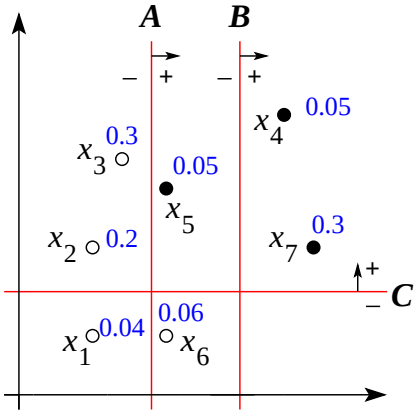
Învăţare automată

— Licenţă, anul III, 2020-2021, examenul parţial II —

Nume student: **Grupa:**

LC.1 (0.6p) **(Algoritmul AdaBoost: exemplu de aplicare pe date din $\mathbb{R}^2$; întrebări calitative)**

In the nearby figure, $\circ$ points correspond to negative examples ($y_i = -1$) and *bullet* points are positive examples ($y_i = +1$). The figure also shows the normalized weights (LC: i.e., probabilities) on the examples resulting from having run the AdaBoost algorithm for some number of iterations. There are also also three decision stumps drawn in the figure, $h(x;\theta_A)$, $h(x;\theta_B)$, and $h(x;\theta_C)$ or A, B and C for short.



a. Which one of the stumps was used at the previous iteration to obtain the weights on the examples shown in the figure (please answer A, B, or C)? Briefly justify your answer.

b. Which one of the stumps would you use at the next iteration (please answer A, B, or C)? Briefly justify your answer.

c. In above figure, circle the training point(s) (possibly none) that the ensemble $H_2(x) = sign(\alpha_A h(x;\theta_A) + \alpha_C h(x;\theta_C))$, with $\alpha_A = 0.3$ and $\alpha_C = 0.5$ cannot classify correctly.

LC.2 (0.7p) **(Concepte din $\mathbb{R}$ *reprezentabile* cu ajutorul combinaţiilor liniare de compaşi de decizie)**

Suppose you want to classify points on the real axis: each sample x_i is a real number, and the labels you want to predict are binary: $y_i \in \{-1, +1\}$. In this problem, you will use ensembles, i.e., linear combinations of weak hypotheses / separators (but, you will NOT need to use AdaBoost!). Recall that your classifier takes this form:

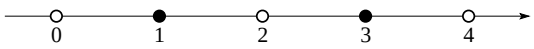
$$\hat{y} = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big),$$

where $\hat{y}$ is your predicted label, $sign(x)$ is $+1$ if $x > 0$ and is -1 otherwise, α_t is a real-valued weight, and $h_t(x)$ is the prediction made by weak hypothesis h_t. Each h_t takes one of the following forms:

$$h_t(x;s,+) = \begin{cases} -1 & \text{if } x \le s \\ +1 & \text{if } x > s \end{cases} \qquad h_t(x;s,-) = \begin{cases} +1 & \text{if } x \le s \\ -1 & \text{if } x > s, \end{cases}$$

for a given (fixed) *split threshold* $s \in \mathbb{R}$.[1]

Let us consider the following dataset, made of 5 instances on the real axis:



a. Show that this data set (LC: concept) is representable with 4 weak hypotheses. Write out the 4 weak hypotheses $h_1, \ldots, h_4$, as well as their weights $\alpha_1, \ldots, \alpha_4$, explicitly.
Describe this concept with a well-labeled drawing.

***Note*: Due to the way the function *sign* was defined above, the *prediction rule* will broke ties by labeling instances -1 when the weighted sum of weak hypothesis predictions is 0; this tie-breaking is useful in answering the above question.**

b. Prove that this data set is NOT representable with fewer than 4 weak hypotheses.

c. Generalize the result obtained at part a to the "representability" (using linear combinations of weak classifiers) of arbitrary data sets on the real axis.

[1] Note that the definition given here for the notion *weak hypothesis* corresponds to the well-known *decision stump*, particularized for data from $\mathbb{R}$.

LC.3 (1p) **(Algoritmul K-means: aplicare pe date din $\mathbb{R}^2$ monotonia criteriului B)**

A	1	1
B	3	3
C	6	6
D	6	12
E	9	9
F	11	11
G	0	3
H	3	0
I	9	3

You are given a dataset in the nearby table, where the rows represent a single data point.

a. Perform two iterations of the K-means algorithm on this dataset with $K = 3$, using the Euclidian distance as the distance function.

