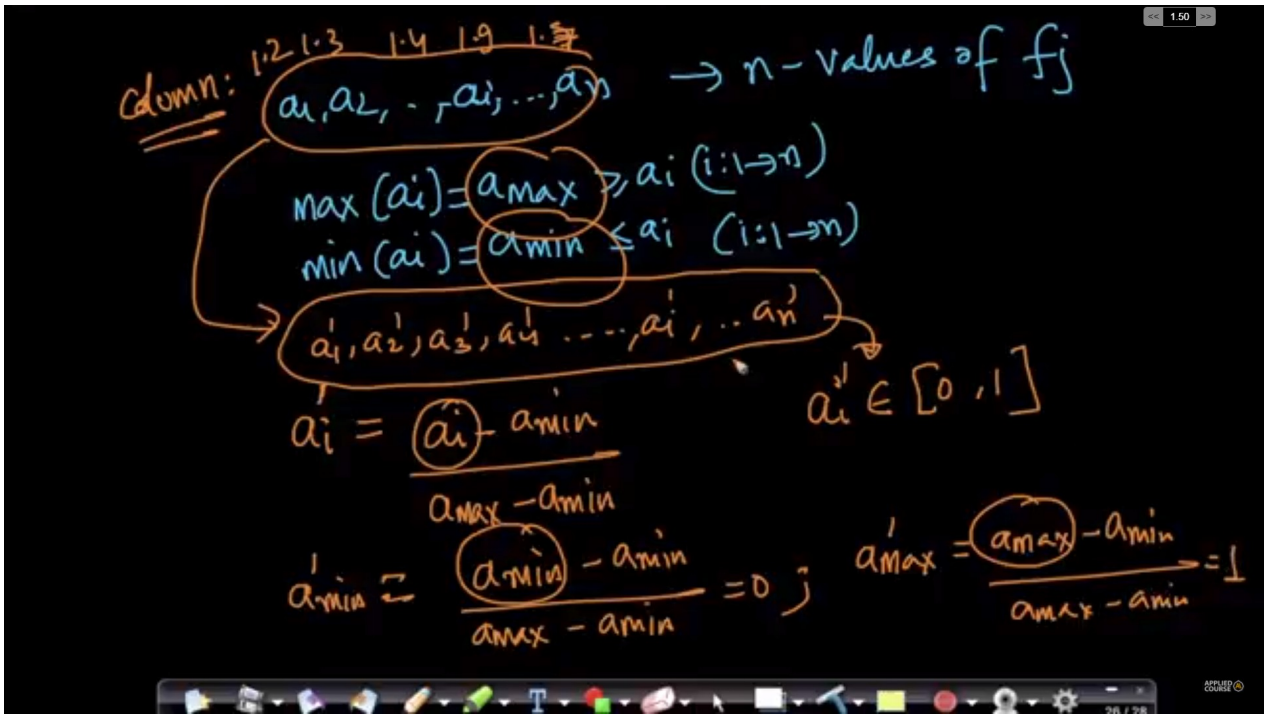
NOTES

Data prepossessing:

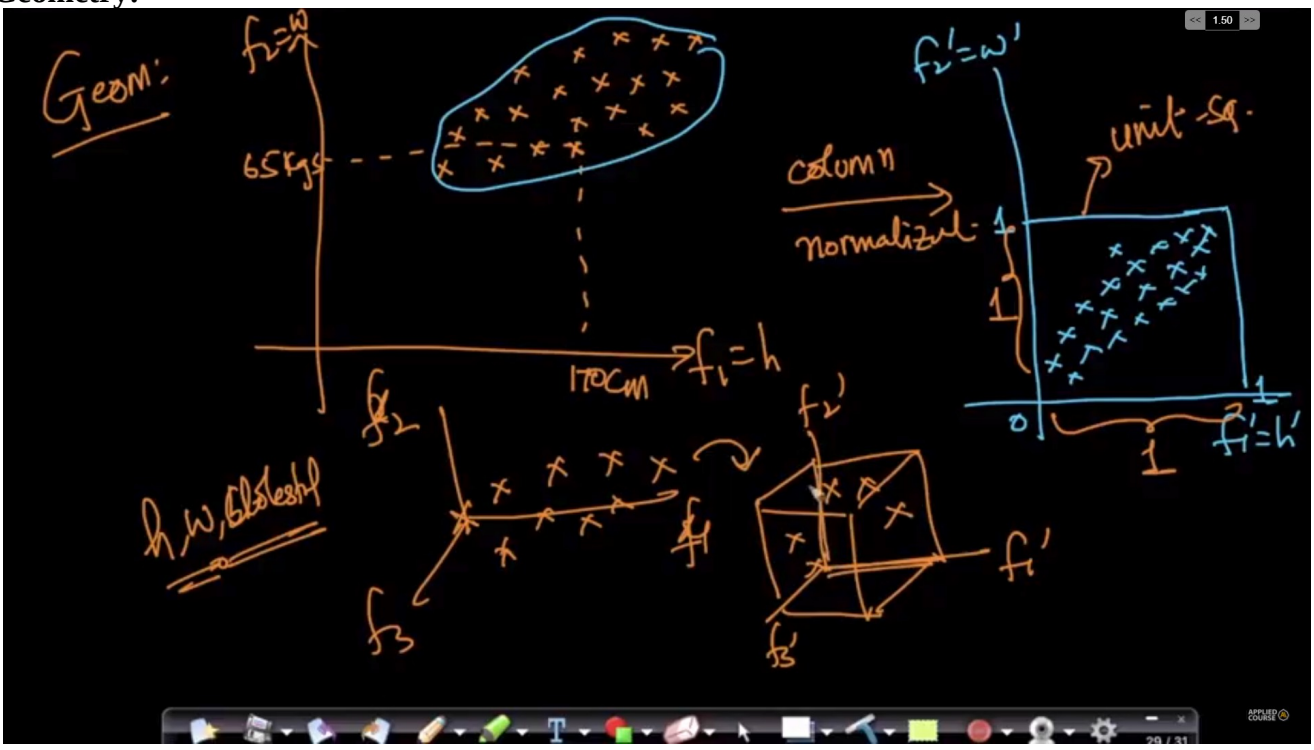
Column Normalization:

getting values between 0 and 1, to get rid of scale of attributes.

Notation:



Geometry:

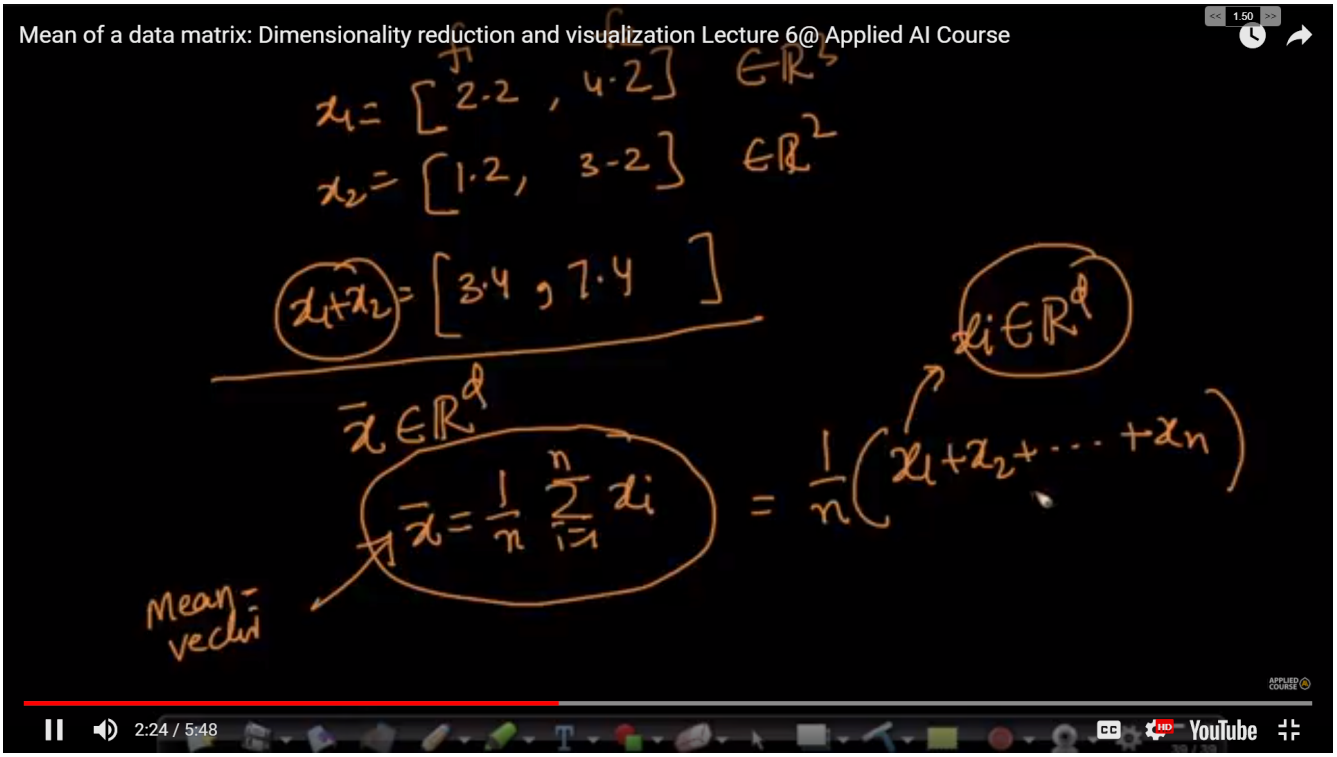


Column Standardization:

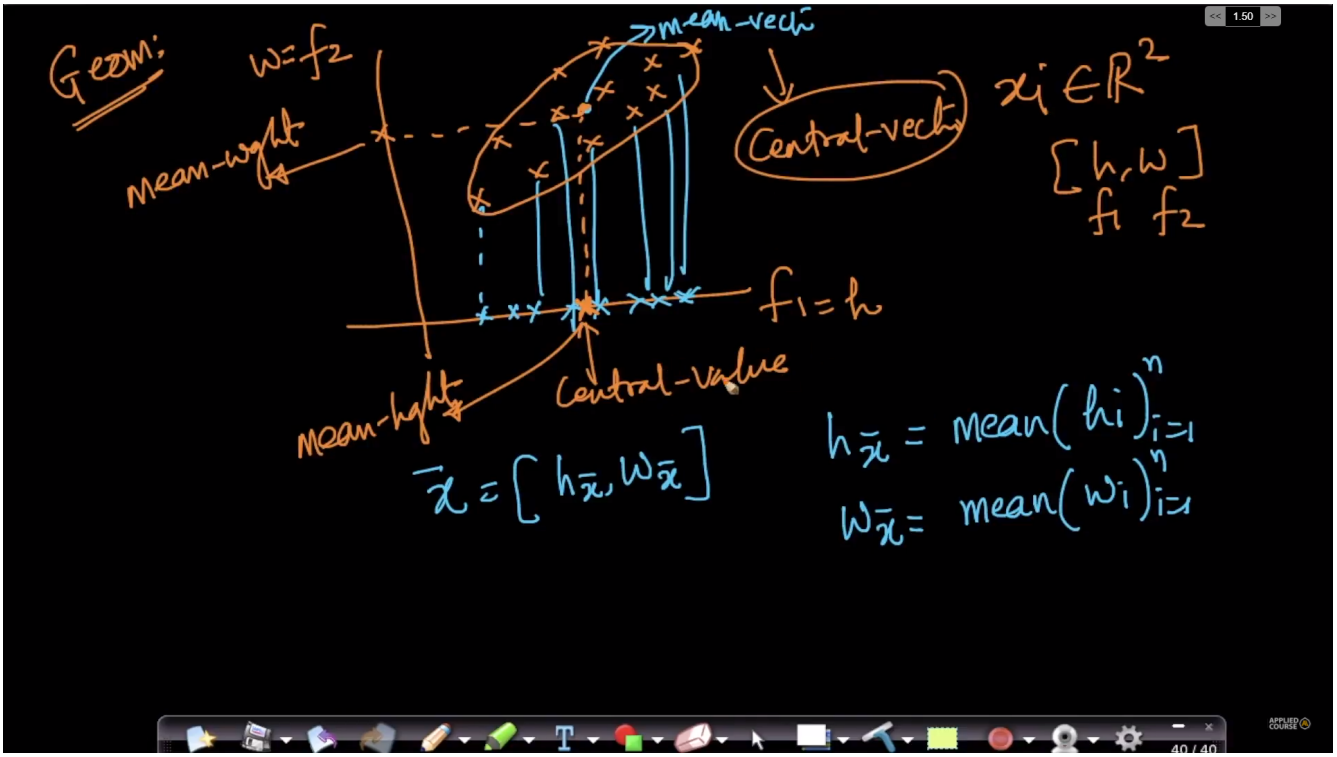
Converts the data into mean=0 and standard deviation = 1.

