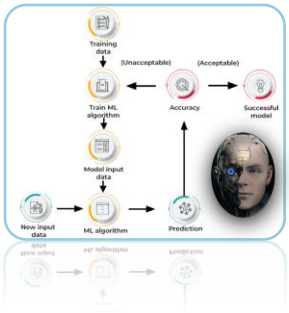
Machine Learning



Dr. Hrishikesh Bhaumik | hrishikesh.bhaumik@rcciit.org.in

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

Distance measures, Nearest Neighbourhood, KNN algorithm.

START: Distance measures, Nearest Neighbourhood, KNN algorithm.

Distance Measure

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

Distance measures in machine learning are essential for various tasks, such as clustering, classification, and recommendation systems. These measures help **quantify the similarity or dissimilarity between data points** in a feature space. Different distance metrics are used based on the nature of the data and the specific problem.

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Common Distance Measures

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

◆1. **Euclidean Distance**:
This is the most common distance measure and is used for continuous data. It computes the **straight-line distance** between two points in Euclidean space. In 2D space, the Euclidean distance between points (x1, y1) and (x2, y2) is given by
d = sqrt((x1 - x2)^2 + (y1 - y2)^2)

◆2. **Manhattan Distance** (city block distance):
Also known as the city block distance or L1 distance, this computes the distance by **summing the absolute differences** between coordinates. In 2D space, the Manhattan distance between points (x1, y1) and (x2, y2) is given by
d = |x1 - x2| + |y1 - y2|

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Common Distance Measures

ML

◆3. **Minkowski Distance**:
The Minkowski distance is a **generalization** of both Euclidean and Manhattan distances.
It is given by:
d = (Σ|x1i - x2i|^p)^(1/p)
The parameter "p" determines whether it behaves like Euclidean distance (p=2) or Manhattan distance (p=1).

◆4. **Cosine Similarity**:
Cosine similarity measures the **cosine of the angle between two non-zero vectors**. It is widely used for text and document analysis. It ranges from -1 (perfectly dissimilar) to 1 (perfectly similar), with 0 indicating **orthogonality**.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Common Distance Measures

ML

◆5. **Jaccard Distance**:
Jaccard distance is used for sets and measures **dissimilarity between two sets** by calculating the size of their intersection divided by the size of their union.
J(A, B) = |A ∩ B| / |A ∪ B|

◆6. **Hamming Distance**:
Hamming distance is used for **categorical or binary data**. It counts the number of positions at which two strings of equal length differ.

◆7. **Mahalanobis Distance**:
This is used when data points have a **multivariate normal distribution**. It considers the correlation between variables and scales the distances accordingly.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Common Distance Measures

ML

◆8. **Edit Distance (Levenshtein Distance):**
Edit distance **measures the minimum number of operations** (insertions, deletions, or substitutions) required to transform one string into another. It is often used in text similarity and spell-checking.

◆9. **KL Divergence (Kullback-Leibler Divergence):**
It measures the **difference between two probability distributions**. It is often used in information theory and can assess how one distribution differs from another.

◆10. **Squared Distance:**
This is squared Euclidean distance, which is a variant of the Euclidean distance measure.
Squared Distance = (x1 - x2)^2 + (y1 - y2)^2

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Nearest Neighbourhood

ML

◆The principle behind nearest neighbour methods is to find predefined **training samples which are closest in distance to the desired point** and which are labeled or categorized.

◆The distance can be computed using **any metric measure**.

◆Euclidean distance is the most common choice.

◆The nearest neighbour method can be used for both **regression** and **classification tasks**.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

k -Nearest Neighbour (*k*NN) Algorithm

ML

- The name of the algorithm originates from the underlying philosophy of ***k*NN** – i.e. people having similar background or mindset tend to stay close to each other.
- The ***k*NN** algorithm is a simple but extremely powerful **classification algorithm**.
- Neighbours in a **locality** have a **similar background**.
- The **unknown** and **unlabelled** data which comes for a prediction problem is judged on the basis of the **elements contained in the training dataset** which are similar to the unknown element.
- So, the class label of the **unknown element** is assigned on the basis of the class labels of the **similar training dataset elements** (metaphorically can be considered as neighbours of the unknown element).

