RNSIT ML2

✖

Points:
0/1

Consider the following Python code:

import pandas as pd

data = {'A': 10, 'B': 20, 'C': 30}
series = pd.Series(data)

print(series)

What will be the output of the above code?

- ○ An error because 'data' is not a valid parameter for creating a pandas Series.
- ○ A pandas Series with index labels 'A', 'B', and 'C' and corresponding values 10, 20, and 30. ✔
- ○ A pandas DataFrame with one row and three columns labeled 'A', 'B', and 'C'.
- ○ An empty pandas Series object.

✖

Points:
0/3

Match the correct answers for the following.

What is the purpose of the tail() method in pandas? - Select - ▾ ✖

Correct answer: It displays the last few rows of a DataFrame.

What does the head() method in pandas do? - Select - ▾ ✖

Correct answer: Displays the first few rows of a DataFrame.

What information does the info() method in pandas provide? - Select - ▾ ✖

Correct answer: It displays a summary of the DataFrame's structure.

✖

Points:
0/1

Consider the following NumPy array:

import numpy as np

arr = np.array([10, 20, 30, 40, 50])

What is the result of the following code?

index = np.searchsorted(arr, 35, side='right')
print(index)

- Select - ▾ ✖

Correct answer: 4

✖

Points:
0/1

What is the dimensionality of a NumPy array with shape `(3, 4, 5)`?

- ○ 12
- ○ 3 ✔
- ○ 4
- ○ 5

✗

Points:
0/1

Consider the following Python code:

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [25, 30, 35, 40]}
df = pd.DataFrame(data)

result = df.loc[[0, 2]]

print(result)
```

What will be the output of the above code?

○ Name Alice

Age 25

Name Charlie

Age 35

Name: 0, dtype: object

○ Name Alice

Age 25

Name: 0, dtype: object

Name Charlie

Age 35

Name: 2, dtype: object

✓

○ Name Alice

Age 25

Name Charlie

Age 35

dtype: object

○ Error: DataFrame has no attribute 'loc'

✗

Points:
0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])
```

What is the result of the following slicing operation?

```
arr_slice = arr[1:, 1:3]
```

○ array([[2, 3], [6, 7], [10, 11]])

○ array([[5, 6, 7], [9, 10, 11]])

○ array([[6, 7], [10, 11]]) ✓

○ array([[1, 2, 3], [5, 6, 7], [9, 10, 11]])

✗

Points:
0/5

Match the follwing definitions with related library

Library for scientific computing and technical computing: - Select - ✗

Correct answer: SciPy

Library for numerical computing and array manipulation: - Select - ✗

Correct answer: NumPy

Library for data manipulation and analysis: - Select - ✗

Correct answer: Pandas

Library for deep learning and neural networks: - Select - ✗

Correct answer: PyTorch

Library for creating plots and visualizations: - Select - ✗

Correct answer: Matplotlib

✗

Points:
0/1

Which of the following statements about the Pandas library is true?

○ Pandas is a machine learning library for building and training models.

○ Pandas is a visualization library for creating plots and charts.

○ Pandas is primarily used for numerical computing.

○ Pandas provides high-level data structures and functions designed to make working with structured or tabular data easier. ✓

✗

Points:
0/3

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3],
               [4, 5, 6]])
```

What is the output of the following code?

```
for row in arr:
    for value in row:
        print(value, end=' ')
    print()
```

○ 1 2 3 4 5 6

○ 1 4 2 5 3 6

○ 14

25

36

○ 1 2 3

4 5 6

✓

✖

Points: 0/1

Consider the following Python code:

```
import numpy as np
from scipy import constants

print(constants.pi)
```

What will be the output of the above code?

- Select - ✖

Correct answer: 3.141592653589793

✖

Points: 0/1

Which of the following data types in NumPy represents a 64-bit floating-point number?

- Select - ✖

Correct answer: np.float64

✖

Points: 0/1

In Python pandas, what is the purpose of labels?

○ Labels are used to determine the shape of a pandas DataFrame.

○ Labels are used to specify the file format when reading or writing data.

○ Labels are used to identify the data type of each column in a DataFrame.

○ Labels are used to assign names to individual elements in a pandas Series or DataFrame. ✔

✖

Points: 0/1

Consider the following Python code:

```
import tensorflow as tf
import matplotlib.pyplot as plt

# Create a white image
white_image = tf.ones((400, 600, 3), dtype=tf.uint8) * 255
white_image_np = white_image.numpy()

plt.axis('off')
plt.imshow(white_image_np)
plt.show()
```

What modification is needed in the code to create a white image?

- Select - ✖

Correct answer: Replace tf.ones() with tf.zeros().

✖

Points: 0/1

What is the difference between a copy and a view of a NumPy array?

○ A copy always has a different shape than the original array, while a view has the same shape.

○ A copy is created using the copy() method, while a view is created using the view() method.

○ A copy shares the same data memory as the original array, while a view has its own data memory.

○ Modifying a copy does not affect the original array, while modifying a view affects the original array. ✔

✖

Points: 0/1

Consider the following NumPy arrays:

```
import numpy as np

arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

```
result = np.dstack((arr1, arr2))
print(result)
```

○ [[1 2 5 6]

[3 4 7 8]]

○ [[1 2]

[3 4]

[5 6]

[7 8]]

○ [[[1 5]

[2 6]]

[[3 7]

[4 8]]]

○ Error: cannot stack arrays of different shapes

✖

Points: 0/1

What is the default number of rows displayed by the `head()` and `tail()` methods in pandas?

- Select - ✖

Correct answer: 5

✖

Points: 0/1

Consider the following Python code:

```
from scipy import constants

print(constants.Avogadro)
```

What will be the output of the above code?

- Select - ✖

Correct answer: 6.02214076e+23

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([10, 20, 30, 40, 50])
```

What is the result of the following code?

```
index = np.searchsorted(arr, [15, 25, 35])
print(index)
```

- Select - ✖

Correct answer: [1 2 3]

✖

Points: 0/1

What is the shape of the following NumPy array?

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
```

○ (5,) ✔

○ (1,)

○ (1, 5)

○ (5, 1)

✖

Points: 0/1

Which method is used to display the first few rows of a DataFrame in pandas?

○ show()

○ head() ✔

○ display()

○ preview()

✖

Points: 0/1

What is the dimensionality of a NumPy scalar?