Mean vector:

Notation:



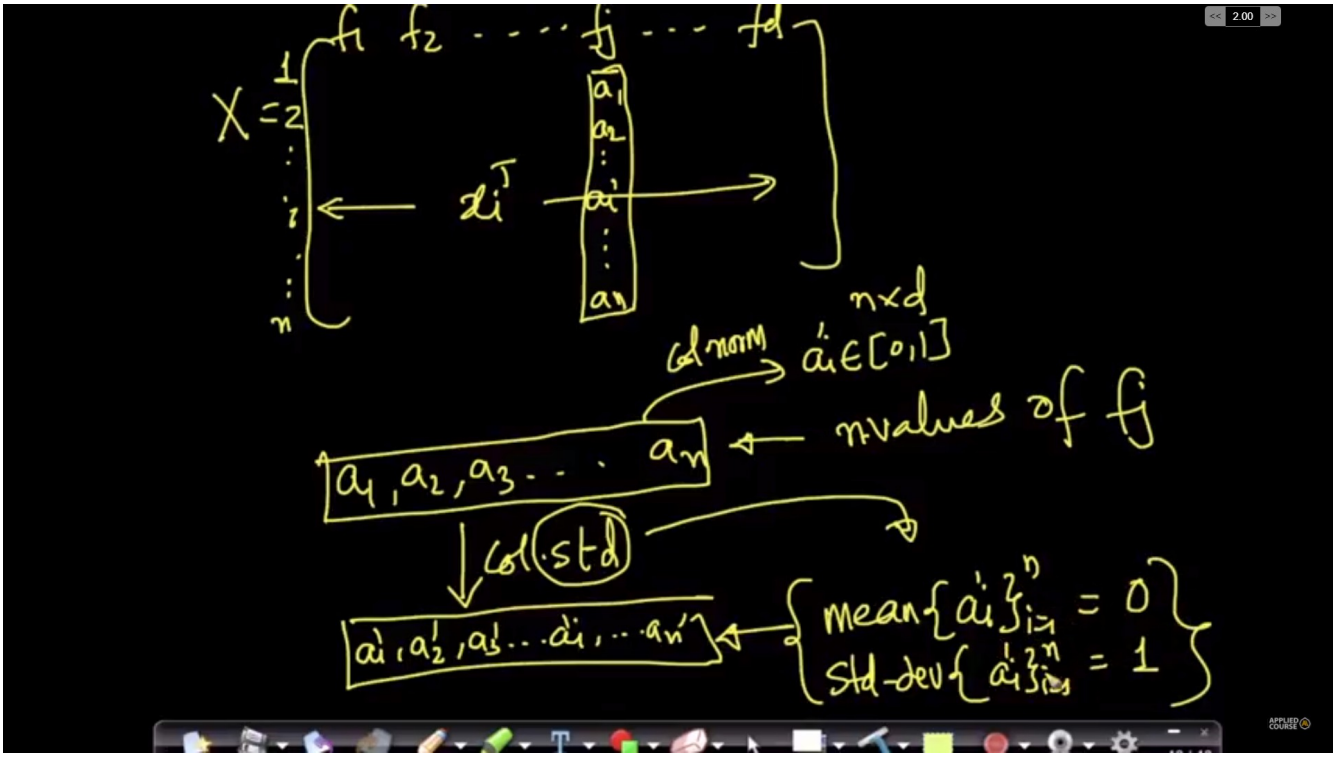
Geometry:

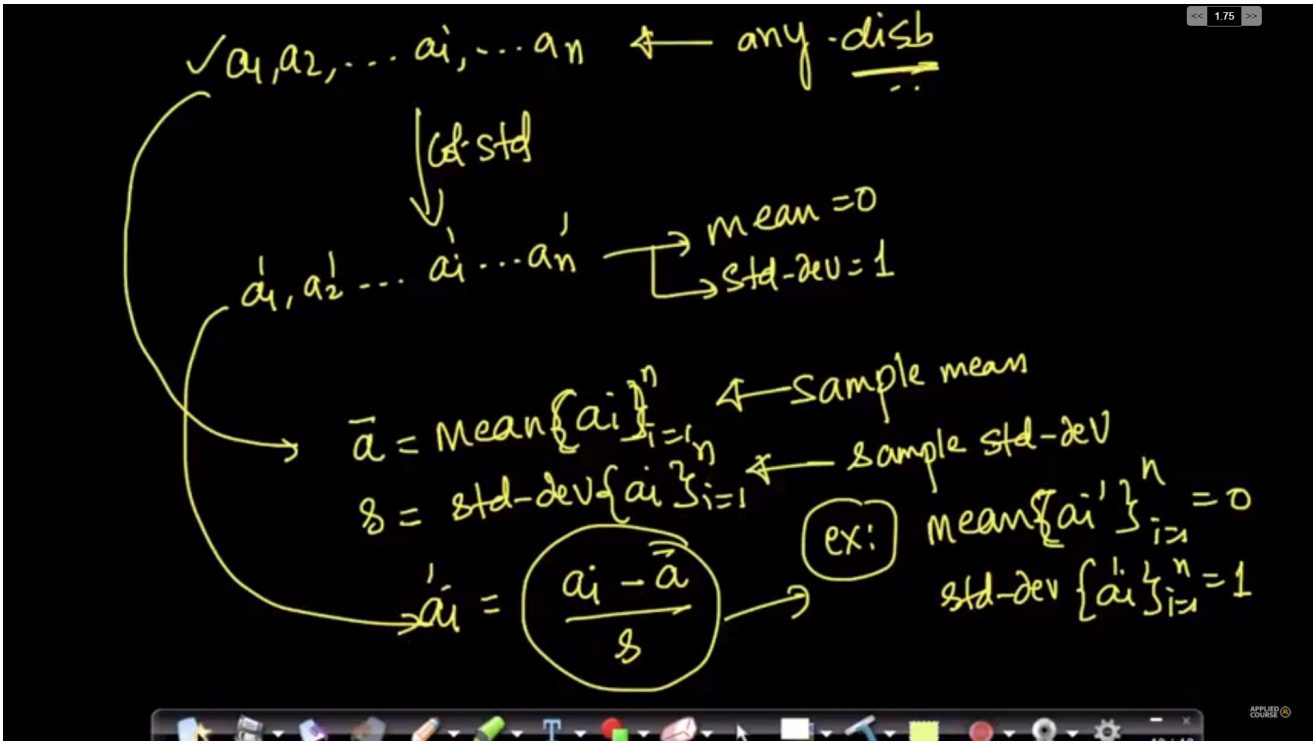


Here mean vector is the central vector of all the data points.

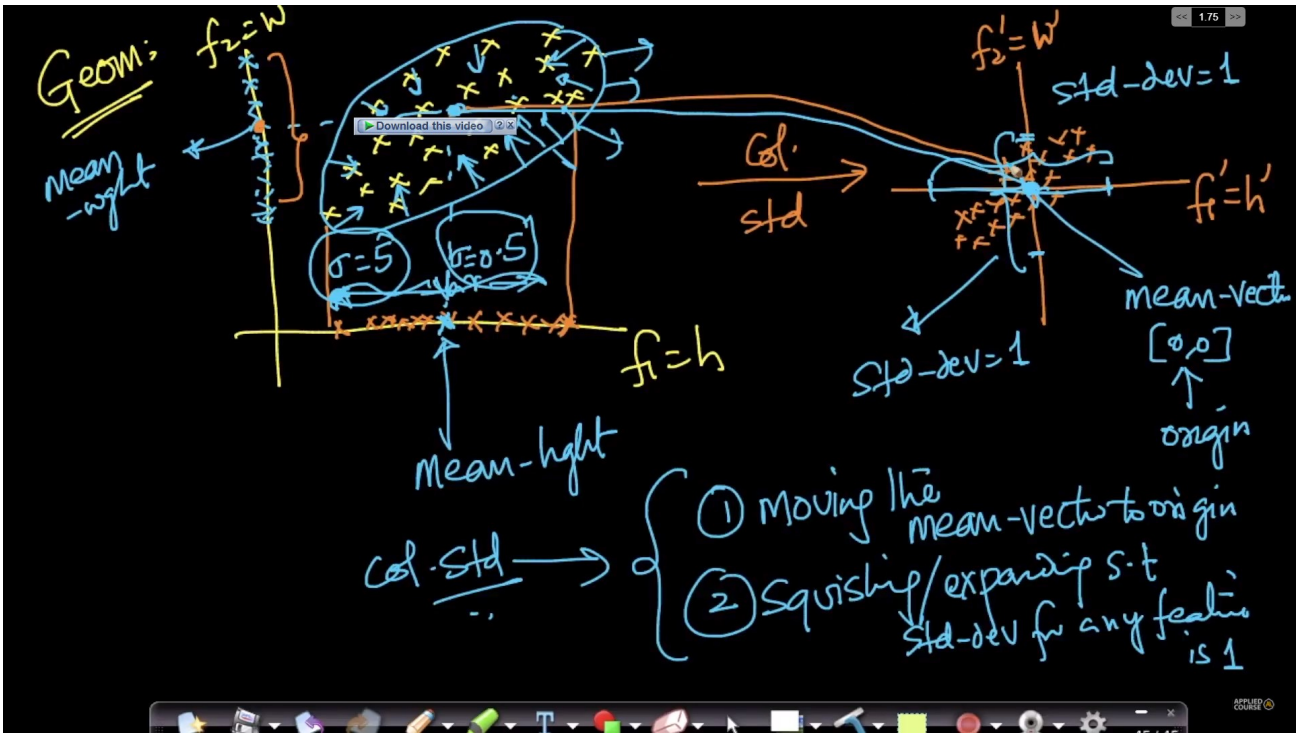
Standardization:

