Unit 3. Unsupervised Learning (10 Hrs.)

Objective:

- ✓ Design and implement unsupervised learning algorithms to solve real world problems.

3.1. Basic Concept of Unsupervised Learning

Unsupervised learning is a type of machine learning where algorithms analyze and cluster data that has not been labeled, categorized, or tagged (i.e. unsupervised learning deals with unlabeled data).

- The goal is for the algorithm to discover or identify hidden patterns and structures, or distributions within the dataset without prior knowledge of the output, enabling it to identify similarities and differences on its own.

Unsupervised learning is well suited for tasks that require exploring large amounts of unlabeled data.

- This approach makes it easier for businesses to gain insights from data when no labels are present, helping them to understand the underlying structure of a dataset and identify patterns and relationships between datasets without the need for a human to teach them.

Example:

Imagine you're at a party with a group of strangers. Without any introductions, you notice people naturally forming groups based on their behaviors - some are at the dance floor, some are at the food table, and others are engaged in quiet conversations. By observing these patterns, you're clustering people into groups without knowing anything about them beforehand.

Key Concepts and Terminology

- **Clustering**: Grouping data points into clusters where points in the same cluster are more similar to each other than to those in other clusters.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

- **Association**: Finding rules that describe relationships between variables in a dataset.
- **Dimensionality Reduction**: Reducing the number of variables under consideration, simplifying the dataset while retaining essential information.
- **Centroid**: The center of a cluster in clustering algorithms like K-Means.
- **Distance Metrics**: Methods to measure the similarity or dissimilarity between data points, such as Euclidean distance or Manhattan distance.

Area of Application of Unsupervised Learning:

Here are some real-world unsupervised learning examples:

- **Anomaly detection:** Unsupervised clustering can process large datasets and discover data points that are atypical in a dataset.
- **Recommendation engines**: Using association rules, unsupervised machine learning can help explore transactional data to discover patterns or trends that can be used to drive personalized recommendations for online retailers.
- **Customer segmentation:** To generate buyer persona profiles by clustering customers' common traits or purchasing behaviors.
- **Fraud detection:** for anomaly detection, revealing unusual data points in datasets.
- **Natural language processing (NLP):** for various NLP applications, such as categorizing articles in news sections, text translation and classification, or speech recognition in conversational interfaces.
- **Genetic research:** for Genetic clustering. Hierarchical clustering algorithms are often used to analyze DNA patterns and reveal evolutionary relationships.

3.2. Types of Unsupervised Learning

In general, there are three types of unsupervised learning tasks:

1. Clustering
2. Association rules
3. Dimensionality reduction

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

3.2.3. Association Rule Learning

Association Rule Learning is an example of Unsupervised Learning, and the goal is to dig into large amounts of data and discover interesting relations between attributes / variables in a given dataset.

- It is most commonly used to analyze retail baskets / market basket or transactional datasets to represent or analyze how often certain items are purchased together, allowing companies to better understand relationships between different products.
- Understanding consumption habits of customers, enables businesses to develop better cross-selling strategies and recommendation engines.
- Unsupervised learning algorithms search for frequent if-then associations, also called rules to discover correlations and co-occurrences within the data and the different connections between data objects.
- **For Example:** Amazon's "Customers Who Bought This Item Also Bought" or Spotify's "Discover Weekly" playlist ae the example of Association Rule Learning.
- Different algorithms such as
    - Apriori and FP-Growth (transactional data)
    - Quantitative Association Rules, Eclat (numerical data),
    - Apriori, FP-Growth with categorical encoding (Categorical and Ordinal data)
    - Sequential Pattern Mining, PrefixSpan (Time Series Data)
    - Spatial Association Rules, Co-location Pattern Mining (Spatial data)
    - Graph Association Rules, Subgraph Mining (Graph Data)
    - Frequent Pattern Mining with text preprocessing, Association Rules on n-grams (Text Data)
    - Pattern Discovery on Spectral Features, Sequential Pattern Mining on audio signals (Audio Data)
    - Frequent Pattern Mining on Image Features, Visual Co-occurrence Mining (Image Data)

are used to generate association rules and the Apriori algorithm is most widely used.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Why to use Association Rule Learning?

Suppose: you are the owner of a Supermarket and you want people to buy your products easily. What you can do is, you can run this algorithm on your sales log and find interesting relations between the items. For example, you find out that people who purchase milk and bread, also tend to purchase butter. Thus, you may want to do the following things to improve the quality of your mart:

- ✓ You can place milk, bread and butter on the same shelf, so that buyers of one item would be prompted to buy another item.
- ✓ You can put milk, bread and butter on discount to increase your item sales.
- ✓ You can also target the buyers of milk or bread with the advertisement of butter.
- ✓ Or you can also combine bread and butter into a whole new product i.e. buttery bread with slightly milky flavor and then put it on sale.

This algorithm counts the frequency of complimentary occurrences, or associations, across a very large dataset with over thousands of attributes. The goal is to find associations that take place together far more often than you would find in a random sampling of possibilities.

Metrics to measure Association:

1. **Support:** To find, how popular an itemset is, i.e. it is used to find the frequency of a certain itemset appearing in the dataset.

   Support (A) = Frequency (A)

2. **Confidence:** To find, how likely item B is purchased when item A is purchased, expressed as (A -> B).

$$Confidence(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(A)}$$

3. **Lift:** To find, how likely an item A is purchased while controlling how popular item B is.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)}$$

If Lift = 1 → The probability of occurrence of A and B is independent of each other.

If Lift > 1 →It determines the degree to which A and B are dependent on each other.

If Lift < 1→ It tells us that A is a substitute for B, which means A has a negative effect on item B.

