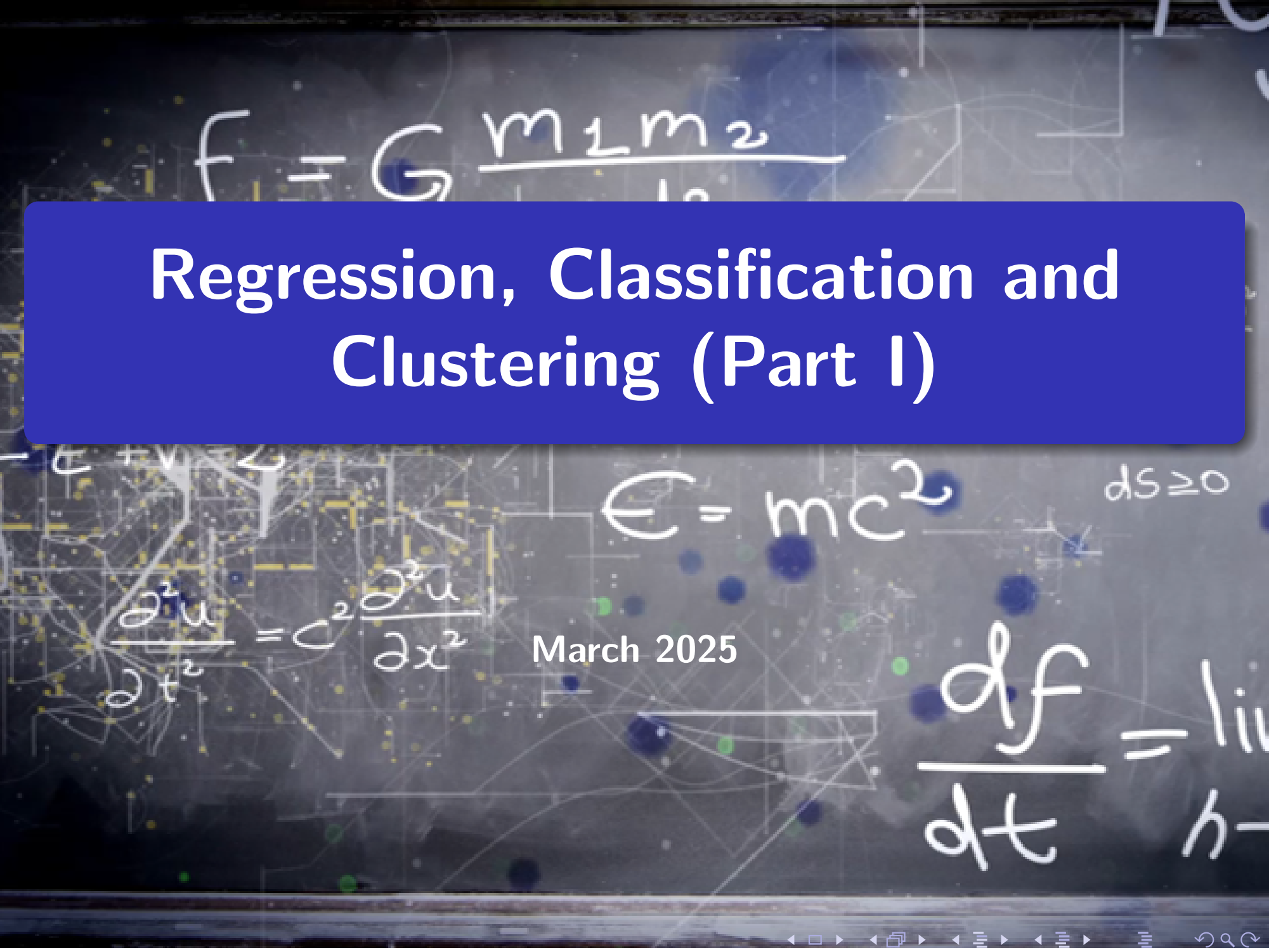


Regression, Classification and Clustering (Part I)

March 2025

Topics to be discussed

Part I. Introduction

1. Introduction to Artificial intelligence

Part II. Search and optimisation.

2. Search - basic approaches
3. Search - optimisation
4. Two-player deterministic games
5. Evolutionary and genetic algorithms

Part III. Machine learning and data analysis.

6. Regression, classification and clustering (Part I & II)
8. Artificial neural networks
9. Bayesian models
10. Reinforcement Learning

Part IV. Logic, Inference, Knowledge Representation

11. Propositional logic and predicate logic
12. Knowledge Representation

Part V. AI in Action: Language, Vision

13. AI in Natural language processing
14. AI in Vision and Graphics

Part VI. Summary

15. AI engineering, Explainable AI, Ethics,

Introduction to Data Analysis

1. Overview of Analytics and Data Analysis
   - Definition and importance of Data Analysis
   - Types of Data Analysis: Descriptive, Inferential, Predictive
   - Types of data analytics
2. Regression Analysis
   - Linear Regression vs. Logistic Regression
3. Classification Analysis
   - Binary and Multi-class Classification
4. Clustering Analysis
   - Hierarchical and Partitioning Clustering
5. Conclusions

Definition and Importance of Data Analysis

Definition of Data Analysis

- The process of cleaning, transforming, and modeling data to extract useful information for decision-making.
- The goal is to discover useful information, draw conclusions, and support decision-making.

Importance of Data Analysis

- Enables informed decision-making by providing insights from data.
- Helps identify patterns, relationships, and trends in data.
- Improves business performance by identifying areas for improvement.
- Helps predict future trends and behaviors.
- Aids in risk management and fraud detection.

Types of Data Analysis: Descriptive, Inferential, Predictive

Descriptive Data Analysis

- Describes and summarizes the main features of a dataset.
- Typically involves measures of central tendency, variability, and frequency distribution.
- Helps in understanding the data and making inferences about it.

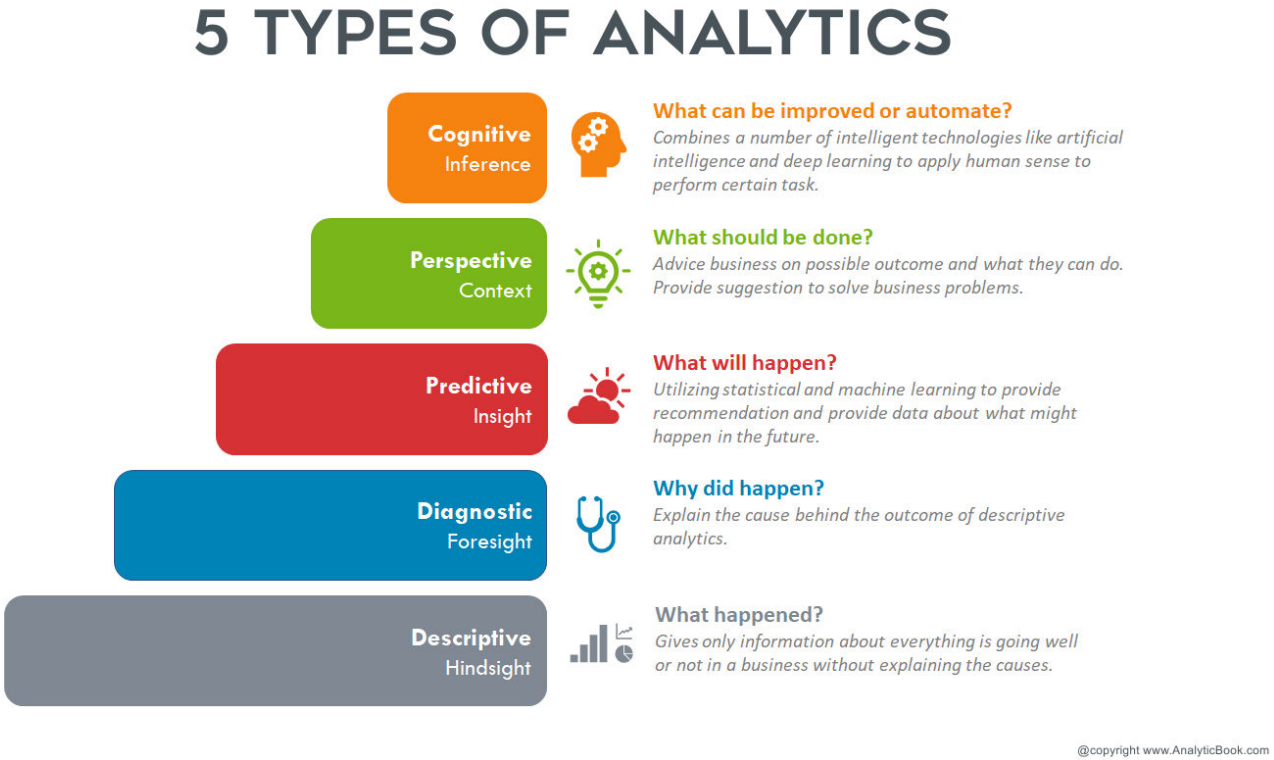
Inferential Data Analysis

- Involves drawing conclusions about a population based on a sample.
- Typically involves hypothesis testing and confidence intervals.
- Helps in generalizing the results to the population from which the sample was taken.

Predictive Data Analysis

- Uses statistical and machine learning techniques to make predictions about future events.
- Involves building models from historical data and using them to predict future outcomes.
- Helps in making informed decisions about future events.