Notation:



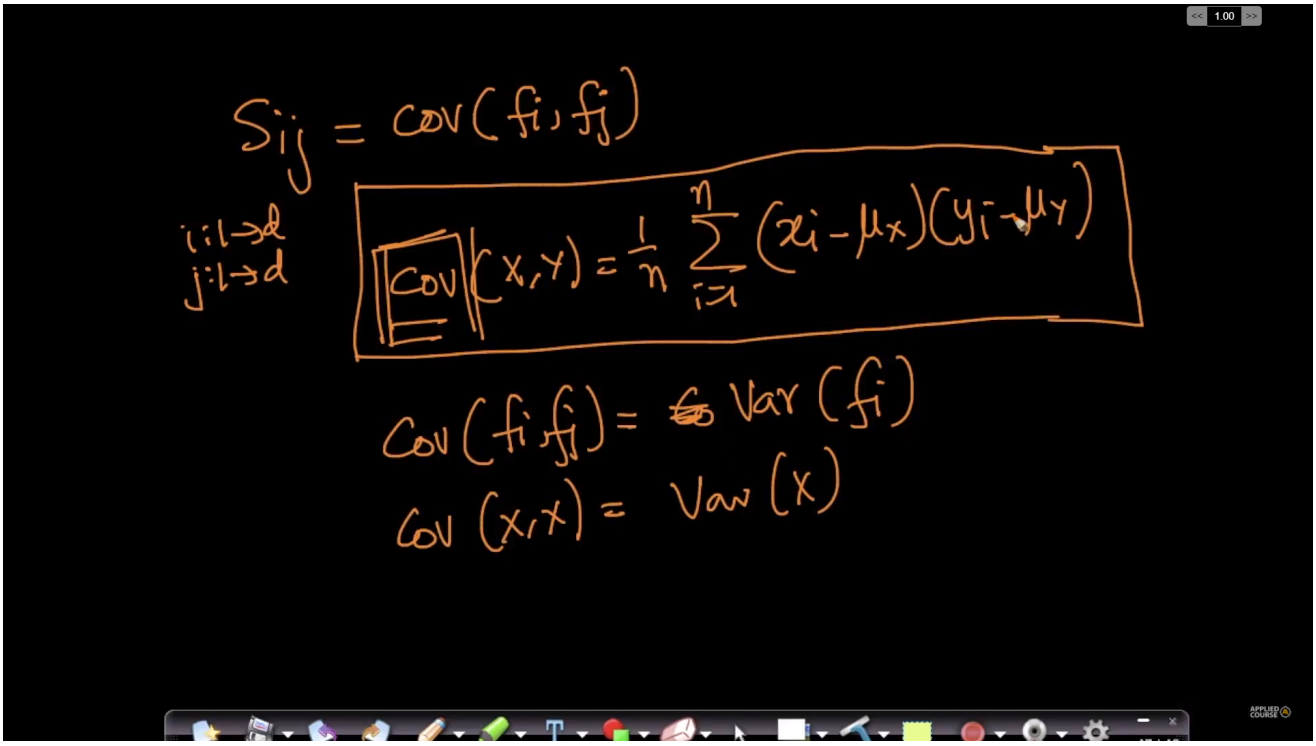


Geometry:



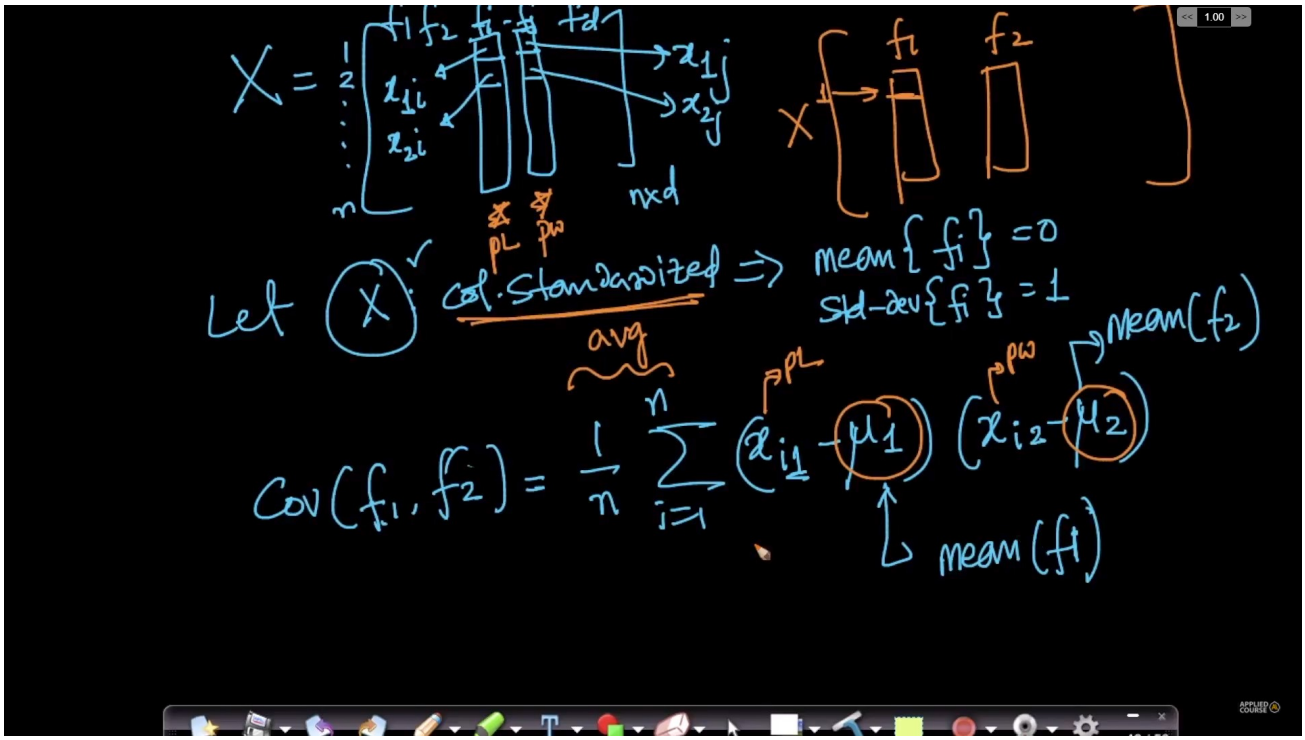
Co – Variance of the Data Matrix.

Notation:

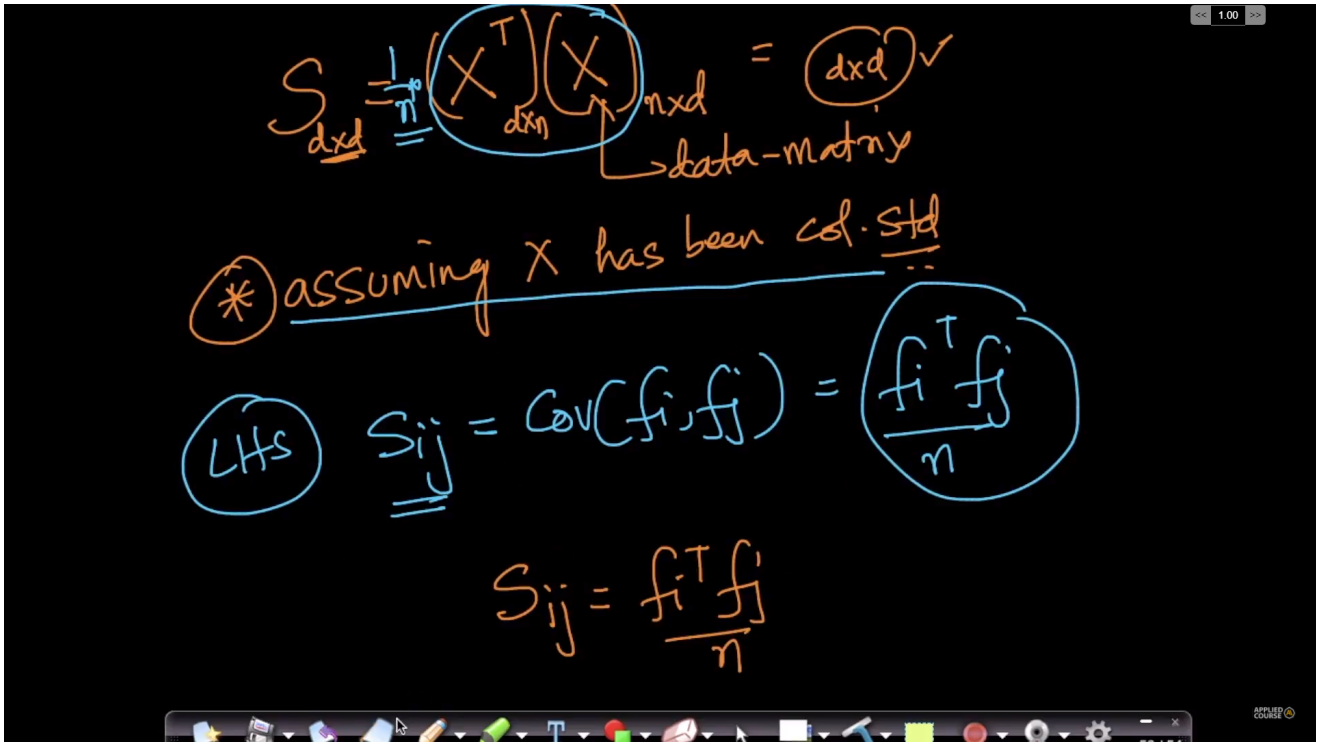


The variance is also called symmetric matrix, along the diagnol we will have variance because c

:



it's mu(greek alphabet), mu_1, mu_2 are 0 ie the means are 0 since data has been column standardised



The formula (X-transpose) * (X) holds because the data is column standardized.