○ 0 ✔

○ 3

○ 2

○ 1

✖

Points: 0/1

Consider the following Python code:

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [25, 30, 35, 40]}
df = pd.DataFrame(data)
result = df.loc[:, 'Name']
print(result)
```

What will be the output of the above code?

○ Alice

Bob

Charlie

David

○ Error: DataFrame has no attribute 'loc'

○ 0

1

2

3

○ 0 Alice

1 Bob

2 Charlie

3 David

Name: Name, dtype: object

✔

✖

Points:
0/2

Consider the following Python code:

```
import pandas as pd

data = {'A': [1, 2, 3, 4, 5],
        'B': ['a', 'b', 'c', 'd', 'e']}
df = pd.DataFrame(data)

result = df.head(3)

print(result)
```

What will be the output of the above code?

○
```
   A  B
0  1  a
1  2  b
2  3  c
3  4  d
4  5  e
```

○
```
   A  B
0  1  a
1  2  b
2  3  c
```
✓

○
```
   A  B
3  4  d
4  5  e
```

○ Error: DataFrame has no attribute 'head'

✖

Points:
0/1

What happens if you try to convert a NumPy array from one data type to another incompatible data type using the `astype()` method?

○ The array is converted successfully without any issues.

○ A ValueError is raised indicating that the conversion is not possible. ✓

○ The array is converted, but the resulting data type may not be consistent.

○ A warning is raised, but the array is still converted with possible data loss.

✖

Points:
0/1

Consider the following Python code:

```
from scipy import constants

print(constants.c)
```

What will be the output of the above code?

- Select - ▾ ✖

Correct answer: 299792458.0

✖

Points:
0/1

Which NumPy attribute provides information about the dimensions of an array?

○ ndim ✓

○ size

○ shape

○ dtype

✖

Points:
0/1

Consider the following Python code:

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [25, 30, 35, 40]}
df = pd.DataFrame(data)

result = df.loc[1]

print(result)
```

What will be the output of the above code?

○ Name Bob

Age 30

Name: 1, dtype: object

✓

○ Bob

○ 30

○ Alice

✖

Points:
0/1

Which of the following statements about labels in pandas is true?

○ Labels are case-sensitive in pandas.

○ Labels are only used for visualization purposes and have no impact on data manipulation.

○ Labels must always be unique within a pandas DataFrame or Series. ✓

○ Labels are optional and not required for indexing in pandas.

✖

Points:
0/1

Consider the following NumPy arrays:

```
import numpy as np

arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

```
result = np.hstack((arr1, arr2))
print(result)
```

○ [[[1 2]

[3 4]]

[[5 6]

[7 8]]]

○ [[1 2]

[3 4]

[5 6]

[7 8]]

○ [[1 2 5 6]

[3 4 7 8]]

✓

○ Error: cannot stack arrays of different shapes

✖

Points: 0/1

Which of the following methods is used to create a NumPy ndarray object from a list?

○ np.array() ✓

○ np.arraylist()

○ np.ndarray()

○ np.list()

✖

Points: 0/1

What is a DataFrame in Pandas?

○ A data structure for storing multidimensional arrays of homogeneous data.

○ A one-dimensional labeled array capable of holding data of any type.

○ A specialized data structure for handling time-series data.

○ A two-dimensional labeled data structure with columns of potentially different types. ✓

✖

Points: 0/1

What is the result of the following code?

```
import numpy as np

arr = np.array([1, 2, 3], dtype=np.float64)
print(arr.dtype)
```

- Select - ✖

Correct answer: np.float64

✖

Points: 0/2

Consider the following Python code:

```
import pandas as pd

data = {'A': [1, 2, 3, 4, 5],
        'B': ['a', 'b', 'c', 'd', 'e']}
df = pd.DataFrame(data)

result = df.head()

print(result)
```

What will be the output of the above code?

○

```
   A  B
0  1  a
1  2  b
2  3  c
3  4  d
4  5  e
```

✓

○ Error: DataFrame has no attribute 'head'

○

```
   A  B
0  1  a
1  2  b
2  3  c
```

○

```
   A  B
1  2  b
2  3  c
3  4  d
4  5  e
```

✖

Points: 0/1

In which programming language is NumPy primarily written?

- ○ Python
- ○ Java
- ○ C++ ✓
- ○ Fortran

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3],
             [4, 5, 6]])
```

What is the result of the following code?

```
arr_copy = arr.copy()
arr_copy[0, 0] = 10
print(arr[0, 0])
```

- ○ 3
- ○ 1 ✓
- ○ 2
- ○ 10

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([10, 20, 30, 40, 50])
```

What is the result of the following code?

```
index = np.searchsorted(arr, 25)
print(index)
```

- Select - ▾ ✖

Correct answer: 2

✖

Points: 0/1

What does the term "dimension" refer to in the context of NumPy arrays?

- ○ The number of elements in the array.
- ○ The data type of the elements in the array.
- ○ The shape of the array, represented as a tuple of integers. ✓
- ○ The total memory occupied by the array.

✖

Points: 0/1

Consider the following NumPy arrays:

```
import numpy as np

arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

```
result = np.stack((arr1, arr2))
print(result)
```

○ [[[1 2]

[3 4]]

[[5 6]

[7 8]]]

✓

○ Error: cannot stack arrays of different shapes

○ [[1 2 5 6]

[3 4 7 8]]

○ [[1 2]

[3 4]

[5 6]

[7 8]]

✖

Points: 0/1

Consider the following Python code:

```
from scipy import constants

print(constants.G)
```

What will be the output of the above code?

- Select - ✖

Correct answer: 6.67430e-11

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([1.5, 2.7, 3.8])
```

What is the data type of the array after converting it to integers using the `astype()` method?

- Select - ✖

Correct answer: np.int64

✖

Points: 0/1

Why is NumPy faster than lists for numerical computations in Python?

○ NumPy arrays support dynamic resizing, resulting in faster data manipulation compared to lists.

○ NumPy arrays use less memory than lists, resulting in faster computation.

○ NumPy provides built-in parallel processing capabilities, enabling faster execution of numerical operations.

○ NumPy arrays store elements of the same data type in contiguous memory locations, allowing for vectorized operations and efficient memory access. ✓

✖

Points: 0/1

What does a one-dimensional NumPy array represent?

○ A row in a two-dimensional array.
○ A matrix in a three-dimensional array.
○ A column in a two-dimensional array.
○ A single sequence of elements. ✓

✖

Points: 0/1

Which of the following statements is true regarding the dimensions of NumPy arrays?

○ The dimensions of a NumPy array must be explicitly specified during initialization.
○ NumPy arrays can have an unlimited number of dimensions.
○ A scalar in NumPy is considered to have zero dimensions. ✓
○ The dimensions of a NumPy array are limited to a maximum of five.

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])
```