Types of analytics

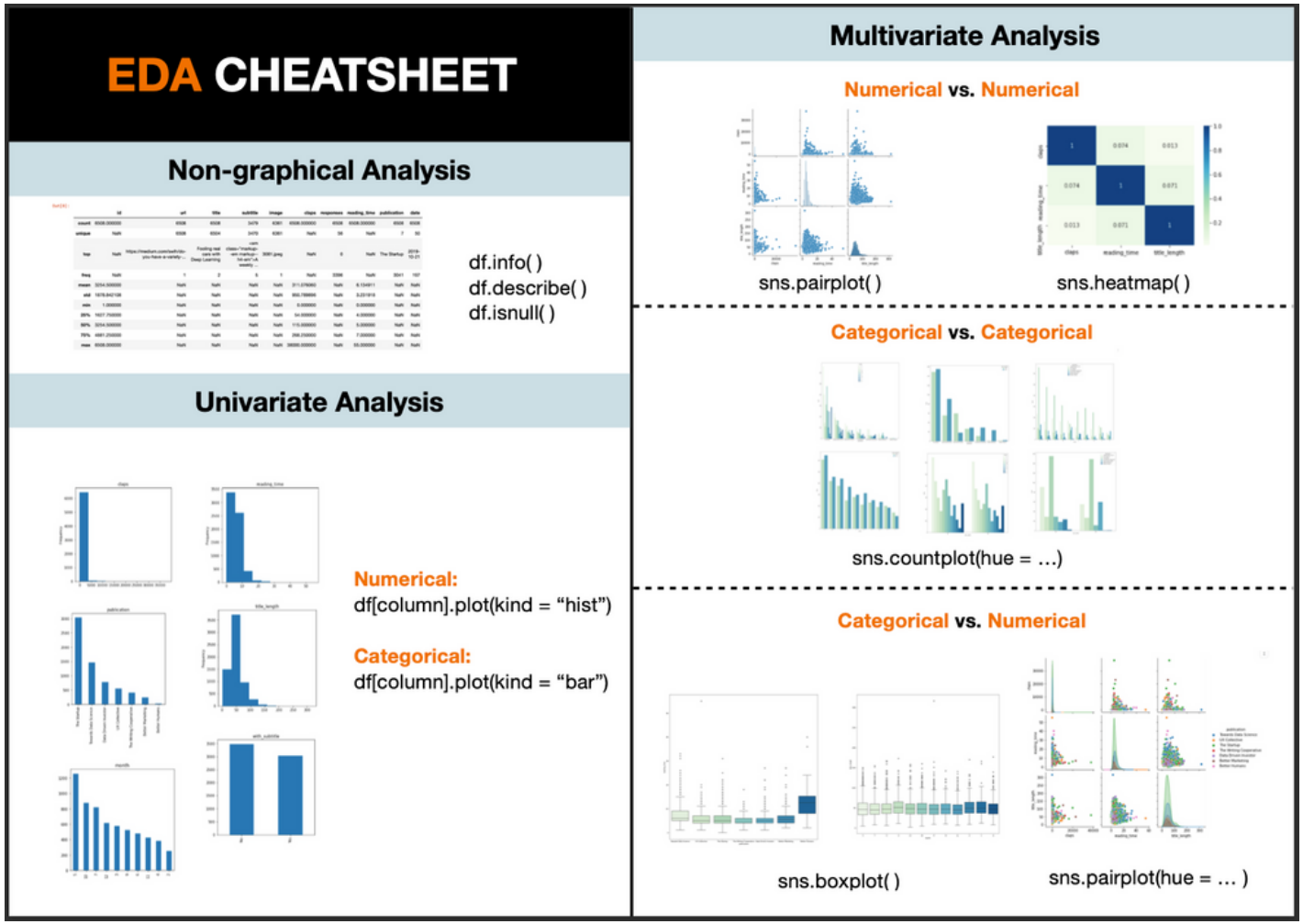


http://www.analyticbook.com/Library?ArticleUId=c81a0c10-2a44-4a77-b6bd-85c35690ae04

Exploratory Data Analysis: Idea and Definition

- **Idea:** The primary goal of EDA is to understand the data sets by summarizing their main characteristics, often with visual methods.
- **Definition:** EDA is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. It allows analysts to uncover patterns, spot anomalies, test a hypothesis, or check assumptions.
- EDA is about making sense of data in hand before making any assumptions.
- Tools used in EDA include: plots, histograms, box plots, scatter plots, and more.

Exploratory Data Analysis: Idea and Definition



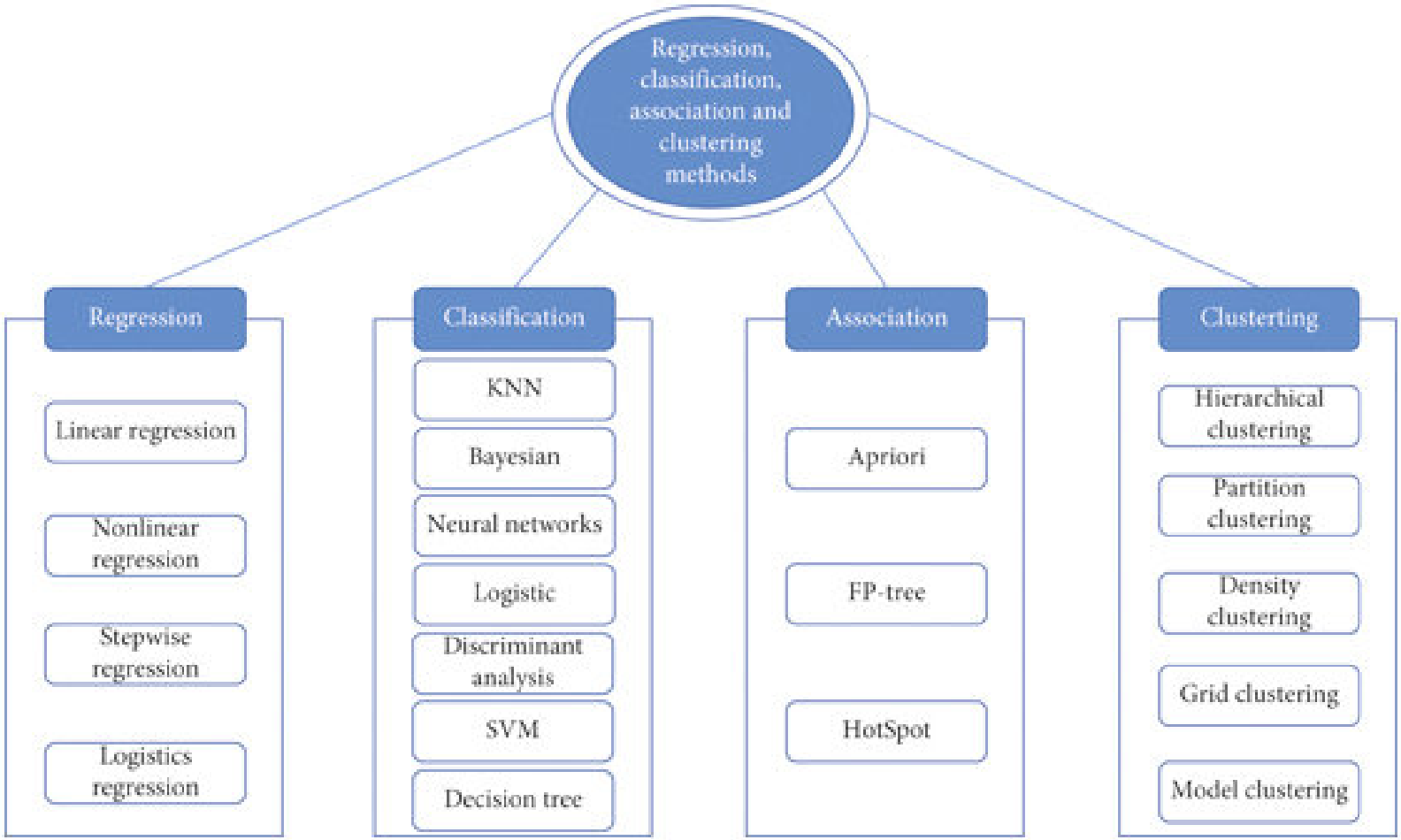
Exploratory Data Analysis: Examples

- **Histograms:** Used to plot the distribution of a numerical variable, understanding the skewness, peaks, and outliers.
- **Scatter Plots:** Explore relationships between two numerical variables; patterns, correlations, or trends might emerge.
- **Box Plots:** Useful for detecting outliers and understanding the distribution and variability of the data.
- **Correlation Matrix:** A table showing correlation coefficients between variables. Each cell shows the correlation between two variables, highlighting potential relationships to investigate further.

Exploratory Data Analysis (EDA) Procedure

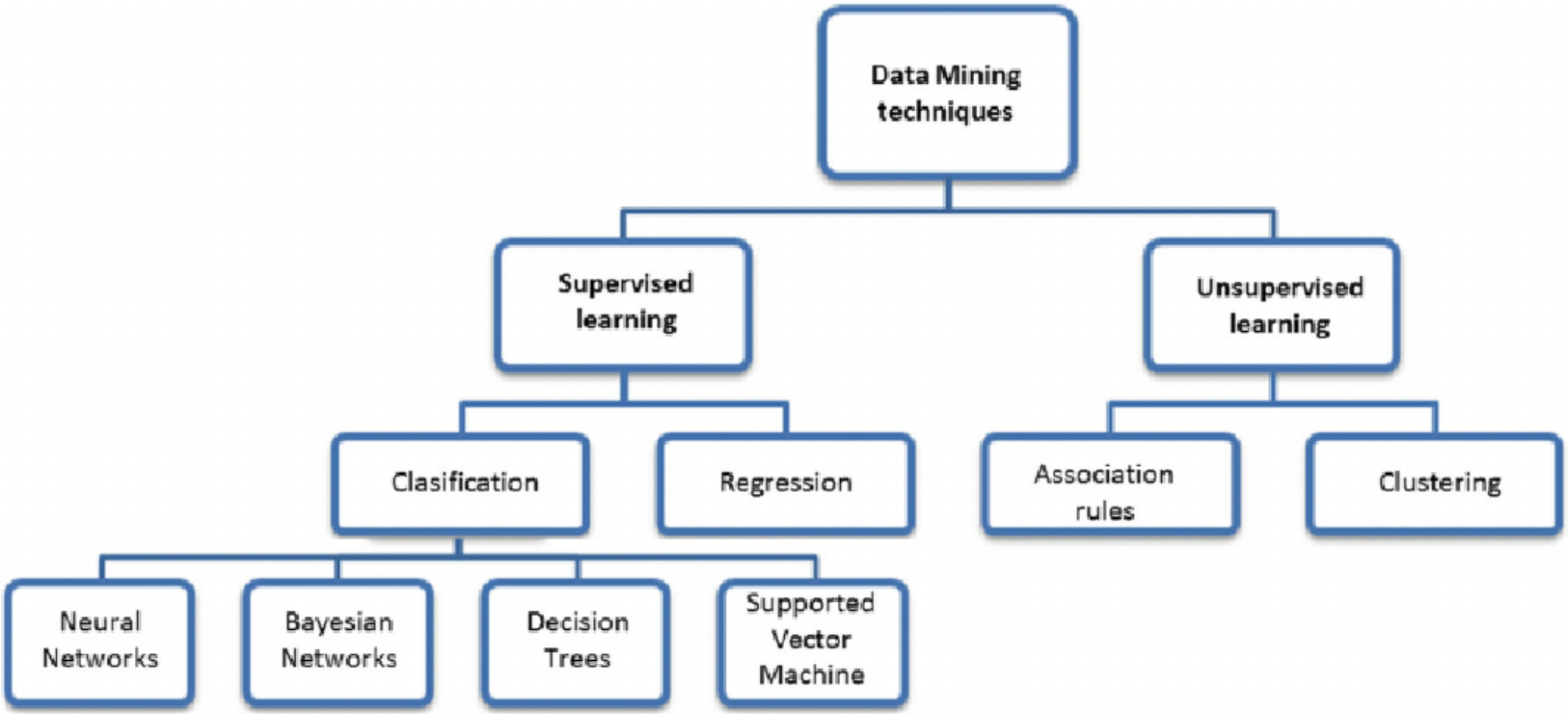
1. **Know Your Data:** Understand dataset characteristics through a preliminary review.
2. **Feature Engineering:** Transform variables to enhance model insight and performance.
3. **Univariate Analysis:**
   - Numerical Data: Use histograms for distribution visualization.
   - Categorical Data: Use bar charts for frequency visualization.
4. **Multivariate Analysis:**
   - Numerical vs. Numerical: Explore with correlation matrices and scatter plots.
   - Categorical vs. Categorical: Understand interactions via grouped bar charts.
   - Numerical vs. Categorical: Analyze distributions with box plots or pair plots with hue.

