================================ **Human Message** =================================

You have been provided some tools to access the files of scikit-learn repository. You are currently in the /scikit-learn/ directory. You will be provided with an issue. Your task is to pin-point those files with line numbers from the repository that are likely to cause that issue.

Problem Statement: Support nullable pandas dtypes in `unique\_labels`

### Describe the workflow you want to enable

I would like to be able to pass the nullable pandas dtypes ("Int64", "Float64", "boolean") into sklearn's `unique\_labels` function. Because the dtypes become `object` dtype when converted to numpy arrays we get `ValueError: Mix type of y not allowed, got types {'binary', 'unknown'}`:

Repro with sklearn 1.2.1

```py

import pandas as pd

import pytest

from sklearn.utils.multiclass import unique\_labels

for dtype in ["Int64", "Float64", "boolean"]:

y\_true = pd.Series([1, 0, 0, 1, 0, 1, 1, 0, 1], dtype=dtype)

y\_predicted = pd.Series([0, 0, 1, 1, 0, 1, 1, 1, 1], dtype="int64")

with pytest.raises(ValueError, match="Mix type of y not allowed, got types"):

unique\_labels(y\_true, y\_predicted)

```

### Describe your proposed solution

We should get the same behavior as when `int64`, `float64`, and `bool` dtypes are used, which is no error:

```python

import pandas as pd

from sklearn.utils.multiclass import unique\_labels

for dtype in ["int64", "float64", "bool"]:

y\_true = pd.Series([1, 0, 0, 1, 0, 1, 1, 0, 1], dtype=dtype)

y\_predicted = pd.Series([0, 0, 1, 1, 0, 1, 1, 1, 1], dtype="int64")

unique\_labels(y\_true, y\_predicted)

```

### Describe alternatives you've considered, if relevant

Our current workaround is to convert the data to numpy arrays with the corresponding dtype that works prior to passing it into `unique\_labels`.

### Additional context

\_No response\_

Hint:

Use tool calls only if necessary. You are only allowed to use one most appropriate tool at a time. Call the tool you think is the most appropriate one and let the tols supply its output to the next agent. The next agent will continue to use more tools as necessary.

Remember to only use one most appropriate tool. If you don't know about the directories or files in the current directories, use 'ls' tool to list them.

Only provide the file paths and the line numbers that are the cause of this issue in the format:

File: ... Line: ...

File: ...

================================== **Ai Message** ==================================

Tool Calls:

find\_files (call\_VOpmyvbwslDlFOXUQqbG6JBB)

Call ID: call\_VOpmyvbwslDlFOXUQqbG6JBB

Args:

file\_name: unique\_labels

================================= **Tool Message** =================================

Name: find\_files

Files found:

================================== **Ai Message** ==================================

Tool Calls:

search\_dir (call\_BKPF7baNHhXOvXYam75XUXFR)

Call ID: call\_BKPF7baNHhXOvXYam75XUXFR

Args:

path: sklearn/utils

search\_term: def unique\_labels

================================= **Tool Message** =================================

Name: search\_dir

Files found:

File: /mnt/d/Ahsan/Job/AI Research/SWE bench/techniques/agentic-workflow/test-repos/scikit-learn/sklearn/utils/multiclass.py, Line: 44

================================== **Ai Message** ==================================

Tool Calls:

open\_file (call\_6cEpDqJK5QaEk9ERfuA1am4H)

Call ID: call\_6cEpDqJK5QaEk9ERfuA1am4H

Args:

path: sklearn/utils/multiclass.py

line\_number: 44

max\_lines: 20

================================= **Tool Message** =================================

Name: open\_file

Showing contents of File: /scikit-learn/sklearn/utils/multiclass.py starting from 44

44: def unique\_labels(\*ys):

45: """Extract an ordered array of unique labels.

46:

47: We don't allow:

48: - mix of multilabel and multiclass (single label) targets

49: - mix of label indicator matrix and anything else,

50: because there are no explicit labels)

51: - mix of label indicator matrices of different sizes

52: - mix of string and integer labels

53:

54: At the moment, we also don't allow "multiclass-multioutput" input type.

55:

56: Parameters

57: ----------

58: \*ys : array-likes

59: Label values.

60:

61: Returns

62: -------

63: out : ndarray of shape (n\_unique\_labels,)

================================== **Ai Message** ==================================

Tool Calls:

search\_file (call\_bO9VemFte520K4Cu6KTlXhx2)

Call ID: call\_bO9VemFte520K4Cu6KTlXhx2

Args:

path: sklearn/utils/multiclass.py

search\_term: ValueError

================================= **Tool Message** =================================

Name: search\_file

Searching for 'ValueError' in /scikit-learn/sklearn/utils/multiclass.py

78: raise ValueError("No argument has been passed.")

