Iniziato	mercoledì, 18 dicembre 2024, 14:23
Stato	Completato
Terminato	mercoledì, 18 dicembre 2024, 14:32
Tempo impiegato	8 min. 32 secondi
Valutazione	**0,00** su un massimo di 20,00 (**0**%)

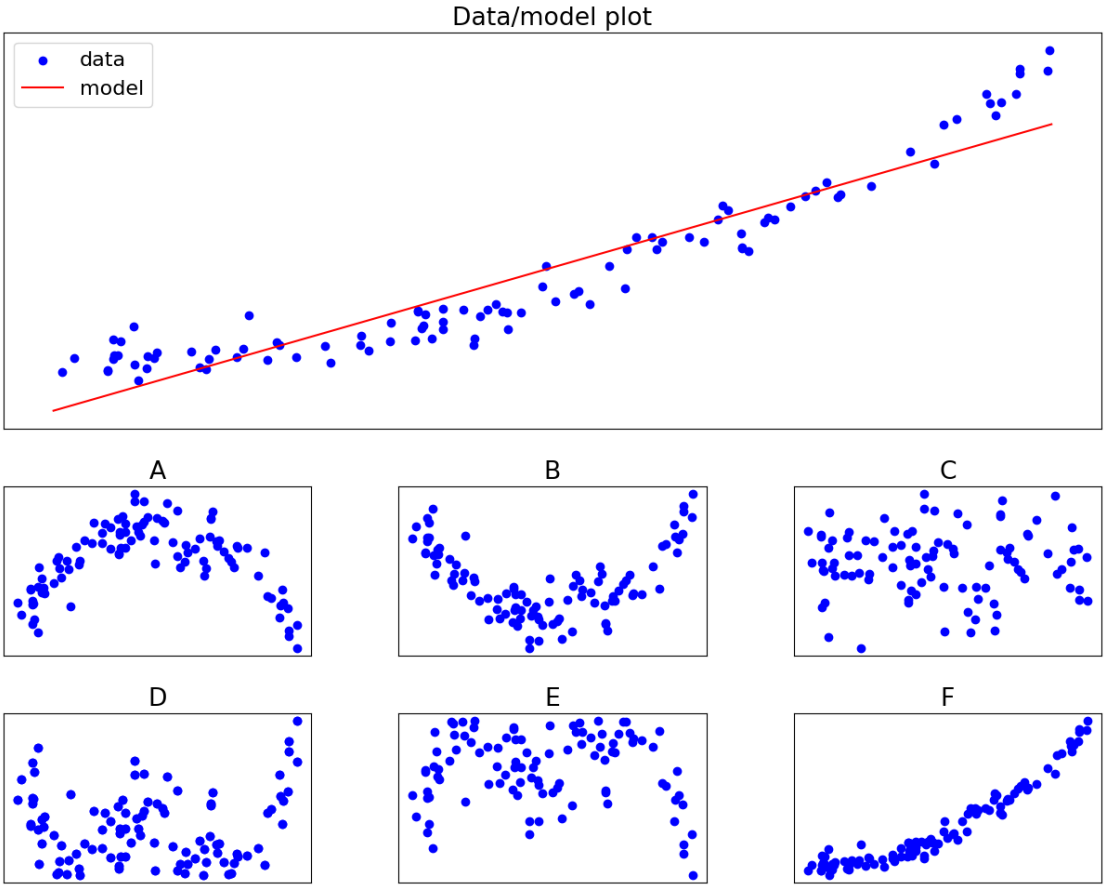
Domanda 1

Risposta non data

Punteggio max.: 1,00

1 point (-15% penalty for a wrong answer)

A 1-dimensional dataset is used for learning the weights of a linear regression model. The data used and the model obtained are shown in the figure below (top-most figure -- "Data/model plot").



We define the residual for the i-th point (x_i, y_i), with prediction y'_i, as $r_i = y_i - y'_i$.

Based on this definition, which of the following plots correctly represents the residuals for the top-most figure?

Note that the x and y labels and values have been removed on purpose. The x-axis contains the values of the 1-dimensional feature of the dataset (independent variable), whereas the y-axis represents the residual r.