Regression, Classification, Association and Clustering



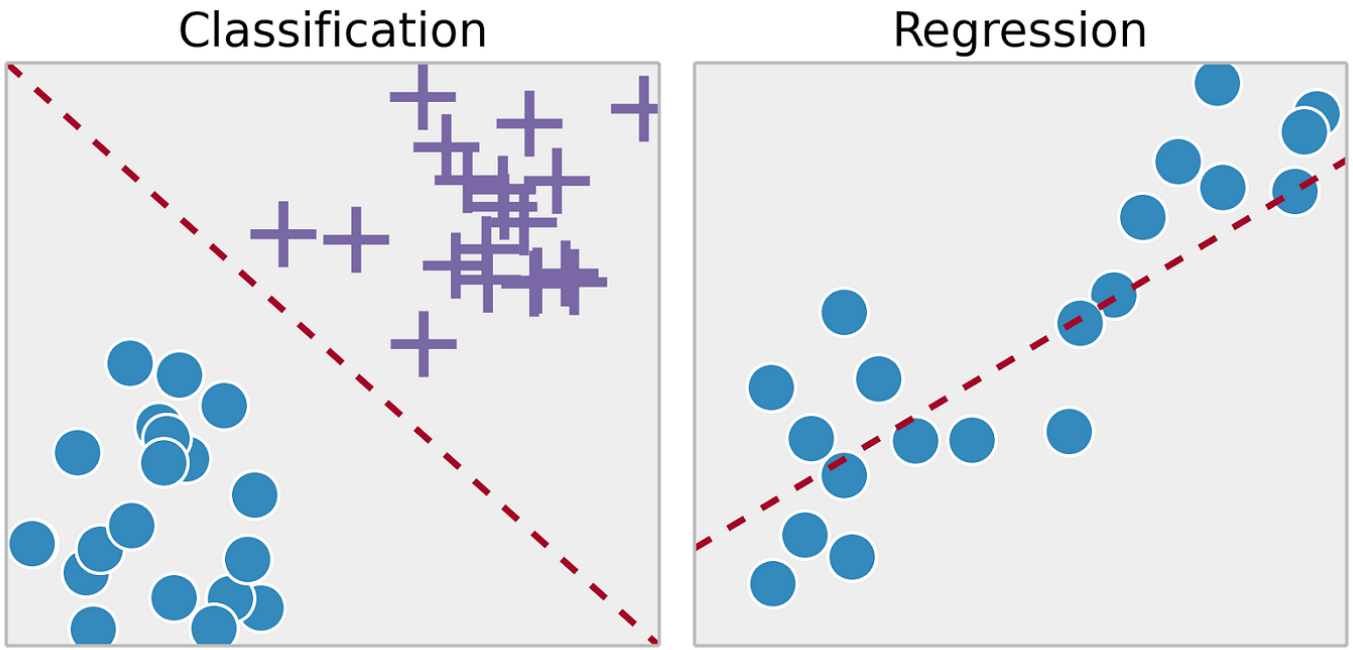
https://www.researchgate.net/publication/361825112

Supervised and Unsupervised Techniques



https://www.researchgate.net/publication/266260663https://pianalytix.com/what-is-data-mining/

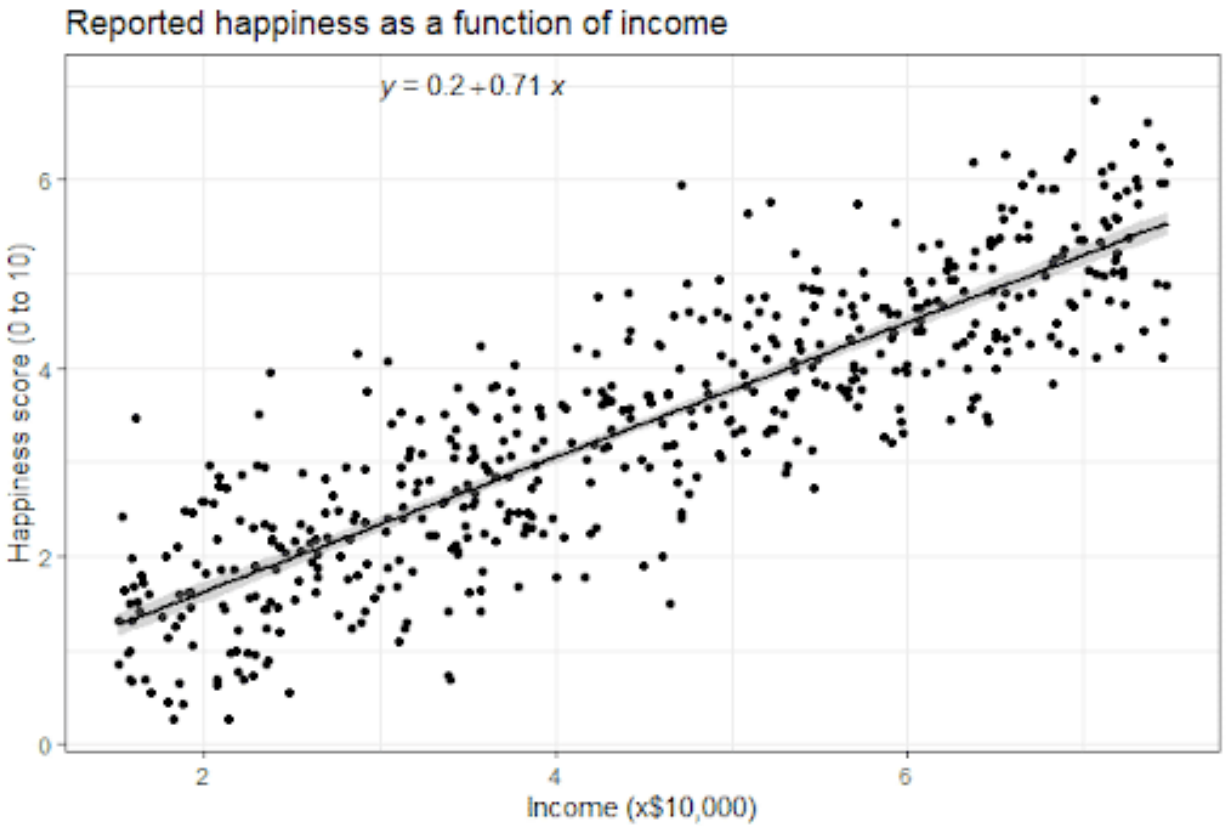
Supervised Learning - example techniques



https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

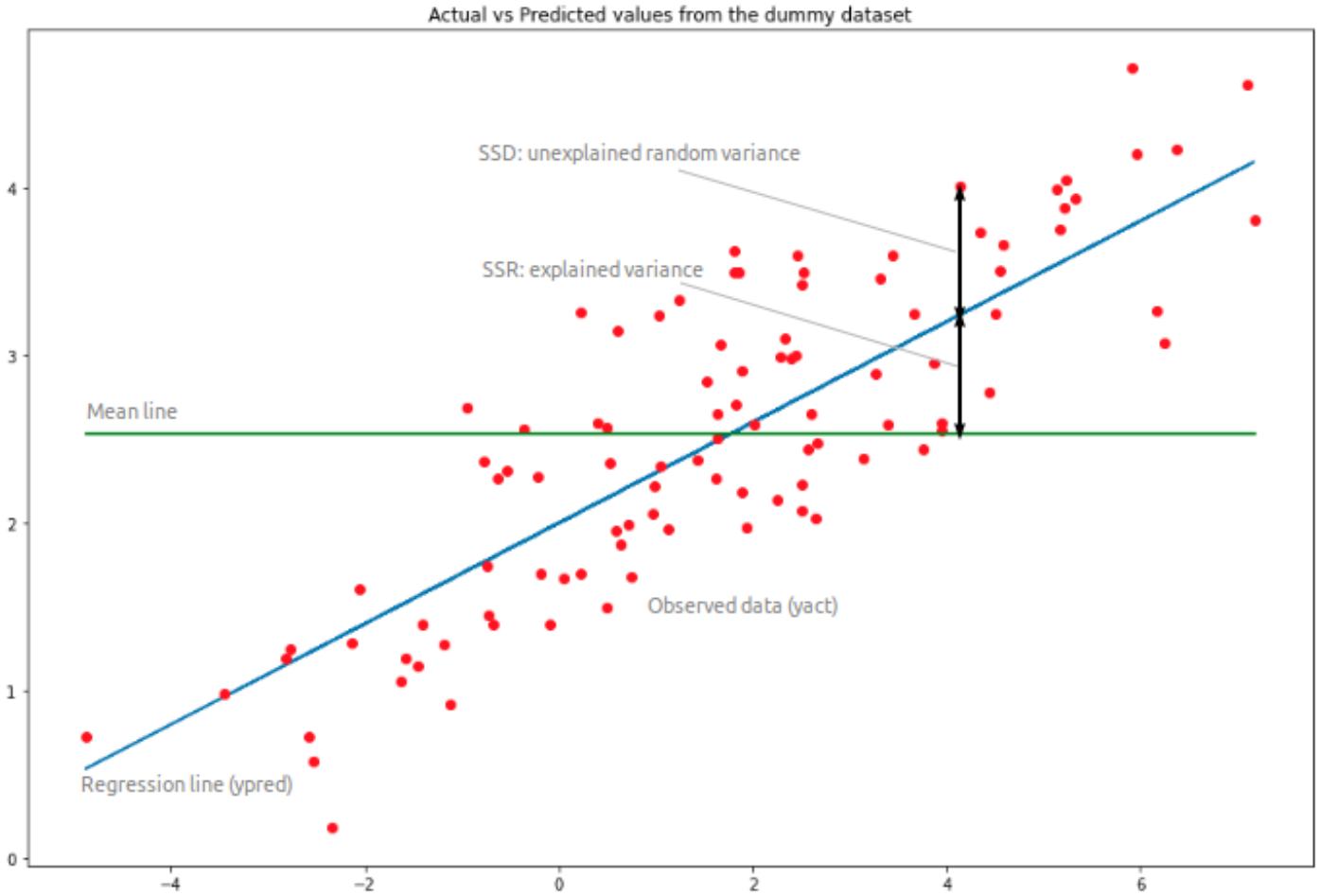
Introduction to Linear Regression

- Regression is a statistical technique used to establish a relationship between a dependent variable and one or more independent variables.
- The goal of regression is to predict the value of the dependent variable based on the values of the independent variables.



https://www.scribbr.com/statistics/simple-linear-regression/

Introduction to Linear Regression (cont.)



https://towardsdatascience.com/introduction-to-linear-regression-in-python-c12a072bedf0

Introduction to Linear Regression (cont.)

Linear regression is a method for modeling the relationship between a dependent variable Y and one or more independent variables $X_1, X_2, ..., X_p$. In its simplest form, linear regression assumes a linear relationship between Y and X:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

where $\beta_0, \beta_1, \beta_2, ..., \beta_p$ are the regression coefficients and ϵ is the error term.

The goal of linear regression is to estimate the values of the regression coefficients that minimize the sum of squared errors between the observed values of Y and the predicted values from the model:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i is the observed value of Y for observation i, $\hat{y}_i$ is the predicted value of Y for observation i, and n is the total number of observations.

Once the regression coefficients are estimated, we can use the model to make predictions for new values of X.