What is the result of the following slicing operation?

```
arr_slice = arr[1:, :2]
```

○ array([[1, 2, 3], [5, 6, 7], [9, 10, 11]])
○ array([[2, 3, 4], [6, 7, 8], [10, 11, 12]])
○ array([[1, 2], [5, 6], [9, 10]]) ✓
○ array([[5, 6], [9, 10]])

✖

Points: 0/1

Consider the following NumPy arrays:

```
import numpy as np

arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
```

What is the result of the following code?

```
result = np.concatenate((arr1, arr2))
print(result)
```

○ Error: cannot concatenate arrays of different dimensions
○ [1 2 3 4 5 6] ✓
○ [[1 2 3]

[4 5 6]]

○ [[1 2 3 4 5 6]]

✖

Points: 0/1

What is the data type of elements in a NumPy array created with the following code?

```
import numpy as np
arr = np.array([1, 2, 3])
```

- Select - ▾ ✖

Correct answer: np.int32

✖

Points: 0/1

Consider the following NumPy array:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
```

What is the result of the following code?

```
arr_view = arr.view()
arr_view[0] = 10
print(arr[0])
```

○ 2

○ 10 ✓

○ 1

○ 3

✖

Points: 0/1

Consider the following Python code:

```
import tensorflow as tf
import matplotlib.pyplot as plt

# Create a black image
black_image = tf.ones((400, 600, 3), dtype=tf.uint8) * 0
black_image_np = black_image.numpy()

plt.axis('off')
plt.imshow(black_image_np)
plt.show()
```

What does the `* 0` operation do in the code?

- Select - ▾ ✖

Correct answer: Sets all pixel values to 0.

✖

Points: 0/2

Consider the following Python code:

```
import pandas as pd

data = {'A': [1, 2, 3, 4, 5],
       'B': ['a', 'b', 'c', 'd', 'e']}
df = pd.DataFrame(data)

result = df.tail()

print(result)
```

What will be the output of the above code?

○
```
   A  B
3  4  d
4  5  e
```

○
```
   A  B
0  1  a
1  2  b
2  3  c
3  4  d
4  5  e
```

○
```
   A  B
0  1  a
1  2  b
2  3  c
```

○
```
   A  B
1  2  b
2  3  c
3  4  d
4  5  e
```
✓

✖

Points: 0/1

Which of the following statements is true regarding creating a NumPy ndarray object from a list?

- ○ The resulting ndarray object has a fixed size and cannot be modified.
- ○ All elements in the list must have the same data type for the conversion to succeed. ✓
- ○ The resulting ndarray object has a data type of object.
- ○ The np.ndarray() function is preferred over np.array() for creating arrays from lists.

✖

Points: 0/1

What is the output of the following code?

```
import numpy as np

arr = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
print(arr.ndim)
```

- ○ 0
- ○ 1
- ○ 2
- ○ 3 ✓

✗

Points:
0/1

What is the result of executing the following code?

```
import numpy as np

my_list = [1, 2, 3, 4, 5]

my_array = np.array(my_list)

print(my_array.dtype)
```

- ○ float64
- ○ object
- ○ ndarray
- ○ int64 ✓

✗

Points:
0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3],
               [4, 5, 6]])
```

What is the dimensionality of the array `arr`?

- ○ 6
- ○ 2 ✓
- ○ 3
- ○ 1

✗

Points:
0/1

Consider the following NumPy arrays:

```
import numpy as np

arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

```
result = np.vstack((arr1, arr2))
print(result)
```

○
```
[[1 2]

[3 4]

[5 6]

[7 8]]
```
✓

○
```
[[[1 2]

[3 4]]

[[5 6]

[7 8]]]
```

○
```
[[1 2 5 6]

[3 4 7 8]]
```

○ Error: cannot stack arrays of different shapes

✖

Points:
0/3

Use the correct NumPy method to change the shape of an array from 1-D to 2-D. (Select all the correct three options)

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(________)
```

- □ (1,4)
- □ (4,3) ✔
- □ (2,6) ✔
- □ (3,4) ✔

✖

Points:
0/1

What is the result of executing the following code?

```
import numpy as np

my_tuple = (1, 2, 3, 4, 5)
my_array = np.array(my_tuple)
print(my_array.dtype)
```

- ○ ndarray
- ○ float64
- ○ int64 ✔
- ○ object

✖

Points:
0/1

What is the output of the following code?

```
import numpy as np

arr = np.array([1, 2, 3, 4], ndmin=5)

print('shape of array :', arr.shape)
```

- ○ shape of array : (1, 1, 1, 1, 4) ✔
- ○ shape of array : (1, 1, 2, 2, 4)
- ○ shape of array : (1, 2, 3, 4, 4)
- ○ shape of array : (4, 4, 3, 2, 1)

✖

Points:
0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
```

What is the dimensionality of the array `arr`?

- ○ 2
- ○ 3 ✔
- ○ 1
- ○ 4

✖

Points:
0/3

What functionality does SciPy provide?

- □ Statistical functions. ✔
- □ Linear algebra operations. ✔
- □ Optimization algorithms. ✔
- □ Creating pixel coloring

✗

Points:
0/1

Which of the following statements is true regarding creating a NumPy ndarray object from tuples?

- ○ The np.ndarray() function should be used instead of np.array() for converting tuples to arrays.
- ○ Tuples with different data types can be converted to ndarray objects without any issues.
- ○ Tuple elements must be explicitly cast to the desired data type for conversion to succeed. ✓
- ○ The resulting ndarray object will have the same shape as the input tuple.

✗

Points:
0/1

Consider the following Python code:

```
import pandas as pd

data = [10, 20, 30, 40, 50]
series = pd.Series(data, index=['a', 'b', 'c', 'd', 'e'])

print(series['c'])
```

What will be the output of the above code?

- ○ 10
- ○ 40
- ○ 30 ✓
- ○ 20

✗

Points:
0/1

Consider the following NumPy array:

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])
```

What is the result of the following slicing operation?

```
arr_slice = arr[:, -2:]
```

- ○ array([[2, 3, 4], [6, 7, 8], [10, 11, 12]])
- ○ array([[2, 3], [6, 7], [10, 11]])
- ○ array([[4], [8], [12]])
- ○ array([[3, 4], [7, 8], [11, 12]]) ✓

✗

Points:
0/3

Which of the following tasks can be performed using SciPy?