Scegli un'alternativa:

- a. C
- b. B
- c. D
- d. The information available is insufficient to answer the question.
- e. E
- f. A

Domanda 3

Risposta non data

Punteggio max.: 2,00

2 points (no penalty for a wrong answer)

```
With a MAX linkage policy, the distance between two clusters is computed as the maximum distance between any pair of
points in the two clusters.
```

You are given the following distance matrix among 5 points, a, b, c, d, e.

	a	b	c	d	e
a	0	4	24	2	23
b	4	0	17	22	13
c	24	17	0	1	25
d	2	22	1	0	5
e	23	13	25	5	0

Apply hierarchical clustering using complete (MAX) linkage.

Write in the box below the clusters obtained at each step of the clustering.

Use a separate line to represent the state after each step. Separate the clusters obtained using a vertical bar (|).

Start from the case where each point belong to its own cluster. End with the case where all points belong to a single cluster.

For example,

```
a | b | c | d | e
a b | c | d | e
a b c | d | e
a b c d | e
a b c d e
```

In this case, the steps of the clustering first merge a with b, then (a, b) with c, then (a, b, c) with d, and finally (a, b, c, d) with e.

Domanda 4

Risposta non data

Punteggio max.: 1,50

1.5 points (-15% penalty for a wrong answer)

What is the difference between the *loc* and *iloc* accessors in pandas? Choose the right answer.

Scegli un'alternativa:

- a. *iloc* is used for label-based indexing and includes both the start and end indices. *loc* is used for integer-location based indexing and excludes the end index.
- b. *loc* is used for integer-location based indexing and includes both the start and end indices. *iloc* is used for label-based indexing and excludes the end index.
- c. *loc* is specifically designed for Series objects, while *iloc* is designed for DataFrame objects.
- d. *loc* is used for label-based indexing and includes both the start and end indices. *iloc* is used for integer-location based indexing and excludes the end index.
- e. *loc* and *iloc* are interchangeable and can be used interchangeably in all scenarios.
- f. Both *loc* and *iloc* are used for integer-location based indexing, but *loc* includes the end index, while *iloc* excludes it.
- g. None of the other statements is correct.
- h. Both *loc* and *iloc* are used for label-based indexing, but *loc* includes the end index, while *iloc* excludes it.

Domanda 5

Risposta non data

Punteggio max.: 2,00

2 points (no penalty for a wrong answer)

For two N-dimensional vectors x and y, the cosine distance is defined as:

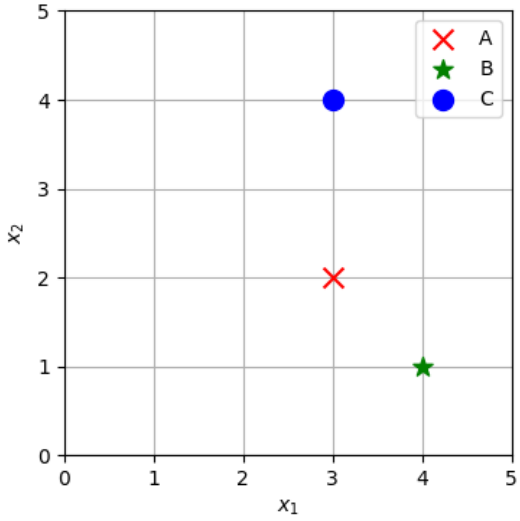
Sequenza di controllo \doty indefinita

where Sequenza di controllo \cdoty indefinita is the dot product of x and y, and $||x||_2$ is the norm 2 of x, defined as $\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$.

The Euclidean distance is defined as:

$$d_{EU}(x, y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$$

You are given the following 3 2-dimensional points, A, B and C.



Compute the Euclidean and cosine distances between A and B, and between A and C.

