

Lecture 3

Supervised Learning Algorithms

Artificial Intelligence Programming
Department: Artificial Intelligence, College of Computer Science and Mathematics,
University of Mosul
Level: 2nd Year

1. Introduction

Supervised learning is one of the most widely used paradigms in machine learning and artificial intelligence. In this approach, models are trained using labeled datasets, where each input is associated with a known output. The main objective is to learn a mapping function that can accurately predict outputs for unseen data.

Supervised learning forms the backbone of many real-world AI systems, including medical diagnosis tools, financial risk assessment, image recognition, speech processing, and intelligent industrial control systems.

2. Concept of Supervised Learning

In supervised learning, the learning process is guided by example. Each training instance consists of:

- **Input features (X):** Descriptive variables
- **Output label (y):** Correct answer or target value

The model learns by minimizing the error between its predictions and the true labels provided in the training data.

3. Main Types of Supervised Learning

Supervised learning problems are generally divided into two major categories.

3.1 Classification

Classification tasks aim to assign an input to one of a finite set of predefined classes.

Examples:

- Spam vs. non-spam emails
- Disease vs. healthy patients
- Faulty vs. non-faulty industrial products

Common Classification Algorithms:

- Logistic Regression
 - k-Nearest Neighbors (k-NN)
 - Support Vector Machines (SVM)
 - Decision Trees
 - Naive Bayes
-

3.2 Regression

Regression tasks aim to predict continuous numerical values.

Examples:

- House price prediction
- Energy consumption estimation
- Temperature forecasting

Common Regression Algorithms:

- Linear Regression
 - Polynomial Regression
 - Ridge and Lasso Regression
-

4. Linear Regression

Linear regression is one of the simplest and most interpretable supervised learning algorithms. It models the relationship between one or more input variables and a continuous output using a linear equation.

Simple Linear Regression

Concept: Linear Regression aims to find the best-fitting straight line through a set of data points to predict a continuous output variable based on an input variable. The line is represented by the equation

$$y = mx + c$$

where m is the slope and c is the y-intercept.

Example Data: Let's predict a student's exam score (y) based on the hours they studied (x).

Hours Studied (x)	Exam Score (y)
2	60
3	70
5	90

Goal: Find the equation of the line $y = mx + c$.

Step-by-Step Solution:

1. Calculate the means of x and y:

- $\bar{X} = (2 + 3 + 5) / 3 = 10 / 3 = 3.33$

- $\bar{Y} = (60 + 70 + 90) / 3 = 220 / 3 = 73.33$

2. Calculate the slope (m):

The formula for the slope is $m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2} \sqrt{\sum(y - \bar{y})^2}}$

- $(2 - 3.33)(60 - 73.33) = (-1.33)(-13.33) = 17.73$

- $(3 - 3.33)(70 - 73.33) = (-0.33)(-3.33) = 1.10$

- $(5 - 3.33)(90 - 73.33) = (1.67)(16.67) = 27.84$

- Sum of numerator: $17.73 + 1.10 + 27.84 = 46.67$

- $(2 - 3.33)^2 = (-1.33)^2 = 1.77$

- $(3 - 3.33)^2 = (-0.33)^2 = 0.11$

- $(5 - 3.33)^2 = (1.67)^2 = 2.79$

- Sum of denominator: $1.77 + 0.11 + 2.79 = 4.67$

- $m = 46.67 / 4.67 = 10.0$

3. Calculate the y-intercept (c):

The formula for the y-intercept is

$$c = \bar{Y} - m\bar{X}$$

- $c = 73.33 - (10.0 * 3.33) = 73.33 - 33.3 = 40.03$

4. Final Equation: $y = 10x + 40$

- Prediction: If a student studies for 4 hours, the predicted score is

$$y = 10(4) + 40 = 80.$$

5. Logistic Regression

Logistic regression is a classification algorithm despite its name. It estimates the probability that an input belongs to a particular class using a sigmoid function.

Key Characteristics:

- Output range between 0 and 1
- Suitable for binary classification
- Widely used in medical and risk analysis

على الرغم من اسمه، يُعدّ الانحدار اللوجستي خوارزمية تصنيف. فهو يُقدّر احتمالية انتماء مُدخل ما إلى فئة مُحددة باستخدام دالة سيجمويد.

الخصائص الرئيسية:
• نطاق المُخرجات بين ٠ و ١
• مناسب للتصنيف الثنائي
• يُستخدم على نطاق واسع في التحليل الطبي وتحليل المخاطر

Step 1: The Linear Model (First Part of Logistic Regression)

The general equation is:

$$z = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

What do these symbols mean?

- $x_1, x_2, \dots, x_n \rightarrow$ input features
- $w_1, w_2, \dots, w_n \rightarrow$ weights
- $w_0 \rightarrow$ bias (intercept)
- $z \rightarrow$ linear output

Logistic Regression starts exactly like Linear Regression.
It computes a weighted sum of the inputs.

