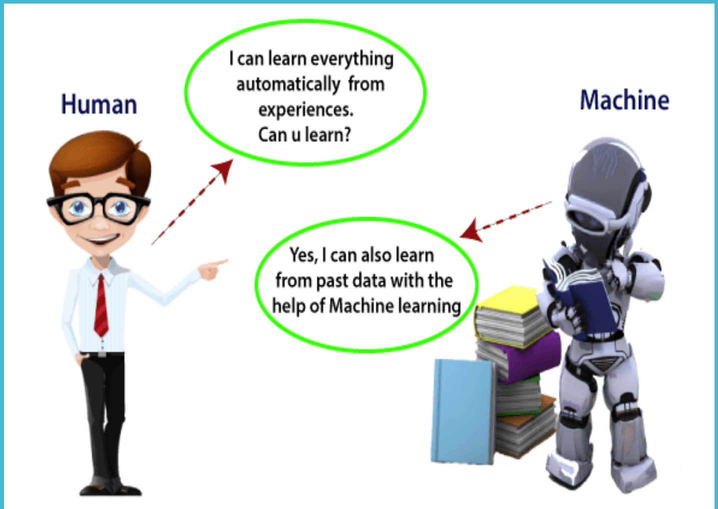
Recommendation System



Marwadi University

Department of Information and Communication Technology

Unit 10: Recommendation Systems

Artificial Intelligence (01CT0703)

Prof. Nishith Kotak

How does Humans Behave

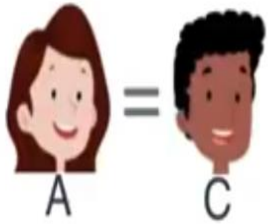
	M 1	M 2	M 3	M 4	M 5
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3
	3	3	3	3	3

	M 1	M 2	M 3	M 4	M 5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

	M 1	M 2	M 3	M 4	M 5
	1	3	2	5	4
	2	1	1	1	5
	3	2	3	1	5
	2	4	1	5	2

How does Humans Behave

	M1	M2	M3	M4	M5
	3	1	1	3	1
	3	1	1	3	1

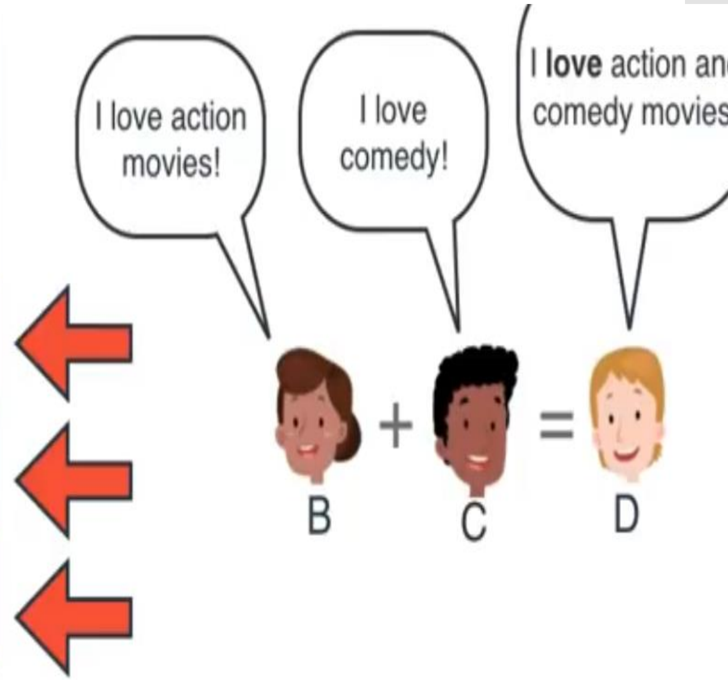


How does Humans Behave

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

How does Humans Behave

	M1	M2	M3	M4	M5
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4



How does Recommender System Behaves

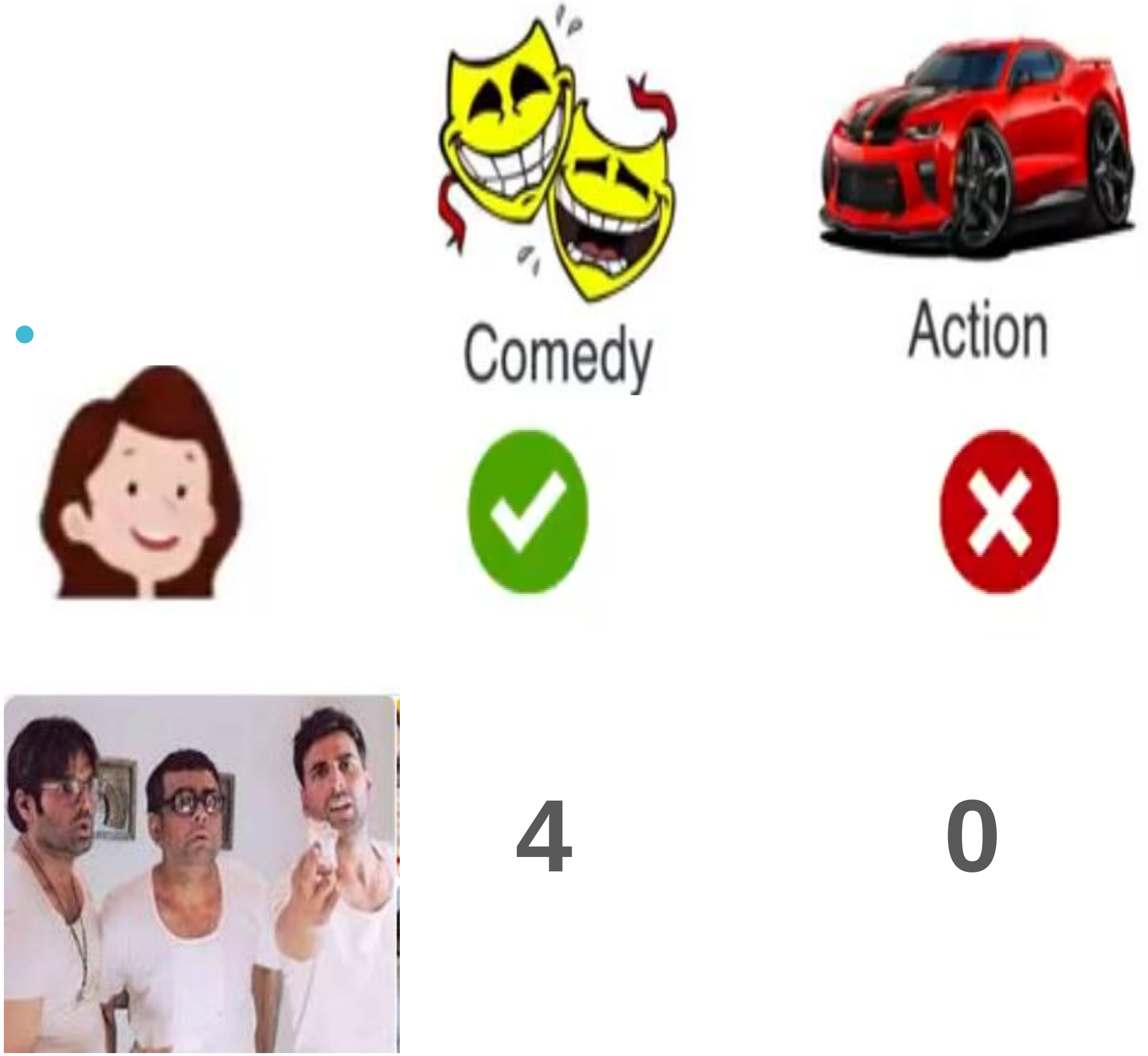
	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	
	4	3	5	4	4

Figuring out the dependencies? ---- Using Matrix Factorization

- Depends on different Features

Figuring out the dependencies?