Write the answers in the box below, using the following syntax:

```
d_EU(A, B) = value
d_EU(A, C) = value
d_COS(A, B) = value
d_COS(A, C) = value
```

For example,

```
d_EU(A, B) = 10
d_EU(A, C) = 20
d_COS(A, B) = 0.5
d_COS(A, C) = 0.7
```

Domanda 6

Risposta non data

Punteggio max.: 1,50

1.5 points (no penalty for a wrong answer)

The following is the documentation for the FunctionTransformer class in scikit-learn.

sklearn.preprocessing.FunctionTransformer

class `sklearn.preprocessing.`**`FunctionTransformer`**(*func=None, inverse_func=None, *, validate=False, accept_sparse=False, check_inverse=True, feature_names_out=None, kw_args=None, inv_kw_args=None*) [source]

Constructs a transformer from an arbitrary callable.

A FunctionTransformer forwards its X (and optionally y) arguments to a user-defined function or function object and returns the result of this function. This is useful for stateless transformations such as taking the log of frequencies, doing custom scaling, etc.

Note: If a lambda is used as the function, then the resulting transformer will not be pickleable.

New in version 0.17.

Read more in the User Guide.

Parameters:

func : *callable, default=None*
The callable to use for the transformation. This will be passed the same arguments as transform, with args and kwargs forwarded. If func is None, then func will be the identity function.

inverse_func : *callable, default=None*
The callable to use for the inverse transformation. This will be passed the same arguments as inverse transform, with args and kwargs forwarded. If inverse_func is None, then inverse_func will be the identity function.

validate : *bool, default=False*
Indicate that the input X array should be checked before calling `func`. The possibilities are:

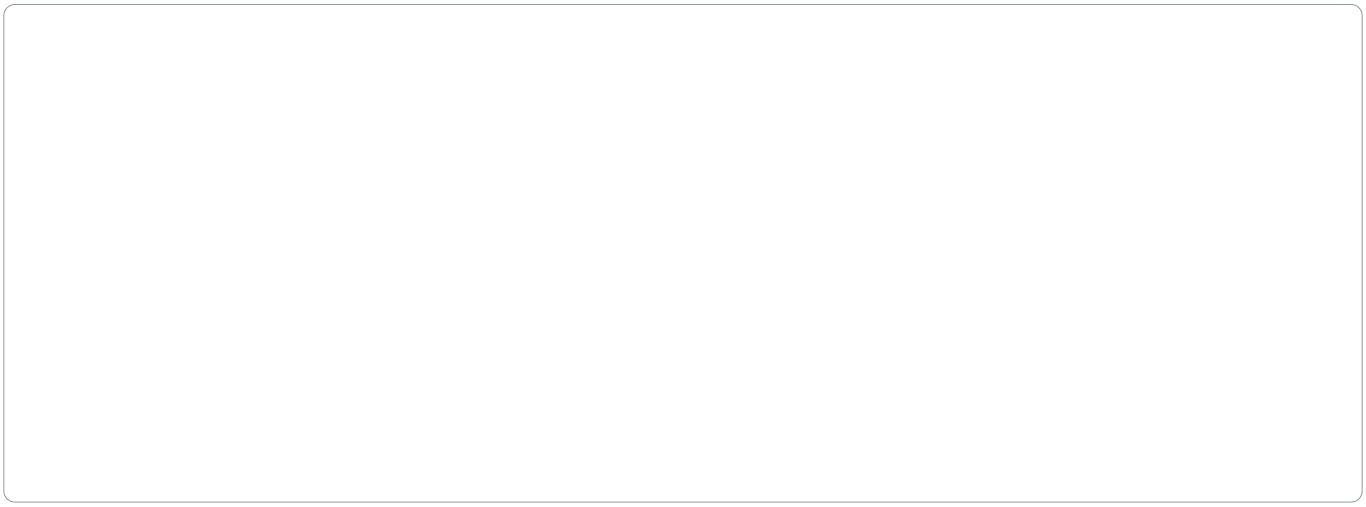
- If False, there is no input validation.
- If True, then X will be converted to a 2-dimensional NumPy array or sparse matrix. If the conversion is not possible an exception is raised.

What is the output of the following snippet of code?

```
from sklearn.preprocessing import FunctionTransformer, MinMaxScaler
from sklearn.pipeline import make_pipeline
import numpy as np

pipe = make_pipeline(
    MinMaxScaler(),
    FunctionTransformer(lambda x: 1/(x**2+1))
)

X = np.array([
    [1,2],
    [3,5],
    [2,10],
])

print(pipe.fit_transform(X))
```

Write the answer in the box below. If an error occurs, write "an error occurs at line X" (X being the line where the error occurs).

