1) $\boxed{\text{data}}$

2) $\boxed{\text{EDA}}$ → .ipynb.

3) $\boxed{\text{Preprocessing.}}$

4) model building.

5) evalution.

Null value

outlier.

Imbalanced.

Scaling

$\boxed{\text{F.S.}}$

$\boxed{\text{Dim red.}}$

PCA.

$\boxed{\begin{array}{l}\text{Lin reg} \\ \text{log reg} \\ \text{Svm.}\end{array}}$

Data :- (R x C) → Housing

f_1 f_2 f_3 .... f_{10} → M_1 → Acc → 75 %.

f_1 f_2 f_3 .... f_{20} → M_2 → Acc → 80 %.

f_1 f_2 f_3 .... f_{50} → M_3 → Acc → 85 %.

f_1 f_2 f_3 .... f_{100} → M_4 → Acc → 70 %.

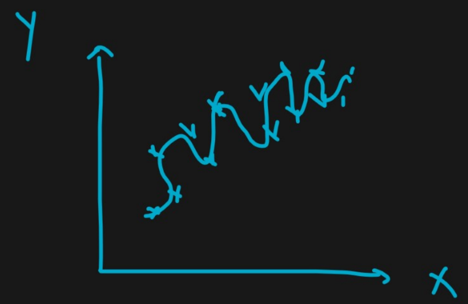
Training accuracy will be high but test accuracy will be low due to over fitting accuracy decreases

because we used features which are not really related to output

Curse of dimensionality → So many feature. → ~~ML~~

100 → Linear reg →

→ overfitting →



feature selection → select subset of the feature.

Dim Red. → transform your data into low dim.

$f_1 \quad f_2 \quad f_3 \quad f_4 \quad \ldots \quad f_{100}$

↓

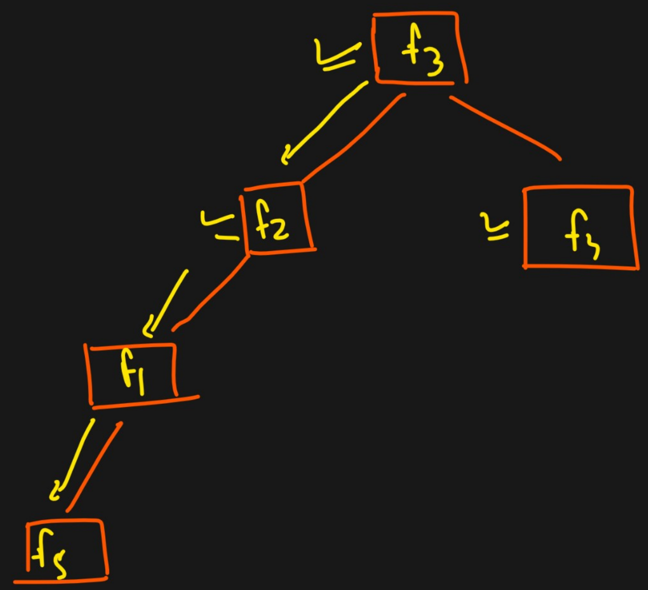
Linearreg. (any model)

$$\left\{ \begin{array}{l} \left. \begin{array}{l} f_1 \longrightarrow f_{10} \\ f_1 \longrightarrow f_{20} \\ f_1 \longrightarrow f_{50} \end{array} \right\} \begin{array}{l} \rightarrow 90 \\ \rightarrow 80 \\ \rightarrow 50 \end{array} \end{array} \right\} X$$

1) corr.
2) Chisqure.
3) ANOVA
4) backward elimation.
5) feature importance

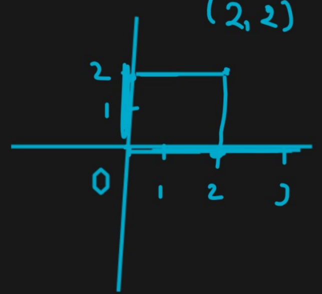
Decision tree.

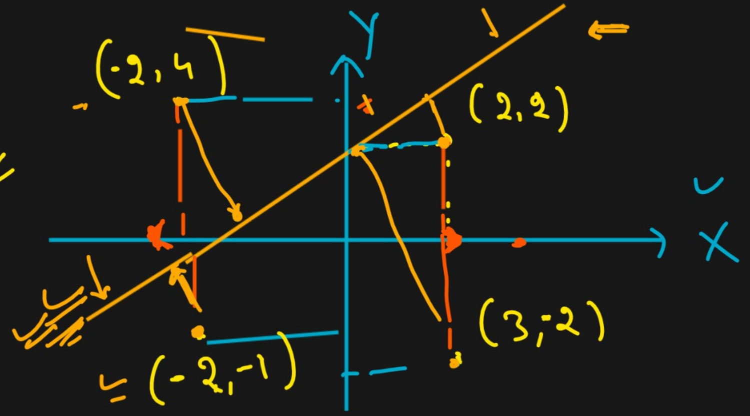
$\boxed{f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5}$



$\{ f_3, \underline{f_2, f_4} \}$.

