Data Engineering

Unit 4

Introduction

Data engineering is a critical discipline within the data ecosystem, focusing on the design, construction, and management of systems that facilitate the collection, storage, transformation, and analysis of data.

As organizations increasingly rely on data to drive decision-making and innovation, the role of data engineering has become more significant.

Data engineering involves a series of processes aimed at making raw data usable for analysis.

It encompasses the creation of data pipelines that gather data from multiple sources, cleans and transforms it into a usable format, and stores it in databases or data warehouses for easy access by data scientists and analysts.

Unlike data scientists who focus on analyzing and interpreting data, data engineers ensure that the data is structured and accessible, enabling effective analysis.

Data Pipeline

Data pipelines are essential components of modern data engineering, enabling organizations to efficiently collect, process, and analyze data from various sources.

The design and monitoring of these pipelines are critical for ensuring data integrity, performance, and scalability.

Data Pipeline

A data pipeline is a series of data processing steps that involve extracting data from source systems, transforming it into a suitable format, and loading it into a target destination for analysis or storage.

The primary goal of a data pipeline is to automate the flow of data between systems while ensuring that it remains accurate and accessible.

Design and Monitoring

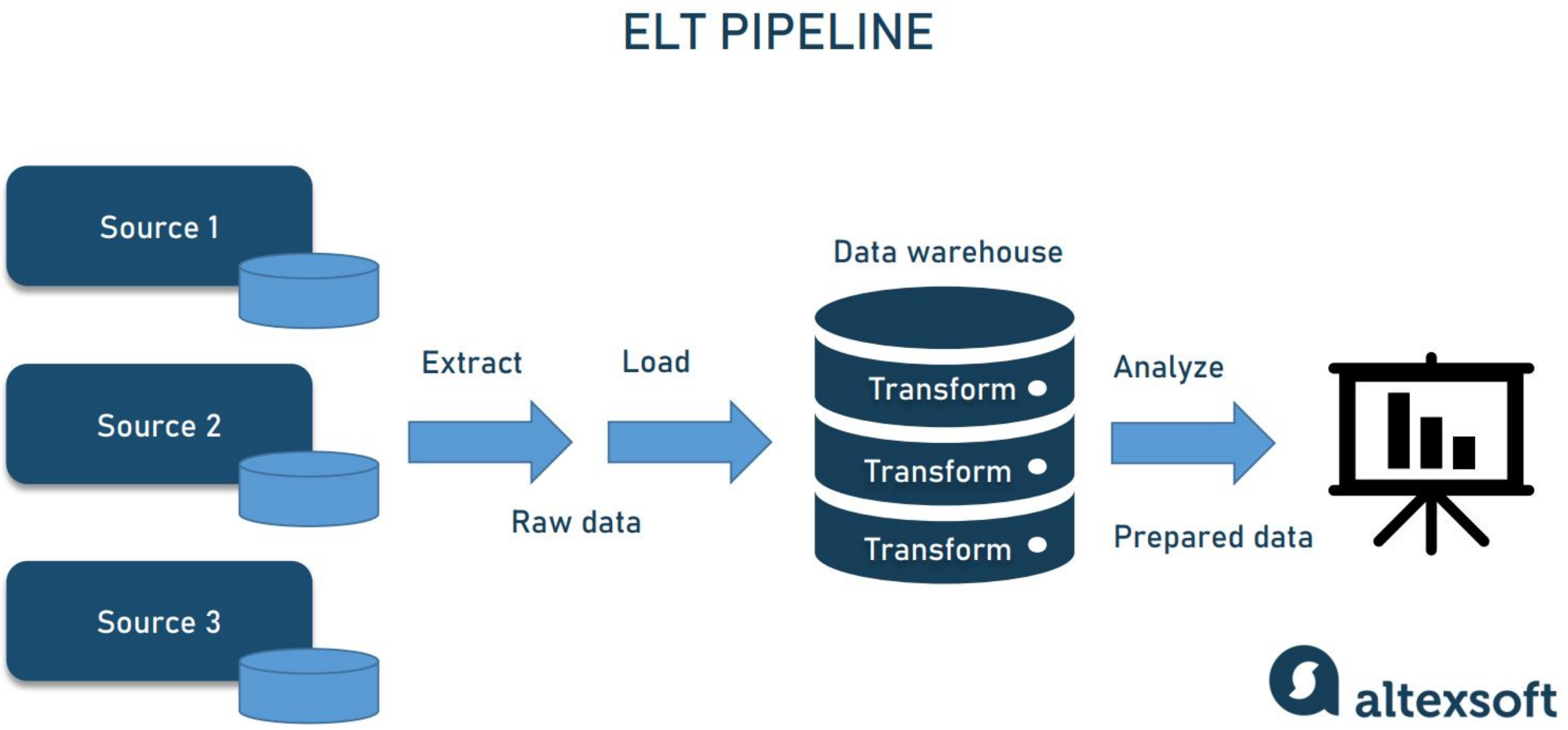
Effective monitoring is crucial for maintaining the reliability and performance of data pipelines:

Real-Time Monitoring

Automated Alerts

Performance Metrics etc.

ETL


ELT PIPELINE
Source 1
Source 2
Source 3
Extract
Load
Raw data
Data warehouse
Transform
Transform
Transform
Analyze
Prepared data
altexsoft

Data engineering is not confined to a single industry; its principles apply across various sectors:

Healthcare: In healthcare, accurate patient records are crucial for effective treatment. Data engineers design systems that integrate patient information from different sources while maintaining compliance with regulations such as HIPAA.

Finance: Financial institutions rely on real-time analytics for risk management and fraud detection. Data engineers build pipelines that process transaction data quickly to identify suspicious activities as they occur.

Retail: Retailers use customer behavior analytics to optimize inventory management and personalize marketing strategies. Data engineers help create systems that analyze purchasing patterns by integrating sales data from multiple channels.

Feature Engineering

Feature engineering is a crucial process in data science and machine learning that involves transforming raw data into a format suitable for modeling.

This process enhances the performance of machine learning models by creating new features or modifying existing ones to provide more relevant information.

Feature Engineering

Feature Creation: This involves generating new features from existing ones. For instance, combining multiple features into a single one or deriving new metrics (e.g., calculating Body Mass Index (BMI) from height and weight) can provide additional insights.

Feature Transformation: Transforming features can include scaling (normalization or standardization), encoding categorical variables (one-hot encoding), or applying mathematical transformations (logarithmic or polynomial transformations) to make relationships more linear.

Feature Extraction: This technique involves reducing dimensionality by extracting important features from raw data while discarding irrelevant ones. Methods like Principal Component Analysis (PCA) are commonly used for this purpose.

Feature Selection: Selecting a subset of relevant features from a larger set helps improve model efficiency and performance.

Correlation quantifies the strength and direction of a relationship between two variables. It is typically represented by a correlation coefficient, denoted as r, which ranges from -1 to +1

- Positive Correlation ($r>0$): Indicates that as one variable increases, the other variable also tends to increase. For example, there may be a positive correlation between hours studied and exam scores.
- Negative Correlation ($r<0$): Suggests that as one variable increases, the other variable tends to decrease. An example could be the relationship between temperature and heating costs; as temperatures rise, heating costs typically fall.
- No Correlation ($r=0$): Implies that there is no discernible relationship between the variables.

Dimensionality Reduction

Dimensionality Reduction is a crucial step in data preprocessing for machine learning and data analysis.

It involves reducing the number of features or dimensions in a dataset while preserving as much information as possible.

This process helps in improving model performance, reducing computational complexity, and enhancing data visualization.

Why Dimensionality Reduction

High-dimensional data can lead to overfitting, where models become too specialized to the training data. By reducing dimensions, you can decrease the risk of overfitting.

Lower-dimensional data can be processed more efficiently, leading to faster training times and better model performance.

Reducing dimensions makes it easier to visualize data, which is essential for understanding complex relationships and patterns.

Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction while preserving as much variance as possible in high-dimensional datasets.

