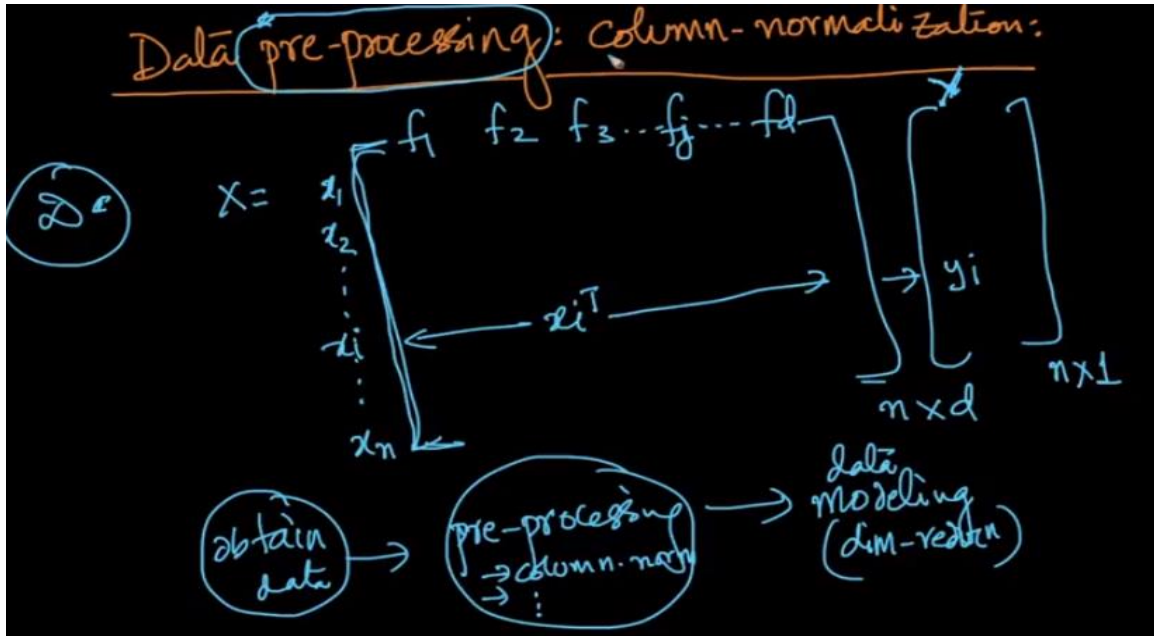
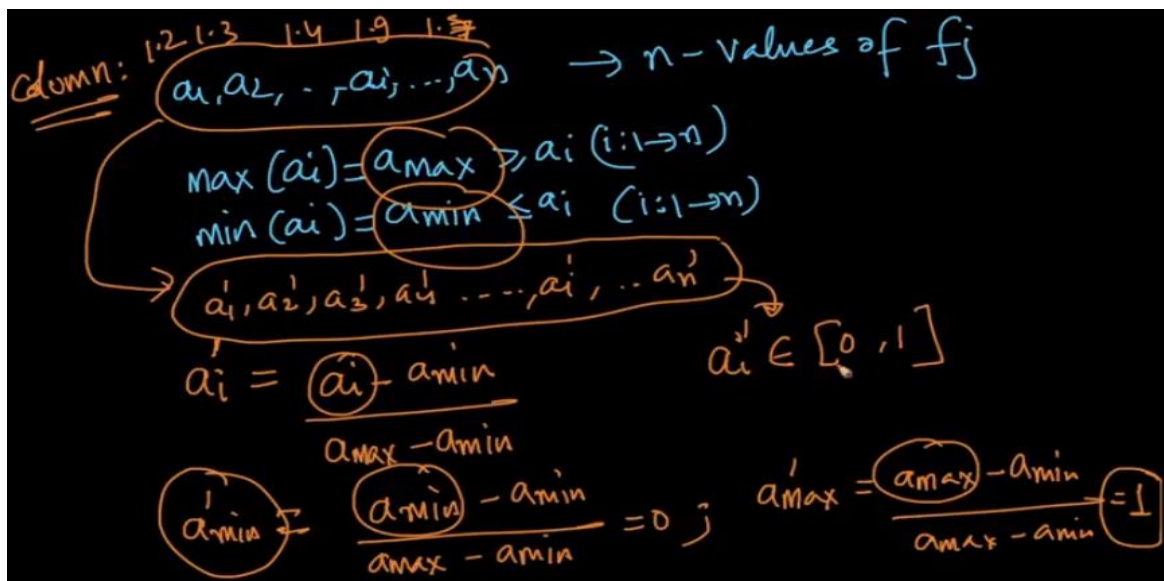


## Dimensionality Reduction & Visualization

Column Normalization: Pre-process data for dimensionality reduction



Column normalization technique – range [0,1] for feature[j]

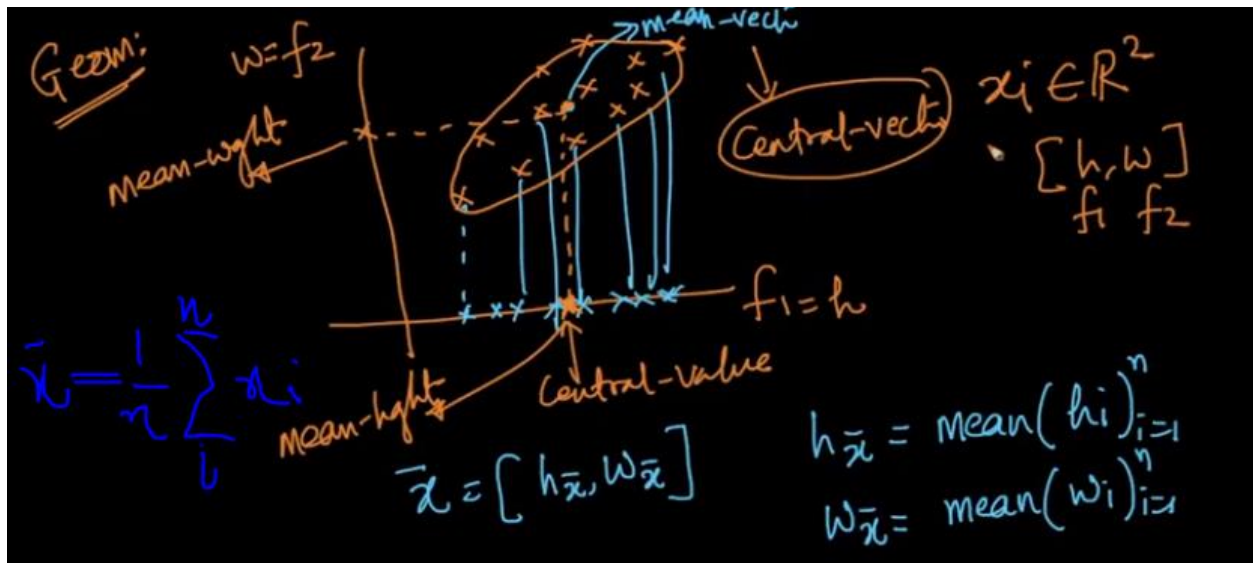


Feature transformed to new real values, ranging from [0,1] – do it for all the features.



