Natural Language Processing with Classification and Vector Spaces || Week — 03 (Vector Space Models)

medium.com/@aakashgoel12/natural-language-processing-with-classification-and-vector-spaces-week-03-vector-space-4c2a6582b8c1

July 26, 2020

Vector space models used

→To capture difference and similarity b/w sentences

- Where are you **heading** ? AND Where are you **from** ?
- What is your age ? AND How old are you ?

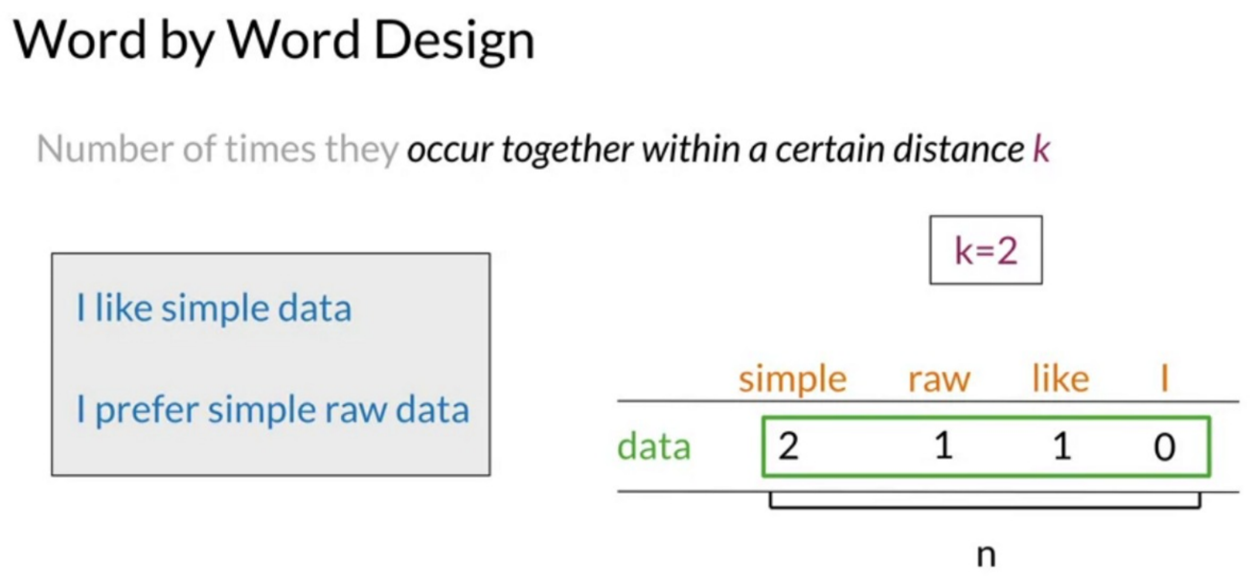
→To capture dependencies b/w words

- You eat **cereal** from a **bowl**
- You **buy** something and someone else **sells** it

