Unsupervised Learning

Ananyapam De

Research Assistant, TU Clausthal

Quick Question!

- Suppose you have a nice square matrix $\boldsymbol{A}$ and a randomly sampled vector $\boldsymbol{v}$ from the space.
- Now suppose you kept transforming your vector $\boldsymbol{v}$ by $\boldsymbol{A}$ for quite a number of times, i.e. you compute $\boldsymbol{Av}, \boldsymbol{A}(\boldsymbol{Av}), \boldsymbol{A}(\boldsymbol{A}(\boldsymbol{Av}))$, … and so on until you are tired.
- Where do you think your vector will finally land up?

Content

- But what is Unsupervised Learning?
- Recap of Linear Regression
- Some Unsupervised Learning methods
  - Latent Variable Models
    - Singular Value Decomposition (SVD)
    - Principle Component Analysis (PCA)
    - Autoencoders
    - t-SNE (visualization)
    - Variational Inference
  - Clustering
    - K-means
  - Food for thought

But what is Unsupervised Learning?

- Generally, predictive ML tasks look like this- given a set of $\{data, targets\}_i \; i \in \{1, n\}$ we need to predict the target for the $\{new\ data\}_{n+1}$
- However for unsupervised learning tasks, we do not have any targets provided to us!
- So the challenging task is to infer the targets, with the data itself. Hence, we are essentially concerned with learning patterns from our data.
- Broad categories in unsupervised learning:
  - Clustering
  - Anomaly detection
  - Latent Variable Models

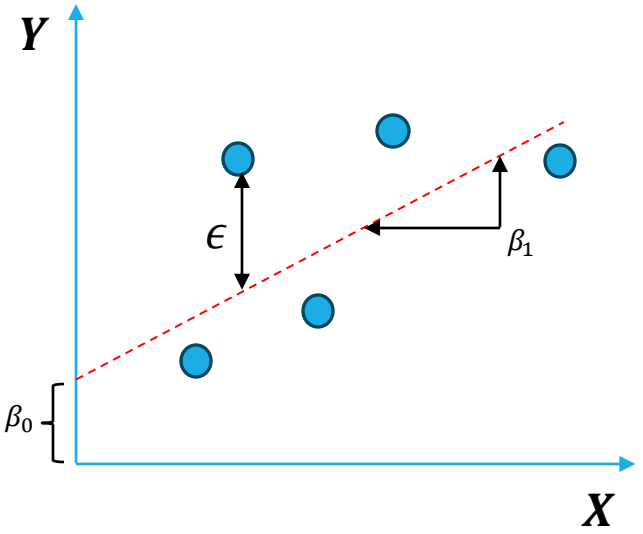
Recap of Linear regression

$$
\begin{aligned}
y_1 &= \beta_0 + \beta_1 x_1 + \epsilon \\
y_2 &= \beta_0 + \beta_1 x_2 + \epsilon \\
&\vdots \\
y_n &= \beta_0 + \beta_1 x_n + \epsilon
\end{aligned}
$$

$y_i = \beta_0 + \beta_1 x_i + \epsilon$ for $i \in \{1, n\}$

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \epsilon$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}
$$



- Least square estimates

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

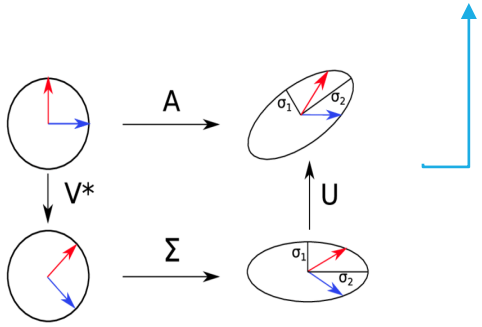
- Problems with Multicollinearity

  $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ in $\boldsymbol{b}$ cannot be calculated if the columns of $\boldsymbol{X}$ are linearly dependent

Singular Value Decomposition

- SVD of a real or complex matrix $A \in R^{m \times n}$ has three main steps:
  - Rotation
  - Followed by rescaling
  - Followed by another rotation

