Hello Everyone,

I hope you all are fine and doing well.

Fuzzy K-Nearest Neighbors (KNN) Method

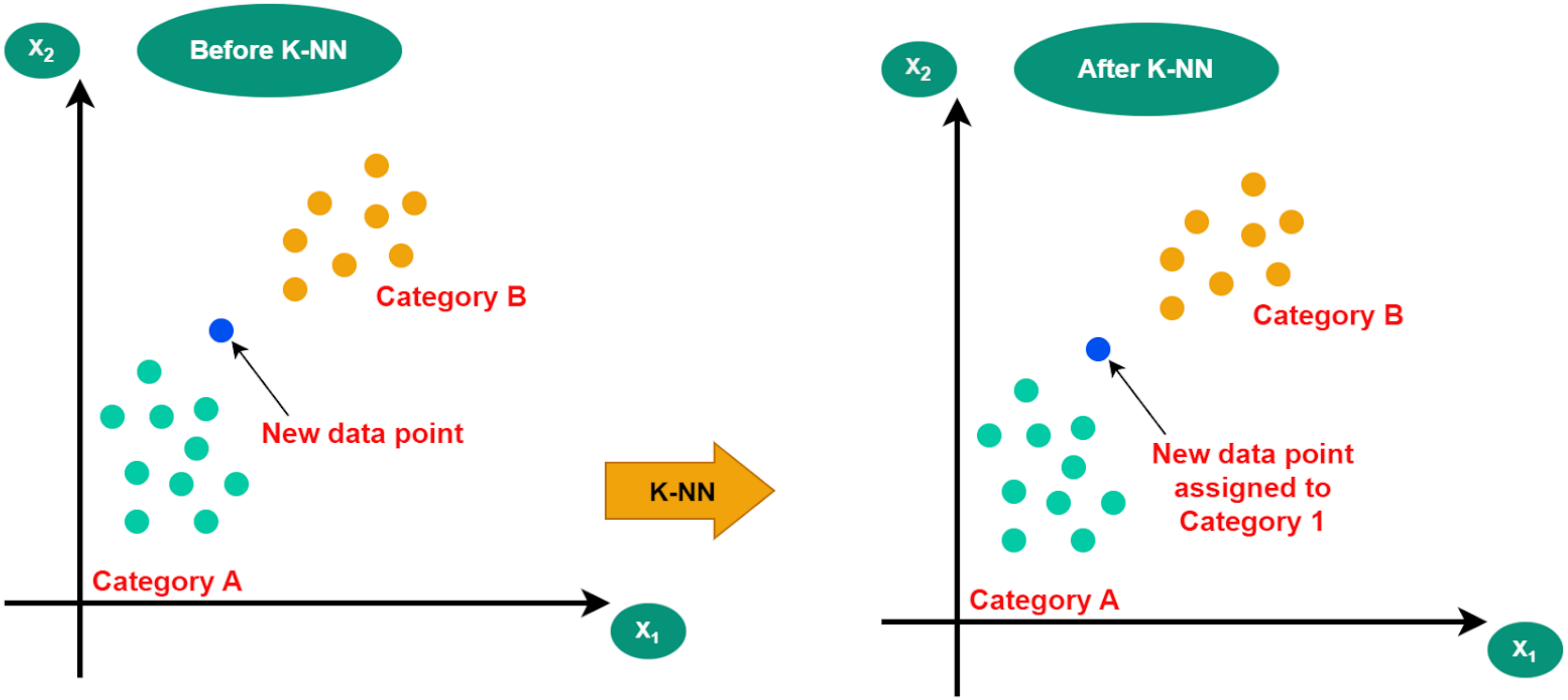
Fuzzy K 最近鄰 (KNN) 方法

Fuzzy K-Nearest Neighbors (Fuzzy K-NN) is an extension of the traditional **K-Nearest Neighbors (K-NN)** algorithm. In classical K-NN, the class label or predicted value is assigned based on the majority class or the average of the K nearest neighbors, with each neighbor contributing equally to the decision.

