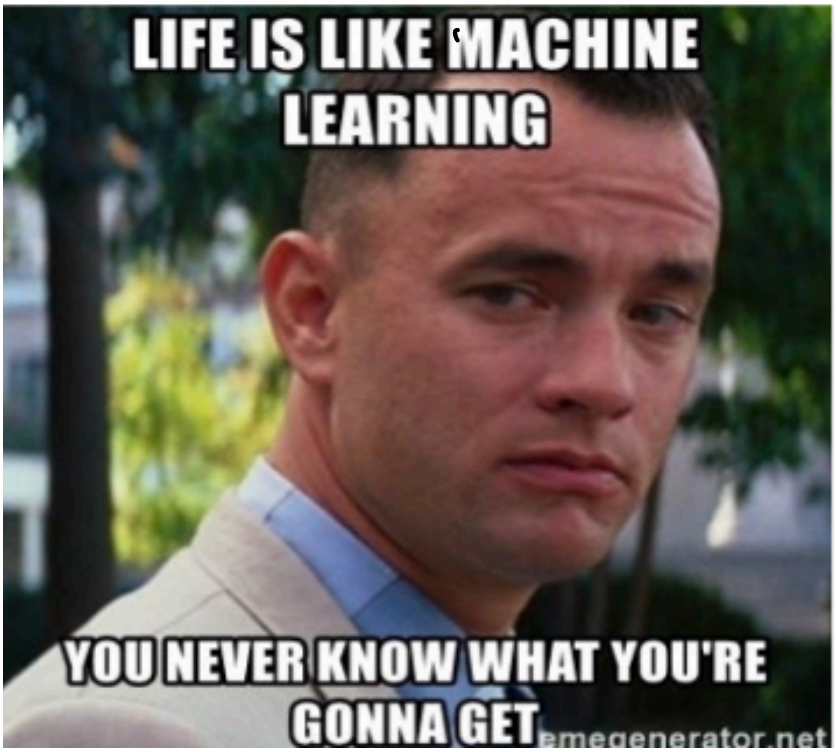
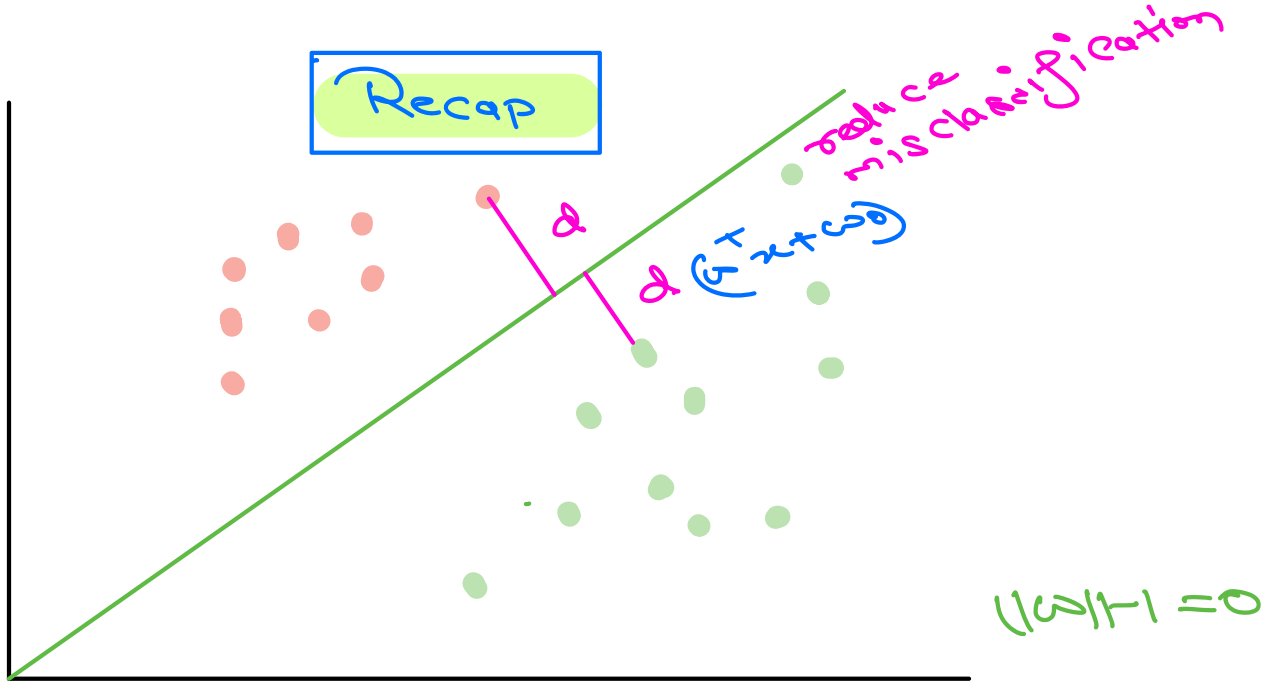


Agenda

- Recap
- Variants of Gradient Descent
  - Vanilla GD
  - SGD
  - Batch GD
- Dimensionality Reduction
- PCA



