Visualizing data:

For 1-D -> Matplotlib

For 2-D,3-D -> Scatter Plots

For 5-D,6-D -> Pair Plots

For n-D? Difficult to visualize so we try to reduce the dimensionality of the data to 2D or 3D , the procss of coverting a Higher Dimensionality Data into a 2D or 3D or just simply lowering the Dimensions is collectively termed as dimensionality Reduction

2 Important Dimensionality Reduction Techniques :

1. Principal Component Analysis
2. t-SNE(t-distributed Stochastic Neighborhood Embedding)

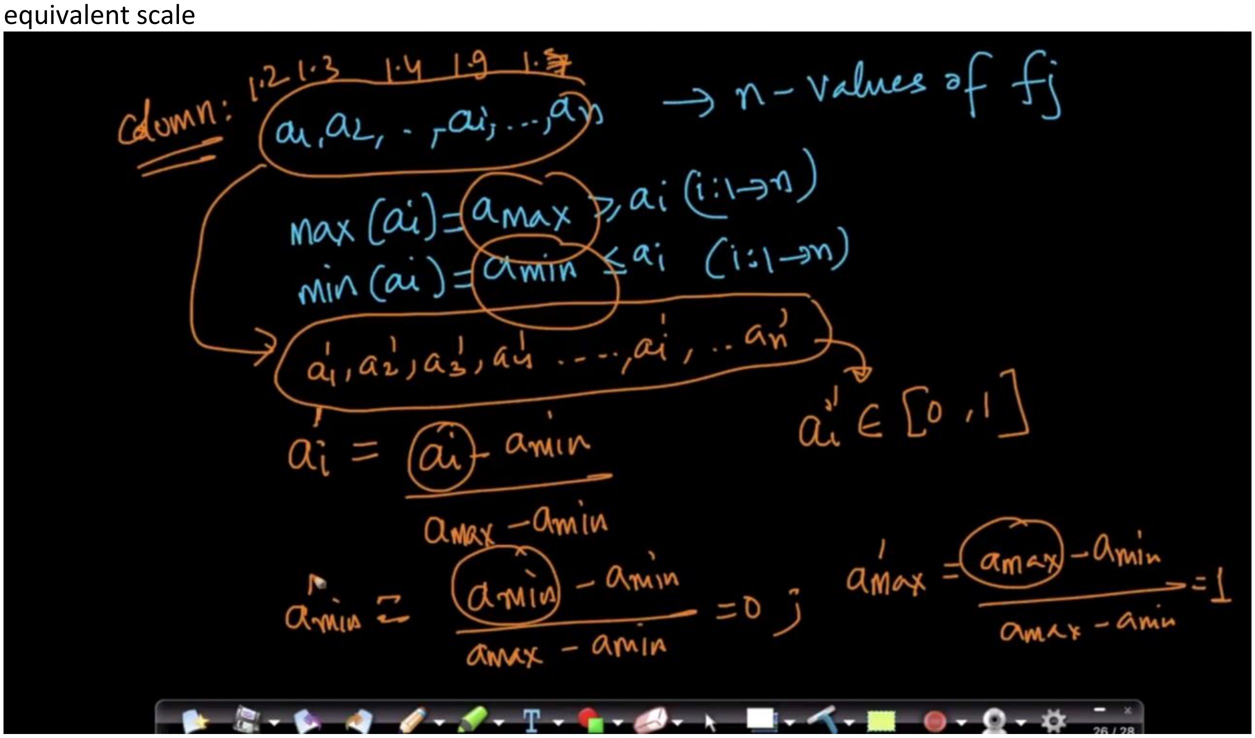
Row Vector and Column Vector:

By convention, any element of a vector space E is represented by a column vector.