Dimension Reduction :-

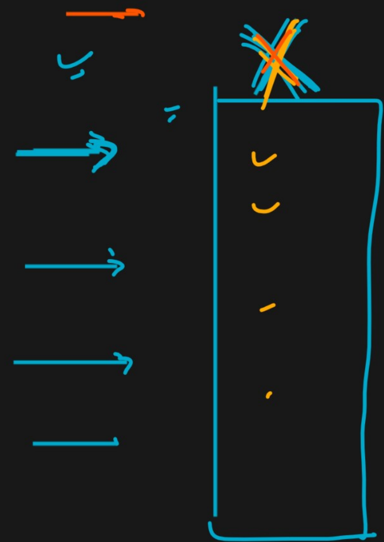




Vectrization.

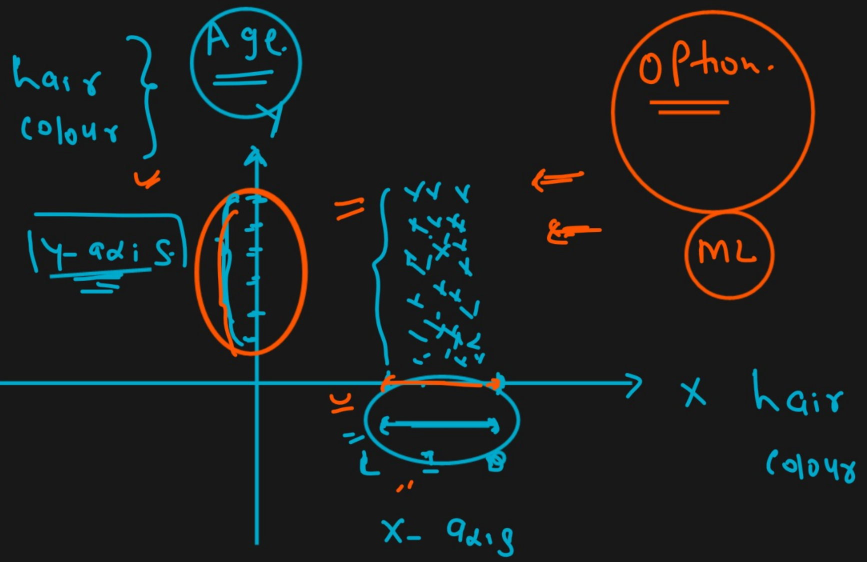
{ eign value
{ eign vector

X	Y
2	2
3	-2
-2	-1
-2	4



PCA- Basic.

INDIA → {Age, hair colour}



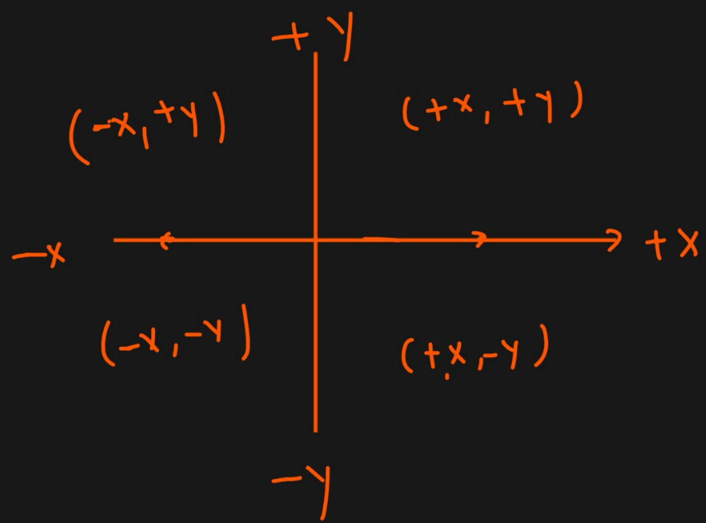
spread is variation?

↓

true

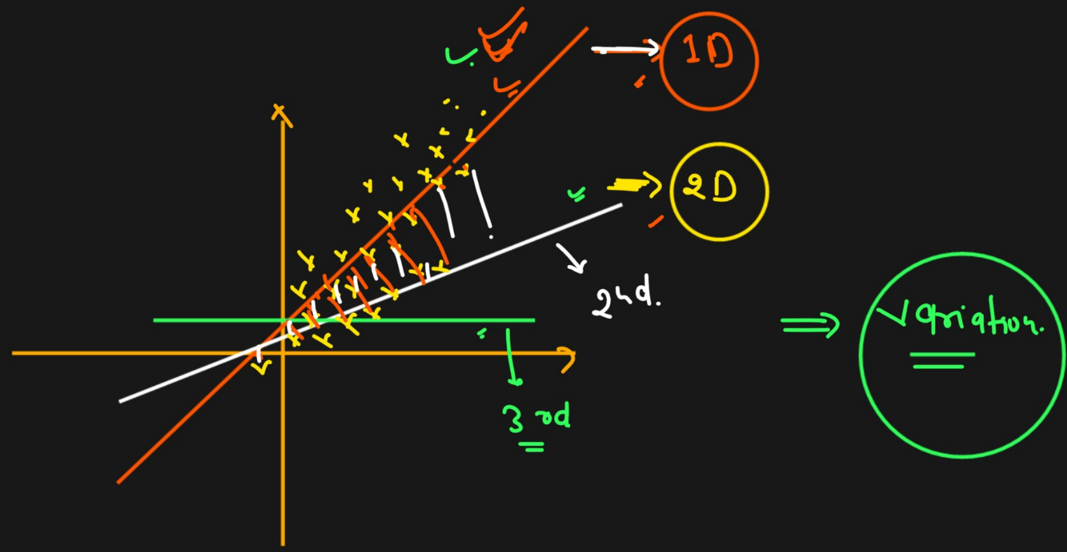
Spread of the data on y axis ⇒ too high.

