

MLP

Section Id :	64065321848
Section Number :	12
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
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 :	64065348704

Question Shuffling Allowed :

No

Question Number : 196 Question Id : 640653345033 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 "MACHINE LEARNING PRACTICES"

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 :

6406531147723. ✓ YES

6406531147724. ✗ NO

Sub-Section Number :

2

Sub-Section Id :

64065348705

Question Shuffling Allowed :

Yes

Question Number : 197 Question Id : 640653345034 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Select Question

Consider the following code snippet:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
X, y = load_iris(return_X_y = True)
```

The size of X and y is (150, 4) and (150,) respectively.

Which of the following option(s) split X and y into training and test data such that test data has 30 samples?

Options :

6406531147725. ✖ `train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=20, random_state=42)`

6406531147726. ✔ `train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=30, random_state=42)`

6406531147727. ✔ `train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.2, random_state=42)`

6406531147728. ✖ `train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.3, random_state=42)`

Question Number : 198 Question Id : 640653345036 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Select Question

We wish to load iris data from sklearn. Which of the following will result in an error?

Options :

6406531147733. ✔ `from sklearn.datasets import load_iris
data = load_iris(load_X_y = True)`

6406531147734. ✖ `from sklearn.datasets import load_iris
data = load_iris(return_X_y = True)`

6406531147735. ✔ `from sklearn.datasets import load_iris
X, y = load_iris(load_X_y = True)`

6406531147736. ✖ `from sklearn.datasets import load_iris
X, y = load_iris(return_X_y = True)`

Question Number : 199 Question Id : 640653345042 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Select Question

The output of the following code block based on the 'Iris' dataset is shown in Figure 1.

```
import seaborn as sns
from matplotlib import pyplot as plt
iris= sns.load_dataset("iris")
sns.set(font_scale=1.5)
plt.figure(figsize=(8,6))
sns.boxplot(data=iris)
sns.swarmplot(data=iris,size=2,color='k')
plt.grid()
plt.title('Box Plot')
plt.ylabel('Range')
plt.show()
```

Based on the Figure 1, mark the correct statements from below options.

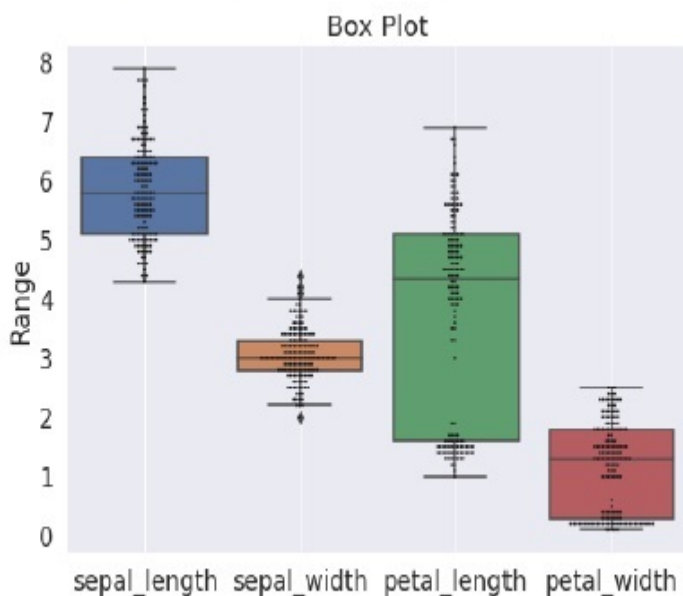


Figure 1

Options :

- 6406531147759. ✓ Median petal_length value is greater than median petal_width.
- 6406531147760. ✗ Median sepal_length value is smaller than the value of median petal_width.
- 6406531147761. ✓ The range of petal_length values is larger than the ranges of all three other

features.

6406531147762. ✖ Only the feature sepal_length has outliers.

Question Number : 200 Question Id : 640653345047 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Select Question

The correct code block or blocks to set adaptive learning rate while using SGDRegressor will be-

Options :

6406531147776. ✖

```
from sklearn.model_selection import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='log2',eta0=1e-2)
```

6406531147777. ✔

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='adaptive',eta0=1e-2)
```

