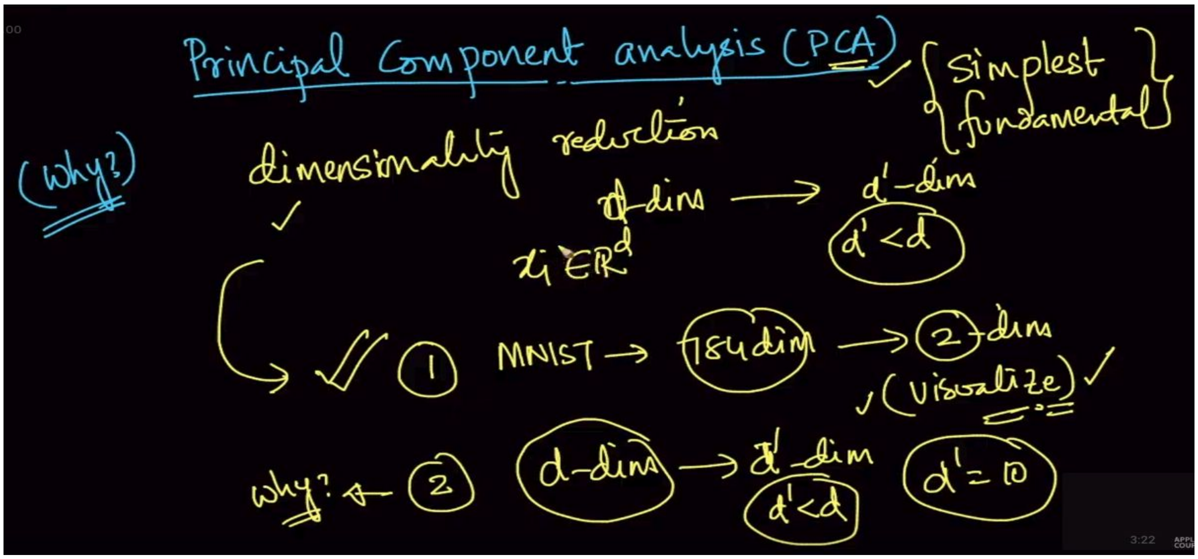
PRINCIPAL COMPONENT ANALYSIS

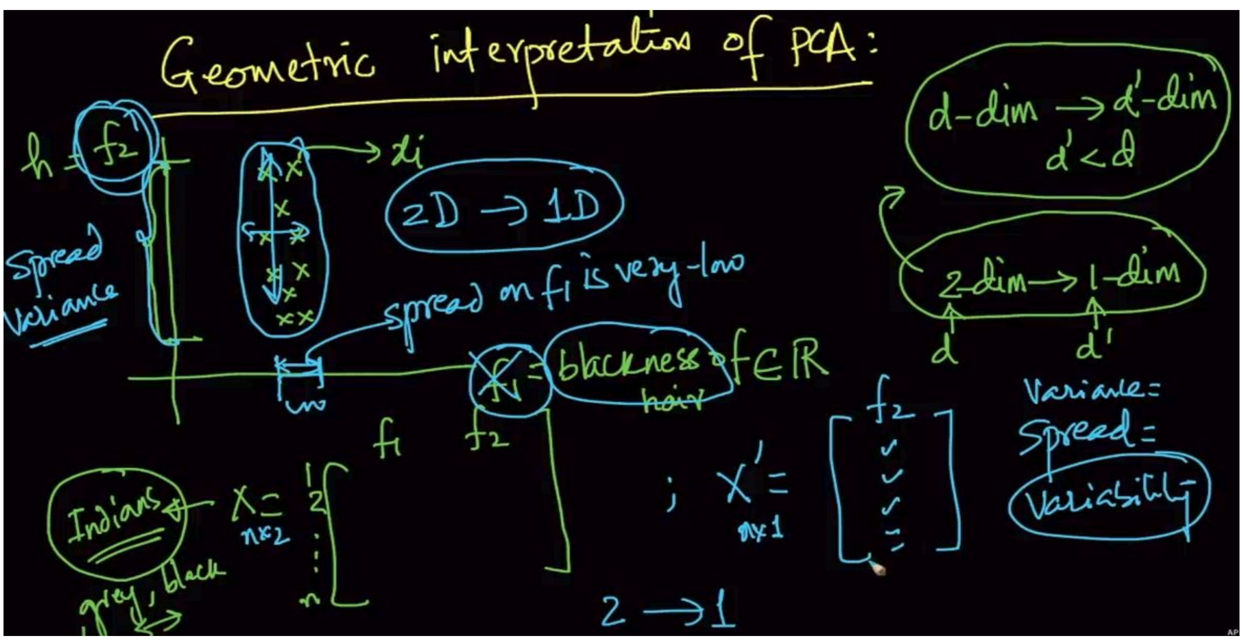
PRINCIPAL COMPONENT ANALYSIS

WHY PCA?

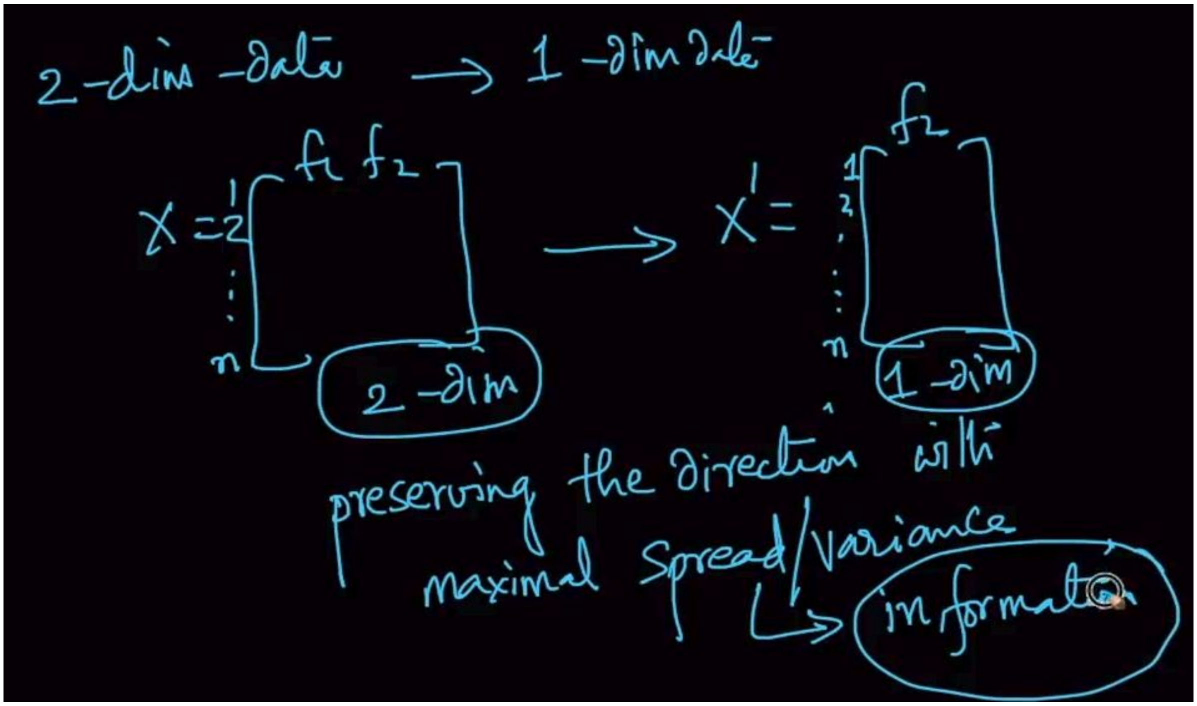


It is used for dimensionality reduction . Can be used to get the most important components with the reduced dimensions. d-dim -> d'-dim where d' < d

