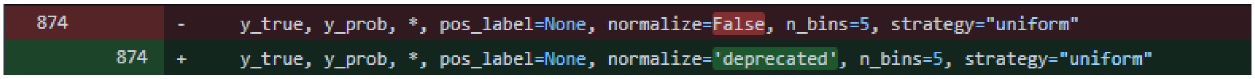
Issue #22482 Deprecate normalize parameter in calibration_curve

Link to Issue: https://github.com/scikit-learn/scikit-learn/issues/22482

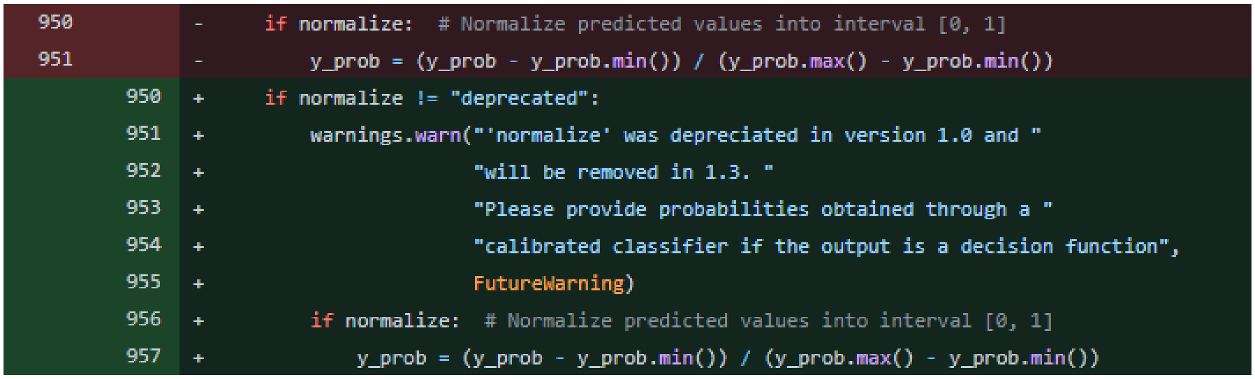
Summary: The method calibration_curve takes in parameters *y_true, y_prob, normalize*(default=False)*, n_bins*(default=5)*,* and *strategy*(default=’uniform’). The problem is, methodologically speaking, this method should only be used with probability input, meaning setting “normalize=True” is equivalent to having “a naive linear calibration with additional clipping for value above/under max/min.” which is unnecessary and usually not what people would want if they understand the implications. We want to warn the user with a deprecation warning when such a case arises so they wouldn’t unnecessarily normalize the input unless they still explicitly want to do so.

Implementation: Since the *naive linear calibration* ("*normalize* = True") is an unusual behavior and potentially dangerous, we changed the default value of *normalize* to "deprecated", as described should be done in contributing.rst, to warn the users that this parameter is being deprecated. As shown below,



```
-    y_true, y_prob, *, pos_label=None, normalize=False, n_bins=5, strategy="uniform"
+    y_true, y_prob, *, pos_label=None, normalize='deprecated', n_bins=5, strategy="uniform"
```

To alert the users, we added a FutureWarning message to warn the users when using *normalize*. Now, they will receive a deprecation warning message to alert them this parameter will be removed after two releases. This is to maintain backwards compatibility.



```
-    if normalize:  # Normalize predicted values into interval [0, 1]
-        y_prob = (y_prob - y_prob.min()) / (y_prob.max() - y_prob.min())
+    if normalize != "deprecated":
+        warnings.warn("'normalize' was depreciated in version 1.0 and "
+                      "will be removed in 1.3. "
+                      "Please provide probabilities obtained through a "
+                      "calibrated classifier if the output is a decision function",
+                      FutureWarning)
+        if normalize:  # Normalize predicted values into interval [0, 1]
+            y_prob = (y_prob - y_prob.min()) / (y_prob.max() - y_prob.min())
```