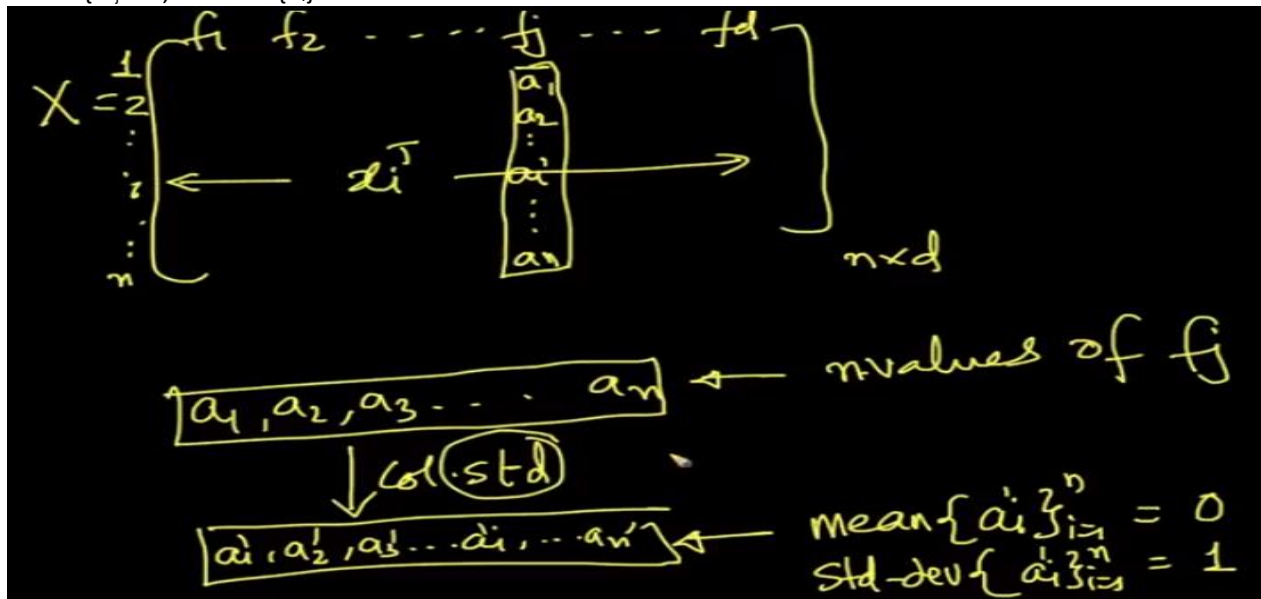
**Terminologies: -**

**Mean Vector:** Central value of the data

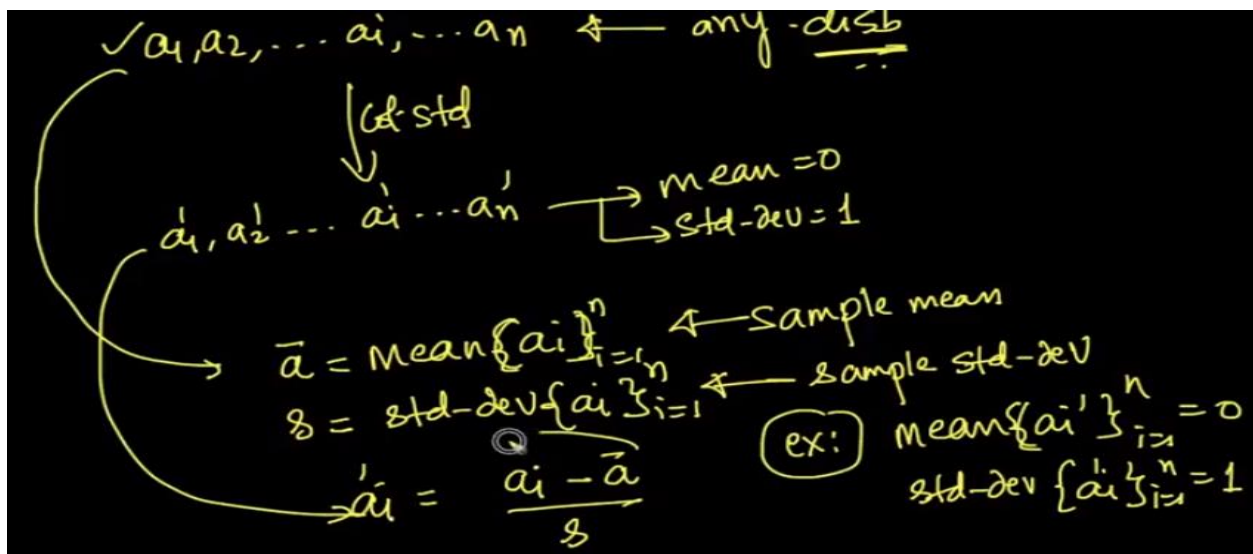


**Column Standardization:** - More often used than normalization

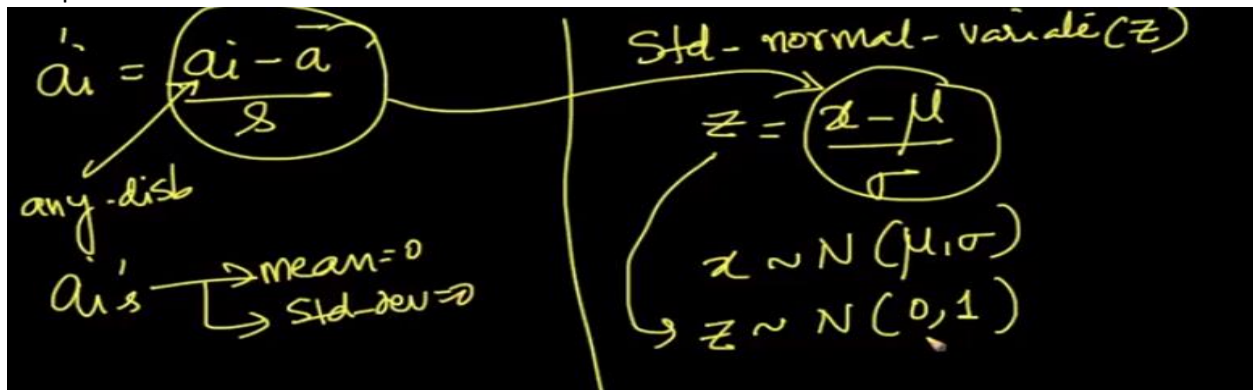
Mean  $\{X_i\} = 0$ , Std-Dev  $\{X_i\} = 1$



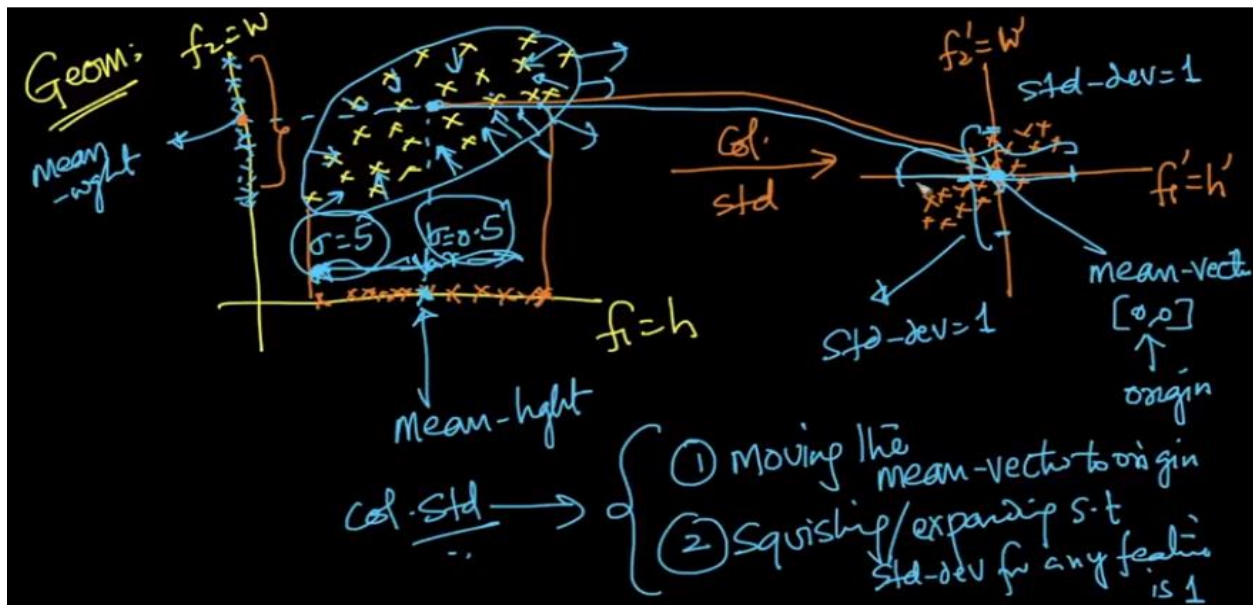
Transform  $a_i \rightarrow a'_i$  such that  $a'_i = (a_i - \text{mean}\{a_i\}) / \text{std-dev}\{a_i\}$  and  $a'_i$  has mean = 0 and std-dev = 1



Comparison with standard normal variate

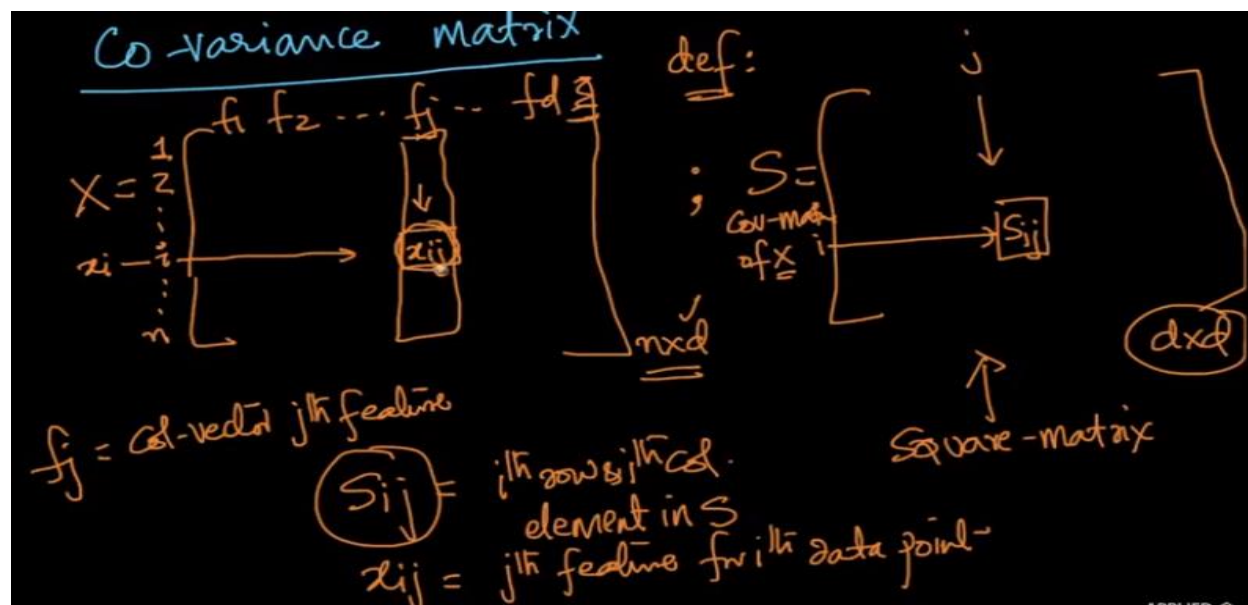


Geometric intuition of column standardization :-





**Covariance Matrix:** - (symmetric and square matrix of d-dimensions)



$S_{ij} = \text{cov}(f_i, f_j)$

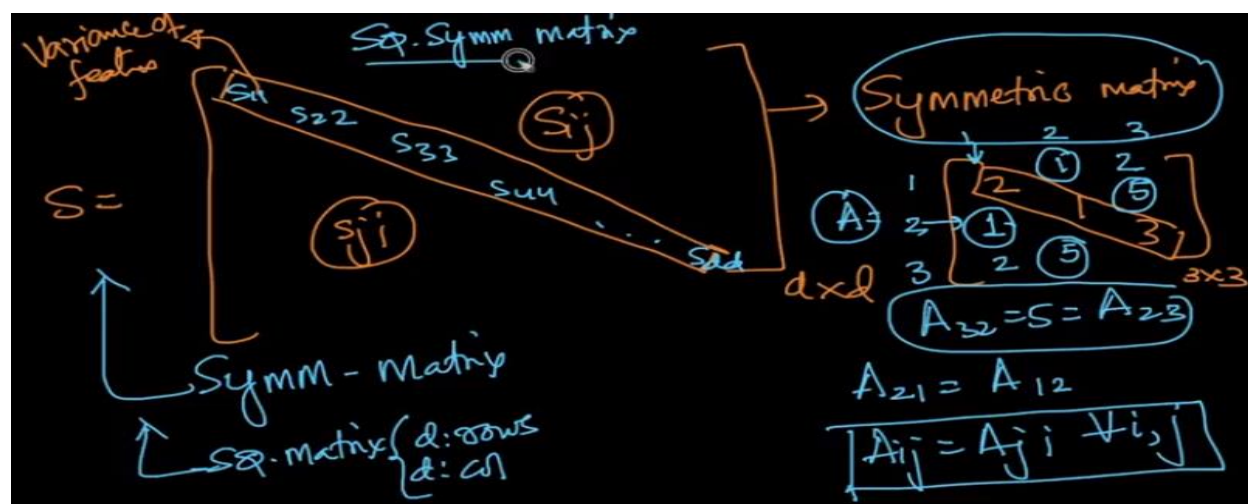
$i: 1 \rightarrow d$   
 $j: 1 \rightarrow d$