Most Common Algorithms for Association Rule Learning

- **Apriori**
  - This algorithm uses frequent datasets to generate association rules.
  - We apply an iterative approach or level-wise search where k-frequent item sets are used to find k+1 item sets.
  - This algorithm uses a Breadth-First Search algorithm and Hash-Tree to calculate the itemset efficiently.
  - Example uses is music recommendation in spotify
- **Eclat**
  - Stands for Equivalence Class Transformation.
  - While the Apriori algorithm works in a horizontal sense imitating the Breadth-First Search of a graph, the ECLAT algorithm works in a vertical manner just like the Depth-First Search of a graph.
  - It performs faster execution than Apriori Algorithm.
- **F-P Growth**
  - Stands for Frequent Pattern, and it is the improved version of the Apriori Algorithm.
  - It is an alternative way to find frequent item sets without using candidate generations, thus improving performance.
  - It uses a Divide-and-Conquer strategy and the core of this method is the usage of a special data structure named *Frequent-Pattern Tree (FP-tree)*, which retains the item set association information. The purpose of this frequent tree is to extract the most frequent patterns.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

3.3. Clustering

Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other.

- Clustering is a technique for exploring raw, unlabeled data and breaking it down into groups (or clusters) based on similarities or differences.
- It is used in a variety of applications such as:
    - Customer / market /Image segmentation,
    - Fraud / Anomaly detection
    - Social network analysis
    - Search results grouping

Example:

Retail companies often use clustering to identify groups of households that are similar to each other.

For example, a retail company may collect the following information on households:

- ✓ Household income
- ✓ Household size
- ✓ Head of household Occupation
- ✓ Distance from nearest urban area

These variables could be input for a clustering algorithm to perhaps identify the following clusters:

Cluster 1: Small family, high spenders

Cluster 2: Larger family, high spenders

Cluster 3: Small family, low spenders

Cluster 4: Large family, low spenders

Then the company can send personalized advertisements or sales letters to each household based on how likely they are to respond to specific types of advertisements.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Unsupervised learning algorithms for clustering:

- **Exclusive clustering**
  - Data is grouped in a way where a single data point can only exist in one cluster. This is also referred to as "hard" clustering.
  - A common example of exclusive clustering is the K-means clustering algorithm, which partitions data points into a user-defined number K of clusters.
- **Overlapping clustering:**
  - Data is grouped in a way where a single data point can exist in two or more clusters with different degrees of membership.
  - This is also referred to as "soft" clustering.
- **Hierarchical clustering:**
  - Data is divided into distinct clusters based on similarities, which are then repeatedly merged and organized based on their hierarchical relationships.
  - There are two main types of hierarchical clustering: agglomerative and divisive clustering.
- **Probabilistic clustering:**
  - Data is grouped into clusters based on the probability of each data point belonging to each cluster.
  - This approach differs from the other methods, which group data points based on their similarities to others in a cluster.

3.3.1. K-Means Clustering

K-Means is one of the most popular, straightforward and an iterative algorithm that tries to partition the dataset into K pre-defined, distinct and non-overlapping subgroups or clusters, where each data point belongs to only one group.

- The algorithm tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is minimized.
- The less variation we have within clusters, the more homogeneous the data points are within the same cluster.

Algorithm:

Step-1: Select K random entries from the dataset and use them as centroids.

Step-2: Find the distance of each entry in the dataset from the centroids (Use any distance metric such as Euclidean distance, Manhattan distance, or squared Euclidean distance).

Step-3: Assign each data point to the cluster with the centroid to which it has the least distance

Step-4: Calculate the new centroid of the clusters by using the mean of each data point in the same cluster as the new centroid.

Step-5: If the newly created centroids are the same as the centroids in the previous iteration, consider the current clusters to be final and stop. BUT, If any of the newly created centroids is different from the centroids in the previous iteration, go to step 2 and continue till step 5.

OR, Stopping Criteria are

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster even after training the algorithm for multiple iterations.
3. Maximum number of iterations is reached

Example:

Q-1. Apply K means clustering with following data of customers behavior in online Shopping Site.

S1(20,500), S2(40,1000), S3 (30,800), S4(18,300), S5 (28,1200), S6 (35,1400) and S7(45,1800).

Q-2. Apply K (=2)-Means algorithm over the data (185, 72), (170, 56), (168, 60), (179,68), (182,72), (188,77) up to two iterations and show the clusters. Initially choose first two objects as initial centroids.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Q-3. Apply K means clustering with the data points A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3 (6,4), C1(1,2) and C2(4,9). Assume A1, B1 and C1 as the center of each cluster.

3.3.2. Hierarchical Clustering

Hierarchical clustering is an unsupervised machine learning algorithm that groups data into a tree of nested clusters.

- It helps to find the patterns and connections in datasets. Results are presented in a dendrogram diagram showing the distance relationships between clusters.
- ***Dendrograms*** are the tree-like morphologies of the dataset, in which the X axis of the dendrogram represents the features or columns of the dataset, and the Y axis of the dendrogram represents the Euclidian distance between data observations.

Hierarchical clustering is also known as hierarchical cluster analysis (HCA).

- There are two types of hierarchal clustering:
    1. Agglomerative clustering
    2. Divisive Clustering

3.3.2.1. Agglomerative Clustering

Agglomerative clustering starts with each data point as an individual cluster and iteratively merges the closest pair of clusters until only one cluster remains or until a stopping condition is met (like a desired number of clusters).

- This method is also called **bottom-up** approach because it starts from the bottom (individual data points) and builds up to the top (final cluster).
- Its data points are isolated as separate groupings initially, and then they are merged together iteratively on the basis of similarity until one cluster has been achieved.
- Euclidean distance is the most common metric used to calculate these distances
- Agglomerative methods are more common due to ease of implementation and widespread software support.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Methods used to measure Similarity:

Four different methods are commonly used to measure similarity:

1. Ward's linkage: This method states that the distance between two clusters is defined by the increase in the sum of squared after the clusters are merged.

