DIMENSIONALITY REDUCTION

Dimensionality Reduction

Two methods

Feature Selection (lasso, chi-square selection)

Feature Extraction (PCA, kernel PCA, latent semantic analysis)

⇒ Curse of Dimensionality

→ PCA (Principal Component Analysis) → comes under unsupervised

high dimension → low dimension (without loosing info)

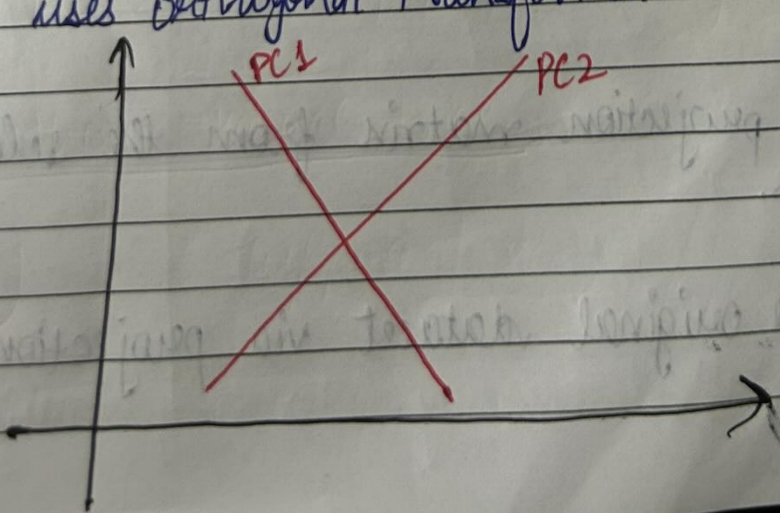
PCA is the analysis where we have lower loss function & high variance in complex data.

⇒ We delete the features having lower loss function & high variance to avoid overfitting.

⇒ Here in this we usually find which features are affecting most to the data.

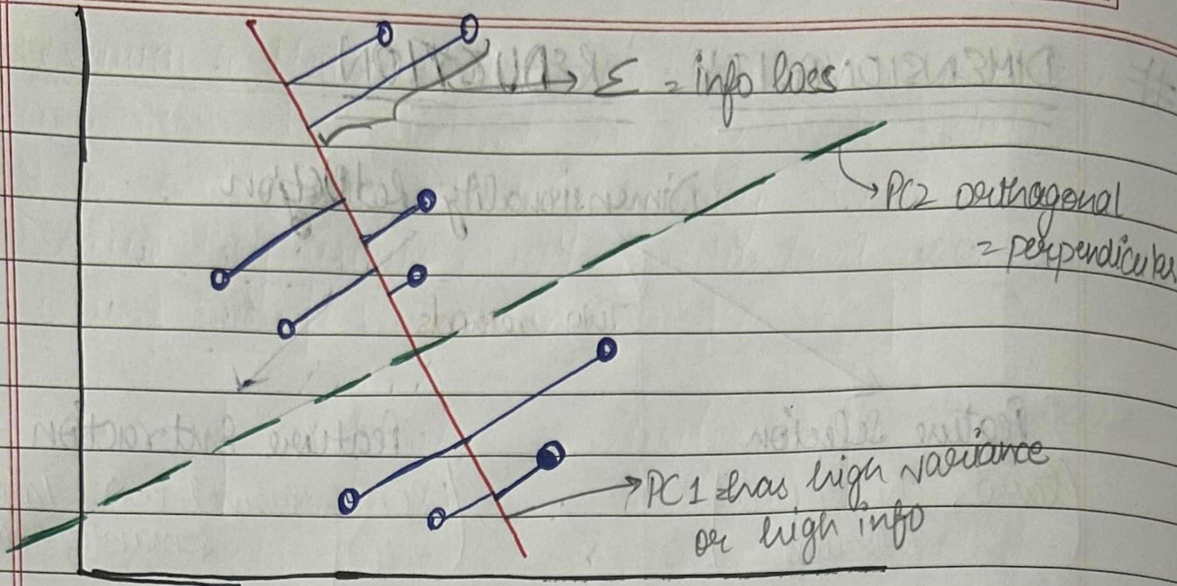
⇒ It is a very useful method for extracting relevant information from confusing dataset.

