Let *t* be a term (or word) and *D* a collection of textual documents. Let *d* be a document in *D*.

tf-idf is defined as follows:

$$tf\text{-}idf(t, d, D) = freq(t, d) * log(\frac{m}{freq(t, D)})$$

where:

- *freq(t, d)* is the frequency of *t* in document *d*,
- *freq(t, D)* is the number of documents in *D* where *t* occurs, and
- *m* is a constant value

Which of the following statements is true?

- (a) The tf-idf of *t* is a numerical value computed separately for each document in *D*, which is equal to the frequency of *t* in the document.
- (b) The tf-idf of *t* is a single numerical value defined for *D* that specifies its relevance to distinguish between documents.
- (c) The tf-idf of *t* is a numerical value computed separately for each document in *D*, which is higher when *t* appears in all documents.
- (d) The tf-idf of *t* is a single numerical value defined for *D* that specifies the frequency of t in the documents.
- (e) The tf-idf of *t* is defined separately for each document in *D* and it is defined only if *t* occurs at least once in the specified document.
- (f) None of the answers is correct.

Given a set of documents $D = \{d_1, d_2, \ldots\}$ and a set of terms $T = \{t_1, t_2, \ldots\}$, the tf-idf for a term $t \in T$ in a document $d \in D$ is defined as:

$$tf - idf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

A possible definition for $tf(t, d)$ is the number of occurrences of t within d.

A possible definition of $idf(t, D)$ is the following:

$$idf(t, D) = log \frac{|D|}{|\{d \in D : t \in d\}|}$$

You are given the following collection of documents:

```
<term 0> <term 0> <term 0> <term 3> <term 0>
<term 0> <term 1> <term 0> <term 0>
<term 3> <term 3> <term 2> <term 2>
```

For the documents, use the same order in which the documents appear in the above list. For the terms, use their natural order (<term 0>, <term 1>, etc).

```
tfidf_term_0_doc_0 tfidf_term_1_doc_0 tfidf_term_2_doc_0 tfidf_term_3_doc_0
tfidf_term_0_doc_1 tfidf_term_1_doc_1 tfidf_term_2_doc_1 tfidf_term_3_doc_1
tfidf_term_0_doc_2 tfidf_term_1_doc_2 tfidf_term_2_doc_2 tfidf_term_3_doc_2
```

For example,

```
0.1234 0.4321 0.5555 0.6667
0.6667 0.5555 0.4321 0.1234
0.1212 0.2121 0.3333 0.3333
```

Where each line represents a document and each term is represented in the form <term X>.

Compute the tf-idf matrix for all terms/documents.

Report the result in the box below, listing one document for each line and one term for each column.

A sparse matrix is a matrix in which most elements are 0.

tfidf(term, document, documents) = tf(term, document) * idf(term, documents)

Where:

- tf(term, document) = term frequency of `term` in `document`
- idf(term, documents) = inverse document frequency of `term` in a set of documents `documents`

The output of TF-IDF is:

- (a) A sparse matrix, since most terms will not occur in most documents
- (b) A dense matrix, since tf(term) is always > 0
- (c) A sparse matrix, since idf(term, documents) can often be 0
- (d) A dense matrix, since idf(term, documents) is always > 0
- (e) A sparse matrix, since idf(term, documents) is always > 0
- (f) A dense matrix, since most documents contain a limited number of terms

A sparse matrix is a matrix in which most elements are 0.

tfidf(term, document, documents) = tf(term, document) * idf(term, documents)

Where:

- tf(term, document) = term frequency of `term` in `document`
- idf(term, documents) = inverse document frequency of `term` in a set of documents `documents`

The output of TF-IDF is:

- (a) A sparse matrix, since idf(term, documents) is always > 0
- (b) A dense matrix, since most documents contain a limited number of terms
- (c) A sparse matrix, since most terms will not occur in most documents
- (d) A dense matrix, since tf(term) is always > 0
- (e) A sparse matrix, since idf(term, documents) can often be 0
- (f) A dense matrix, since idf(term, documents) is always > 0

Tf-idf can be defined as:

$tf - idf(t, d, D) = tf(t, d) \cdot idf(t, D)$

Where t and d are the term and the document under consideration respectively, D is the set of all documents, $tf(t, d)$ is the term frequency and idf is the inverse document frequency.

A possible definition of the idf is $log \frac{|D| - n_{tD}}{n_{tD}}$, where n_{tD} it represents the number of documents where t it appears, ie $n_{tD} = |\{d \in D : t \in d\}|$.

What properties must the idf function (an example of which is provided above) follow? Select all correct answers. Multiple answers may be correct.

Choose one or more alternatives:

- [ ] (a) idf must be positive (> 0)
- [ ] (b) idf must be defined for $n_{tD} \in [1, |D|]$
- [ ] (c) idf must monotonically increase with n_{tD}
- [ ] (d) idf must be defined for $n_{tD} = 0$
- [ ] (e) idf must monotonically decrease with n_{tD}
- [ ] (f) idf must be defined for $n_{tD} \in (-\infty, +\infty)$
- [ ] (g) idf must be negative (< 0)
- [ ] (h) None of the other options is correct

The z-score normalization for a feature with mean μ and variance σ^2 is computed as:

$$x' = \frac{x-\mu}{\sigma}$$

You sample a data distribution and obtain samples $f = (f_1, f_2, \ldots, f_n)$, with mean μ and standard deviation σ.

You know that $max(f) = \mu + 3\sigma$, and $min(f) = \mu - 2\sigma$.

You apply z-score normalization on f to obtain $f' = (f'_1, f'_2, \ldots, f'_n)$.

Which of the following statements are correct? (select all correct answers)

Scegli una o più alternative:

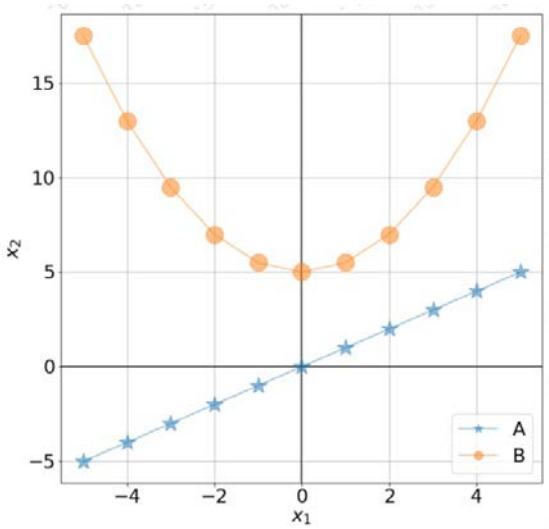
- [ ] (a) The upper bound for f' depends on max(f)
- [ ] (b) The upper bound for f' depends on both max(f) and min(f)
- [ ] (c) There is both an upper and a lower bound to f'
- [ ] (d) There is a lower bound to f', but not an upper bound
- [ ] (e) There is an upper bound to f', but not a lower bound
- [ ] (f) The lower bound for f' depends on min(f)
- [ ] (g) The boundaries of f' are independent of max(f) and min(f)
- [ ] (h) The lower bound for f' depends on both max(f) and min(f)

The Pearson correlation between two variables $x = \{x_1, x_2, \ldots, x_n\}$ having mean $\bar{x}$ and $y = \{y_1, y_2, \ldots, y_n\}$ having mean $\bar{y}$ is defined as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

Two sequences A and B have been sampled from a distribution. The following sequences are included below both in numerical and graphical form. B can be computed as $0.5 \cdot A^2 + 5$.

A	B
-5	17.5
-4	13
-3	9.5
-2	7
-1	5.5
0	5
1	5.5
2	7
3	9.5
4	13
5	17.5



What is the Pearson correlation between A and B (r_{AB})?

- ○ (a) -0.5
- ○ (b) 0
- ○ (c) +0.5
- ○ (d) -1
- ○ (e) -0.25
- ○ (f) +1
- ○ (g) +0.25

The Simple Matching Coefficient (smc) between two binary sequences $p_1 = \{p_{01}, p_{11}, \dots, p_{n1}\}$ and a point $p_2 = \{p_{02}, p_{12}, \dots, p_{n2}\}$ is defined as:

$$smc(p1, p2) = \frac{M_{11}+M_{00}}{M_{00}+M_{01}+M_{10}+M_{11}}$$

Where:

- M_{00} is the number of elements that are both zeros in the two sequences, i.e. $|\{i \in [0, n]\ s.t.\ p_{i1} = 0 \wedge p_{i2} = 0\}|$.
- M_{01} and M_{10} are the number of elements that are either 0 and 1 or 1 and 0 in the two sequences, i.e. $|\{i \in [0, n]\ s.t.\ p_{i1} = 0 \wedge p_{i2} = 1\}|$ and $|\{i \in [0, n]\ s.t.\ p_{i1} = 1 \wedge p_{i2} = 0\}|$ respectively.
- M_{11} is the number of elements that are both ones in the two sequences, i.e. $|\{i \in [0, n]\ s.t.\ p_{i1} = 1 \wedge p_{i2} = 1\}|$.

	x1	x2
pt0	2	7
pt1	3	8
pt2	5	7
pt3	8	3
pt4	6	8

The following 2-dimensional dataset is provided.

You apply **equal-width** binning on the two features to obtain a discrete version of this dataset. Use a number of **bins = 3**. You can infer the range of possible values for the features directly from the dataset.

Q1) What dataset do you obtain after discretization? Assign a **letter of the alphabet** (a, b, c, ...) to each bin.

Q2) Next, you apply **1-hot encoding** to the resulting discrete features. What dataset do you get after applying 1-hot encoding? Specify the encoded column names as *feature_name=value* (e.g. x1=a, x2=b, etc).

Q3) Finally, you use a simple matching to compute **smc(pt2, pt4)** and **smc(pt4, pt3)**. What distances do you get?

When matching two binary sequences $v_1 = (v_{11}, v_{12}, \dots v_{1n})$, $v_2 = (v_{21}, v_{22}, \dots v_{2n})$, the Jaccard similarity is computed as follows:

$$J(v_1, v_2) = \frac{M_{11}(v_1,v_2)}{M_{11}(v_1,v_2)+M_{01}(v_1,v_2)+M_{10}(v_1,v_2)}$$

Where $M_{ij}(v_1, v_2)$ is the number of instances of $k \leq n$ for which $v_{1k} = i$ and $v_{2k} = j$

You are given the following categorical dataset with features x0, x1, x2.

1. What dataset do you obtain by applying 1-hot encoding to all of its features? Specify the column names in the form "attribute_value"
2. Compute J(x1, x2) and J(x1, x4)

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

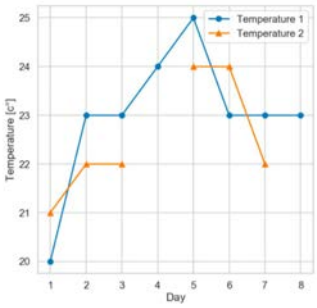
```
1.
attribute1_value1 attribute1_value2 ... attributeM_valueN
value_1.1 value_1.2 ... value_1.X
value_2.1 value_2.2 ... value_2.X
...
value_Y.1 value_Y.2 ... value_Y.X
2.
J(x1, x2) = value
J(x1, x4) = value
```

x0	x1	x2
a	c	y
a	b	z
b	b	z
a	a	y

For example

```
1.
v1_a v1_b v2_c v2_d
0 1 0 1
1 0 1 0
0 1 0 1
0 1 0 1
2.
J(x1, x2) = 0.9
J(x1, x4) = 0.8
```

You have to compare the daily temperature differences between two cities, sample by sample. The trends are showed in the picture below.



Question 1) Which kind of preprocessing would you choose?

- Add zero padding after Temperature 2
- Fill missing values with 0
- Fill missing values with the previous value
- Consider 0 the difference between the series when missing values

Question 2) Given the strategy that you have selected, calculate the mean of the daily differences calculated with the following NumPy function.

```
avg_daily_diff = np.mean(temperature_1-temperature_2)
```

In contrast with a grid search, a *randomized search* randomly samples and tests a subset of all possible configurations of hyperparameters, instead of testing all of them.

You apply a randomized search and a grid search at the same time, tuning the same hyperparameters with the same candidate values sets.

Which of the following statements is correct?

- (a) The randomized search tests more configurations of hyperparameters with respect to the grid search
- (b) The randomized search will take a longer time for the tuning of the hyperparameters with respect to the grid search
- (c) None of the other answers is correct
- (d) It is advisable to run the randomized search after the grid search
- (e) The randomized search might find a better configuration than the one found by the grid search
- (f) The randomized search is guaranteed to find a configuration no better than the one found by the grid search

A neural network is trained on a training set with 2 features x1,x2.

- x1 is a continuous value
- x2 is a nominal value among the following: “high”,”med”,”low”

Before training, the features are preprocessed in the following way:

- x1 is normalized with z-score: `x1norm = (x1-np.mean(x1))/np.std(x1)`
- x2 is encoded with a one hot vector (“high” : 001, “med” : 010, “low” : 100)

When applying the trained classifier on unlabeled data (denoted as test set), which of the following statements are true (multiple answers may be true)?

Scegli una o più alternative:

- (a) the x1 feature of the test record must be normalized with the mean and standard deviation computed for the test set
- (b) a record with a value of x2 not included in “high”,”med”,”low” cannot be processed
- (c) the x1 feature of the test record can be provided to the classifier without normalization
- (d) it is suggested to process a record with a value of x2 not included in “high”,”med”,”low” by adding an input neuron to the neural network
- (e) it is suggested to process a record with a value of x2 not included in “high”,”med”,”low” by choosing a random value for the one hot features
- (f) the x1 feature of the test record must be normalized with the mean and standard deviation previously computed for the training set

The z-score normalization of a feature $x = (x_1, x_2, \ldots, x_n)$ is given by:

$$x' = \frac{x - \frac{1}{n}\sum_i x_i}{\sqrt{\frac{1}{n}\sum_i (x_i - \frac{1}{n}\sum_j x_j)^2}}$$

You are given a feature f with mean μ and standard deviation σ.

Which of the following statements on the z-score normalization is correct?

- (a) It normalizes all points to the $[-\sigma^2, +\sigma^2]$ range
- (b) None of the other statements is correct
- (c) It normalizes all points to the [-1, +1] range
- (d) It normalizes all points to the $[0, +\sigma]$ range
- (e) It normalizes all points to the $[-\sigma, +\sigma]$ range
- (f) It normalizes all points to the [0, +1] range

Equal width discretization applied to a feature f creates N intervals with the same width $w = \frac{max(f) - min(f)}{N}$

Given the dataset below (4 points, 2 features), apply equal width discretization to f1 and f2 with N = 3 bins.
Write in the answer box:

1. The discretized features of x1
2. The discretized features of x2

	f1	f2
x1	6	13
x2	2	11
x3	2	10
x4	11	16

For each feature, you must use the 3 bins L (for Low), M (for Medium), H (for High), such that:

1. $\forall l \in L, m \in M, l < m$, and
2. $\forall m \in M, h \in H, m < h$

Use the following notation:
x1: [bin_f1, bin_f2]
x2: [bin_f1, bin_f2]
where bin_f1 and bin_f2 are the selected bins (i.e., one among L, M, H)

Example:
x1: [L, L]
x2: [H, H]

Association rules

You are given the following list of transactions.

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

Write the following:

1. B-CPB (B - Conditional Pattern Base)
2. B-CHT (B - Conditional Header Table)
3. BD-CPB (BD - Conditional Pattern Base)

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

Use the following syntax:

```
B-CPB = { element1: support1, element2: support2, ... }
B-CHT = { element1: support1, element2: support2, ... }
BD-CPB = { element1: support1, element2: support2, ... }
```

For example:

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

A C D
C D
A C
C
B D
A B C D
C D E
A B C
A C
D
C D
A B
A B D E

An itemset is closed if none of its immediate supersets has the same support as the itemset

Given the transactional dataset shown in the figure below, apply the Apriori algorithm to extract all frequent itemsets.

The value of minsup is 2.

An itemset is considered to be frequent if its support count is equal to or higher than the minsup.

	Transactions
0	B E
1	B C
2	A B D E
3	B C D E
4	B C
5	A B E
6	B E
7	A C E
8	B D E
9	A B

Question 1) List all frequent itemsets having length 2, along with their support count.
Question 2) List all itemsets of length 3 that have been generated by Apriori after the join and prune steps, **before counting their support in the database**.
Question 3) List all frequent itemsets that are **not closed**, along with their support count.

Use the following notation:

```
A1) { ... list of itemsets w/ support count ... }
A2) { ... list of itemsets ... }
A3) { ... list of itemsets w/ support count ... }
```

For example

```
A1) { ab: 2, ac: 3, ad: 2 }
A2) { abc, abd, abe }
A3) { ab: 2, ad: 2, bce: 3 }
```

You are given the following list of itemsets:

```
ADE
AB
BCD
BDE
ABC
AE
CD
ABE
ABCDE
ABDE
```

Apply the Apriori algorithm to extract all frequent itemsets, considering minsup = 2 (an itemset is frequent if it appears in at least two transactions). Only consider the extraction of itemsets of length up to 3 (included).

1. Which are the itemsets that can be pruned during the prune step (ie after the join step)?
2. Which are the itemsets that are pruned after calculating their support count?

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

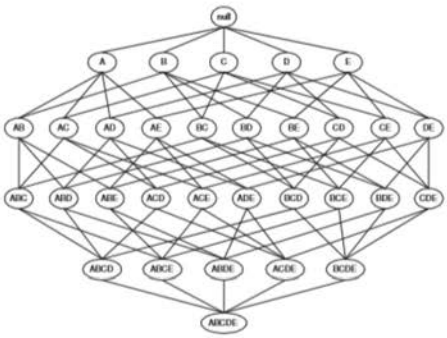
```
1. (itemset1, itemset2, ..., itemsetN)
2. (itemset1, itemset2, ..., itemsetM)
```

For example:

```
1. (A, B, AB)
2. (C, D, CD)
```

A maximal frequent itemset is a frequent itemset whose immediate supersets are all infrequent.

Considering the itemsets generated as in the figure below, it is known that the itemset {ACE} is a **maximal frequent itemset** . Which of the following statements is correct?



- (a) The itemset {ABCD} cannot be a frequent itemset
- (b) The itemset {ACDE} is also a maximal frequent itemset
- (c) None of the other statements are correct
- (d) The itemset {ACD} cannot be a frequent itemset
- (e) The itemset {BC} cannot be a frequent itemset
- (f) The itemset {ACDE} is guaranteed to be a frequent itemset
- (g) The itemset {ABCE} cannot be a frequent itemset

Given the transactional dataset below, apply the Apriori algorithm to extract all frequent itemsets.

Transactions
A B D
A B
C E
A D E
A E
B C E
A C D E
A B C
B C E
A E

The value of minsup is 2.