Introduced by mathematician Karl Pearson in 1901, PCA transforms correlated variables into a set of uncorrelated variables called principal components.

These components capture the maximum variance in the data, allowing for simplified analysis and visualization.

Purpose of Principal Component Analysis

PCA is widely used for several purposes, including:

- Dimensionality Reduction: Reducing the number of variables in a dataset while retaining the most important information.
- Data Visualization: Enabling the visualization of high-dimensional data in two or three dimensions.
- Feature Selection: Identifying the most significant variables in a dataset, which is particularly useful in machine learning.
- Noise Reduction: Eliminating noise by removing components with low variance that do not contribute significantly to the data structure.
- Multicollinearity Handling: Addressing issues where independent variables are highly correlated, which can affect regression analysis.

Algorithm

1. Standardize the dataset to have a mean of 0 and a standard deviation of 1 for each feature. This ensures that PCA is not biased by the scale of the variables.

2. Compute the covariance matrix to understand how variables vary together.

3. Calculate the eigenvalues and eigenvectors of the covariance matrix. The eigenvectors represent the directions of maximum variance (principal components), while the eigenvalues indicate their magnitude.

4. Sort the eigenvalues in descending order and arrange their corresponding eigenvectors accordingly. The top k eigenvalues will determine how many principal components to keep.

5. Project the original standardized data onto the selected principal components to obtain a lower-dimensional representation.