Using Matrix Factorization



Figuring out the dependencies?

Using Matrix Factorization

4

0

Figuring out the dependencies? ---- Using Matrix Factorization

Comedy

Action

- 

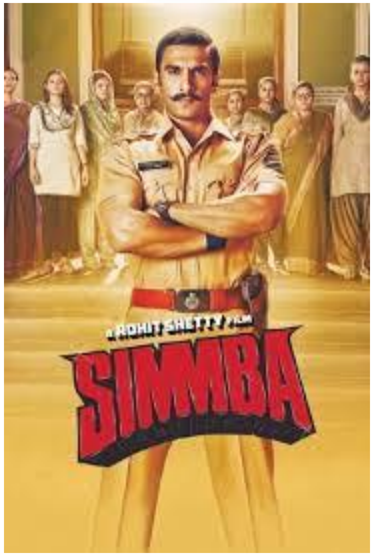
3.5

0.5

Figuring out the dependencies? ---- Using Matrix Factorization

Comedy

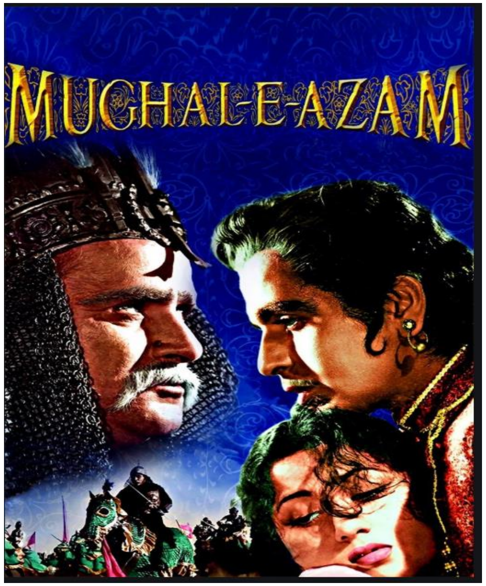
Action



3.0

1.0

Figuring out the dependencies? ---- Using Matrix Factorization



0

0.5

	Comedy	Action
M1	3	1
M2	1	2
M3	1	4
M4	3	1
M5	1	3

	Comedy	Action
	✓	✗
	✗	✓
	✓	✗
	✓	✓

	Comedy	Action
M1	3	1
M2	1	2
M3	1	4
M4	3	1
M5	1	3

	Comedy	Action
	✓	✗
	✗	✓
	✓	✗
	✓	✓

=

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Matrix Factorization

	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

	Comedy	Action
A	✓	✗
B	✗	✓
C	✓	✗
D	✓	✓

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Matrix Factorization

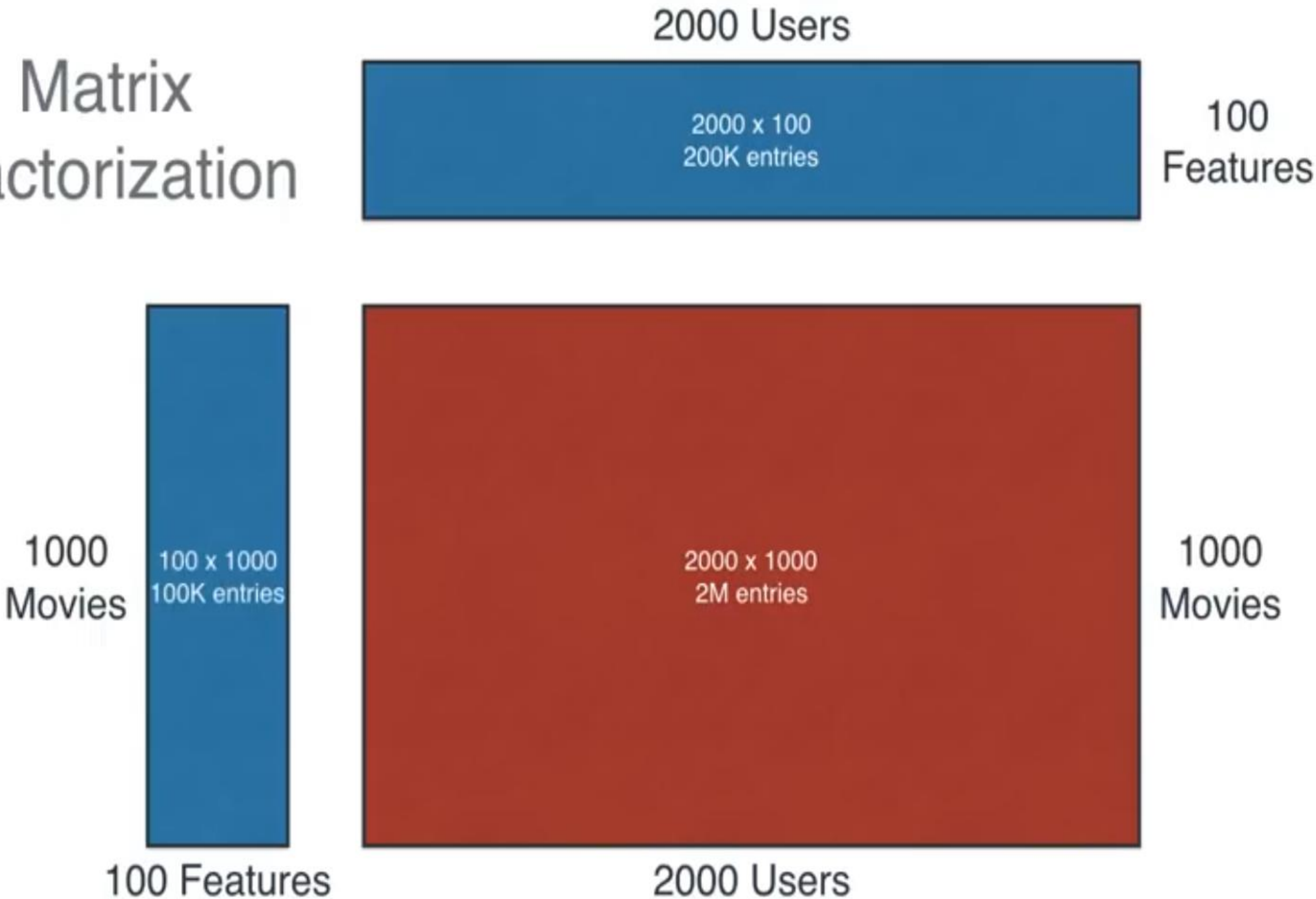
	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