模糊 K 最近鄰 (Fuzzy K-NN) 是傳統 K 最近鄰 (K-NN) 演算法的擴展。在經典 K-NN 中，類別標籤或預測值是根據多數類別或 K 個最近鄰居的平均值來分配的，每個鄰居對決策的貢獻相同。

However, in **Fuzzy K-NN, membership degrees** are introduced, allowing neighbors to contribute to the prediction based on a **degree of membership** rather than a strict "vote." This makes the algorithm more flexible, particularly when data points do not clearly belong to a single class but can belong to multiple classes to varying degrees. In essence, **Fuzzy K-NN** allows for **fuzziness** in classification and regression.

然而，在模糊 K-NN 中，引入了隸屬度，允許鄰居根據隸屬度而不是嚴格的「投票」為預測做出貢獻。這使得演算法更加靈活，特別是當資料點並不明顯屬於單一類別而是可以不同程度地屬於多個類別時。本質上，Fuzzy K-NN 允許分類和回歸中的模糊性



X_2
Before K-NN
Category B
New data point
Category A
X_1
K-NN
X_2
After K-NN
Category B
New data point
assigned to
Category 1
Category A
X_1

Importance of Fuzzy K-Nearest Neighbors (Fuzzy K-NN)

模糊 K 最近鄰 (Fuzzy K-NN) 的重要性

Fuzzy K-Nearest Neighbors (Fuzzy K-NN) is an extension of the traditional K-Nearest Neighbors (K-NN) algorithm, which introduces the concept of **fuzziness** into classification and regression. This has several significant advantages in real-world problems where the data may not neatly fall into distinct categories or where the relationships between data points are not clear-cut. Here are the key reasons why Fuzzy K-NN is important:

模糊K近鄰（Fuzzy K-NN）是傳統K近鄰（K-NN）演算法的擴展，它將模糊性的概念引入分類和迴歸中。這在現實世界的問題中具有幾個顯著的優勢，在這些問題中，數據可能無法完全劃分為不同的類別，或者數據點之間的關係不明確。以下是模糊 K-NN 如此重要的主要原因