⇒ It uses orthogonal transformation.



Orthogonal dimension dimensions will change but features will not change

Date / /
Page No.



⇒ Goals -
1) To identify pattern in data
2) Detect correlation b/w variables
3) Attempts to reduce dimensionality

PCA Approach

1) Standardize the data $\left\{ \begin{matrix} \mu = 0 \\ \sigma = 1 \end{matrix} \right\}$
(because we have to bring random variables to normal Gaussian distance)

2) Perform singular vector decomposition to get eigen vectors & eigen values.

3) Sort eigen values in descending order & choose the K-eigen vectors.
denote them in matrix form

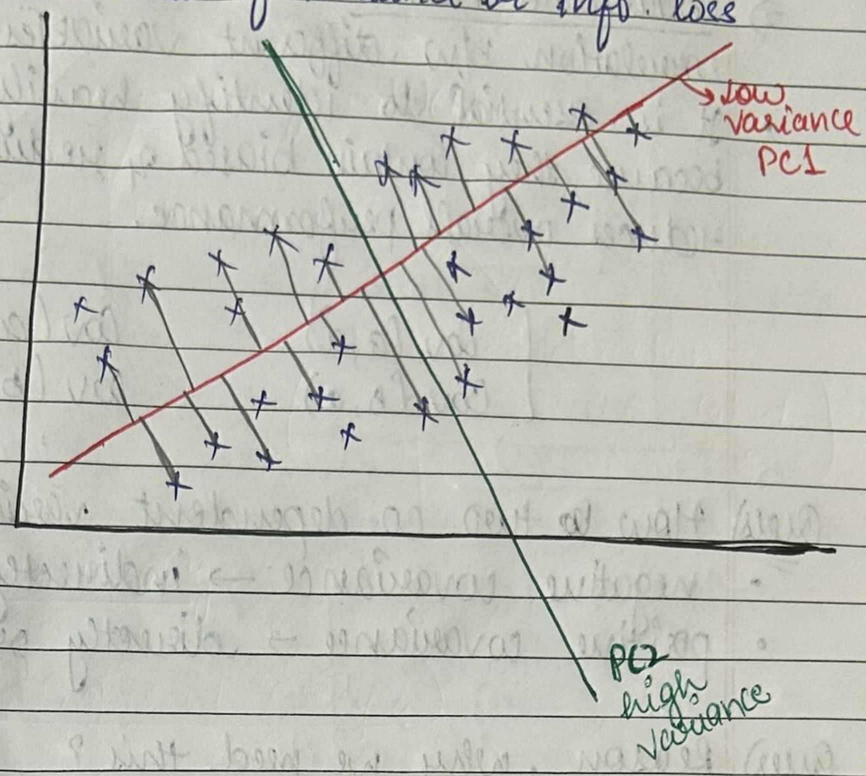
4) Construct the projection matrix from the selected k eigen vectors.

5) Transform the original dataset via projection matrix.

Date / /
Page No.

→ There should be less no. of variance or info. loss

SNDC ($\mu = 0, \sigma = 1$)
Rescale the values on small scale.



→ Principal Component Analysis - is a dimensionality reduction technique that enables you to identify correlations & patterns in data set so that it can be transformed into a dataset of lower dimension with any loss of important information

→ If two features are highly correlated then they will have high bias.

→ Standardization
↓
Computing the covariance matrix
↓
Calculate the eigen vectors
↓
Computing principal component
↓
Reducing dimensions

relation b/w & diff. datasets

⇒ <u>Covariance Matrix</u> - A covariance matrix expresses the correlation b/w different variables in the dataset.
- It is essential to identify heavily dependent variables because they contain biased & redundant info which reduces overall performance.

$$\begin{bmatrix} Cov(a,a) & Cov(a,b) \\ Cov(b,a) & Cov(b,b) \end{bmatrix}$$

Ques) How two co-dependent variables are
- negative covariance → indirectly proportional
- positive covariance → directly proportional

Ques) Reason, why we need this?
⇒ Eigen vectors & Eigen values so that we can determine principal component of dataset.