	Comedy	Action
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4


1000
Movies
2000 Users

Matrix Factorization



	M1	M2	M3	M4	M5
F1	1.2	3.1	0.3	2.5	0.2
F2	2.4	1.5	4.4	0.4	1.1

	F1	F2
 A	0.2	0.5
 B	0.3	0.4
 C	0.7	0.8
 D	0.4	0.5

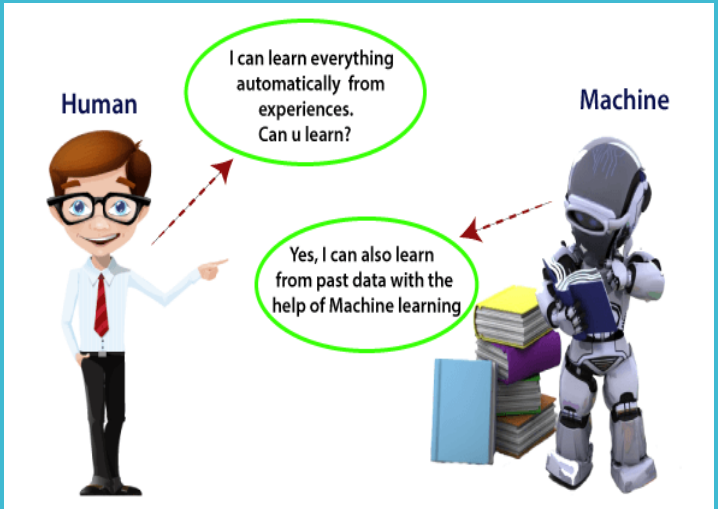
	M1	M2	M3	M4	M5
	1.44	1.37	2.26	0.7	0.59
	1.32	1.53	1.85	0.91	0.5
	2.76	3.37	3.73	2.07	1.02
	1.68	1.99	2.32	1.2	0.63

	M1	M2	M3	M4	M5
F1	3	1	1	3	1
F2	1	2	4	1	3

	F1	F2
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
A	3		1		1
B	1		4	1	
C	3	1		3	1
D		3		4	4

Linear Transformation

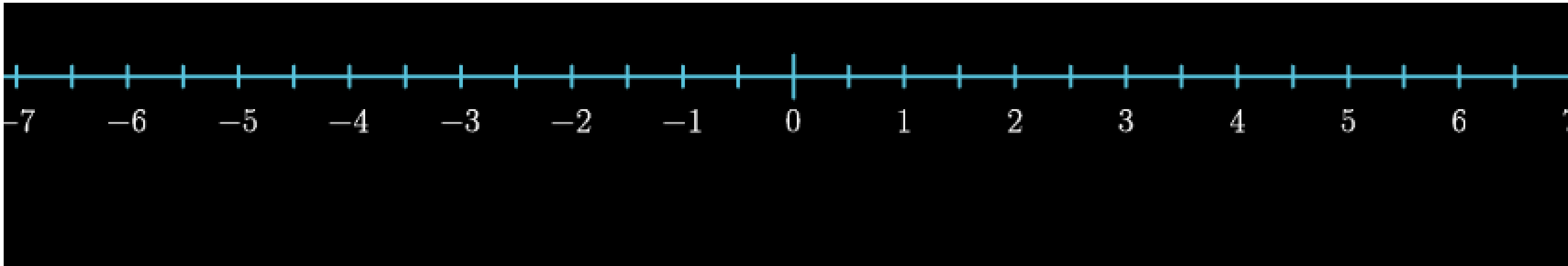


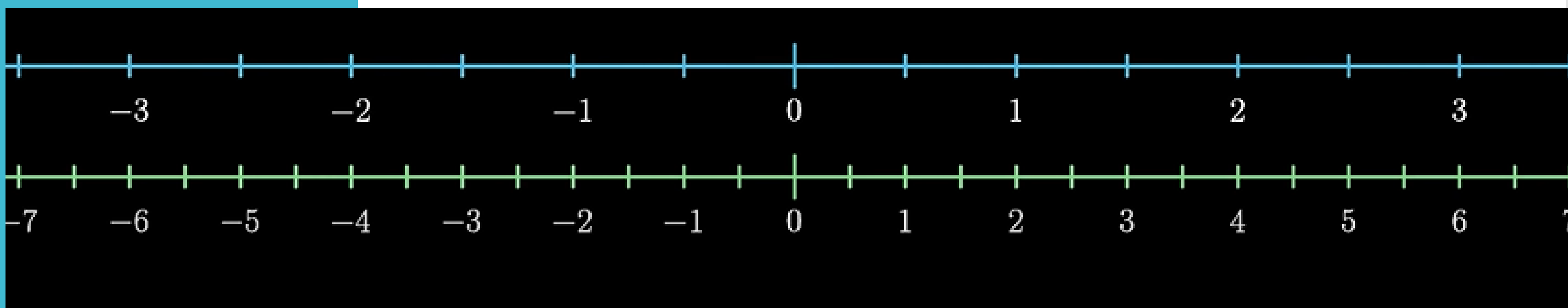
Department of Information and Communication Technology