The clusters are initialized as follows:
$C_1^{(0)} = \{A, B, F\}$, $C_2^{(0)} = \{C, H, I\}$, $C_3^{(0)} = \{D, E, G\}$.

You will represent the data on the following grids.

***Requirement*:**
In those situations when the *geometrical method* [based on representing the mediators determined by pairs of centroids] is not very concludent with respect to the assignation of an instance to a certain cluster / centroid, you *must* use the *analytical method* (based on effectively calculating the distances) in a rigorous manner.

b. Let $\{x_1, \ldots, x_n\}$ be a set of instances to be clusterized using the K-means algorithm. We define

$$J(\mu^{(t)}) = \sum_{i=1}^{n} (x_i - \mu^{(t)}(x_i))^2,$$

where $\mu^{(t)}$ designates the set of centroids at iteration t, and $\mu^{(t)}(x_i)$ is the centroid [of the cluster] to which the instance x_i is assigned at iteration t. Prove in *analitical manner* (NOT numerically!) that for the dataset used at part a, at the end of iteration $t = 1$ we have

$$J(\mu^{(t)}) \leq J(\mu^{(t-1)}).$$

Răspuns:

a.

Iniţializare:

$C_1^{(0)} = \{A, B, F\}$
$C_2^{(0)} = \{C, H, I\}$
$C_3^{(0)} = \{D, E, G\}$.

Iteraţia 1:

$\mu_1^{(1)} =$

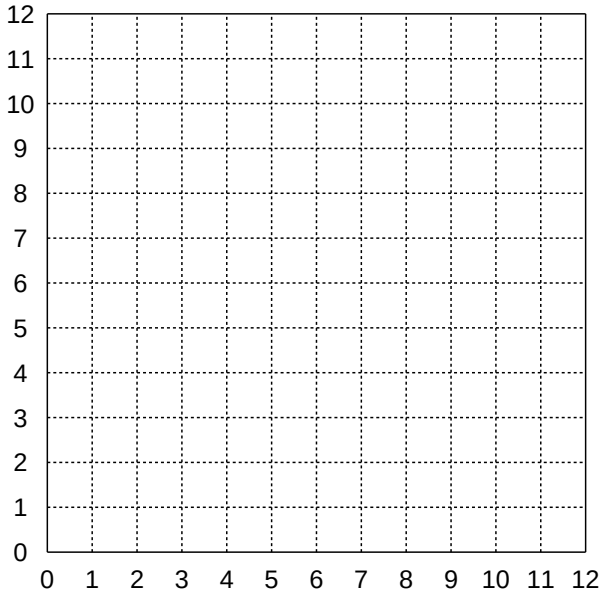
$\mu_2^{(1)} =$

$\mu_3^{(1)} =$

$C_1^{(1)} =$

$C_2^{(1)} =$

$C_3^{(1)} =$



Iteraţia 2:

$\mu_1^{(2)} =$

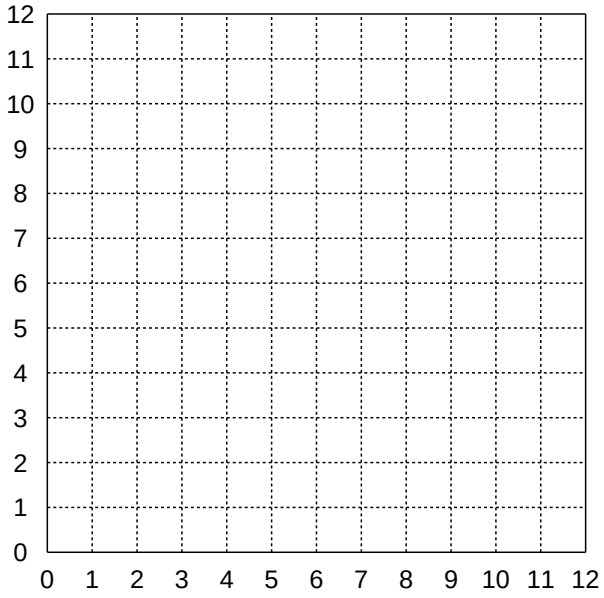
$\mu_2^{(2)} =$

$\mu_3^{(2)} =$

$C_1^{(2)} =$

$C_2^{(2)} =$

$C_3^{(2)} =$



b. ...