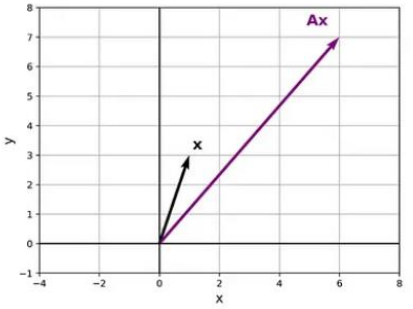
$$A = U\Sigma V^T$$

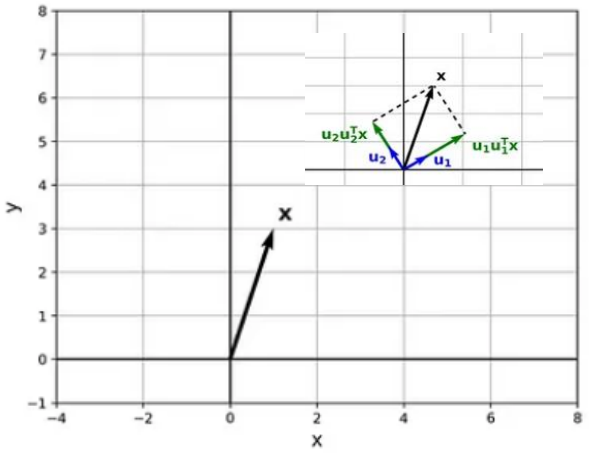


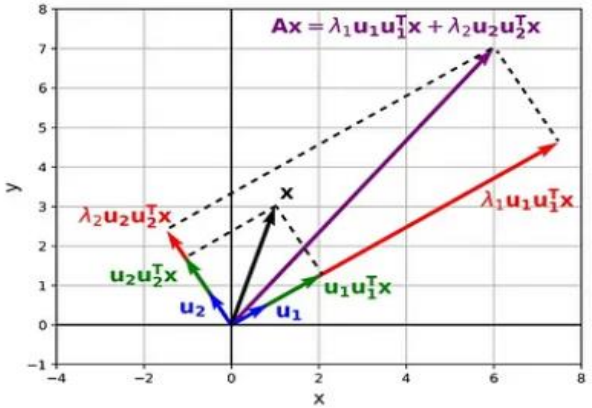
- It generalizes the eigen decomposition of a square normal matrix with an orthonormal eigen basis to any $m \times n$ matrix.

$m \geq n$ with $\text{rank}(A) = n$ as large as possible

$U \in R^{m \times m}$ is matrix of orthonormal eigenvectors of AA^T
$V \in R^{m \times m}$ is matrix of orthonormal eigenvectors of $A^T A$
$\Sigma \in R^{r \times r}$ is a diagonal matrix with elements equal to the root of the positive eigenvalues of AA^{T}



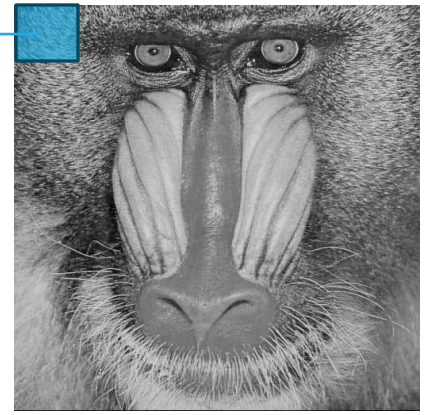




Singular Value Decomposition: Application

$$A = U\Sigma V^T$$

Each entry of matrix represents a pixel value

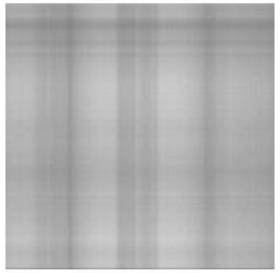


- Think now about $\boldsymbol{A}$ as containing the grayscale values of a black-and-white image
- SVD of the baboon represents the baboon image as a superposition of 512 simple images, each one only showing horizontal/vertical stripes.