Constraint

$L(D, \bar{w}, w_0) = -\left(\bar{w}^T \bar{x} + w_0\right) y_i$ s.t $\|w\|=1$

$+ \lambda(\|w\|-1)$

$\lambda(\epsilon_q = 0)$

Gradient Descent

① $\bar{w}^{(t+1)} = \bar{w}^{(t)} - \eta \nabla_{\bar{w}} L(D, \bar{w}, w_0)$

$\downarrow \bar{w}_{4\times 1} + w_0 = 1.4$

y_1	$x_1 \dots x_q$
y_2	x_2
$\vdots$	
y_n	$x_n \quad x_n^q$

$\bar{w}^{(t+1)} - \eta\left(-\sum_{i=1}^{n} y_i \bar{x}_i + \lambda \frac{\bar{w}}{\|w\|}\right)$ regularizer

$\frac{\partial L}{\partial \bar{w}}$

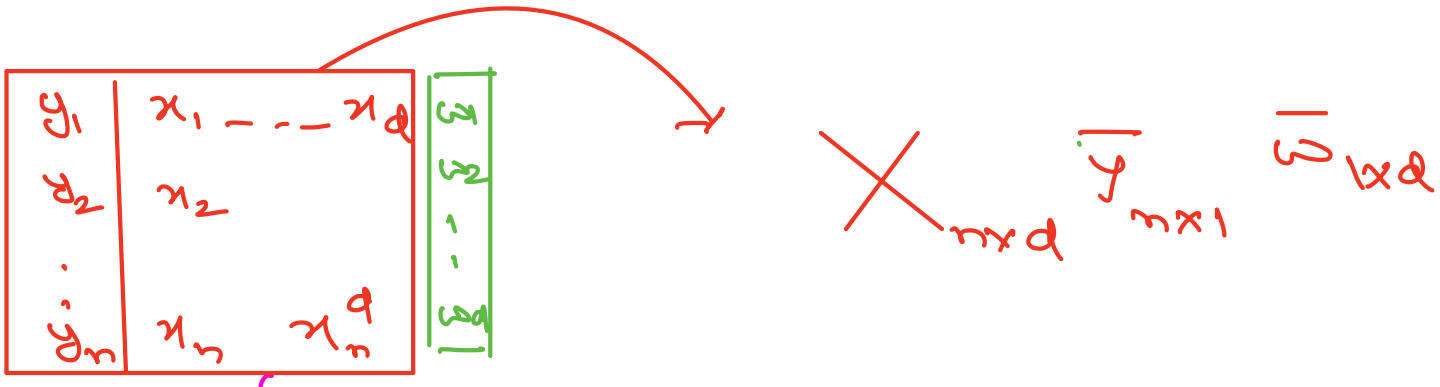
② $w_0^{(t+1)} = w_0^{(t)} - \eta\left(-\sum_{i=1}^{n} y_i\right)$

$\frac{\partial L}{\partial w_0}$

$t+1 \rightarrow$ Timestamp / iteration

Variants of Batch Gradient Descent

Epoch : A full scan/pass through dataset



① Vanilla GD : Full Batch GD

10 million DP

$$w^{(t)} - \eta \left(- \sum_{i=1}^{n} y_i x_i + \lambda \frac{w}{\|w\|} \right)$$

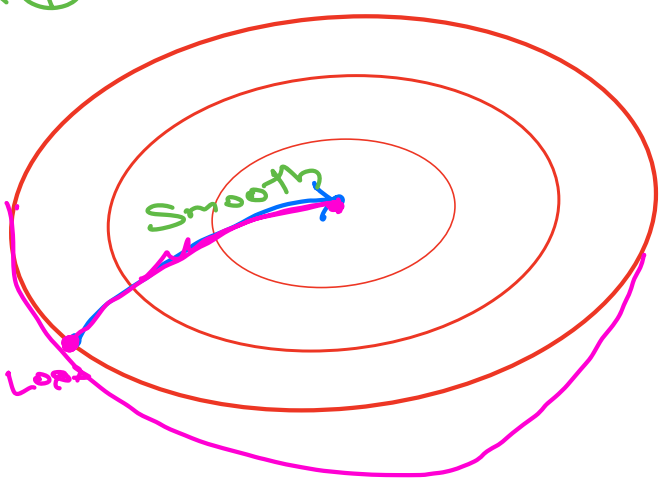
$$w^{t} - \left(\eta \left(Y \cdot X + \lambda \frac{w}{\|w\|} \right) \right)$$

$X_{(3) \times d}$

	x_1	x_2	x_3	------ x_d	y
1					
2					
(3)					
n					

n data points

d - dimension



Loss function

* Full dataset is processed
Fastest but Requires High Computation

② SGD : Stochastic Gradient Descent

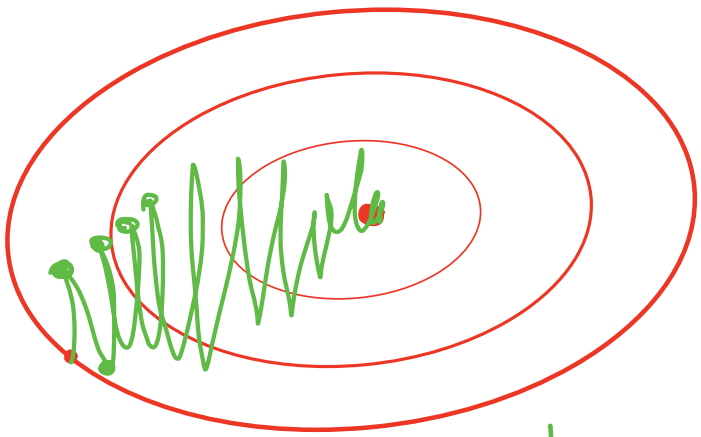
Pick i index Randomly

$$\bar{\omega}^{(t)} - \eta\left(-\sum_{i=1}^{n} y_i \bar{x}_i + \lambda \frac{\bar{\omega}}{\|\omega\|}\right)$$

