DATE 26 12 2024

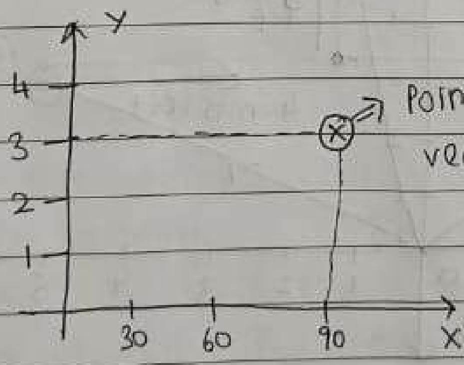
SCALARS AND VECTORS

⇒ scaler is a single numerical value. It represents a magnitude or quantity and has no direction.
In a dataset, count, mean, median, average etc are scalers since they are single numerical values and no direction.

⇒ vector is a numerical value which has both magnitude and direction. In computer science field, a vector can be defined as an ordered list of numbers, which can represent a point in space or quantity with both magnitude and direction

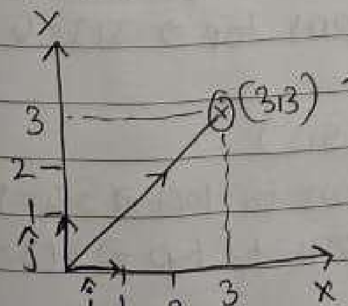
Application of vectors in data science :-
Let us take a student dataset for ~~fea~~ understanding which contains IQ, no. of study hours and result as features.

A vector representing IQ and no. of study hours = [90, 3hrs] } 2 dimensions



⇒ Point representing the vector [90,3hrs] ⇒ [x,y]

⇒ Unit vectors have a magnitude of 1.



→ can be represented as $3\hat{i}+3\hat{j}$

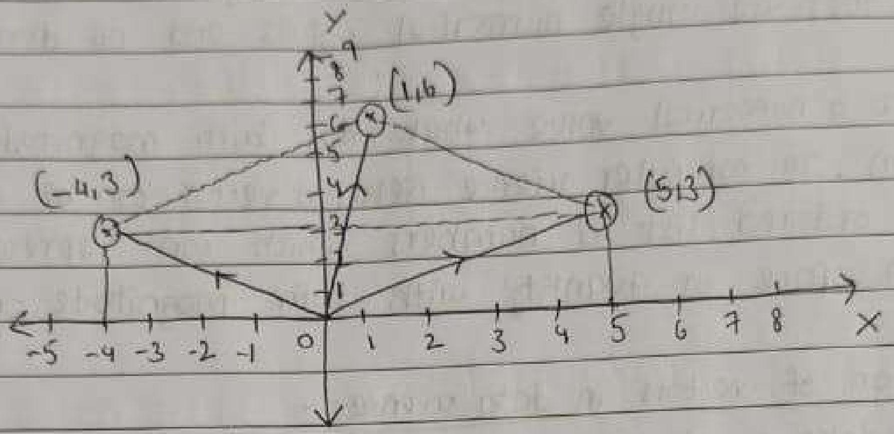
$\hat{i}$ ⇒ unit vector along X-axis

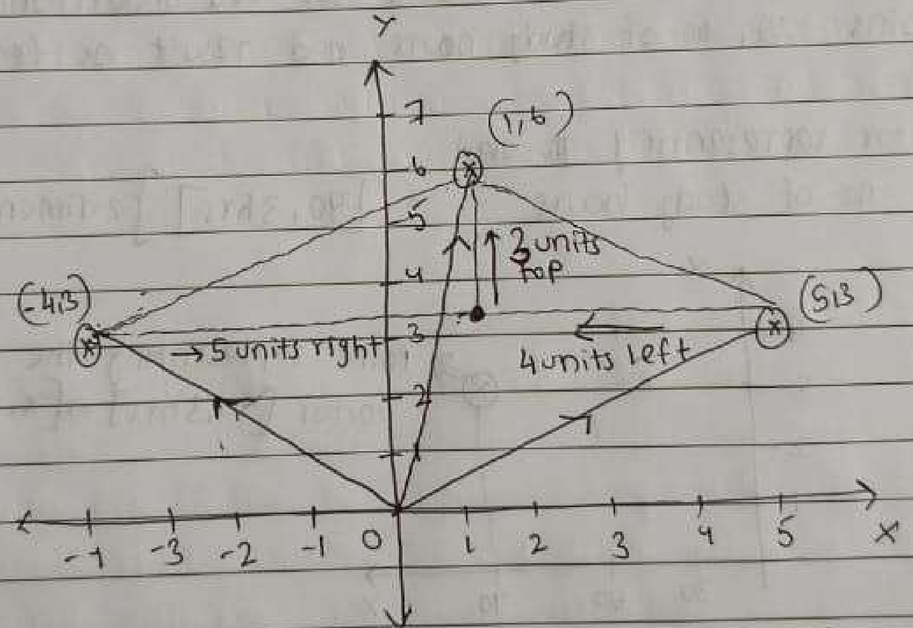
$\hat{j}$ ⇒ unit vector along Y-axis

ADDITION OF VECTORS

$\Rightarrow \quad P_1 = \begin{bmatrix} -4 \\ 3 \end{bmatrix}, P_2 = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$

