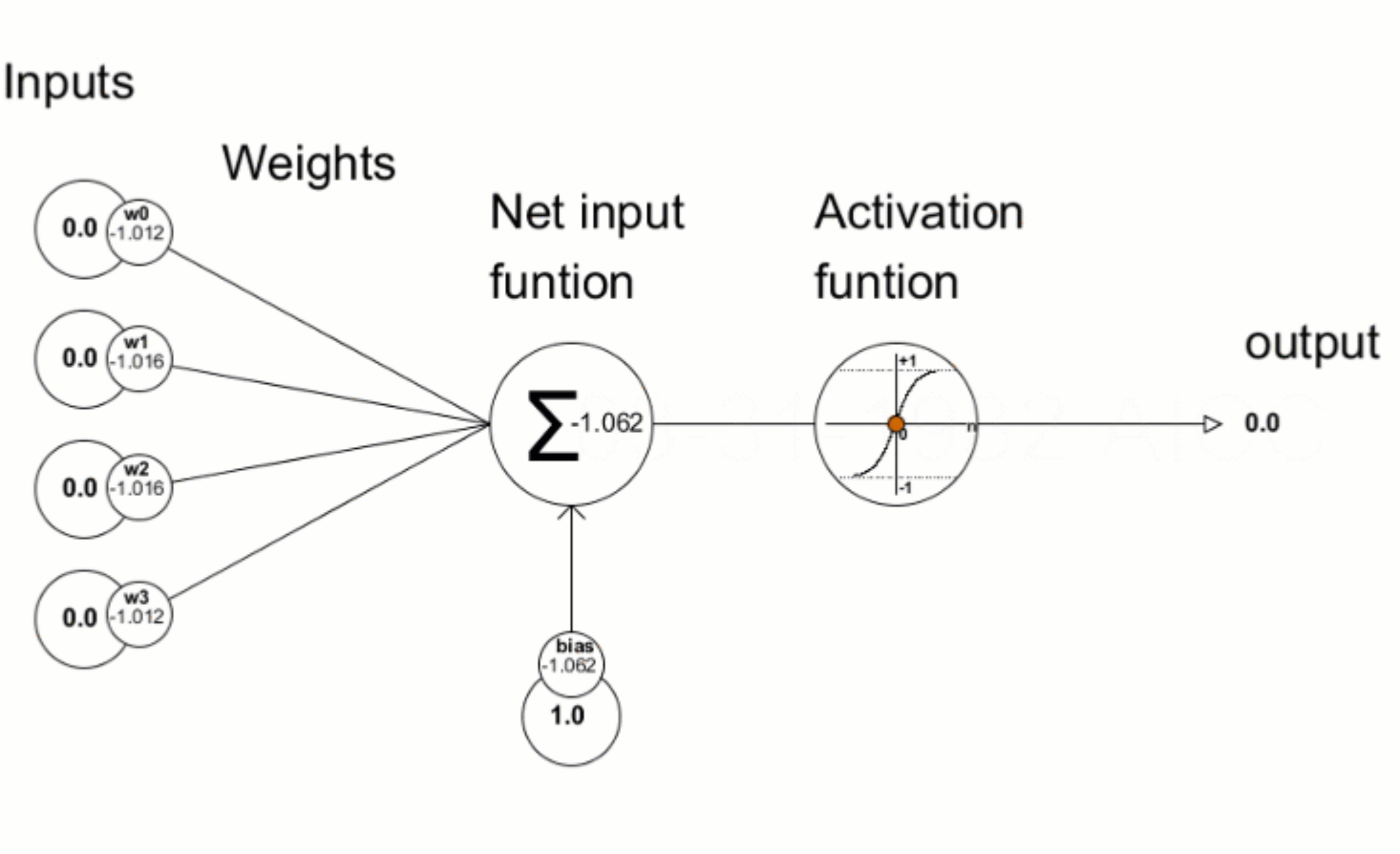
- **Optimization Algorithms** refer to a procedure for **finding the input parameters** or arguments to a function that result in the **minimum or maximum output** of the function.
- **Optimization in ML**: finding the parameters or arguments to get the **minimum of Loss Function**
- Some optimize algorithms:
  - ✓ Gradient Descent
  - ✓ Momentum
  - ✓ Adagrad
  - ✓ RMSProp
  - ✓ Adam
  - ✓ And many others …

Optimizer

- **Optimizer:** is an algorithm used to adjust the network's parameters (weights) to **minimize the loss function** on the training set.
- This process is **crucial for training** the neural network and enabling it to learn from data.