- □ Signal processing ✓
- □ Image processing. ✓
- □ Solving differential equations. ✓
- □ Data Cleaning

ETHNOTECH

QUESTION BANK

MACHINE LEARNING
SESSION1, SESSION2, SESSION 3

RNSIT-COE
2023-24

✘

Points: 0/1

What is machine learning?

○ An approach for designing hardware components with high computational power
○ A branch of artificial intelligence that deals with algorithms that can learn from data ✓
○ A method for creating virtual reality simulations
○ A technique for programming computers to perform specific tasks

✘

Points: 0/1

Which of the following is NOT a type of machine learning algorithm?

○ Semi-supervised Learning
○ Supervised Learning
○ Deterministic Learning ✓
○ Unsupervised Learning

✘

Points: 0/1

What is the main goal of unsupervised learning?

○ To make predictions based on labeled data
○ To learn from feedback provided by a teacher
○ To classify data into predefined categories
○ To discover hidden patterns or structures in data ✓

✘

Points: 0/1

Which evaluation metric is commonly used for classification tasks in machine learning?

○ Mean Absolute Error (MAE)
○ R-squared (R²)
○ Root Mean Squared Error (RMSE)
○ Accuracy ✓

✘

Points: 0/1

What is overfitting in machine learning?

○ When the model learns noise in the training data and performs poorly on unseen data ✓
○ When the model underfits the training data
○ When the model performs well on unseen data
○ When the model generalizes well to new data

✘

Points: 0/3

Which of the following are types of supervised learning algorithms? (Select all the 3 for full marks)

□ Decision Trees ✓
□ K-Nearest Neighbors
□ Support Vector Machines ✓
□ K-Means Clustering
□ Linear Regression ✓

✘

Points: 0/3

What are common methods for handling missing data in machine learning?

□ Using predictive models to estimate missing values ✓
□ Ignoring missing values during training
□ Dropping features with missing values
□ Removing observations with missing values ✓
□ Replacing missing values with the mean of the column ✓

✘

Points: 0/3

What are the different categories of algorithm classified based on desired outputs?

- □ Clustering type ✓
- □ Molding type
- □ Classification Type ✓
- □ Observational Type
- □ Regression Type ✓

✘

Points: 0/4

What are some limitations of machine learning techniques? (There are four right answers)

- □ Susceptibility to Overfitting ✓
- □ Limited Generalization to Unseen Data ✓
- □ Lack of Interpretability ✓
- □ Dependency on Quality of Data ✓
- □ Difficulty in Handling Non-Numeric Data
- □ Inability to Handle Large Datasets

✘

Points: 0/8

Match the following step by step process of Machine Learning in the right order of numbers

Pre-process the Data	- Select - ✘ **Correct answer:** 3
Train the Model	- Select - ✘ **Correct answer:** 6
Fine-tune the Model	- Select - ✘ **Correct answer:** 8
Choose a Model	- Select - ✘ **Correct answer:** 5
Define the Problem	- Select - ✘ **Correct answer:** 1
Split the Data	- Select - ✘ **Correct answer:** 4
Evaluate the Model	- Select - ✘ **Correct answer:** 7
Collect Data	- Select - ✘ **Correct answer:** 2

✘

Points: 0/1

```
import csv

data = [
    ['Name', 'Age', 'City'],
    ['John', 30, 'New York'],
    ['Alice', 25, 'San Francisco'],
    ['Bob', 35, 'Los Angeles']
]

# Writing to CSV file
with open(___________, mode='w', newline=' ') as file:
    writer = csv.writer(file)
    writer.writerows(data)

print("CSV file created successfully!")
```

For the given program, select the appropriate replacements at the missing section

- ○ Pass Nothing
- ○ Just File Name
- ○ File Path
- ○ File Name with CSV extension ✓

✘

Points: 0/2

```
import csv

with open(__________, mode='r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

For the given program, select both the appropriate replacements at the missing section

- □ File Path ✓
- □ File Name with csv extension ✓
- □ Pass Nothing
- □ File Name

✘

Points: 0/1

What does the `writerows()` function do in Python's `csv` module?

- ○ Reads multiple rows of data from a CSV file
- ○ Deletes multiple rows from a CSV file
- ○ Writes multiple rows of data to a CSV file ✓
- ○ Writes a single row of data to a CSV file

✘

Points: 0/1

What is the purpose of the `writerow()` function in Python's `csv` module?

- ○ Writes multiple rows of data to a CSV file
- ○ Deletes a single row of data from a CSV file
- ○ Reads a single row of data from a CSV file
- ○ Writes a single row of data to a CSV file ✓

✘

Points: 0/2

Which of the following statements is true regarding `writerows()` and `writerow()` functions?

- ○ writerows() is used for writing a single row of data, while writerow() is used for writing multiple rows of data
- ○ writerows() is used for writing multiple rows of data, while writerow() is used for writing a single row of data ✓
- ○ Both functions can only write a single row of data to a CSV file
- ○ Both functions are used for reading data from a CSV file

✘

Points: 0/1

```
import csv
csv_file_name = 'example.csv'
with open(csv_file_name, 'r') as csv_file:
  csv_reader = csv.reader(csv_file)
  for____in csv_reader:
        print(row)
```

For the given Program, fill in the missing code.

- ○ row ✓
- ○ len(rows in csv_file)
- ○ len(csv_file)
- ○ i

✘

Points: 0/1

```
import csv
csv_file = "data.csv"
with open(csv_file, mode='r') as file:
    reader = csv.reader(file)
    header =_______(reader)
print("Header:", header)
```

In the above program, fill the missing key word used to read the header of csv file.

- ○ readrow(0)
- ○ next ✓
- ○ head
- ○ read(1)

✘

Points: 0/2