GEOMETRIC INTERPRETATION OF PCA

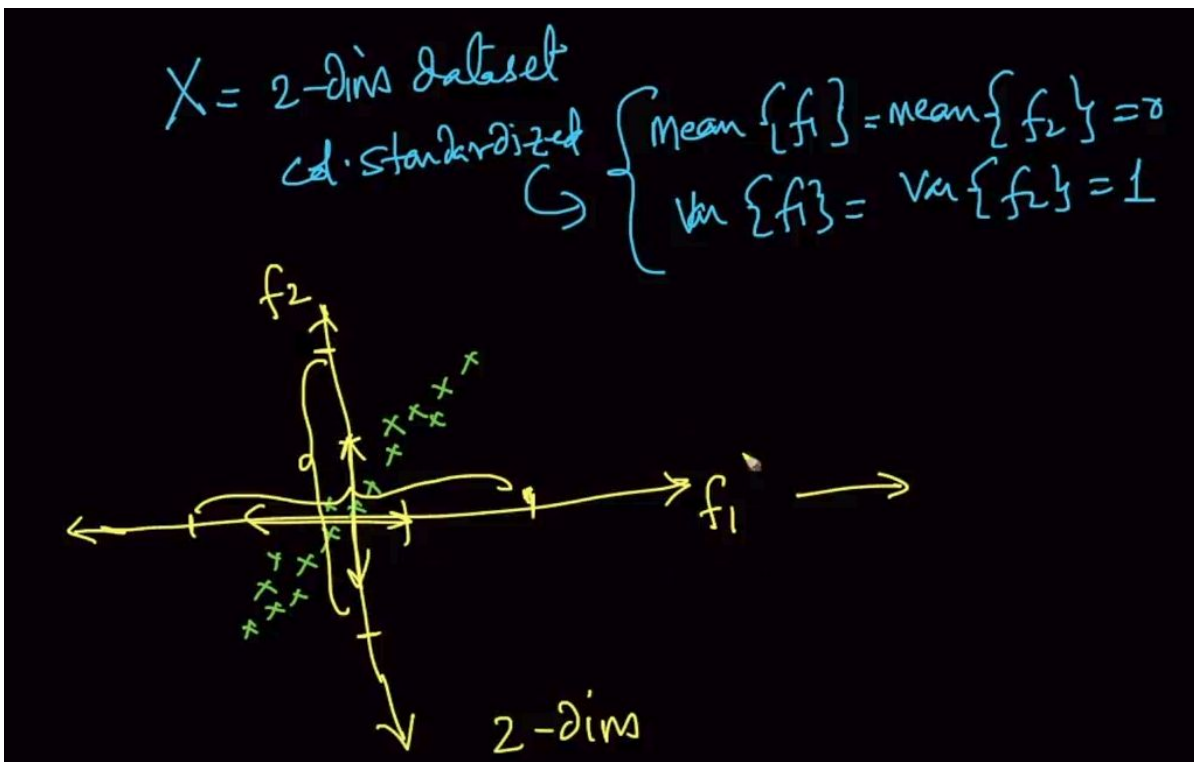


Here we've two features f1 and f2 and suppose we are supposed to go from 2d to 1d we'll check the dataset. It's clear that spread in f2 is clearly far while f1 has very low spread . So we'll choose f2

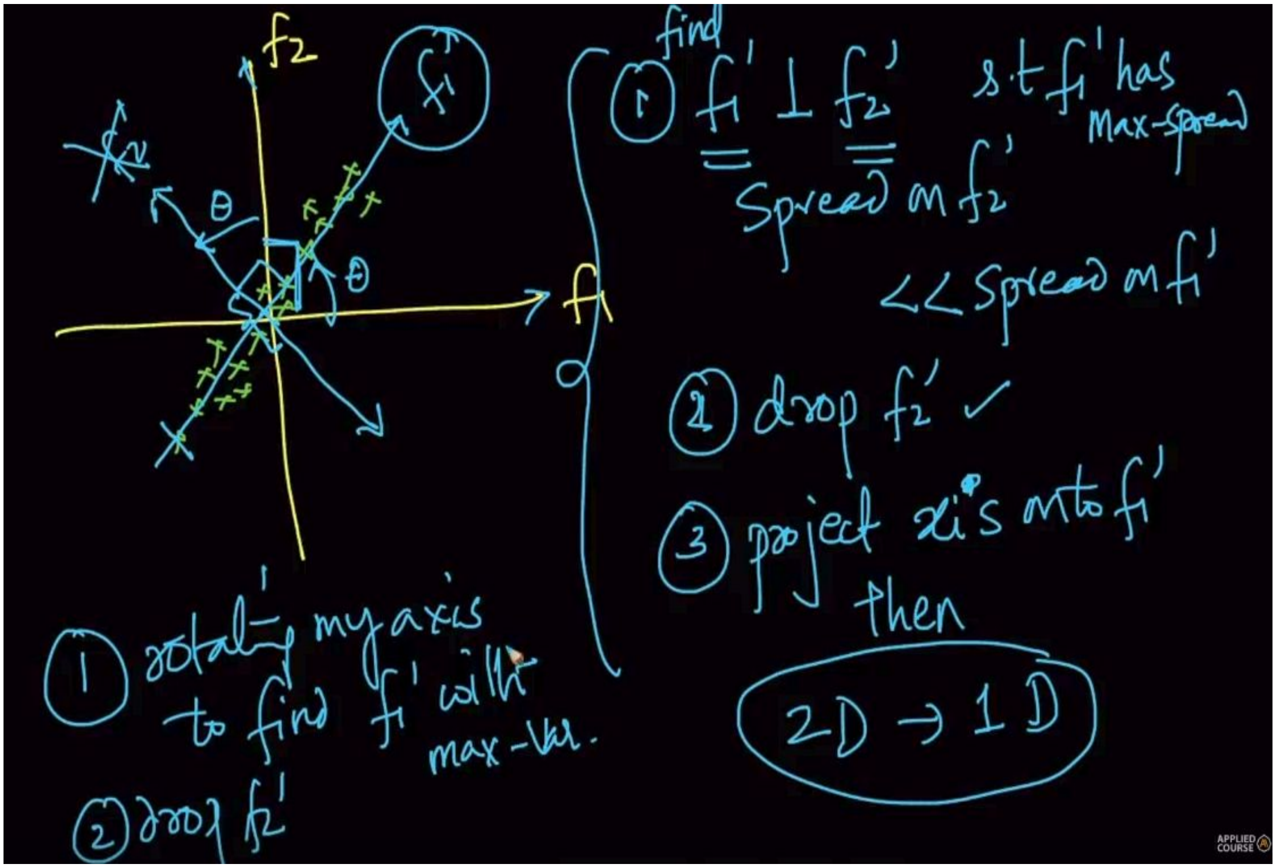


So we'll choose the feature with maximum spread because more spread is more information.

CASE 2

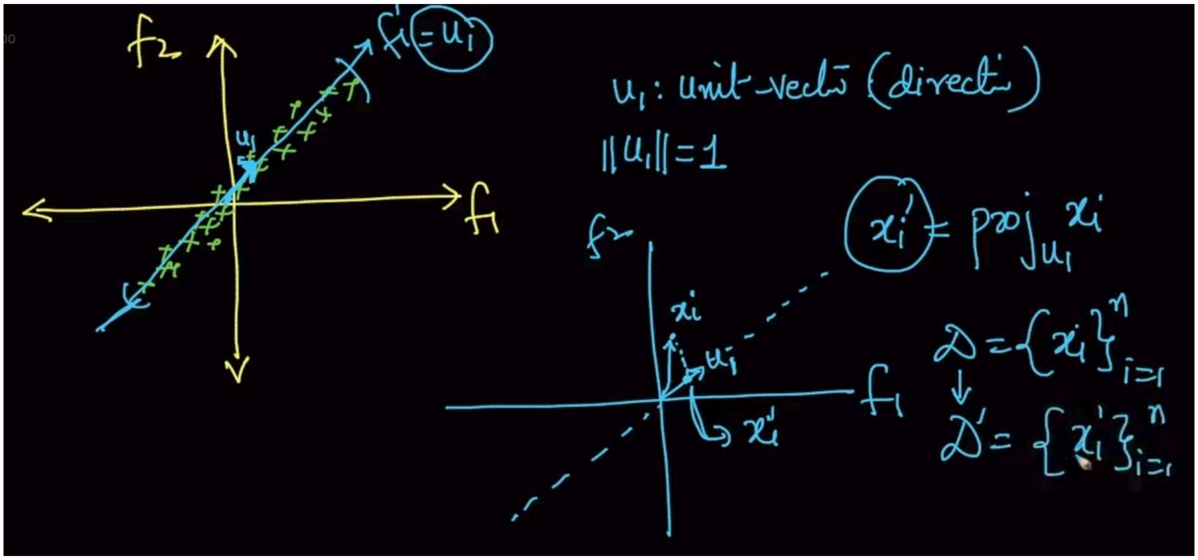


We took a 2d dataset but column standardized so the variance for both will be same (1).
So, we can't drop any feature .What will we do now?