Given the table reduce the dimension from 2 to 1 using PCA algorithm

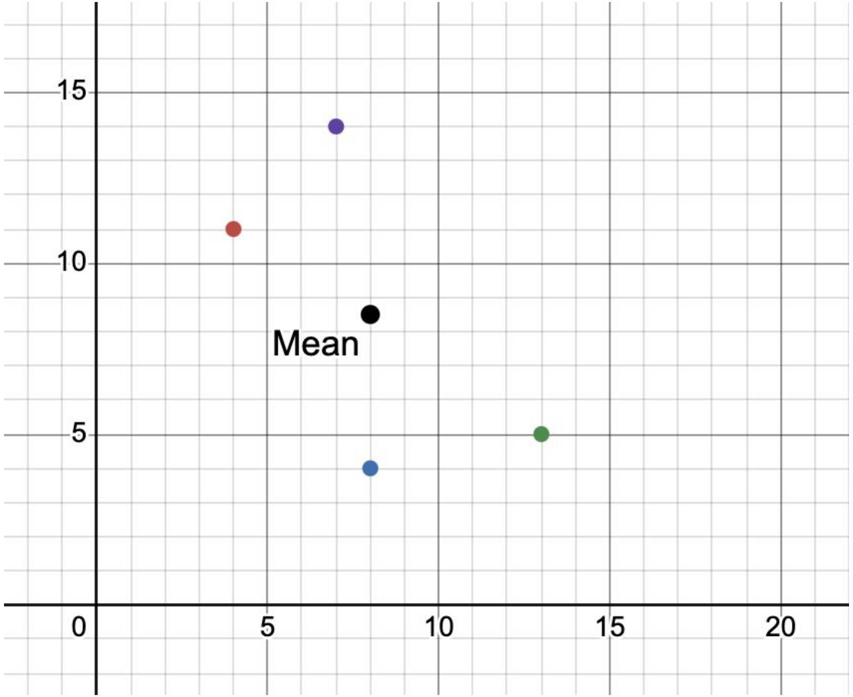
X1	X2
4	11
8	4
13	5
7	14

Step1: Calculate Mean value

Mean of X1 = (4+8+13+7)/4 = 8

Mean of X2 = (11 + 4 + 5 + 14) /4 = 8.5

Given the table reduce the dimension from 2 to 1 using PCA algorithm

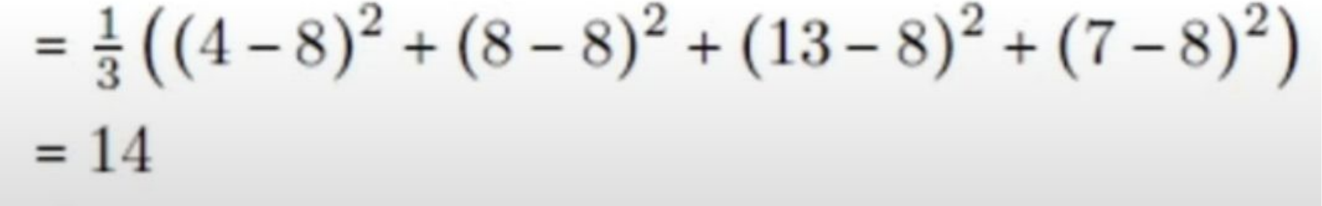


Given the table reduce the dimension from 2 to 1 using PCA algorithm

Step 2: Calculation of the covariance matrix

$$S = \begin{bmatrix} Cov(x1, x1) & Cov(x1, x2) \\ Cov(x2, x1) & Cov(x2, x2) \end{bmatrix}$$

$$Cov(x1, x1) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{1k} - \overline{X_1})(X_{1k} - \overline{X_1})$$



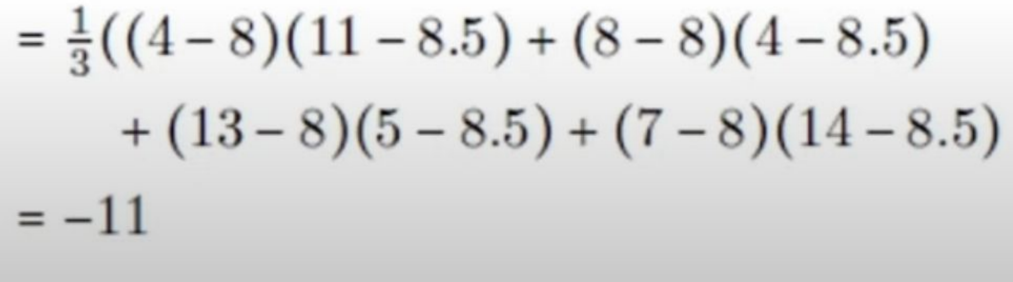
$$= \frac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right)$$

$$= 14$$

Given the table reduce the dimension from 2 to 1 using PCA algorithm

Step 2: Calculation of the covariance matrix

$$Cov(x1, x2) = \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \overline{X_1})(X_{2k} - \overline{X_2})$$

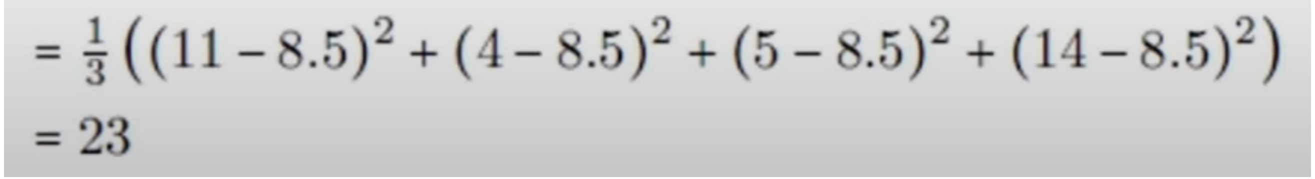


$$= \frac{1}{3}((4-8)(11-8.5)+(8-8)(4-8.5)$$
$$+(13-8)(5-8.5)+(7-8)(14-8.5)$$
$$= -11$$

Given the table reduce the dimension from 2 to 1 using PCA algorithm

Step 2: Calculation of the covariance matrix

$$Cov(x2, x2) = \frac{1}{N-1}\sum_{k=1}^{N}(X_{2k} - \overline{X_2})(X_{2k} - \overline{X_2})$$



$$= \frac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right)$$

$$= 23$$

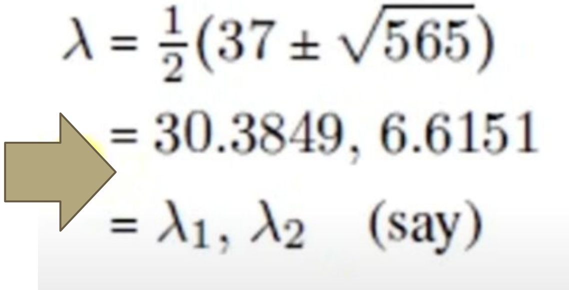
$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 3: Calculate the eigen value of the covariance matrix
The characteristics equation of the covariance matrix is