" " " x axis ⇒ too low



reducing dimensionality of data by transforming data points on a plane feature selection - Where we select subset of feature those are important .

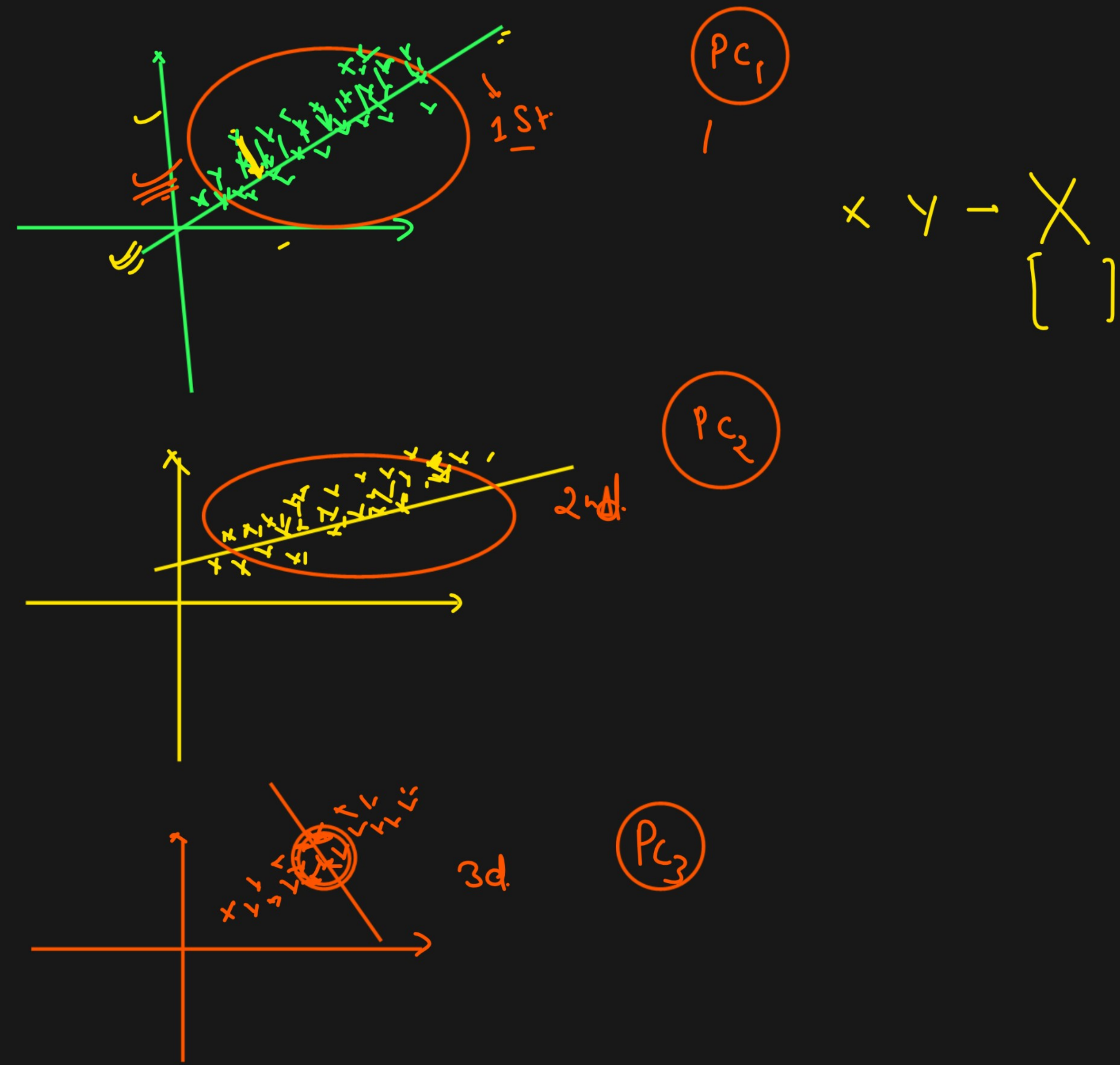
in feature selection we eliminate features but in dimension reduction we project data in one diamension not eliminate



PCA ⇒ Dim red. 4 D. → 2D , 1D. 3D.