An itemset is considered to be frequent if its support count is equal to or higher than the minsup.

Question 1) List all frequent itemsets having length 2, along with their support count.
Question 2) List all itemsets of length 3 that have been generated by Apriori after the join and prune steps, before counting their support in the database.
Question 3) List all frequent itemsets that are not closed, along with their support count.

Use the following notation:

```
A1) { ... list of itemsets w/ support count ... }
A2) { ... list of itemsets ... }
A3) { ... list of itemsets w/ support count ... }
```

For example

```
A1) { ab: 2, ac: 3, ad: 2 }
A2) { abc, abd, abe }
A3) { ab: 2, ad: 2, bce: 3 }
```

Given an association rule r: A $\longrightarrow$B, lift(r) is defined as $\frac{conf(r)}{sup(B)}$, where conf(r) is the confidence of r, $\frac{sup(A,B)}{sup(A)}$ and sup(X) is the support of X.

Which of the following statements is correct?

- (a) lift(r) $\in$[-1, 1]
- (b) if lift(A $\longrightarrow$B) > 1, then lift(A $\longrightarrow$not B) < 1
- (c) None of the other statements are correct
- (d) lift(A $\longrightarrow$B) $\neq$lift(B $\longrightarrow$A)
- (e) lift(r) > 0 implies positive correlation
- (f) lift(r) < 0 implies independence

The support of an itemset A (sup(A)) is the fraction of transactions that contains A and is represented by P(A).

The confidence of an association rule A $\longrightarrow$ B is the conditional probability P(B|A).

The following rule ab $\longrightarrow$ c with confidence(ab $\longrightarrow$ c) = 1.0 is given.

Which statement is true?

- (a) If a transaction contains ab, it will optionally contain c.
- (b) If sup(ab) = 1.0, then c must be present in all transactions.
- (c) None of the other answers is correct.
- (d) sup(c) can be 0.
- (e) If sup(c) = 1.0, then sup(ab) cannot take any value.
- (f) P(abc) must be 1.0.

The support of an itemset A (sup(A)) is the fraction of transactions that contains A and is represented by P(A).

The confidence of an association rule A $\longrightarrow$ B is the conditional probability P(B|A).

The following rule ab $\longrightarrow$ c with confidence(ab $\longrightarrow$ c) = 1.0 is given.

Which statement is true?

- (a) None of the other answers is correct.
- (b) If sup(c) = 1.0, then sup(ab) cannot take any value.
- (c) P(abc) must be 1.0.
- (d) If sup(ab) = 1.0, then c must be present in all transactions.
- (e) sup(c) can be 0.
- (f) If a transaction contains ab, it will optionally contain c.

Given an association rule r: A $\longrightarrow$ B, lift(r) is defined as $\frac{conf(r)}{sup(B)}$, where conf(r) is the confidence of r, $\frac{sup(A,B)}{sup(A)}$ and sup(X) is the support of X.

Which of the following statements is correct?

- (a) lift(r) > 0 implies positive correlation
- (b) None of the other statements is correct
- (c) lift(A $\longrightarrow$ B) $\neq$ lift(B $\longrightarrow$ A)
- (d) lift(r) $\in$ [-1, 1]
- (e) lift(r) < 0 implies independence
- (f) if lift(A $\longrightarrow$ B) > 1, then lift(A $\longrightarrow$ not B) < 1

The support of a set of items A (sup(A)) is the fraction of transactions that contains A and is represented by P(A).

The confidence of an association rule A $\longrightarrow$ B is the conditional probability P(B|A).

The following rule ab $\longrightarrow$ c with confidence(ab $\longrightarrow$ c) = 1.0 is given.

Which statement is true?

- (a) If sup(c) = 1.0, then sup(ab) cannot take any value.
- (b) If sup(ab) = 1.0, then c must be present in all transactions.
- (c) If a transaction contains ab, it will optionally contain c.
- (d) None of the other answers is correct.
- (e) P(abc) must be 1.0.
- (f) sup(c) can be 0.

An itemset I is closed if none of its immediate supersets has the same support as I.

An itemset I is represented by a collection of literals (e.g. abc) and a number representing its support count (e.g. abc: 12).

After the analysis of a transactional dataset, the list of frequent itemsets found (with support counts) is the following:

length=1 a:150 b:160 c:180 d:150 e:150
length=2 ab:140 ac:120 ad:130 ae:140
length=3 abc:120 acd:110 abe:130
length=4 abce:120

Which of the following statements is true?

- (a) The only non-closed itemsets are **ac**, **ad**
- (b) The only non-closed itemsets are **ac**, **abc**
- (c) The only non-closed itemsets are **ac**, **ad**, **abc**, **abce**
- (d) The only non-closed itemsets are **ac**, **ad**
- (e) The only closed itemsets are **ac**, **abc**
- (f) None of the other answers is correct

The confidence of a rule is defined as:

$$confidence(A \rightarrow B) = \frac{sup(A,B)}{sup(A)}$$

where sup(x) indicates the fraction of transactions where x occurs (i.e. the support).

Let R: a,b -> c,d be an association rule.

In the transactional database you find the following records:

a b c d
a e f
a c d
a b
b c d f
b c d e
a e
b f

Which of the following statements is true?

- (a) confidence(R) = 8/6
- (b) confidence(R) = 24/25
- (c) confidence(R) = 2/3
- (d) confidence(R) = 1/3
- (e) None of the answers is correct
- (f) confidence(R) = 1/2

An itemset is defined to be frequent maximal if none of its immediate supersets (i.e. including one more item) is frequent.

After the analysis of a transactional dataset, the complete set of maximal frequent itemsets with minsup=900 are:

abc: 930

def: 980

Which of the following statements is true?

- (a) sup(ab) can be in range [900, 929], sup(abcd) cannot be in range [0, 930]
- (b) sup(ab) can be a value in range [900, 929], sup(cd) cannot be in range [900, 930]
- (c) sup(ab) can be a value in range [930, 1000], sup(abcd) can be a value in range [900, 930]
- (d) None of the other answers is correct
- (e) sup(ab) can be a value in range [930, 1000], sup(cd) cannot be in range [900, 930]
- (f) sup(ab) can be a value in range [900, 929], sup(abcd) can be a value in range [900, 930]

Given the transactional dataset shown in the figure below, apply the Apriori algorithm to extract all frequent itemsets.

The value of minsup is 2.

An itemset is considered to be frequent if its support count is equal to or higher than the minsup.

Question 1) List all frequent itemsets having length 2, along with their support count.

Question 2) List all itemsets of length 3 that have been generated by Apriori after the join and prune steps, before counting their support on the database.

Question 3) List all frequent itemsets that are **NOT** closed, along with their support count.

Transactions
ab
cde
bc
ab
acd
abce
de
acde
ac
de

(a) sup(B) < 1/2

(b) Not enough information is provided to address the question

(c) None of the other answers is correct

(d) sup(B) > 1/2

(e) sup(B) > 1/10

(f) sup(B) > 1/5

(g) sup(B) < 1/5

(h) sup(B) < 1/10

For a rule $A \rightarrow B$, the following can be computed:

$$confidence(A \rightarrow B) = \frac{sup(A,B)}{sup(A)}$$

$$lift(A \rightarrow B) = correlation(A \rightarrow B) = \frac{confidence(A \rightarrow B)}{sup(B)}$$

Where sup(X) is the support of an itemset X, i.e. the fraction of transactions that contain X.

A frequent itemset is maximal if none of its supersets is frequent.

You are given a transactional dataset. You extract all frequent itemsets using a minimum support of 0.1 (an itemset is considered frequent if it appears in at least 10% of the transactions).

You find that the itemset A = {a1, a2, … } has support sup(A) = 0.5, and it is maximal.

Which of the following constraints on the support of B = {b1, b2, …} guarantees that the rule $A \rightarrow B$ has positive correlation?

If applicable (i.e. a solution can be found and is among the provided answers), choose the most permissive constraint, which is the constraint that produces the largest possible range of permissible values. For example, if both sup(B) > x and sup(B) > y satisfy the request, you must choose sup(B) > min(x, y). Likewise if the options are in the form sup(B) < x and sup(B) < y, choose sup(B) < max(x, y).

You are given the following list of transactions.

cd
abde
bcde
abcd
abc
d
abcd
cd
abde
bd

Write the following:

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

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

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

An itemset is maximal if it is frequent and none of its supersets is frequent

After extracting the frequent itemsets from a transactional dataset, you find that -- among others -- the itemsets ABC and BD are maximal.

Which of the following statements is correct?

- (a) AC is frequent ✓
- (b) ABCD is frequent
- (c) ABD is maximal
- (d) AC is maximal
- (e) ABCD is frequent only if BCD is frequent
- (f) None of the other statements is correct
- (g) ABD is frequent
- (h) ABCD is maximal

D E
A C D E
A B C E
A B C
C E
A C D
C D
A C D
A C D E
B D

Given the transactional dataset shown in the figure below, apply the Apriori algorithm to extract all frequent itemsets.

The value of minsup is 2 (an itemset is frequent if it appears in at least 2 transactions).

An itemset is considered to be frequent if its support count is equal to or higher than the minsup.

1. List all frequent itemsets having length 2, **along with their support count**.
2. List all candidate itemsets of length 3 that have been generated by Apriori **after the join and prune steps**, before counting their support in the database.
3. List all frequent itemsets that are **not** closed, along with their support count.

3 points (no penalty for a wrong answer)

You are given the following list of transactions.

```
bcd
acde
bd
b
abe
e
abe
bd
d
abde
```

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. AD-CPB (AD - Conditional Pattern Base)

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

Classification

1.5 points, -15% penalty for a wrong answer

The true positive rate (TPR) and false positive rate (FPR) are defined as:

$$TPR = \frac{TP}{TP+FN}$$

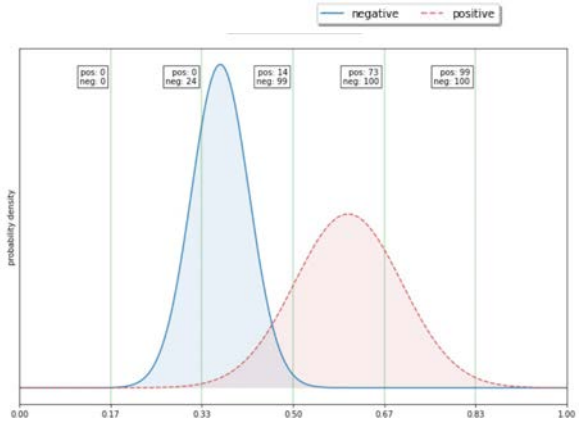
$$FPR = \frac{FP}{FP+TN}$$

Where TP = true positive, FP = false positive, TN = true negative, FN = false negative

The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the TPR (y axis) against the FPR (x axis) at various threshold settings.

A binary classifier (output: positive, negative) produces a posterior probability x = P(class=positive | t) for any test instance t. The class label is obtained based on a discrimination threshold applied to *x* . Values above the threshold are predicted *positive* , values below the threshold are predicted *negative* .

The following plot shows how the posterior probabilities are distributed within a dataset comprised of 200 records (100 positive records, 100 negative records).



- (a) (1/2, 1/2), (100/176, 76/176)
- (b) (1, 0), (76/100, 0)
- (c) (100/101, 1), (100/127, 1)
- (d) (86/100, 1/100), (27/100, 0)
- (e) The information provided is not sufficient to answer the question
- (f) (1, 0), (0, 73/127)

In particular, the 5 boxes in the upper part of the figure show the cumulative number of records that belong to the positive (pos) and negative (neg) classes, from to the $-\infty$ threshold defined by the respective vertical line.
For example, there are 14 positive records and 99 negative records that have $x \leq 0.5$.

Consider the ROC curve for this classifier. Which of the following pairs of points belong to the curve? (each of the points is provided as a pair of (x, y) coordinates).

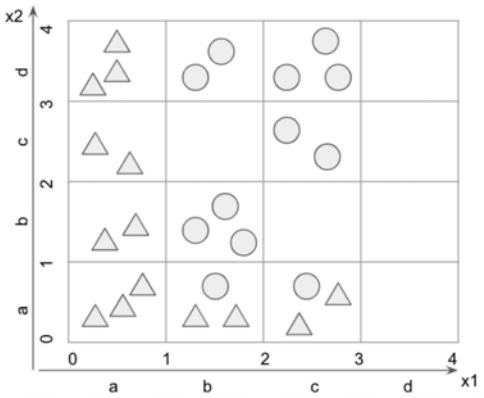
Let X be a feature vector, $X \in \mathbb{R}^N$, ie X = (x1, x2, ..., xn).

Naive Bayes classifiers compute the output class based on the following definition:

P(class|X)=P(x1|class)*P(x2|class)*...*P(xn|class)*P(class) It is

given the dataset represented in figure, where x1 and x2 are two continuous features, while triangle and circle are two class labels.

The two features are discretized in four bins [a,b,c,d], depicted in the figure with a grid.



Suppose that a Naive Bayes classifier has been trained on the provided dataset, which contains 14 triangles and 12 circles.

Classify the test data sample X with features x1=2.5, x2=1.5
Write in the box the following values (**one answer per row**):

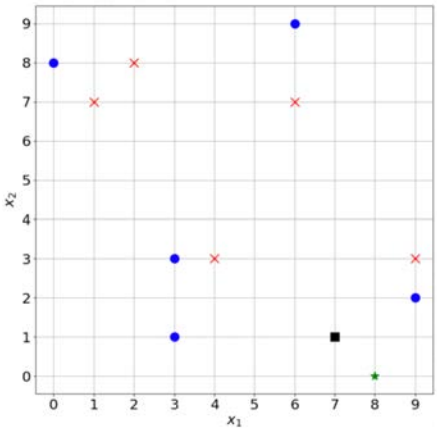
1. P(triangle|X) = ?
2. P(circle|X) = ?
3. The class assigned to P (circle or triangle?)

The KNN algorithm can assign a class probability based on the votes assigned to each class. If $vote_Y(x)$ represents the unnormalized vote 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 Chebyshev distance (L_∞ norm) is the limit case for $r \to \infty$ of the Minkowski distance, as shown below for two points $x, y \in \mathbb{R}^n$:

$$lim_{r\to\infty}\left(\sum_{i=1}^{n} |x_i - y_i|^r\right)^{\frac{1}{r}} = max_{i=1}^{n}|x_i - y_i|$$

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 Chebyshev 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, when neighbors' votes are weighted uniformely?

Question 2) What are the class probabilities for A and B, if the vote cast by each neighbor is $w = \frac{1}{1+distance}$?

Use the following notation:

```
A1) A=class_for_A, B=class_for_B
A2)
p(x|A)=prob_cross_A, p(o|A)=prob_circle_A
p(x|B)=prob_cross_B, p(o|B)=prob_circle_B
```

For example

```
A1) A=circle, B=circle
A2)
p(x|A)=0.5, p(o|A)=0.5
p(x|B)=0.9, p(o|B)=0.1
```

A binary classifier is trained to separate between images of cats and dogs. The test set used to evaluate this model is balanced, with 10,000 images of dogs and 10,000 images of cats.

The classifier only predicts 50 images as being cats. All of those predictions are correct.

What can be said about such a classifier?

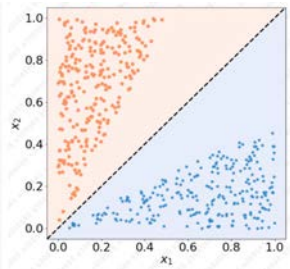
- (a) None of the other answers is correct
- (b) It has low precision for the class "cat"
- (c) It has high recall for the class "dog"
- (d) It has high F1 score for the class "dog"
- (e) It has high recall for the class "cat"
- (f) It has high accuracy
- (g) It has high precision for the class "dog"
- (h) It has high F1 score for the class "cat"

The following image represents a two-dimensional dataset with two numerical features, x1 and x2, and two classes, blue (star) and orange (circle).

A classifier is trained to distinguish between the blue (star) class and the orange (circle) one.

The dashed line represents the decision boundary identified by the classifier to separate between the two classes.

Which of the following statements is correct?



- (a) A decision tree can generate the boundary represented in the figure, when trained with the feature x1 / x2
- (b) A decision tree can generate the boundary represented in the figure, when trained with the feature x1
- (c) A decision tree can generate the boundary represented in the figure, when trained with the feature x1 * x2
- (d) A decision tree can generate the boundary represented in the figure, when trained with the features x1 and x2
- (e) A decision tree can generate the boundary represented in the figure, when trained with the feature x2
- (f) A decision tree can generate the boundary represented in the figure, when trained with the feature x1

A classification model can be tuned on P hyperparameters. You identify a subset of p < P hyperparameters that require tuning. For each hyperparameter, you identify n candidate values.

For n = 10, p = 5, P = 20, which of the following statements is correct regarding the number of configurations to be assessed?

- (a) Introducing an additional value for each hyperparameter introduces fewer new configurations to be assessed than if introducing an additional hyperparameter (with n candidate values)
- (b) Introducing an additional hyperparameter (with n candidate values) does not increase the number of configurations to be assessed
- (c) Introducing an additional candidate value for each hyperparameter does not increase the number of configurations to be assessed
- (d) The number of candidate values n cannot exceed P
- (e) Introducing an additional hyperparameter (with n candidate values) introduces fewer new configurations to be assessed than if introducing an additional value for each hyperparameter
- (f) It is advisable to always choose p = P and the largest possible value for n

Consider the training process of a Random Forest classifier on a dataset X characterized by a set of features F.

What is used to train *each individual tree* of the forest?

- (a) All the records from X and all the features F. However, each tree is initialized to a different random state
- (b) None of the other answers is correct
- (c) All the records from X and a subset of features from F. However, only a subset of X is considered at each split
- (d) A sample of records from X and a sample of features from F. The samples of records and features can overlap among the trees
- (e) A sample of records from X and a sample of features from F. The samples of records and features do not overlap among the trees
- (f) A sample of records from X and a sample of features from F. The samples of records can overlap among the trees but the samples of features do not

A classifier is incremental if, when adding a single new point in the training set, the training algorithm analyzes the new data point and does not require the availability of the training set.

Inference time is the time required to classify a new test object.

Which of the following statements on the Naive Bayes classification algorithm is correct?

- (a) The inference time of Naive Bayes is proportional to the training set size
- (b) Naive Bayes requires storing the entire training set to make new inferences
- (c) Naive Bayes cannot be used with continuous attributes
- (d) The Naive hypothesis is used to compute the prior probability (i.e. p(C) for a class C)
- (e) Naive Bayes is not an incremental algorithm
- (f) None of the other answers is correct

Consider the training process of a Random Forest classifier on a dataset X characterized by a set of features F.

What is used to train *each individual tree* of the forest?