⇒ Principal component?
↓
New set of variables obtained from initial set after we perform dimensional reduction
↓
These components have to be highly significant for predicting output & also independent of each other
↓
They compress & they possess most of the useful info that is actually scattered among all different variables.
↓
after all PC's (PC1, PC2, ----)
↓
arrange them in descending order & you can just take

Covariance only gives direction
Correlation gives strength & direction

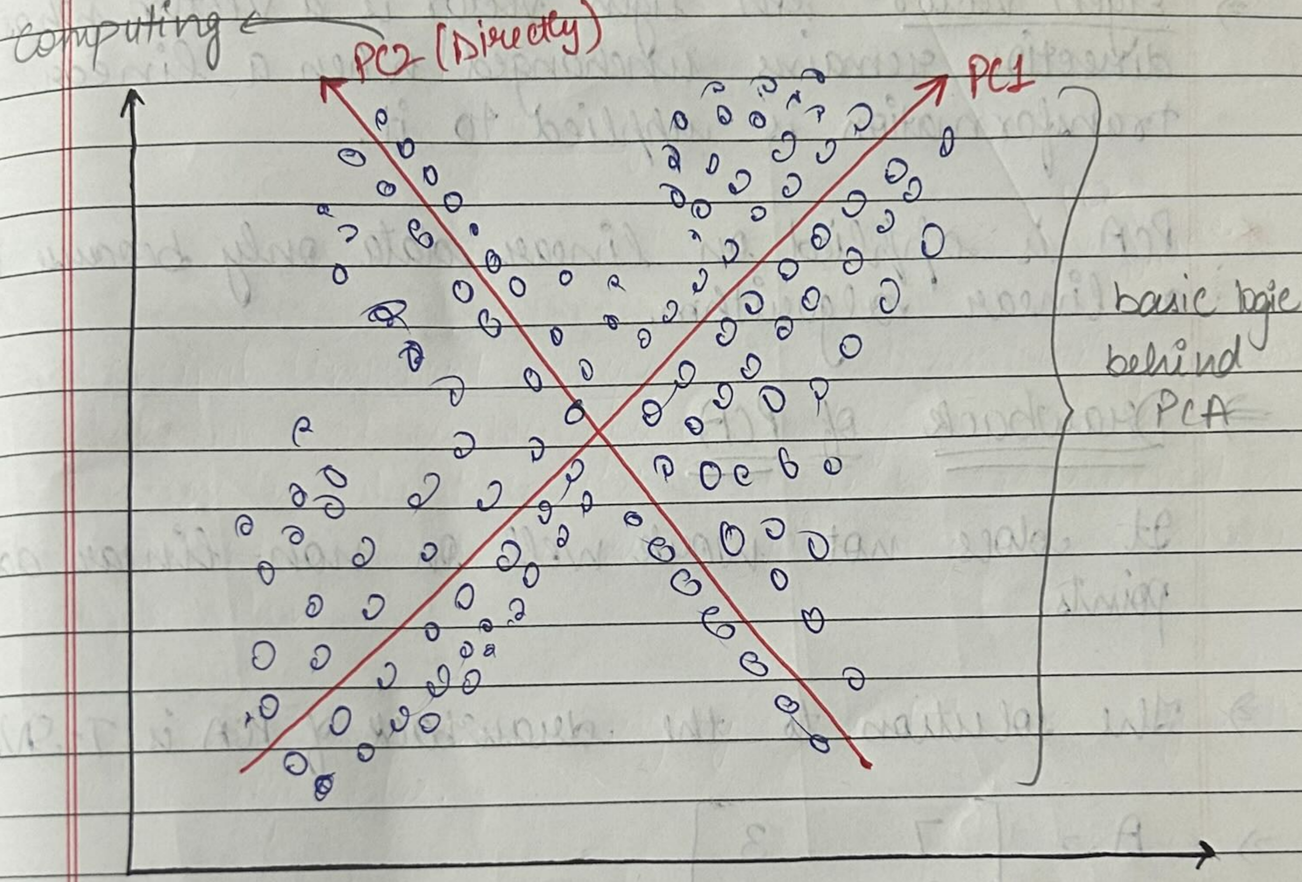
Date / /
Page No.

important ones into your analysis.

no. of dimensions = no. of PC's.

