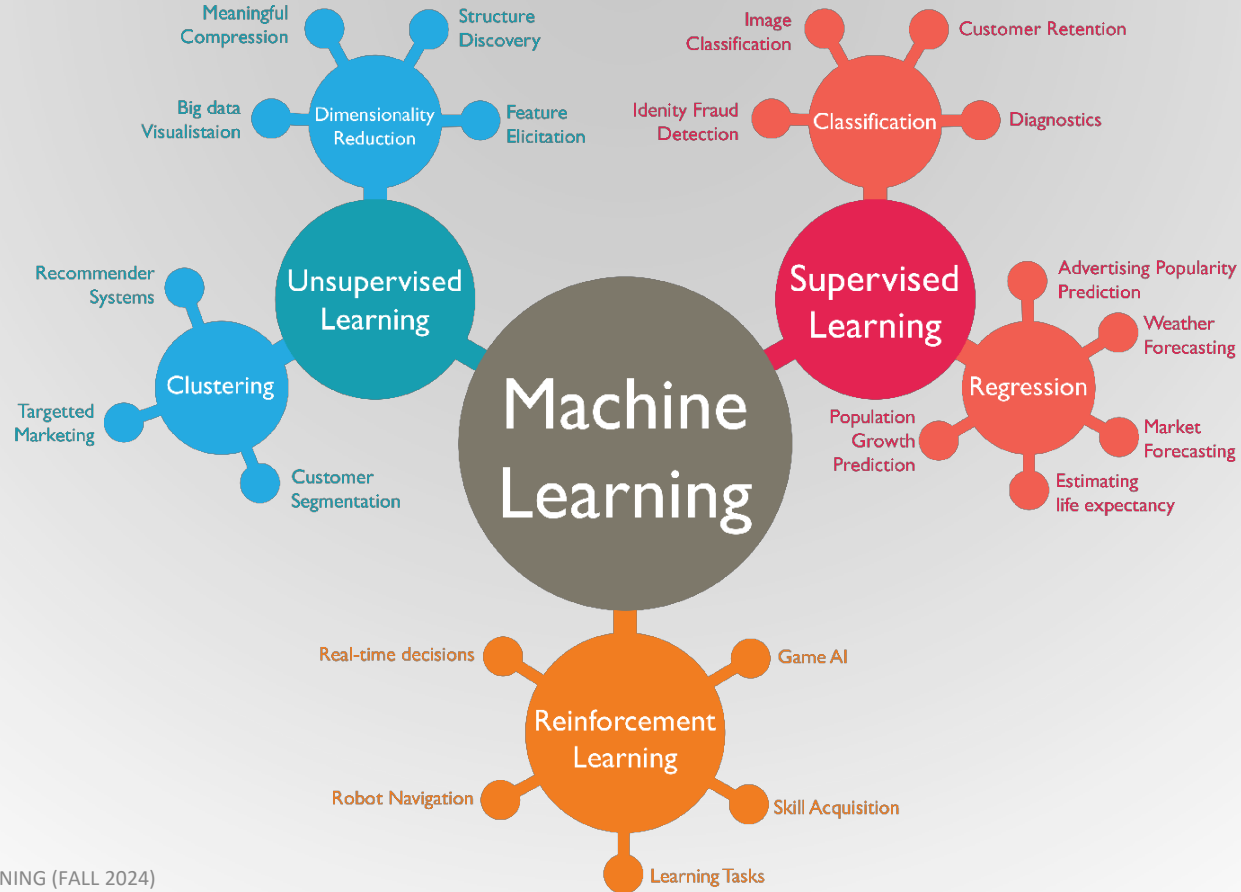


A decorative graphic on the left side of the slide, consisting of a network of thin, dark blue lines that branch out and connect to small, empty circles, resembling a circuit board or a neural network diagram.