LC.4 (0.7p) **(EM/GMM, cazul unidimensional: executarea manuală a pasului M (precum şi a următorului pas E), pentru o mixtură de un tip particular, parametrii liberi fiind π_1, μ_1, μ_2)**

Suppose that we are fitting a Gaussian mixture model for data items consisting of a single real value, x, using $K = 2$ components. We have $N = 5$ instances, in which the values of x are as follows:

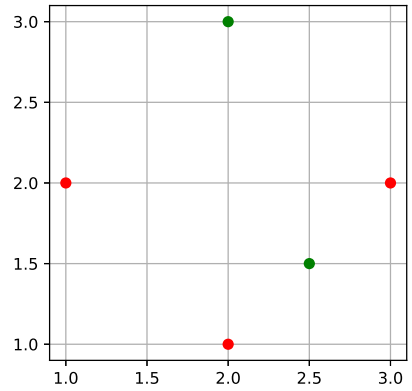
$$5,\ 15,\ 25,\ 30,\ 40.$$

We use the EM algorithm to find the maximum likeihood estimates for the model parameters, which are the mixing proportions for the two components, π_1 and π_2, and the means for the two components, μ_1 and μ_2. The standard deviations for the two components are fixed at 10. Suppose that at some point in the EM algorithm, the E step found that the *responsibilities* of the two components for the five data items were as follows:

p_{i1}	0.2	0.2	0.8	0.9	0.9
p_{i2}	0.8	0.8	0.2	0.1	0.1

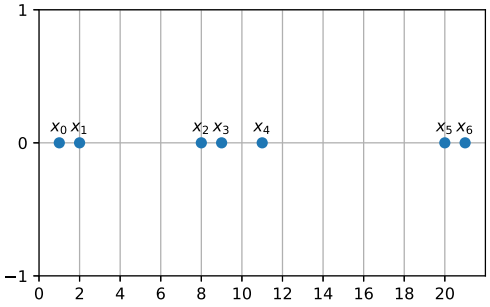
What values for the parameters π_1, π_2, μ_1, and μ_2 will be found in the next M step of the algorithm?

Ex. 1 — (0.25p)
What is the output of the following code on the dataset below?

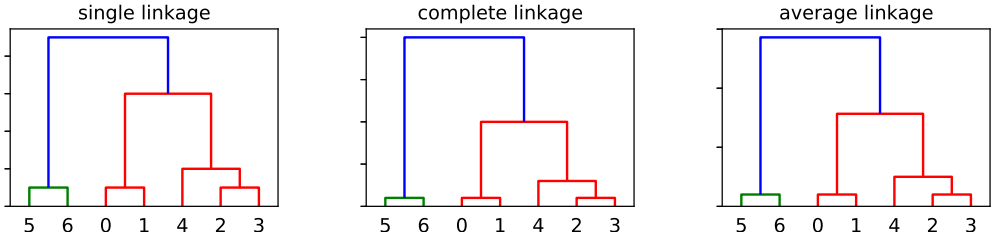



```
# [...] X and Y are initialised as usual with the features
# and the classes, respectively
from sklearn.ensemble import AdaBoostClassifier
ab = AdaBoostClassifier(n_estimators=1).fit(X, Y)
epsilon_1 = 1-ab.score(X, Y)
print(epsilon_1)
```

Ex. 2 — (0.5p)
For the dataset below, the dendrogram for agglomerative clustering has been generated for each type of linkage.



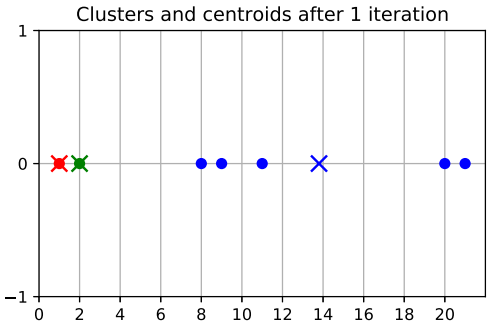
What is the height of the *blue* cluster for each of the three dendrograms?



Ex. 3 — (0.5p)
The results of the code below are displayed in the corresponding graph.