Regression Analysis Metrics

Definitions and Formulas

- **SSD (Sum of Squared Differences)**:

$$SSD = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **SSR (Sum of Squares due to Regression)**:

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

- **SSE (Sum of Squared Errors)**:

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

- **MSE (Mean Squared Error)**:

$$MSE = \frac{SSE}{n}$$

- **RMSE (Root Mean Squared Error)**:

$$RMSE = \sqrt{\frac{SSE}{n}}$$

Definition and Assumptions of Linear Regression

Key Assumptions:

- Linearity: The relationship between Y and X is linear.
- Independence: The observations are independent of each other.
- Homoscedasticity: The variance of the error term ϵ is constant across all values of X.
- Normality: The error term ϵ follows a normal distribution.

Violations of these assumptions can lead to biased or inefficient estimates of the regression coefficients and incorrect conclusions about the relationship between Y and X.

Deriving the Linear Regression Model (least square method)

Algorithm Deriving the Linear Regression Model

Input: Dataset with n observations, each with m predictor variables and a response variable
Output: Linear regression model
for $j \leftarrow 1$ **to** m **do**
    Compute the mean $\bar{x}_j$ and standard deviation s_j of the jth predictor variable
end
Compute the mean $\bar{y}$ of the response variable

for $j \leftarrow 1$ **to** m **do**
    Compute the correlation coefficient r_{yj} between the response variable and the jth predictor variable
    Compute the slope b_j of the regression line between the response variable and the jth predictor variable using $b_j = r_{yj} \cdot (s_y / s_j)$
end
Compute the intercept a of the regression line using $a = \bar{y} - \sum_{j=1}^{m} b_j \cdot \bar{x}_j$

return Linear regression model with coefficients a and $b_1, b_2, \ldots, b_m$

where:

- n: the number of training examples; m predictor variables; a response variable
- x_i: the input variable for the i^{th} training example; y_i: the output variable for the i^{th} training example.
- $\bar{x}$: the mean of the input variabl; $\bar{y}$: the mean of the output variable.
- r_{yj}: the correlation coefficient between the response variable and the jth predictor variable
- s_x^2: the variance of the input variable; s_{xy}: the covariance between the input and output variables.

Deriving the Linear Regression Model (least square method) (cont.)

correlation coefficient:

$$r_{yj} = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x_j})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x_j})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where x_{ij} is the j^{th} feature of the i^{th} input variable and $\bar{x_j}$ is the mean of the j^{th} feature.

The numerator of the formula represents the covariance between x_j and y, while the denominator is the product of their standard deviations.

The resulting coefficient r_{yj} is a value between -1 and 1, where -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship.

Linear Regression Model: Derivation example

Input: Training dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$
Output: Regression coefficients w_0 and w_1

- The linear regression model aims to predict the output variable y from the input variable x using a linear relationship.
- The model is represented by the equation: $y = w_0 + w_1 x$.
- The goal is to find the values of w_0 and w_1 that minimize the difference between the predicted output and the actual output.
- The algorithm for deriving the linear regression model involves computing the mean, variance, and covariance of the input and output variables, and using them to calculate the regression coefficients.

Linear Regression Model: Derivation example (cont.)

Algorithm Linear Regression Model Derivation

Input : Training dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$

Output: Regression coefficients w_0 and w_1

Compute the mean of input and output variables:
$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$

Compute the variance of the input variable:
$s_x^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$

Compute the covariance between the input and output variables:
$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$

Compute the regression coefficient w_1:
$w_1 = \frac{s_{xy}}{s_x^2}$

Compute the regression coefficient w_0:
$w_0 = \bar{y} - w_1 \bar{x}$

One-Dimensional Regression: Example

Suppose we have the following data points:

x	y
1	2
2	4
3	6
4	8
5	10
6	12

First, we calculate the mean and variance of the input variable x and the output variable y:

$$\bar{x} = \frac{1+2+3+4+5+6}{6} = 3.5$$

$$\bar{y} = \frac{2+4+6+8+10+12}{6} = 7$$

$$s_x^2 = \frac{\sum_{i=1}^{6}(x_i - \bar{x})^2}{6-1} = 3.5$$

$$s_{xy} = \frac{\sum_{i=1}^{6}(x_i - \bar{x})(y_i - \bar{y})}{6-1} = 7$$

Next, we can compute the slope and intercept of the regression line:

$$\hat{\beta}_1 = \frac{s_{xy}}{s_x^2} = \frac{7}{3.5} = 2$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 7 - 2(3.5) = 0$$

Therefore, the one-dimensional regression line is given by:

$$\hat{y} = 2x$$

Two-Dimensional Regression Function and Gradients of the Loss Function

- Let X be a two-dimensional input space, where each input $x \in X$ is a vector of input features.
- Let Y be the output space, where $y \in Y$ is a real-valued output.
- We define a simple linear regression function $\hat{f}(x) = a + bx_1 + cx_2$, where a, b, c are real-valued model parameters and x_1, x_2 are the input features.
- The goal is to learn the model parameters a, b, c such that $\hat{f}(x)$ approximates the true function $y = g(x)$, where $g(x)$ is an unknown function that generates the output y for input x.
- The parameters are learned by minimizing the mean squared error loss function $\mathcal{L}(a, b, c) = \frac{1}{4} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)})^2$, where $(x^{(i)}, y^{(i)})$ are the input-output pairs in the training set.
- To use gradient descent, we need to compute the gradients of the loss function with respect to the model parameters a, b, c.
- For our simple linear regression example, the gradients are:
  $$\frac{\partial \mathcal{L}}{\partial a} = \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)}), \quad \frac{\partial \mathcal{L}}{\partial b} = \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)}) x_1^{(i)}, \quad \frac{\partial \mathcal{L}}{\partial c} = \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)}) x_2^{(i)}$$
- These gradients tell us the direction and magnitude of the steepest ascent in the loss function for each parameter. We will use them to update the parameters in each (k+1) iteration of gradient descent.
  $$a_{k+1} \leftarrow a_k - \alpha \frac{\partial \mathcal{L}}{\partial a} = a_k - \alpha \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)})$$
  $$b_{k+1} \leftarrow b_k - \alpha \frac{\partial \mathcal{L}}{\partial b} = b_k - \alpha \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)}) x_1^{(i)}$$
  $$c_{k+1} \leftarrow c_k - \alpha \frac{\partial \mathcal{L}}{\partial c} = c_k - \alpha \frac{1}{2} \sum_{i=1}^{4} (\hat{f}(x^{(i)}) - y^{(i)}) x_2^{(i)}$$
  where α is the learning rate, a hyperparameter that determines the step size of the updates.

Deriving the Linear Regression Model (gradient method)

Algorithm Linear Regression Model (Gradient method)

Data: Training data: $(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})$, Learning rate: α, Number of iterations: *num_iters*

Result: Optimal parameters θ

Initialization: $\theta \leftarrow [0, ..., 0]$

for *iter* $\leftarrow 1$ **to** *num_iters* **do**

- Compute the hypothesis function: $h_\theta(x^{(i)}) = \theta^T x^{(i)}$
- Compute the cost function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- Compute the gradient: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$
- Update parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta)$

Introduction to Classification

What is Classification?

- Classification is a type of supervised learning where the goal is to predict the categorical class labels of new instances, based on past observations.
- The training process involves learning a model that assigns class labels to instances or data points.

Importance of Classification

- Serves as a fundamental approach in machine learning and data mining to solve various predictive modeling problems.
- Widely applied in many fields such as spam detection, sentiment analysis, medical diagnosis, and image recognition.

Overview of Classification Types

- **Binary Classification**: Differentiates between two classes.
- **Multi-class Classification**: Differentiates between more than two classes.
- **Multi-label Classification**: Each instance may belong to multiple classes.

Classification: Binary vs Multi-class

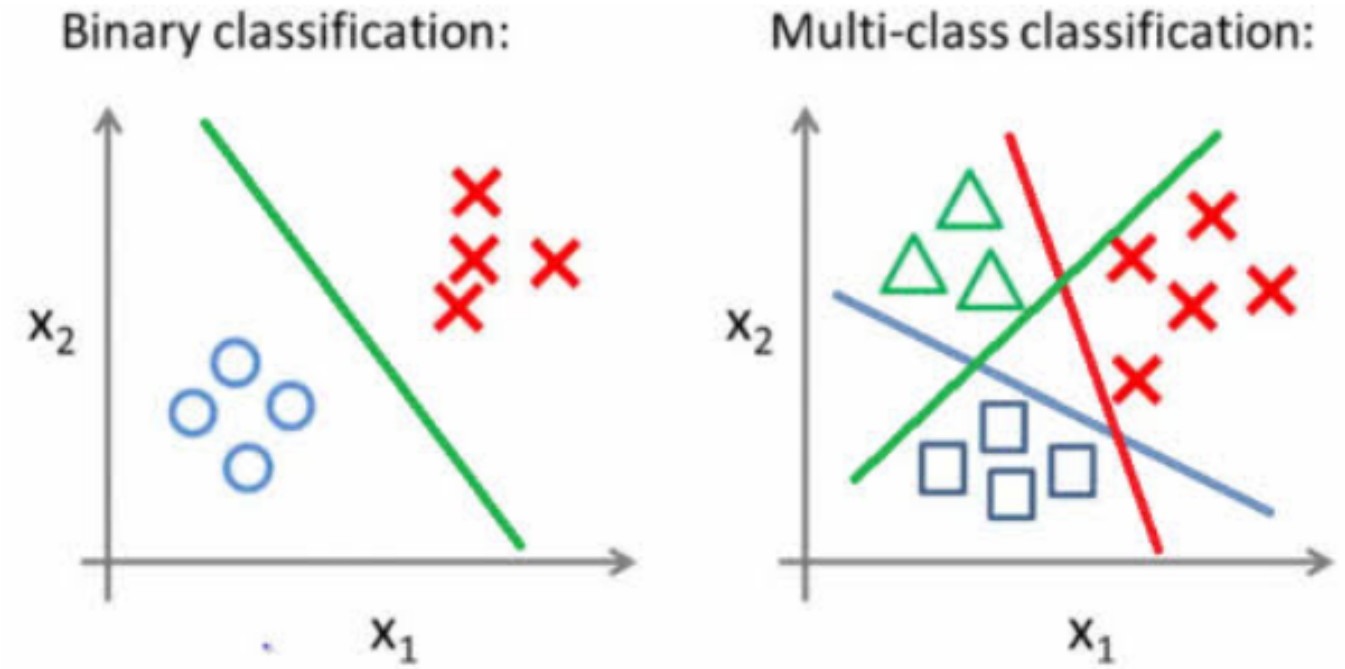
Binary Classification

- In binary classification, the goal is to predict one of two possible outcomes (e.g., yes or no, 0 or 1).
- The output of the classifier is a probability score between 0 and 1, which represents the likelihood of the input belonging to one of the two classes.
- The decision boundary that separates the two classes is typically a straight line or plane.

Multi-class Classification

- In multi-class classification, the goal is to predict one of more than two possible outcomes (e.g., red, green, or blue).
- The output of the classifier is a probability score for each possible class, which represents the likelihood of the input belonging to that class.
- The decision boundary that separates the classes is typically a curved surface.

Classification: Binary vs Multi-class (cont.)