$\boxed{\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}$

$\text{Cov}(f_i, f_i) = \text{Var}(f_i)$

$\checkmark \text{Cov}(X, X) = \text{Var}(X) \text{ --- (1)}$

$\checkmark \text{Cov}(f_i, f_j) = \text{Cov}(f_j, f_i) \text{ --- (2)}$



Covariance matrix of column standardized dataset.

$$X = \begin{bmatrix} f_1 & f_2 & \dots & f_d \\ x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix} \quad n \times d$$

Let  $(X)$  col-standardized  $\Rightarrow \begin{cases} \text{mean}\{f_i\} = 0 \\ \text{std-dev}\{f_i\} = 1 \end{cases}$

$$\text{Cov}(f_1, f_2) = \frac{1}{n} \sum_{i=1}^n (x_{i1} - \underbrace{\mu_1}_{\text{mean}(f_1)}) (\underbrace{x_{i2} - \mu_2}_{\text{mean}(f_2)})$$

$$\text{Cov}(f_1, f_2) = \frac{1}{n} \sum_{i=1}^n x_{i1} * x_{i2}$$

$$X = \begin{bmatrix} f_1 & f_2 & \dots & f_d \\ 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 2 & x_{21} & x_{22} & \dots & x_{2d} \\ 3 & x_{31} & x_{32} & \dots & x_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$$

$$\text{Cov}(f_1, f_2) = (f_1^T f_2) * \frac{1}{n}$$

$$S_{d \times d} = \frac{1}{n} (X^T)_{d \times n} (X)_{n \times d} = (d \times d) \checkmark$$

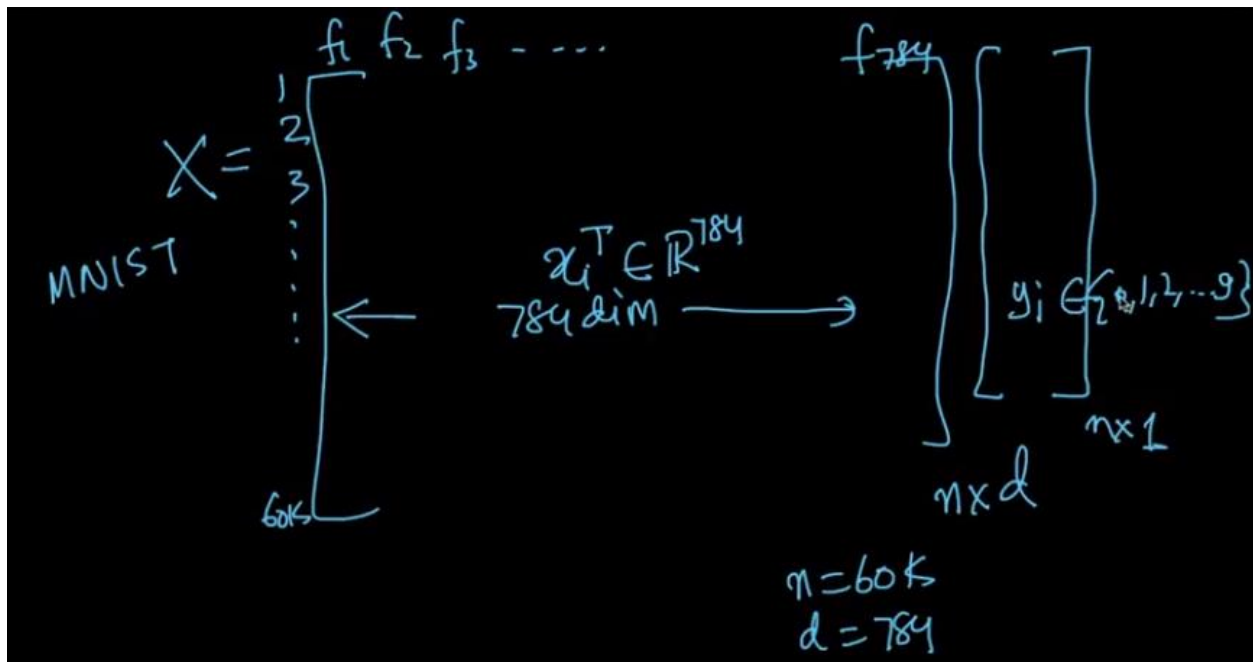
data-matrix

(\*) assuming X has been col-std

$$S_{ij} = \text{Cov}(f_i, f_j) = \frac{f_i^T f_j}{n}$$

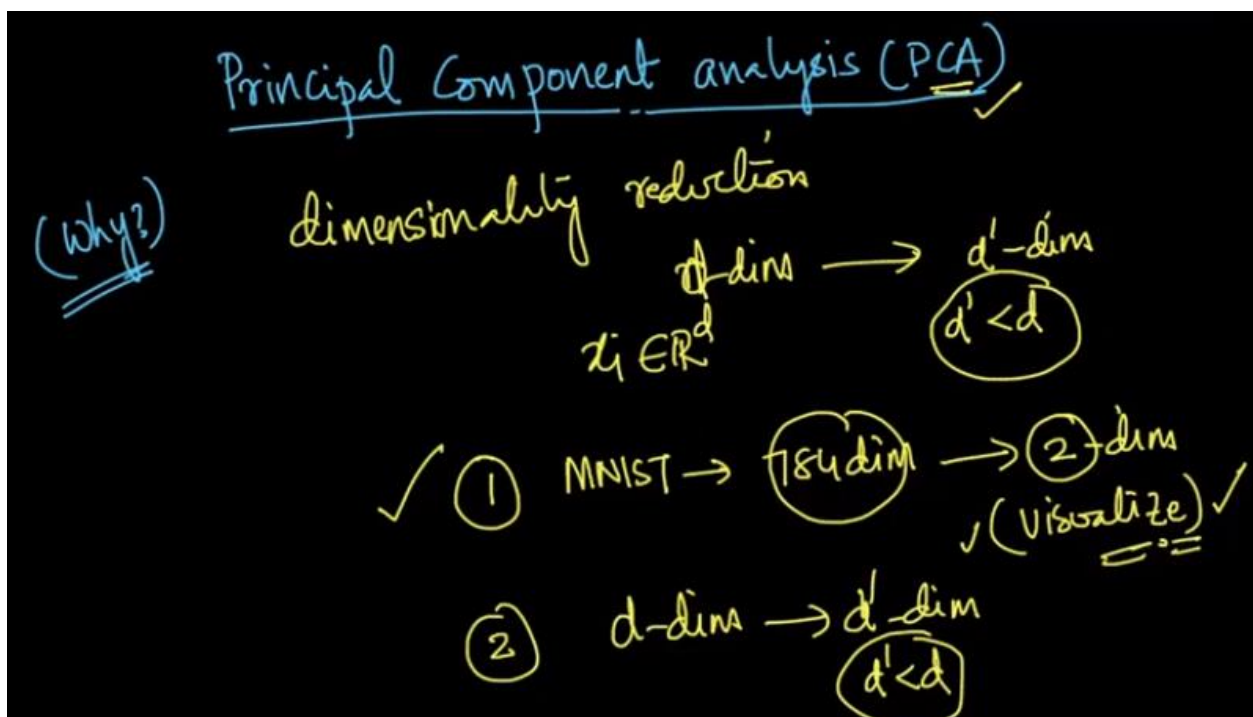
## Principle Component Analysis: - (Example MNIST Dataset)

Data Representation: - N (60K) features flattened to 784 dimensions



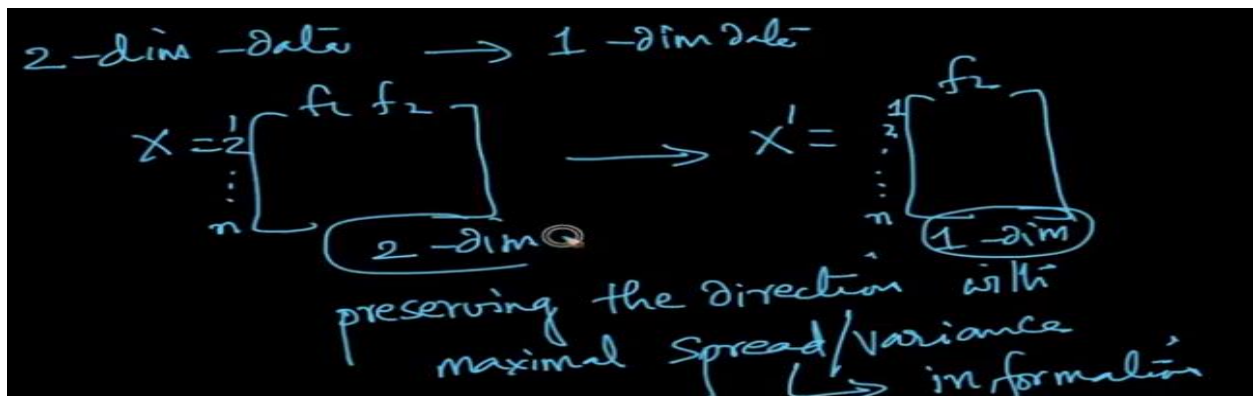
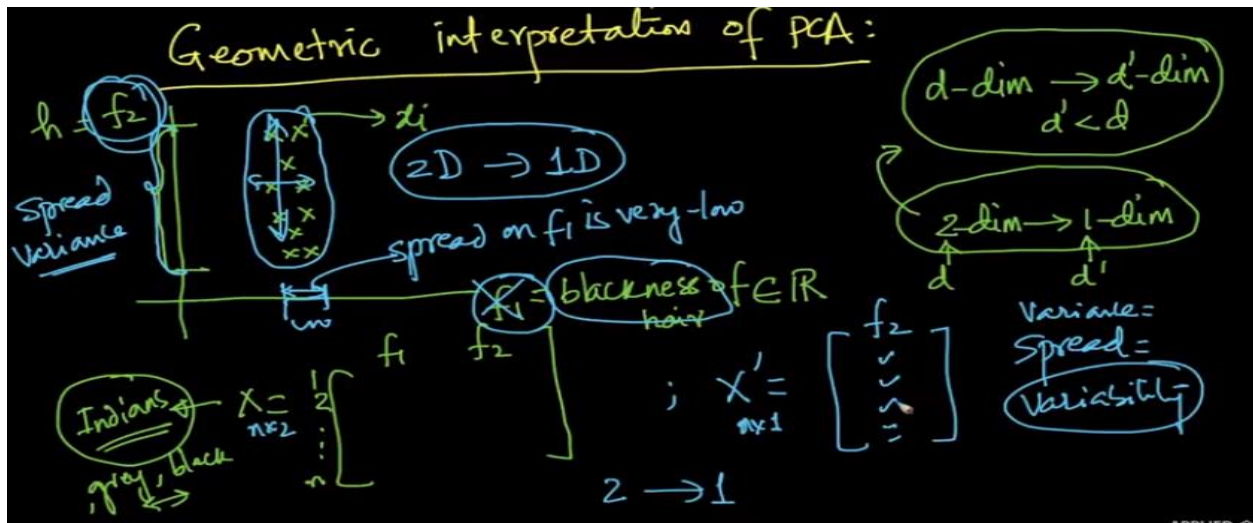
Difficult to visualize for classification task – where PCA comes into the picture.