```
import csv
csv_file_name = 'example.csv'
column_to_read = 'Name'
with open(csv_file_name, 'r') as csv_file:
  csv_reader = csv.reader(csv_file)
  header = next(csv_reader)
  column_index = header.index(column_to_read)
 ____________________
        _____________________
```

For the given program select the misisng code lines:

- ○ for row in csv_reader: print(row[column_index]) ✓
- ○ for row in csv_reader: print(row[row])
- ○ for col in csv_reader: for row in csv_reader: print(row[column_index])
- ○ for row in csv_reader: print(row[column_to_read])

✘

Points: 0/5

Which of the following are common forms of data in machine learning? (Select all that apply)

- □ Text data ✓
- □ Audio data ✓
- □ Graph data
- □ Numerical data ✓
- □ Categorical data ✓
- □ Image data ✓

✘

Points: 0/1

Data can come from various sources such as:

- ○ Surveys
- ○ User-generated content
- ○ All of the above ✓
- ○ Scientific experiments
- ○ Social media
- ○ Sensor data

✘

Points: 0/1

What is labeled data in the context of machine learning?

- ○ Data that contains only numerical values
- ○ Data that has been processed for visualization purposes
- ○ Data that is accompanied by its corresponding target variable or output ✓
- ○ Data that is not suitable for training machine learning models

✘

Points: 0/1

Which of the following is an example of labeled data?

- ○ A dataset of customer transactions with each transaction labeled as fraudulent or not fraudulent ✓
- ○ Social media posts without any annotations
- ○ A collection of images with no accompanying descriptions
- ○ Sensor readings from environmental monitoring stations

✘

Points: 0/1

What distinguishes unlabeled data from labeled data in machine learning?

- ○ Unlabeled data is always textual, while labeled data can be numerical or categorical
- ○ Unlabeled data contains missing values, while labeled data does not
- ○ Unlabeled data is exclusively used for testing machine learning models, while labeled data is used for training
- ○ Unlabeled data lacks any accompanying target variable or output ✓

✘

Points: 0/1

Labeled data is essential for supervised learning tasks.

- Select - ▾ ✘

Correct answer: True

✘

Points: 0/1

Unlabeled data is not useful for training machine learning models.

- Select - ▾ ✘

Correct answer: False

✘

Points: 0/1

In semi-supervised learning, both labeled and unlabeled data are utilized during model training.

- Select - ▾ ✘

Correct answer: True

✘

Points: 0/1

In machine learning, it is common practice to split a dataset into training, validation, and test sets.

- Select - ▾ ✘

Correct answer: True

✘

Points: 0/1

The training set is typically used to train the model, while the test set is used to tune hyperparameters.

- Select - ▾ ✘

Correct answer: False

✘

Points: 0/1

A common splitting ratio for the training, validation, and test sets is 70%, 15%, and 15%, respectively.

- Select - ▾ ✘

Correct answer: False

✘

Points: 0/1

What is the primary purpose of splitting a dataset into training and test sets in machine learning?

- ○ To visualize the data distribution
- ○ To reduce the computational complexity of the model
- ○ To evaluate the model's performance on unseen data ✓
- ○ To increase the size of the dataset

✘

Points: 0/1

Which of the following is a common technique for splitting a dataset into training and test sets?

- ○ Random Sampling ✓
- ○ Principal Component Analysis
- ○ Mean Squared Error
- ○ K-Means Clustering

✘

Points: 0/1

What is the purpose of using a validation set in addition to training and test sets?

- ○ To increase the size of the training set
- ○ To select the features for model training
- ○ To evaluate the model's performance during training
- ○ To reduce overfitting by tuning hyperparameters ✓

✘

Points: 0/3

Which of the following splitting ratios is commonly used for dividing a dataset into training and test sets? (Select all the three right answers)

- □ 90% training, 10% test ✓
- □ 80% training, 20% test ✓
- □ 50% training, 50% test
- □ 70% training, 30% test ✓

✘

Points: 0/1

What is the purpose of data validation in machine learning?

- ○ To ensure that the data is accurate and complete ✓
- ○ To visualize the distribution of the data
- ○ To prevent overfitting of the model
- ○ To reduce the computational complexity of the model

✘

Points: 0/1

What is the primary goal of data cleaning in machine learning?

- ○ To improve the quality of the dataset ✓
- ○ To reduce the computational complexity of the model
- ○ To make the dataset more complex
- ○ To increase the size of the dataset

✘

Points: 0/2

Which of the following are common techniques used in data cleaning?

- □ Normalizing data
- □ Adding noise to the dataset
- □ Imputing missing values ✓
- □ Removing duplicates ✓

✘

Points: 0/1

What is imputation in the context of data cleaning?

- ○ Splitting the dataset into training and test sets
- ○ Replacing missing values with estimated values ✓
- ○ Removing outliers from the dataset
- ○ Adding noise to the dataset

✘

Points:
0/1

Which of the following is a common method for handling outliers during data cleaning?

- ○ Ignoring the outlier during model training
- ○ Adding the outlier to a separate dataset
- ○ Replacing the outlier with the mean of the column
- ○ Deleting the entire row containing the outlier ✓

✘

Points:
0/1

What is the purpose of feature scaling in data cleaning?

- ○ To add noise to the dataset
- ○ To remove features with missing values
- ○ To standardize or normalize the range of features ✓
- ○ To reduce the number of features in the dataset

✘

Points:
0/1

What are the advantages of data processing in machine learning?

- ○ Reduced accuracy of predictions
- ○ Increased computational complexity
- ○ Improved model performance ✓
- ○ Decreased dataset size

✘

Points:
0/1

How does data processing contribute to better machine learning outcomes?

- ○ By optimizing the training process for faster convergence ✓
- ○ By enhancing the interpretability of the model
- ○ By reducing the need for feature engineering
- ○ By increasing the noise in the dataset

✘

Points:
0/1

```
   CustomerID       Name    Age   Gender           Address   Income
0           1   John Doe   25.0     Male  123 Main St, City  50000.0
1           2 Alice Smith  32.0   Female   456 Elm St, City  60000.0
2           3 Bob Johnson  NaN      Male  NaN                55000.0
3           4 Sarah Brown  28.0   Female  789 Oak St, City   70000.0
4           5 Michael Lee   42.0     NaN   NaN               65000.0
5           6         NaN  45.0     Male   987 Pine St, City     NaN
```

Select the appropritate code **to handle missing values in the "Age" column by replacing them with the mean age:**

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

b) df['Age'].fillna(mean_age, inplace=True)

c) df['Gender'] = df['Gender'].str.lower()

d) df['Address'] = df['Address'].str.strip()

e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - ▾ ✘

Correct answer: a

✘

Points:
0/1

```
   CustomerID         Name    Age    Gender             Address     Income
0           1     John Doe   25.0      Male    123 Main St, City   50000.0
1           2   Alice Smith  32.0    Female     456 Elm St, City   60000.0
2           3  Bob Johnson    NaN      Male    NaN                 55000.0
3           4  Sarah Brown   28.0    Female    789 Oak St, City    70000.0
4           5   Michael Lee   42.0      NaN    NaN                 65000.0
5           6          NaN   45.0      Male    987 Pine St, City       NaN
```

Select the appropritate code **to remove rows with missing values in the dataset:**

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

b) df.dropna(inplace=True)

c) df['Gender'] = df['Gender'].str.lower()

d) df['Address'] = df['Address'].str.strip()

e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - ▾ ✘

Correct answer: b

✘

Points:
0/1