$$0 = |S - \lambda I|$$

$$0 = \begin{vmatrix} 14 - \lambda & -11 \\ -11 & 23 - \lambda \end{vmatrix}$$

$$= \lambda^2 - 37\lambda + 201$$



$$\lambda = \frac{1}{2}(37 \pm \sqrt{565})$$

$$= 30.3849,\ 6.6151$$

$$= \lambda_1,\ \lambda_2 \quad \text{(say)}$$

Step 4: Computation of the eigen vectors

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$(S - \lambda I)U = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 14 - \lambda I & -11 \\ -11 & 23 - \lambda I \end{bmatrix} \begin{bmatrix} u1 \\ u2 \end{bmatrix}$$

$$= \begin{bmatrix} (14 - \lambda)u_1 - 11u_2 \\ -11u_1 + (23 - \lambda)u_2 \end{bmatrix}$$

$$(14 - \lambda)u_1 - 11u_2 = 0$$

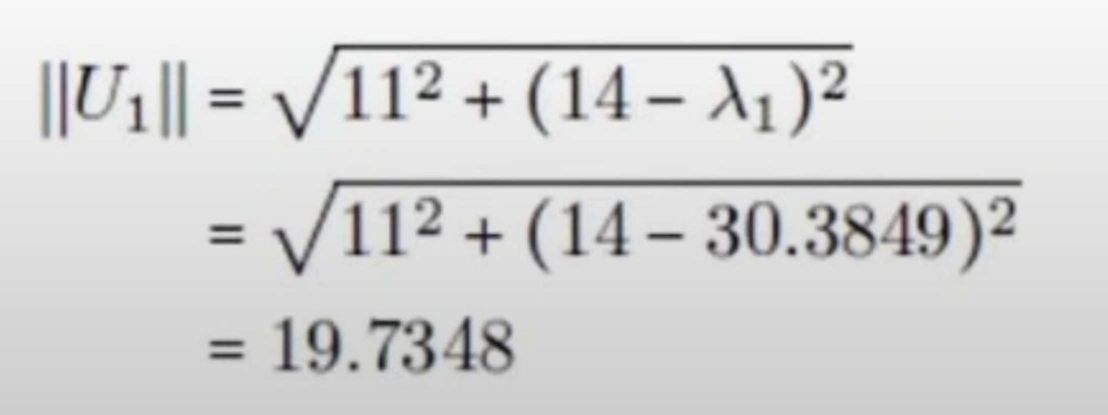
$$-11u_1 + (23 - \lambda)u_2 = 0$$

$$\frac{u_1}{11} = \frac{u_2}{14 - \lambda} = t(say)$$

$$u_1 = 11t, u_2 = (14 - \lambda)t$$

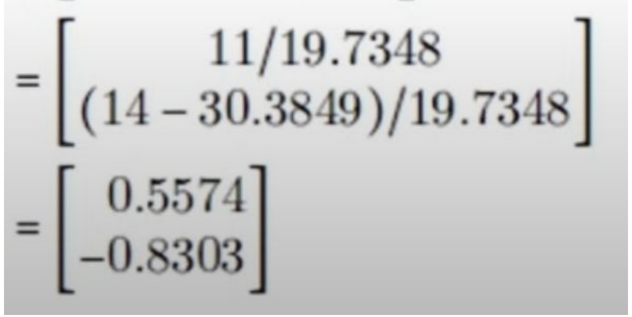
$$U_1 = \begin{bmatrix} 11 \\ 14 - \lambda \end{bmatrix}$$