→Information extraction, Machine Translation, Chatbot

Different ways to get vector space

→ Word by word



Word by Word Design

Number of times they *occur together within a certain distance* k

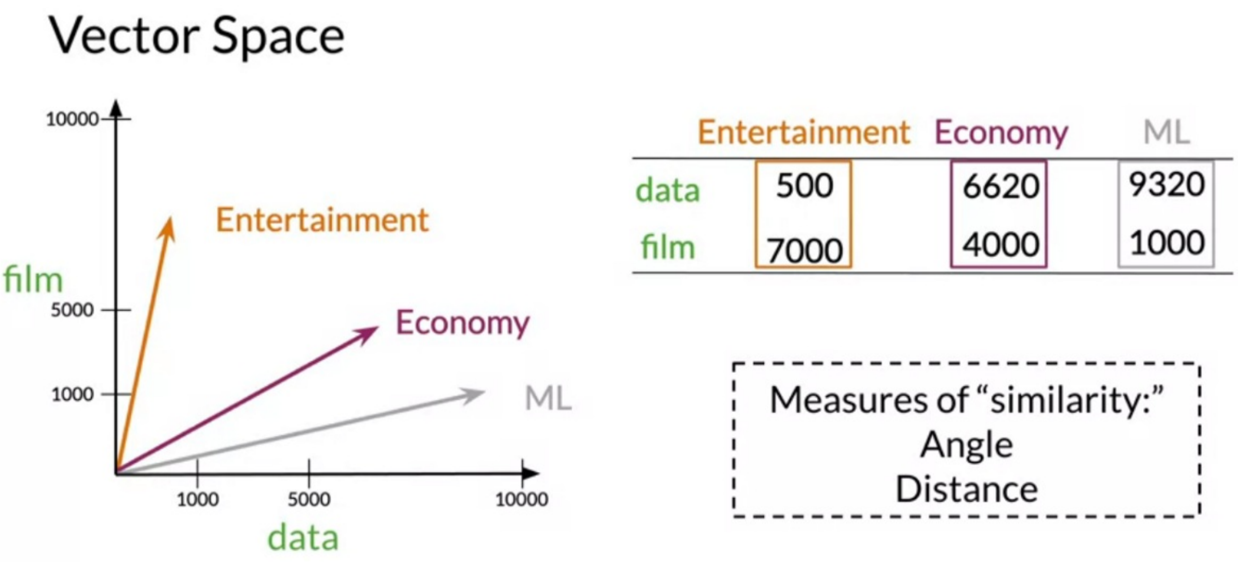
I like simple data

I prefer simple raw data

k=2

	simple	raw	like	I
data	2	1	1	0

→Word by Doc



Vector Space

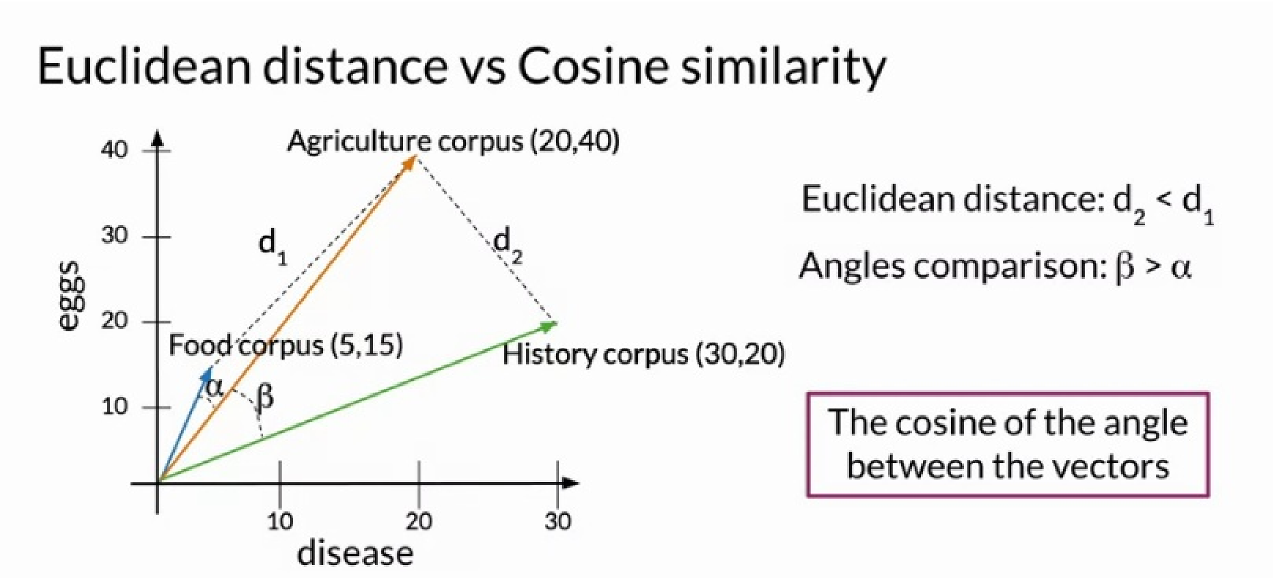
	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