↑

Visulaation.

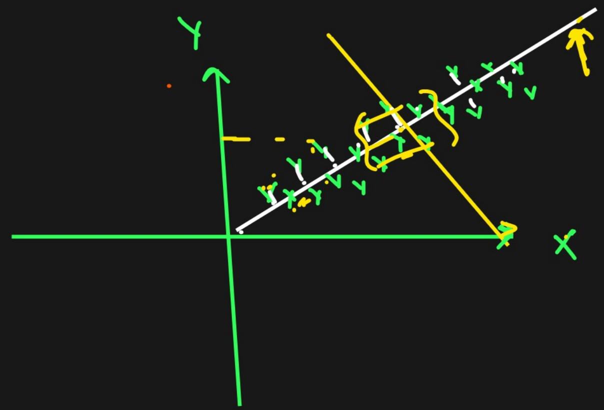


when pca is use and why?

when there is to much features

pca is used when our data is very large

When we have more features in our dataset and which causes curse of dimensionality



$PC_1 \Rightarrow 95\%$ variation.

$PC_2 = 5\%$

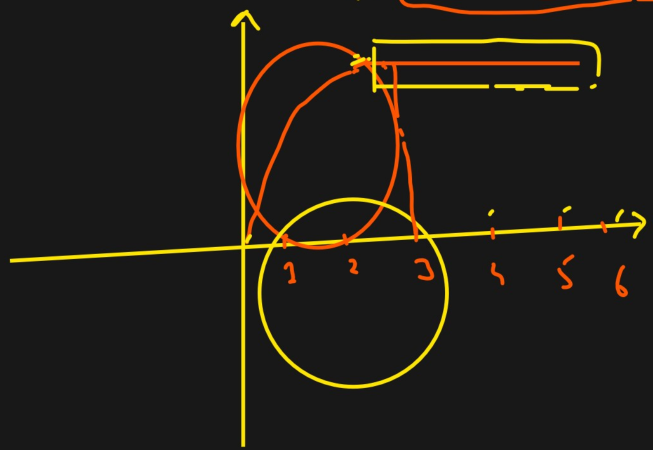
mixture

We can have more than one Principle Component

$$\frac{PC_1}{PC_1 + PC_2} = \frac{95}{100} = 0.95$$

$N \rightarrow 100\%$

Variation $= PC_1 + PC_2 + PC_3$



No. of Principle

PCA

1. Scaling or Std.
   - Scaling → Min-max
   - Std. → Standard Scala
2. Covariance matrix
3. Eigen value and eigen vector
4. Principle component

= (Scaling) :- transforming a data within certain range.

Min - Max.

[0 , 1]

min   max

Data Points :- $(a_1, a_2, a_3, a_4 ---- a_{100})$

Scaling of the data.

[min, max]   [0,1]

min

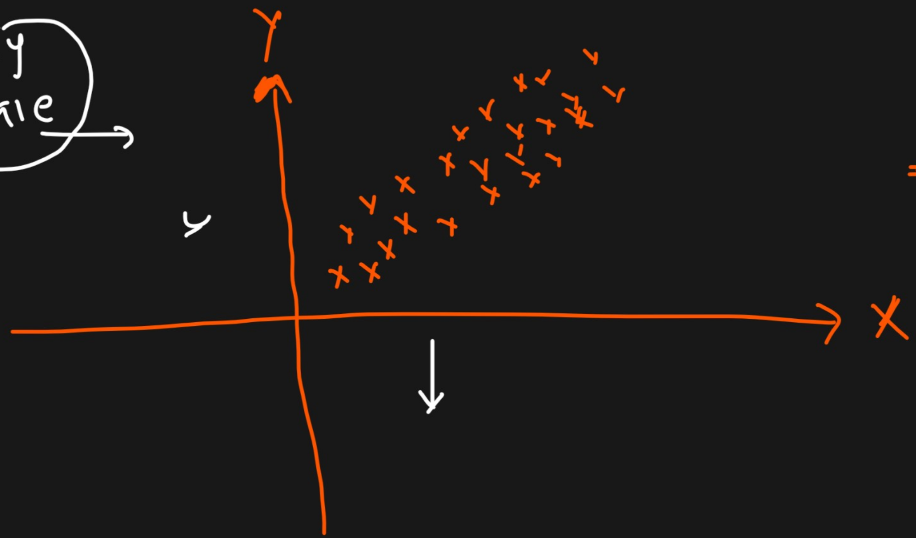
$$\boxed{\frac{\text{Data Point} - \text{min}}{\text{max} - \text{min}}}$$

$$\Rightarrow \frac{\text{min} - \text{min}}{\text{max} - \text{min}} = 0 \quad \boxed{[0, 1]}$$