2. Average linkage: This method is defined by the mean distance between two points in each cluster.

3. Complete (or maximum) linkage: This method is defined by the maximum distance between two points in each cluster.

4. Single (or minimum) linkage: This method is defined by the minimum distance between two points in each cluster.

Algorithm Steps:

1. Compute the proximity matrix using a distance metric.
2. Use a linkage function to group objects into a hierarchical cluster tree based on the computed distance matrix from the above step.
3. Data points with close proximity are merged together to form a cluster.
4. Repeat steps 2 and 3 until a single cluster remains.

Example:

Q-1. Consider the one-dimensional data points 18,22,25,27,42,43 and apply the agglomerative hierarchical clustering to cluster and plot the dendrogram.

Q-2. For the given dataset, find the clusters using a single linkage technique. Use the Euclidean distance and draw the Dendrogram.

Sample	X	Y
P1	0.40	0.53
P2	0.22	0.38

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

3.3.2.2. Divisive Clustering

Divisive clustering starts with all data points in a single cluster and recursively splits them into smaller clusters. The process continues until each data point is in its own individual cluster.

- It is also known as a top-down approach, as it starts at the top (single cluster) and breaks it down into smaller clusters.
- Divisive methods are typically more computationally expensive due to their recursive nature, and accuracy depends greatly on the splitting strategy.
- Divisive hierarchical clustering is not as readily supported in standard Python libraries as agglomerative clustering.
- These clustering processes are usually visualized using a dendrogram.

Algorithm Steps:

1. Split into clusters using any flat-clustering method, say k-means.
2. Choose the best cluster among the clusters to split further, choose the one that has the largest Sum of Squared Error (SSE).
3. Repeat steps 2 and 3 until a single cluster is formed.

Example:

3.3.3. Density-based Clustering

Density-based clustering is based on the idea that clusters are regions of high density separated by regions of low density.

3.3.3.1. DBSCAN

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.

- DBSCAN is one of the most common density-based clustering algorithms.
- DBSCAN algorithm requires two parameters:
    1. The minimum number of neighbors **(minPts)** and
    2. The maximum distance between core data points **(eps).**
- For each cluster, DBSCAN will define three types of data points:
    1. The algorithm works by first identifying "**core" data points**, which are data points that have a minimum number of neighbors within a specified distance. These core data points form the center of a cluster.
    2. Next, the algorithm identifies **"border" data points**, which are data points that are not core data points but have at least one core data point as a neighbor.
    3. Finally, the algorithm identifies "**noise /Outlier" data points**, which are data points that are not core data points or border data points.

Let's Understand DBSCAN steps by Real life **Example:**

Q-1: Apply the DBSCAN algorithm to the given dataset P1(3,7), P2(4,6), P3(5,5), P4(6,4), P5(7,3), P6(6,2), P7(7,2), P8(8,4), P9(3,3), P10(2,6), P11(3,5), P12(2,4). Create the clusters with 4 minpts and value of epsilon is 1.9.

Solution:

Given that,

Minpts = 4 (i.e. each cluster should contain minimum 4 data points)

And epsilon (ϵ) = 1.9 (i.e. the distance between each data points within a cluster should be less than or equal to 1.9 unit)

Now,

Calculate the distance between the data points:

We know, Euclidian distance for points P1 (x1, y1) and P2 (x2, y2),

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Distance between data points are:

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.0	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

Step-2: after calculating the distance from one data points to another datapoints, ***find the nearest data points*** from each of the datapoints.

Now, start from P1:

Here, try to find the nearest distance from P1 to other datapoints horizontally then vertically.

Hence, distance from P1 to P1 horizontally is 0. i.e. P1 will be the part of P1 here. (horizontally)

Again, look vertically from P1 to all other datapoints and try to find the distance which is less than or equal to 1.9 unit. Because we have given the epsilon = 1.9 unit.

Therefore. P2 and P10 are nearest one here (vertically)

P1 = P2, P10

Similarly,

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

P1= P2, P10	P7= P5, P6
P2=P1, P3, P11	P8=P5
P3=P2, P4	P9=P12
P4=P3, P5	P10=P1, P11
P5=P4, P6, P7, P8	P11=P2, P10, P12
P6=P5, P7	P12=P9, P11

Step-3: Identify the ***core, border and noise*** / outlier data datapoints

Now, for P1: P2, P10 (here, including P1, P1 has only 3 datapoints but question tells us that minpts must be minimum 4)

Therefore, P1 is Noise datapoint.

Similarly, for P2: P1, P3, P11 (here, including P2, P2 has 4 datapoints)

Therefore, P2 is core datapoint.

For other datapoints:

P1	Noise	P7	Noise
P2	Core	P8	Noise
P3	Noise	P9	Noise
P4	Noise	P10	Noise
P5	Core	P11	Core
P6	Noise	P12	Noise

Step-4: Check Border datapoints (if any datapoints is the border datapoints then it should be a part of any core datapoints)

For P1, if P1 is the part of any core points it is border datapoints, but p1 is included in P2: P1, P3, P11 hence, it is border data points.