PCA – Method to reduce the dimensions of the dataset.



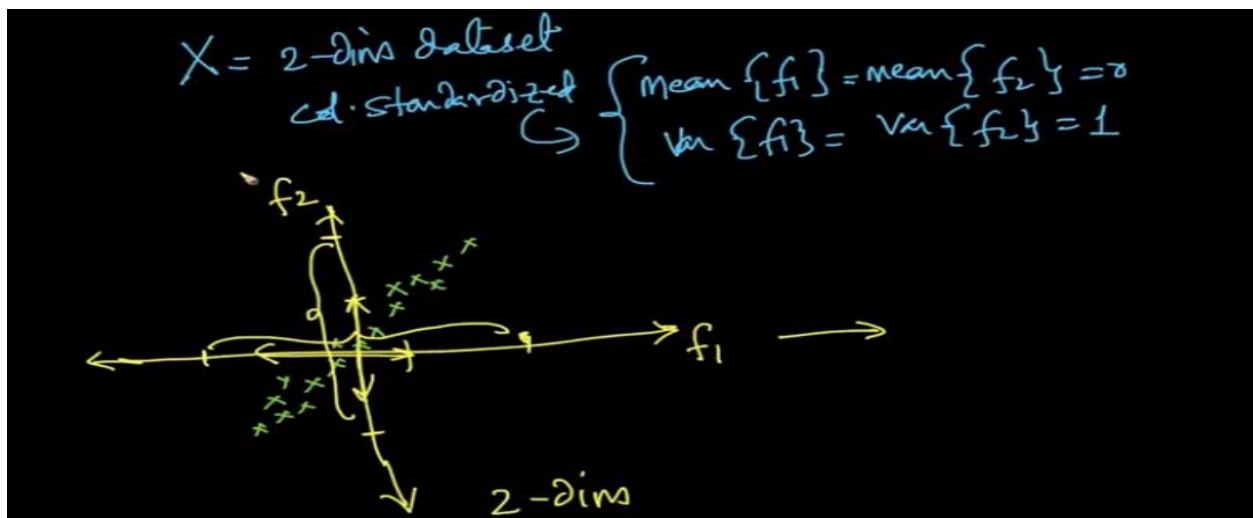
## Geometric Interpretation of PCA: -

Dropping the feature which has less variance (drop by intuition but has problem of losing information)



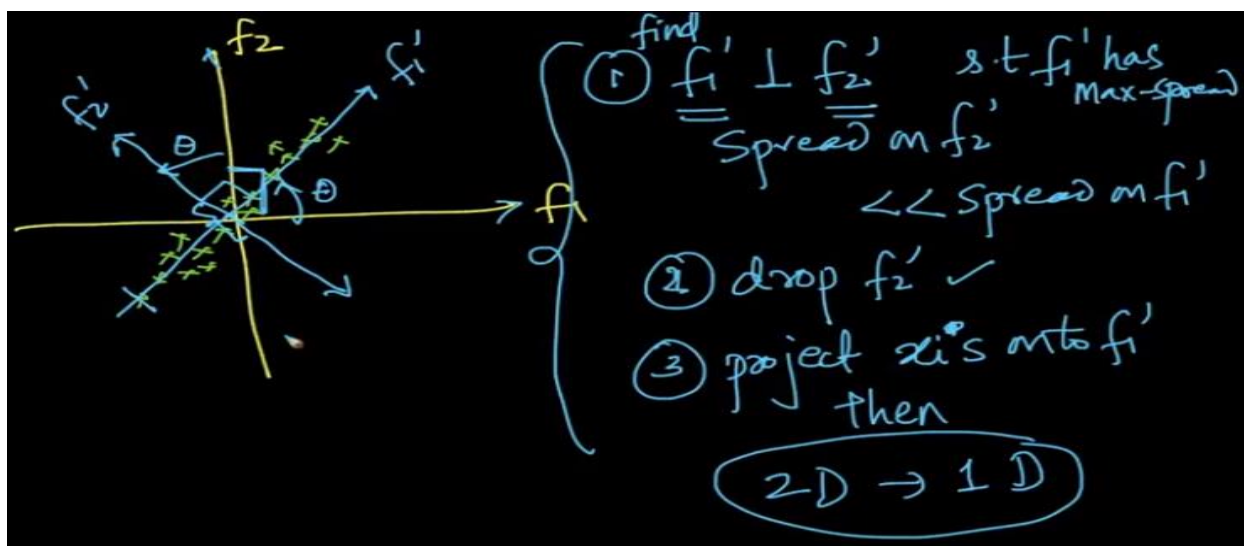
Feature  $f_2$  has most information and  $f_1$  has less. So  $f_1$  can be dropped directly.

Feature have enough spread in both the directions. Dropping a feature based on intuition is not good.

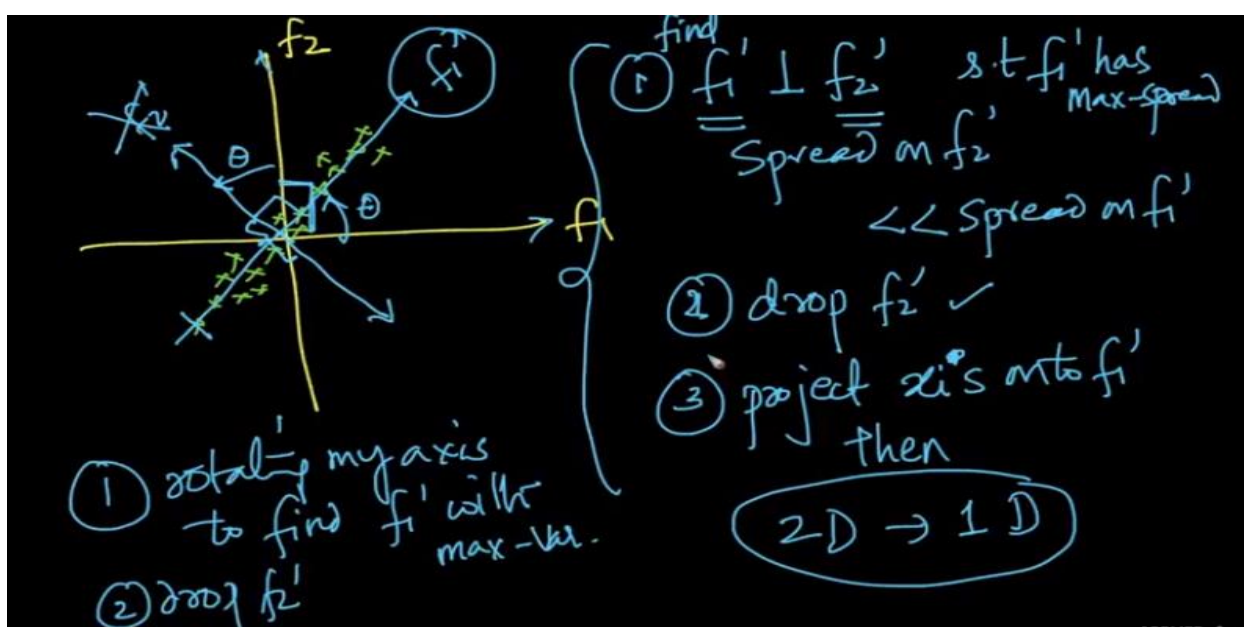




Find method for maximizing the variance of data in one direction.

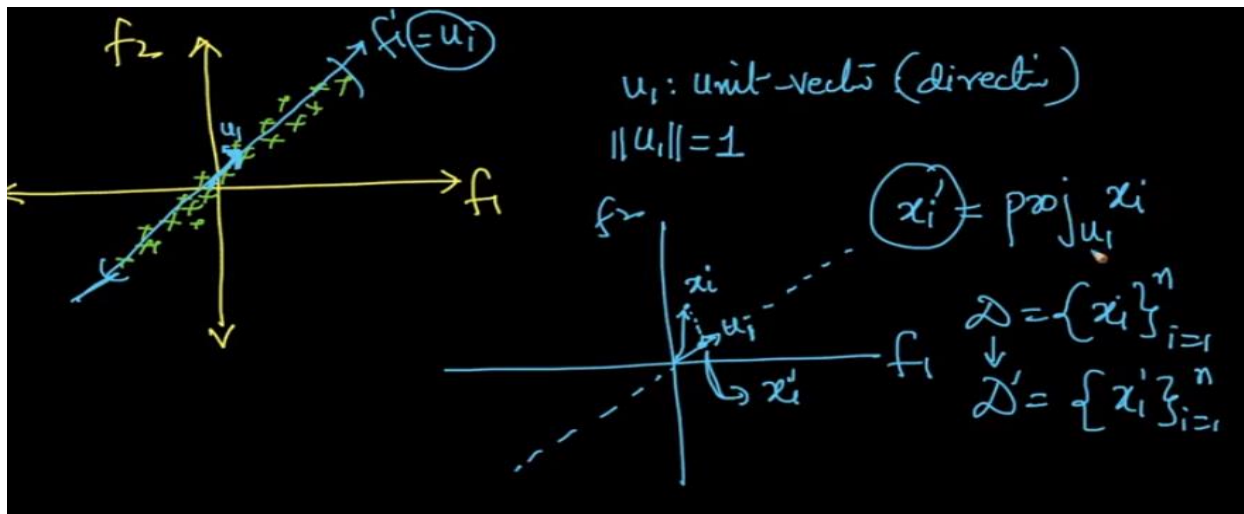


We want to find a direction  $f_1'$   
 s.t. the variance of  $x_i$ 's projected onto  
 $f_1'$  is maximal



## Mathematical Interpretation: - as optimization problem

(Maximizing Variance): Find unit vector  $U_1$  such that projection of  $X_i$  on  $U_1$  is maximized.