ML core – Optimization (Training)



Data preprocessing

- **Data preprocessing** involves transforming raw data into a format suitable for machine learning models.
- **Importance**: Real-world data is often incomplete, noisy, and inconsistent, requiring preprocessing for effective analysis.

Data preprocessing
Common techniques

- **Data Cleaning**: correcting errors and inconsistencies in the data.
- **Handling Missing Values**: addressing missing values through imputation or removal.
- **Data Normalization:**
  - ✓Scaling variables to a specific range to prevent variables with larger domains from dominating the analysis.
  - ✓Min-Max Normalization: Scales data linearly onto the interval.
  - ✓Standardization: Centers data around zero with a standard deviation of one.

- **Feature Engineering**: transforming categorical attributes into numerical format, such as one-hot encoding.
- **Feature Selection:** discards attributes that appear to be irrelevant.
- **Quantization**: forcing continuous valued input into fixed bins.

Data analysis

- **Data analysis** involves describing data with simple parameters to understand its characteristics.
- This is **especially important** for analyzing training data in machine learning.

Key Statistical Measures
Average Value

- The average value μi for each variable xi is defined as:

$$\mu_i := \frac{1}{N} \sum_{p=1}^{N} x_i^{(p)}$$

- Where:
  - ✓N is the total number of patients (data points)
  - ✓xi(p) is the value of variable xi

Key Statistical Measures
Standard deviation

- The **standard deviation** σi measures the average difference from the average value.

$$\sigma_i := \sqrt{\frac{1}{N-1}\sum_{p=1}^{N}(x_i^{(p)} - \mu_i)^2}$$

Key Statistical Measures
Covariance & Correlation Coefficience

- **Covariance** (σij) indicates how two variables change together.

$$\sigma_{ij} = \frac{1}{N-1}\sum_{p=1}^{N}(x_i^{(p)} - \mu_i)(x_j^{(p)} - \mu_j)$$

- The **correlation coefficient** (Kij) is a normalized covariance that measures the strength and direction of a linear relationship between two variables i and j .