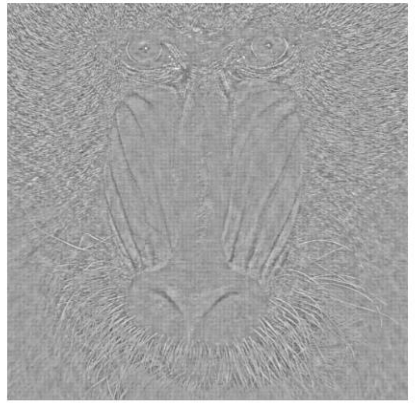
```
baboon.svd  <-  svd(bab) # May take some time

baboon.1  <-  sweep(baboon.svd$u[, 1, drop=FALSE], 2,
              baboon.svd$d[1], "*") %*%
                   t(baboon.svd$v[, 1, drop=FALSE])

baboon.20 <-  sweep(baboon.svd$u[, 1:20, drop=FALSE], 2,
            baboon.svd$d[1:20], "*") %*%
                   t(baboon.svd$v[ , 1:20, drop=FALSE])
```



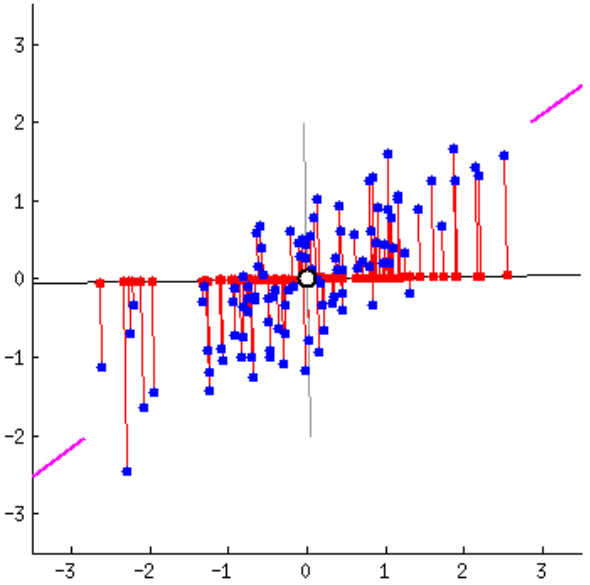
Low rank decomposition of the image with 1 and 20 components

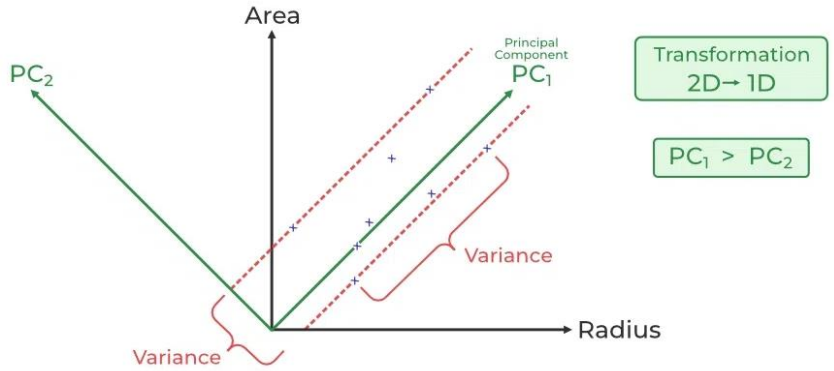


- Let us finally look at the "residual image", the image reconstructed from the 20 rank-one images with the lowest singular values.

Principle Component Analysis

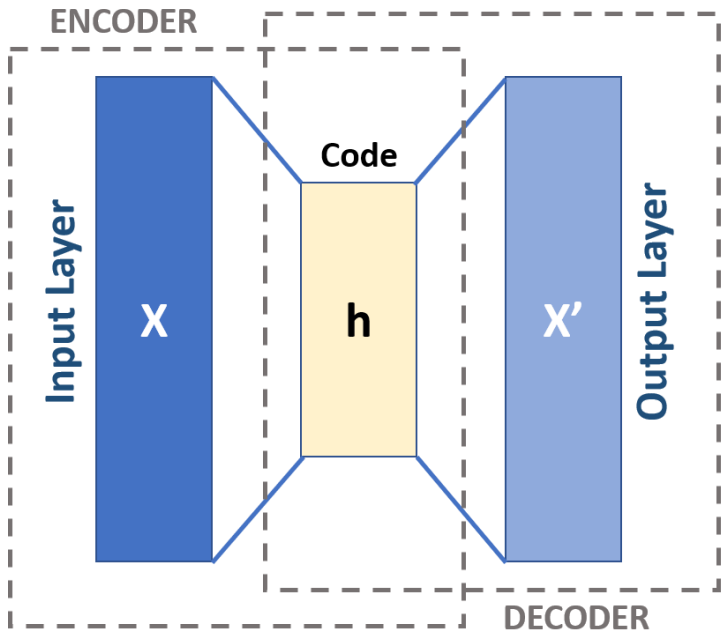
- A special case of SVD (applied on the $\boldsymbol{X}$ matrix)!
- Principal components account for the variance in the data set amongst different axes.
- Purple line: the line that maximizes the variance. (also the first component)
- The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.