6406531147778. ✔

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='adaptive')
```

6406531147779. ✖

```
from sklearn.model_selection import SGDRegressor
SGD_regressor = LinearRegressor(learning_rate='adaptive',eta0=1e-2)
```

Sub-Section Number : 3

Sub-Section Id : 64065348706

Question Shuffling Allowed : Yes

Question Number : 201 Question Id : 640653345035 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

We wish to download a data set available online. Assume that the url of the csv containing the data is 'url_1'. Which of the following is the correct code to load the data?

Options :

6406531147729. ✖

```
import pandas as pd
data = pd.fetch_csv(url_1)
```

6406531147730. ✖

```
import pandas as pd
data = pd.fetch_data(url_1)
```

6406531147731. ✖

```
import pandas as pd
data = pd.load_data(url_1)
```

6406531147732. ✔

```
import pandas as pd
data = pd.read_csv(url_1)
```

Question Number : 202 Question Id : 640653345038 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 be the shapes of X and y?

```
from sklearn.datasets import make_classification
X, y = make_classification(n_features=20,
                           n_informative=10,
                           n_redundant = 1,
                           n_classes=2,
                           n_clusters_per_class=1,
                           random_state=20,
                           n_samples=30)
```

Options :

6406531147743. ✖ (30,10) and (30,)

6406531147744. ✔ (30,20) and (30,)

6406531147745. ✖ (20,30) and (20,)

6406531147746. ✖ (20,10) and (20,)

Question Number : 203 Question Id : 640653345039 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 is the output of the following block of the code?

```
data = [{'plot number': 51, 'Carpet area':251.8},
        {'Plot number': 69, 'Carpet area':611.9}]
from sklearn.feature_extraction import DictVectorizer
dv = DictVectorizer(sparse=False)
data_transformed = dv.fit_transform(data)
data_transformed
```

Options :

6406531147747. ✖ array([[610.9, 0. , 77.], [51. , 251.8, 0.]])

6406531147748. ✖ array([[611.9, 0. , 69.], [251.8, 51. , 0.]])

6406531147749. ✖ array([[251.8, 0. , 69.], [611.9, 51. , 0.]])

6406531147750. ✔ array([[251.8, 0. , 51.], [611.9, 69. , 0.]])

Question Number : 204 Question Id : 640653345041 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 is the output of the following block of the code?

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
X_california, y_california = fetch_california_housing(return_X_y=True)
# select a subset of data
X, y = X_california[:2000, :], y_california[:2000]
estimator = LinearRegression()
selector = RFE(estimator, n_features_to_select = 3, step=1)
selector = selector.fit(X, y)
print(selector.support_)
```

Options :

6406531147755. ✖ [True False False True False True True True]

6406531147756. ✔ [True False False False False False True True]

6406531147757. ✖ [True False False False False True True True]

6406531147758. ✖ [True False False False False False True False]

Question Number : 205 Question Id : 640653345043 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

Output of the following code is-

```
movie_genres = [{'action', 'comedy' },
                {'comedy'},
                {'action', 'thriller'},
                {'science-fiction', 'action', 'thriller'}]

from sklearn.preprocessing import MultiLabelBinarizer
mlb = MultiLabelBinarizer()
mlb.fit_transform(movie_genres)
```

Options :

6406531147763. ✔ array([[1, 1, 0, 0], [0, 1, 0, 0], [1, 0, 0, 1], [1, 0, 1, 1]])

6406531147764. ✖ array([[1, 0, 0, 0], [0, 1, 0, 0], [1, 0, 0, 1], [1, 0, 1, 1]])

6406531147765. ✖ array([[1, 0, 0, 0], [0, 1, 0, 0], [1, 0, 0, 0], [1, 0, 1, 1]])

6406531147766. ✖ array([[1, 0, 0], [0, 1, 0], [1, 0, 0], [1, 0, 1]])

Question Number : 206 Question Id : 640653345044 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