It is similar to the equation of a straight line:

$$y = mx + b$$

But in multiple dimensions:

$$z = w_0 + w_1x_1 + w_2x_2$$

So the model simply multiplies each feature by its weight and adds them together.

Step 2: Apply the Example

Suppose we are predicting **student success** (Pass = 1, Fail = 0).

Features:

- x_1 = Study hours
- x_2 = Absence days

Weights:

$$w_0 = -4$$

$$w_1 = 0.8$$

$$w_2 = -0.5$$

Student data:

$$x_1 = 6$$

$$x_2 = 2$$

Calculate the Linear Output

$$z = -4 + (0.8 \times 6) + (-0.5 \times 2)$$

$$z = -4 + 4.8 - 1$$

$$z = -0.2$$

So the linear result is:

$$z = -0.2$$

But here is the important point:

- 👉 This number can be anything (negative or positive).
- 👉 It is NOT yet a probability.

We must transform it.

Step 3: The Sigmoid Function

The sigmoid function converts any real number into a value between 0 and 1.

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

This gives us the probability.

Step 4: Apply Sigmoid to Our Example

We have:

$$z = -0.2$$

$$\sigma(-0.2) = \frac{1}{1 + e^{0.2}}$$

We know:

$$e^{0.2} \approx 1.22$$

So:

$$\begin{aligned}\sigma(-0.2) &= \frac{1}{1 + 1.22} \\ &= \frac{1}{2.22} \\ &\approx 0.45\end{aligned}$$

2 Decision Rule (Classification Step)

After computing the probability:

$$\hat{y} = \begin{cases} 1 & \text{if } P \geq 0.5 \\ 0 & \text{if } P < 0.5 \end{cases}$$

(Threshold can be changed depending on the problem.)

Step 5: Interpret the Result

$$P(y = 1) = 0.45$$

This means:

Probability of passing = 45%

Since it is less than 0.5:

$$\hat{y} = 0$$

So the student is classified as **Fail**.

The Complete General Mathematical Model

$$P(y = 1|x) = \frac{1}{1 + e^{-(w_0 + w_1x_1 + \dots + w_nx_n)}}$$

How to Explain This Clearly in Class

You can say:

1. Multiply each feature by its weight.
 2. Add them together → this gives z .
 3. Apply sigmoid to convert z into probability.
 4. If probability ≥ 0.5 → Class = 1.
 5. If probability < 0.5 → Class = 0.
-

5 Steps to Compute Logistic Regression (Manual Calculation)

Let's compute a simple example.

Given:

$$z = -3 + 2x$$

Assume:

$$x = 4$$

Step 1: Compute Linear Output

$$z = -3 + 2(4)$$

$$z = -3 + 8 = 5$$

Step 2: Apply Sigmoid

$$\sigma(5) = \frac{1}{1 + e^{-5}}$$

$$e^{-5} \approx 0.0067$$

$$\sigma(5) = \frac{1}{1 + 0.0067}$$

$$\sigma(5) \approx 0.993$$

Step 3: Interpret Probability

$$P(y = 1) = 0.993$$

Probability = 99.3%

Since it is greater than 0.5:

$$\hat{y} = 1$$

6. k-Nearest Neighbors (k-NN)

k-NN is an instance-based learning algorithm. Instead of learning a global model, it makes predictions based on the similarity between data points.

Key Ideas:

- Distance metrics (e.g., Euclidean distance)
- Choice of k affects performance

Advantages:

- Simple to understand
- No training phase

Disadvantages:

- Computationally expensive for large datasets
- Sensitive to noise

k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is a **supervised learning algorithm** used for classification and regression.

It is a **lazy learning algorithm** because it does not build an explicit model. Instead, it stores the training data and makes predictions based on distance.

1 General Steps of k-NN Algorithm

Step 1: Choose the value of k

- k = number of nearest neighbors
- Small $k \rightarrow$ sensitive to noise
- Large $k \rightarrow$ smoother decision boundary

Step 2: Compute Distance Between the New Point and All Training Points

The most common distance metric is **Euclidean Distance**:

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2}$$

For 2D:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Other distance measures:

- Manhattan Distance:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

- Minkowski Distance:

$$d = \left(\sum |x_i - y_i|^p \right)^{1/p}$$

Step 3: Sort Distances

Arrange all distances in ascending order.

Step 4: Select the k Smallest Distances

Choose the k nearest neighbors.