A column vector is an (n×1) [1]matrix, and a row vector is a (1×n) matrix

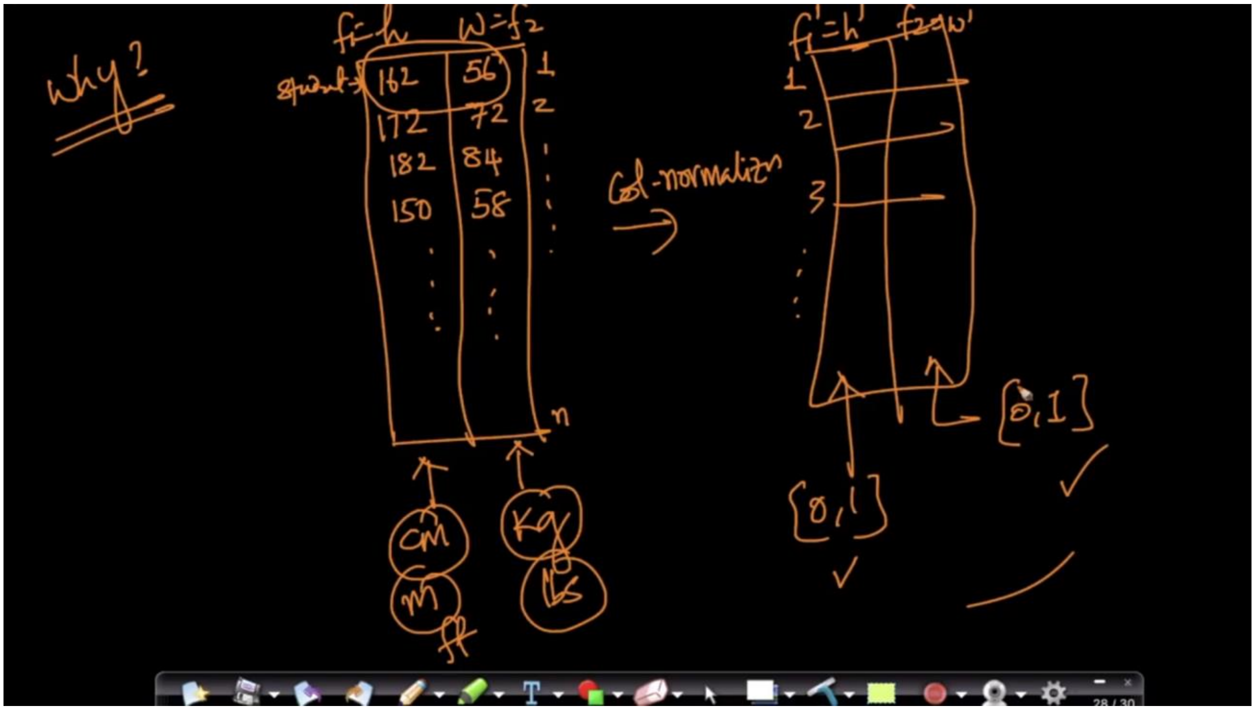
Data Preprocessing : Column Normalization:

The purpose of normalization is, primarily, to scale numeric data from different columns down to an equivalent scale

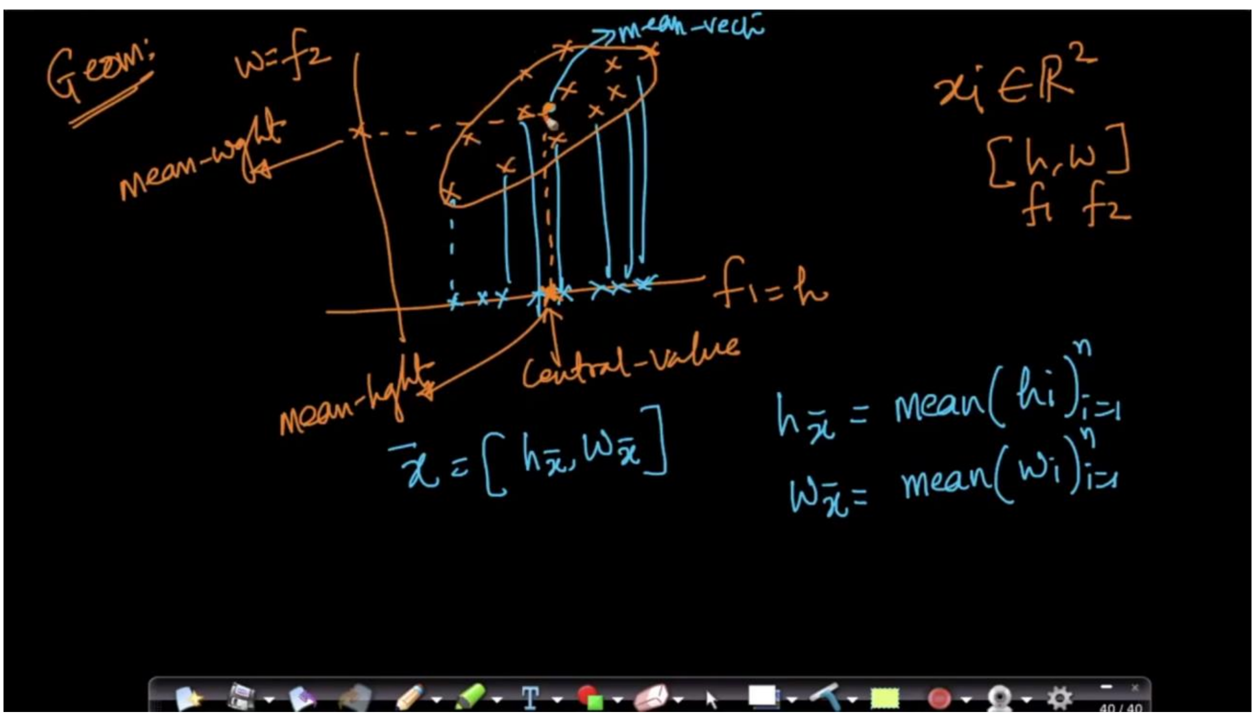


Why Column Normalization?