```
   CustomerID         Name    Age    Gender             Address     Income
0           1     John Doe   25.0      Male    123 Main St, City   50000.0
1           2   Alice Smith  32.0    Female     456 Elm St, City   60000.0
2           3  Bob Johnson    NaN      Male    NaN                 55000.0
3           4  Sarah Brown   28.0    Female    789 Oak St, City    70000.0
4           5   Michael Lee   42.0      NaN    NaN                 65000.0
5           6          NaN   45.0      Male    987 Pine St, City       NaN
```

Select the appropritate code **to convert the "Gender" column to lowercase for consistency:**

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

b) df['Age'].fillna(mean_age, inplace=True)

c) df['Gender'] = df['Gender'].str.lower()

d) df['Address'] = df['Address'].str.strip()

e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - ▾ ✘

Correct answer: c

✘

Points: 0/1

```
   CustomerID         Name   Age  Gender            Address   Income
0           1     John Doe  25.0    Male  123 Main St, City  50000.0
1           2   Alice Smith  32.0  Female   456 Elm St, City  60000.0
2           3  Bob Johnson   NaN    Male                NaN  55000.0
3           4  Sarah Brown  28.0  Female   789 Oak St, City  70000.0
4           5  Michael Lee  42.0     NaN                NaN  65000.0
5           6          NaN  45.0    Male  987 Pine St, City      NaN
```

Select the appropritate code **to remove leading and trailing whitespaces in the "Address" column:**

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

b) df['Age'].fillna(mean_age, inplace=True)

c) df['Gender'] = df['Gender'].str.lower()

d) df['Address'] = df['Address'].str.strip()

e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - ✘

Correct answer: d

✘

Points: 0/1

```
   CustomerID         Name   Age  Gender            Address   Income
0           1     John Doe  25.0    Male  123 Main St, City  50000.0
1           2   Alice Smith  32.0  Female   456 Elm St, City  60000.0
2           3  Bob Johnson   NaN    Male                NaN  55000.0
3           4  Sarah Brown  28.0  Female   789 Oak St, City  70000.0
4           5  Michael Lee  42.0     NaN                NaN  65000.0
5           6          NaN  45.0    Male  987 Pine St, City      NaN
```

Select the appropritate code **to convert numeric values in the "Name" column to NaN (missing values):**

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

b) df['Age'].fillna(mean_age, inplace=True)

c) df['Gender'] = df['Gender'].str.lower()

d) df['Address'] = df['Address'].str.strip()

e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - ✘

Correct answer: e

✘

Points: 0/1

What is a key feature of NumPy?

- ○ Data manipulation and analysis through DataFrames
- ○ Efficient handling of large datasets
- ○ Implementation of advanced mathematical functions ✓
- ○ Plotting of data visualizations

✘

Points: 0/1

What is a key feature of Pandas?

- ○ Implementation of advanced mathematical functions
- ○ Efficient handling of large datasets
- ○ Plotting of data visualizations
- ○ Data manipulation and analysis through DataFrames ✓

✘

Points: 0/1

What is a key feature of Matplotlib?

- ○ Implementation of advanced mathematical functions
- ○ Efficient handling of large datasets
- ○ Data manipulation and analysis through DataFrames
- ○ Plotting of data visualizations ✓

✘

Points: 0/1

What is a key feature of SciPy?

- ○ Data manipulation and analysis through DataFrames
- ○ Efficient handling of large datasets
- ○ Implementation of advanced mathematical functions ✓
- ○ Plotting of data visualizations

✘

Points: 0/1

Which library is primarily used for scientific computing and mathematical operations in Python? a

- ○ Matplotlib
- ○ SciPy ✓
- ○ Pandas
- ○NumPy

✘

Points: 0/1

```
import numpy as np
data =_________([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])
np.savetxt("Filename.csv", data, delimiter=',')
```

The given data need to be converted to array, select the missing code to be get the program running properly.

- ○ np.random.rand()
- ○ np.linspace()
- ○ np.data()
- ○ np.array() ✓

✘

Points: 0/1

What is the purpose of the `numpy.savetxt()` function?

○ To load data from a text file into a numpy array.
○ To save a numpy array to a text file. ✓
○ To concatenate two numpy arrays along a specified axis.
○ To perform element-wise multiplication on two numpy arrays.

✘

Points: 0/1

What is the purpose of the `numpy.loadtxt()` function?

○ To convert a text file into a numpy array. ✓
○ To concatenate two numpy arrays along a specified axis.
○ To save a numpy array to a text file.
○ To perform element-wise multiplication on two numpy arrays.

✘

Points: 0/1

What is the code line to save a NumPy array to a text file using `numpy.savetxt()`?

- Select - ✘

Correct answer: np.savetxt("data.txt", arr, delimiter=",")

✘

Points: 0/1

What is the code line to load data from a text file into a NumPy array using `numpy.loadtxt()`?

- Select - ✘

Correct answer: np.loadtxt("data.txt", delimiter=",")

✘

Points: 0/1

What is the code line to save a NumPy array to a text file with a specified header using `numpy.savetxt()`?

- Select - ✘

Correct answer: np.savetxt("data.txt", arr, header="Header")

✘

Points: 0/1

What is the code line to skip the first row while loading data from a text file into a NumPy array using `numpy.loadtxt()`?

- Select - ✘

Correct answer: np.loadtxt("data.txt", skiprows=1)

✘

Points: 0/1

What is the correct way to create a Pandas DataFrame from a dictionary?

- Select - ✘

Correct answer: pd.DataFrame.from_dict(dict)

✘

Points: 0/1

What is the correct way to create a Pandas DataFrame from a CSV file?

- Select - ✘

Correct answer: pd.read_csv("data.csv")

✘

Points: 0/1

```
import pandas as pd

data = {'Name': ['John', 'Alice', 'Bob', 'Sarah'],
        'Age': [25, 32, 28, 35],
        'City': ['New York', 'San Francisco', 'Los Angeles', 'Chicago']}

df = pd.DataFrame(data)

________________________
```

You have to write the data in to a csv file using pands, select the approriate missing code lines

- Select - ✘

Correct answer: df.to_csv('output.csv', index=False)

✘

Points: 0/1

for given

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

What type of plot is most suitable for visualizing the relationship between the given `x` and `y` data?

- Select - ✘

Correct answer: Scatter plot

✘

Points: 0/1

for given

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

Which Matplotlib function is used to plot a bargraph?

- Select - ✘

Correct answer: plt.bar()

✘

Points: 0/1

for given

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

Which Matplotlib function is used to plot a linegraph?

- Select - ✘

Correct answer: plt.plot()

✘

Points: 0/1

What is the purpose of adding labels to the axes in a plot?

- Select - ✘

Correct answer: To provide context and interpretation to the plotted data

✘

Points: 0/1

Which Matplotlib function is used to display the plot?

- Select - ✘

Correct answer: plt.show()

✘

Points: 0/1

What does the following Matplotlib code snippet do?

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('My Graph')
plt.show()
```

- Select - ✘

Correct answer: It creates a line plot with the given x and y values.