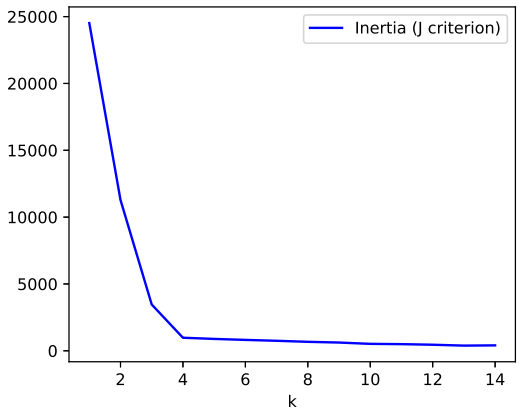
```
from sklearn.cluster import KMeans
```

```
# [...] Initialise dataset d
init_centroids = np.array([[1], [2], [3]])
km = KMeans(init=init_centroids, n_init=1, max_iter=1, n_clusters=3)
clusters = km.fit_predict(d)
centroids = km.cluster_centers_[:,0]
# [...] Plot clusters and centroids
print(km.inertia_)
```

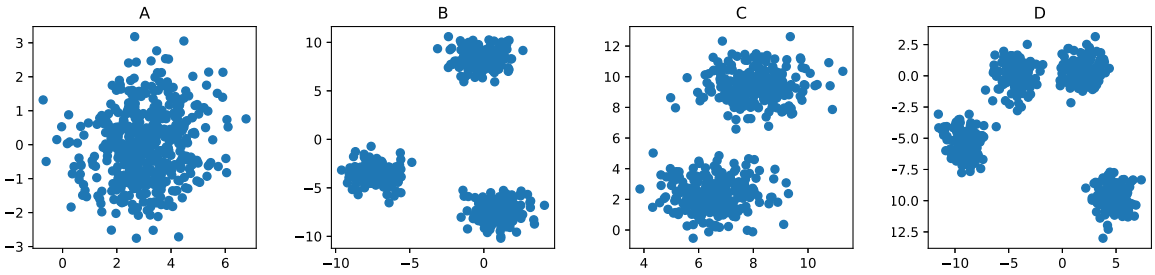


1. What is the value printed by the last line?
2. How does the graph change if `max_iter` is changed from 1 to 2?

Ex. 4 — (0.25p)
The line chart below shows the value of inertia (or the J criterion) for different values of k in the k-means algorithm.



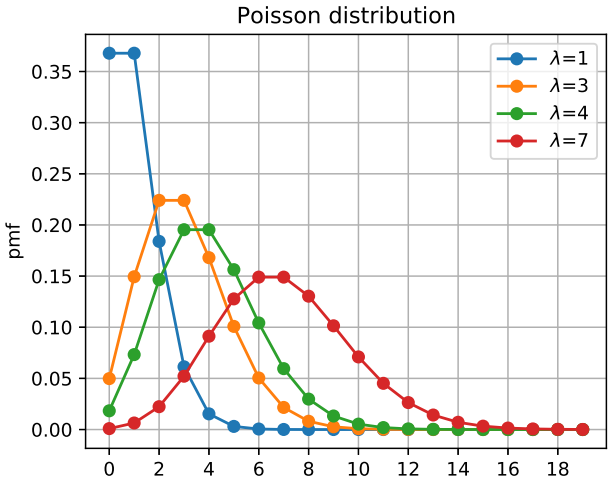
1. If the line chart would not be limited to 14, but could go to any values, where would the J criterion have its minimum value?
2. Which of the following datasets corresponds to the line chart above and why?



Ex. 5 — (0.5p)
What is the output of the following code?
(Please include any calculations as well!)

```
import pandas as pd
d = pd.DataFrame({
    'X': [1, 2, 6],
})
from scipy.stats import norm
print(norm.fit(d))
```

Ex. 6 — (0.5p)
An API receives 4 calls in one minute and 10 calls in the next. You learn that this is a Poisson process, with a λ parameter. Using the graphs below, which value for λ best describes your data? Is it 1, 3, 4 or 7?
(Please include any calculations as well! You can use approximations from the chart; you don't have to use the exact pmf formula.)



Ex. 7 — (0.5p)
The expectation-maximisation (EM) algorithm for Gaussian mixture models (GMM) has reached the state shown below. The algorithm is only adjusting the mean, while the standard deviation is fixed at 1. How will the Gaussian distributions change after running the expectation and maximisation steps one more time?

(Please include any calculations. You can approximate the values in the chart to the nearest grid point, so you don't have to use the exact pmf formula.)