Review Test Submission: Mock test 2023/24 - Second assessed online test

User	Rakshit Yadav
Course	BMAN73701 Programming in Python for Business Analytics 2023-24 1st Semester
Test	Mock test 2023/24 - Second assessed online test
Started	10/11/23 04:17
Submitted	30/11/23 06:22
Status	Completed
Attempt Score	Mark not available.
Time Elapsed	482 hours, 5 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback

Question 1

9.0909 out of 10 points

Let us assume 'sales' is a Numpy matrix that contains amount sold per department (rows) per year (columns).

Fill in the missing blanks so that each line of code matches the comment the precedes the line.

Every input to a blank is exactly one word, symbol or number. NO spaces should be used (either before or after a word/symbol).

```
# 1. Calculate total number of sales of each department

total = np.[sum](sales, axis=[axis1])

# 2. Find the store with the largest total sales

best_store = np.[argmax](total)


# 3. Find if any store in any year did not have sales :
```

← OK

```
np.[all](sales > 0)

# 4. Find which stores had at least one year without sales:


np.[any](sales == 0, axis=[axis_1])


# 5. Find the largest sale amounts per year:


np.[max](sales, axis=[axis_0])

# 6. For each store, calculate its maximum sales of all
years, then find index of the store with the smallest
maximum.


np.[argmin](np.[max2](sales, axis=[axis_11]))
```

Specified Answer for: sum	sum
Specified Answer for: axis1	1
Specified Answer for: argmax	max
Specified Answer for: all	all
Specified Answer for: any	any
Specified Answer for: axis_1	1
Specified Answer for: max	max
Specified Answer for: axis_0	0
Specified Answer for: argmin	argmin
Specified Answer for: max2	max
Specified Answer for: axis_11	1

Correct Answers for: sum		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	sum	

Correct Answers for: axis1		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	1	

Correct Answers for: argmax

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	argmax	
Correct Answers for: all		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	all	Case Sensitive
Correct Answers for: any		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	any	Case Sensitive
Correct Answers for: axis_1		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	1	Case Sensitive
Correct Answers for: max		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	max	Case Sensitive
Correct Answers for: axis_0		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	0	Case Sensitive
Correct Answers for: argmin		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	argmin	Case Sensitive
Correct Answers for: max2		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	max	Case Sensitive
Correct Answers for: axis_11		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	1	

Question 2

10 out of 10 points

Assuming that 'A' is a Numpy matrix with 1000 rows and 500 columns, match each line of code with the most appropriate description.

Question	Correct Match	Selected Match
Select all rows and the first column	3. `A[:,:1]`	3. `A[:,:1]`
Select all columns and the first row	2. `A[:1,:]`	2. `A[:1,:]`

Select all columns and the last row	4. `A[-1:, 0:]`	4. `A[-1:, 0:]`
Select all rows and the last column	5. `A[::-1, -1:]`	5. `A[::-1, -1:]`
Select all columns except the last one and all rows	1. `A[::-1, :-1]`	1. `A[::-1, :-1]`

All Answer Choices

1. `A[::-1, :-1]`
2. `A[:1,:]`
3. `A[:,:1]`
4. `A[-1:, 0:]`
5. `A[::-1, -1:]`

Question 3

10 out of 10 points

You are given a Numpy matrix P that contains N rows and M columns. Each row of the matrix P represents one data point (i.e., you have N data points) and each column one numerical feature (i.e., you have M numerical features). You wish to apply Standardization (Z-score normalization) to this data, however, you missed the lecture about Data Preprocessing and you don't know of any function to do this. You decide to implement it yourself only using Numpy. To perform Standardization, you need to do the following steps:

1. For all values of each feature X, that is, $X = \{x_1, x_2, \ldots, x_N\}$, you need to do the following operation:

$$x'_i = \frac{x_i - \overline{X}}{\sigma_X}$$

where $\overline{X}$ and σ_X are, respectively, the mean and standard deviation of all values of X. This transformation must be appplied to each of the M features of the matrix.

2. If the standard deviation of a feature before preprocessing is less than 1e-5, that feature is constant for all purposes and it must be removed from the data.