Measures of "similarity:"
Angle
Distance

Refer Jupyter NB for **Linear Algebra.**

Euclidean and Cosine

Problem with Euclidean usually when compare vector representation of doc or corpora



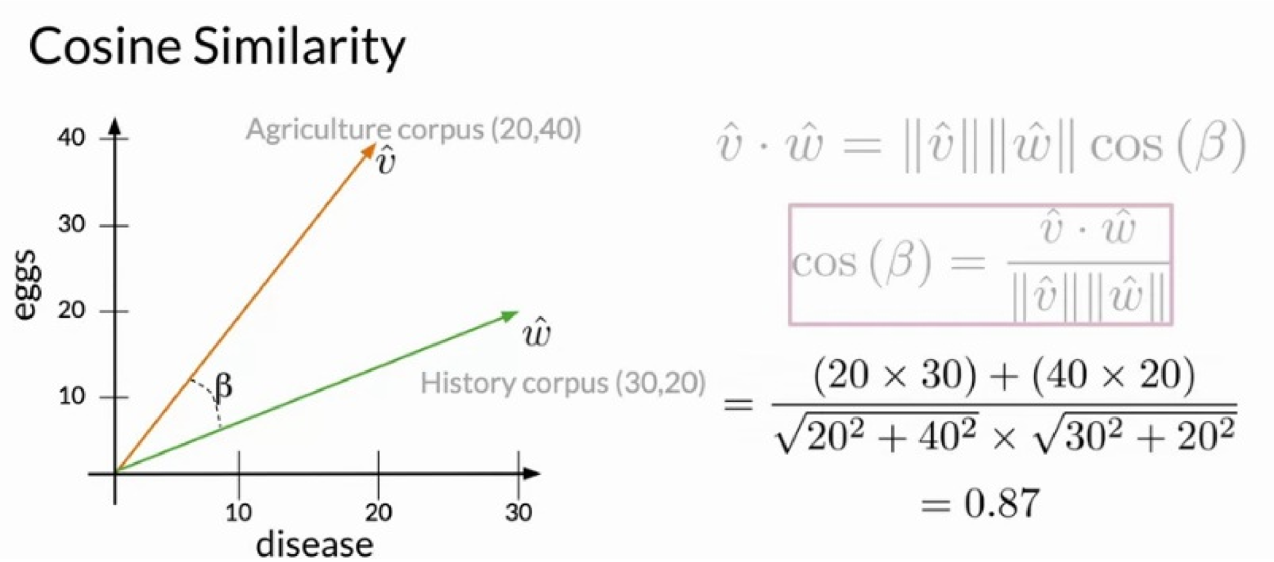
Euclidean distance vs Cosine similarity

Euclidean distance: $d_2 < d_1$

Angles comparison: $\beta > \alpha$

The cosine of the angle between the vectors

Use Cosine similarity when corpora are of different sizes. It isn't biased by the size difference b/w representations.