Stochastic = Randomness

$$\bar{\omega}^{(t)} - \eta\left(- y_i \bar{x}_i + \lambda \frac{\bar{\omega}}{\|\omega\|}\right)$$

where i is picked randomly from $(1, n)$



Loss function

d is very high

③ BGD : Batch Gradient Descent

↓

Mini Batch

Batch-size → Hyperparameter → Based on Compute / Budget

stochastic → $1 < K < N$ ← Vanilla

$$\bar{w}^{(t)} - \eta\left(-\sum_{i=1}^{K} y_i \bar{x}_i + \lambda \frac{\bar{w}}{\|\bar{w}\|}\right)$$

10 <

Q) 10 samples Batch-Size 5

How many iteration will be required for 1 Epoch?

$10 // 5 \Rightarrow 2$

for i in num_batches(2):

rand_ind = [----]

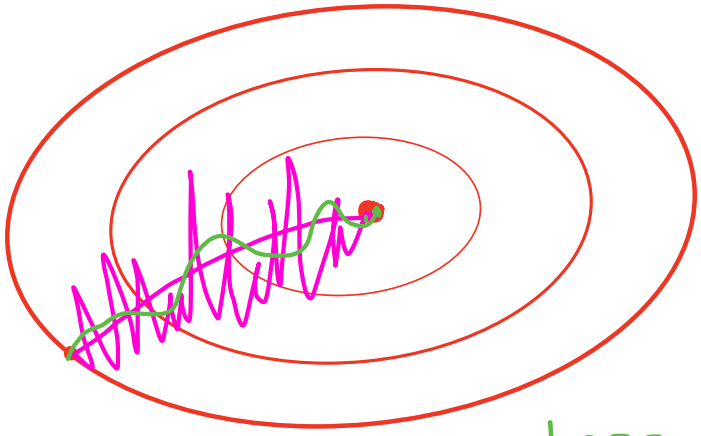
$w = w - \eta \nabla L$

$10 \times 5 \Rightarrow 50$

$5 \times 2 = 10$ 10

* 1 Epoch = $\frac{\text{Total Datapoints}}{\text{Batch-size}}$ ⇒ $\frac{100}{10} = 10$