$$K_{ij} = \frac{\sigma_{ij}}{\sigma_i \cdot \sigma_j}$$

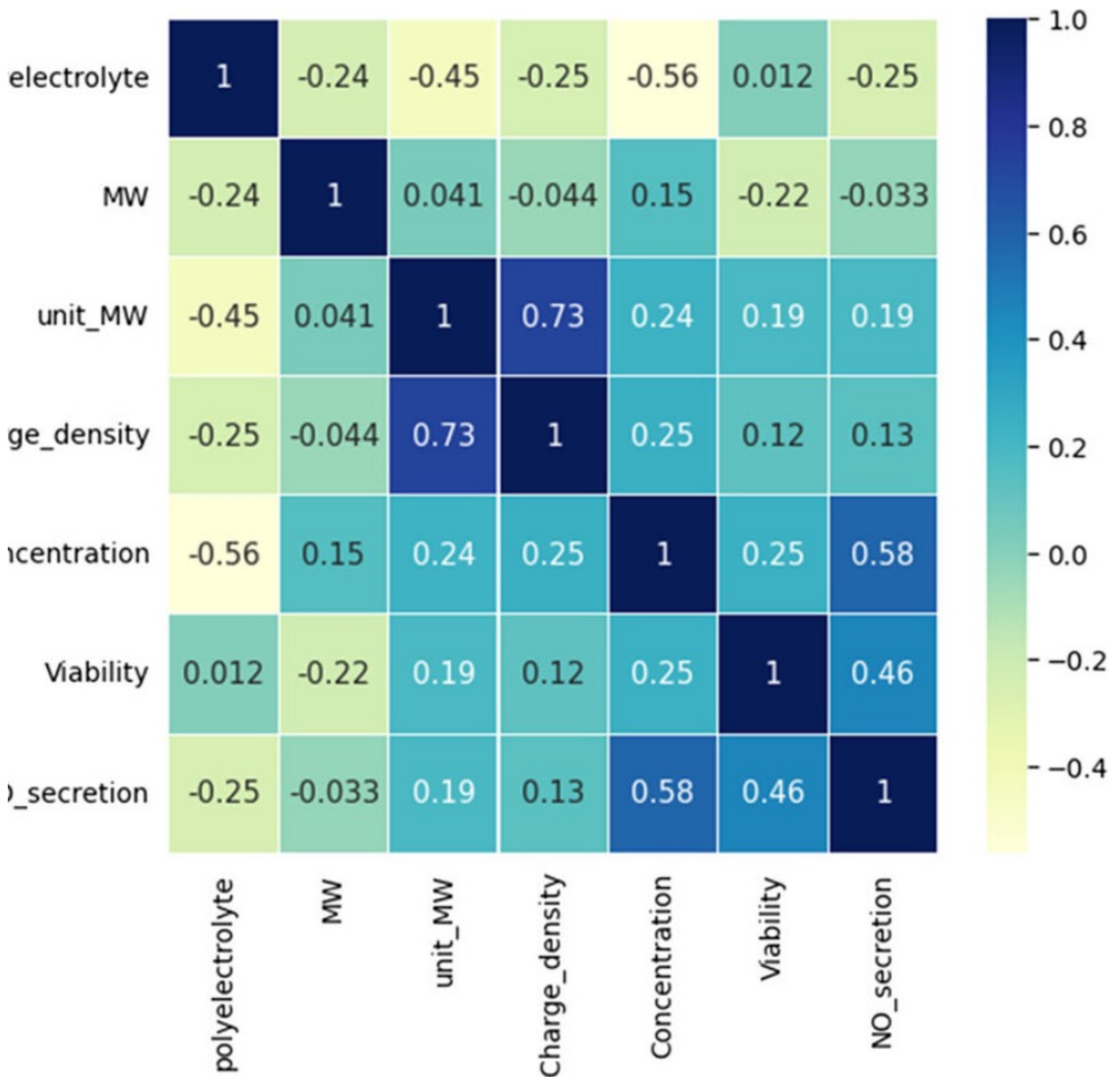
Correlation Matrix

- A correlation matrix contains the correlation coefficients between all pairs of variables.
- The matrix is symmetric, and all diagonal elements are 1.
- It can be visualized as a density plot to quickly identify strong or weak dependencies between variables

Correlation Matrix

Example of correlation matrix.

Student task: Try to interpret the correlation matrix?



Demo

- Demonstrate common techniques for data preprocessing and analysis based on **Week4-Data1.csv** dataset.

Supervised Learning
Definition

- More formally, the task of supervised learning is this:

  Given a training set of N example input–output pairs **training set**:

  $(x1, y1),(x2, y2),\ldots(xN, yN),$

  where each pair was generated by an unknown function $y = f(x)$,

  the goal is discovering a function h that approximates the true function f .

- The function h is called a hypothesis about the world.

Supervised Learning
Model Selection and Optimization

- The goal is to select a hypothesis *(h)* that will optimally fit future examples.
- Optimal Fit: The hypothesis that **minimizes the error** rate. The error rate of a hypothesis can be estimated by measuring its performance on a **test set** of examples.

Supervised Learning
Key features

- **Learning from labeled data:** Supervised learning relies on a training set of **input-output pairs**, where each **input is accompanied by a label** that represents the correct output.
- **Function approximation:** learn a hypothesis *h* that closely approximates the true function *f* mapping inputs x to outputs y.
- **Tasks:** classification and regression.
- **Stationarity assumption** Supervised learning relies on the stationarity assumption, which posits that future examples will be similar to those in the past.
- **Realizability:** A learning problem is considered realizable if the hypothesis space *H* actually contains the true function *f.*

The Learning Process

- **Data Collection**: Gather a dataset of input-output.
- **Model Selection**: Choose a hypothesis space H (e.g., polynomials, decision trees, neural networks).
- **Training** (Optimization): Find the best hypothesis h within H that minimizes the error on the training data.
- **Testing**: Evaluate the performance of h on a separate test set to estimate its error rate.
- **Goal**: Select a hypothesis that will optimally fit future examples