→ Eigen values & Eigen vectors → To understand where in the data where is most amount of variance.

* $\boxed{\text{More variance in data = more info in data}}$



⇒ PC 1 is the most significant & stores the max possible info

⇒ PC2 remaining & so on.

Date / / Page No.

23/2/23

⇒ Dimensionality Reduction

PCA
(feature extraction karta hai)

T-SNe
t-Distributed Stochastic Neighbour Embedding

⇒ Eigen Vector - An eigen vector is a vector whose direction remains unchanged when a linear transformation is applied to it.

* PCA is applied on linear data only because it a linear algorithm.

⇒ Drawback of PCA

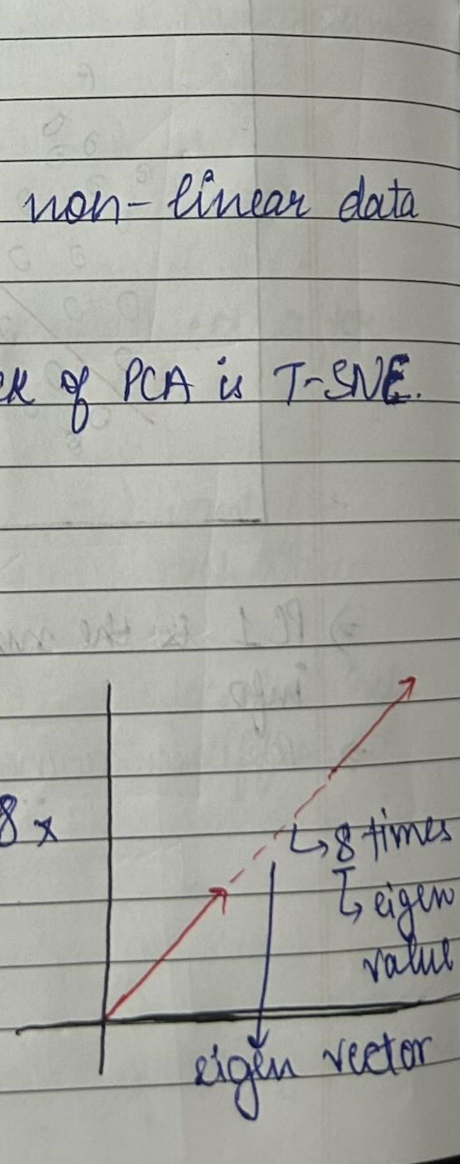
It does not work well on non-linear data points.

⇒ The solution to the drawback of PCA is T-SNE.

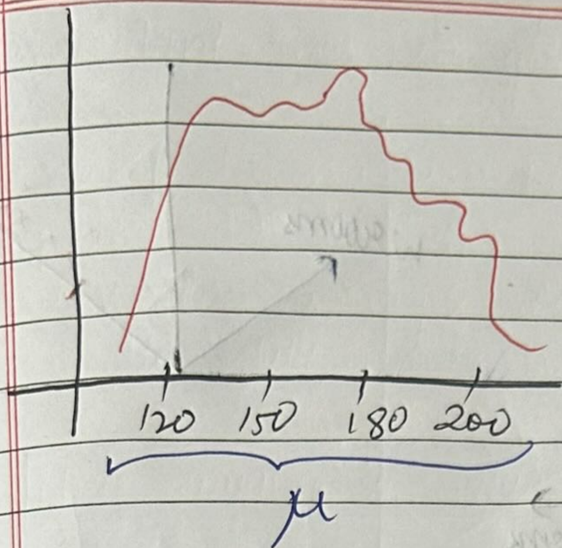
⇒ $A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix}$

Vector, $\vec{x} = 3(i) + 1(j)$

$\begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 8x$



Date / /
Page No.