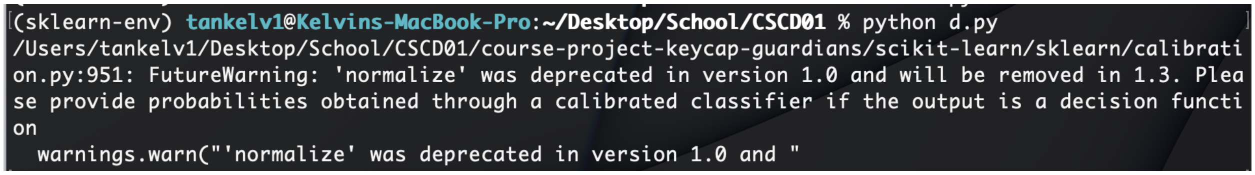
Acceptance Testing: When using the calibration_curves function, the “normalize” parameter should not be used. If it is used then a deprecation warning message should appear. To test for acceptance, call calibration_curves() with the “normalize” parameter. After taking out the “normalize” parameter the warning should go away.

Steps to reproduce:

1) Save the following code below into a file called d.py, notice normalize is set:

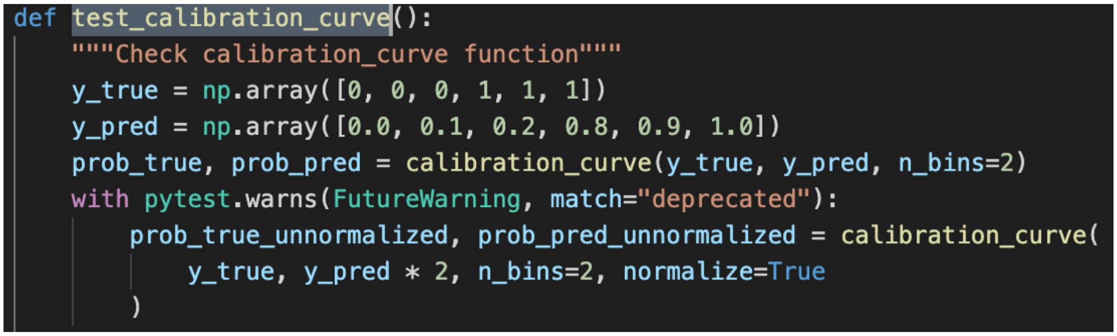
```
import numpy as np
from sklearn.calibration import calibration_curve
y_true = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1])
y_pred = np.array([0.1, 0.2, 0.3, 0.4, 0.65, 0.7, 0.8, 0.9,  1.])
prob_true, prob_pred = calibration_curve(y_true, y_pred, n_bins=3,
normalize=True)
```

2) Run d.py and see that a deprecation warning message appears.



```
(sklearn-env) tankelv1@Kelvins-MacBook-Pro:~/Desktop/School/CSCD01 % python d.py
/Users/tankelv1/Desktop/School/CSCD01/course-project-keycap-guardians/scikit-learn/sklearn/calibrati
on.py:951: FutureWarning: 'normalize' was deprecated in version 1.0 and will be removed in 1.3. Plea
se provide probabilities obtained through a calibrated classifier if the output is a decision functi
on
  warnings.warn("'normalize' was deprecated in version 1.0 and "
```

Unit Testing: We updated the test_calibration_curve() function in test_calibration.py to check that a depreciation warning was given when the “normalize” flag was set. Specifically, pytest.warns() will assert that FutureWarning is returned within the code block and that the warning message contains the string “deprecated”. To run the unit test use the following command: $ pytest -k test_calibration_curve



```python
def test_calibration_curve():
    """Check calibration_curve function"""
    y_true = np.array([0, 0, 0, 1, 1, 1])
    y_pred = np.array([0.0, 0.1, 0.2, 0.8, 0.9, 1.0])
    prob_true, prob_pred = calibration_curve(y_true, y_pred, n_bins=2)
    with pytest.warns(FutureWarning, match="deprecated"):
        prob_true_unnormalized, prob_pred_unnormalized = calibration_curve(
            y_true, y_pred * 2, n_bins=2, normalize=True
        )
```