$$x_i' = \text{proj}_{u_1} x_i = \frac{u_1 \cdot x_i}{\|u_1\|^2 = 1} = \boxed{u_1^T x_i}$$

$$x_i' = u_1^T x_i$$

$$\checkmark \quad \bar{x}' = u_1^T \bar{x}$$

$\bar{x}' \leftarrow \text{mean}\{x_i'\}_{i=1}^n$   
 $\bar{x} \leftarrow \text{mean}\{x_i\}_{i=1}^n$

⊛ find  $u_1$  s.t.  $\text{Var}\left\{\text{proj}_{u_1} \bar{x}_i\right\}_{i=1}^n$  is maximal.

$$\text{Var}\left\{u_1^T x_i\right\}_{i=1}^n = \frac{1}{n} \sum_{i=1}^n \left( \underbrace{u_1^T x_i}_{x_i'} - \underbrace{u_1^T \bar{x}}_{\text{mean}\{x_i\}_{i=1}^n} \right)^2$$

$\text{scale} = (u_1)^T x_i \text{ (n x 1)}$   
 $X$ : Col. Standardized  
 $\checkmark \bar{x} = [0, 0, 0, \dots, 0]$

$$\text{Var} \{x_i\}_{i=1}^n = \frac{1}{n} \sum_{i=1}^n (u_1^T x_i)^2$$

objective of an optmzn problem  $\rightarrow$   $\max_{u_1} \frac{1}{n} \sum_{i=1}^n (u_1^T x_i)^2$   $\rightarrow$  optmzn problem

$\text{Var} \{x_i\}$

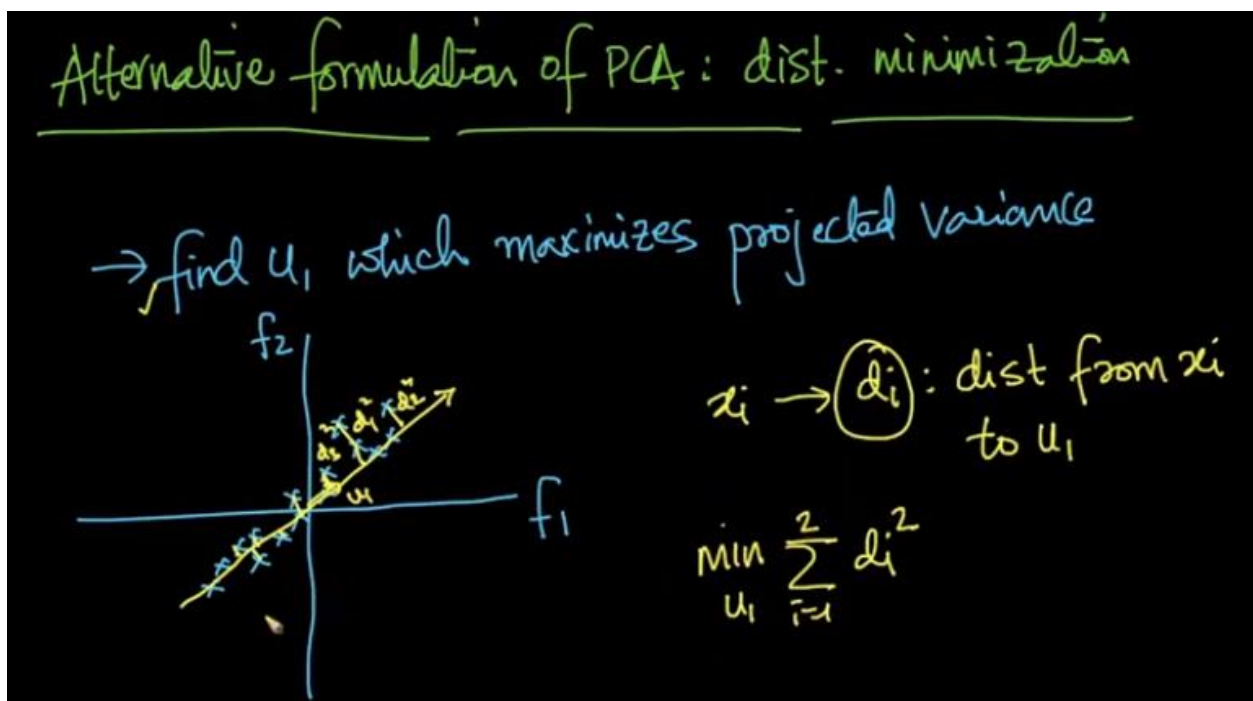
Data-matrix  $\checkmark$

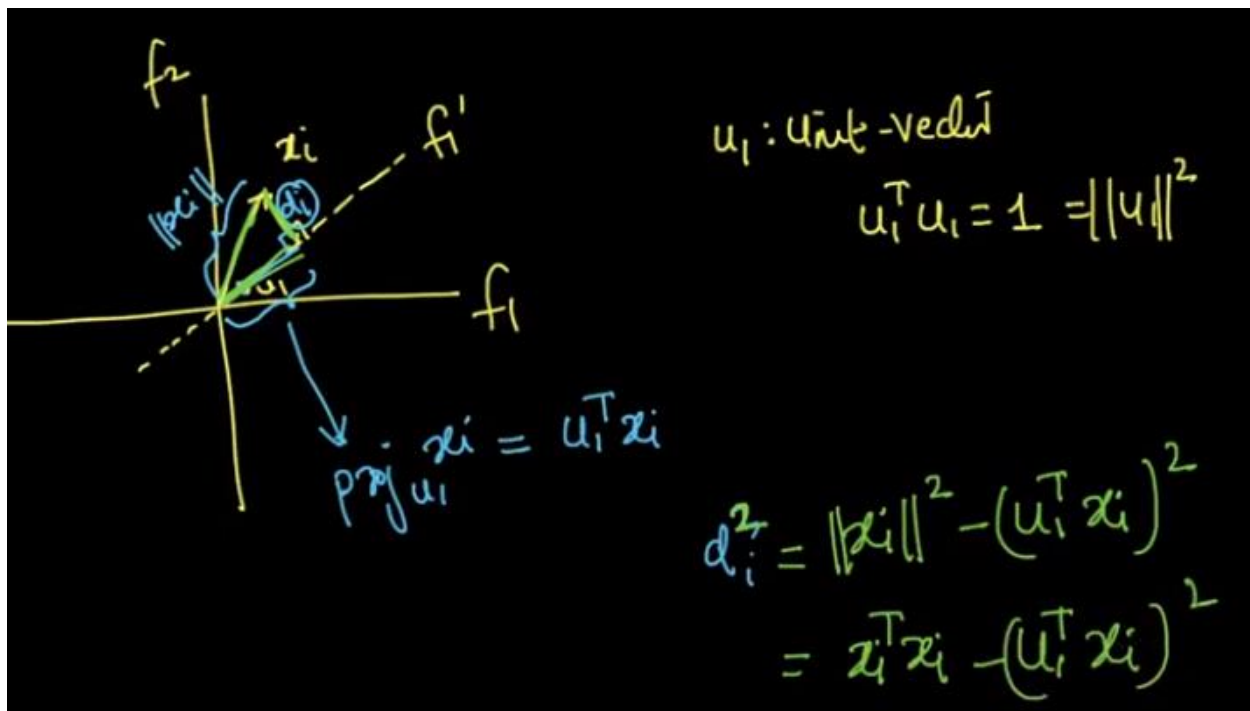
s.t.  $u_1^T u_1 = 1 = \|u_1\|^2$

Constraint  $\rightarrow u_1$  is a unit vector

$u_1 = [\infty, \infty]$

Distance Minimization: Alternate optimization statement for PCA





dist min PCA  $\left\{ \begin{array}{l} \min_{u_1} \sum_{i=1}^n \left( x_i^T x_i - (u_1^T x_i)^2 \right) \\ \text{s.t. } u_1^T u_1 = 1 \end{array} \right.$

$X = \begin{bmatrix} \leftarrow x_i^T \rightarrow \end{bmatrix}$

---

$\max_{u_1} \left\{ \frac{1}{n} \sum_{i=1}^n (u_1^T x_i)^2 \right\}$  - Variance maximization PCA  
 s.t.  $u_1^T u_1 = 1$



Solution to optimization problem:

Eigen vector (direction of the maximal variance)

Eigen values (percentage of variance explained)

Solution to our Optimization problems:  $\lambda_1, V_1$

$X = \frac{1}{2} \begin{bmatrix} 1 & 2 & 3 & \dots & d \\ \vdots & & & & \\ n & & & & \end{bmatrix}$   $n \times d$

Est. std.

Covariance matrix of  $X = S$

$S_{d \times d} = X_{d \times n}^T X_{n \times d}$

Sq. Symm. matrix

eigen-values ( $\lambda_1, \lambda_2, \dots, \lambda_d$ )

eigen-vector ( $V_1, V_2, \dots, V_d$ )

$S_{d \times d}$

maximal eigen-value  $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4 \dots > \lambda_d$

eigen-values of  $(S) = \lambda_1, \lambda_2, \lambda_3, \lambda_4, \dots, \lambda_d$

eigen vectors of  $(S) = V_1, V_2, V_3, V_4, \dots, V_d$

def:  $\lambda_1 V_1 = S V_1$   $d \times 1$  vector

$\lambda_1$ : eigen value of  $S$

$V_1$ : eigen vec to  $S$

$$\lambda_1 > \lambda_2 > \lambda_3 \dots > \lambda_d$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \dots \quad \downarrow$$

$$v_1, v_2, v_3, \dots, v_d$$

$S_{d \times d}$

✓  $\boxed{v_i \perp v_j} : v_i^T v_j = 0 = v_i \cdot v_j = 0$

✓  $\textcircled{u_1} = v_1 = \text{eigen-vector of } S (= X^T X)$   
 corr. to largest eigen-value ( $= \lambda_1$ )  
 max-variance direction