Output of the following block of code is -

```
import numpy as np
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest, mutual_info_regression
# download data
X_california, y_california = fetch_california_housing(return_X_y=True)
# select a subset of data
X, y = X_california[:2000, :], y_california[:2000]
skb = SelectKBest(mutual_info_regression, k = 3)
X_new = skb.fit_transform(X,y)
print(X_new.shape)
```

Options :

- 6406531147767. ✖ (2000,8)
- 6406531147768. ✔ (2000,3)
- 6406531147769. ✖ (3, 2000)
- 6406531147770. ✖ (2000, 5)

Question Number : 207 Question Id : 640653345045 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 is the 'explained variance score' for the following code?

```
from sklearn.metrics import explained_variance_score
y_true = [1, 2, 3]
y_pred = [2, 5, 8]
explained_variance_score(y_true, y_pred)
```

Options :

6406531147771. ✓ -3

6406531147772. ✖ 3

6406531147773. ✖ -2

6406531147774. ✖ 2

Question Number : 208 Question Id : 640653345048 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

Calculate the coefficient of determination (R^2) from the following block of code?

```
from sklearn.metrics import r2_score
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
print(r2_score(y_true, y_pred))
```

Options :

6406531147780. ✖ 0.789

6406531147781. ✓ 0.948

6406531147782. ✖ 0.922

6406531147783. ✖ 0.824

Question Number : 209 Question Id : 640653345050 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

Options :

6406531147800. ✖ [0.85,0.1,0.05]

6406531147801. ✖ [3,2,1]

6406531147802. ✔ [0.85,0]

6406531147803. ✖ There are some mistakes in the 3rd /4th line of code, hence it will produce an error.

Question Number : 211 Question Id : 640653345057 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:

```
from sklearn import linear_model
clf = linear_model.Ridge(alpha=0.01)
X= [[1,0], [2, 1], [3, 2]]
y= [1, 2, 3]
clf.fit(X, y)
linear_model.Ridge(alpha=0.01, max_iter=1000,
                    tol=0.0001, fit_intercept=True)
clf.score(X,y)
```

Which of the following is likely to be the correct output of the code above?

Options :

6406531147812. ✖ 5

6406531147813. ✖ 99

6406531147814. ✔ 0.999

6406531147815. ✖ No evaluation metrics is mentioned, hence it will produce an error.

Sub-Section Number :

4

Sub-Section Id :

64065348707

Question Shuffling Allowed :

Yes

Question Number : 212 Question Id : 640653345037 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the following code snippet.

```
from sklearn.datasets import load_iris
one = load_iris()
two = load_iris(return_X_y = True)
three = load_iris(as_frame = True)
four = load_iris(as_frame = True, return_X_y = True)
```

What will be the data types of one, two, three and four?

Options :

6406531147737. ✖ Bunch, Tuple, DataFrame, DataFrame

6406531147738. ✖ Bunch, Tuple, DataFrame, Tuple

6406531147739. ✖ Bunch, Narray, Bunch, Narray

6406531147740. ✖ Narray, Tuple, Narray, Tuple

6406531147741. ✔ Bunch, Tuple, Bunch, Tuple

6406531147742. ✖ It will result in an error.

Sub-Section Number : 5

Sub-Section Id : 64065348708

Question Shuffling Allowed : Yes

Question Number : 213 Question Id : 640653345040 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

What is the output of the following block of the code?

```
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
estimators = [('simpleImputer', SimpleImputer()),
              ('standardScaler', StandardScaler())]
pipe = Pipeline(steps=estimators)
print(len(pipe.steps))
```

Options :

6406531147751. ✓ 2

6406531147752. ✖ 3

6406531147753. ✖ 4

6406531147754. ✖ 1

Sub-Section Number :

6

Sub-Section Id :

64065348709

Question Shuffling Allowed :

Yes

Question Number : 214 Question Id : 640653345046 Question Type : SA Calculator : None

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

Correct Marks : 1

Question Label : Short Answer Question

What will be the output of following block of code?

```
from sklearn.model_selection import ShuffleSplit
rs = ShuffleSplit(n_splits=3, random_state=0)
rs.get_n_splits(X)
```

NOTE: Enter your answer to the nearest integer.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

Sub-Section Number : 7
 Sub-Section Id : 64065348710
 Question Shuffling Allowed : Yes

Question Number : 215 Question Id : 640653345049 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Which of the following code snippets represent correct methods to obtain cross validated performance measure using LeaveOneOut?