When representing a NumPy array, make sure you use square brackets to correctly identify dimensions. For instance, [1,2,3] and [[1], [2], [3]] represent two different results.



Domanda 7

Risposta non data

Punteggio max.: 1,50

1.5 points (-15% penalty for a wrong answer)

The Gini index of a child node N where points belong to one of M classes is calculated as follows:

$Gini(N) = 1 - \sum_{i=1}^{M} p_i^2$

Where p_i is the proportion of points that belong to class i in the child node.

The Gini index of a split having children nodes with N1 and N2 points is calculated as follows:

$Gini = \frac{N1}{N1+N2} Gini(N1) + \frac{N2}{N1+N2} Gini(N2)$

Where Gini(N1) and Gini(N2) are the Gini indices of the left and right child nodes, respectively.

You are training a decision tree model to predict one of three classes, C1, C2, or C3.

During the training phase, a total of 100 points reach a specific node.

The point's classes are distributed as follows:
C1: 40
C2: 1
C3: 59

The following are 5 possible splits that are being considered for the node.

```
Split 1: [9, 0, 3]      [31, 1, 56]
Split 2: [34, 0, 52]    [6, 1, 7]
Split 3: [39, 0, 2]     [1, 1, 57]
Split 4: [13, 0, 8]     [27, 1, 51]
Split 5: [0, 0, 53]     [40, 1, 6]
```

Each split is reported as two lists of values.
The left-most list represents the distribution of the classes in the left child node.
The right-most list represents the distribution of the classes in the right child node.
The distribution of the classes represents the number of points of each class (C1, C2, C3) that reaches the child node.

What is the value of the Gini index for the best split? All options are rounded to 4 significant figures.

Scegli un'alternativa:

- a. None of the other answers is correct
- b. 0.077
- c. 0.3265
- d. 0.1264
- e. 0.0547
- f. 0.0154
- g. 0.1217
- h. 0.3088
- i. 0

Domanda 8

Risposta non data

Punteggio max.: 1,50

1.5 points (no penalty for a wrong answer)

You are given a dataset X, containing 50,000 images. Each image is represented as a 28x28 grid of gray-scale pixels.

As such, X can be represented as a NumPy array with shape (50000, 28, 28).

The label of each image (a number between 0 and 9) is encoded in a 1-dimensional array y, with shape (50000,).

The labels are equally distributed (i.e., 5,000 images belong to each of the 10 classes).

What is the output of the following snippet of code?

```
a = X[y==0, :, ::-1]
b = X[y==8][0]

c = (a - b)**2
d = c.sum(axis=1).sum(axis=1)

print(d.shape)
```

Write the answer in the box below. If an error occurs, write "an error occurs at line X" (X being the number of the line).

Domanda 9

Risposta non data

Punteggio max.: 2,50

2.5 points (no penalty for a wrong answer)

You are given the following list of transactions.

```
cde
cd
ade
bd
abe
bde
acde
abde
bde
bd
```

Apply the FP-growth algorithm using minsup = 2 (an itemset is frequent if it appears in at least two transactions).

Write the following:

1. A-CPB (A - Conditional Pattern Base)
2. A-CHT (A - Conditional Header Table)
3. AB-CPB (AB - Conditional Pattern Base)

Write all and only the correct itemsets, along with their support counts.

Use the following syntax:

```
A-CPB = { element1: support1, element2: support2, ... }
A-CHT = { element1: support1, element2: support2, ... }
AB-CPB = { element1: support1, element2: support2, ... }
```

For example:

```
A-CPB = { a: 1, abc: 2 }
A-CHT = { a: 3, b: 2, c: 1 }
AB-CPB = { a: 1, b: 2, c: 3 }
```