Select the correct lines of Python for this task.

Selected Answer:

```
tmp = P[:, P.std(axis=0) >= 1e-5]
P_norm = (tmp - tmp.mean(axis=0)) /
tmp.std(axis=0)
```

Answers:

```
tmp = (P - P.mean(axis=1)[:,np.newaxis]) /
P.std(axis=1)[:,np.newaxis]
P_norm = tmp[tmp.std(axis=1) >= 1e-5, :]
```

```
P_norm = (P[:, P.std(axis=0) >= 1e-5] -
P.mean(axis=0)) / P.std(axis=0)
```

```
tmp = P[:, P.std(axis=0) >= 1e-5]
P_norm = (tmp - tmp.mean(axis=0)) /
tmp.std(axis=0)
```

```
if P.std() >= 1e-5:
    P_norm = P - P.mean() / P.std()
```

```
P_norm = (P[P.std(axis=1) >= 1e-5, :] -
P.mean(axis=1)[:,np.newaxis]) / (P.std(axis=1)
[:,np.newaxis])
```

```
tmp = P[:, P.std(axis=0) < 1e-5]
P_norm = (tmp - tmp.mean(axis=0)) /
tmp.std(axis=0)
```

```
tmp = P[P.std(axis=1) < 1e-5, :]
P_norm = (tmp - tmp.mean(axis=1)[:,np.newaxis]) /
(tmp.std(axis=1)[:,np.newaxis])
```

```
tmp = (P - P.mean(axis=0)) / P.std(axis=0)
P_norm = tmp[:, tmp.std(axis=0) >= 1e-5]
```

```
tmp = P[P.std() >= 1e-5]
P_norm = (tmp - tmp.mean(tmp)) / tmp.std()
```

```
tmp = P[P.std(axis=1) >= 1e-5, :]
P_norm = (tmp - tmp.mean(axis=1)[:,np.newaxis]) /
(tmp.std(axis=1)[:,np.newaxis])
```

Question 4

4 out of 10 points

Assuming that x is a numpy vector of shape (10,) and A is a numpy matrix of shape (10,5), match each erroneous expression with the corect explanation of the error.

Question	Correct Match	Selected Match
`A + x`	2. Gives an error because the number of columns of A is incompatible with the shape of x.	1. There is no error. The code is correct.
`A.reshape(10,5) + x.reshape(10,5)`	5. Gives an error because x does not have enough elements to fill the shape requested.	6. Gives an error because x has only one dimension, i.e., x is a vector not a matrix.
`A.reshape(10,) + x.reshape(10,1)`	3. Gives an error because A has too many elements to fill the shape requested.	3. Gives an error because A has too many elements to fill the shape requested.
`A.reshape(5,10) + x.reshape(10,1)`	4. Gives an error because the number of rows of A is incompatible with the shape of x.	4. Gives an error because the number of rows of A is incompatible with the shape of x.
`A.sum(axis=1) + x.sum(axis=1)`	6. Gives an error because x has only one dimension, i.e., x is a vector not a matrix.	2. Gives an error because the number of columns of A is incompatible with the shape of x.

All Answer Choices

1. There is no error. The code is correct.

2. Gives an error because the number of columns of A is incompatible with the shape of x.

3. Gives an error because A has too many elements to fill the shape requested.

4. Gives an error because the number of rows of A is incompatible with the shape of x.

5. Gives an error because x does not have enough elements to fill the shape requested.

6.
Gives an error because x has only one dimension, i.e., x is a vector not a matrix.

Question 5

7.14285 out of 10 points

Given the following two data frames:

Sales_Company_A

"Date"	"Profit"	"Phone"
01-01-2021	2.2	"iPhone"
01-01-2021	2.5	"Android"
02-01-2021	2.4	"iPhone"
02-01-2021	2.9	"Android"

Sales_Company_B

"Date"	"iPhone"	"Android"
01-01-2021	2.2	2.5
02-01-2021	2.4	2.9
03-01-2021	2.5	3.1
04-01-2021	2.6	2.0

Fill the blanks of the following sentences:

- The shape of Sales_Company_B is **[wide]** because **[at_least_one_var]** appears **[two_columns]**.
- The dataset Sales_Company**_[A]** is tidy because it is **[long_two]** and **[each_obs]** appears **[different_row]**.