Example Problem: Restaurant Waiting

- Problem: Deciding whether to wait for a table at a restaurant.
- Output (y): Boolean variable willWait (true if we wait).
- Input (x): Vector of ten attribute values:
  - ✓Alternate: Suitable alternative restaurant nearby
  - ✓Bar: Comfortable bar area
  - ✓Fri/Sat: True on Fridays and Saturdays
  - ✓Hungry: Whether we are hungry
  - ✓Patrons: How many people are in the restaurant (None, Some, Full)
  - ✓Price: Restaurant's price range ($, $$, $$$)
  - ✓Raining: Whether it is raining outside
  - ✓Reservation: Whether we made a reservation
  - ✓Type: Kind of restaurant (French, Italian, Thai, burger)
  - ✓WaitEstimate: Host's wait estimate (0–10, 10–30, 30–60, >60 minutes)

Introduction to Linear Regression

- **Linear regression** is a parametric supervised learning model that predicts a continuous output variable based on a linear combination of input features.
- **Model**: Assumes a **linear relationship** between the input variables (x) and the output variable (y): y=w1.x+ w0
  - y is the predicted output.
  - xi are the input featur
  - wi are the weights (coefficients) for each feature.
  - w0 is the bias (intercept)
- Goal: find the best set of weights that minimize the difference between the predicted and actual values:

$$h_{\mathbf{w}}(x) = w_1 x + w_0$$

Introduction to Linear Regression

- **Multivariable Linear regression: o**ur hypothesis space is the set of functions of the form:

$$h_{\mathbf{w}}(\mathbf{x}_j) = w_0 + w_1 x_{j,1} + \cdots + w_n x_{j,n} = w_0 + \sum_i w_i x_{j,i}$$

Introduction to Linear Regression
How it works

- **Training Phase**: Given a set of training examples, the algorithm learns the weights (w) that best fit the data.
- **Loss Function:** typical the squared-error loss (L2), which measures the difference between the predicted and actual values.

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2$$

- **Optimization**: The weights are adjusted using optimization algorithms to minimize the *Loss* such as:
  - **Gradient Descent**: Iteratively updates the weights by moving in the direction of the negative gradient of the loss function.
  - **Normal Equations**: Solve directly for the weights that minimize the loss function.
- **Prediction Phase:** Once the model is trained, it can predict the output for new input values using the learned weights as the function *h(x)*.

Introduction to Linear Regression
Applications

- Economics: Predicting economic indicators such as GDP, inflation, and unemployment rates.
- Finance: Estimating stock prices, real estate values, and credit risk.
- Marketing: Predicting sales, customer behavior, and advertising effectiveness.
- Environmental Science: Modeling climate change, pollution levels, and resource depletion.
- Medicine: Determining a linear score in medical diagnosis, such as for appendicitis.
- And so much more ...

Introduction to Linear Regression
Practice

- Example: ***Advertising results prediction***
- Problem: Sales (in thousands of units) for a particular product as a function of advertising budgets (in thousands of dollars) for TV, radio, and newspaper media. The requirements are:
  - ✓Which media contribute to sales?
  - ✓Visualize the relationship between the features and the response using scatter plots.
  - ✓Find a model that given input budgets for TV, radio and newspaper predicts the output sales.
- Data: ***Week 4 - dataset_2_advertising.csv***
- Model: A simple linear regression model.
- See the ***Week 4 – Tutorial.docx*** for more detail.

Introduction to Logistic Regression

- A method for classification, where the output is one of a finite set of values.
- Uses a logistic function (sigmoid function) to classify data or also known as log-linear classifier.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}}$$