10 epochs ⇒ 10^4 10 = 100 steps

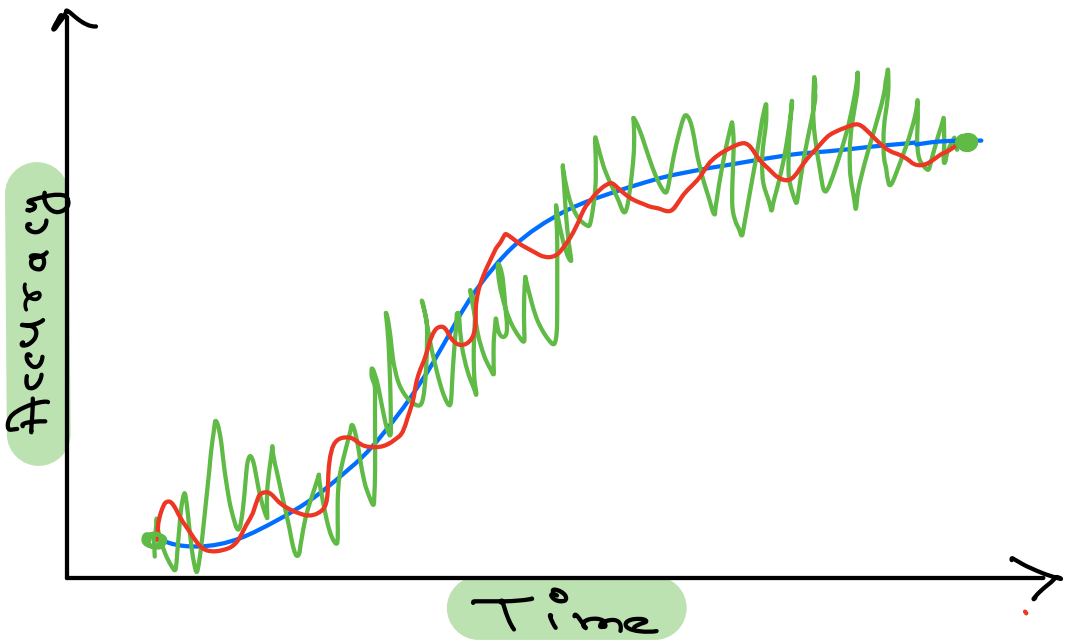


Loss function

$\frac{\log}{10} \Rightarrow 10$

Loss function

Questions

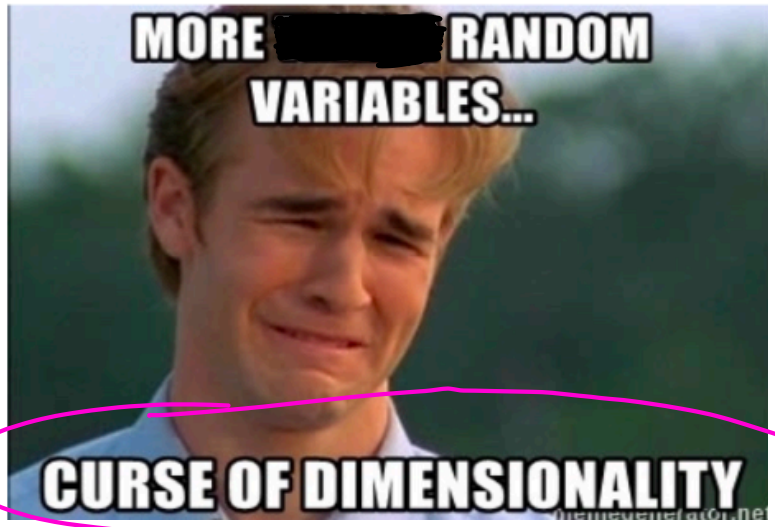


Blue ⇒ Vanilla

Green ⇒ Stochastic

Red ⇒ Mini Batch

Dimensionality Reduction



d is high → Curse of Dimensionality

	x_1	x_2	x_3	------ x_d	y
1					
2					
3					
n					

n data points

d-dimension

$$D = \{(\bar{x}_i, y_i) : \bar{x}_i \in \mathbb{R}^d\}_{i=1}^{n}$$

* Issues with Higher Dimensions:

① Visualization is difficult (3d-dimen EDA

② Training and processing Time

③ Computational Cost

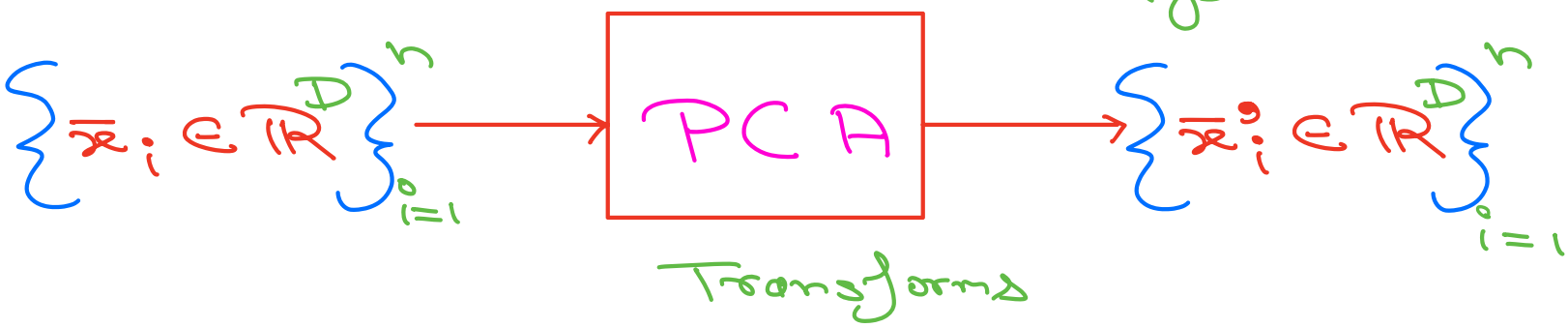
④ Difficult Maths

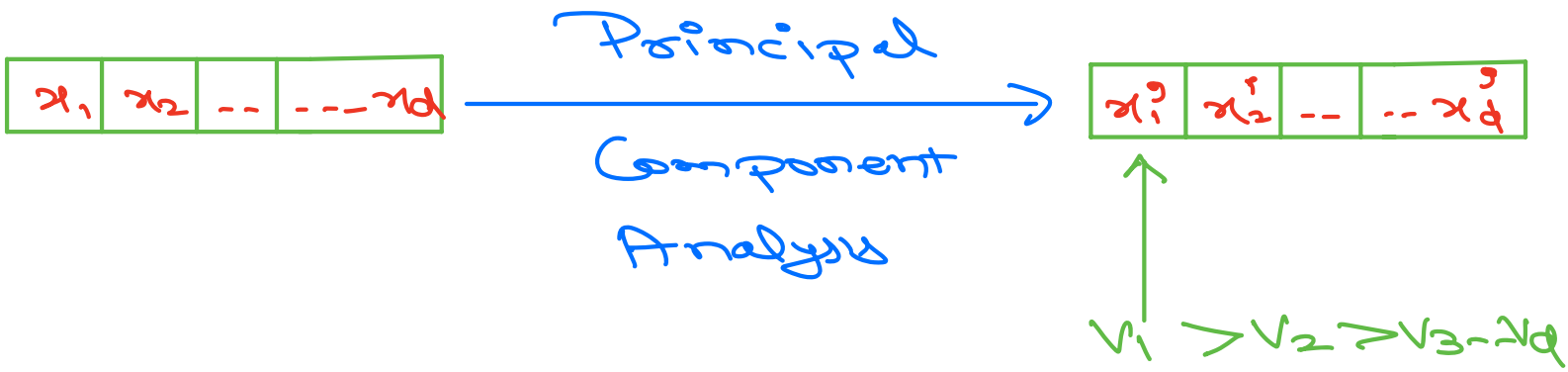
Dimensionality Reduction Algos

- PCA
- T-SNE
- UMAP

PCA

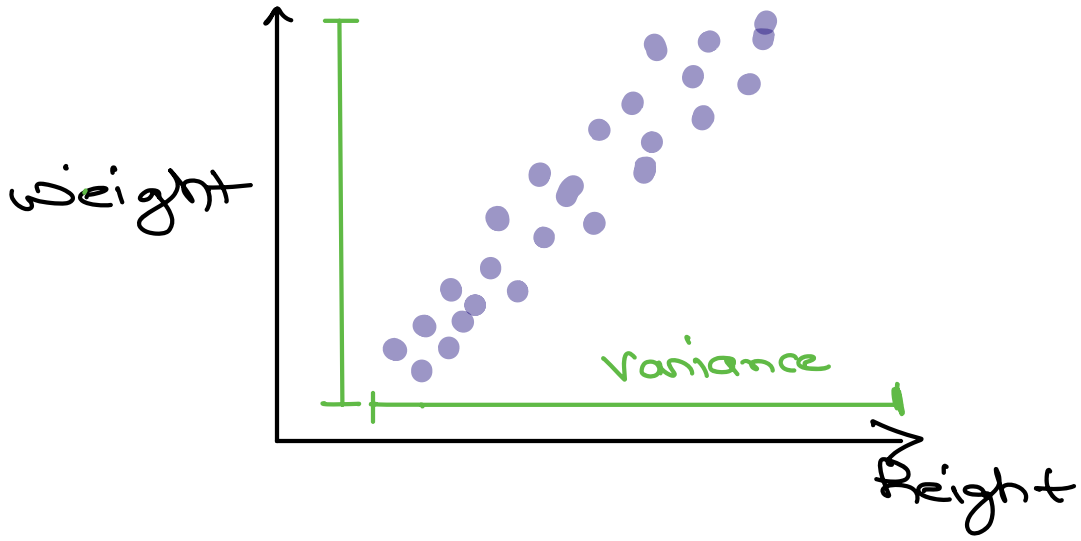
Principal Component Analysis