$a_1 - a_{100} \Rightarrow$

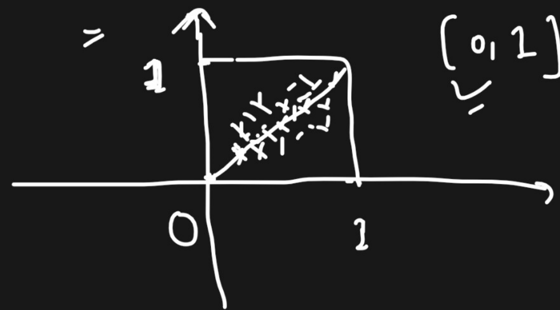
$$\frac{max - min}{max - min} = 1$$

any Scale



$\longrightarrow$ $\boxed{\text{min-max}}$

=



[0, 1]

Standard Scala :-

$[a_1 \ldots\ldots\ldots a_{100}]$

$\Downarrow$

S.S. formula $\Rightarrow \dfrac{\text{Data Point} - \mu}{\sigma} \Rightarrow$ mean = 0, std deviation = 1

SND

$2, 5, 8, 9, 10 \Rightarrow \mu \Rightarrow \dfrac{34}{5} = 7$

$\Downarrow$

SND

$\sigma \Rightarrow$

$\left(\dfrac{D.P - \mu}{\sigma}\right) \Rightarrow \dfrac{2-\mu}{\sigma}, \dfrac{5-\mu}{\sigma}, \dfrac{8-\mu}{\sigma}, \dfrac{9-\mu}{\sigma}, \dfrac{10-\mu}{\sigma}$

SND

$\Rightarrow [\ ,\ ,\ ,\ ,\ ] \Rightarrow$ mean = 0, $\Rightarrow$ std = 1

mean.
SND
0
1
Zero centric data
SND

② Variance

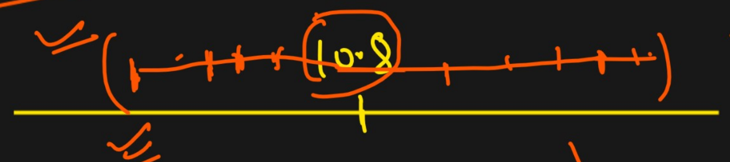
(Co-Var)

2, 8, 10, 12, 6, 5, 13, 14, 18, 20

↓ mean

$$\frac{1}{n}\sum_{i=1}^{n}(\text{Data Point} - \text{mean})^2$$

deviation from mean.



mean.

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

X (Y)

$$\rightarrow \operatorname{cov}(x,y) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}(X-\bar{X})(Y-\bar{Y})$$

$\cancel{X}$ $\quad cov(x, x) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})^2$

variance $\Leftarrow$ $cov(x, x) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x-\bar{x})(x-\bar{x})$

$\boxed{cov(x, y) = cov(y, x)}$

2 variable $\Rightarrow$ 2×2

$\xrightarrow{PCA}$

$$\begin{array}{cc} & \begin{array}{cc} X & Y \end{array} \\ & \begin{bmatrix} cov(x, y) & cov(x, x) \\ cov(y, y) & cov(y, x) \end{bmatrix} \end{array}$$

3 variable

3×3

PCA $\Rightarrow$ 1 Std 2 Cov.

$\boxed{\text{Data Points}} \rightarrow$ SND $\Rightarrow$ $\mu = 0$, $\sigma = 1$

mean = 0

1

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

$$\text{Variance} = \mathrm{cov}(x, x) \Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_i)^2$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_i - 0)(y_i - 0)$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_i)(y_i) \leftarrow \text{Dot Product}$$

X	Y
0.5	0.8
1	2.5
10	20
20	40
2.5	50

$\Rightarrow$ $\boxed{1}$ $\mathrm{cov}(x, y)$ = value

$\Rightarrow$ 2 SND $\mathrm{cov}(x, y)$ = Value

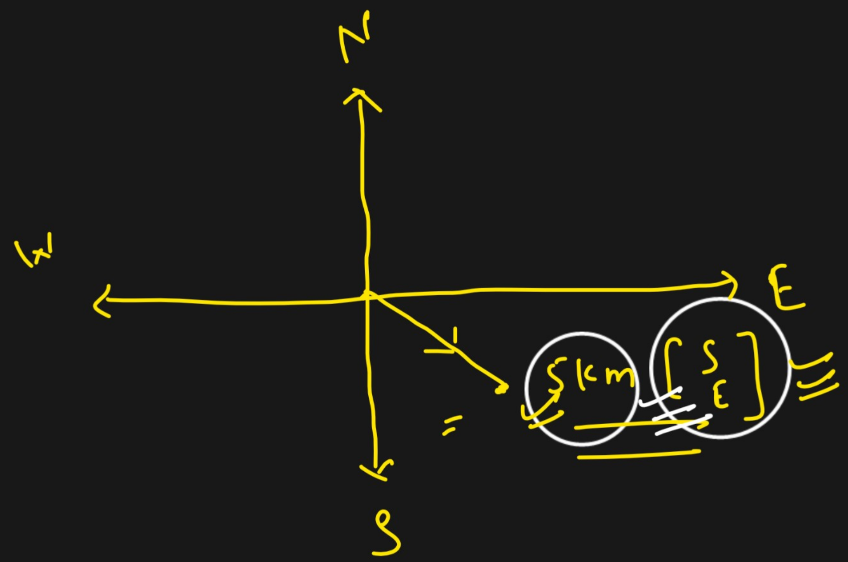
Dot Product :- multiplication of the data Point followed by Summistion.