With Column Normalization we are getting rid of scale, we are putting all the values of a column between 0 and1 as shown in above diagram



Mean Vector: Central Vector



Column Standardization:

Geometrically, column standardization means squishing the data points such that the mean vector comes at origin and the variance (by either squishing or expanding) on any axes would be 1 in the transformed space.

it will transform features ***coming from any distribution*** so that, it will have zero mean and unit variance.
in other words
Xnew = (x - mean(D)) /std-deviation(D)
where,
Xnew = standardized value of feature $\mathbf{x}$
D = set of all the values of feature x

Usage of normalization instead of standardization:

Normalization is better if we want all resulting values in the interval [0,1] as Standardization can result in any value, both positive and negative.

Usage of Standardization instead of Normalization:

Standardization is better when we have outliers as outliers will have large negative or positive values while inliers will have values around 0. Normalization (using min and max) in the case of data with outliers could result in outliers having values closer to 0 and 1 and most inliers concentrated in a small band of values.

Downside of normalization:

Min-max normalization has one important downside that it does not handle outliers with ease. Lets assume we have 99 values between 0 and 30, and one value is 110, then the 99 values will all be transformed to a value between 0 and 0.4. That data is just as squished as before!

Covariance:

The covariance of two variables (x and y) can be represented as cov(x,y). If E[x] is the expected value or mean of a sample 'x', then cov(x,y) can be represented in the following way:

$$\begin{aligned} \mathrm{cov}(x,y) &= E[(x-\mu_x)(y-\mu_y)] \\ &= E[xy] - E[x]E[y] \\ &= E[xy] - \mu_x\mu_y \end{aligned}$$

$\forall\ \mu_x$ & μ_y = $E[x]$ & $E[y]$ respectively.

If we look at a single variable, say 'y', cov(y,y), the expression can be written in the following way:

$cov(y, y) = Var(y) = \sigma^2(y) = \sigma_y^2$

Here, Var → variance of variable y

$$\boxed{Var(y) = E[(y - \mu_y)^2]}$$

Also; $Var(y) = s^2$

where s^2 ⇒ sample variance

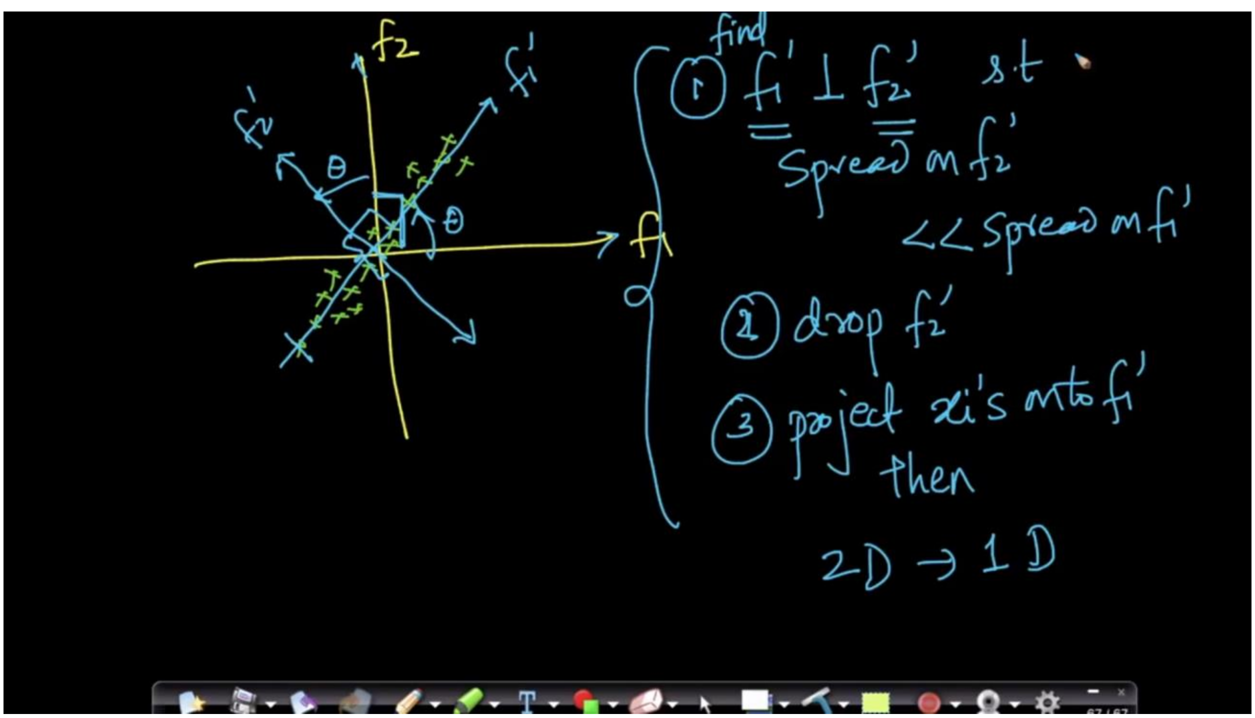
PCA: Traditional method to convert N-dimensional Data into 1D or 2D data

