Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Select all true statements.

Options :

6406532577751. ✔ In Decision tree, if a question Q_1 is "better" than question Q_2, then information gains for Q_1 is greater than information gains Q_2 always.

6406532577752. ✔ The training dataset is required while predicting the label of a test-point in the k-NN algorithm.

6406532577753. ✖ A question of the form $f_k \leq \theta$ always partitions the dataset into two non-empty sets.

6406532577754. ✔ The depth of the tree is a hyperparameter and has to be chosen using cross validation.

6406532577755. ✖ Decision trees are prone to overfit if the maximum depth is set too low.

MLP

Section Id :	64065353269
Section Number :	13
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	23
Number of Questions to be attempted :	23
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0

Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653112639
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 205 Question Id : 640653770628 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING PRACTICE (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406532577761. ✔ YES

6406532577762. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653112640
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 206 Question Id : 640653770629 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider following code snippet:

```
from sklearn.utils.multiclass import type_of_target
import numpy as np
print(type_of_target(np.array([['horror','fantasy'],
                               ['adventure','fantasy'],
                               ['adventure','fantasy']])))
print(type_of_target([72, 17.89, 63.00]))
print(type_of_target([0, 1, 1, 0]))
```

What will be the output of the above code snippet in the correct sequence?

Options :

6406532577763. ✖ 'multilabel-indicator'
'multiclass'
'binary'

6406532577764. ✖ 'multiclass'
'multiclass'
'binary'

6406532577765. ✖ 'binary'
'multiclass'
'multilabel-indicator'

6406532577766. ✔ 'multilabel-multioutput'
'continuous'
'binary'

Question Number : 207 Question Id : 640653770630 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code snippet and assume all the dependencies are imported:

```
from sklearn.linear_model import Perceptron
clf = Perceptron(max_iter=100,random_state=1729)
```

He learnt that every time he calls fit() method on 'clf', the parameters learnt from the previous training session (i.e. previous call to 'fit()') are lost. What should he change in code so that this problem is removed?

Options :

6406532577767. ✔ Set warm_start=True

6406532577768. ✖ Combine training data from different training sessions

6406532577769. ✖ Set retain_parameters=True

6406532577770. ✖ This problem can not be solved.

Question Number : 208 Question Id : 640653770631 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider a binary classification dataset with labeled as 98% negative samples and 2% positive samples. A model is trained on this data, which of the following evaluation metrics are suitable for measuring effectiveness of this model:

Options :

6406532577771. ✖ accuracy

6406532577772. ✖ Mean Absolute Error

6406532577773. ✖ smote

6406532577774. ✔ F-1 score

Question Number : 209 Question Id : 640653770632 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code block:

```
from sklearn.datasets import make_regression
X, y = make_regression(n_samples = 1000,
                       n_features = 5,
                       n_informative = 2,
                       random_state=42)

from sklearn.linear_model import SGDRegressor
sgd1 = SGDRegressor(alpha=1e-3,
                    random_state=42,
                    penalty='_________________', )
sgd1.fit(X, y)
print(sgd1.coef_)

sgd2 = SGDRegressor(alpha=1e-3,
                    random_state=42,
                    penalty='___________________')
sgd2.fit(X, y)
print(sgd2.coef_)
```

What are the most suitable values to be filled in the two blank spaces (in that order) in the code to expect the following output?:

[1.68059576e+01, 1.89752021e+01, 7.49212536e-04, -6.53455275e-04, 3.01471918e-04]

[16.82258106, 18.99248887, 0., 0., 0.]

Options :

6406532577775. ✖ '11', '12'

6406532577776. ✖ '11', None

6406532577777. ✔ '12', '11'

6406532577778. ✖ '12', None

Question Number : 210 Question Id : 640653770633 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What might be the possible output of the following code:

```
from sklearn.metrics import mean_absolute_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
mean_absolute_error(y_true, y_pred)
```

Options :

6406532577779. ✖ 0.00

6406532577780. ✔ 0.50

6406532577781. ✖ 0.72

6406532577782. ✖ 1.00

Question Number : 211 Question Id : 640653770634 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

You're building a machine learning pipeline to preprocess data and train a model on a

classification task. You decide to use a pipeline that includes data preprocessing and a support vector machine (SVM) classifier. The following code snippet demonstrates the pipeline creation and usage:

```
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
import numpy as np

# Simulated data (features: X, target: y)
X = np.array([[2, 3], [5, 7], [8, 10]])
y = np.array([0, 1, 0])

# Create a pipeline with StandardScaler and SVM classifier
pipeline = Pipeline([('scaler', StandardScaler()),
                     ('svm', SVC())])

# Fit the pipeline on training data
pipeline.fit(X, y)

# Make predictions using the trained pipeline
predictions = pipeline.predict(X)
```

What is the purpose of using the pipeline in this code snippet?

