A0597203 AI Business Applications

Introduction to Machine Learning

AI Business Applications

Introduction to Machine Learning

Introduction to Machine Learning

In this part of the course, we will cover 3 algorithms in Machine Learning:

1. Regression (Supervised Learning)
2. Classification (Supervised Learning)
3. Clustering (Unsupervised Learning)

What Is Linear Regression?

Purpose: Predict a continuous numeric outcome (dependent variable) using one or more independent variables.

Use Case Example:

Problem	Example Features
House Price Prediction	Location, size (sqft), number of bedrooms, age of property
Sales Forecasting	Advertising budget, seasonality, past sales, promotions
Stock Price Prediction	Trading volume, past prices, news sentiment, technical indicators
Temperature Prediction	Date, time of day, humidity, wind speed, cloud cover
Medical Cost Estimation	Age, BMI, smoking status, number of children, region

Simple Linear Regression

- In **simple** linear regression, we use one independent variable `X` to predict `Y`.
- **Model equation:**

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where:

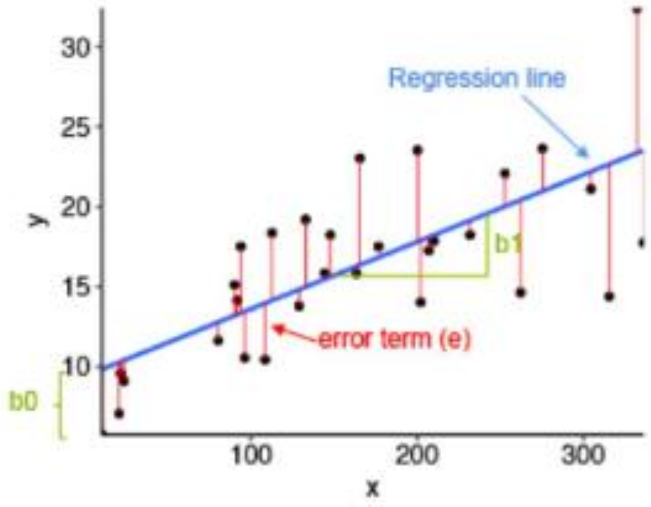
- Y: dependent variable
- X: independent variable (predictor)
- β_0: intercept
- β_1: slope (effect of X on Y)
- ε: error term (difference between actual and predicted values)

Goal of Regression

- Find values of β_0 and β_1 that minimize the **Sum of Squared Errors (SSE)**:

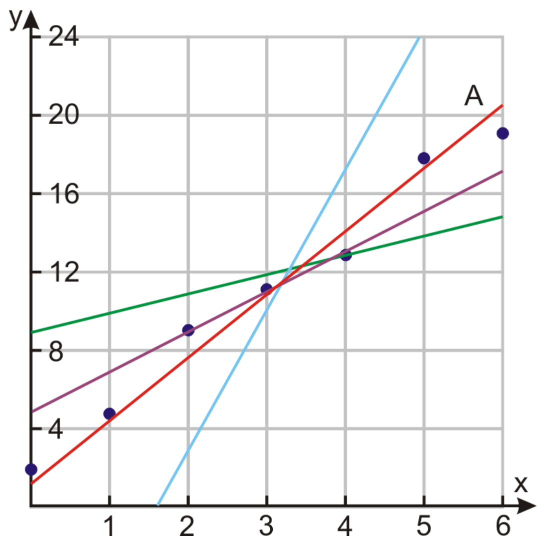
$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

- Use **Ordinary Least Squares (OLS)** to estimate coefficients.



Interpreting Coefficients

- β_1: For every unit increase in X, Y is expected to increase by β_1, holding all else constant.



Solving Simple Linear Regression using the Closed Form Method

The closed-form solution for Simple Linear Regression (SLR) is a direct mathematical formula used to compute the slope ($\beta 1$) and intercept ($\beta 0$) of the best-fit line without iterative optimization.

1. **Slope (β_1)** is given by:

$$\beta_1 = \frac{\mathrm{Cov}(X, y)}{\mathrm{Var}(X)}$$

Where:

- $\mathrm{Cov}(X, y)$ is the covariance between X and y,
- $\mathrm{Var}(X)$ is the variance of X.