Selected Answer:

Given the following two data frames:

Sales_Company_A

"Date"	"Profit"	"Phone"
01-01-2021	2.2	"iPhone"
01-01-2021	2.5	"Android"
02-01-2021	2.4	"iPhone"
02-01-2021	2.9	"Android"

Sales_Company_B

"Date"	"iPhone"	"Android"
01-01-2021	2.2	2.5

02-01-2021	2.4	2.9
03-01-2021	2.5	3.1
04-01-2021	2.6	2.0

Fill the blanks of the following sentences:

- The shape of Sales_Company_B is ✓ **wide** because ✓ **at least one variable** appears ✓ **in two columns**.
- The dataset Sales_Company_✓ **A** is tidy because it is ✓ **long** and ✗ **each variable** appears ✗ **in a different column**.

Answers: Given the following two data frames:

Sales_Company_A

"Date"	***"Profit"***	***"Phone"***
01-01-2021	2.2	"iPhone"
01-01-2021	2.5	"Android"
02-01-2021	2.4	"iPhone"
02-01-2021	2.9	"Android"

Sales_Company_B

"Date"	***"iPhone"***	***"Android"***
01-01-2021	2.2	2.5
02-01-2021	2.4	2.9
03-01-2021	2.5	3.1
04-01-2021	2.6	2.0

Fill the blanks of the following sentences:

- The shape of Sales_Company_B is ✓ **wide** because ✓ **at least one variable** appears ✓ **in two columns**.
- The dataset Sales_Company_✓ **A** is tidy because it is ✓ **long** and ✓ **each observation** appears ✓ **in a different row**.

All Answer Choices

- wrong
- tidy
- wide
- long
- each variable
- each observation
- in a different row
- in a different column
- at least one observation
- at least one variable
- in two columns
- in two rows
- A
- B

