| Exercise No. | Points |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 3 | 6 |
| 4 | 6 |
| Total: | 20/20 |

# Ex 1: General definitions and concepts

1. *What are the main differences between Data Science and Machine Learning, and how do their goals and functionalities differ, even though they are closely linked?* (**1 pt**)

   - **Goals:** DS uncover insights for decision-making (**0.25 pt**), while ML develops predictive models (**0.25 pt**).
   - **Functionality:** DS has broader activities from data collection to visualization (**0.25 pt**), while ML is specialized in building and training predictive models (**0.25 pt**).

2. *Enumerate the steps involved in the data science lifecycle, highlighting the key tasks and considerations at each stage?* (**3 pts**)

   - **Business requirements:** identify the business problem or question to be addressed, the objectives and success criteria. (**0.25 pt + 0.25 pt**)
   - **Data Acquisition:** collect relevant data from various sources. (**0.25 pt + 0.25 pt**)
   - **Data Processing:** preprocess the data, perform feature engineering. (**0.25 pt + 0.25 pt**)
   - **Data Exploration:** explore the data to understand its structure, patterns, and relationships, visualize it using graphs and charts to gain further insights. (**0.25 pt + 0.25 pt**)
   - **Model Development:** select ML algorithms, evaluate their performance, compare them to select the best-performing one. (**0.25 pt + 0.25 pt**)
   - **Deployment and optimization:** Deploy the chosen model into production systems for real-world use, integrate the model with existing infrastructure and software. Update and retrain the model periodically with new data to maintain its relevance and accuracy. (**0.25 pt + 0.25 pt**)

# Ex 2: Similarity between objects

Consider two objects represented by binary strings: Object A = "110010" and Object B = "101011".

1. *Define and calculate the Hamming distance between* A *and* B. *Can we use the Hamming distance if* A *and* B *had different lengths and why?* (**1 pt**)

   - Hamming distance: $d_H(A, B) = \sum_{i=1}^{n} |A_i - B_i|$ (**0.25 pt**)
   - To calculate $d_H(A, B)$ we compare corresponding elements of the two strings and count the number of positions where they differ: $d_H(A, B) = 3$ (**0.25 pt**)
   - If the lengths of the strings are not the same, there won't be a one-to-one correspondence between elements, and thus the concept of "corresponding elements" doesn't apply. (**0.5 pt**)

2. *Under which conditions a distance measure is considered as a metric? Demonstrate that Hamming distance is a metric (provide proofs to support each part of the proof)* (**3 pts**)

   - A distance measure is considered a metric if it satisfies the following properties:
     (a) **Non-negativity:** the distance between any two points is non-negative. $d_H(A, B)$ is the count of positions at which the corresponding symbols differ between two strings. Since we are counting the differences, the distance is always non-negative. (**0.25 pt + 0.25 pt + 0.25 pt**)
     (b) **Identity of Indiscernible**: The distance between two points is zero if and only if the points are identical. (**0.25 pt + 0.25 pt + 0.25 pt**)
     (c) **Symmetry**: The distance from point A to point B is the same as the distance from point B to point A. In $d_H(A, B)$, the order of comparison doesn't affect the count of differing symbols. In $d_H(A, B)$ if two strings are identical, there are no positions at which the symbols differ. Therefore, $d_H(A, B) = 0$. (**0.25 pt + 0.25 pt + 0.25 pt**)
     (d) **Triangle Inequality**: The distance from point A to point C is always less than or equal to the sum of the distances from point A to point B and from point B to point C. Let's consider three strings $A$, $B$, and $C$ of the same length. By the triangle inequality, we have $d_H(A, C) \leq d_H(A, B) + d_H(B, C)$ (**0.25 pt + 0.25 pt**)

     Demonstration: (**0.25 pt**)

     $$A = \text{"110010"}, B = \text{"101011"}, C = \text{"100111"}$$

     $$d_H(A, B) = 3, d_H(B, C) = 2, d_H(A, C) = 3$$

     $$3 \leq 3 + 2$$

     Therefore, $d_H$ satisfies all the properties of a metric, making it a valid metric for measuring dissimilarity between binary strings.