Handling Ambiguity and Overlapping Classes

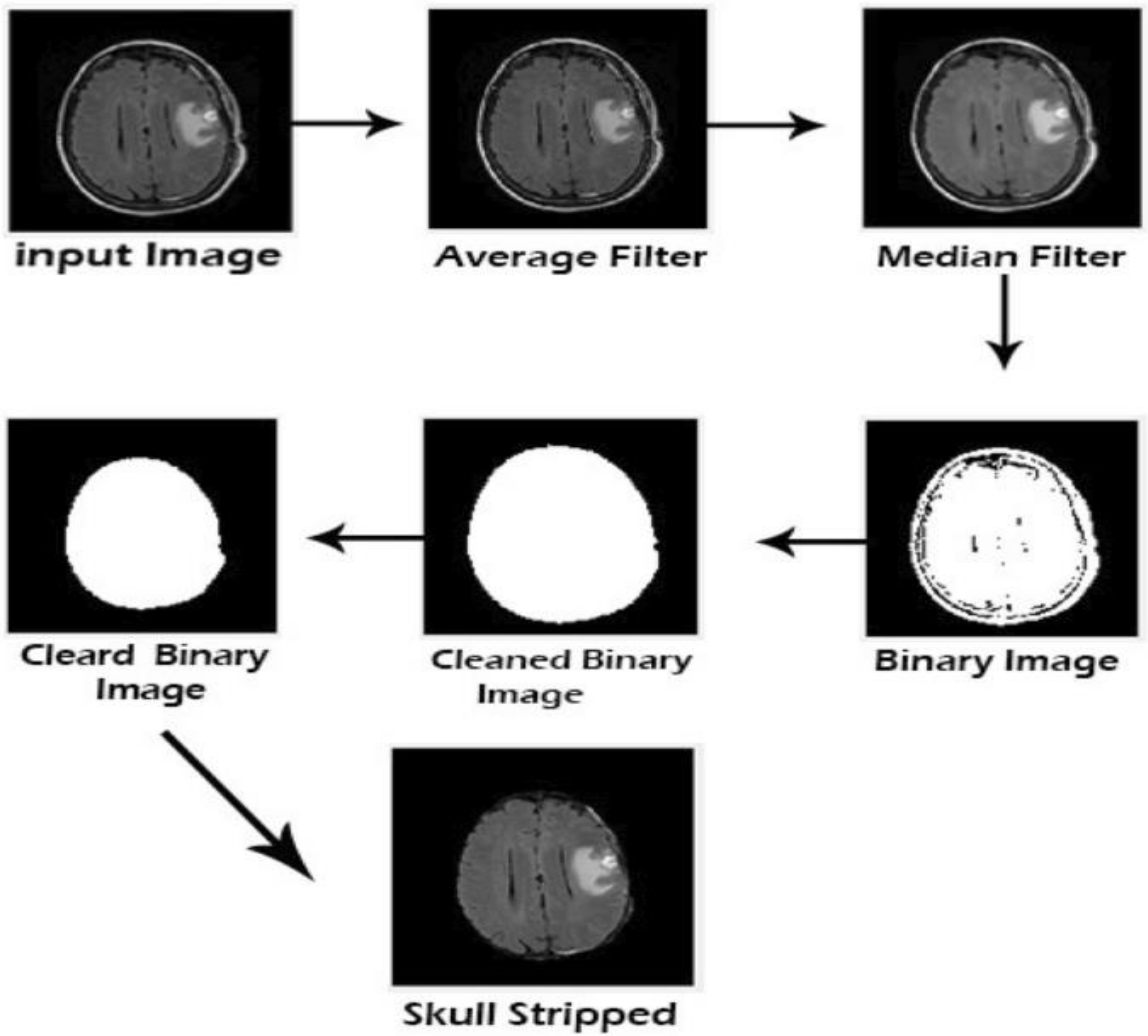
In many real-world datasets, especially in areas like **image recognition, medical diagnostics**, or **speech processing**, data points may not belong to a single, well-defined class. For instance, in medical data, a patient's symptoms could be ambiguous, and a disease might be hard to classify strictly into one category.

Fuzzy K-NN allows for a **soft membership** for data points, meaning a data point can belong to multiple classes with varying degrees of confidence. This helps handle **overlapping or ambiguous data**, where hard classification would lead to poor predictions or misclassifications.

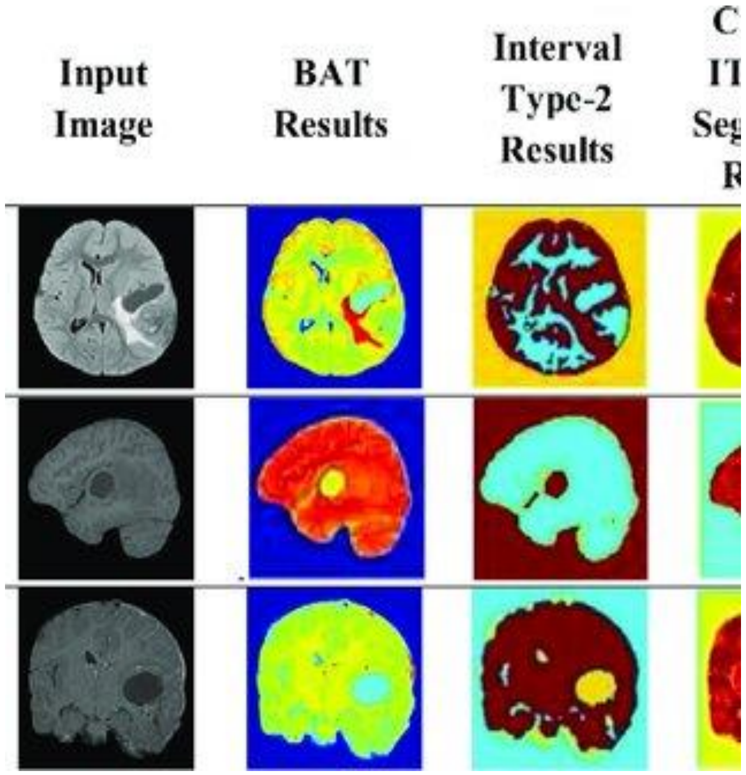
1. 處理歧義和重疊類 在許多現實世界的資料集中，特別是在影像辨識、醫療診斷或語音處理等領域，資料點可能不屬於單一的、定義明確的類別。例如，在醫療數據中，患者的症狀可能不明確，疾病可能很難嚴格歸類為一類。 模糊 K-NN 允許資料點的軟隸屬關係，這意味著資料點可以屬於具有不同置信度的多個類別。這有助於處理重疊或模糊的數據，在這些數據中，硬分類會導致糟糕的預測或錯誤分類