Options :

6406532577783. ✖ The pipeline combines multiple models for better model performance.

6406532577784. ✖ The pipeline allows for simultaneous training of the scaler and classifier.

6406532577785. ✔ The pipeline simplifies the code by encapsulating preprocessing and modeling steps.

6406532577786. ✖ The pipeline ensures that only linear SVM can be used for this classification task.

Question Number : 212 Question Id : 640653770635 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Given below code to load a huge file name as filename.csv and this file is not loading at once in the system which parameter should be added to pd.read_csv to load this file ?

```
import pandas as pd
from sklearn.linear_model import SGDRegressor
for train_df in pd.read_csv("filename.csv", __________=1024):
    X = train_df.iloc[:, :-1]
    y = train_df.iloc[:, -1]
    model = SGDRegressor()
    model.partial_fit(prep_X,y)
```

Options :

6406532577787. ✖ max_depth

6406532577788. ✖ C

6406532577789. ✔ chunksize

6406532577790. ✖ warm_start

Question Number : 213 Question Id : 640653770636 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will the output for below code

```
from sklearn.feature_extraction.text import CountVectorizer
corpus = [ 'This is the first document.',
           'This document is the second document.']
vectorizer = CountVectorizer()
vectorizer.fit_transform(corpus)
print(vectorizer.get_feature_names_out())
```

Options :

6406532577791. ✖ {'this': 5, 'is': 2, 'the': 4, 'first': 1, 'document': 0, 'second': 3}

6406532577792. ✖ [3,1,2,1,2,2]

6406532577793. ✔ ['document', 'first', 'is', 'second', 'the', 'this']

6406532577794. ✖ [0,1,2,3,4,5]

Question Number : 214 Question Id : 640653770637 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Imagine you're training a Perceptron using sklearn with the following code:

```
from sklearn.linear_model import Perceptron
X = [[0, 0.5], [1, 1.5], [1, 2], [2, 3]]
y = [-1, -1, 1, 1]
clf = Perceptron(eta0 = 1, tol=None, shuffle=True, random_state=42)
clf.fit(X, y)
iterations = clf.n_iter_
```

Given the linearly separable nature of the data, how many iterations would it most likely take for the perceptron to converge? What will be the value of iterations?

Options :

6406532577795. ✖ iterations = 1

6406532577796. ✖ iterations = 10

6406532577797. ✔ iterations value can vary since the data is being shuffled in each epoch.

6406532577798. ✖ iterations = 5

Question Number : 215 Question Id : 640653770638 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code snippet using scikit-learn:

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

pipeline = Pipeline([('scaler', StandardScaler()),
                     ('classifier', SVC())])

param_grid = {'scaler__with_mean': [True, False],
              'classifier__C': [0.1, 1, 10],
              'classifier__kernel': ['linear', 'rbf'],
              'classifier__gamma': [0.1, 1, 10]}

grid_search = GridSearchCV(estimator= pipeline,
                           param_grid= param_grid,
                           cv=5,
                           scoring='accuracy',
                           verbose=2)
grid_search.fit(X_train, y_train)
```

Assuming that `X_train` and `y_train` are given and the features are not sparse, which of the following statements about the given code is correct?

Options :

6406532577799. ✖ The `StandardScaler` will scale both `X_train` and `y_train` before training a classifier.

6406532577800. ✔ All the classifiers will not be trained on the scaled data with zero mean and unit variance.

6406532577801. ✖ The pipeline always uses a radial basis function (`'rbf'`) as the kernel for the `SVC()` classifier.

6406532577802. ✖ A total of 18 different combinations of hyperparameters were trained during the GridSearchCV fitting.

Question Number : 216 Question Id : 640653770639 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code snippet that employs LogisticRegression from sklearn on a feature matrix X and corresponding label vector y:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(class_weight='balanced', C=0.5)
model.fit(X, y)
```

Given the code above, which of the following statements is true?

Options :

6406532577803. ✖ The logistic regression model will give equal importance to both classes in an imbalanced dataset.

6406532577804. ✖ The model does not use any regularization because the parameter C is set.

6406532577805. ✖ The model will perform equally well on both imbalanced and balanced datasets due to the class_weight parameter.

6406532577806. ✔ The value of C indicates that the model will apply a regularization.