The Geometrical Intuition of PCA

Step 1: Standardize all values of feature f1 and f2 and plot them against each other

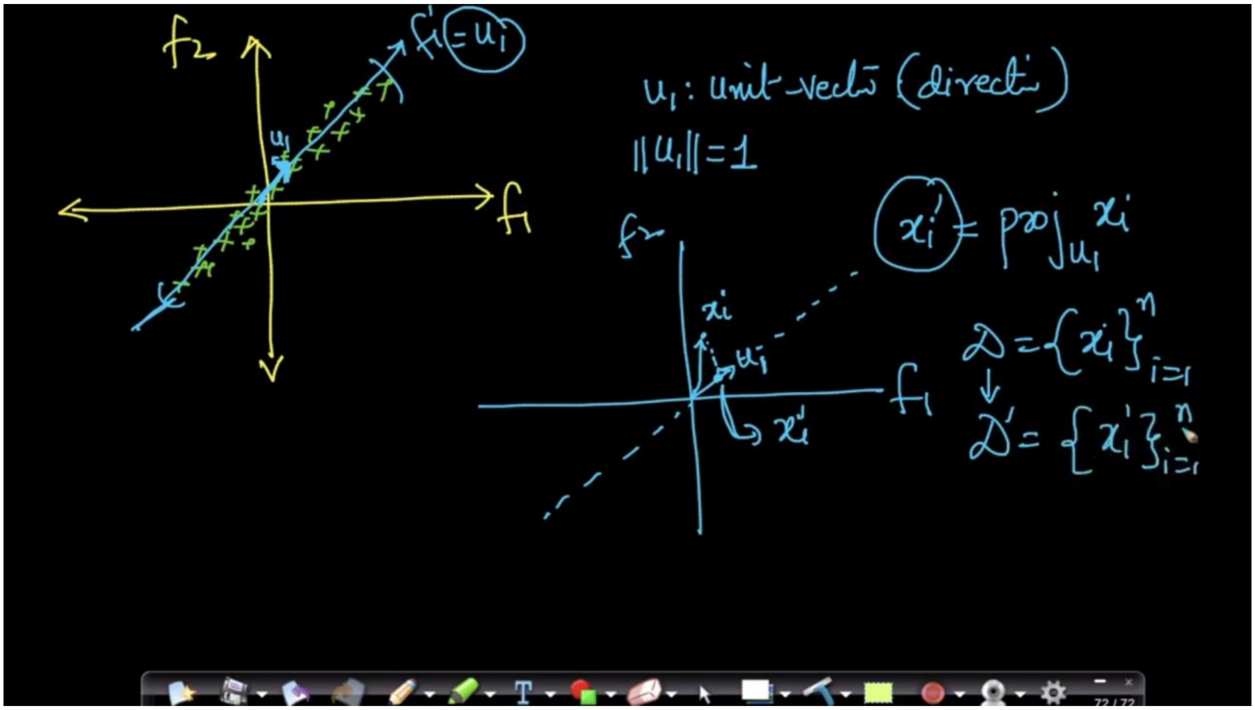
Step 2: find f1' and f2' by rotating an angle theta such that spread on f2' is much smaller than spread on f1'. Drop f2' and keep f1' as it gives max variance/information