9.3 Dimensionality Reduction

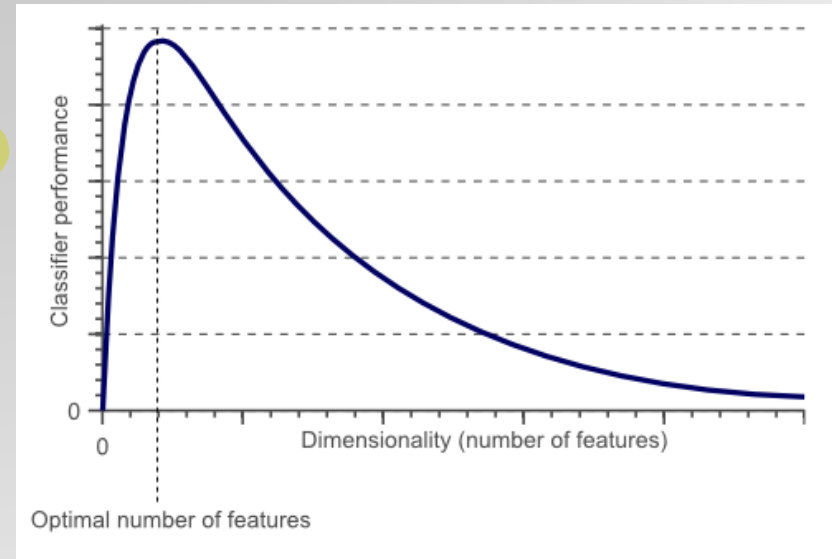
Dr. Sultan Alfarhood

Machine Learning Approaches



Hughes Phenomenon

- According to Hughes phenomenon, If the number of **training samples is fixed** and we keep on **increasing the number of dimensions** then the predictive power of our machine learning model first increases, but after a certain point it tends to decrease.
- Also known as the **curse of dimensionality**



Dimensionality Reduction

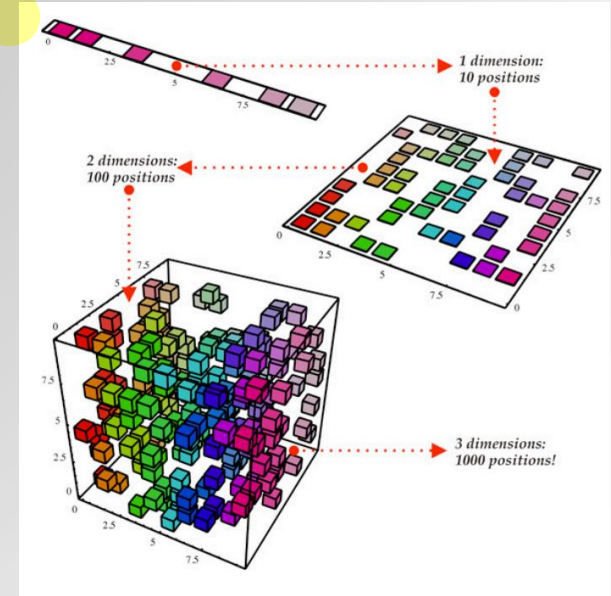
Dimensionality reduction, or dimension reduction, is the transformation of data **from a high-dimensional space into a low-dimensional space** so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension.

- **Applications**

1. The most frequent use case for dimensionality reduction is **data visualization**
 - humans can only interpret on a plot the maximum of three dimensions.
2. Dimensionality reduction **removes redundant** or highly correlated features;
3. It also **reduces the noise** in the data