Question Number : 217 Question Id : 640653770649 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following is true for a hard margin SVM algorithm ?

Options :

6406532577834. ✖ It does not create hyperplanes as a classification decision boundary

6406532577835. ✖ It is robust to outliers

6406532577836. ✔ It will correctly classify all the datapoints if the data is linearly separable.

6406532577837. ✖ It is mostly used for clustering the data

Sub-Section Number :	3
Sub-Section Id :	640653112641
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 218 Question Id : 640653770640 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following is/are correct regarding RadiusNeighborsClassifier

Options :

6406532577807. ✖ Only n_neighbors in the range of some radius R are used to compute the label of a sample.

6406532577808. ✔ All the neighbours in the range of some radius R are used to compute the label of a sample.

6406532577809. ✔ Feature Scaling helps in improving the score of RadiusNeighborsClassifier model

6406532577810. ✖ LabelEncoder helps in improving the score of RadiusNeighborsClassifier model

Question Number : 219 Question Id : 640653770641 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following is correct?

Options :

6406532577811. ✔ SGDClassifier(loss= "perceptron") is a stochastic version of a perceptron model

6406532577812. ✖ SGDClassifier(loss= "percept") is a stochastic version of a perceptron model

6406532577813. ✔ SGDClassifier(loss= "log_loss") is a stochastic version of a logistic classifier model

6406532577814. ✖ SGDClassifier(loss= "sigmoid") is a stochastic version of a logistic classifier model

Question Number : 220 Question Id : 640653770642 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following option(s) are correct for the precision-recall curve

Options :

6406532577815. ✔ A high area under the curve represents both high recall and high precision.

6406532577816. ✔ The precision-recall curve shows the trade-off between precision and recall for different threshold values.

6406532577817. ✖ The precision-recall curve used to evaluate unsupervised algorithm for imbalanced clustered data.

6406532577818. ✖ None of these

Question Number : 221 Question Id : 640653770643 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following statements are true?

Options :

6406532577819. ✔ KNeighborsClassifier with low values of n_neighbors produces complex decision boundaries.

6406532577820. ✖ KNeighborsClassifier with low values of n_neighbors produces smooth decision boundaries.

6406532577821. ✔ In KNeighborsClassifier the scale of the features(columns) can impact the decision boundaries.

6406532577822. ✖ None of these

Question Number : 222 Question Id : 640653770645 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following code snippet that employs `LogisticRegression` from sklearn on a feature matrix `X` and corresponding label vector `y`:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(C=0.8, multi_class='multinomial', max_iter=1000)
model.fit(X, y)
```

Given the code above, which of the following statements is true?

Options :

6406532577827. ✖ The `LogisticRegression` model is set up for binary classification.

6406532577828. ✖ The model does not use any regularization because the parameter `C` is set.

6406532577829. ✔ The model has been specifically set up to handle a multi-class classification problem using a `softmax` regression approach.

6406532577830. ✔ The model might iterate through the data multiple times, with a maximum limit set at 1000 iterations.

Sub-Section Number : 4

Sub-Section Id : 640653112642

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 223 Question Id : 640653770644 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Fill in the missing parameter value in the following estimator that can be used to classify the data

```
from sklearn.svm import SVC
clf = SVC(kernel = ______)
clf.fit(X, y)
```

Options :

6406532577823. ✔ 'poly'

6406532577824. ✖ 'lasso'

6406532577825. ✖ 'scale'

6406532577826. ✔ 'sigmoid'

Sub-Section Number :	5
Sub-Section Id :	640653112643
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 224 Question Id : 640653770646 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Short Answer Question

Consider the following code snippet:

```
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import FeatureUnion
X = load_iris().data # X.shape = (150,4)

poly_feature = PolynomialFeatures(degree=2, include_bias=True)
union = FeatureUnion([('poly', poly_feature),
                      ('pca', PCA(n_components=2))])

X_transformed = union.fit_transform(X)
print(X_transformed.shape)
```

If the shape of X is $(150, 4)$. How many total columns are there in the X_transformed ?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

17

Question Number : 225 Question Id : 640653770647 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Short Answer Question

Please consider the following data and code for a regression problem with symbols in mind:

- >>>: Represents input code
- # : Represents comment in a code
- . . . : Represents code continuation
- Without any symbols at the beginning of a line then it is output of just above input line of code.