{PCA approach}

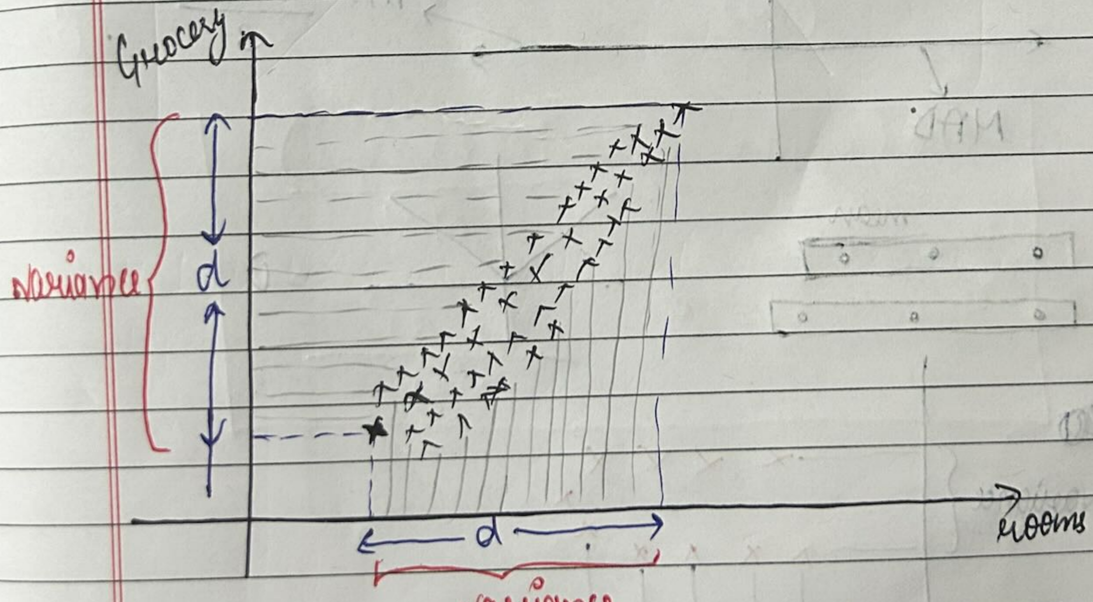
in ① step standarisation, we have to make $\mu = 0$ & $\sigma = 1$

* In PCA mein Principal component
we have select the value.