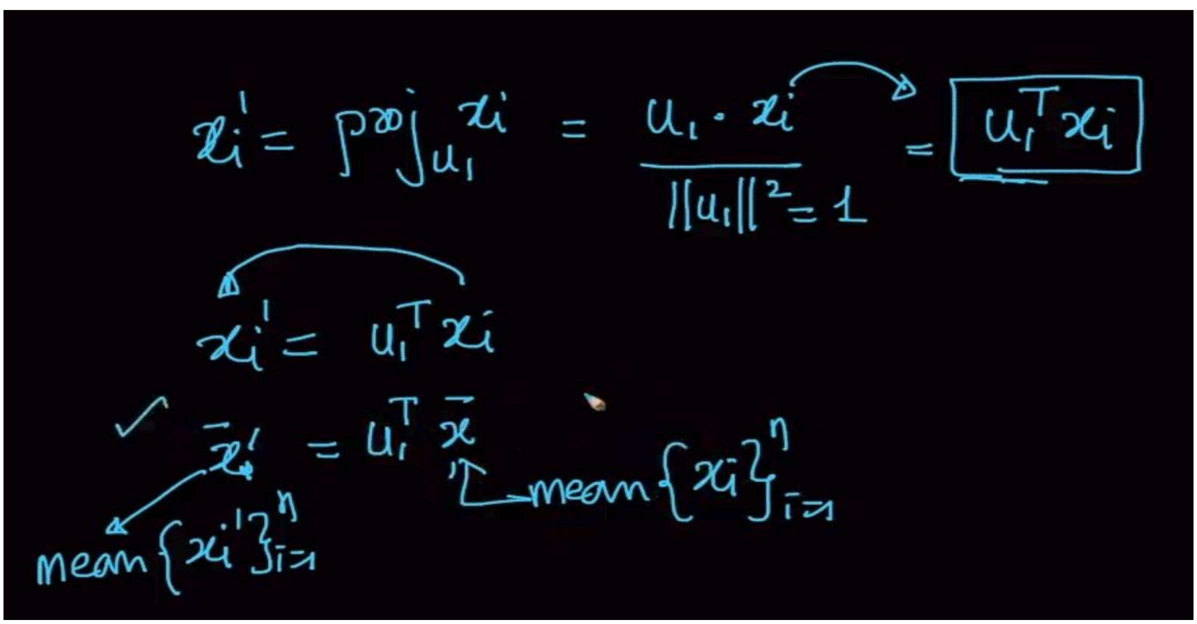


We cant drop f1 or f2 but it is visible that in f1' the spread is much greater than f2' and f1' ⊥ f2'. Since f1' has the max spread which means more information our task is to rotate our axis to find f1' with maximum - variance/spread and drop f2' to convert data from 2D - > 1D

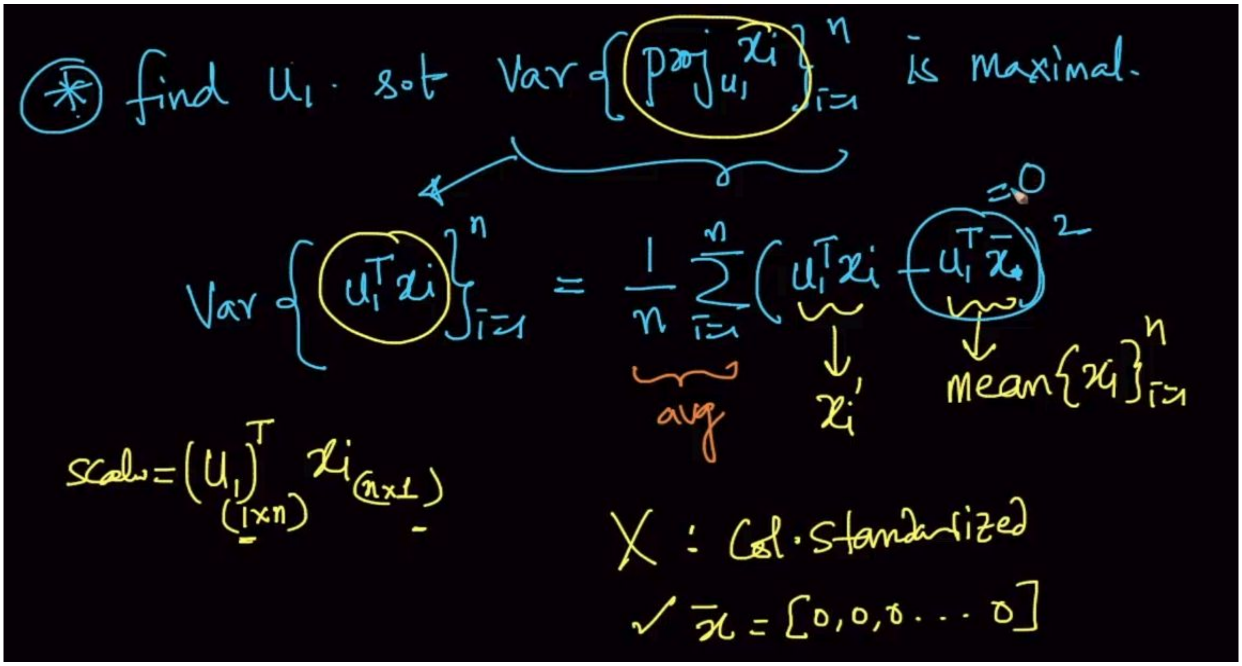
Mathematical objective function of PCA



We want to find f1' as seen above . So we need to find only the direction i.e unit vector u_1 because once we know the direction we can project ant point in that direction. So we are trying to get x'_i where $x_i' = proj_{u_1} x_i$ i.e projection of x_i on u_1

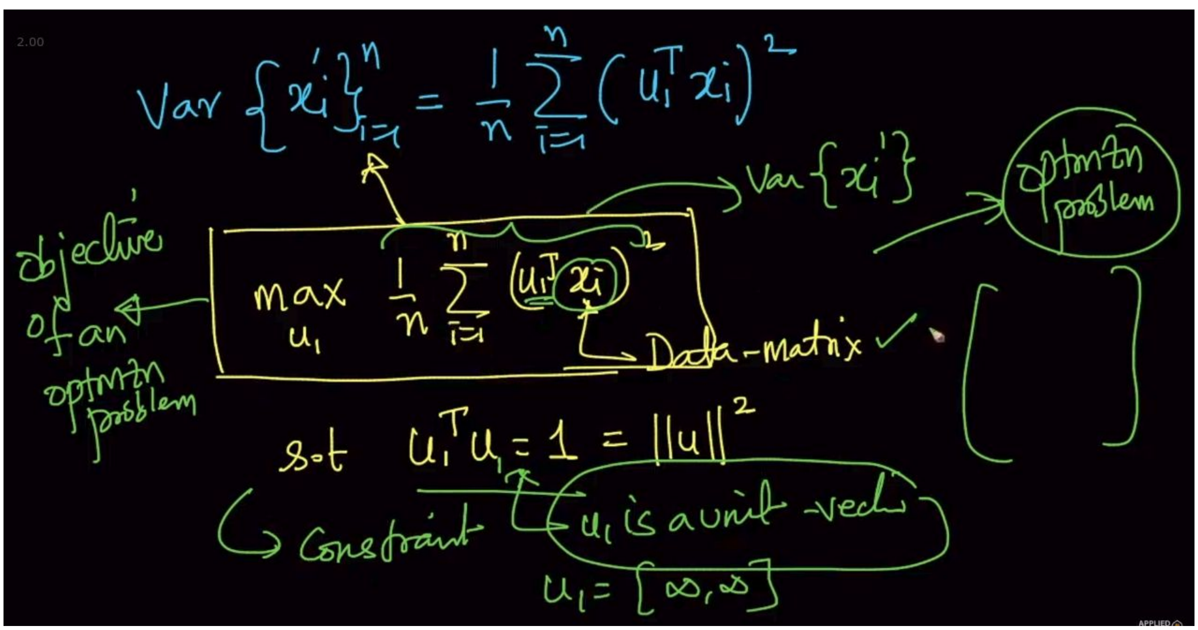


The formula has been mentioned above .Note that we are also calculating the mean of $\overline{x_i}$ and $\overline{x_i'}$
We'll know why are we doing this.



We need to find u_1 such that variance of the projection becomes maximum. The value of $u_1^T.x_i$ is a scalar.