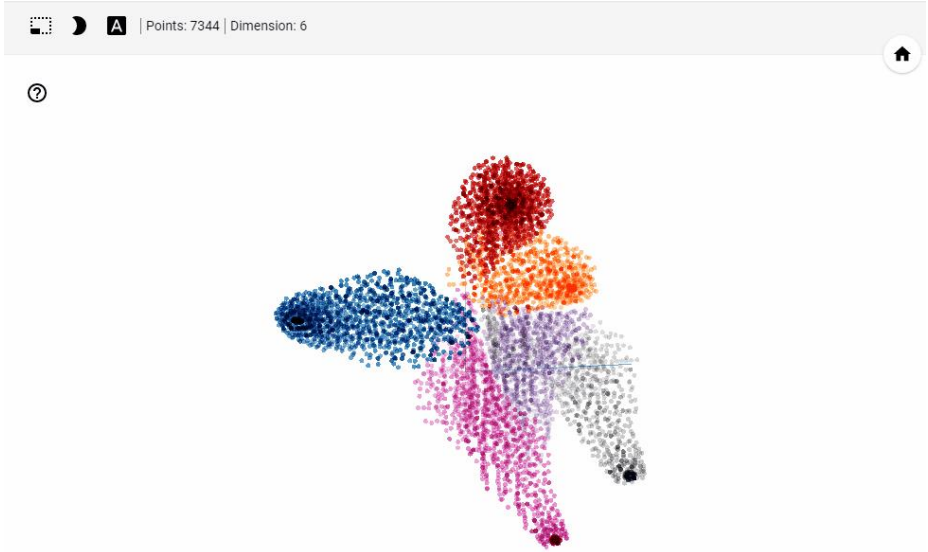
Autoencoders

- A type of neural network used to learn efficient latent representations of unlabeled data by forcing the data to be pushed into a lower dimension.
- An autoencoder learns two functions: an ***encoding function*** that transforms the input data, and a ***decoding function*** that recreates the input data from the encoded representation.
- Finally, the loss between the input and the output is minimized.
- This forces the encoding function to learn only the most important features from the data which will help it reconstruct it back.



t-distributed Stochastic Neighbour Embedding (t-SNE)

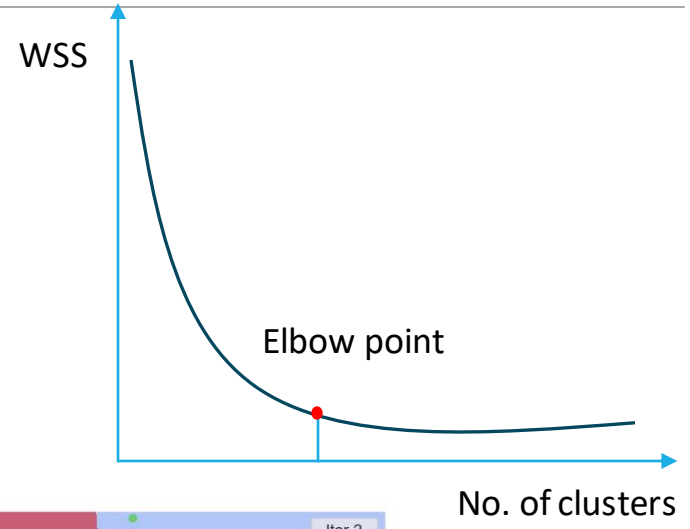
- Finds similarity measure between pairs of points in the higher and lower dimensional space and tries to optimize two similarity measures.
- Calculates pairwise similarity between all data points in the high-dimensional space using a Gaussian kernel.
- The algorithm computes pairwise conditional probabilities and tries to minimize the sum of the difference of the probabilities in higher and lower dimensions.
- The algorithm takes a lot of time to compute and has a quadratic time and space complexity in the number of data points.