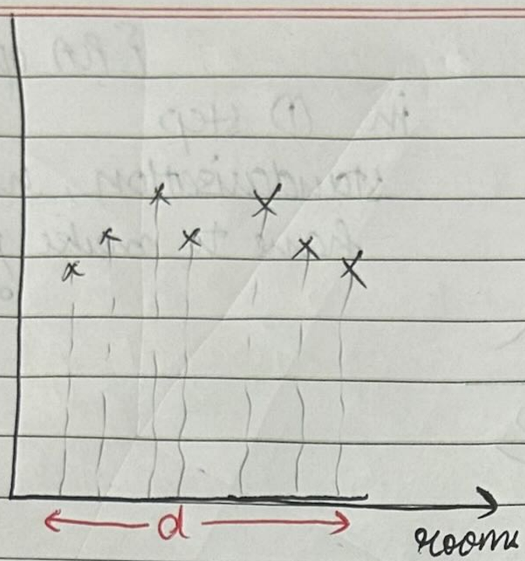
24/2/23

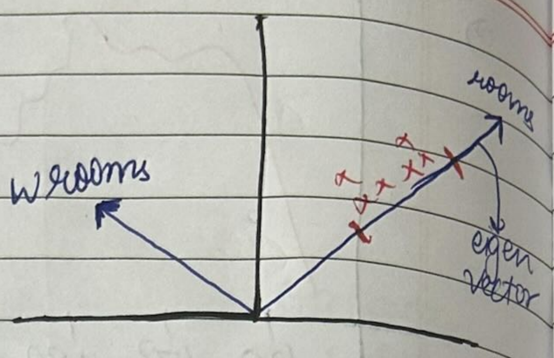
→ Geometric Intution

rooms	Grocery shop	L
3	2	60
4	0	130
5	6	170
2	10	90

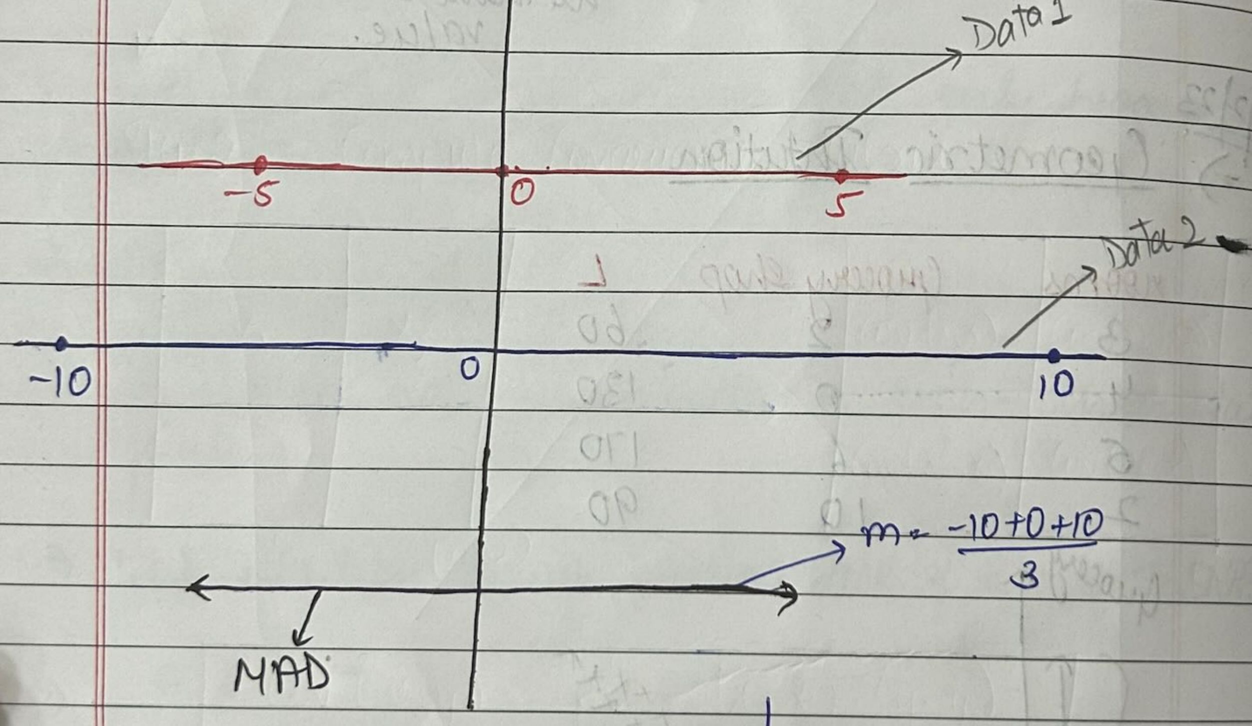


Variance - It is a statistical technique through which we can tell how much data is spread

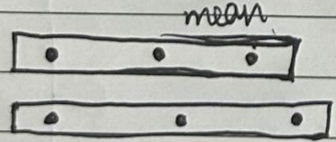


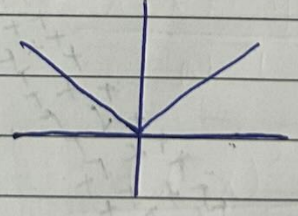


Ques) Why variance is important?