✘

Points: 0/1

What does the following Matplotlib code snippet do?

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.scatter(x, y, color='red')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('My Graph')
plt.show()
```

- Select - ✘

Correct answer: It creates a scatter plot with the given x and y values.

✘

Points: 0/1

What does the following Matplotlib code snippet do?

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.hist(y, bins=5, color='green')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('MyGraph')
plt.show()
```

- Select - ✘

Correct answer: It creates a histogram of the y values.

✘

Points:
0/1

What does the following Matplotlib code snippet do?

```
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C', 'D']
sizes = [20, 30, 40, 10]

plt.pie(sizes, labels=labels)
plt.title('My Chart')
plt.show()
```

- Select - ✘

Correct answer: It creates a pie chart.

ML Scenario

✖

Points: 0/1

Scenario: Your dataset df contains an imbalanced binary target variable target. You want to ensure that the train-test split maintains the same proportion of each class as in the original dataset.

Which code snippet achieves this?

○ from sklearn.model_selection import train_test_split X = df.drop('target', axis=1) y = df['target'] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=X, random_state=42)

○ from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(df.drop('target', axis=1), df['target'], test_size=0.2)

○ from sklearn.model_selection import train_test_split X = df.drop('target', axis=1) y = df['target'] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

○ from sklearn.model_selection import train_test_split X = df.drop('target', axis=1) y = df['target'] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42) ✓

✖

Points: 0/1

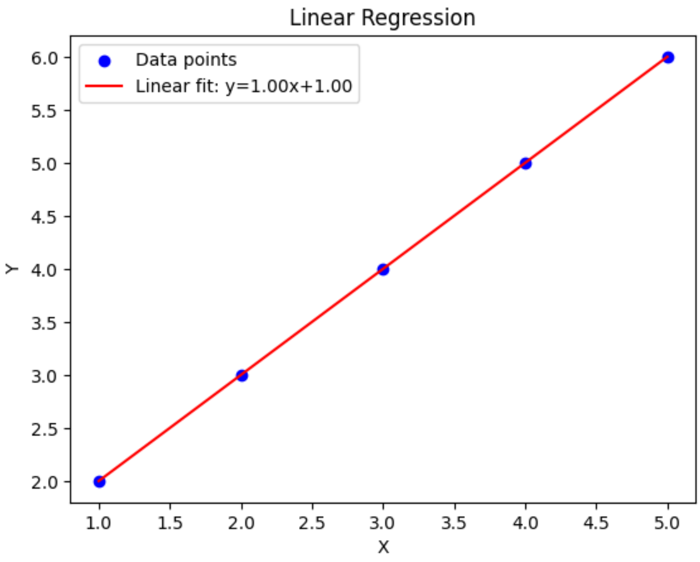
John is working on a machine learning project where he needs to perform a simple linear regression to fit a line to his data points. He uses the following code to achieve this:

```
import numpy as np

# Sample data
X = np.array([1, 2, 3, 4, 5])
Y = np.array([2, 3, 4, 5, 6])

# Perform linear regression
slope, intercept = np.polyfit(X, Y, 1)
```

Data Visualization for your reference:



○ slope = 1.5, intercept = 0.5

○ slope = 1.0, intercept = 1.0

○ slope = 0.5, intercept = 1.5

○ slope = 1.0, intercept = 0.0 ✓

✖

Points: 0/1

Sarah is using the LinearRegression class from the sklearn.linear_model module to perform a simple linear regression on her dataset. She uses the following code:

```
from sklearn.linear_model import LinearRegression

# Sample data
X = [[2], [3], [4], [5], [6]]
Y = [3, 5, 7, 9, 11]

model = LinearRegression()
model.fit(X, Y)

slope = model.coef_[0]
intercept = model.intercept_
```

Data Visualization for the reference:

Output image

○ slope = 2.0, intercept = -1.0

○ slope = 1.0, intercept = 1.0

○ slope = 1.5, intercept = 0.5

○ slope = 2.0, intercept = 1.0 ✓

✗

Points: 0/1

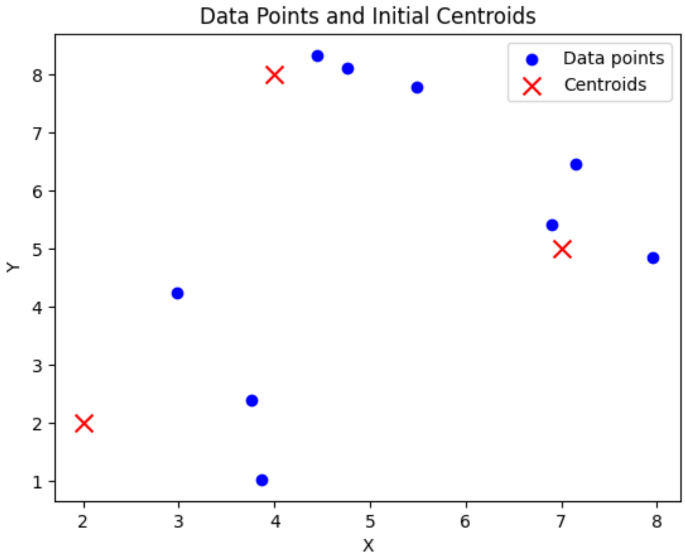
Alice is working on a clustering problem using a simple dataset. She has data points and initial centroid positions as shown in the code snippet below:

```
import numpy as np
import matplotlib.pyplot as plt

# Data points
data = np.array([[3.76405235, 2.40015721],
            [2.97873798, 4.2408932 ],
            [3.86755799, 1.02272212],
            [7.95008842, 4.84864279],
            [6.89678115, 5.4105985 ],
            [7.14404357, 6.45427351],
            [4.76103773, 8.12167502],
            [4.44386323, 8.33367433],
            [5.49407907, 7.79484174]])

# Initial centroids
centroids = np.array([[2, 2], [7, 5], [4, 8]])
```

Data Visualiztion for the reference:



○ Data points are scattered randomly with centroids at (2, 2), (7, 5), (4, 8). ✓

○ Data points form clusters with centroids at (2, 2), (7, 5), (4, 8).