86: raise ValueError("Mix type of y not allowed, got types %s" % ys\_types)

100: raise ValueError(

107: raise ValueError("Unknown label type: %s" % repr(ys))

117: raise ValueError("Mix of label input types (string and number)")

156: # DeprecationWarning will be replaced by ValueError, see NEP 34

162: except (np.VisibleDeprecationWarning, ValueError):

207: raise ValueError("Unknown label type: %r" % y\_type)

290: raise ValueError(

296: raise ValueError("y cannot be class 'SparseSeries' or 'SparseArray'")

301: # DeprecationWarning will be replaced by ValueError, see NEP 34

310: except (np.VisibleDeprecationWarning, ValueError):

322: raise ValueError(

385: raise ValueError("classes must be passed on the first call to partial\_fit.")

390: raise ValueError(

================================== **Ai Message** ==================================

The issue is likely caused by the following lines in the file where the `ValueError` is raised for mixed types:

- File: `sklearn/utils/multiclass.py`, Line: 86

This line checks for mixed types in the input and raises a `ValueError`, which is consistent with the issue described.

================================== **Ai Message** ==================================

Tool Calls:

get\_files\_content (call\_Uiu8EXBVGJRyGPBnx7HJ5MbV)

Call ID: call\_Uiu8EXBVGJRyGPBnx7HJ5MbV

Args:

paths: ['sklearn/utils/multiclass.py']

line\_numbers: [86]

================================= **Tool Message** =================================

Name: get\_files\_content

Showing contents of File: /scikit-learn/sklearn/utils/multiclass.py starting from 86

86: raise ValueError("Mix type of y not allowed, got types %s" % ys\_types)

87:

88: label\_type = ys\_types.pop()

89:

90: # Check consistency for the indicator format

91: if (

92: label\_type == "multilabel-indicator"

93: and len(

94: set(

95: check\_array(y, accept\_sparse=["csr", "csc", "coo"]).shape[1] for y in ys

96: )

97: )

98: > 1

99: ):

100: raise ValueError(

101: "Multi-label binary indicator input with different numbers of labels"

102: )

103:

104: # Get the unique set of labels

105: \_unique\_labels = \_FN\_UNIQUE\_LABELS.get(label\_type, None)

106: if not \_unique\_labels:

107: raise ValueError("Unknown label type: %s" % repr(ys))

108:

109: if is\_array\_api:

110: # array\_api does not allow for mixed dtypes

111: unique\_ys = xp.concat([\_unique\_labels(y) for y in ys])

112: return xp.unique\_values(unique\_ys)

113:

114: ys\_labels = set(chain.from\_iterable((i for i in \_unique\_labels(y)) for y in ys))

115: # Check that we don't mix string type with number type

116: if len(set(isinstance(label, str) for label in ys\_labels)) > 1:

117: raise ValueError("Mix of label input types (string and number)")

118:

119: return xp.asarray(sorted(ys\_labels))

120:

121:

122: def \_is\_integral\_float(y):

123: return y.dtype.kind == "f" and np.all(y.astype(int) == y)

124:

125:

126: def is\_multilabel(y):

127: """Check if ``y`` is in a multilabel format.

128:

129: Parameters

130: ----------

131: y : ndarray of shape (n\_samples,)

132: Target values.

133:

134: Returns

135: -------

================================== **Ai Message** ==================================

diff --git a/sklearn/utils/multiclass.py b/sklearn/utils/multiclass.py

index 1234567..89abcde 100644

--- a/sklearn/utils/multiclass.py

+++ b/sklearn/utils/multiclass.py

@@ -86,6 +86,12 @@ def unique\_labels(\*ys):

raise ValueError("Mix type of y not allowed, got types %s" % ys\_types)

label\_type = ys\_types.pop()

+

+ # Handle nullable pandas dtypes

+ if label\_type == "unknown":

+ ys = [y.astype('float64') if y.dtype.name in ["Int64", "Float64"] else y.astype('bool') for y in ys]

+ return unique\_labels(\*ys)

+

# Check consistency for the indicator format

if (

label\_type == "multilabel-indicator"

================================== **Ai Message** ==================================

diff --git a/sklearn/utils/multiclass.py b/sklearn/utils/multiclass.py

index 1234567..89abcde 100644

--- a/sklearn/utils/multiclass.py

+++ b/sklearn/utils/multiclass.py

@@ -86,6 +86,12 @@ def unique\_labels(\*ys):

raise ValueError("Mix type of y not allowed, got types %s" % ys\_types)

label\_type = ys\_types.pop()

+

+ # Handle nullable pandas dtypes

+ if label\_type == "unknown":

+ ys = [y.astype('float64') if y.dtype.name in ["Int64", "Float64"] else y.astype('bool') for y in ys]

+ return unique\_labels(\*ys)

+

# Check consistency for the indicator format

if (

label\_type == "multilabel-indicator"

Executing command:

['cd test-repos/scikit-learn', 'git apply patch.diff --ignore-whitespace -v']

=============================== **Function Message** ===============================

Name: Patch\_Application\_Output

**Checking patch sklearn/utils/multiclass.py...**

**Applied patch sklearn/utils/multiclass.py cleanly.**

/home/ahsan/.local/lib/python3.10/site-packages/langchain\_community/tools/shell/tool.py:33: UserWarning: The shell tool has no safeguards by default. Use at your own risk.

warnings.warn(