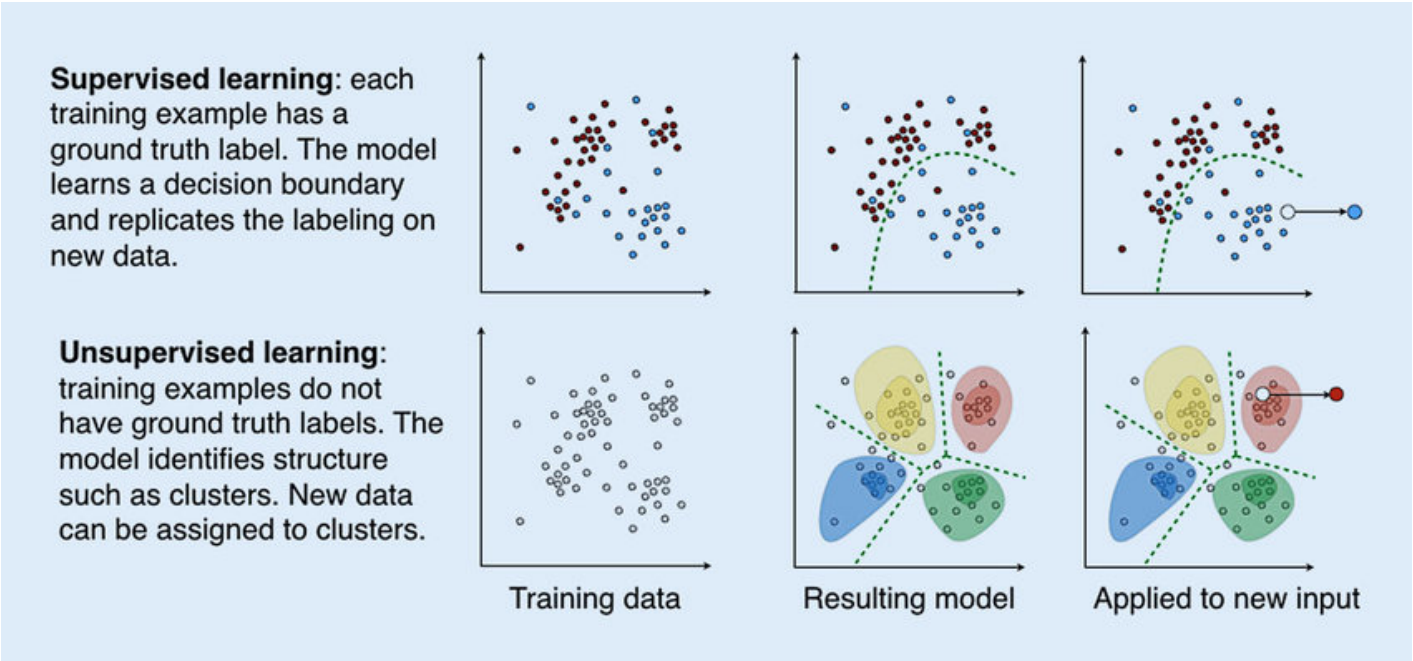


https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b

Types of Classification Algorithms

- Supervised classification algorithms
  1. Logistic Regression
  2. Decision Trees
  3. Naive Bayes
  4. Support Vector Machines (SVMs)
  5. Random Forests
- Unsupervised classification algorithms (Clustering)
  1. K-Means Clustering
  2. Hierarchical Clustering

Supervised vs. Unsupervised Learning



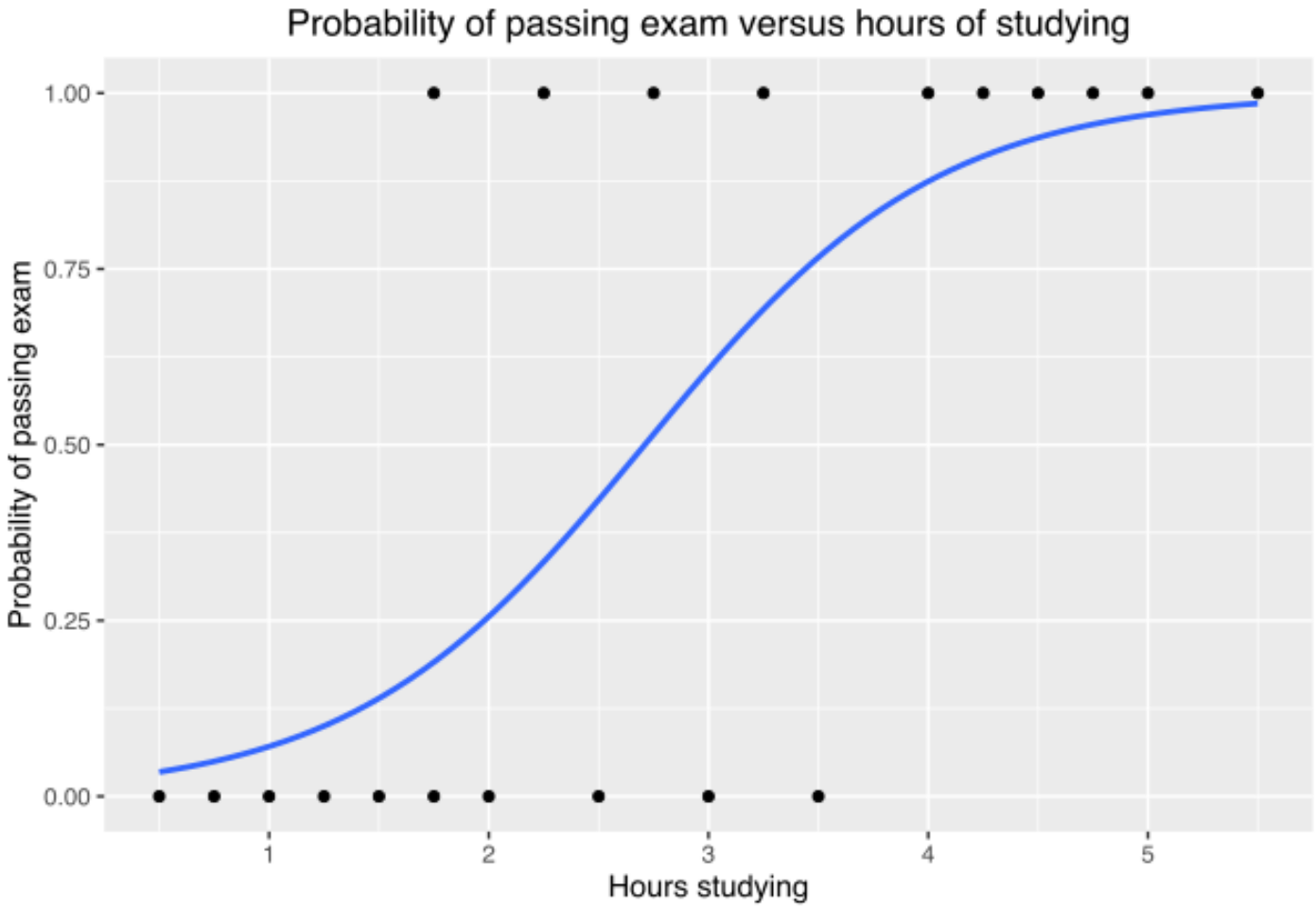
https://www.researchgate.net/publication/325867536

Logistic Regression

Overview:

- Logistic regression is a statistical method used to analyze a dataset in which there are one or more independent variables that determine an outcome.
- The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).
- The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable) and a set of independent (predictor or explanatory) variables.

Logistic regression



https://en.wikipedia.org/wiki/Logistic_regression

Logistic regression

Definitions:

- y: the dependent variable, which is dichotomous (e.g., success/failure, 1/0, true/false).
- $x_1, x_2, ..., x_p$: independent variables or predictors.
- $\beta_0, \beta_1, \beta_2, ..., \beta_p$: coefficients of the logistic regression equation.

Logistic Regression Equation:

The logistic regression equation is:

$$log \frac{P(y=1)}{1-P(y=1)} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$$

where $P(y=1)$ is the probability of the dependent variable y being equal to 1 (i.e., success) divided by the probability of y being equal to 0 (i.e., failure).

Logistic Regression

Solving the Logistic Regression Equation:

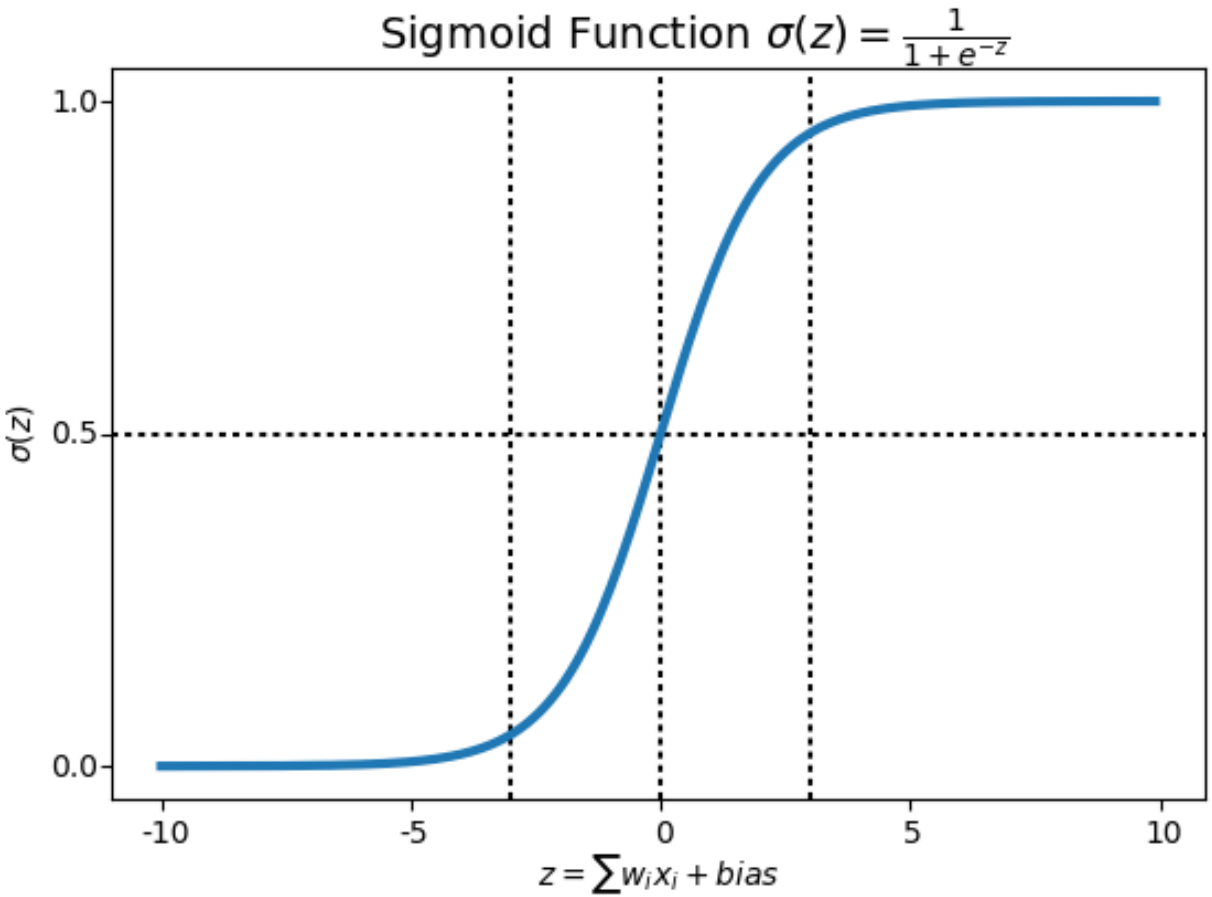
- The logistic regression equation cannot be solved directly, but the coefficients can be estimated using maximum likelihood estimation.
- Maximum likelihood estimation finds the values of the coefficients that maximize the likelihood of observing the data.
- The likelihood function is the joint probability of observing the data given the values of the coefficients.
- The goal is to find the values of the coefficients that make the observed data most likely.