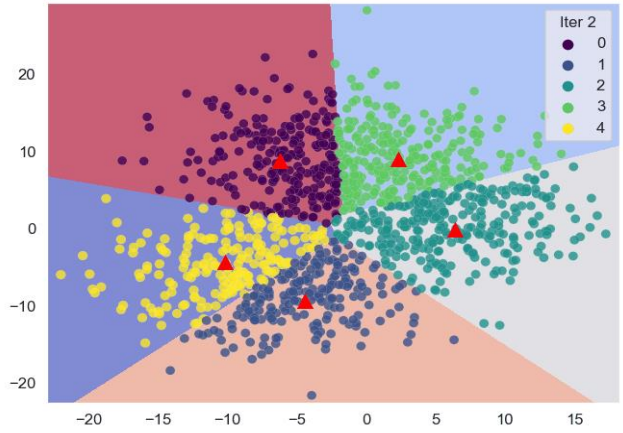
K-means clustering

- Iterative process where we minimize the distance of the data point from the average data point in the cluster.
- Distance Measures:
  - Euclidian
  - Manhattan
  - Cosine
- Partitions n observations into k clusters.
- Choosing k

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

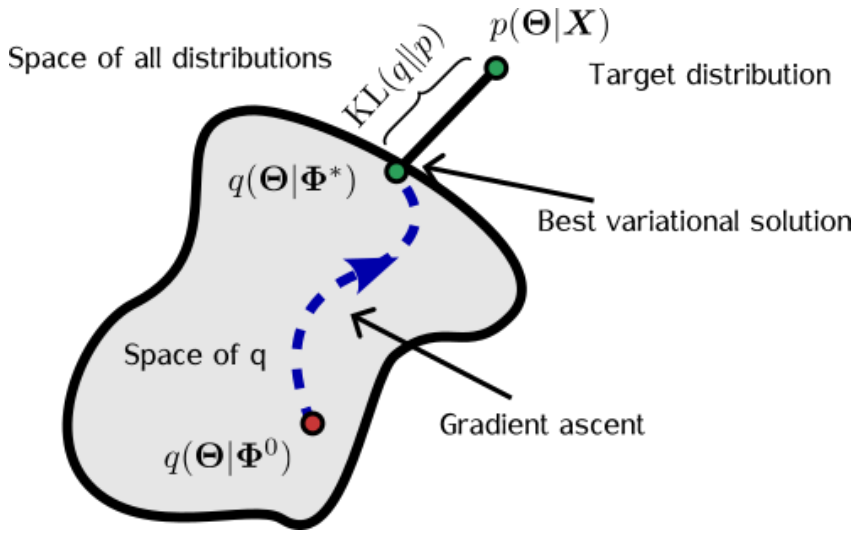


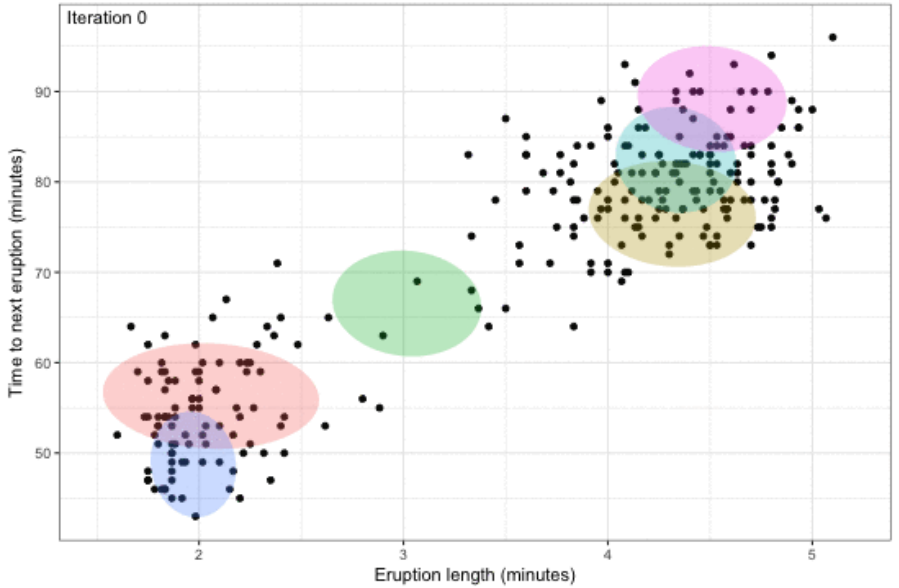
For each value of k. The value of k, which has the largest change in amount of WSS, is taken as the optimum value.