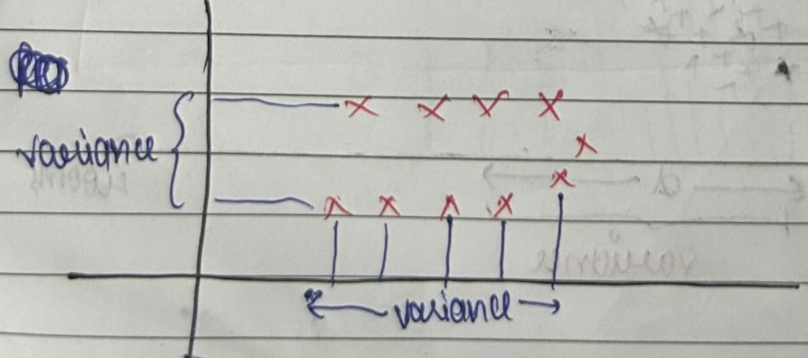


$m = \frac{-10+0+10}{3}$





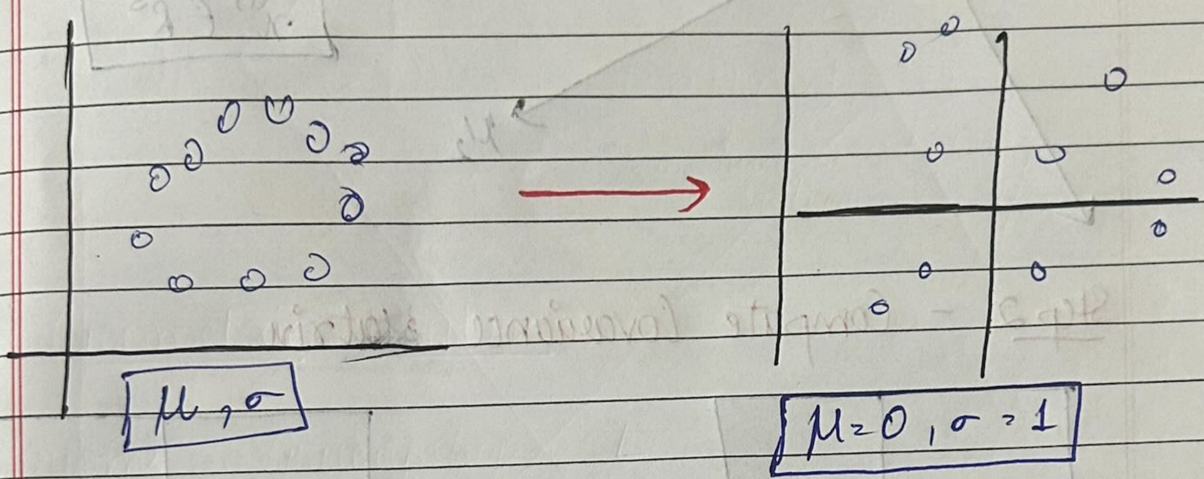
$= 0$



* PCA finds new set of co-ordinates
→ It will form new axis, by rotating other axis
→ In PCA, we have to find max variance.

→ **Steps Involved in PCA** -

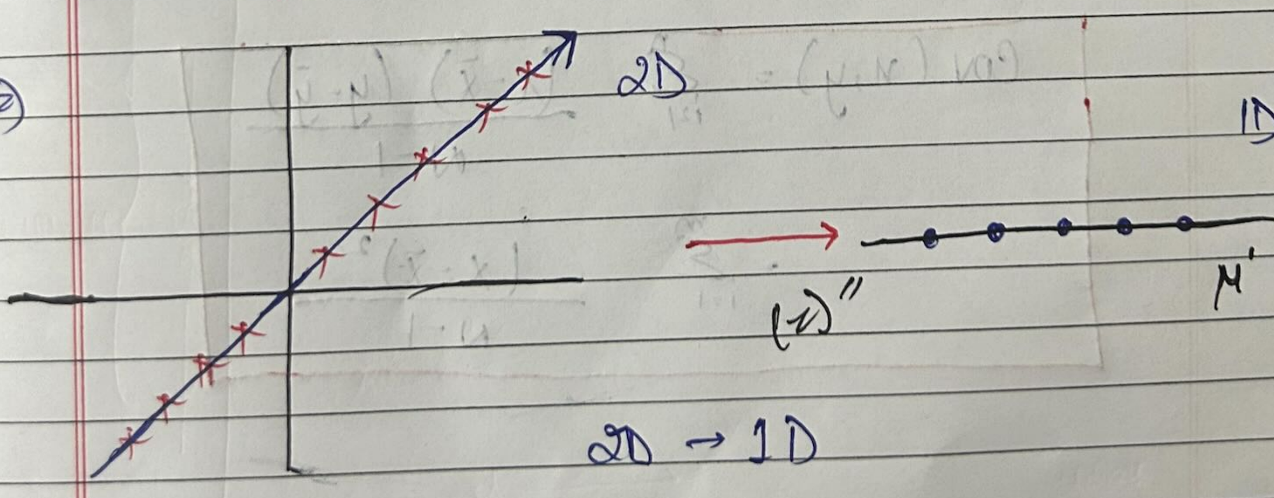
Step 1 - Standardize
↳ sklearn {standard scalar ()}