Sigmoid Function:
The sigmoid function is used to convert the linear combination of coefficients and independent variables to a probability of the dependent variable being equal to 1.

$$P(y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p)}}$$

The sigmoid function has an S-shaped curve that ranges from 0 to 1. When the input is large and positive, the output approaches 1. When the input is large and negative, the output approaches 0 ...

Logistic regression sigmoid function



https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

Logistic regression sigmoid function (cont.)

The Sigmoid Function
The sigmoid function is defined as:

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

where z is the linear combination of the input variables $x_1, x_2, ..., x_n$ and their corresponding weights $w_1, w_2, ..., w_n$:

$$z = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n = \sum_{i=0}^{n} w_i x_i$$

where w_0 is the bias term. The bias term shifts the sigmoid function to the left or right, depending on the sign of w_0.

Cross-Entropy Loss Function
The goal of logistic regression is to find the optimal values of the weights w_i that minimize the error in the prediction The error is measured using the cross-entropy loss function:

$$L(w) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where m is the number of training examples, y_i is the true label (either 0 or 1) for the i^{th} example, $\hat{y}_i$ is the predicted label for the i^{th} example, and w is the vector of weights.

Logistic regression sigmoid function (cont.)

Gradient Descent
To minimize the cross-entropy loss function, we use gradient descent to update the weights:

$$w_i = w_i - \alpha \frac{\partial L}{\partial w_i}$$

where α is the learning rate, and $\frac{\partial L}{\partial w_i}$ is the partial derivative of the loss function with respect to the i^{th} weight.

Binary Classification
In practice, logistic regression can be used for binary classification problems, where the goal is to predict one of two classes based on the input variables. The predicted label for a new input is obtained by passing the input through the sigmoid function, and thresholding the result at 0.5.

Logistic Regression: Pseudocode

Algorithm Logistic Regression Algorithm

Data: Training dataset with input features $x_1, x_2, \ldots, x_n$ and binary target variable y

Result: Logistic regression model

initialization: $w_0, w_1, \ldots, w_n$, learning rate α

while *not converged* **do**

calculate the sigmoid function for each training example i:

$\hat{y}_i = \sigma(w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in})$

calculate the cost function for all training examples:

$L(w) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$

update the weights: $w_j = w_j - \alpha \frac{\partial L(w)}{\partial w_j}$

Entropy in AI and Data Science

Definition
Entropy is a measure of the uncertainty or randomness of a system. In the context of machine learning and data science, entropy is often used to quantify the amount of information contained in a dataset or a probability distribution.

- For a discrete probability distribution P, the entropy $H(P)$ is defined as:

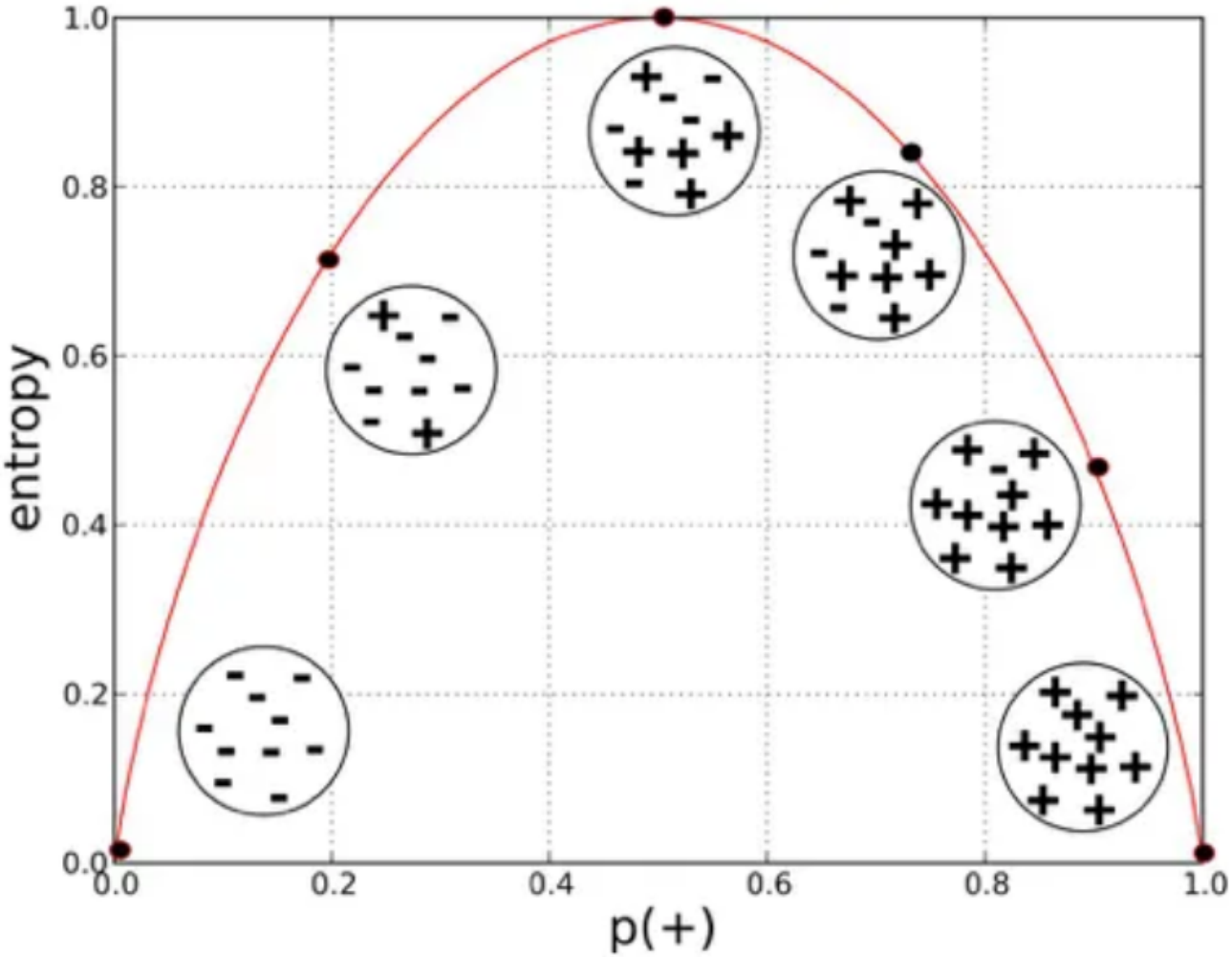
$$H(P) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

  where x_i are the possible outcomes and n is the total number of outcomes.

- For a continuous probability distribution with probability density function $f(x)$, the differential entropy $H(f)$ is defined as:

$$H(f) = -\int_{-\infty}^{\infty} f(x) \log_2 f(x) dx$$

Entropy in AI and Data Science (cont.)



https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Entropy in AI and Data Science (cont.)

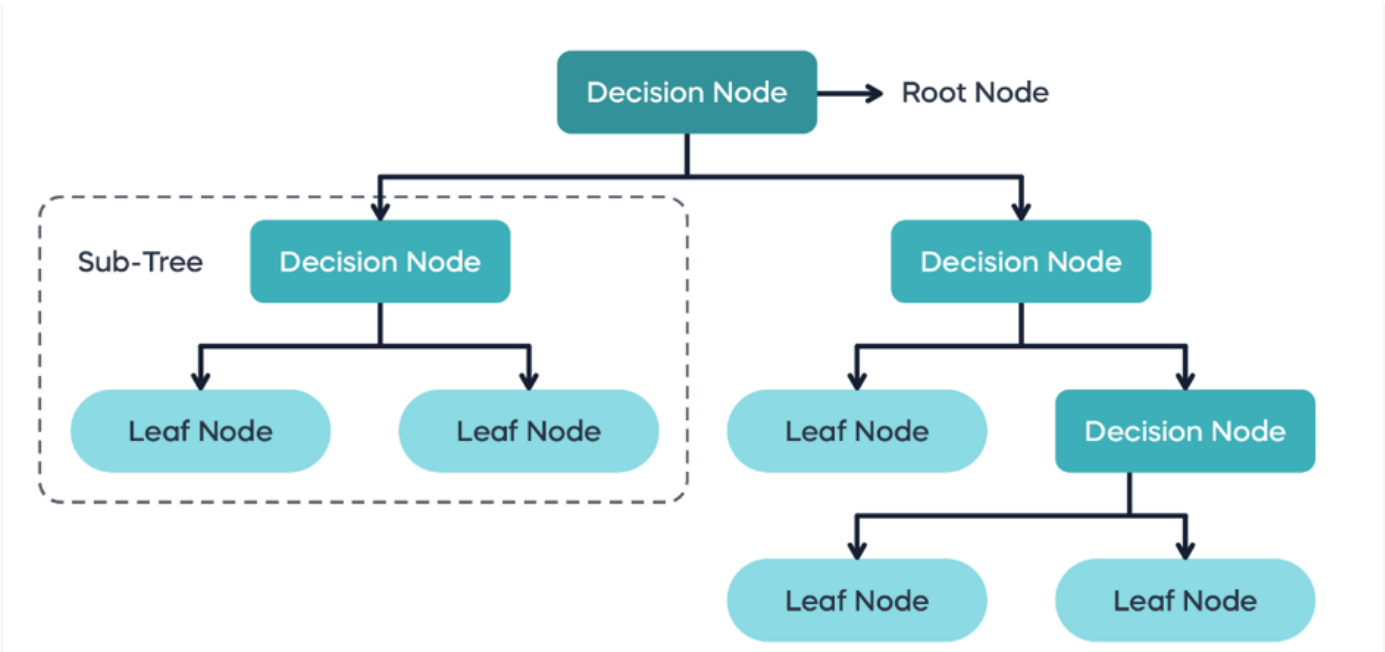
Role in AI and Data Science

Entropy plays a critical role in many areas of AI and data science, including:

- **Decision trees**: Entropy is used as a measure of the purity of a split in a decision tree. The goal is to maximize the information gain, which is the difference between the entropy before and after the split.
- **Clustering**: Entropy is used to measure the quality of a clustering solution. The goal is to minimize the entropy of the clusters, which indicates that the clusters are homogeneous and well-separated.
- **Reinforcement learning**: Entropy is used to balance exploration and exploitation in reinforcement learning algorithms. The goal is to encourage the agent to explore new actions while also exploiting the actions that have been successful in the past.
- **Image compression**: Entropy coding is a technique used in image compression to minimize the number of bits required to represent the image. The most common entropy coding method is Huffman coding, which assigns shorter codes to more frequently occurring symbols.
- **Information retrieval**: Entropy is used to measure the relevance or importance of a document or web page in a search engine, based on the information content of the keywords and the frequency of their occurrence.