- (a) A sample of records from X and a sample of features from F. The samples of records and features do not overlap among the trees
- (b) All the records from X and a subset of features from F. However, only a subset of X is considered at each split
- (c) None of the other answers is correct
- (d) All the records from X and all the features F. However, each tree is initialized to a different random state
- (e) A sample of records from X and a sample of features from F. The samples of records and features can overlap among the trees
- (f) A sample of records from X and a sample of features from F. The samples of records can overlap among the trees but the samples of features do not

A classifier is incremental if, when adding a single new point in the training set, the training algorithm analyzes the new data point and does not require the availability of the training set.

Inference time is the time required to classify a new test object.

Which of the following statements on the Naive Bayes classification algorithm is correct?

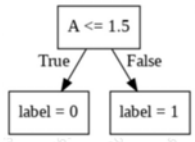
- (a) Naive Bayes is not an incremental algorithm
- (b) Naive Bayes requires storing the entire training set to make new inferences
- (c) Naive Bayes cannot be used with continuous attributes
- (d) None of the other answers is correct
- (e) The inference time of Naive Bayes is proportional to the training set size
- (f) The Naive hypothesis is used to compute the prior probability (i.e. p(C) for a class C)

You are training a decision tree classifier and need to evaluate its performance. To this end, you use 3-fold cross-validation. The following are the decision trees obtained during the validation -- the fold used for testing is shown with each tree.

Fold 1

	A	B	label
0	0	9	1
1	6	2	1
2	0	2	0
3	2	5	0

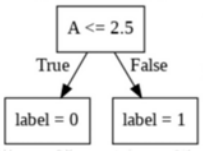
Decision tree 1



Fold 2

	A	B	label
0	9	6	1
1	9	4	1
2	0	9	0

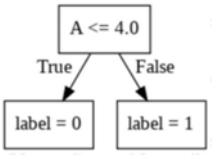
Decision tree 2



Fold 3

	A	B	label
0	8	4	1
1	3	8	1
2	0	5	0

Decision tree 3



1. Compute the accuracy for each fold. Use the syntax:
   fold1: <accuracy>
   fold2: <accuracy>
   fold3: <accuracy>
2. Compute the overall accuracy. Use the syntax: *overall: <accuracy>*

Example:

fold1: 0.25
fold2: 0.3
fold3: 0.35
overall: 0.3

A random forest is trained on a problem with C classes (0, 1, … C-1), using K decision trees.

For each input point x, each decision tree returns a class probability for each of the C classes.

The random forest assigns a class label to each point using soft voting.

Soft voting works as follows: given a point x, each of the K decision trees produces a class probability for each of the C classes. For each class, the respective K probabilities are then added together. Finally, the random forest assigns to x the label of the class with the highest sum.

You are given a test set X containing N points.

The following syntax will be used:

- $y_i \in \{0, 1, \dots, C-1\}^N$ represents the vector of labels predicted for all N instances of X by the i-th decision tree

When applying this random forest to X, we notice that $y_1 = y_2 = y_3 = \dots = y_k$.

Which of the following statements is true? (Multiple answers may be true)

Scegli una o più alternative:

- [ ] (a) There is no benefit in having an ensemble, since all predictions by the random forest will be the same obtained with a single decision tree
- [ ] (b) All K models have learned the same tree even though they have been trained on different subsets of the training set, hence the same predictions
- [ ] (c) The optimal number of features used at each split is of $\sqrt{C}$
- [ ] (d) For a different test set Z, different decision trees in the random forest may yield different predictions
- [ ] (e) The optimal number of features used at each split is of $\sqrt{N}$
- [ ] (f) All K decision trees have been trained on the same subsets of data, hence the same predictions

You are building a K-NN classifier A on a data set D split into a training set $D_{A,tr}$ and a test set $D_{A,te}$, such that $D = D_{A,tr} \cup D_{A,te}$.

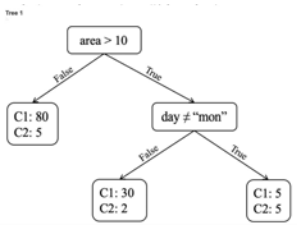
Suppose you build a second K-NN classifier B using the same hyper-parameters and the same dataset D, split into a training set $D_{B,tr}$ and a test set $D_{B,te}$, such that $D_{A,tr} \cup D_{A,te} = D_{B,tr} \cup D_{B,te}$.

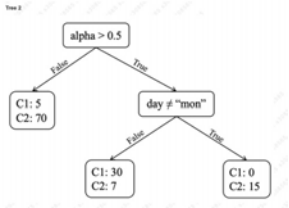
Which of the following statements is true?

- (a) None of the other statements is correct
- (b) A and B will have the same performance on the respective test sets since they are trained using the same hyperparameters
- (c) We can obtain a meaningful evaluation of A by testing it on $D_{B,te}$
- (d) If $D_{A,tr} = D_{B,tr}$, A and B will make the same predictions on the respective test sets
- (e) A and B will have the same performance on the respective test sets since they are evaluated on the same test sets
- (f) If we choose the same K for both classifier A and B, they will make the same predictions on the respective test sets

Given a point x to classify by means of a random forest, each tree of the ensemble assigns to point x a probability for each class. For each tree, the probability for a class C is given by the fraction of the training set points at the leaf reached by point x that belong to class C. The random forest assigns to point x the class with the highest average probability across all trees.

Consider the Random Forest shown in the figure below, which is composed of two trees. The represented trees show the split conditions and, for each leaf, the number of training samples that belong to each class (i.e., after applying Hunt's algorithm).





Write in the answer box the predicted class vector of the test samples below.

	Importance	alpha	area	day
x1	high	0.7	5	wed
x2	low	0.8	100	mon
x3	low	0.4	50	fri
x4	high	0.6	20	wed

Consider the training process of a Random Forest classifier on a dataset X characterized by a set of features F.

What is used to train *each individual tree* of the forest?

- (a) A sample of records from X and a sample of features from F. The samples of records and features can overlap among the trees
- (b) None of the other answers is correct
- (c) All the records from X and all the features F. However, each tree is initialized to a different random state
- (d) A sample of records from X and a sample of features from F. The samples of records and features do not overlap among the trees
- (e) A sample of records from X and a sample of features from F. The samples of records can overlap among the threes but the samples of features do not
- (f) A sample of records from X and all the features F. However, only a subset of F is considered at each split

A KNN classifier is trained on the training set in the table below, which only contains categorical features.

	width	weight	speed	Class
x1	big	light	fast	A
x2	big	heavy	fast	B
x3	small	heavy	slow	B
x4	big	heavy	slow	A
x5	small	light	slow	A

With categorical features, the distance between two samples *a* and *b* can be computed as the number of features with different values:

$$dist(a,b) = \sum_i \ell(a_i, b_i)$$

Where:

$$\ell(m,n) = \begin{cases} 0 & if\ m = n \\ 1 & otherwise \end{cases}$$

For each neighbor X_i of x the vote is weighted as follows:

$$weight(x, x_i) = \frac{1}{dist(x, x_i) + 0.5}$$

Given the test point below, write in the answer box:

1. The list of the 3 neighbors of t1
2. The class assigned to t1 with K = 3

	width	weight	speed	Class
t1	big	heavy	fast	?

Use the following notation:
neighbors={list of neighbors for t1}
class=class assigned

Example:
neighbors={point1 point2 point3}
class=A

A classifier is incremental if, when adding a single new point in the training set, the training algorithm analyzes the new data point and does not require the availability of the training set.

Inference time is the time required to classify a new test object.

Which of these statements is correct?

- (a) Decision tree and Naive Bayes classifiers are not incremental. However, when adding a data point to the training set, the inference time of a decision tree does not increase.
- (b) Decision tree is incremental, while Naive Bayes is not. However, when adding a data point to the training set, the inference time of Naive Bayes does not increase.
- (c) Naive Bayes is incremental. When adding a data point to the training set, the inference time of Naive Bayes does not increase.
- (d) None of the answers is correct.
- (e) Decision tree is incremental. Moreover, when adding a data point to the training set, the inference time of the decision tree does not increase.
- (f) Naive Bayes is incremental, while Decision tree is not. When adding a data point to the training set, the inference time of Decision tree always increases.

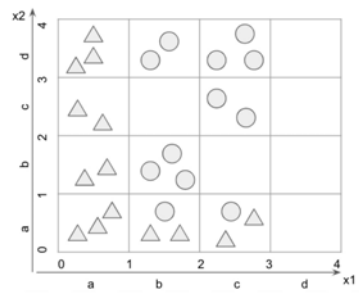
Let X be a feature vector, $X \in \mathbb{R}^N$, i.e. X = (x1, x2, ..., xn).

Naive Bayes classifiers compute the output class based on the following definition:

P(class|X)=P(x1|class)*P(x2|class)*...*P(xn|class)*P(class)

It is given the dataset represented in figure, where x1 and x2 are two continuous features, while triangle and circle are two class labels.

The two features are discretized in four bins [a,b,c,d], depicted in the figure with a grid.



Suppose that a Naive Bayes classifier has been trained on the provided dataset, which contains 14 triangles and 12 circles.

Classify the test data sample X with features x1=2.5, x2=1.5
Write in the box the following values (**one answer per row**):

1. P(triangle|X) = ?
2. P(circle|X) = ?
3. The class assigned to X (circle or triangle?)

You are given a dataset containing 300,000 records divided into 3 classes with the following schema:

```
Id | value_0 | ... | value_100 | Label
```

You have to build a classification model to avoid overfitting. Which is the best strategy to tune and test the classifier?

(a)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
kf = KFold(n_splits=4)
for train, test in kf.split(X_train):
    # Train and Validate
# Test the model
```

(b)

```
loo = LeaveOneOut()
for train_index, test_index in loo.split(X):
    # Train and Test
```

(c)

```
tscv = TimeSeriesSplit(n_splits=4)
for train, test in tscv.split(X):
    # Train and Test
```

(d)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
# Train and test the model
```

You are given a decision tree trained on a dataset with categorical features only.

Which statement is true?

- (a) Each internal node of a decision tree can have at most C children, where C is equal to the number of attributes of the training set
- (b) Each internal node of a decision tree can have at most C children, where C is equal to the number of distinct values of the attribute associated with the internal node
- (c) Each internal node of a decision tree can have at most C children, where C is equal to the number of training records
- (d) Each internal node of a decision tree can have at most C children, where C is equal to the number of distinct class labels

The matrix in the figure below represents the distances between data points, while the table on the right lists the ground truth labels. Points a, b, c and d represent training samples, while x_1 and x_2 are test samples.

	a	b	c	d	x1	x2
a	0	1	2	3	2	2
b	1	0	2	1	2	5
c	2	2	0	3	4	2
d	3	1	3	0	1	4
x1	2	2	4	1	0	4
x2	2	5	2	4	4	0

sample	ground truth
a	True
b	True
c	False
d	False

Question 1) If the value of K is set to 2, what is the number of distances computed by the K-Nearest- Neighbor algorithm to obtain the label for x_1?

- The number of computed distances is 2
- The number of computed distances is 4
- The number of computed distances is at most 2
- The number of computed distances is 5

Question 2) Apply a K-Nearest-Neighbor classifier and use K=3 neighbors as hyperparameter. Use a weighted voting with the following schema:

$$vote(C, x) = \frac{1}{K_c} \sum_{n_c \in C} \frac{1}{d^2(n_c, x)}$$

Where C is a class label, x is the sample being labeled, K_c is the number of neighbors belonging to class C ($Kc \leq K$), n_c is one of the K_c neighbors of x belonging to class C, $d(n_c, x)$ is the distance between n_c and x.

What is the value of $vote(True, x_2)$?

$vote(True, x_2)$ = ✕

Question 3) Apply a K-Nearest-Neighbor classifier and use $K = 3$ neighbors as hyperparameter. Use the voting schema of the previous question and label the samples x_1 and x_2 with the class having the highest vote.

x_1: ✕

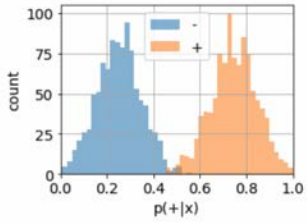
x_2: ✕

A dataset is comprised of 2,000 points assigned to either of two labels: positive (+) or negative (-). The dataset is well-balanced, with 1,000 points belonging to the positive class, and 1,000 points belonging to the negative class.

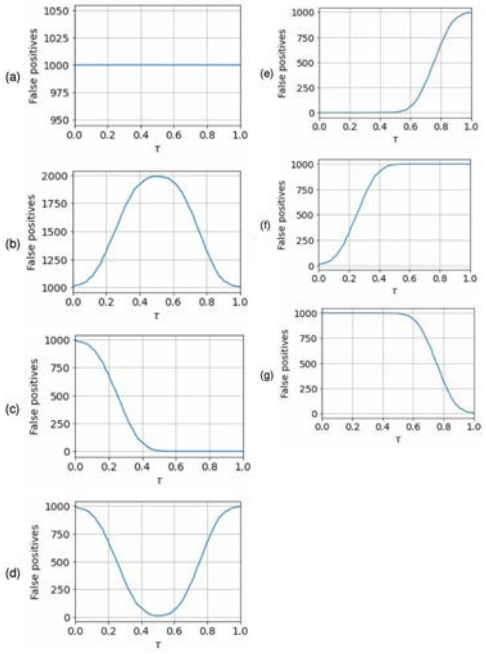
A classifier trained on this dataset produces, for a given point x, a probability of belonging to the positive class p(+|x). The probability p(-|x) can be computed as 1 - p(+|x).

A threshold value τ can be defined to assign a class to any given point x. In particular, if p(+|x) > τ the point is labelled as positive; negative otherwise.

The figure below shows how the positive and negative points are distributed w.r.t. p(+|x). In particular, the points known to have a negative ground truth are shown in the blue (left-most) distribution, whereas the points having a positive ground truth are shown in the orange (right-most) one.



Which of the following figures correctly represents the number of false positives obtained as τ varies between 0 and 1?



A random forest (RF) and a decision tree (DT) classifiers have been trained on a dataset D.

You accidentally train the random forest using only 1 decision tree.

Which of the following statements is correct? Choose all correct answers (multiple answers may be correct)

Note: when referring to "RF" in the answers, it means the specific 1-tree random forest mentioned above.

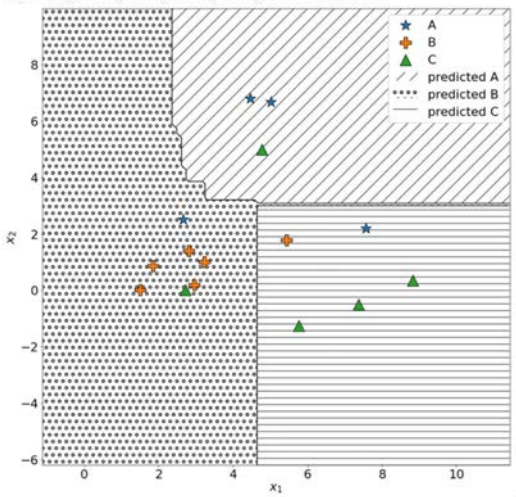
Scegli una o più alternative:

- [ ] (a) The RF and the DT will produce different results, since the voting scheme of the RF will affect its predictions
- [ ] (b) None of the other statements is correct
- [x] (c) The RF and the DT will produce different results, since the pool of features used changes for each split of RF ✓
- [ ] (d) The RF and the DT will produce different results, since the data used for training them will differ
- [ ] (e) The RF and the DT will produce the same results
- [ ] (f) The RF and the DT will produce different results, since the pool of features used changes for each split of DT

- recall(X) is the fraction of correct predictions among the samples with actual class X
- precision(X) is the fraction of correct predictions among the samples predicted with class X

A random forest classifier has been trained on a 2-dimensional dataset (features x_1, x_2). Each point in the dataset is labeled as either A, B or C (star, cross, triangle respectively).

The following figure represents a test set that is used to validate the classifier.



The decision boundaries of the model are shown in the figure:

- Diagonal lines represent areas of the input space where the model predicts class A
- Small circles represent areas of the input space where the model predicts class B
- Horizontal lines represent areas of the input space where the model predicts class C

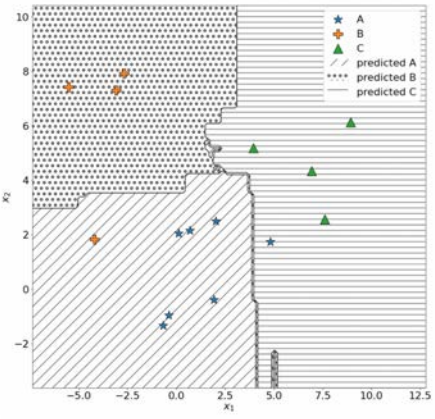
Write in the box below:

- accuracy(B)
- accuracy(C)
- recall(C)

recall(X) is the fraction of correct predictions among the samples with actual class X
precision(X) is the fraction of correct predictions among the samples predicted with class X

A random forest classifier has been trained on a 2-dimensional dataset (features x_1, x_2). Each point in the dataset is labelled as either A, B or C (star, cross, triangle respectively).

The following figure represents a test set that is used to validate the classifier.



The decision boundaries of the model are shown in the figure:

- Diagonal lines represent areas of the input space where the model predicts class A
- Small circles represent areas of the input space where the model predicts class B
- Horizontal lines represent areas of the input space where the model predicts class C

Write in the box below:

precision(B)

precision(C)

recall(B)

Rand index is defined as follows, given a ground truth y_true and a cluster assignment y_pred.

$$RI(y_true, y_pred) = \frac{a+b}{\binom{n}{2}}$$

Where

- a is the number of pairs of objects with different labels in y_true and different labels in y_pred
- b is the number of pairs of objects having the same label in y_true and the same label in y_pred
- n = number of elements in y_true and y_pred

You are given an **unlabelled** dataset X (n rows, m columns) and are required to run a cluster analysis using DBSCAN. Which of the following statements is correct?

- (a) The number of clusters to be extracted is a hyperparameter that can be fine-tuned based on the knee of the curve of the sum of squared errors
- (b) The number of clusters extracted is independent of the hyperparameter ϵ (`eps`, maximum distance between two points for them to be considered neighbors)
- (c) DBSCAN may extract at most n clusters
- (d) You can use the Rand index to assess the quality of the resulting clusters
- (e) DBSCAN may extract at most m clusters
- (f) None of the other answers is correct

- Precision(C) is the fraction of correct predictions among the samples predicted with class C
- Recall(C) is the fraction of correct predictions among the samples with actual class C
- Accuracy is the fraction of correct predictions among all samples

Let y_pred, y_true be the prediction vector and the ground truth vector respectively.

```
y_true: [A A B C B A A A B A]
y_pred: [C A C A A B A B A B]
```

Write in the answer box:

- Accuracy
- Precision(A)
- Recall(A)

The Gini index of a node is computed as follows:

$$gini(node) = 1 - \sum_j P(j|t)^2$$

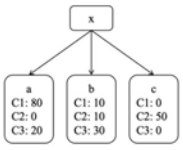
where $P(j|t)$ is the relative frequency of class j at node t

The Gini index of a *split* with parent p and children C_i is computed as follows:

$$gini(split) = \sum_i \frac{n_i}{n} gini(c_i)$$

where n_i is the number of records at child C_i and n is the number of records in p.