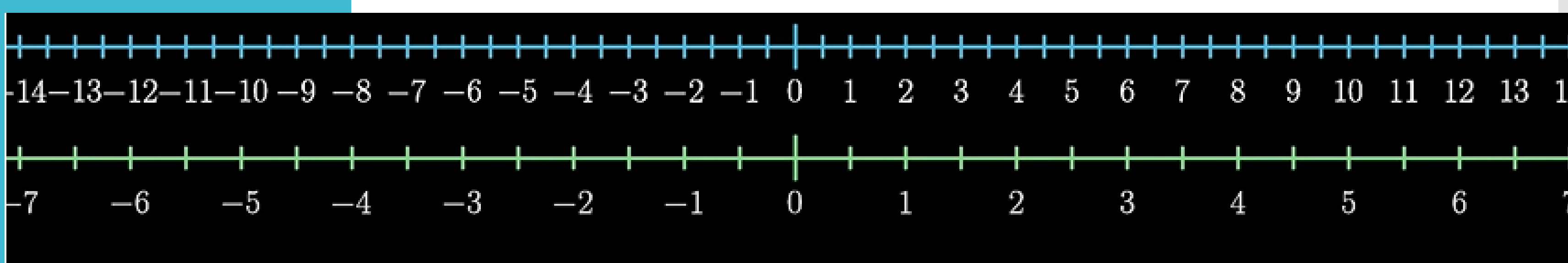
Unit 10: Recommendation Systems

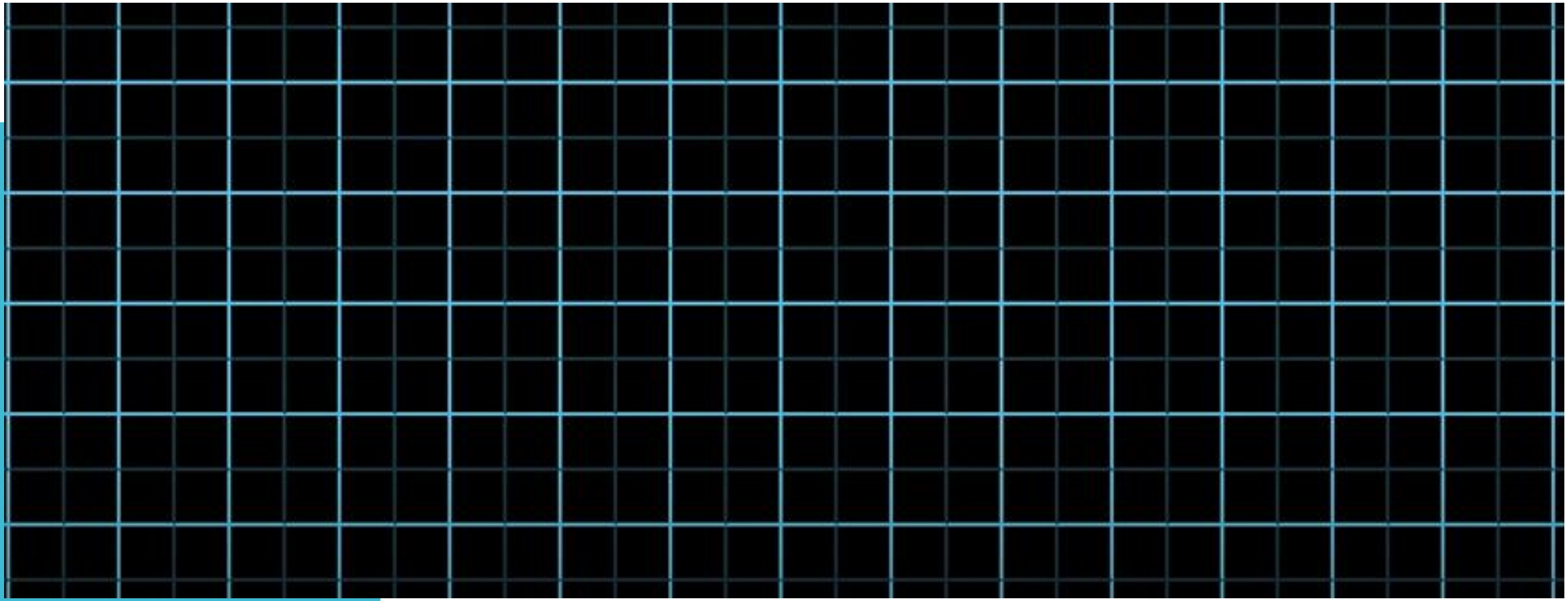
Artificial Intelligence (01CT0703)