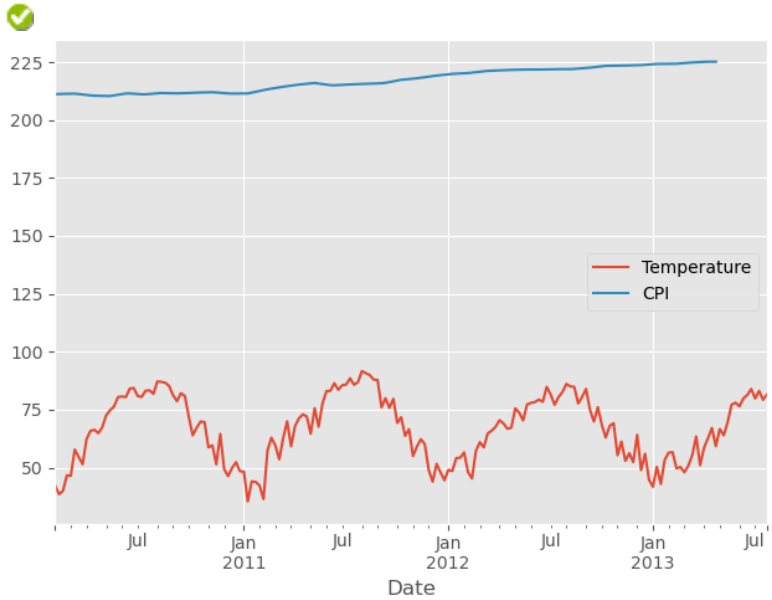
Question 6

10 out of 10 points

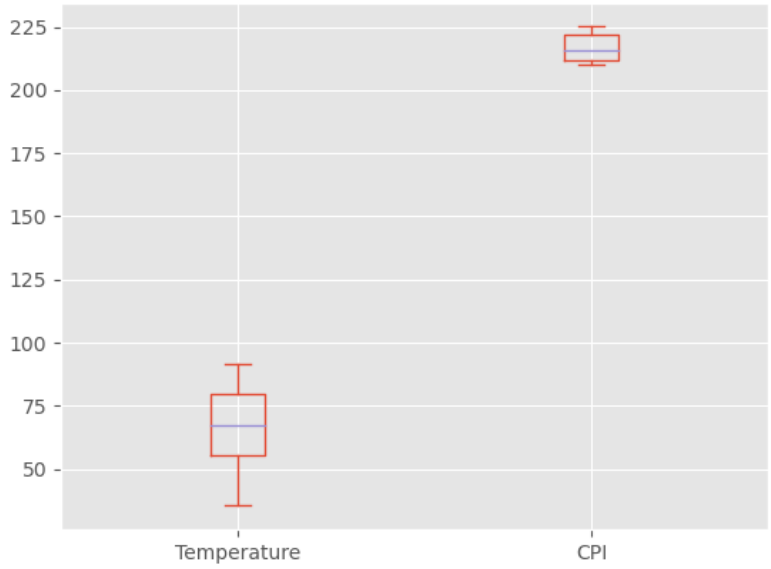
Select the plot that most closely matches the expected output of this line of Python code:

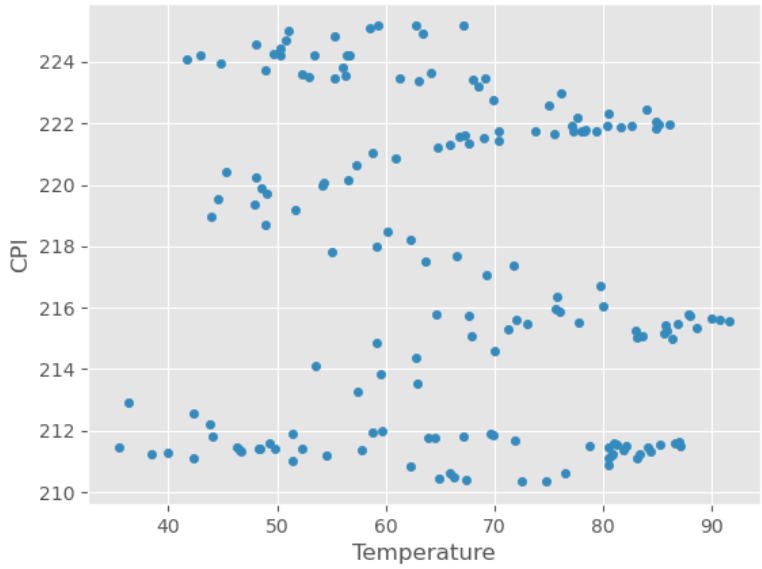
```
df[['Temperature','CPI']].plot()
```

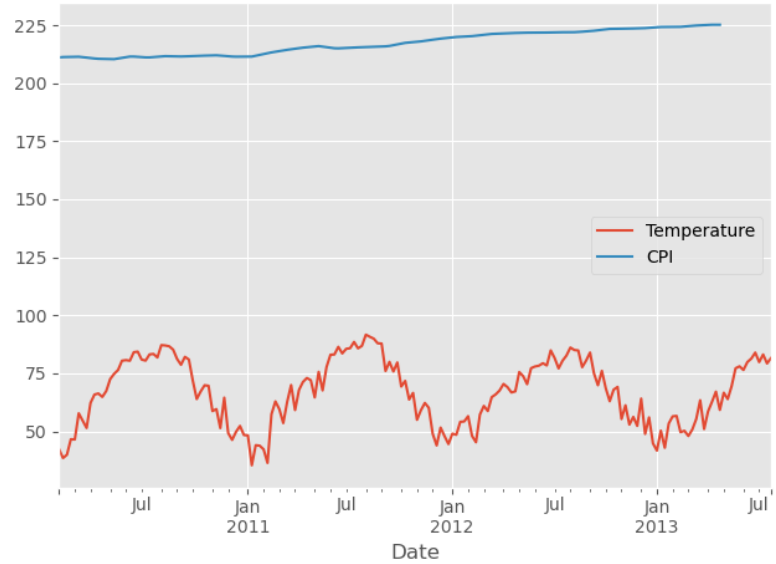
Selected Answers:

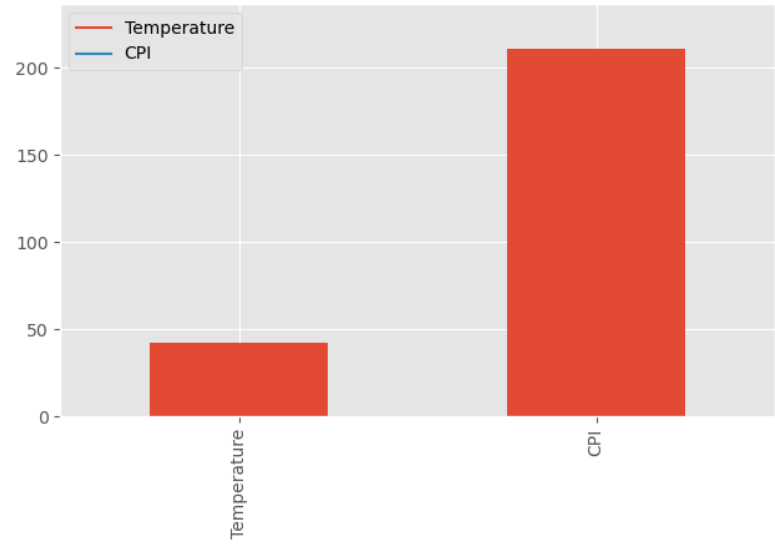


Answers:


225
200
175
150
125
100
75
50
Temperature
CPI


224
222
220
218
216
214
212
210
CPI
40
50
60
70
80
90
Temperature





Question 7

0 out of 10 points

Which of the following statements are true regarding DataFrames in Pandas? Mark all that apply. Incorrect answers penalize.

Selected Answers:

Automatic index alignment between DataFrames only happens after we set the index of the DataFrames to the values of some column.

Setting the index of a DataFrame to a column changes the labels of the rows to the values of the column chosen.

When summing two columns of two different DataFrames, for example
`df1['col1'] + df2['col2']`
the result is a new column where each row is the sum of the values of the corresponding rows of each DataFrame.

Answers:

When summing two columns of two different DataFrames, for example
`df1['col1'] + df2['col2']`
the result depends on the index of each DataFrame.

Automatic index alignment between DataFrames only happens after we set the index of the DataFrames to the values of some column.

Setting the index of a DataFrame to a column changes the labels of the rows to the values of the column chosen.

When summing two columns of two different DataFrames, for example
`df1['col1'] + df2['col2']`
If one DataFrame has fewer rows than the other, then we will get missing values.

When summing two columns of two different DataFrames, for example
`df1['col1'] + df2['col2']`
the result is a new column where each row is the sum of the values of the corresponding rows of each DataFrame.

Question 8

4 out of 10 points

Mark all the sentences that are true regarding matplotlib. Incorrect answers penalise.

Selected Answers:

In matplotlib, when you place an axes within an axes you create a plot on top of another plot.

In matplotlib, a figure may contain several axes but axes do not contain figures.

In matplotlib, plots are objects that have their own methods (functions), which may be used to modify the plots.

If you want to create a large number of similar plots, it is better to use **plt.show()** after creating each plot to save it to a file.

Answers:

In matplotlib, when you place an axes within an axes you create a plot on top of another plot.

Plots created with Pandas and with Matplotlib are incompatible because they use different ways of plotting.

In matplotlib, a figure may contain several axes but axes do not contain figures.

Pandas uses matplotlib to create plots.

The Sypder GUI allows creating more complex plots than matplotlib.

In matplotlib, when you place a figure within a figure you create a plot on top of another plot.

In matplotlib, a figure may contain several axes and an axes may contain several figures.

In matplotlib, plots are objects that have their own methods (functions), which may be used to modify the plots.

If you want to create a large number of similar plots, it is better to NOT use **plt.show()** after creating each plot.

If you want to create a large number of similar plots, it is better to use **plt.show()** after creating each plot to save it to a file.