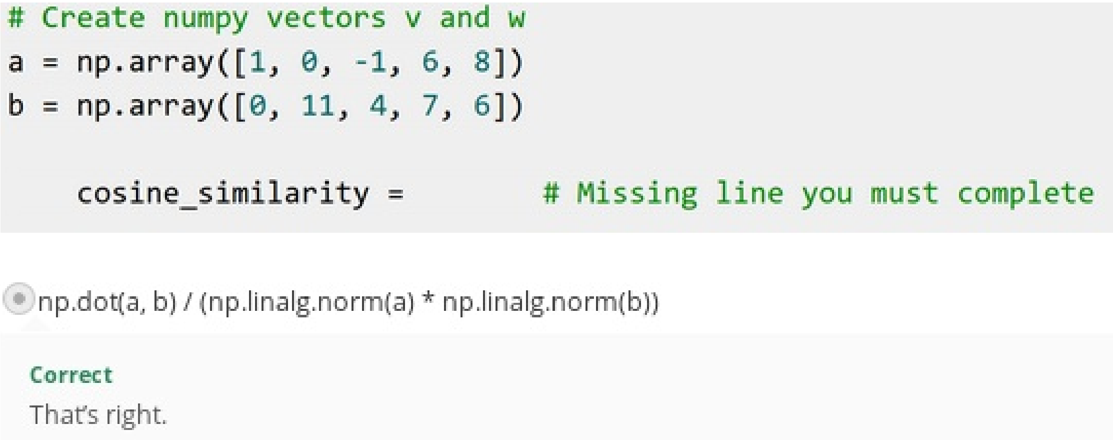
Cosine Similarity

$$\hat{v} \cdot \hat{w} = \|\hat{v}\|\|\hat{w}\| \cos(\beta)$$

$$\boxed{\cos(\beta) = \frac{\hat{v} \cdot \hat{w}}{\|\hat{v}\|\|\hat{w}\|}}$$

$$= \frac{(20 \times 30) + (40 \times 20)}{\sqrt{20^2 + 40^2} \times \sqrt{30^2 + 20^2}}$$

$$= 0.87$$



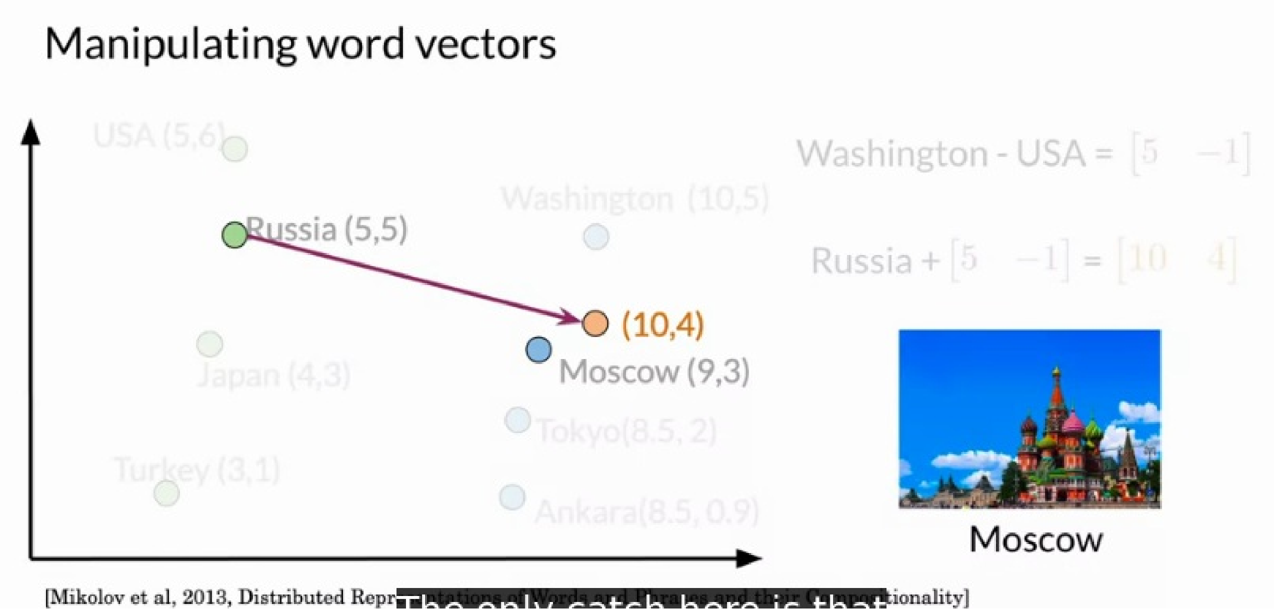
```
# Create numpy vectors v and w
a = np.array([1, 0, -1, 6, 8])
b = np.array([0, 11, 4, 7, 6])

    cosine_similarity =         # Missing line you must complete
```

● np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b))

Correct

That's right.