PCA:

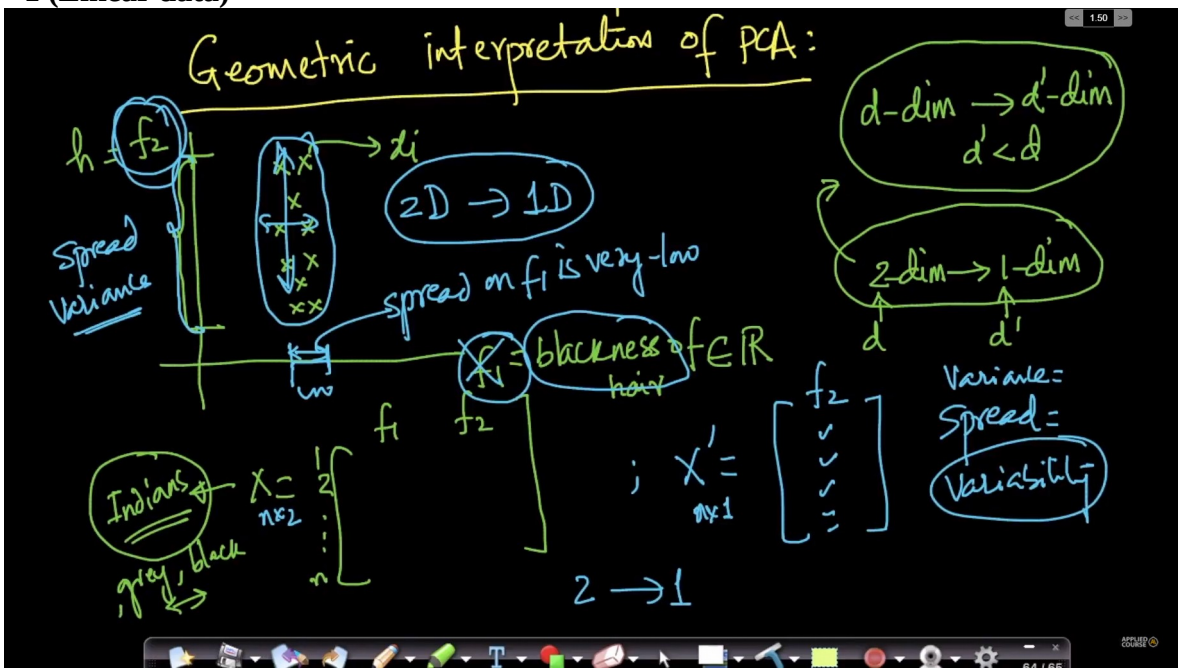
There are two kinds of defining the PCA

1. **Variance maximization**
2. **Distance minimization**

Variance Maximization

Geometry:

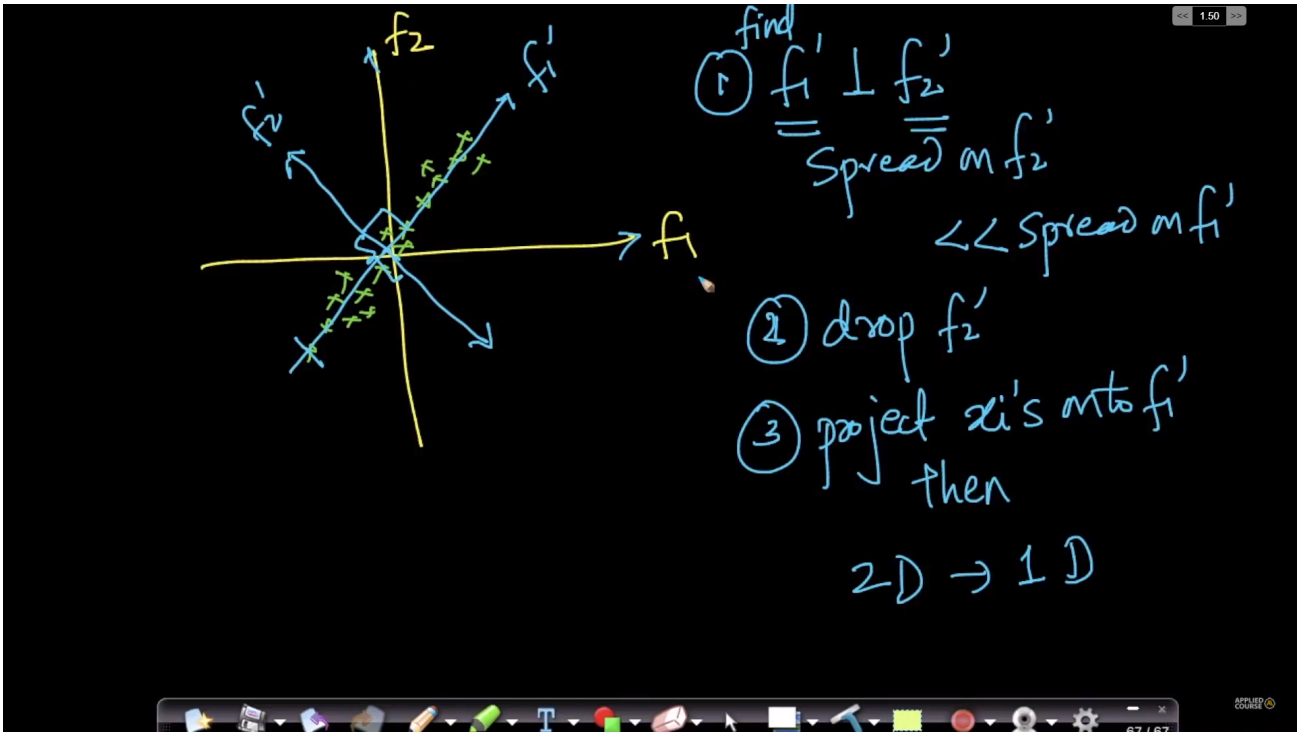
Case – 1 (Linear data)

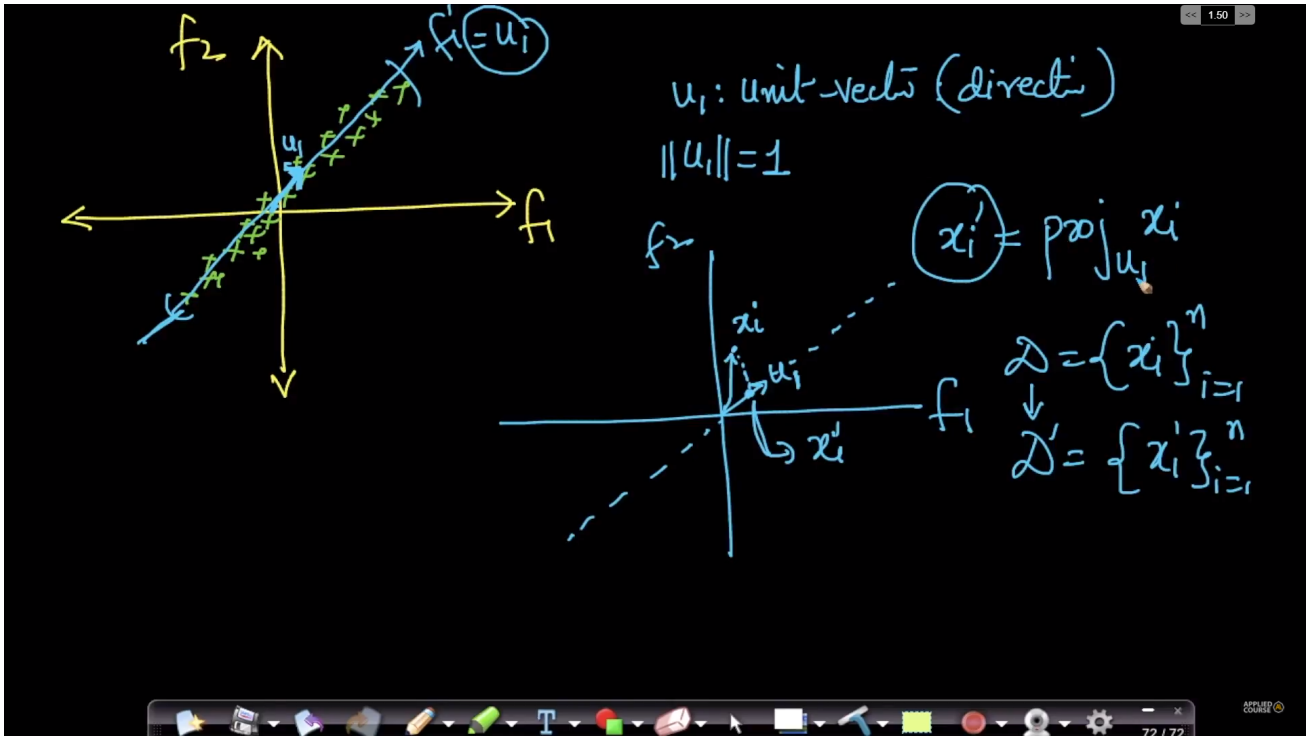


Here, Skipping the Feature 1 is the good choice as the maximum variance can be captured by feature 2.

Case – 2(inclined data)

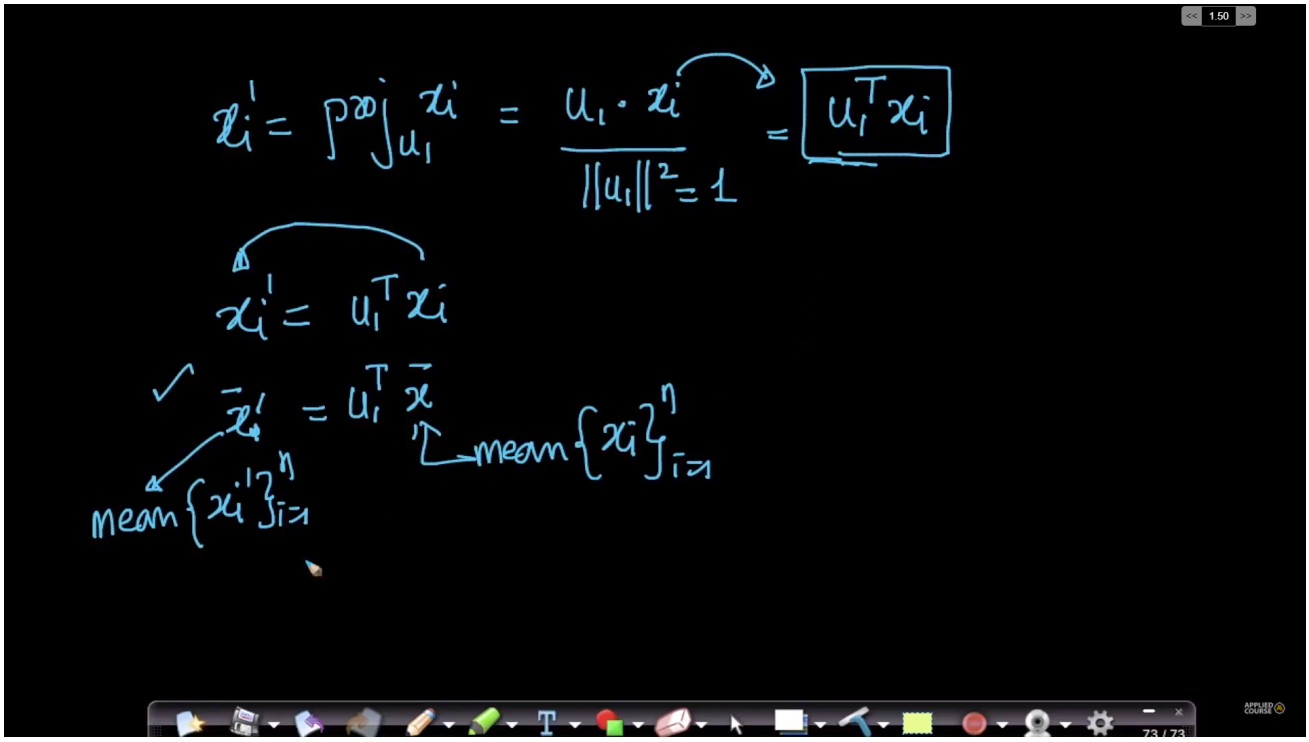
Towards the direction of f1' the maximum variance is captured. Here, We can drop f2' the perpendicular to f1'.