2. **Intercept (β_0)** is calculated as:

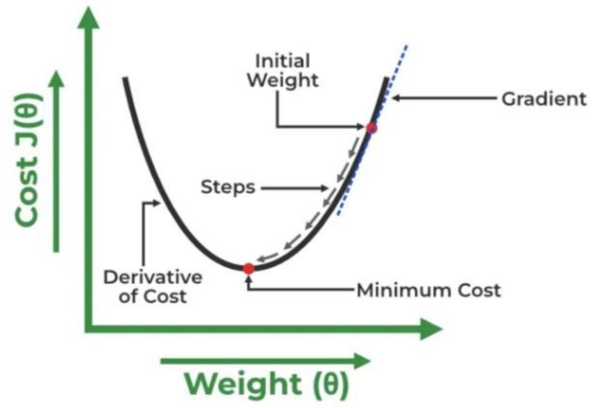
$$\beta_0 = \bar{y} - \beta_1 \bar{X}$$

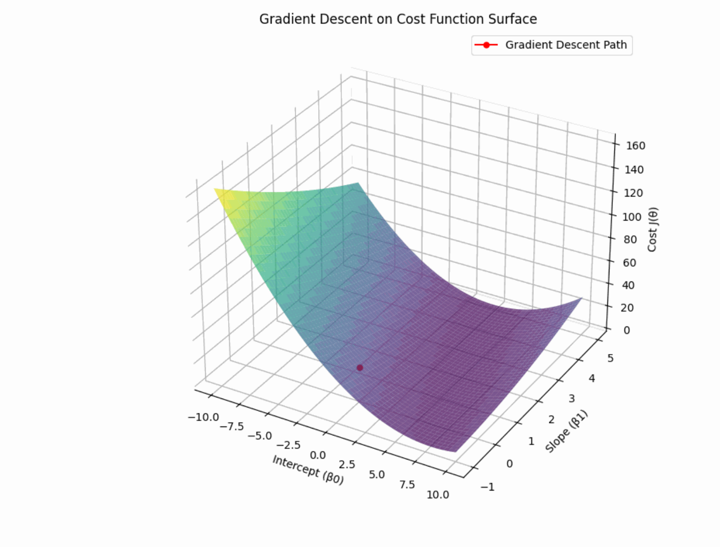
Where:

- $\bar{y}$ is the mean of the dependent variable y,
- $\bar{X}$ is the mean of the independent variable X.

Solving Regression using the Gradient Descent Method

- A linear regression model can be trained using **gradient descent** method, which adjusts the model's parameters to minimize the mean squared error (MSE).
- To update the Intercept (Beta 0) and the Slope (Beta 1) and reduce the cost function (minimizing the RMSE).
- Gradient descent starts with random values for the Intercept and the Slope and iteratively improves them to find the best-fit line.
- A gradient is simply the derivative, showing how small changes in inputs affect the output.
- By moving in the direction of the Mean Squared Error negative gradient with respect to the coefficients, the coefficients can be changed.





Model Evaluation Metrics

1. Total Sum of Squares (SST)
Measures the total variance in the actual data.
Formula: $\mathbf{SST = \Sigma(y_i - \bar{y})^2}$
Interpretation: How much the actual values vary from their mean.

2. Sum of Squares for Error (SSE)
Measures the unexplained variance (residuals).
Formula: $\mathbf{SSE = \Sigma(y_i - \hat{y}_i)^2}$
Interpretation: How far the predictions are from the actual values.

3. Sum of Squares for Regression (SSR)
Measures the variance explained by the regression model.
Formula: $\mathbf{SSR = \Sigma(\hat{y}_i - \bar{y})^2}$

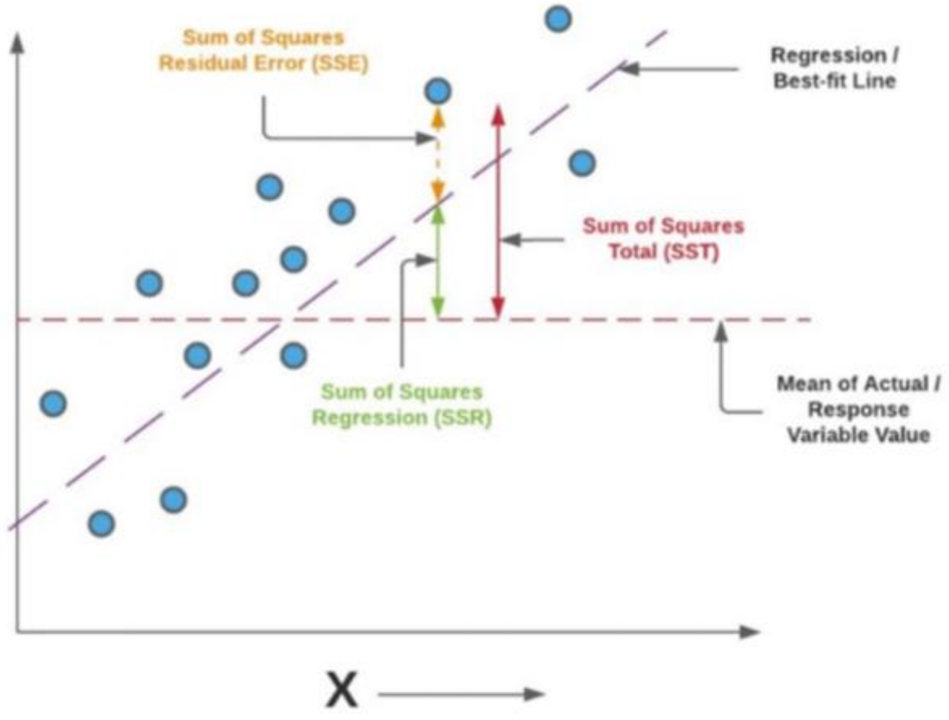
Interpretation: How much of the variation is captured by the model.