Step 3: Project xi's to f1' then 2D to 1D

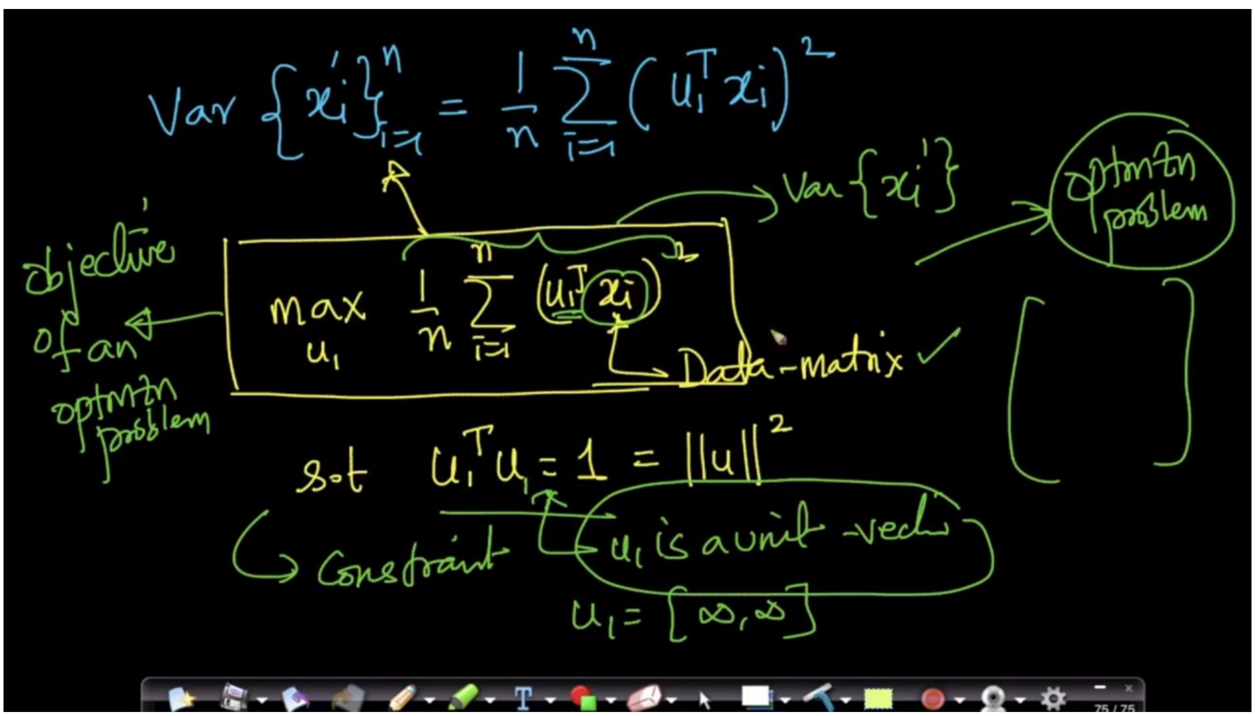


Mathematoical Intuition of PCA:

1. **Find u1 so that var{proj xi}, where i=1 to n is maximal.**

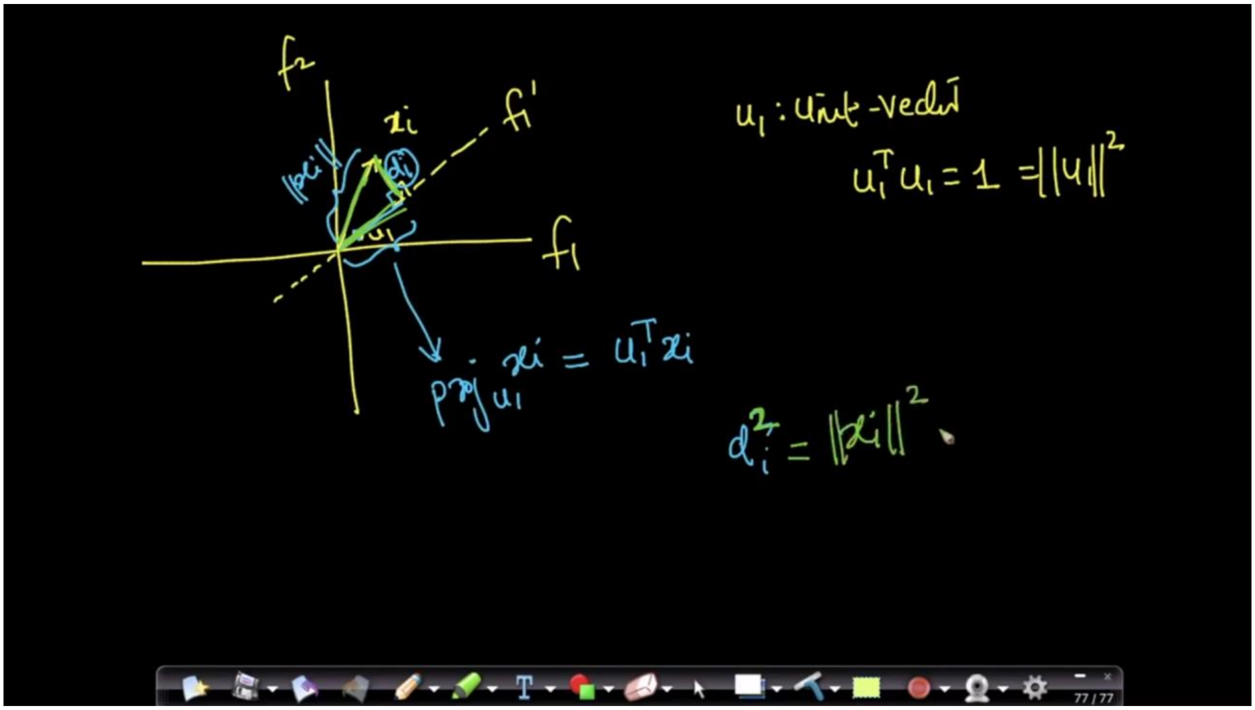


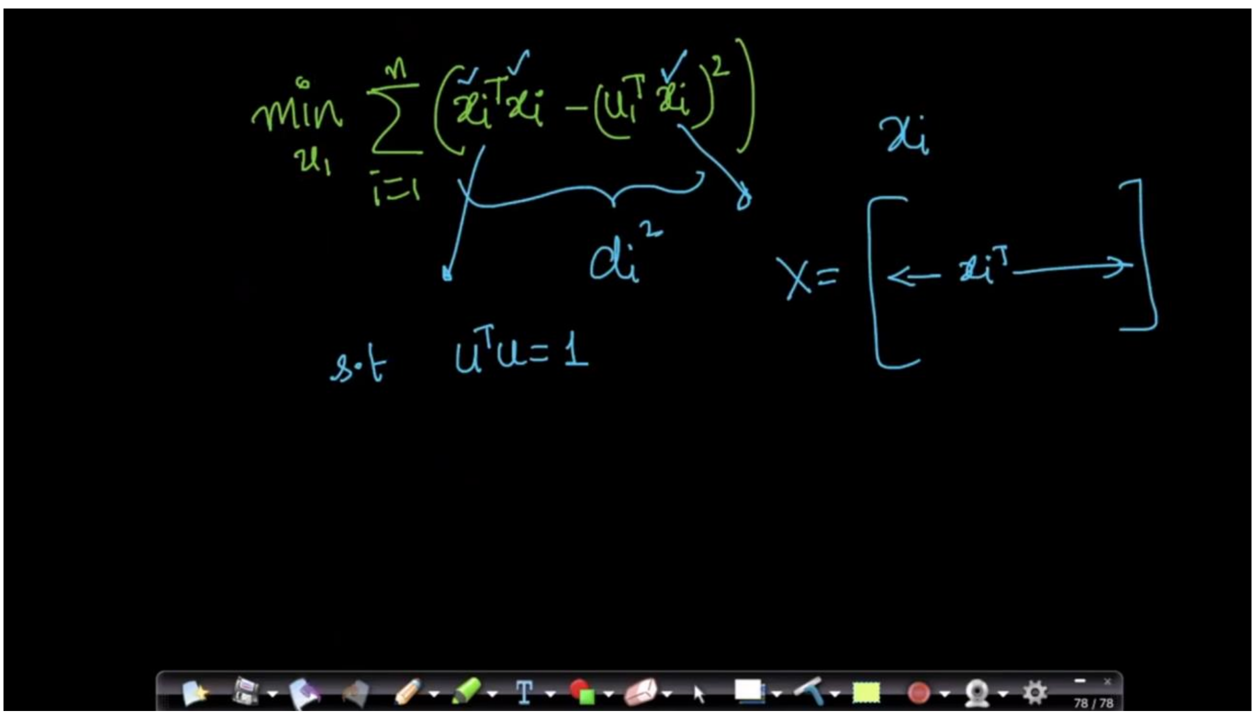
We have to find u1 which maximizes u1 as illustrated below