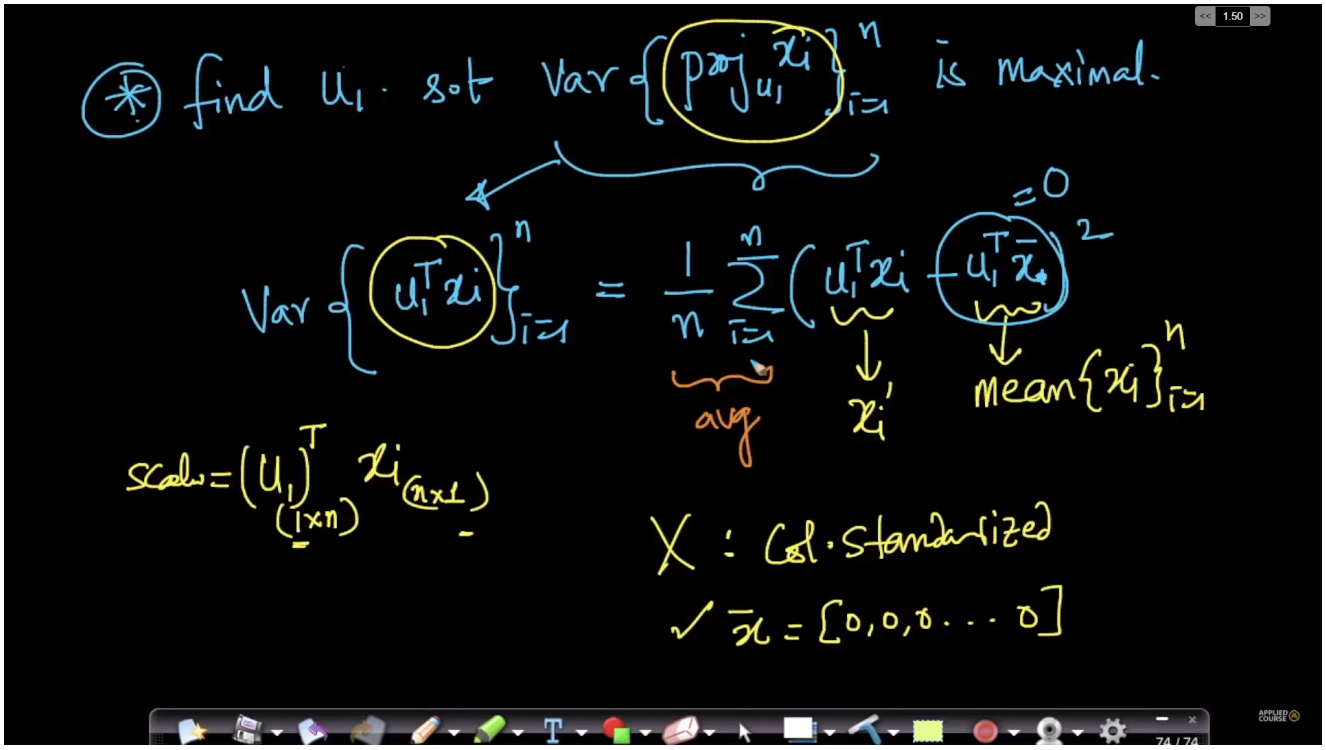


Here, the xi' is the projection on ui'(unit vector along the maximum variance).

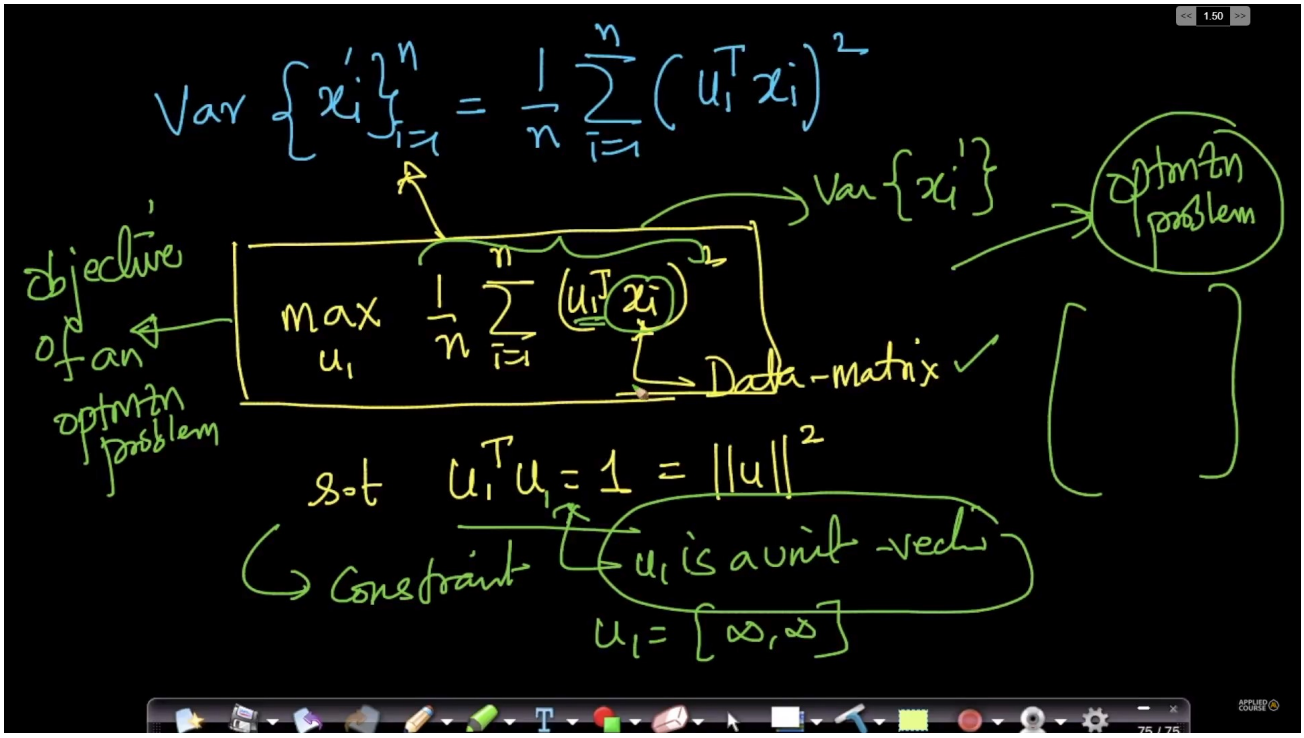
Formulation of ui:



The objective is to find the ui' vector of maximum projected variance values of xi on ui is maximal.

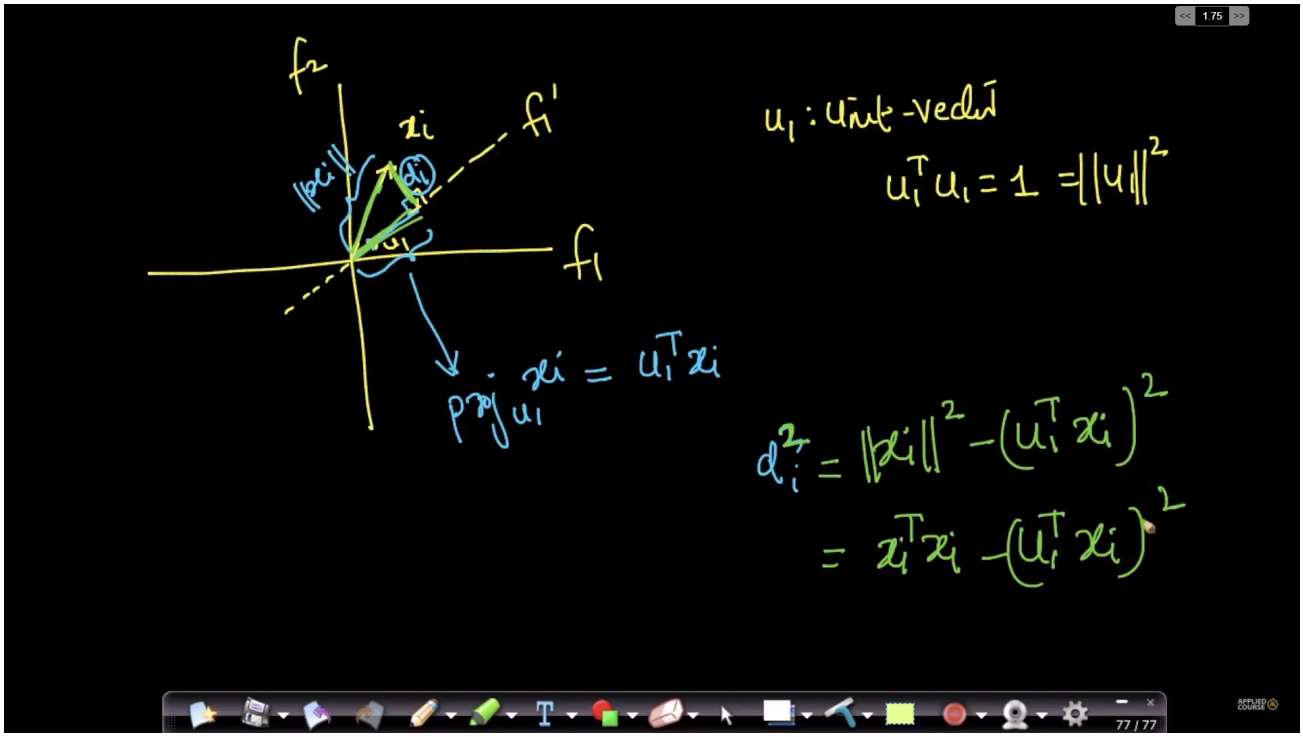


Defining PCA Objective function for variance maximization:

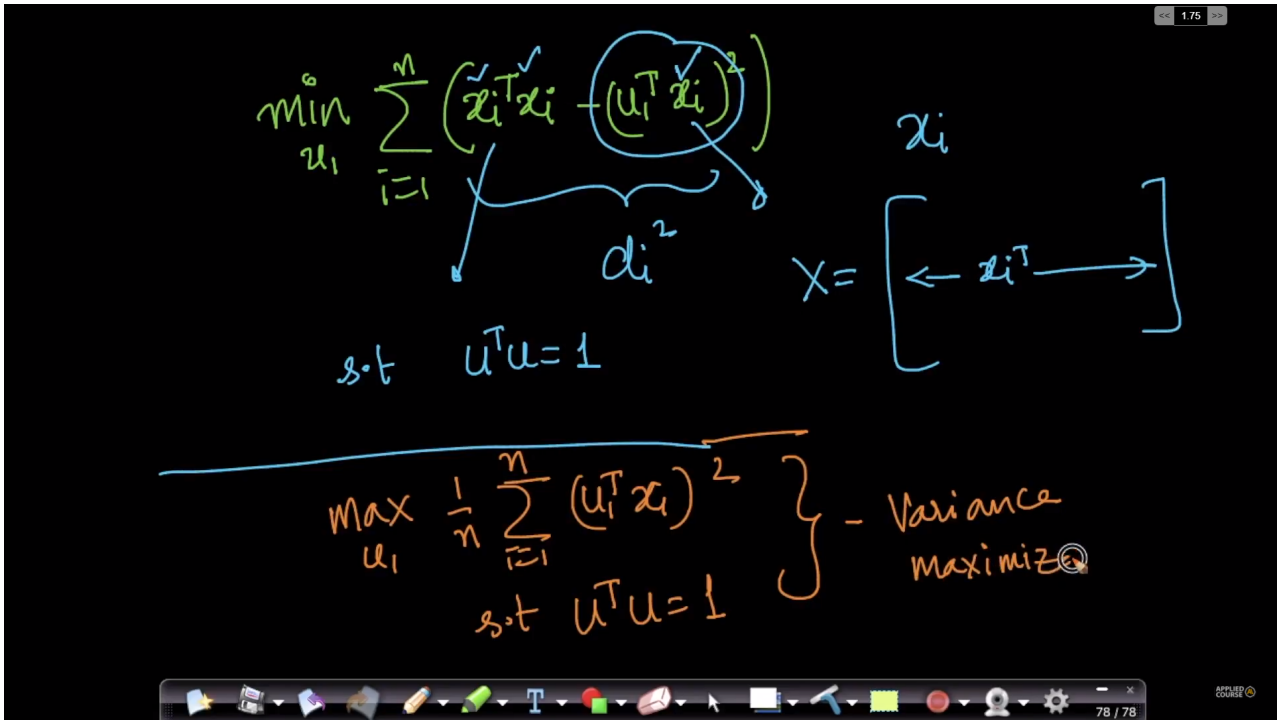


The objective function if to maximize the function on **constraint** ui is a unit vector.