$(x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4 \ldots) \Rightarrow$ scalar value

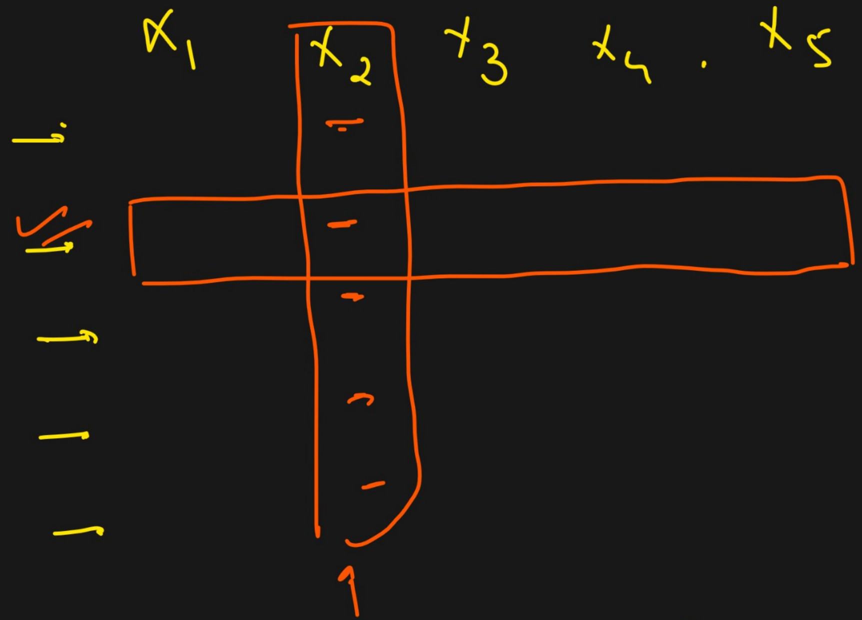
③ Eigen value & eigen vector

Scalar value :- Simple int value.

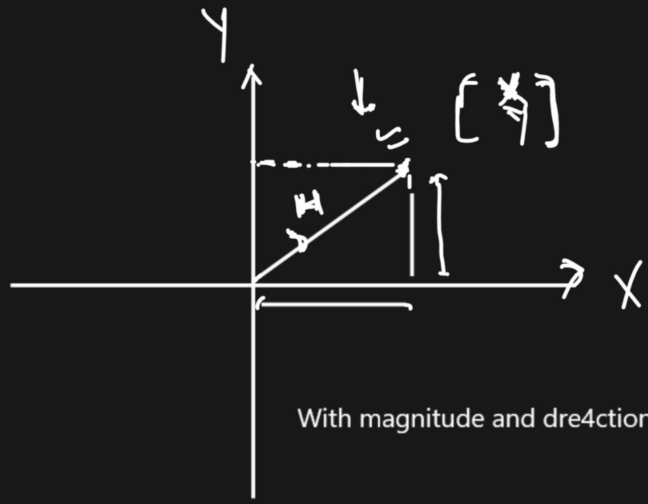
Vector :- Scalar value + Direction.



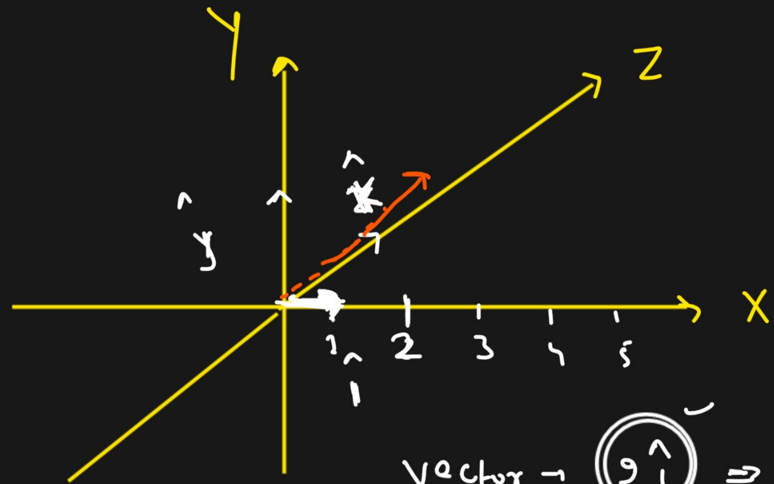
Machine learing :- (R x c)



= Colum vector = [ ]