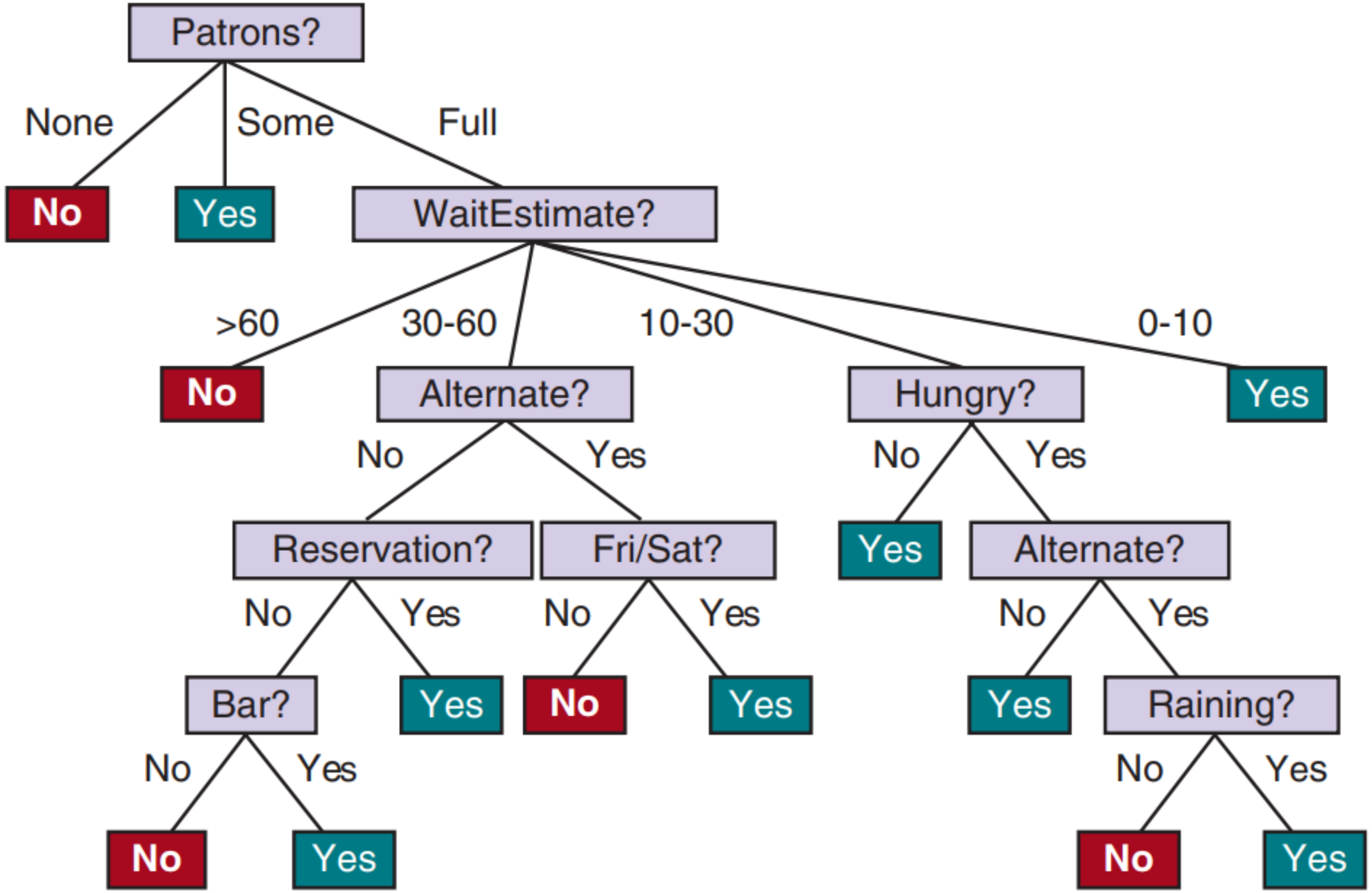
- Applies a soft threshold to the output of a linear function to classify data, called **activation function.**

Decision Tree

- A decision tree is a model that maps attribute values to a single output value or decision.
- A decision tree is a model that maps attribute values to a single output value or decision.
- It consists of a **sequence of tests** performed in a hierarchical structure.
- Each **internal node** represents a test on an attribute.
- **Branches** represent the outcomes of the test.
- **Leaf nodes** represent the final decision or classification.

Decision Tree
Example

- A decision tree for deciding whether to wait for a table.



Decision Tree
How it works

- The Decision Tree learning algorithm uses a greedy divide-and-conquer strategy.
- Chooses the attribute with the highest **IMPORTANCE**, using the notion of information gain, which is defined in terms of **entropy.**
- The entropy of a random variable *V* with values v_k having probability *P(* v_k *)* is defined as:

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$

- Choose the most important attribute to test first.
- Recursively solve smaller subproblems based on the test results.
- The algorithm generates a tree, and one may use pruning to combat overfitting.