Question 9

8 out of 10 points

Assuming that 'X' is a Numpy matrix and 'df' is a Pandas DataFrame, fill the blanks in the following code to match the comments.

Every input to a blank is exactly one word, number or symbol. NO spaces should be used (either before or after a word/symbol).

1. Calculate the mean value of each column of X taking into account missing

values.

result = np.**[mean]**(X, axis=**[axis0]**)

2. Calculate the mean value of each column of X ignoring missing values.

result = np.**[nanmean]**(X, axis=**[axis0_]**)

3. Replace missing values in df with the value 1

df = df.**[fillna]**(1)

4. Remove columns with missing values

df = df.**[dropna]**(axis=**[axis1__]**)

5. Remove rows with missing values

df = df.**[dropna2]**(axis=**[axis0__]**)

6. Calculate the mean value of each column of df ignoring missing values.

result = df.**[mean2]**()

Specified Answer for: mean	✗ nanmean
Specified Answer for: axis0	✓ 0
Specified Answer for: nanmean	✗ mean
Specified Answer for: axis0_	✓ 0
Specified Answer for: fillna	✓ fillna
Specified Answer for: dropna	✓ dropna
Specified Answer for: axis1__	✓ 1
Specified Answer for: dropna2	✓ dropna
Specified Answer for: axis0__	✓ 0
Specified Answer for: mean2	✓ mean

Correct Answers for: mean		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
✓ *Exact Match*	mean	
Correct Answers for: axis0		

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	0	
Correct Answers for: nanmean		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	nanmean	
Correct Answers for: axis0_		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	0	
Correct Answers for: fillna		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	fillna	Case Sensitive
Correct Answers for: dropna		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	dropna	Case Sensitive
Correct Answers for: axis1__		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	1	
Correct Answers for: dropna2		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	dropna	Case Sensitive
Correct Answers for: axis0__		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	0	
Correct Answers for: mean2		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	mean	

Question 10

8 out of 10 points

Fill the blanks in the following python code to match the comments.

Every input to a blank is exactly one word, number or symbol.

NO spaces should be used (either before or after a word/symbol).

Assume that any required '**import**' or preprocessing step has already been done earlier in the code.

```
# 1. Split the data into 75% training and 25% test data sets

X_train, X_test, y_train, y_test = [train_test_split]
```

```
(data.drop('answer' axis=1), data.answer, test_size = [.75])


# 2. Train a decision tree on the training data to predict
when the 'answer' is 0 or 1 (column 'answer' of the data).

dt = [DecisionTreeClassifier].[fit2]([X_train], [y_train])

# 3. Calculate the scores for 5-fold cross-validation

scores = [cross_val_score](dt, [X_train2], [y_train2], cv = [five])
```

Specified Answer for: train_test_split	train_test_split
Specified Answer for: .75	0.25
Specified Answer for: DecisionTreeClassifier	DecisionTreeClassifier
Specified Answer for: fit2	fit
Specified Answer for: X_train	X_train
Specified Answer for: y_train	y_train
Specified Answer for: cross_val_score	crass_val_score
Specified Answer for: X_train2	X_train
Specified Answer for: y_train2	y_train
Specified Answer for: five	5

Correct Answers for: train_test_split

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	train_test_split	Case Sensitive

Correct Answers for: .75

Evaluation Method	Correct Answer	Case Sensitivity
Contains	.75	

Correct Answers for: DecisionTreeClassifier

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	DecisionTreeClassifier	

Correct Answers for: fit2

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	fit	Case Sensitive

Correct Answers for: X_train

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	X_train	

Correct Answers for: y_train

Evaluation Method	Correct Answer	Case Sensitivity
Exact Match	y_train	
Correct Answers for: cross_val_score		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	cross_val_score	Case Sensitive
Correct Answers for: X_train2		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	X_train	
Correct Answers for: y_train2		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	y_train	
Correct Answers for: five		
Evaluation Method	**Correct Answer**	**Case Sensitivity**
Exact Match	5	

Thursday, 30 November 2023 06:23:54 o'clock GMT