# Truncated SVD

Snowball Stemming
Porter stemming

$$A = U * S * V^T$$

$$S = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \sigma_3 & & \\ & & & \ddots & \\ & & & & \sigma_r \end{bmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \geq \sigma_r \geq 0$$

k-largest Singular values → keep them.
others → = zero.

first

Use only K colums of U, V.

K → tradeoff b/o time and data. (accuracy).

87773 × 11524
↓
features

→ First step is to get words Co-occurrence matrix.

↓

Has 2 or more words occur together in a given corpus.

Example

penny wise and pound foolish.
a penny saved is a penny earned

| | a | and | earned | foolish | is | penny | pound | saved | wise |
|---|---|---|---|---|---|---|---|---|---|
| a | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| penny | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

Bigram frequency.

Need N×N to represent bigram frequencies
↳ this table is highly sparse.
(as most are 0's)

568454 x 10 ≈ 500K ⇒ After EDA, Data preprocessing, featurization

87773 x [Some columns + 11524 word2vec vector]

87773 x 11524 → Need to reduce dimensions.

1) Bag of words
2) TFIDF
3) word2vec.

Co-occurance matrix – [11524 x 11524]

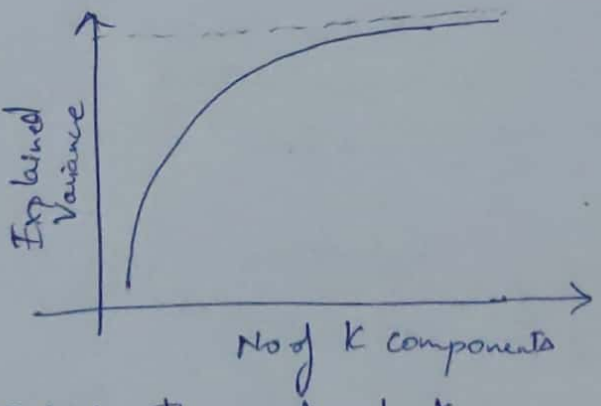
↓

get the features from Tfidfvectorizer.get_feature_names()

→ Do svd(co-occurance matrix) for different n-components (ie K-values)

→ get svd.explained_variance_ratio_.sum() for different K-values.

↳ ?




Select the optimal K.

→ Decompose 87773 x 11524 matrix.

$A_{mxn} = U_{mxm} S_{mxn} V^T_{nxn}$

$B_{mxk} = U_{mxm} S_{mxn} V^T_{nxn}$ =

$\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k \\ 0 & 0 & 0 & 0 \end{bmatrix}_{mxn}$