Decision Tree - Introduction

- A decision support tool that uses a tree-like graph or model of decisions and their possible consequences.
- Consists of nodes representing decisions or actions, branches representing the possible outcomes, and leaves representing the consequences.
- Can be used for classification, regression, and other prediction tasks.



https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/

Decision Tree - Introduction (cont.)

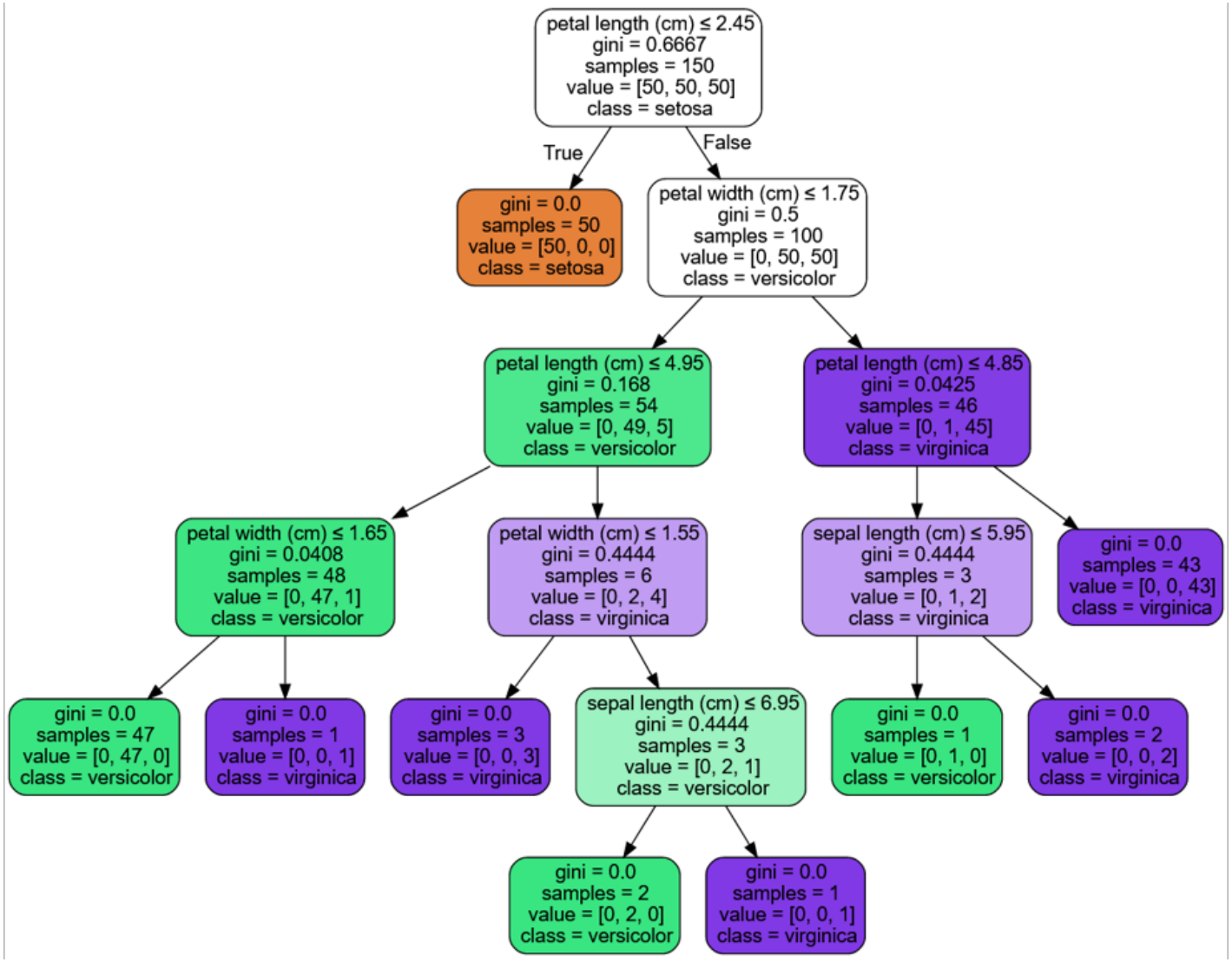
Entropy

- $Entropy = -p_1 * log_2(p_1) - p_2 * log_2(p_2) - ... - p_n * log_2(p_n)$
- Where p_i is the proportion of examples in class i
- Used to decide how to split the data in a decision tree.
- Higher entropy means more randomness or uncertainty.
- Lower entropy means more purity or homogeneity.

Information Gain

- Information gain = entropy(parent) - [weighted average] * entropy(children)
- Used to decide which feature to split on in a decision tree.
- Measures the reduction in entropy achieved by splitting the data on a particular feature.
- Choose the feature with the highest information gain.

Decision Tree: Example



https://scikit-learn.org/stable/modules/tree.html

Decision Tree: ID3 Pseudocode Algorithm

Algorithm ID3 Decision Tree Algorithm

Input: Training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is a feature vector and y_i is a label
Output: Decision tree T

if *all instances in S belong to the same class* **then**
  return *a leaf node with the class label*

if *the feature set is empty* **then**
  return *a leaf node with the majority class label in S*

Select the best feature f to split the data based on some metric, e.g., information gain, Gini index
Divide the training set S into subsets $S_1, S_2, \ldots, S_m$ based on the values of feature f
Create a decision tree node *node* with attribute f
foreach S_i **do**
  Attach a subtree to *node* recursively by calling the algorithm with inputs (S_i, f'), where f' is the best feature to split S_i

return *node*

Decision Tree: Purity and Satisfaction Measures

Purity Measures for A

- Purity measures evaluate the quality of a split based on how well the split separates the classes in the data.
- Examples of purity measures include:
  - **Entropy**: measures the impurity of a node in terms of the probability of each class occurring in that node.

$$H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

    where p_i is the proportion of instances in class i in node S and c is the number of classes in the dataset.
  - **Gini index**: measures the probability of misclassification of a random instance in a node.

$$G(S) = \sum_{i=1}^{c} p_i(1 - p_i)$$

  - **Classification error rate**: measures the proportion of instances in a node that do not belong to the majority class.

$$E(S) = 1 - max_{i=1}^{c} p_i$$

Decision Tree: Purity and Satisfaction Measures (cont.)

Satisfaction Measures for S

- Satisfaction measures evaluate the quality of a tree based on its structure and size.
- Examples of satisfaction measures include:
  - **Maximum depth of the tree**: measures the maximum number of splits from the root to a leaf node.
  - **Minimum number of instances in a leaf node**: specifies the minimum number of instances required in a leaf node.
  - **Maximum number of leaf nodes**: specifies the maximum number of leaf nodes allowed in the tree.

Note:

- Selecting the best attribute A means finding the attribute that maximizes the purity gain or satisfaction improvement.
- S is satisfied when a stopping criterion is met, such as reaching the maximum depth or minimum number of instances in a leaf node.

Algorithm C4.5 Decision Tree Algorithm

Input: Training dataset D with features F, target attribute T, confidence threshold δ
Output: Decision tree T

if *all instances in D belong to the same class c* **then**
  return *a leaf node labeled c*
end
if *F is empty* **then**
  return *a leaf node labeled with the most common class in D*
end
$A_{best} \leftarrow$ attribute with the highest information gain ratio in D with respect to T
if *the information gain ratio of* A_{best} *is less than* δ **then**
  return *a leaf node labeled with the most common class in D*
end
$T \leftarrow$ a new decision tree with root node labeled A_{best}
foreach *possible value* v_i *of* A_{best} **do**
  $D_i \leftarrow$ subset of instances in D with $A_{best} = v_i$
  if D_i *is empty* **then**
    return *a leaf node labeled with the most common class in D*
  end
  $T_i \leftarrow$ C4.5(D_i, $F - \{A_{best}\}$, T, δ)
  add branch from root labeled v_i to subtree T_i
end
return T

ID3 vs. C4.5

- ID3 (Iterative Dichotomiser 3) was the predecessor of C4.5, both are decision tree algorithms for classification.
- C4.5 is an extension of ID3 that addresses some of its limitations.
- Key differences between ID3 and C4.5:
  - Handling of missing attribute values: ID3 cannot handle missing attribute values, while C4.5 can replace missing values with the most common value in the training set.
  - Handling of numerical attributes: ID3 can only handle categorical data, while C4.5 can handle numerical attributes by discretizing them into discrete intervals.
  - Pruning: ID3 does not have a pruning mechanism, while C4.5 uses a pruning algorithm to reduce overfitting and improve generalization.
  - Rule generation: ID3 only generates decision trees, while C4.5 can also generate decision rules.

Introduction to Clustering

What is Clustering?

- Clustering is a type of unsupervised learning technique used to group sets of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups.
- It's about discovering a structure in the data, grouping the data based on similarity, without pre-labeled responses or outcomes.

Goals of Clustering

- To discover the inherent groupings in the data.
- To understand the structure of the data by categorizing it into clusters.
- To simplify further analysis by segmenting the dataset into manageable groups.

Applications of Clustering

- Market segmentation to find customer groups with similar preferences.
- Organizing computing clusters for efficient data processing.
- Social network analysis to identify communities within larger networks.
- Image segmentation in computer vision.

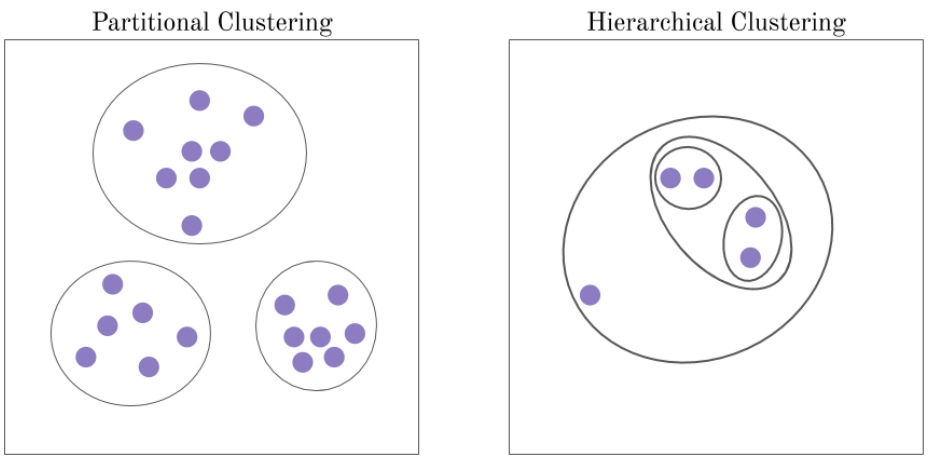
Clustering

Hierarchical Clustering

- Divides the data into a tree of clusters, with the leaves being individual points.
- Two types: Agglomerative and Divisive.
- Requires a linkage criterion to determine the distance between clusters.
- Produces a dendrogram that shows the hierarchy of clusters.