Alternative formulation of PCA: Distance Maximization

We have to minimize distance di such that

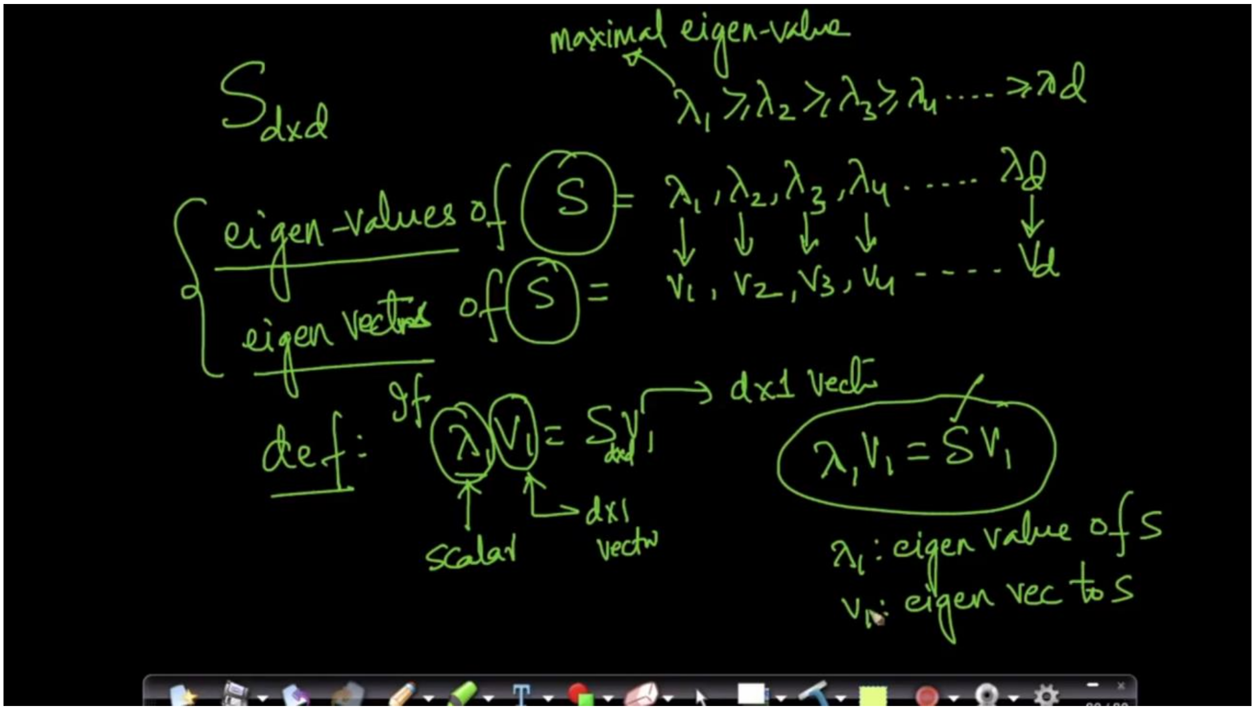




Eigen Values and Eigen Vectors:

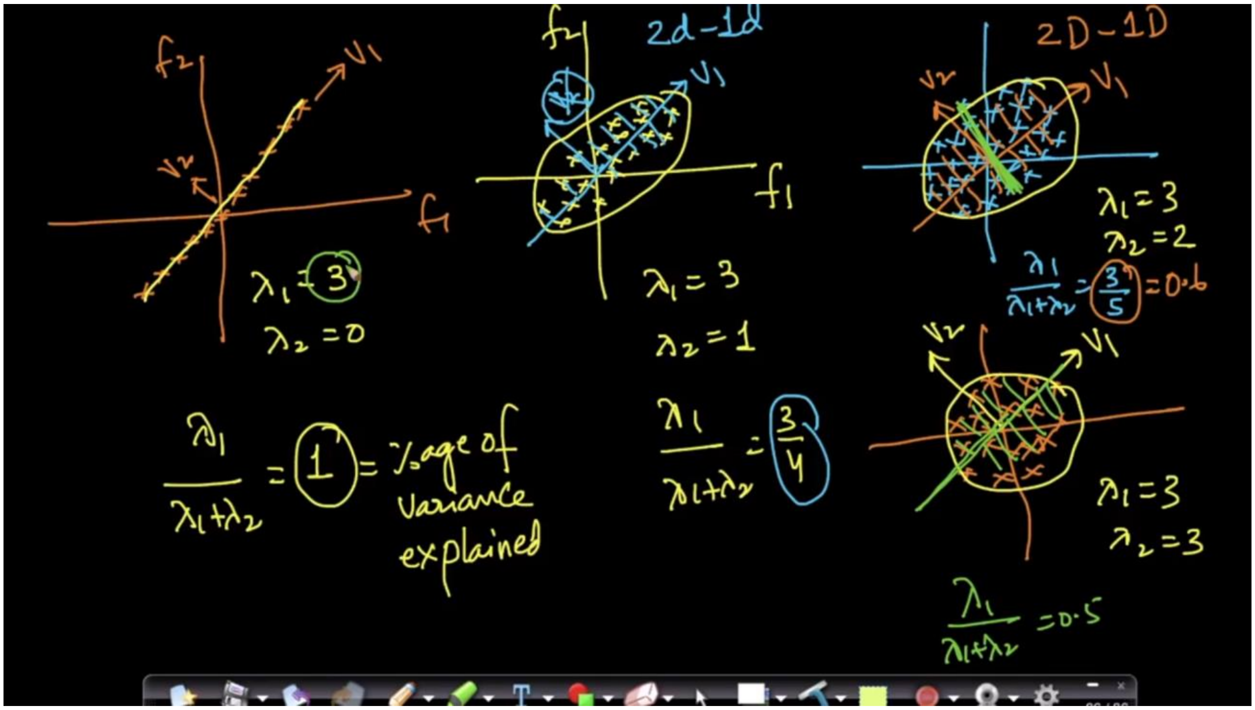
If S is a d*d matrix, lambda1 is eigen value and V1 is corresponding eigen vector, then