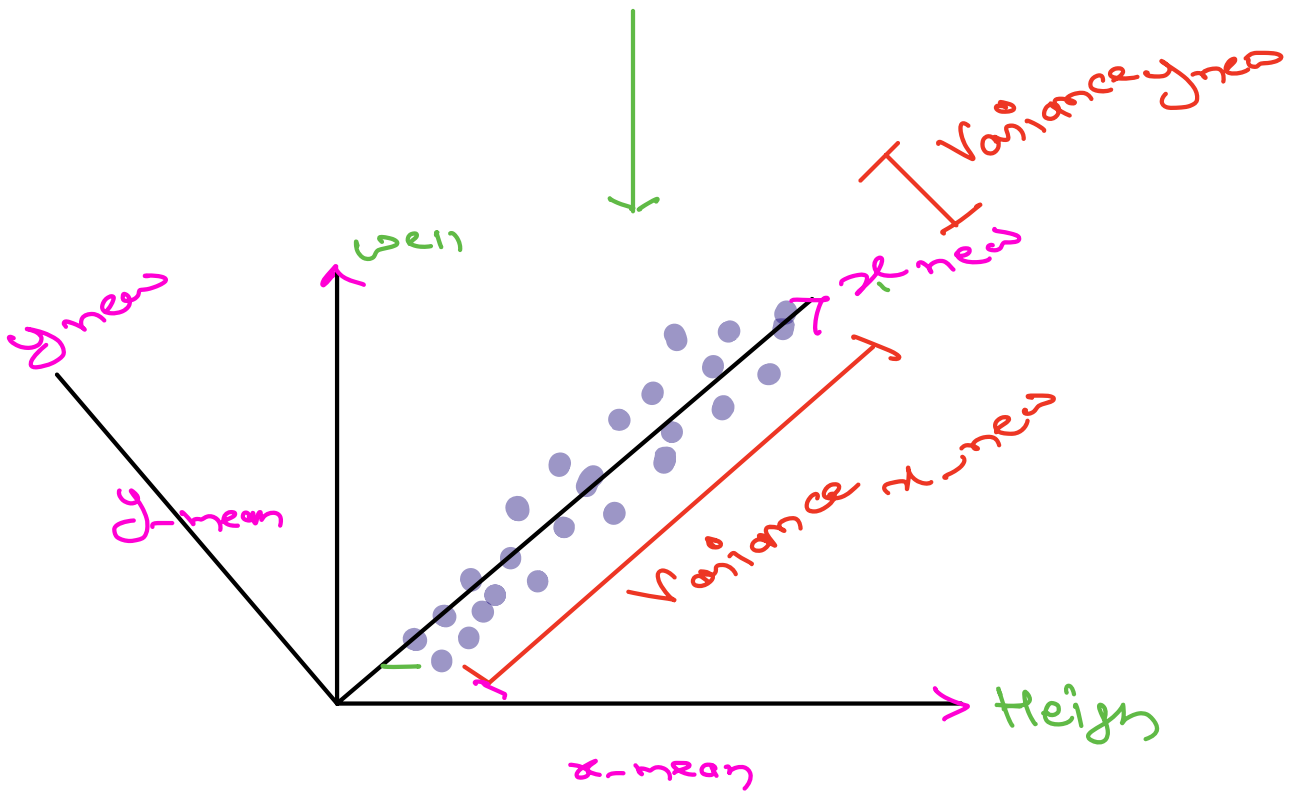




* Variance ⇒ The new dimensions capture variance in reverse sorted order

W	H	BMI



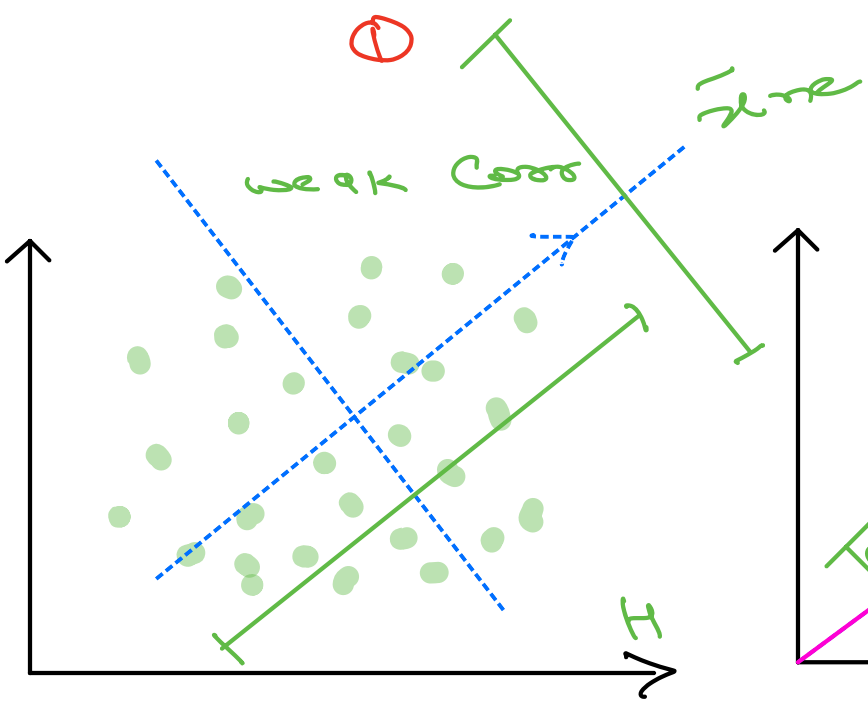


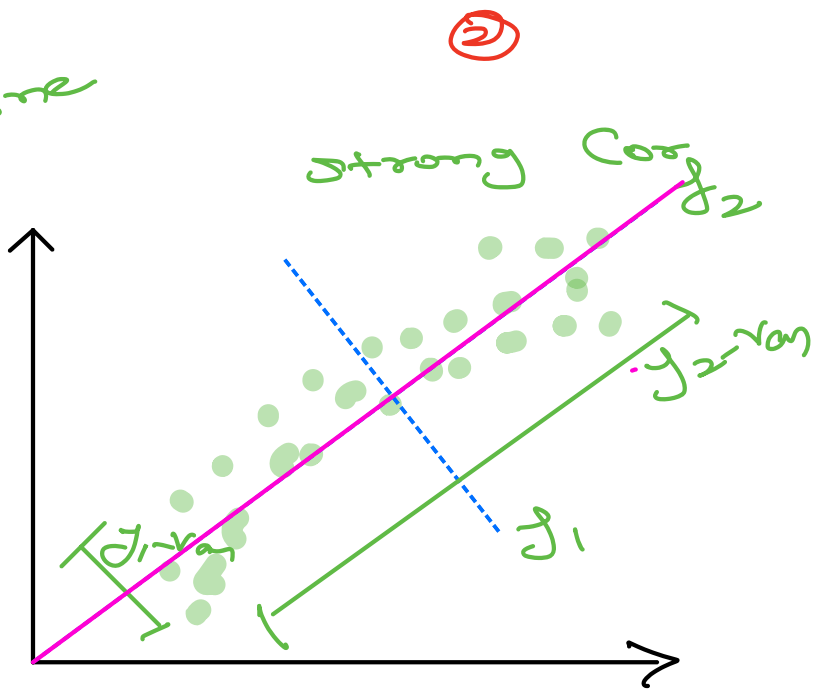
x y

x_{new} y_{new}

$Var_x > Var_y$

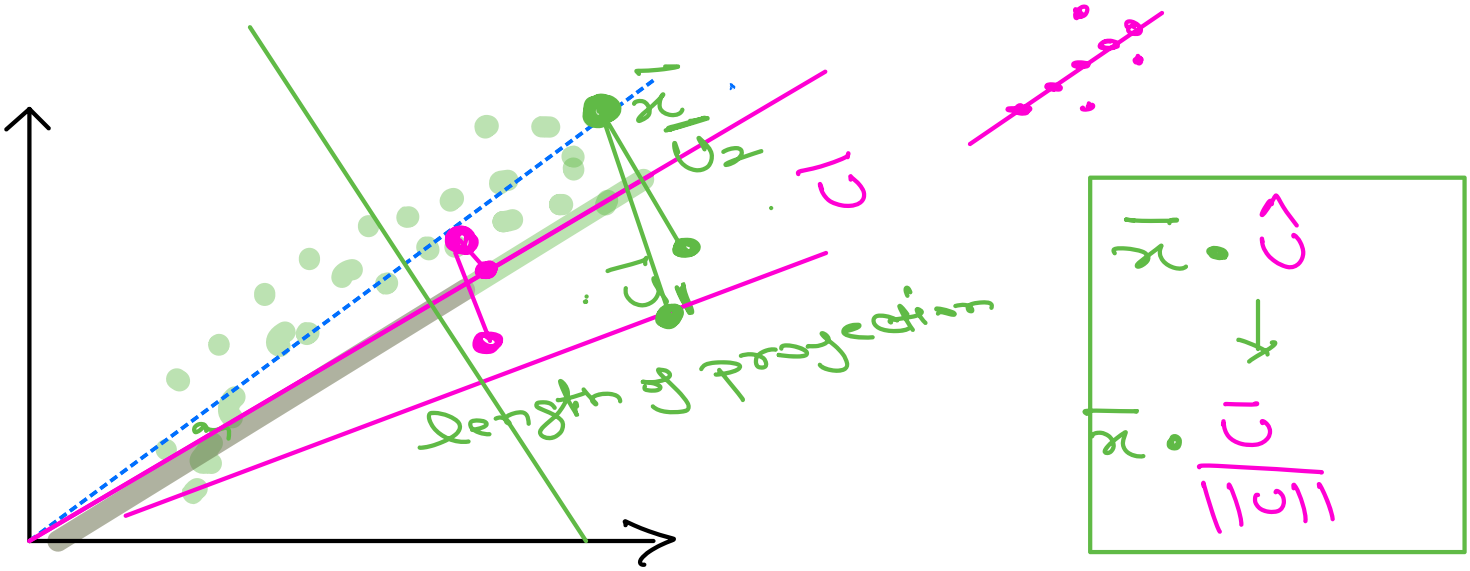
If we eliminate y_{new} we won't lose a lot of Information