Domanda 10

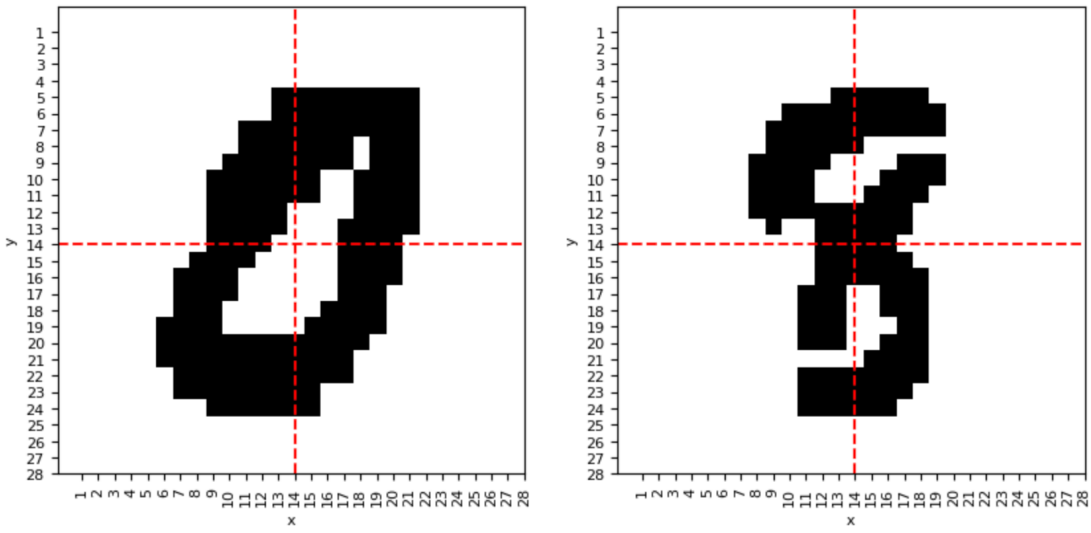
Risposta non data

Punteggio max.: 1,50

1.5 points (-15% penalty for a wrong answer)

A dataset of 28x28 black and white images is given. Each image in the dataset represents either a digit "0", or a digit "8". Each image is represented as a 28x28 matrix of 0/1 features (0 for white pixels, 1 for black pixels).

Examples of two digits are shown in the image below (the red lines and the crosses are included for your convenience).



Your goal is to extract features that could be useful in separating the two classes.

Which of the following features is best suited for this task?

Scegli un'alternativa:

- a. None of the proposed features can help separate between the two classes.
- b. The number of black pixels contained in the top half of the image
- c. The number of black pixels contained in the right half of the image
- d. The value of the pixel at position x=21, y=7
- e. The value of the pixel at position x=7, y=21
- f. The number of black pixels contained in the top-right quarter of the image
- g. The number of black pixels contained in the left half of the image
- h. The value of the pixel at position x=14, y=14
- i. The number of black pixels contained in the top-left quarter of the image
- j. The value of the pixel at position x=7, y=7
- k. The number of black pixels contained in the bottom-left quarter of the image
- l. The number of black pixels contained in the bottom-right quarter of the image
- m. The value of the pixel at position x=21, y=21
- n. The number of black pixels contained in the bottom half of the image

Domanda 11

Risposta non data

Punteggio max.: 2,00

2 points (no penalty for a wrong answer)

The KNN algorithm can assign a class probability based on the votes assigned to each class. If $vote_Y(x)$ represents the (unnormalized) number of votes assigned to class Y for sample x, then, for any class $c \in C$:

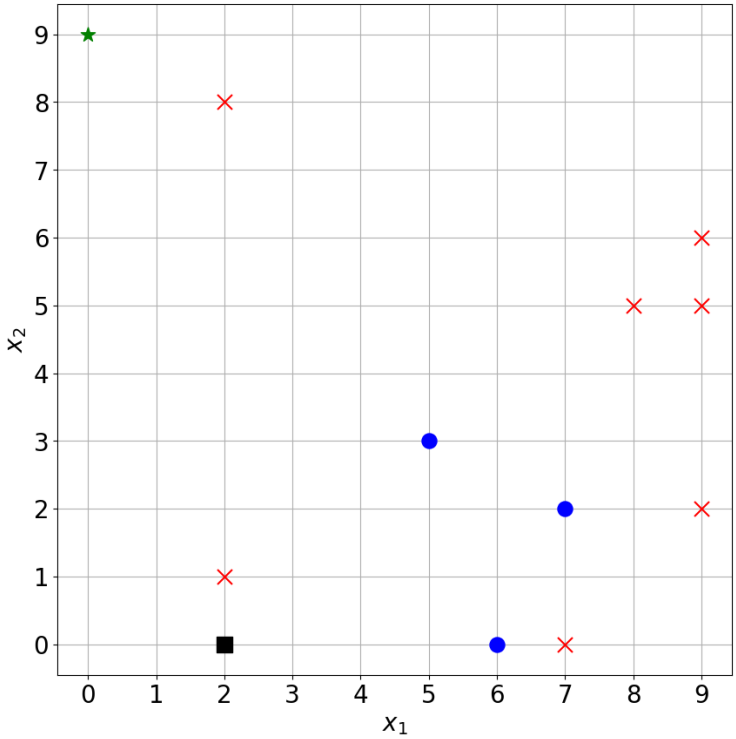
$$p(c|x) = \frac{vote_c(x)}{\sum_{i \in C} vote_i(x)}$$

The L_∞ distance between two N-dimensional points x and y is defined as:

$$d(x, y) = \max |x_i - y_i|$$

where x_i is the i-th dimension of x.

The figure below shows a 2-dimensional dataset. Points in this dataset belong to either of two classes: circles (in blue) or crosses (in red).



You are given two test points to be labelled, A (green star) and B (black square). The K-NN algorithm with L_∞ distance and K = 3 is used to label A and B.

Answer the following questions.

Question 1) What is the label assigned to A and B?

Question 2) What are the class probabilities for A and B, if each distance is weighted with a weight $w = \frac{1}{1+distance}$?

Use the following notation:

```
A1) A=class_for_A, B=class_for_B
A2)
A: cross=prob_cross_A, circle=prob_circle_A
B: cross=prob_cross_B, circle=prob_circle_B
```

For example

```
A1) A=circle, B=circle
A2)
A: cross=0.5, circle=0.5
B: cross=0.9, circle=0.1
```

Domanda 12

Risposta non data

Punteggio max.: 1,50

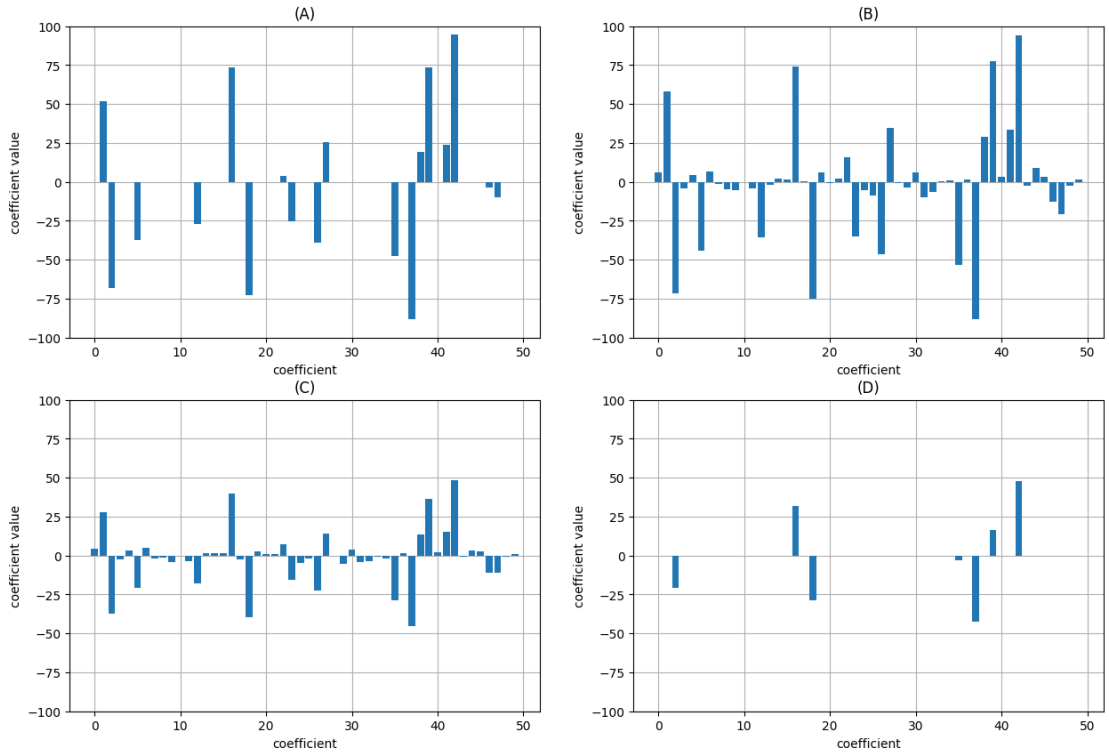
1.5 points (-15% penalty for each wrong answer)