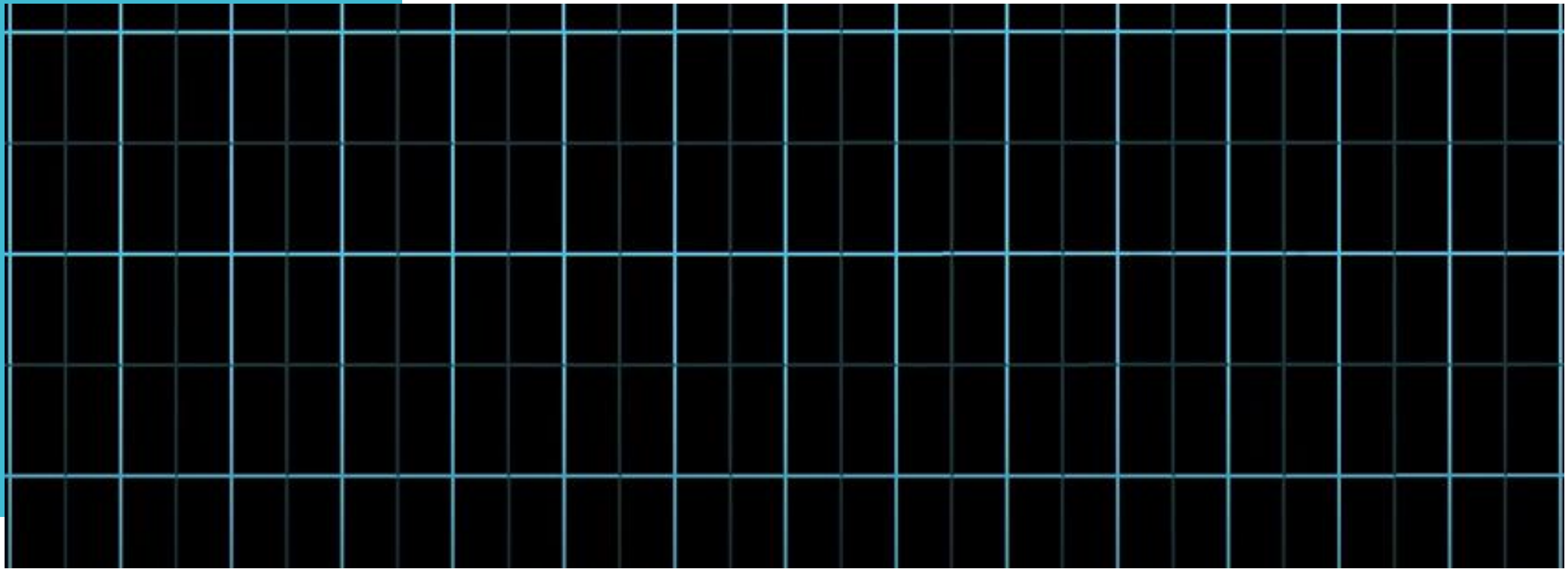
Prof. Nishith Kotak

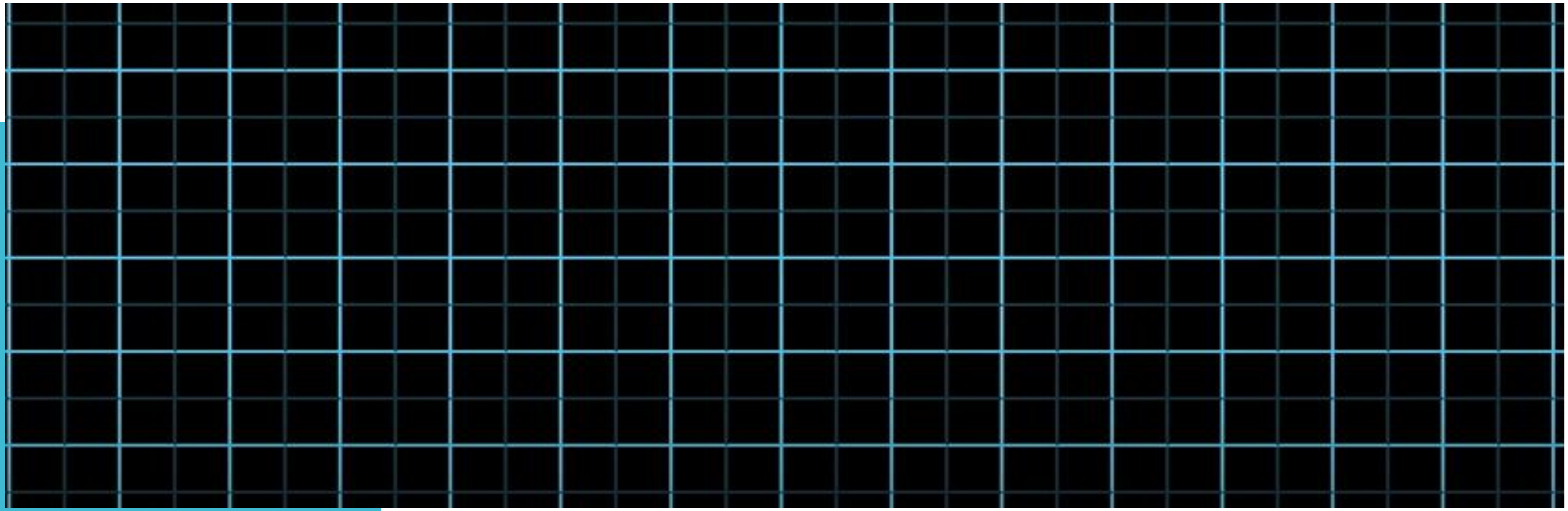

−7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7

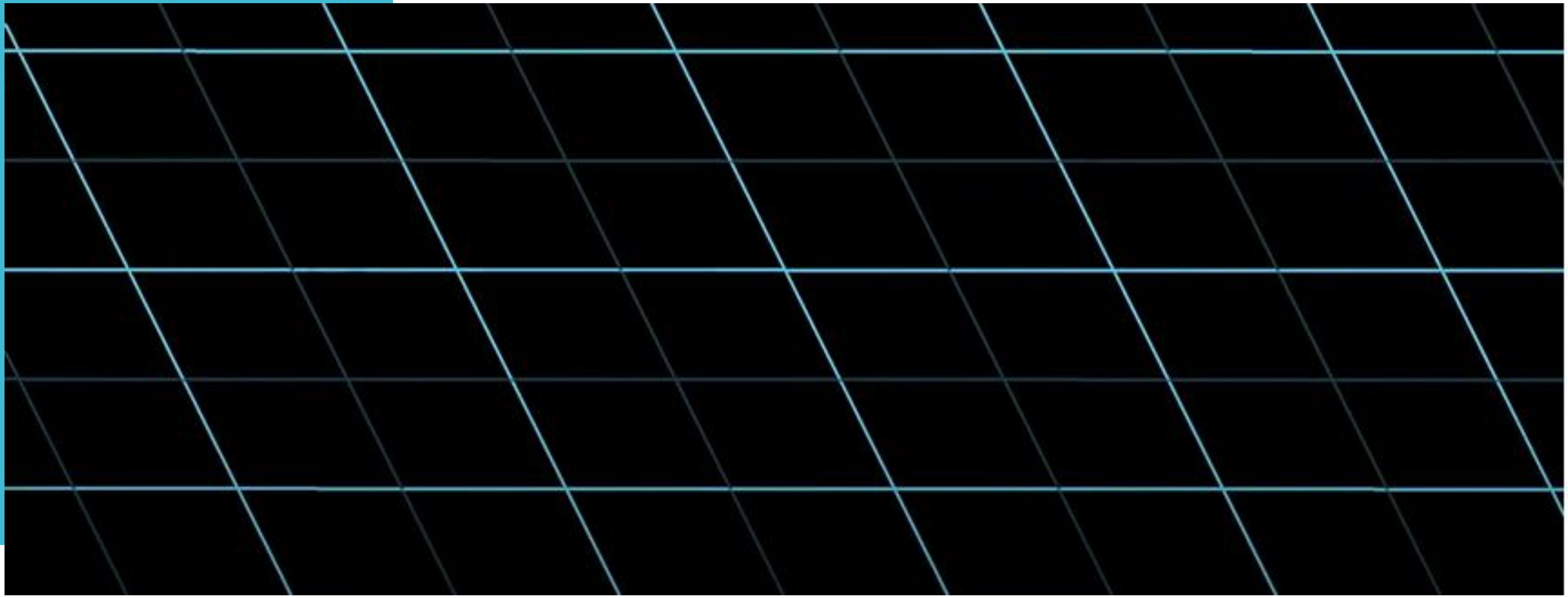

−3 −2 −1 0 1 2 3
−7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7


−14 −13 −12 −11 −10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
−7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7







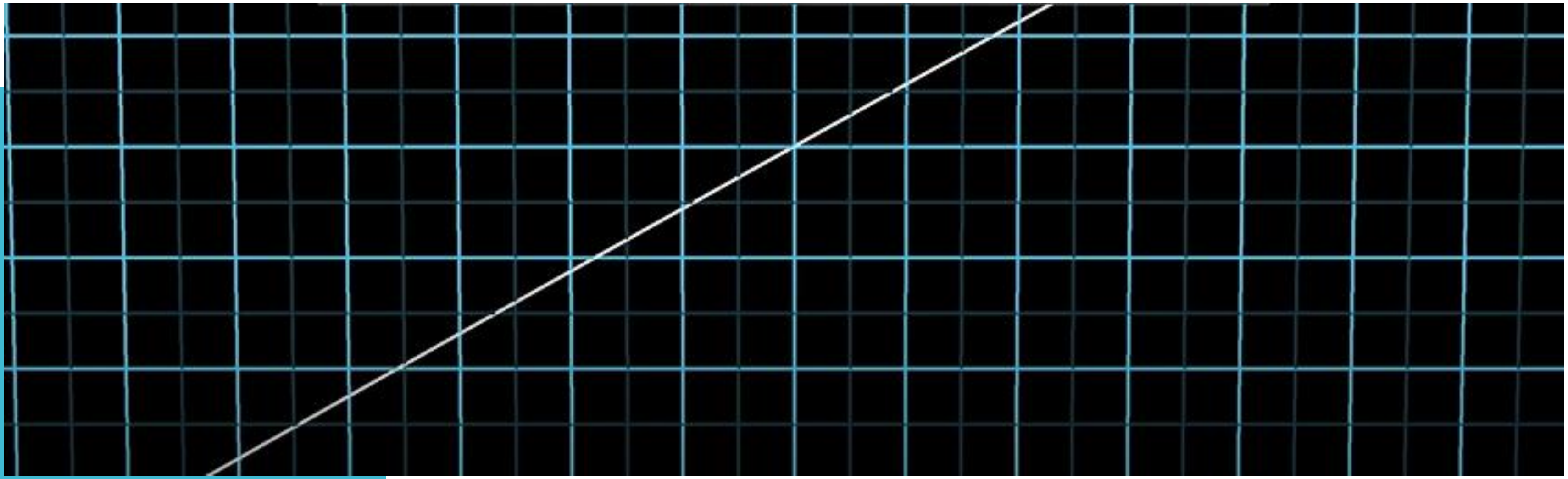


Linearity Property

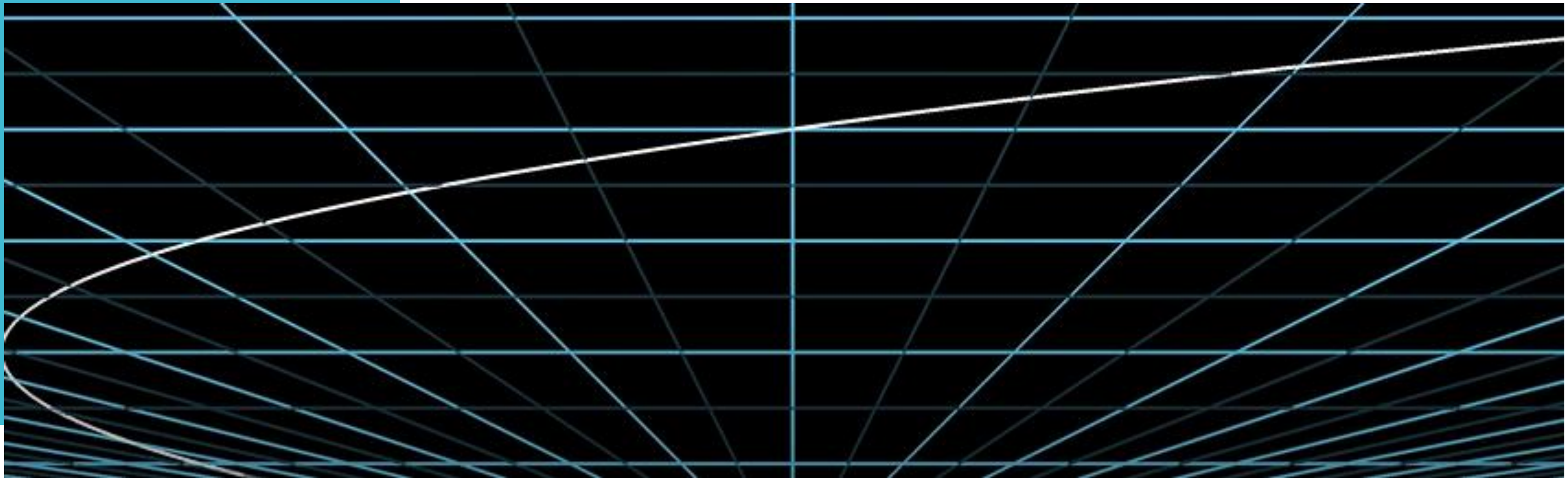
what makes a transformation linear is the following geometric rule: The *origin must remain fixed*, and *all lines must remain lines*.

