

# Lecture 06 : Vectors, Embeddings & Vector Databases

---

## ◆ Introduction: Problem Statement

Netflix, Amazon, YouTube ke **recommendation systems** kaise kaam karte hain?  
Jab hum koi product kharidte hain, to **related products** kaise suggest hote hain?  
YouTube ke **search results** itne **relevant** kaise hote hain?

✂ Aaj hum in sab problems ko solve karte hue **vectors** ki **requirement** samjhenge.

---

## ◆ Solution 1: Manual Array Approach (Brute Force)

### ◆ Kaise Kaam Karta Hai?

📁 **MANUAL CATEGORIES BANAYE:**

Gym Items			Fruits			Kitchen Items		
• Protein			• Apple			• Blender		
• BCAA			• Orange			• Toaster		
• Creatine			• Banana			• Coffee Maker		
• Shaker			• Mango					
• Yogamat			• Watermelon					

- ➡ **User ne Protein kharida** → Pure Gym Items array recommend kar do
  - ➡ **User ne Onion kharida** → Vegetables array ke saare items recommend kar do
-

## ⚠ Problems Is Approach Mein

- 🤖 **Not Scalable:**  
1 million products ko manually categorize karna impossible
  - 🔄 **Manual Work:**  
Naye product aaye to engineer ko manually arrays mein add karna padega
  - 🚫 **Rigid Boundaries:**
    - Protein ke saath Banana bhi use hota hai, lekin different arrays mein hain
    - Blender bhi Protein ke saath use hota hai, lekin recommend nahi ho pata
  - ❌ **New Relationships Discover Nahi Kar Pata:**  
Walmart ki **"Diaper aur Beer"** wali story – Friday ko dono sath bikte the  
→ Yeh relationship manual system discover nahi kar pata
  - 🗒 **Context Problem:**  
"Orange" fruit bhi hai aur color bhi – system context nahi samajh pata
- 

## 💠 Solution 2: Graph-Based Approach

### 💠 Kaise Kaam Karta Hai?

- Har product ek **Node** ban jata hai
- Products ke beech **Edges hote hain** with weights

### 🎯 GRAPH REPRESENTATION:

Protein —50— BCAA

| \ |

| \ |

30 \ 1 40

| \ |

| \ |

Creatine Shaker |

| |

└──25──┘

🧠 Jab 2 products sath mein kharide jate hain, unke beech edge ka weight badh jata hai

➡ **Recommendation:** Highest weight wale connections prioritize karo

---

## ◆ Implementation - 2D Array





### WEIGHT MATRIX:

	Prot	BCAA	Crea	Shaker
Prot	0	50	30	1
BCAA	50	0	0	40
Crea	30	0	0	25
Shaker	1	40	25	0

➡ **User ne Protein kharida** → Sort karo weights ko → BCAA (50), Creatine (30), Shaker (1) recommend karo

---

### ⚠ Problems Is Approach Mein

-  **Storage Issue:**  
1M products ke liye 1M x 1M matrix chahiye – bahut bada
  -  **Sorting Slow:**  
Har product ke liye 1M items sort karna padega
  -  **Cold Start Problem:**  
Shuru mein sab weights zero – koi recommendation nahi
  -  **Semantic Meaning Nahi Samajhta:**
    - ON Protein aur My Protein dono **proteins** hain, lekin graph dono ko alag treat karta hai
    - Agar ON Protein **out-of-stock** hai, to My Protein recommend nahi hoga
- 

## ◆ Solution 3: Number Line Approach (1-Dimensional)

### ◆ Kaise Kaam Karta Hai?

- Har product ko ek **unique number** assign karo
- Similar products ko **number ranges mein group** karo

## NUMBER LINE SYSTEM:

1-100: Fruits

101-150: Gym Items




151-200: Kitchen

1 Apple	101 Prot	151 Blend
2 Banana	102 BCAA	152 Toast
3 Waterm	103 Crea	153 Coffe
...	...	...

→ User ne Watermelon (3) kharida → Recommend Banana (2) aur ? (4)

→ User ne Creatine (103) kharida → Recommend BCAA (102) aur Protein (104)

### Problems Is Approach Mein

-  **Boundary Problem:**
  - Product number 1 ka sirf ek hi neighbor hai
  - Product number 100 ka sirf ek hi neighbor hai
-  **Long-Distance Relationships Fail:**
  - Protein (101) aur Blender (151) bahut door hain
  - Protein ke saath **coffee bhi pite hain**, lekin coffee number line mein door hai
-  **Insertion Problem:**

Naya gym product aaya – **kahan insert karenge?** Number ranges already full hain

### Key Insight:

👉 Real-world relationships **multi-dimensional** hote hain, hum **1D** mein solve nahi kar sakte

## The Ultimate Solution: Vectors & Embeddings

### Movie Recommendation Example - 2D Space

#### MOVIES IN 2D SPACE:

High Comedy 😄

↑

| 3 Idiots • Golmaal •

| Hera Pheri •

|

| War • Pathan •

|

Low Action —————→ High Action 🥊

| • Lagaan

| • Swades

| • Taare Zameen Par

↓

Serious 😐

🕒 **X-axis:** Action Level (-10 = Peaceful, +10 = High Action)

👋 **Y-axis:** Comedy Level (-10 = Serious, +10 = High Comedy)

➡ **User ne 3 Idiots dekhi** → Recommend nearby: Golmaal, Hera Pheri

---

## ◆ Multi-Dimensional Vectors (Vector Embeddings)

Real life mein 2 dimensions **kafi nahi hain**

🎯 **5-DIMENSIONAL MOVIE VECTOR:**

War Movie = [Action, Comedy, Drama, Romance, Realism]

= [10, -5, -4, 2, -7]




📌 Yeh list of numbers hi **VECTOR** hai

Modern AI models **496+ dimensions** tak ke vectors bana sakte hain!

