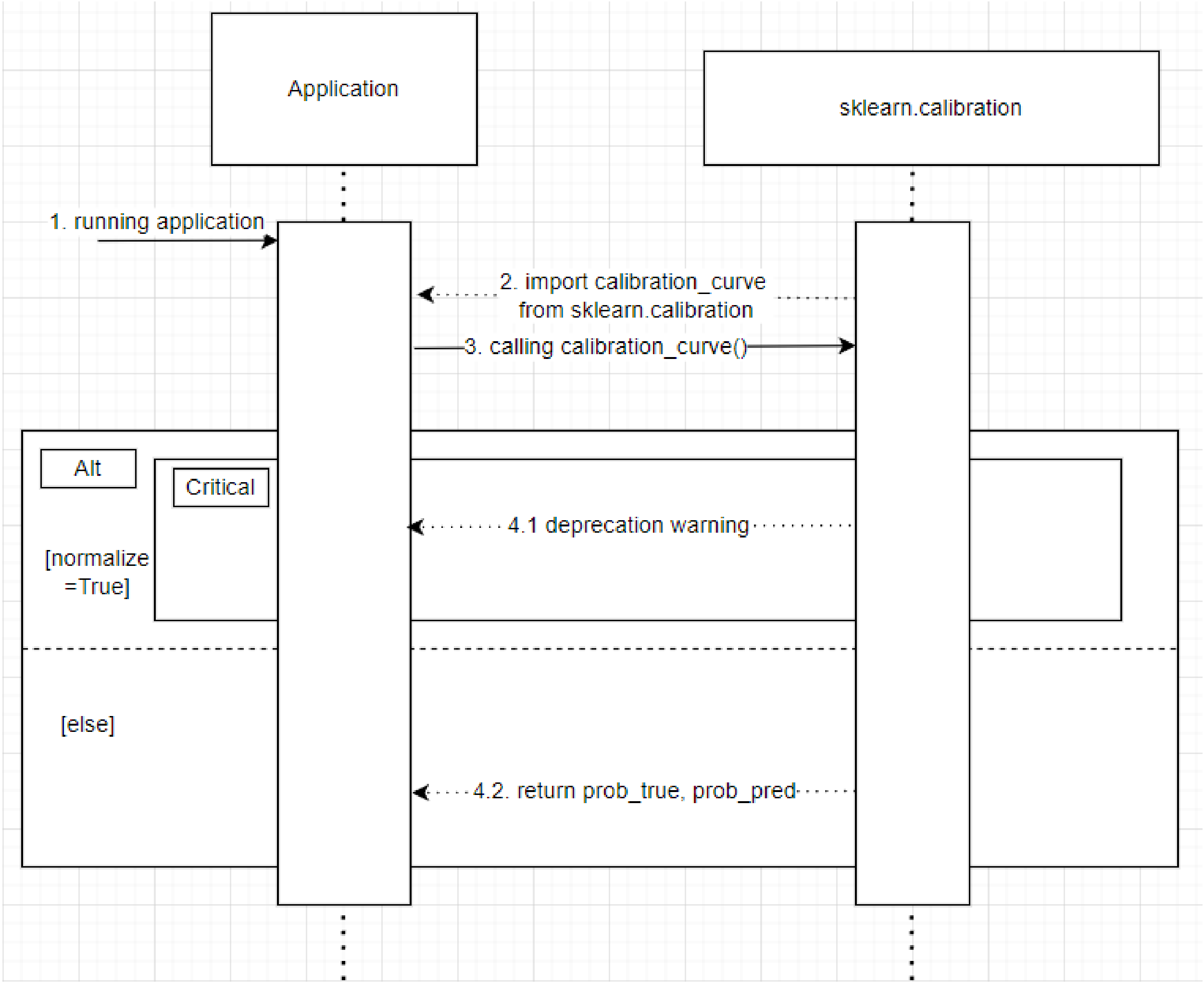
A3 Design Document

Keycap Guardians

Understanding the Issue

To understand how to approach the issue better, we came up with a sequence diagram detailing how our fix will be reflected when a client calls the calibration_curves function.


Application
sklearn.calibration
1. running application
2. import calibration_curve from sklearn.calibration
3. calling calibration_curve()
Alt
Critical
[normalize =True]
4.1 deprecation warning
[else]
4.2. return prob_true, prob_pred

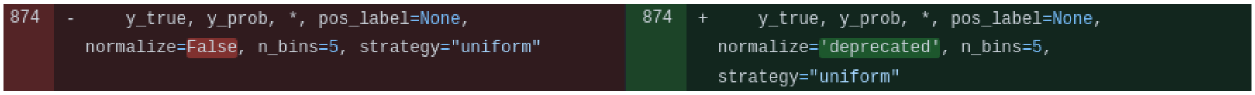
From this, we understood that when a client calls the calibration_curve function with the parameter “normalize=True”, Scikit-learn should return a deprecation warning to the user explaining they currently have set “normalize=True” and that is unusual behavior.

We also needed to understand how to properly add a deprecation warning formally that complies with Scikit-learn's policies, for this we investigated the official Scikit-learn guidelines for contribution.

Design Changes

File changed: *scikit-learn/sklearn/calibration.py*

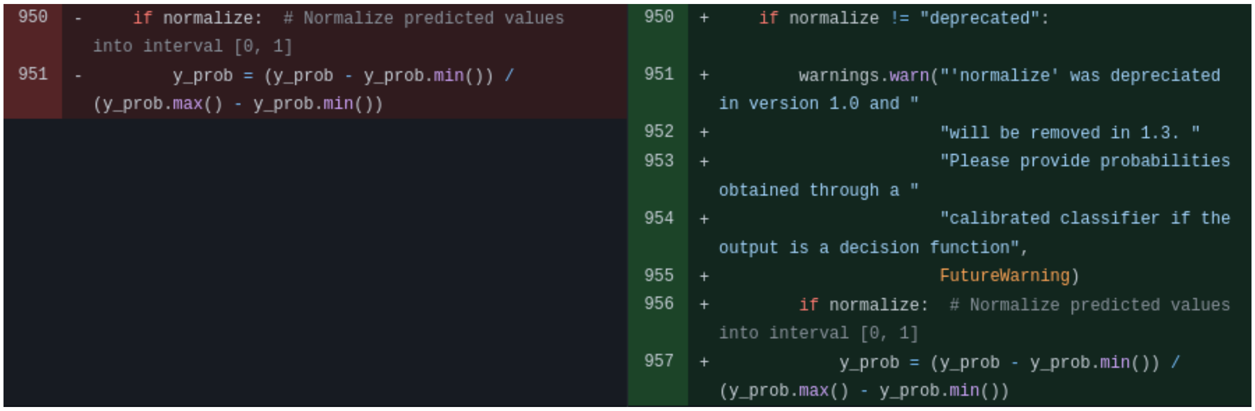
Below, we have changed the default value of *normalize* to "deprecated". This is to warn users that the usage of this method with *naive linear calibration ("normalize=True")* is potentially dangerous, and should be used explicitly.



```
-     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"
```

Thus, we created a warning message to alert the users of this fact. Now, if users try to use the method with *"normalize=True"*, they should receive a deprecation message.



```
-     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())
```