Why am I getting the error “No module named ‘sklearn.cross_validation'” when trying to use the Fix module?

Authored by

stats writer

June 27, 2024

The error "No module named 'sklearn.cross_validation'" occurs when attempting to use the Fix module because the cross_validation module has been removed from the latest version of scikit-learn (sklearn). This module was deprecated in version 0.18 and completely removed in version 0.20. In order to use the Fix module, a different module such as sklearn.model_selection should be used instead, as it contains similar functionality. It is recommended to update the code to use the appropriate module in order to resolve this error.

Fix: No module named 'sklearn.cross_validation'

One error you may encounter when using Python is:

ModuleNotFoundError: No module named 'sklearn.cross_validation'

This error usually occurs when you attempt to import the train_test_split function from sklearn using the following line:

from sklearn.cross_validationimport train_test_split

However, the cross_validation sub-module has been replaced with the model_selection sub-module, so you need to use the following line instead:

from sklearn.model_selectionimport train_test_split

The following example shows how to resolve this error

in practice.

How to Reproduce the Error

Suppose we would like to use the train_test_split function from sklearn to split a pandas DataFrame into training and testing sets.

Suppose we attempt to use the following code to import the train_test_split function:

```
from sklearn.cross_validationimport train_test_split

ModuleNotFoundError: No module named 'sklearn.cross_validation'
```

We receive an error because we used the wrong sub-module name when attempting to import the train_test_split function.

How to Fix the Error

To fix this error, we simply need to use the model_selection sub-module instead:

```
from sklearn.model_selectionimport train_test_split
```

This time we don't receive any error.

We could then proceed to use the train_test_split function to split a pandas DataFrame into a training and testing set:

```
from sklearn.model_selectionimport train_test_split
import pandas as pd
import numpy as np

#make this example reproducible
np.random.seed(1)

#create DataFrame with 1000 rows and 3 columns
df = pd.DataFrame({'x1': np.random.randint(30, size=1000),
'x2': np.random.randint(12, size=1000),
'y': np.random.randint(2, size=1000)})

#split original DataFrame into training and testing sets
train, test = train_test_split(df, test_size=0.2, random_state=0)

#view first few rows of each set
print(train.head())
```

```
x1 x2 y
687 16 2 0
500 18 2 1
332 4 10 1
979 2 8 1
817 11 1 0

print(test.head())

x1 x2 y
993 22 1 1
859 27 6 0
298 27 8 1
553 20 6 0
672 9 2 1
```

We're successfully able to use the train_test_split function without any error.

The following tutorials explain how to fix other common errors in Python:

How to Fix: ValueError: cannot convert float NaN to integer