In the figure below it is shown a split x, with three children (a, b, c). For each child you are given the number of elements belonging to each of three classes (C1, C2, C3).



Write in the box below:
1. the Gini index of child a
2. the Gini index of the split x

Use the following notation:
gini(a)=value
gini(x)=value

Example
gini(a)=1/2
gini(x)=1/3

A binary classifier uses the following spherical decision boundary:

$$(x-3)^2 + (y-4)^2 + (z-1)^2 = r^2$$

Points strictly inside the sphere are classified with "Reject", otherwise they are classified with "Accept".

Assuming $r^2 = 15$, report the overall precision and recall for both the classes on the following test set.

x	y	z	Ground truth
2	1	3	Accept
0	2	-1	Accept
-3	1	1	Accept
1	3	-2	Reject
0	0	0	Accept

The Fowlkes-Mallows score* is a quality index designed for evaluating clustering predictions against a ground truth.

It is defined as follows:

$$FM_{score} = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}}$$

Where:

- TP = number of pairs of elements that are in the same set in the ground truth and in the same set in the prediction
- FP = number of pairs of elements that are in different sets in the ground truth and in the same set in the prediction
- FN = number of pairs of elements that are in the same set in the ground truth and in different sets in the prediction

Given the vectors:

```
y_pred = [1 1 2 3 2] # prediction
y_true = [1 3 2 3 2] # ground truth
```

Write in the box below the value of Fowlkes-Mallows score computed for y_pred and y_true.

- Precision(C) is the fraction of correct predictions among the samples predicted with class C
- Recall(C) is the fraction of correct predictions among the samples with actual class C
- Accuracy is the fraction of correct predictions among all samples

Let y_pred, y_true be the prediction vector and the ground truth vector respectively.

```
y_true: [C A A B C B A B A A]
y_pred: [A A C A C C C B A C]
```

Write in the answer box:

- Accuracy
- Precision(A)
- Recall(A)

Use the following notation:

```
accuracy=value
precision(A)=value
recall(A)=value
```

Example:

```
accuracy=1
precision(A)=1/3
recall(A)=0.25
```

The true positive rate (TPR) and false postive rate (FPR) are defined as:

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

Where TP = true positive, FP = false positive, TN = true negative, FN = false negative

The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the TPR (y axis) against the FPR (x axis) at various threshold settings.

A binary classifier (output: positive, negative) produces a posterior probability x = P(class=positive | t) for any test instance t. The class label is obtained based on a discrimination threshold applied to *x*. Values above the threshold are predicted *positive*, values below the threshold are predicted *negative*.

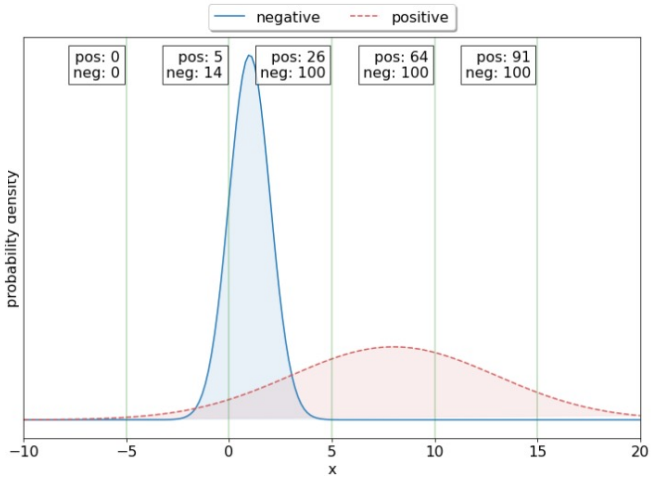
The following plot shows how the posterior probabilities are distributed within a dataset comprised of 200 records (100 positive records, 100 negative records).

In particular, the 5 boxes in the upper part of the figure show the cumulative number of records that belong to the positive (pos) and negative (neg) classes, from $-\infty$ to the threshold defined by the respective vertical line.

For example, there are 26 positive records and 100 negative records that have $x \leq 5$.

Consider the ROC curve for this classifier. Which of the following pairs of points belong to the curve? (each of the points is provided as a pair of (x, y) coordinates).

- (a) (1, 1), (5/100, 14/100)
- (b) The information provided is not sufficient to answer the question
- (c) (0, 1), (86/100, 95/100)
- (d) (86/100, 95/100), (0, 74/100)
- (e) (5/100, 86/100), (0, 1)
- (f) (5/100, 86/100), (91/100, 0)



The softmax activation function is defined as:

$softmax(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

for $i = 1, \ldots, K$ and $\mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$

Where $\mathbf{z}$ is the K-dimensional input vector.

Which of the following statements is true?

- (a) The output $softmax(\mathbf{z})_i$ can be computed using only z_i
- (b) None of the other statements is correct
- (c) Softmax is typically used for lasso regression
- (d) Softmax is typically used as an activation function for hidden layers of a neural network
- (e) Softmax is idempotent, i.e. $softmax(\mathbf{z})_i = softmax(softmax(\mathbf{z})_i)_i$
- (f) The output of softmax is a probability distribution of a continuous variable

- Precision(C) is the fraction of correct predictions among the samples predicted with class C
- Recall(C) is the fraction of correct predictions among the samples with actual class C
- Accuracy is the fraction of correct predictions

Let y_pred, y_true be the prediction vector and the ground truth vector respectively.

```
y_pred = [A B C A A B B A]
y_true = [B B C B A B C C]
```

Write in the answer box:

- Accuracy
- Precision(A)
- Recall(A)

- Precision(C) is the fraction of correct predictions among the samples predicted with class C
- Recall(C) is the fraction of correct predictions among the samples with actual class C

After training two models (**M1**, **M2**), we obtain the confusion matrices shown below. Consider class **b** only. Which of the following statements is true?

predicted class

true class	**M1**	a	b	c	d
	a	15	10	1	0
	b	0	50	5	5
	c	5	20	70	0
	d	5	10	0	60

predicted class

true class	**M2**	a	b	c	d
	a	14	10	0	2
	b	5	40	5	10
	c	5	5	80	5
	d	10	5	10	50

- (a) M1 has higher recall than M2 and lower precision.
- (b) M1 has higher recall than M2 and higher precision.
- (c) M1 has lower recall than M2 and lower precision.
- (d) M1 has lower recall than M2 and higher precision.
- (e) M1 has the same recall as M2 and lower precision.
- (f) None of the other answers is correct

You are given the ground truth labels of a dataset and the predicted values obtained with a clustering algorithm (`y_true` and `y_pred` respectively).

```
y_true = [1,1,2,2,3,3]
y_pred = [2,2,1,1,1,1]
```

Compute the Rand Index (RI) between `y_true` and `y_pred`.

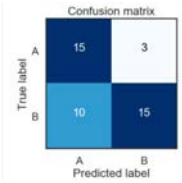
Remember that the Rand Index is given by:

$$RI = \frac{a+b}{DP} = \frac{TP+TN}{DP}$$

Where:

- a is the number of pairs of elements (without ordering) that are in the same set in `y_true` and in the same set in `y_pred`
- b is the number of pairs of elements (without ordering) that are in different sets in `y_true` and in different sets in `y_pred`
- DP is the total number of possible pairs in the dataset (without ordering)

You are given the confusion matrix in the figure below obtained after the validation phase of a classifier.



Question 1) Compute the accuracy of the classifier.

Accuracy:

Question 2) Compute the precision for class with label A.

Precision for A:

Given a set of n ground truth values y_i and predictions made by a regressor, $\hat{y}_i$, the Mean Absolute E rror (MAE) is defined as:

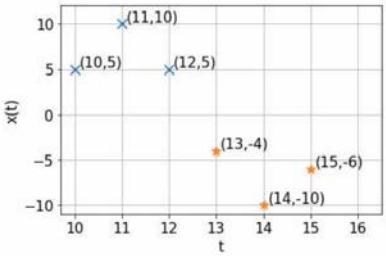
$$MAE = \frac{1}{n}\sum_i |y_i - \hat{y}_i|$$

You are given a 1-dimensional time series. You are interested doing time series forecasting (i.e. predicting the next values of the time series, based on the past ones).

Using a window containing the past 3 readings of the time series, x(t-2), x(t-1), x(t), you build the following model:

$$x(t+1) = a \cdot x(t-2) + b \cdot x(t-1) + c \cdot x(t) + d$$

Where a = -0.68, b = 0.26, c = 0.4, d = 0



The following figure represents the ground truth values for x(t), ($t \in [10, 15]$).

You use x(10), x(11), x(12) to predict x(13), x(14), x(15).
After acquiring the ground truth values for x(13), x(14), x(15), you validate your results by computing the mean absolute error.

Answer the following question:

1. What are the values predicted for x(13), x(14), x(15)?
2. What is the MAE of the regressor, as evaluated on x(13), x(14), x(15)?

To assess the quality of a classification result the following measures were defined:

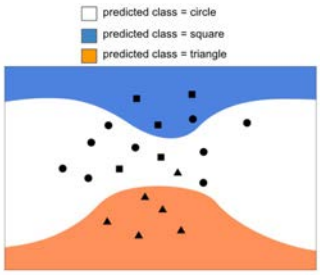
- accuracy: the percentage of correctly classified items
- precision(C): the percentage of correctly classified items among those predicted with class C
- recall(C): the percentage of correctly classified items among those with actual class C

Let C be a supervised classifier and I a two-dimensional input space. The points drawn from I belong to one of the three classes: *Circle*, *Square*, or *Triangle*. After the training stage, C learns the following decision boundaries:

Where the label assigned by C is given by the background color of the region in I.

Suppose that the symbols in the image are test points, whose shape represents the respective ground truth label (either a *Circle*, a *Square*, or a *Triangle*).

Which of the following statements is true?

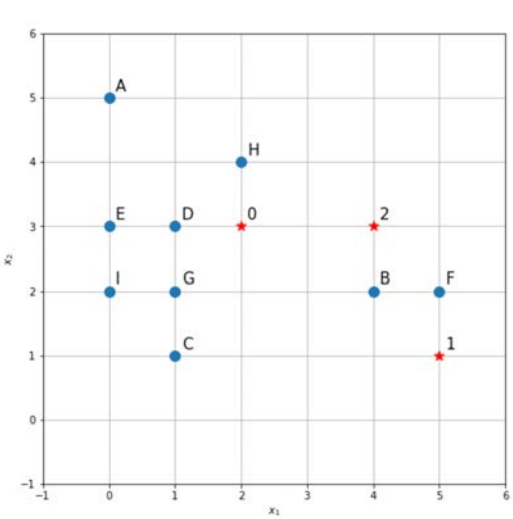


- (a) The precision for Square is ⅗
- (b) The precision for Triangle is 1
- (c) The precision for Circle is 1
- (d) None of the answers is correct
- (e) The recall for Square is ¾
- (f) The accuracy is 1

Clustering

You are given a dataset containing 9 points in 2 dimensions (x1, x2). Each point is labeled A through I.

You apply K-means clustering with K = 3. The figure below represents the 9 points (blue dots) and the initializations for the three centroids (red stars). Each centroid is labeled with a number (0, 1, 2).



What are the new centroids computed after 1 iteration of the K-means algorithm?

Use the Euclidean distance when computing any distance.

Write your answer in the box below, using the following syntax:

```
0: (x_10, x_20)
1: (x_11, x_21)
2: (x_12, x_22)
```

For example:

```
0: (5.3, 1.2)
1: (2.212, 2.345)
2: (0.1, 3.17)
```

The clusters assigned in a clustering U can be identified as a sequence of cluster identifiers $(u_1, u_2, \ldots, u_n)$, where n is the number of points to be labelled. u_i can be any of the values contained in S_U, the set of symbols used to identify each cluster. In other words, each cluster is identified by a unique symbol. S_U is the set of all such symbols.

Similarly, we can identify the cluster assigned in a clustering V as $(v_1, v_2, \ldots, v_n)$, with $v_i \in S_V$, S_V being the set of symbols used to identify the clusters of V.

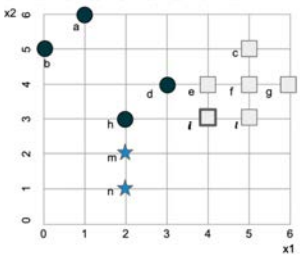
We define a as the number of pairs of points that have the same clusters in U and in V, and b as the number of points that have different clusters in U and in V. In other words, $a = |\{(i,j)\ s.t.\ u_i = u_j \wedge v_i = v_j\}|$ and $b = |\{(i,j)\ s.t.\ u_i \neq u_j \wedge v_i \neq v_j\}|$.

We define the rand index as:

$$R(U,V) = \frac{a+b}{\binom{n}{2}}$$

Which of the following statements about the Rand index is correct?

- (a) The Rand index is symmetric (ie R(U,V) = R(V,U))
- (b) The Rand index requires the two clusterings U and V to use the same symbols to identify the clusters (ie $S_U = S_V$)
- (c) The Rand index has values contained in the range [-1, +1]
- (d) A low Rand index implies that the two clusterings are in agreement with each other
- (e) The Rand index is not a suitable metric for evaluating the quality of a cluster
- (f) None of the other statements are correct
- (g) The Rand index requires the two clusterings U and V to have the same number of clusters (ie $|S_U| = |S_V|$)
- (h) The Rand index has values contained in the range $(-\infty, +\infty)$



For each data point p_i

let $intra(p_i)$ be the average dissimilarity of p_i with all other points within the same cluster

let $inter(p_i)$ be the lowest average dissimilarity of p_i to any other cluster, of which p_i is not a member

The Silhouette score of point pi is defined as:

$$silh(p_i) = \frac{inter(p_i) - intra(p_i)}{max(inter(p_i), intra(p_i))}$$

For two n-dimensional points $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the Manhattan distance is defined as follows:

$$dist(a,b) = \sum_i^n |a_i - b_i|$$

In the figure below, three different clusters are represented with the following shapes: circle, star, square. Use the Manhattan distance as the dissimilarity metric.

Consider point i belonging to the square cluster. Write in the answer box:

- intra(i)
- inter(i)
- silh(i)

For each data point p_i

- let $inter(p_i)$ be the average dissimilarity of p_i with all other points within the same cluster
- let $extra(p_i)$ be the lowest average dissimilarity of p_i to any other cluster, of which p_i is not a member

The Silhouette score of point p_i is defined as:

$$silh(p_i) = \frac{extra(p_i) - inter(p_i)}{max(inter(p_i), extra(p_i))}$$

Suppose we have the following three clusters:

- C_1 = {a, b, c}
- C_2 = {d, f}
- C_3 = {e, g}

The matrix represented in the figure below specifies the dissimilarity between each pair of points.
Compute and report in the answer box the Silhouette score of:

- point **b**
- point **e**

(Write one answer per line)

	a	b	c	d	e	f	g
a		1	2	5	6	7	9
b	1		3	4	6	8	8
c	2	3		7	6	6	7
d	5	4	7		7	2	7
e	6	6	6	7		9	1
f	7	8	6	2	9		5
g	9	8	7	7	1	5	

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).

Rand index is defined as follows, given a ground truth y_true and a cluster assignment y_pred.

$$RI(y_true, y_pred) = \frac{a+b}{\binom{n}{2}}$$

Where

- a is the number of pairs of objects with different labels in y_true and different labels in y_pred
- b is the number of pairs of objects having the same label in y_true and the same label in y_pred
- n = number of elements in y_true and y_pred

You are given an **unlabelled** dataset X (n rows, m columns) and are required to run a cluster analysis using DBSCAN. Which of the following statements is correct?

- (a) DBSCAN can extract no more than m clusters
- (b) The number of clusters extracted is independent of the hyperparameter ϵ (`eps`, maximum distance between two points for them to be considered neighbors)
- (c) You can use the Rand index to assess the quality of the resulting clusters
- (d) The number of clusters to be extracted is a hyperparameter that can be fine-tuned based on the knee of the curve of the sum of squared errors
- (e) DBSCAN can extract no more than n clusters
- (f) None of the other answers is correct

Rand index is defined as follows, given a ground truth y_true and a cluster assignment y_pred.

$$RI(y_true, y_pred) = \frac{a+b}{\binom{n}{2}}$$

Where

- a is the number of pairs of objects with different labels in y_true and different labels in y_pred
- b is the number of pairs of objects having the same label in y_true and the same label in y_pred
- n = number of elements in y_true and y_pred

You are given an **unlabelled** dataset X (n rows, m columns) and are required to run a cluster analysis using DBSCAN. Which of the following statements is correct?

- (a) None of the other answers is correct
- (b) You can use the Rand index to assess the quality of the resulting clusters
- (c) DBSCAN can extract no more than n clusters
- (d) The number of clusters extracted is independent of the hyperparameter ϵ (`eps`, maximum distance between two points for them to be considered neighbors)
- (e) The number of clusters to be extracted is a hyperparameter that can be fine-tuned based on the knee of the curve of the sum of squared errors
- (f) DBSCAN can extract no more than m clusters

1.5 points (15% penalty for a wrong answer)

The cosine similarity between two vectors u, v is:

$$similarity_{cos}(u, v) = \frac{u \cdot v}{||u|| \, ||v||}$$

The cosine distance between two vectors u, v is:

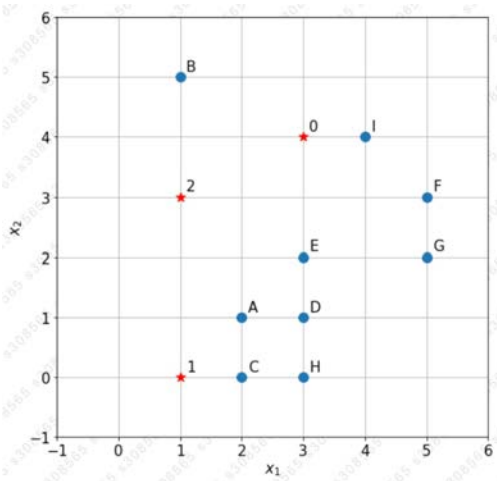
$$distance_{cos}(u, v) = 1 - similarity_{cos}(u, v)$$

What is the range of values that can be assumed by the cosine distance?

- (a) [-2, +2]
- (b) [0, +2]
- (c) $[0, +\infty)$
- (d) $(-\infty, 1]$
- (e) $(-\infty, +\infty)$
- (f) [-1, +1]
- (g) [0, +1]
- (h) [-1, 0]

You are given a dataset containing 9 points in 2 dimensions (x1, x2). Each point is labelled A through I.