$$f(\mathbf{v}+\mathbf{w}) = f(\mathbf{v}) + f(\mathbf{w})$$

$$f(c\mathbf{v}) = cf(\mathbf{v})$$



Non-Linear Transformation



https://youtu.be/XUw95PFP1RE

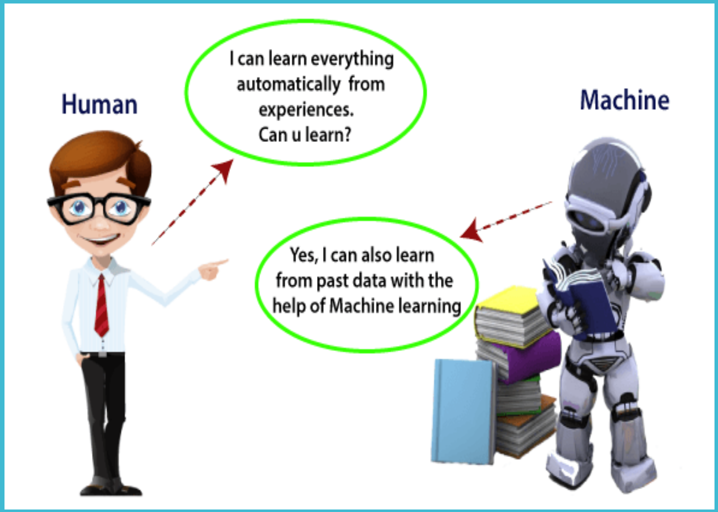
2-D Linear Transformation

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

In this matrix, the first *column* tells us where $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ lands, and the second *column* tells us where $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ lands. Now we can describe where any vector $\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix}$ lands very compactly as the matrix-vector product

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}.$$

Singular Value Decomposition



Marwadi University

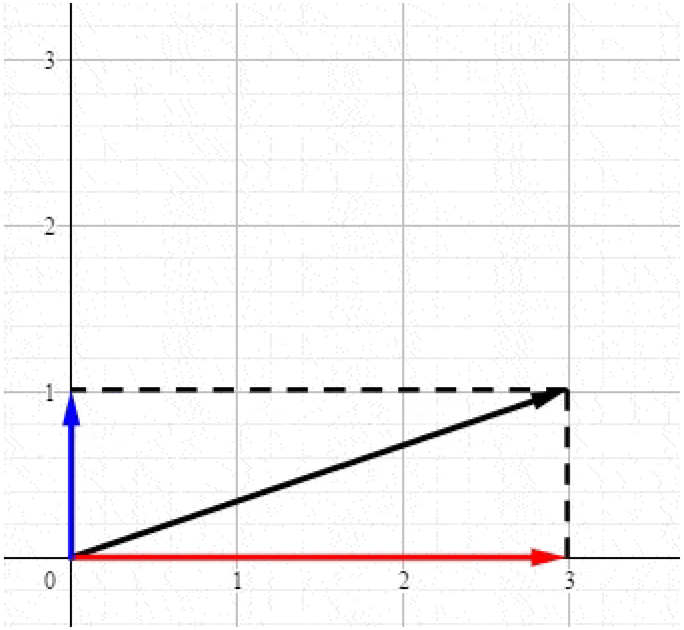
Department of Information and Communication Technology

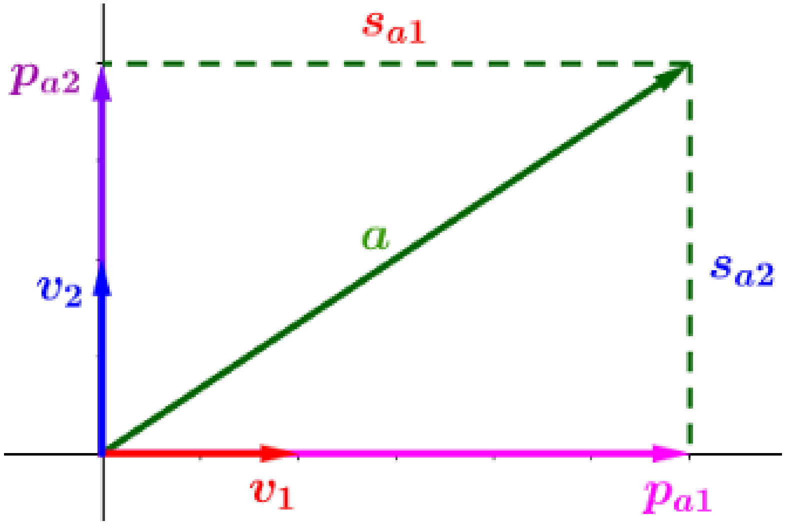
Unit 10: Recommendation Systems

Artificial Intelligence (01CT0703)