Similarly, P3 is also the part of P2 (core) datapoints. Therefore, P3 is also border

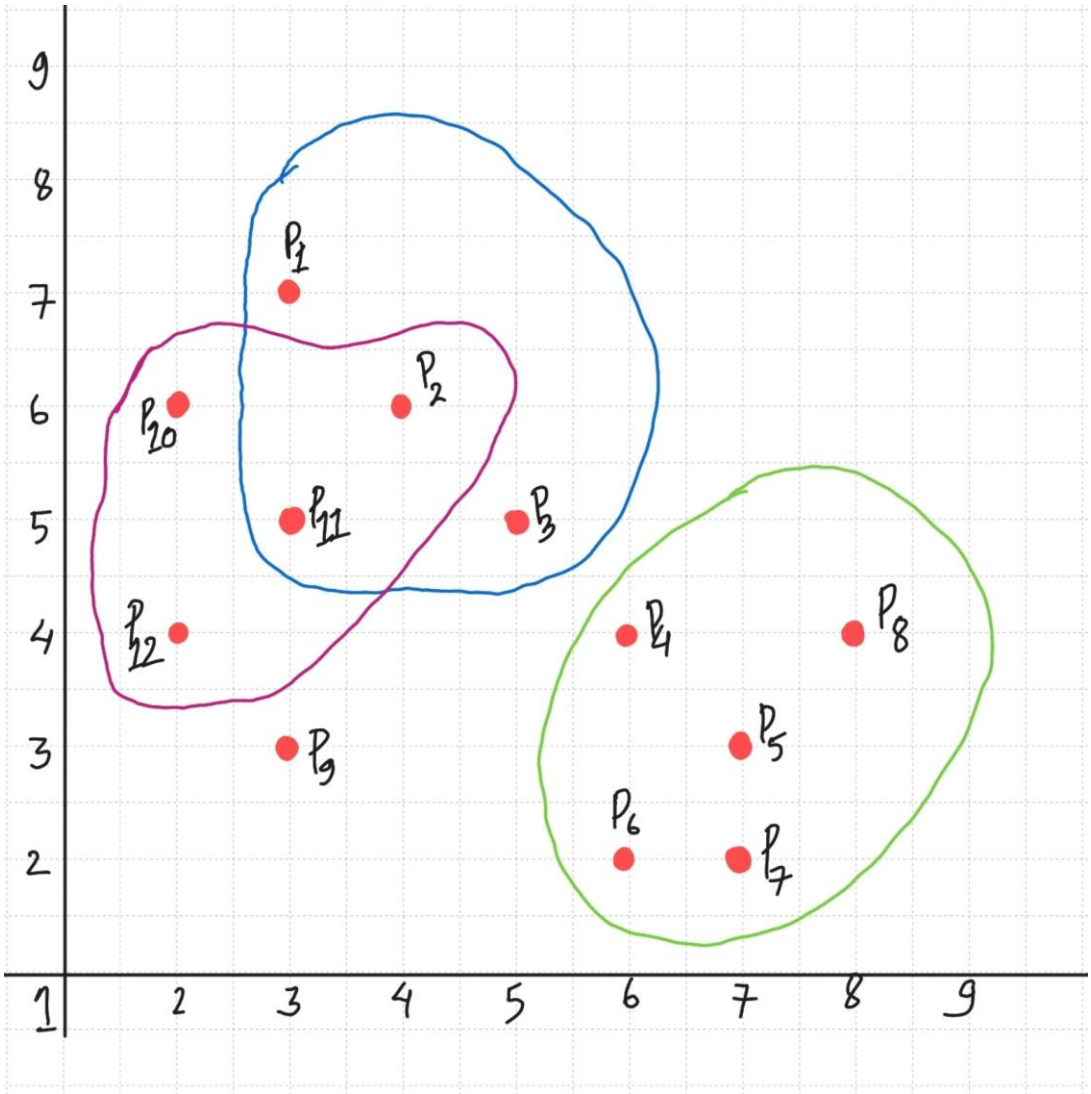
By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Therefore,

P1	Noise	Border	P5	Core		P9	Noise	
P2	Core		P6	Noise	Border	P10	Noise	Border
P3	Noise	Border	P7	Noise	Border	P11	Core	
P4	Noise	Border	P8	Noise	Border	P12	Noise	Border

But, here P9 remains noise. i.e. P9 is not a part of any core datapoints. And we have 3 core data points (i.e. we will have 3 cluster and P2, P5 and P11 will be center of cluster).

Step-5: Plot the Datapoints and ***draw the cluster*** with respect to core data points.



By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

3.4. Dimensionality Reduction

Dimensionality reduction is an unsupervised learning technique that reduces the number of variables / features, or dimensions, in a dataset.

- More data is generally better for machine learning, but it can also make it more challenging to visualize the data.
- Dimensionality reduction extracts important features from the dataset, reducing the number of irrelevant or random features present.

Lets' Understand it with an **Example:**

Suppose, you train a model that can forecast the next day's weather based on the current climatic variables such as the amount of sunlight, rainfall, temperature, humidity, and several other environmental factors. Analyzing all these variables is a complex and challenging task. Hence, to accomplish the task with a limited set of features, you can target only specific features that show a stronger correlation and can be clubbed into one.

For instance, we can combine the humidity and temperature variables into one dependent feature as they tend to show a stronger correlation. With such a clubbing method, dimensionality reduction compresses complex data into a simpler form and ensures that the end objective is achieved without losing the data's crux.

Advantages	Disadvantages
1. Fastr Computation	1. Data loss and reduced accuracy
2. Better Visualization	2. Choosing right component is difficult
3. Prevent overfitting	

To reduce the number of data inputs without compromising the integrity of the properties in the original data ***Dimensionality reduction techniques can be broadly divided into two categories:***

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

1. **Feature Selection**

   This refers to retaining the relevant (optimal) features and discarding the irrelevant ones to ensure the high accuracy of the model. Feature selection methods such as filter, wrapper, and embedded methods are popularly used

2. **Feature Extraction**

   This process is also termed feature projection; wherein multidimensional space is converted into a space with lower dimensions.

   - Common feature extraction techniques and dimensionality reduction are as:
     - Principal component analysis (PCA)
     - Linear discriminant analysis (LDA)
     - Kernel PCA (K-PCA)
     - Quadratic discriminant analysis (QCA) and
     - Singular value decomposition (SVD)

3.4.1. Principal Component Analysis (PCA)

Principal component analysis (PCA) is a type of dimensionality reduction technique which is used to reduce redundancies and to compress datasets through feature extraction.