Example: A tumor's characteristics in an image may not clearly belong to "benign" or "malignant" categories. With Fuzzy K-NN, the model can assign partial membership to both categories, representing the uncertainty in the data.

例如：影像中的腫瘤特徵可能不明確屬於「良性」或「惡性」類別。透過 Fuzzy K-NN，模型可以將部分隸屬度分配給兩個類別，表示資料中的不確定性



input Image
Average Filter
Median Filter
Binary Image
Cleaned Binary Image
Cleard Binary Image
Skull Stripped



Input Image
BAT Results
Interval Type-2 Results

Mathematical Form of Fuzzy K-Nearest Neighbors (Fuzzy K-NN)

模糊 K 最近鄰 (Fuzzy K-NN) 的數學形式

The **Fuzzy K-Nearest Neighbors (Fuzzy K-NN)** algorithm extends the traditional K-NN algorithm by allowing for **fuzzy membership** of the data points in each class. In Fuzzy K-NN, each data point is assigned a **degree of membership** for each class based on its proximity to the K nearest neighbors. This allows for a **soft classification**, where a point can belong to multiple classes with varying degrees of confidence.

模糊 K 最近鄰 (Fuzzy K-NN) 演算法透過允許每個類別中資料點的模糊隸屬度來擴展傳統的 K-NN 演算法。在模糊 K-NN 中，每個資料點根據其與 K 個最近鄰居的接近度被分配給每個類別的隸屬度。這允許軟分類，其中一個點可以屬於具有不同置信度的多個類別

Steps in Fuzzy K-NN:

Step 1:

Distance Calculation: The first step is to compute the distance between the test point x and all points in the dataset using a distance metric (commonly **Euclidean distance**). For two points x_i and x_j, the Euclidean distance is given by:

$$d(x_i, x_j) = \left(\sum_{k=1}^{n}(x_{ik} - x_{jk})^2\right)^{\frac{1}{2}}.$$

距離計算：第一步是使用距離度量（通常為歐幾里德距離）計算測試點 x 與資料集中所有點之間的距離。對於兩點 x_i和 x_j，歐幾里德距離由下式給出：

Where n is the number of features (dimensions), and x_{ik} and x_{jk} are the values of the k-th feature for points x_i and x_j, respectively.

其中n是特徵（維度）的數量，x_{ik}和x_{jk}分別是點x_i和x_j的第k個特徵的值。

Step 2:

Membership Degree Calculation: Once the distances are calculated, the next step is to calculate the **membership degrees** for each class. The degree of membership of a data point x_i in class C_j is determined by how close x_i is to the points that belong to class C_j.

In Fuzzy K-NN, the membership degree $\mu_{i,j}$ of point x_i to class C_j is calculated using the **inverse distance** between x_i and its KKK nearest neighbors. The closer a neighbor is, the higher its membership degree for the class.