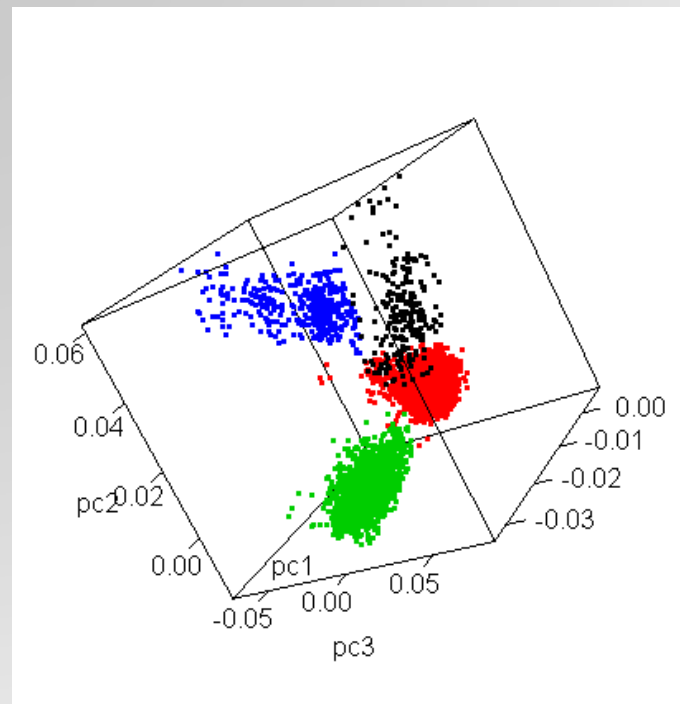
- **Widely used techniques** of dimensionality reduction

- Principal Component Analysis (PCA)
- Uniform Manifold Approximation and Projection (UMAP)
- Autoencoders



Principal Components Analysis (PCA)

- Principal Components Analysis (PCA) is a technique that finds underlying variables (known as principal components) that best differentiate your data points.
- Principal components are **dimensions** along which your **data points are most spread out**.



Applications of PCA

- Visualize multidimensional data.
- Reduce the number of dimensions in data.
- Help resize an image.
- Used in finance to analyze stock data and forecast returns.
- Find patterns in the high-dimensional datasets.



تحسين الاداء

How PCA Works?

1. Normalize the data

- Standardize the data before performing PCA.
- This will ensure that each feature has a mean = 0 and variance = 1.

$$Z = \frac{x - \mu}{\sigma}$$

2. Build the **covariance matrix**

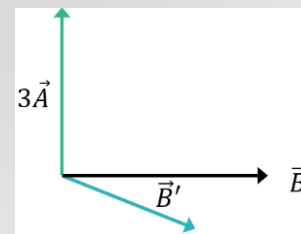
- Construct a square matrix to express the correlation between two or more features in a multidimensional dataset.



Var(x)	Cov(x,y)	...	Cov(x,m)
Cov(x,y)			
Cov(z,x)			
...			
Cov(n,x)	Var(n)

3. Find the **Eigenvectors** and Eigenvalues

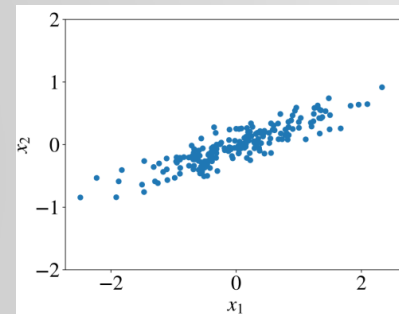
- Calculate the eigenvectors/unit vectors and eigenvalues.
- Eigenvalues are scalars by which we multiply the eigenvector of the covariance matrix.



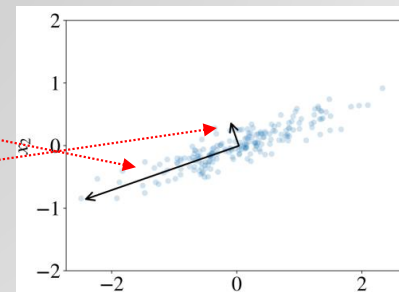
4. Sort the **eigenvectors** in highest to lowest order and select the number of principal components.

PCA Example

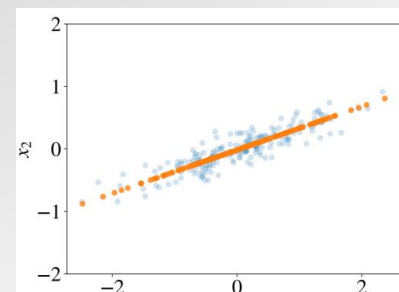
- Consider a two-dimensional data as shown in figure on the right:
 - Principal components are vectors that define a new coordinate system in which the **first axis** goes in the direction of the **highest variance** in the data.
 - The second axis is **orthogonal** to the first one and goes in the direction of the second highest variance in the data.
- If our data was three-dimensional:
 - The third axis would be orthogonal to both the first and the second axes and go in the direction of the third highest variance
- And so on for high dimensional data.