- This method uses a linear transformation to create a new data representation, yielding a set of "principal components."
- The first principal component is the direction which maximizes the variance of the dataset. While the second principal component also finds the maximum variance in the data, it is completely uncorrelated to the first principal component, yielding a direction that is perpendicular, or orthogonal, to the first component.
- This process repeats based on the number of dimensions, where a next principal component is the direction orthogonal to the prior components with the most variance.
- The number of principal components is usually less than or equal to the number of original features, and can be chosen based on a desired level of explained variance or information loss.
- PCA only considers the variation or structure of the data

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

When to use PCA

PCA is more suitable for exploratory data analysis, data visualization, feature extraction, or dimensionality reduction for unsupervised learning tasks, such as clustering or anomaly detection.

What are the Advantages and disadvantages of PCA?

PCA can reduce noise and redundancy in the data by removing correlated features, reveal hidden patterns or trends in the data by capturing the most significant dimensions, and be applied to any type of data. However, it can also lose important information or features that have low variance but high relevance, be affected by outliers or scaling issues, and produce principal components that are difficult to interpret or explain.

Steps in PCA:

1. Standardized the data and calculate the mean value for each feature ($\overline{X_\iota}$)
2. Compute the Covariance matrix (S)

   Cov (X_{ik}) = $\frac{1}{N}\sum_{K=1}^{N}(X_k - \overline{X_\iota})$ for all individual feature.

3. Computer Eigenvalues of covariance matrix

   Characteristic equation for covariance matrix is det (S-λ I) = 0

   Where I → is identity matrix.
4. Calculate the eigen vectors (U) = $\binom{u1}{u2}$. Here, U considered zero (0) initially.
5. Compute the principal component.
6. Project data onto new axis.

Example:

Reduce the dimension of given data from 2 to 1 using the PCA Algorithm.

Feature	**Example 1**	**Example 2**	**Example 3**	**Example 4**
X1	4	8	13	7
X2	11	4	5	14

Solution:

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Step-1: Calculate the mean value

for X_1 feature $(\bar{X}_1) = \frac{4+8+13+7}{4} = 8$

for X_2 feature $(\bar{X}_2) = \frac{11+4+5+14}{4} = 8.5$

Step-2: Calculate Covariance matrix (S)

$$S = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} Cov(X_1X_1) & Cov(X_1X_2) \\ Cov(X_2X_1) & Cov(X_2X_2) \end{bmatrix}$$

Now, $Cov(X_1, X_1) = \frac{1}{N-1} \sum_{K=1}^{N} (X_{1K} - \bar{X}_1)(X_{2K} - \bar{X}_2)$

$= \frac{1}{4-1} ((4-8)(4-8) + (8-8)(8-8) + (13-8)(13-8) + (7-8)(7-8))$ → 0

$= \frac{1}{3} (4-8)^2 + (13-8)^2 + (7-8)^2 = 14$ ✓

$Cov(X_1X_2) = \frac{1}{4} (4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)$

$= -11$ ✓

$Cov(X_2X_1) =$ Same as $(X_1X_2) = -11$ ✓

$Cov(X_2X_2) = \frac{1}{3} (11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5) = 23$ ✓

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

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

Step-3: Calculate Eigenvalues of the Cov. matrix

As we know, Characteristic eqn of the Cov. matrix is $\det(S - \lambda I) = 0$

$$\Rightarrow \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}$$

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

So solving,

$\lambda_1 = 30.3849$ & $\lambda_2 = 6.6151$

Step-4: Calculate the Eigenvectors!

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \Rightarrow (S - \lambda I)u = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} * \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} (14-\lambda)u_1 - 11u_2 \\ -11u_1 + (23-\lambda)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow (14-\lambda)u_1 - 11u_2 = 0 \quad \text{—(I)}$$

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

On solving these equations:

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

from (1)

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

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

so, $\frac{u_1}{11} = t \quad \therefore u_1 = 11t$

and, $\frac{u_2}{14-\lambda} = t \Rightarrow u_2 = (14-\lambda)t$

Here, to solve (1) & (2) we have assumed the value of $t = 1$,

Therefore, $u_1 = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix}$

Now, TO calculate the pricipal component we need to consider the largest value of λ. i.e. $\lambda = 30.3849$

(if you want next principal component you need to consider next largest eigen value and so on).

Here, we have only one principal component so only λ_1 is used.

Therefore $u_1 = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$

Again, we need to compute the length of u_1. so, calculate unit eigenvector.

i.e $\|u_1\| = \sqrt{(11^2 + (14-\lambda)^2} = 19.7348$

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

So, $e_1 = \begin{pmatrix} \frac{11}{114.11} \\ \frac{14-\lambda}{114.11} \end{pmatrix} = \begin{pmatrix} 0.5574 \\ -0.8308 \end{pmatrix}$

if there were next principal component then we need to calculate next e_2 by utilizing the x_2 but question feels only one principal component so e_1 is enough for this question.

Step-5: Compute 1st principal component

eqn for principal component $= e^T \begin{bmatrix} x_{1k} - \bar{x}_1 \\ x_{2k} - \bar{x}_2 \end{bmatrix}$

$\Rightarrow [0.5574 \quad -0.8308] \begin{bmatrix} x_{1k} - \bar{x}_1 \\ x_{2k} - \bar{x}_2 \end{bmatrix}$

for Example 1:

$\Rightarrow 0.5574 (x_{11} - \bar{x}_1) - 0.8308 (x_{21} - \bar{x}_2)$

$\Rightarrow 0.5574 (4-8) - 0.8308 (11 - 8.5)$

$= -4.30535$

Similarly,

for Example 2 = 3.7361

for Example 3 = 5.9628

for Example 4 = -5.1238

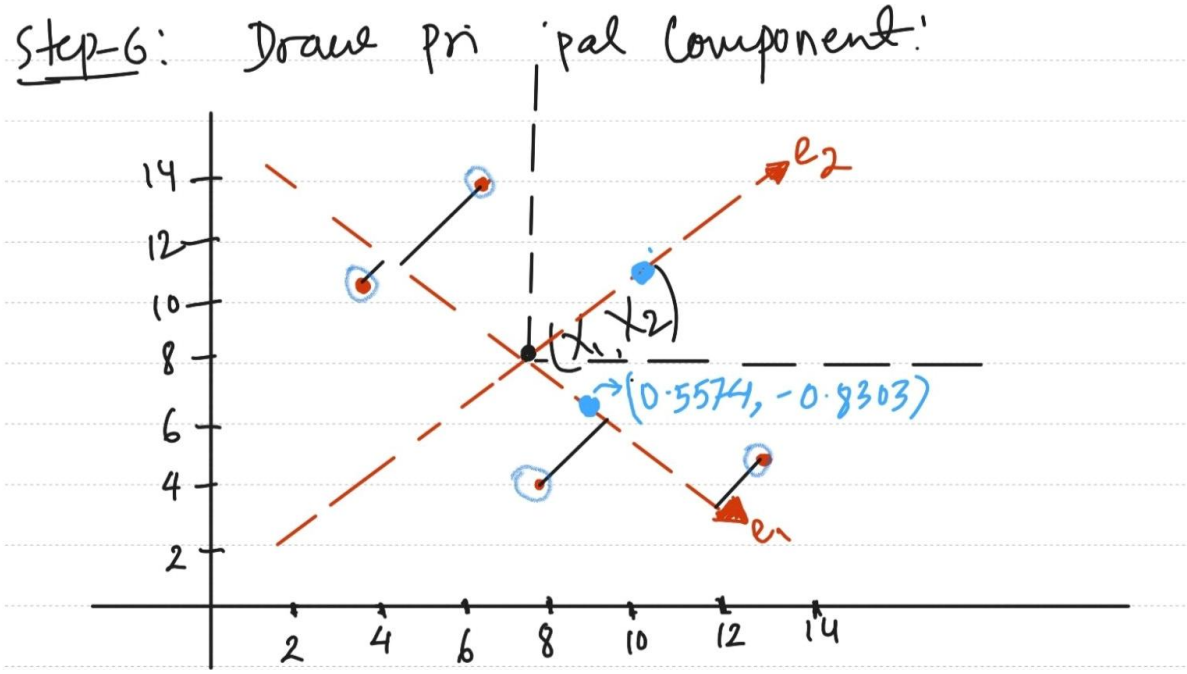
This shows that,

Every Example has 2 components as features but we reduced them into single principal component as shown in table below.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Feature	Example 1	Example 2	Example 3	Example 4
X1	4	8	13	7
X2	11	4	5	14
First Principal Components	-4.3052	3.7361	5.6928	-5.1238

Step-6: Draw the principal Component:



3.4.2. Linear Discriminant Analysis (LDA)

- Also known as Normal Discriminant Analysis (NDA) / Discriminant Function Analysis (DFA)
- LDA is commonly used for supervised classification problems for feature extraction.
- Commonly used to project the features in higher dimensional space into lower dimensional space.

Example:

Suppose that a bank is deciding whether to approve or reject loan applications. The bank uses two features to make this decision: the applicant's credit score and annual income.

Here, the two features or classes are plotted on a 2-dimensional (2D) plane with an X-Y axis. If we tried to classify approvals using just one feature, we might observe overlap. By applying LDA,

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

we can draw a straight line that completely separates these two class data points. LDA achieves this by using the X–Y axis to create a new axis, separating the different classes with a straight line and projecting data onto the new axis.

The Goal of LDA is to find a linear combination of features that:

- Maximize the distance between the means of two classes.
- Minimize the variance (spread) within individual classes.

Key Assumptions of LDA:

1. **Gaussian Distribution**: LDA assumes that the features within each class follow a normal (Gaussian) distribution.
2. **Equal Covariance Matrices**: It assumes that the variance (or spread) of data points is the same across all classes.
3. **Linearity**: The relationship between the features and the target variable is assumed to be linear.
4. **Independence**: The features used should ideally be independent of each other.

Advantages and Disadvantages of LDA

LDA can improve the classification performance and accuracy of machine learning models by enhancing the class separation, preserve class information and labels of the data by using them as criteria for dimensionality reduction, and handle multicollinearity or high correlation among features.

LDA can overfit the data or produce biased results if the class distribution or number of samples per class is imbalanced, be affected by assumptions of normality and independence of features and classes; and produce linear discriminants that are limited by the linear relationship between features and classes.

When to use LDA?

LDA is more suitable for predictive data analysis, data preprocessing, feature selection, or dimensionality reduction for supervised learning tasks, such as classification or regression.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

However, there is no definitive rule or formula for choosing between PCA and LDA, and it is advisable to experiment with both methods and compare the results and performance of the machine learning models.

Steps in LDA

1. Calculate the mean vectors for each class.

$$\mu_i = \frac{1}{N} \sum_{x \in \omega_i} x$$

2. Find the covariance matrix of each class variable.

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)\,(x - \mu_i)^T$$

where x is the data and μ_i *is the mean of the particular class.*

3. Compute within the class Scatter matrix

$$S_w = S1 + S2 + \ldots$$

4. Compute between the 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.

$$S_w^{-1}\, S_B\, w = \lambda\, w$$

6. Sort the values of eigen values and select the top k values
7. Find the eigen vectors corresponds to the k eigen vectors:

$$(S_w^{-1}\, S_B - \lambda\, I) \begin{pmatrix} W1 \\ W2 \end{pmatrix} = 0$$ (if class are more than two then w3, w4 … are added simultaneously.

8. Plot dimension reduction:

$y = W^T . x$

where, W^T is the projection vector and x is input data sample.