步驟2： 隸屬度計算：計算距離後，下一步就是計算每個類別的隸屬度。類別 C_j中資料點 x_i的隸屬度取決於 x_i與屬於類別 C_j的點的接近程度。 在模糊 K-NN 中，點 x_i到類別 C_j的隸屬度 $\mu_(i,j)$是使用 x_i與其 KKK 最近之間的反距離計算的鄰鄰。鄰居越近，其在類中的隸屬度就越高

The **membership degree** for point x_i and class C_j is given by:

$$\mu_{i,j} = \frac{1}{\sum_{k=1}^{K} d(x_i, x_k)^{-\beta}}$$

Where:

- $d(x_i, x_k)$ is the distance between the test point x_i and the k-th nearest neighbor x_k.
- K is the number of nearest neighbors.
- β is a parameter that controls the **degree of fuzziness** (how quickly the membership value decreases with distance). A higher value of β results in **crisper** membership (less fuzziness), while a lower value results in **fuzzier** membership (more overlap).

點x_i和類別C_j的隸屬度由下式給出： μ_(i,j)=1/(Σ24_(k=1)^K▒〖d(x_i,x_k)^(−β) 〗) 在哪裡：
d(x_i,x_k) 是測試點x_i和第 k 個最近鄰x_k之間的距離。 K 是最近鄰居的數量。 β 是控制模糊程度（隸屬度值隨距離下降的速度）的參數。 β 值越高，隸屬度越清晰（模糊性越少），而值越低，隸屬度越模糊（重疊越多）

The membership degree $\mu_{i,j}$ indicates the degree to which x_i belongs to class C_j, based on its distance to the K nearest neighbors.

隸屬度μ_(i,j)表示x_i屬於類別C_j的程度，基於其與K最近鄰居的距離

Step 3:

Fuzzy Voting (Weighted Voting): After calculating the membership degrees for each class, the next step is to determine the class label or predicted value. In Fuzzy K-NN, the prediction is based on the **fuzzy voting** principle, where the contribution of each class to the final decision is weighted by its membership degree.

For Classification: The class label for point x_i is determined by the class with the highest total membership degree across all its neighbors. The total membership for class C_j is:

步驟3： 模糊投票（加權投票）：計算出每個類別的隸屬度後，下一步是確定類別標籤或預測值。在Fuzzy K-NN中，預測基於模糊投票原理，其中每個類別對最終決策的貢獻按其隸屬度進行加權。 對於分類：點x_i的類別標籤由其所有鄰居中總隸屬度最高的類別決定。 C_j類別的會員總數為

$$M_{i,j} = \sum_{k=1}^{K} \mu_{i,j}$$

Where:

- $\mu_{i,j}$ is the membership degree of the k-th neighbor to class C_j.
- $M_{i,j}$ represents the total membership of class C_j for the test point x_i.

在哪裡： μ_(i,j)是類別 C_j的第 k 個鄰居的隸屬度。 M_(i,j)表示測試點x_i的類別C_j的總成員資格。

The final classification label $\hat{y}_i$ for the test point x_i is:

$$\hat{y}_i = argmax_j(M_{i,j})$$

This means that the predicted class is the one with the **maximum total membership degree**.

For Regression: In fuzzy regression, the predicted value for the test point x_i is computed as the **weighted average** of the values of the K nearest neighbors, with the membership degrees serving as weights. The predicted value $\hat{y}_i$ is:

測試點*x_i*的最終分類標籤（*y_i*）̂為：(*y_i*)̂=*argmax_j* (*M_*(*i*,*j*)) 這意味著預測的類別是總隸屬度最大的類別。 對於迴歸：在模糊迴歸中，測試點*x_i*的預測值計算為 K 個最近鄰值的加權平均值，其中隸屬度作為權重。預測值 (*y_i*)̂ 為：

$$\hat{y_i} = \frac{\sum_{k=1}^{K} \mu_{k,j} . y_k}{\sum_{k=1}^{K} \mu_{k,j}}$$

Where:

- y_k is the target value for the k-th nearest neighbor.
- $\mu_{k,j}$ is the membership degree of the k-th nearest neighbor in class C_j.

