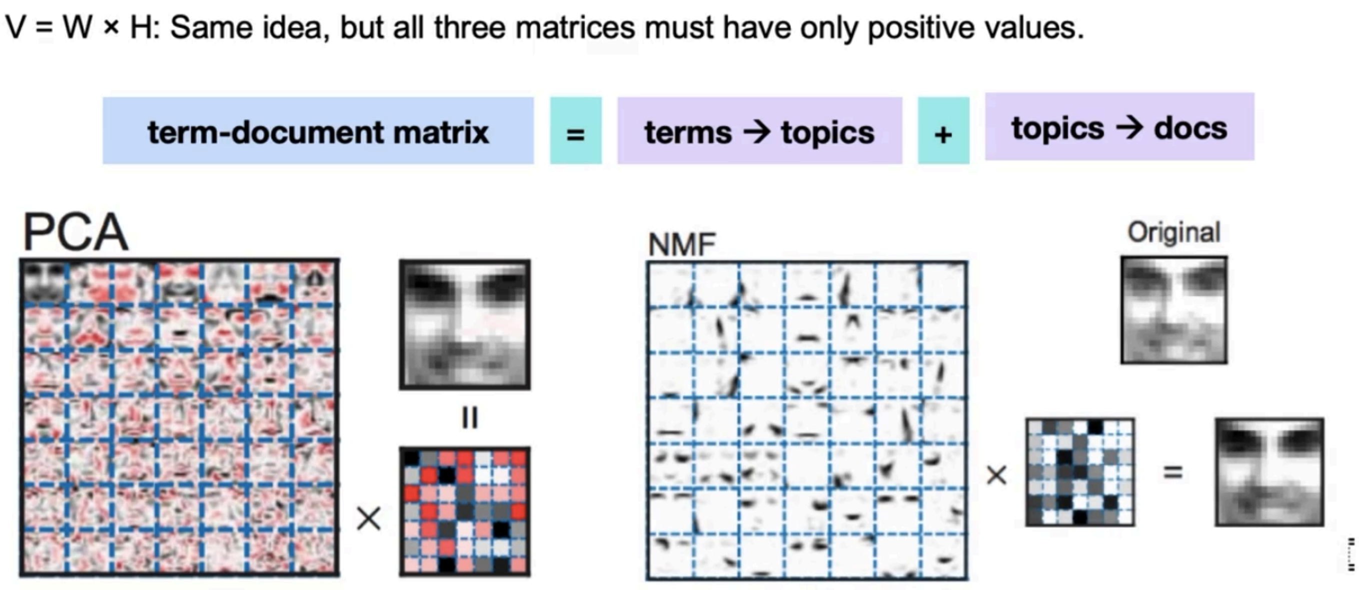
16) NMF

- Stands for "Non-negative Matrix Factorisation"
- V = W * H: Same idea, but all three matrices must have only positive values
- "Highly recommended when you have to transform higher dimensions into lower dimensions, and you're okay to lose their original features in the process as new ones are being introduced"
- It has been proven to be powerful for word and vocabulary recommendation, image processing problems, text mining, transcription processes, and it can also handle decomposition of non-interpretable data objects (like videos, music, or images)
- Since NMF only works with positive values, it cannot undo the application of a latent feature, it is much more careful about what it adds at each step. Thus each included feature must be important, it cannot cancel it out down the line
- Since its only +ve values, this leads to features that may be interpretable, as they must all add together to recreate our original data
- Since NMF has the extra constraint of +ve values, it will tend to lose more information when truncating (meaning if we ever end up with -ve values, the algo will convert/truncate them to 0)
- Unlike PCA, theres no constraint of only orthogonal latent vectors



V = W × H: Same idea, but all three matrices must have only positive values.

term-document matrix = terms → topics + topics → docs

```
from sklearn.decomposition import NMF

model = NMF(n_components=3, init='random')
X_nmf = NMF.fit(X_orig)
```