Decision Tree
Applications

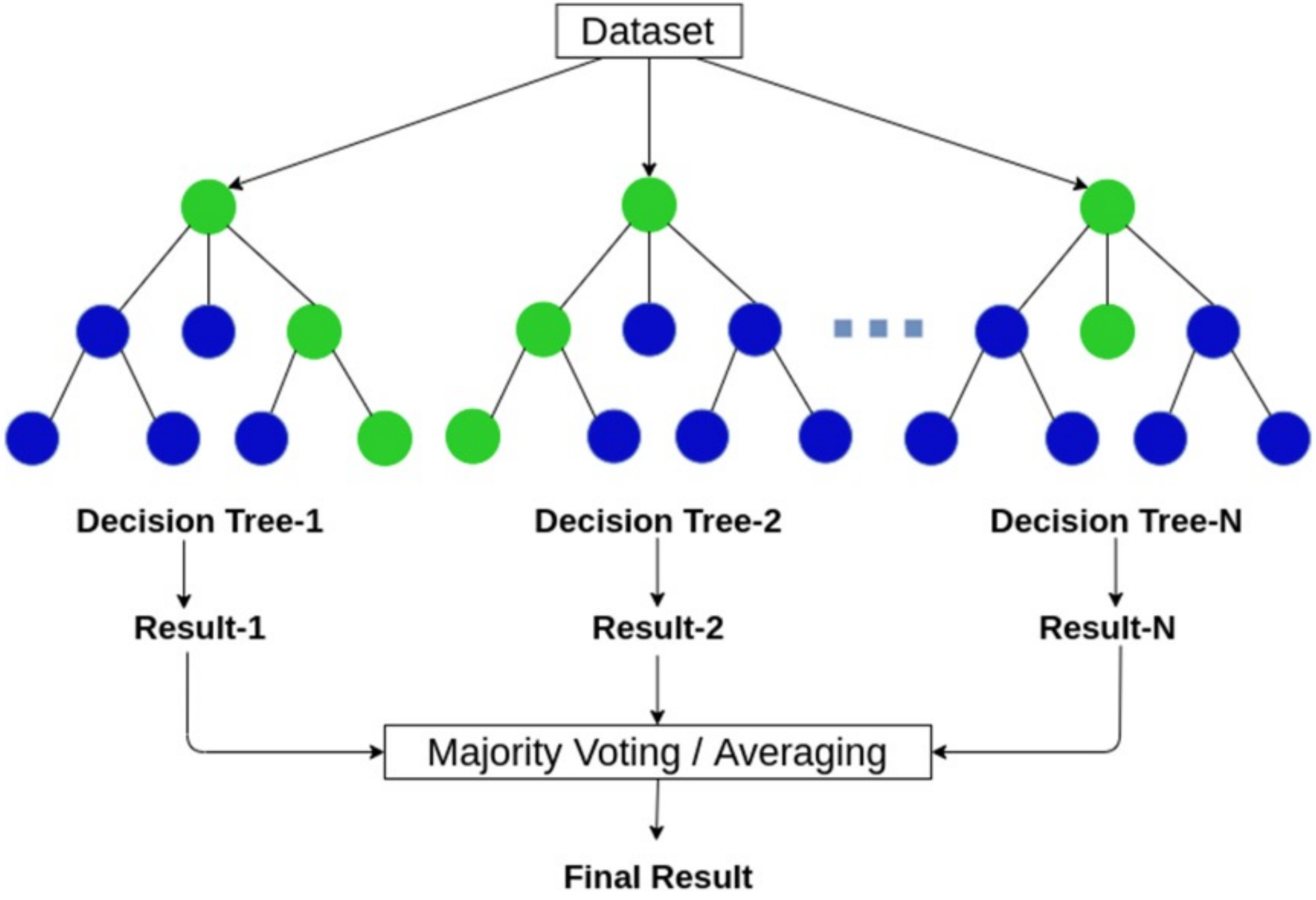
- **Classification**: Predict a category or class label.
- **Regression**: Predict a continuous numeric value.
- Decision trees can be used in:
  - ✓Medical diagnosis.
  - ✓Autonomous robots.
  - ✓Data mining, ...