With magnitude and dre4ction we can represent vector



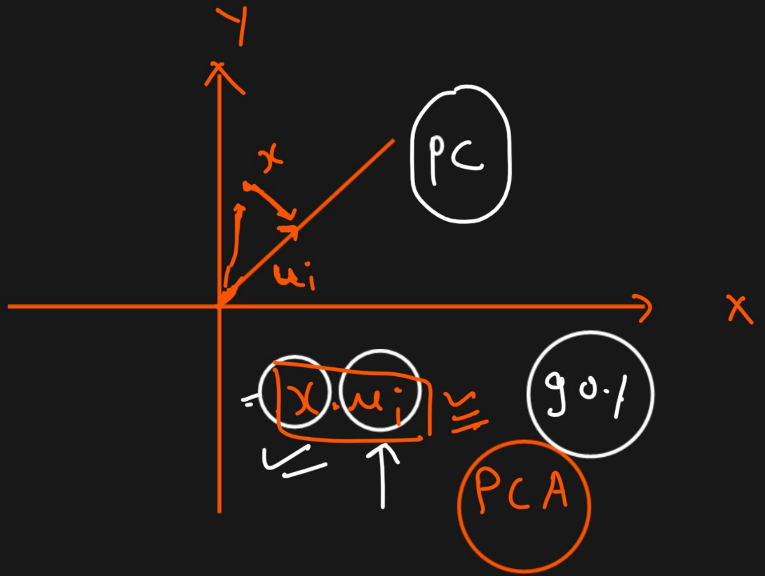
Vector → $2\hat{i} \Rightarrow X$

$2\hat{j} \Rightarrow Y$

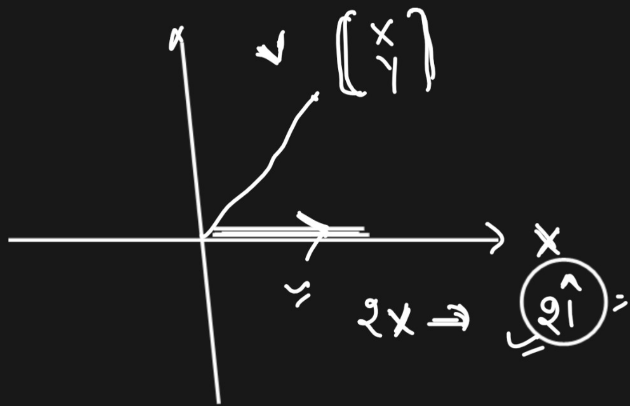
$2\hat{k} \Rightarrow Z$

$\Rightarrow \boxed{X, Y, Z}$

$u = 2\hat{i} + 2\hat{j} + 2\hat{k}$



→ (Vector multiplication.)



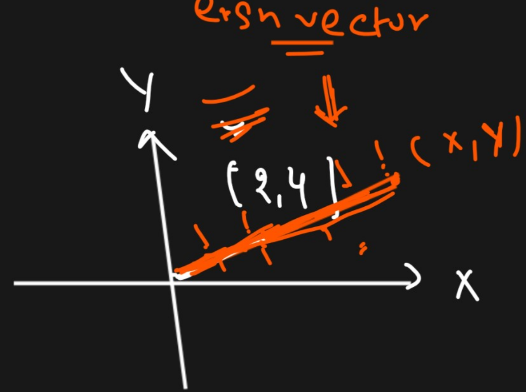
2 is scaler and I, j, k is unit vector

Unit vector is nothing just its a magnitude with value 1 and its represents a specific direction

Eigen value and Eigen vector.

$$\boxed{C - \lambda I = 0}$$

$$\boxed{CV = \lambda V}$$



2, 4, 6, 8

$\begin{bmatrix}1\\2\end{bmatrix} \to \times 2 \Rightarrow \begin{bmatrix}2\\4\end{bmatrix}$

4

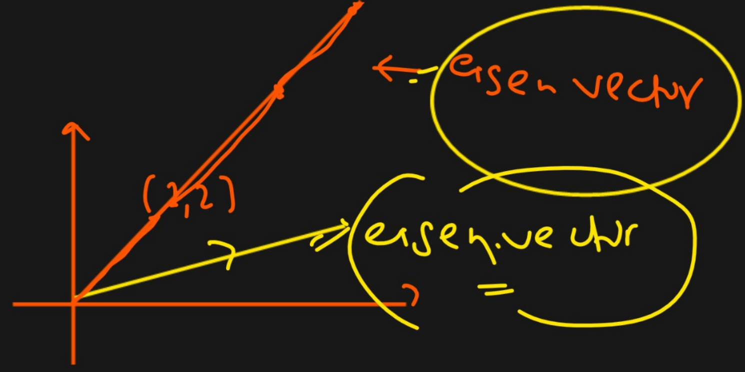
6

Data:- $\begin{bmatrix}3\\3\end{bmatrix} \Rightarrow$ Vector $\to$ PC

$$\begin{bmatrix}1\\2\end{bmatrix}$$

$$2 \leftarrow \begin{bmatrix}2\\4\end{bmatrix}$$

$$4 \leftarrow \begin{pmatrix}4\\8\end{pmatrix}$$



$$\boxed{AV = \lambda V} \Rightarrow AV - \lambda V = 0$$

$$\boxed{(A - \lambda I)V = 0}$$

$$A = \begin{bmatrix}1 & 2\\1 & 0\end{bmatrix}$$

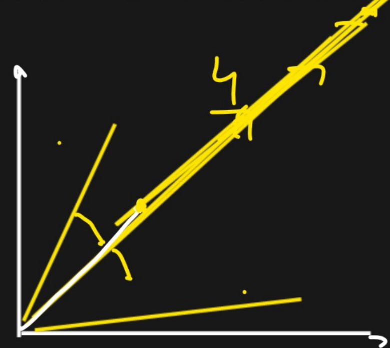
Ex. $\Rightarrow$

$$\begin{bmatrix}1 & 2\\1 & 0\end{bmatrix}\begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}4\\2\end{bmatrix}$$