○ Data points are clustered with centroids at (3, 3), (6, 5), (5, 7).

○ Data points form a straight line with centroids at (2, 2), (7, 5), (4, 8).

✖

Points: 0/1

Jane is analyzing a dataset containing measurements of various samples. She decides to use the KMeans clustering algorithm to partition the data into three clusters. The following code snippet shows how she performs the clustering:

```
from sklearn.cluster import KMeans
import numpy as np

# Data points
X = np.array(
[[3.76405235, 2.40015721],
[2.97873798, 4.2408932 ],
[3.86755799, 1.02272212],
[7.95008842, 4.84864279],
[6.89678115, 5.4105985 ],
[7.14404357, 6.45427351],
[4.76103773, 8.12167502],
[4.44386323, 8.33367433],
[5.49407907, 7.79484174]])

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(X)

# Get cluster centroids
centroids = kmeans.cluster_centers_

# Get cluster labels for each data point
labels = kmeans.labels_
```

- ○ The KMeans algorithm will create three clusters, but the cluster centroids cannot be determined.
- ○ The data points will be partitioned into three clusters, and each cluster will have a single centroid. ✓
- ○ The labels array will contain the distances of each data point from the cluster centroids.
- ○ The random_state parameter in KMeans ensures that the initial centroids are chosen randomly each time the code is run.

✖

Points: 0/1

What will be the output of the following Python code snippet?

```
from nltk.tokenize import word_tokenize, sent_tokenize

text = "NLTK makes it easy to perform NLP tasks."
tokens = word_tokenize(text)
sentences = sent_tokenize(text)

print(tokens)
print(sentences)
```

- ○ ['NLTK', 'makes', 'it', 'easy', 'to', 'perform', 'NLP', 'tasks', '.']
- ○ ['NLTK', 'makes', 'it', 'easy', 'to', 'perform', 'NLP', 'tasks', '.'] ✓
- ○ ['NLTK', 'makes', 'it', 'easy', 'to', 'perform', 'NLP', 'tasks', '.']
- ○ ['NLTK', 'makes', 'it', 'easy', 'to', 'perform', 'NLP', 'tasks']

✖

Points: 0/1

What will be the output of the following Python code snippet?

```
import torch

List1 = [12, 34, 56]
List2 = [123, 456, 789]
V1 = torch.tensor(List1)
V2 = torch.tensor(List2)

VTMUL = V1 * V2

print("Result:", VTMUL)

print("Shape of vector1:", V1.shape)
print("Size of Result Vector:", VTMUL.size())
```

○ Result: tensor([1476, 15424, 44064])
Shape of vector1: torch.Size([3])
Size of Result Vector: 3

○ Result: tensor([1476, 15424, 44064])
Shape of vector1: [3]
Size of Result Vector: torch.Size([3])

○ Result: tensor([1476, 15424, 44064])
Shape of vector1: [3]
Size of Result Vector: 3

○ Result: tensor([1476, 15424, 44064])
Shape of vector1: torch.Size([3])
Size of Result Vector: torch.Size([3])
✓

✖

Points: 0/1

Suppose you are developing a program to analyze student performance in a class. The program uses Python to process student grades stored in a list and calculates various statistics. Consider the following modified code snippet:

```
import numpy as np

# Sample student grades
grades = [82, 91, 78, 85, 90, 87, 89, 95, 88, 84]

# Convert grades to a NumPy array
grades_array = np.array(grades)

# Calculate mean and standard deviation of grades
mean_grade = np.mean(grades_array)
std_dev_grade = np.std(grades_array)

# Determine students above the mean grade
above_mean_count = np.sum(grades_array > mean_grade)

print("Mean grade:", mean_grade)
print("Standard deviation of grades:", std_dev_grade)
print("Number of students above the mean grade:", above_mean_count)
```

○ Mean grade: 86.5
Standard deviation of grades: 4.674
Number of students above the mean grade: 7

○ Mean grade: 87.9
Standard deviation of grades: 4.674
Number of students above the mean grade: 6
✓

○ Mean grade: 86.5
Standard deviation of grades: 4.674
Number of students above the mean grade: 6

○ Mean grade: 87.9
Standard deviation of grades: 4.674
Number of students above the mean grade: 7

✖

Points: 0/1

You are developing a program to calculate the area of a circle using PyTorch for scientific computation.

The program prompts the user to input the radius of the circle and then computes the area using the formula Area=π×R2\text{Area} = \pi \times R^2Area=π×R2, where π\piπ is approximated as 3.14.

Consider the following code snippet with the formula missing:

```
import torch

S1 = torch.tensor(3.14)

R = eval(input("Enter the radius of circle: "))
R = torch.tensor(R)

# Missing formula to calculate area A

__________________________

print("Area of circle is: ", A)
```

Which of the following options correctly completes the missing formula to compute the area A of the circle?

○ A = S1 * R * R

○ A = S1 * (R * R)

○ A = S1 * R ** 2 ✓

○ A = S1 * R * R * R

✗

Points: 0/1

You are working with a Python program that uses the Pandas library to create and manipulate Series objects. Consider the following code snippet:

```
import pandas as pd

Data = ["Boing747", "Boing787_Dreamliner", "Boing777"]

A = pd.Series(Data)
print(A)
print(A[2],"\n")

B = pd.Series(Data, index=["A1","A2","A3"])
print(B)
print(B["A1"])
```

What will be the output of the following Python code snippet?

○
```
0            Boing747
1    Boing787_Dreamliner
2            Boing777
dtype: object
Boing777

A1            Boing747
A2    Boing787_Dreamliner
A3            Boing777
dtype: object
Boing777
```

○
```
0            Boing747
1    Boing787_Dreamliner
2            Boing777
dtype: object
Boing777

A1            Boing747
A2    Boing787_Dreamliner
A3            Boing777
dtype: object
Boing787_Dreamliner
```

○
```
0            Boing747
1    Boing787_Dreamliner
2            Boing777
dtype: object
Boing747

A1            Boing747
A2    Boing787_Dreamliner
A3            Boing777
dtype: object
Boing787_Dreamliner
```

○
```
0            Boing747
1    Boing787_Dreamliner
2            Boing777
dtype: object
Boing777

A1            Boing747
A2    Boing787_Dreamliner
A3            Boing777
dtype: object
Boing747
```

✓