Manipulating Words in Vector Spaces



Manipulating word vectors

Washington - USA = $\begin{bmatrix} 5 & -1 \end{bmatrix}$

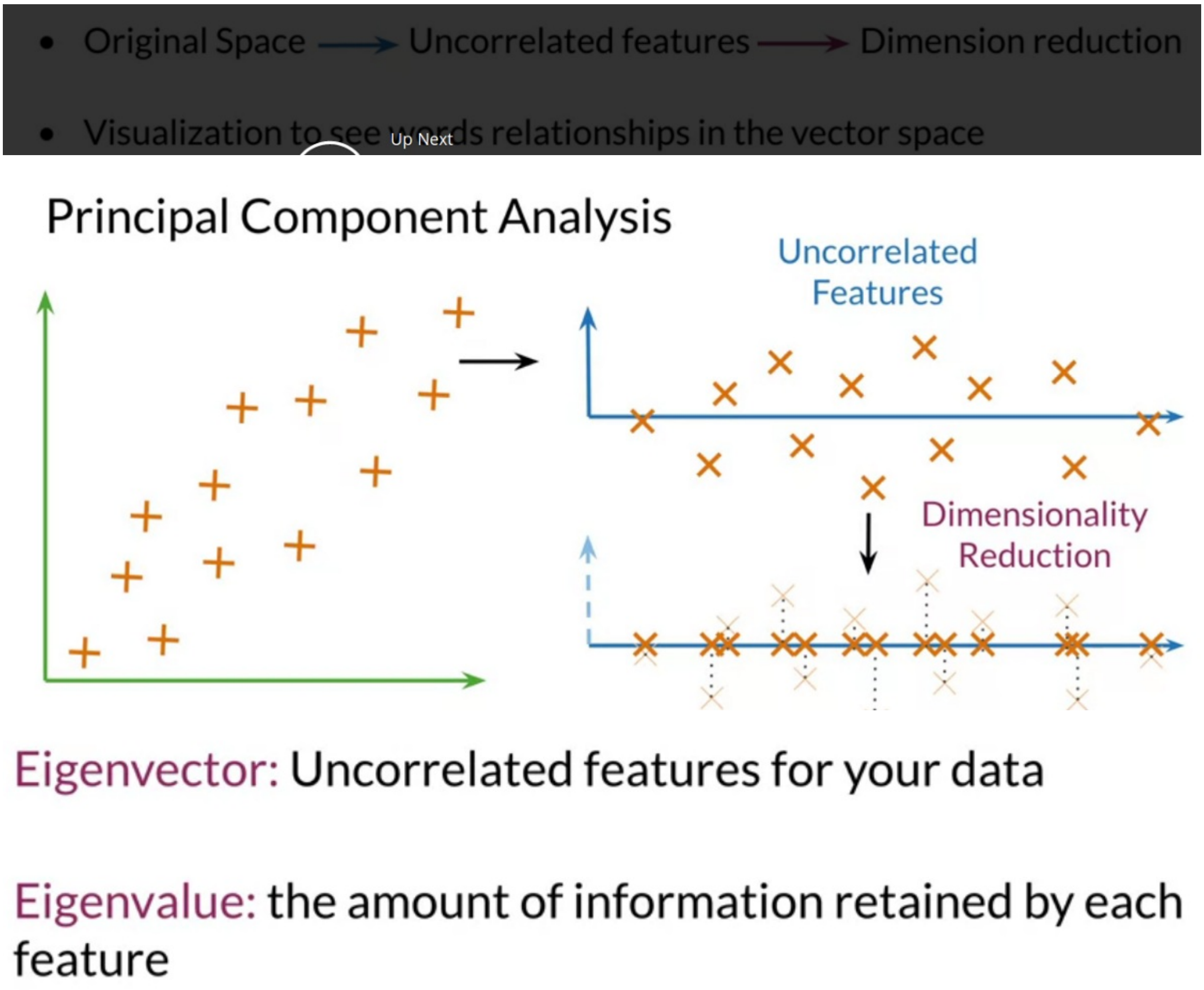
Russia + $\begin{bmatrix} 5 & -1 \end{bmatrix}$ = $\begin{bmatrix} 10 & 4 \end{bmatrix}$

[Mikolov et al, 2013, Distributed Representations of Words and Phrases and their Compositionality]

A

PCA on Word2Vec

PCA is a statistical technique invented in 1901 by Karl Pearson that uses orthogonal transformations to map a set of variables into a set of linearly uncorrelated variables called Principal Components.



