Singular value decomposition:

- Goal:
  - Examine this matrix and uncover its linear algebraic properties to:
  - 1.Approximate A with a smaller matrix B that is easier to store but contains similar information as A
  - 2.Dimensionality Reduction / Feature Extraction
  - 3.Anomaly Detection & Denoising
- Linear algebra review:
  - **Definition**: The vectors in a set **V = {** $\vec{v}_1$, …, $\vec{v}_n$ **}** are **linearly independent** if
    - $a_1\vec{v}_1 + \dots + a_n\vec{v}_n = \vec{0}$
  - can only be satisfied by $a_i = 0$
  - **Note**: this means no vector in that set can be expressed as a **linear combination** of other vectors in the set.
- Definition:
  - The **determinant** of a square matrix A is a scalar value that encodes properties about the **linear mapping** described by A.
  - $$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$
- Property:
  - **n** vectors **{** $\vec{v}_1$, …, $\vec{v}_n$ **}** in an n-dimensional space are **linearly independent** iff the matrix **A**:
    - $A = [\vec{v}_1, \dots, \vec{v}_n]$ (n x n)
  - has non-zero determinant.
  - **Q**: Can **m > n** vectors in an **n**-dimensional space be linearly independent?
- Definition:
  - A **basis B** of a vector space (over a field **F**) is a **linearly independent** subset of **V** that **spans V**. **B spans V** if for every vector **v** in **V** it is possible to choose v_1, …, v_n in **F** and $\vec{b}_1$, …, $\vec{b}_n$ in **B** such that:
    - $v = v_1\vec{b}_1 + \dots + v_n\vec{b}_n$
  - Ex: North & East in 2d-plane
- The **rank** of a matrix **A** is the dimension of the vector space spanned by its column space. This is equivalent to the maximal number of linearly independent columns / rows of **A**.
- **Definition**:
  - A matrix **A** is **full-rank** iff **rank(A) = min(m, n)**
  - **Note:** Get the rank of a matrix through the **Gram-Schmidt process**
- Approximation:
  - In practice, matrices describing our dataset contain a lot of redundant information.
  - It would be great to capture all the information of our dataset in the least amount of space possible.

- To store an **n x m** matrix **A** requires storing **m · n** values.
- However, if the rank of the matrix of **A** is **k**, **A** can be factored as
    - **A = UV**
- Where **U** is **n x k**; **V** is **k x m**; which requires storing **k(m + n)** values.
- Frobenius Distance:
    - $$d_F(A, B) = \|A - B\|_F = \sqrt{\sum_{i,j}(a_{ij} - b_{ij})^2}$$
    - The **i^{th} singular vector** represents the direction of the i^{th} most variance.
- To find the right **k** you can:
    - Look at the singular value plot to find the elbow point
    - Look at the residual error of choosing different **k**
- Principal component analysis:
    - Idea: project the data onto a subspace generated from a subset of singular vectors / principal components.
    - We want to project onto the components that capture most of the variance / information in the data.
- **Latent semantic analysis:**
    - Inputs are documents. Each word is a feature. We can represent each document by:
        - The presence of the word (0 / 1)
        - Count of the word (0, 1, …)
        - Frequency of the word (n_i / $\boldsymbol{\Sigma} n_i$)
        - TfiDf
- Anomaly detection:
    - Define $\mathbf{O = A - A^{(k)}}$
    - The largest rows of **O** could be considered anomalies