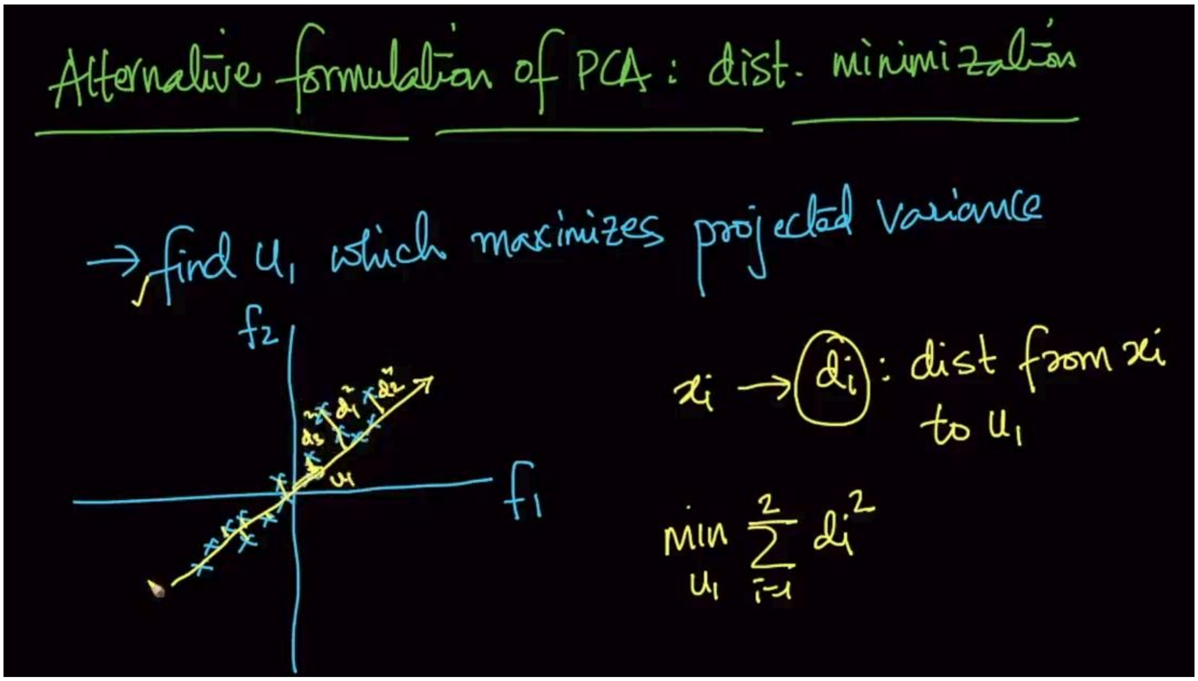
Note : If X (dataset) is column standardized then $\bar{x} = 0$, therefore, $u_1^T.\bar{x} = 0$



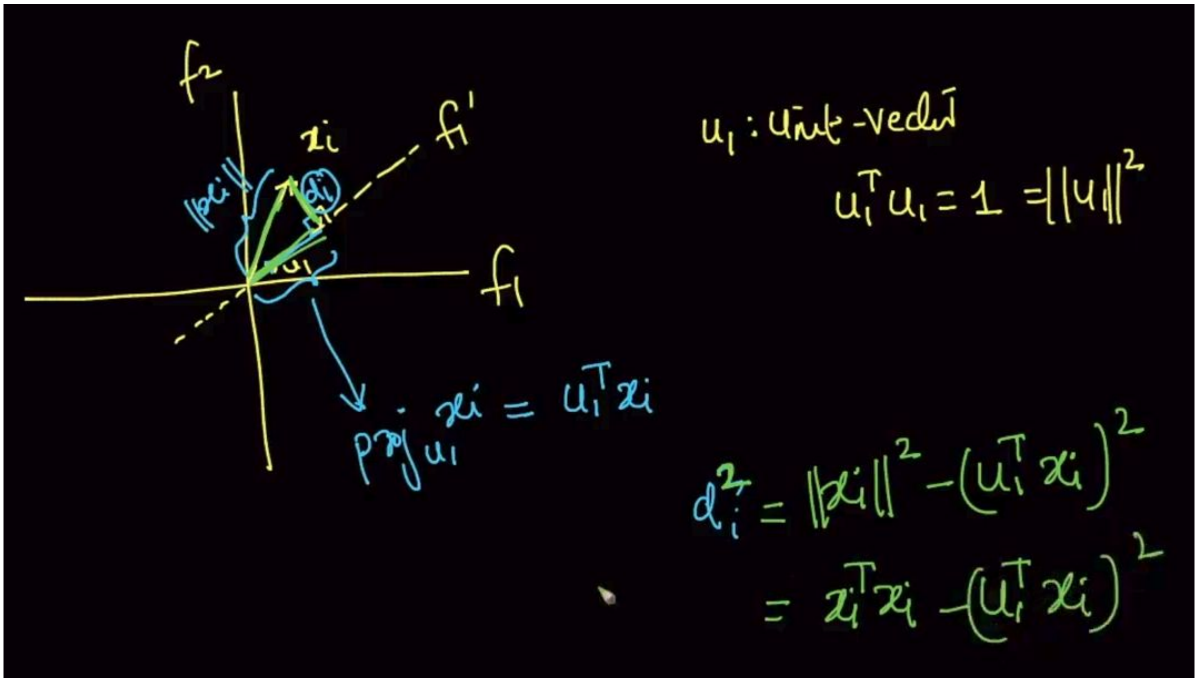
So mean ($u_1^T.\bar{x}$) will be removed to calculate variance . Therefore we want to find u_1 such that variance gets maximized as shown above.

Note : Constraint is u_1 should be a unit vector because if u_1 is infinite or very large value then variance will always be maximal so it should be a unit vector.

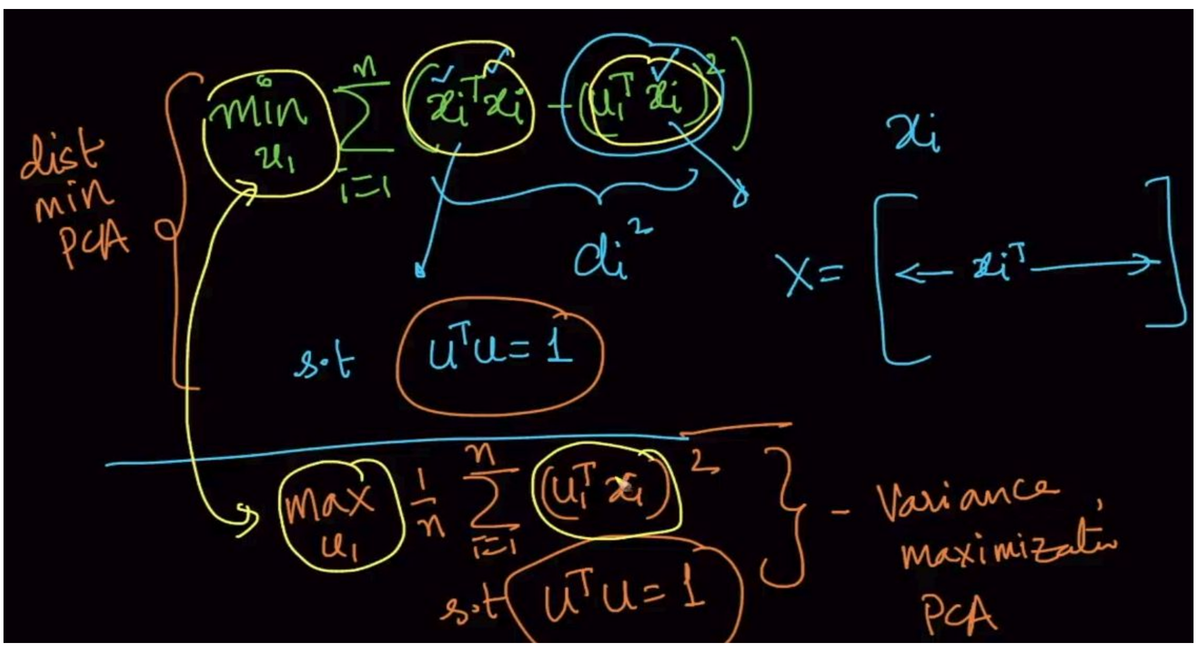
Alternative formulation of PCA: distance minimization



We've seen above that by finding u_1 such that variance gets maximum we can find our PCA.There's an alternative method to this that finding u_1 such that we find the minimum of distance d_i^2 as seen above