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Working of *k*NN Algorithm

ML

- **Training:**

The algorithm begins with a **training dataset**, which consists of **labeled data points**. Each data point has **features** (attributes) and a corresponding **class label** (for classification) or **target value** (for regression). These data points are used to learn the **relationships** between the **features** and the **labels**.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Working of *k*NN Algorithm

ML

- **Prediction for a New Data Point:**

Step 1: When a prediction is to be made for a new, **unlabeled data point**, the algorithm calculates the distance between this data point and all data points in the training set. The most common distance metric used is the Euclidean distance, but other distance metrics can be used depending **on the problem** and **type of data**.

Step 2: The algorithm selects the **k-nearest data points** from the **training set** based on the computed distances. These are the data points with the **smallest distances** to the new data point.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Working of *k*NN Algorithm

ML

- **Prediction for a New Data Point:**

Step 3A (Classification): For a **classification task**, the algorithm assigns a **class label to the new data point** by taking a **majority vote among the *k*-nearest neighbors**. The class label that occurs most frequently among the neighbours is assigned to the new data point.

Step 3B (Regression): For a **regression task**, the algorithm assigns a **numerical value to the new data point** by taking the **average** (or **weighted average**) of the target values of the **k-nearest neighbours.**

The key hyperparameter in kNN is "*k*," which represents the number of nearest neighbours to consider. **The choice of "*k*" can significantly impact the algorithm's performance.**

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

*k*NN Example

ML

No. of Students = 15
Scores are on a scale of 10
Two performance parameters:
'**Aptitude**' and '**Communication**'.

Class value assigned based on following criteria:
1. '**Leader**' = Good communication skills & Good level of aptitude.
2. '**Speaker**' = Good communication skills but not so good level of aptitude.
3. '**Intel**' = Not so good communication skill but Good level of aptitude.

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

*k*NN Example

ML

Name	Aptitude	Communication	Class
Karuna	2	5	Speaker
Bhuvna	2	6	Speaker
Gaurav	7	6	Leader
Parul	7	2.5	Intel
Dinesh	8	6	Leader
Jani	4	7	Speaker
Bobby	5	3	Intel
Parimal	3	5.5	Speaker
Govind	8	3	Intel
Susant	6	5.5	Leader
Gouri	6	4	Intel
Bharat	6	7	Leader
Ravi	6	2	Intel
Pradeep	9	7	Leader
Josh	5	4.5	?

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

*k*NN Example

ML

Class label of the test data elements is decided by the **class label of the training data elements** which are **neighbouring**, i.e. similar in nature.

But there are two challenges:
- 1. What is the **basis of this similarity** or when can we say that two data elements are similar?
- 2. **How many** similar elements should be considered for **deciding the class label** of each test data element?

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

*k*NN Example

ML

Euclidean distance
Considering a very simple data set having two features (say f_1 and f_2), Euclidean distance between two data elements d_1 and d_2 can be measured by

$$\text{Euclidean distance} = \sqrt{(f_{11} - f_{12})^2 + (f_{21} - f_{22})^2}$$

where f_{11} = value of feature f_1 for data element d_1

f_{12} = value of feature f_1 for data element d_2

f_{21} = value of feature f_2 for data element d_1

f_{22} = value of feature f_2 for data element d_2

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rcciit.org.in

*k*NN Example

The record of the student named **Josh** is assumed to be the test data.

	Name	Aptitude	Communication	Class
Training Data	Karuna	2	5	Speaker
	Bhuvna	2	6	Speaker
	Gaurav	7	6	Leader
	Parul	7	2.5	Intel
	Dinesh	8	6	Leader
	Jani	4	7	Speaker
	Bobby	5	3	Intel
	Parimal	3	5.5	Speaker
	Govind	8	3	Intel
	Susant	6	5.5	Leader
	Gouri	6	4	Intel
	Bharat	6	7	Leader
	Ravi	6	2	Intel
	Pradeep	9	7	Leader
Test Data	Josh	5	4.5	?

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

*k*NN Example

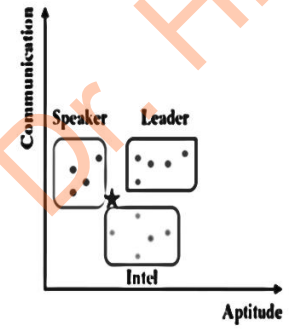
The training data points of the Student data set considering only the features '**Aptitude**' and '**Communication**' can be represented as dots in a **two-dimensional feature space**.

The training data points having the **same class value** are coming **close to each other**. The reason for considering **two-dimensional data space** is that we are considering just the two features of the Student data set, i.e. 'Aptitude' and 'Communication', for doing the classification.