- Original Space → Uncorrelated features → Dimension reduction
- Visualization to see words relationships in the vector space

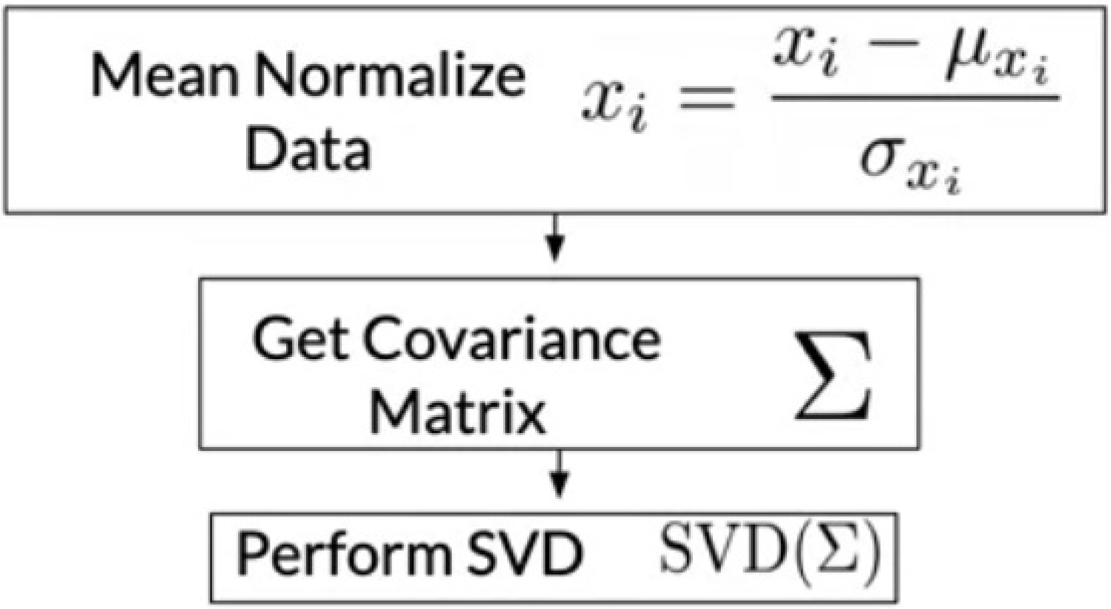
Principal Component Analysis

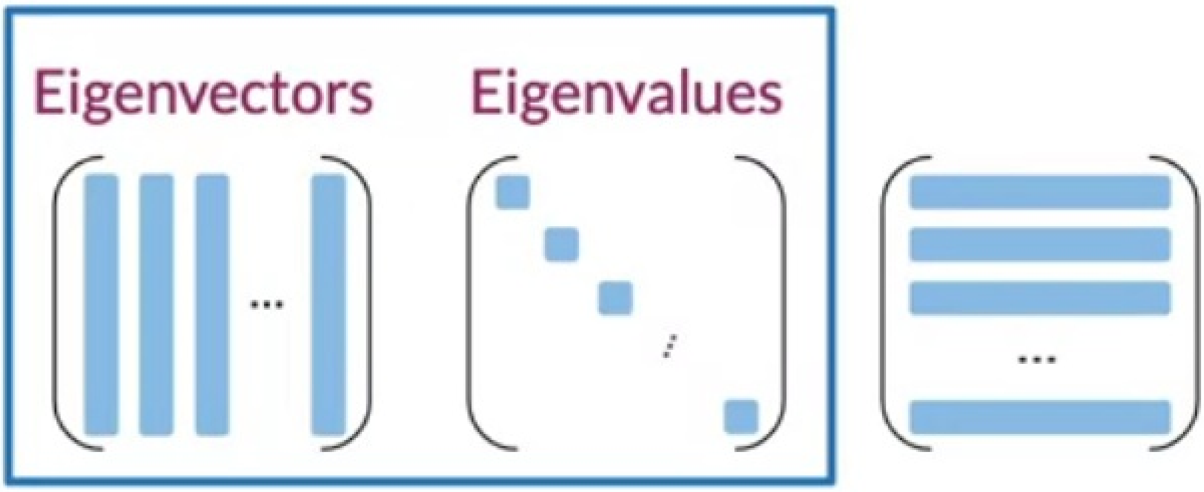
Eigenvector: Uncorrelated features for your data