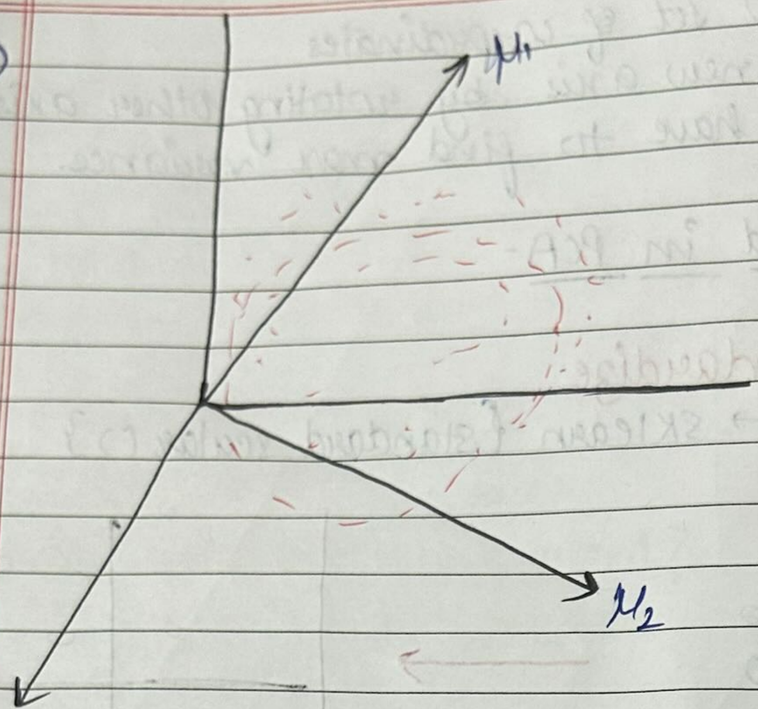
→ The formula used to make $\mu = 0$ & $\sigma = 1$ is

$$\boxed{x' = \frac{x - \bar{x}}{\sigma}}$$

2)



→



3D
↓
2D

$\boxed{x^{(i)} \in R^3}$

Step 2 - Compute Covariance Matrix

X_1	X_2

$(m \times 2)$ →

cov (x,x)	cov (x₁,x₂)
cov (x₂,x₁)	cov (y,y)

(2×2)

$$\mathrm{cov}(x,y) = \sum_{i=1}^{n} \frac{(x-\bar{x})(y-\bar{y})}{N-1}$$

$$= \sum_{i=1}^{m} \frac{(x-\bar{x})^2}{N-1}$$

Date / /
Page No.

Step 3 - Apply SVD on Covariance Matrix
↳ Singular value Decomposition

$\mu^{(i)} \longrightarrow$ compute eigen vectors of covariance

SVD returns μ, $\underbrace{S, V}_{\text{not required}}$

$\mu \downarrow$ eigen values

$\{SVD(cov) = \mu, S, V\}$

~~[illegible]~~ Mathematics behind SVD-
The SVD of (m×n) matrix A is given by formula:

$$\boxed{A = UWV^T}$$

where,
U = (m×n) matrix of the orthogonal eigen vectors of AA^T
V^T = transpose of a (n×n) matrix containing the orthonormal eigenvectors of A^{T} A
W = a (n×n) diagonal matrix of the singular values which are square roots of the eigen values of A^TA.