Eigen Vectors are perpendicular to each other eg, V1 _|_V2, V2_|_V4 etc



Steps to be followed

1. **Column Standardization of X is done**
2. **S = X^t . X, where S= co-variance matrix of dxd**
3. **Eigen Values and Eigen vectors of X such that λ1> λ2> λ3..> λd and v1>v2…vd**
4. **U1 = v1**



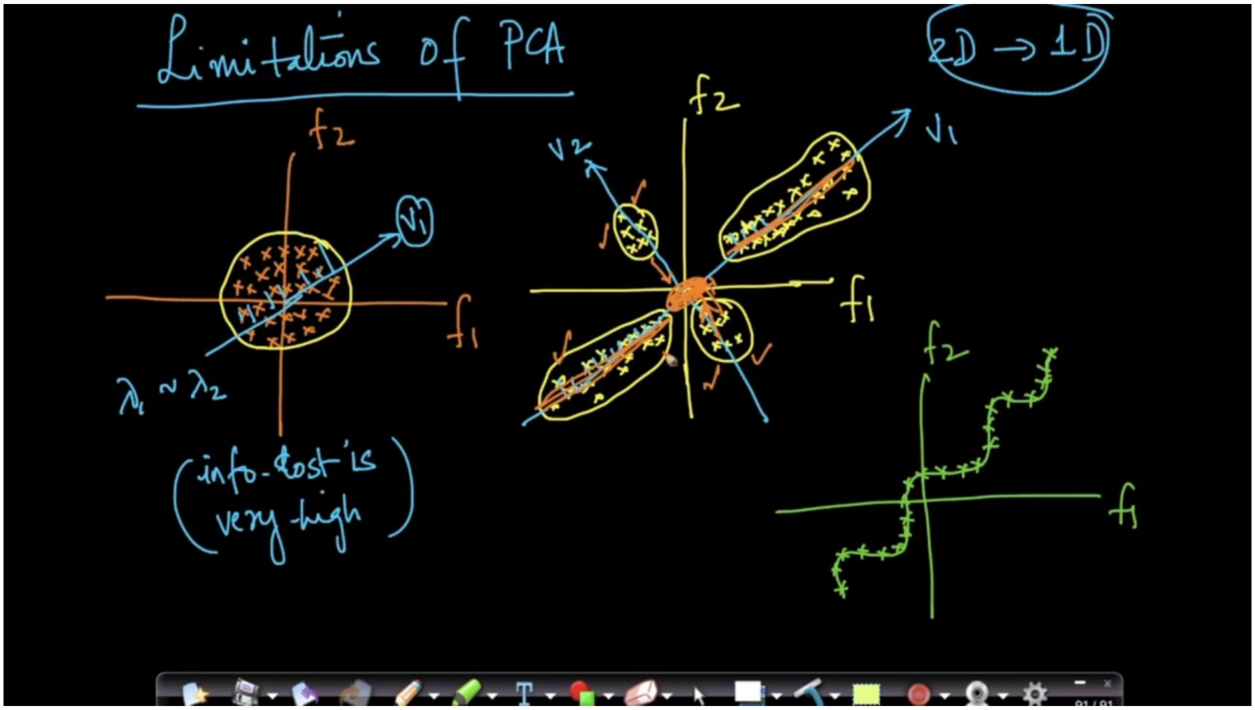
Suppose we have 2 D data which we want to project in 1D, in that case,

Case 1: If all points lie on v1, then we retain maximum information as shown in fig. 1

Case 2: if the distribution of datapoints is as in fig. 2, we retain 75% information

Case 3: if λ1 = 3, λ2 = 2, we will retain 60% of data as shown in figure 3

Case 4: if λ1=3 , λ2 = 3, we will retain 50% of the information when projecting data from 2D to 1D



T-SNE:

State of the art Dimensionality Reduction Technique, also good for visualizing Data

Other techniques for Dimensionality Reduction: Multi-Dimensional Scaling, Sammon Mapping, Graph Based techniques.

PCA reserves Global structure, T-SNE preserves local structure