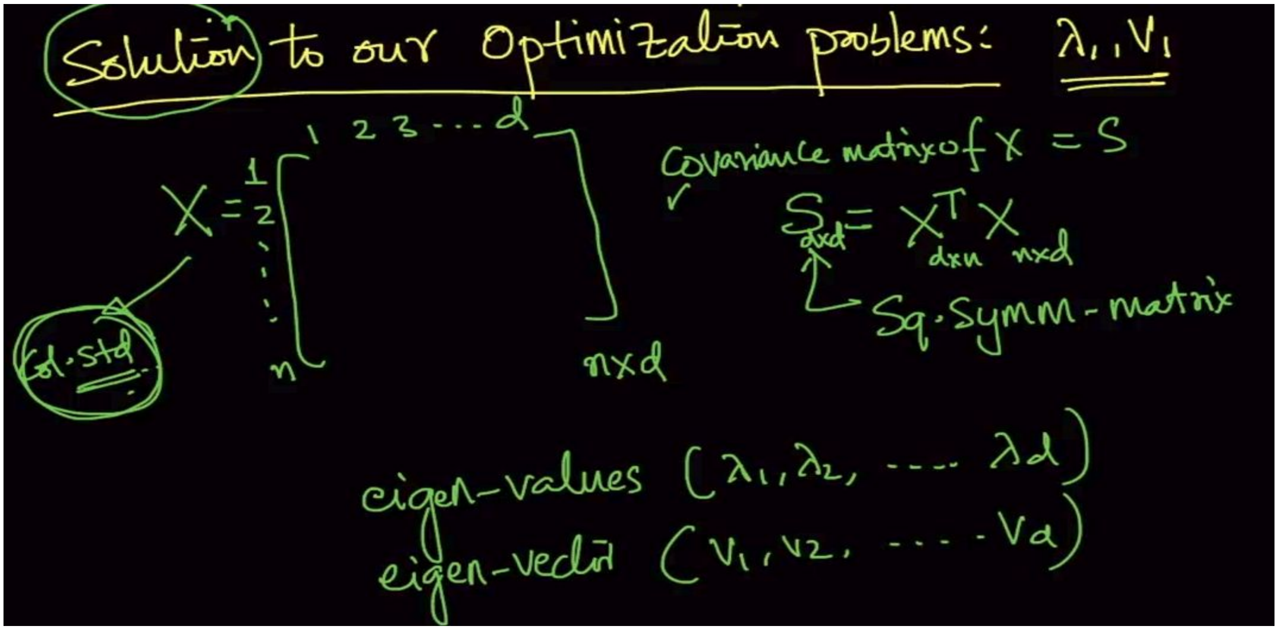
Formulation of d_i^2 where $||x||^2$ is length of x_i and $u_1^T x_i$ is projection of x_i on u_1. By trigonometry/Pythagoras we are calculating distance d_i



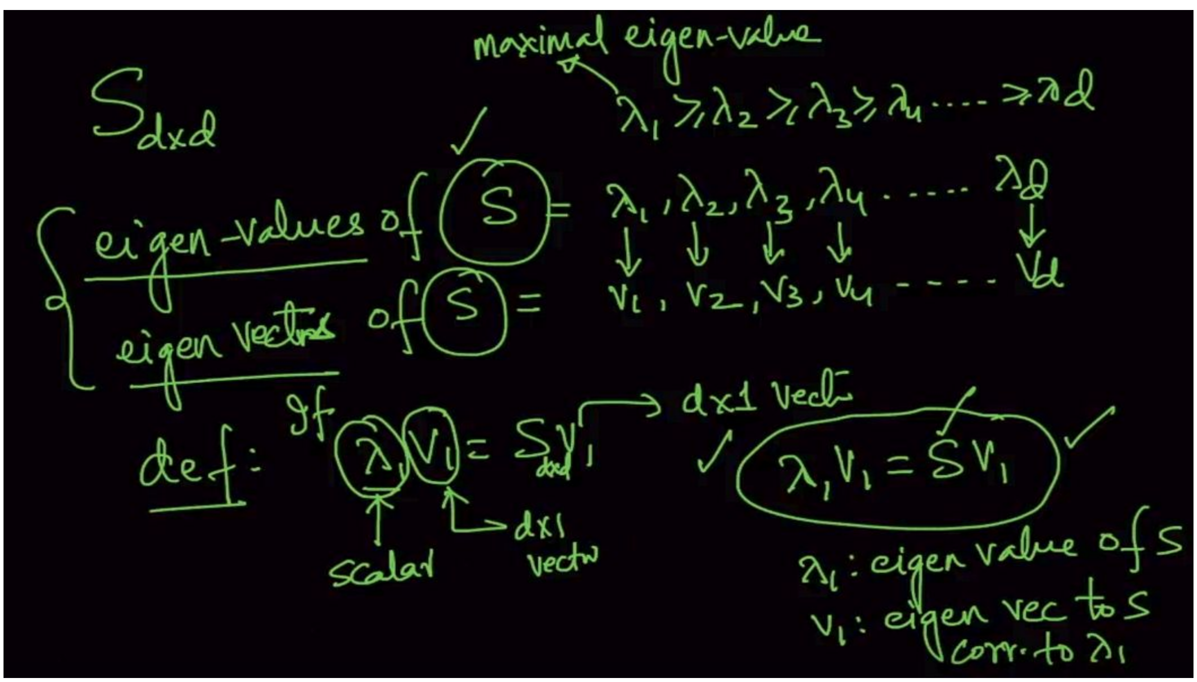
Distance minimization formula is shown . We need to find u_1 such that it gets minimal.

Note: We know variance maximization where we need to find u_1 such that var gets max. That u_1 can also be used for distance minimization with some changes

Eigen values and Eigen vectors (PCA)

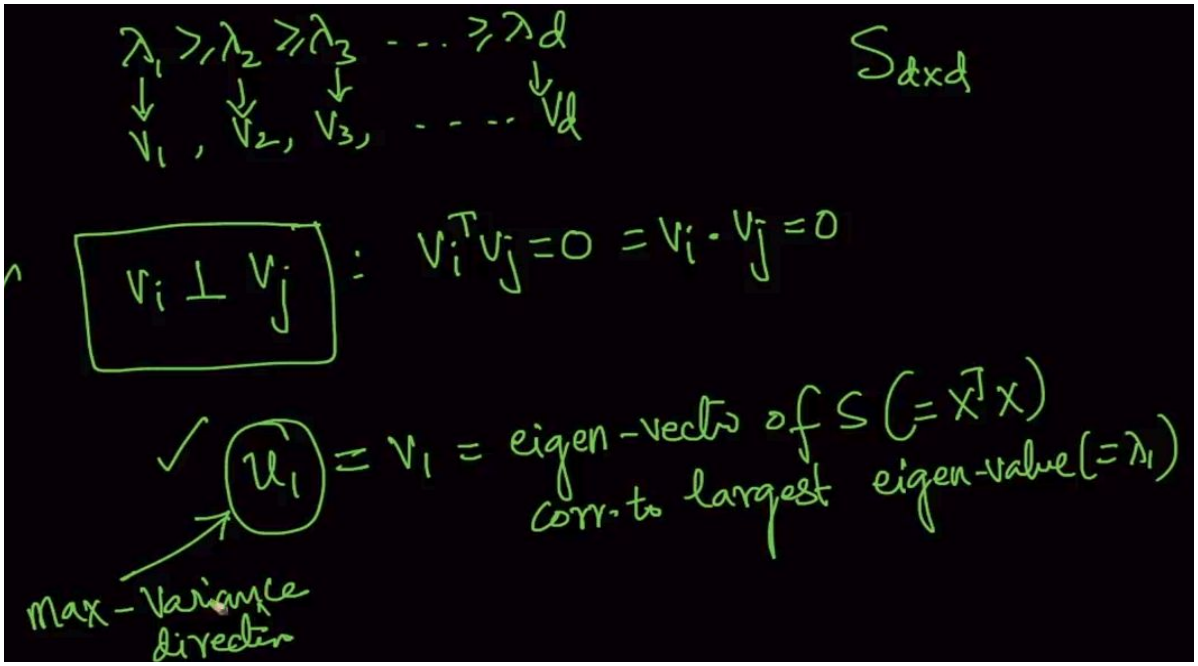


We've already performed the Column standardization and then we get the COvariance matrix of S.



We calculate Eigen values of S and get the corresponding eigen vectors by using NumPy. If the condition is satisfied then λ is the eigen value and v is eigen vector

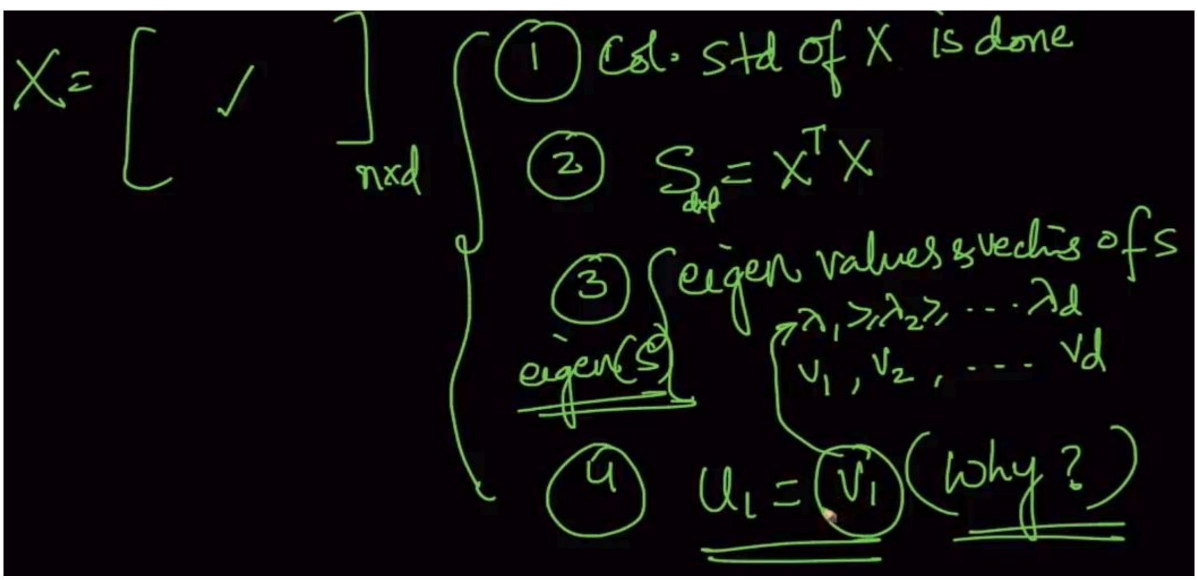
If there is a d*d matrix then it'll have d eigen values as seen above.