Random Forest

- A **random forest** is an ensemble learning method that operates by constructing a ***multitude of decision trees*** during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Random forests correct for decision trees' habit of overfitting to their training set.
- Random forests are complex, unpruned models that are **resistant to overfitting**.
- Random forests can be used for both **classification** and **regression** tasks.

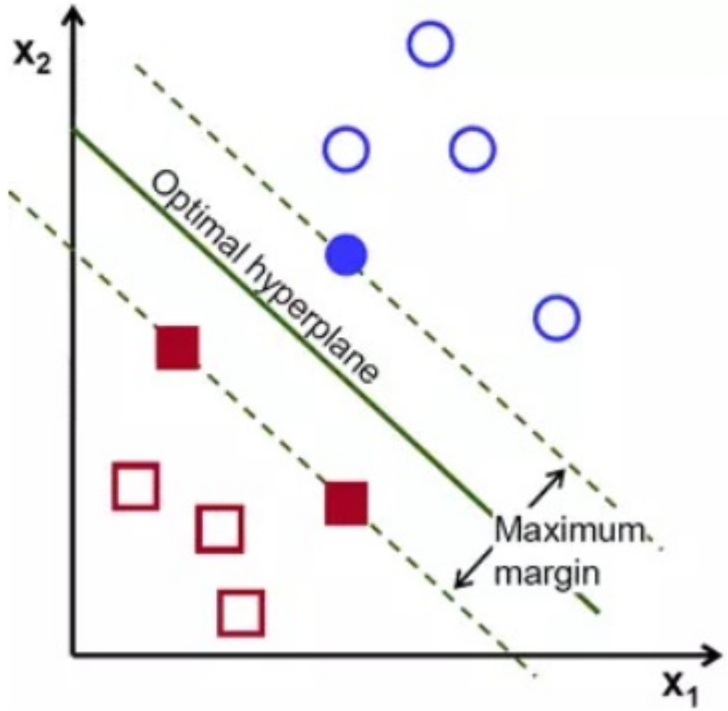
Random Forest

- Random forest



Support Vector Machines (SVM)

- A **Support Vector Machine (SVM)** is a model that constructs a **maximum margin separator**, a **decision boundary** with the largest possible distance to example points.
- SVMs aim to find a **hyperplane** that best divides the data into different classes with the maximum margin.
- SVMs are effective for **high-dimensional data**.
- SVMs can be used for both **classification** and **regression** problems.



Support Vector Machines (SVM)
How it works

- SVMs work by constructing a maximum margin separator, which serves as the decision boundary.
- Support Vectors are data points that are closest to the maximum margin separator. Finding the support vectors involves an efficient optimization algorithm.
- The **hyperplane** is defined as the set of points $\{x : w \cdot x + b = 0\}$. We could search the space of *w* and *b* to find the parameters that maximize the margin while correctly classifying all the examples.