You apply K-means clustering with K = 3. The figure below represents the 9 points (blue dots) and the initializations for the three centroids (red stars). Each centroid is labelled with a number (0, 1, 2).



What are the new centroids computed after 1 iteration of the K-means algorithm?

Use the Euclidean distance when computing any distance.

Write your answer in the box below, using the following syntax:

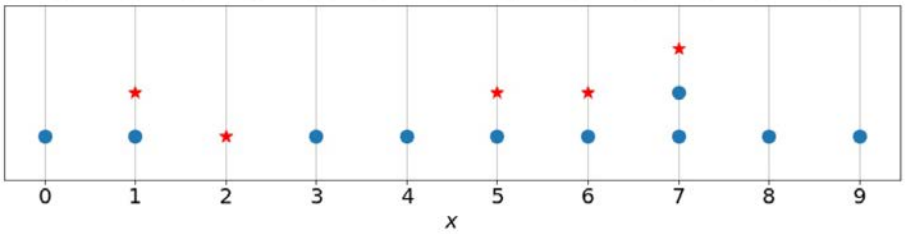
```
0: (x_10, x_20)
1: (x_11, x_21)
2: (x_12, x_22)
```

For example:

```
0: (5.3, 1.2)
1: (2.212, 2.345)
2: (0.1, 3.17)
```

1.5 points (no penalty for a wrong answer)

The following figure represents a 1-dimensional dataset. The blue circles represents the point of a dataset. The red stars represent the five initial centroids selected by a k-means algorithm. Please note that the vertical separation between points that share the same x has been introduced for visualization purposes (ie to avoid overlapping points).



Apply a single iteration of k-means. What are the new centroids computed?

Use the following syntax (you can use any order for the centroids):

```
centroid_A, centroid_B, centroid_C, centroid_D, centroid_E
```

For example:

```
10, 11, 12, 13, 14
```

```
1.
dist(Px, Py) = value
dist(Cz, Px) = value
dist(Ca, Cb) = value
...
2.
Cx = (Pa, Pb, ...)
Cy = (Px, Py, . ..)
...
```

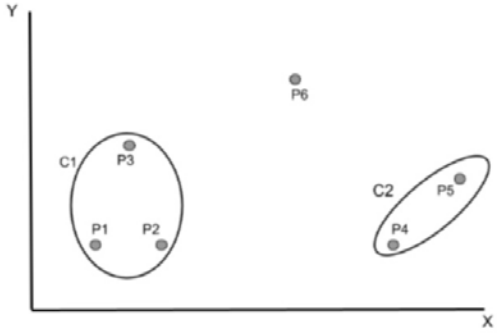
For example,

```
1.
dist(P1, P2) = 3
dist(C3, P4) = 4
dist(C1, C2) = 5

2.
C1 = (P1, P2, P3)
C2 = (P4, P5, P6)
C3 = (P7 , P8, P9)
```

An agglomerative clustering algorithm has been applied, using the Euclidean distance between centroids to measure cluster similarity.

After the first few steps, the following clusters have been obtained.



The following two-dimensional unlabelled dataset is given.

	X	Y
P1	2	2
P2	4	2
P3	3	5
P4	11	2
P5	13	4
P6	8	7

Answer the following questions:

1. What is the proximity matrix at this step? Write the answer as distances between pairs of points/clusters
2. What are the clusters obtained after applying an additional step?

Answer the questions in the box below, using the following syntax:

Answer the following questions:

1. What is the proximity matrix at this step? Write the answer as distances between pairs of points/clusters
2. What are the clusters obtained after applying an additional step?

Answer the questions in the box below, using the following syntax:

The following definitions refer to any two vectors $a = (a_1, a_2, \ldots a_n)$ and $b = (b_1, b_2, \ldots, b_n) \in \mathbf{R}^n$.

The cosine similarity is defined as:

$$cos(a, b) = \frac{a \cdot b}{||a|| ||b||}$$

The Minkowski distance of order p between is defined as:

$$D_p(a, b) = \left(\sum_{i=1}^{n} |a_i - b_i|^p\right)^{\frac{1}{p}}$$

You are given a categorical dataset with two independent features, x1 and x2. The two features can assume one of 13 and 7 distinct values respectively. More specifically x1 = (n00, n01, n02, ... n12) and x2 = (m00, m01, m02, ..., m06).

From this dataset, two records r1 = (n04, m06) and r2 = (n11, m01) are extracted.

Answer the following questions:

1. What is the cosine similarity between r1 and r2 (cos(r1, r2))?
2. What is the Minkowski distance of order 1 between r1 and r2 (D_1(r1, r2))?
3. What is the Minkowski distance of order 2 between r1 and r2 (D_2(r1, r2))?

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

```
cos(r1,r2)=value
D1(r1,r2)=value
D2(r1,r2)=value
```

For example:

```
cos(r1,r2)=0.5
L1(r1,r2)=0.4
L2(r1,r2)=0.3
```

You are given the following 2-dimensional dataset.

x1	x2
-6.7	-11.4
-3.3	-9.8
9.7	4.1
-4.9	-11.4
-4.1	-7.6
11.2	3.8
-4.2	-9.8
-6.6	-9.8
8.5	3.5
-4.7	-9.1

K-means is trained on this dataset using the following configuration of hyperparameters:

- k = 3
- Max iterations = 20
- Centroids initialization = random
- Number of separate runs = 5

Based on this information, write in the box below:

1. The number of distances computed at each iteration of k-means
2. The maximum number of distances computed for a single run of k-means
3. The maximum number of distances that need to be computed to train the model

Use the following syntax:

```
1. value1
2. value 2
3. value 3
```

For example:

```
1. 3
2. 6
3. 9
```

Kmeans converges when two subsequent iterations do not yield different cluster assignments

The j-th centroid at the t-th iteration is computed as follows:

$$m_j^{(t)} = \frac{1}{|S_j^{(t-1)}|} \sum_{p_i \in S_j^{(t-1)}} p_i$$

Where $S_j^{(t-1)}$ is the set of points that is closest to $m_j^{(t-1)}$.

The following figure represents a 1-dimensional dataset. Each blue circle represents a point of a dataset. The red crosses represent the three initial centroids selected by a k-means algorithm.

When two or more points overlap, a vertical separation has been introduced for visualization purposes. For example there are two points in the dataset having x = 5.

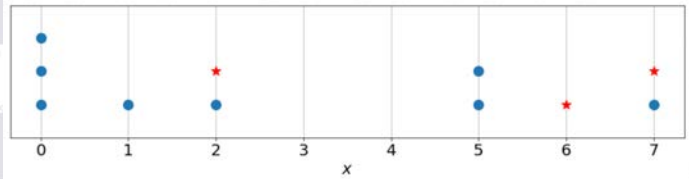
Question 1) Apply a single iteration of k-means. What are the new centroids computed?
Question 2) After the first iteration performed in Question 1, how many additional iterations of k-means will be applied until convergence?

Enter one answer per row. Use the following syntax (you can use any order for the centroids):

```
A1) centroid_A, centroid_B, centroid_C
A2) X
```

For example:

```
A1) 3, 4, 5
A2) 3
```



For two n-dimensional points $P_1 = (P_{11}, P_{12}, \dots P_{1n})$ and $P_2 = (P_{21}, P_{22}, \dots P_{2n})$ the Manhattan distance is defined as follows:

$$dist(P_1, P_2) = \sum_i |P_{1i} - P_{2i}|$$

Using the Manhattan distance, apply the DBSCAN algorithm to the following points in the bidimensional space.

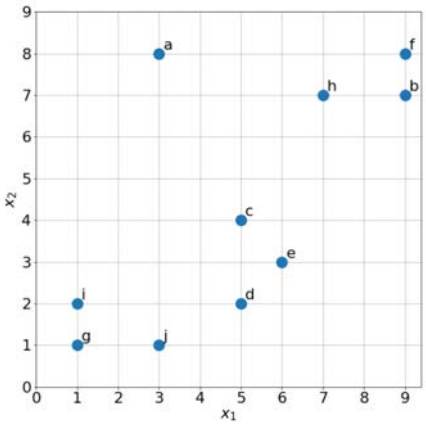
Use the following hyperparameters: ϵ = 2.5, minpoints=2 (at least 2 points as neighbors)

For each point write:

- The assigned label (N=noise, B=border, C=core)
- The assigned cluster id (order of cluster ids is not important, use -1 for noise points)

Use the following format: "point label cluster"
Example:

```
a N -1
b C 0
c C 0
d C 1
e B 1
```

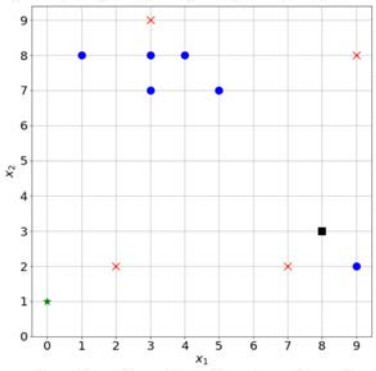


The KNN algorithm can assign a class probability based on the votes assigned to each class. If $vote_Y(x)$ represents the unnormalized vote assigned to class Y for sample x, then, for any class $c \in C$: $p(c|x) = \frac{vote_c(x)}{\sum_{i \in C} vote_i(x)}$

The Chebyshev distance (L_∞ norm) is the limit case for $p \to \infty$ of the Minkowski distance, as shown below for two points $x, y \in \mathbb{R}^n$:

$$lim_{r \to \infty} \left(\sum_{i=1}^n |x_i - y_i|^r\right)^{\frac{1}{r}} = max_{i=1}^n |x_i - y_i|$$

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 Chebyshev 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 the vote cast by each neighbor is $w = \frac{1}{1+distance}$?

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
```

Kmeans converges when two subsequent iterations do not yield different cluster assignments

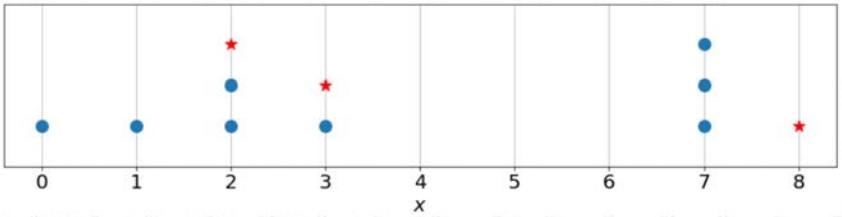
The j-th centroid at the t-th iteration is computed as follows:

$$m_j^{(t)} = \frac{1}{|S_j^{(t-1)}|} \sum_{p_i \in S_j^{(t-1)}} p_i$$

Where $S_j^{(t-1)}$ is the set of points that is closest to $m_j^{(t-1)}$.

The following figure represents a 1-dimensional dataset. Each blue circle represents a point of a dataset. The red crosses represent the three initial centroids selected by a k-means algorithm.

When two or more points overlap, a vertical separation has been introduced for visualization purposes. For example there are two points in the dataset having x = 2.



Question 1) Apply a single iteration of k-means. What are the new centroids computed?
Question 2) After the first iteration performed in Question 1, how many additional iterations of k-means will be applied until convergence?

Enter one answer per row. Use the following syntax (you can use any order for the centroids):

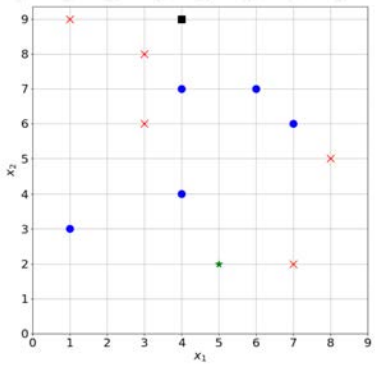
```
A1) centroid_A, centroid_B, centroid_C
A2) X
```

The KNN algorithm can assign a class probability based on the votes assigned to each class. If $vote_Y(x)$ represents the unnormalized vote 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 Chebyshev distance (L_∞ norm) is the limit case for $r \to \infty$ of the Minkowski distance, as shown below for two points $x, y \in \mathbb{R}^n$:

$$lim_{r\to\infty}\left(\sum_{i=1}^{n} |x_i - y_i|^r\right)^{\frac{1}{r}} = max_{i=1}^{n}|x_i - y_i|$$

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 Chebyshev 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 the vote cast by each neighbor is $w = \frac{1}{1+distance}$?

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
```

You run K-means (K = 5) on a dataset. After the algorithm converges, you obtain the following sums of squared errors (SSE) from the respective centroid, for each cluster:

SSE = { 500, 150, 15, 200, 900 }

Mean SSE: 353

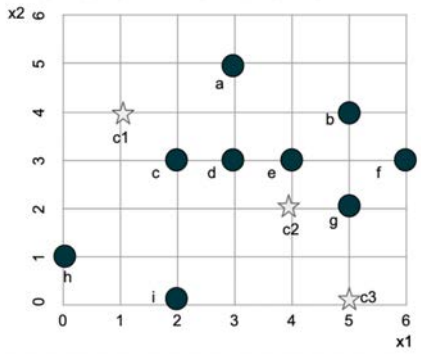
Which of the following statements is true?

- (a) The sum of squared errors is not a suitable metric for evaluating the quality of the clusters
- (b) None of the other answers is true
- (c) Running k-means with k = 10 on the same dataset will typically produce lower mean SSE
- (d) Using the Rand index will result in a better cluster evaluation
- (e) Running an additional iteration of k-means may lower the sums of squared errors obtained
- (f) The third cluster contains a lower number of points with respect to the others

For two n-dimensional points $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the Manhattan distance is defined as follows:

$$dist(a, b) = \sum_i^n |a_i - b_i|$$

The k-means algorithm is run on the points in figure, using the Manhattan distance and k=3. The initial positions of the centroids are represented by the three stars (c1, c2, c3).



Write in the answer box:

- The clusters obtained after 1 iteration
- The position of the centroids after 1 iteration

Use the following notation:

c1(xc1,yc1) = {points of c1}
c2(xc2,yc2) = {points of c2}
c3(xc3,yc3) = {points of c3}

Example:

c1(75,-156) = {point1 point2 point3 point4 ....}

Given the **similarity matrix** in the figure, apply agglomerative hierarchical clustering with **single (MIN) linkage**. The similarity metric ranges from 0 (more distant) to 1 (less distant).

	a	b	c	d	e
a	1	0.4	0.4	0.8	0.5
b	0.4	1	0.3	0.3	0.4
c	0.4	0.3	1	0.4	0.9
d	0.8	0.3	0.4	1	0.5
e	0.5	0.4	0.9	0.5	1

Let K be the number of obtained clusters. Write in the answer box:

1. The clusters obtained with K=3
2. The clusters obtained with K=2

Use the following notation:
K=3 {points of the first cluster} {points of the second cluster} {points of the third cluster}
K=2 {points of the first cluster} {points of the second cluster}

Example:
K=3 {point1 point2} {point3 point4} {point5}
K=2 {point1 point2} {point 3 point4 point5}

The Manhattan distance is defined as follows for two points $P_1 \in \mathbb{R}^2 = (x_1, y_1)$ and $P_2 \in \mathbb{R}^2 = (x_2, y_2)$

$dist(P_1, P_2) = |x_1 - x_2| + |y_1 - y_2|$

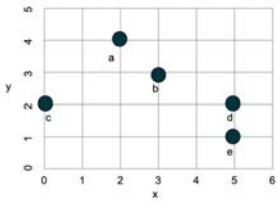
The MIN linkage between two clusters C_1 and C_2 is defined as the minimum among the distances between every pair of points $P_1 \in C_1, P_2 \in C_2$.

Suppose that hierarchical agglomerative clustering is run on the points represented in the figure below, using the MIN linkage (also called single-linkage) and the Manhattan distance.
Write in the answer box (one answer per line):

- the cluster subdivision with K = 3 clusters
- the cluster subdivision with K = 2 clusters
- with K = 2, the value of the linkage distance between the two obtained clusters

Use the following notation to represent a cluster subdivision:
K=k: {points of the first cluster}, {points of the second cluster}, ..., {points of the k-th cluster}
where k is the number of clusters.
For example:

```
K=3: {point1 point2}, {point3 point4 point5}, {point6 point7}
```



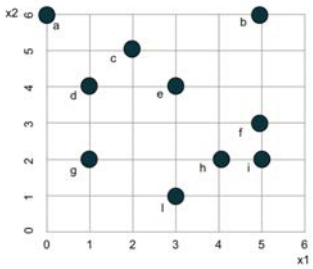
The Manhattan distance is defined as follows:

dist(P1,P2) = |x1-x2|+|y1-y2|

In the DBSCAN algorithm the dataset points can be labeled as:

- core (points with at least *minpoints* neighbors within a distance ϵ),
- border (at least a core point in the neighborhood),
- noise (neither a core point, nor a border point)

Using the Manhattan distance, apply the DBSCAN algorithm to the following points in the bidimensional space.



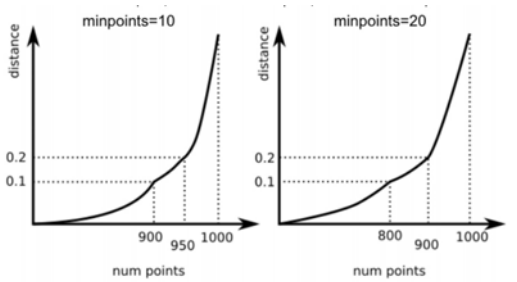
Use the following hyperparameters: ϵ = 2.5, minpoints=2 (at least 2 points as neighbors)

For each point write:

- The assigned label (N=noise, B=border, C=core)
- The assigned cluster id (order of cluster ids is not important, use -1 for noise points)

The chart shown in the figure below is typically used to set up epsilon (eps) and minpoints values in DBSCAN.
It shows on the y-axis, for each point in the dataset, the distance from the k-th nearest point, where k is equal to minpoints.
The x-axis represents all the dataset points, ordered by increasing distance from the k-th nearest point.

Let *nc* be the number of core points, *nb* the number of border points, *nn* the number of noise points.



Which of the following statements is true?

- (a) with *minpoints* = 20, *eps* = 0.2: *nc* = 900, *nn* = 100
- (b) with *minpoints* = 10, *eps* = 0.1: *nc* = 900, *nb* < 50
- (c) with *minpoints* = 20, *eps* = 0.2: *nc* < 900, *nb* $\leq$ 100
- (d) with *minpoints* = 20, *eps* = 0.1: *nc* = 800, *nn* $\leq$ 200
- (e) with *minpoints* = 10, *eps* = 0.2: *nc* + *nb* = 950, *nn* = 50
- (f) with *minpoints* = 10, *eps* = 0.1: *nc* $\leq$ 900, *nn* = 100

For the DBSCAN algorithm, the *epsilon* parameter defines the radius considered when searching for n eighboring points, whereas *min_points* represents the number of neighboring points expected to be fo und within a radius of *epsilon* for a point to be considered *core*.