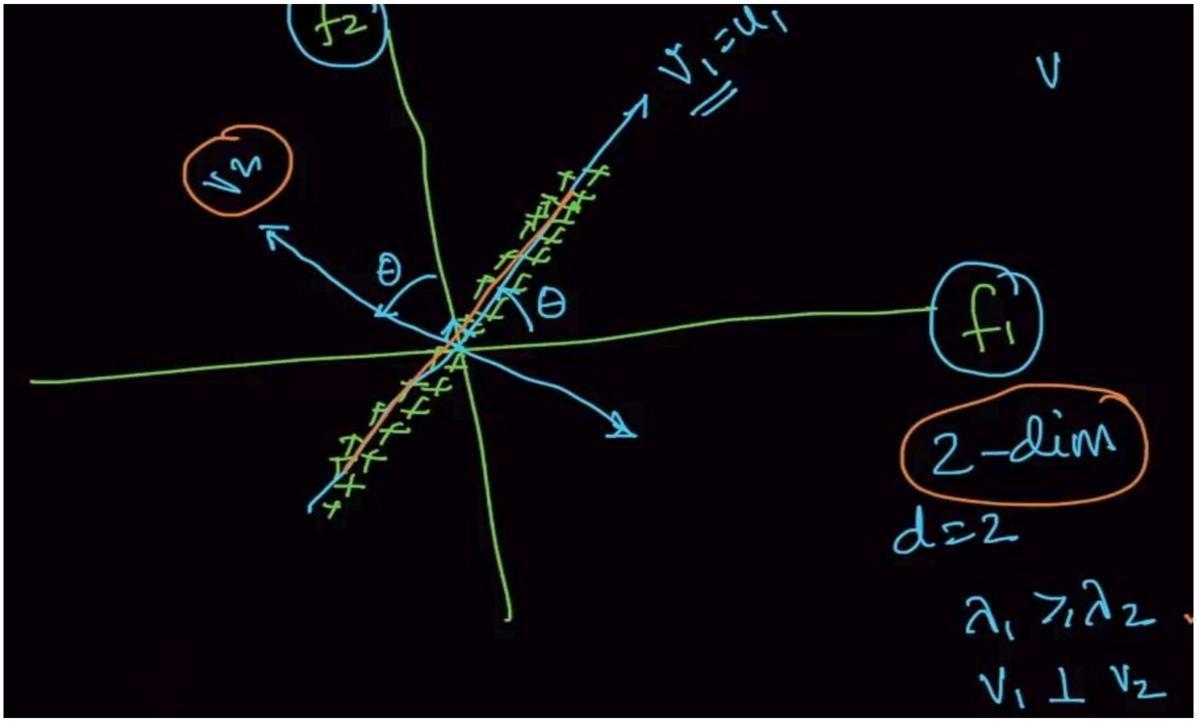
λ is calculated such that $\lambda_1 > \lambda_2 > \ldots\ldots > \lambda_d$ and if we take any two corresponding vectors then they'll always be perpendicular.

We wanted u_1 for maximum-variance/minimum-distance and u_1 = v_1 (where v_1 is the vector corresponding to largest eigen value λ_1)

STEPS FOR PCA

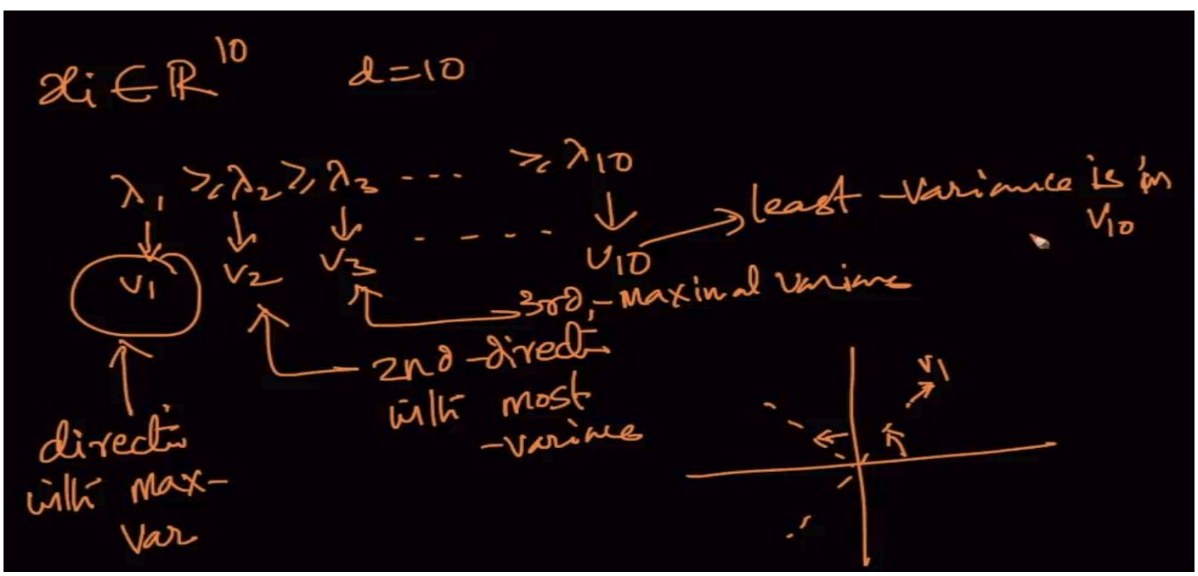


GEOMETRIC INTERPRETATION OF EIGENVECTORS



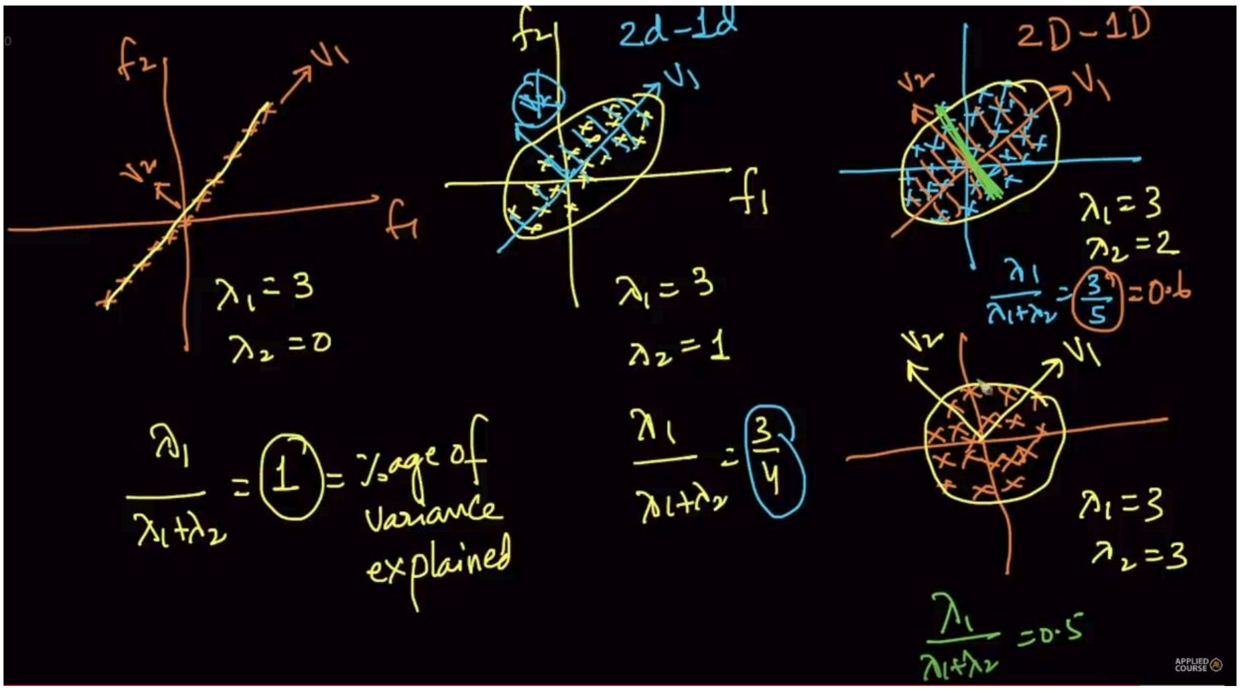
Suppose we've two dimensions f_1 and f_2. Then we get the eigen values λ_1, λ_2.