The weights w of a Ridge model are chosen so as to minimize $MSE + \lambda||w||_2^2$
The weights w of a Lasso model are chosen so as to minimize $MSE + \lambda||w||_1$

Two Ridge models and two Lasso models have been trained on the same dataset comprised of 1,000 50-dimensional points. In particular, the following are the characteristics of the four models:

- Ridge, $\lambda = 10.0$
- Ridge, $\lambda = 100.0$
- Lasso, $\lambda = 1.0$
- Lasso, $\lambda = 5.0$

After training, you plot the weights (coefficients) learned by the four models as bar charts. The following are the plots you obtain.



You labelled the four plots with letters (A, B, C, D respectively). Which model corresponds to which plot?

Select all correct mappings. Each plot is associated to only one correct answer.

Scegli una o più alternative:

- a. (B) Ridge, $\lambda = 100.0$
- b. (C) Lasso, $\lambda = 1.0$
- c. (B) Ridge, $\lambda = 10.0$
- d. (C) Ridge, $\lambda = 100.0$
- e. (B) Lasso, $\lambda = 1.0$
- f. (D) Lasso, $\lambda = 1.0$
- g. (C) Lasso, $\lambda = 5.0$
- h. (D) Lasso, $\lambda = 5.0$
- i. (A) Lasso, $\lambda = 1.0$
- j. (A) Lasso, $\lambda = 5.0$
- k. (A) Ridge, $\lambda = 10.0$
- l. (C) Ridge, $\lambda = 10.0$
- m. (A) Ridge, $\lambda = 100.0$
- n. (D) Ridge, $\lambda = 10.0$
- o. (C) Lasso, $\lambda = 1.0$
- p. (D) Ridge, $\lambda = 100.0$
- q. (B) Lasso, $\lambda = 5.0$

Domanda 13

Risposta non data

Punteggio max.: 1,50

1.5 points (no penalty for a wrong answer)

A clustering C can be represented as a set of n clusters $\{C_1, C_2, \ldots, C_n\}$. Each cluster C_i is a set of points that have been assigned to that cluster.

For a given point $x \in C_i$, its silhouette score can be computed as:

$silh(x) = \frac{b(x)-a(x)}{max(b(x),a(x))}$

Where:
- a(x) is the mean distance between x and all points in C_i, i.e. $a(x) = \frac{1}{|C_i|-1} \sum_{j \in C_i, j \neq x} dist(j,x)$

- b(x) is the minimum of the average distance between x and any other cluster (the distance between a point and a cluster is defined as the average distance between the point and all points in the cluster):
$b(x) = min_{C_k, k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} dist(j,x)$

(dist(w,z) is the distance between points w and z)

The following is a distance matrix among 6 2-dimensional points (the Manhattan distance has been used).

	a	b	c	d	e	f
a	0	5	9	5	2	2
b	5	0	8	10	5	5
c	9	8	0	8	11	7
d	5	10	8	0	5	5
e	2	5	11	5	0	4
f	2	5	7	5	4	0

A clustering algorithm has been applied to this dataset, obtaining 3 clusters (0, 1, 2). The cluster labels assigned to the 6 points are the following:

a: 2
b: 1
c: 0
d: 2
e: 0
f: 2

What are the silhouette scores for points b and d?

Write the answer in the box below.

Use the following syntax:

```
silh(b) = value
silh(d) = value
```

For example,

```
silh(b) = 0.5
silh(d) = 0.1
```