The feature 'Name' **is ignored** because, it has no role to play in deciding the class value.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

*k*NN Example



ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

*k*NN Example

Distance calculation between test and training points.
If k=1 Then
Training data1=Gouri -> Class='Intel'

If k=2 Then
Training data1=Gouri -> Class='Intel'
Training data2=Susant -> Class='Leader'

If k=3 Then
Training data1=Gouri -> Class='Intel'
Training data2=Susant -> Class='Leader'
Training data2=Bobby -> Class='Intel'
=> Josh Class value is 'Intel'.
In this case, the class value of Josh is decided by majority voting.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

*k*NN Example

Name	Aptitude	Communication	Class	Distance	k = 1	k = 2	k = 3
Karuna	2	5	Speaker	3.041			
Bhuvna	2	6	Speaker	3.354			
Parimal	3	5.5	Speaker	2.236			
Jani	4	7	Speaker	2.693			
Bobby	5	3	Intel	1.500			1.500
Ravi	6	2	Intel	2.693			
Gouri	6	4	Intel	1.118	1.118	1.118	1.118
Parul	7	2.5	Intel	2.828			
Govind	8	3	Intel	3.354			
Susant	6	5.5	Leader	1.414			
Bharat	6	7	Leader	2.693			
Gaurav	7	6	Leader	2.500			
Dinesh	8	6	Leader	3.354			
Pradeep	9	7	Leader	4.717			
Josh	5	4.5	???				

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

What will be the value of *k* in *k*NN?

It is often a tricky decision to decide the value of k.

The reasons are as follows:

- If the value of **k is very large** (in the extreme case equal to the total number of records in the training data), the class label of the **majority class** of the training data set will be assigned to the test data regardless of the class labels of the neighbours nearest to the test data.

- If the value of **k is very small** (in the extreme case equal to 1), the class value of a **noisy data** or **outlier** in the training data set which is the nearest neighbour to the test data will be assigned to the test data.

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

Strategies for deciding value of *k*

- 1. Set *k* equal to the square root of the number of training records.

- 2. Test several *k* values on a variety of test data sets and choose the one that delivers the best performance.

- 3. Choose a larger value of *k*, but apply a weighted voting process in which the vote of close neighbours is considered more influential than the vote of distant neighbours.

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

*k*NN Algorithm

Input: Training data set, test data set (or data points), value of 'k' (i.e. number of nearest neighbours to be considered)

Do for all test data points

- Calculate the distance (usually Euclidean distance) of the test data point from the different training data points.
- Find the closest 'k' training data points, i.e. training data points whose distances are least from the test data point.

  If *k* = 1 Then

  Assign class label of the training data point to the test data point.

  Else

  Whichever class label is predominantly present in the training data points, assign that class label to the test data point

End do

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

ML
Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

Why *k*NN algorithm is a lazy learner?

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

Eager learners follow the general steps of machine learning, i.e. perform an **abstraction of the information** obtained from the input data and then follow it through by a **generalization step**.

In the case of the kNN algorithm, these steps are completely skipped.

kNN stores the training data and directly applies the philosophy of nearest neighbourhood finding to arrive at the classification.

So, for kNN, there is no learning happening in the real sense. Therefore, kNN falls under the category of lazy learner.

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Strengths of the *k*NN algorithm

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

- Extremely simple algorithm – easy to understand
- Very effective in certain situations, e.g. for recommender system design
- Very fast or almost no time required for the training phase

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Weaknesses of the kNN algorithm

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

- **Does not learn anything** in the real sense. Classification is done completely on the basis of the training data. So, it has a **heavy reliance on the training data**. If the training data does not represent the problem domain comprehensively, the algorithm **fails to make an effective classification**.
- Because **there is no model trained** in real sense and the classification is done completely on the basis of the training data, the **classification process may be very slow**.
- Also, a **large amount of computational space** is required to load the training data for classification.

continue: Distance measures, Nearest Neighbourhood, KNN algorithm.

Application of the *k*NN algorithm

ML

Dr. Hrishikesh Bhaumik ☐ hrishikesh.bhaumik@rccit.org.in

- One of the most popular areas in machine learning where the kNN algorithm is widely adopted is **recommender systems**. Recommender systems recommend users different items which are similar to a particular item that the user seems to like. The liking pattern may be revealed from past purchases or browsing history and the similar items are identified using the kNN algorithm.
- Another area where there is widespread adoption of kNN is searching documents/contents similar to a given document/content. This is a core area under information retrieval and is known as **concept search**.

END: Distance measures, Nearest Neighbourhood, KNN algorithm.