Original data

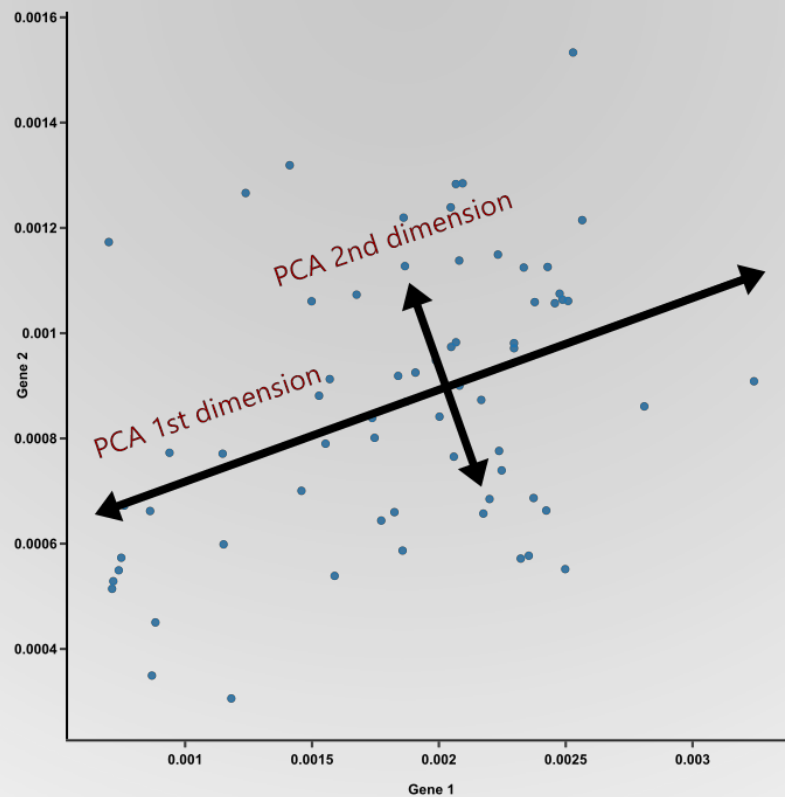


2 principal components displayed as vectors

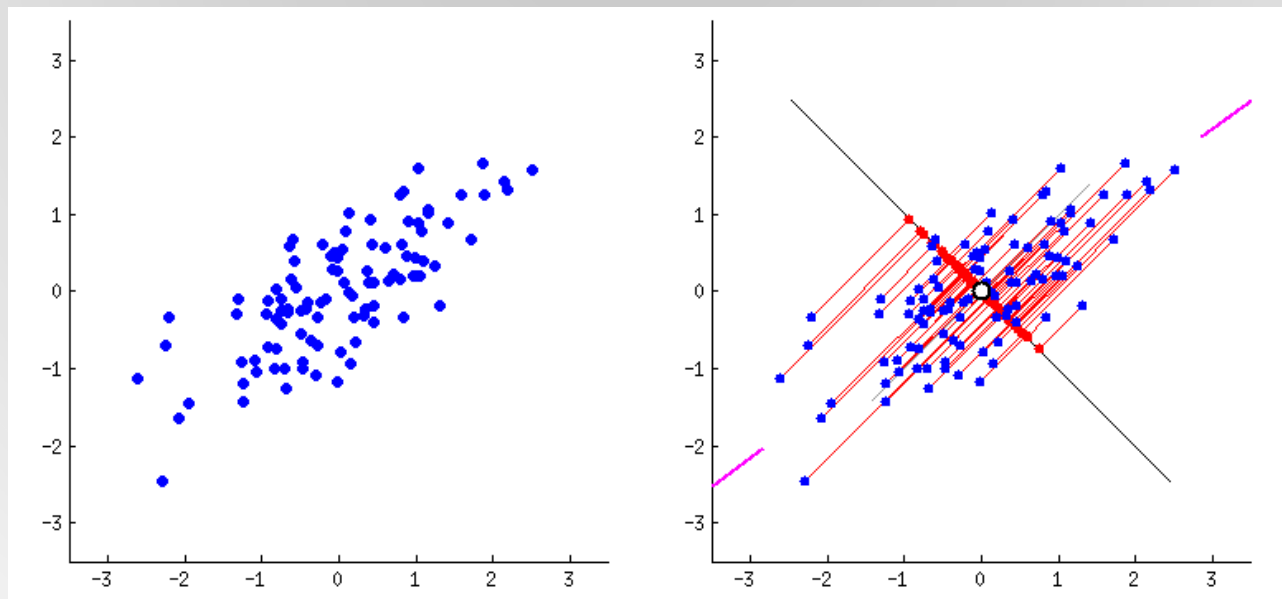


The data projected on the first principal component