To find the unit eigen vector, we compute the length of U1



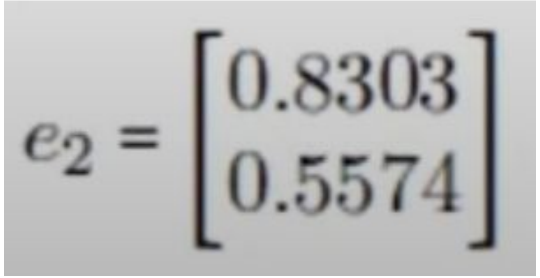
$$\begin{aligned} \|U_1\| &= \sqrt{11^2 + (14 - \lambda_1)^2} \\ &= \sqrt{11^2 + (14 - 30.3849)^2} \\ &= 19.7348 \end{aligned}$$

$$e_1 = \begin{bmatrix} 11/\left\|U_1\right\| \\ (14-\lambda)/\left\|U_1\right\| \end{bmatrix}$$



$$= \begin{bmatrix} 11/19.7348 \\ (14-30.3849)/19.7348 \end{bmatrix}$$

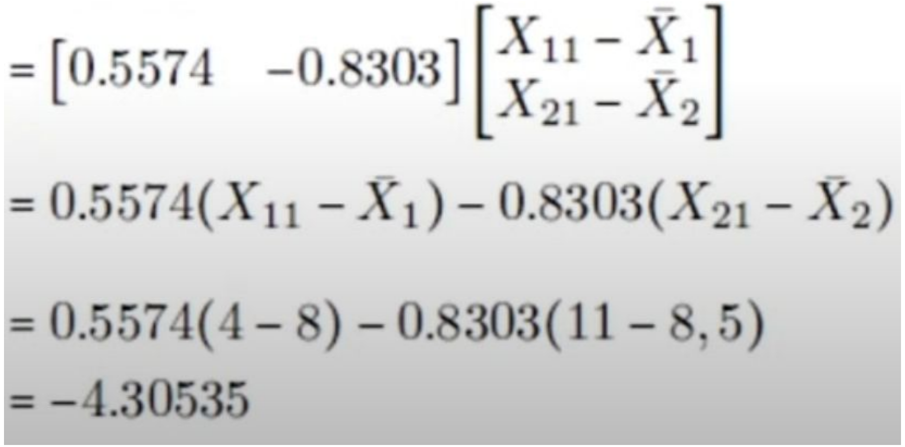
$$= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$



$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Step 5: Computation of first principal Components

$$e_1^T \begin{bmatrix} X_{1k} - \overline{X_1} \\ X_{2k} - \overline{X_2} \end{bmatrix}$$



$$= \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix}$$

$$= 0.5574(X_{11} - \bar{X}_1) - 0.8303(X_{21} - \bar{X}_2)$$

$$= 0.5574(4 - 8) - 0.8303(11 - 8,5)$$

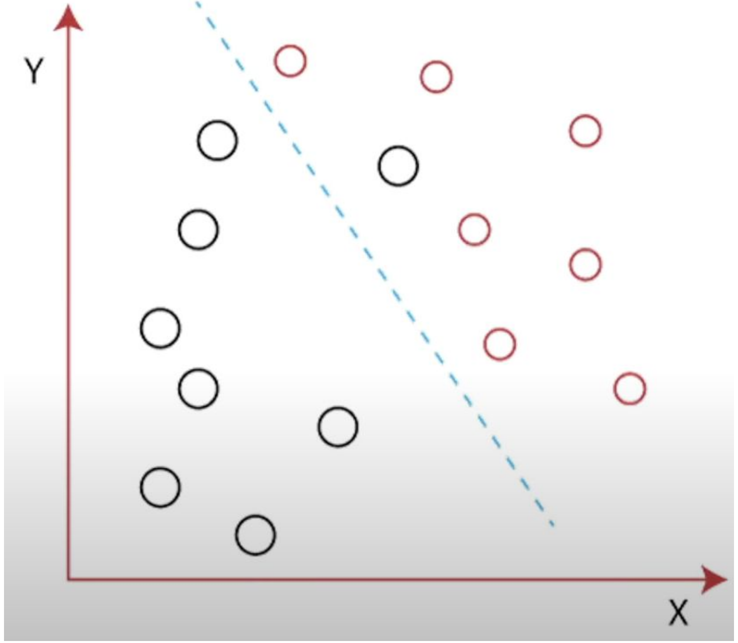
$$= -4.30535$$

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) or Normal Discriminant Analysis is also a dimensionality reduction technique used in machine learning and pattern recognition for supervised classification tasks.