What we are doing here is we are rotating the axes such that our top eigen vector v_1 of covariance matrix of X corresponds with the direction where spread is maximum



If we've a 10 dimensional data then 10 eigenvalues and the largest λ will correspond to direction with max var. Second largest λ will correspond to second direction with most variance and so on

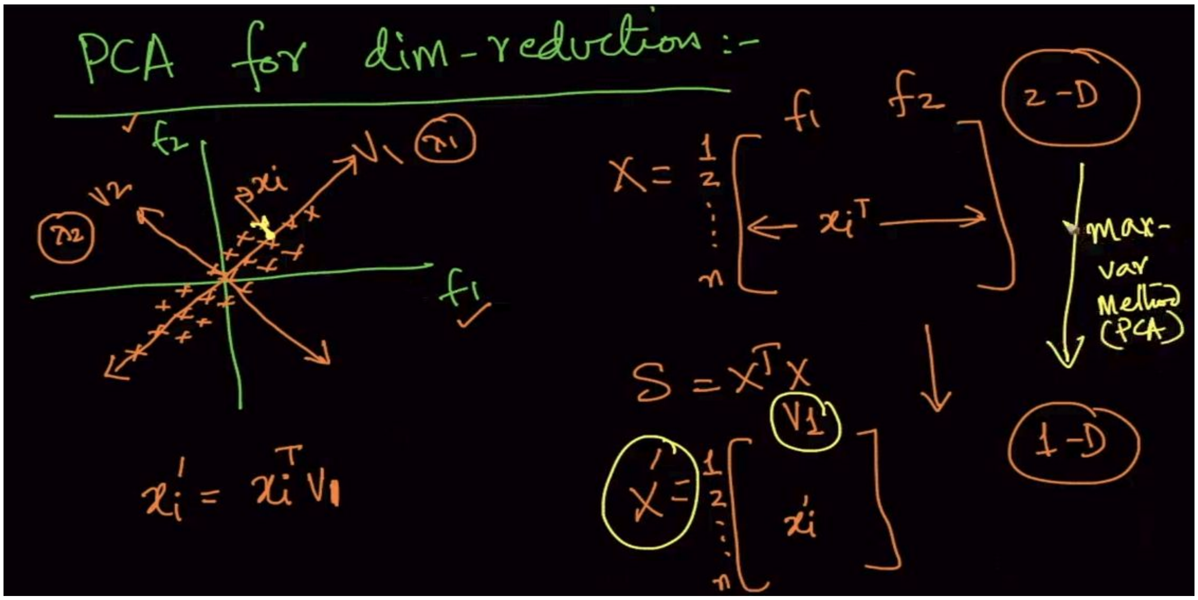
GEOMETRIC INTERPRETATION OF EIGENVALUES



Here different datasets are plotted with different spreads and variances. As mentioned $\frac{\lambda_1}{\lambda_1+\lambda_2}$ = 1 = percentage of variance explained. So, in the 1st fig it's 1 i.e it says that if we project all our data into v1 we won't lose any data since there's no spread in v2 .

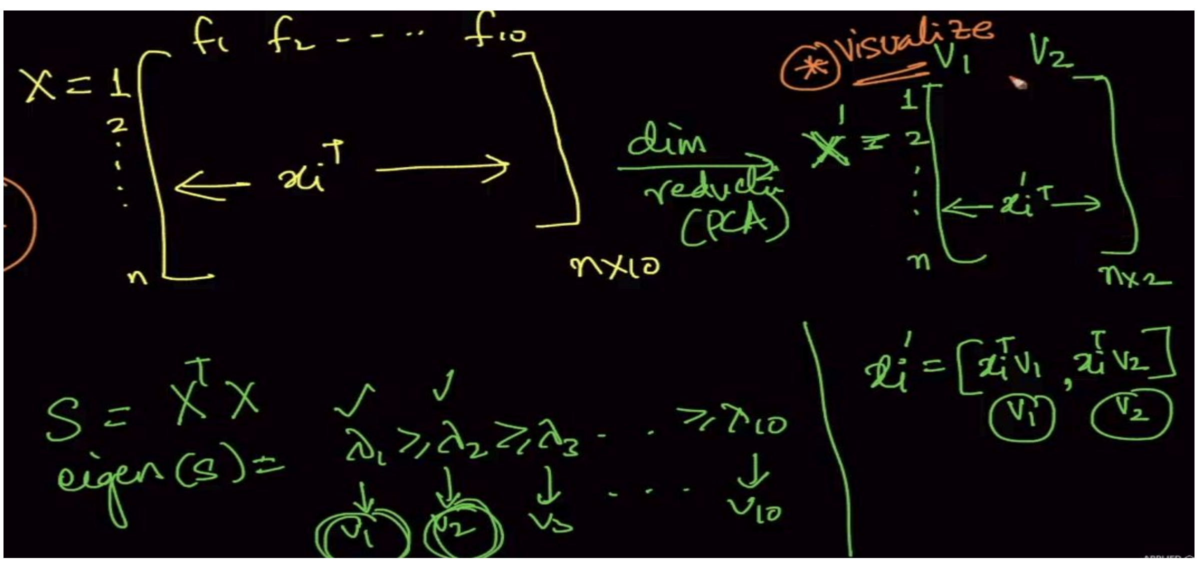
In the 2nd fig if it is 75% that means if we project all our data into v1 or convert 2d-1d then we conserve 75% of the data since there's some spread in v2 as well. Same for the rest of the figures. So, λ gives a relative idea or scale of the variance in our data i.e $\lambda's$ tells us if there's spread in one axis (1st fig) or there are spread in the other axes as well (fig2 or 3) .

PCA for Dimensionality Reduction and Normalization



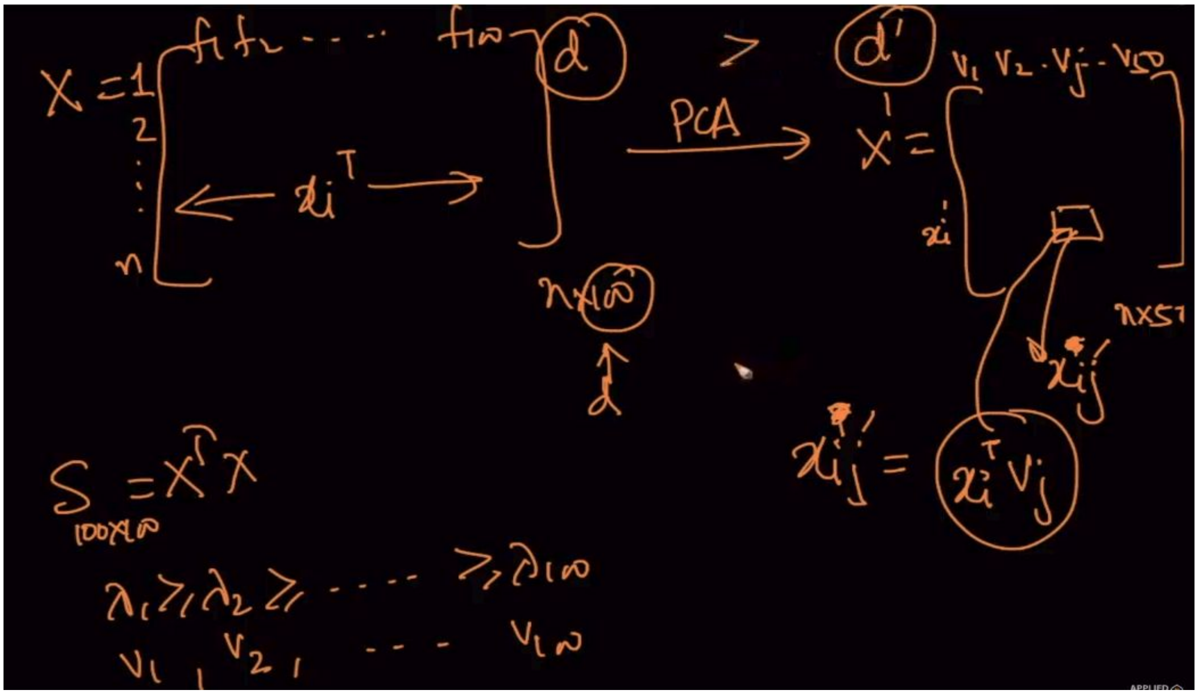
Suppose we want to convert our data from 2-D from 1-D . Then firstly you'll try to project data into v1 you got from $\lambda 1$ where you'll get the maximum variance. The obtained data is nothing but $x_i' = x_i^T v1$ which is projection

What if we want it for 10-D?