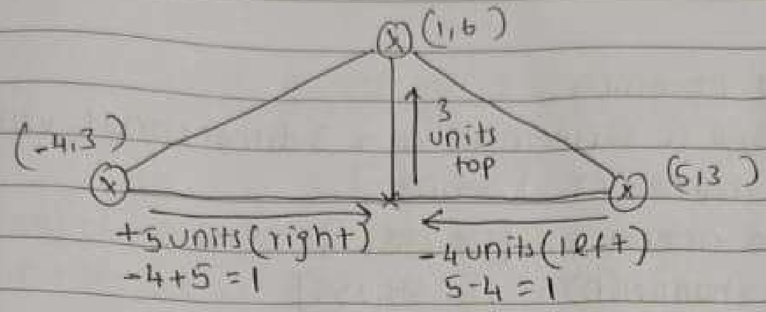
$P_1 + P_2 = \begin{bmatrix} -4 \\ 3 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} -4+5 \\ 3+3 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \end{bmatrix}$





At position (-4,3), when we add (5,3),
In x direction ⇒ we move 5 positions to right ⇒ 5-4 = 1
In y direction ⇒ we move 3 positions top ⇒ 3+3 = 6

similarly, when we add (-4,3) to (5,3)
In x direction ⇒ we move 4 positions to left ⇒ 5-4 = 1
In y direction ⇒ we move 3 positions to top ⇒ 3+3 = 6



→ **Applications of vector addition:-**

(i) Data aggregation tasks

(ii) Feature engineering

(iii) In Natural language Processing, when we perform tasks like sentiment analysis.

Ex:

Review	sentiment
The product is good	1
The product is bad	0

$\boxed{\text{Review}}$ → (Model) → sentiment

The model cannot understand text information. so we convert the text into vectors

To convert text into vectors, we require different techniques

(i) one hot encoding
(ii) TF-IDF
(iii) Bag of words
(iv) word2vec

} These techniques convert every sentence in the text to some number of vectors.

(iv) word embeddings:-

suppose, the word data is represented by vector $[0.2, 0.1, 0.4]$

the word science is represented by $[0.3, 0.7, 0.2]$

then,

$$\text{Data science} = V_{Data} + V_{science} = [0.2, 0.1, 0.4] + [0.3, 0.7, 0.2]$$
$$= [0.5, 0.8, 0.6]$$

we use vector addition to combine words

DATE 26 12 2024

⑤ In image processing,
Every image is represented as a 3-dimensional vector [R,G,B]
If red channel(R) = [255,128,0]
green channel (G) = [128,255,0]
blue channel(B) = [64,64,255]

~~Grey scale (RGB) = $(V_R + V_G + V_B)/3$ = [255+128+64, 128+255+64, 0+0+255]~~
~~= [447, 447, 255]~~
~~Grey scale = Average~~

Grey scale(RGB) = $(V_R + V_G + V_B)/3$

$$= \left[\frac{255+128+64}{3}, \frac{128+255+64}{3}, \frac{0+0+255}{3}\right] = [149, 149, 85]$$

To obtain black and white image (grey scale), we perform vector addition of red, blue and green scale vectors and find average of them.

VECTOR MULTIPLICATION

⇒ 3 types of vector multiplication
(i) Dot product
(ii) Element wise multiplication
(iii) Scalar multiplication

⇒ **Dot product:-**
The dot product of 2 vectors results in a scalar and is calculated as sum of products of their corresponding components.

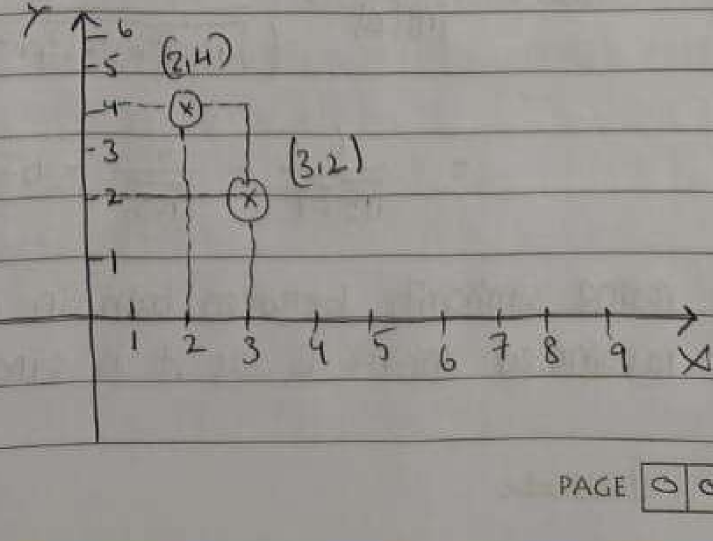
$A = [2 \;\; 3]$ $\qquad$ $B = [4 \;\; 5]$