Eigenvalue: the amount of information retained by each feature

- Eigenvectors of covariance matrix from your data give directions of uncorrelated features.
- And eigen values are the variance of your data sets in each of those new features.

STEP 01: ***Get Uncorrelated features***





U Eigen vectors are stacked column wise and **S** eignevalues are present at diagonal side. It should be in descending order in order to choose high variance eigen vectors.

— — — — — — — -

On **Normalization,**

In the literature people sometimes talk about ‘normalization’ and sometimes about ‘standardization’. In general, this terminology is used somehow freely.

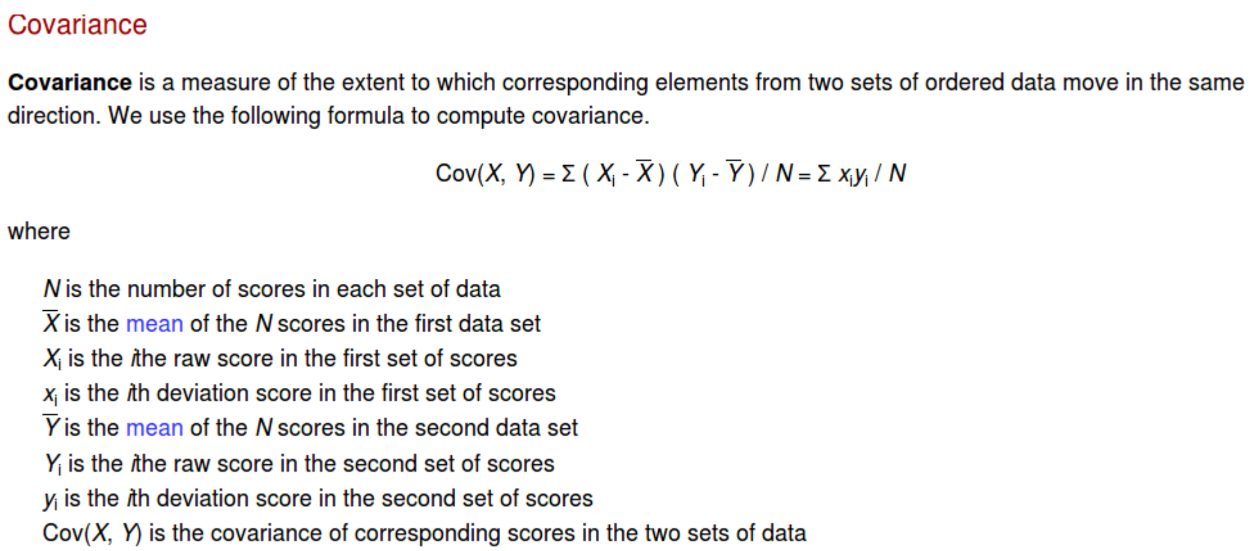
The formula presented in the video is usually referred to as ‘standardization’. The score we get using this formula is so-called ‘z-score’ or ‘standard score’.

The formula we’re supposed to use in the exercise (assignment):

$X=X-\mu(X)$ is sometimes referred to as ‘mean centering’.

It might be very useful to use standardization instead of mean centering, when performing PCA when your variables have very different scales. Note, that standardization would result in **correlation-based PCA** as opposed to **covariance-based PCA** (this is because correlation == standardized covariance).