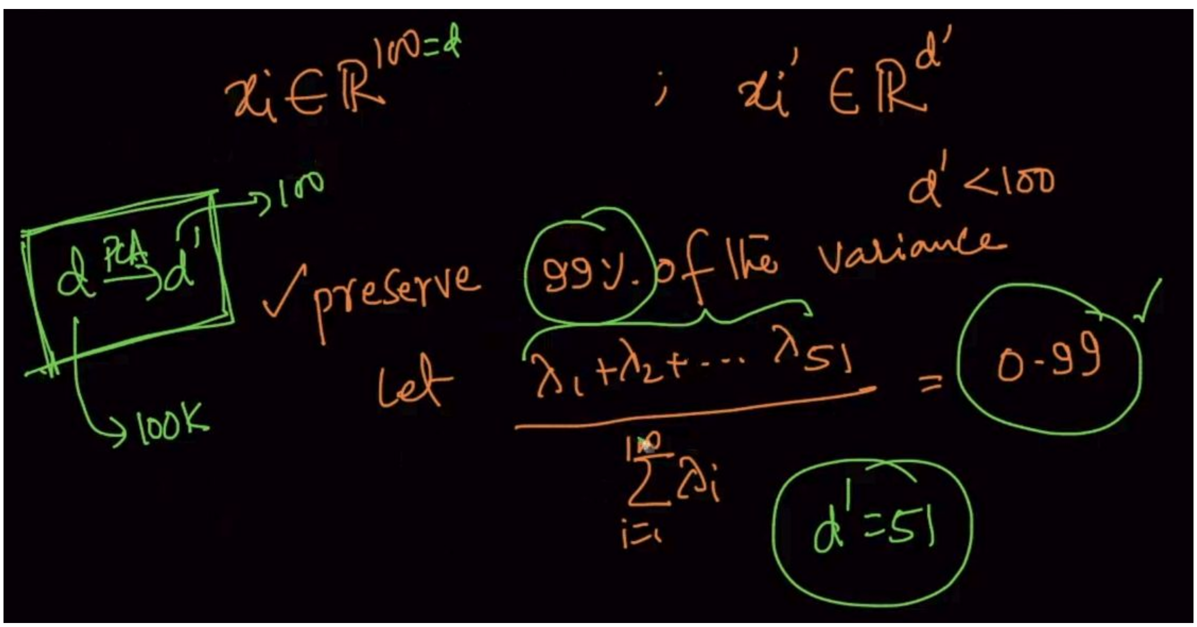


Firstly calculate S (covariance matrix) then get it's λ.Select the top two λ and then multiply it with $x_i(datapoint)$ and now you've reduced it to 2-D

What if we've 100 dimensional data?

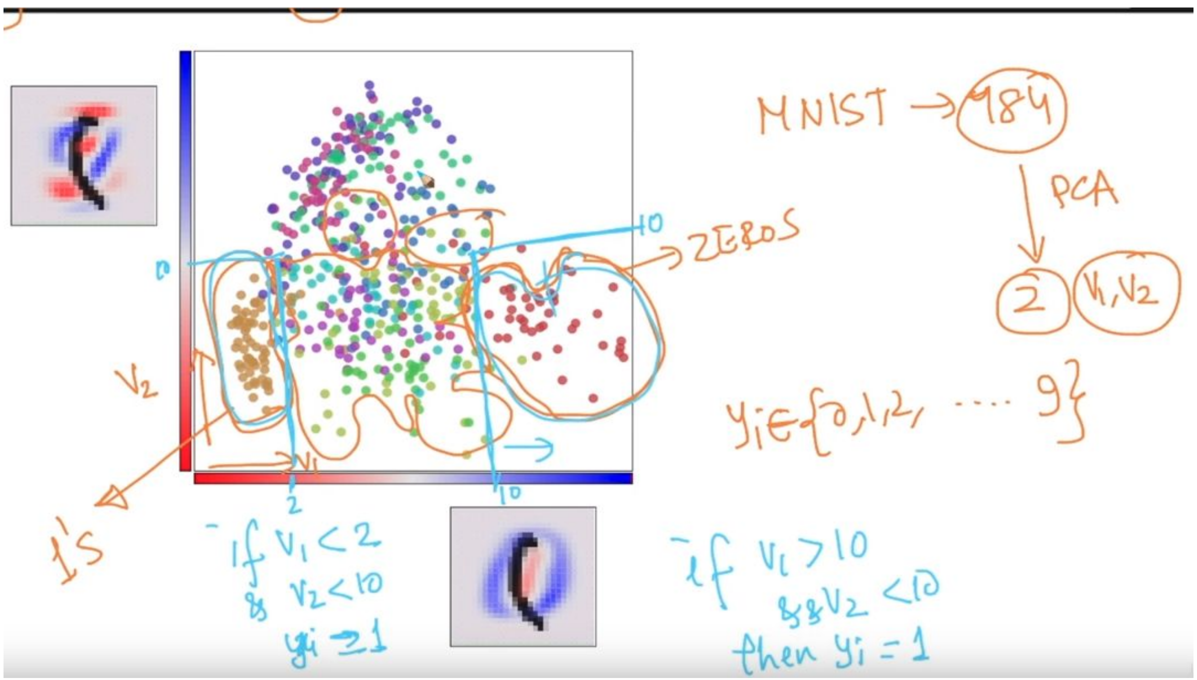


Do the same steps and get the top 50 eigenvalues (λ) and then multiply eigenvectors(v) with datapoints.



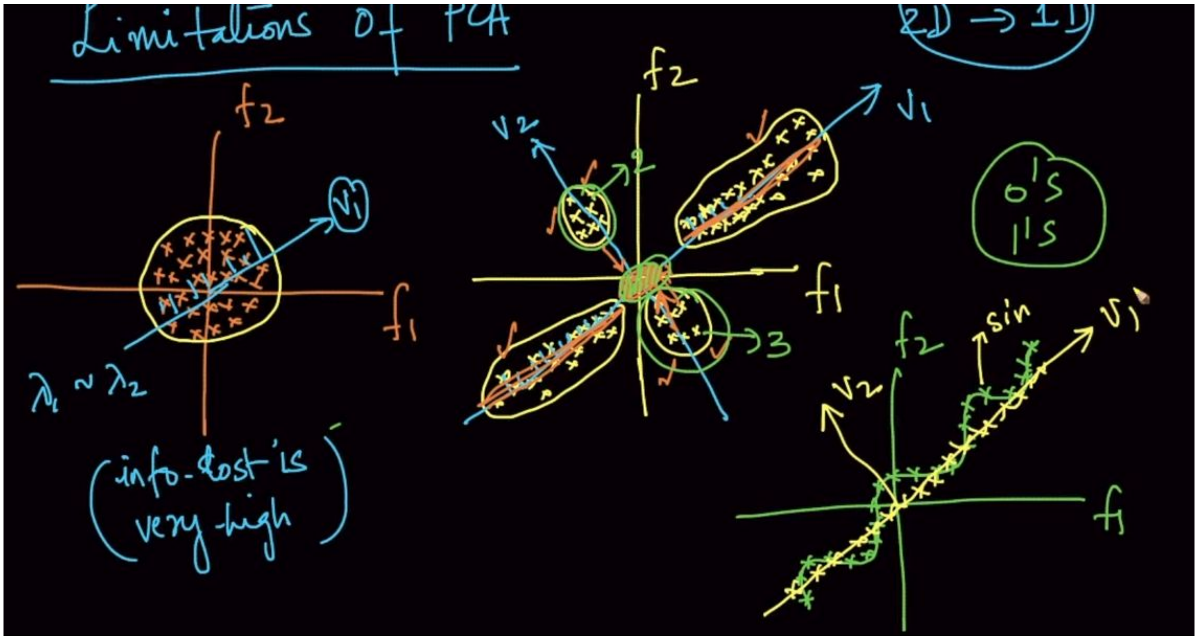
If we've to preserve 99% of the data from a 100d data and top 51 eigenvalues (above) is preserving 99% of the data. We select them and reduce our data to 51-D

Visualize MNIST data set :Dimensionality reduction and visualization



Here our 1 and 0 are seperated well so we can apply if conditions like above to predict them on our PCA performed dataset plotted on top 2 eigenvectors (v1,v2) but it didnt seperated the other numbers well like t-SNE .

Limitations of PCA



In the first figure, $\lambda_1 \sim \lambda_2$ so data loss will be very high after converting it to 1-D

In the second figure, the clusters in the middle will get projected in the same line so it will confuse us from which cluster it came from

In the third figure , we are losing the sinusoidal wave when projecting it into a single line