$$\Rightarrow \boxed{\underbrace{\begin{bmatrix}1 & 2\\1 & 0\end{bmatrix}}_{A} = \underbrace{\begin{bmatrix}2\\1\end{bmatrix}}_{V} = 2\begin{bmatrix}2\\1\end{bmatrix}}$$

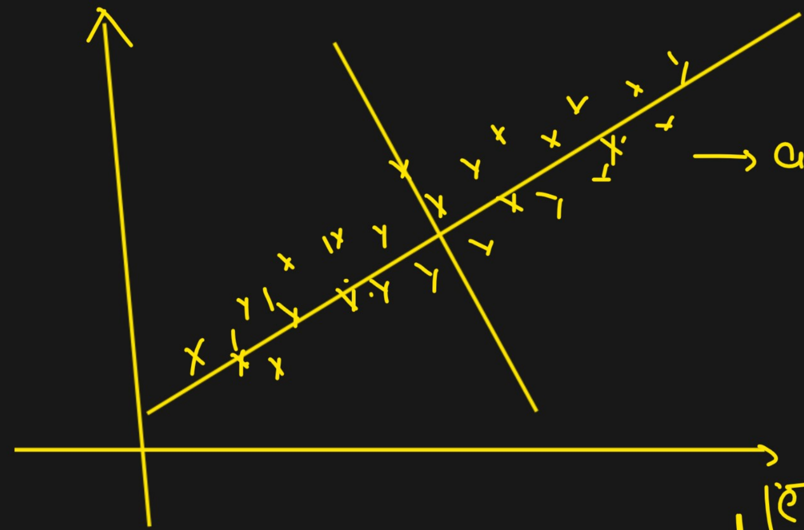
$$\begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 9 \\ 3 \end{bmatrix}$$

when matrix * vector our tend to rotate but if vector is eigenvector it will not rotate it will just stretch the vector



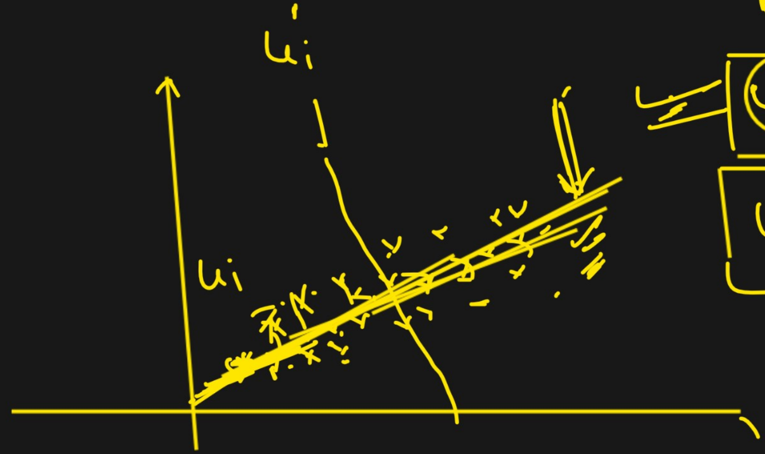
2, 4, 8

$= 2(1, 2, 4)$



→ actual data point

eigen value and eigen vector



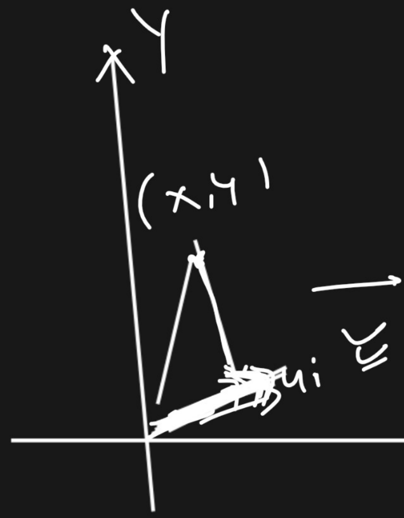
$\boxed{u_i \cdot x_i} = [\ \ ]$

$\boxed{u_i' \cdot x_i}$

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

1. Std.
2. Cov.
3. eigen value and eigen vector
4. Principle component

$\Rightarrow$ Single vector



$(x \cdot u_i)$

$\longrightarrow$ Projection or transforming.

$\begin{bmatrix} x \\ y \end{bmatrix} (u_i) \Rightarrow$