· **PCA.** (Principle Component Analysis)

↳ unsupervised ML Algorithm.

↳ A feature extract^n technique

↳ It is used used to reduce ~~essence~~ dimension of data while keeping essence of data

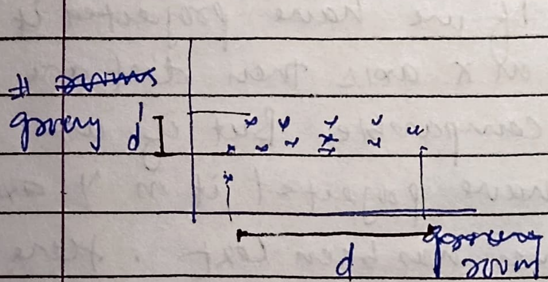
· Benifits of using it -

1) faster exec^n of algo
2) Visualizatn

· Geometric Intution -

I/P		O/P
No. of rooms	No. of grocery shop	Price
3	2	60
4	0	130
5	6	170
2	10	90

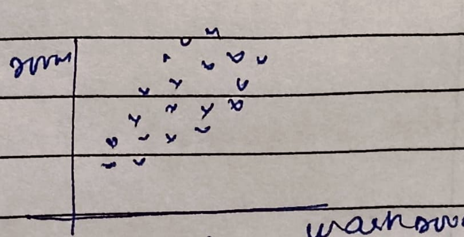
By seeing features we can remove No. of grocery shop feature. But how ~~that~~ to know which column to remove mathematical



variance of room is more. and it doesn't depend on grocery so we will remove grocery.

$d > d'$

·) rooms | washroom | Price

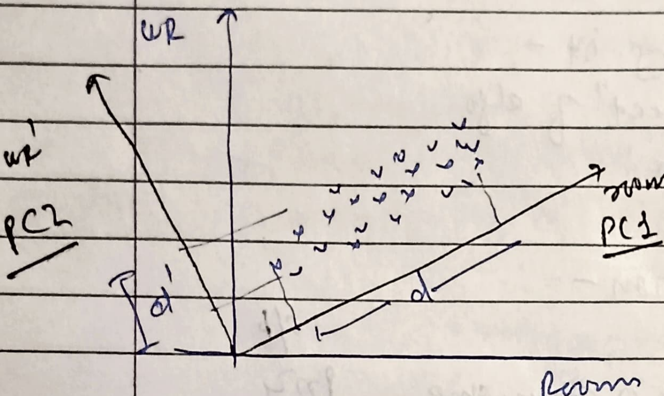


$(d = d')$

Since $d = d'$ So here feature Select^n can't eliminate.

So here we will use feature extractn

→ Since we can't select so we will merge room & washrooms together and make a new feature called Size



Here we have (trans form) rotated the axis and now we have two ~~com~~ PCs.

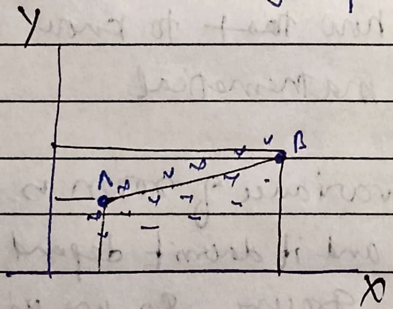
here since $d > d'$

So we will take PC1

always →

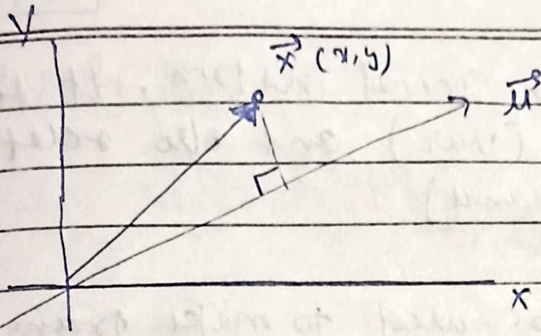
$$\# PC \leq n \qquad n \Rightarrow \# \text{features}$$

- variance is proportional to spread not exactly spread.



we want dist b/w AB

If we have projected it on x axis then dist are comparable But if we have projected it on Y axis then distance has been lost. Here variance will be important



projection of $\vec{x}$ on $\vec{u}$ (unit vector) $= \dfrac{\vec{u} \cdot \vec{x}}{|u|}$

$$= \vec{u} \cdot \vec{x} = u^T x$$

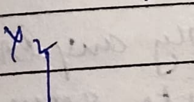
$$\begin{bmatrix} x_1 & y_1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = x_1 x_2 + y_1 y_2 = \vec{u} \cdot \vec{x}$$

→ we want that unit vector on which if we project all the points we get max. variance

$$\text{var} = \frac{1}{n} \sum_{i=1}^{n} \left(u^T x_i - u^T \bar{x} \right)^2$$

⇒ Covariance → measures how two variables change together (direction of relationship)

⇒ Cov matrix



$$\begin{array}{c|cc} & x_1 & x_2 \\ \hline x_1 & \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) \\ x_2 & \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) \end{array}$$

$\text{cov}(x_1, x_1) = \text{var}(x_1, x_1)$ & $\text{cov}(a, b) = \text{cov}(b, a)$

$$= \begin{array}{c} x_1 \\ x_2 \end{array} \begin{bmatrix} \text{var}(x_1) & \text{cov}(x_1, x_2) \\ \text{cov}(x_1, x_2) & \text{var}(x_2) \end{bmatrix}$$

→ It is square, symmetric matrix

Page No.
Date

→ Cov matrix is special matrix. It tells about spread (var) and also relatⁿ b/w axis (covariance)

:) Matrix can be used to make transformatⁿ in the co-ordinate system.
They can rotate, expand etc the co-ordinate system

:) Eigen vectors are vectors on which if we apply linear transformatn then directⁿ of vector is not changed, magnitude may change. [2D → 2 Eigen vectors]

:) Eigen values are factor by which magnitude of Eigen vector changes

:) $A\vec{v} = \lambda \vec{v}$

↑ Eigen vector ↳ Eigen value

↳ This eqⁿ means if we apply any linear transformatn on Eigen vector its same as multiplying Eigen vector with constant

:) If we find the largest Eigen vector of covariance matrix, then directⁿ of that Eigen vector we will have highest variance and that Eigen vector will be PC

· Step by Step Solⁿ :-

1) Do mean Center (means ~~bringt~~ make mean of that data zero)

2) find cov matrix
3) find Eigen value & Eigen vectors

3D → 3 vector

λ_1 λ_2 λ_3

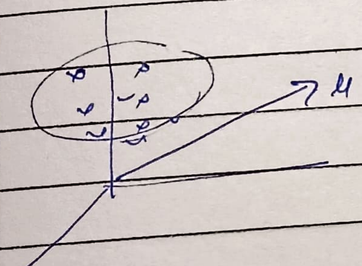
PC1 PC2 PC3

1D 2D

→ After we got Principal Component (PC) we transform our pts to that PC coordinate

⇒ $f_1 | f_2 | f_3 |$ o/p ⇒ PC1 | op
(1000×4) ⇒ (1000, 2)

we do $\underline{u^T \cdot X}$



x = (1000, 3) u = (1, 3)

So ~~dot~~ (1000, 3) (3, 1)
= (1000, 1) → $\underline{PC1}$