Variational Inference

- A OP technique to learn probability distributions (which are often nasty or not expressible in closed form).
- ***Example:*** Suppose $Z \sim \mathcal{N}(\mu, \sigma^2)$ and $Y \sim \text{Pois}(e^z)$ How can we infer μ, σ given only Y?
- ***Approach:*** Approximate your target distribution using a simpler but more accurate distribution.
- ***Measure of dissimilarity***: KL divergence. This reduces the sampling problem to an optimization problem!





Food for thought

Q. Which of the following techniques would perform better for reducing the dimensions of a data set?

A. Removing columns that have too many missing values

B. Removing columns that have high variance in data

C. Removing columns with dissimilar data trends

D. None of these

Food for thought

Q. Which of the following techniques would perform better for reducing the dimensions of a data set?

A. Removing columns that have too many missing values

B. Removing columns that have high variance in data

C. Removing columns with dissimilar data trends

D. None of these

Q. In the context of image compression using SVD, which factor primarily contributes to the reduction in storage space?

A. Reduction in the number of singular vectors retained

B. Reduction in the number of singular values retained

C. Reduction in the dimensions of the original image

D. Reduction in the rank of the decomposed matrices

Food for thought

Q. Which of the following techniques would perform better for reducing the dimensions of a data set?

A. Removing columns that have too many missing values
B. Removing columns that have high variance in data
C. Removing columns with dissimilar data trends
D. None of these

Q. In the context of image compression using SVD, which factor primarily contributes to the reduction in storage space?

A. Reduction in the number of singular vectors retained
B. Reduction in the number of singular values retained
C. Reduction in the dimensions of the original image
D. Reduction in the rank of the decomposed matrices

Q. In the context of data compression using SVD, which factor primarily affects the quality of the reconstructed data?

A. The number of singular vectors retained
B. The magnitude of the singular values retained
C. The dimensions of the original data
D. Reduction in the rank of the decomposed matrices

Food for thought

Q. Which of the following techniques would perform better for reducing the dimensions of a data set?

A. Removing columns that have too many missing values
B. Removing columns that have high variance in data
C. Removing columns with dissimilar data trends
D. None of these

Q. In the context of image compression using SVD, which factor primarily contributes to the reduction in storage space?

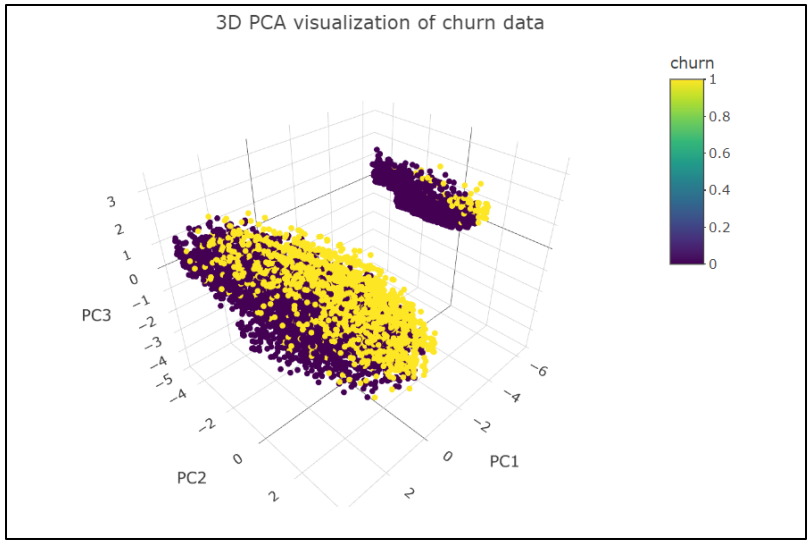
A. Reduction in the number of singular vectors retained
B. Reduction in the number of singular values retained
C. Reduction in the dimensions of the original image
D. Reduction in the rank of the decomposed matrices