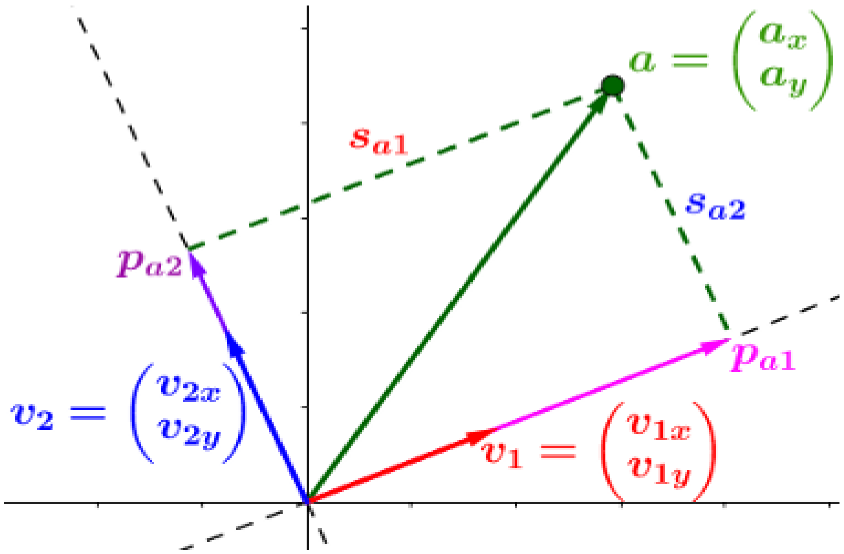
Prof. Nishith Kotak

SVD Projection





SVD Projection



Projecting a on v1 and v2

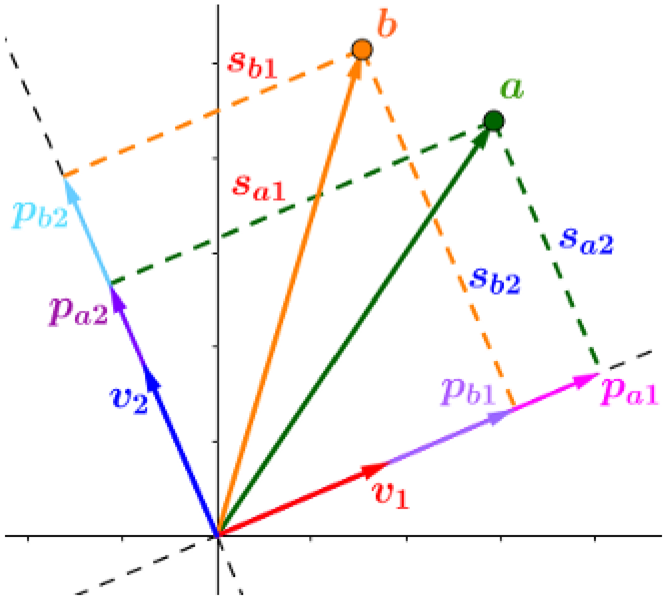
$$a^T \cdot v_1 = \begin{pmatrix} a_x & a_y \end{pmatrix} \cdot \begin{pmatrix} v_{1x} \\ v_{1y} \end{pmatrix} = s_{a1}$$

$$a^T \cdot v_2 = \begin{pmatrix} a_x & a_y \end{pmatrix} \cdot \begin{pmatrix} v_{2x} \\ v_{2y} \end{pmatrix} = s_{a2}$$

Combining both the axis projections, v1 and v2:

$$a^T \cdot V = \begin{pmatrix} a_x & a_y \end{pmatrix} \cdot \begin{pmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{pmatrix} = \begin{pmatrix} s_{a1} & s_{a2} \end{pmatrix}$$

SVD Projection



$$a^T \cdot v_1 = \begin{pmatrix} a_x & a_y \end{pmatrix} \begin{pmatrix} v_{1x} \\ v_{1y} \end{pmatrix} = \begin{pmatrix} s_{a1} \end{pmatrix}$$

$$A \cdot V = \begin{pmatrix} a_x & a_y \\ b_x & b_y \end{pmatrix} \cdot \begin{pmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{pmatrix} = \begin{pmatrix} s_{a1} & s_{a2} \\ s_{b1} & s_{b2} \end{pmatrix} = S$$

SVD Projection

$$A \cdot V = S$$

Matrix of points

The dot product performs the projection

Matrix of decomposition axes

Matrix of the lengths of projections

Because **V** contains orthonormal columns, its inverse = its transpose (property of orthogonal matrices).

$$A = S\,V^{-1} = S\,V^{T}$$

Any set of vectors (A) can be expressed in terms of their lengths of projections (S) on some set of orthogonal axes (V).

SVD Projection

$$S = \begin{pmatrix} s_{a1} & s_{a2} \\ s_{b1} & s_{b2} \end{pmatrix}$$

$$\text{Magnitude of 1st column} = \sigma_1 = \sqrt{(s_{a1})^2 + (s_{b1})^2}$$

$$\text{Magnitude of 2nd column} = \sigma_2 = \sqrt{(s_{a2})^2 + (s_{b2})^2}$$

$$S = \begin{pmatrix} \frac{s_{a1}}{\sigma_1} & \frac{s_{a2}}{\sigma_2} \\ \\ \frac{s_{b1}}{\sigma_1} & \frac{s_{b2}}{\sigma_2} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} = \underset{\displaystyle\downarrow \atop \displaystyle U}{\begin{pmatrix} u_{a1} & u_{a2} \\ u_{b1} & u_{b2} \end{pmatrix}} \underset{\displaystyle\downarrow \atop \displaystyle \Sigma}{\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}}$$