Support Vector Machines (SVM)
Applications

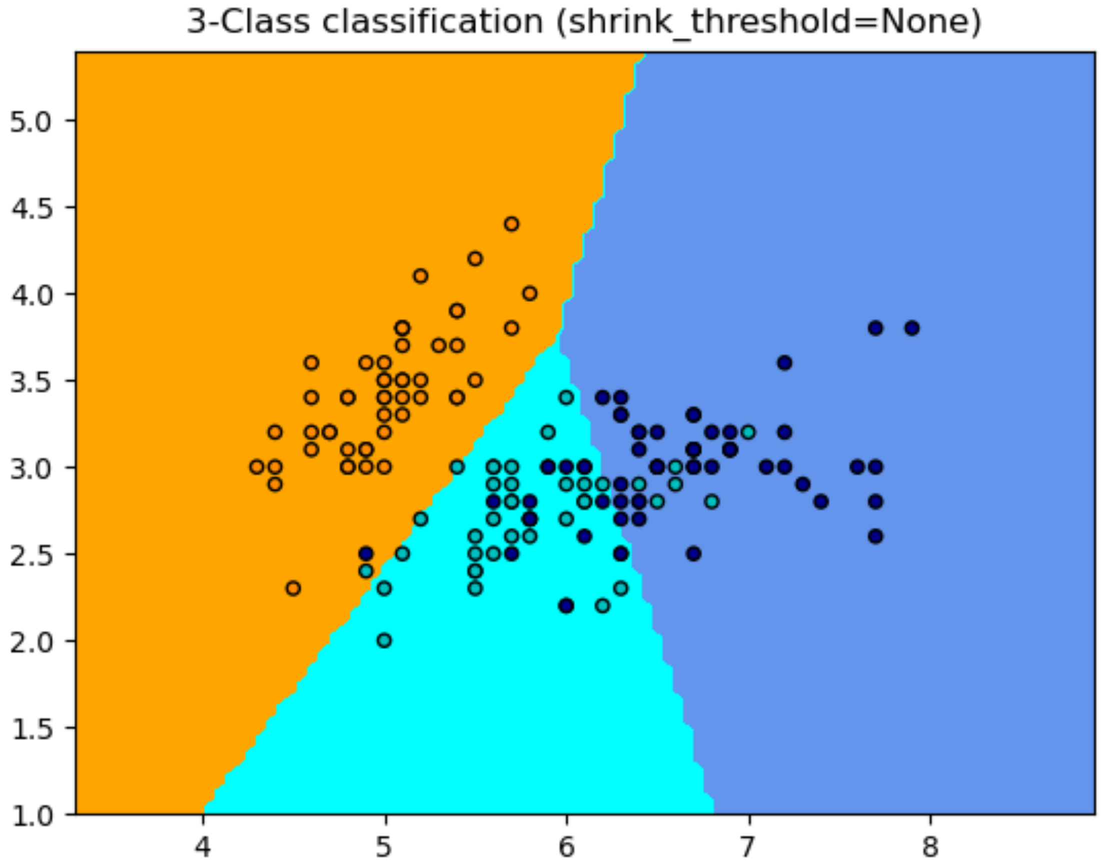
- **General Use:** SVMs can be applied to both **classification** and **regression** problems.
- **Pattern Recognition:** SVMs are useful in pattern recognition.
- **Robotics:** SVMs have applications in robotics.
- **Text Classification:** SVM is applied in spam filters, tracking websites with criminal content and customize search engines, ...
- **Medical Diagnosis**: SVMs are used in medical expert systems for diagnosis.
- **Bioinformatics**: SVMs are used for identifying human genes by reference to mouse genes.
- **Other fields:** SVMs have a range from simple calculation of averages to the construction of complex models, throughout computer science, engineering, computational biology, neuroscience, psychology, and physics

k-Nearest Neighbors Models (k-NN)

- The **k-Nearest Neighbors** (k-NN) algorithm is a simple and both **supervised and unsupervised** *machine learning algorithm* used for both **classification** and **regression** tasks.
- It's a non-parametric method, meaning it doesn't make any assumptions about the underlying distribution of the data.
- But k-NN is a *uncertain reasoning model*, cause of uncertainty in data, uncertainty in prediction and the k as a factor of uncertainty.

k-Nearest Neighbors Models (k-NN)

- KNN



k-Nearest Neighbors Models
How it works

- **Store the training data:** k-NN memorizes all the training data points.
- **Calculate distances:** to predict, k-NN *calculates the distance* between this new point and all the points in the training data. Common distance metrics include *Euclidean distance, Manhattan distance, and Minkowski distance*.
- **Find the nearest neighbors:** It identifies the *k* closest data points (neighbors) to the new point based on the calculated distances.
- **Make a prediction:**
  - **Classification:** k-NN assigns the class that is most frequent among the *k* neighbors to the new data point.
  - **Regression:** k-NN predicts the average (or weighted average) of the target values of the *k* neighbors as the value for the new data point.

Applications of k-NN

- **Recommendation systems:** Suggesting products or content based on user preferences.
- **Image recognition:** Classifying images based on their similarity to known images.
- **Spam detection:** Identifying spam emails based on their content and features.
- **Customer segmentation:** Grouping customers based on their behavior and characteristics.

Discovery and discussion on Supervised Learning models

- There are still many machine learning / Supervised Learning models out there that cannot be introduced within the framework of this course.
- Students try to list them out and try to understand their algorithms.
- Some suggestions: Kernel ridge regression (KRR), Stochastic Gradient Descent (SGD), Gaussian Processes (GP), Gradient Boosted Decision Trees (GBDT), Bagging, Voting, AdaBoost, XGBoost, ...

UNIVERSITY OF GREENWICH

Alliance with FPT Education

Thank you