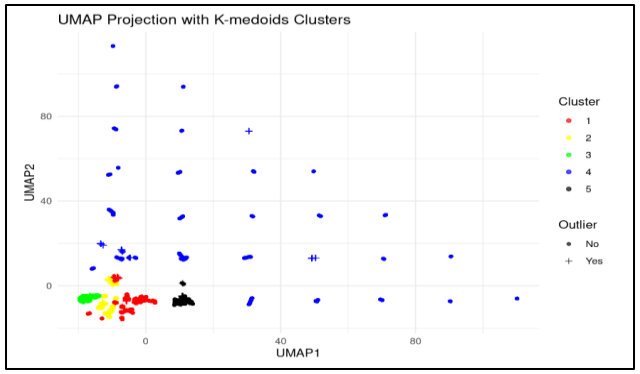
Q. In the context of data compression using SVD, which factor primarily affects the quality of the reconstructed data?

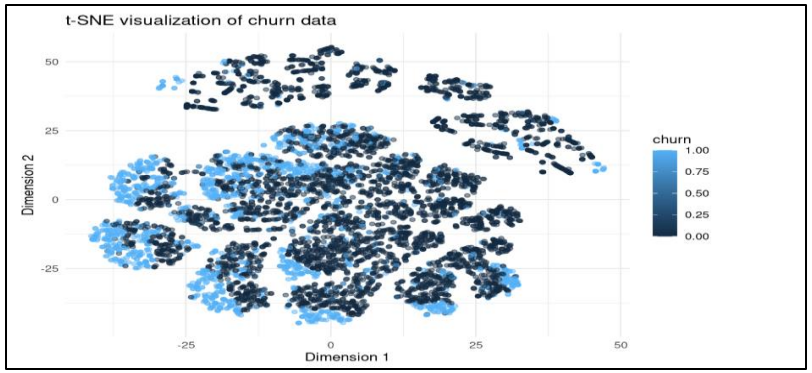
A. The number of singular vectors retained
B. The magnitude of the singular values retained
C. The dimensions of the original data
D. Reduction in the rank of the decomposed matrices

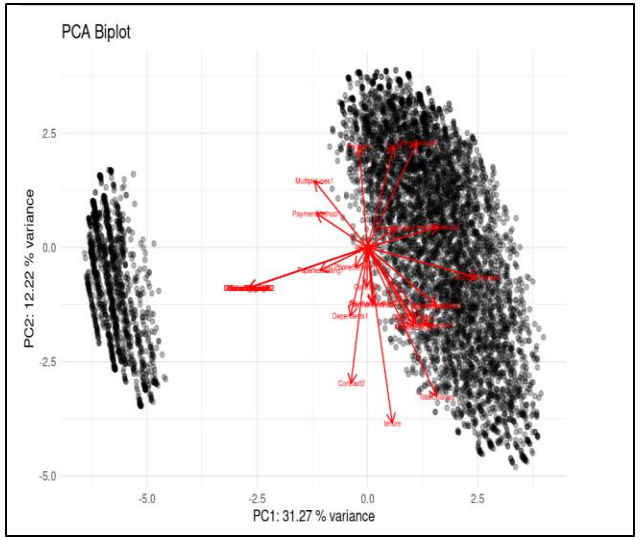
Customer Churn Prediction (Demo Project)

Repo:
https://github.com/Ananyapam7/Customer-Churn-Prediction?tab=readme-ov-file
Feel free to star it ;)









R HTML Page:
https://students.iiserkol.ac.in/~ad18ms075/notebooks/notebook-eda.html

Thank you!

❖ References:

- A nice explanation about PCA and SVD: https://intoli.com/blog/pca-and-svd/
- A go to LA spot: 3b1b Essence of Linear Algebra
- Relationship between SVD and PCA: https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca?noredirect=1&lq=1
- *Machine Learning and Pattern Recognition* - CM Bishop
- *Introduction to Statistical Learning* - James, D. Witten, T. Hastie, R. Tibshirani
- *Computer Age Statistical Inference* – T. Hastie

In case you want to discuss further, feel free to connect me on any of these platforms :)

ananyapam7@gmail.com

ananyapam7.github.io

https://github.com/Ananyapam7

https://www.linkedin.com/in/ananyapam-de

This is me :-)