$$\text{Dot product } (A\cdot B) = \underbrace{(2\times4)}_{\text{product of } x \text{ components}} + \underbrace{(3\times5)}_{\text{product of } y \text{ components.}} = \underbrace{23}_{\text{scaler value}}$$

$$\underbrace{A\cdot B^T = [2 \;\; 3]\cdot\begin{bmatrix}4\\5\end{bmatrix} = (2\times4)+(3\times5)}_{} = 23$$

Actual matrix multiplication. So, if we want to find dot product, we should transpose one of the vectors if we want to find it out using matrix multiplication.

Ex:-
$A = [2,4]$
$B = [3,2]$
$A\cdot B = (2\times3)+(4\times2) = 14$



→ Applications of dot product in data science:-

In GenAI applications like RAG, we use cosine similarity. cosine similarity is a measure used to determine how similar 2 vectors are. It provides/calculates the cosine of the angle between 2 vectors, providing a similarity score that ranges from -1 to +1.

-1 ⇒ completely dissimilar
+1 ⇒ completely similar

$$\cos\theta = \frac{\overbrace{A \cdot B}^{\text{Dot product}}}{|A||B|}$$

In movie recommendation as well, we do use cosine similarity to recommend new movies.
suppose, if we saw a movie A with representation:-

[horror, violence, comedy, drama] = [2, 1, 0, 0]

Then we try to find cosine similarity of other movies with this movie for recommendation.

If cosine similarity is close to 1, then we recommend the movie, otherwise not.

so, if we have 2 movies,

$A = [1, 2, 0, 3, 1]$
$B = [2, 0, 1, 1, 1]$

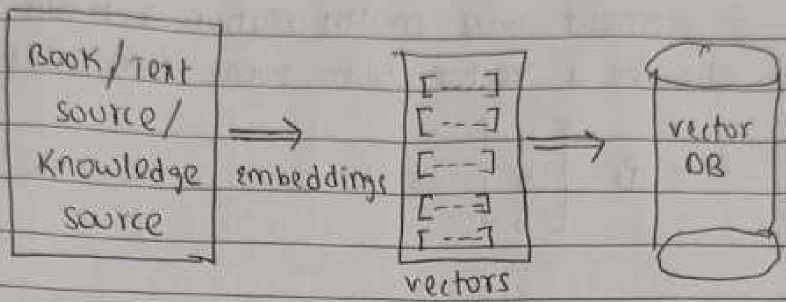
$$\cos\theta = \frac{A \cdot B}{|A||B|} = \frac{(1\times2)+(2\times0)+(0\times1)+(3\times1)+(1\times1)}{\left(\sqrt{1^2+2^2+0^2+3^2+1^2}\right)\left(\sqrt{2^2+0^2+1^2+1^2+1^2}\right)}$$

$$= \frac{6}{\sqrt{15}\sqrt{7}} = \frac{6}{\sqrt{105}} = 0.586$$

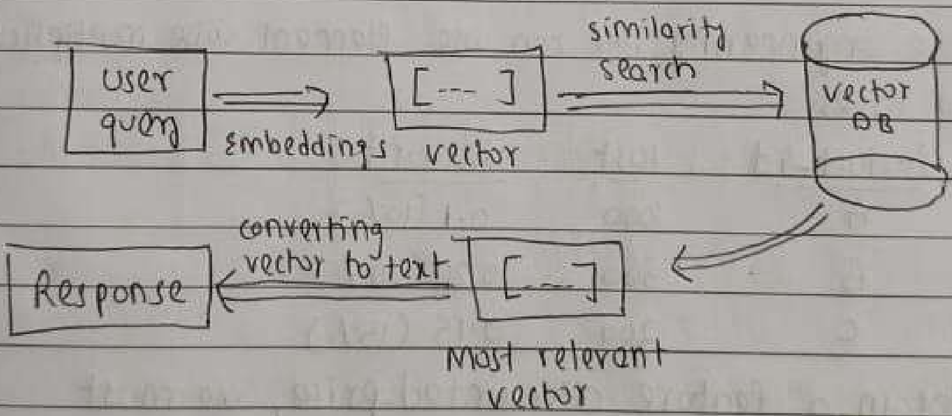
cosine similarity between both in near 1, so we can recommed movie B as it is similar to A.