Example:

Compute the Linear Discriminant Projection for the given two-dimensional dataset.

Samples for class $\omega_1 = X_1 = (x_1, x_2)$ = {(4,1), (2,4), (2,3), (3,6), (4,4)}

Samples for class $\omega_2 = X_2 = (x_1, x_2)$ = {(9,10), (6,8), (9,5), (8,7), (10,8)}

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Solution:

Step-1: Compute the class mean

$$M_1 = \frac{1}{N}\sum_{x\in\omega_1}^{x} = \frac{1}{5}\left[\begin{pmatrix}4\\1\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right] = \begin{bmatrix}3\\3.6\end{bmatrix}$$

and $$M_2 = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right] = \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Step-2: Compute Covariance matrix of classes

$$(x-M_1) = \begin{bmatrix}1 & -1 & -1 & 0 & 1\\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4\end{bmatrix}$$

for each x, calculate $(x-M_1)(x-M_1)^T$ so:

$$\begin{bmatrix}1\\-2.6\end{bmatrix} * \begin{bmatrix}1 & -2.6\end{bmatrix} = \begin{bmatrix}1 & -2.6\\ -2.61 & 6.76\end{bmatrix}$$

Similarly,

for $\begin{bmatrix}-1\\0.4\end{bmatrix}\begin{bmatrix}-1 & 0.4\end{bmatrix} = \begin{bmatrix}1 & -0.4\\ -0.4 & 0.16\end{bmatrix}$

for $\begin{bmatrix}-1\\-0.6\end{bmatrix}\begin{bmatrix}-1 & -0.6\end{bmatrix} = \begin{bmatrix}1 & 0.6\\ 0.6 & 0.36\end{bmatrix}$

for $\begin{bmatrix}0\\2.4\end{bmatrix}\begin{bmatrix}0 & 2.4\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 5.76\end{bmatrix}$

for $\begin{bmatrix}1\\0.4\end{bmatrix}\begin{bmatrix}1 & 0.4\end{bmatrix} = \begin{bmatrix}1 & 0.4\\ 0.4 & 0.16\end{bmatrix}$

Adding all above matrix and averaging the element

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

· We get $Cov(S_1)$

$$\text{i.e. } Cov(S_1) = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$$

for second class (ω_2)

Similar like S_1 we will get $Cov(S_2)$ where M_2 is $[8.4 \quad 7.6]$

$$\text{i.e. } Cov(S_2) = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

Step-3: Compute within class scatter matrix (S_W)

We know $S = S_1 + S_2$

$$\begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.26 \end{bmatrix}$$

this matrix gives the idea about how well the data is scattered within the class.

Step-4: Compute Between the class scatter matrix (S_B)

We know, $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

$$\Rightarrow \begin{pmatrix} -5.4 \\ -4 \end{pmatrix} (-5.4 \quad -4) \Rightarrow \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16.00 \end{bmatrix}$$

This matrix gives the idea about how well the data is scattered across the classes.

Step-5: Find the Eigen values and Eigen vectors (i.e finding of best LDA projection vectors)

(Same as PCA)

2 / 5 100%

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

we know $S_W^{-1} \cdot S_B \vec{W}^{\text{eigen}} = \lambda W$ → eigen/projector vector ——— (*)

$\Rightarrow |S_W^{-1} \cdot S_B - \lambda I| = 0$

Let's find the S_W^{-1}.

we know that $S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.26 \end{bmatrix}$

$$\Rightarrow S_W^{-1} = \frac{1}{2.64 \times 5.26 - 0.44 \times 0.44} \begin{bmatrix} 5.26 & 0.44 \\ 0.44 & 2.64 \end{bmatrix}$$

$$= \begin{bmatrix} 0.384 & 0.0321 \\ 0.0321 & 0.1928 \end{bmatrix}$$

Therefore, $(S_W^{-1} \cdot S_B) - \lambda I = 0$ will be

$$\Rightarrow \begin{pmatrix} 0.384 & 0.0321 \\ 0.0321 & 0.1928 \end{pmatrix} \begin{pmatrix} 29.16 & 21.6 \\ 21.6 & 16.00 \end{pmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{pmatrix} 11.891 & 8.808 \\ 5.105 & 3.781 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = 0$$

$$\Rightarrow \begin{pmatrix} 11.891 - \lambda & 8.808 \\ 5.105 & 3.781 - \lambda \end{pmatrix} = 0$$

$\Rightarrow (11.891 - \lambda)(3.781 - \lambda) - 5.105 \times 8.808 = 0$

$\Rightarrow 44.959 - 15.672\lambda + \lambda^2 - 44.985 = 0$

$\Rightarrow -0.006 - 15.672\lambda + \lambda^2 = 0$

on solving, $\lambda_1 = -0.000382$ & $\lambda_2 = 15.67238$

Select largest λ so λ_2 is selected.

3 / 5 100%

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

Now, put λ in ⊛ to find eigen vector.

$$S_W^{-1} S_B W = \lambda W$$

$$\Rightarrow \begin{pmatrix} 11.891 & 8.808 \\ 5.105 & 3.71 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 15.67238 \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

On solving we get $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0.91 \\ 0.39 \end{pmatrix}$

Step-6 plot Dimension Reduction

we know, $y = W^T X_i$

for $X_1(4,1)$

1st LD $\Rightarrow \begin{bmatrix} 0.91 & 0.39 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix}$

$\Rightarrow 0.91*4 + 0.39*1 = 3.95$

Similarly,

Similarly, for all the points in X1 and X2 we will get following 1st LDA.

X1	4	2	2	3	4	9	6	9	8	10
X2	1	4	3	6	4	10	8	5	7	8
1st LD	**3.95**	**3.48**	**3.06**	**5.2**	**5.3**	**12.35**	**8.8**	**10.2**	**10.19**	**12.42**

Here, dimensionality of all the points has been reduced as shown in above table.