— — — — —



Covariance

Covariance is a measure of the extent to which corresponding elements from two sets of ordered data move in the same direction. We use the following formula to compute covariance.

$$\text{Cov}(X, Y) = \Sigma (X_i - \overline{X}) (Y_i - \overline{Y}) / N = \Sigma x_i y_i / N$$

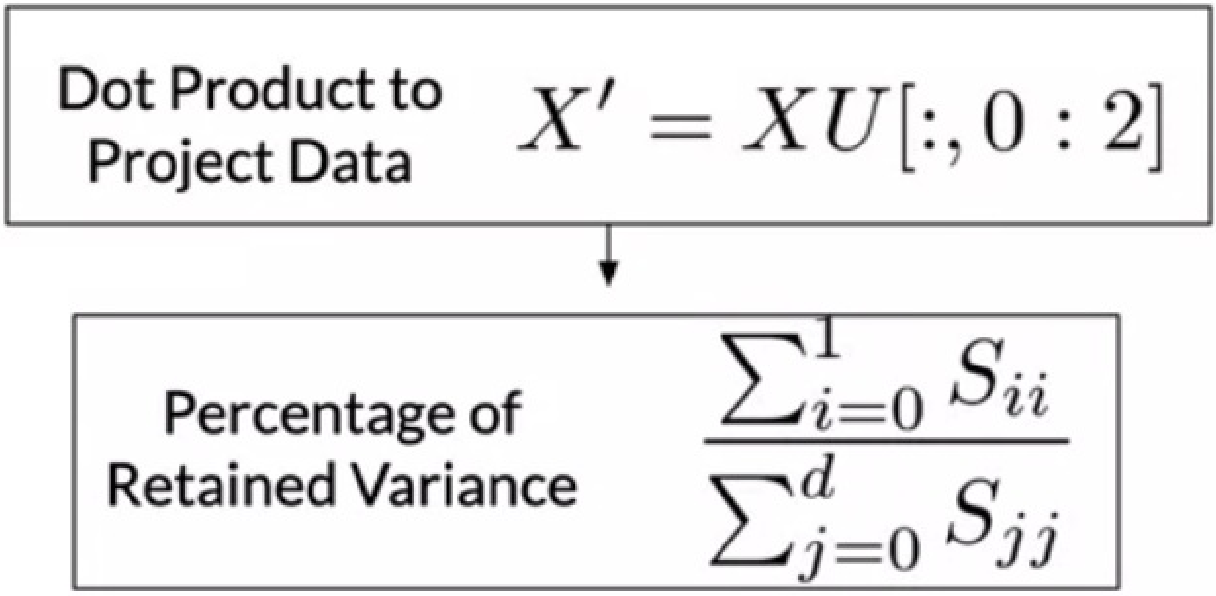
where

N is the number of scores in each set of data
$\overline{X}$ is the mean of the N scores in the first data set
X_i is the $i$the raw score in the first set of scores
x_i is the ith deviation score in the first set of scores
$\overline{Y}$ is the mean of the N scores in the second data set
Y_i is the $i$the raw score in the second set of scores
y_i is the ith deviation score in the second set of scores
Cov(X, Y) is the covariance of corresponding scores in the two sets of data

https://stattrek.com/matrix-algebra/covariance-matrix.aspx

STEP 02: ***Dimension reduction***

Take n columns of eigen vector matrix and perform its dot product with original data matrix.



Dot Product to Project Data

$$X' = XU[:, 0:2]$$

Percentage of Retained Variance

$$\frac{\sum_{i=0}^{1} S_{ii}}{\sum_{j=0}^{d} S_{jj}}$$

Word Analogies Task (Semantic and syntactic analogies data sets**)**

→ semantic analogy reasoning dataset based on countries and their capitals. For example: *Paris* is to *France* as *Rome* is to *Italy*;

→ syntactic analogy reasoning dataset based on positive-comparative form relationship in adjectives. For example: *big* is to *bigger* as *young* is to *younger*;