→ Another important application of vectors :- (vector databases)

when we build a RAG application (Retrieval Augmented Generation), we convert huge file into vectors and store them into vector database.



when a user sends a query, the query will be embedded on converted into a vector. Then similarity search will be performed on query vector and vectors in vector database to find most relevent vector and the most relevant vector will be sent back to user after converting to text.



Dot product of vectors is used to find cosine similarity, which will be used in RAG applications to find most relevent/similar data, from vector database.

④ Element wise multiplications:-

In element wise multiplication, corresponding elements of two vectors are multiplied to form a new vector of same dimension.
The result of element wise multiplication between two vectors will also be a vector with same dimensions

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad B = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

$A \otimes B$ = Element wise multiplication of A and B

$$= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \otimes \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1\times4 \\ 2\times5 \\ 3\times6 \end{bmatrix} = \begin{bmatrix} 4 \\ 10 \\ 18 \end{bmatrix}$$

Application of element wise multiplication:-

In feature engineering, we can use element wise multiplication in some cases.

Ex:

Product-id	cost	discount
A	1000	0.1 (10%)
B	500	0.20 (20%)
C	200	0.15 (15%)

To obtain a feature discounted-price, we must perform element-wise multiplication of cost and discoun

discounted-price = cost ⊗ discount

$$= \begin{bmatrix} 1000 \\ 500 \\ 200 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.2 \\ 0.15 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 30 \end{bmatrix}$$

similarly, to obtain final price, we can perform element wise subtraction

Product-id	cost	discount	discounted-Price
A	1000	0.1	100
B	500	0.2	100
C	200	0.15	30

final price = cost $\ominus$ discounted-Price

$$= \begin{bmatrix} 1000 \\ 500 \\ 200 \end{bmatrix} - \begin{bmatrix} 100 \\ 100 \\ 30 \end{bmatrix} = \begin{bmatrix} 900 \\ 400 \\ 170 \end{bmatrix}$$

In Deeplearning, we have architectures of RNN like LSTM RNN and GRU RNN, where we can find applications of element wise multiplication, element wise addition etc.

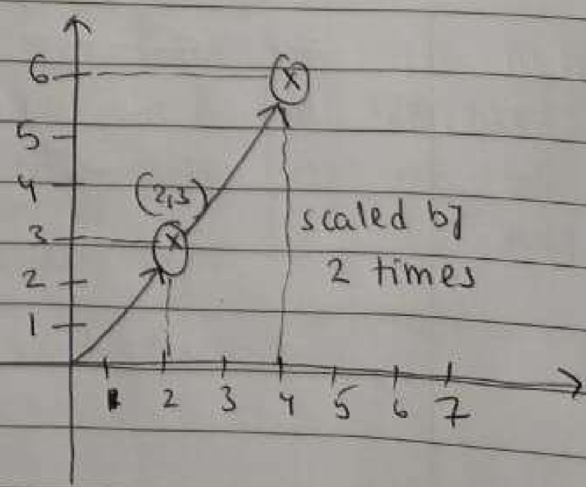
(i) scalar multiplication:-

when we multiply a vector with a scalar, each component of the vector will be scaled by scalar.

$$A = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, B = 3 \qquad A = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad B = 2$$

$$AB = \begin{bmatrix} 2\times3 \\ 3\times3 \\ 4\times3 \end{bmatrix} = \begin{bmatrix} 6 \\ 9 \\ 12 \end{bmatrix} \qquad AB = \begin{bmatrix} 2\times2 \\ 3\times2 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Resultant of scalar multiplication i.e product of a vector and scalar will be a vector with same dimensions.



Applications of scalar multiplication:-

In normalization technique, we use scalar multiplication to reduce complex computations. It has application in image processing.

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} \end{bmatrix} \xrightarrow{\text{normalization}} \begin{bmatrix} a_0/255 & a_1/255 & a_2/255 \\ -- & -- & -- \\ -- & -- & -- \\ a_9/255 & a_{10}/255 & a_{11}/255 \end{bmatrix}$$

~~All pixel values will be in range~~ 0-255

All pixel values will be in range 0-1

Similarly, we perform scalar multiplication for scaling purposes. For example, to convert cms to ms

$$A = \begin{bmatrix} 170cm \\ 150cm \\ 180cm \end{bmatrix} \qquad \text{scale-factor} = 0.01$$

$$B = A \cdot \text{scale-factor} = \begin{bmatrix} 170 \times 0.01 \\ 150 \times 0.01 \\ 180 \times 0.01 \end{bmatrix} = \begin{bmatrix} 1.7m \\ 1.5m \\ 1.8m \end{bmatrix}$$