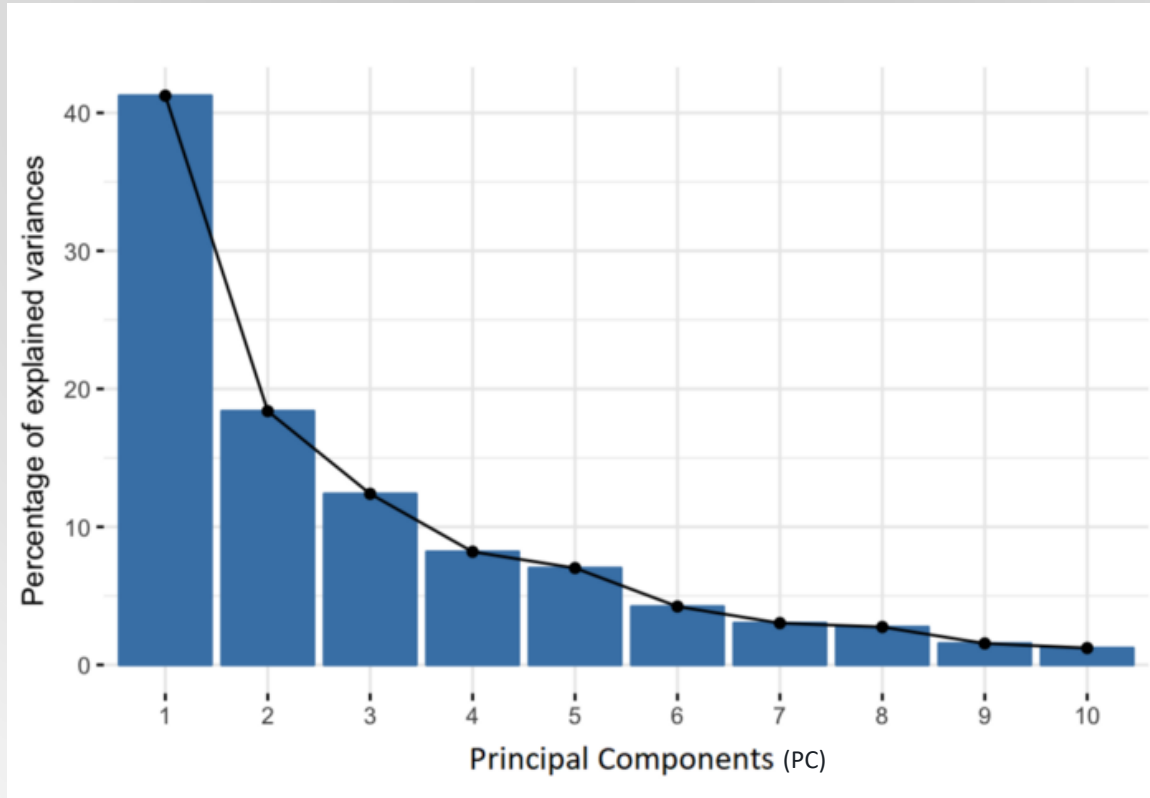
PCA Example



Example



Percentage of Variance (Information) for each by PC





Python Example

<https://colab.research.google.com/drive/1eIddjy28cpp26pyM8qmJqdCABzG0VbBF?usp=sharing>