It is used to project the features in higher dimension data into lower dimension.

Suppose we have two sets of data points belonging to two different classes that we want to classify. When the data points are plotted on the 2D plane, there’s no straight line that can seperate the two classes of the data points completely

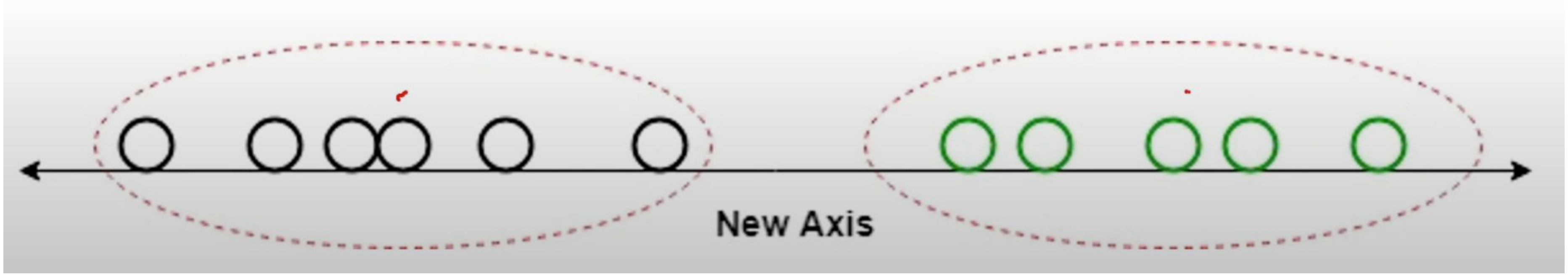


Linear Discriminant Analysis (LDA)

- Two criteria are used by LDA to create a new axis:
  - Maximize the distance between means of the two classes.
  - Minimize the variation within each class.

Linear Discriminant Analysis (LDA)

- Two criteria are used by LDA to create a new axis:
  - Maximize the distance between means of the two classes.
  - Minimize the variation within each class.



Linear Discriminant Analysis (LDA)

1. Compute the class means of dependent variable

$$\mu_{1} = \frac{1}{N_1}\sum_{x\in\omega_1} x$$

2. Derive the covariance matrix of the class variable.

$$S_1 = \sum_{x \epsilon w_1} (x-\mu_1)(x-\mu_1)^T$$

3. Compute the within class - scatter matrix (S1+S2)

$$S_w = S1 + S2$$

4. Compute the between class scatter matrix

$$S_B = (\mu_1-\mu_2)(\mu_1-\mu_2)^T$$

5. Compute the eigen values and eigen vectors from the within class and between class scatter matrix

$$\left|S_w^{-1}.\, S_B - \lambda I\right| = 0$$

Linear Discriminant Analysis (LDA)

6. Sort the values of eigen values and select the top k values

7. Find the eigen vectors that corresponds to the top k eigen values

$$(S_w^{-1} S_B w - \lambda I) \begin{pmatrix} w1 \\ w2 \end{pmatrix} = 0$$

8. Obtain the LDA by taking the dot product of eigen vectors and original data

Compute the LDA for the following two dimensional dataset

Samples for class $\boldsymbol{\omega}_1$: $\mathbf{X}_1=(\mathrm{x}_1,\mathrm{x}_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$

Sample for class $\boldsymbol{\omega}_2$: $\mathbf{X}_2=(\mathrm{x}_1,\mathrm{x}_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$

$$\mu_1=\frac{1}{N_1}\sum_{x\in\omega_1}x=\frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right]=\begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2=\frac{1}{N_2}\sum_{x\in\omega_2}x=\frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right]=\begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Solve yourself?? Using the above steps