在哪裡： y_k 是第 k 個最近鄰的目標值。 $\mu_(k,j)$是類別C_j中第 k 個最近鄰的隸屬度

Normalization of Membership Degrees (Optional): To ensure that the sum of the membership degrees across all classes for each test point is equal to 1, the membership degrees can be **normalized**:

隸屬度歸一化（可選）：為了確保每個測試點的所有類別的隸屬度總和等於1，可以對隸屬度進行歸一化：

$$\widehat{\mu_{i,j}} = \frac{\mu_{i,j}}{\sum_{j=1}^{C} \mu_{i,j}}$$

Where C is the number of classes, and $\widehat{\mu_{i,j}}$ is the **normalized membership degree**.

其中 C 是類別數，(μ_(i,j))^是歸一化隸屬度。

Summary of Key Formulas for Fuzzy K-NN

1. Distance between two points:

$$d(x_i, x_j) = \left(\sum_{k=1}^{n} (x_{ik} - x_{jk})^2\right)^{\frac{1}{2}}$$

2. Membership degree for point $\boldsymbol{x_i}$ and class $\boldsymbol{C_j}$:

$$\mu_{i,j} = \frac{1}{\sum_{k=1}^{K} d(x_i, x_k)^{-\beta}}$$

3. Class prediction (for classification):

$$\hat{y}_i = argmax_j\left(\sum_{k=1}^{K} \mu_{i,j}\right)$$

4. **Prediction for regression:**

$$\widehat{y_i} = \frac{\sum_{k=1}^{K} \mu_{k,j} \cdot y_k}{\sum_{k=1}^{K} \mu_{k,j}}$$

5. **Optional normalization of membership degrees:**

$$\widehat{\mu_{i,j}} = \frac{\mu_{i,j}}{\sum_{j=1}^{C} \mu_{i,j}}$$

Numerical Example of Fuzzy K-Nearest Neighbors (Fuzzy K-NN)

Let's go through a simple example of how **Fuzzy K-Nearest Neighbors (Fuzzy K-NN)** works in practice. We will use a **2D dataset** and perform **classification** using **Euclidean distance** and fuzzy membership degrees. This example involves **K = 3 nearest neighbors** and **2 classes**.

讓我們透過一個簡單的範例來了解模糊 K 最近鄰 (Fuzzy K-NN) 在實踐中的工作原理。我們將使用二維資料集並使用歐幾里德距離和模糊隸屬度進行分類。此範例涉及 K = 3 個最近鄰和 2 個類別。

Dataset:

Suppose we have the following training dataset with 5 points and their corresponding class labels:

數據集：
假設我們有以下包含 5 個點及其對應類別標籤的訓練資料集：

Points (x, y)	Class
$(1,1)$	A
$(2,1)$	A
$(3,3)$	B
$(6,5)$	B
$(7,8)$	B

Test Point:

We want to classify the point $P = (4,4)$.

Step 1: Calculate the Euclidean Distance between Test Point and Training Points

The Euclidean distance $d(x_i, x_j)$ between two points $x_i = (x_{i1}, x_{i2})$ and $x_j = (x_{j1}, x_{j2})$ in 2D space is given by:

$$d(x_i, x_j) = \left(\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2\right)^{\frac{1}{2}}$$

Let's calculate the Euclidean distance between the test point $P = (4,4)$ and each training point.

1. Distance to (1,1):

$$d\big((4,4),(1,1)\big) = \big((4-1)^2 + (4-1)^2\big)^{\frac{1}{2}} = \big((3)^2 + (3)^2\big)^{\frac{1}{2}}$$
$$= (9+9)^{\frac{1}{2}} = (18)^{\frac{1}{2}} \approx 4.24$$

2. Distance to (2,1):

$$d\big((4,4),(2,1)\big) = \big((4-2)^2 + (4-1)^2\big)^{\frac{1}{2}} = \big((2)^2 + (3)^2\big)^{\frac{1}{2}}$$
$$= (4+9)^{\frac{1}{2}} = (13)^{\frac{1}{2}} \approx 3.61$$

3. Distance to (3,3):

$$d\big((4,4),(3,3)\big) = \big((4-3)^2 + (4-3)^2\big)^{\frac{1}{2}} = \big((1)^2 + (1)^2\big)^{\frac{1}{2}}$$
$$= (1+1)^{\frac{1}{2}} = (2)^{\frac{1}{2}} \approx 1.41$$