Relationship Between the Three
SST = SSR + SSE

4. R-squared (R^2)
Proportion of total variance explained by the model.
Formula: $\mathbf{R^2 = 1 - (SSE / SST)}$
Range: 0 to 1. Higher R^2 indicates better fit.

5. Mean Squared Error (MSE)
Average of the squared residuals.
Formula: **MSE = SSE / n**
Used to assess the model's prediction error.

6. Root Mean Squared Error (RMSE)
Square root of MSE.
Formula: $\mathbf{RMSE = \sqrt{MSE}}$



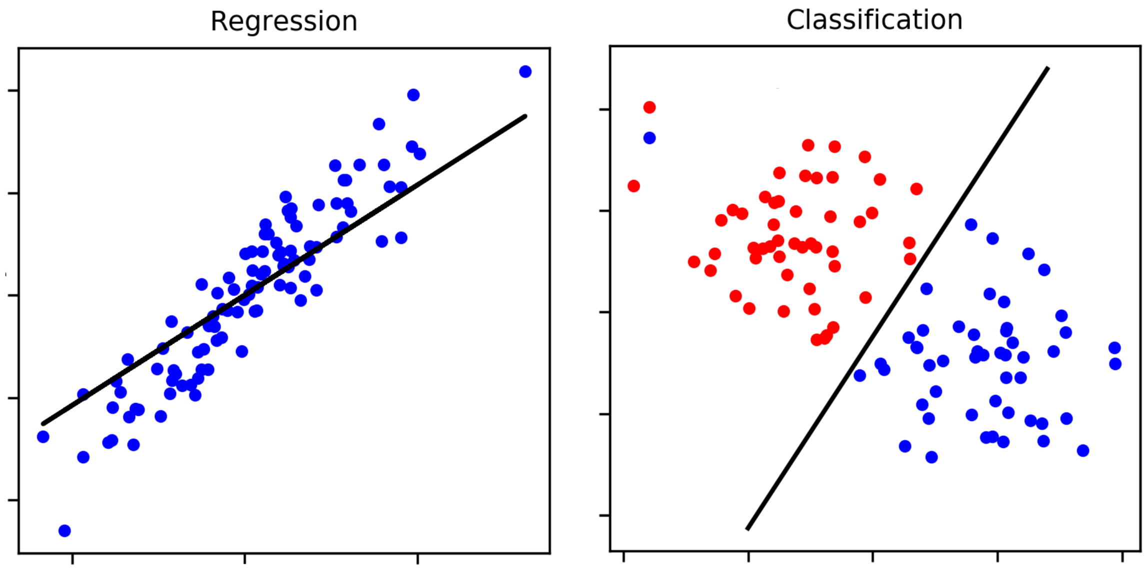
Classification

What is Classification?

Classification is a type of supervised learning where the goal is to predict a **categorical label** (like "yes" or "no") instead of a continuous value.

Examples:

- Predicting if an email is spam or not
- Determining if a customer will make a purchase
- Medical diagnosis