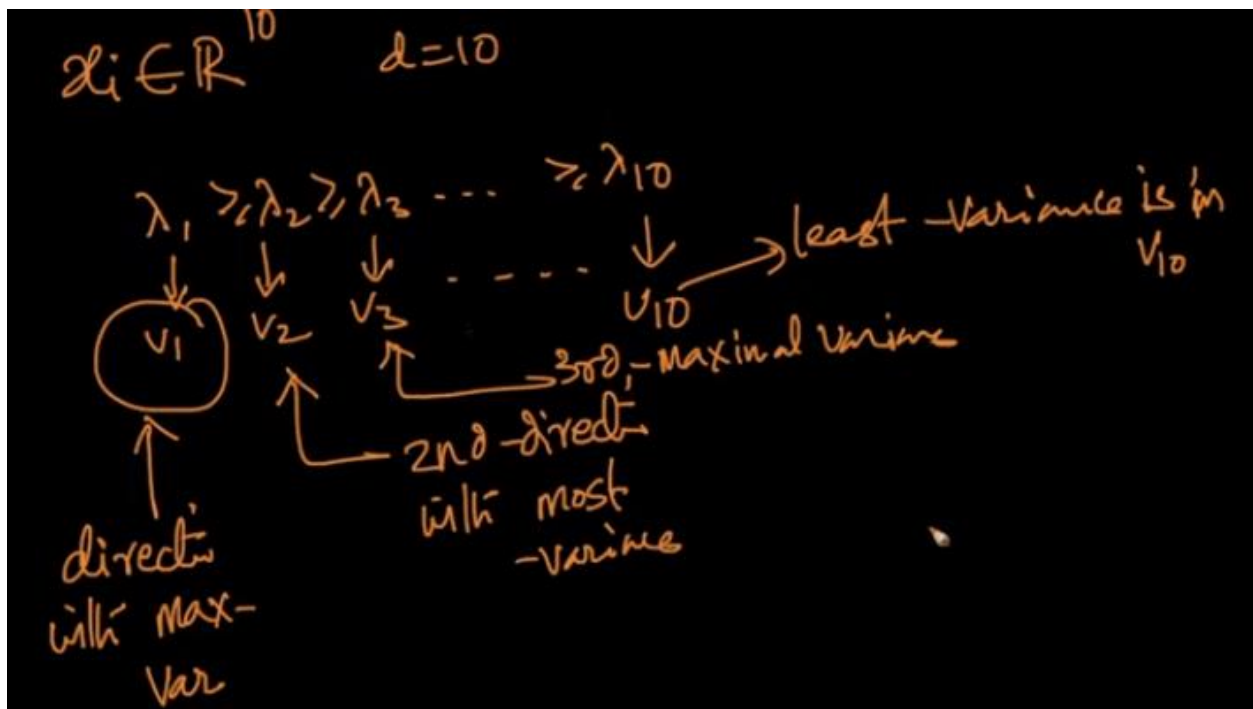
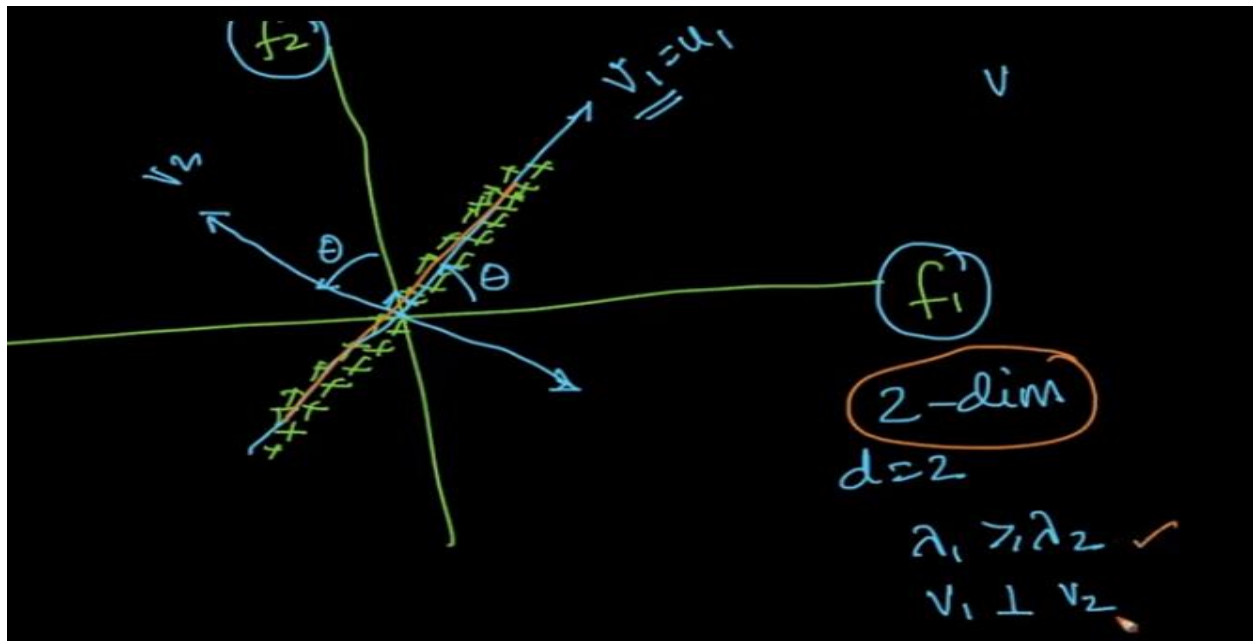
$$X = \begin{bmatrix} \checkmark \end{bmatrix}_{n \times d}$$

① Col. std of  $X$  is done

②  $S_{d \times d} = X^T X$

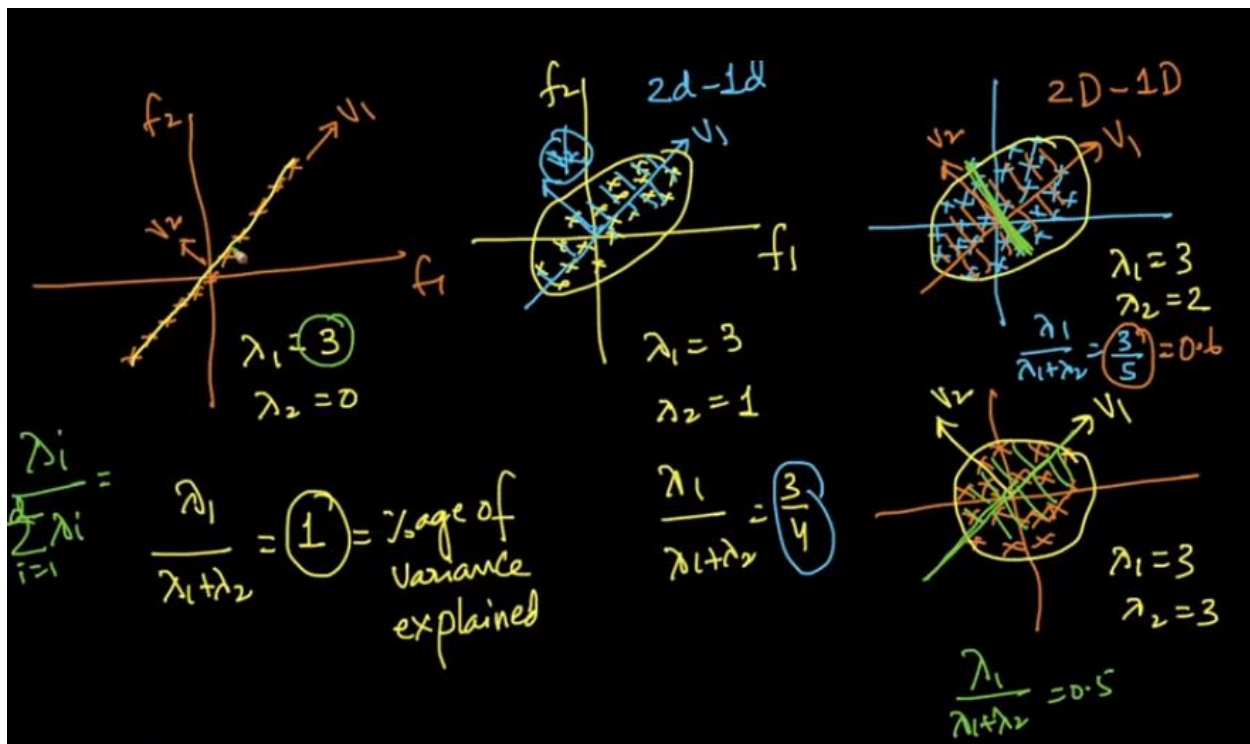
③  $\underbrace{\text{eigen}(S)}_{\text{eigen values \& vectors of } S}$   
 $\lambda_1 > \lambda_2 > \dots > \lambda_d$   
 $v_1, v_2, \dots, v_d$

④  $u_1 = v_1$  (why?)



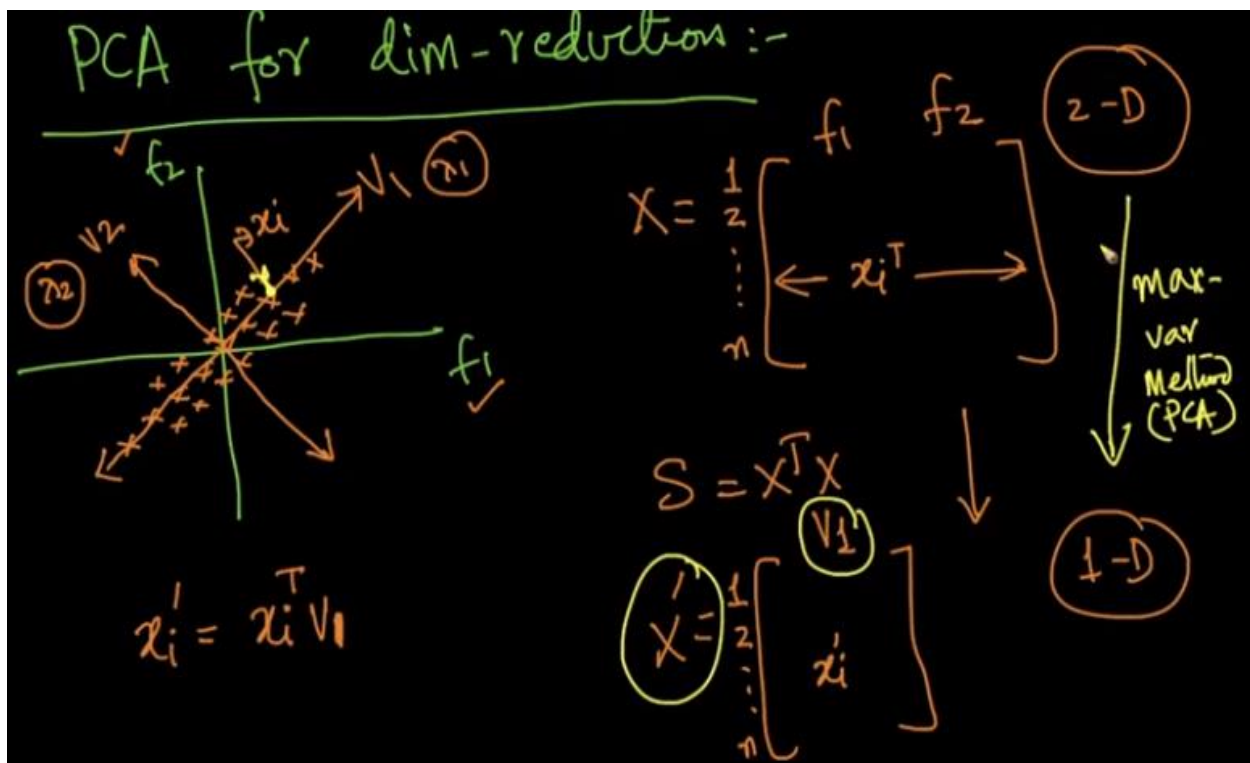
### Projecting the data and variance explained: -