4. Distance to (6,5):

$$d\big((4,4),(6,5)\big) = \big((4-6)^2 + (4-5)^2\big)^{\frac{1}{2}} = \big((2)^2 + (1)^2\big)^{\frac{1}{2}}$$

$$= (4+1)^{\frac{1}{2}} = (5)^{\frac{1}{2}} \approx 2.24$$

5. Distance to (7,8):

$$d\big((4,4),(7,8)\big) = \big((4-7)^2 + (4-8)^2\big)^{\frac{1}{2}} = \big((3)^2 + (4)^2\big)^{\frac{1}{2}}$$

$$= (9+16)^{\frac{1}{2}} = (25)^{\frac{1}{2}} \approx 5$$

Step 2: Select the K Nearest Neighbors

For **K = 3**, we choose the 3 closest points to the test point $P = (4,4)$. From the distances calculated, the three closest points are:

- (3,3) with distance 1.41 (Class B)
- (2,1) with distance 3.61 (Class A)
- (6,5) with distance 2.24 (Class B)

Step 3: Calculate Membership Degrees for Each Class

The next step is to calculate the **membership degrees** of the test point $P = (4,4)$ for each class based on the distances of the K nearest neighbors. We use the formula for the membership degree:

$$\mu_{i,j} = \frac{1}{\sum_{k=1}^{K} d(x_i, x_k)^{-\beta}}$$

For simplicity, we will set $\beta = 2$ in this case (this is a common choice for Fuzzy K-NN).

Membership for Class A:

The 3 nearest neighbors to P are:

- Neighbor 1: (2,1) (Class A), distance = 3.61
- Neighbor 2: (3,3) (Class B), distance = 1.41
- Neighbor 3: (6,5) (Class B), distance = 2.24

We calculate the membership for class A based on the distances (with β=2):

$$\mu_A = \frac{1}{d(4,2)^{-2} + d(4,3)^{-2} + d(4,6)^{-2}}$$

$$= \frac{1}{3.61^{-2} + 1.41^{-2} + 2.24^{-2}}$$

Calculating the individual terms:

$$3.61^{-2} = 0.077$$
$$1.41^{-2} = 0.507$$
$$2.24^{-2} = 0.199$$

So:

$$\mu_A = \frac{1}{3.61^{-2} + 1.41^{-2} + 2.24^{-2}}$$
$$= \frac{1}{0.077 + 0.507 + 0.199}$$
$$= \frac{1}{0.7831} \approx 1.28$$

It means that

$$\mu_A \approx 1.28$$

Membership for Class B:

The 3 nearest neighbors to P are:

- Neighbor 1: (2,1) (Class A), distance = 3.61
- Neighbor 2: (3,3) (Class B), distance = 1.41
- Neighbor 3: (6,5) (Class B), distance = 2.24

We calculate the membership for class A based on the distances (with β=2):

$$\mu_B = \frac{1}{d(4,2)^{-2} + d(4,3)^{-2} + d(4,6)^{-2}}$$
$$= \frac{1}{3.61^{-2} + 1.41^{-2} + 2.24^{-2}}$$

Calculating the individual terms:

$$3.61^{-2} = 0.077$$
$$1.41^{-2} = 0.507$$
$$2.24^{-2} = 0.199$$

So:

$$\mu_B = \frac{1}{3.61^{-2} + 1.41^{-2} + 2.24^{-2}}$$
$$= \frac{1}{0.077 + 0.507 + 0.199}$$
$$= \frac{1}{0.7831} \approx 1.28$$

It means that

$$\mu_B \approx 1.28$$

Final Thoughts:

Fuzzy K-NN offers a more **flexible and nuanced approach** to classification compared to traditional K-NN, especially in situations where the decision boundaries are not crisp. By incorporating **fuzzy membership**, it provides more realistic predictions for complex datasets, making it a valuable tool in **uncertain or overlapping data scenarios**.