# Ex 3: Unsupervised learning - Clustering

1. *What are the hyper-parameters of k-means clustering method and how do we set them up?* (**4 pts**)

   - Number of clusters k: determine it using domain knowledge or techniques like the elbow method. (**0.5 pt + 0.5 pt**)
   - Centroid initialization: use k-means++ by default, experiment if necessary. (**0.5 + 0.5 pt**)
   - Maximum number of iterations: choose a large enough value to ensure convergence without unnecessary runtime. (**0.5 pt + 0.5 pt**)
   - Convergence tolerance: choose a small enough value to ensure precision without excessive computation. (**0.5 pt + 0.5 pt**)

2. *In hierarchical agglomerative clustering, how would you determine the optimal number of clusters without relying on pre-defined stopping criteria?* (**1 pt**)

   - Visually inspecting the dendrogram representing the clustering process and identifying the point at which the clusters seem to be most cohesive or meaningful (dendrogram cutting).
     Validate the clustering solution using internal/external validation metrics, internal such as silhouette score or external if ground truth labels are available.

3. *If you set a large value for $\epsilon$ in DBSCAN, what would be the potential consequences on the clustering results?* (**1 pt**)

   - It cause the algorithm to produce fewer, larger clusters, potentially merging together distinct clusters and increasing sensitivity to noise.

# Ex 4: Supervised learning - Regression

1. *Discuss the difference between simple linear regression and multiple linear regression.* (**2 pts**)

   - In SLR models, the relationship between the independent variable $x$ and the dependent variable $y$ is modeled as a straight line, described by the equation $y = \beta_0 + \beta_1 x + \epsilon$. (**1 pt**)
   - In MLR models, the relationship between the multiple independent variables $x_1, x_2, \ldots, x_n$ and the dependent variable $y$ is modeled as a linear combination, described by the equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$. (**1 pt**)

2. *Describe the process by which gradient descent is employed to refine the parameters of a linear regression model.*(**2 pts**)

   (a) Initialize the coefficients of the linear regression model to some arbitrary values (randomly or set to zero).

   (b) Define a function $J(\theta)$ that quantifies the error between the actual values of the dependent variable and the predicted values by the linear regression model such as Mean Squared Error (MSE), defined as:

   $$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

   (c) Calculate the gradient of the cost function with respect to each parameter $\theta_j$

   $$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

   (d) Update each parameter $\theta_j$ using the gradient descent update rule:

   $$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

   (e) Repeat steps (c) and (d) until convergence.

   **Simplified answer:** By iteratively updating the parameters of the linear regression model in the direction of the steepest decrease in the cost function, gradient descent effectively refines the model to minimize prediction error and improve its fit to the data.

3. *How does regularization help in addressing the challenges associated with employing polynomial regression models, particularly in mitigating overfitting and controlling model complexity?* (**2 pts**)

   - L1 regularization encourages sparsity in the coefficient values, effectively shrinking some coefficients to zero and eliminating irrelevant features. By doing so, L1 prevent overfitting by reducing model complexity as it keeps only the most important features.
   - L2 regularization encourages small but non-zero coefficients for all features, effectively reducing the impact of irrelevant features on the model. This helps to prevent overfitting by reducing the sensitivity of the model to noisy or irrelevant features.
   - By controlling the coefficients sizes, regularization prevent the model from fitting the training data too closely and capturing noise or random fluctuations, promoting better generalization performance on unseen data and helps to produce simpler, more interpretable models.
   - **Simplified answer:** regularization techniques like Lasso and Ridge add penalty terms to the cost function in polynomial regression. They mitigate overfitting by shrinking coefficients, thus controlling model complexity and promoting generalization performance.