Eigen vector explained the direction of maximum variance and lambda (eigen values) explains the percentage of variance on the projected eigen vector ( $v_1$ )



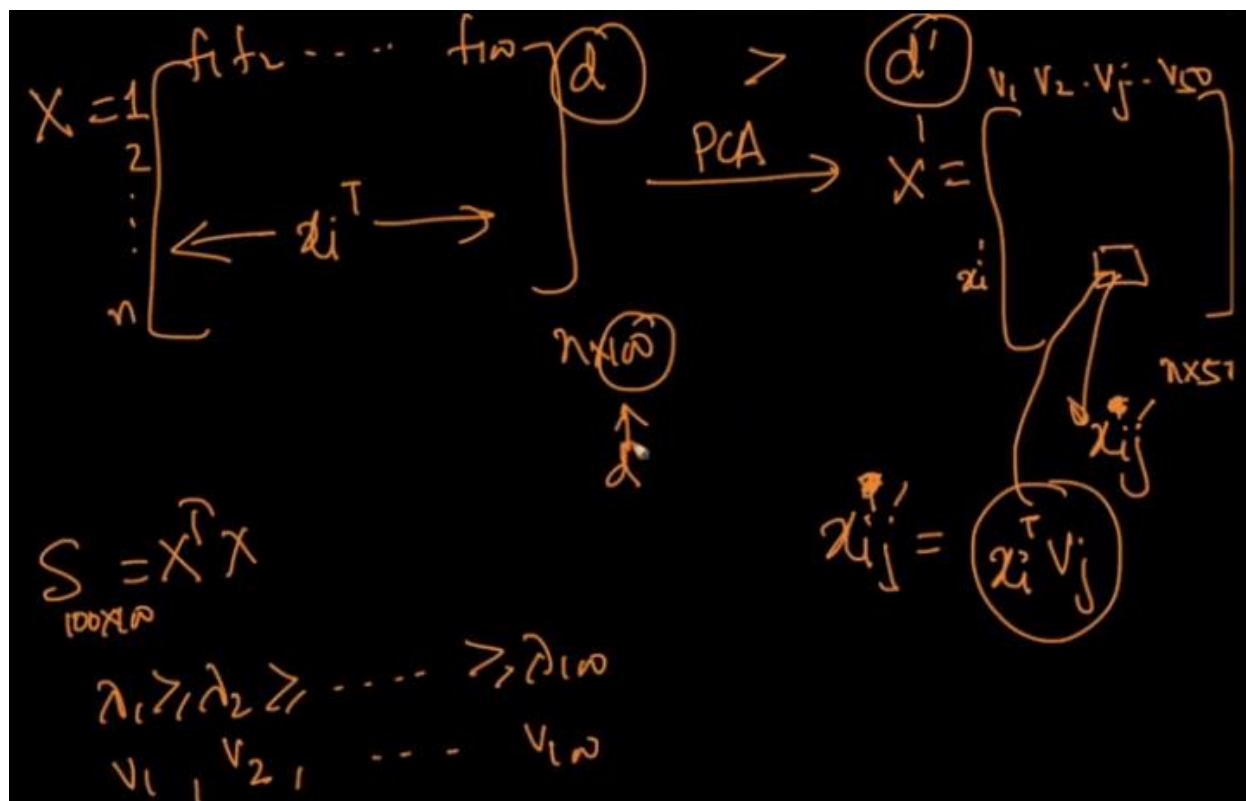
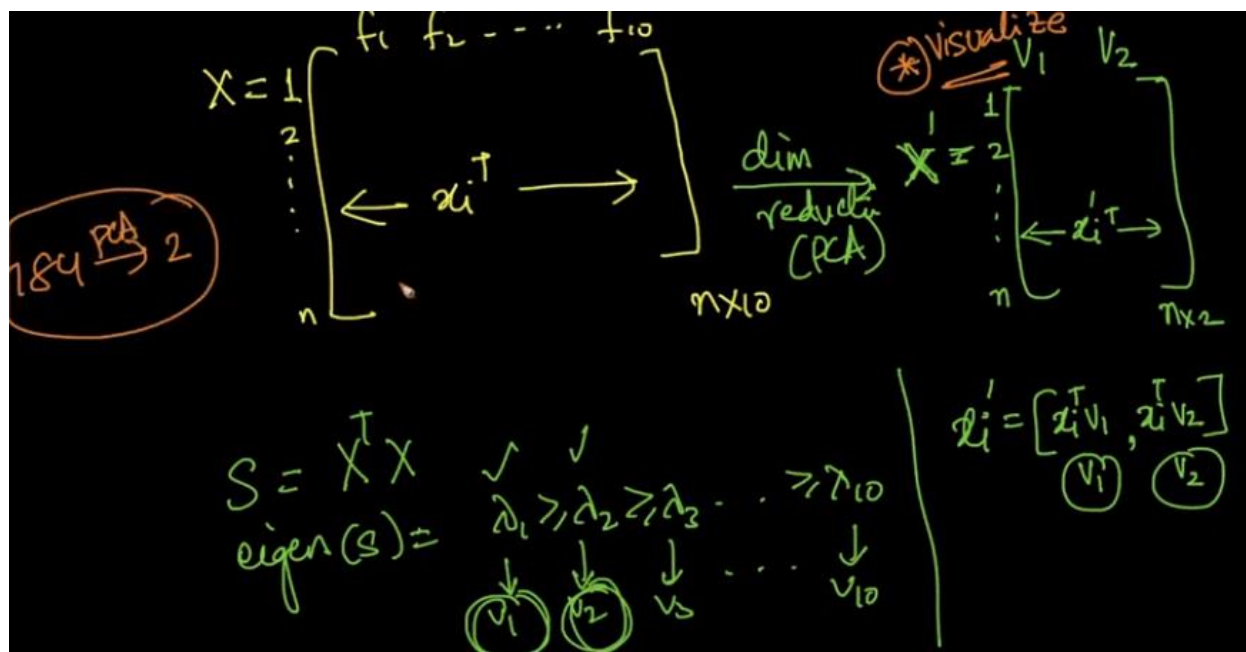
Idea to reduce dimension using PCA: -

(A) Take maximum variance eigen vector as feature ( $X'_i = X_i^T V_1$ )





(B) Take maximum variance eigen vector as feature ( $X'_i = X_i^T V_1$ ,  $X'_i = X_i^T V_2$ )



Eigen values (measure of maximum variance) – select feature to retain the amount of variance.

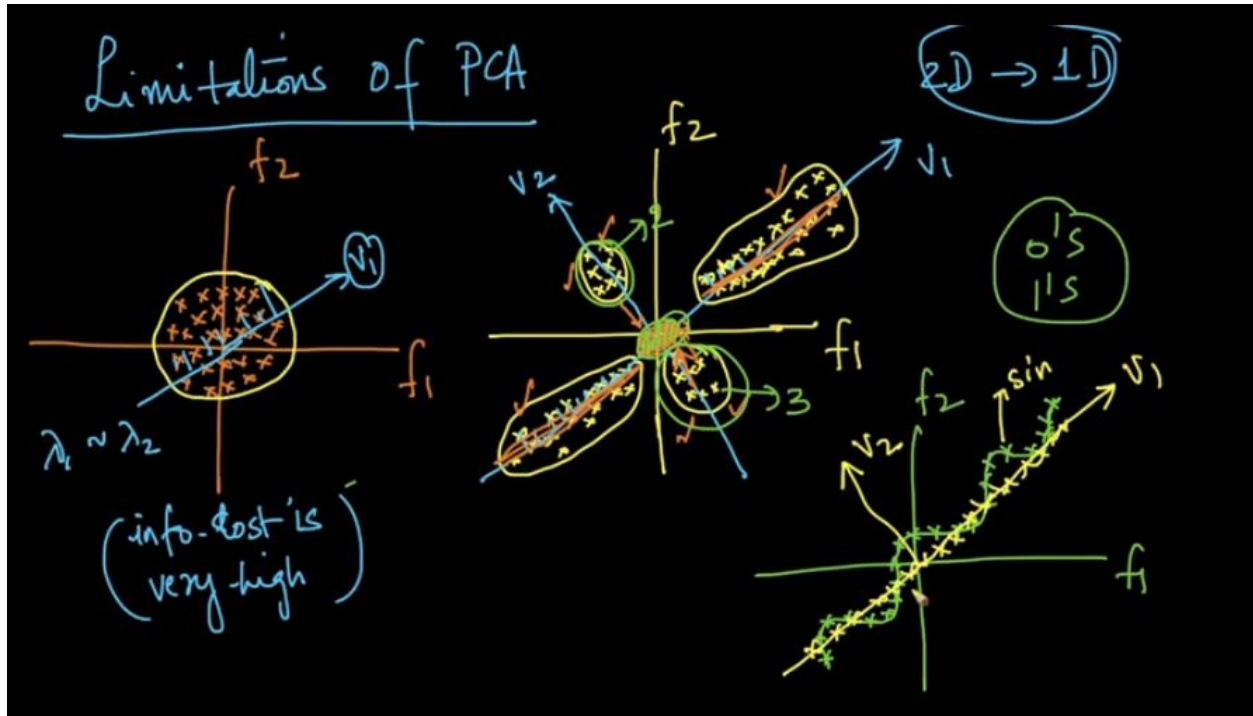
$$x_i \in \mathbb{R}^{100=d} \quad ; \quad x_i' \in \mathbb{R}^{d'}$$