```
>>> import pandas as pd
>>> from sklearn.preprocessing import OneHotEncoder
>>> from sklearn.linear_model import LinearRegression
>>> data_array = [[19, 'Black', 74],
...               [19, 'Blue', 75],
...               [19, 'Red', 85],
...               [24, 'Black', 70],
...               [24, 'Blue', 70],
...               [24, 'Red', 89],
...               [30, 'Black', 78],
...               [30, 'Blue', 76],
...               [30, 'Red', 90]]

>>> data = pd.DataFrame(data_array,columns=["Age",
                                            "Car_color",
                                            "Accidents_per_1000_Driver"])

>>> X = data.drop("Accidents_per_1000_Driver", axis=1)
>>> y = data["Accidents_per_1000_Driver"]

>>> ohe = OneHotEncoder(sparse_output=False)
>>> X[['Black', 'Blue', 'Red']] = ohe.fit_transform(X[["Car_color"]])
>>> X.drop("Car_color", axis=1, inplace=True)

>>> lr = LinearRegression().fit(X, y)

>>> print(lr.coef_)
[0.32, -4.55, -4.88, 9.44]

>>> print(lr.intercept_)
70.75
```

How many Accidents per 1000 Driver predicted by the model for Age 27 and driving a Red car ?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

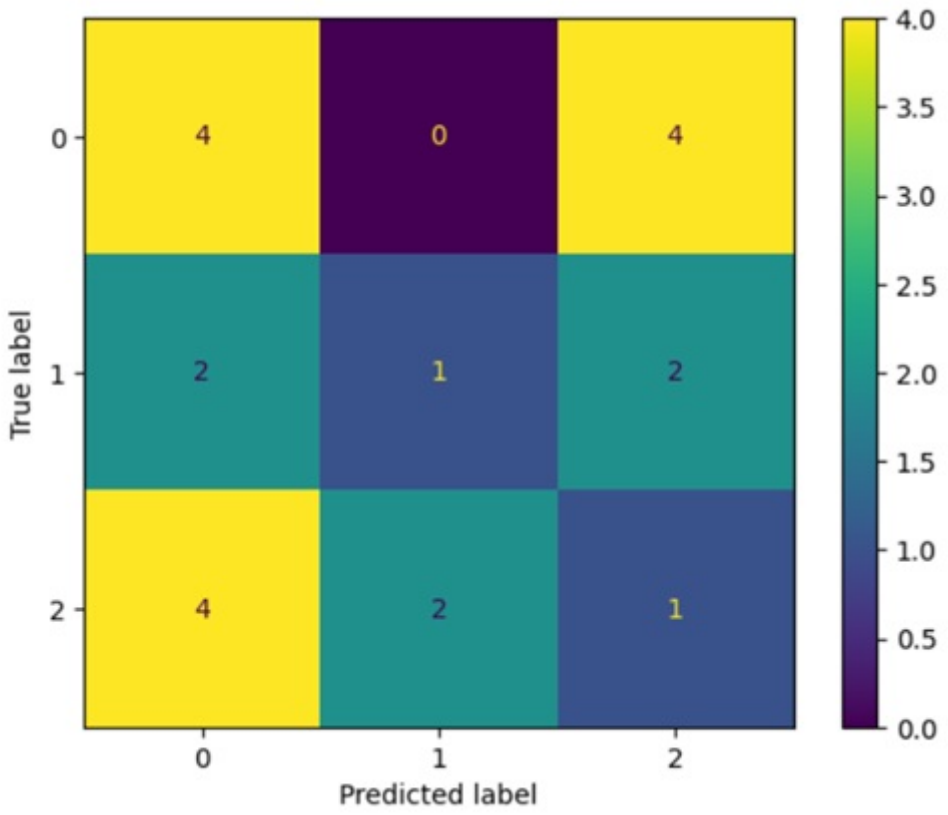
88.3 to 89.3

Question Number : 226 Question Id : 640653770650 Question Type : SA Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Short Answer Question

After training a multi-class classifier, you obtain the following confusion matrix. What will be the weighted average of the recall score for each class?



Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

0.295 to 0.315

Sub-Section Number : 6

Sub-Section Id : 640653112644

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 227 Question Id : 640653770648 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Short Answer Question

What is the output of the following code?

```
from sklearn.neighbors import KNeighborsClassifier
X = [[2,3], [5,6], [8,9], [10, 11], [15,16], [20,21]]
y = [2, 1, 0, 1, 2, 1]

knn = KNeighborsClassifier (n_neighbors=3,
                            metric='euclidean',
                            weights='uniform')

knn.fit (X, y)
print (knn.predict([[8,9]]))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

1

Business Analytics

Section Id : 64065353270

Section Number : 14

Section type : Online

Mandatory or Optional : Mandatory

Number of Questions : 5