Finally,

let's visualized the complete results in graph. And after LDA you can clearly see that, there is clear separation between the two class over single axis. Which was the objective of LDA.

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com

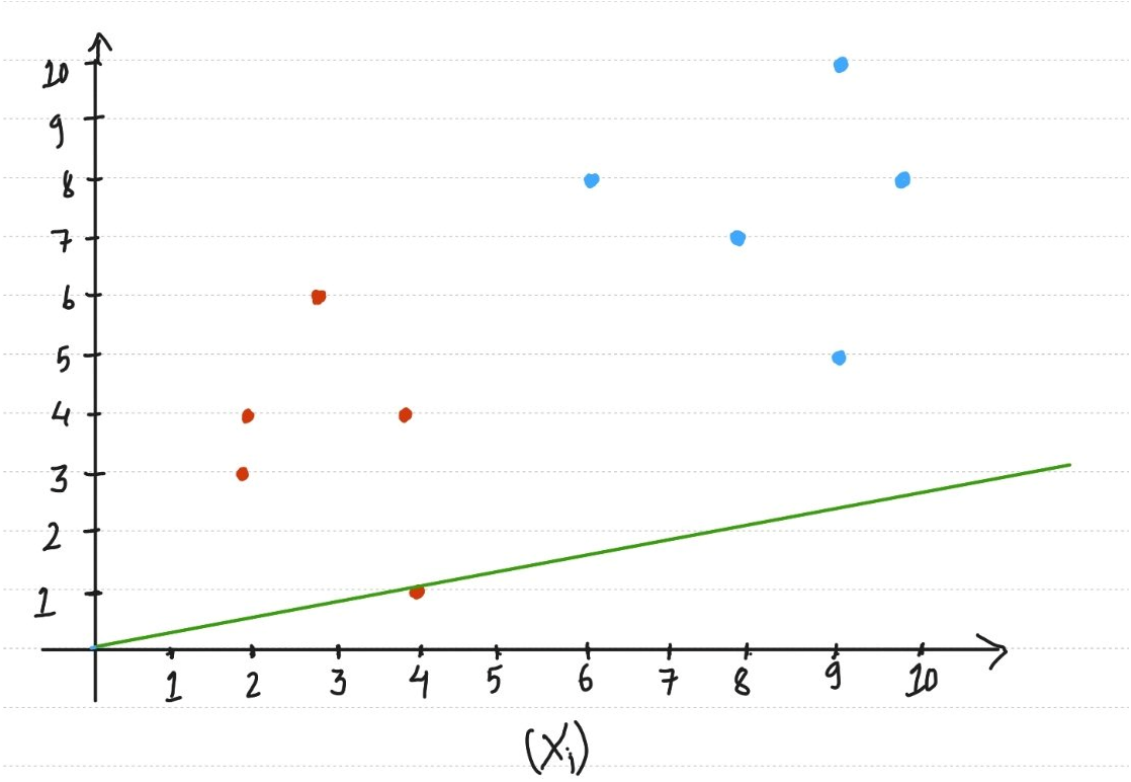


Figure: Data points before LDA

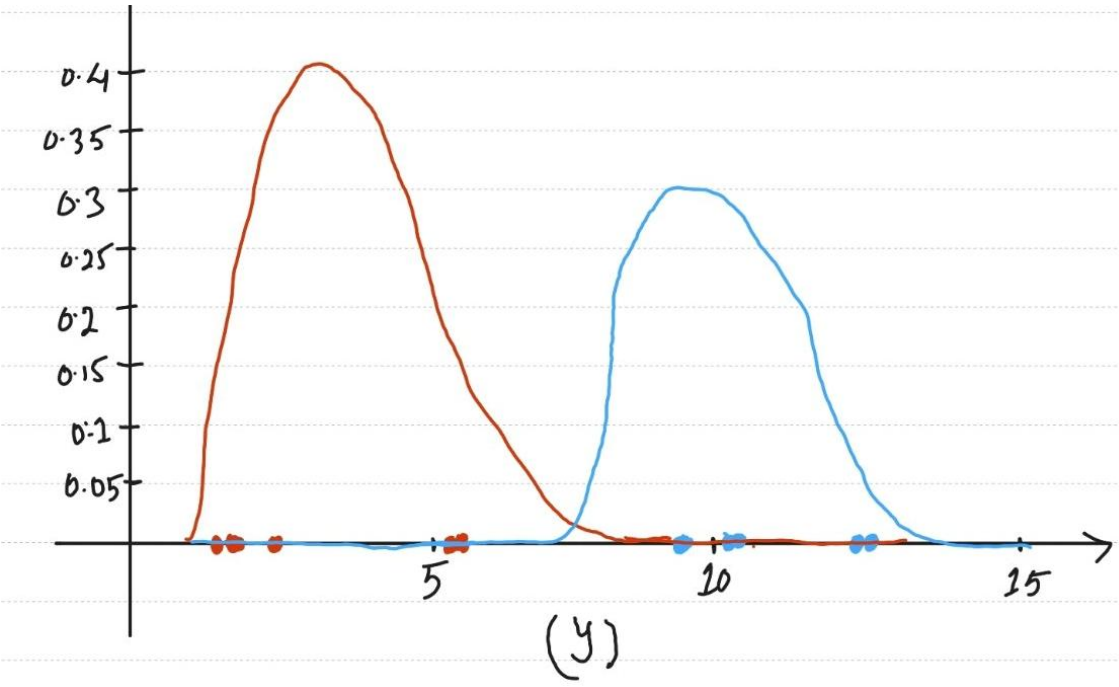


Figure: Data points (Class) after LDA

End of Chapter

By: Er. Mukunda Paudel, *M.E. Computer*
Email: paudelmuku@gmail.com