---

## ◆ Vectors Kaise Create Hote Hain?

Neural Networks **automatically** vectors create karte hain.

 **Input:** "King"  
 **Output:** [10, 7, 3, 8, 9, ...]  
 Hum manually numbers assign nahi karte

---

## ◆ Famous "King - Man + Woman = Queen" Example


### VECTOR ARITHMETIC:

King = [Royalty, Male, Power, Wealth] = [10, 10, 9, 10]

Man = [Royalty, Male, Power, Wealth] = [1, 10, 2, 3]

Woman = [Royalty, Female, Power, Wealth] = [1, 0, 2, 3]

King - Man + Woman = [10-1+1, 10-10+0, 9-2+2, 10-3+3]

= [10, 0, 9, 10] ≈ Queen 

### Why it works:

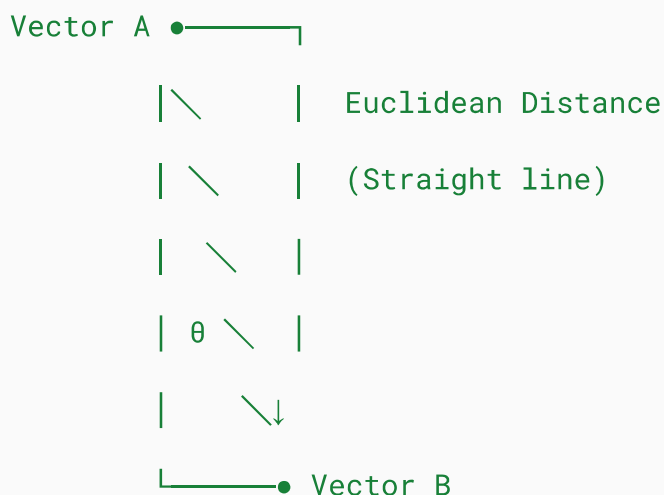
King - Man = Royalty, Power, Wealth (jo king mein extra hai)


- + Woman = Yeh qualities woman mein add karo to Queen milta hai




## Similarity Kaise Measure Karein?

### ◆ Cosine Similarity vs Euclidean Distance

#### COMPARISON DIAGRAM:



 **Cosine Similarity** → angle  $\theta$  ko measure karti hai

Method	What it Does	Best For
 Euclidean Distance	Distance between points	General geometry
 Cosine Similarity	Angle between vectors (ignores size)	Text & Semantics 

 Cosine Similarity:

- 1 = Same direction (high similarity)
- 0 = No relation
- -1 = Opposite

 **Cosine is best for semantic similarity**

---

## Vector Databases

### Traditional DB vs Vector DB

 Traditional Database:

```
|_____|
| ID: 123      | → Finds EXACT matches
| Name: "King" |
|_____|
```

 Vector Database:

```
|_____|
| Vector: [10,7,3..] | → Finds SIMILAR items (nearest neighbors)
|_____|
```

---

◆ YouTube Search Example

- 🔍 **User searches:** "What is Array?"
- ➡ Convert to vector → [8, -2, 5, 1, -3, ...]
- ➡ Find closest vectors:
- ✅ "Array Tutorial" → [7, -1, 6, 2, -2]
- ✅ "Data Structures" → [8, -3, 4, 1, -4]
- ❌ "Cooking Recipe" → [-5, 2, -3, 8, 1]
- ➡ YouTube **keywords nahi, meaning match karta hai!**

📖 Complete Summary

🎯 **TIMELINE:**

Manual Arrays → Graph Weights → Number Lines → VECTOR EMBEDDINGS			
❌	❌	❌	✅
Rigid	Co-occurrence	1D Thinking	Multi-dimensional
Static	Only	Limited	Semantic Understanding

◆ Key Definitions

Concept	Definition	Simple Example
Vector	List of numbers in multi-dimensional space	Movie ka fingerprint: [Action, Comedy, Drama]
Vector Embedding	Text/images ko vector mein convert karna	Word ko AI-friendly form mein lana
Vector Database	Similar items dhoondhne wali smart library	YouTube, Google Search, Amazon Recommendations



## Final Thought

❌ **Purana Tarika:** "If you bought X, you might like Y" (manual rules)

✅ **Naya AI Tarika:** "Items that are conceptually similar in high-dimensional space are recommended"

➡ **Vectors computers ko semantic relationships samajhne ki capability dete hain**

---

## Visual Recap:

1 Manual Arrays →

2 Graphs →

3 Number Lines →

4 Multi-dimensional **Vector Embeddings** ✅

📌 **Yahi hai secret of modern AI recommendations, search, aur content understanding!**