Options :

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import LeaveOneOut
from sklearn.linear_model import linear_regression
```

```
lin_reg = linear_regression()
loocv = LeaveOneOut()
score = cross_val_score(lin_reg, X, y, cv=loocv)
```

6406531147784. ✓

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(loss='squared_error',
                                early_stopping=True,
                                max_iter=500,
                                tol=1e-3,
                                validation_fraction=0.2,
                                n_iter_no_change=5)
```

6406531147785. ✗

```
from sklearn.linear_model import linear_regression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import LeaveOneOut
lin_reg = linear_regression()
shuffle_split = ShuffleSplit(n_splits=5, test_size=0.2,
                             random_state=42)
```

6406531147786. ✗ score = cross_val_score(lin_reg, X, y, cv=loocv)

```
from sklearn.model_selection import cross_validate
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import LeaveOneOut
cv = ShuffleSplit(n_splits=40, test_size=0.3, random_state=0)
cv_results = cross_validate(regressor, data,
                             target, cv=cv,
                             scoring="loocv_neg_mean_absolute_error")
```

6406531147787. ✖

Question Number : 216 Question Id : 640653345051 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What is the recall value for the class "2" for the following block of code?

```
from sklearn.metrics import confusion_matrix
y_true = [2, 0, 2, 2]
y_pred = [0, 0, 2, 2]
confusion_matrix(y_true, y_pred)
from sklearn.metrics import classification_report
print(classification_report(y_true, y_pred))
```

Options :

6406531147792. ✖ 1.00

6406531147793. ✔ 0.67

6406531147794. ✖ 0.50

6406531147795. ✖ 0.81

Question Number : 217 Question Id : 640653345052 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following code snippet:

```
data=np.array([ 1, 2, 3, 4])
from sklearn.preprocessing import PolynomialFeatures
poly= PolynomialFeatures(degree=3, interaction_only=True)
data = data.reshape(2,2)
print(poly.fit_transform(data))
```

Which of the following is the correct output?

Options :

6406531147796. ✖ array([[1., 1., 2., 2.], [1., 5., 10., 50.], [1., 3., 4., 12.]])

6406531147797. ✖ array([[1., 1., 2., 3., 2., 3., 6.], [1., 4., 5., 10., 20., 40., 50.]])

6406531147798. ✖ array([[1., 4., 5., 10., 20., 40., 50.], [1., 1., 2., 3., 2., 3., 6.]])

6406531147799. ✔ array([[1., 1., 2., 2.], [1., 3., 4., 12.]])

Sub-Section Number : 8

Sub-Section Id : 64065348711

Question Shuffling Allowed : No

Question Id : 640653345054 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Question Numbers : (218 to 219)

Question Label : Comprehension

Consider the following code snippet, and answer the given subquestions.


```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import RidgeCV
from sklearn.datasets import fetch_california_housing
california_housing = fetch_california_housing(as_frame=True)
com_train_features=california_housing.data
com_train_labels=california_housing.target

lf= np.logspace(-3, 0, num=4)

ridge_reg_pipeline = Pipeline([("poly", PolynomialFeatures(degree=2)),
                                ("feature_scaling", StandardScaler())
                                ])

ridge= RidgeCV(alphas=lf,
               scoring="neg_mean_squared_error",
               fit_intercept = False)
results = ridge.fit(com_train_features, com_train_labels)
```

Sub questions

Question Number : 218 Question Id : 640653345055 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 be the output of the following code?

```
print(results.intercept_)
```

Options :

6406531147804. ✖ 0.0001

6406531147805. ✖ 0.001

6406531147806. ✖ 0.1

6406531147807. ✔ 0.0

Question Number : 219 Question Id : 640653345056 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What will be the output of the following code?

```
print(results.best_score_)
```

Options :

6406531147808. ✖ 0.32

6406531147809. ✖ 0.528

6406531147810. ✖ 0.681

6406531147811. ✔ -0.609