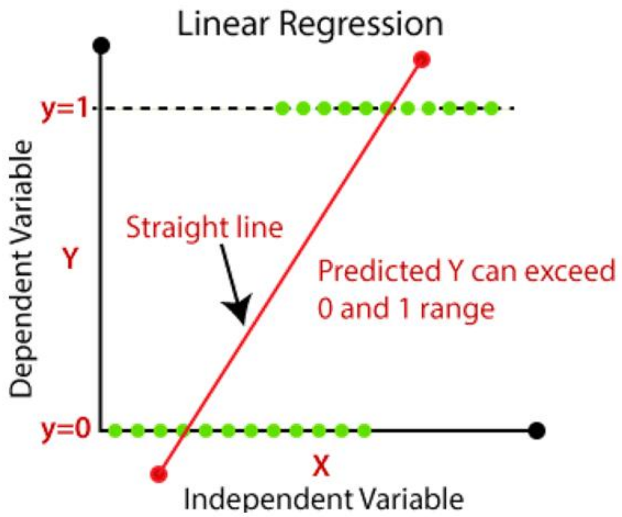
Binary Classification Examples

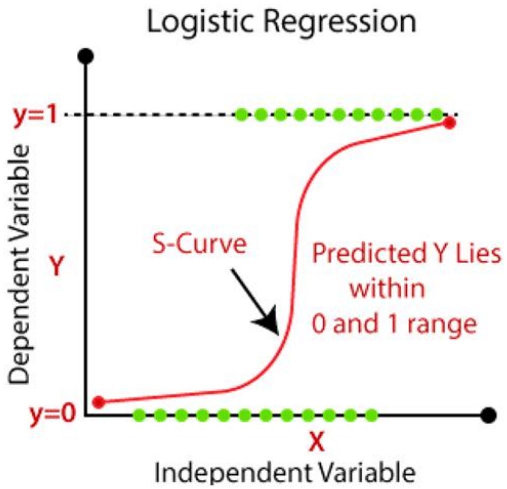
Problem	Example Features
Churn Prediction	Customer tenure, monthly charges, contract type, support calls
Spam Detection	Email subject length, sender reputation, word frequency
Loan Approval	Income, credit score, loan amount, employment status
Fraud Detection	Transaction amount, location, card usage frequency
Disease Diagnosis	Age, symptoms, test results, exposure history

Why Not Linear Regression?

Linear Regression Problems:

- Great for predicting continuous numbers (like prices)
- **Struggles with binary outcomes** (yes/no, spam/not spam)
- Can predict values outside 0-1 range
- Doesn't handle categorical data well
- **Solution:** Logistic Regression





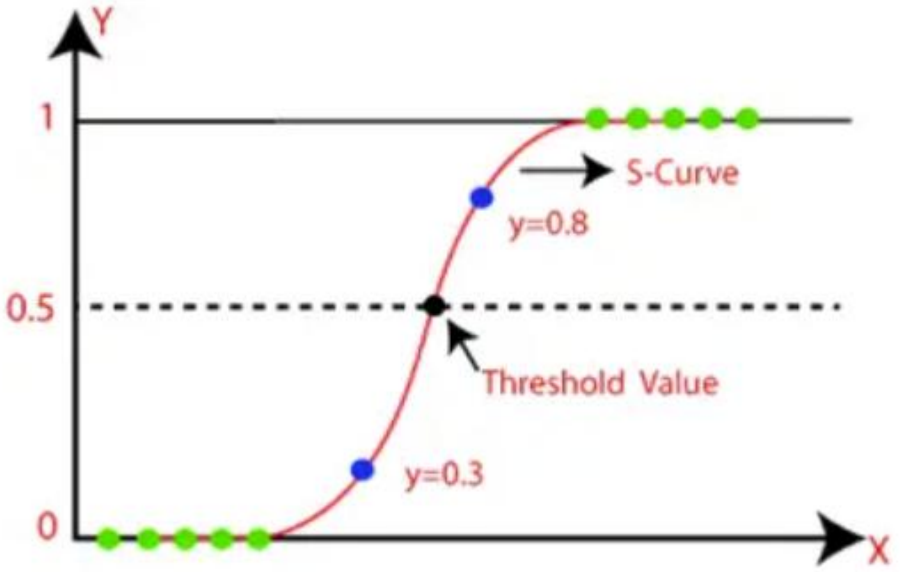
What is Logistic Regression?

- **Logistic regression** predicts the **probability** that an instance belongs to a certain class.
- **Key Component: Sigmoid Function**

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

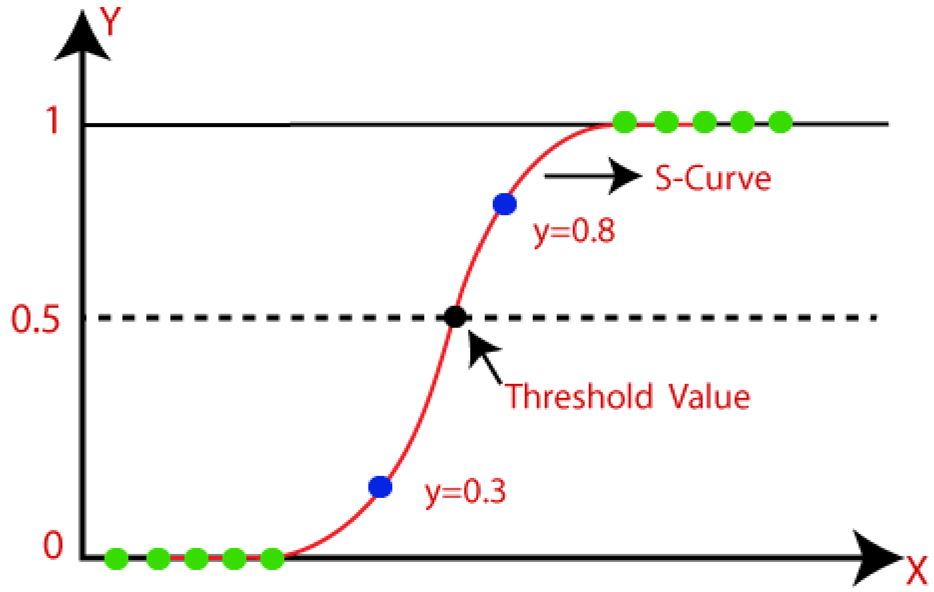
How it works:

- Takes any input value
- Squeezes it between 0 and 1
- Perfect for probability predictions!