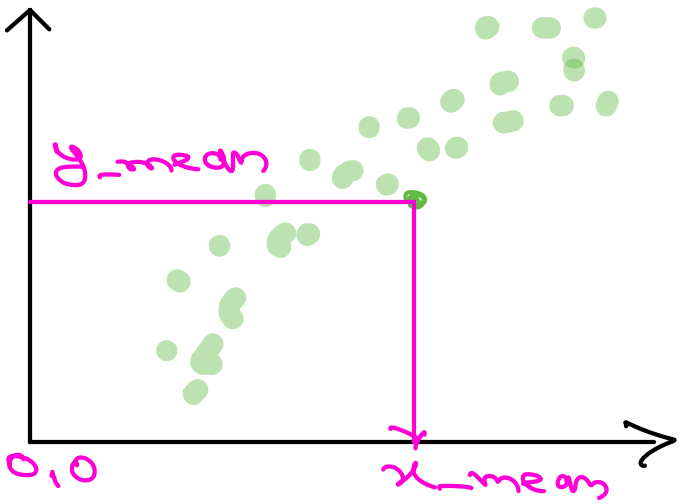
What is the Goal of PCA

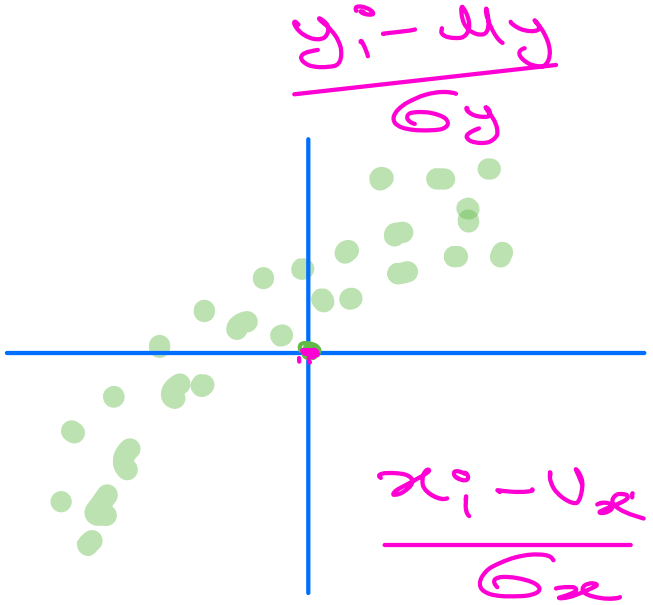
① Find new axis such that we can eliminate some features



Visualizing PCA

strong Cor





Step 1: Shift the origin to the mean of features

* Normal ⊖ −1 −1 (0,0)
* Standardization ⊖ $(\bar{x}, \bar{y})$

How can we do this Mathematically?

Objective ⊖ maximize

$$\sum_{i=1}^{n} \left(\frac{\bar{x} \cdot \bar{v}}{\|\bar{v}\|} - \mu_{\bar{v}} \right)^2$$

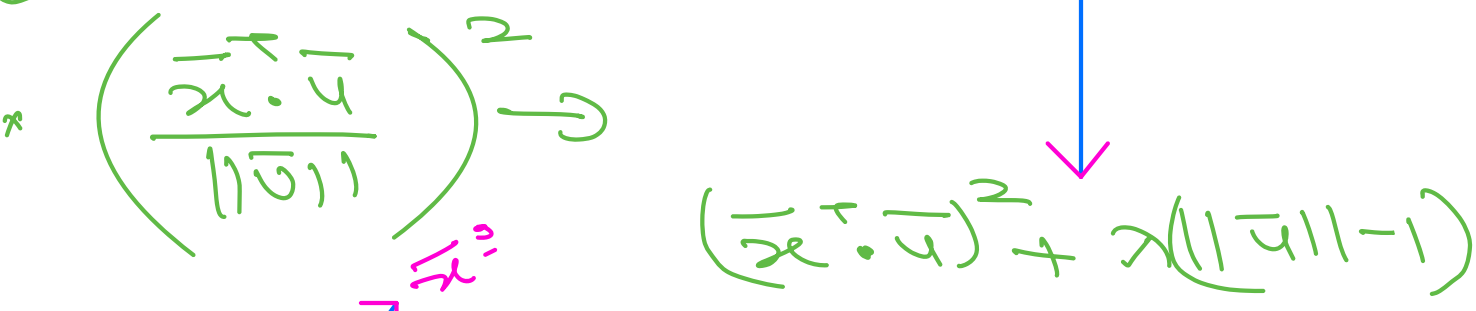
Step 2:

Rotate axis

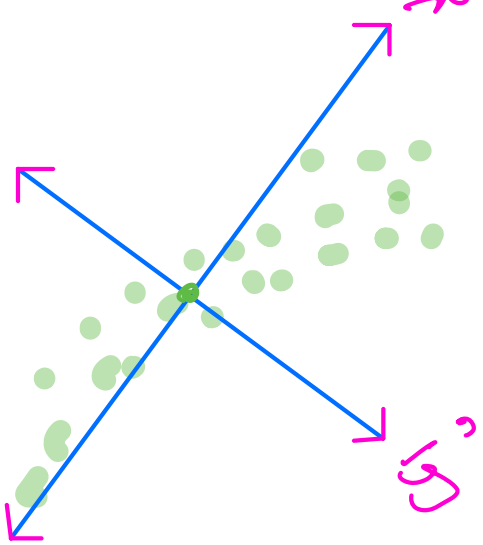
with objective

by maximizing projection

$$\max \left(\frac{\bar{x}^T \cdot \bar{u}}{\|\bar{u}\|} \right)^2 \rightarrow$$



$$(\bar{x}^T \cdot \bar{u})^2 + \lambda(\|\bar{u}\| - 1)$$

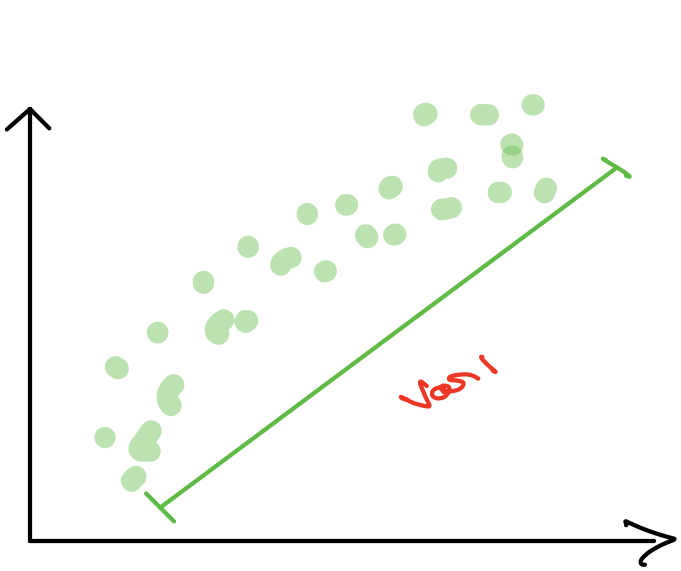


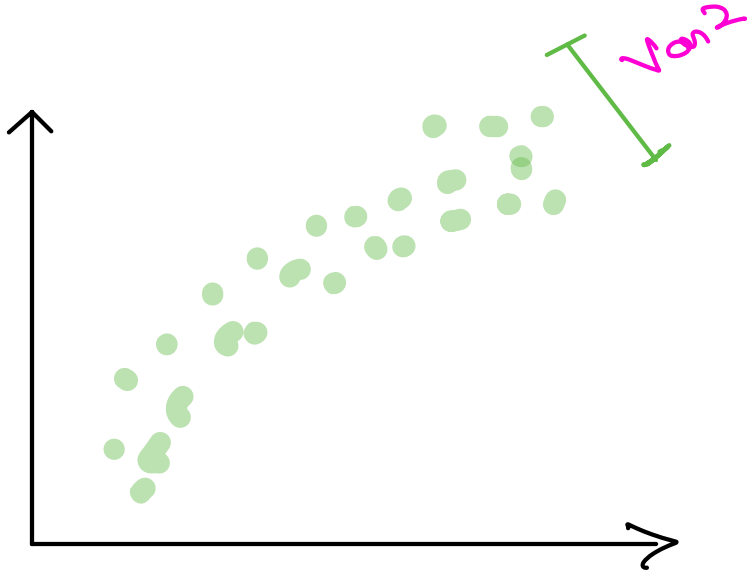
* Note: PCA Return new Orthogonal features

↓

Feature at 90° with Each Other

Implementing PCA





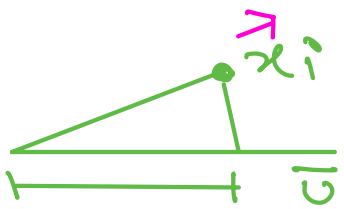
* Higher the Variance Captured on new axis more important the feature

Step 1 : Standardization

$$\frac{x_i - \mu_{x_i}}{\sigma_{x_i}}$$

Step 2 : Maximize the projections

$$\vec{u}^* = \underset{\vec{u}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \frac{\vec{x}_i^T \vec{u}}{\|u\|}$$



$$\bar{u}^* = \underset{\bar{u}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i^T \bar{u} + \lambda(\|u\|-1)$$

① Gradient Descent to find best $\bar{u}$

② Closed Form Solution / Eigen Value Decomposition

$$(X^T X)\,\bar{u} = \lambda \bar{u}$$

- $(X^T X)$ → Covariance Matrix
- λ → Eigen Value
- $\bar{u}$ → new feature axis

X → Dataset

f_1	f_2	$\cdots$	f_d
x_1			
x_2			
x_n			

λ → Eigen Value $\quad X \cdot X$

$\bar{u}$ → Eigen Vector

- 3 features Input $\bar{x}_1, \bar{x}_2, \bar{x}_3$
- 3 Eigen Value
- 3 Eigen Vector

λ_{10} ✗

$$\frac{\sum_{i=1}^{j=9} \lambda}{\sum_{i=1}^{j=10} \lambda} \Rightarrow \% \text{ of Information} \quad 99\%$$

② How much varian $\vec{U}_1$ store

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \Rightarrow$$