Distance minimization:

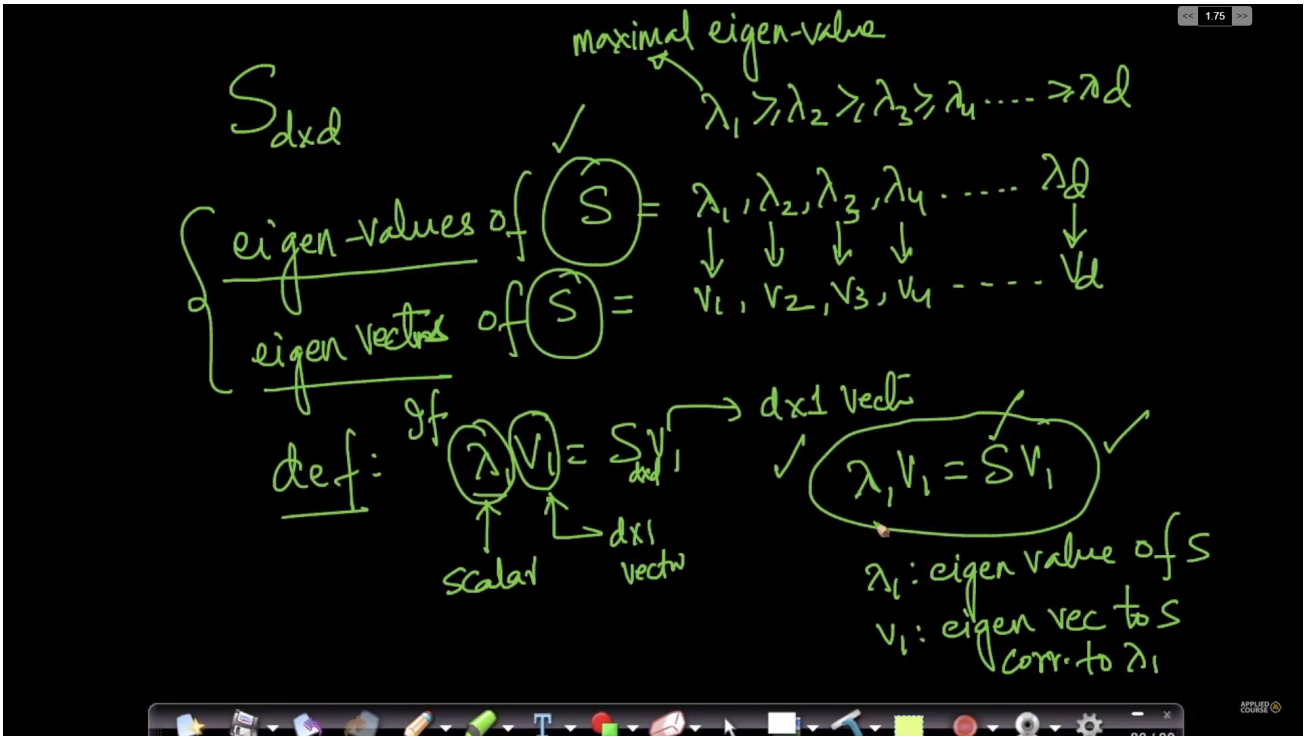


Here di is the is the distance of the point xi on the unit vector, the distance di must be minimization.



Here the distance must be minimized with (ui transpose) * (ui) is equal to 1.
Two kinds of minimization:

Eigen values and Eigen Vectors:
These are used to get the direction of the ui.

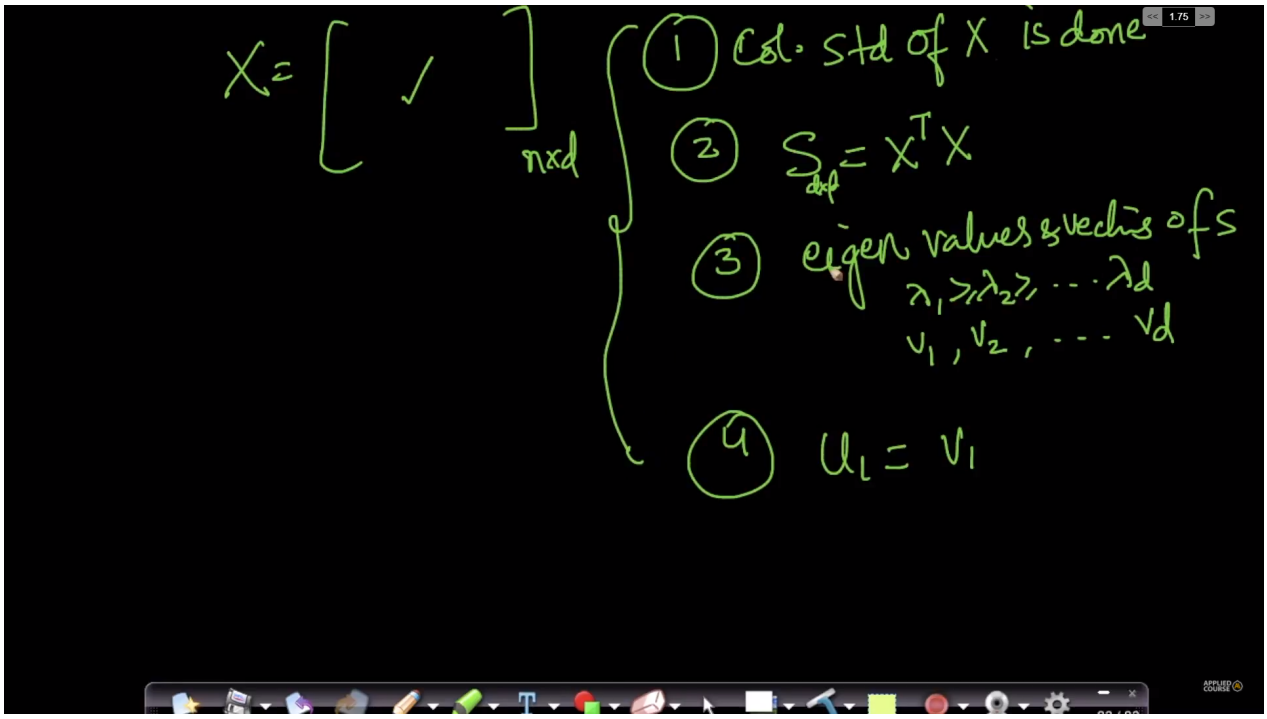


Every pair of Eigen vectors are perpendicular to each other.

Step by step procedure of compunction PCA and obtaining ui.

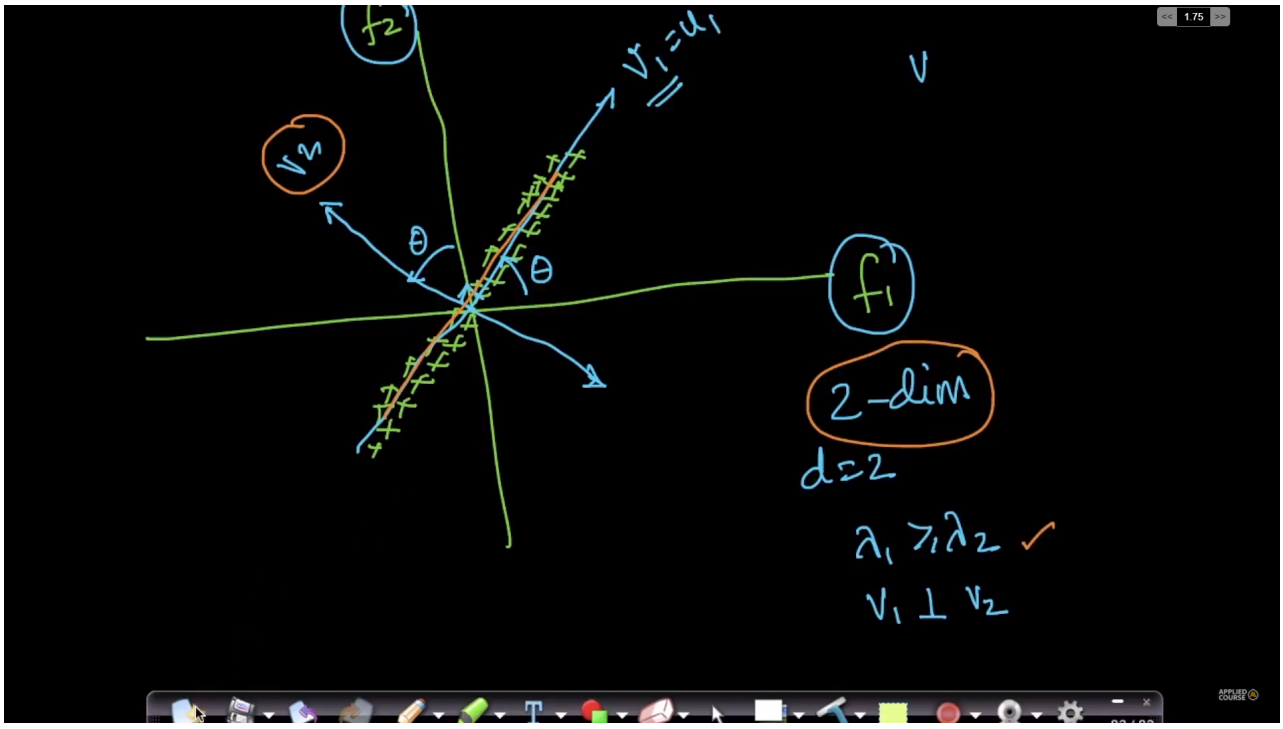
Here each eigen vector corresponds to each feature.