Which of the following statements on DBSCAN is correct?

(a) The values of min_points and epsilon do not affect the total number of clusters found

(b) Only epsilon affects the total number of clusters found

(c) Only min_points affects the total number of clusters found

(d) For a fixed value of min_points, increasing the value of epsilon generally increases the number of core points found.

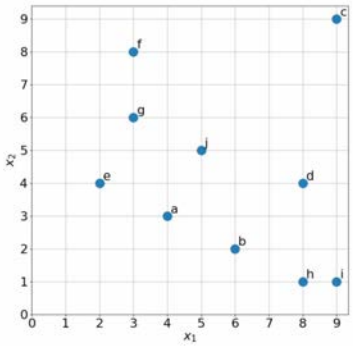
(e) None of the other statements is correct

(f) For a fixed value of min_points, increasing the value of epsilon generally increases the number of border points found. ✓

(g) For a fixed value of epsilon, increasing min_points generally increases the number of core points found

For two n-dimensional points $P_1 = (P_{11}, P_{12}, \ldots P_{1n})$ and $P_2 = (P_{21}, P_{22}, \ldots P_{2n})$ the Manhattan distance is defined as follows:

$$dist(P_1, P_2) = \sum_i |P_{1i} - P_{2i}|$$



Using the Manhattan distance, apply the DBSCAN algorithm to the following points in the bidimensional space.

Use the following hyperparameters: ϵ = 3.5, minpoints=3 (at least 3 points as neighbors)

For each point write:

* The assigned label (N=noise, B=border, C=core)
* The assigned cluster id (order of cluster ids is not important, use -1 for noise points)

The KNN algorithm can assign a class probability based on the votes assigned to each class. If $vote_Y(x)$ represents the unnormalized vote 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_∞ (or Chebyshev) distance between points $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$ in d dimensions is defined as:

$$L_\infty(x, y) = max_i |x_i - y_i|$$

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

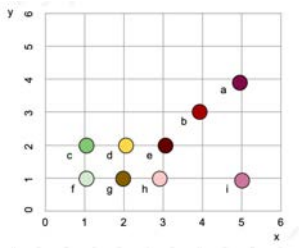
For two n-dimensional points $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the Manhattan distance is defined as follows:

$$dist(a, b) = \sum_i^n |a_i - b_i|$$

The plots below show the (x, y) and (x, z) projections of 9 points in a 3-dimensional space (x, y, z).

Using the Manhattan distance, apply the DBSCAN algorithm to the provided points.

ϵ = 2.5, minpoints=2 (at least 2 points as neighbors)

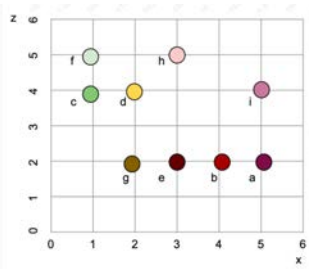


For each point write:

- The assigned label (N=noise, B=border, C=core)
- The assigned cluster id (the order of cluster ids is not important, use -1 for noise points)

Use the following format: "point label cluster"

Example:
a N -1
b C 0
c C 0
d C 1
e B 1



For a vector of predictions $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ and ground truths $y = (y_1, y_2, \ldots, y_n)$, the R 2 score is defined as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where $\bar{y} = \frac{1}{n} \sum_i y_i$.

A training set is comprised of 10,000 15-dimensional points and 1 target variable. The target variable is continuous and has mean = 5.9 and standard deviation 3.4. The training set is used to train a regression algorithm to predict the target variable.

A separate test set comprised of 5,000 points is next used for assessing the quality of the regression model. The target variable in the test set can be assumed to have the same distribution as the target variable in the training set.

The sum of squared errors for the model is of 14,807.7651.

Based on the available information, what is the R2 of the model, as evaluated on the test set?

Write the answer in the box below. If not enough information is provided, write "not enough information available".

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

	a	b	c	d	e
a	0	13	22	17	23
b	13	0	16	7	21
c	22	16	0	24	6
d	17	7	24	0	20
e	23	21	6	20	0

Apply agglomerative 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
```

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

What are the silhouette scores for points b and d?

	a	b	c	d	e	f
a	0	8	8	7	11	4
b	8	0	2	5	5	4
c	8	2	0	3	3	4
d	7	5	3	0	4	3
e	11	5	3	4	0	7
f	4	4	4	3	7	0

Regression

The R^2score is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where:

- $y_i$is the ith target value
- $\hat{y}_i$is the ith predicted value
- $\bar{y}$is the mean overy_i

Which of the following statements on the R^2score are correct? (select all correct answers, multiple answers may be correct)

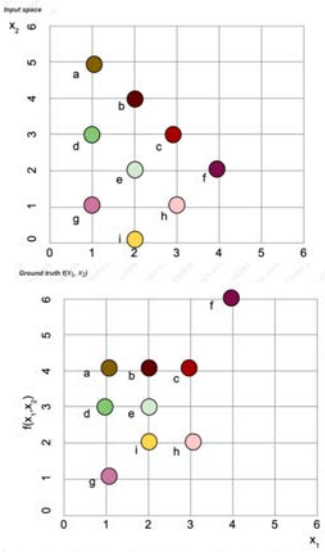
Choose one or more alternatives:

- [ ] (a) It has no upper bound ($+\infty$)
- [ ] (b) It is used as a classification evaluation metric
- [ ] (c) It has no lower bound ($-\infty$)
- [ ] (d) Its upper bound is 1
- [ ] (e) It is used as a clustering evaluation metric
- [ ] (f) Its lower bound is 0
- [ ] (g) It is used as a regression evaluation metric
- [ ] (h) Its lower bound is -1

The Mean Squared Error between two vectors $z, z' \in \mathbb{R}^n$ is defined as:

$$MSE(z, z') = \frac{1}{n} \sum_i (z_i - z'_i)^2$$

Consider the points in the figures below with features x_1, x_2 and ground truth values $f(x_1, x_2)$.



Apply the following regression model to all the data points:

$$f'(x_1, x_2) = 0.5x_1 + 0.5x_2 + 1$$

Write in the answer box the value of the MSE between the prediction $f'(x_1, x_2)$ and the ground truth $f(x_1, x_2)$

The R^2 score is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where:

- y_i is the i-th target value
- $\hat{y}_i$ is the i-th predicted value
- $\bar{y}$ is the mean over y_i

Which of the following statements on the R^2 score are correct?

Scegli una o più alternative:

- [ ] (a) Its upper bound is 1
- [ ] (b) It is used as a regression evaluation metric
- [ ] (c) It is used as a classification evaluation metric
- [ ] (d) It has no upper bound ($+\infty$)
- [ ] (e) It is used as a clustering evaluation metric
- [ ] (f) It has no lower bound ($-\infty$)
- [ ] (g) Its lower bound is -1
- [ ] (h) Its lower bound is 0

1 point - 15% penalty for a wrong answer

The R^2 score is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where:

- y_i is the i-th target value
- $\hat{y}_i$ is the i-th predicted value
- $\bar{y}$ is the mean over y_i

Which of the following statements on the R^2 score are correct? (select all correct answers, multiple answers may be correct)

Scegli una o più alternative:

- [ ] (a) It has no upper bound ($+\infty$)
- [ ] (b) Its upper bound is 1
- [ ] (c) Its lower bound is 0
- [ ] (d) It has no lower bound ($-\infty$)
- [ ] (e) It is used as a classification evaluation metric
- [ ] (f) It is used as a clustering evaluation metric
- [ ] (g) Its lower bound is -1
- [ ] (h) It is used as a regression evaluation metric

When training a Lasso regressor, the following objective function L is minimized: Where MSE represents the mean squared error (ie the sum of the squared distances from the predicted values to the targets), is a regularization coefficient and represents the coefficients of the Lasso model.

$L = MSE + \lambda||\theta||_2$

$\lambda \geq 0\theta$

You are required to train the hyperparameter $\lambda$for a Lasso regressor. What is the most suitable approach for choosing it?

- (a) By defining $f = \frac{L-MSE}{||\theta||_2}$, the optimal λcan be found as the minimum of f (ie $\lambda$for which $\frac{df}{d\lambda} = 0$)
- (b) Any value of $\lambda \geq 0$will result in the same model being learned, so the choice of $\lambda$is not important
- (c) The default value for λprovided by the library used for the implementation (eg scikit-learn) will already provide the best results, as such it should not be changed
- (d) $\lambda$is learned during the training of the model, therefore it does not need to be chosen manually
- (e) λshould be chosen with a grid search or some other hyperparameter tuning technique
- (f) Set λ= 0. In this way, the objective function can reach its minimum value
- (g) None of the other answers is correct
- (h) The value of λonly depends on the dataset used for the training. It therefore can only be computed by having the dataset available

A linear regression model is trained for a regression problem on a dataset containing the information of the employees of a company. For each employee, their age A (in years) and their yearly salary S (in euros) are used as input features.

The output feature is a number in the range [0,100], which represents a rescaled probability that an employee will leave the company, given its age and salary (i.e. p(leave|A, S)).

The following are the coefficients learned for each feature:

- Age: -2.17
- Salary: 1.09 * 10^{-4}
- Bias (intercept): 117

The performance of the model, as measured in terms of R2 score on a separate test set, is of 0.74.
Which of the following statements are correct? Select all correct answers (multiple answers may be correct)

Scegli una o più alternative:

- [ ] (a) There is a positive correlation between age and probability of leaving
- [ ] (b) Without feature normalization, the most influential feature in predicting the likelihood of leaving the company cannot be easily identified
- [ ] (c) There is a negative correlation between salary and probability of leaving
- [ ] (d) There is a positive correlation between salary and probability of leaving
- [ ] (e) There is a negative correlation between age and probability of leaving
- [ ] (f) The age feature is found to be more relevant than the salary in predicting the likelihood of leaving the company
- [ ] (g) The salary feature is found to be more relevant than the age in predicting the likelihood of leaving the company

You are given the following 1-dimensional dataset (feature x) and target feature (y). Point e. represents an anomalous measurement within the dataset.

	x	y
a.	2	-3b
.	3	-4
ch.	2	-3
d.	1	-2
e.	2	123

You fit this dataset using a linear regression model.
Which of the following statements is correct?

- (a) The model learned when including point e. is in any case correct and does not require any additional intervention
- (b) The presence of the anomalous point e. has a significant effect on the model learned, since linear regression is <u>not</u> robust to noise and outliers
- (c) It is incorrect to consider e. an outlier, since its x feature is within a valid range of values
- (d) The presence of the anomalous point e. does not affect the model learned in any way
- (e) It is incorrect to consider e. an outlier, since the only anomaly is the target value
- (f) The presence of the anomalous point e. has a negligible effect on the model learned, since linear regression is robust to noise and outliers

The R^2 score is defined as follows:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where:

- y_i is the i-th target value
- $\hat{y}_i$ is the i-th predicted value
- $\bar{y}$ is the mean over y_i

A linear regression model is trained for a regression problem on a dataset containing the information of the employees of a company. For each employee, their age (in years) and their yearly salary (in euros) are used as input features.

The output feature is a number in the range [0,100], which represents how likely it is for the employee to leave the company.

The following are the coefficients learned for each feature:

- Age: -2.17
- Salary: $1.09 \cdot 10^{-4}$
- Bias (intercept): 117

The performance of the model, as measured in terms of R^2 score on a separate test set, is of 0.37.

Which of the following statements are correct? Select all correct answers (multiple answers may be correct)

Scegli una o più alternative:

- ☐ (a) The coefficients learned would change if the features were normalized with min-max scaling
- ☐ (b) The R^2 score is low because linear regressions perform poorly when the features have different units of measurement
- ☐ (c) The R^2 score is low because linear regressions perform poorly when the features have different orders of magnitude
- ☐ (d) The coefficients learned would **not** change if the features were normalized with min-max scaling
- ☐ (e) The performance of the model would be the same if the features were normalized with min-max scaling
- ☐ (f) The performance of the model would be lower if the features were normalized with min-max scaling
- ☐ (g) The performance of the model would be higher if the features were normalized with min-max scaling

A lasso regression, when applied to a dataset (x_i, y_i) minimizes the following function

$$\sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \lambda \sum_j |\beta_j|$$

You are given a dataset with 4096 real features and a real target variable. You apply the following three regressors:

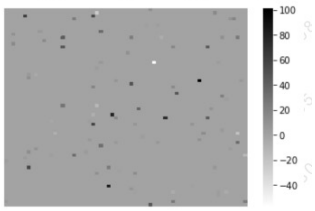
a. Lasso, with $\lambda = 1$
b. Linear regression
c. Lasso, with $\lambda = 100$

For each model, you rearrange the coefficients learned in a 64x64 grid and plot a heatmap.

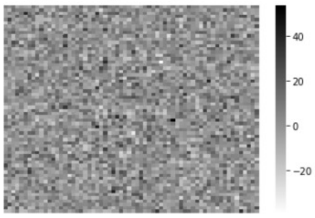
Select the correct mapping between each figure and the model (a, b, c, as listed above)

- ○ (a) 2a 1b 3c
- ○ (b) 1a 2b 3c
- ○ (c) 1a 2c 3b
- ○ (d) 3a 1c 2b
- ○ (e) 3a 1b 2c
- ○ (f) 2a 1c 3b

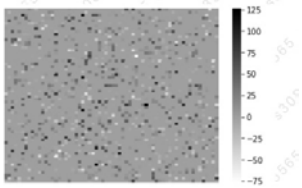
(1)



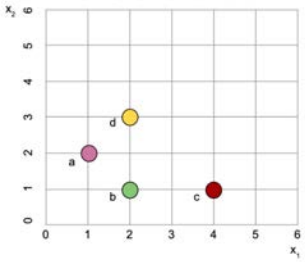
(2)



(3)



Consider the points with features x1, x2 in the figure below.



A polynomial regression model extracts the following additional features: $[x_1, x_2, x_1x_2, x_1^2, x_2^2]$

The regression weights learned are w = [1, -1, 0.5, 0, 0], the intercept is b = -1

Write in the answer box the predicted values for all the data points.

Use the following notation:

y=[f(a), f(b), f(c), f(d)], where f(p) is the prediction for point p

The z-score normalization is applied to a feature x by removing the mean μ and dividing by its standard deviation σ. The resulting feature x' is:

$$x' = \frac{x-\mu}{\sigma}$$

Consider the regression task described by the training set and the test set below, in which **the features have not been normalized**. The mean and the standard deviation (std) for each column are included at the bottom of the tables for your convenience.

The dataset contains **three input features** (freq, rpm, power) and an **output variable** (Y).

The input features have been normalized with z-score (not shown in the tables) before training a regression model to predict the value of Y.

You are required to perform the normalization on the test point t1 (see test set table): write in the answer box the normalized row corresponding to the test point t1, after applying the proper z-score normalization on the features.

Training set

	freq	rpm	power	Y
x1	10	20	0	0
x2	5	0	2	10
x3	5	0	0	10
x4	10	20	2	40
mean	7.5	10	1	15
std	2.5	10	1	15

Test set

	freq	rpm	power	Y
t1	5.5	1	4	3
t2	0.5	6	8	6
mean	3	3.5	6	4.5
std	2.5	2.5	2	1.5

Use the following notation:

t1=[freq, rpm, power, Y]

Example:

t1=[1, 2, 3, 4]

Lasso is a regression model that chooses the best theta configuration based on the following score:

$$score(y, y_{pred}) = MSE(y, y_{pred}) + ||\theta||_1 = \frac{1}{N}\sum_i (y[i] - y_{pred}[i])^2 + \sum_j |\theta[j]|$$

where MSE is the mean squared error and $||\theta||_1$ is the regularization term that sums the absolute values of the coefficients in theta.

Let y = [1, 6, 8] be the ground truth output we require from a regression task on N = 3 points.

Let y_{pred_1} = [1, 5, 6] and y_{pred_2} = [1, 6, 6] be the output vectors obtained with two linear regression models (r1, r2 respectively).

The coefficient vector of r1 is θ_1 = [1/3, -2/3, 1, -1/3], while the one of r2 is θ_2 = [4/3, -6/3, 2/3, -2/3]

Write in the answer box (one answer per line)

- score(y, y_{pred_1})
- score(y, y_{pred_2})
- which is the best configuration between the ones of r1 and r2?

A model's capacity defines its ability of fitting more complex data.

Regularization is defined as a set of techniques to avoid overfitting.

Referring to a polynomial regression problem, which of the following statements is true?

- (a) Regularization is always necessary when the output variable presents complex non linear relationships with the features.
- (b) Regularization decreases model capacity and it is only suitable for output variables that are linearly correlated with features.
- (c) Adding higher power degrees (e.g. x^4, x^5) to the input features increases model capacity and the likelihood of having overfitting.
- (d) All the statements are correct
- (e) Adding higher power degrees (e.g. x^4, x^5) is always useful to obtain better predictions, since it increases model capacity.
- (f) Regularization helps to decrease model capacity and fit more complex data.

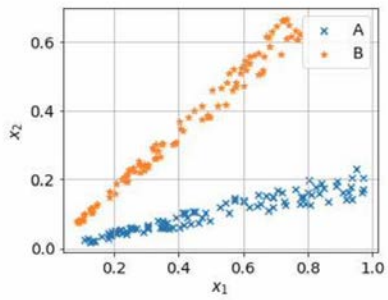
You are given a two-dimensional labelled dataset (features x1, x2), with labels A (blue crosses) and B (orange stars), as represented in the figure below.

The following are transformations that you can apply to your input features.

- ρ, r are the polar coordinates of the points in the original dataset (ρ represents the angles, r represents the radii)
- x_1^2, x_2^2, $x_1 \cdot x_2$ are the second order polynomials of the input features
- x_1', x_2' represent the original input features after min-max scaling (i.e. scaled in the range [0,1])
- x_1^*, x_2^* represent the original input features after z-score normalization (i.e. transformed to have mean 0 and variance 1)

Which of these transformations allows training a decision tree with a single decision node (i.e. a root with two leaves) that perfectly separates the points between the two classes?

Select the smallest subset of features that allows achieving this result. The set may be comprised of multiple features from the list above.

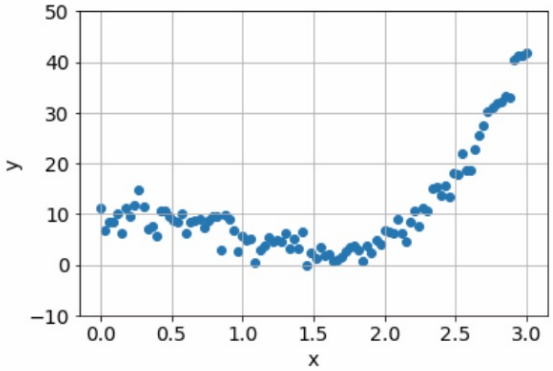


(a) x_2'
(b) x_2^*
(c) x_1'
(d) $x_1 \cdot x_2$
(e) x_1^* ✗
(f) x_2
(g) x_1
(i) x_2^2
(j) x_1^2
(k) ρ
(l) r
(h) None of the other features can be used to achieve the required result

Ridge is a linear regression model in which a regularization penalty is applied to the weights learned. For a model $\hat{y} = w^T x + b$, the parameters are chosen to minimize: the function L, defined as:

$$L = MSE + \lambda \cdot ||w||_2^2$$

You want to model the relationship between an independent variable (x) and a dependent variable (y). The figure below shows the relationship between the two variables.



(a) The model behaves as a linear regression with polynomial features
(b) None of the other statements is correct
(c) The model overfits the training data
(d) The model will make constant predictions, all exactly equal to 0
(e) The model will make approximately constant predictions, all positive
(f) The model behaves as a normal linear regression
(g) The model will make approximately constant predictions, all close to 0
(h) The model training does not converge

You apply a Ridge regressor to predict y given x as input.

However, you choose an excessively large value for λ. Which of the following behaviors occur, when testing the model on points in the range $[0, 3]$?

1 Point, -15% penalty for a wrong answer

A model's capacity defines its ability of fitting more complex data.
Regularization is defined as a set of techniques to avoid overfitting.

Referring to a polynomial regression problem, which of the following statements is true?

- (a) Adding higher power degrees (e.g. x^4, x^5) to the input features increases model capacity and the likelihood of having overfitting.
- (b) Regularization helps to decrease model capacity and fit more complex data.
- (c) Regularization decreases model capacity and it is only suitable for output variables that are linearly correlated with features.
- (d) Regularization is always necessary when the output variable presents complex non linear relationships with the features.
- (e) All the statements are correct
- (f) Adding higher power degrees (e.g. x^4, x^5) is always useful to obtain better predictions, since it increases model capacity.

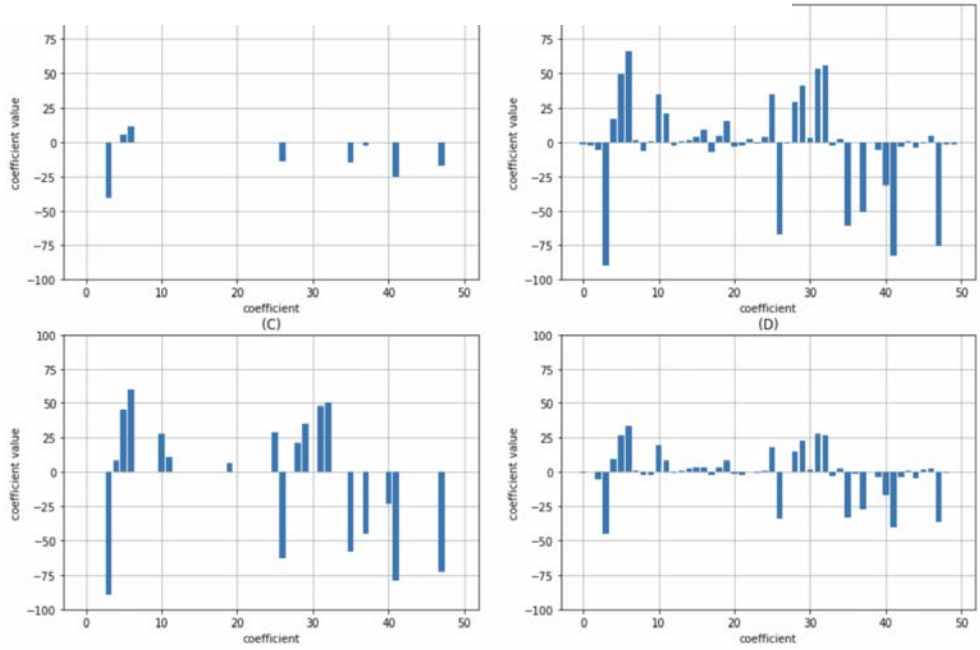
The weights w of a Ridge model are chosen so as to minimize $MSE + \lambda||w||_2^2$
The weights w of a Lasso model are chosen so as to minimize $MSE + \lambda||w||_1$

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, $\lambda = 10$
- Ridge, $\lambda = 100$
- Lasso, $\lambda = 1$
- Lasso, $\lambda = 5$

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?



(a) (A) Ridge, $\lambda = 10$
(D) Ridge, $\lambda = 100$
(B) Lasso, $\lambda = 1$
(C) Lasso, $\lambda = 5$

(b) (A) Ridge, $\lambda = 10$
(B) Ridge, $\lambda = 100$
(C) Lasso, $\lambda = 1$
(D) Lasso, $\lambda = 5$

(c) (B) Ridge, $\lambda = 10$
(D) Ridge, $\lambda = 100$
(C) Lasso, $\lambda = 1$
(A) Lasso, $\lambda = 5$

(d) (C) Ridge, $\lambda = 10$
(B) Ridge, $\lambda = 100$
(A) Lasso, $\lambda = 1$
(D) Lasso, $\lambda = 5$

(e) (B) Ridge, $\lambda = 10$
(C) Ridge, $\lambda = 100$
(A) Lasso, $\lambda = 1$
(D) Lasso, $\lambda = 5$

(f) (A) Ridge, $\lambda = 10$
(C) Ridge, $\lambda = 100$
(B) Lasso, $\lambda = 1$
(D) Lasso, $\lambda = 5$

(g) (B) Ridge, $\lambda = 10$
(C) Ridge, $\lambda = 100$
(D) Lasso, $\lambda = 1$
(A) Lasso, $\lambda = 5$

(h) (B) Ridge, $\lambda = 10$
(A) Ridge, $\lambda = 100$
(C) Lasso, $\lambda = 1$
(D) Lasso, $\lambda = 5$

(i) None of the other answers is correct.

(j) (D) Ridge, $\lambda = 10$
(B) Ridge, $\lambda = 100$
(C) Lasso, $\lambda = 1$
(A) Lasso, $\lambda = 5$

1 point, no penalty for a wrong answer

The Fowlkes-Mallows score* is a quality index designed for evaluating clustering predictions against a ground truth.

It is defined as follows:

$$FM_{score} = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}}$$

Where:

- TP = number of pairs of elements that are in the same set in the ground truth and in the same set in the prediction
- FP = number of pairs of elements that are in different sets in the ground truth and in the same set in the prediction
- FN = number of pairs of elements that are in the same set in the ground truth and in different sets in the prediction

Given the vectors:

```
y_pred = [1 1 2 3 2] # prediction
y_true = [1 3 2 3 2] # ground truth
```

Write in the box below the value of Fowlkes-Mallows score computed for y_pred and y_true.

1/sqrt((1+1)*(1+1))=0.5

The coefficients of a linear regression model are found by minimizing the residual sum of squares (RSS) between the predictions $y' = (y'_1, y'_2, \ldots, y'_n)$ and the ground truth $y = (y_1, y_2, \ldots, y_n)$.

$$RSS = \frac{1}{n} \sum_i (y'_i - y_i)^2$$

x	y
2	-3
3	-4
2	-3
1	-2

You are given the following 1-dimensional dataset (feature x) and target variable (y).

You fit this data with a linear regression model that uses no intercept (y' = m * x).

Compute the coefficient *m* learned by the linear regression model.

Write the answer in the box below. Use the following syntax:

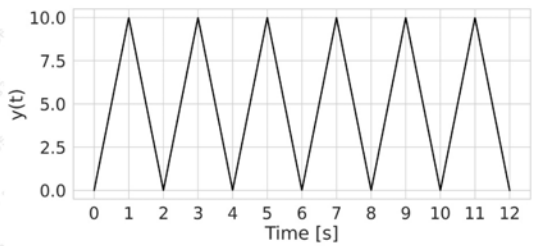
```
m=value
```

For example:

```
m=3
```

Consider the time series y(t) in the figure below defined between 0 and 12 seconds. We decide to encode it in tabular format using a sliding window approach with window size=W seconds and step=S seconds. We discard any window of size smaller than W seconds.

For each window, we collect in a Pandas DataFrame (*df*) the mean (column *mean*), the standard deviation (column *std*), and the area under the curve (column *area*).



Which statement is true?

- (a) None of the other answers is correct
- (b) If W=1 seconds and S=1 seconds, then df.area.mean() = 10
- (c) If W=1 seconds and S=2 seconds, then df.area.mean() = 10
- (d) If W=2 seconds and S=1 seconds, then df.area.mean() = 5
- (e) If W=2 seconds and S=2 seconds, then df.area.mean() = 20
- (f) If W=1 seconds and S=1 seconds, then df.area.mean() = 5

Python

2 points (no penalty for a wrong answer)

You are given the following Python code.
What are the contents of variable X?

```
import numpy as np

a = list(range(0,4,2))
b = np.ones(len(a), dtype=int)
c = (1 + b * b.shape[0]) - a

M = np.array([
    [0, 3, 0, 2, 3],
    [2, 4, 0, 2, 2],
    [2, 1, 0, 3, 4],
    [1, 2, 0, 3, 1],
    [2, 0, 4, 1, 0]])

X = M[a,c]
```

Write the answer in the box below.
Separate each row with a new line, and each column with a space.
For example, for np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]]), write:

```
1 2 3 4
5 6 7 8
9 10 11 12
```

Consider the following Pandas DataFrame (`df`)

	Age	Gender	Occupation
0	23	M	student
1	27	M	consultant
2	34	F	manager
3	44	F	manager
4	54	M	manager
5	29	M	freelancer
6	21	F	student

The following operation is performed

The following indexing techniques are applied to obtain a and b.

```
df.groupby("Occupation") \
  .filter( \
    lambda x: (x["Gender"]=="F").sum()==2 \
  )["Age"] \
  .mean()
```

Which of the following statements is correct?

- (a) The returned value is 44
- (b) The operation raises an exception because `sum()` cannot be applied to `x["Gender"]=="F"`
- (c) The returned value is 33
- (d) The returned value is 30
- (e) The operation raises an exception because `filter()` cannot be applied to `df.groupby("Occupation")`
- (f) The returned value is 39

- recall for class X is the fraction of correct predictions among the samples with actual class X
- precision for class X is the fraction of correct predictions among the samples predicted with class X

Below you can find the precision and recall signatures for the respective functions in sklearn.

> `sklearn.metrics.precision_score`(*y_true, y_pred, *, labels=None, pos_label=1, average='binary', sample_weight=None, zero_division='warn'*) [source]

> `sklearn.metrics.recall_score`(*y_true, y_pred, *, labels=None, pos_label=1, average='binary', sample_weight=None, zero_division='warn'*) [source]

You are required to evaluate a binary classifier in terms of precision on a test set containing N elements.
The labels for the ground truth are contained in a vector y_true $\in \{0,1\}^N$.
The labels predicted by the classifier are contained in a vector y_pred $\in \{0,1\}^N$.
When computing the precision for the positive class (1), however, you mistakenly invert the ground truth and the predictions when calling the precision function:

```
from sklearn.metrics import precision_score
...
score = precision_score(y_pred, y_true)
```

What is computed in the variable *score* by the code above?

- (a) 1 - Recall for the positive class
- (b) 1 - Precision for the positive class
- (c) Precision for the negative class
- (d) Precision for the positive class
- (e) Recall for the positive class
- (f) Recall for the negative class
- (g) None of the other options is correct
- (h) 1 - Precision for the negative class
- (i) 1 - Recall for the negative class

The following snippet of Python code is executed.

```
a = list(range(12))
b = np.reshape(a, (4,3))
c = np.array([1,-1,1,-1])
d = b + c
```

What is the result of this operation?

- (a) An error occurs when computing b
- (b) An error occurs when computing d
- (c) d = [[2, 3, 8, 9], [3, 4, 9, 10], [4, 5, 10, 11]]
- (d) d = [[0, 1, 2], [-3, -4, -5], [6, 7, 8], [-9, -10, -11]]
- (e) None of the other answers is correct
- (f) d = [[1, 2, 3], [-4, -5, -6], [7, 8, 9], [-10, -11, -12]]
- (g) d = [[1, 2, 7, 8], [2, 3, 8, 9], [3, 4, 9, 10]]

2 points (no penalty for a wrong answer)

The following is the pandas documentation for the sub() operator.

pandas.DataFrame.sub

`DataFrame.sub(other, axis='columns', level=None, fill_value=None)` [source]

Get Subtraction of dataframe and other, element-wise (binary operator *sub*).

Equivalent to `dataframe - other`, but with support to substitute a fill_value for missing data in one of the inputs. With reverse version, *rsub*.

Among flexible wrappers (*add, sub, mul, div, mod, pow*) to arithmetic operators: *+, -, *, /, //, %, **.*

Parameters: **other** : ***scalar, sequence, Series, or DataFrame***

Any single or multiple element data structure, or list-like object.

axis : ***{0 or 'index', 1 or 'columns'}***

Whether to compare by the index (0 or 'index') or columns (1 or 'columns'). For Series input, axis to match Series index on.

level : ***int or label***

Broadcast across a level, matching Index values on the passed MultiIndex level.

fill_value : ***float or None, default None***

Fill existing missing (NaN) values, and any new element needed for successful DataFrame alignment, with this value before computation. If data in both corresponding DataFrame locations is missing the result will be missing.

Returns: **DataFrame**

Result of the arithmetic operation.

You are given the following Python snippet.

```
X = [[4, 2, 1],
     [2, 4, 4],
     [4, 4, 2],
     [1, 1, 2],
     [3, 2, 1]]

df = pd.DataFrame(data=X, columns=["a", "b", "c"], index=range(0,10,2))
out = df - df.loc[:4, "a":"b"].min()
```

You are given the following Python snippet.

What does *out* contain? Write your answer in the box below, specifying:

- Type (DataFrame or Series)
- Columns (for DataFrames only)
- Index
- Data (the contents of the Series/DataFrame)

If the execution of the code results in an error, write "an error occurs" instead.

For "Data", you can separate multiple columns (in the case of DataFrames) using a space. You can separate rows using newlines.

In the case of a multi-level index/column, you must represent them in the same way as with the data (separating with spaces and newlines). You need to repeat the outermost index/column whenever needed.

The following are some examples of the representation you should use.

Example 1

DataFrame

	a	b	c
0	3	3	2
1	2	4	4

Expected representation

```
Type
DataFrame

Columns
a b c

Index
0 1

Data
3 3 2
2 4 4
```

Example 2

Multi-index/column DataFrame

		W		X
		a	b	c
y	**0**	2	3	3
	1	4	3	4
z	**0**	3	1	4
	1	3	3	4

Expected representation

```
Type
DataFrame

Columns
W W X
a b c

Index
y y z z
0 1 0 1

Data
2 3 3
4 3 4
3 1 4
3 3 4
```

Example 3

Series

a	11
b	7
c	5
d	−1

Expected representation

```
Type
Series

Index
a b c d

Data
11 7 5 -1
```

The following documentation has been extracted from the official documentation for scikit-learn's DummyClassifier.

class sklearn.dummy.DummyClassifier(*, *strategy='prior', random_state=None, constant=None*) [source]

DummyClassifier makes predictions that ignore the input features.

This classifier serves as a simple baseline to compare against other more complex classifiers.

The specific behavior of the baseline is selected with the `strategy` parameter.

All strategies make predictions that ignore the input feature values passed as the `X` argument to `fit` and `predict`. The predictions, however, typically depend on values observed in the `y` parameter passed to `fit`.

Note that the "stratified" and "uniform" strategies lead to non-deterministic predictions that can be rendered deterministic by setting the `random_state` parameter if needed. The other strategies are naturally deterministic and, once fit, always return a the same constant prediction for any value of `X`.

Read more in the User Guide.

New in version 0.13.

Parameters:

strategy : *{"most_frequent", "prior", "stratified", "uniform", "constant"}, default="prior"*

Strategy to use to generate predictions.

- "most_frequent": the `predict` method always returns the most frequent class label in the observed `y` argument passed to `fit`. The `predict_proba` method returns the matching one-hot encoded vector.
- "prior": the `predict` method always returns the most frequent class label in the observed `y` argument passed to `fit` (like "most_frequent"). `predict_proba` always returns the empirical class distribution of `y` also known as the empirical class prior distribution.
- "stratified": the `predict_proba` method randomly samples one-hot vectors from a multinomial distribution parametrized by the empirical class prior probabilities. The `predict` method returns the class label which got probability one in the one-hot vector of `predict_proba`. Each sampled row of both methods is therefore independent and identically distributed.
- "uniform": generates predictions uniformly at random from the list of unique classes observed in `y`, i.e. each class has equal probability.
- "constant": always predicts a constant label that is provided by the user. This is useful for metrics that evaluate a non-majority class.

Changed in version 0.24: The default value of `strategy` has changed to "prior" in version 0.24.

random_state : *int, RandomState instance or None, default=None*

Controls the randomness to generate the predictions when `strategy='stratified'` or `strategy='uniform'`. Pass an int for reproducible output across multiple function calls. See Glossary.

constant : *int or str or array-like of shape (n_outputs,), default=None*

The explicit constant as predicted by the "constant" strategy. This parameter is useful only for the "constant" strategy.