SVD Projection

$$A \cdot V = S$$

Matrix of points

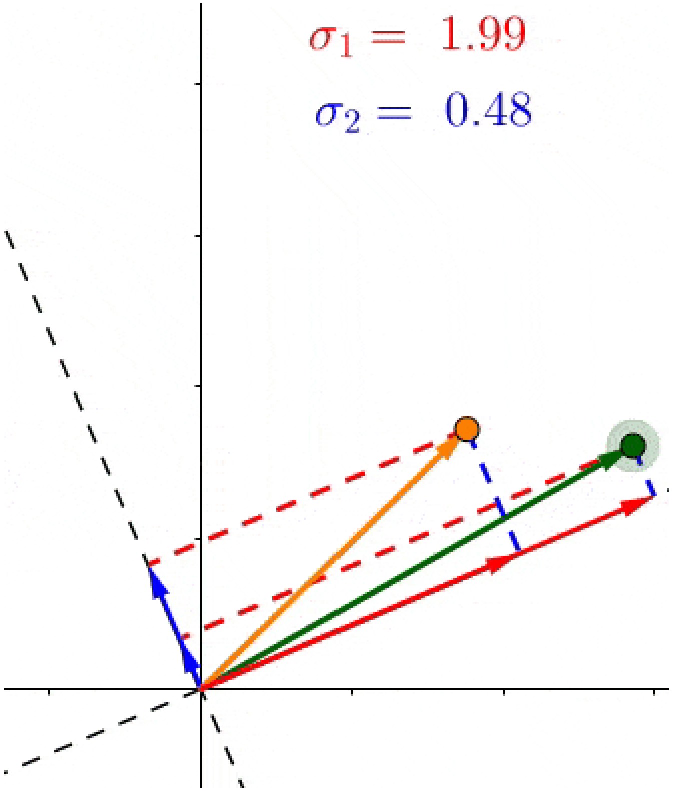
The dot product performs the projection

Matrix of decomposition axes

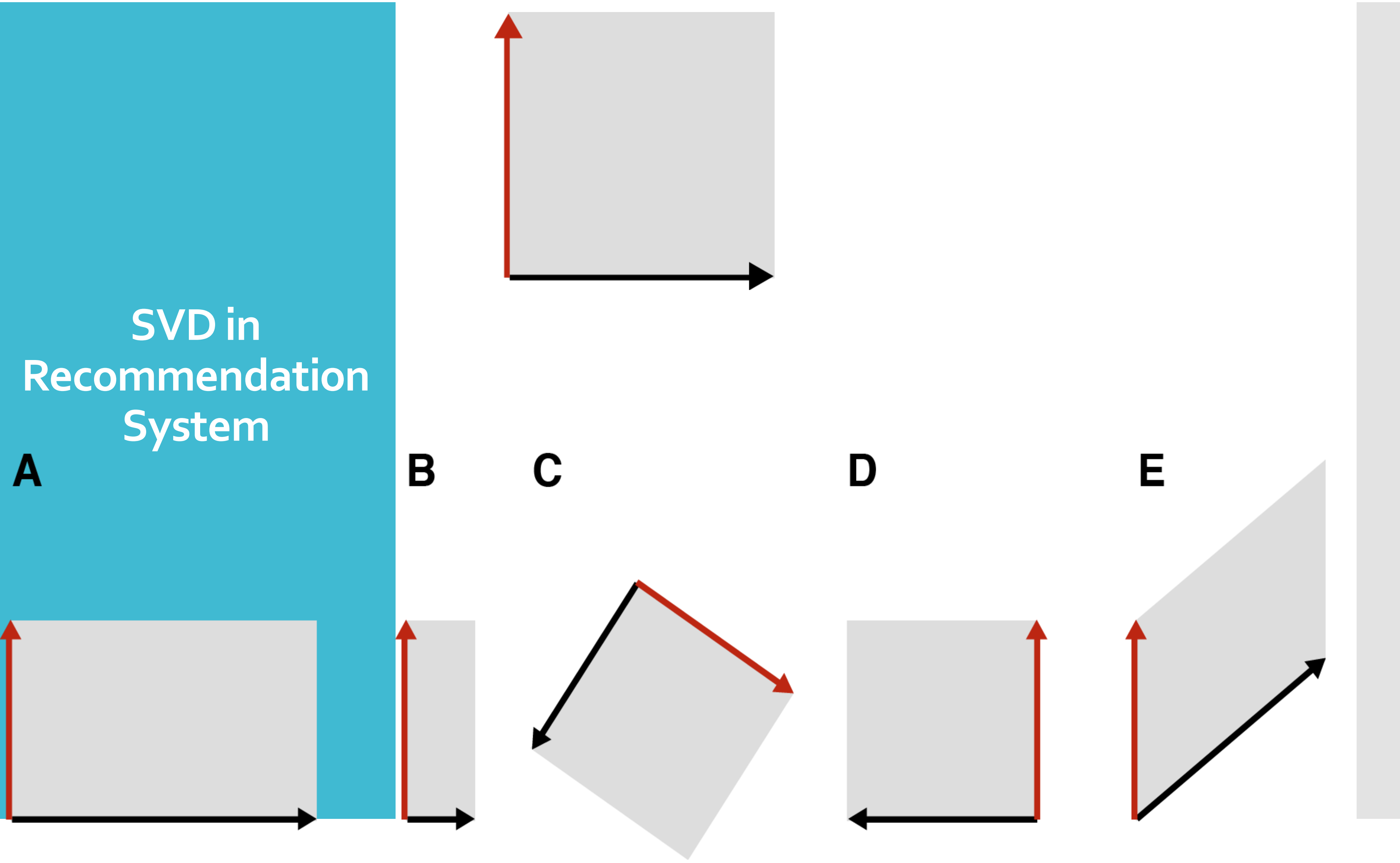
Matrix of the lengths of projections

$$A = U \Sigma V^T$$

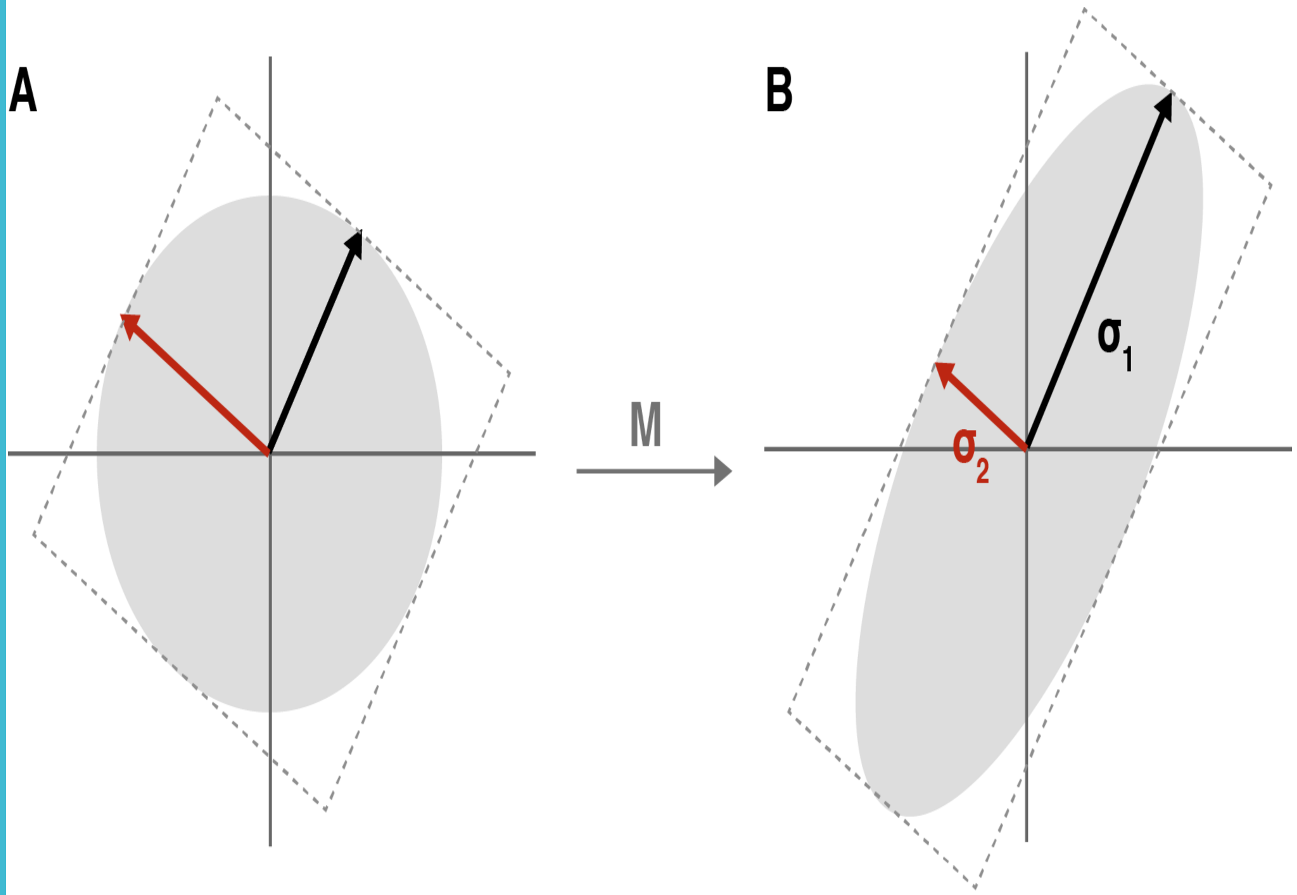
SVD Projection



SVD in Recommendation System



SVD in Recommendation System



SVD in Recommendation System

The factorisation of this matrix is done by the singular value decomposition. It finds factors of matrices from the factorisation of a high-level (user-item-rating) matrix.

The singular value decomposition is a method of decomposing a matrix into three other matrices.

$$A = USV^T$$

Where A is a m x n utility matrix,

U is a m x r orthogonal left singular matrix, which represents the relationship between users and latent factors,

S is a r x r diagonal matrix, which describes the strength of each latent factor and

V is a r x n diagonal right singular matrix, which indicates the similarity between items and latent factors.

Recommendation System

Refer Apply Recommendation.ipynb

	Harry Potter	Star Wars	Tomb Raider
User 1	3	1	
User 2	3		4
User 3	1	4	
User 4		2	1