Formulation:

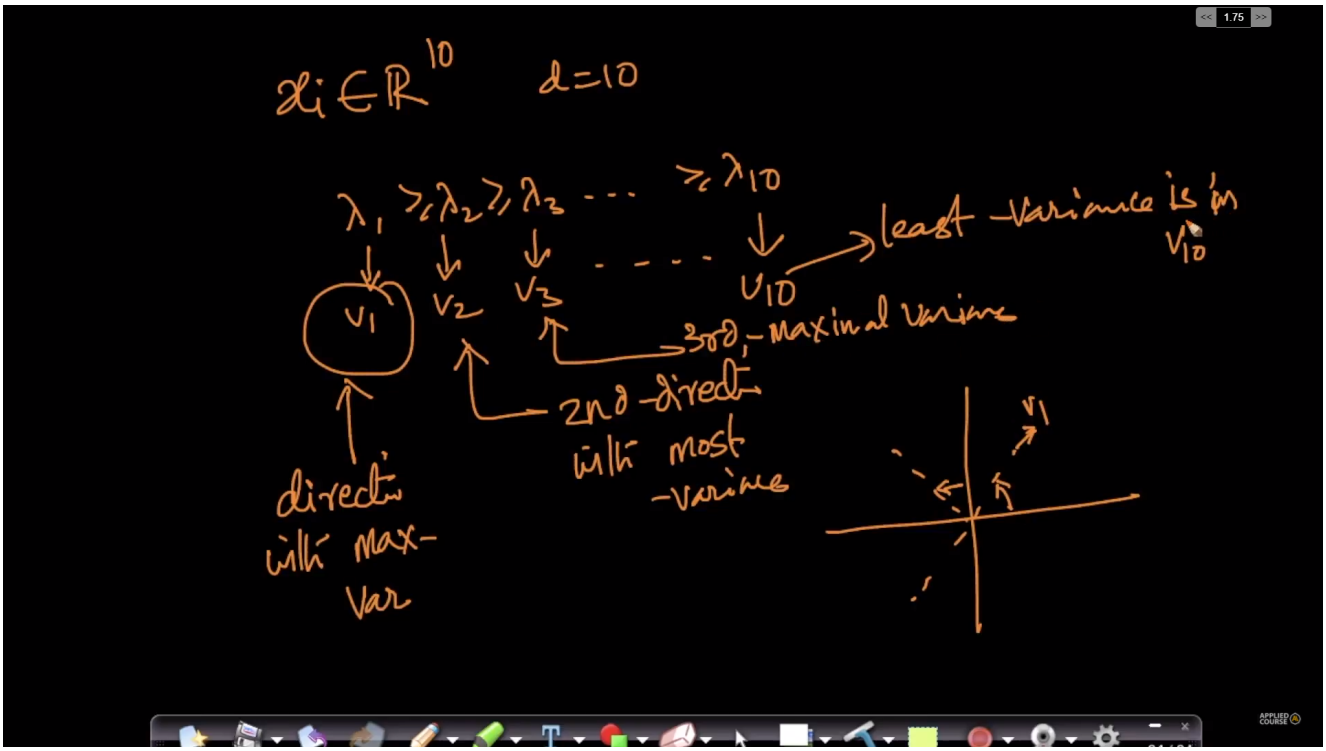


Geometry:

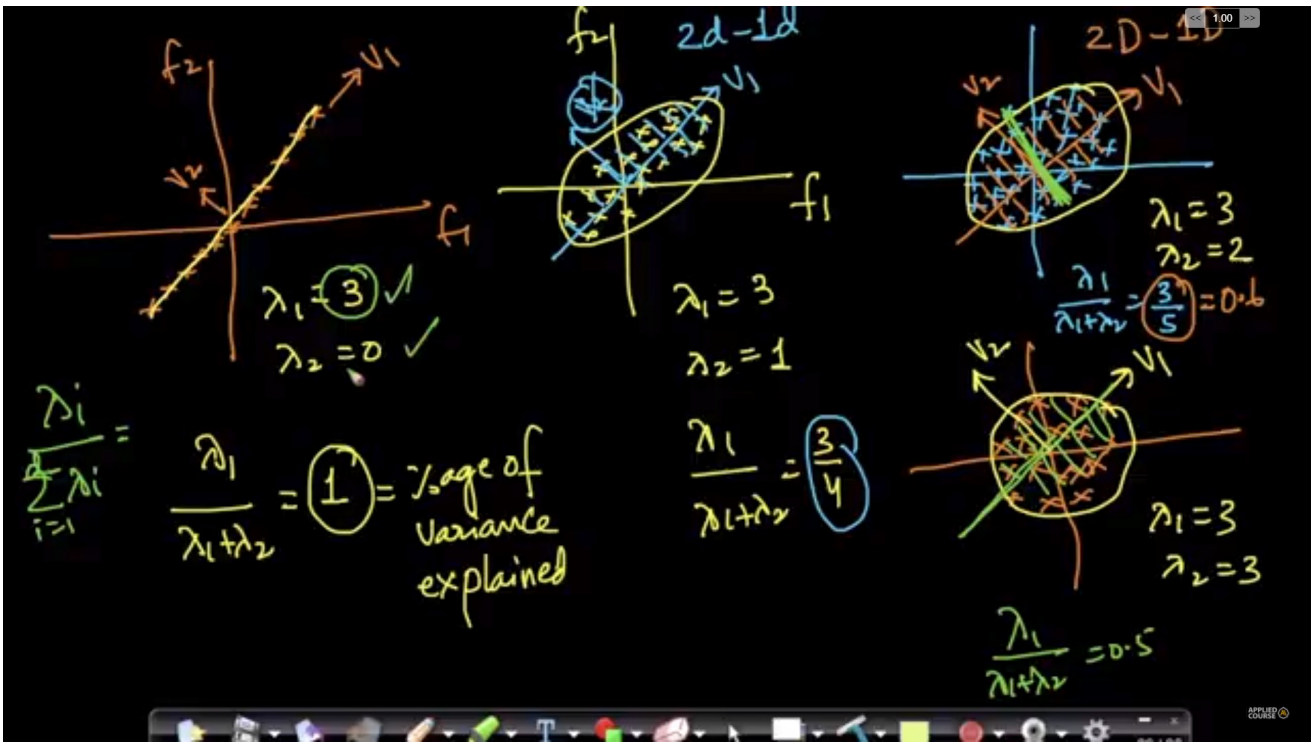
This is for two dimensional data.



This if for 10 dimensional data.



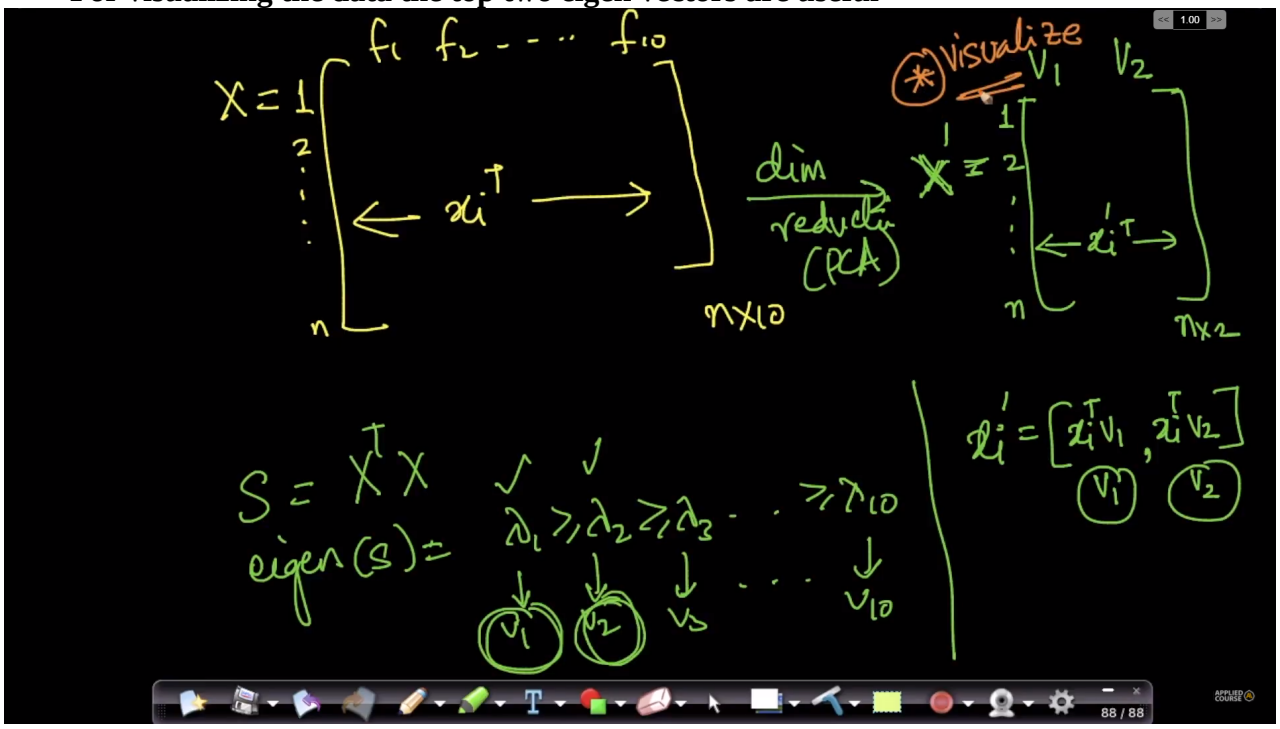
Explanation of lambdas: lambda's explain percentage of variance, the percentage shows



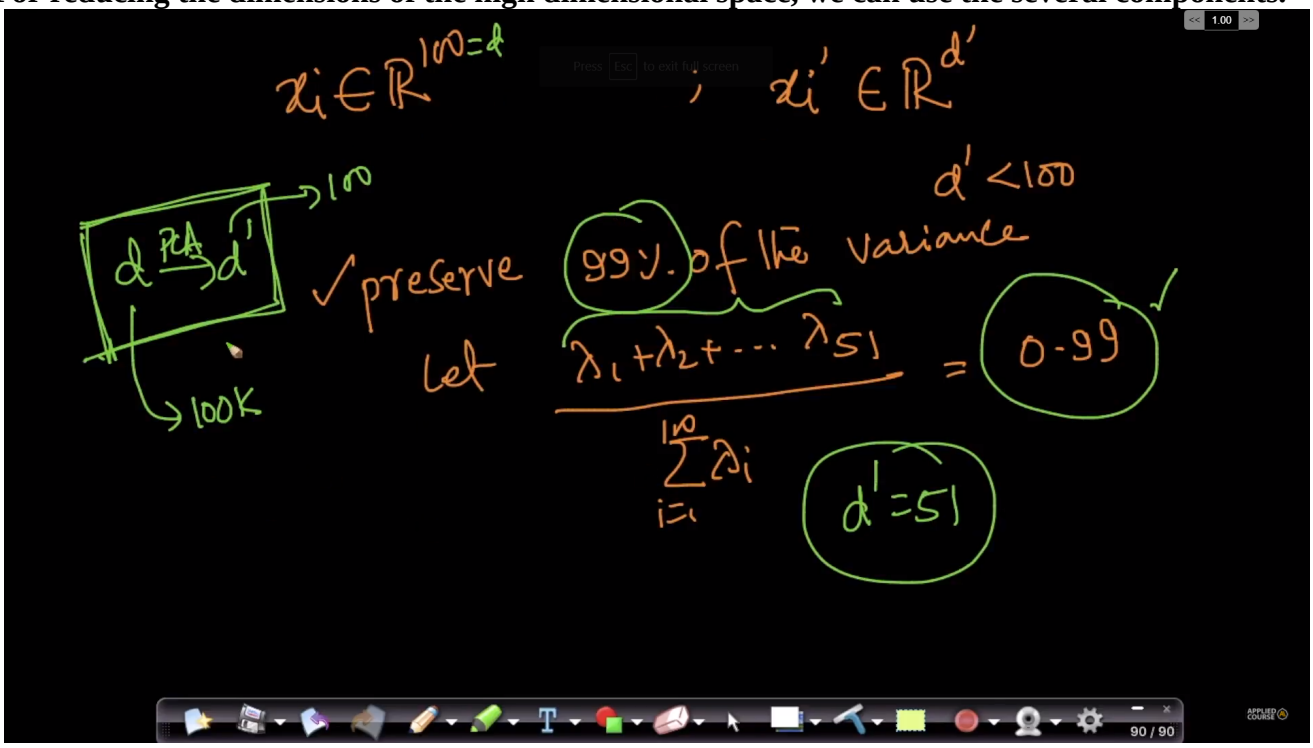
Percentage of variance preserved can be known by lambdas.