Decision Making

- **Classification Rules:**
- If probability > 0.5 → Predict "Yes" (Class 1)
- If probability ≤ 0.5 → Predict "No" (Class 0)
- **Visual Comparison:**
- Linear Regression: Straight line, can go beyond 0-1
- Logistic Regression: S-curve, bounded between 0-1



Solving Logistic Regression

We solve logistic regression using **numerical optimization techniques**:

A. Gradient Descent (or variants like SGD, mini-batch GD)

We minimize the **negative log-likelihood** (i.e., cross-entropy loss):

$$\mathrm{Loss}(\beta) = -\sum_{i=1}^{n} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]$$

Gradient of the loss w.r.t. β is:

$$\nabla_\beta = X^T (\hat{y} - y)$$

Then you update the parameters using:

$$\beta \leftarrow \beta - \eta \cdot \nabla_\beta$$

Confusion Matrix

Use these four statistics to calculate other evaluation metrics, such as overall accuracy, true positive rate, and false positive rate

	Predicted class positive	Predicted class negative
True class positive	TRUE POSITIVE	FALSE NEGATIVE
True class negative	FALSE POSITIVE	TRUE NEGATIVE

1. TRUE POSITIVE (**TP**): Actual and predicted class is positive
2. TRUE NEGATIVE (**TN**): Actual and predicted class is negative
3. FALSE NEGATIVE (**FN**): Actual class is positive and predicted negative
4. FALSE POSITIVE (**FP**): Actual class is negative and predicted positive

Overall Accuracy

- Definition:

$$\boldsymbol{Overall\ accuracy} = \frac{\#\ \boldsymbol{Correct\ classifications\ (test\ set)}}{\#\ \boldsymbol{All\ events\ (test\ set)}}$$

- The proportion of correct classifications

- Downsides:
    - Only considers the performance in general and not for the different classes
    - Therefore, not informative when the class distribution is unbalanced

Confusion Matrix for Sailing Example

Diabetes yes / no	Predicted class: yes	Predicted class: no
True class: yes	22	3
True class: no	12	328

Diabetes yes / no	Predicted class: yes	Predicted class: no
True class: yes	0	25
True class: no	0	340

$$Accuracy = \frac{350}{365} = 0{,}96$$

$$Accuracy = \frac{340}{365} = 0{,}93$$

- Rows – true class values
- Columns – predicted class values
- Numbers on main diagonal – correctly classified samples
- Numbers off the main diagonal – misclassified samples

Unsupervised Learning

Clustering

What Is Unsupervised Learning?

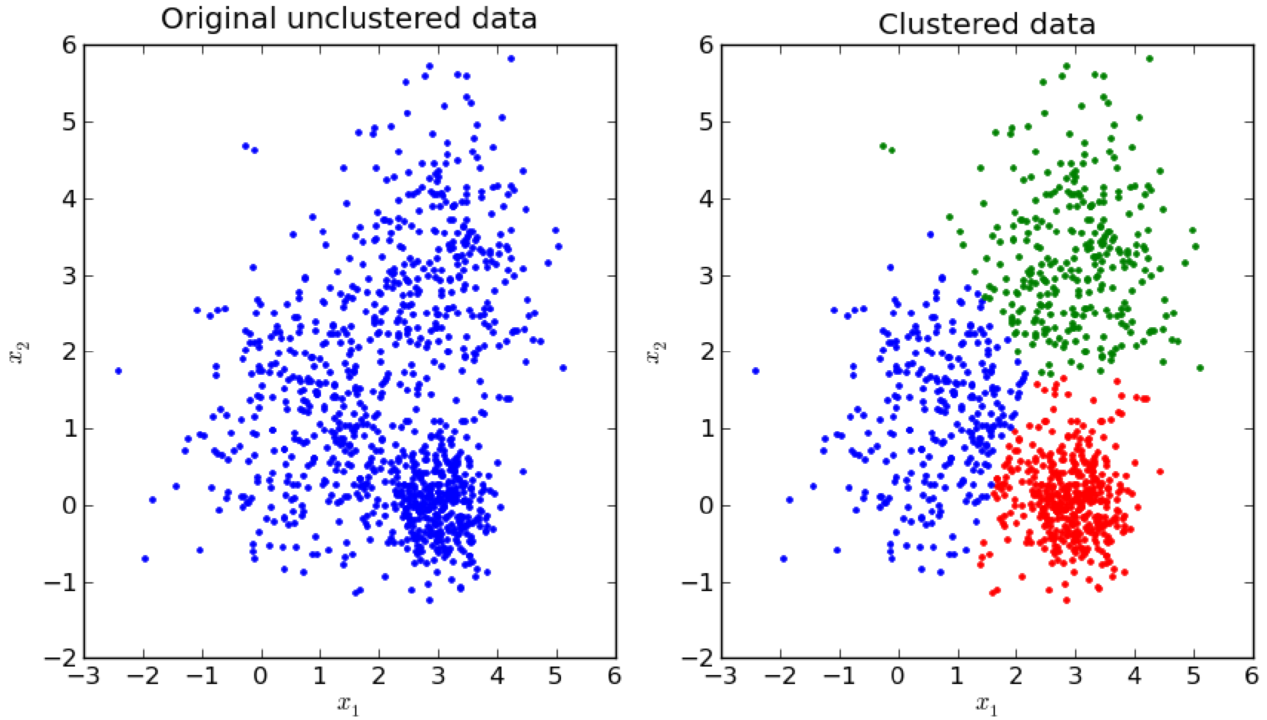
- **Definition**: Learning from data without labeled outcomes.
- **Goal**: Discover hidden patterns or structures.
- **Contrast with Supervised Learning**:
  - Supervised: Has labels (e.g., spam / not spam)
  - Unsupervised: No labels (e.g., group customers)