Step 5: Make Prediction

For **Classification**:

- Use Majority Voting

$$\hat{y} = \text{mode}(y_1, y_2, \dots, y_k)$$

For **Regression**:

- Use Average

$$\hat{y} = \frac{1}{k} \sum_{i=1}^k y_i$$

2 Worked Example (Step-by-Step)

Given Training Data (2D Classification)

Point	X	Y	Class
A	1	2	0
B	2	3	0
C	3	3	1
D	6	5	1

We want to classify new point:

$$P = (3, 2)$$

Let:

$$k = 3$$

Step 1: Compute Euclidean Distances

Distance to A (1,2)

$$\begin{aligned} d &= \sqrt{(3-1)^2 + (2-2)^2} \\ d &= \sqrt{4+0} \\ d &= 2 \end{aligned}$$

Distance to B (2,3)

$$\begin{aligned} d &= \sqrt{(3-2)^2 + (2-3)^2} \\ d &= \sqrt{1+1} \\ d &= \sqrt{2} \\ d &\approx 1.41 \end{aligned}$$

Distance to C (3,3)

$$d = \sqrt{(3 - 3)^2 + (2 - 3)^2}$$
$$d = \sqrt{0 + 1}$$
$$d = 1$$

Distance to D (6,5)

$$d = \sqrt{(3 - 6)^2 + (2 - 5)^2}$$
$$d = \sqrt{9 + 9}$$
$$d = \sqrt{18}$$
$$d \approx 4.24$$

Step 2: Sort Distances

Point	Distance	Class
C	1	1
B	1.41	0
A	2	0
D	4.24	1

Step 3: Select k = 3 Nearest Neighbors

The 3 nearest:

- C → Class 1
- B → Class 0
- A → Class 0

Step 4: Majority Voting

Classes:

$$[1, 0, 0]$$

Majority class = 0

Final Prediction

$$\hat{y} = 0$$

So the new point (3,2) belongs to **Class 0**.

3 When to Use k-NN

Use k-NN when:

- Dataset is small to medium size
- Data is non-linear
- You want simple implementation
- No need for model interpretability

Example 2 — k-NN Classification (Step-by-Step)

Training Data

Point	X	Y	Class
A	2	4	0
B	4	4	0
C	4	2	1
D	6	2	1
E	6	4	1

We want to classify the new point:

$$P = (5, 3)$$

Let:

$$k = 3$$

Step 1: Compute Euclidean Distance

Formula:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Distance to A (2,4)

$$d = \sqrt{(5 - 2)^2 + (3 - 4)^2}$$

$$d = \sqrt{9 + 1}$$

$$d = \sqrt{10}$$

$$d \approx 3.16$$

Distance to B (4,4)

$$d = \sqrt{(5 - 4)^2 + (3 - 4)^2}$$

$$d = \sqrt{1 + 1}$$

$$d = \sqrt{2}$$

$$d \approx 1.41$$

Distance to C (4,2)

$$d = \sqrt{(5 - 4)^2 + (3 - 2)^2}$$

$$d = \sqrt{1 + 1}$$

$$d = \sqrt{2}$$

$$d \approx 1.41$$

Distance to D (6,2)

$$d = \sqrt{(5 - 6)^2 + (3 - 2)^2}$$

$$d = \sqrt{1 + 1}$$

$$d = \sqrt{2}$$

$$d \approx 1.41$$

Distance to E (6,4)

$$d = \sqrt{(5 - 6)^2 + (3 - 4)^2}$$

$$d = \sqrt{1 + 1}$$

$$d = \sqrt{2}$$

$$d \approx 1.41$$

Step 2: Sort Distances

Point	Distance	Class
B	1.41	0
C	1.41	1
D	1.41	1
E	1.41	1
A	3.16	0

Step 3: Choose $k = 3$ Nearest Neighbors

The 3 closest points are:

- B $\rightarrow$ Class 0
- C $\rightarrow$ Class 1
- D $\rightarrow$ Class 1

Step 4: Majority Voting

Classes:

[0, 1, 1]

Class 1 appears twice.

Final Prediction

$$\hat{y} = 1$$

So the new point (5,3) belongs to Class 1.

7. Support Vector Machines (SVM)

Support Vector Machines aim to find the optimal hyperplane that separates classes with the maximum margin.

Key Concepts:

- Margin maximization
- Support vectors
- Kernel functions

Applications:

- Image classification
 - Text categorization
 - Bioinformatics
-

8. Decision Trees

Decision trees use a tree-like structure to model decisions and their possible consequences.

Characteristics:

- Easy to interpret
- Handles both numerical and categorical data

Limitations:

- Prone to overfitting
-

9. Model Evaluation Metrics

Evaluating supervised learning models is essential to measure their performance.

Classification Metrics:

- Accuracy
- Precision
- Recall
- F1-score

Regression Metrics:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- R-squared (R^2)

10. Practical Examples Using Python

Example 1: Linear Regression

```
from sklearn.linear_model import LinearRegression

X = [[1], [2], [3], [4], [5]]
y = [100, 150, 200, 250, 300]

model = LinearRegression()
model.fit(X, y)

print(model.predict([[6]]))
```

Example 2: Classification with k-NN

```
from sklearn.neighbors import KNeighborsClassifier

X = [[2, 60], [4, 70], [6, 80], [8, 90]]
y = [0, 0, 1, 1]

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X, y)

print(model.predict([[5, 75]]))
```

11. Educational Notes

- Supervised learning is data-driven and sensitive to data quality.
 - Proper feature selection improves model accuracy.
 - Evaluation metrics must be chosen carefully based on the task.
-