```
from sklearn.dummy import DummyClassifier
import numpy as np
from sklearn.metrics import accuracy_score

X = np.array(
[[2, 4, 2, 1],
[0, 1, 4, 3],
[2, 1, 2, 3],
[2, 3, 1, 2],
[3, 1, 2, 1],
[1, 3, 0, 3],
[1, 3, 1, 0],
[1, 1, 3, 3],
[4, 1, 3, 3],
[3, 2, 0, 0]])
train_size = int(0.6 * X.shape[0])

X_train = X[:train_size]
X_test = X[train_size: ]
y_train = X_train[:, ::2].sum(axis=1)
y_test = X_test[:, ::2].sum(axis=1)
print(y_test)
print(y_train.shape)

dummy_clf = DummyClassifier()
dummy_clf.fit(X_train, y_train)
acc = accuracy_score(y_test, dummy_clf.predict(X_test))
print(acc)
```

You are given the following snippet of code that creates a dummy classifier (see documentation above) and validates it. What is the output of the snippet of code (based on the print instructions shown in bold)? Write the answer in the box below. If an error occurs, write "an error occurs".

For a feature x, the min-max normalization is defined as $\frac{x-min(x)}{max(x)-min(x)}$

You are given a feature matrix X with shape (n_samples, n_features), represented as a NumPy array. Which of the following code snippets correctly calculates the feature-wise min-max normalization?

- ○ (a) (X - X.min()) / (X.max() - X.min())
- ○ (b) (X - X.min(axis=1)) / (X.max(axis=1) - X.min(axis=1))
- ○ (c) X - X.min(axis=0) / (X.max(axis=0) - X.min(axis=0))
- ○ (d) X - X.min(axis=1) / (X.max(axis=1) - X.min(axis=1))
- ○ (e) (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
- ○ (f) X - X.min() / (X.max() - X.min())

You are given the following pandas DataFrame *df* , which contains information on phone calls made by the customers of a mobile carrier.

	caller_number	called_number	duration	month	year
0	511-807-5185	240-958-8917	13	9	2020
1	511-807-5185	947-769-1946	88	11	2022
2	240-958-8917	947-769-1946	17	1	2019
3	441-471-2495	327-125-4734	67	12	2019
4	700-702-6706	252-112-9463	76	9	2020
...	...	...	...	...	...

You want to address the following query: *extract the list of phone numbers that, during the month of February (month=2) 2022 made at least 2 calls.*

Which of the following code snippets addresses the query?

○ (to)

```
df.groupby("caller_number")[(df["month"]==2)&(df["year"]==2022)].filter(lambda x: len(x) >= 2)
```

○ (b)

```
gb = df.groupby("caller_number")[(df["month"]==2)&(df["year"]==2022)].size()
gb[gb>=2]
```

○ (c)

```
df.groupby("caller_number").filter(lambda x: len(x) >= 2)[(df["month"]==2)&(df["year"]==2022)]
```

○ (d)

```
gb = df.groupby("caller_number").size()
gb[(df["month"]==2)&(df["year"]==2022)][gb>=2]
```

○ (And)

```
gb = df[(df["month"]==2)&(df["year"]==2022)].groupby("caller_number").size() gb[gb>=2
]
```

○ (f)

```
df.filter(lambda x: len(x) >= 2).groupby("caller_number")[(df["month"]==2)&(df["year"]==2022)]
```

Consider the following Python code. Write down what it prints to the standard output (write "an error occurs" if, at any point, an error occurs during the execution of the code).

```
import numpy as np

a = np.random.random((32, 224, 224, 3))
b = np.random.random((32, 224, 224, 3))
b = b.mean(axis=0)
print(b.shape)
c = a * b
c = (c - c.mean()) / c.std()
out = c.sum(axis=-1).reshape(32, -1)
print(out.shape)
```

The following is the pandas' documentation for the unstack() method.

pandas.DataFrame.unstack

DataFrame.unstack(*level=- 1, fill_value=None*) [source]

Pivot a level of the (necessarily hierarchical) index labels.

Returns a DataFrame having a new level of column labels whose inner-most level consists of the pivoted index labels.

If the index is not a MultiIndex, the output will be a Series (the analogue of stack when the columns are not a MultiIndex).

Parameters: **level** : ***int, str, or list of these, default -1 (last level)***

Level(s) of index to unstack, can pass level name.

fill_value : ***int, str or dict***

Replace NaN with this value if the unstack produces missing values.

Returns: **Series or DataFrame**

You are given the following Python snippet.

```
import pandas as pd

X = [
   [ 4, 2, 4 ],
   [ 4, 2, 4 ],
   [ 3, 2, 3 ],
   [ 1, 3, 2 ],
   [ 3, 3, 3 ],
   [ 3, 2, 1 ],
   [ 3, 3, 3 ],
   [ 4, 1, 4 ]
]

df = pd.DataFrame(data=X, columns=["height", "width", "weight"])
out = df.groupby(["height", "width"]).count().unstack()
```

What does *out* contain? Write your answer in the box below, specifying:

- Type (DataFrame or Series)
- Columns (for DataFrames only)
- Index
- Data (the contents of the Series/DataFrame)

If the execution of the code results in an error, write “an error occurs” instead.

For "Data", you can separate multiple columns (in the case of DataFrames) using a space. You can separate rows using newlines.

In the case of a multi-level index/column, you must represent them in the same way as with the data (separating with spaces and newlines). You need to repeat the outermost index/column whenever needed.

The following are some examples of the representation you should use.

Example 1

DataFrame

	a	b	c
0	3	3	2
1	2	4	4

Expected representation

```
Type
DataFrame

Columns
a b c

Index
0 1

Data
3 3 2
2 4 4
```

Example 2

Multi-index/column DataFrame

		W		X
		a	b	c
y	**0**	2	3	3
	1	4	3	4
z	**0**	3	1	4
	1	3	3	4

Expected representation

```
Type
DataFrame

Columns
W W X
a b c

Index
y y z z
0 1 0 1

Data
2 3 3
4 3 4
3 1 4
3 3 4
```

Example 3

Series

a	11
b	7
c	5
d	−1

Expected representation

```
Type
Series

Index
a b c d

Data
11 7 5 -1
```

Consider the following Python code.

```
import numpy as np
import sklearn as sk

X = np.random.random((3, 4))
pipe = sk.pipeline.make_pipeline(
    sk.preprocessing.PolynomialFeatures(degree=3),
    sk.linear_model.Lasso(alpha=0.8)
)
pipe.fit(X, np.arange(3))
```

Which of the following statements is correct regarding the *.fit()* method?

- (a) It concatenates all polynomial features up to degree 3 to the original ones and trains a polynomial regressor on them
- (b) It concatenates all polynomial features up to degree 3 to the original ones and trains a linear regressor on them
- (c) None of the other answers is correct
- (d) It trains a polynomial regressor followed by a linear regressor with weights regularization
- (e) It transforms the original features into polynomial ones (up to degree 3) and trains a linear regressor on them
- (f) It transforms the original features into polynomial ones (up to degree 3) and trains a polynomial regressor on them

You are given the following python snippet of code.

```
m = np.array([
        [2, 2, 1, 1],
        [2, 0, 1, 1],
        [0, 2, 1, 2],
        [2, 0, 0, 1]])

mask1 = [True, True, False, False]
mask2 = [ not m for m in mask1 ]

a = m[mask1][:,mask1]
b = m[mask2][:,mask2]
c = a * b

print(c + np.array([1,2]))
```

What is the output returned? Write the answer in the box below. Use spaces to separate columns, newlines to separate rows. For example, for array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]), write:

```
1 2 3
4 5 6
7 8 9
```

The following are the contents of file.csv

```
c1,c2
4,e
8,d
6,b
5,c
0,a
```

What is the output of the following Python snippet of code?

```
df = pd.read_csv("file.csv", index_col=0).sort_index()
df2 = df[df["c2"]>="b"]
df3 = df2.loc[:8]
df4 = df3.iloc[:2]
print(df4)
```

Write the output in the box below. Use a separate row for each row of the DataFrame. Remember to include the index and the column names. For example,

```
Index_name column_name
val1 val2
val3 val4
val5 val6
```

You are given the following python snippet of code.

```
m = np.array([
       [1, 2, 0, 1],
       [0, 1, 0, 0],
       [0, 0, 2, 1],
       [1, 2, 2, 1]])

mask1 = [True, True, False, False]
mask2 = [ not m for m in mask1 ]

a = m[mask1][:,mask1]
b = m[mask2][:,mask2]
c = a * b

print(c + np.array([1,2]))
```

What is the output returned? Write the answer in the box below. Use spaces to separate columns, newlines to separate rows. For example, for array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]), write:

```
1 2 3
4 5 6
7 8 9
```

The following are the contents of file.csv

```
c1,c2
1,d
9,c
6,a
8,b
0,e
```

What is the output of the following Python snippet of code?

```
df = pd.read_csv("file.csv", index_col=0).sort_index()
df2 = df[df["c2"]>="b"]
df3 = df2.loc[:8]
df4 = df3.iloc[:2]
print(df4)
```

Write the output in the box below. Use a separate row for each row of the DataFrame. Remember to include the index and the column names. For example,

```
Index_name column_name
val1 val2
val3 val4
val5 val6
```

What is the output of the following snippet of Python code?

```
x = np.array([
    [1, 2, 2, 1],
    [2, 1, 1, 2],
    [3, 2, 2, 2],
    [2, 3, 3, 2]
])

mask = x[:, 0]==2

a = x[mask]
b = x[~mask]

print(b ** a)
```

Write the answer in the box below. **Use spaces to separate columns, newlines to separate rows.**

For example, the answer [[1,2,3], [4,5,6], [7,8,9]] should be written as:

```
1 2 3
4 5 6
7 8 9
```

The following are the contents of file.csv

```
1,1,(2,2),(3,4)
2,2,(3,4),(2,2)
```

What is the output of the following python code snippet?

```
df = pd.read_csv("file.csv", header=None)
df["a"] = df[0] + df[1]
df["b"] = df[2] + df[3]
print(df[["a","b"]])
```

○ (a)

	a	b
0	2	(2,2)(3,4)
1	4	(3,4)(2,2)

○ (b)

	a	b
0	2	(22)
1	4	(34)

○ (c)

	a	b
0	11	(2, 2, 3, 4)
1	22	(3, 4, 2, 2)

○ (d)

	a	b
0	2	(2, 2, 3, 4)
1	4	(3, 4, 2, 2)

○ (e)

	a	b
0	11	(2,2)(3,4)
1	22	(3,4)(2,2)

○ (f)

	a	b
0	11	(22)
1	22	(34)

Consider the following Pandas DataFrame (df)

	Age	Gender	Occupation
0	23	M	student
1	27	M	consultant
2	34	F	manager
3	44	F	manager
4	54	M	manager
5	29	M	freelancer
6	21	F	student

- (a) The operation raises an exception because `sum()` cannot be applied to `x["Gender"]=="F"`
- (b) The returned value is 30
- (c) The returned value is 39
- (d) The returned value is 33
- (e) The returned value is 44
- (f) The operation raises an exception because `filter()` cannot be applied to `df.groupby("Occupation")`

The following operation is performed

The following indexing techniques are applied to obtain a and b.

```
df.groupby("Occupation") \
  .filter( \
    lambda x: (x["Gender"]=="F").sum()==2 \
  )["Age"] \
  .mean()
```

Which of the following assertions is correct?

Consider the following Python operations:

```
x = np.array([[2, 4,  6,  8],
              [3, 6,  9, 12],
              [4, 8, 12, 16]
])
y = np.array([1,2])
z = x[0::2, y]-y
```

Which of the following assertions is correct?

- (a) The value of z is [[3, 4],[7, 10]]
- (b) The value of z is [[3, 5],[4, 7]]
- (c) The value of z is [[3, 5],[6, 10]]
- (d) The value of z is [[2, 8],[2, 10]]
- (e) The value of z is [[3, 4],[5, 7]]
- (f) The value of z in [[2, 7],[3, 10]]

Consider the following numpy array:

```
x = np.array([
              [2, 4,  6,  8],
              [3, 6,  9, 12],
              [4, 8, 12, 16]
])
```

The following indexing techniques are applied to obtain a and b.

```
a = x[1:, ::-1]
b = x[[False, True, True], :0:-1]
```

Which of the following assertions is correct?

- (a) The value of a is [[12, 9, 6, 3], [16, 12, 8, 4]]. The variables a and b contain the same values.
- (b) The value of a is [[12, 9, 6, 3], [16, 12, 8, 4]]. The variables a and b contain different values.
- (c) The value of a is [[3, 6, 9, 12], [4, 8, 12, 16]]. The variables a and b contain the same values.
- (d) None of the other answers is correct.
- (e) The value of a is [[3, 6, 9, 12], [4, 8, 12, 16]]. The variables a and b contain different values.
- (f) The script will provide an error because b cannot be computed.

Consider the following Python operations:

```
l1 = [1,2,3]
l2 = [2,4,6]
d = {'mon':l1, 'tue':l2}
res = d.values()
l1.append(4)
d['tue'].append(8)
print(res)
```

What is the output of the above script?

- (a) dict_values([[1, 2, 3, 4], [2, 4, 6]])
- (b) dict_values([[1, 2, 3], [2, 4, 6]])
- (c) dict_values([[1, 2, 3], [2, 4, 6, 8]])
- (d) The script will generate an error.
- (e) dict_values([[1, 2, 3, 4], [2, 4, 6, 8]])
- (f) None of the other answers is correct.

Let x be a three-dimensional NumPy array defined as:

```
import numpy as np
x = np.array([
              [[1,2],
               [0,2],
               [0,4]],
              [[0,2],
               [1,3],
               [4,3]]
])
```

Which of the following statements is true?

- (a) x[:,-1,:] is [[0, 4], [4,3]]
- (b) x[:,-1,:] is [[[1,2]], [[0,2]]]
- (c) x[:,1,:] is [[0,2],[1,3],[4, 3]]
- (d) x[:,1,:] is [[2,2,4], [2,3,3]]
- (e) x[:,1,:] is [[[0,2]], [[1,3]]]
- (f) None of the answers is correct

Broadcasting rules for executing an operation between two NumPy vectors are defined as follows:

- The shape of the array with fewer dimensions is padded with leading ones
- If the shape along a dimension is 1 for one of the arrays and >1 for the other, the array with shape = 1 in that dimension is stretched to match the other array
- If there is a dimension where both arrays have shape >1 then broadcasting cannot be performed

Apply the broadcasting rules to the following numpy vectors.

```
x = np.array([[1,2,3],
              [0,0,0],
              [1,1,1]])
y = np.array([1,2,3])
z = np.array([[1],
              [0],
              [-1]])
```

Which of the following statements is true?

- (a) `x+y = [[2,3,4],[2,2,2],[4,4,4]]` and `y+z = [[2,1,0],[3,2,1],[4,3,2]]`
- (b) `x+y = [[2,3,4],[2,2,2],[4,4,4]]` and `y+z = [2,2,2]`
- (c) `x+y = [[2,4,6],[1,2,3],[1,1,1]]` and `y+z = [[2,1,0],[3,2,1],[4,3,2]]`
- (d) `x+y = [[2,4,6],[1,2,3],[2,3,4]]` and `y+z` cannot be executed
- (e) `x+y = [[2,4,6],[1,2,3],[2,3,4]]` and `y+z = [[2,3,4],[1,2,3],[0,1,2]]`
- (f) `x+y = [[2,3,4],[2,2,2],[4,4,4]]` and `y+z = [[2,3,4],[1,2,3],[0,1,2]]`

numpy.vstack

numpy.vstack(*tup, *, dtype=None, casting='same_kind'*) [source]

Stack arrays in sequence vertically (row wise).

This is equivalent to concatenation along the first axis after 1-D arrays of shape *(N,)* have been reshaped to *(1,N)*. Rebuilds arrays divided by **vsplit**.

This function makes most sense for arrays with up to 3 dimensions. For instance, for pixel-data with a height (first axis), width (second axis), and r/g/b channels (third axis). The functions **concatenate**, **stack** and **block** provide more general stacking and concatenation operations.

`np.row_stack` is an alias for **vstack**. They are the same function.

Parameters: **tup** : *sequence of ndarrays*

The arrays must have the same shape along all but the first axis. 1-D arrays must have the same length.

dtype : *str or dtype*

If provided, the destination array will have this dtype. Cannot be provided together with *out*.

.. **versionadded:: 1.24**

casting : *{'no', 'equiv', 'safe', 'same_kind', 'unsafe'}, optional*

Controls what kind of data casting may occur. Defaults to 'same_kind'.

.. **versionadded:: 1.24**

Returns: **stacked** : *ndarray*

The array formed by stacking the given arrays, will be at least 2-D.

What is the output of the following snippet of code?

```
import numpy as np

a = np.linspace(0, 50, 100)
b = np.ones((75, 2))
c = np.random.random((25,2))
d = np.vstack([b,c]) * a
e = d.reshape(25, -1, 2)
print(e.shape)
```

(a) None of the other answers is correct

(b) (75, 2)

(c) An error occurs

(d) (25, 4, 2)

(e) (25, 2)

(f) (100, 2)

(g) (25, -1, 2)

(h) (100,) ✗

The following is the pandas DataFrame *df.*

	height	weight	width
0	3	2	9
1	2	2	2
2	8	3	4
3	6	9	2
4	9	8	4

Write your answer in the box below, specifying:

- Type (DataFrame or Series)
- Columns (for DataFrames only)
- Index
- Data (the contents of the Series/DataFrame)

If the execution of the code results in an error, write “an error occurs” instead.

What is the output of the following snippet of code?

```
(df - df[df["height"] < 5].mean()).sum()
```

The following are the pandas DataFrames *df1* and *df2,* respectively (note the column names):

	height	weight	width
0	2	1	2
1	5	7	2
2	6	3	5
3	1	8	7
4	8	1	6

	width	height	weight
0	8	5	3
1	1	3	6
2	9	4	6
3	2	3	1
4	9	9	9

What are the contents of *df5,* after executing the following snippet of code?

```
df3 = (df1 + df2)
df4 = df3 - df3.min()
df5 = df4.loc[:3]
```

Write your answer in the box below, specifying:

- Type (DataFrame or Series)
- Columns (for DataFrames only)
- Index
- Data (the contents of the Series/DataFrame)

If the execution of the code results in an error, write “an error occurs” instead.

You are given a dataset X with target classes y. Both X and y are represented as NumPy arrays.
The contents of X and y are unknown, you only know the following:

- X.ndim == 2
- y.ndim == 1
- X.shape[0] == y.shape[0]
- X.shape[0] > 5
- X and y do not contain any missing values
- X contains real numbers in the range [0, 100]
- y contains one of 10 classes encoded as integers from 0 to 9

The following piece of code is used to find the optimal set of hyperparameters for a decision tree model.

```
from sklearn.model_selection import KFold, ParameterGrid
from sklearn.tree import DecisionTreeClassifier

params = {
    "max_depth": [5, 10, None],
    "criterion": ["gini", "entropy", "log_loss"],
    "min_samples_split": [2, 5]
}

for config in ParameterGrid(params):
    for train_ndx, test_ndx in KFold(5).split(X, y):
        clf = DecisionTreeClassifier(max_depth=config["max_depth"], criterion=config["criterion"], min_sa
mples_split=config["min_samples_split"])
        clf.fit(X[train_ndx], y[train_ndx])
        y_pred = clf.predict(X[test_ndx])
        # ... code for handling the evaluation
```

How many decision trees are trained based on the above code?

(a) 1

(b) 90

(c) 5

(d) 18

(e) An error occurs

(f) 180

(g) None of the other answers is correct

(h) 40

(i) Not enough information is provided to address the question

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

What is the output of the following code? (you can assume infinite precision in the representation of floating point numbers)

```
x = np.random.random((128, 16))
mu = x.mean(axis=0)
sigma = x.std(axis=0)

x2 = (x - mu) / sigma

w = np.random.random(16)
b = np.random.random()

y = (x2 * w).sum(axis=1) + b

print(x2.mean(axis=0).mean(), y.shape)
```

(a) 0 (16,)

(b) 0.5 (16,)

(c) 0.5 (128, 16)

(d) An error occurs

(e) 0.5 (128,)

(f) 0 (128,)

(g) 1 (128, 16)

(h) 0 (128, 16)

(i) None of the other answers is correct

(j) 1 (16,)

(k) 1 (128,)