Use Cases of Unsupervised Learning

Problem	Example Features	Common Techniques
Customer Segmentation	Purchase history, age, income, browsing behavior	K-Means, Hierarchical Clustering
Anomaly Detection	Network activity, transaction amount, login frequency, location	Isolation Forest, DBSCAN, Autoencoders
Document Clustering	Word frequency, TF-IDF scores, topic distribution	K-Means, LDA (Latent Dirichlet Allocation)
Market Basket Analysis	Product IDs, purchase combinations, quantity, purchase sequence	Apriori, FP-Growth
Image Compression	Pixel intensity values, color channels, image dimensions	PCA, Autoencoders

What Is Clustering?

- **Definition**: Grouping similar data points into clusters.
- **Objective**: Points in the same cluster are more like each other than to those in other clusters



Introduction to K-Means Clustering

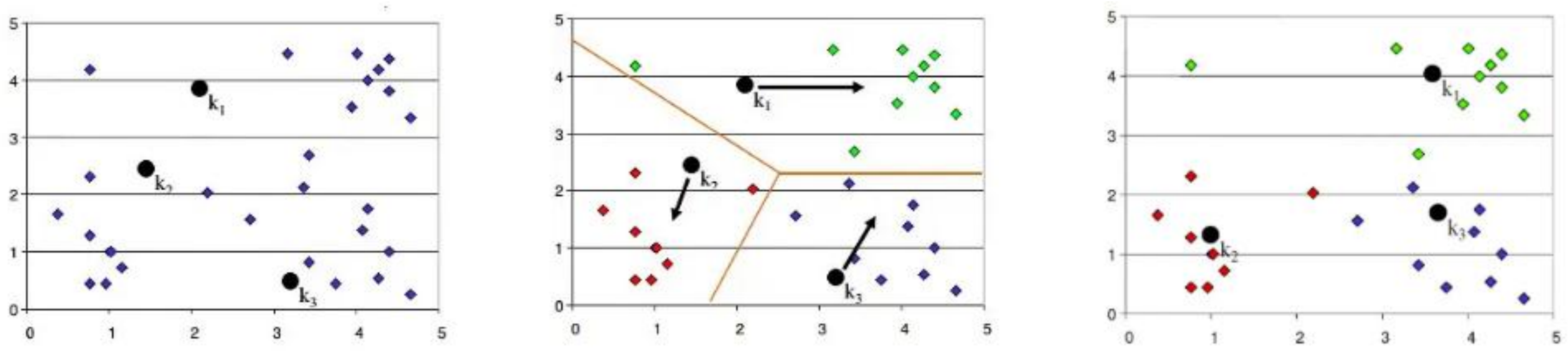
K-Means Clustering is an **unsupervised learning algorithm** used to group similar data points into **K** clusters, where each point belongs to the cluster with the nearest **centroid** (mean of the cluster).

How It Works:

- **Choose K**: the number of clusters.
- **Initialize** K centroids randomly.
- **Assign** each data point to the nearest centroid.
- **Update** centroids by calculating the mean of the assigned points.
- **Repeat** steps 3–4 until assignments no longer change (convergence).

Goal: Minimize the **within-cluster sum of squares (WCSS)** — i.e., the total distance between each point and its cluster centroid.

It's simple, efficient, and widely used for tasks like **customer segmentation**, **image compression**, and **pattern discovery**.