$d \xrightarrow{\text{PCA}} d'$  ✓ preserve 99% of the variance  $d' < 100$

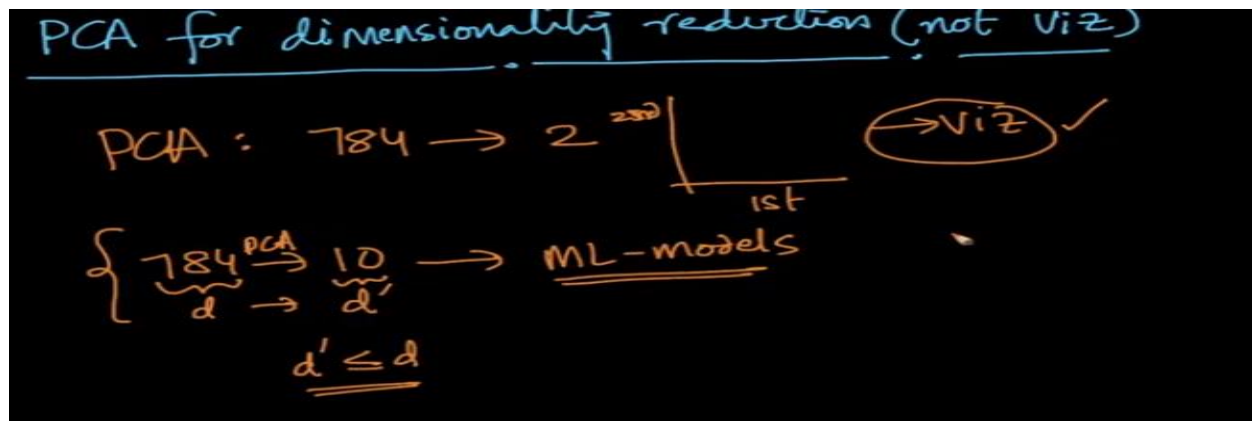
let  $\frac{\lambda_1 + \lambda_2 + \dots + \lambda_{51}}{\sum_{i=1}^{100} \lambda_i} = 0.99$  ✓

$d' = 51$

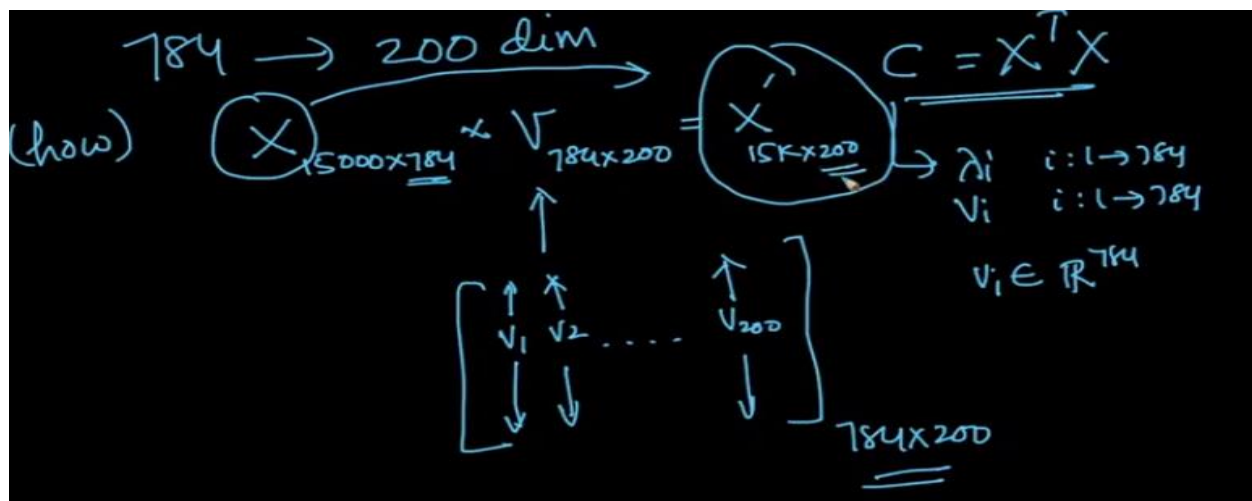
Limitations of PCA: - information loss for types of data spread



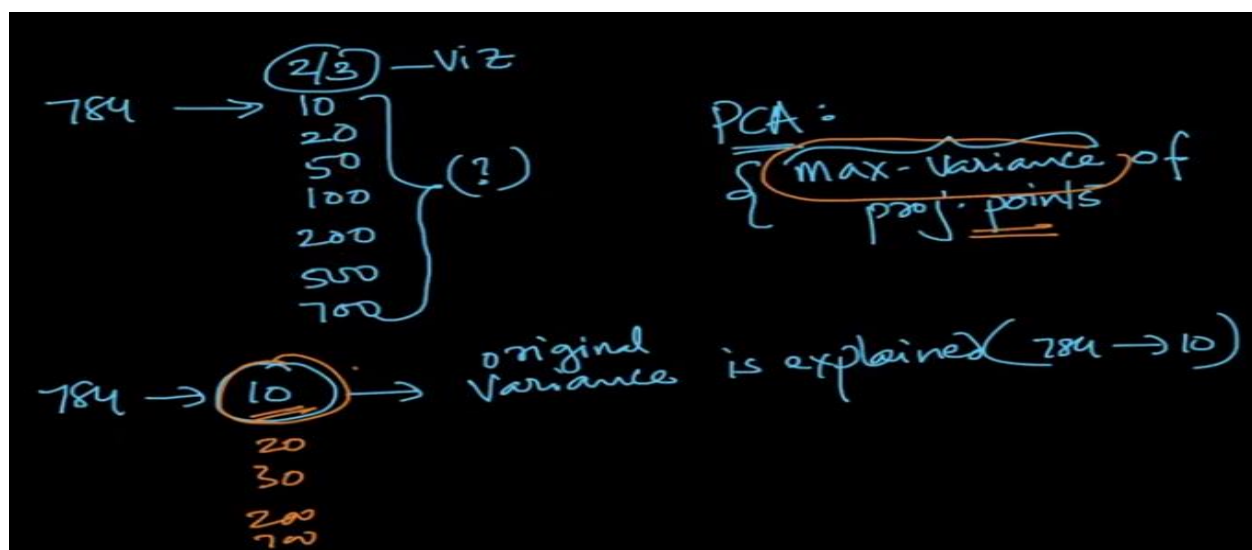
# Machine Learning Model (Dimensionality Reduction)



Convert into lower dimension



What dimension to choose – maximum variance retained



PCA:

$$C = X^T X$$

$784 \times 784$

$\lambda_i, V_i$

$\lambda_1, \lambda_2, \dots, \lambda_{784}$

$784 \rightarrow 10 \text{ dim}$

percentage of Variance explained in 10-dim

$$= \frac{\lambda_1 + \lambda_2 + \dots + \lambda_{10}}{\sum_{i=1}^{784} \lambda_i}$$

20% of the total variance in 784-dim is explained in 10-dim

Find  $d'$  which retains 90% variance explained.

$784 \xrightarrow{\text{PCA}} d'$

90% of info/Variance

$$\left\{ \frac{\lambda_1 + \lambda_2 + \dots + \lambda_{d'}}{\sum_{i=1}^d \lambda_i} = \underline{\underline{0.9}} \right.$$



```

# PCA for dimensionality redcution (non-visualization)

pca.n_components = 784
pca_data = pca.fit_transform(sample_data)

percentage_var_explained = pca.explained_variance_ / np.sum(pca.explained_v
cum_var_explained = np.cumsum(percentage_var_explained)

# Plot the PCA spectrum
plt.figure(1, figsize=(6, 4))

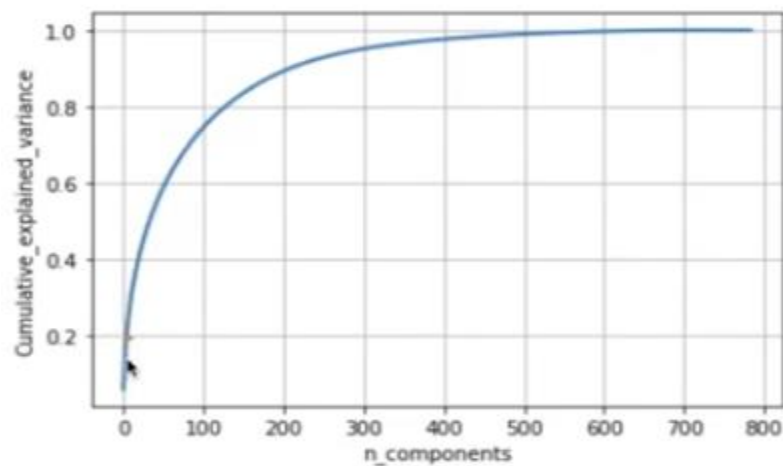
plt.clf()
plt.plot(cum_var_explained, linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cumulative explained variance')
plt.show()

```

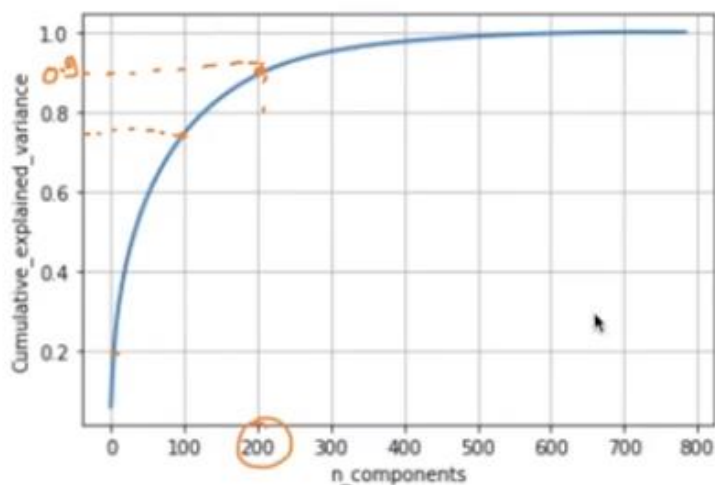
$$\frac{\lambda_i}{\sum \lambda_i} \quad \forall i$$

$$\frac{\lambda_1}{\sum \lambda_i}, \frac{\lambda_2}{\sum \lambda_i}, \frac{\lambda_3}{\sum \lambda_i} \dots$$

$$\frac{\lambda_1}{\sum \lambda_i}, \frac{\lambda_1 + \lambda_2}{\sum \lambda_i}, \frac{\lambda_1 + \lambda_2 + \lambda_3}{\sum \lambda_i} \dots$$



90% variance explained by 200 principle components.



784  $\rightarrow$  100 (~0.75)  
 784  $\rightarrow$  d' (~90%)  
 PCA  $\rightarrow$  200