Applying PCA:

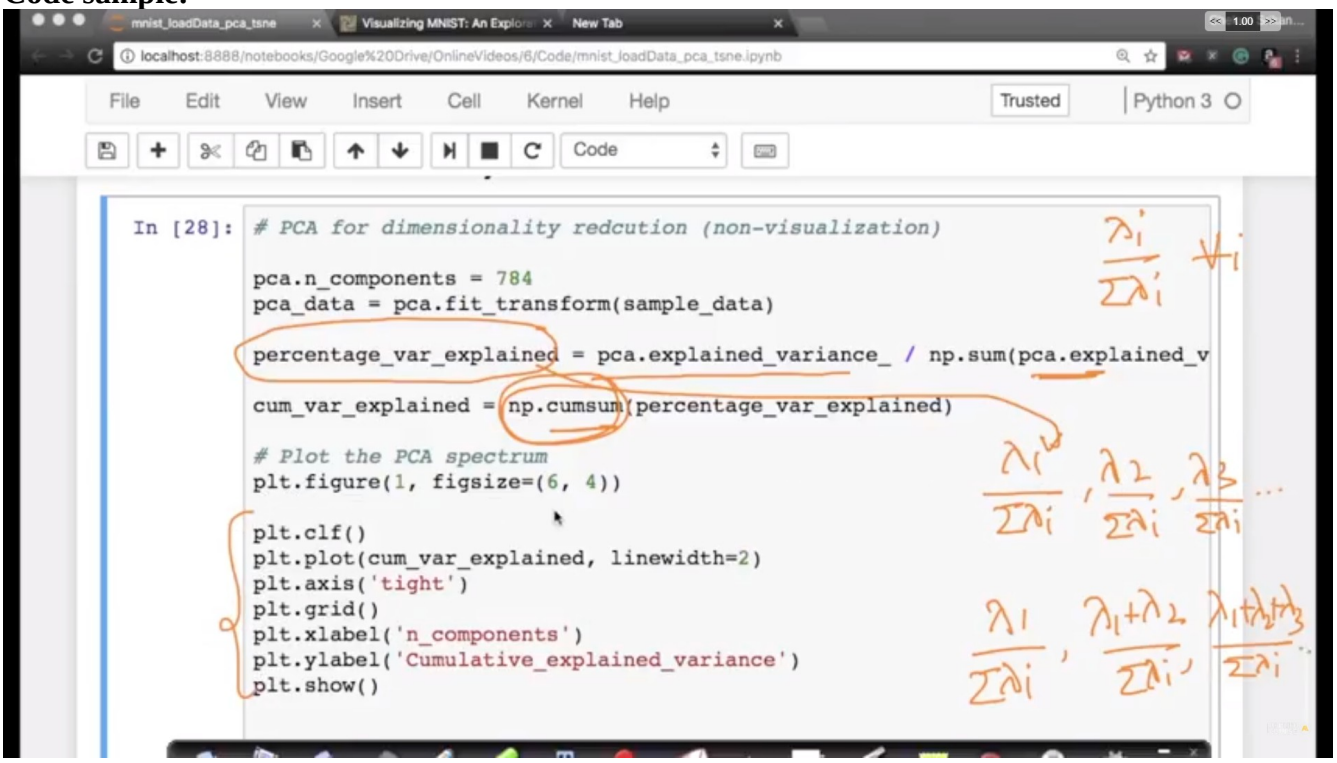
For visualizing the data the top two eigen vectors are useful

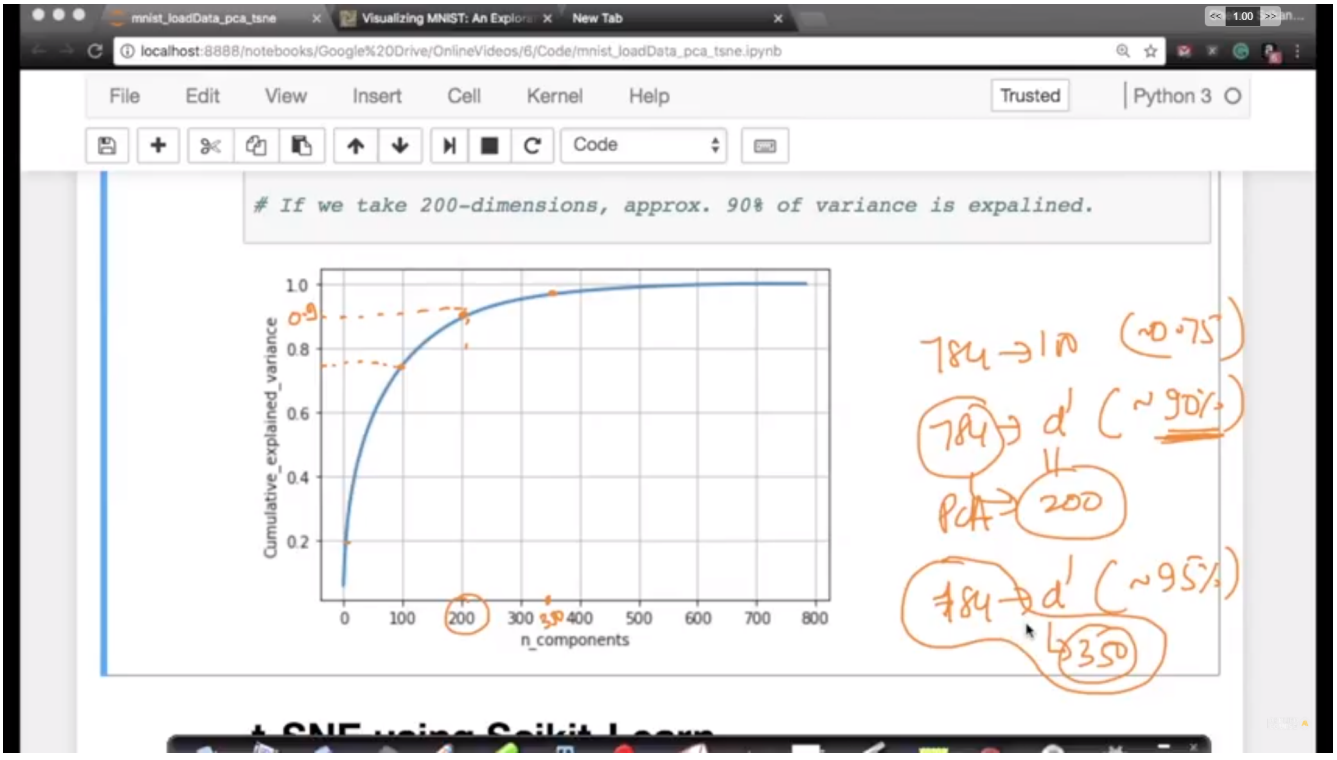


For reducing the dimensions of the high dimensional space, we can use the several components.

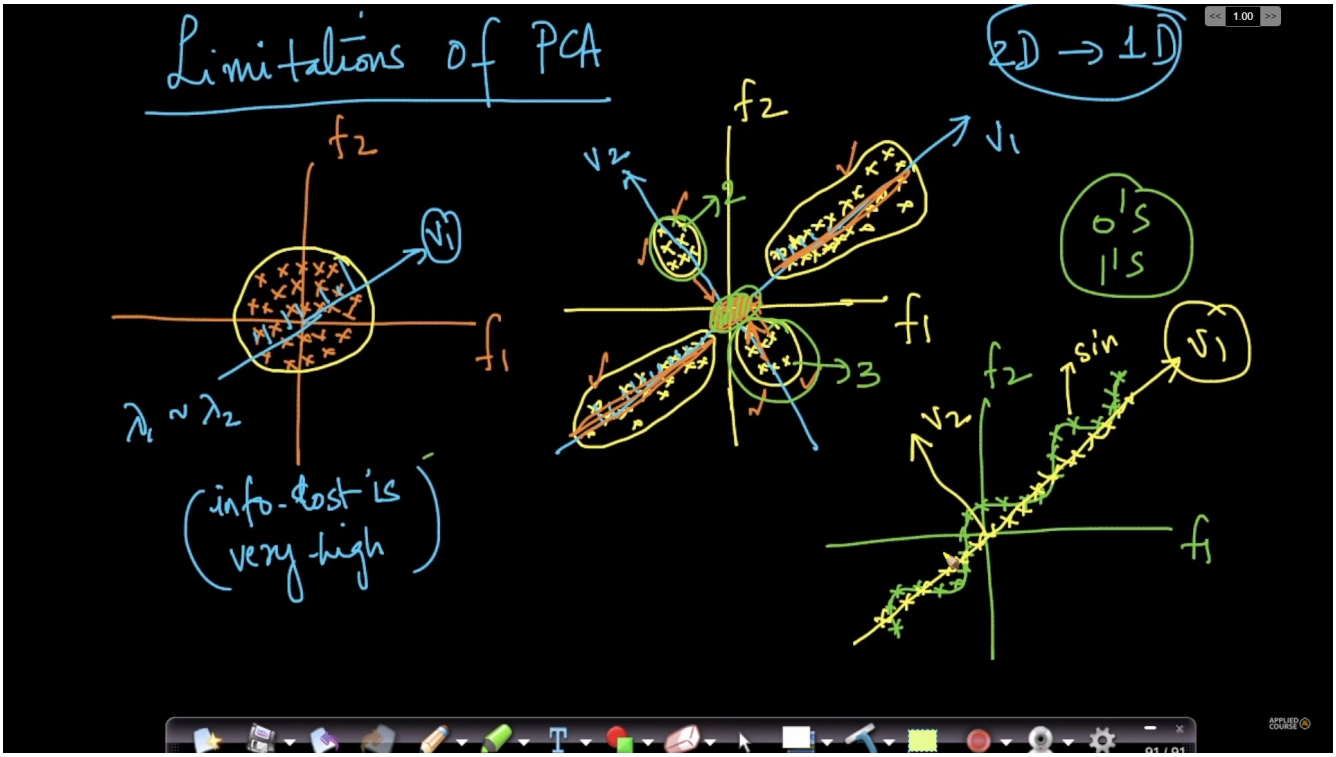


Code sample:





Limitation of PCA:



Here we PCA cannot work because the projection of the features on 'ui' will overlap on each other. And the relationship of blobs of data that are previously present is lost.

Jupyter notebooks:

C:\Users\rahul\Downloads\AppliedAIcourse\Notes\13.10#14.9#14.10#15.7.pdf