K-Means Step by Step Example

Step 1: Euclidean Distance Formula

The Euclidean distance measures the straight-line distance between two points in a 2D space. The formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

We will use this formula to compute the distance between each point and the centroids.

Problem Setup

- **Data Points**: (x, y)
  - A1: (2, 10)
  - A2: (2, 5)
  - A3: (8, 4)
  - A4: (5, 8)
  - A5: (7, 5)
  - A6: (6, 4)
  - A7: (1, 2)
  - A8: (4, 9)
- **Initial Centroids**:
  - Mean 1: (2, 10)
  - Mean 2: (5, 8)
  - Mean 3: (1, 2)

Iteration 1

Steps:

1. **Compute Distances**: Use the Euclidean distance formula to calculate the distance from each point to all centroids.
2. **Assign Clusters**: Each point is assigned to the cluster of the nearest centroid.

Results:

- Distances and cluster assignments are shown in the table below.

Point	Coordinates	Dist Mean 1 ($\sqrt{(2-x)^2 + (10-y)^2}$)	Dist Mean 2 ($\sqrt{(5-x)^2 + (8-y)^2}$)	Dist Mean 3 ($\sqrt{(1-x)^2 + (2-y)^2}$)	Cluster
A1	(2, 10)	$\sqrt{(2-2)^2 + (10-10)^2} = 0.00$	$\sqrt{(5-2)^2 + (8-10)^2} = 3.61$	$\sqrt{(1-2)^2 + (2-10)^2} = 8.25$	1
A2	(2, 5)	$\sqrt{(2-2)^2 + (10-5)^2} = 5.00$	$\sqrt{(5-2)^2 + (8-5)^2} = 3.61$	$\sqrt{(1-2)^2 + (2-5)^2} = 3.16$	3
A3	(8, 4)	$\sqrt{(2-8)^2 + (10-4)^2} = 9.43$	$\sqrt{(5-8)^2 + (8-4)^2} = 5.00$	$\sqrt{(1-8)^2 + (2-4)^2} = 7.28$	2
A4	(5, 8)	$\sqrt{(2-5)^2 + (10-8)^2} = 3.61$	$\sqrt{(5-5)^2 + (8-8)^2} = 0.00$	$\sqrt{(1-5)^2 + (2-8)^2} = 6.71$	2
A5	(7, 5)	$\sqrt{(2-7)^2 + (10-5)^2} = 8.06$	$\sqrt{(5-7)^2 + (8-5)^2} = 3.61$	$\sqrt{(1-7)^2 + (2-5)^2} = 6.08$	2
A6	(6, 4)	$\sqrt{(2-6)^2 + (10-4)^2} = 8.94$	$\sqrt{(5-6)^2 + (8-4)^2} = 4.47$	$\sqrt{(1-6)^2 + (2-4)^2} = 5.39$	2
A7	(1, 2)	$\sqrt{(2-1)^2 + (10-2)^2} = 8.00$	$\sqrt{(5-1)^2 + (8-2)^2} = 7.62$	$\sqrt{(1-1)^2 + (2-2)^2} = 1.00$	3
A8	(4, 9)	$\sqrt{(2-4)^2 + (10-9)^2} = 2.24$	$\sqrt{(5-4)^2 + (8-9)^2} = 1.41$	$\sqrt{(1-4)^2 + (2-9)^2} = 7.07$	2

Recompute Centroids

Using the provided table, the **new centroids** are computed as follows:

Cluster Assignments

From the table:

- **Cluster 1**: A1 (Coordinates: $(2, 10)$)
- **Cluster 2**: A3, A4, A5, A6, A8 (Coordinates: $(8, 4), (5, 8), (7, 5), (6, 4), (4, 9)$)
- **Cluster 3**: A2, A7 (Coordinates: $(2, 5), (1, 2)$)

Centroid Calculations

1. **Centroid for Cluster 1**: Since Cluster 1 has only one point, the centroid remains at the coordinates of A1: $\text{Centroid}_1 = (2.0, 10.0)$
2. **Centroid for Cluster 2**: $\text{Centroid}_2 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right)$ $\text{Centroid}_2 = (6.0, 6.0)$
3. **Centroid for Cluster 3**: $\text{Centroid}_3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right)$ $\text{Centroid}_3 = (1.5, 3.5)$

Updated Centroids for Iteration 1

- **Cluster 1**: $(2.0, 10.0)$
- **Cluster 2**: $(6.0, 6.0)$
- **Cluster 3**: $(1.5, 3.5)$

Iteration 2

Steps:

1. **Recalculate Centroids**: For each cluster, compute the new centroid as the mean of all points in that cluster.
2. **Reassign Clusters**: Recompute distances to the updated centroids and assign points to the nearest cluster.