Partitioning Clustering

- Divides the data into non-overlapping partitions or clusters.
- Two popular algorithms: K-Means and PAM (Partitioning Around Medoids).
- Requires a distance metric to determine the similarity between data points.
- The number of clusters must be specified ahead of time.



https://towardsdatascience.com/a-brief-introduction-to-unsupervised-learning-20db46445283

Partitioning Clustering: K-Means Clustering

- **K-Means Clustering** is a popular unsupervised machine learning algorithm used for clustering data points into groups or clusters.
- The algorithm works by iteratively partitioning the data into K clusters, where K is a pre-defined number chosen by the user.
- The objective of K-Means Clustering is to minimize the sum of squared distances between each data point and its assigned cluster centroid.

K-Means Clustering: Key Formulas

- The distance between a data point x_i and a cluster centroid c_j is typically computed using the Euclidean distance formula:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^{m} (x_{i,k} - c_{j,k})^2}$$

where m is the number of features in the data and $x_{i,k}$ and $c_{j,k}$ are the kth features of data point x_i and centroid c_j, respectively.

- The algorithm updates the cluster centroids using the following formula:

$$c_j = \frac{1}{n_j} \sum_{i=1}^{n} z_{i,j} x_i$$

where n_j is the number of data points in cluster j, n is the total number of data points, and $z_{i,j}$ is a binary indicator variable that equals 1 if data point i is assigned to cluster j and 0 otherwise.

K-Means Clustering: Pseudocode Algorithm

Algorithm K-Means Clustering Algorithm

Input: Data set $X = \{x_1, x_2, \ldots, x_n\}$, number of clusters K
Output: Cluster assignments for each data point

Randomly initialize K cluster centroids $\{c_1, c_2, \ldots, c_K\}$
repeat
Assign each data point x_i to the nearest centroid c_j:

$$z_{i,j} = \begin{cases} 1 & \text{if } j = \arg\min_k d(x_i, c_k) \\ 0 & \text{otherwise} \end{cases}$$

Update the centroids:

$$c_j = \frac{1}{n_j} \sum_{i=1}^{n} z_{i,j} x_i$$

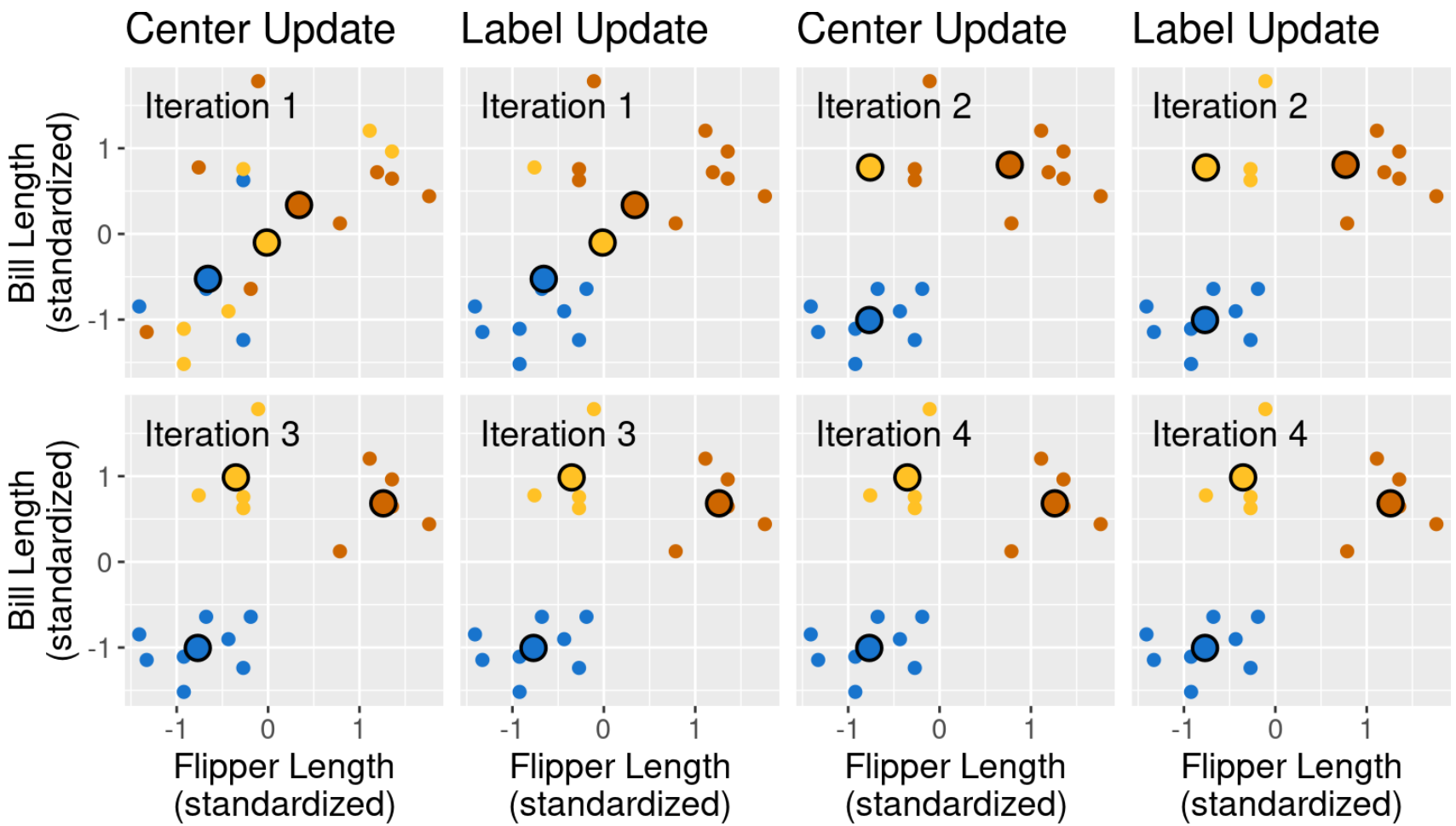
until *convergence*
return *Cluster assignments* $\{z_1, z_2, \ldots, z_n\}$

Note: $d(x_i, c_k)$ represents the distance between data point x_i and cluster centroid c_k. The most commonly used distance metric is Euclidean distance, defined as:

$$d(x_i, c_k) = \sqrt{\sum_{m=1}^{M} (x_{i,m} - c_{k,m})^2}$$

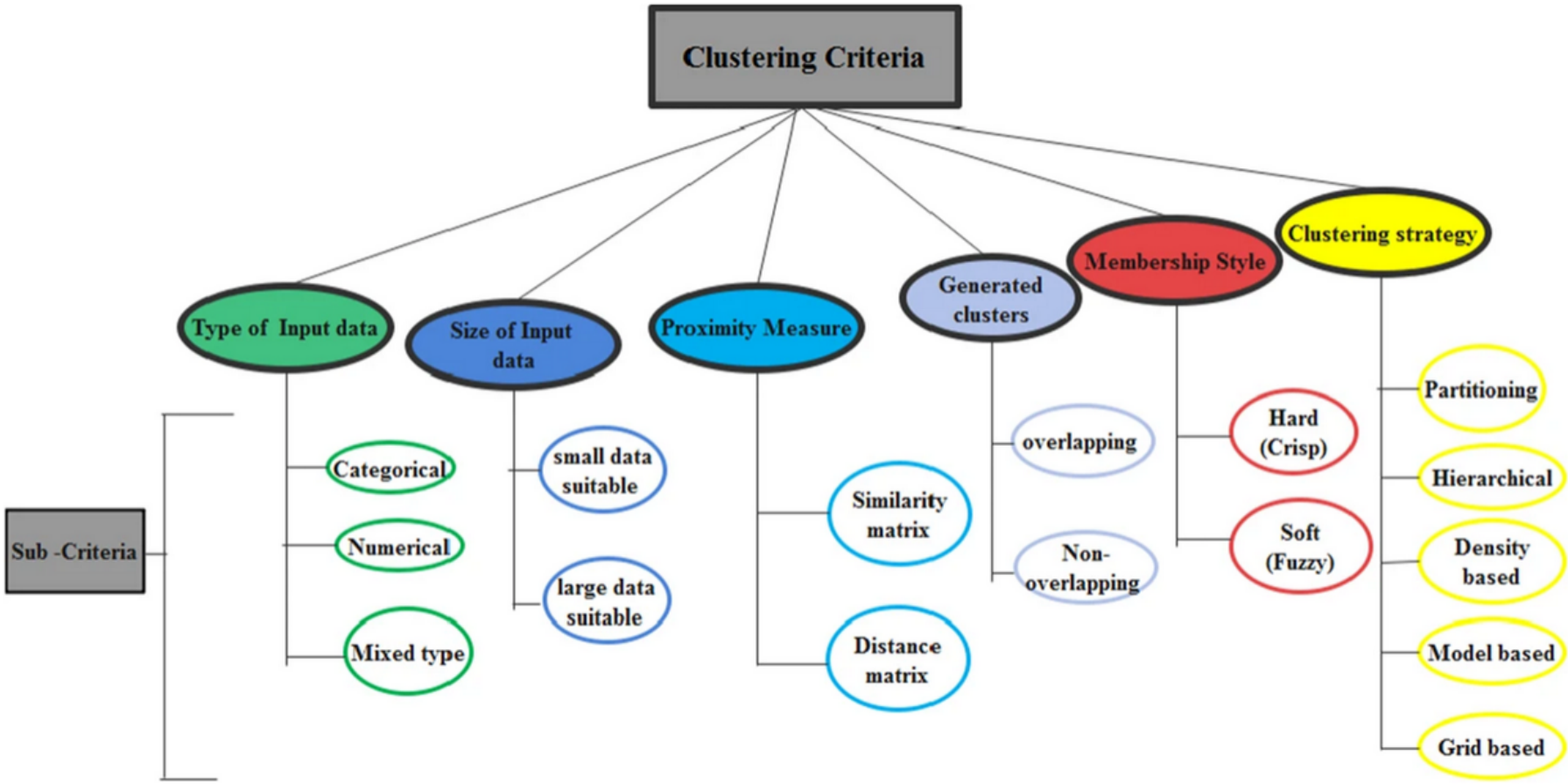
where M is the number of dimensions in the data and $x_{i,m}$ and $c_{k,m}$ are the m-th dimension values for data point x_i and cluster centroid c_k, respectively.

K-Means Clustering: Examples of iterations



https://datasciencebook.ca/clustering.html

Clustering Criteria



https://link.springer.com/article/10.1007/s10462-022-10325-y

Summary - Data Analysis (Part I)

1. Overview of Analytics and Data Analysis
2. Linear Regression Analysis
3. Logistic Regression Analysis
4. Classification Analysis
5. Clustering Analysis

References

1. S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach", Financial Times Prentice Hall, 2019.
2. T. Nield, "Essential Math for Data Science: Take Control of Your Data with Fundamental Linear Algebra, Probability, and Statistics", O'Reilly Media, 2022
3. A. Geron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, 3rd ed., 2023
4. M. Flasiński, Introduction to Artificial Intelligence", Springer Verlang, 2016
5. M. Muraszkiewicz, R. Nowak (ed.), Sztuczna Inteligencja dla inżynierów", Oficyna Wydawnicza PW, 2022
6. J. Prateek , Artificial Intelligence with Python", Packt 2017