Results:

- Distances and new cluster assignments are shown in the table below.
- **Used Centroids**: $(2.0, 10.0)$, $(6.0, 6.0)$, $(1.5, 3.5)$

Point	Coordinates	Dist Mean 1 ($\sqrt{(x1-x)^2 + (y1-y)^2}$)	Dist Mean 2 ($\sqrt{(x2-x)^2 + (y2-y)^2}$)	Dist Mean 3 ($\sqrt{(x3-x)^2 + (y3-y)^2}$)	Cluster
A1	(2, 10)	$\sqrt{(2-2)^2 + (10-10)^2} = 0.00$	$\sqrt{(6-2)^2 + (6-10)^2} = 5.39$	$\sqrt{(1.5-2)^2 + (3.5-10)^2} = 8.83$	1
A2	(2, 5)	$\sqrt{(2-2)^2 + (10-5)^2} = 5.00$	$\sqrt{(6-2)^2 + (6-5)^2} = 4.47$	$\sqrt{(1.5-2)^2 + (3.5-5)^2} = 2.50$	3
A3	(8, 4)	$\sqrt{(2-8)^2 + (10-4)^2} = 9.22$	$\sqrt{(6-8)^2 + (6-4)^2} = 3.61$	$\sqrt{(1.5-8)^2 + (3.5-4)^2} = 6.86$	2
A4	(5, 8)	$\sqrt{(2-5)^2 + (10-8)^2} = 3.61$	$\sqrt{(6-5)^2 + (6-8)^2} = 2.24$	$\sqrt{(1.5-5)^2 + (3.5-8)^2} = 6.92$	2
A5	(7, 5)	$\sqrt{(2-7)^2 + (10-5)^2} = 8.06$	$\sqrt{(6-7)^2 + (6-5)^2} = 2.24$	$\sqrt{(1.5-7)^2 + (3.5-5)^2} = 5.92$	2
A6	(6, 4)	$\sqrt{(2-6)^2 + (10-4)^2} = 8.94$	$\sqrt{(6-6)^2 + (6-4)^2} = 2.83$	$\sqrt{(1.5-6)^2 + (3.5-4)^2} = 5.22$	2
A7	(1, 2)	$\sqrt{(2-1)^2 + (10-2)^2} = 8.06$	$\sqrt{(6-1)^2 + (6-2)^2} = 7.21$	$\sqrt{(1.5-1)^2 + (3.5-2)^2} = 1.80$	3
A8	(4, 9)	$\sqrt{(2-4)^2 + (10-9)^2} = 2.24$	$\sqrt{(6-4)^2 + (6-9)^2} = 3.61$	$\sqrt{(1.5-4)^2 + (3.5-9)^2} = 6.80$	1

Iteration 2

Explanation

- **Dist Mean 1**: Calculated using the first centroid ((2, 10)).
- **Dist Mean 2**: Calculated using the second centroid ((6, 6)).
- **Dist Mean 3**: Calculated using the third centroid ((1.5, 3.5)).
- **Cluster**: Points are assigned to the cluster of the closest centroid (smallest distance).

Comments:

- Notice how centroids shift toward the center of their clusters.
- Some points might change clusters as centroids update.

Formula for New Centroids

For each cluster, the new centroid is calculated as the mean of the (x)- and (y)-coordinates of all points in the cluster:

$$\text{Centroid}_i = \left(\frac{\sum x_{\text{cluster}}}{\text{n}}, \frac{\sum y_{\text{cluster}}}{\text{n}} \right)$$

Cluster Assignments

Based on the given table:

- **Cluster 1**: A1, A8 (Coordinates: ((2, 10), (4, 9)))
- **Cluster 2**: A3, A4, A5, A6 (Coordinates: ((8, 4), (5, 8), (7, 5), (6, 4)))
- **Cluster 3**: A2, A7 (Coordinates: ((2, 5), (1, 2)))

Iteration 2

Centroid Calculations

Cluster 1 (Centroid 1):

$$\text{Centroid}_1 = \left(\frac{2+4}{2}, \frac{10+9}{2}\right) = (3.0, 9.5)$$

Cluster 2 (Centroid 2):

$$\text{Centroid}_2 = \left(\frac{8+5+7+6}{4}, \frac{4+8+5+4}{4}\right) \text{Centroid}_2 = \left(\frac{26}{4}, \frac{21}{4}\right) = (6.5, 5.25)$$

Cluster 3 (Centroid 3):

$$\text{Centroid}_3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5, 3.5)$$

Updated Centroids

1. **Centroid 1**: ((3.0, 9.5))
2. **Centroid 2**: ((6.5, 5.25))
3. **Centroid 3**: ((1.5, 3.5))

These centroids are the updated positions for the next iteration of the K-Means algorithm.

Repeat the process until convergence

- Repeat the process and compute the distance to the new centroids and move the points to their new clusters.
- Keep repeating the process until the points stops changing their clusters.